Dynamic stochastic block models, clustering and segmentation in dynamic graphs

Marco Corneli

Université Paris 1 Pahntéon-Sorbonne
Laboratoire SAMM

Thèse

présentée en première version en vu d'obtenir le grade de Docteur,
spécialité "Mathématiques Appliquées"

par

Marco Corneli

Dynamic stochastic block models, clustering and segmentation in dynamic graphs

Thèse soutenue le 17 Novembre 2017 devant le jury composé de :

M.	Christophe Biernacki	Université Lille 1 Sciences et Technologies	(Rapporteur)
M.	Thomas Brendan Murphy	University College Dublin	(Rapporteur)
M.	Allou-Badara Samé	IFSTTAR	(Examinateur)
Mme.	Clémence Magnien	Université Pierre et Marie Curie	(Examinatrice)
Mme.	Sophie Lèbre	Université Paul Valéry Montpellier 3	(Examinatrice)
M.	Fabrice Rossi	Université Paris 1 Panthéon-Sorbonne	(Directeur)
M.	Pierre Latouche	Université Paris 1 Panthéon-Sorbonne	(Co-encadrant)

Alla mia famiglia...

Remerciements

Je remercie Fabrice Rossi et Pierre Latouche pour m'avoir fait confiance et avoir encadré ce travail de thèse. Je les remercie également pour leur soutien et pour la liberté qu'ils m'ont donné. J'adresse aussi un grand remerciement à tous les membres du laboratoire SAMM pour ces trois ans passés ensemble dans une ambiance chaleureuse et très solidaire.

M. Corneli, Paris, le 20 octobre 2017.

Contents

List of Figures

Préface

Les graphes sont des structures mathématiques très adaptées pour modéliser les interactions parmi des objet/individus à étudier. De nombreux types de réseaux réels peuvent être modélisés à travers des graphes, tels que les réseaux de transport, les réseaux de transactions financières ou les réseaux sociaux comme Facebook ou Linkedin. Quand on observe un réseau d'interactions, le *temps* entre en jeu de deux manières différentes : on peut étudier les instants auxquels les interactions ont lieu et les durées de ces interactions. Les travaux de cette thèse se limitent à la première dimension temporelle. Chaque interaction est donc considérée comme instantanée pour des raisons de simplicité. L'évolution du réseau repose ainsi sur les temps des interactions uniquement. Dans ce contexte, les graphes peuvent être utilisés de deux manières différentes pour modéliser les réseaux :

1. **Temps discret.** Un réseau est observé à des instants différents et un graphe est associé à chacun de ces instants. Deux nœuds d'un graphe sont connectés si une ou plusieurs interactions entre eux sont observées dans le réseau à l'instant correspondant. Les interactions sont donc agrégées entre un instant d'observation et le suivant et les dates exactes des interactions sont perdues. Un réseau dynamique est enfin représenté par une séquence de graphes.

2. **Temps continu.** Plusieurs arcs connectent les nœuds d'un graphe. Chaque arc est donc uniquement associé à une paire de nœuds et à un instant temporel. Il n'y a pas d'agrégation temporelle dans ce cas et les instants exacts des interactions ne sont pas perdus. Le réseau dynamique est donc représenté par un seul graphe multiple dont les arcs sont étiquetés par les temps d'interaction.

Dans cette thèse ces deux visions sont adoptées alternativement. Nous proposons de nouvelles méthodes d'apprentissage non supervisé qui visent à partitionner les sommets d'un graphe dynamique en classes homogènes au sens où les sommets d'une même classe ont des profils d'interaction similaires. Pour éviter des problèmes d'identifiabilité les groupes de nœuds ne changent pas dans le temps. Par ailleurs, les approches proposées visent à détecter des changements structurels dans la façon dont les groupes de nœuds interagissent entre eux. Le point de départ de cette thèse est le stochastic block model (SBM), une approche probabiliste initialement utilisée en sciences sociales. Dans la version standard du modèle, les nœuds d'un graphe sont répartis dans des classes et la probabilité d'apparition d'un arc entre deux nœuds dépend uniquement des classes auxquelles ils appartiennent. Comme aucune hypothèse n'étant faite sur les probabilités d'interaction, SBM est un modèle très flexible qui permet

de capturer des structures topologiques différentes et variées (hubs, stars, communautés, etc.).

Tout en gardant une approache de modélisation par blocs (comme dans SBM) dans le contexte des graphes dynamiques, les principales contributions de cette thèse sont les suivantes :

1. Nous introduisons une nouvelle extension dynamique du SBM, appelée dSBM, qui utilise des processus de Poisson non homogènes pour modéliser les interactions parmi les paires de nœuds d'un graphe dynamique, en temps discret et continu. Les fonctions d'intensité des processus ne dépendent que des classes des nœuds comme dans SBM. De plus, ces fonctions d'intensité ont des propriétés de régularité sur des intervalles temporels qui sont à estimer, et à l'intérieur desquels les processus de Poisson redeviennent homogènes.

2. Un récent algorithme d'estimation pour SBM, qui repose sur la maximisation d'un critère exact (ICL exacte) est ici adopté pour estimer les paramètres de dSBM et sélectioner simultanément le modèle optimal. À notre connaissance, c'est la première fois que cet algorithme est utilisé dans le cadre d'un modèle SBM dynamique.

3. Un algorithme exact pour la détection de rupture dans les séries temporelles, la méthode « pruned exact linear time » (PELT), est étendu pour faire de la détection de rupture dans des données de graphe dynamique selon le modèle dSBM.

4. Le modèle dSBM est étendu ultérieurement pour faire de l'analyse de réseau *textuel* dynamique. Les réseaux sociaux sont un exemple de réseaux textuels : les acteurs s'échangent des documents (posts, tweets, etc.) dont le contenu textuel peut être utilisé pour faire de la classification et détecter la structure temporelle du graphe dynamique. Le modèle que nous introduisons est appelé « dynamic stochastic topic block model » (dSTBM).

Ce manuscrit est organisé de la façon suivante.

Dans le premier chapitre nous faisons état des principales notions de théorie des graphes et des propriétés connues des réseaux réels. Deux définitions formelles de graphe dynamique sont énoncées. Ensuite, nous présentons les principaux modèles génératifs existants pour les graphes (statiques et dynamiques) et les méthodes d'estimation introduites dans la littérature pour ces modèles. Enfin, nous introduisons des outils statistiques (pas forcement liés à l'analyse de réseau) qui sont à la base de nos travaux.

Dans le deuxième chapitre, deux versions du modèle dSBM sont présentées pour l'analyse des réseaux dynamiques en temps discret. Une procédure d'inférence est ensuite détaillée. Elle vise à maximiser (de façon gloutonne) la vraisemblance intégrée des données complétées : ceci permet d'estimer les paramètres du modèle tout en sélectionnant simultanément le nombre de classes.

Le troisième chapitre introduit une version du modèle dSBM pour l'analyse de graphes dynamiques en temps continu. La méthode proposée assure une forme de détection de rupture dans l'évolution temporelle de ce type de graphes. L'inférence repose sur une approche variationnelle classique dont une partie est basée sur le PELT.

Le quatrième chapitre revient sur les graphes dynamiques en temps discret. Les réseaux dynamiques textuels sont pris en compte, le modèle dSTBM est présenté et une procédure d'inférence est détaillée. Un critère de sélection de modèle est enfin formellement dérivé.

À la fin de chaque chapitre, nous conduisons des expériences sur des données simulées et réelles. Ces expériences nous servent à la fois à tester les points forts et les faiblesses de nos méthodes et à les comparer avec des approches concurrentes.

Cette thèse a fait l'objet de quatre articles dont trois d'ores et déjà publiés. Le deuxième chapitre du présent manuscrit traite des thèmes introduits en Corneli et al. (2016b;a). Le troisième chapitre correspond à Corneli et al. (2017) et le quatrième chapitre fait l'objet d'un papier qui à été rècemment soumis.

Abstract

Graphs are mathematical structures very suitable to model interactions between objects or actors of interest. Several real networks such as communication networks, financial transaction networks, mobile telephone networks and social networks (Facebook, Linkedin, etc.) can be modelled via graphs. When observing a network, the *time* variable comes into play in two different ways: we can study the time dates at which the interactions occur and/or the interaction time spans. This thesis only focuses on the *first* time dimension and each interaction is assumed to be instantaneous, for simplicity. Hence, the network evolution is given by the interaction time dates only. In this framework, graphs can be used in two different ways to model networks:

1. **Discrete time.** A network is observed at several times and a graph is associated with each observation time. Two vertices of a graph are connected if one or more interactions occurred between them in the corresponding time frame. Thus, interactions are aggregated between two consecutive observation times and the *exact* interaction dates are lost. In this context, a dynamic network is represented by a sequence of graphs.

2. **Continuous time.** Several edges are allowed to connect the vertices of a graph at different times. One edge is uniquely associated with a pair of nodes and a time point. No aggregation is required and interaction times are never lost. Therefore, a dynamic network is represented by a single multiple graph whose edges are labelled by the interaction times.

In this thesis both these perspectives are adopted, alternatively. We consider new unsupervised methods to cluster the vertices of a graph into groups of homogeneous connection profiles. In this manuscript, the node groups are assumed to be time invariant to avoid possible identifiability issues. Moreover, the approaches that we propose aim to detect structural changes in the way the node clusters interact with each other. The building block of this thesis is the stochastic block model (SBM), a probabilistic approach initially used in social sciences. The standard SBM assumes that the nodes of a graph belong to hidden (disjoint) clusters and that the probability of observing an edge between two nodes only depends on their clusters. Since no further assumption is made on the connection probabilities, SBM is a very flexible model able to detect different network topologies (hubs, stars, communities, etc.).

By adapting the block modelling perspective of SBM to dynamic graphs, the main contributions of this thesis are the following:

1. We introduce a new extension of SBM for dynamic graphs. The proposed approach, called dSBM, adopts non homogeneous Poisson processes to model the interaction times between pairs of nodes in dynamic graphs, either in discrete or continuous time. The intensity functions of the processes only depend on the node clusters, in a block modelling perspective. Moreover, all the intensity functions share some regularity properties on hidden time intervals that need to be estimated.

2. A recent estimation algorithm for SBM, based on the greedy maximization of an exact criterion (exact ICL) is adopted for inference and model selection in dSBM. To the best of our knowledge, this is the first time this algorithm is adopted for inference in dynamic stochastic block models.

3. An exact algorithm for change point detection in time series, the "pruned exact linear time" (PELT) method is extended to deal with dynamic graph data modelled via dSBM. The approach we propose can be used for change point analysis in graph data.

4. A further extension of dSBM is developed to analyse dynamic networks with *textual* edges (like social networks, for instance). In this context, the graph edges are associated with documents exchanged between the corresponding vertices. The textual content of the documents can provide additional information about the dynamic graph topological structure. The new model we propose is called "dynamic stochastic topic block model" (dSTBM).

This manuscript is organized as follows.

In the first chapter, we pass through the main notions of graph theory and review some stylized facts about real networks. Two formal definitions of dynamic graph are provided. Then, the main existing generative models for static and dynamic random graphs are presented along with their associated inference procedures. Finally, some statistical tools not necessarily related with network analysis are described in detail since they are used in later chapters.

In the second chapter, two versions of dSBM are introduced, both dealing with discrete time dynamic graphs. The corresponding inference procedure aims to maximize the complete data integrated log-likelihood, thus allowing us to learn the model parameters and select the number of clusters at the same time.

In the third chapter, we model continuous time dynamic graphs via dSBM and focus on clustering and change point analysis in graph data. A standard variational approach is adopted for the inference and one step of the estimation algorithm relies on the PELT method.

Finally, the fourth chapter introduces the dSTBM for discrete time dynamic graph with textual edges. The inference procedure is detailed and a model selection criterion is formally obtained.

The last part of each chapter is devoted to experiments on both simulated and real data. These experiments allow us to highlight the features of the proposed approaches and to compare them with alternative methods. Three papers were published during this thesis. The two works of

Corneli et al. (2016b;a) are discussed in detail in the second chapter of the present manuscript. The third chapter focuses on the topics detailed in Corneli et al. (2017) and the fourth chapter corresponds to a recently submitted paper.

Background

1

This preliminary chapter introduces the main concepts and existing works this thesis builds upon. In Section 1.1, we recall some definitions of graph theory and list some stylized facts about real networks. Two different definitions of dynamic graphs are provided, in continuous and discrete time, respectively. After reviewing some of the main existing methods to cluster vertices in static graphs, Section 1.2 focuses on generative models for random graphs. The stochastic block model (SBM) is a building block of this thesis, therefore it is discussed apart and treated in more detail. In the last part of the section, some existing probabilistic approaches for dynamic graph analysis are mentioned and briefly discussed. Section 1.3 details a variational expectation maximization (EM) procedure to perform inference in SBM and introduces the main existing model selection criteria to select the number of latent groups in mixture models. Finally, Section 1.4 is divided into three independent parts. The former introduces a stochastic process that will be employed in the following chapters to generalize SBM to dynamic graphs. The second part of this section recalls some results on change point analysis in time series. These results will be employed in Chapter 3. Finally, we introduce a probabilistic model for statistical analysis of documents that will be extended to the context of dynamic network analysis in Chapter 4.

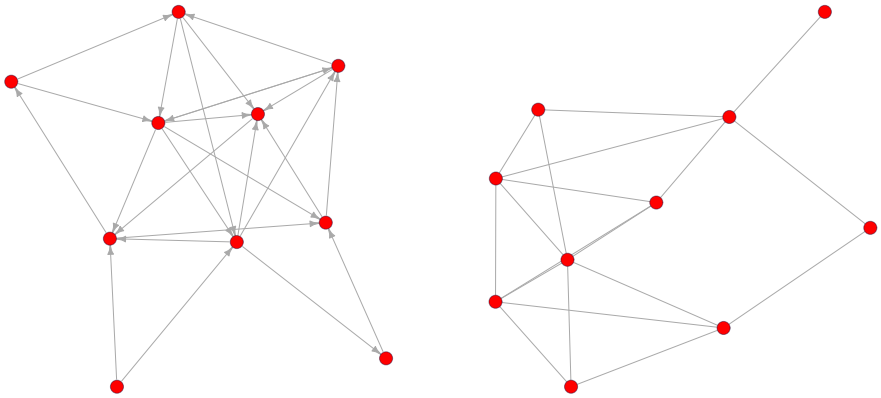
1.1 Graphs and data modelling

Roughly speaking graphs are mathematical structures used to model pairwise relations between objects. Since the pioneer work of Moreno (1934), graphs have been used to model phenomena of interest in many scientific fields. A non-exhaustive list of such fields includes physics (Albert and Barabási 2002), economics (Snyder and Kick 1979), biology (Barabási and Oltvai 2004, Palla et al. 2005) and history (Villa et al. 2008). This thesis focuses on applications in social sciences where graphs are used to model relational ties between actors (see e.g. Nowicki and Snijders 2001, Isella et al. 2011).

1.1.1 Basic graph theory and networks

We start by providing a formal definition of *graph*.

Definition 1.1 *A graph is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of N* vertices *(or* nodes*) and $\mathcal{E}$ is a set of* edges*. One edge connects two nodes and the graph is* directed *if connections between vertices are asymmetric and the pairs (i, j) are ordered. It is* undirected *if the pairs (i, j) are* not *ordered and hence interactions between vertices are symmetric.*



(a) *Directed graph.* **(b)** *Unirected graph.*

Figure 1.1 – *Two Graphs. The red points are the graph nodes and both directed links (Figure 1.1a) and undirected links (Figure 1.1b) are shown.*

A directed graph can be seen in Figure 1.1a and an undirected one in Figure 1.1b. Self loops are not considered in both figures, meaning that vertices are not allowed to interact with themselves. The edges of a graph can be weighted by a function $f : \mathcal{E} \to \mathbf{E}$, for any set $\mathbf{E}$.

Definition 1.2 *A graph is* binary *if $\mathbf{E} = \{0, 1\}$.*

When a graph is binary an interaction between two nodes i and j either occurs (an edges connects i and j) or does not (no edge connects the two nodes). In Chapters 2 and 3, we focus on *weighted* graphs (i.e. not binary) where $\mathbf{E} = \mathbb{N}^*$. In our case, an edge connecting two nodes is associated with the number of interactions that occurred between them in

a predefined time period, but interpretations are possible depending on the context. For instance, in the last chapter of this thesis a more complex framework is considered, in which edges are associated with textual contents.

A graph is entirely characterized by its *adjacency* matrix, defined in the following

Definition 1.3 *A $N \times N$ adjacency matrix X can be associated with a graph $\mathcal{G}$, such that the entry (i, j) of such matrix, denoted X_{ij}, is equal to* one *if an edge connects i to j,* zero *otherwise.*

Notice that the adjacency matrix of an undirected graph is symmetric whereas the one of a directed graph is not. An $N \times N$ *weight* matrix W can also be associated with a weighted graph, such that if $X_{ij} = 1$ then W_{ij} is equal to the weight associated with the edge connecting i and j. Conversely, when $X_{ij} = 0$, also $W_{ij} = 0$.

Remark 1.1 *In the reminder of this thesis, with a slight abuse of notation, we refer to W as to the adjacency matrix and denote it X.*

For instance, when $\mathbf{E} = \mathbb{N}^*$ and each edge is associated with the number of connections that occurred between the corresponding pair of nodes, the following notation is adopted

$$X_{ij} = \begin{cases} k \in \mathbb{N}^* & \text{if } k \text{ interactions occured between } i \text{ and } j \\ 0 & \text{if no interaction occured between } i \text{ and } j \end{cases}.$$

An important notion in graph theory is the one of *degree*.

Definition 1.4 *The* degree *of a node is the number of nodes to whom it is connected (a.k.a. its neighbours) whereas the* total degree *of a graph is $|\mathcal{E}|$, namely the total number of edges.*

The total degree of a graph as well as each node's degree can be inferred from the adjacency matrix. For instance, in undirected binary graphs without self loops the total degree is equal to the number of non-null entries in the adjacency matrix divided by two. Similarly, the degree of node i is equal to the sum of the elements on the i-th row (or column) of the adjacency matrix. The definition of path follows.

Definition 1.5 *A* path *from a vertex i to a vertex j is a sequence of edges in $\mathcal{E}$ starting at vertex i and ending at vertex j.*

If there exists at least one path between every pair of vertices then the graph is said to be *connected*.

As mentioned at the beginning of this section, graphs are used to model pairwise relations between actors. More specifically, actors are represented by the graph vertices and relations by the graph edges. The best suited kind of data that can be modelled by graphs is **network** data.

Remark 1.2 *In the reminder of this thesis, the word* network *is referred to the real object corresponding to the mathematical object called* graph.

For instance transportation networks, electric networks, social networks (Facebook, Twitter, etc.) are *not* graphs themselves and different modelling choices can lead to different graphs corresponding to the same network (this point is clarified in the following section).

The graphs associated with most real networks are known to share some properties (Albert et al. 1999, Barabasi and Albert 1999, Amaral et al. 2000, Newman 2003), for example

1. *Sparsity*: the total degree is linear in N.
2. *Heterogeneity*: very high degree for few vertices and low degree for most of them. In scale-free networks the node degrees follow a power law distribution.
3. *Assortative mixing* or *homophily*: a vertex tends to associate preferentially with other vertices that are similar to him in some way.
4. *Small-World Effect*: most pairs of vertices seem to be connected by a short path through the graph.
5. *Community structure*: this very important feature is discussed in detail in Section 1.2.

1.1.2 Dynamic graphs

Now, let us assume that some network data involving N actors is observed. The actors interacted with each other (possibly repeatedly) at specific times and these times are recorded. For instance, an interaction could correspond to a tweet or an e-mail sent from one actor to another at some time of the day. What we observe is an *interaction dataset* $\mathcal{D} = \{(i_m, j_m, t_m)\}_{1 \leq m \leq M}$, subset of $\{1, \dots, N\}^2 \times \mathbb{R}^+$, in which a triple (i, j, t) represents an interaction between actors i and j, at time t. The temporal period under study is the interval $[0, T]$ and t_{M+1} (i.e. the next interaction time) is not observed before T.

How can this interaction dataset be translated into a graph? Although the answer is not unique, this issue is addressed by Casteigts et al. (2012) in very general terms. They extended the definition of graph 1.1 to the following

Definition 1.6 *A dynamic graph is defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \phi, \zeta)$ where*

1. *The* lifetime *$\mathcal{T}$ is the period under study, in our case $[0, T]$.*
2. *$\phi : \mathcal{E} \times \mathcal{T} \to \{0, 1\}$ is called* presence *function and indicates whether a given edge is available at a given time.*
3. *$\zeta : \mathcal{E} \times \mathcal{T} \to \mathbb{R}^+$ called* latency *function, indicates the time it takes to cross a given edge if starting at a given date.*

Based on this definition we propose some changes to introduce two new definitions of *dynamic* graph that can be used to model the dataset $\mathcal{D}$. First of all, in this thesis the function $\zeta(\cdot)$ is neglected and all the interactions in $\mathcal{D}$ are assumed to be instantaneous.

Remark 1.3 *All the network analysis approaches described in the following make the implicit assumption that interaction time spans do not play a significant role in the general behaviour of the actors.*

As in Casteigts et al. (2012) two nodes i and j can be connected by multiple edges, each one associated with a different time point. Such a graph is called *multiple* graph and the presence function $\phi(\cdot)$ is null everywhere but at the interaction times.

Definition 1.7 **Continuous time dynamic graph.** *A continuous time dynamic graph is a multiple graph defined by* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \phi)$.

Notice that time is modelled as a continuous variable rather than a discrete one[1]. This definition of dynamic graph will be employed in Chapter 3, where the interaction times $t_1, \dots, t_M$ in the dataset $\mathcal{D}$ are seen as random variables generated by stochastic point processes (see Section 3.1.2).

Although the continuous time approach illustrated so far is completely general and has the advantage of preserving information (e.g. the exact order and times in which interactions occur), statistical models in dynamic network analysis usually are in *discrete* time. To illustrate this alternative approach, a time partition of the interval $[0, T]$ is introduced such that

$$[0, T] = \bigcup_{u=1}^{U} I_u, \qquad \exists U \in \mathbb{N}^* \tag{1.1}$$

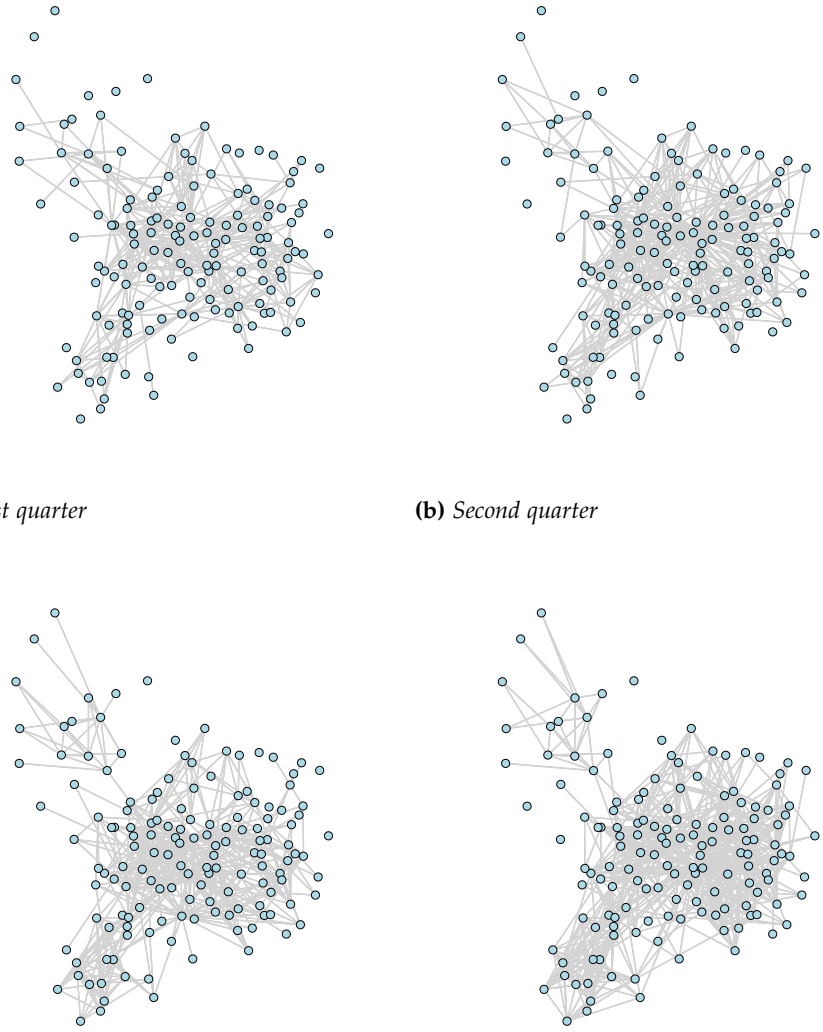
where $I_1, \dots, I_U$ are pairwise disjoint time sub-intervals not necessarily of the same size. For each pair of vertices (i, j) in $\mathcal{D}$ the corresponding edges can be aggregated on the time intervals of the above partition to obtain a sequence of U adjacency matrices. Hence, the entry (i, j) in the u-th matrix counts the number of interactions between the vertices i and j that occurred during I_u (see Remark 1.1). Alternatively, in a binary framework, the same entry is one if at least one interaction between i and j occurred over I_u, zero otherwise. As seen in the previous section, an adjacency matrix corresponds to a static graph as defined in 1.1. The dataset $\mathcal{D}$ is then modelled as a sequence of static (possibly weighted) graphs. Hence, the following definition can be provided

Definition 1.8 **Discrete time dynamic graph.** *In a discrete time framework, dynamic graph is synonym of sequence of static graphs (a.k.a. snapshots).*

Notice that the time partition introduced in (1.1) defines the finest level of information we have access to. The exact time at which an interaction occurred as well as the interaction orders inside each time interval of the partition are lost with aggregation. As we will see in the following chapters, a three dimensional tensor Y can be introduced to keep notations uncluttered. For instance, in the binary case $Y_{iju} = 1$ means that an interaction from node i to node j occurred during the time interval I_u.

Remark 1.4 *The tensor Y is the natural extension of the adjacency matrix X to the dynamic framework.*

[1] A similar framework for dynamic graphs was proposed by Guigourès et al. (2015).



(a) *First quarter* **(b)** *Second quarter*

(c) *Third quarter* **(d)** *Fourth quarter*

Figure 1.2 – *The Enron e-mail data set viewed as a dynamic graph. Each graph corresponds to a quarter of 2001.*

In Figure 1.2, the popular Enron communication network (`http://www.cs.cmu.edu/~./enron/`), containing all e-mail exchanges between 149 employees of the company is represented as a discrete time dynamic graph. Each snapshot corresponds to a quarter of the year 2001. The discrete time view described so far will play a central role in Chapters 2 and 4.

1.2 Generative models for random graphs

In a probabilistic perspective, the edges of a (static or dynamic) graph can be modelled as random variables. In this context, the graph is said to be *random*. This modelling choice allows us to answer to questions like: what is the probability that one edge occurs between two vertices? Which is the

expected number of edges between two vertices in a multiple graph? Etc. This section reviews the main existing generative models for static and dynamic random graphs.

1.2.1 Static graphs

Several existing methods in network analysis aim to detect *communities*.

Definition 1.9 *A community is a densely connected group of vertices having fewer connections outside the group.*

A graphical representation of a graph with three communities can be seen in Figure 1.3. Most of the existing approaches for community detec-

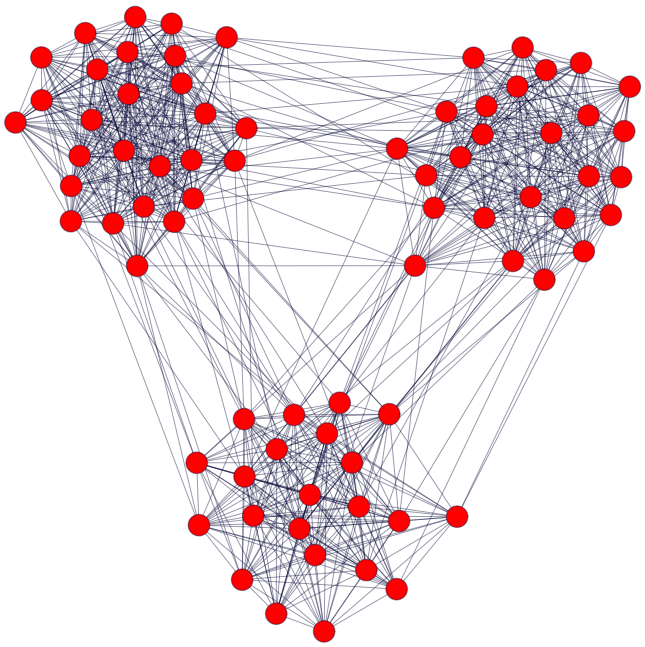


Figure 1.3 – *Example of an undirected network with three communities.*

tion either rely on the maximization of the *modularity* score (Newman and Girvan 2004) or the spectral properties of the graph Laplacian (this is the case of spectral clustering algorithms, see von Luxburg 2007, for instance). For a detailed survey of existing techniques for community detection the reader is referred to Fortunato (2010). These non-probabilistic techniques are popular in the literature and ad-hoc algorithms have been developed to reduce the computational burden and deal with large dataset (see e.g. Blondel et al. 2008, Noack and Rotta 2008). However, algorithms based on the modularity maximization have been proved to be biased, even asymptotically (Bickel and Chen 2009).

Another class of probabilistic approaches modelling graph edges as random variables proved to be flexible and capable of retrieving complex heterogeneous structures in networks (Airoldi et al. 2008, Goldenberg et al. 2009). In contrast, at the time of writing these approaches cannot compete with the ad-hoc algorithms mentioned above in terms of computational

complexity (however there are some recent very promising contributions in this direction, see e.g. Brault and Channarond 2016).

This thesis mainly focuses on a probabilistic approach known as stochastic block model (SBM, Holland et al. 1983, Wang and Wong 1987, Nowicki and Snijders 2001). Due to the importance of this generative model in the following chapters, we take some time to describe it in details.

Stochastic block model. SBM (Holland et al. 1983, Wang and Wong 1987, Nowicki and Snijders 2001) assumes that the vertices of a graph are partitioned into K hidden groups, denoted by $\mathcal{A}_1, \dots, \mathcal{A}_K$. Each node i is associated with a random variable (r.v.) Z_i, such that $Z_i = k$ if and only if $i \in \mathcal{A}_k$. The latent r.v. Z_i is assumed to follow a multinomial distribution

$$\mathbb{P}(Z_i = k) = \pi_k, \quad \forall k \in \{1, \dots, K\},$$

where the vector of cluster proportions is denoted by $\pi = (\pi_1, \dots, \pi_K)$ and

$$\sum_{k=1}^{K} \pi_k = 1.$$

We denote by $Z = \{Z_1, \dots, Z_N\}$ the set of all the latent variables Z_i. The equivalent 0-1 notation $Z_i = (Z_{i1}, \dots, Z_{iK})$, with $Z_{ik} = 1$ if node i belongs to the k-th cluster, 0 otherwise, will be used interchangeably when no confusion arises. In this case, Z is an $N \times K$ matrix whose i-th row is the vector Z_i. Recall that X denotes the adjacency matrix of the graph. This matrix entries are now assumed to be random variables. For instance, when dealing with binary graphs the random variable X_{ij} is equal to 1 if one edge connects node i to node j. Otherwise $X_{ij} = 0$. Conditionally on Z, X_{ij} is assumed to be drawn from a Bernoulli distribution

$$X_{ij} | Z_{ik} Z_{jg} = 1 \sim \mathcal{B}(X_{ij}; \theta_{kg}), \qquad \forall k, g \in \{1, \dots, K\}$$

whose parameter θ only depends on the groups of i and j, respectively. So, if vertex i belongs to cluster $\mathcal{A}_k$ and vertex j to cluster $\mathcal{A}_g$, the probability that one edge between them occurs is θ_{kg}. The last very important assumption in the SBM is that the entries of X are all independent random variables, conditionally on Z.

A variant of the SBM focuses on weighted graphs in which X_{ij} counts the number of interactions between i and j. In this context conditionally on Z to be known, X_{ij} follows a Poisson distribution

$$X_{ij} | Z_{ik} Z_{jg} = 1 \sim \mathcal{P}(X_{ij}; \lambda_{kg}), \qquad \forall k, g \in \{1, \dots, K\} \tag{1.2}$$

where

$$\mathcal{P}(X_{ij}; \lambda_{kg}) := \frac{\lambda_{kg}^{X_{ij}}}{X_{ij}!} \exp(-\lambda_{kg})$$

and Λ denotes a $K \times K$ matrix whose entry (k, g) is λ_{kg}, the expected number of interactions between any node in cluster k and any node in cluster g. A graphical representation of this model can be seen in Figure 1.4. Since no further constraints are imposed on the matrix Λ, SBM can de-

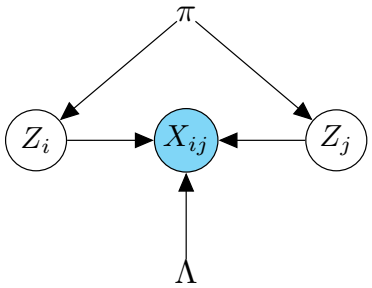


Figure 1.4 – *Graphical representation of SBM for weighted graphs with Poisson distributed weights. This representation enlightens the statistical dependence between the observed (blue circle) random variable X_{ij} and the hidden (white circle) random variables Z_i and Z_j. The model parameters Λ and π are not circled.*

tect communities of nodes but also more complicated structures like *hubs* which make networks locally dense (Daudin et al. 2008), *stars* (Latouche et al. 2011) and non-assortative structures (Corneli et al. 2016b). Several inference procedures have been developed for SBM aiming at estimating both the set Z and the number of components K. They are discussed in the next sections.

Extensions of SBM and other generative models. Several extensions of SBM have been proposed for static graphs, for instance:

1. The mixed membership stochastic block model (MMSBM) of Airoldi et al. (2008), which captures partial membership and allows each vertex to have a distribution over a set of classes, while SBM assumes that each vertex of a graph belongs to a single class.

2. The degree corrected SBM allows nodes in the same group to have different degrees, hence providing a more flexible model for real-world networks (Karrer and Newman 2011).

3. The random subgraph model (RSM, Jernite et al. 2014) is a generalization of SBM dealing with static weighted graphs. A partition of nodes (corresponding to subgraphs) is assumed to be available and node memberships to latent clusters vary from one subgraph to another. Notice that, when a single subgraph coinciding with the whole graph is provided a priori, RSM reduces to SBM.

Several probabilistic approaches for random graphs, alternative to SBM, were introduced in the literature. Two of them are reported without exhaustive intent.

1. The exponential random graph model (ERGM, Robins et al. 2007) employs a logistic regression to model the probability of interactions between vertices of a static graph. In this context, a set of user defined statistics are introduced to account for statistical dependence between edges.

2. The latent position model (LPM, Hoff et al. 2002) assumes that vertices in a graph have unobserved positions in a d-dimensional Euclidean latent space. Relying on a logistic regression, the probability of one interaction between two nodes depends on both network statistics and the nodes latent position. The basic idea is that the nearer two nodes are, the more likely they are to interact. The LPM was extended for community detection purposes by Handcock et al. (2007), where latent positions of nodes are assumed to follow multivariate Gaussian mixtures, whose parameters as well as proportions have to be estimated.

1.2.2 Dynamic graphs

Statistical models for dynamic graphs are usually discrete in time (see Section 1.1.2), i.e. predefined time intervals are considered and interactions during those time intervals are aggregated to obtain snapshots. Some of these models aim to extend SBM to the dynamic case. Yang et al. (2011) proposed a dynamic version of SBM allowing nodes to change cluster at time $t+1$ depending on their cluster at time t. The switching probabilities are all characterized by an homogeneous transition matrix. An alternative approach, relying on a non-homogeneous Markov chain, is proposed in Xu and Hero III (2013). Based on the central limit theorem, the sequence of connectivity matrices is seen as the hidden states sequence of a dynamic system, generating noisy, observed statistics. The authors rely on Kalman filtering along with the Rauch-Tung-Striebel smoother for inference. The work of Yang et al. (2011) was generalized by Matias and Miele (2017) to deal with more general types of edges. In their paper, they also showed that restrictions have to be imposed to the connectivity matrix in dynamic extensions of SBM in which both the connectivity parameters and the cluster memberships vary over time. These restrictions are needed in order to avoid identifiability issues. Other static variants of the SBM have also been adapted to the dynamic context. For instance, Xing et al. (2010), Ho et al. (2011) and Kim and Leskovec (2013) extended the MMSBM in order to look for overlapping clusters of nodes, through time.

Other existing approaches in discrete time dynamic network analysis are based on the generative models, other than SBM, mentioned in the previous section. For example, the dynamic random subgraph model (DRSM, Zreik et al. 2016) was built upon the RSM to uncover clusters within subgraphs provided a priori. Based on the exponential random graph model (ERGM), Hanneke et al. (2010) developed a more general class of models not limited to clustering purposes: the temporal exponential random graph model (TERGM). In this framework, the evolution of network snapshots is modelled through a Markov Chain whose transition probabilities depend on some user-defined functions accounting for sufficient statistics which usually involve the adjacency matrices at time

t and $t-1$. For inference, the authors rely on an approximated maximum likelihood approach involving both Gibbs sampling and a gradient descent algorithm. A similar view is adopted by Krivitsky and Handcock (2014) who introduced an hypothesis of separability (i.e. conditional independence) between appearing and disappearing connections between two consecutive snapshots of the dynamic graph. This assumption justifies the name STERGM (separable TERGM) and allows the model to gain in ease of specification and tractability. While in ERGM network statistics are defined globally (e.g. density, stability, reciprocity, etc.), they are built at the actors level (e.g. degree, past behaviour, etc.) in stochastic actor-oriented models for network change (SAOM, Snijders 1996). In SAOM, a dynamic graph undergoes instantaneous changes involving a single edge (appearing or disappearing) at a time. The waiting time between two change opportunities is an exponential r.v. and when a node i has the opportunity to change connection, the probability that it connects to j is computed via a logit regression involving several statistics. Inference is based on method of moments. Finally, the latent position model (LPM) was also extended by Sarkar and Moore (2005), Friel et al. (2016), Sewell and Chen (2015; 2016) to deal with dynamic binary or weighted graphs. In a recent work, Durante et al. (2016) allow the node coordinates in latent space to evolve in continuous time via nested Gaussian processes to account for non stationarity in real networks.

Although very popular, the discrete dynamic network models described so far share a common drawback: aggregating data leads to a loss of information and the choice of the time intervals used to build the snapshots has a strong impact on the inference results (Matias et al. 2015). As mentioned in Section 1.1.2, in order to deal with dynamic interactions on a continuous time frame a natural choice is to consider point processes. Thus, Matias et al. (2015) relied on the so called (doubly stochastic) non homogenous Poisson processes (NHPP, see Section 1.4.1). Following a SBM like approach, nodes are assumed to belong to hidden clusters. Each pair of nodes is then associated to a NHPP whose intensity function depends on the respective clusters. A variational expectation maximization (VEM, see Section 1.3.1) algorithm is finally employed to estimate these functions non-parametrically and to uncover the clusters. This work is partially related to Dubois et al. (2013) who relied on a parametric form for the intensity functions, which depend on the past network history and other predefined statistics. An alternative approach is detailed in Chapter 2 of this thesis, where the intensity functions of the Poisson processes are assumed to be piecewise constant on predefined time intervals, each time interval belonging to a hidden time cluster. In this model, the value of each intensity function at time t not only depends on the clusters of nodes, but also on the corresponding time cluster.

Not relying on Poisson processes, but still in continuous time Guigourès et al. (2012; 2015) proposed a different model. Based on the non-parametric MODL approach, (Boullé 2010), the triclustering technique of Guigourès et al. (2012; 2015) aims to simultaneously uncover groups of nodes and time segments characterized by a stationary edges distribution. Although powerful and flexible, this technique is somewhat blind to some intensity changes: time segments cannot be detected when

Model/Paper	Time D	Time C	Reference Model
Xu and Hero III (2013)	✓		SBM
Yang et al. (2011)	✓		SBM
Matias and Miele (2017)	✓		SBM
Xing et al. (2010)	✓		MMSBM
Ho et al. (2011)	✓		MMSBM
Kim and Leskovec (2013)	✓		MMSBM
Zreik et al. (DRSM, 2016)	✓		RSM
Hanneke et al. (TERGM, 2010)	✓		ERGM
Krivitsky and Handcock (STERGM, 2014)	✓		ERGM
Sarkar and Moore (2005)	✓		LPM
Friel et al. (2016)	✓		LPM
Sewell and Chen (2015)	✓		LPM
Sewell and Chen (2016)	✓		LPM
Durante et al. (2016)	✓		LPM
Matias et al. (2015)		✓	SBM
Dubois et al. (2013)		✓	ERGM, SBM
Guigourès et al. (2012; 2015)		✓	MODL

Table 1.1 – *Summary of the models for dynamic graph modelling mentioned in Section 1.2.2.*

the type of connectivity structure is persistent through time but subject to parallel shifts in the interaction intensity levels. This point will be discussed in more detail in Section 3.3.1, when one of the models that we developed is compared with MODL.

We finally cite some recent works that could be extended to model interactions between vertices in dynamic graphs relying on Hawkes point processes (Hawkes 1971). These processes take into account mutual dependence between pairs of nodes and self-exciting dynamics. Several approaches have been used to perform inference, both in parametric and non parametric frameworks. For instance, Xu et al. (2016) rely on maximum likelihood whereas Achab et al. (2016) developed a moment matching method. Although Hawkes processes are a natural extension of Poisson processes, they are less easily interpretable and their use implies an important additional effort during the inference step.

Table 1.1 summarizes the main features of the approaches mentioned so far.

1.3 Inference in stochastic block model

As said in the previous section, SBM is the generative model we rely on in the following chapters. It is then crucial to review the inference techniques introduced in the literature for this model. In the following, we consider weighted graphs in which the entries of the adjacency matrix X count the number of interactions between the corresponding pairs of nodes (see

Remark 1.1). Hence, (1.2) can be rewritten as

$$p(X_{ij}|Z_i, Z_j, \Lambda) = \mathcal{P}(X_{ij}; \lambda_{Z_i Z_j}) = \prod_{k,g}^{K} \left(\mathcal{P}(X_{ij}; \lambda_{kg})\right)^{Z_{ik} Z_{ig}}.$$

Recalling that Z_i and Z_j are independent and follow a multinomial distribution of parameter π, the following joint probability distribution is obtained

$$p(X_{ij}, Z_i, Z_j|\Lambda, \pi) = \prod_{k,g}^{K} \left(\mathcal{P}(X_{ij}; \lambda_{kg})\pi_k \pi_g\right)^{Z_{ik} Z_{jg}} \tag{1.3}$$

and summing over all possible values of Z_i and Z_j leads to the marginal probability distribution

$$p(X_{ij}|\Lambda, \pi) = \sum_{k,g}^{K} \mathcal{P}(X_{ij}; \lambda_{kg})\pi_k \pi_g.$$

As it can be seen, the number of edges between a pair of nodes (i, j) follows a mixture of Poisson distributions. It is well known that the standard approach to obtain maximum-likelihood (ML) and maximum a posteriori (MAP) estimates in mixture models is the expectation maximization (EM) algorithm (A. P. Dempster 1977). However, due to the graphical structure of SBM, the posterior probability of Z given X, Λ and π is not tractable and the EM algorithm cannot be used to estimate the model parameters. To tackle this issue, many inference procedures have been introduced in the literature such as variational EM (VEM, Daudin et al. 2008), variational Bayes EM (VBEM, Latouche et al. 2012), Gibbs sampling (Nowicki and Snijders 2001), allocation sampler (Mc Daid et al. 2013) and greedy search (Côme and Latouche 2015).

Since the last two chapters of this thesis rely on the VEM algorithm for the inference, we describe here its main features.

1.3.1 Variational decomposition and EM algorithm

In this section, the number of clusters K is assumed to be known. This hypothesis will be relaxed in the following section. Recalling that X is a $N \times N$ adjacency matrix and Z an $N \times K$ cluster matrix (in 0-1 notations), the complete data log-likelihood for SBM can be explicitly obtained

$$\log p(X, Z|\Lambda, \pi) = \sum_{i=1}^{N} \sum_{j \neq i}^{N} \log p(X_{ij}, Z_i, Z_j|\Lambda, \pi), \tag{1.4}$$

where

1. the probability distributions on the right hand side of the equality are detailed in (1.3) and
2. we used the conditional independence of $\{X_{ij}\}_{i,j}$ given Z.

Maximizing this log-likelihood with respect to Λ and π is feasible but would require the knowledge of Z.

Let us now denote by $q(\cdot)$ a generic probability distribution on the matrix Z. In the following, $\mathbf{E}_q$ denotes the expectation taken with respect to the probability distribution $q(\cdot)$ and $\mathbf{z}$ denotes an outcome of the random matrix Z. It can be proven that the *observed* data log-likelihood can be decomposed as (Neal and Hinton 1998)

$$\log p(X|\Lambda, \pi) = \mathcal{L}(q(\cdot); \Lambda, \pi) + \mathrm{KL}(q(\cdot)||p(\cdot|X, \Lambda, \pi)), \tag{1.5}$$

where

$$\begin{aligned} \mathcal{L}(q(\cdot); \Lambda, \pi) :&= \mathbf{E}_q \left[\log \left(\frac{p(X, Z|\Lambda, \pi}{q(Z)} \right) \right] \\ &= \sum_{\mathbf{z}} q(\mathbf{z}) \log \left(\frac{p(X, \mathbf{z}|\Lambda, \pi}{q(\mathbf{z})} \right), \end{aligned} \tag{1.6}$$

where the sum is taken over all the possible outcomes of Z and KL denotes the Kullback-Leibler divergence between $q(\cdot)$ and $p(\cdot|X, \Lambda, \pi)$

$$\begin{aligned} \mathrm{KL}(q(\cdot)||p(\cdot|X, \Lambda, \pi)) :&= -\mathbf{E}_q \left[\log \left(\frac{p(Z|X, \Lambda, \pi}{q(Z)} \right) \right] \\ &= -\sum_{\mathbf{z}} q(\mathbf{z}) \log \left(\frac{p(\mathbf{z}|X, \Lambda, \pi}{q(\mathbf{z})} \right). \end{aligned} \tag{1.7}$$

Since the Kullback-Leilbler divergence is non negative and null if and only if $q(\cdot) = p(\cdot|X, \Lambda, \pi)$, it is clear that

1. the functional (1.6) is a *lower bound* for $\log p(X|\Lambda, \pi)$ and
2. the following equality holds

$$\log p(X|\Lambda, \pi) = \mathcal{L}(p_{Z|X}(\cdot); \Lambda, \pi), \qquad \forall \Lambda, \pi$$

where $p_{Z|X}(\cdot)$ is a shorthand notation for the posterior distribution of Z given X, Λ and π.

The distribution $p_{Z|X}(\cdot)$, however, is not tractable in SBM (see Daudin et al. 2008, for details) and we can only minimize the KL divergence with respect to a tractable family of distributions $q(\cdot)$. The approximating distributions are assumed to be of the form

$$q(Z) = \prod_{i=1}^{N} q(Z_i) = \prod_{i=1}^{N} \prod_{k=1}^{K} \tau_{ik}^{Z_{ik}}, \tag{1.8}$$

where the τ_{ik}s are positive and $\sum_{k=1}^{K} \tau_{ik} = 1$, for all i. This hypothesis can be rephrased by saying that under the $q(\cdot)$ distribution, node to cluster assignments are independent given the adjacency matrix. Hence, the VEM algorithm consists in alternatively maximizing the lower bound (1.6) with respect to the probability distribution $q(\cdot)$ in the above equation and the model parameters (Λ, π) up to convergence. Two important works of Celisse et al. (2012) and Bickel et al. (2013) proved the consistency and asymptotic normality of maximum-likelihood and variational estimators obtained using the mean field variational approximation (1.8) for standard SBM.

1.3.2 Model Selection

So far, the number of groups K was assumed to be known but in real applications this hypothesis is often too simplistic. Two popular model selection criteria, the Akaike information criterion (AIC, Akaike 1974) and the Bayesian information criterion (BIC, Schwarz 1978), rely on asymptotic approximations of the observed data integrated log-likelihood to select the number of groups in mixture models. However, they have both been shown to be prone to over fitting in particular circumstances (see for example Celeux and Soromenho 1996, Biernacki et al. 2000) and, more important, the observed data log-likelihood is not tractable in SBM. Hence neither AIC nor BIC can be computed for SBM.

A third model selection criterion, the integrated classification likelihood (ICL, Biernacki et al. 2000) plays a crucial role in the next chapters and it is illustrated in the following in the context of SBM with Poisson distributed weights. For a more general and detailed survey about model selection criteria the reader is referred to Claeskens and Hjort (2008).

1.3.3 Integrated classification likelihood (ICL)

This criterion was developed as a model selection criterion for Gaussian mixtures and it focuses on retrieving relevant clustering rather than density estimation as in BIC. If BIC relies on an asymptotic approximation of the *observed* data integrated log-likelihood, ICL is based on an asymptotic approximation of the *complete* data integrated log-likelihood. In the SBM case

$$p(X, Z|K) = \int_{\Lambda,\pi} p(X, Z, |\Lambda, \pi) p(\Lambda, \pi|K) d\Lambda d\pi, \tag{1.9}$$

where $p(\Lambda, \pi|K)$ is any prior distribution over the pair (Λ, π) conditional on K. To keep notations uncluttered, conditioning on K is omitted henceforth, nonetheless the following equations are conditioned on K. Making the further assumption that $p(\Lambda, \pi)$ factorizes over Λ and π, the following holds

$$\begin{aligned} p(X, Z) &= \int_{\Lambda,\pi} p(X|Z, \Lambda) p(Z|\pi) p(\Lambda) p(\pi) d\Lambda d\pi \\ &= \int_{\Lambda} p(X|Z, \Lambda) p(\Lambda) d\Lambda \int_{\pi} p(Z|\pi) p(\pi) d\pi \\ &= p(X|Z) p(Z), \end{aligned} \tag{1.10}$$

and hence

$$\log p(X, Z) = \log p(X|Z) + \log p(Z).$$

In order to approximate the two log-probabilities on the right hand side of the equality, the following strategy can be adopted

1. A Laplace approximation is employed to approximate the first term in an analogous manner to the standard derivation of BIC

$$\log p(X|Z) \approx \max_{\Lambda} \log p(X|Z, \Lambda) - \frac{K^2}{2} \log N(N-1),$$

where K^2 is the dimension of Λ and $N(N-1)$ the number of observations, namely the entries of the adjacency matrix X in a directed graph without self loops.

2. A Jeffreys' non informative prior is used to obtain a formulation of $p(Z)$ and an approximation of the Gamma function through the Stirling formulas (see Biernacki et al. 2000, for details) finally leads to

$$\log p(Z) \approx \max_{\pi} \log p(Z|\pi) - \frac{K-1}{2}\log(N).$$

Notice that this last approximation would be the same for any mixture model with N observations and Z following a multinomial distribution of parameter π. The ICL criterion for SBM can finally be formulated as

$$ICL(K) = \max_{\Lambda,\pi} \log p(X, Z|\Lambda, \pi) - \frac{K^2}{2}\log N(N-1) - \frac{K-1}{2}\log N. \quad (1.11)$$

The above criterion is obtained for standard SBM in Daudin et al. (2008). In their case, the first penalty term in the above equation looks slightly different since they consider undirected graphs.

The above criterion can be computed for several values of K and the the number of groups leading to the highest ICL is finally retained. We recall that Z is not observed and it needs to be estimated in order to use the ICL. A possible approach consists into applying the VEM algorithm to the data (for a fixed K) and replace Z by its MAP estimates according to $q(\cdot)$ in (1.8).

The ICL criterion plays a fundamental role in the following chapters, since it is used for model selection in Chapter 4, in the context of a dynamic extension of SBM. For the same model, in Chapter 2, an exact version of this criterion is formally obtained and maximized relying on a Bayesian approach.

1.4 Other important statistical tools

The three independent topics discussed in this section are employed in the reminder of this thesis and although not necessarily relating with network analysis, they are illustrated in some details.

1.4.1 Non homogeneous Poisson process

This stochastic *point* process is used in the following to model interactions between a pair of nodes in dynamic graphs. The reader in referred to Thompson (1988), Kallenberg (2006) for a formal definition of point process and a general treatment of point processes theory, whose deeper understanding is outside the scope of the present thesis. However, when the Poisson point process is defined on the real line (as it is the case in this thesis), it is possible to adopt the useful interpretation of *counting* process which simplifies the exposition. This view is adopted for instance in Norris (1998) to define the well known homogeneous Poisson process. In the reminder of this manuscript, with a slight abuse of notation, when writing "non homogeneous Poisson process" we refer to the *counting* counterpart of the non homogeneous Possion point process.

Definition 1.10 *Let $\{M(t)\}_{t\geq 0}$ be an increasing, right continuous integer-valued process starting from 0. Let $\lambda(\cdot)$ be a strictly positive integrable function. Then $\{M(t)\}_{t\geq 0}$*

is a non homogeneous Poisson process (NHPP) if it has independent increments and for all $s \leq t$

$$M(t) - M(s) \sim \mathcal{P}\left(\int_s^t \lambda(u)du\right). \tag{1.12}$$

Obviously, when $\lambda(\cdot)$ is a constant function, $\{M(t)\}_{t\geq 0}$ reduces to a homogeneous Poisson process whose increments are *stationary*, which is not the case for the NHPP.

Consider the following sequence of random *arrival* times

$$0 < \nu_1 < \nu_2 < \cdots < \nu_m < \ldots, \tag{1.13}$$

counted by $\{M(t)\}_{t\geq 0}$. The following two events are the same one

$$\{M(t) = m\} \qquad \text{iff} \qquad \{\nu_m \leq t < \nu_{m+1}\}.$$

In the following chapters, arrival times will model the interaction times between nodes of a graph. However, when dealing with discrete time dynamic graphs (Definition 1.8) the observed interaction times are in some way "neglected", since we are only interested in their number in a certain time frame. Conversely, when modelling dynamic graphs in continuous time (Definition 1.7), the interaction times will be fully part of the inference procedure. In this last case, the following proposition will play a crucial role.

Proposition 1.1 *Consider the following ordered time points*

$$0 = t_0 < t_1 < \cdots < t_m < T.$$

The event of observing the first m arrival times in (1.13) *at* $t_1, \ldots, t_m$ and *not observing* ν_{m+1} *before T has the following likelihood*

$$p_{\nu,m}(t_1, \ldots, t_m)\mathbb{P}(\nu_{m+1} \geq T | \nu_m = t_m) = \exp\left(-\int_0^T \lambda(u)du\right)\prod_{j=1}^m \lambda(t_j), \tag{1.14}$$

where $p_{\nu,m}(\cdot)$ *is a shorthand notation for the joint density of the first m arrival times* $\nu_1, \ldots, \nu_m$.

Proof. First of all, notice that the likelihood on the left hand side of the above equation includes the additional term $\mathbb{P}(\nu_{m+1} \geq T | \nu_m = t_m)$ to account for the incomplete knowledge of the event ν_{m+1}, which is only known to happen after T. This phenomenon is called *right censoring* and is very common in survival analysis (see e.g. Zhou 2015).

We now condition on the event $\{\nu_j = t_j\}$ for $j \in \{1, \ldots, m-1\}$. The following equality holds

$$\begin{aligned}
\mathbb{P}(\nu_{j+1} > t | \nu_j = t_j) &= \mathbb{P}\left(M(t) - M(t_j) = 0 | \nu_j = t_j\right) \\
&= \mathbb{P}\left(M(t) - M(t_j) = 0\right) \\
&= \exp\left(-\int_{t_j}^t \lambda(u)du\right),
\end{aligned}$$

for all $t \geq t_j$, due to 1.12. By taking the first derivative with respect to t of the above probability and changing the sign, the following conditional

probability density function is obtained for ν_{j+1}

$$f_{\nu_{j+1}}(t|\nu_j = t_j) = \lambda(t) \exp\left(-\int_{t_j}^{t} \lambda(u)du\right).$$

Notice that, due to the incremental independence of the NHPP, the arrival times previous to ν_j can be neglected. Hence the following equalities hold

$$\begin{aligned} p_{\nu,m}(t_1, \ldots, t_m) &= \prod_{j=1}^{m} f_{\nu_j}(t_j|\nu_{j-1} = t_{j-1}) = \prod_{j=1}^{m} \left(\lambda(t_j) \exp\left(-\int_{t_{j-1}}^{t_j} \lambda(u)du\right)\right) \\ &= \exp\left(-\int_{0}^{t_m} \lambda(u)du\right) \prod_{j=1}^{m} \lambda(t_j), \end{aligned}$$

where we used $t_0 = 0$ and the proof is concluded by observing that

$$\mathbb{P}(\nu_{m+1} > T|\nu_m = t_m) = \exp\left(-\int_{t_m}^{T} \lambda(u)du\right)$$

□

1.4.2 Multiple change point detection in univariate time series.

In the time series literature, change point analysis is a central and widely studied topic. An exhaustive review of this field is outside of the scope of this thesis. However, in Chapter 3, an existing algorithm for multiple change point detection in time series (PELT, Killick et al. 2012) is adapted to deal with graph data. Hence, this section introduces some basics in change point analysis and details the PELT algorithm.

An ordered sequence of real data $x_1, \ldots, x_N$ is assumed to be observed. The generative model for such data includes $D-1$ ordered change points $\eta_1 < \eta_2 < \cdots < \eta_{D-1}$, each one being a natural number between 1 and $N-1$. Roughly speaking, when a change point occurs the statistical properties of the data change. More specifically, the change points define D segments $]\eta_{d-1}, \eta_d]$ and

$$p(x_i) = g(x_i; \theta_d) \qquad \text{if } x_i \in]\eta_{d-1}, \eta_d], \forall i \tag{1.15}$$

where $g(\cdot)$ is a density function depending on a parameter θ varying between segments. The data points are all assumed to be independent. In contrast they are identically distributed *only* on each time segment. The generative model outlined so far can be used to estimate the number $D-1$ of change points and their locations. Formally, by adopting the convention $\eta_0 = 0$ and $\eta_D = N$ the following minimization problem can be stated

$$\min_{\eta_1, \ldots, \eta_{D-1}, D} \left\{ \sum_{d=1}^{D} \left[\mathcal{C}(x_{\eta_{d-1}+1}, \ldots, x_{\eta_d})\right] + h(D)\alpha \right\}, \tag{1.16}$$

where $\mathcal{C}(\cdot)$ is a *cost* function associated with the observations on segment $]\eta_{d-1}, \eta_d]$, $h(D)$ accounts for the number of free parameters in the model and α is a constant depending on the number of observations only. Henceforth, the function $h(\cdot)$ is assumed to be linear in D

$$h(D) := kD, \qquad \exists k \in \mathbb{N}^* \tag{1.17}$$

which is a common assumption in the literature. Similarly, two standard choices for the constant α are $\alpha = N$ (AIC penalty) or $\alpha = \log(N)/2$ (BIC penalty), but other penalties can be considered (see e.g. Picard et al. 2005, Eckley et al. 2011). Although several choices are possible for the cost function $\mathcal{C}(\cdot)$, in this thesis we only focus on the negative log-likelihood

$$\begin{aligned}\mathcal{C}(x_{\eta_{d-1}+1}, \dots, x_{\eta_d}) :&= -\max_{\theta_d} \log p(x_{\eta_{d-1}+1}, \dots, x_{\eta_d}|\theta_d) \\ &= -\max_{\theta_d} \sum_{i=\eta_{d-1}+1}^{\eta_d} \log p(x_i|\theta_d),\end{aligned} \tag{1.18}$$

where the last equality comes from the independence assumption formulated above.

Notice that in real applications, the number of change points is unknown and has to be estimated. Hence, the minimization problem (1.16) involves both $\eta_1, \dots, \eta_{D-1}$ (change point locations) and D (the number of change points).

Binary Segmentation. Several approaches have been proposed to solve (1.16). One of the most popular is Binary Segmentation (A. J. Scott 1974), an *approximate* method relying on the "divide and conquer" paradigm. This method begins by looking for a single change point on the entire time series. This means that it looks for η such that

$$\mathcal{C}(x_1, \dots, x_\eta) + \mathcal{C}(x_{\eta+1}, \dots, x_N) + 2k\alpha < \mathcal{C}(x_1, \dots, x_N) + k\alpha.$$

If no change point is detected the method stops. If a change point is detected it splits the dataset into two segments and the method looks for a single change point on each segment and so on until no further change points are detected. Binary Segmentation is an *approximate* method since it does not explore all the possible combinations of $\eta_1, \dots, \eta_{D-1}$ for all D smaller equal than N. However, it has the advantage of being computationally efficient, resulting in an $\mathcal{O}(N \log N)$ calculations to detect both the number of change points and their location.

Optimal Partitioning. A very popular *exact* method, exploring the whole segmentation space to provide estimates of both $\{\eta_1, \dots, \eta_{D-1}\}$ and D, in an $\mathcal{O}(N^2)$ calculation, was proposed by Jackson et al. (2005). This method (a.k.a. "Optimal Partitioning") relies on *dynamic programming* (Bellman 1954) and employs the *value* function $F(\cdot)$

$$F(N, D) := \min_{\eta_1, \dots, \eta_{D-1}} \left\{ \sum_{d=1}^{D} [\mathcal{C}(x_{\eta_{d-1}+1}, \dots, x_{\eta_d}) + k\alpha] \right\}, \tag{1.19}$$

where we made use of (1.17). Therefore, $F(\cdot)$ keeps track of the lowest value of (1.16) attainable by segmenting N observations into D segments. The method relies on the following crucial recursion

$$\begin{aligned}F(N, D) &= \min_{\eta_{D-1}} \left\{ \min_{\eta_1, \dots, \eta_{D-2}} \sum_{d=1}^{D-1} [\mathcal{C}(x_{\eta_{d-1}+1}, \dots, x_{\eta_d}) + k\alpha] + \mathcal{C}(x_{\eta_{D-1}+1}, \dots, x_N) + k\alpha \right\} \\ &= \min_{\eta_{D-1}} \left\{ F(\eta_{D-1}, D-1) + \mathcal{C}(x_{\eta_{D-1}+1}, \dots, x_N) + k\alpha \right\}.\end{aligned} \tag{1.20}$$

Algorithm 1: Optimal Partitioning

Require:
A set of data $(x_1, \dots, x_N)$, where $x_i \in \mathbb{R}$.
A cost function $\mathcal{C}(\cdot)$.
A penalty α not depending on change points.

Initializations: $F(0) = -k\alpha$, $cp(0) = NULL$.
for η^* in $1, \dots, N$ **do**
Calculate $F(\eta^*) = \min_{0 \leq \eta < \eta^*} \left[F(\eta) + \mathcal{C}(x_{\eta+1}, \dots, x_{\eta^*}) + k\alpha \right]$.
Let $\overline{\eta} = \text{argmin}_{0 \leq \eta < \eta^*} \left[F(\eta) + \mathcal{C}(x_{\eta+1}, \dots, x_{\eta^*}) + k\alpha \right]$.
$cp(\eta^*) = (cp(\overline{\eta}), \overline{\eta})$.
end for

Ensure: The change points stored in $cp(N)$.

The main intuition in the above recursion is that the optimal segmentation on a subset of data can be used to inform the optimal segmentation when adding one more data to the sequence. And this is exactly what Optimal Partitioning does by using backward the above recursion. Pseudocode Algorithm 1 illustrates how the algorithm works. The first line inside the *for* loop looks for the optimal last change point location between 0 and $\eta^* - 1$ and store it in $\overline{\eta}$. Once this is done, the subset of data $x_1, \dots, x_{\eta^*}$ is optimally partitioned and this information is used in the following steps to find and place other change points (if any) before N. Notice that, for each data point x_i, the algorithm tests *each* time point previous to i as possible last change point location. This explains the quadratic complexity of the algorithm.

Pruned Exact Linear Time (PELT). The PELT algorithm was introduced by Killick et al. (2012) as an exact segmentation algorithm, based on Optimal Partitioning, which is speeded up via *pruning*. As mentioned, at the i-th step of the *for* loop, in Algorithm 1, all the time points previous to i are tested as locations for the last optimal change point. However, it proves that some removed values not optimal at one step of the *for* loop can never be optimal in subsequent iterations and do not need to be checked again. More in details, assume that t_1 is a time point such that $0 \leq t_1 < t_2$ and

$$F(t_1) + \mathcal{C}(x_{t_1+1}, \dots, x_{t_2}) + k\alpha > F(t_2).$$

The above equation states that t_1 is *not* the location of the last change point prior to t_2. If moreover

$$F(t_1) + \mathcal{C}(x_{t_1+1}, \dots, x_{t_2}) > F(t_2), \tag{1.21}$$

then t_1 can never be the optimal last change point location prior to t_3 for all $t_3 > t_2$.

Proof. First of all, notice that due to the definition provided in (1.18) the following statement holds

$$\mathcal{C}(x_{t_1+1}, \dots, x_{t_2}) + \mathcal{C}(x_{t_2+1}, \dots, x_{t_3}) \leq \mathcal{C}(x_{t_1+1}, \dots, x_{t_3}).$$

Then, from (1.21) it follows

$$\begin{aligned} F(t_2) + \mathcal{C}(x_{t_2+1}, \dots, x_{t_3}) + k\alpha \\ < F(t_1) + \mathcal{C}(x_{t_1+1}, \dots, x_{t_2}) + \mathcal{C}(x_{t_2+1}, \dots, x_{t_3}) + k\alpha \\ \leq F(t_1) + \mathcal{C}(x_{t_1+1}, \dots, x_{t_3}) + k\alpha, \end{aligned}$$

stating that t_2 can never be the last optimal change point location before t_3. □

In Killick et al. (2012), the authors proved that, under some assumptions concerning the generative models of both data and change points, the PELT algorithm has a computational cost linear in the number of observations ($\mathcal{O}(N)$). The reader is referred to that paper for more details and a pseudocode of the original algorithm. A pseudocode of the modified algorithm dealing with graph data is provided in Chapter 3.

1.4.3 Latent Dirichlet allocation for statistical analysis of texts

The last chapter 4 of this thesis describes a new model for clustering and performing statistical analysis in weighted dynamic graphs with textual edges (i.e. a text is associated with each edge). Communications via social media like Facebook, Twitter or Linkedin are examples of textual networks: interactions between individuals consist in messages whose content can be used to capture information. Probabilistic approaches for network analysis *not* involving text analysis were discussed in the previous sections. In Chapter 4, some methods improving joint analysis of texts and networks will be illustrated. This section offers an overview of some existing models for statistical analysis of documents and focuses on one of such models in particular, the latent Dirichlet allocation (LDA, Blei et al. 2003).

One of the earliest models for statistical analysis of documents is LSI (Latent Semantic Indexing, Papadimitriou et al. 1998). It allows to detect linguistic notions such as synonymy from a data weighting called "term frequency - inverse document frequency" (tf-idf). A probabilistic extension of the latent semantic indexing (a.k.a. pLSI) was proposed by (Hofmann 1999), who modelled the words of a document as a mixture of multinomial random variables. In this context, the hidden groups of words were referred to as "topics", meaning that each word of a document is associated with a single topic.

Based on pLSI, LDA assumes that words in a document follow a mixture distribution over latent topics. Moreover, when several documents are taken into account a vector of topic proportions (i.e. the relative number of words associated with each topic) is independently generated for each document, according to a Dirichlet distribution. More in details, let us consider a corpus of D documents made out of words from a dictionary containing $T^{(W)}$ words. Each document is denoted by W^d, $d \leq D$, and contains a sequence of N_{W^d} words such that, using a zero-one coding, we have

$$W^d_{nw} = 1,$$

if the n-th word in the d-th document is the word w in the dictionary, zero otherwise. Each document W^d is associated with a latent vector θ_d drawn

from a Dirichlet distribution

$$\theta_d \sim \text{Dir}(\alpha = (\alpha_1, \ldots, \alpha_L)),$$

where L is the number of topics. Hence θ_d is a L-vector such that $\sum_{l=1}^{L} \theta_{dl} = 1$ and θ_{dl} is the expected proportion of words extracted from the l-th topic in document W^d. As said, the topic proportions $(\theta_1, \ldots, \theta_D)$ associated with each document are assumed to be independent. Moreover, a latent vector V^d whose length is N_{W^d} is associated with each document such that

$$V_{nl}^d = 1$$

if the n-th word in document W^d is extracted from the l-th topic, zero otherwise. V^d is assumed to follow a multinomial distribution, such that

$$V_n^d \sim \mathcal{M}(1, \theta_d)$$

and the random variables

$$V_1^d, \ldots, V_{N_{W^d}}^d$$

are conditionally independent given θ_d.

In a similar fashion, W_n^d follows a multinomial distribution conditionally on V^d

$$W_n^d | V_{nl}^d = 1 \sim \mathcal{M}(1, \beta_l = (\beta_{l1}, \ldots, \beta_{lT^{(W)}})),$$

where $\sum_{w=1}^{T^{(W)}} \beta_{lw} = 1$. Under this assumption, β_{lw} is the probability that world w in the dictionary is extracted from the l-th topic. The random variables

$$W_1^d, \ldots, W_{N_{W^d}}^d$$

are conditionally independent given V^d.

Based on these assumptions, the complete data likelihood of LDA can be decomposed as follows

$$\begin{aligned} p(W, V, \theta | \beta, \alpha) &= p(W|V, \beta) p(V|\theta) p(\theta|\alpha) \\ &= \prod_{d=1}^{D} \left(p(\theta_d|\alpha) \prod_{n=1}^{N_{W^d}} p(W_n^d | V_n^d, \beta) p(V_n^d | \theta_d) \right). \end{aligned}$$

A graphical model representation of the LDA model can be seen in Figure 1.5.

Several inference procedures have been proposed for LDA. For instance a VEM approach was adopted in the original paper of Blei et al. (2003) and a collapsed variational Bayes EM algorithm was presented in Teh et al. (2006).

Due to the independence of the variables V_n^d, one drawback of LDA consists in its blindness with respect to eventual correlation between topics. A correlated topic model (CTM) was developed by Blei and Lafferty (2007) to address this issue. Similarly the relational topic model (RTM, Chang and Blei 2009) models the links between documents as binary random variables conditioned on their content, but ignores the community ties between the authors of these documents. RTM is extended to deal with weighted graphs by Sun et al. (2009). For a detailed survey on probabilistic topic models the reader is referred to Blei (2012).

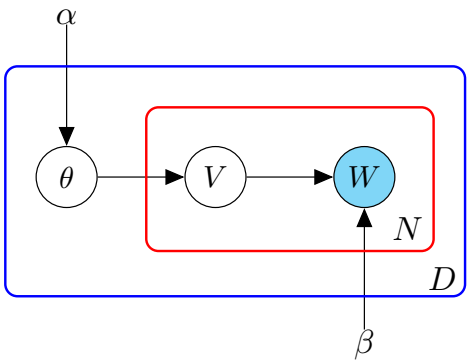


Figure 1.5 – *Graphical model representation of LDA for the d-th document. The boxes are plates representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.*

Conclusion

In this chapter, we reviewed important notions of graph theory and graph clustering. Several approaches for static and dynamic network analysis were presented with a particular emphasis on the stochastic block model (SBM). Estimation and model selection for this model were addressed and some extensions to dynamic graphs were presented, both in discrete and continuous time. Finally, we introduced and detailed three topics (NHPP, PELT and LDA) massively used in the following chapters.

A dynamic extension of the stochastic block model

2

This chapter introduces and details a dynamic stochastic block model (dSBM) in which interactions between nodes of a discrete time dynamic graph (see Section 1.1.2) are counted by non homogeneous Poisson processes (NHPPs). The proposed approach aims to cluster vertices of a dynamic network in time invariant groups, whose number has to be estimated. Section 2.1 describes the dSBM model in detail. Possible overfitting problems of this model are discussed and a regularized version solving these issues is proposed. In Section 2.2, an exact version of the ICL criterion for dSBM is formally obtained. A maximization algorithm based on a greedy search approach is detailed. This approach allows to simultaneously estimate both the cluster memberships and the number of clusters. In the last part of the section, a maximum-likelihood estimator is developed to estimate the integrated intensity functions of the NHPPs in a non parametric fashion. Section 2.3 focuses on experiments on both simulated and real data allowing us to highlight the main features of the proposed methodology.

2.1 The dynamic stochastic block model (dSBM)

In this chapter we employ stochastic processes to count the interactions between the pairs of nodes of a dynamic graph. Each process is uniquely associated with a pair of nodes. We adopt a discrete time modelling approach and focus on the increments of the stochastic processes on a user defined partition. A more general, continuous time view (see Section 1.1.2) is adopted in the next chapter. Graphs are assumed to be directed and self loops are not allowed. All results presented in this chapter can easily be generalized to account for undirected interactions.

Block modelling. Since the proposed approach relies on the stochastic block model (SBM), we revise the notations introduced in Section 1.2. A graph consists in N nodes, interacting as frequently as wanted during the time interval $[0, T]$. SBM assumes that nodes belong to hidden groups that solely explain the way in which nodes interact to each other. Using the same notations of the previous chapter, vertices are clustered in K groups $\mathcal{A}_1, \dots, \mathcal{A}_K$ and a hidden cluster membership set $Z = \{Z_1, \dots, Z_N\}$ is introduced such that

$$Z_i = k \qquad \text{iff} \qquad i \in \mathcal{A}_k, \qquad k \in \{1, \dots, K\}.$$

The random component Z_i is assumed to follow a multinomial distribution with parameter vector $\pi = (\pi_1, \dots, \pi_K)$, such that

$$\mathbb{P}(Z_i = k) = \pi_k \qquad \text{with} \qquad \sum_{k=1}^{K} \pi_k = 1.$$

The above equation says that the i-th node belongs to group $\mathcal{A}_k$ with probability π_k. In addition, the random variables $\{Z_i\}_{1 \le i \le N}$ are assumed to be independent. Thus

$$p(Z|\pi, K) = \prod_{i=1}^{N} \pi_{Z_i} = \prod_{k=1}^{K} \pi_k^{|\mathcal{A}_k|}, \tag{2.1}$$

where $|\mathcal{A}_k|$ denotes the cardinal of $\mathcal{A}_k$ or, equivalently, the number of nodes assigned to the k-th cluster. So far the setup is identical to the one in Section 1.2 and we recall that the number of groups K is unknown and has to be estimated.

Non homogeneous Poisson processes. As mentioned in Section 1.1.2, stochastic processes can be introduced to count interactions between pairs of nodes in a dynamic graph. Indeed, the value of the process associated with the pair (i, j) at time t is the number of interactions that took place from i to j up to time t. A possible choice for the counting process associated with each pair of nodes is the non homgeneous Poisson process (NHPP), introduced in Section 1.4.1. In more details, we denote by $\{M_{ij}(t)\}_{t \le T}$ the stochastic process counting the interactions from node i to node j up to time t. Two assumptions are made:

1. The processes associated with different pairs of nodes are conditionally independent given Z. Then, since we are considering directed graphs, there are $N \times (N-1)$ independent processes.

2. $M_{ij}(\cdot)$ is a non homogeneous Poisson process conditionally on Z, whose intensity function only depends on Z_i and Z_j

$$p(M_{ij}(t)|Z, \boldsymbol{\lambda}, K) = \frac{(\int_0^t \lambda_{Z_i Z_j}(s)ds)^{M_{ij}(t)}}{M_{ij}(t)!} \exp\left(-\int_0^t \lambda_{Z_i Z_j}(s)ds\right),$$

for all $t \leq T$ where $\boldsymbol{\lambda} = \{\lambda_{kg}(t)\}_{1\leq k,g\leq K}$ denotes the set of the instantaneous intensity functions.

For each $t \in [0, T]$ and conditionally on Z, the above assumptions tell us that $M_{i_1 j_1}(t)$ and $M_{i_2 j_2}(t)$ are two independent random variables whenever the pair (i_1, j_1) is different from (i_2, j_2). Moreover these two r.v. are Poisson distributed and they have the same distribution only if $(Z_{i_1}, Z_{j_1}) = (Z_{i_2}, Z_{j_2})$.

Remark 2.1 *In the reminder of this thesis, with a slight abuse of language, we say that the process $M_{ij}(\cdot)$ is a "non homogeneous Poisson process", but this statement is correct uniquely conditionally on Z to be known.*

This can easily be seen via the characteristic function of the random variable $M_{jl}(t)$, for a fixed t and a fixed pair of nodes (j, l)

$$\mathbb{E}\left[\exp\left(i\mu M_{jl}(t)\right)|Z\right] = \exp\left(g(t, Z_j, Z_l)(e^{i\mu} - 1)\right), \quad \forall t \leq T \tag{2.2}$$

where

$$g(t, Z_j, Z_l) := \int_0^t \lambda_{Z_j Z_l}(s)ds,$$

i is the imaginary unit and μ is a real constant. On the right hand side of (2.2) we can see the characteristic function of a Poisson distributed r.v. whose parameter is $g(t, Z_j, Z_l)$. When Z is unobserved, however, $g(t, Z_j, Z_l)$ is a random variable and computing the unconditional expectation leads to

$$\begin{aligned}\mathbb{E}\left[\exp\left(i\mu M_{jl}(t)\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(i\mu M_{jl}(t)\right)|Z\right]\right] \\ &= \sum_{k=1}^{K}\sum_{g=1}^{K} \exp\left(g(t, k, g)(e^{i\mu} - 1)\right)\pi_k \pi_g,\end{aligned}$$

which is not at all the characteristic function of a Poisson distributed random variable.

2.1.1 Discrete time version

Consider a partition of the interval $[0, T]$ based on a set of $U + 1$ time points

$$0 = t_0 < t_1 < \cdots < t_{U-1} < t_U = T, \tag{2.3}$$

that defines U intervals $I_u := [t_{u-1}, t_u[$ of arbitrary length Δ_u. Consider two nodes i and j. The number of interactions between these two nodes, on each time interval I_u is counted by the following random variable

$$X_{ij}^{I_u} := M_{ij}(t_u) - M_{ij}(t_{u-1}), \qquad u \in \{1, \ldots, U\}. \tag{2.4}$$

Hence, $X_{ij}^{I_u}$ measures the increment over the time interval I_u of the NHPP counting the interactions from i to j. If we denote X_{ij} the random vector

$$X_{ij} := (X_{ij}^{I_1}, \dots, X_{ij}^{I_U})^T,$$

thanks to the incremental independence of NHPPs, the following conditional probability can be obtained

$$p(X_{ij}|Z, \boldsymbol{\lambda}) = \prod_{u=1}^{U} \left(\frac{(\int_{I_u} \lambda_{Z_i Z_j}(s)ds)^{X_{ij}^{I_u}}}{X_{ij}^{I_u}!} \exp\left(- \int_{I_u} \lambda_{Z_i Z_j}(s)ds \right) \right). \quad (2.5)$$

Remark 2.2 *Notice that the above probability distribution is conditional on K being known and it should be written $p(X_{ij}|Z, \boldsymbol{\lambda}, K)$. However, to keep notation uncluttered, such dependency is omitted.*

Remark 2.3 *The vector X_{ij} can be seen as a time series of independent* not *identically distributed random variables.*

Notations can be simplified further by employing *integrated* intensity functions (a.k.a. IIFs) $\Lambda_{kg}(\cdot)$, defined on $[0, T]$ by

$$\Lambda_{kg}(t) := \int_0^t \lambda_{kg}(s)ds,$$

for all k, g. The increments of the IIFs on I_u are denoted by

$$\Delta\Lambda_{kg}^{I_u} := \Lambda_{kg}(t_u) - \Lambda_{kg}(t_{u-1}), \qquad \forall u \in \{1, \dots, U\}. \quad (2.6)$$

Remark 2.4 *If $\lambda_{kg}(\cdot)$ is assumed constant on I_u, namely*

$$\lambda_{kg}(t) := \sum_{u=1}^{U} \lambda_{kgu} \mathbf{1}_{I_u}(t),$$

where $\mathbf{1}_{I_u}(\cdot)$ is the indicator function on I_u and $\lambda_{kgu} > 0$, then the following equality holds

$$\Delta\Lambda_{kg}^{I_u} = \Delta_u \lambda_{kgu}.$$

This condition, however, is not *required in the reminder of this chapter and no assumption is formulated about the shape of $\lambda_{kg}(\cdot)$[1].*

Equation (2.5) can be rewritten as

$$p(X_{ij}|Z, \boldsymbol{\Delta\Lambda}) = \prod_{u=1}^{U} \left(\frac{(\Delta\Lambda_{Z_i Z_j}^{I_u})^{X_{ij}^{I_u}}}{X_{ij}^{I_u}!} \exp\left(-\Delta\Lambda_{Z_i Z_j}^{I_u} \right) \right), \quad (2.7)$$

where $\boldsymbol{\Delta\Lambda}$ is a $K \times K \times U$ tensor, whose entry (k, g, u) is $\Delta\Lambda_{kg}^{I_u}$. In a similar manner, the $N \times N \times U$ tensor $\boldsymbol{X} = \{X_{ij}^{I_u}\}_{i,j,u}$ is defined.

[1] On the contrary, a very similar assumption will play a crucial role in the following chapter.

The assumption of conditional independence of the processes $\{M_{ij}(\cdot)\}_{i,j}$ immediately leads to

$$p(\boldsymbol{X}|Z, \boldsymbol{\Delta\Lambda}) = \prod_{i,j}^{N} p(X_{ij}|Z, \boldsymbol{\Delta\Lambda}). \tag{2.8}$$

To simplify the rest of this chapter the following short hand notations are used

$$\prod_{i,j}\prod_{k,g}\prod_{u} := \prod_{i=1}^{N}\prod_{\substack{j=1\\ j\neq i}}^{N}\prod_{k=1}^{K}\prod_{g=1}^{K}\prod_{u=1}^{U}$$

$$\prod_{Z_i=k}\left(\prod_{Z_j=g}\right) := \prod_{\substack{i:\\ Z_i=k}}\left(\prod_{\substack{j:\\ Z_j=g}}\right).$$

The conditional distribution of $\boldsymbol{X}$ given Z and $\boldsymbol{\Delta\Lambda}$ is

$$\begin{aligned} p(\boldsymbol{X}|Z, \boldsymbol{\Delta\Lambda}) &= \prod_{i,j}\prod_{u}\left(\frac{(\Delta\Lambda^{I_u}_{Z_iZ_j})^{X^{I_u}_{ij}}}{X^{I_u}_{ij}!}\exp\left(-\Delta\Lambda^{I_u}_{Z_iZ_j}\right)\right) \\ &= \prod_{k,g}\prod_{u}\left(\frac{(\Delta\Lambda^{I_u}_{kg})^{S_{kgu}}}{P_{kgu}}\exp\left(-|\mathcal{A}_k||\mathcal{A}_g|\Delta\Lambda^{I_u}_{kg}\right)\right), \end{aligned} \tag{2.9}$$

where

$$S_{kgu} = \sum_{Z_i=k}\sum_{Z_j=g} X^{I_u}_{ij}$$

is the total number of interactions from cluster k to cluster g (possibly equal to k) during the time interval I_u and

$$P_{kgu} = \prod_{Z_i=k}\prod_{Z_j=g} X^{I_u}_{ij}!.$$

Relying on (2.1) and (2.9) the complete data likelihood for dSBM is obtained

$$p(\boldsymbol{X}, Z|\boldsymbol{\Delta\Lambda}, \pi) = p(\boldsymbol{X}|Z, \boldsymbol{\Delta\Lambda})p(Z|\pi). \tag{2.10}$$

and a graphical representation can be seen in Figure 2.1.

2.1.2 Constraints on the integrated intensity functions

As it will be shown in Section 2.3.1, the model presented so far is prone to over fitting when the number of sub-intervals U is large compared to N. For example, if too many intervals are used then there will be just one event per interval and thus over fitting will occur.

However, imposing some constraints to the intensity functions $\{\Lambda_{kg}(t)\}_{k,g}$ can solve the over fitting problem.

Let us consider a fixed pair of clusters (k, g). So far, the increments $\{\Delta\Lambda^{I_u}_{kg}\}_{u\leq U}$ were all distinct parameters (indeed the tensor $\boldsymbol{\Delta\Lambda}$ has dimension $K \times K \times U$). A constraint can be introduced by assigning the

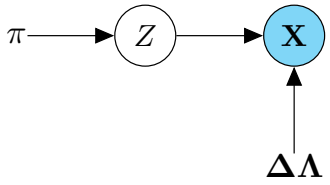


Figure 2.1 – *Graphical representation of dSBM. The observed three dimensional tensor* $\boldsymbol{X}$ *depends on the hidden set* Z *and the model parameter* $\boldsymbol{\Delta\Lambda}$.

time intervals $(I_1, \dots I_U)$ to different time clusters and by assuming that the IIF increments are identical for all the time intervals belonging to the same time cluster. Formally D time clusters $(\mathcal{C}_1, \dots, \mathcal{C}_D)$ are introduced and a set of hidden variables $Y = \{Y_1, \dots, Y_U\}$ is associated with the time intervals $(I_1, \dots, I_U)$ such that

$$Y_u = d \qquad \text{iff} \qquad I_u \in \mathcal{C}_d.$$

Each random variable Y_u is assumed to follow a multinomial distribution depending on parameter ρ

$$\mathbb{P}(Y_u = d) = \rho_d \qquad \text{with} \qquad \sum_{d=1}^{D} \rho_d = 1.$$

This means that the time interval I_u is assigned to the time cluster $\mathcal{C}_d$ with probability ρ_d. In addition, the variables $Y_1, \dots, Y_U$ are assumed to be independent

$$p(Y|\rho, D) = \prod_{u=1}^{U} \rho_{Y_u} = \prod_{d=1}^{D} \rho_d^{|\mathcal{C}_d|}, \tag{2.11}$$

where $|\mathcal{C}_d|$ is the cardinality of $\mathcal{C}_d$, i.e. the number of time intervals assigned to the d-th time cluster. The constraint introduced in this section can be illustrated as follows. Assume that two distinct time intervals I_u and I_l not necessarily adjacent belong to the same time cluster $\mathcal{C}_d$. It follows that

$$Y_u = Y_l = d.$$

With a slight abuse of notation, we then assume that

$$\Delta\Lambda_{kg}^{I_u} = \Delta\Lambda_{kg}^{I_l} =: \Delta\Lambda_{kg}^{d} \qquad \forall k, g. \tag{2.12}$$

Hence, in this new formulation of dSBM, the random variable $X_{ij}^{I_u}$ is assumed to depend on both Z and Y

$$p(X_{ij}^{I_u}|Z, Y, \boldsymbol{\Delta\Lambda}) = \frac{\left(\Delta\Lambda_{Z_iZ_j}^{Y_u}\right)^{X_{ij}^{I_u}}}{X_{ij}^{I_u}!} \exp\left(-\Delta\Lambda_{Z_iZ_j}^{Y_u}\right). \tag{2.13}$$

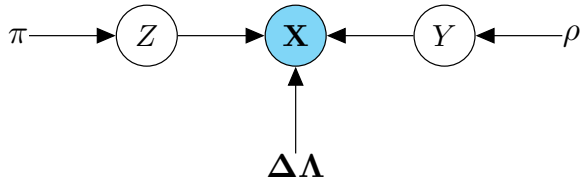


Figure 2.2 – *Graphical representation of CdSBM. The observed tensor* $\boldsymbol{X}$ *now depends on both* Z *(node clusters) and* Y *(time clusters).*

Remark 2.5 *Notice that the new Poisson parameter* $\Delta\Lambda^{Y_u}_{Z_iZ_j}$ *replaces* $\Delta\Lambda^{I_u}_{Z_iZ_j}$ *in the previous version. It means that the dependence on the time interval* I_u *is replaced by the dependence on the time cluster of* I_u*, namely* Y_u*.*

In this framework, a unique parameter $\Delta\Lambda^d_{kg}$ is associated with the interactions from nodes in cluster $\mathcal{A}_k$ to nodes in cluster $\mathcal{A}_g$ during the time intervals in time cluster $\mathcal{C}_d$. As a consequence, the tensor $\boldsymbol{\Delta\Lambda}$ has now dimension $K \times K \times D$ instead of $K \times K \times U$. Since in practical applications D is significantly smaller than U, we obtained an important reduction of the number of parameters in the model.

The conditional distribution of $\boldsymbol{X}$ given Z and Y is

$$\begin{aligned} p(\boldsymbol{X}|Z,Y,\boldsymbol{\Delta\Lambda}) = &\prod_{i,j}\prod_{u}\left(\frac{(\Delta\Lambda^{Y_u}_{Z_iZ_j})^{X^{I_u}_{ij}}}{X^{I_u}_{ij}!}\exp\left(-\Delta\Lambda^{Y_u}_{Z_iZ_j}\right)\right) \\ &\prod_{k,g}\prod_{d}\left(\frac{(\Delta\Lambda^{d}_{kg})^{S_{kgd}}}{P_{kgd}}\exp\left(-|\mathcal{A}_k||\mathcal{A}_g||\mathcal{C}_d|\Delta\Lambda^{d}_{kg}\right)\right), \end{aligned} \tag{2.14}$$

where

$$S_{kgd} = \sum_{Z_i=k}\sum_{Z_j=g}\sum_{Y_u=d} X^{I_u}_{ij} \tag{2.15}$$

is the number of interactions from nodes in cluster $\mathcal{A}_k$, to nodes in cluster $\mathcal{A}_g$, during the time intervals in time cluster $\mathcal{C}_d$ and where

$$P_{kgd} = \prod_{Z_i=k}\prod_{Z_j=g}\prod_{Y_u=d} X^{I_u}_{ij}!. \tag{2.16}$$

Assuming that Z and Y are independent and using (2.14), (2.11) and (2.1) the complete data likelihood is given by

$$p(\boldsymbol{X},Z,Y|\boldsymbol{\Delta\Lambda},\pi,\rho) = p(\boldsymbol{X}|Z,Y,\boldsymbol{\Delta\Lambda})p(Z|\pi)p(Y|\rho) \tag{2.17}$$

and a graphical representation can be seen in Figure 2.2.

Summary

Two generative dSBM models were defined:

dSBM K is the number of latent node clusters, Z labels node memberships to clusters and π is the parameter of the multinomial distribution followed by Z. The tensor $\mathbf{\Delta\Lambda}$ has dimension $K \times K \times U$. Given Z and $\mathbf{\Delta\Lambda}$, the model generates a tensor of interaction counts $\boldsymbol{X}$ according to (2.9).

CdSBM: it is a **constrained** version of dSBM. The number of time clusters is D and Y labels the time intervals memberships to time clusters. The parameter ρ characterizes the multinomial distribution of Y. In CdSBM, $\mathbf{\Delta\Lambda}$ is a $K \times K \times D$ tensor and given Z, Y and $\mathbf{\Delta\Lambda}$, the model generates a tensor of interaction counts $\boldsymbol{X}$ according to (2.14).

In the remaining of this chapter, when not differently stated the expression "the model" refers to dSBM.

The following very important remark concludes the section.

Remark 2.6 *The way CdSBM is formulated is in no way the only possible solution to impose regularity constraints to the integrated functions* $\{\Lambda_{kg}(\cdot)\}_{k,g}$*. An alternative approach would be to formulate a* segmentation *constraint, i.e. forcing each temporal cluster to contain* only *adjacent time intervals. This approach is adopted and detailed in the following chapter.*

2.2 Inference

The present section derives an inference procedure to estimate the labels Z and Y as well as the number of clusters K and time clusters D in a dynamic graph simulated according to CdSBM. Notice that, if dSBM is considered instead of CdSBM, the inference task reduces to the estimation of Z and K only. Therefore, an inference procedure allowing us to learn CdSBM is a more general one.

A standard solution to estimate Z, Y, K and D would be to rely on a variational EM algorithm (see Sections 1.3.1 and 1.3.2) to estimate Z and Y for any pair (K, D) varying in a certain range. Then, the ICL model selection criterion (described in Section 1.3.3) could be used to select the values of K and D. However, this approach can be computationally very expensive since a VEM algorithm needs to be run for each value of K and D varying in $\{1, \dots, K_{max}\} \times \{1, \dots, D_{max}\}$ for some K_{max} and D_{max}.

An alternative inference procedure can be developed by following Côme and Latouche (2015). From a Bayesian perspective, they obtained an exact version of the ICL criterion for the standard stochastic block model and maximized it *directly* with respect to the number of clusters (K) and cluster memberships (Z). The maximization was performed relying on a *greedy search* approach. They ran several experiments on simulated and real data showing that their approach provided more accurate estimates than those obtained by variational inference or MCMC techniques. Similar findings are provided in Wyse et al. (2017), in the context of latent block models (LBMs) for bipartite graphs: the greedy ICL approach outperforms its competitors in both computational terms and the accuracy of

the provided estimates. In the following section we adapt the approach introduced by Côme and Latouche (2015) to our context. We formally obtain the *exact* ICL for CdSBM and detail a *greedy* search strategy to maximize it.

2.2.1 Exact ICL for dSBM

In the remaining of this chapter the expressions "ICL" or "exact ICL" will be used interchangeably when no confusion arises.

From a Bayesian perspective the CdSBM model parameters $\mathbf{\Delta\Lambda}$, π and ρ are assumed to be latent random variables following a *prior* joint distribution. This distribution is conditional on K and D being known and it is denoted by $p(\mathbf{\Delta\Lambda}, \pi, \rho|K, D)$. When introducing the ICL criterion for SBM (Section 1.3.3) we saw that this criterion aimed to approximate the complete data integrated log-likelihood. Similarly, the exact ICL for CdSBM is nothing more than its *complete data* integrated log-likelihood

$$\begin{aligned} ICL(Z, Y, K, D) :&= \log p(\boldsymbol{X}, Z, Y|K, D) \\ &= \log\left(\int p(\boldsymbol{X}, Z, Y|\mathbf{\Delta\Lambda}, \Phi) p(\mathbf{\Delta\Lambda}, \Phi|K, D) d\mathbf{\Delta\Lambda} d\Phi\right), \end{aligned} \tag{2.18}$$

where $\Phi = \{\pi, \rho\}$. Notice that the marginalization over all model parameters naturally induces a penalization on the number of clusters and time clusters. If the function $ICL(\cdot)$ in (2.18) could be maximized directly, estimates of Z, Y, K and D would be available. Obviously, no closed form solution for such a maximization exists. However, the integral in the above equation can be decomposed by adopting the following independence assumption on the prior distribution

$$p(\mathbf{\Delta\Lambda}, \pi, \rho|K, D) = p(\mathbf{\Delta\Lambda}|K, D) p(\pi|K) p(\rho|D).$$

Hence

$$\begin{aligned} ICL(Z, Y, K, D) =& \log\left(\int_{\mathbf{\Delta\Lambda}} p(\boldsymbol{X}|\mathbf{\Delta\Lambda}, Z, Y, K, D) p(\mathbf{\Delta\Lambda}|K, D) d\mathbf{\Delta\Lambda}\right) \\ &+ \log\left(\int_{\pi} p(Z|\pi, K) p(\pi|K) d\pi\right) \\ &+ \log\left(\int_{\rho} p(Y|\rho, D) p(\rho|D) d\rho\right) \\ =& \log p(\boldsymbol{X}|Z, Y, K, D) + \log p(Z|K) + \log p(Y|D). \end{aligned} \tag{2.19}$$

In general, the last three terms on the right hand side of above equality do not have an explicit form, since the corresponding integrals cannot be explicitly computed. However, a sensible choice of prior distributions over the model parameters can fix this issue. Conjugate prior distributions are the most natural choice.

Gamma prior distribution. In order to integrate $\mathbf{\Delta\Lambda}$ out and obtain a closed formula for $\log p(\boldsymbol{X}|Z, Y, K, D)$, on the right hand side of (2.19),

$\mathbf{\Delta\Lambda}^d_{kg}$ is assumed to follow a Gamma distribution

$$p(\Delta\Lambda^d_{kg}|a,b) = \frac{b^a}{\Gamma(a)}(\Delta\Lambda^d_{kg})^{a-1}e^{-b\Delta\Lambda^d_{kg}}, \tag{2.20}$$

where $a, b > 0$ and $\Gamma(\cdot)$ is the gamma function

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt, \qquad x \in \mathbb{R}^+.$$

Notice that the positive parameters a and b could vary as a function of (k, g, d). However, in the following we assume that they are constant for simplicity. The joint density of $\mathbf{\Delta\Lambda}$ is obtained via the following independence assumption

$$p(\mathbf{\Delta\Lambda}|K,D) = \prod_{k,g}^{K}\prod_{d}^{D} p(\Delta\Lambda^d_{kg}|a,b). \tag{2.21}$$

Equations (2.20) and (2.21) can be rephrased by saying that the random variables $\{\Delta\Lambda^d_{kg}\}_{k,g,d}$ are i.i.d following a Gamma$(\Delta\Lambda; a, b)$ distribution.

By multiplying the likelihood in (2.14) with the above probability density function the joint distribution of the pair $(\boldsymbol{X}, \mathbf{\Delta\Lambda})$ follows

$$p(\boldsymbol{X}, \mathbf{\Delta\Lambda}|Z,Y,K,D) = \prod_{k,g}^{K}\prod_{d}^{D}\left[\frac{b^a}{\Gamma(a)P_{kgd}}e^{-\Delta\Lambda^d_{kg}[|\mathcal{A}_k||\mathcal{A}_g||\mathcal{C}_d|+b]}(\Delta\Lambda^d_{kg})^{S_{kgd}+a-1}\right],$$

where S_{kgd} and P_{kgd} were defined in (2.15) and (2.16), respectively. The functional form of a Gamma distribution can be recognised and the above probability can now be integrated w.r.t. $\mathbf{\Delta\Lambda}$ to obtain

$$p(\boldsymbol{X}|Z,Y,K,D) = \prod_{k,g}^{K}\prod_{d}^{D} L_{kgd}, \tag{2.22}$$

with

$$L_{kgd} := \frac{b^a}{\Gamma(a)P_{kgd}}\frac{\Gamma(S_{kgd}+a)}{[|\mathcal{A}_k||\mathcal{A}_g||\mathcal{C}_d|+b]^{S_{kgd}+a}}. \tag{2.23}$$

The above probability depends on the values of the hyper parameters a and b. A non informative prior for the Poisson distribution would correspond to limiting cases of the Gamma family, when b tends to zero. A possible choice is to set $a = 1$ to obtain DK^2 i.i.d. exponential random variables $\{\Delta\Lambda^d_{kg}\}_{k,g,d}$. The value of b remains an open issue. In all the experiments we carried out, the parameters a and b were set equal to one in order to have unitary mean and variance for the Gamma distribution.

Dirichlet prior distribution. In order to obtain a closed form for $p(Z|K)$ and $p(Y|D)$ a factorizing Dirichlet prior distribution is attached to the pair (π, ρ), namely

$$p(\pi,\rho|K,D) = \mathrm{Dir}_K(\pi;\alpha,\ldots,\alpha) \times \mathrm{Dir}_D(\rho;\beta,\ldots,\beta),$$

where the parameters of each distribution (α and β) were set constant for simplicity. As proven in Appendix 2.4.1 the joint integrated distribution for the pair (Z, Y), reduces to

$$p(Z, Y|K, D) = \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \frac{\prod_k^K \Gamma(|\mathcal{A}_k| + \alpha)}{\Gamma(N + \alpha K)} \frac{\Gamma(\beta D)}{\Gamma(\beta)^D} \frac{\prod_d^D \Gamma(|\mathcal{C}_d| + \beta)}{\Gamma(U + \beta D)}. \quad (2.24)$$

A common choice (adopted in the experiments in the next section) consists in fixing α and β to 1 to get uniform distributions. Alternatively, α and β could be set equal to $1/2$ to obtain Jeffreys' non informative prior distributions.

A graphical representation of Bayesian dSBM and CdSBM can be seen in Figure 2.3.

2.2.2 ICL maximization

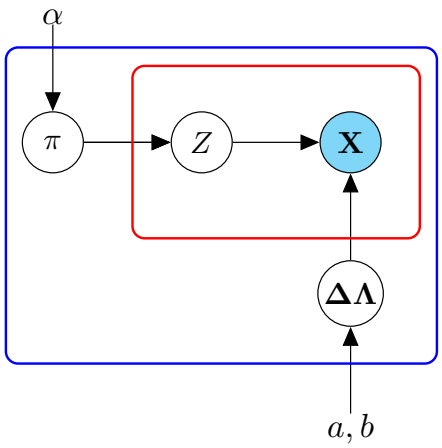
The complete data integrated log-likelihood (a.k.a. exact ICL) in (2.19) has to be maximized with respect to the four unknowns Z, Y, K and D, which are discrete variables. Obviously no closed formulas can be obtained and it would be computationally prohibitive to test every combination of the four unknowns. However, a *greedy search* strategy can be adopted to get a local optimum. The main idea is to start with an initial clustering of nodes and time intervals and then to alternate between an exchange phase where nodes/intervals can move from one cluster to another and a merge phase where clusters/time clusters are merged. Exchange and merge operations are *locally* optimal and the following remark holds

Remark 2.7 *The* greedy search *algorithm detailed in the following is guaranteed to increase the ICL at each step and to converge to a* local *maximum. Randomization can be used to explore several local maxima but the convergence to a global maximum is not guaranteed.*

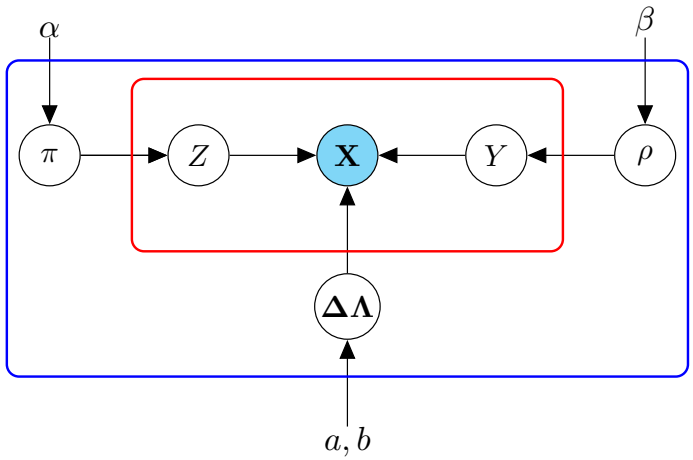
The algorithm is described in detail in the rest of this section. An analysis of its computational complexity is provided in Appendix 2.4.2.

Initialization. Initial values are fixed for both K and D, say $K_{\max}$ and $D_{\max}$. These values may be fixed equal to N and U respectively and each node (respectively time interval) would be alone in its own cluster (resp. time cluster). Alternatively, simple clustering algorithms (k-means, hierarchical clustering or spectral clustering) may be used to obtain initial values of Z and Y for some $K_{\max} \ll N$ and $D_{\max} \ll U$. This choice should be preferred to speed up the greedy search.

Greedy - Exchange (GE). A shuffled sequence of all the nodes (resp. time intervals) in the graph is created. One node (resp. time interval) is chosen and is moved from its current cluster (resp. time cluster) into the cluster (resp. time cluster) leading to the highest increase in the ICL, if any. This step is called greedy exchange (GE). GE is applied to every node (resp. time interval) in the shuffled sequence. This iterative procedure is repeated until no further improvement in the exact ICL is possible. Notice that, when a node (resp. time interval) is alone inside its cluster, an exchange becomes a merge of two clusters (see below).



(a) *Graphical representation of Bayesian dSBM.*



(b) *Graphical representation of Bayesian CdSBM.*

Figure 2.3 – *Graphical representation of dSBM and CdSBM from a Bayesian perspective. The red plates contain the random variables (both observed and hidden) in the Frequentist version of the two models. From a Bayesian view, the model parameters* $(\mathbf{\Delta\Lambda}, \pi, \rho)$ *are seen as latent random variables. The blue plates contain the random variables (both observed and hidden) in the Bayesian version of the two models.*

The ICL needs not to be evaluated before and after each swap since possible increases can be computed directly, thus reducing the computational cost. Let us consider first the case of time intervals. Moving an interval I_u from the cluster $\mathcal{C}_{d'}$ to cluster $\mathcal{C}_l$ induces a change in the ICL

given by

$$\begin{aligned}\Delta^{E,T}_{d'\to l} :=& ICL(Z, Y^*, K, D) - ICL(Z, Y, K, D),\\ =& \left[\log p(Z, Y^*|K, D) + \sum_{k,g,d} \log L^*_{kgd}\right]\\ &- \left[\log p(Z, Y|K, D) + \sum_{k,g,d} \log L_{kgd}\right],\end{aligned}$$

where Y^* and L^*_{kgd} refer to the new configuration where $I_u \in \mathcal{C}_l$. It can easily be shown that $\Delta_{d'\to l}$ reduces to

$$\Delta^{E,T}_{d'\to l} = \log\left(\frac{\Gamma(|\mathcal{C}_{d'}| - 1 + \beta)\Gamma(|\mathcal{C}_l| + 1 + \beta)}{\Gamma(|\mathcal{C}_{d'}| + \beta)\Gamma(|\mathcal{C}_l| + \beta)}\right) + \sum_{k,g} \log\left(\frac{L^*_{kgd'} L^*_{kgl}}{L_{kgd'} L_{kgl}}\right). \quad (2.25)$$

The case of nodes is slightly more complex. When a node is moved from cluster $\mathcal{A}_{k'}$ to $\mathcal{A}_l$, with $k' \neq l$, the change in the ICL is

$$\Delta^{E,V}_{k'\to l} := ICL(Z^*, Y, K, D) - ICL(Z, Y, K, D),$$

which simplifies into

$$\begin{aligned}\Delta^{E,V}_{k'\to l} =& \log\left(\frac{\Gamma(|\mathcal{A}_{k'}| - 1 + \alpha)\Gamma(|\mathcal{A}_l| + 1 + \alpha)}{\Gamma(|\mathcal{A}_{k'}| + \alpha)\Gamma(|\mathcal{A}_l| + \alpha)}\right)\\ &+ \sum_{g\le K}\sum_{d\le D} \log L^*_{k'gd} + \sum_{g\le K}\sum_{d\le D} \log L^*_{lgd}\\ &+ \sum_{k\le K}\sum_{d\le D} \log L^*_{kk'd} + \sum_{k\le K}\sum_{d\le D} \log L^*_{kld}\\ &- \sum_d \log(L^*_{k'k'd} + \log L^*_{k'ld} + \log L^*_{lk'd} + \log L^*_{lld})\\ &- \sum_{g\le K}\sum_{d\le D} \log L_{k'gd} - \sum_{g\le K}\sum_{d\le D} \log L_{lgd}\\ &- \sum_{k\le K}\sum_{d\le D} \log L_{kk'd} - \sum_{k\le K}\sum_{d\le D} \log L_{kld}\\ &+ \sum_d (\log L_{k'k'd} + \log L_{k'ld} + \log L_{lk'd} + \log L_{lld}),\end{aligned}$$

where Z^* and L^*_{kgd} refer to the new configuration where the node is in cluster $\mathcal{A}_l$.

Greedy - Merge (GM). Once the GE step is concluded, all possible merges of pairs of clusters (resp. time clusters) are tested and the best merge is finally retained. This step is called greedy merge (GM) and it is repeated until no further improvement in the ICL is possible.

In this case too, the ICL does not need to be explicitly computed. Merging in fact time clusters $\mathcal{C}_{d'}$ and $\mathcal{C}_l$ into $\mathcal{C}_l$ leads to the following ICL modification

$$\begin{aligned}\Delta^{M,T}_{d'\to l} :=& ICL(Z, Y^*, K, D-1) - ICL(Z, Y, K, D)\\ =& \log\left(\frac{p(Z, Y^*|K, D-1)}{p(Z, Y|K, D)}\right) + \sum_{k,g}\left(\log L^*_{kgl} - \log L_{kgd'} L_{kgl}\right) \quad (2.26)\end{aligned}$$

Notice that if $d \leq l$, then l has to be replaced by $l-1$ inside L^*_{kgl}.

When merging clusters $\mathcal{A}_{k'}$ and $\mathcal{A}_l$ into the cluster $\mathcal{A}_l$, the change in the ICL can be expressed as follows

$$\begin{aligned}
\Delta^{M,V}_{k' \to l} :=& ICL(Z^*, Y, K-1, D) - ICL(Z, Y, K, D) = \\
=& \log\left(\frac{p(Z^*, Y|K-1, D)}{p(Z, Y|K, D)}\right) + \\
&+ \sum_{g \leq K} \sum_{d \leq D} (\log L^*_{lgd} + \log L^*_{kld}) - \sum_d \log L^*_{lld} \\
&- \sum_{g \leq K} \sum_{d \leq D} \log L_{k'gd} - \sum_{g \leq K} \sum_{d \leq D} \log L_{lgd} \\
&- \sum_{k \leq K} \sum_{d \leq D} \log L_{kk'd} - \sum_{k \leq K} \sum_{d \leq D} \log L_{kld} \\
&+ \sum_d (\log L_{k'k'd} + \log L_{k'ld} + \log L_{lk'd} + \log L_{lld}).
\end{aligned}$$

Optimization strategies. Two issues emerge:

1. The optimization order of nodes and time intervals. We could either run the greedy algorithm for nodes and time intervals separately or choose a hybrid strategy that switches and merges nodes and time intervals alternatively, for instance;

2. whether to execute merge or switch movements at first.

The second topic has been largely discussed in the context of modularity maximization for community detection in static graphs (see Section 1.2.1). One of the most commonly used algorithms is the so-called Louvain method (Blondel et al. 2008) which proceeds in a rather similar way as the one chosen here, i.e. switching nodes from clusters to clusters and then merging clusters. This is also the strategy used in Côme and Latouche (2015) for static SBM. Combined with a choice of sufficiently small values of K_{max} and D_{max}, this approach gives very good results at a reasonable computational cost. We recall that more complex approaches based on multilevel refinements of a greedy merge procedure have been shown to give better results than the Louvain method in the case of modularity maximization. The reader is referred to (Noack and Rotta 2008) for a detailed review of these approaches. However, the computational complexity of those approaches is acceptable only because of the very specific nature of the modularity criterion and with the help of specialized data structures. Such tools cannot be leveraged for ICL maximization.

The first issue (the optimization order) is hard to manage since the shape of the function $ICL(\cdot)$ is unknown. Three optimization strategies are developed in the following:

1. *GE* + *GM* for time intervals and then *GE* + *GM* for nodes (**Strategy A**);

2. *GE* + *GM* for nodes and then *GE* + *GM* for times (**Strategy B**);

3. Mixed *GE* + mixed *GM* (**Strategy C**).

In mixed *GE* a node is chosen in the shuffled sequence of nodes and moved to the cluster leading to the highest increase in the ICL. Then a time interval is chosen in the shuffled sequence of time intervals and placed in the best time cluster and so on, alternating between nodes and time intervals until no further increase in the ICL is possible. The mixed *GM* works similarly. In all the experiments involving CdSBM, the three optimization strategies are tested and the one leading to the highest ICL is retained. A pseudocode of the greedy search algorithm (strategy **A**) can be seen in Algorithm 2.

2.2.3 Non-parametric estimation of integrated intensities

This section focuses at first on dSBM and assumes that the pair (Z, K) is either known or estimated via the inference procedure described in the previous section. In such a framework, no hypothesis has been formulated about the shape of the functions $\{\Lambda_{kg}(\cdot)\}_{\{k,g\leq K\}}$. In the following we show how the tensor $\mathbf{\Delta\Lambda}$ can be estimated and how its estimates can be used to further estimate the integrated intensity functions $\{\Lambda_{kg}(\cdot)\}_{\{k,g\leq K\}}$. Over the considered time partition, $\mathbf{\Delta\Lambda}$ can be directly estimated by maximum likelihood (ML) from (2.9)

$$\log p(\boldsymbol{X}|Z, \mathbf{\Delta\Lambda}) = \sum_{k,g}\sum_{u}\left[S_{kgu}\log(\Delta\Lambda_{kg}^{I_u}) - |\mathcal{A}_k||\mathcal{A}_g|\Delta\Lambda_{kg}^{I_u}\right] + c,$$

where c regroups all terms in (2.9) not depending on $\mathbf{\Delta\Lambda}$. By taking the first order derivative with respect to $\Delta\Lambda_{kg}^{I_u}$ and setting it equal to zero, it follows that

$$\widehat{\Delta\Lambda}_{kg}^{I_u} = \frac{S_{kgu}}{|\mathcal{A}_k||\mathcal{A}_g|}, \qquad \forall(k,g), \tag{2.27}$$

where $\widehat{\Delta\Lambda}_{kg}^{I_u}$ denotes the ML estimator of $\Delta\Lambda_{kg}^{I_u} = \Lambda_{kg}(t_u) - \Lambda_{kg}(t_{u-1})$. Hence, $\Delta\Lambda_{kg}^{I_u}$ can be estimated by maximum likelihood as the total number of interactions on the sub-graph corresponding to the connections from cluster $\mathcal{A}_k$ to cluster $\mathcal{A}_g$, over the time interval I_u, divided by the number of potential binary connections on this sub-graph.

Once the tensor $\mathbf{\Delta\Lambda}$ estimated, a point-wise, non-parametric estimator of $\Lambda_{kg}(t_u)$ is defined by

$$\widehat{\Lambda}_{kg}(t_u) = \sum_{l=1}^{u}\widehat{\Delta\Lambda}_{kg}^{I_l}, \qquad \forall(k,g) \tag{2.28}$$

for all time points t_u in the user defined partition (2.3), recalling that $\Lambda_{kg}(0) = 0$. Thanks to the properties of the ML estimator, together with the linearity of (2.28), $\widehat{\Lambda}_{kg}(t_u)$ is known to be an unbiased and convergent estimator of $\Lambda_{kg}(t_u)$. Notice, however, that this is true only if the estimated Z is the true one.

Remark 2.8 *The maximum likelihood estimator in* (2.28) *can be viewed as an extension to random graphs and mixture models of the non-parametric estimator proposed in Leemis (1991). In that article, N-trajectories of independent NHPPs sharing the same intensity function are observed and the proposed estimator is obtained via method of moments.*

Algorithm 2: Greedy search algorithm (CdSBM, **Strategy A**)

Require:
A $(N \times N \times U)$ tensor X whose entry (i, j, u) is $X_{ij}^{I_u}$
An initial number of clusters K and time clusters D
Initial clustering algorithms $f(\cdot)$ (nodes) and $g(\cdot)$ (time intervals)

Initializations:
$Z \leftarrow f(X, K)$
$Y \leftarrow g(X, D)$
%% *Exchange - time intervals*
seq $\leftarrow$ shuffle$(\{1, \ldots, U\})$
while ICL increases **do**
  for idx in $1, \ldots, U$ **do**
    $u \leftarrow$ seq$[idx]$
    $d' \leftarrow Y[u]$
    bck $\leftarrow 0$ %% *benchmark*
    bs $\leftarrow$ (null, null) %% *best switch*
    for l in $1, \ldots, D$ **do**
      if $l == d'$ **then**
        break
      end if
      $\Delta_{d' \to l}^{E,T} \leftarrow$ TestSwitch(u, l)
      if $\Delta_{d' \to l}^{E,T} >$ bck **then**
        bck $\leftarrow \Delta_{d' \to l}^{E,T}$
        bs $\leftarrow (u, l)$
      end if
    end for
    if bck > 0 **then**
      DoSwitch(bs[1], bs[2])
      DoUpdate %% *Y, ICL, D and other statistics*
    end if
  end for
end while
%% *Merge - Time intevals*
while ICL increases **do**
  bck $\leftarrow 0$
  bm $\leftarrow$ (null, null) %% *best merge*
  for d in $1, \ldots, D-1$ **do**
    for l in $d+1, \ldots, D$ **do**
      $\Delta_{d \to l}^{M,T} \leftarrow$ TestMerge(d, l)
      if $\Delta_{d \to l}^{M,T} >$ bck **then**
        bck $\leftarrow \Delta_{d \to l}^{M,T}$
        bm $\leftarrow (d, l)$
      end if
    end for
    if bck > 0 **then**
      DoMerge(bm[1], bm[2])
      DoUpdate %% *Y, ICL, D and other statistics*
    end if
  end for
end while
%% *Exchange - Nodes*
...
%% *Merge - Nodes*
...

Ensure: Estimates of Z, Y, K and D.

Of course, (2.28) is only defined on the time points t_u in (2.3), whereas it would be desirable to have a an estimator of $\Lambda_{kg}(\cdot)$ over the whole time interval $[0, T]$. A straightforward approach (adopted in the experiments) is to consider the following piecewise linear estimator

$$\widehat{\Lambda}_{kg}(t) = \sum_{u=1}^{U} \left[\widehat{\Lambda}_{kg}(t_{u-1}) + \frac{\widehat{\Lambda}_{kg}(t_u) - \widehat{\Lambda}_{kg}(t_{u-1})}{t_u - t_{u-1}} (t - t_{u-1}) \right] \mathbf{1}_{[t_{u-1}, t_u[}(t), \tag{2.29}$$

defined for all $t \in [0, T]$, which is a linear interpolation of the estimators in (2.28). Notice, once more, that this is a consistent and unbiased estimator of $\Lambda_{kg}(t)$ at times $\{t_u\}_{u \leq U}$ only, even when the estimated Z is the actual one.

Remark 2.9 *In the context of dynamic SBMs with non-homogeneous Poisson counting processes, a very similar approach is independently developed by Matias et al. (2015) with two main differences*

1. *They focus on the estimation of the* instantaneous *intensity functions* $\{\lambda_{kg}(\cdot)\}_{k,g}$.
2. *The considered time partition is not fixed a priori but adaptively selected (via inference) at each step of the VEM algorithm that they use.*

When considering the CdSBM, instead of dSBM, (2.27) and (2.28) are replaced by

$$\widehat{\Delta\Lambda}_{kg}^{d} = \frac{S_{kgd}}{|\mathcal{A}_k||\mathcal{A}_g||\mathcal{C}_d|} \tag{2.30}$$

$$\widehat{\Lambda}_{kg}(t_u) = \sum_{l=1}^{u} \widehat{\Delta\Lambda}_{kg}^{Y_l}, \tag{2.31}$$

where the first equation is an immediate consequence of (2.14). Although the estimator in (2.29) can also be used in CdSBM, one point should be noted. In dSBM each interval I_u corresponds to a potentially different slope of the function $\widehat{\Lambda}_{kg}(\cdot)$. In contrast, in CdSBM only D different slopes are allowed, one for each time cluster.

Remark 2.10 *In CdSBM, the estimated integrated intensity functions defined in* (2.29) *have the same slope on the time intervals belonging to the same cluster.*

Due to this constraint, (2.29) cannot be defined "non-parametric" in case of CdSBM.

2.3 Experiments

This section focuses on experiments on both synthetic and real data. The greedy algorithm described in Section 2.2.2 was implemented in C++ and it is referred to as "greedy ICL" henceforth. A Euclidean hierarchical clustering was used to initialize Z (and Y when testing CdSBM) .

2.3.1 Simulated Data

First Scenario. We start by showing how dSBM can be used to efficiently estimate Z and K in frameworks where a static SBM fails. The dynamic graphs simulated in this section consist in 50 (N) nodes, grouped in two hidden clusters $\mathcal{A}_1$ and $\mathcal{A}_2$ and 100 (U) time intervals of unitary length. Two time clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ are considered, each one containing a certain number of time intervals $I_1, \dots I_U$. If I_u is in $\mathcal{C}_1$, $X_{ij}^{I_u}$ is drawn from a Poisson distribution $\mathcal{P}(P_{Z_iZ_j})$, otherwise from $\mathcal{P}(Q_{Z_iZ_j})$, where $P_{Z_iZ_i}$ (respectively $Q_{Z_iZ_j}$) is the element in position (Z_i, Z_j) in the matrix P (resp. Q). The matrices P and Q are given by

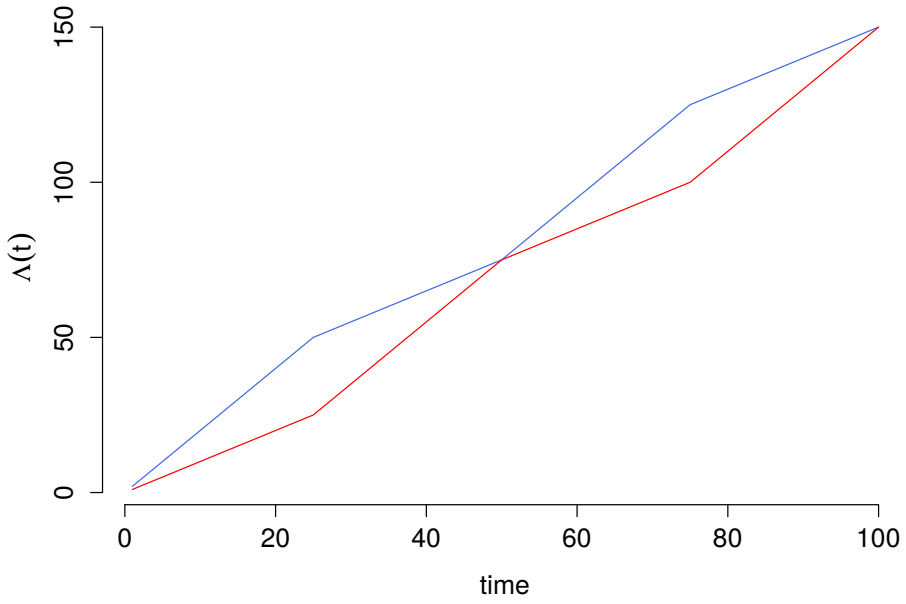
$$P = \begin{pmatrix} \psi & 1 \\ 1 & \psi \end{pmatrix} \qquad \text{and} \qquad Q = \begin{pmatrix} 1 & \psi \\ \psi & 1 \end{pmatrix},$$

where ψ is a free parameter in $[1, \infty)$. The setup $\psi = 1$ corresponds to the degenerate case in which all nodes belong to the same, unique cluster. During $\mathcal{C}_1$, higher values of $\psi > 1$ correspond to a stronger *community* structure. During $\mathcal{C}_2$, higher values of ψ correspond to a higher *non-assortative* structure, in which there are more between cluster interactions than within cluster interactions. In this section, ψ is set equal to 4 and the proportions to each group are set equal, namely $\pi = (1/2, 1/2)$. The number of time intervals assigned to each time cluster is assumed to be equal to $U/2$ and the following assignment is used

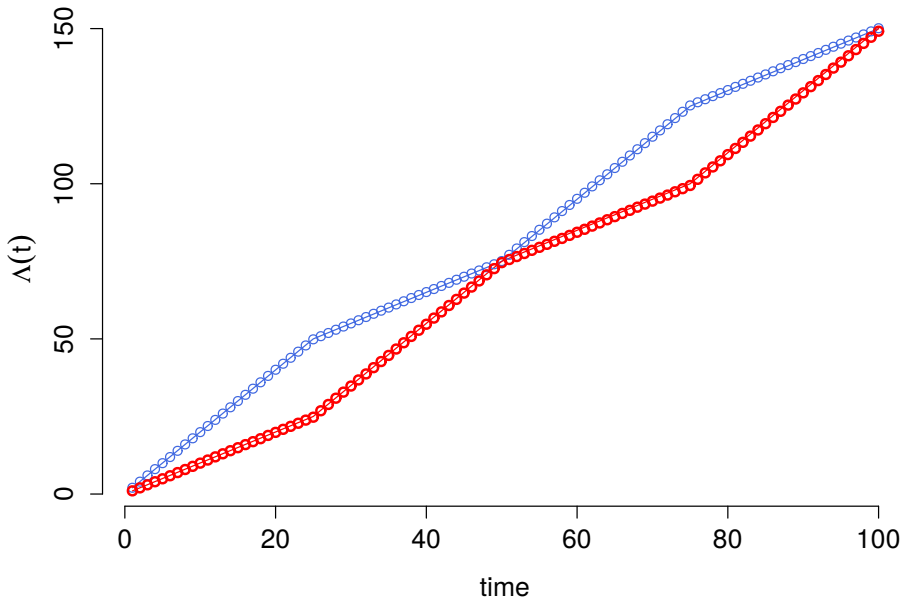
$$\begin{aligned} \mathcal{C}_1 :=& \{I_1, \dots, I_{25}\} \cup \{I_{51}, \dots, I_{75}\}, \\ \mathcal{C}_2 :=& \{I_{26}, \dots, I_{50}\} \cup \{I_{76}, \dots, I_{100}\}. \end{aligned}$$

This setup defines two integrated intensity functions (IIFs), called $\Lambda_1(\cdot)$ and $\Lambda_2(\cdot)$. The former is associated with the NHPPs counting interactions *within* clusters, the latter is associated with the NHPPs counting interactions *between* clusters. These IIFs can be observed in Figure 2.4a and, as explained in Section 2.2.3, they can be estimated non-parametrically once Z and K are known.

A tensor X, of dimension $N \times N \times U$, is simulated. Its entry (i, j, u) counts the number of interactions from node i to node j over the time interval I_u. The greedy ICL algorithm was run on X. In order to compare dSBM with a static SBM, we relied on the Gibbs sampling approach introduced by Nouedoui and Latouche (2013) to fit a static SBM with Poisson distributed edges. Henceforth, their approach is referred to as Poisson-SBM. Hence, the simulated interactions stored in X were aggregated over the whole time horizon $[0, 100]$ to obtain an $N \times N$ adjacency matrix. The Poisson-SBM was run on this adjacency matrix. The experiment was repeated 50 times and estimates of Z were provided at each iteration. Each estimate $\hat{Z}$ is compared with the true Z for both dSBM and SBM. Notice that, while the estimation procedure greedy ICL allows us to select the number of clusters K, this is not true for the Gibbs sampling of Nouedoui and Latouche (2013). Hence, SBM was provided with the true number of clusters, $K = 2$. Adjusted rand indexes (ARI, Rand 1971) were employed to asses the estimates. The ARI takes values between zero and one. An ARI equal to one means that the estimated Z coincides with the actual



(a)



(b)

Figure 2.4 – *Real 2.4a and estimated 2.4b integrated intensity functions (IIFs) according to the generative model in the first scenario ($\psi = 4$). In blue, the intensity function $\Lambda_1(\cdot)$ represents the mean number of interactions* within *clusters. In red, $\Lambda_2(\cdot)$ represents the mean number of interactions* between *clusters.*

one (up to label switching). An ARI equal to zero corresponds to an inconsistent clustering. While the true structure (Z, K) is always recovered by

	Firs Scenario: ARIs	
Model	ARI (Z)	ARI (Y)
CdSBM	0.98 (0.1414)	0.96 (0.1385)
dSBM	0 (0)	-
SBM	0 (0)	-

Table 2.1 – *Average ARIs for dSBM models and Poisson-SBM (standard deviations inside parenthesis). 50 dynamic graphs were simulated according to the first setup with* $N = 50$ *and* $U = 1000$*, SBM was provided with the true value of* $K = 2$*.*

dSBM and 50 unitary values of the ARI are obtained, SBM never succeeded in recovering any hidden structure present in the data and produced 50 null ARIs. This is not surprising, since the time clusters exhibit opposite interactivity patterns (community vs. non-assortative) which cancel each other out when aggregating interactions through time.

Relying on an efficient estimate of Z, the two integrated intensity functions $\Lambda_1(\cdot)$ and $\Lambda_2(\cdot)$ can be estimated via (2.29). The results can be observed in Figure 2.4b, where the estimated functions (coloured dots) overlap the real functions.

Over fitting. So far, we did not need to introduce the constrained model to recover the true values of Z and K. This is due to the reasonably lower value of U. This section shows how dSBM can no longer recover the true structure (Z, K) when the number of time intervals U (which is proportional to the number of free parameters) grows. The same setup of the previous paragraph is considered with a lower ψ

$$P = \begin{pmatrix} 1.4 & 1 \\ 1 & 1.4 \end{pmatrix} \qquad \text{and} \qquad Q = \begin{pmatrix} 1 & 1.4 \\ 1.4 & 1 \end{pmatrix}.$$

Despite a lower contrast (ψ reduces form 4 to 1.4) for $U = 100$ dSBM still estimates the true pair (Z, K) at each iteration (not reported). Consider now a finer partition of $[0, 100]$, obtained by setting $U = 1000$ and $\Delta_u = 0.1$. The intensity matrices Q and P are scaled coherently with Δ_u, leading to

$$P = \begin{pmatrix} 0.14 & 0.1 \\ 0.1 & 0.14 \end{pmatrix} \qquad \text{and} \qquad Q = \begin{pmatrix} 0.1 & 0.14 \\ 0.14 & 0.1 \end{pmatrix}.$$

The time cluster assignment now is

$$\mathcal{C}_1 := \{I_1, \dots, I_{250}\} \cup \{I_{501}, \dots, I_{750}\}$$
$$\mathcal{C}_2 := \{I_{251}, \dots, I_{500}\} \cup \{I_{751}, \dots, I_{1000}\}$$

According to this modified setup, 50 dynamic graphs were simulated over the interval $[0, 100]$. Notice that each unitary time interval now contains 10 graph snapshots. The tensor X associated which each dynamic graph has dimension $50 \times 50 \times 1000$.

The greedy ICL algorithm for both models dSBM and CdSBM was run on each simulated tensor X. The clustering results are reported in Table 2.1. Let us start with dSBM. It placed all nodes in the same, unique group. This leads to a null ARI for Z, at each iteration. As mentioned in Section 2.2.1, the ICL penalizes the number of parameters and since the tensor $\mathbf{\Delta\Lambda}$ has dimension $K \times K \times U$, for a fixed K when moving from the

larger decomposition ($U = 100$) to the finer one ($U = 1000$) the number of free parameters in the model is approximatively[2] multiplied by 10. The increase in the complete-data log-likelihood occurring when increasing the number of groups from $K = 1$ to $K = 2$ is not sufficient to compensate the penalty due to the higher number of parameters and hence the ICL decreases. Therefore, the maximum value of ICL is attained for $K = 1$ and a single cluster of nodes is detected. Model CdSBM allows to tackle this issue. When the integrated intensity functions $\Lambda_1(\cdot)$ and $\Lambda_2(\cdot)$ are constrained to have the same slope on time intervals belonging to the same time cluster (see remark 2.10), we basically reduce the third dimension of the tensor $\mathbf{\Delta\Lambda}$ from U (1000) to D (2). A hierarchical clustering algorithm was used to initialize the time labels Y, and the initial number of time clusters was set to $D_{max} = \sqrt{U}$. In an attempt to avoid convergence to local maxima, ten estimates are built for each tensor and the estimate leading to the best ICL is finally retained. The clustering results for CdSBM can be observed in the first line of Table 2.1. These results were obtained relying on the optimization strategy **A**. The other two strategies described in Section 2.2.2, namely **B** and **C**, led to similar results in terms of both final ICL and ARIs (not reported).

Second Scenario. This paragraph focuses on CdSBM only. The simulated dynamic graphs consist in 50 nodes, belonging to three clusters $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$. The interactions take place over 50 times intervals of unitary length, belonging to three time clusters (denoted $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$). Both node and time clusters are assumed to be balanced, on average, by fixing $\pi = \rho = (1/3, 1/3, 1/3)$.

A random tensor X, whose dimension is $N \times N \times U$ is simulated according to the following rule

$$X_{ij}^{I_u} \sim \mathcal{P}(P_{Z_i Z_j}(u))$$

where

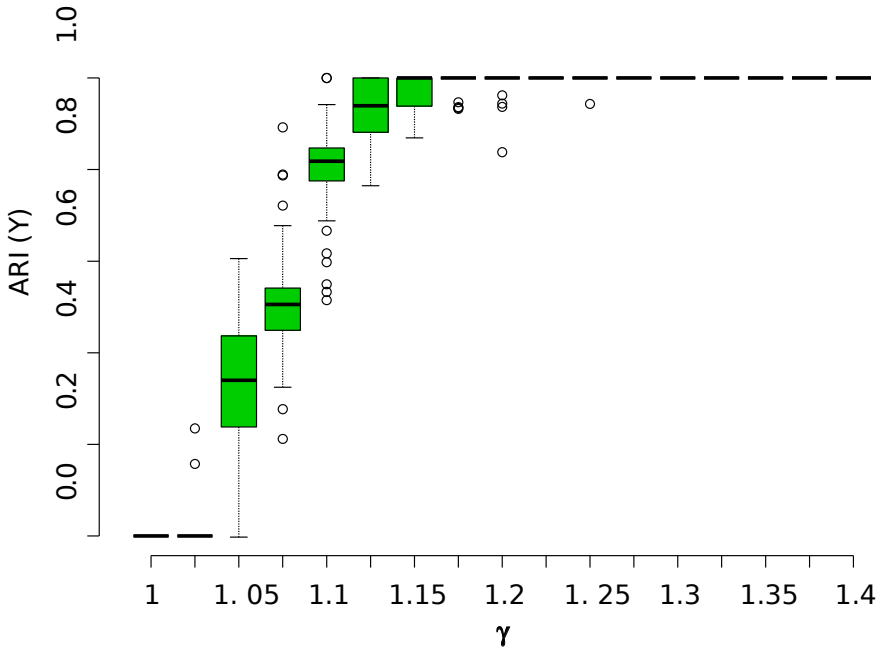
$$P(u) = Q\mathbf{1}_{\mathcal{C}_1}(u) + \sqrt{\gamma}Q\mathbf{1}_{\mathcal{C}_2}(u) + \gamma Q\mathbf{1}_{\mathcal{C}_3}(u), \quad u \in \{1, \dots, 50\} \tag{2.32}$$

and

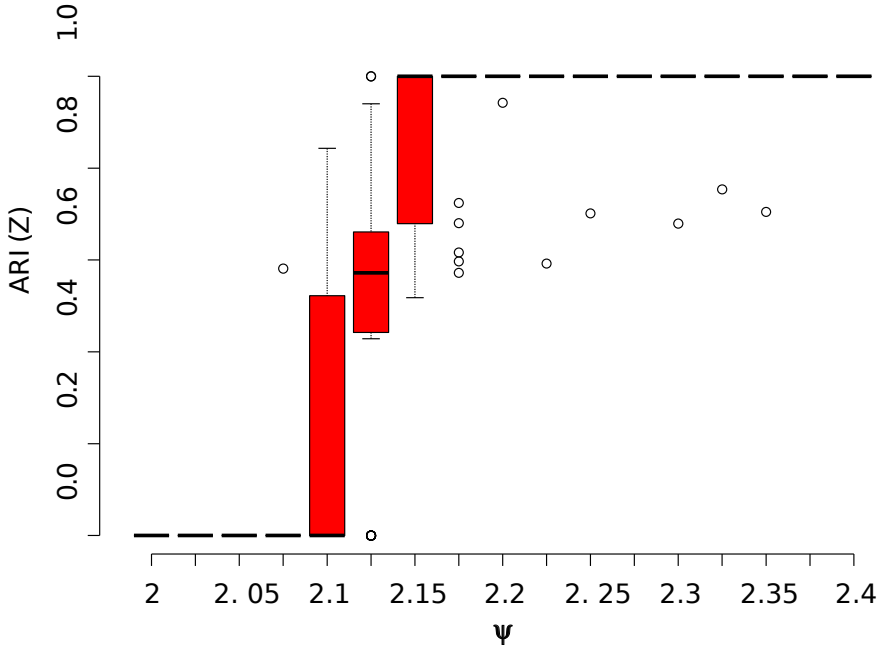
$$Q = \begin{pmatrix} \psi & 2 & 2 \\ 2 & \psi & 2 \\ 2 & 2 & \psi \end{pmatrix}.$$

Here, ψ is a free parameter in $[2, +\infty)$, $\gamma > 0$ and $\mathbf{1}_{\mathcal{C}}$ is the indicator function over the set $\mathcal{C}$. Hence, $P(u)$ is equal to Q when I_u belongs to $\mathcal{C}_1$, to $\sqrt{\gamma}Q$ when I_u belongs to $\mathcal{C}_2$ and to γQ when I_u belongs to $\mathcal{C}_3$. The simulated dynamic graphs are affected by a persistent community structure whereas the expected number of interactions differs from a time cluster to another. Both the community structure and the non-stationary behaviour can be more or less obvious based on the value of ψ and γ. This section does not consider the non-parametric estimation of the NHPP intensities and only inspects how the greedy ICL algorithm behaves for different values of the pair (ψ, γ). Hence, for a fixed value of the pair, 50 dynamic

[2]The dimension of the vector π does not change.



(a) *Time clustering ($\psi = 2$).*



(b) *Node clustering ($\gamma = 1$).*

Figure 2.5 – *Box plots of ARIs for both clusterings of nodes and time intervals (CdSBM). Both clusterings reach the maximum effectiveness for higher values of the contrast parameters.*

graphs were generated and estimates of the labels Z and Y were provided for each graph. The greedy search algorithm following the optimization strategy **A**, led to the best results (see next section for more details). In

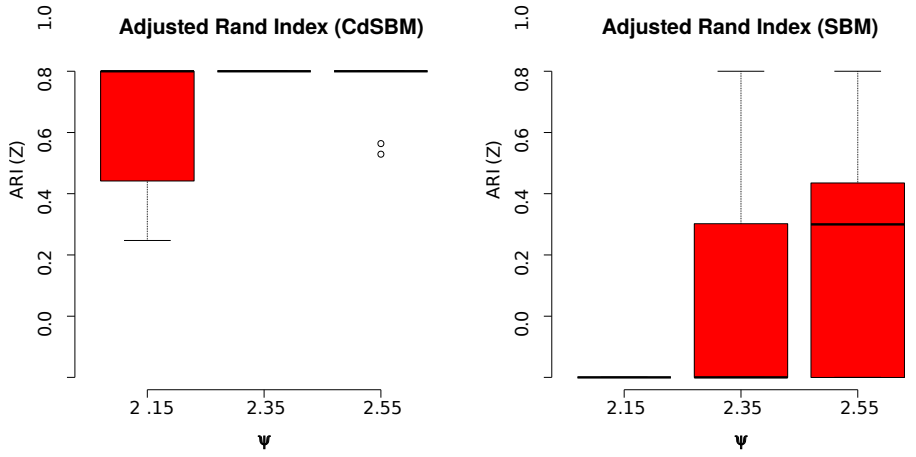


Figure 2.6 – *Comparison between CdSBM and SBM with Poisson links in a stationary framework (a single time cluster, $\gamma = 1$).*

order to avoid convergence to local maxima, ten estimates of labels are provided for each graph and the pair $(\hat{Z}, \hat{Y})$ leading to the highest ICL is finally retained. The experiments show that for sufficiently large values of ψ and γ, the true structure can always be recovered. This can be seen in detail for two special cases illustrated in Figure 2.5. In Figure 2.5a, γ varies in the range $[1, 1.05, \ldots, 1.4]$ and ψ is set equal to 2, corresponding to a single community (the Erdős-Rényi model). The setup $\gamma = 1$ corresponds to a degenerate case and no time structure affects the interaction frequencies. Hence it is not surprising that the algorithm assigns all the time intervals to the same cluster (null ARI). The higher the value of γ the more effective the clustering is up to a perfect recovery of the planted structure (ARI of 1). In particular the true time structure is fully recovered for all the fifty graphs when γ is higher than 1.3. Similar findings can be observed in Figure 2.5b, about node clustering, when $\gamma = 1$. In this case, any time structure is present and persistent communities are detected by the model as ψ increases. In this last setup, it is interesting to make a comparison with Poisson-SBM, which is expected to give similar results to those shown in Figure 2.5b. As done in the previous scenario, SBM was run on the adjacency matrix obtained by aggregating the simulated interactions. The experiment was repeated 50 times for each value of ψ in the set $\{2.15, 2.35, 2.55\}$. Figure 2.6 compares the ARIs produced by the two models. The greedy ICL for CdSBM recovers the true structure at levels of contrast lower than those required by the Gibbs sampling algorithm for Poisson-SBM. This comparison shows that, in a stationary framework, the dSBM model works at least as well as a static SBM. The difference in terms of performance between the two models in this context, is certainly due to the greedy search approach which is more effective than Gibbs sampling, as expected (Côme and Latouche 2015).

Optimization strategies As mentioned in the previous section, the optimization strategy **A** was more efficient than the two other strategies outlined in Section 2.2.2. This can also be seen in the following test. The pair (γ, ψ) is set to $(1, 2.15)$ and 50 dynamic graphs are simulated according

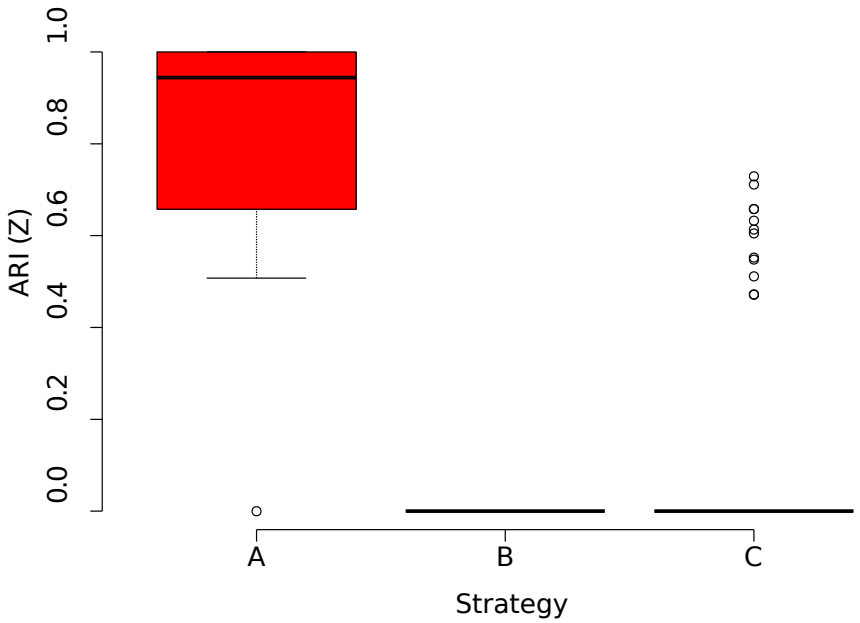


Figure 2.7 – *Adjusted Rand indexes for Z produced by CdSBM according to the three different optimization strategies outlined in Section 2.2.2.*

to the same settings discussed so far. Three different estimates of Z are obtained, one for each strategy, and ARIs for Z are computed. Results in Figure 2.7 can be compared with the mean value of the final ICL for each strategy, in Table 2.2.

	mean ICL
strategy A	-70845.64
strategy B	-70894.67
strategy C	-70885.22

Table 2.2 – *Mean values of the final ICL attained by different strategies.*

2.3.2 Real data

The dataset used in this section was collected during the **ACM Hypertext** conference held in Turin, June 29th - July 1st 2009. We focus on the first conference day (24 hours) and consider a dynamic graph with 113 (N) nodes (conference attendees) and 96 (U) time intervals (the consecutive quarter-hours in the period: 8am of June 29th - 7.59am of June 30th). The graph edges model the proximity face to face interactions between the conference attendees. An interaction is recorded when two attendees are face to face, nearer than 1.5 meters for a time period of at least 20 seconds. More information about the way the data were collected can be found in Isella et al. (2011) or by visiting the website `http://www.sociopatterns.org/datasets/hypertext-2009-dynamic-contact-network/`.

The data set we considered consists of several lines similar to the following one

*ID*1	*ID*2	*Time Interval* (15*m*)	*Number of interactions*
52	26	5	16

It means that conference attendees 52 and 26, between 9am and 9.15am, have spoken for $16 \times 20s \approx 5m30s$.

The initial number of groups K_{max} was set to 20 and Z was initialized randomly, namely each node was assigned to a cluster following a multinomial distribution. The greedy search algorithm for dSBM was run 10 times on the considered dataset, each time with a different initialization and estimates of Z and K were provided in 13.81 *seconds*, on average. The final values of the ICL can be observed as a box plot in Figure 2.8.

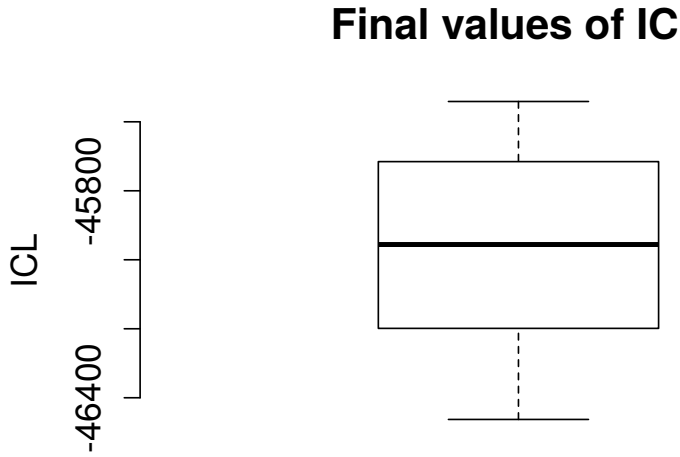
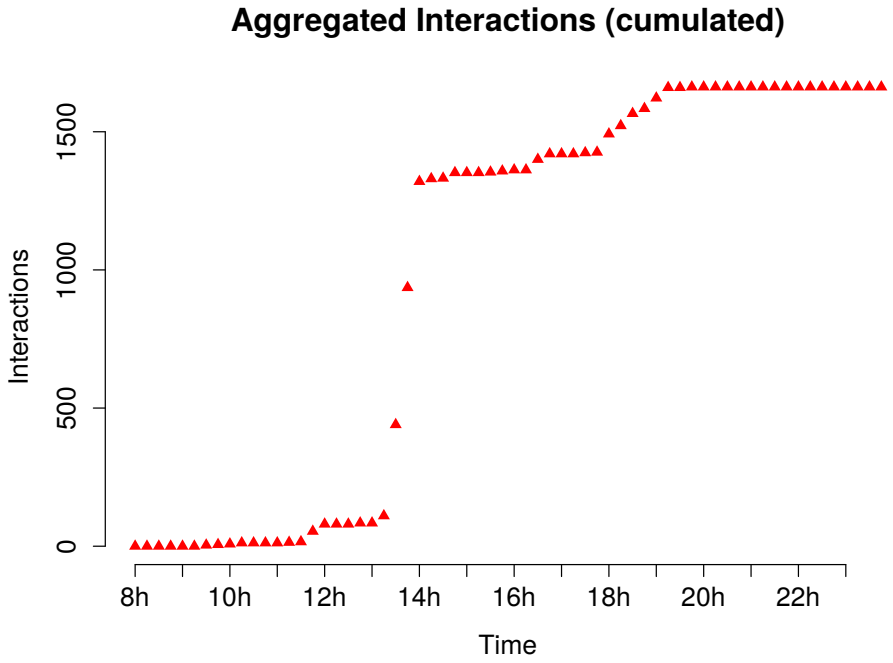


Figure 2.8 – *Box plot of the ten final values of the ICL produced by the greedy ICL algorithm for different initializations (dSBM).*

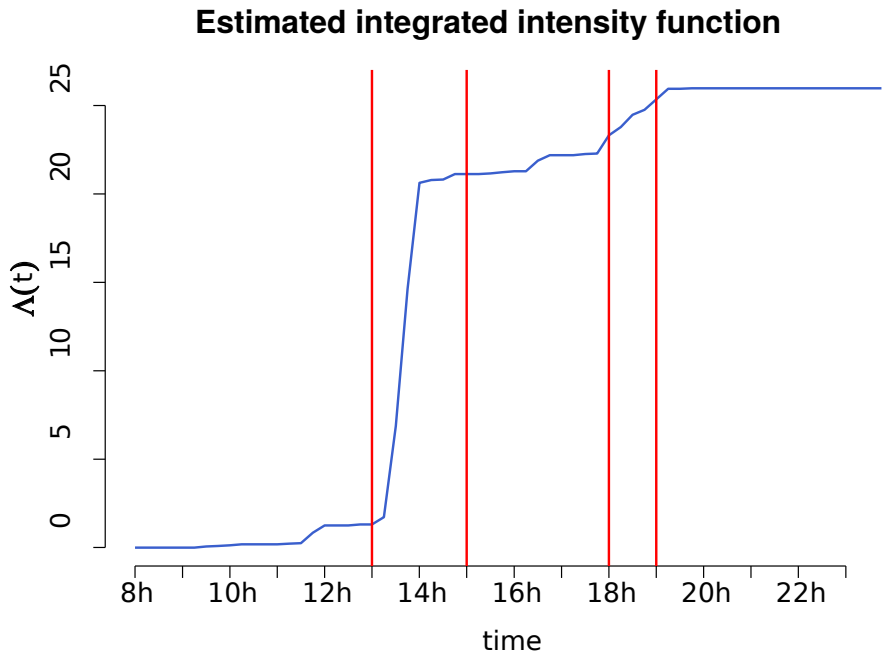
The estimates associated with the highest ICL correspond to 5 groups of nodes. Figure 2.9 focuses on cluster $\mathcal{A}_4$, containing 48 nodes. The top figure shows the time cumulated interactions within the cluster. As it can be seen the connectivity pattern for this cluster is very representative of the entire graph (see also Figure 2.10a): between 13pm and 14pm and 18pm and 19.30pm there are significant increases in the interaction intensity. The estimated integrated intensity function (IIF) for the interactions inside cluster $\mathcal{A}_4$ can be observed in Figure 2.9b. The function has a higher slope on those time intervals where attendees in the cluster are more likely to have interactions. The vertical red lines delimit two important times of social gathering[3]:

1. 13.00-15.00 - lunch break.
2. 18.00-19.00 - wine and cheese reception.

[3]More information at `http://www.ht2009.org/program.php`.



(a) *Cumulated aggregated connections within cluster $\mathcal{A}_4$.*



(b) *Estimated IIF for interactions within cluster $\mathcal{A}_4$*

Figure 2.9 – *in Figure 2.9a, cumulated aggregated connections for each time interval for cluster $\mathcal{A}_4$. In Figure 2.9b the estimated IIF for interactions inside cluster $\mathcal{A}_4$. Vertical red lines delimit the lunch break and the wine and cheese reception.*

The CdSBM can be used to assign time intervals on which interactions have similar instantaneous intensity to the same time cluster. The greedy ICL algorithm for CdSBM was run on the dataset by using the optimiza-

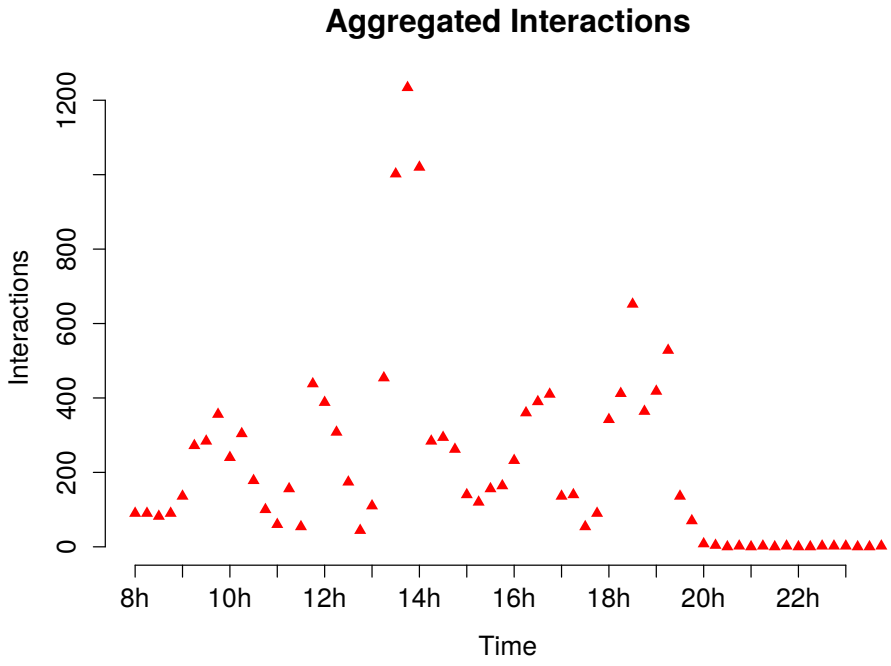
tion strategy **C** described at the end of Section 2.2.2 (other strategies lead in this case to similar results) and the initial number of clusters D_{max} was set equal to 20. The time clustering provided by the greedy ICL algorithm can be observed in Figure 2.10. The top figure shows the aggregated interactions in the whole network for each quarter-hour during the first day. In the bottom figure, interactions taking place in time intervals assigned to the same time cluster have the same shape/color. Two important things should be noticed:

1. The obtained clustering seems meaningful: the three time intervals corresponding to the highest interaction levels are placed in the same cluster (blue), apart from all the others. More in general, each cluster is associated with a certain intensity level, so time intervals in the same time cluster share the same global interactivity pattern.

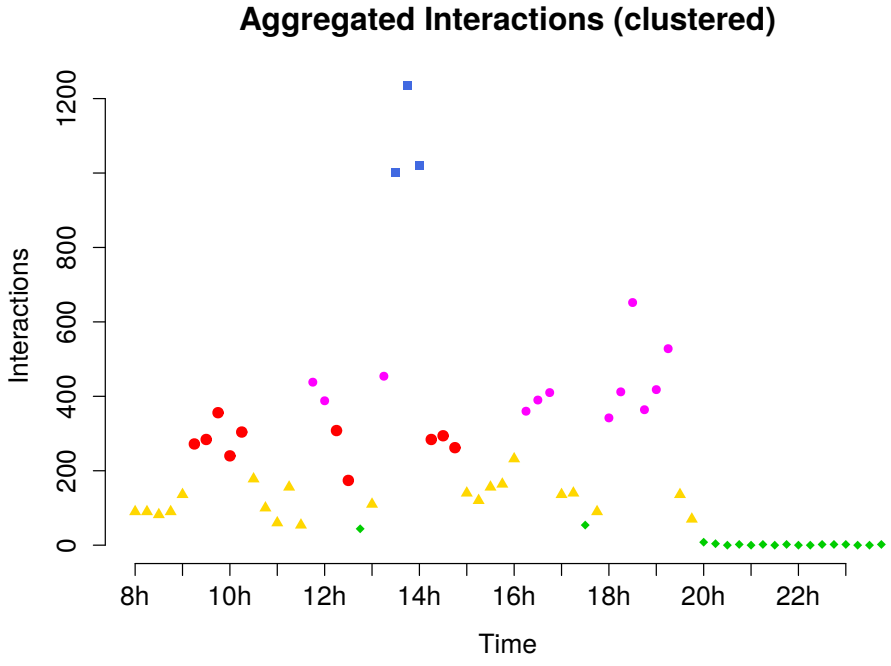
2. Once more, we recall that time clusters do not have to contain adjacent time intervals and this is one of the main differences between the approach considered in this chapter (*time clustering*) and the one adopted in the following chapter (*segmentation*).

Conclusion

This chapter introduced and detailed a dynamic extension of SBM (called dSBM) to cluster the nodes of a discrete time dynamic graph in scenarios where the static SBM fails. The chosen approach consists into partitioning the time horizon over which interactions are observed into sub-intervals of fixed length. Those intervals provide aggregated interaction counts that are increments of non homogeneous Poisson processes. To avoid overfitting problems, a constrained version of dSBM (CdSBM) is developed. CdSBM assigns the time intervals of the user defined partition in such a way that the time series of the interaction counts is stationary on each time cluster. The next chapter addresses two main topics. First, it extends the approach presented in this chapter to model continuous time dynamic graphs (see Definition 1.7). Then, it focuses on change point analysis for graph data.



(a) *Aggregated connections.*



(b) *Clustered time intervals.*

Figure 2.10 – *in Figure 2.10a, aggregated connections for each time interval for the whole network. In Figure 2.10b interactions of the same form/color take place on time intervals assigned to the same cluster (CdSBM).*

2.4 Appendix

2.4.1 Joint integrated probability of labels

Consider at first Z, following a multinomial distribution of parameter π. The vector π, of length K is assumed to follow a Dirichlet prior distribution

$$p(\pi|\alpha, K) = \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \pi_k^{\alpha-1}.$$

The joint probability for the pair (Z, π) is obtained by multiplying the above equation by (2.1)

$$p(Z, \pi|\alpha, K) = \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \pi_k^{|\mathcal{A}_k|+\alpha-1}.$$

This is still a Dirichlet probability density function of parameters $(|\mathcal{A}_1| + \alpha, \ldots, |\mathcal{A}_K| + \alpha)$ and integration with respect to π is straightforward

$$\begin{aligned}
p(Z|\alpha, K) =& \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \int_\pi \prod_{k=1}^{K} \pi_k^{|\mathcal{A}_k|+\alpha-1} d\pi, \\
=& \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \frac{\prod_{k\leq K} \Gamma(|\mathcal{A}_k| + \alpha)}{\Gamma(\sum_{k=1}^{K}(|\mathcal{A}_k| + \alpha))} \\
& \times \int_\pi \mathrm{Dir}(\pi; |\mathcal{A}_1| + \alpha, \ldots, |\mathcal{A}_K| + \alpha) d\pi, \\
=& \frac{\Gamma(\alpha K)}{\Gamma(\alpha)^K} \frac{\prod_{k\leq K} \Gamma(|\mathcal{A}_k| + \alpha)}{\Gamma(N + \alpha K)}.
\end{aligned}$$

This integrated probability corresponds to the first term on the right hand side of (2.24). The second term of (2.24) is obtained similarly and the joint probability distribution $p(Z, Y|K, D)$ follows by independence.

2.4.2 Computational complexity of the greedy search

To evaluate the computational complexity of the proposed algorithm, we assume that the gamma function can be computed in constant time (see Press et al. 2007). The core computational task consists in evaluating the change in ICL induced by exchanges and merges. The main quantities involved in those computations are the $(L_{kgd})_{1\leq k\leq g, 1\leq d\leq D}$. The next paragraph details how to handle those quantities and the following one analyses the cost of the exchange and merge operations.

Data structures. To keep formulas uncluttered, the following additional notation is introduced

$$P_{kgd} := |\mathcal{A}_k||\mathcal{A}_g||\mathcal{C}_d|.$$

The quantities $(L_{kgd})_{1\leq k\leq g, 1\leq d\leq D}$ are stored in a three dimensional array that is never resized (it occupies a $\mathcal{O}(K_{max}^2 D_{max})$ memory space) so that at any time during the algorithm, accessing to a value or modifying it can be done in constant time. The needed quantities to compute an

increase/decrease of the ICL are L_{kgd}, S_{kgd}, P_{kgd} and R_{kgd} and they are handled in a similar way.

In addition, aggregated interaction counts are maintained and updated for each time interval and each node. For instance, for a time interval I_u the following statistics is considered

$$S_{kgu} := \sum_{Z_i=k} \sum_{Z_j=g} X_{ij}^{I_u},$$

and similar quantities such as P_{kgu}. In a similar manner, for a node i

$$S_{igd} := \sum_{c_j=g} \sum_{y_u=d} X_{ij}^{I_u}$$

and other related quantities are stored. The memory occupied by those structures is in $\mathcal{O}(N^2U)$. Cluster memberships and clusters sizes are also stored in arrays.

In order to evaluate the ICL change induced by an operation, we need to compute its effect on L_{kgd} to obtain L^*_{kgd}. This can be done in constant time for one value. For instance moving time interval I_u from $C_{d'}$ to C_l implies the following modifications:

1. $S_{kgd'}$ is reduced by S_{kgu} while S_{kgl} is increased by the same quantity;
2. $P_{kgd'}$ is divided by P_{kgu} while P_{kgl} is multiplied by the same quantity;
3. $R_{kgd'}$ is decreased by $|\mathcal{A}_k||\mathcal{A}_g|$ (or $|\mathcal{A}_k|(|\mathcal{A}_k|-1)$) while R_{kgl} is increased by the same quantity.

When an exchange or a merge is actually implemented, all the data structures are updated. The update cost is dominated by the other phases of the algorithm. For instance when I_u is moved from d' to l, the following updates are needed:

1. cluster memberships and cluster sizes, which is done in $\mathcal{O}(1)$;
2. $L_{kgd'}$ and L_{kgl} for all k and g, which is done in $\mathcal{O}(K^2)$;
3. aggregated counts and products, such as $S_{igd'}$ and S_{igl}, which is done in $\mathcal{O}(NKD)$.

Considering that $K \leq N$ and $D \leq U$, the total update cost is in $\mathcal{O}(NKD)$ for time interval related operations and in $\mathcal{O}(UK^2)$ for node related operations.

Exchanges. The calculation of $\Delta^{E,T}_{d'\to l}$, for a time interval cluster exchange, from (2.25) involves a sum with K^2 terms. As explained above each term is obtained in constant time, thus the total computational time is in $\mathcal{O}(K^2)$. This has to be evaluated for all time clusters and for all time intervals, inducing a total cost of $\mathcal{O}(UDK^2)$.

Similarly, the calculation of $\Delta^{E,V}_{d'\to l}$ involves a fix number of sums with at most KD terms in each sum. The total computational time is therefore in $\mathcal{O}(KD)$. This had to be evaluated for each node and for all node clusters inducing a total cost of $\mathcal{O}(NK^2D)$.

Notice that the total cost evaluated so far is the one of a full exchange round where all time intervals (or all nodes) are considered once. This evaluation does not take into account the reduction in the number of clusters generally induced by exchanges.

Merges. Merges are very similar to exchanges in terms of computational complexity. They involve comparable sums that can be computed efficiently using the data structures described above. The computational cost for one time cluster merge round is in $\mathcal{O}(D^2K^2)$ while it is in $\mathcal{O}(K^3D)$ for node clusters.

Total cost. The worst case complexity of one full exchange phase (with each node and each time interval considered once) is $\mathcal{O}((N + U)D_{max}K_{max}^2)$. The worst case complexity of one merge with mixed GM is $\mathcal{O}(D_{max}K_{max}^2(D_{max} + K_{max}))$ which is smaller than the previous one for $N \geq K_{max}$ and $U \geq D_{max}$. Thus the worst case complexity of one "iteration" of the algorithm is $\mathcal{O}((N + U)D_{max}K_{max}^2)$.

Unfortunately, the actual complexity of the algorithm, while obviously related to this quantity, is difficult to evaluate for two reasons. First, there is no way to estimate the number of exchanges needed in the exchange phase (apart from bounding them with the number of possible partitions). Secondly, in practice we observed that exchanges reduce the number of clusters, especially when D_{max} and K_{max} are high (i.e. close to U and N, respectively). Thus the actual cost of one individual exchange reduces very quickly during the first exchange phase leading to a vast overestimation of its cost using the proposed bounds. As a consequence, the merge phase is also quicker than evaluated by the bounds.

A practical evaluation of the behaviour of the algorithm, while outside the scope of this thesis, would be very interesting to assess it potential use on large data sets.

Multiple change point detection in dynamic graphs

3

As explained in Section 1.1, by adopting a continuous time point of view a sequence of time stamped interactions in continuous time can be seen as a dynamic graph. This definition is alternative to the one employed in the previous chapter and does not require any aggregation of the data. The approach outlined in this chapter aims to uncover hidden node groups in *continuous time* dynamic graphs, based on a SBM-like generative model. As in the previous chapter, node groups are not allowed to change in time. However, the interaction intensities between groups are assumed to display abrupt changes (a.k.a. "change points") whose number and location must be inferred from the data. Hence, we develop a VEM algorithm (introduced in Section 1.3.1 for SBM) to estimate the model parameters, the number of clusters and their content and the number and locations of the change points. As it will be seen, the change point detection is part of the maximization step of the algorithm. We show that the pruned exact linear time method (PELT, Killick et al. 2012), originally developed for univariate time series change point detection, can be adapted to perform change point detection in dynamic graphs.

3.1 A generative model for continuous time dynamic graphs

As detailed in Section 1.1.2, the finite set

$$\mathcal{D} := \{(i_m, j_m, \nu_m)\}_{1 \leq m \leq M},$$

where ν_m denotes the m-th interaction time and (i_m, j_m) is the pair of nodes actually interacting at time ν_m defines a continuous time dynamic graph with N interacting nodes. Here, the interaction times are denoted by the Greek letter ν (and no longer by t) since they are seen as random times, from a generative point of view. In this chapter, self interactions are not considered and the pair (i_u, j_u) is assumed to be not ordered (undirected graph). The illustrated approach can easily be extended to deal with directed graphs. The time period under study is the interval $[0, T]$ and M is the total number of interactions occurring up to time T. Without loss of generality, $\mathcal{D}$ can be sorted with respect to the time variable and instantaneous interaction times (see Remark 1.3) are assumed to be unique

$$0 = \nu_0 < \nu_1 < \cdots < \nu_M < T.$$

The interaction time ν_{M+1} is not observed before time T.

3.1.1 Time-stamped interactions as point processes

Let us consider two fixed nodes, i and j. We denote $M^{(i,j)}$ the number of edges between i and j, namely the number of distinct times ν such that $(i, j, \nu) \in \mathcal{D}$. Without loss of generality, those interaction times can be sorted into the following list[1]

$$\mathcal{A}^{(i,j)} := \{\nu_1^{(i,j)}, \ldots, \nu_{M^{(i,j)}}^{(i,j)}\}, \tag{3.1}$$

with $\nu_1^{(i,j)} < \nu_2^{(i,j)} < \cdots < \nu_{M^{(i,j)}}^{(i,j)}$. In probabilistic terms, $\mathcal{A}^{(i,j)}$ can be seen as a point process. As such a point process takes values in $[0, T]$, it is naturally associated to a counting process $\{M^{(i,j)}(t)\}_{t \in [0,T]}$. The random variable $M^{(i,j)}(t)$ counts the number of interactions, between i and j, that happened before (or exactly at) t, i.e.

$$M^{(i,j)}(t) = \left| \mathcal{A}^{(i,j)} \cap]0, t] \right|,$$

where $|S|$ denotes the cardinal of the set S.

As we saw in the previous chapter, a simple yet flexible generative model to generate the interaction times in $\mathcal{A}^{(i,j)}$ is the non-homogeneous Poisson process (NHPP)[2]. This chapter, however, does not only focuses on the increments of such process on a predefined grid, but it looks at NHPPs in more details. Hence, the process associated with the pair (i, j)

[1] In the previous chapter, $\mathcal{A}_k$ denoted the k-th cluster of nodes. This notation is no longer used in this chapter and the letter $\mathcal{A}$ is uniquely involved in (3.1).

[2] As detailed in Section 1.4.1, when saying non-homogeneous Poisson process, we refer to the *counting* process.

is characterized by an intensity function $\kappa^{(i,j)}(\cdot)$, positive and integrable on $[0,T]$. Denoting

$$\bar{\kappa}^{(i,j)}(t) = \int_0^t \kappa^{(i,j)}(s)ds, \qquad t \leq T,$$

assuming that $\mathcal{A}^{(i,j)}$ is associated with a NHPP with intensity function $\kappa^{(i,j)}$ means that for all $t \in [0,T]$, $M^{(i,j)}(t)$ follows a Poisson distribution with parameter $\bar{\kappa}^{(i,j)}(t)$.

As in the previous chapter, the proposed model adopts a block modelling perspective. Hence, Z still denotes a set of cardinality N, whose elements are independent hidden random variables following a multinomial distribution of parameter π

$$\mathbb{P}(Z_i = k) = \pi_k, \qquad \forall k \in \{1, \ldots, K\}$$

and $\sum_{k=1}^K \pi_k = 1$. In the reminder of this chapter, an equivalent 0-1 notation will be used interchangeably for Z. Hence, Z also denotes a $N \times K$ matrix, whose line is the vector $Z_i = (Z_{i1}, \ldots, Z_{iK})$ and $Z_{ik} = 1$ if and only if the i-th node is in the k-th cluster, zero otherwise. We stress that the pair (Z, K) is unknown and fixed in time.

Then, the following assumptions hold:

1. given Z, for all $i > j$ the interaction times in $\mathcal{A}^{(i,j)}$ are counted by $N(N-1)/2$ independent non-homogeneous Poisson processes with intensity functions $\{\kappa^{(i,j)}(\cdot)\}_{i>j}$;
2. there are $K(K+1)/2$ positive integrable functions $\boldsymbol{\lambda} = \{\lambda_{kg}(\cdot)\}_{k,g}$ defined on $[0,T]$ such that $\kappa^{(i,j)}(t) = \lambda_{Z_i Z_j}(t)$ for all $t \in [0,T]$.

With those assumptions, the conditional likelihood of a set of interactions $\mathcal{A}^{(i,j)}$ between two nodes i and j is given by (Proposition 1.1)

$$p(\mathcal{A}^{(i,j)} | Z_i = k, Z_j = g, \lambda_{kg}) = \prod_{m=1}^{M^{(i,j)}} \lambda_{kg}(\nu_m^{(i,j)}) \exp\left(-\Lambda_{kg}(T)\right), \qquad (3.2)$$

with

$$\Lambda_{kg}(t) := \int_0^t \lambda_{kg}(s)ds.$$

Remark 3.1 *Notice that as the interaction times in $\mathcal{A}^{(i,j)}$ are assumed to be counted by a NH Poisson process, all the interaction times are distinct (almost surely). This justifies the assumption used at the beginning of the present section.*

The data set $\mathcal{D}$ can be seen as the union of all the $\mathcal{A}^{(i,j)}$, with the added information of the interacting pairs at each interaction time: the pair (i_m, j_m) corresponds to the nodes actually having an interaction at time ν_m. Combining (3.2) applied to all the NHPPs with the conditional independence assumption between them, we obtain the following complete data likelihood

$$\begin{aligned} p(\mathcal{D}, Z | \boldsymbol{\lambda}, \pi) &= p(\mathcal{D} | Z, \boldsymbol{\lambda}) p(Z | \pi) \\ &= \exp\left(-\sum_{j>i}^{N} \Lambda_{Z_i Z_j}(T)\right) \prod_{m=1}^{M} \lambda_{Z_{i_m} Z_{j_m}}(\nu_m) \prod_{i=1}^{N} \pi_{Z_i}. \end{aligned} \qquad (3.3)$$

Once more, it is worth stressing that no aggregation of data was employed so far.

3.1.2 Modelling the intensity functions

The generative model introduced in the previous section does not make any assumption on the shape of the intensity functions $\boldsymbol{\lambda}(\cdot)$. As explained in Section 2.1.2, some restrictions on $\boldsymbol{\lambda}(\cdot)$ are needed to avoid over-fitting problems. Moreover, we look for a model being able to emphasize abrupt changes in the way existing clusters interact with each other. To account for these issues, we assume that the intensity functions of the NHPPs are piecewise constant and that the D time intervals on which they are constant are shared between *all* functions. Hence, we assume that there are $D-1$ discontinuity points

$$0 = \eta_0 < \eta_1 < \cdots < \eta_{D-1} < \eta_D = T, \tag{3.4}$$

such that for all $1 \leq d \leq D$, $1 \leq k \leq K$ and $1 \leq g \leq K$, $\lambda_{kg}(\cdot)$ is constant on $[\eta_{d-1}, \eta_d[$. Therefore

$$\lambda_{kg}(t) = \sum_{d=1}^{D} \lambda_{kgd} \mathbf{1}_{[\eta_{d-1}\eta_d[}(t), \qquad \forall k, g \in \{1, \ldots, K\}, \tag{3.5}$$

where $\lambda_{kgd} := \lambda_{kg}(\eta_{d-1})$ and $\mathbf{1}_{\mathcal{G}}(.)$ is the indicator function over a set $\mathcal{G}$[3]. In the following, $\eta = \{\eta_1, \ldots, \eta_{D-1}\}$ denotes the set of discontinuity points and $\boldsymbol{\lambda}$ is the $(K \times K \times D)$ tensor[4] with elements λ_{kgd}.

A crucial consequence of the assumption (3.5) is that on an interval $[\eta_{d-1}, \eta_d[$, all the Poisson processes are homogeneous and thus the graph does not exhibit any temporal structure. On the contrary, the intensity functions are allowed to change arbitrarily from one interval to the next one, accounting for abrupt changes in the interaction patterns between clusters. Therefore, a discontinuity point η_d corresponds to a sudden change in the graph structure. For this reason, the discontinuity points in η are called "change points" henceforth.

Taking into account (3.5) allows us to simplify the complete data log-likelihood as shown in the following proposition.

Proposition 3.1 *Using the constraint* (3.5)*, the complete data log-likelihood becomes*

$$\begin{aligned}\log p(\mathcal{D}, Z|\theta) = \\ - \sum_{d=1}^{D} \sum_{k,g}^{K} \left[\lambda_{kgd} \Delta_d \left(\sum_{j>i}^{N} Z_{ik} Z_{jg} \right) - \log(\lambda_{kgd}) \left(\sum_{j>i}^{N} Z_{ik} Z_{jg} X_{ij}^{(d)} \right) \right] \\ + \sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} \log \pi_k, \end{aligned} \tag{3.6}$$

where

1. $\theta := \{\eta, \boldsymbol{\lambda}, \pi\}$

[3]Since the whole time interval $[0, T]$ is considered, the intensity functions can be assumed (exceptionally) left continuous in $t = T$, i.e. $\lambda_{kg}(T) = \lambda_{kg}(\eta_{D-1})$.

[4]We use the same notation to denote the set of $K(K+1)/2$ intensity functions and the tensor because under our assumptions they correspond to two different views of the same object. Notice that the frontal slices of $\boldsymbol{\lambda}$ are symmetric $K \times K$ matrices since we are dealing with undirected graphs.

2. Δ_d *is the size of the interval* $[\eta_{d-1}, \eta_d[$,

3. $X_{ij}^{(d)} := M^{(i,j)}(\eta_d) - M^{(i,j)}(\eta_{d-1})$ *is the increment of the process* $\{M^{(i,j)}(t)\}_{t\in[0,T]}$ *over the segment* $[\eta_{d-1}, \eta_d[$, i.e. *the number of interactions that occurred between i and j during the time interval* $[\eta_{d-1}, \eta_d[$.

Proof. See appendix 3.5.1. □

3.2 Estimation

This section focuses on the inference of the model proposed above. This involves estimating the number of clusters (K), the number of segments (D) and the model parameters θ. Therefore, we introduce first a penalized likelihood criterion to maximize for model selection. A variational approximation for this criterion is then adopted, leading to a variational expectation maximization (VEM) algorithm, introduced in Section 1.3.1 for static SBM. It is finally shown how to integrate an efficient change point detection algorithm in the maximization step of VEM to estimate the piecewise constant intensities.

3.2.1 Penalized likelihood

Given the set of all the observed interactions $\mathcal{D}$, our goal is to estimate the number K of node clusters and their content Z. Similarly we must estimate the number D of change points as well as their location η.

A natural quality measure in this context is the observed data (integrated) log-likelihood

$$\begin{aligned}\log p(\mathcal{D}|K,\eta,D) &= \log\left(\int_{\lambda,\pi} p(\mathcal{D},\boldsymbol{\lambda},\pi|K,\eta,D)d\boldsymbol{\lambda}d\pi\right)\\ &= \log\left(\int_{\lambda,\pi} p(\mathcal{D}|\boldsymbol{\lambda},\pi,K,\eta,D)p(\boldsymbol{\lambda},\pi)d\boldsymbol{\lambda}d\pi\right),\end{aligned} \tag{3.7}$$

where $p(\boldsymbol{\lambda},\nu)$ is any prior distribution over the pair $(\boldsymbol{\lambda},\nu)$. Unfortunately this marginal log-likelihood does not have an analytical form, so we propose to replace it with a penalized log-likelihood (BIC-like) term

$$\max_{\boldsymbol{\lambda},\pi} \log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi) - \frac{1}{2}C(K,D)\log\alpha, \tag{3.8}$$

where

$$C(K,D) := K - 1 + \frac{K(K+1)D}{2},$$

accounts for the number of model parameters. The term α in (3.8) is related to the number of observations and will be discussed in Section 3.2.4. We consider the following optimization problem for inference

$$\max_{K,\eta,D,\boldsymbol{\lambda},\pi}\left[\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi) - \frac{1}{2}C(K,D)\log\alpha\right]. \tag{3.9}$$

While this allows us to avoid to consider directly the data (integrated) log-likelihood, two difficulties remain: $\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi)$ is not directly calculable and the optimization over η and D is a complex non convex problem. This issues are tackled as following: a variational approach

is used in order to derive a tractable lower bound of $\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi)$ (see Sections 3.2.2 and 3.2.3) while a change point detection technique is considered to address the optimization over η and D (see Section 3.2.4).

3.2.2 A variational bound

The first difficulty mentioned above is that computing the log-likelihood term $\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi)$ is not feasible. Indeed, it involves summing over all the K^N possible outcomes of the set Z, i.e.

$$\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi) = \log \left(\sum_{\mathbf{z}} p(\mathcal{D}, \mathbf{z}|K, \eta, D, \boldsymbol{\lambda}, \pi) \right),$$

where $\mathbf{z}$ denotes an outcome of Z. Moreover, $p(Z|\mathcal{D}, K, \eta, D, \boldsymbol{\lambda}, \pi)$ cannot be factorized so the standard EM algorithm cannot be considered for inference. This point was discussed for the static SBM in Section 1.3.1. For more details, see also Daudin et al. (2008).

Therefore, we introduce an approximate distribution $q(Z)$ for Z and use a standard variational decomposition

$$\begin{aligned} \log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi) = \\ \mathcal{L}(q; K, \eta, D, \boldsymbol{\lambda}, \pi) + KL(q(\cdot)||p(\cdot|\mathcal{D}, K, \eta, D, \boldsymbol{\lambda}, \pi)), \end{aligned}$$

where

$$\begin{aligned} \mathcal{L}(q; K, \eta, D, \boldsymbol{\lambda}, \pi) &= \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathcal{D}, \mathbf{z}|K, \eta, D, \boldsymbol{\lambda}, \pi)}{q(\mathbf{z})} \\ &= \mathrm{E}_q \left[\log \frac{p(\mathcal{D}, Z|K, \eta, D, \boldsymbol{\lambda}, \pi)}{q(Z)} \right], \end{aligned}$$

and KL denotes the Kullback-Leibler divergence between the true and approximate posterior distribution $q(\cdot)$ of Z, given the data and model parameters

$$\begin{aligned} KL(q(\cdot)||p(\cdot|\mathcal{D}, K, \eta, D, \boldsymbol{\lambda}, \pi)) &= -\sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathcal{D}, K, \eta, D, \boldsymbol{\lambda}, \pi)}{q(\mathbf{z})} \\ &= -\mathrm{E}_q \left[\log \frac{p(Z|\mathcal{D}, K, \eta, D, \boldsymbol{\lambda}, \pi)}{q(Z)} \right]. \end{aligned}$$

In the above equations, E_q denotes the expectation taken with respect to the distribution $q(\cdot)$.

Since $\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi)$ does not depend on the distribution $q(\cdot)$, maximizing $\mathcal{L}$ with respect to $q(\cdot)$ is equivalent to minimizing the KL divergence (over $q(\cdot)$ also).

As the KL divergence is non negative, $\mathcal{L}(q; K, \eta, D, \boldsymbol{\lambda}, \pi)$ is obviously a lower bound of $\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi)$ for all $q(\cdot)$, thus we use the standard variational bounding method. In our case, it consists in replacing the problem (3.9) by

$$\max_{K, \eta, D, \boldsymbol{\lambda}, \pi, q(\cdot)} f(q(\cdot), K, \eta, D, \boldsymbol{\lambda}, \pi), \tag{3.10}$$

where

$$f\left(q(\cdot), K, \eta, D, \boldsymbol{\lambda}, \pi\right) = \mathcal{L}\left(q; K, \eta, D, \boldsymbol{\lambda}, \pi\right) - \frac{1}{2} C(K, D) \log \alpha. \tag{3.11}$$

As in Section 1.3.1, the posterior Z distribution, is approximated by a factorizing distribution

$$q(Z) = \prod_{i=1}^{N} q(Z_i) = \prod_{i=1}^{N} \prod_{k=1}^{K} \tau_{ik}^{Z_{ik}}, \tag{3.12}$$

where $\tau_{ik} \geq 0$, for all k and

$$\sum_{k=1}^{K} \tau_{ik} = 1.$$

This leads to the following expression for $\mathcal{L}$.

Proposition 3.2 *If $q(\cdot)$ is of the form* (3.12)*, then*

$$\begin{aligned}\mathcal{L}\left(q; K, \eta, D, \boldsymbol{\lambda}, \pi\right) = \\ - \sum_{d=1}^{D} \sum_{k,g}^{K} \left[\lambda_{kgd} \Delta_d \left(\sum_{j>i}^{N} \tau_{ik} \tau_{jg} \right) - \log(\lambda_{kgd}) \left(\sum_{j>i}^{N} \tau_{ik} \tau_{jg} X_{ij}^{(d)} \right) \right] \\ + \sum_{i=1}^{N} \sum_{k=1}^{K} \tau_{ik} \log \frac{\pi_k}{\tau_{ik}}. \end{aligned} \tag{3.13}$$

Proof. This expression can easily be obtained by taking the expectation with respect to $q(\cdot)$ of the log-likelihood in (3.6) and by adding the following entropy term

$$\begin{aligned}\mathcal{H}(q) : &= -\mathrm{E}_q[\log q(Z)] = \\ &= -\mathrm{E}_q \left[\sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} \log(\tau_{ik}) \right] = \sum_{i=1}^{N} \sum_{k=1}^{K} \tau_{ik} \log(\tau_{ik}).\end{aligned}$$

□

3.2.3 Variational expectation maximization

The problem (3.10) is solved relying on a VEM algorithm which optimizes the function $f(q(\cdot); K, \eta, D, \lambda, \pi)$ with respect to η, D, $\boldsymbol{\lambda}$, π, and with respect to $q(\cdot)$, alternately. In contrast, the number K of clusters is considered fixed in the present section. Section 3.2.5 details how the selection of K is handled.

Given η, D, $\boldsymbol{\lambda}$ and π, the optimization with respect to $q(\cdot)$ is straightforward. This corresponds to the E step of the algorithm and it is illustrated in the next section. Then, given $q(\cdot)$, the parameters $\boldsymbol{\lambda}$ as well as π are optimized away, and we use a change point detection procedure to maximize the criterion with respect to η and D. The optimization with respect to $\eta, D, \boldsymbol{\lambda}$ and π corresponds to the M step of the algorithm. The E and M steps are then iterated until convergence.

Maximization with respect to τ (E step)

The E step is based on the following proposition.

Proposition 3.3 *A first order condition for $f(q(\cdot), K, \eta, D, \boldsymbol{\lambda}, \pi)$ to be maximal with respect to $q(\cdot)$ in (3.12) is*

$$\tau_{ik} = \frac{\pi_k}{C} \exp\left\{ -\sum_{d=1}^{D}\sum_{g=1}^{K}\left[\lambda_{kgd}\Delta_d\left(\sum_{j\neq i}\tau_{jg}\right) - \log(\lambda_{kgd})\left(\sum_{j\neq i}\tau_{jg}X_{ij}^{(d)}\right)\right]\right\},$$

where C is a normalizing constant, such that

$$\sum_{k=1}^{K}\tau_{ik} = 1,$$

$\forall i \in \{1,\ldots,N\}$, $k \in \{1,\ldots,K\}$.

Proof. See Appendix 3.5.2. □

In the E step of the algorithm, the τ_{ik}s are updated in turn until convergence of. This corresponds to a fixed point procedure, as in Daudin et al. (2008) for instance. We emphasize that τ_{ik} is the (approximate) posterior probability for node i to be in cluster k, given the data and model parameters. Thus, the clustering structure uncovered by the method is encoded through the $N \times K$ matrix τ, whose element (i,k) is τ_{ik}.

Maximization with respect to π

Notice that $f(q(\cdot), K, \eta, D, \boldsymbol{\lambda}, \pi)$ can be written as

$$f(q(\cdot), K, \eta, D, \boldsymbol{\lambda}, \pi) = \sum_{i=1}^{N}\sum_{k=1}^{K}\tau_{ik}\log\frac{\pi_k}{\tau_{ik}} + g(q(\cdot), K, \eta, D, \boldsymbol{\lambda}),$$

where the function g regroups all the term in (3.11) not depending on π. Thus, $q(\cdot)$ and K being fixed, maximizing f with respect to η , D, $\boldsymbol{\lambda}$, π can be done independently on π and on the other parameters. The estimated value for π (under the constraint $\sum_{k=1}^{K}\pi_k = 1$) is then

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N}\tau_{ik}}{N}, \qquad \forall k \in \{1,\ldots,K\}. \tag{3.14}$$

Maximization with respect to $\boldsymbol{\lambda}$

Maximizing f with respect to $\boldsymbol{\lambda}$ leads to the following estimates

$$\hat{\lambda}_{kgd} = \begin{cases} \dfrac{\sum_{j>i}^{N}\tau_{ik}\tau_{jg}X_{ij}^{d}}{\Delta_d\sum_{j>i}^{N}\tau_{ik}\tau_{jg}} & \text{when } g > k, \\ \dfrac{\sum_{j\neq i}^{N}\tau_{ik}\tau_{jk}X_{ij}^{d}}{\Delta_d\sum_{j\neq i}^{N}\tau_{ik}\tau_{jk}} & \text{when } g = k. \end{cases} \tag{3.15}$$

Notice that contrary to $\hat{\pi}$, $\hat{\boldsymbol{\lambda}}$ does depend on η and D, which are also considered in the M optimization step.

Maximization with respect to η and D

Obviously, we have

$$\max_{\eta,D,\boldsymbol{\lambda},\pi} f\left(q(\cdot),K,\eta,D,\boldsymbol{\lambda},\pi\right) = \max_{\eta,D}\max_{\boldsymbol{\lambda},\pi} f\left(q(\cdot),K,\eta,D,\boldsymbol{\lambda},\pi\right).$$

Thus, $q(\cdot)$ and K being fixed, for any value of η (and hence D) f can be evaluated in $\hat{\boldsymbol{\lambda}}$ and $\hat{\pi}$ obtained via (3.14) and (3.15). In more details

$$\max_{\boldsymbol{\lambda},\pi} f\left(q(\cdot),K,\eta,D,\boldsymbol{\lambda},\pi\right) = \sum_{d=1}^{D}\mathcal{G}([\eta_{d-1},\eta_d[) - \frac{1}{2}\frac{K(K+1)D}{2}\log\alpha + const, \tag{3.16}$$

where all the terms which do not depend on η and/or D have been absorbed into the constant *const* and

$$\mathcal{G}([\eta_{d-1},\eta_d[) := \\ -\sum_{k,g}^{K}\left[\hat{\lambda}_{kgd}\Delta_d\left(\sum_{j>i}^{N}\tau_{ik}\tau_{jg}\right) - \log(\hat{\lambda}_{kgd})\left(\sum_{j>i}^{N}\tau_{ik}\tau_{jg}X_{ij}^{(d)}\right)\right]. \tag{3.17}$$

Notice that the criterion to maximize (now with respect to η and D) is a sum of independent components: each *gain* function $\mathcal{G}([\eta_{d-1},\eta_d[)$ applies only to interactions that take place in the time interval $[\eta_{d-1},\eta_d[$. Notice in particular that for a given d, the $\hat{\lambda}_{kgd}$ are obtained from the quantities $X_{ij}^{(d)}$ which correspond themselves to interaction counts during the time segment $[\eta_{d-1},\eta_d[$.

3.2.4 Segmentation

The criterion (3.16) must be maximized with respect to η and D and it has the general form used in change point detection problems. This can be seen by comparing (3.16) with equation (1.16) in Section 1.4.2. In (1.16), a *cost* function (to minimize) is associated with each time segment. In contrast, we introduce in this chapter a *gain* function (to maximize) to be consistent with the problem formulation used so far. Anyway, the two definitions are equivalent since the cost function can be thought as a gain function multiplied by -1. We now show how the efficient algorithms for change point detection introduced in Section 1.4.2 can be adapted to solve the maximization problem (3.16).

Dynamic programming

Let us first recall that the maximization problems based on additive criteria, like (3.17), can be solved exactly via a form of dynamic programming (Jackson et al. 2005). In general terms, dynamic programming leverages the structure of an optimization problem in order to formulate it with the help of recurrence equations than can in turn be solved efficiently in an iterative way, via some form of memorization. In order to apply this principle to the maximization of (3.16), let us denote $F(s,W)$ the maximum of said criterion for *at most* $W-1$ change points $\eta_1,\ldots,\eta_{W-1}$ restricted to the interval $]0,s]$ (and thus $\eta_0 = 0$ and $\eta_D = W$). We denote $\eta \subset [0,s]$ and

$|\eta| = W$ the corresponding constraints on the set of change points. The true maximization problem consists in finding $F(T, D)$, but as always in dynamic programming, solving a more complex problem enables to solve more efficiently the one under study.

It can be shown (following Jackson et al. 2005, Killick et al. 2012) that

$$\begin{aligned} F(T,D) &= \max_{\eta\subset[0,T],|\eta|=D',D'\leq D} \left[\sum_{d=1}^{D'} \left(\mathcal{G}([\eta_{d-1},\eta_d[) - \frac{1}{2}\frac{K(K+1)}{2}\log\alpha\right)\right], \\ &= \max_{\zeta\in]0,T]} \left\{ \max_{\eta'\subset[0,\zeta],|\eta'|=W,W\leq D-1} \left[\sum_{d=1}^{W} \left(\mathcal{G}([\eta'_{d-1},\eta'_d[) - \frac{1}{2}\frac{K(K+1)}{2}\log\alpha\right)\right] \right. \\ &\qquad \left. + \mathcal{G}([\zeta,T[) - \frac{1}{2}\frac{K(K+1)}{2}\log\alpha \right\}, \\ &= \max_{\zeta\in]0,T]} \left[F(\zeta, D-1) + \mathcal{G}([\zeta,T[) - \frac{1}{2}\frac{K(K+1)}{2}\log\alpha\right]. \end{aligned} \tag{3.18}$$

This shows that finding $F(T, D)$ can be done recursively by finding the values of $F(\zeta, D-1)$ for any ζ. This recursion is very similar to the one in (1.20). Intuitively, the idea consists in moving the position of the last change point ζ in order to maximize the criterion by using the knowledge of the maximal value of the criterion when using one less change point (hence the use of $F(\zeta, D-1)$).

Restriction on the change point locations

There are two issues in (3.18): a maximal number of change points has to be specified and the optimization over ζ (i.e. over the position of a given change point) remains an open problem. While in theory this optimization is straightforward, the key point of the recurrence in (3.18) is the possibility of memorizing $F(\zeta, D-1)$ for all values of ζ. Indeed, the dynamic programming algorithm proceeds by computing and memorizing $F(\zeta, 1)$ for all ζ, and then computes and memorizes $F(\zeta, 2)$ using $F(\zeta, 1)$, etc.

Therefore, one has to reduce the search space for the change points to a finite set. In practice, this corresponds to fix a $U \in \mathbb{N}^*$ and introduce a grid of points which are *a priori* change point candidates:

$$\mathcal{P} = \{t_0, \ldots, t_U\}$$

such that

$$0 = t_0 < t_1 < \cdots < t_U = T.$$

A natural choice for $\mathcal{P}$ is the set of all interaction times in $\mathcal{D}$, but other choices can be adopted, such as intermediate times between interactions (e.g., times of the form $\frac{\nu_{m+1}+\nu_m}{2}$) or arbitrary regular grids. Notice that choosing a grid immediately solves the problem of choosing the maximal value for D: it is exactly U.

The choice of $\mathcal{P}$ has several consequences. Firstly, the computational cost of the dynamic programming (with or without pruning) is directly linked to U (see below for details). Secondly, $\mathcal{P}$ acts as a minimal time resolution constraint. Thus, a high value of U allows to pinpoint change points very precisely but at a high computational cost, and vice versa.

Therefore, both computational and expert considerations should be taken into account for choosing $\mathcal{P}$. If the computational load is acceptable, using the set of all interaction times offers a maximal resolution, but this choice might emphasize unneeded details.

The last consequence of the choice of $\mathcal{P}$ concerns the value of α in the penalized log-likelihood (3.8). According to the hypotheses on the generative model, we observe $N(N-1)/2$ conditionally independent NHPP trajectories. Each trajectory is observed on the intervals $[t_u, t_{u+1}[$ via interaction counts. Those interaction counts are independent per the general definition of NHPP (1.10). Thus, we have $\alpha = UN(N-1)/2$ independent observations. Notice that the choice of $\mathcal{P}$ affects both the gain functions $\mathcal{G}(\cdot)$, via different change point candidates, and the penalty term, via α. Hence, the inference procedure adapts to the choice of U. Moreover, in all the experiments we carried out, we observed that the estimated values for D do not change for a sufficiently large U[5]. This remark certainly supports the choice of fixing $\mathcal{P}$ equal to the set of all interactions that occurred in $\mathcal{D}$, leading to a maximal resolution setup.

Pruned exact linear time

In the reminder of this section, we assume that a partition $\mathcal{P}$ is set, inducing U time sub-intervals. The pruned exact linear time (PELT) method of Killick et al. (2012) for change point detection in univariate time series was introduced in Section 1.4.2 and is now considered to solve (3.18).

First of all, notice that directly using the recursive decomposition in (3.18) for maximization has a cost of $\mathcal{O}(K^2U^2)$. Indeed this recursion corresponds to the Optimal Partitioning method (Jackson et al. 2005) which has a quadratic complexity in the number of observations as we showed in Section 1.4.2. In contrast, the cost function introduced in that section can be efficiently computed in constant time (for each time segment) whereas the gain function in (3.17) involves $\mathcal{O}(K^2)$ calculations. Moreover, we recall that, in the Optimal Partitioning method, for each point t_u in $\mathcal{P}$ the gain of setting $t_{u'}$ as the last change point before t_u has to be computed for all $t_{u'} < t_u$.

Fortunately, as it was the case for univariate time series, some candidate change points can be pruned through the optimization routine. This is the principle of PELT and in practice it allows to speed up the exploration of the segmentation space. We use the following result whose general statement and formal proof are given in Section 1.4.2.

Again, consider $t_{u'}$ and t_u such that $t_{u'} < t_u$ and $t_{u'}$ is *not* the last change point before t_u. Namely

$$\left(F(t_{u'}, U-1) + \mathcal{G}([t_{u'}, t_u[) - \frac{1}{2}\frac{K(K+1)}{2}\log\alpha\right) < F(t_u, U).$$

Moreover if $t_{u'}$ fulfils the condition

$$F(t_{u'}, U-1) + \mathcal{G}([t_{u'}, t_u[) < F(t_u, U),$$

then the time point $t_{u'}$ can never be the optimal last change point prior to $t_{u''}$, for all $t_{u''} > t_u$. This statement is true for all gain functions satisfying

[5]The same holds for the selected number of groups K (see Section 3.2.5).

the following condition

$$\mathcal{G}([t_{u'}, t_u[) + \mathcal{G}([t_u, t_{u''}[) \geq \mathcal{G}([t_{u'}, t_{u''}[), \qquad t_{u'} < t_u < t_{u''}. \tag{3.19}$$

Proposition 3.4 *The condition* (3.19) *is fulfilled by the gain function* $\mathcal{G}(\cdot)$ *defined in* (3.17).

Proof. See appendix 3.5.3. □

Roughly speaking, the above proposition allows us to speed up the change point detection algorithm by reducing the number of candidate change points to look for. The PELT algorithm to perform the maximization of $f(\cdot)$ with respect to η and D is detailed in the pseudocode Algorithm 3.

Algorithm 3: PELT for dynamic SBM

Require:
A grid $0 = t_0 < t_1 < \cdots < t_U = T$.
An $(N \times N \times U)$ tensor X whose entry[6] (i, j, u) is $X_{ij}^{(u)}$
A matrix τ of variational probabilities.
The penalty $\alpha = UN(N-1)/2$.
A fixed positive number of clusters K.
The gain function $\mathcal{G}(.)$.

Initializations: $F(0) = \frac{1}{4}K(K+1)\log\alpha$, $cp(0) = NULL$, $R_1 = \{0\}$.
for η^* in $1, \ldots, U$ **do**
Calculate $F(\eta^*) = \max_{\eta \in R_{\eta^*}} \left[F(\eta) + \mathcal{G}([t_\eta, t_{\eta^*}[) - F(0)\right]$.
Let $\overline{\eta} = \text{argmax}_{\eta \in R_{\eta^*}} \left[F(\eta) + \mathcal{G}([t_\eta, t_{\eta^*}[) - F(0)\right]$.
Set $cp(\eta^*) = [cp(\overline{\eta}), \overline{\eta}]$.
Set $R_{\eta^*+1} = \{\eta \in R_{\eta^*} \cup \{\eta^*\} \quad | \quad F(\eta) + \mathcal{G}([t_\eta, t_{\eta^*}[) \geq F(\eta^*)\}$.
end for

Ensure: The change points stored in $cp(U)$.

3.2.5 Selection of K and initialization clusters

As any EM like approach, the algorithm proposed for inference depends on some initializations. Notice, however, that once initial values of τ and K are provided, the other model parameters $(D, \theta) = (D, \eta, \boldsymbol{\lambda}, \pi)$ are estimated in the maximization (M) step and those estimates are employed in the E step to obtain a better estimate of τ and so on until the criterion $f(q(\cdot), K, \eta, D, \lambda, \pi)$ in (3.11) no longer increases (i.e. convergence). For a fixed value of K the initialization of τ can be obtained in several ways. For example a $N \times N$ adjacency matrix can be built by aggregating all the interactions over the time interval $[0, T]$. Hence the entry (i, j) of this adjacency matrix corresponds to $M^{(i,j)}$, with the previous notations. Then, clustering algorithms like k-means, hierarchical clustering or spectral clustering can be used to get an estimate of Z. Finally the initial matrix τ is built such that τ_{ik} is one if $Z_i = k$, zero otherwise. Another method

[6] Notice that in case the minimal partition is used, $X_{ij}^{(u)}$ is trivially equal to one for the pair (i, j) interacting at t_{u-1} and zero for all the other pairs.

to initialize τ consists in applying a k-means clustering on the rows of the $N \times UN$ matrix corresponding to the mode-1 unfolding of the tensor X (see Kolda and Bader 2009, for more details). In the experiments in Section 3.3, all mentioned initialization techniques are attempted. The initialization leading to the highest final value of the lower bound is finally retained.

So far, we have assumed that the number of clusters was fixed. However, in practice K is unknown and has to be inferred from the data. Again, we rely on the criterion defined in (3.11) which involves a penalization term over K. Recalling that the optimal number of segments is selected by the PELT procedure (see Section 3.2.3), the VEM algorithm described in this section is run for different values of K in $\{1, \ldots, K_{\max}\}$, for some fixed $K_{\max}$, and the value K maximizing the criterion is retained. The pseudocode Algorithm 4 summarizes the whole estimation routine.

Algorithm 4: VEM algorithm

Require:
A maximum number of clusters $K_{\max}$.
A set $\mathcal{D}$ of M interactions.
The criterion $f(\cdot)$ in (3.11).

Initializations: Store $\leftarrow$ vector($K_{\max}$), Pmts $\leftarrow$ list($K_{\max}$)
for K in $1, \ldots, K_{\max}$ **do**
  $\tau \leftarrow$ Some clustering algorithm
  $\{D, \theta\} \leftarrow$ Maximization($\tau, \mathcal{D}$)
  while $f(\cdot)$ increases **do**
    $\tau \leftarrow$ Expectation($\mathcal{D}, D, \theta$)
    $\{D, \theta\} \leftarrow$ Maximization($\tau, \mathcal{D}$)
  end while
  Store[K] $\leftarrow f(K, D, \tau, \theta)$
  Pmts[K] $\leftarrow \{\tau, \theta\}$
end for
$K^* \leftarrow$ argmax(Store).

Ensure: The estimated parameters in Pmts$[K^*]$.

3.3 Experiments

3.3.1 Simulated datasets

Some experiments on simulated data are carried out to test the proposed approach. Our model (called hereafter PELT-Dynamic SBM) is compared with the triclustering approach proposed in Guigourès et al. (2012; 2015), which is referred to as MODL, although MODL is a more generic technique (Boullé 2010). As pointed out in the introduction, this method is non-parametric and looks for node clusters and time segments. It is based on a combinatorial generative model estimated via a maximum a posteriori approach. As our model, it has no user tunable parameter and is therefore fully automated.

(a) $[0, 20[$. **(b)** $[20, 30[$. **(c)** $[30, 60[$. **(d)** $[60, 100[$.

Figure 3.1 – *Interactions pattern between clusters on each time segment. Each node in a graph represents a group of vertices. Each group only interacts with neighbour groups during the corresponding time segment (e.g. clusters 1 and 2 only interact with each other in Figure 3.1c, as well as clusters 3 and 4).*

First scenario. The experiments considered in this section are related to the simulation setup in Section III.A of Guigourès et al. (2012). Each simulated graph is made of 40 nodes, grouped into four clusters: 5 vertices are in clusters 1 and 2, 10 vertices in cluster 3 and 20 vertices in cluster 4. The time interval $[0, 100]$ is split into four segments ($I_1 = [0, 20[$, $I_2 = [20, 30[$, $I_3 = [30, 60[$ and $I_4 = [60, 100[$) and each segment is associated with a specific interaction pattern between clusters, as illustrated in Figure 3.1. For each number of edges varying from 50 to 10000, 50 dynamic graphs are simulated according to the following procedure:

1. A vertex as well as a random interaction time are drawn uniformly in $\{1, \ldots, 40\}$ and $[0, 100]$, respectively. The vertex is then assigned to its cluster and the interaction time to its segment.

2. If the cluster of the selected vertex is connected to one or more clusters over the considered time segment (see Figure 3.1), a second vertex is drawn uniformly at random in the union of these clusters, and an edge is generated.

3. The first two steps are repeated until the desired number of edges is reached.

4. Finally, 30% of edges are rewired uniformly at random.

The only difference between the current setup and the one used in Guigourès et al. (2012) is that we assume that graphs are undirected. Notice that the generative process for those data is neither a Poisson based model nor the combinatorial model used by MODL.

For estimation purposes, a regular grid $\mathcal{P}$ (Section 3.2.4) with unitary length time intervals is used for PELT-Dynamic SBM. Both algorithms (PELT-Dynamic SBM and MODL) are applied to the generated interaction data and results are assessed at an aggregated level (cluster numbers) and at a more refined level relying on the adjusted Rand indexes (Rand 1971, introduced in the previous chapter).

In Figure 3.2 the mean number of clusters K (respectively time segments, D) found by the two methods is plotted in blue (resp. green) as a function of the number of edges.

As it can be seen, MODL provides more accurate estimates of both K and D for a small number of edges, while PELT-Dynamic SBM needs denser graphs to recover the true number of clusters and time segments.

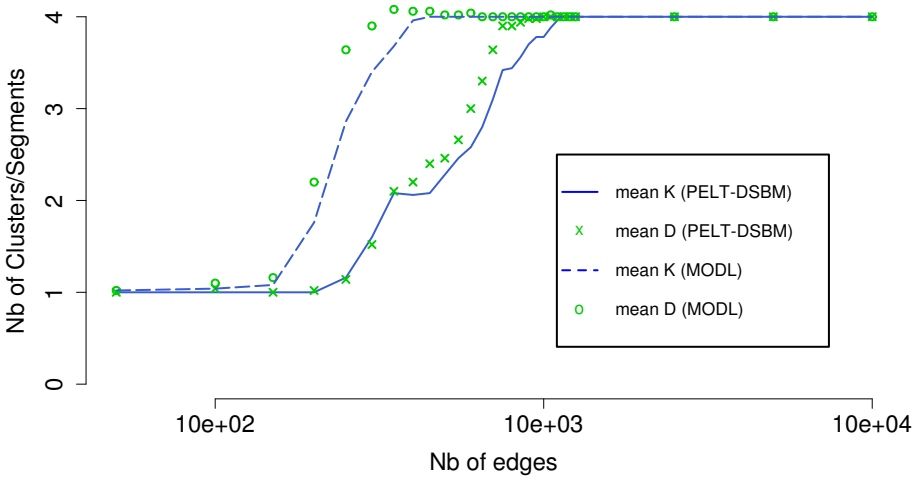


Figure 3.2 – *The average number of clusters and time segments detected by MODL and PELT-Dynamic SBM versus number of simulated edges.*

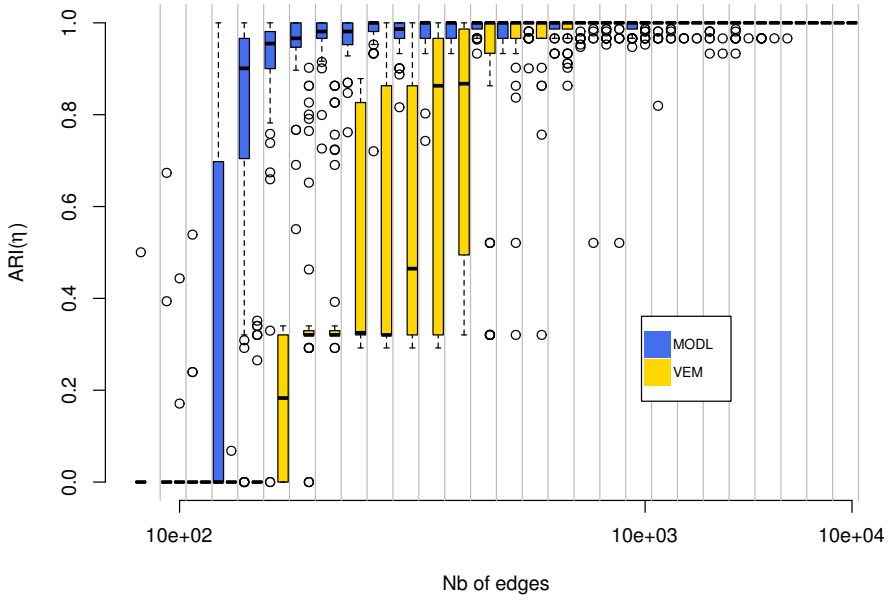


Figure 3.3 – *ARIs for the change points (η).*

These results are confirmed in Figures 3.3 and 3.4. For each number of edges, adjusted Rand indexes (ARIs) are computed to asses the quality of the estimates provided for Z and η by the two models. Regarding the change point locations, since the selected grid $\mathcal{P}$ contains 100 time intervals, it is natural to introduce "label" random variables, say $Y := \{Y_1, \ldots, Y_U\}$, such that $Y_u = d$ iff the u-th interval in the grid is assigned to I_d, $d \in \{1, \ldots, 4\}$. Hence, having a true Y and an estimated one, the ARI compares them to assess the quality of the found segmentation. When no change point is detected the ARI is zero, conversely when $\hat{\eta}_1 = 20$,

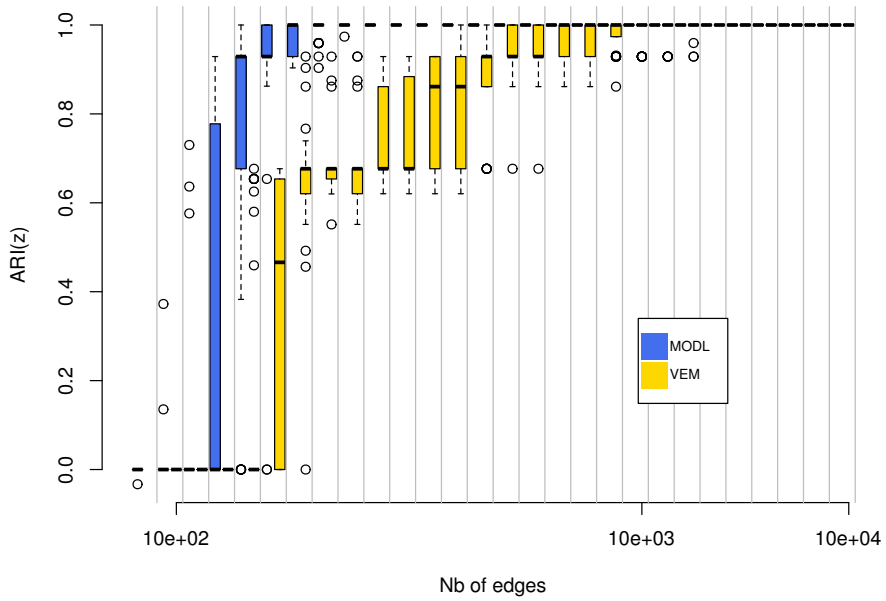


Figure 3.4 – *ARIs for the cluster memberships (Z).*

$\hat{\eta}_2 = 30$ and $\hat{\eta}_3 = 60$ the ARI is one. For each number of edges, two box-and-whiskers plots are produced, one for MODL (on the left hand side) and the other for our method (right hand side).

As pointed out above, the data are not generated according to our model nor to MODL combinatorial one. However, our model is still parametric which might explain the less accurate estimates provided here as compared to MODL. We show in the following sections that when the data are generated with a model closer to our model, the results are quite different.

Second scenario. The graphs generated in the second simulation scenario are made of 75 nodes, grouped into two clusters and undirected interactions are simulated over the time interval $[0, 10]$. This interval is split into three segments $I_1 = [0, \eta_1[$, $I_2 = [\eta_1, \eta_2[$, $I_3 = [\eta_2, 10[$, where the change points η_1 and η_2 are set equal to 2.1 and 6.9, respectively. Interactions are simulated by thinning (Lewis and Shedler 1979) according to the model we introduced in Section 3.1, based on the following intensity functions (IFs)

$$\lambda_{Z_i Z_j}(t) = \begin{cases} 0.1\mathbf{1}_{I_1}(t) + 0.2\mathbf{1}_{I_2}(t) + 0.05\mathbf{1}_{I_3}(t) & \text{if} \quad Z_i = Z_j \\ 0.05\mathbf{1}_{I_1}(t) + 0.1\mathbf{1}_{I_2}(t) + 0.025\mathbf{1}_{I_3}(t) & \text{if} \quad Z_i \neq Z_j, \end{cases}$$

for all $j > i$. Thus, the IFs define a persistent community structure through time in which the intensity of the interactions *within* clusters is twice the intensity of the interactions *between* clusters. The following simulating procedure is used to generate 50 dynamic graphs:

1. Each vertex is assigned to one of the two clusters with probability $1/2$.

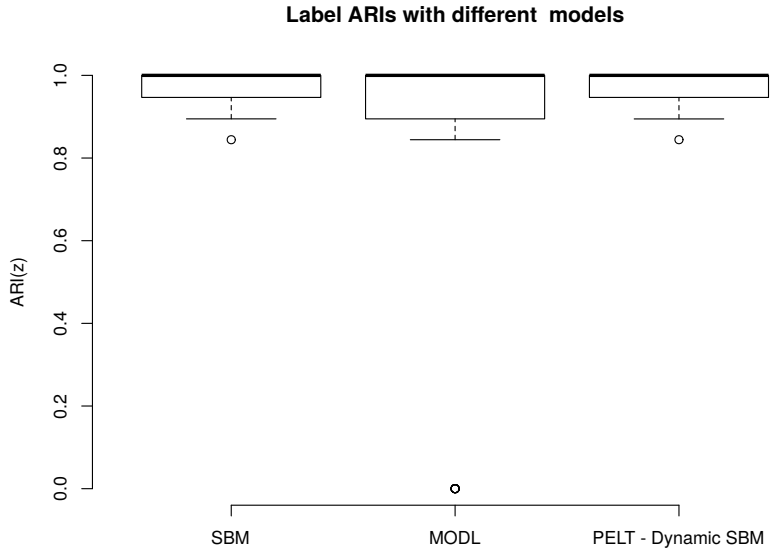


Figure 3.5 – *Boxplots over 50 simulations of the ARIs for the clustering structures obtained by SBM, MODL and PELT-Dynamic SBM for scenario 2. More details in the text.*

2. Interactions between each pair of nodes (i, j) are simulated according to the NHPP with IF $\lambda_{Z_i,Z_j}(t)$.

Again, to introduce some noise in the data, 10% of edges are randomly rewired. Considering this new setup, we aim at evaluating the clusters uncovered by our methodology along with the estimates of the change point locations η_1 and η_2. The grid $\mathcal{P}$ of all observed interactions in $\mathcal{D}$ was considered for inference. As mentioned in Section 3.2.4, this allows to pinpoint change points more accurately. The use of such a grid is made possible here because of the limited number of interactions generated.

Figure 3.5 compares the clustering results obtained by applying three different models, namely MODL, PELT-Dynamic SBM and SBM dealing with Poisson links, as in Section 2.3.1. Static SBM was applied on the aggregated adjacency matrix, where the interactions between each pair of nodes are summed up over the time interval $[0, 10]$. PELT-Dynamic SBM was initialized relying on a spectral clustering algorithm applied to the aggregated graph Laplacian. Note that, unlike MODL and PELT-Dynamic SBM, static SBM was provided with the true number of clusters. Not surprisingly, the three models can recover the hidden node groups most of the time, with SBM/PELT-Dynamic SBM slightly outperforming MODL. Notice also that, due to a persistent community structure, SBM works well on the aggregated dataset. In contrast, the next section illustrates a setup in which interactions aggregation leads to a huge loss of information.

The interest of PELT-Dynamic SBM shows up when looking at change point detection. Indeed, SBM cannot deal with it and due to the particular generative structure adopted in this section, MODL cannot recover any time cluster and considers the dynamic graph as stationary. More precisely, in order to avoid parametric assumptions, MODL uses rank based modeling for numerical values. In the triclustering context of Guigourès

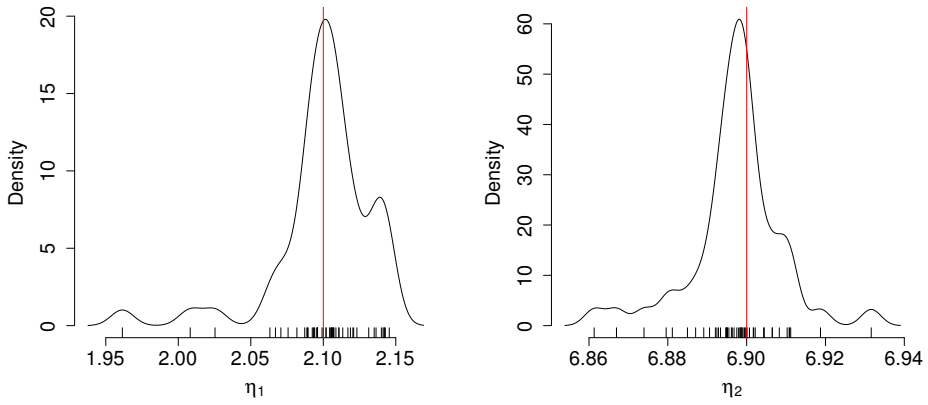


Figure 3.6 – *Kernel density estimates over 50 simulations of the change points estimated by PELT-Dynamic SBM, for scenario 2. The true values of η_1 and η_2 are given by the red vertical lines.*

et al. (2012; 2015) this means that interaction times are replaced by interaction ranks. This explains why MODL is blind to the time structure in the present scenario.

In contrast, PELT-Dynamic SBM always retrieves the right number of change points in the data. The change point estimates can be observed in Figure 3.6 and Kernel density estimates are plotted along with the true change points as red vertical lines. This illustrates the accuracy of the proposed estimation procedure and its superiority to MODL in this situation.

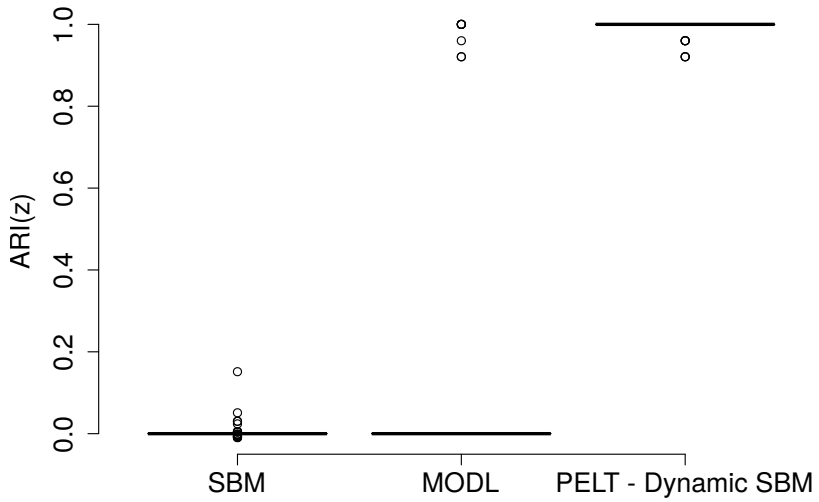
Third scenario. This section aims at illustrating that aggregating interactions can lead to an important loss of information. Thus, each graph generated is made of 100 nodes clustered in two groups, with 50 nodes each. Moreover, the time interval $[0, 12]$ is split into four segments of equal size delimited by the change points $\eta_1 = 3$, $\eta_2 = 6$ and $\eta_3 = 9$. Finally, interactions are simulated by thinning according to the following IFs

$$\lambda_{Z_i Z_j}(t) = \begin{cases} 0.05\mathbf{1}_{I_1}(t) + 0.1\mathbf{1}_{I_2}(t) + 0.05\mathbf{1}_{I_3}(t) + 0.1\mathbf{1}_{I_4} & \text{if} \quad Z_i = Z_j \\ 0.1\mathbf{1}_{I_1}(t) + 0.05\mathbf{1}_{I_2}(t) + 0.1\mathbf{1}_{I_3}(t) + 0.05\mathbf{1}_{I_4} & \text{if} \quad Z_i \neq Z_j, \end{cases}$$

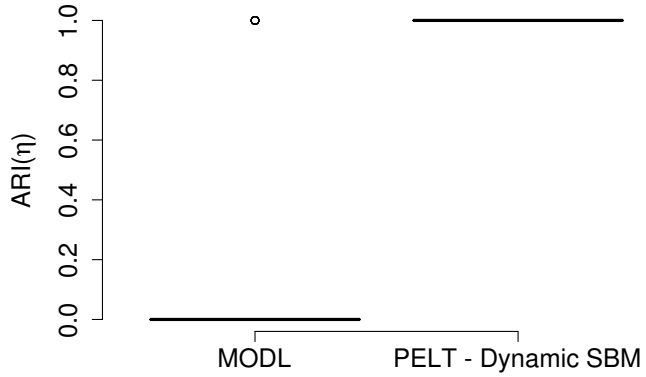
for all $j > i$ and I_d denotes the d-th segment. By construction, integrating the IFs over $[0, 12]$ leads to

$$\Lambda_{11}(T) = \Lambda_{12}(T) = \Lambda_{21}(T) = \Lambda_{22}(T) = 3.8.$$

Thus, the average number of interactions is the same for all pairs of clusters which makes clusters indistinguishable when aggregating the interactions over the whole time interval. As for the previous simulation scenarios, 50 dynamic graphs are generated and 10% of edges of each graph are rewired uniformly at random. MODL as well as PELT-Dynamic SBM are then used to uncover clusters of vertices and to segment the time interval. A regular grid $\mathcal{P}$ with unitary length time intervals is considered. The results are presented in Figure 3.7 as ARIs for both the change points and the cluster memberships. In this context, PELT-Dynamic SBM provides



(a) *ARIs for the cluster memeberships (Z).*



(b) *ARIs for the change points (η).*

Figure 3.7 – *Boxplots of the ARIs for groups of nodes 3.7b and change points 3.7a as found by SBM, MODL and PELT-Dynamic SBM in Setup 3. In Figure 3.7a, SBM is not considered since it cannot provide estimates for change points.*

more reliable estimates than MODL, which fails to retrieve any cluster or temporal structure, in most cases. It's a form of extreme blindness to the whole structure of the data induced by the blindness to the temporal structure. Note that the results for PELT-Dynamic SBM are similar when relying on a grid with a higher time resolution. The results for SBM are not reported in Figure 3.7b since this model cannot provide any time segmentation. Moreover, as illustrated in Figure 3.7a, the clustering results obtained via SBM are poor. As anticipated, when aggregating interactions through time, the assortative and non-assortative structures cancel out.

Summary. The scenarios studied in this section show that PELT-Dynamic SBM is able to recover both the cluster and the temporal structure of dynamic graphs without the need for a strong prior aggregation of the interactions (a minimal aggregation is used for computational reasons in some of the experiments). In particular, we highlighted how PELT-Dynamic SBM outperforms static SBM when clustering nodes, in situations where aggregating interactions leads to important information losses. Moreover, unlike MODL, PELT-Dynamic SBM discovers structural changes that are only based on a modification of the interaction intensities. Thus, both approaches have different use cases. In particular, the temporal structure of the data is more easily captured by our model than by MODL.

3.3.2 Real data

We now focus on a cycle hire usage dataset, publicly available at `http://api-portal.tfl.gov.uk/docs`. It characterizes the interactions that occurred on September 9, 2015, between the Santander stations of London. The considered dynamic graph is made of 735 nodes and 64514 undirected edges (with no self loops), collected with a minute precision over the day. One edge connecting nodes i and j at a given time corresponds to a cycle hire from station i to station j or, conversely from station j to station i. To limit the computational burden of the segmentation step, we relied on a regular grid $\mathcal{P}$ corresponding to 96 time intervals of 15 minutes. PELT-Dynamic SBM was then applied several times, for different values of K ranging from 0 to 20. The highest value of the criterion f (defined in (3.11)) was attained for $K = 11$ clusters and $D = 5$ time segments. MODL does not find any temporal structure in those data despite obvious changes in the aggregated intensities (see Figure 3.9). Results are presented in Figure 3.8. The Santander stations are plotted on a London map[7], different symbols/colors correspond to different clusters identified by the model. Interestingly (and as expected), generally nearby stations are placed in the same cluster and the geographical distance between them plays a key role. In Figure 3.9, an histogram of the interaction times in the whole graph is provided. Two peaks are visible around 8.30 and 18.30. The five segments detected by the methodology are delimited by the vertical red lines in the figure. A strong alignment can be observed between the histogram and the estimated segments. In particular, the two observed picks are clearly associated with segment 2 and 4, respectively. We now focus on the results for node cluster 3 (identified by the symbol $+$ in Figure 3.8) which is made of stations from central London. This cluster is a clear *community* with higher interaction values within the group than outside. This can be seen in Figure 3.10, which gives some examples of intra-group and inter-groups IFs related to cluster 3. The results are presented for clusters 1 ($\boxtimes$) and 7 ($\bigcirc$) which are geographically adjacent to cluster 3, and for clusters 4 ($\times$) and 10 ($\otimes$), which are not. Overall, as mentioned, the within group IF (on the top) is the highest at each time. However, this figure also highlights an interesting temporal pattern. Indeed, it appears

[7]Map data are available from `http://www.openstreetmap.org` and copyrighted OpenStreetMap contributors.

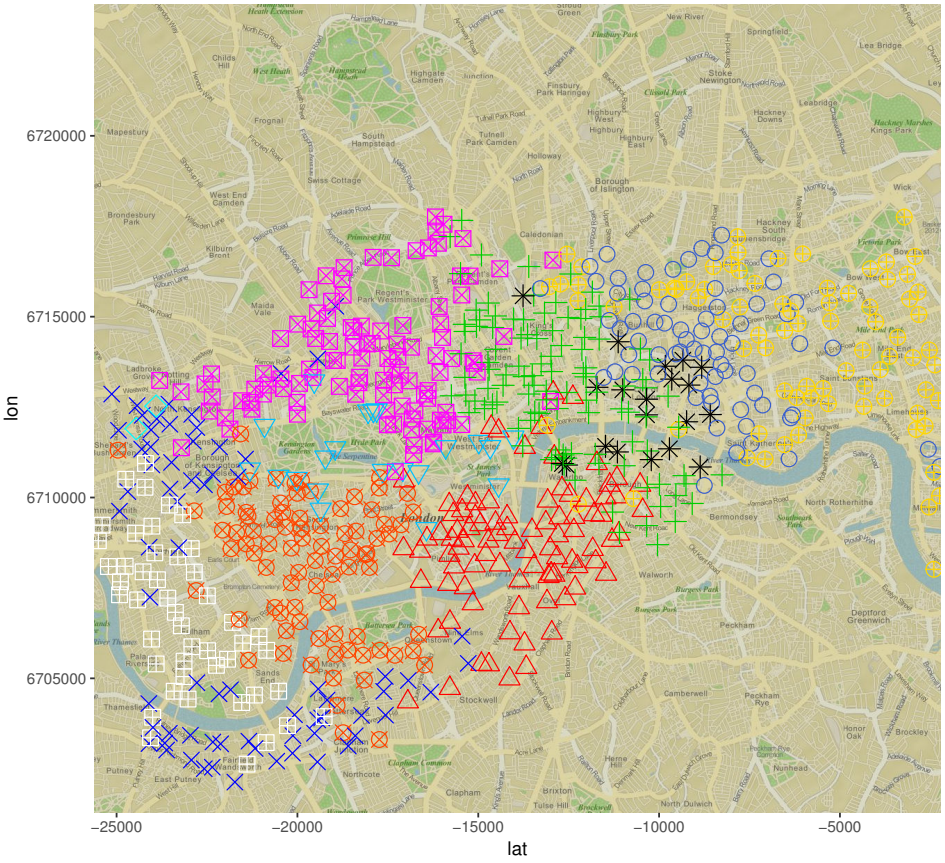


Figure 3.8 – *The 11 clusters found by PELT-Dynamic SBM represented here with 11 different symbols/colors on the left hand side.*

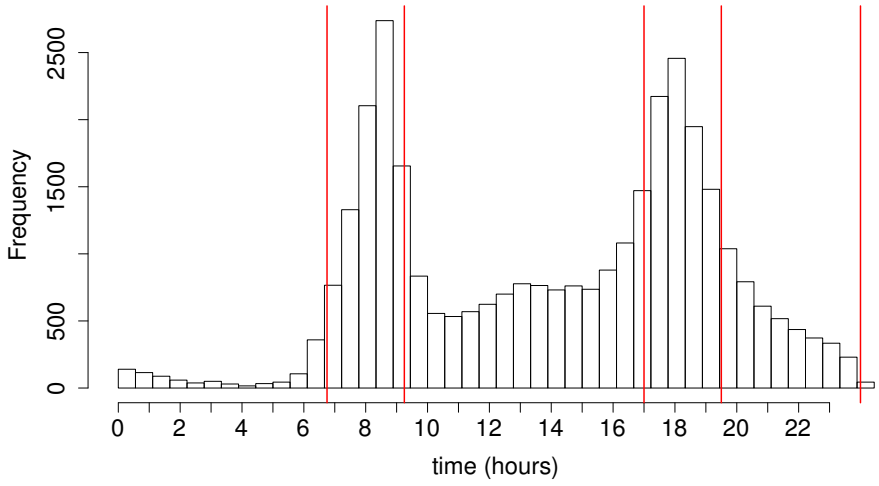


Figure 3.9 – *An histogram shows how frequent interactions (cycle hires) are during the day. The vertical lines correspond to the estimated change points.*

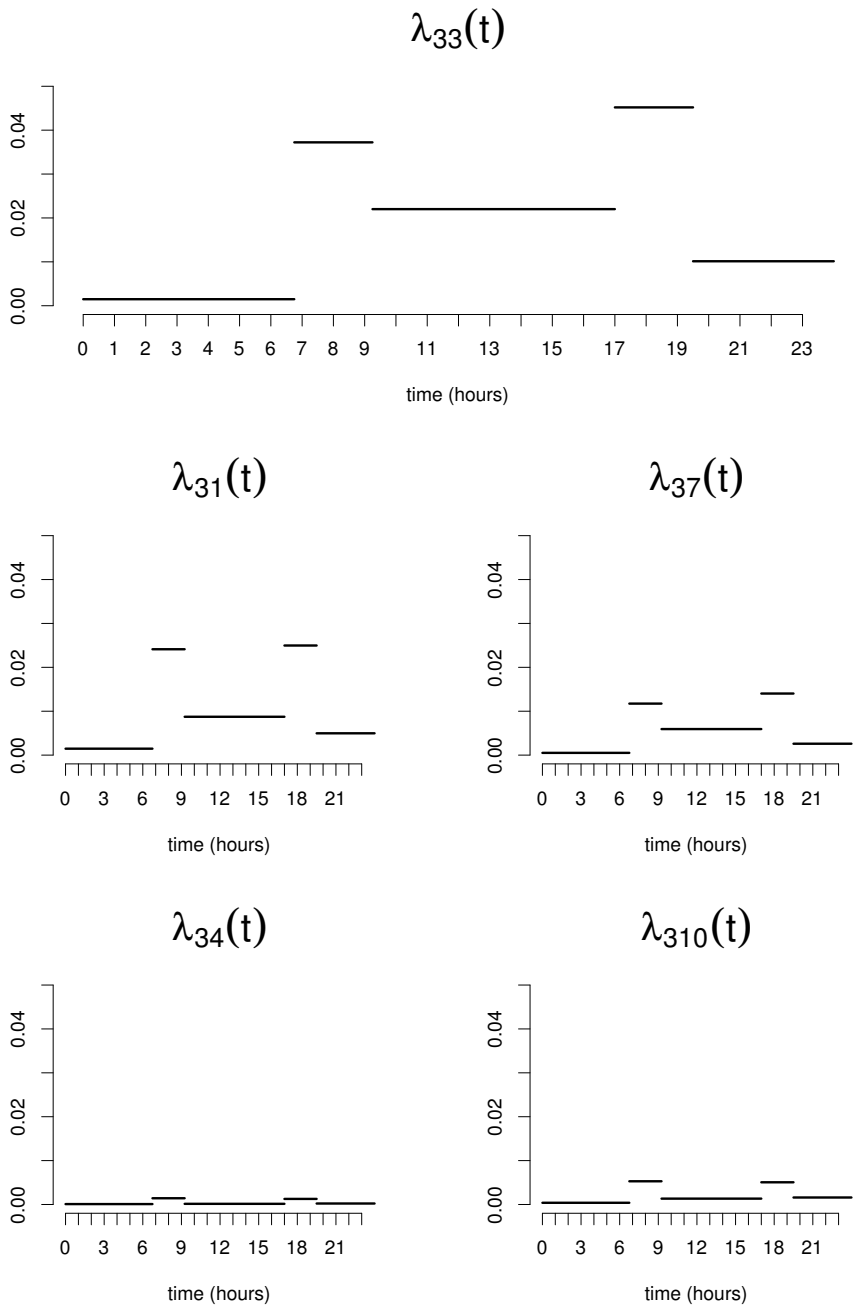


Figure 3.10 – *Estimated IFs for groups* $(3, \cdot)$. *Groups 1 and 7 are geographically adjacent to cluster 3 whereas 4 and 10 are not.*

that the between-groups IFs for the adjacent clusters are higher in the morning and in the evening than for the rest of the day. A somehow similar pattern is observed for clusters 4 and 10 but with much lower values in general. This is coherent with cycles being hired more often to go to a station which is not geographically far.

In order to highlight another feature of the proposed methodology, some results regarding cluster 8 are discussed. More specifically, Table 3.1 provides the aggregated interactions between clusters 7 ($\bigcirc$) and 8 ($*$), over the segments uncovered. In the first segment, 16 interactions occurred be-

tween vertices of cluster 8 and 47 between vertices of cluster 8 and vertices of cluster 7. Thus, cluster 8 exhibits a non-assortative connectivity pattern with less within group edges than between groups edges. Conversely, during the time segment C_2, there are more intra-group edges (502), within cluster 8, than between groups 8 and 7. This corresponds to a community pattern. Looking through all the time segments, we can observe that the community and non-assortative pattern for cluster 8 alternates through time. Thus, clustering the vertices while detecting change points in the intensity of the interactions is mandatory here since the connectivity patterns of the data set keep changing. Therefore, any method aggregating the data would miss important information present in the data. In conclu-

104	47
47	16

midnight-6.45

742	441
441	502

6.45-9.45

1106	368
368	338

9.45-17

912	419
419	984

17-19.45

572	128
128	108

19.45-midnight

Table 3.1 – *Aggregated interactions for clusters 7 (◯) and 8 (∗) during the five segments uncovered. On the main diagonal of each table, the numbers of interactions within clusters are reported: for cluster 7 on the left/top, for cluster 8 on the right/bottom. Interactions between clusters are outside the main diagonal. Community structure for cluster 8 is highlighted in blue. Non-assortative structure for cluster 8 is highlighted in red.*

sion, the uncovered clusters as well as the change point locations seems to be meaningful on a ground truth basis and PELT-Dynamic SBM proved to be fit to uncover interaction patterns that could not easily be detected by other static or dynamic clustering algorithms.

3.4 Conclusion

In this chapter, we proposed a new model for continuous time dynamic graphs adopting non-homogeneous Poisson processes. This model allows us to perform change point analysis of graph data and cluster the graph vertex simultaneously. The next chapter goes back to discrete time dynamic graph analysis and focuses on a special family of networks, i.e. communication networks. As we will see, this kind of networks can be modelled via graphs whose hedges are associated with textual contents. The dSBM introduced in Chapter 2 will be extended to analyse such graphs and exploit the textual information.

3.5 Proofs

3.5.1 Proof of Proposition 3.1

Proof. Notice, first, that the central factor on the right hand side of the equality in (3.3) can be written as

$$\prod_{m=1}^{M} \lambda_{Z_{i_m} Z_{j_m}}(\nu_m) = \prod_{m=1}^{M} \prod_{j>i}^{N} \left(\lambda_{Z_i Z_j}(\nu_m)\right)^{\mathbf{1}_{\mathcal{A}^{(i,j)}}(\nu_m)},$$

where $\mathbf{1}_{\mathcal{G}}(\cdot)$ is the indicator function on a set $\mathcal{G}$ and $\mathcal{A}^{(i,j)}$ has been defined in (3.1). By inverting the product on the right hand side, because of the indicator function we get

$$\begin{aligned}\prod_{j>i}^{N} \prod_{m=1}^{M} \left(\lambda_{Z_i Z_j}(\nu_m)\right)^{\mathbf{1}_{\mathcal{A}^{(i,j)}}(\nu_m)} &= \prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \lambda_{Z_i Z_j}(\nu_m^{(i,j)}) \\ &= \prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \left[\prod_{k,g}^{K} (\lambda_{kg}(\nu_m^{(i,j)}))^{Z_{ik} Z_{jg}}\right],\end{aligned}$$

where $\nu_m^{(i,j)}$ are the interaction times in the set $\mathcal{A}^{(i,j)}$, whose cardinality is $M^{(i,j)}$. Thanks to (3.5), the following holds

$$\begin{aligned}\prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \left[\prod_{k,g}^{K} (\lambda_{kg}(\nu_m^{(i,j)}))^{Z_{ik} Z_{jg}}\right] &= \prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \left[\prod_{k,g}^{K} \prod_{d=1}^{D} \lambda_{kgd}^{Z_{ik} Z_{jg} \mathbf{1}_{[\eta_{d-1}, \eta_d[}(\nu_m^{(i,j)})}\right] \\ &= \prod_{k,g}^{K} \prod_{d=1}^{D} \lambda_{kgd}^{\sum_{j>i}^{N} Z_{ik} Z_{jg} (M^{(i,j)}(\eta_d) - M^{(i,j)}(\eta_{d-1}))}.\end{aligned} \tag{3.20}$$

Note that the last equality employs the definition of counting process

$$M^{(i,j)}(t) = \sum_{m=1}^{M^{(i,j)}} \mathbf{1}_{]0,t]}(\nu_m^{(i,j)}).$$

By replacing (3.20) into (3.3) and using that

$$\Lambda_{Z_i Z_j}(T) = \sum_{k,g}^{K} \Lambda_{kg}(T) Z_{ik} Z_{jg} = \sum_{d=1}^{D} \sum_{k,g}^{K} \lambda_{kgd} \Delta_d Z_{ik} Z_{jg},$$

it suffices to take the logarithm of the likelihood and the proposition is proven. □

3.5.2 Proof of Proposition 3.3

Proof. The following objective function is taken into account

$$\mathcal{L}(q(Z); K, \eta, D, \boldsymbol{\lambda}, \pi) + \sum_{i=1}^{N} l_i \left(\sum_{k=1}^{K} \tau_{ik} - 1\right).$$

This function has to be maximized with respect to τ. Moreover, since the lines of the matrix τ sum to one N Lagrange multipliers $l_1, \ldots, l_N$ are

introduced. The most difficult step consists in taking the partial derivative of the objective function with respect to $\tau_{i_0k_0}$. We first focus on those terms of $\mathcal{L}(\cdot)$ depending on d

$$Q(\tau,\theta) := -\sum_{d=1}^{D}\sum_{k,g}^{K}\left(\lambda_{kgd}\Delta_d\left(\sum_{i=1}^{N}\sum_{j>i}^{N}\tau_{ik}\tau_{jg}\right) - \log(\lambda_{kgd})\left(\sum_{i=1}^{N}\sum_{j>i}^{N}\tau_{ik}\tau_{jg}X_{ij}^{(d)}\right)\right).$$

Hence

$$\begin{aligned}
\frac{\partial Q(\tau,\theta)}{\partial\tau_{i_0k_0}} &= \sum_{d=1}^{D}\frac{\partial}{\partial\tau_{i_0k_0}}\left(-\sum_{i=1}^{N}\sum_{j>i}^{N}\sum_{k,g}^{K}\tau_{ik}\tau_{jg}\lambda_{kgd}\Delta_d\right)\\
&\quad+\sum_{d=1}^{D}\frac{\partial}{\partial\tau_{i_0k_0}}\left(\sum_{i=1}^{N}\sum_{j>i}^{N}\sum_{k,g}^{K}\tau_{ik}\tau_{jg}X_{ij}^{(d)}\log(\lambda_{kgd})\right)\\
&= -\sum_{d=1}^{D}\left[\sum_{j>i_0}^{N}\sum_{g=1}^{K}\tau_{jg}\Delta_d\lambda_{k_0gd}+\sum_{j<i_0}^{N}\sum_{g=1}^{K}\tau_{jg}\Delta_d\lambda_{gk_0d}\right]\\
&\quad+\sum_{d=1}^{D}\left[\sum_{j>i_0}^{N}\sum_{g=1}^{K}\tau_{jg}X_{i_0j}^{(d)}\log(\lambda_{k_0gd})+\sum_{j<i_0}^{N}\sum_{g=1}^{K}\tau_{jg}X_{ji_0}^{(d)}\log(\lambda_{gk_0d})\right]\\
&= -\sum_{d=1}^{D}\left[\sum_{j\neq i_0}^{N}\sum_{g=1}^{K}\tau_{jg}\Delta_d\lambda_{k_0gd}-\sum_{j\neq i_0}^{N}\sum_{g=1}^{K}\tau_{jg}X_{i_0j}^{(d)}\log(\lambda_{k_0gd})\right],
\end{aligned}$$

where the last equality comes from the symmetry (i.e. interactions are undirected) of the frontal slices of tensors X and $\boldsymbol{\lambda}$. Notice that the last term in the above equation is the function inside the exponential in Proposition 3.3. The remaining terms of $\mathcal{L}(\cdot)$, not involving d, can be differentiated straightforwardly. Imposing the partial derivatives of $\mathcal{L}(\cdot)$ equal to zero and using the above equation leads to the following system

$$\begin{cases}\log(\tau_{i_0k_0}) = \log(\pi_{k_0}) - \frac{\partial Q(\tau,\theta)}{\partial\tau_{i_0k_0}} + l_{i_0} - 1\\ \sum_{k_0=1}^{K}\tau_{i_0k_0} = 1 \qquad\qquad \forall(i_0,k_0).\end{cases}$$

The solution is obtained after some manipulations and this concludes the proof. □

3.5.3 Proof of Proposition 3.4

Proof. The following definitions are introduced to keep the notation uncluttered

$$\begin{aligned}
S_{kg} &:= \sum_{j>i}^{N}\tau_{ik}\tau_{jg}\\
Y_{kg}^{[s,t[} &:= \sum_{j>i}^{N}\tau_{ik}\tau_{jg}(M^{(i,j)}(t)-M^{(i,j)}(s)) \qquad \forall s<t.
\end{aligned}$$

Moreover, for every $t_{u_e} < t_{u_f} < t_{u_g}$, the following short hand notation is used

$$\Delta^{e,f} := t_{u_f} - t_{u_e}$$

and similarly for $\Delta^{f,g}$ and $\Delta^{e,g}$. Hence, we get

$$
\begin{aligned}
\mathcal{G}([t_{u_e}, t_{u_g}[) &= \sum_{k,g} \left[\max_{\lambda_{kg}^{e,g} \in]0,+\infty[} \left(-\lambda_{kg}^{e,g} \Delta^{e,g} S_{kg} + \log(\lambda_{kg}^{e,g}) Y_{kg}^{[t_{u_e}, t_{u_g}[} \right) \right] \\
&= \sum_{k,g} \left[\max_{\lambda_{kg}^{e,g} \in]0,+\infty[} \left(-\lambda_{kg}^{e,g} \Delta^{e,f} S_{kg} + \log(\lambda_{kg}^{e,g}) Y_{kg}^{[t_{u_e}, t_{u_f}[} \right) \right] \\
&+ \sum_{k,g} \left[\max_{\lambda_{kg}^{e,g} \in]0,+\infty[} \left(-\lambda_{kg}^{e,g} \Delta^{f,g} S_{kg} + \log(\lambda_{kg}^{e,g}) Y_{kg}^{[t_{u_f}, t_{u_g}[} \right) \right] \\
&\leq \sum_{k,g} \left[\max_{\lambda_{kg}^{e,f} \in]0,+\infty[} \left(-\lambda_{kg}^{e,f} \Delta^{e,f} S_{kg} + \log(\lambda_{kg}^{e,f}) Y_{kg}^{[t_{u_e}, t_{u_f}[} \right) \right] \\
&+ \sum_{k,g} \left[\max_{\lambda_{kg}^{f,g} \in]0,+\infty[} \left(-\lambda_{kg}^{f,g} \Delta^{f,g} S_{kg} + \log(\lambda_{k,g}^{f,g}) Y_{kg}^{[t_{u_f}, t_{u_g}[} \right) \right] \\
&= \mathcal{G}([t_{u_e}, t_{u_f}[) + \mathcal{G}([t_{u_f}, t_{u_g}[),
\end{aligned}
$$

where the first and the last equalities come from the definition of $\mathcal{G}(\cdot)$. This concludes the proof. □

Topic modelling in dynamic networks with textual edges

4

So far in this thesis we focused on generative models allowing us to perform clustering, time clustering or segmentation in dynamic graphs, based on the observed interactions between vertices. Either in a discrete time framework (Chapter 2) or in a continuous one (Chapter 3), the interaction frequency was the unique source of information exploited by the proposed approaches. More generally, as detailed in the first chapter, several existing clustering techniques for static and/or dynamic network analysis are based on the graph structure, namely the presence/absence of interactions between nodes, the frequency of such interactions, the number of neighbours of nodes, etc. However, the increasing volume of communications via social networks such as Linkedin, Twitter and Facebook has being motivating researches on new techniques accounting for both the network connectivity and the textual contents associated with interactions. Such a networks can be modelled via graphs as detailed in Section 4.1. As we saw in the previous chapters, when dealing with dynamic graphs, it is of interest to be able to detect changes in the graph structure (structural changes) that can affect either the node groups composition or the way existing groups interact. In both cases, as shown in this chapter, a joint analysis of text contents and network connectivity can provide very important insights.

Section 4.2 introduces a new probabilistic approach for the clustering of *nodes* in dynamic graphs accounting for texts associated with graph edges. We consider discrete time dynamic graphs (as in Chapter 2) and partially rely on the CdSBM model (introduced in Section 2.1.2) which is referred to as dSBM henceforth. Hence, vertices are clustered in groups which are homogeneous both in terms of interaction frequency and discussed topics. Two edges are clustered together if the corresponding messages share the same majority topic. Moreover, a dynamic graph will be considered stationary on a time horizon if the proportions of topics discussed between each pair of nodes do not change in time during that horizon. In Section 4.3, a classification variational expectation-maximization (C-VEM) algorithm is adopted to perform inference and a model selection criterion is also developed to select the number of node groups, time clusters and topics. In Sections 4.4 and 4.5, experiments on both simulated and real data are carried out to assess the proposed methodology.

4.1 Statistical approaches for the joint analysis of texts and networks

The following definition is adopted henceforth.

Definition 4.1 *A* textual *network is a network in which interactions between agents are characterized by a textual content.*

Social networks or e-mail communication networks are example of textual networks.

Among probabilistic methods for text analysis, the latent Dirichlet allocation model (LDA, Blei et al. 2003) is quite popular. This generative model was illustrated in the first chapter, Section 1.4.3. As explained, the basic idea of LDA is that documents are represented as random mixtures over latent topics where each topic is characterized by a distribution over words. The topic proportions follow a Dirichlet distribution. The author-topic (AT, Steyvers et al. 2004, Rosen-Zvi et al. 2004) and the author-recipient-topic (ART McCallum et al. 2005) models partially extend LDA to deal with textual networks. Although providing authorship and information about recipients, these models do not account for the network structure, e.g. the way vertices are connected.

A first attempt to take into account the network structure along with the textual content of the interactions is due to Zhou et al. (2006). The authors propose two community-user topic (CUT) models: CUT1, modeling the communities based on the network structure only and the CUT2, modeling the communities based on the textual information alone. More recently, Pathak et al. (2008) extended the ART model by introducing the community-author-recipient-topic (CART) model. In this context, authors and recipients are assigned to latent communities and they are clustered by CART based on homogeneity criteria, both in terms of connectivity structure and textual content. Interestingly, the vertices in the associated graph are allowed to belong to multiple communities and each pair of nodes is associated with a specific topic. Although flexible, the models illustrated so far rely on Gibbs sampling for the inference procedure, which can be prohibitive when dealing with large graphs. An alternative model, that can be fitted via variational EM inference, is the topic-link LDA (Liu et al. 2009) performing both community detection and topic modeling. This model employs a logistic transformation based on topic proportions as well as author latent features. A family of 4 topic-user-community models was proposed by Sachan et al. (2012). These models, accounting for multiple community/topic memberships, discover topic-meaningful communities in graphs with different types of edges. This is of particular interest in social network analysis. For instance, in Twitter there are different types of interactions: follow, tweet, re-tweet, etc.

In order to overcome the limitations of previous methods in terms of scalability and flexibility, the recent work of Bouveyron et al. (2016) introduced the stochastic topic block model (STBM) along with an inference procedure. This approach can exhibit node partitions that are meaningful both regarding the graph structure and the topics, in directed and undirected graphs. The graph structure analysis relies on SBM, whereas

the textual analysis relies on LDA, allowing the model to characterize the construction of documents. The inference procedure is based on a classification variational EM algorithm.

The methods described so far in this section deal with static graphs whereas the main focus of this thesis is dynamic graphs. The new generative model introduced in the next section, called dynamic stochastic topic block model (dSTBM), is inspired by STBM and relies on the dynamic SBM (defined in Section 2.1.2) for network analysis and on LDA for topic modeling. However, contrarily to LDA the topic proportions associated with a document are allowed to change in time. We highlight that a dynamic extension of the LDA model allowing both topics and topic proportions to evolve in time was proposed by Blei and Lafferty (2006). Nonetheless, the approach adopted in that paper is very different from the one presented in this chapter (see Section 4.2.3 for further details.)

4.2 The dynamic stochastic topic block model (dSTBM)

The following section quickly reviews the dSBM introduced in Section 2.1.2 as "CdSBM" and introduces some simplifying assumptions. Section 4.2.2 describes how this model can be extended to deal with textual networks.

4.2.1 Simplified Block modelling

As in Chapter 2, graphs are assumed to be directed with N nodes and without self loops. A discrete time view is adopted and a dynamic graph is a sequence of static graphs (Definition 1.8). Recalling the notations used in Chapter 2, $X_{ij}^{I_u}$ is the number of *directed* interactions from node i to node j during the time interval I_u, defined according to the time partition (2.3). To keep notations uncluttered, in the remaining of this chapter we use

$$X_{iju} := X_{ij}^{I_u}.$$

As in Chapter 2 X_{iju} still follows a Poisson distribution whose parameter only depends on the clusters of nodes i and j (respectively Z_i and Z_j) and the time cluster of I_u (namely Y_u). However, to simplify the exposition two additional assumptions are made

1. The condition in Remark 2.4 is assumed to hold. Namely, the instantaneous intensity functions $\{\lambda_{kg}(\cdot)\}_{k,g}$ are assumed to be constant on each time interval $\{I_u\}_u$ of the user-defined partition.

2. Moreover, the user-defined partition in (2.3) is assumed to be regular, namely
$$\Delta_u := t_u - t_{u-1} = \Delta, \qquad \forall u \in \{1, \dots, U\}.$$

Notice that Δ is now a simple time scale factor and can be set equal to one without loss of generality. Indeed, when $\Delta \neq 1$, we can safety define $\tilde{\lambda}_{kgu} := \Delta\lambda_{kgu}$ and reduce to the previous case, with the following equality being true

$$\Delta\Lambda_{kg}^{I_u} = \tilde{\lambda}_{kgu},$$

where $\Delta\Lambda_{kg}^{I_u}$ is defined in (2.6).

The previous assumptions holding, the time constraint introduced in Section 2.1.2 reduces to the following one

$$X_{iju}|Z_{ik}Z_{jg}Y_{ud} = 1 \sim \mathcal{P}(\lambda_{kgd}),$$

where $\mathcal{P}(\cdot)$ denotes the Poisson probability distribution function and Z and Y are the very same as in Chapter 2. Furthemore, we recall that the random variables X_{iju} are all independent conditionally on Z and Y to be known.

A $K \times K \times D$ tensor $\boldsymbol{\lambda} = \{\lambda_{kgd}\}_{k,g,d}$ is introduced and the complete-data likelihood of the model described so far can be obtained

$$p(\boldsymbol{X}, Z, Y|\boldsymbol{\lambda}, \pi, \rho) = p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda})p(Z|\pi)p(Y|\rho), \tag{4.1}$$

where $p(Z|\pi)$ and $p(Y|\rho)$ are defined in (2.1) and (2.11), respectively, and

$$\begin{aligned} p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda}) &\propto \prod_{k,g}^{K}\prod_{d}^{D} \left(\lambda_{kgd}\right)^{S_{kgd}} \exp(-\lambda_{kgd}P_{kgd}), \\ S_{kgd} &:= \sum_{j\neq i}^{N}\sum_{u=1}^{U} Z_{ik}Z_{jg}Y_{ud}X_{iju}, \\ P_{kgd} &:= \sum_{j\neq i}^{N}\sum_{u=1}^{U} Z_{ik}Z_{jg}Y_{ud}. \end{aligned} \tag{4.2}$$

We point out that the likelihood $p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda})$ in the above equation is a simplified version of (2.9). Moreover the 0-1 notation of Z and Y is employed in the definitions of S_{kgd} and P_{kgd}.

4.2.2 Dynamic modelling of documents

The dSBM discussed so far can easily be extended to deal with textual networks, by assuming that a directed interaction characterizing the pair (i, j) corresponds to a document sent from i to j. More specifically, X_{iju} corresponds to the number of documents sent from i to j over the time interval I_u. The documents counted by X_{iju} are considered as a single document, obtained by concatenation and L_{iju} denotes the number of words of such a document. In the following, a dictionary containing $T^{(W)}$ words will be considered and all words are extracted from the dictionary. Hence, W_n^{iju} will denote the n-th word (in the aggregated document) sent from i to j during the time interval I_u. Using a zero-one notation, $W_{nw}^{iju} = 1$ if the word W_n^{iju} is the w-th in the dictionary, 0 otherwise.

In line with the LDA model, a list of Q *topics* is introduced and each word of a document is associated with one topic through a latent L_{iju}-vector, noted V^{iju}. More in details, $V_n^{iju} = q$ iff the word W_n^{iju} is associated with the q-th topic. For each pair of node clusters ($\mathcal{A}_k$, $\mathcal{A}_g$) and a time cluster $\mathcal{C}_d$, a vector of topic proportions $\theta_{kgd} := (\theta_{kgdq})_{q\leq Q}$ is assumed to follow a Dirichlet distribution

$$\theta_{kgd} \sim \text{Dir}(\alpha = (\alpha_1, \ldots, \alpha_Q)),$$

such that $\sum_{q=1}^{Q} \theta_{kgdq} = 1$. Hence, the n-th word in the document associated with the triplet (i, j, I_u), namely W_n^{iju}, is extracted from the latent topic q according to the following conditional probability distribution

$$\mathbf{P}(V_{nq}^{iju} = 1 | \boldsymbol{X}, Z, Y, \theta) = \prod_{k,g}^{K} \prod_{d}^{D} \theta_{kgdq}^{Z_{ik} Z_{jg} Y_{ud}}$$

corresponding to a multinomial distribution of parameter θ_{kgd}. The following full conditional distribution is obtained

$$\begin{aligned} p(V | \boldsymbol{X}, Z, Y, \theta) &= \prod_{j \neq i}^{N} \prod_{u}^{U} \prod_{n}^{L_{iju}} \theta_{Z_i Z_j Y_u V_n^{iju}} \\ &= \prod_{k,g}^{K} \prod_{d}^{D} \prod_{q}^{Q} \theta_{kgdq}^{\sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} Z_{ik} Z_{jg} Y_{ud} V_{nq}^{iju}}, \end{aligned} \tag{4.3}$$

where the exponent counts the total occurrences, in the dynamic graph, of words associated with the q-th topic, sent from cluster $\mathcal{A}_k$ to cluster $\mathcal{A}_g$, during the time cluster $\mathcal{C}_d$ and $V := (V^{iju})_{i,j,u}$. Given V, the word W_n^{iju} is finally assumed to be drawn from a multinomial distribution

$$W_n^{iju} | V_{nq}^{iju} = 1 \sim \mathcal{M}(1, \beta_q = (\beta_{q1}, \ldots, \beta_{qT^{(W)}})).$$

As a consequence, once we know that the word W_n^{iju} is extracted from the q-th topic, it is equal to the *first* word of the dictionary with probability β_{q1}, to the *second* word of the dictionary with probability β_{q2}, etc. Hence, β defines a $Q \times T^{(W)}$ matrix of word assignment probabilities.

Remark 4.1 *Notice that β (unlike θ and V) depends neither on node clusters nor on time clusters. In particular, while the mean topic proportions in each document evolve in time the mean word proportions in each topic do not.*

Denoting by $W = (W^{iju})_{i,j,u}$ the whole set of documents appearing in the whole network, the following conditional distribution is obtained by independence

$$\begin{aligned} p(W | V, \boldsymbol{X}, \beta) &= \prod_{j \neq i}^{N} \prod_{u}^{U} \prod_{n}^{L_{iju}} \beta_{V_n^{iju} W_n^{iju}} \\ &= \prod_{q=1}^{Q} \prod_{w=1}^{T^{(W)}} \beta_{qw}^{\sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} V_{nq}^{iju} W_{nw}^{iju}}, \end{aligned} \tag{4.4}$$

where the exponent counts the total occurrences, in the dynamic graph, of the w-th word of the dictionary associated with the q-th topic.

The complete-data conditional distribution for the textual part of the model is finally obtained by conditioning

$$p(W, V, \theta | \boldsymbol{X}, Z, Y, \beta) = p(W | V, \boldsymbol{X}, \beta) p(Z | \boldsymbol{X}, Z, Y, \theta) p(\theta)$$

and the joint distribution of the whole dSTBM model is

$$p(\boldsymbol{X}, Z, Y, W, V, \theta | \boldsymbol{\lambda}, \pi, \rho, \beta) = p(W, V, \theta | \boldsymbol{X}, Z, Y, \beta) p(\boldsymbol{X}, Z, Y | \boldsymbol{\lambda}, \pi, \rho).$$

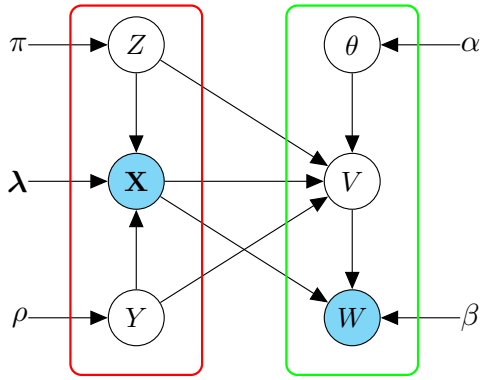


Figure 4.1 – *Graphical representation of the dynamic STBM model (dSTBM). The complete data likelihood can be decomposed in two components: the dSBM component (red plate) and the LDA component (green plate). More details in the text.*

A graphical representation can be seen in Figure 4.1. Before to go further, let us clarify the relation between dSTBM and LDA. Assuming that Z and Y are known, the set of documents W can be reorganized such that $W = (\tilde{W}_{kgd})_{kgd}$ where

$$\tilde{W}_{kgd} = \{W^{iju} | Z_{ik} Z_{jg} Y_{ud} = 1\}$$

is the set of all documents sent from any vertex in $\mathcal{A}_k$ to any vertex in $\mathcal{A}_g$ during the time cluster $\mathcal{C}_d$. By marginalization over V, it can easily be seen that each word W_n^{iju} has a mixture distribution over topics which only depends on the clusters of i and j and the time cluster of I_u. As a consequence, all words in $\tilde{W}_{kgd}$ share the same mixture distribution over topics and removing the knowledge of (k, g, d), $\tilde{W}_{kgd}$ can be seen as one of $K^2 \times D$ independent documents. This means that, if the pair (Z, Y) is known, the generative model described so far in this section is the one of a LDA model with $K^2 \times D$ independent documents, each one having its own vector of topic proportions and sharing a matrix β of word probabilities.

4.2.3 Link with other existing models

Before discussing the inference, in the next section, it is worth highlighting the relation between dSTBM and some of the existing models mentioned so far, in this thesis.

1. **Single time cluster** $(D = 1)$**.** In this case both $\boldsymbol{\lambda}$ and θ are constant in time and dSTBM reduces to STBM (Bouveyron et al. 2016).

2. **Single topic** $(Q = 1)$**.** When a single topic is used in the whole network (e.g. actors talk about a single argument) there is no additional information that can be extrapolated by the network relying on text analysis. In this case, dSTBM reduces to the dSBM.

3. **Single cluster** $(K = 1)$**.** When all vertices are clustered into a single group, the set of documents can be reorganized as $W = (\tilde{W}_d)_{d \leq D}$ corresponding to D documents. Each one corresponds to a time cluster and has its own topic proportions $(\theta_d)_{d \leq D}$. This could be seen as an original dynamic extension of the LDA model in which the topic proportions evolve in time. From a generative point of view, we stress that only D i.i.d. topic proportion vectors $\theta_1, \ldots, \theta_D$ are simulated. With respect to the original time partition, *all* documents sent in time intervals belonging to the same time cluster share the same (previously) extracted topic proportion parameter. Notice that the dynamic approach described so far is completely different from the one adopted by Blei and Lafferty (2006). In that paper, sequentially organized corpus of documents are taken into account and both the Dirichlet parameter (α) and the topic parameter (β) change in time according to (unit-root) autoregressive models combined with multinomial-logit probabilities. Hence, from a generative point of view at each time step t a *new* vector of topic proportions is simulated based on α_t.

4. **Case** $K = D = 1$**.** In line with the previous case, the set W can now be considered as a single document with its own topic proportions. The dSTBM model reduces in this case to an LDA model.

5. **Case** $Q = D = 1$**.** In presence of a single topic discussed in the whole network (i.e. text analysis is useless), with Λ constant in time, the dSTBM model reduces to SBM with weighted Poisson distributed links, already used for comparison purposes in the experiences of previous chapters.

4.3 Estimation

This section focuses on the inference procedure adopted to learn the model parameters and provide estimates of Z, Y and V. In the last part of the section, a model selection criterion is developed to select K, D and Q.

4.3.1 Variational inference

Let us assume for now, that the number of clusters (K), time clusters (D) and the number of topics (Q) are known.

Consider the following complete-data integrated log-likelihood

$$\log p(\boldsymbol{X}, Z, Y, W | \boldsymbol{\lambda}, \pi, \rho, \beta) = \log \sum_{V} \int_{\theta} p(X, Z, Y, W, V, \theta | \boldsymbol{\lambda}, \pi, \rho, \beta) d\theta. \tag{4.5}$$

We aim to maximize it with respect to the model parameters $(\boldsymbol{\lambda}, \pi, \rho, \beta)$ and the hidden label vectors (Z, Y). Unfortunately, (4.5) is not tractable due to the summation over all possible values of V inside the logarithm. Nonetheless, a variational decomposition of the above log-likelihood can be employed to obtain a lower bound which can be directly maximized.

This approach relies on the following equality

$$\begin{aligned}\log p(\boldsymbol{X}, Z, Y, W|\zeta) = \mathcal{L}(R(\cdot); \boldsymbol{X}, W, Z, Y, \zeta) \\ + \mathrm{KL}(R(\cdot)||p(\cdot|\boldsymbol{X}, W, Z, Y, \zeta))\end{aligned} \tag{4.6}$$

where $\zeta := \{\boldsymbol{\lambda}, \pi, \rho, \beta\}$, $R(\cdot)$ is any distribution over the pair (V, θ),

$$\mathcal{L}(R(\cdot); \boldsymbol{X}, W, Z, Y, \zeta) := \mathbf{E}_R \left(\log \frac{p(\boldsymbol{X}, Z, Y, W, V, \theta|\zeta)}{R(V, \theta)} \right) \tag{4.7}$$

and, as usual, $\mathrm{KL}(\cdot)$ denotes the Kullback-Leibler divergence between the approximate and the true posterior distribution of the pair (V, θ) given $\{\boldsymbol{X}, W, Z, Y, \zeta\}$

$$\mathrm{KL}(R(\cdot)||p(\cdot|\boldsymbol{X}, W, Z, Y, \zeta)) := -\mathbf{E}_R \left(\log \frac{p(V, \theta|\boldsymbol{X}, Z, Y, W, \zeta)}{R(V, \theta)} \right).$$

In the above equations, $\mathbf{E}_R$ denotes the expectation taken with respect to the distribution $R(\cdot)$. A mean field variational approximation is adopted to approximate the true posterior distribution of the pair (V, θ)

$$R(V, \theta) = R(V)R(\theta) = R(\theta) \prod_{j \neq i}^{N} \prod_{u=1}^{U} \prod_{n=1}^{L_{iju}} R(V_n^{iju}).$$

As previously observed the above equation corresponds to an independence assumption on the hidden variables (in this case V and θ) with respect to the approximate posterior distribution.

Since the integrated likelihood in (4.5) cannot be directly maximized, the idea is to replace it with the lower bound $\mathcal{L}$ and maximize it with respect to the parameters ($\boldsymbol{\lambda}$, π, ρ, β), the approximate posterior distribution $R(V, \theta)$ in the above equation and the hidden vectors Z and Y. Furthermore, as it can be seen in the graphical model in Figure 4.1, the full joint distribution of the dSTBM can be decomposed into two parts. The one in the red plate does *not* depend on the pair (V, θ). As a consequence, the lower bound defined in (4.7), can be split into two parts also

$$\mathcal{L}(R(\cdot); \boldsymbol{X}, W, Z, Y, \zeta) = \tilde{\mathcal{L}}(R(\cdot); \boldsymbol{X}, W, Z, Y, \beta) + \log p(\boldsymbol{X}, Z, Y|\boldsymbol{\lambda}, \pi, \rho), \tag{4.8}$$

where

$$\tilde{\mathcal{L}}(R(\cdot); \boldsymbol{X}, W, Z, Y, \beta) := \mathbf{E}_R \left(\log \frac{p(W, V, \theta|\boldsymbol{X}, Z, Y, \beta)}{R(V, \theta)} \right). \tag{4.9}$$

Note that the joint distribution $p(\boldsymbol{X}, Z, Y|\boldsymbol{\lambda}, \pi, \rho)$ is the same as in (4.1) and corresponds to the dynamic SBM part of the model. Modulo the simplifying assumptions discussed in the previous section, this joint distribution appeared for the first time in (2.17). Furthermore, given Z and Y, the first term on the right hand side of (4.8) only involves the pair $(R(\cdot), \beta)$ while the second term only involves $(\boldsymbol{\lambda}, \pi, \rho)$. Hence, the maximization algorithm that is detailed in the next section consists in alternating the following two steps, up to convergence:

1. **VEM step.** For a given pair (Z, Y), the lower bound $\mathcal{L}$ is maximized with respect to the pair $(R(\cdot), \beta)$ (involving $\tilde{\mathcal{L}}$) and the triplet $(\boldsymbol{\lambda}, \pi, \rho)$ (involving the dSBM complete-data likelihood).

2. **Classification step.** The lower bound $\mathcal{L}$ is maximized in a greedy fashion with respect to the pair (Z, Y).

This algorithm alternating a variational EM routine with a clustering step was used in Bouveyron et al. (2016) and is built upon the C-EM algorithm (Celeux and Govaert 1991).

4.3.2 Maximization of the lower bound

In this section, the updating formulas for $R(V, \theta)$ and the model parameters $(\lambda, \pi, \rho, \beta)$ are provided by the following propositions. At the end of the section, we discuss the maximization with respect to the pair (Z, Y).

Maximization of $\mathcal{L}$ with respect to $R(V, \theta)$. The updating formulas corresponding to the E step of the VEM algorithm are given in the following two propositions.

Proposition 4.1 *The VEM update step for distribution $R(V_n^{iju})$ is given by*

$$R(V_n^{iju}) = \mathcal{M}(V_n^{iju}; 1, \phi_n^{iju} = (\phi_{n1}^{iju}, \ldots, \phi_{nQ}^{iju}))$$

where

$$\phi_{nq}^{iju} \propto \left(\prod_{w=1}^{T^{(W)}} \beta_{qw}^{W_{nw}^{iju}} \right) \prod_{k,g}^{K} \prod_{d}^{D} \exp \left(\psi(\gamma_{kgdq}) - \psi(\sum_{q=1}^{Q} \gamma_{kgdq}) \right)^{Z_{ik} Z_{jg} Y_{ud}}, \qquad \forall (n, q)$$

where ϕ_{nq}^{iju} is the approximate posterior probability of word W_n^{iju} being in topic q and $\psi(\cdot)$ is the digamma function.

Proof. In Appendix 4.7.1. □

Proposition 4.2 *The VEM update step for distribution $R(\theta)$ is given by*

$$R(\theta) = \prod_{k,g}^{K} \prod_{d}^{D} \mathrm{Dir}(\theta_{kgd}; \gamma_{kgd} = (\gamma_{kgd1}, \ldots, \gamma_{kgdQ}))$$

where

$$\gamma_{kgdq} = \alpha_q + \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} Z_{ik} Z_{jg} Y_{ud} \phi_{nq}^{iju}, \qquad \forall (k, g, d).$$

Proof. In Appendix 4.7.2 □

Maximization of $\mathcal{L}$ with respect to the model parameters. The following proposition provides the estimates of the model parameters $(\lambda, \pi, \rho, \beta)$ obtained through maximizing the lower bound in (4.7). The lower bound $\tilde{\mathcal{L}}$ in (4.9) is computed in the appendix.

Proposition 4.3 *The estimates of* $(\beta, \boldsymbol{\lambda}, \pi)$ *and* ρ *are given by*

$$\beta_{qw} \propto \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{l=1}^{L_{iju}} W_{nw}^{iju} \phi_{nq}^{iju}, \qquad \forall (q, w) \tag{4.10}$$

$$\lambda_{kgd} = \frac{S_{kgd}}{P_{kgd}}, \qquad \forall (k, g, d) \tag{4.11}$$

$$\pi_k \propto |\mathcal{A}_k|, \qquad \forall k, \tag{4.12}$$

$$\rho_d \propto |\mathcal{C}_d|, \qquad \forall d, \tag{4.13}$$

where S_{qrl} *and* P_{qrl} *were defined in* (4.2).

Proof. In Appendix 4.7.4. □

Maximization of $\mathcal{L}$ with respect to the label vectors. Other parameters being fixed, we now attempt to maximize $\mathcal{L}$ with respect to the pair (Z, Y). Since this combinatorial problem cannot be attacked directly, due to the huge number of cluster assignments to test $(K^N D^U)$, a *greedy* search strategy is employed to look for a local maximum. Greedy search strategies were discussed in Chapter 2 when maximizing the ICL for the dSBM. In this context, however, there are two important differences with respect to the previous framework:

1. The function to be maximized is not the same.
2. The number of node clusters (K) and time clusters (D) is *fixed*. Hence, cluster merges are not considered in this framework and nodes (respectively time intervals) are only allowed to switch cluster (resp. time cluster). If a node (resp. time interval) is alone in its own cluster (resp. time cluster) then it cannot be moved.

Let us consider Z at first and assume that nodes are clustered in K initial groups (see section 4.3.3 for more details about this initial assignment). If node i is currently in cluster $\mathcal{A}_k$, the algorithm assesses the increase/decrease in the lower bound $\mathcal{L}$ due to switching node i to the cluster $\mathcal{A}_g$ for each $g \neq k$. The switch (if any) leading to the highest increase of the lower bound is actually performed and the entire routine is iteratively applied to *all* nodes until no further increase of $\mathcal{L}$ is possible. The maximization with respect to Y works similarly: nodes are replaced by time subintervals I_u and node clusters $\mathcal{A}_k$ by time clusters $\mathcal{C}_d$.

As previously explained, a greedy search is never guaranteed to converge to a global maximum. Hence a good strategy consists in performing several independent greedy maximizations randomizing over the node/time intervals moving order and finally choosing the values of (Z, Y) leading to the highest value of the lower bound (see Section 2.2.2 for further details on this point).

4.3.3 Further issues

Initialization. Assuming that K, D and Q are known, the C-VEM algorithm described in the previous section still needs some initial values of (Z, Y) in order to provide estimates for the model parameters and the

variational posterior distribution $R(V, \theta)$. The approach proposed in this chapter for the initializations relies on a spectral clustering algorithm applied to proper similarity matrices. The initialization of Z is considered at first. Recalling the definition of $\boldsymbol{X} = \{X_{iju}\}_{iju}$ we proceed as follows

1. The VEM algorithm for the LDA model is a applied to the collection of documents exchanged from all pair of nodes in the whole time horizon. Note that these documents correspond to the entries of $\boldsymbol{X}$ and the VEM algorithm provides the main topic discussed in each document. Hence an $N \times N \times U$ tensor MT (main topic) is obtained, such that $MT_{iju} = q$ if and only if q is the main topic discussed in the document sent form i to j, during the time interval I_u.

2. An $M \times M$ similarity matrix Ξ is obtained as follows

$$\begin{aligned}\Xi(i,j) = &\sum_{u=1}^{U}\sum_{h=1}^{N} \delta(MT_{ihu} = MT_{jhu}) X_{ihu} X_{jhu} \\ &+ \sum_{u=1}^{U}\sum_{h=1}^{N} \delta(MT_{hiu} = MT_{hju}) X_{hiu} X_{hju}.\end{aligned}$$

   The rationale behind the above equation is quite intuitive: if i and j have a common neighbour *and* they share with him the same main topic, then the similarity between i and j increases. Two terms appear on the right hand side of the equality because we are dealing with directed graphs.

3. A spectral clustering algorithm is applied to the graph Laplacian associated with Ξ. This allows to cluster nodes into K groups and produce an initial estimate of Z.

The initialization of Y is performed similarly. A $U \times U$ similarity matrix Σ is built such that two time intervals are similar if they share the same main topic discussed in the whole network

$$\Sigma(u,v) = \sum_{i=1}^{N}\sum_{j=1}^{N} \delta(MT_{iju} = MT_{ijv}) X_{iju} X_{ijv}$$

for all pairs $(u, v) \in U \times U$ such that $u \neq v$. A spectral clustering algorithm if finally applied to the graph Laplacian associated with the similarity matrix Σ to produce an initial estimate of Y.

Model selection. So far, the parameters K, D and Q were assumed to be known, but in real world datasets this assumption is unrealistic. In order to estimate these parameters, we adopts the ICL criterion (introduced in Section 1.3.2 for SBM) to approximate the complete-data integrated log-likelihood in (4.5). This approach extends the model selection criterion proposed in Bouveyron et al. (2016) to the dynamic framework of this chapter.

Proposition 4.4 *An integrated classification criterion (ICL) for the dSTBM is*

$$\begin{aligned} ICL_{dSTBM} = \tilde{\mathcal{L}}(R(\cdot); \boldsymbol{X}, W, Z, Y, \beta) - \frac{Q(T^{(W)}-1)}{2}\log(DK^2) \\ + \max_{\boldsymbol{\lambda},\pi,\rho} \log p(\boldsymbol{X}, Z, Y|\boldsymbol{\lambda}, \pi, \rho) \\ - \frac{DK^2}{2}\log(UN(N-1)) - \frac{K-1}{2}\log(N) - \frac{D-1}{2}\log(U). \end{aligned} \tag{4.14}$$

Proof. In Appendix 4.7.5. □

4.4 Numerical Experiments

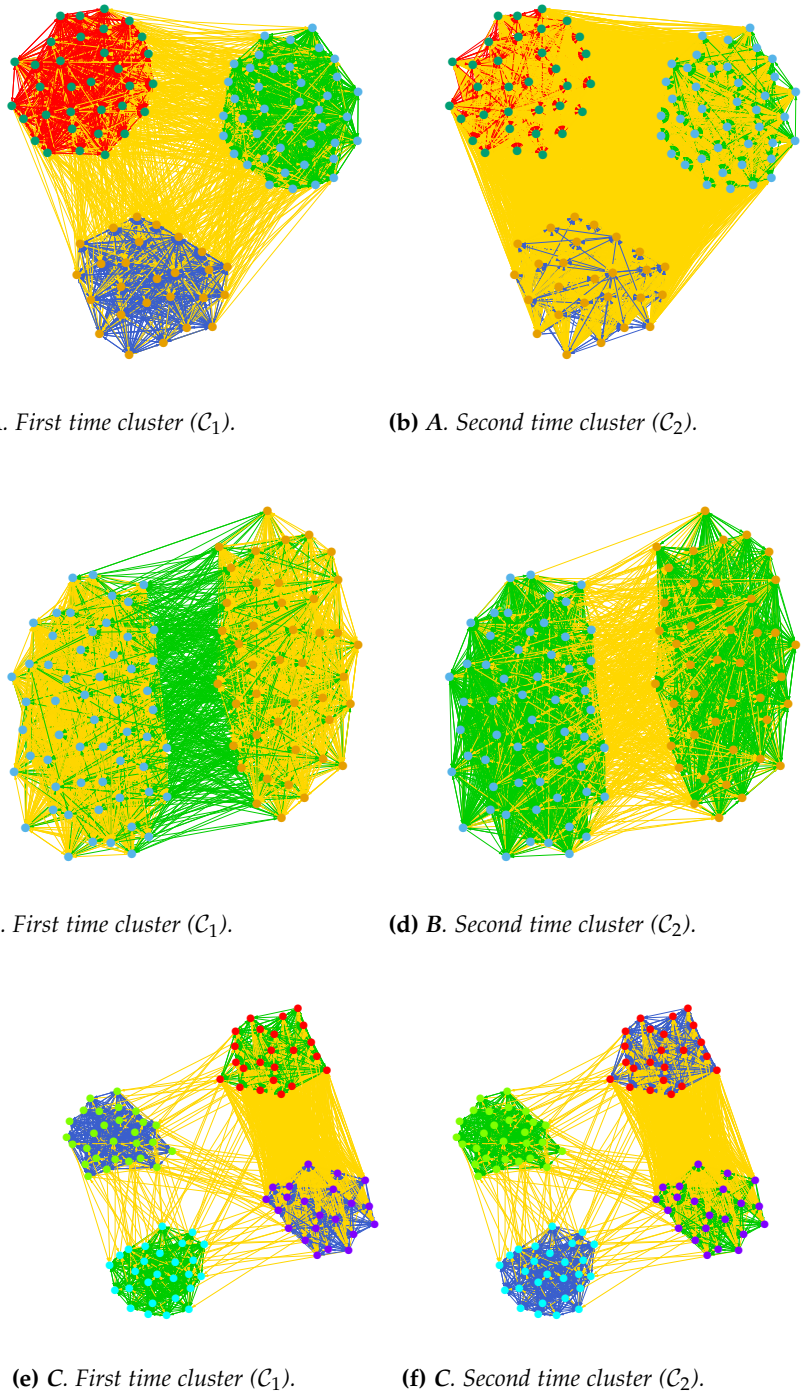
In the first part of this section, both dSTBM and the ICL model selection criterion presented in the previous section, are tested on simulated data. In order to highlight some peculiarities, dSTBM tested in three different scenarios with four other models: the dynamic SBM , STBM (Bouveyron et al. 2016), a standard SBM using the mixer package `https://cran.r-project.org/web/packages/mixer/index.html` and LDA using the topicmodels package (Grün and Hornik 2011).

4.4.1 Simulation setups

In the following simulation setups, the parameter α_q is assumed to be equal to 1, inducing a uniform distribution over the topic proportions θ_{kgd}. In each setup, 50 dynamic graphs are independently simulated and the messages associated with graph edges are sampled from four texts from BBC news. One text is about the birth of Princess Charlotte, the second is about black holes in astrophysics, the third one focuses on UK politics and the fourth on cancer diseases. Each message, associated with one directed interaction, is made of 75 words. We finally stress that, the message sampling procedure adopted in the following scenarios is *not* exactly the one described in the previous sections for dSTBM. Each setup is detailed in the following.

Scenario A. Nodes are grouped into three clusters and time intervals in two time clusters. During the first time cluster, the graph exhibits a clear community structure: interactions *within* groups are more frequent than interactions *between* groups. An opposite non-assortative structure characterizes the graph during the second time cluster: interactions between groups are more frequent than interactions inside groups. Each group talks about a single topic and a fourth, shared topic, is associated with the interactions between two different groups ($Q = 4$). In order to introduce some noise, 10% of interactions within each group is (randomly) associated to the shared topic (see Figures 4.2a and 4.2b). In this first scenario, the topic proportions do not change in time.

Scenario B. In this second scenario, the dynamic graph maintains a persistent community structure, whereas a structural time change occurs in the topic proportions. Nodes are grouped into two clusters and time intervals into two time clusters. Two topics are taken into account,



(a) ***A***. *First time cluster* ($\mathcal{C}_1$).

(b) ***A***. *Second time cluster* ($\mathcal{C}_2$).

(c) ***B***. *First time cluster* ($\mathcal{C}_1$).

(d) ***B***. *Second time cluster* ($\mathcal{C}_2$).

(e) ***C***. *First time cluster* ($\mathcal{C}_1$).

(f) ***C***. *Second time cluster* ($\mathcal{C}_2$).

Figure 4.2 – *Dynamic graphs simulated according to three different setups (A,B and C). The graph on the left (respectively right) hand side of each line is obtained through aggregation of the interactions taking place on the first (resp. second) time cluster.*

Scenario	A	B	C
N	100		
U	100		
K	3	2	4
D	2		
Q	4	2	3
π	$(1/K,\dots,1/K)$		
ρ	$(1/D,\dots,1/D)$		
λ on $\mathcal{C}_1$	$\begin{cases}\lambda_{kk1}=0.03\\ \lambda_{kg1}=0.0075 \quad g\neq k\end{cases}$	$\begin{cases}\lambda_{kk1}=0.03\\ \lambda_{kg1}=0.0075 \quad g\neq k\end{cases}$	$\begin{cases}\lambda_{kk1}=\lambda_{141}=\lambda_{411}=0.03\\ \lambda_{gk1}=0.0075 \quad \text{otherwise}\end{cases}$
λ on $\mathcal{C}_2$	$\begin{cases}\lambda_{kk2}=0.0075\\ \lambda_{gk2}=0.03 \quad g\neq k\end{cases}$	$\begin{cases}\lambda_{kk2}=0.03\\ \lambda_{gk2}=0.0075 \quad g\neq k\end{cases}$	$\begin{cases}\lambda_{kk2}=\lambda_{142}=\lambda_{412}=0.03\\ \lambda_{gk2}=0.0075 \quad \text{otherwise}\end{cases}$
θ on $\mathcal{C}_1$	$\begin{cases}\theta_{1111}=\theta_{2212}=\theta_{3313}=1\\ \theta_{kg14}=1 \quad g\neq k\\ \text{otherwise} \quad 0\end{cases}$	$\begin{cases}\theta_{1112}=\theta_{2212}=1\\ \theta_{kg11}=1 \quad g\neq k\\ \text{otherwise} \quad 0\end{cases}$	$\begin{cases}\theta_{1112}=\theta_{3312}=1\\ \theta_{2211}=\theta_{4411}=1\\ \theta_{kg13}=1 \quad g\neq k\\ \text{otherwise} \quad 0\end{cases}$
θ on $\mathcal{C}_2$	$\begin{cases}\theta_{1121}=\theta_{2222}=\theta_{3323}=1\\ \theta_{kg24}=1 \quad g\neq k\\ \text{otherwise} \quad 0\end{cases}$	$\begin{cases}\theta_{1121}=\theta_{2221}=1\\ \theta_{kg22}=1 \quad g\neq k\\ \text{otherwise} \quad 0\end{cases}$	$\begin{cases}\theta_{1121}=\theta_{3321}=1\\ \theta_{2222}=\theta_{4422}=1\\ \theta_{kg23}=1 \quad g\neq k\\ \text{otherwise} \quad 0\end{cases}$

Table 4.1 – *Parametrization in different setups.*

corresponding to two of the four texts from the BBC news. During the first time cluster, each community talks preferentially about the same topic (say T_1) and a second topic T_2 is reserved to interactions between communities (Figure 4.2c). During the second time cluster, the two topics have the opposite role. Hence, T_2 is used for the within community interactions whereas T_1 is discussed between members of different groups (Figure 4.2d). As in the previous setup, 10% of interactions inside each group is (randomly) associated with the shared topic to introduce some noise.

Scenario C. This third scenario consists in a dynamic graph whose nodes are grouped in four clusters. However, only two of these clusters are real communities, with actors talking preferentially about a unique topic inside the community. The other two clusters form a single community and the topic they discuss about is the only discriminant. Hence, three topics are considered: two clusters use one topic, the other two clusters use another topic and a third topic is used for communications between all different groups. In order to induce a relevant time structure, the topics used within groups change from a time cluster to another as illustrated in Figures 4.2e and 4.2f.

4.4.2 Benchmark results

A detailed description of the three scenarios mentioned so far can be seen in Table 4.1. The C-VEM algorithm for dSTBM was run on 50 simulated dynamic graphs in each scenario. In a first time, we focus on the clustering produced by the model when the number of clusters K, time clusters D and topics Q are known. The clustering results for dSTBM, dSBM and

	Setup A		
Model	node ARI	time ARI	edge ARI
dSTBM	0.99 (0.06)	1 (0)	0.99 (0.06)
dSBM	1 (0)	1 (0)	-
STBM	1 (0)	-	0.66 (0.21)
SBM	0.01 (0.06)	-	-
LDA	-	-	0.73 (0.20)

	Setup B		
Model	node ARI	time ARI	edge ARI
dSTBM	1 (0)	1 (0)	1 (0)
dSBM	0.98 (0.03)	0.00 (0.01)	-
STBM	0.5 (0.5)	-	0.02 (0.03)
SBM	0.99 (0.04)	-	-
LDA	-	-	1 (0)

	Setup C		
Model	node ARI	time ARI	edge ARI
dSTBM	1 (0)	1(0)	1 (0)
dSBM	0.67 (0.05)	0.00 (0.01)	-
STBM	1 (0)	-	0.70 (0.10)
SBM	0.65 (0.04)	-	-
LDA	-	-	0.69 (0.15)

Table 4.2 – *Clustering results for dSTBM, dSBM, STBM, SBM and LDA on 50 graphs simulated according to the different setups. The true values of K, D and Q is assumed to be known. The average ARI values are reported, with standard deviations inside brackets.*

STBM can be seen in Table 4.2, where, as usual, ARI stands for adjusted Rand index. The ARI was already used in previous chapters to assess both node and time clusterings. In this chapter they are also used to asses the edge clustering. In this context, the clustering measure "edge ARI" in Table 4.2 is equal to one when the main topic used in each exchanged document is correctly retrieved by the model. We recall that one document is uniquely associated with a triplet (i, j, I_u) in the dynamic graph: source node, destination node and time interval. Hence, the number of exchanged documents coincides with the total degree of the simulated dynamic graph. It follows that the edge ARI defined so far is not available for both dSBM and STBM: the former does not deal with topics, the latter cannot recover information about the interactions taking place at time I_u since this information is definitely lost, due to aggregation. However, STBM can cluster the edges of the aggregated graph. Namely, it estimates the main topic used by each pair of nodes during the whole time horizon. Hence, the edge ARI for STBM can be calculated by assigning to *all* edges associated with the pair (i, j) in the dynamic graph, the main topic estimated for that pair by STBM.

Let us start from the first setup **A**. Not surprisingly, dSTBM and dSBM have very similar performances and dSBM is slightly more accurate in clustering nodes (ARI equal to one versus ARI equal to 0.99). This small

difference however is not very significant and can be explained by the different initializations adopted by the two approaches. As mentioned above, in this scenario the proportion of assigned topics (θ) is constant in time, hence the structural change in the dynamic graphs can be fully detected by dSBM and the analysis of documents does not bring any further information. This is the reason why the time ARI is equal to one for both the approaches: the time structure can be recovered with or without the analysis of documents. Since STBM cannot deal with dynamic graphs, the C-VEM algorithm for this model is run on the static graph obtained by aggregating the interactions/e-mails on the whole time horizon (September, 2001 - January, 2002). Despite of the structural change in the dynamic graph (Figures 4.2a and 4.2b), the topics used for communications within each community and between communities remain distinct on the whole time horizon. This is the reason why STBM can correctly cluster nodes. Similarly to STBM, the SBM model is run on the aggregated graph. Its performance is poor since the community structure in $\mathcal{C}_1$ and the non-assortative structure in $\mathcal{C}_2$ cancel each other out when aggregating interactions over time.

Looking at the edge ARI, when aggregating interactions over time information is lost: this explains the edge ARI of 0.66 for STBM. The edge ARI is slightly better for LDA which is applied to the whole collection of documents (there is no aggregation).

Consider now the second setup **B**. Since the topic proportions are the only time varying parameter, dSBM cannot see any time cluster (null time ARI). Nonetheless, the persistent community structure allows it to recover the actual node partition most of the time (node ARI of 0.98). A similar result can be seen for SBM. Conversely, since each topic is alternatively used for intra and inter community interactions (Figures 4.2c and 4.2d), STBM suffers in recovering the actual node partition (node ARI of 0.5). As explained before, the LDA model can be applied to the original set of documents and in this case, not particularly noised, it performs very well.

The last scenario **C** is the hardest for dSBM. As in the previous case, the topic proportions are the only time varying parameter and the time clusters are not correctly detected by the model (null time ARI). Moreover, two clusters form a single community (Figures 4.2e and 4.2f) and are only discriminated by the used topic. Hence the node ARI is never higher than 0.7 for dSBM (and SBM too). Instead, in contrast with the previous scenario, the inter-community topic (yellow color) is never employed for intra-community interactions and STBM can recover the actual node partition. Notice, however, that both STBM and LDA are performing worse than dSTBM in clustering edges.

4.4.3 Model Selection

So far, the C-VEM algorithm for dSTBM was run on fifty simulated dynamic graphs for each setup and the actual number of groups K, time clusters D and topics Q was assumed to be known. In real applications, these three parameters must be estimated and this can be done for dSTBM relying on the ICL model selection criterion developed in Proposition 4.4. In terms of model selection, the third scenario **C** is by far the hardest to

Setup C, ICL (dSTBM)						
K/D	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	**48**	1	0	0	0
5	0	1	0	0	0	0
6	0	0	0	0	0	0

Table 4.3 – *Frequency of selections by ICL for dSTBM* (K, D, Q) *on 50 simulated graphs in the third scenario* ***C****. The actual values of* (K, D, Q) *are* $(4, 2, 3)$*, respectively and the actual value of* Q *is always selected by ICL and therefore it is not reported in the table.*

deal with, due to the quite sophisticated dynamic graph structure. Hence, we focus on this setup to assess the ICL criterion. The estimates of K, D and Q, provided by ICL for dSTBM, are illustrated in Table 4.3. The actual number of topics $(Q = 3)$ is always detected by ICL and it is not reported in the table. As it can be seen, the actual values of K and D are recovered in 48 out of 50 cases.

4.5 Analysis of the Enron scandal

The famous scandal involving the energy company Enron Corporation was publicized in October 2001. Two moths later, USA experienced the largest bankruptcy failure up to that time. The first part of this section describes the Enron data set we used, while the second part illustrates the results obtained through applying the dSTBM model to the dataset.

4.5.1 Context and data

The Enron communication network is a popular data set containing all e-mail exchanges between the 149 employees of the company. The original dataset is available at `http://www.cs.cmu.edu/~./enron/` and cover the time horizon 1999-2002. The time window considered in the present section spans from September, 3rd, 2001 to January, 28th, 2002, including three key dates

1. September, 11th, 2001: the terrorist attacks to the Twin Towers and the Pentagon (USA).
2. October, 31st, 2001: the Securities and Exchange Commission (SEC) opened on investigation for fraud concerning Enron.
3. December, 2nd, 2001: Enron failed bankruptcy, resulting in more than 4,000 lost jobs.

The selected time window is partitioned in weekly subintervals, thus corresponding to $U = 21$ weeks. As previously explained, the documents/e-mails sent from i to j during each time interval I_u (a week) are aggregated into a single document, obtained by concatenation. Each document is pre-processed in a classical way: words are stemmed, very short words and stop words are removed, punctuation and numbers are ignored.

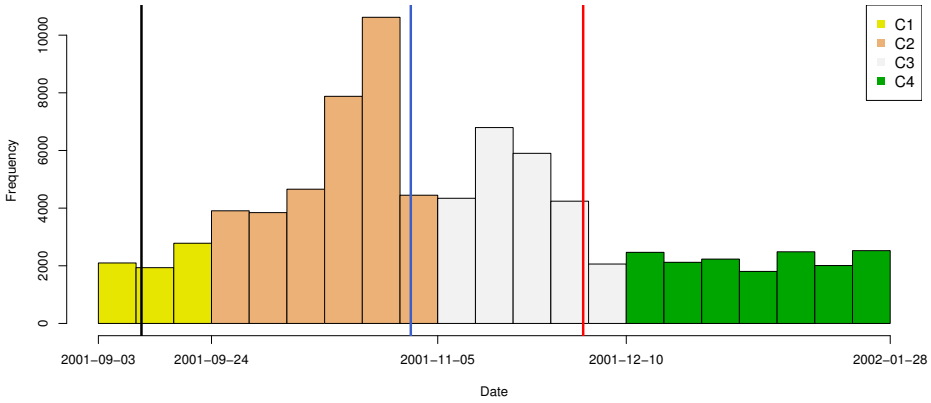


Figure 4.3 – *Time clustering results obtained by ICL-dSTBM for the Enron data set (Sept. 2001 - Jan. 2002). The black vertical line marks the day September, 11, 2001, the blue vertical line marks the day October, 31st, 2001 (investigation opened by the SEC), the red vertical line marks the day December, 2nd, 2001 (Enron's bankruptcy).*

Thus, each week is associated with a graph and one directed edge of such graph, from i to j, corresponds to the e-mails sent from i to j during the week. The whole dynamic graph is made of 4321 directed edges, corresponding to the same number of exchanged documents. The dictionary associated to these documents contains 49955 words.

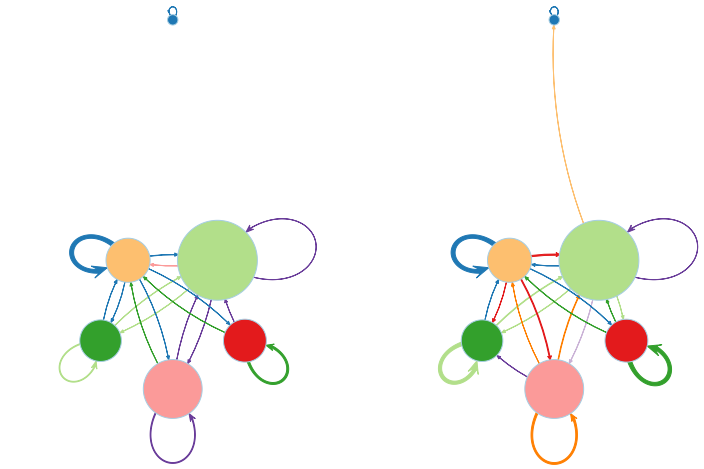
4.5.2 Results

The VEM algorithm for dSTBM was run on this dataset for all values of K, D and Q varying between 1 and 10, corresponding to 1000 runs and ICL finally selected nine topics $(Q = 9)$, six groups $(K = 6)$ and four time clusters $(D = 4)$. For these values of K, D and Q the algorithm was run several times, corresponding to different initializations and the clustering results associated with the highest value of the ICL criterion were finally retained.

In Figure 4.3, an histogram reports the frequency of exchanged e-mails in the whole graph, each rectangle covers one week. Rectangles/weeks of the same color are assigned to the same time cluster by dSTBM. Notice that, although in dSTBM time intervals in the same cluster do *not* have to be adjacent, the clustering reported in Figure 4.3 clearly detects four segments of adjacent time intervals and three corresponding change points, one for each color change. It is worth noticing that the three change points occur some days after the three key dates mentioned at the beginning of the present section and represented in the figure by three vertical lines, black, blue and red, respectively.

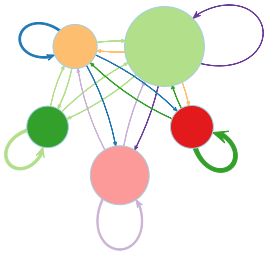
Figure 4.4 summarizes the main clustering results. Four graphs are associated with the time clusters detected by the model. Each node in a graph corresponds to a cluster of vertices and node sizes are proportional to group membership probabilities π. Edge colors indicate the main topic associated with group interactions. The larger the arrow is, the more frequent the respective interactions are.

Some remarks can be made by looking at this figure.

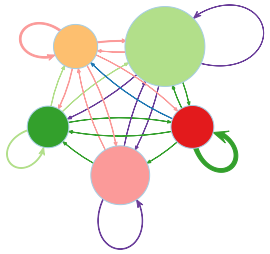


(a) *Time cluster* $\mathcal{C}_1$.

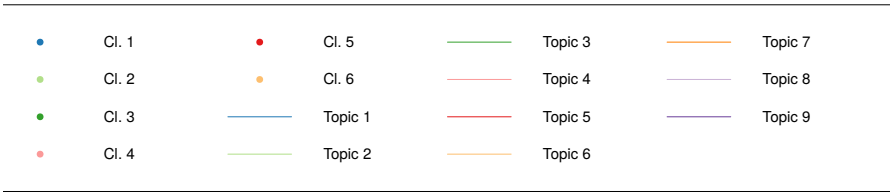
(b) *Time cluster* $\mathcal{C}_2$.



(c) *Time cluster* $\mathcal{C}_3$



(d) *Time cluster* $\mathcal{C}_4$



(e) *Legend.*

Figure 4.4 – *Summary of the interaction intensities* (λ, *edge widths), group proportions* (π, *node size) and majority topic for group interactions (edge colors) during each time cluster.*

.

1. Consider Group 4, consisting of 32 agents (mainly vice presidents, CEOs and managers). The topic used by this group for internal communications changes on each time segment: topic 9 in time clusters 1 and 4, topic 7 in time clusters 2, topic 8 in time cluster 3. Figure 4.5 shows the most representative words of each topic and can be used in the attempt to understand the main theme of each topic. Topic 7

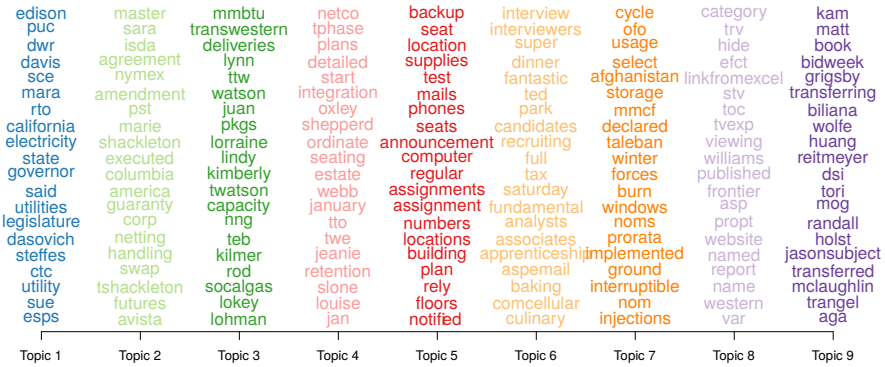


Figure 4.5 – *The 20 most representative words for each topic.*

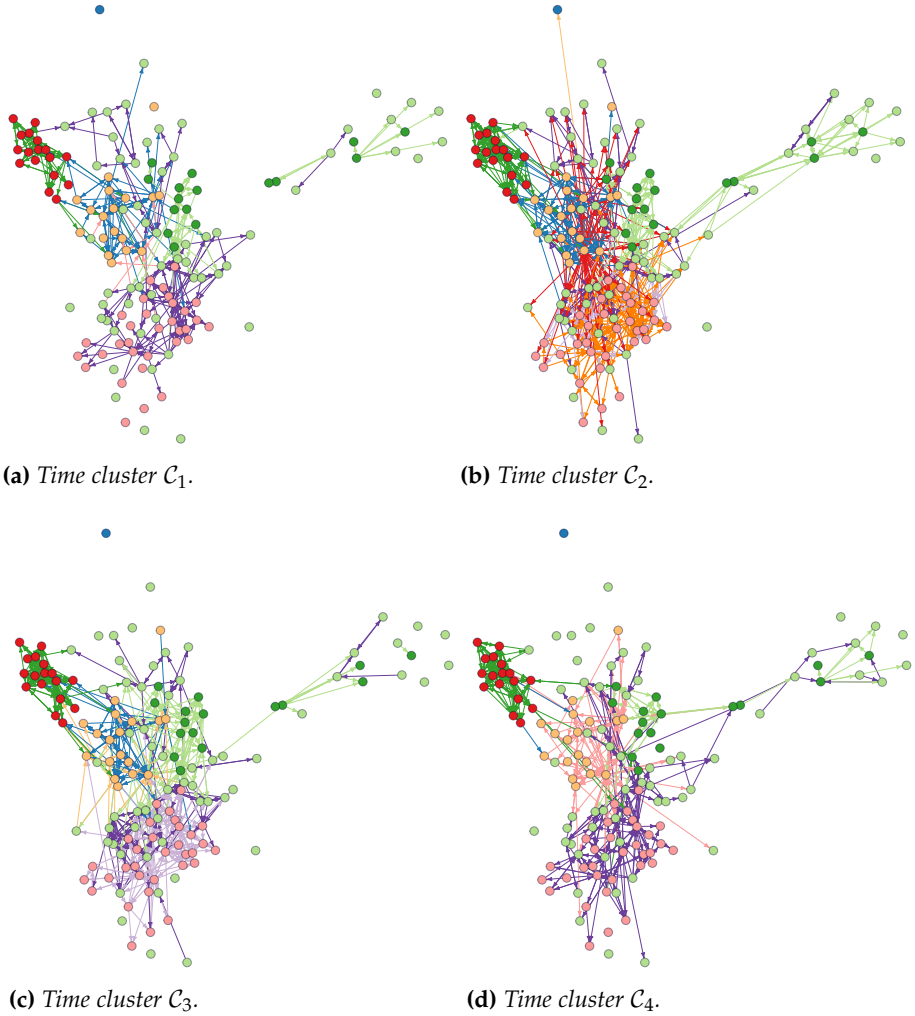
is discussed in details in the following. Topic 9 seems to be related to trading activities, as the words "book","transferring" and "bid week" suggest. The bid week, in particular, is the last week of the month when producers sell their core production and consumers buy for their core natural gas needs for the upcoming month. Topic 8 is very difficult to decode. It seems to focus on TRV (Trader Report Viewer), a project allowing traders to share their reports about particular issues. For example, an e-mail dating November, 13, 2001 announced to several employees that a report on West NG (West Virginia Natural Gas) prices was available. A "link from Excel" was provided in the e-mail.

2. Topic 7 contains words like "afghanistan" and "taleban" and it is clearly related to Enron activities in Afghanistan: Enron and the Bush administration were suspected to work secretly with Talebans before the 9/11 attacks. It is interesting to observe that this topic appears in the graph during time cluster $\mathcal{C}_2$, starting on September, 24th, 2001, exactly two weeks after the 9/11 attacks.

3. Topic 5, only used for communication between clusters during the time segment $\mathcal{C}_2$ is related to a backup plan developed to face possible work stoppages. In fact, some areas of the Enron Center North building were put aside for recovery purposes and backup seats assignments were announced to employees in November 2001. Topic 5 as well as Topic 7 disappeared during the other time clusters.

4. Made of 18 components, with a similar composition of Group 4, Group 6 mainly uses Topic 1 during the first three time clusters and switches to Topic 4 after the company bankruptcy, during the fourth segment. Topic 1 is related to the California electricity crisis in which Enron was involved and which almost caused the bankruptcy of the SCE-corp (Southern California Edison Corporation). Topic 4 seems to be related to Netco, a set of trading activities bought by the Swiss bank UBS after the Enron bankruptcy.

5. Group 5, 17 employees, looks like a real persistent community both



(a) *Time cluster* $\mathcal{C}_1$. **(b)** *Time cluster* $\mathcal{C}_2$.

(c) *Time cluster* $\mathcal{C}_3$. **(d)** *Time cluster* $\mathcal{C}_4$.

Figure 4.6 – *Clustering results obtained by ICL-dSTBM for the Enron data set (Sept. 2001 - Jan. 2002). Each graph corresponds to a time cluster.*

in terms of interactivity pattern and used topic. This group focuses during the whole time horizon on the technical Topic 3, about gas deliveries (mmBTU are British thermal units).

Finally, Figure 4.6 shows four snapshots of the original Enron dataset. Each snapshot is obtained by aggregating the interactions/e-mails over the corresponding time cluster. Vertices of the same color are assigned to the same cluster by the C-VEM algorithm and edges of the same color are associated with the same majority topic on the considered time cluster.

4.6 Conclusion

We introduced in this chapter the dynamic stochastic topic block model (dSTBM), a new probabilistic model aiming to cluster both vertices and edges of a textual dynamic network. Moreover, relying on an external time partition, it allows to uncover time clusters during which the graph is stationary both in terms of structure (interaction frequency between groups of nodes) and discussed topics. The inference procedure relies

on a classification VEM approach and an ICL model selection criterion is developed in order to estimate the number of node groups, time clusters and discussed topics. Numerical experiments on simulated data allowed us to highlight the main features of the proposed methodology, which proves to generalize several existing approaches. Finally, the application of dSTBM to the Enron communication network leaded to meaningful results.

Future researches could focus on a "clever" way to set a time partition, either including this partition between the model parameters or adopting a data driven choice (as done by Matias et al. 2015, for a dynamic SBM-like model). Alternatively, the dSTBM model could be extended to deal with overlapping clusters, allowing individuals to belong to multiple groups. In this context, a starting point could be the mixed memberships SBM (MMSBM, Airoldi et al. 2008).

4.7 Proofs

4.7.1 Proof of Proposition 4.1

Proof. The VEM update step for the distribution $R(V_n^{iju})$, for all i, j, u and n, is given by

$$\begin{aligned}\log R(V_n^{iju}) &= \mathbf{E}_{R(V^{\setminus i,j,u,n},\theta)}[\log p(W|V,\boldsymbol{X},\beta) + \log p(V|\boldsymbol{X},Z,Y,\theta)] + C \\ &= \sum_{q=1}^{Q} V_{nq}^{iju} \sum_{w=1}^{T^{(W)}} W_{nw}^{iju} \log \beta_{qw} \\ &+ \sum_{k,g}^{K}\sum_{d}^{D} Z_{ik} Z_{jg} Y_{ud} \sum_{q=1}^{Q} V_{nq}^{iju} \mathbf{E}_{\theta_{kgd}}[\log \theta_{kgd}] + C \\ &= \sum_{q=1}^{Q} V_{nq}^{iju} \left(\sum_{w=1}^{T^{(W)}} W_{nw}^{iju} \log \beta_{qw} + \sum_{k,g}^{K}\sum_{d}^{D} Z_{ik} Z_{jg} Y_{ud} \mathbf{E}_{\theta_{kgd}}[\log \theta_{kgd}] \right) \\ &+ C,\end{aligned} \tag{4.15}$$

where the expectation is taken with respect to the distribution $R(V, \theta)$ conditional on V_n^{iju} to be fixed and C includes all the terms not depending on V_n^{iju}. The functional form of a multinomial distribution can be recognised

$$R(V_n^{iju}) = \mathcal{M}\left(V_n^{iju}; 1, \phi_n^{iju} = \{\phi_{n1}^{iju}, \ldots, \phi_{nQ}^{iju}\}\right),$$

where

$$\phi_{nq}^{iju} \propto \left(\prod_{w=1}^{T^{(W)}} \beta_{qw}^{W_{nw}^{iju}}\right) \prod_{k,g}^{K}\prod_{d}^{D} \exp\left(\psi(\gamma_{kgdq}) - \psi(\sum_{q=1}^{Q} \gamma_{kgdq})\right)^{Z_{ik} Z_{jg} Y_{ud}},$$

recalling that

$$\mathbf{E}_{\theta_{kgd}}[\log \theta_{kgd}] = \psi(\gamma_{kgdq}) - \psi(\sum_{q=1}^{Q} \gamma_{kgdq}),$$

where $\psi(\cdot)$ denotes the digamma function. □

4.7.2 Proof of Prososition 4.2

Proof. The VEM update step for distribution $R(\theta)$ is given by

$$\begin{aligned}\log R(\theta) &= \mathbf{E}_{R(V)}[\log p(V|\boldsymbol{X},Z,Y,\theta)] + C \\ &= \sum_{j\neq i}^{N}\sum_{u=1}^{U}\sum_{n=1}^{L_{iju}}\sum_{k,g}^{K}\sum_{d}^{D} Z_{ik} Z_{jg} Y_{ud} \sum_{q=1}^{Q} \mathbf{E}_{R(V)}[V_{nq}^{iju}] \log \theta_{kgdq} \\ &+ \sum_{k,g}^{K}\sum_{d}^{D}\sum_{q=1}^{Q} (\alpha_q - 1) \log \theta_{kgdq} + C \\ &= \sum_{k,g}^{K}\sum_{d}^{D}\sum_{q=1}^{Q} \left(\alpha_q + \sum_{j\neq i}^{N}\sum_{u=1}^{U}\sum_{d=1}^{N_{iju}} Z_{ik} Z_{jg} Y_{ud} \phi_{nq}^{iju} - 1\right) \log \theta_{kgdq} + C,\end{aligned} \tag{4.16}$$

where C contains those terms not depending on θ. The functional form of a Dirichlet distribution can be recognized

$$R(\theta) = \prod_{k,g}^{K} \prod_{d}^{D} \text{Dir}(\theta_{kgd}; \gamma_{kgd} = \{\gamma_{kgd1}, \dots, \gamma_{kgdQ}\}),$$

with

$$\gamma_{kgdq} = \alpha_q + \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} Z_{ik} Z_{jg} Y_{ud} \phi_{nq}^{iju}.$$

□

4.7.3 Derivation of the lower bound

Provided $R(V, \theta)$ given in Proposition 4.2 and Proposition 4.3, the functional $\tilde{\mathcal{L}}(R(\cdot); \boldsymbol{X}, W, Z, Y, \beta)$ in (4.9) is given by

$$\begin{aligned}
\tilde{\mathcal{L}}(R(\cdot); \boldsymbol{X}, W, Z, Y, \beta) = & \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} \sum_{q=1}^{Q} \sum_{w=1}^{T^{(W)}} W_{nw}^{iju} \phi_{nq}^{iju} \log(\beta_{qw}) \\
& + \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} \sum_{q=1}^{Q} \phi_{nq}^{iju} \left(\sum_{k,g}^{K} \sum_{d}^{D} Z_{ik} Z_{jg} Y_{ud} \left(\psi(\gamma_{kgdq}) - \psi(\sum_{q=1}^{Q} \gamma_{kgdq}) \right) \right) \\
& + \sum_{k,g}^{K} \sum_{d}^{D} \left(\log \Gamma(\sum_{q=1}^{Q} \alpha_q) - \sum_{q=1}^{q} \log \Gamma(\alpha_q) + \sum_{q=1}^{Q} (\alpha_q - 1) \left(\psi(\gamma_{kgdq}) - \psi(\sum_{q=1}^{Q} \gamma_{kgdq}) \right) \right) \\
& - \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} \sum_{q=1}^{Q} \phi_{nq}^{iju} \log(\phi_{nq}^{iju}) \\
& - \sum_{k,g}^{K} \sum_{d}^{D} \left(\log \Gamma(\sum_{q=1}^{Q} \gamma_{kgdq}) - \sum_{q=1}^{Q} \log \Gamma(\gamma_{kgdq}) + \sum_{q=1}^{Q} (\gamma_{kgdq} - 1) \left(\psi(\gamma_{kgdq}) - \psi(\sum_{q=1}^{Q} \gamma_{kgdq}) \right) \right)
\end{aligned}$$

where $\Gamma(\cdot)$ denotes the gamma function.

4.7.4 Proof of Proposition 4.3

Proof. The maximization of the functional in (4.9) with respect to β is considered at first. By isolating the terms depending on β and introducing Q Lagrange multipliers accounting for the constraints $\sum_{w=1}^{T^{(W)}} \beta_{qw} = 1$, $\forall q$, we obtain the following objective function

$$f(\beta) := \sum_{j \neq i}^{N} \sum_{u=1}^{U} \sum_{n=1}^{L_{iju}} \sum_{q=1}^{Q} \sum_{w=1}^{T^{(W)}} \phi_{nq}^{iju} \log \beta_{qw} + \sum_{q=1}^{Q} \lambda_q \left(\sum_{q=1}^{Q} \beta_{qw} - 1 \right),$$

whose gradient can be easily computed and set equal to zero to find the β_{qw} in (4.10).

In a similar fashion, when optimizing with respect to π, the following objective function is introduced

$$f(\pi) := \sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} \log \pi_k + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \tag{4.17}$$

and its first derivative with respect to π_k is set equal to zero to obtain the stationary point in (4.12). The optimization with respect to ρ is analogous and (4.11) is a consequence of the likelihood in (4.2). □

4.7.5 Proof of Proposition 4.4

Proof. A factorizing prior distribution being attached to the model parameters, $(\boldsymbol{\lambda}, \pi, \rho, \beta)$, the integrated complete-data log-likelihood $\log p(\boldsymbol{X}, W, Z, Y|K, D, Q)$ can easily be written as

$$\begin{aligned}\log p(\boldsymbol{X}, W, Z, Y|K, D, Q) = &\log \int_{\beta} p(W|\boldsymbol{X}, Z, Y, \beta, K, D, Q)p(\beta|Q)d\beta \\ &+ \log \int_{\boldsymbol{\lambda}} p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda}, K, D)p(\boldsymbol{\lambda}|K, D)d\boldsymbol{\lambda} \\ &+ \log \int_{\pi} p(Z|\pi, K)p(\pi|K)d\pi \\ &+ \log \int_{\rho} p(Y|\rho, D)p(\rho|D)d\rho,\end{aligned} \tag{4.18}$$

where the dependency on (K, D, Q) is made explicit and the pair (V, θ) is integrated out as in Section 4.3.1. Following the derivation of the ICL criterion (Section 1.3.2), we rely on a BIC-like approximation of the second term on the right hand side of the above equation to obtain

$$\begin{aligned}\log \int_{\boldsymbol{\lambda}} p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda}, K, D)p(\boldsymbol{\lambda}|K, D)d\boldsymbol{\lambda} \approx &\max_{\boldsymbol{\lambda}} \log p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda}, K, D) \\ &- \frac{K^2 D}{2} \log(NU(N-1)).\end{aligned}$$

Similarly the last two terms can be approximated as

$$\log \int_{\pi} p(Z|\pi, K)p(\pi|K)d\pi \approx \max_{\pi} \log p(Z|\pi, K) - \frac{K-1}{2} \log(N)$$

and

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

Notice that the last three approximations lead to the ICL criterion for dSBM

$$\begin{aligned}ICL_{dSBM} := &\max_{\boldsymbol{\lambda}} \log p(\boldsymbol{X}|Z, Y, \boldsymbol{\lambda}, K, D) - \frac{K^2 D}{2} \log(NU(N-1)) \\ &+ \max_{\pi} \log p(Z|\pi, K) - \frac{K-1}{2} \log(N) \\ &+ \max_{\rho} \log p(Y|\rho, D) - \frac{D-1}{2} \log(U).\end{aligned}$$

Notice also that the exact version of this criterion was maximized relying on a greedy search approach in Chapter 2 for CdSBM.

Consider now the first term on the right hand side of (4.18). Recalling that the documents W can be organized as $W = (\tilde{W}_{kgd})_{k,g,d}$ such that all words in $\tilde{W}_{kgd}$ follow the same mixture distribution over topics, we adopt the BIC-like approximation obtained in Bouveyron et al. (2016) corrected by the number of documents in dSTBM

$$\begin{aligned}\log \int_{\beta} p(W|\boldsymbol{X}, Z, Y, \beta, K, D, Q)p(\beta|Q)d\beta \approx &\max_{\beta} \log p(W|\boldsymbol{X}, Z, Y, \beta, K, D, Q) \\ &- \frac{Q(T^{(W)}-1)}{2} \log(K^2 D).\end{aligned}$$

Since the first term on the right hand side of the above approximation is not tractable, it is replaced by its variational approximation $\tilde{\mathcal{L}}(R(\cdot); \boldsymbol{X}, W, Z, Y, \beta)$, defined in (4.9), and the proposition is proven. □

Conclusion

Due to the increasing amount of network data sets available, the statistical analysis of graphs is more and more developed. In this thesis we focused on dynamic graph analysis. Two different definitions of dynamic graph were provided, in continuous and discrete time, respectively. Both the definitions were employed to develop new unsupervised methods to

1. cluster the vertices of a dynamic graph in classes of homogeneous interactivity patterns and
2. detect structural changes in the way the node groups interact with each other.

The building block of this thesis is the stochastic block model (SBM), a mixture model that assigns the vertex of a graph to hidden, disjoint groups. The probability that one edge occurs between two nodes only depends on their groups and since no further assumption is formulated about the interaction probability, SBM is a very flexible tool accounting for different topological structures.

The original SBM (Holland et al. 1983, Wang and Wong 1987, Nowicki and Snijders 2001) does not apply to dynamic graphs. Therefore, after illustrating several approaches introduced in the literature to extend SBM to dynamic graphs, in Chapter 2 we introduced our dynamic model, called dSBM. In dSBM the vertices of a dynamic graph are grouped into hidden clusters not time varying and the interactions between each pair of nodes are modelled via a non-homogeneous Poisson process (NHPP) whose intensity function only depends on the corresponding nodes clusters. All the NHPPs are conditionally independent, given the groups. In a first time, we focused on a discrete time framework by introducing a partition of the whole time horizon. Interactions were aggregated on each time interval of the partition and the intensity functions of the NHPPs were assumed constant over hidden time clusters. Each time cluster contains some intervals of the time partition. The time intervals in the same time cluster do *not* have to be adjacent. The inference for dSBM is based on the greedy maximization of the exact integrated classification likelihood (ICL). This technique allowed us to perform clustering and model selection at the same time.

In Chapter 3, dSBM was extended in order to model continuous time dynamic graphs and not requiring any data aggregation. We developed an exact algorithm for change point detection in graph data, based on the pruned exact linear time (PELT) method for univariate time series and we saw that a time segment can be seen as a time cluster only containing *adjacent* time intervals. A variational expectation maximization (VEM) algorithm was used to estimate the model parameters and a BIC-based criterion was employed for model selection.

Finally, in Chapter 4 we focused on communication networks (e.g. social networks like Facebook, Twitter, etc.) and extended dSBM in order to model such networks accounting for the annexed textual information. The dynamic stochastic topic block model (dSTBM) we introduced associates one interaction in a dynamic graph with an exchanged document. The words of the document are assumed to follow a mixture distribution over latent topics. Being a generalisation of the latent Dirichlet allocation (LDA) method, our approach defines node groups that are homogeneous both in terms of connection profiles and used topics. Moreover, a dynamic graph is considered stationary on a hidden time segment if the interaction intensities and the used topics do not change over the segment. A variational EM was used for the inference and the number of topics, time clusters and node groups was selected via an ad-hoc model selection criterion.

Perspectives

As mentioned in the previous chapters, there are several possible directions for future researches. Mainly:

1. For dSBM/CdSBM in discrete time it would be crucial to include the time partition between the model parameters and infer the best partition directly from the data. Furthermore, in order to make the model more realistic, the assumption of common discontinuity points should certainly be relaxed. Indeed, in real networks, some external events could affect the behaviour of some groups uniquely. Hence, we could imagine discontinuity points that are specific to each pair of node groups and not common to the whole graph. Not even in their number.

2. About dSBM in continuous time, we introduced an exact change point detection algorithm that can be adopted to perform change point analysis in graph data. The algorithm was speeded up via *pruning* and in future work we will investigate in more details the resulting computational complexity. In particular, it is crucial to understand if the complexity has an upper bound linear in U (for a given K) and under what conditions.

3. One of the main drawbacks in the inference procedure for dSTBM is the model selection, because the number of topics must be estimated in addition to the number of cluster and node clusters in dSBM. Hence, it would be relevant to develop a faster model selection criterion, perhaps based on a greedy strategy as done in Chapter 2 for dSBM.

Bibliography

M. Knott A. J. Scott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30(3):507–512, 1974. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2529204. (Cited in page 28.)

D. B. Rubin A. P. Dempster, N. M. Laird. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. URL http://www.jstor.org/stable/2984875. (Cited in page 22.)

Massil Achab, Emmanuel Bacry, Stéphane Gaïffas, Iacopo Mastromatteo, and Jean-Francois Muzy. Uncovering causality from multivariate hawkes integrated cumulants. *arXiv preprint arXiv:1607.06333*, 2016. (Cited in page 21.)

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. (Cited in pages 16, 18, and 115.)

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974. (Cited in page 24.)

R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Modern Physics*, 74:47–97, 2002. (Cited in page 11.)

R. Albert, H. Jeong, and A.L. Barabasi. Diameter of the world-wide web. *Nature*, 401:130–131, 1999. (Cited in page 13.)

L.A.N Amaral, A. Scala, M. Barthélémy, and H.E. Stanley. Classes of small-world networks. In *Proceedings of the National Academy of Sciences*, volume 97, pages 11149–11152, 2000. (Cited in page 13.)

A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999. (Cited in page 13.)

A.L. Barabási and Z.N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Rev. Genet*, 5:101–113, 2004. (Cited in page 11.)

Richard Bellman. The theory of dynamic programming. Technical report, RAND CORP SANTA MONICA CA, 1954. (Cited in page 28.)

Peter Bickel, David Choi, Xiangyu Chang, and Hai Zhang. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *Ann. Statist.*, 41(4):1922–1943, 08 2013. doi: 10.1214/13-AOS1124. URL http://dx.doi.org/10.1214/13-AOS1124. (Cited in page 23.)

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. (Cited in page 16.)

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7):719–725, 2000. (Cited in pages 24 and 25.)

D. Blei and J. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35, 2007. (Cited in page 31.)

D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4): 77–84, 2012. (Cited in page 31.)

David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006. (Cited in pages 96 and 100.)

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944937`. (Cited in pages 30, 31, and 95.)

Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. (Cited in pages 16 and 47.)

M. Boullé. *Data grid models for preparation and modeling in supervised learning*. Microtome, 2010. (Cited in pages 20 and 78.)

Charles Bouveyron, Pierre Latouche, and Rawya Zreik. The stochastic topic block model for the clustering of vertices in networks with textual edges. *Statistics and Computing*, 2016. doi: 10.1007/s11222-016-9713-7. URL `https://hal.archives-ouvertes.fr/hal-01299161`. (Cited in pages 95, 99, 102, 104, 105, and 118.)

Vincent Brault and Antoine Channarond. Fast and consistent algorithm for the latent block model. *arXiv preprint arXiv:1610.09005*, 2016. (Cited in page 17.)

A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012. doi: 10.1080/17445760.2012.668546. (Cited in pages 13 and 14.)

G. Celeux and G. Soromenho. An entropy criterion for assessing the number of clusters in a mixture model. *Classification Journal*, 13:195–212, 1996. (Cited in page 24.)

Gilles Celeux and Gérard Govaert. A classification EM algorithm for clustering and two stochastic versions. Research Report RR-1364, INRIA, 1991. URL `https://hal.inria.fr/inria-00075196`. Projet CLOREC. (Cited in page 102.)

Alain Celisse, Jean-Jacques Daudin, and Laurent Pierre. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electron. J. Statist.*, 6:1847–1899, 2012. doi: 10.1214/12-EJS729. URL http://dx.doi.org/10.1214/12-EJS729. (Cited in page 23.)

J. Chang and D. M. Blei. Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, pages 81–88, 2009. (Cited in page 31.)

Gerda Claeskens and Nils Lid Hjort. *Model selection and model averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge Univ. Press, Leiden, 2008. URL https://cds.cern.ch/record/1251912. (Cited in page 24.)

Etienne Côme and Pierre Latouche. Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Statistical Modelling*, 15(6):564–589, 2015. doi: 10.1177/1471082X15577017. (Cited in pages 22, 41, 42, 47, and 56.)

Marco Corneli, Pierre Latouche, and Fabrice Rossi. Block modelling in dynamic networks with non-homogeneous poisson processes and exact ICL. *Social Network Analysis and Mining*, 6(1):1–14, 2016a. ISSN 1869-5469. doi: 10.1007/s13278-016-0368-3. URL http://dx.doi.org/10.1007/s13278-016-0368-3. (Cited in pages 3 and 7.)

Marco Corneli, Pierre Latouche, and Fabrice Rossi. Exact ICL maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. *Neurocomputing*, 192:81 – 91, 6 2016b. ISSN 0925-2312. doi: 10.1016/j.neucom.2016.02.031. (Cited in pages 3, 7, and 18.)

Marco Corneli, Pierre Latouche, and Fabrice Rossi. Multiple change points detection and clustering in dynamic networks. *Statistics and Computing*, Sep 2017. ISSN 1573-1375. doi: 10.1007/s11222-017-9775-1. URL https://doi.org/10.1007/s11222-017-9775-1. (Cited in pages 3 and 7.)

J-J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and Computing*, 18(2):173–183, 2008. (Cited in pages 18, 22, 23, 25, 71, and 73.)

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. (Cited in pages 20 and 21.)

Daniele Durante, David B Dunson, et al. Locally adaptive dynamic networks. *The Annals of Applied Statistics*, 10(4):2203–2232, 2016. (Cited in pages 20 and 21.)

Idris A Eckley, Paul Fearnhead, and Rebecca Killick. Analysis of changepoint models. *Bayesian Time Series Models.*, pages 205–224, 2011. (Cited in page 28.)

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486 (3-5):75 – 174, 2010. ISSN 0370-1573. (Cited in page 16.)

Nial Friel, Riccardo Rastelli, Jason Wyse, and Adrian E. Raftery. Interlocking directorates in irish companies using a latent space model for bipartite networks. *Proceedings of the National Academy of Sciences*, 113 (24):6629–6634, 2016. doi: 10.1073/pnas.1606295113. (Cited in pages 20 and 21.)

A. Goldenberg, X. Zheng, Sthephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Machine Learning*, 2(2):129–133, 2009. doi: 10.1561/2200000005. (Cited in page 16.)

Bettina Grün and Kurt Hornik. topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13):1–30, 2011. doi: 10.18637/jss.v040.i13. (Cited in page 105.)

Romain Guigourès, Marc Boullé, and Fabrice Rossi. A triclustering approach for time evolving graphs. In *Co-clustering and Applications, IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012)*, pages 115–122, Brussels, Belgium, 12 2012. ISBN 978-1-4673-5164-5. doi: 10.1109/ICDMW.2012.61. (Cited in pages 20, 21, 78, 79, and 82.)

Romain Guigourès, Marc Boullé, and Fabrice Rossi. Discovering patterns in time-varying graphs: a triclustering approach. *Advances in Data Analysis and Classification*, pages 1–28, 2015. ISSN 1862-5347. doi: 10.1007/s11634-015-0218-6. URL `http://dx.doi.org/10.1007/s11634-015-0218-6`. (Cited in pages 14, 20, 21, 78, and 83.)

Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007. (Cited in page 19.)

Steve Hanneke, Wenjie Fu, Eric P Xing, et al. Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605, 2010. (Cited in pages 19 and 21.)

Alan G Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 438–443, 1971. (Cited in page 21.)

Qirong Ho, Le Song, and Eric P Xing. Evolving cluster mixed-membership blockmodel for time-evolving networks. In *International Conference on Artificial Intelligence and Statistics*, pages 342–350, 2011. (Cited in pages 19 and 21.)

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. (Cited in page 19.)

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. (Cited in page 30.)

P.W. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5:109–137, 1983. (Cited in pages 17 and 121.)

Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271 (1):166–180, 2011. ISSN 0022-5193. doi: DOI:10.1016/j.jtbi.2010.11.033. (Cited in pages 11 and 57.)

B. Jackson, J.D. Sargle, D. Barnes, S. Arabhi, A. Alt, P. Giomousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T.T. Tsai. An algorithm for optimal partitioning of data on an interval. *Signal Processing Letters*, pages 105–108, 2005. (Cited in pages 28, 74, 75, and 76.)

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. (Cited in page 18.)

Olav Kallenberg. *Foundations of modern probability*. Springer Science & Business Media, 2006. (Cited in page 25.)

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011. doi: 10.1103/PhysRevE.83.016107. URL `https://link.aps.org/doi/10.1103/PhysRevE.83.016107`. (Cited in page 18.)

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012. doi: 10.1080/01621459.2012.737745. URL `http://dx.doi.org/10.1080/01621459.2012.737745`. (Cited in pages 27, 29, 30, 65, 75, and 76.)

M. Kim and J. Leskovec. Nonparametric multi-group membership model for dynamic networks. In *Advances in Neural Information Processing Systems (25)*, pages 1385–1393, 2013. (Cited in pages 19 and 21.)

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. (Cited in page 78.)

Pavel N Krivitsky and Mark S Handcock. A separable model for dynamic networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):29–46, 2014. (Cited in pages 20 and 21.)

P. Latouche, E Birmelé, and C. Ambroise. Variational bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12(1):93–115, 2012. (Cited in page 22.)

Pierre Latouche, Etienne Birmelé, and Christophe Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, pages 309–336, 2011. (Cited in page 18.)

Lawrence M. Leemis. Nonparametric estimation of the cumulative intensity function for a nonhomogeneous poisson process. *Management Science*, 37(7):886–900, 1991. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2632541. (Cited in page 48.)

P.A.W. Lewis and G.S. Shedler. Simulation of nonhomogeneous poison processes by thinning. *Naval Res. Logist. Quart.*, 26(3):403–413, 1979. (Cited in page 81.)

Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: Joint models of topic and author community. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 665–672, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553460. URL http://doi.acm.org/10.1145/1553374.1553460. (Cited in page 95.)

C. Matias, T. Rebafka, and F. Villers. Estimation and clustering in a semiparametric Poisson process stochastic block model for longitudinal networks. *ArXiv e-prints*, December 2015. (Cited in pages 20, 21, 50, and 115.)

Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1119–1141, 2017. (Cited in pages 19 and 21.)

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. (Cited in page 22.)

A. McCallum, A. Corrada-Emmanuel, and X. Wang. The author-recipient-topic model for topic and role discovery in social networks. In *Workshop on Link Analysis, Counterterrorism and Security*, 2005. (Cited in page 95.)

J.L. Moreno. *Who shall survive?: A new approach to the problem of human interrelations.* Nervous and Mental Disease Publishing Co, 1934. (Cited in page 11.)

Radford M Neal and Geoffrey 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. (Cited in page 23.)

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004. doi: 10.1103/PhysRevE.69.026113. URL http://link.aps.org/doi/10.1103/PhysRevE.69.026113. (Cited in page 16.)

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. (Cited in page 13.)

Andreas Noack and Randolf Rotta. Multi-level algorithms for modularity clustering. *CoRR*, abs/0812.4073, 2008. URL http://arxiv.org/abs/0812.4073. (Cited in pages 16 and 47.)

James R. Norris. *Markov chains.* Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1998. ISBN 978-0-521-48181-6. (Cited in page 25.)

Laetitia Nouedoui and Pierre Latouche. Bayesian non parametric inference of discrete valued networks. In *21-th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013)*, pages 291–296, Bruges, Belgium, 2013. (Cited in page 51.)

K. Nowicki and T.A.B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455): 1077–1087, 2001. (Cited in pages 11, 17, 22, and 121.)

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. (Cited in page 11.)

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. (Cited in page 30.)

Nishith Pathak, Colin DeLong, Arindam Banerjee, and Kendrick Erickson. Social topic models for community extraction. In *The 2nd SNA-KDD workshop*, volume 8, 2008. (Cited in page 95.)

Franck Picard, Stephane Robin, Marc Lavielle, Christian Vaisse, and Jean-Jacques Daudin. A statistical approach for array cgh data analysis. *BMC bioinformatics*, 6(1):27, 2005. (Cited in page 28.)

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007. ISBN 0521880688. (Cited in page 62.)

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. (Cited in pages 51 and 79.)

Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. *Social networks*, 29(2):173–191, 2007. (Cited in page 19.)

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6. URL `http://dl.acm.org/citation.cfm?id=1036843.1036902`. (Cited in page 95.)

Mrinmaya Sachan, Danish Contractor, Tanveer A. Faruquie, and L. Venkata Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 331–340, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/

2187836.2187882. URL http://doi.acm.org/10.1145/2187836.2187882. (Cited in page 95.)

Purnamrita Sarkar and Andrew W Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7 (2):31–40, 2005. (Cited in pages 20 and 21.)

Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6 (2):461–464, 03 1978. doi: 10.1214/aos/1176344136. URL http://dx.doi.org/10.1214/aos/1176344136. (Cited in page 24.)

Daniel K Sewell and Yuguo Chen. Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657, 2015. (Cited in pages 20 and 21.)

Daniel K Sewell and Yuguo Chen. Latent space models for dynamic networks with weighted edges. *Social Networks*, 44:105–116, 2016. (Cited in pages 20 and 21.)

Tom AB Snijders. Stochastic actor-oriented models for network change. *Journal of mathematical sociology*, 21(1-2):149–172, 1996. (Cited in page 20.)

David Snyder and Edward L. Kick. Structural position in the world system and economic growth, 1955-1970: A multiple-network analysis of transnational interactions. *American Journal of Sociology*, 84(5):pp. 1096–1126, 1979. ISSN 00029602. URL http://www.jstor.org/stable/2778218. (Cited in page 11.)

Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 306–315, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014087. URL http://doi.acm.org/10.1145/1014052.1014087. (Cited in page 95.)

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. (Cited in page 31.)

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. (Cited in page 31.)

W. A. Thompson. *Point process models with applications to safety and reliability*. Chapman & Hall Ltd, London, New York, 1988. (Cited in page 25.)

N. Villa, F. Rossi, and Q.D. Truong. Mining a medieval social network by kernel som and related methods. *Arxiv preprint arXiv:0805.1374*, 2008. (Cited in page 11.)

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. ISSN 1573-1375. doi: 10.1007/s11222-007-9033-z. URL http://dx.doi.org/10.1007/s11222-007-9033-z. (Cited in page 16.)

Y.J. Wang and G.Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987. (Cited in pages 17 and 121.)

Jason Wyse, Nial Friel, and Pierre Latouche. Inferring structure in bipartite networks using the latent blockmodel and exact icl. *Network Science*, 5 (1):45–69, 2017. (Cited in page 41.)

Eric P. Xing, Wenjie Fu, and Le Song. A state-space mixed membership blockmodel for dynamic network tomography. *Ann. Appl. Stat.*, 4(2): 535–566, 06 2010. doi: 10.1214/09-AOAS311. (Cited in pages 19 and 21.)

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1717–1726, 2016. (Cited in page 21.)

Kevin S Xu and Alfred 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. (Cited in pages 19 and 21.)

Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. Detecting communities and their evolutions in dynamic social networks a bayesian approach. *Machine learning*, 82(2):157–189, 2011. (Cited in pages 19 and 21.)

Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 173–182, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135807. URL http://doi.acm.org/10.1145/1135777.1135807. (Cited in page 95.)

M. Zhou. *Empirical Likelihood Method in Survival Analysis*. Chapman & Hall/CRC Biostatistics Series. CRC Press, 2015. ISBN 9781466554931. URL https://books.google.fr/books?id=9-b5CQAAQBAJ. (Cited in page 26.)

Rawya Zreik, Pierre Latouche, and Charles Bouveyron. The dynamic random subgraph model for the clustering of evolving networks. *Computational Statistics*, pages 1–33, 2016. ISSN 1613-9658. doi: 10.1007/s00180-016-0655-5. URL http://dx.doi.org/10.1007/s00180-016-0655-5. (Cited in pages 19 and 21.)