Modèles de graphes aléatoires à structure cachée pour l'analyse des réseaux

Pierre Latouche

Université d'Évry Val d'Essonne
Laboratoire Statistique et Génome

Thèse

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

par

Pierre Latouche

Modèles de graphes aléatoires à structure cachée pour l'analyse des réseaux

Thèse soutenue le la date de soutenance devant le jury composé de :

M.	Christophe Biernacki	Université de Lille	(Rapporteur)
M.	Jean-Philippe Vert	Mines, Fontainebleau & Institut Curie, Paris	(Rapporteur)
M.	Geoff McLachlan	Université du Queensland, Australie	(Jury)
M.	Stéphane Robin	AgroParisTech, Paris	(Jury)
M.	Christophe Ambroise	Université d'Évry Val D'Essonne	(Directeur)
M.	Etienne Birmelé	Université d'Évry Val d'Essonne	(Co-Directeur)

À toute ma famille, affectueusement

Remerciements

Je tiens à remercier Christophe Ambroise et Etienne Birmelé pour avoir encadré ce travail de thèse et pour la liberté qu'ils m'ont donnée. Je remercie également Julien Chiquet, Marie-Agnès Dillies, Gilles Grasseau, Catherine Matias, et Bernard Prum. Chacun à votre manière, vous m'avez donné confiance en moi et aidé à avancer. Enfin, un grand remerciement à tous les membres du laboratoire Statistique et Génome. Merci pour tout.

P. Latouche, Evry, le November 30, 2010.

Contents

List of Figures

Préface

Les réseaux sont très largement utilisés dans de nombreux domaines scientifiques afin de représenter les intéractions entre objets d'intérêt. Ainsi, en Biologie, les réseaux de régulation s'appliquent à décrire les mécanismes de régulation des gènes, à partir de facteurs de transcription, tandis que les réseaux métaboliques permettent de représenter des voies de réactions biochimiques. En sciences sociales, ils sont couramment utilisés pour réprésenter les intéractions entre individus.

Dans le cadre de cette thèse, nous nous intéressons à des méthodes d'apprentissage non supervisé dont l'objectif est de classer les nœuds d'un réseau en fonction de leurs connexions. Il existe une vaste littérature se référant à ce sujet et un nombre important d'algorithmes ont été proposés depuis les premiers travaux de Moreno en 1934. Il apparaît dès lors que les approches existantes peuvent être regroupées dans trois familles différentes. Tout d'abord, un certain nombre de méthodes se concentrent sur la recherche de communautés où les nœuds sont classés de manière à ce que deux nœuds aient une plus forte tendance à se connecter s'ils appartiennent à la même classe. Ces techniques sont parmi les plus utilisées en particulier pour analyser les réseaux de type Internet. Par ailleurs, d'autres approches recherchent des structures topologiques différentes où, au contraire, deux nœuds ont une plus forte tendance à intéragir s'ils sont dans des classes distinctes. Elles sont particulièrement adaptées à l'étude des réseaux de type biparpite. Enfin, quelques méthodes s'intéressent à la recherche de structures hétérogènes où les nœuds peuvent avoir des profiles de connexion très différents. En particulier, ces techniques peuvent être employées pour retrouver à la fois des commautés et des structures bipartites dans les réseaux. Le point de départ de cette thèse est le modèle à blocs stochastiques, Stochastic Block Model (SBM) en anglais, qui appartient à cette dernière famille d'approches et qui est également très largement utilisé.

SBM est un modèle de mélange qui est le résultat de travaux en sciences sociales. Il fait l'hypothèse que les nœuds d'un réseau sont répartis dans des classes et décrit la probabilité d'apparition d'un arc entre deux nœuds uniquement en fonction des classes auquelles ils appartiennent. Aucune hypothèse n'ait faite *a priori* en ce qui concerne ces probabilités de connexion de sorte que SBM puisse prendre en compte des structures topologiques très variées. En particulier, le modèle permet de caractériser la présence de hubs, c'est à dire de nœuds ayant un nombre de liens élevés par rapport aux autres nœuds d'un réseau. Pour finir, il généralise la plupart des modèles existants pour la classification de nœuds dans les réseaux.

L'estimation des paramètres de SBM a déjà fait l'objet de nombreuses

études. Cependant, à notre connaissance un seul critère de sélection de modèles a été développé pour estimer le nombre de composantes du mélange (Daudin et al. 2008). Malheureusement, il a été montré que ce critère était trop conservateur dans le cas de réseaux de petites tailles.

Par ailleurs, il apparaît que SBM ainsi que la plupart des modèles existants pour la classification dans les réseaux sont limités puisqu'ils partitionnent les nœuds dans des classes disjointes. Or, de nombreux objets d'étude dans le cadre d'applications réelles sont connus pour appartenir à plusieurs groupes en même temps. Par exemple, en Biologie, des protéines appelées *moonlighting proteins* en anglais ont plusieurs fonctions dans les cellules. De la même manière, lorsqu'un ensemble d'individus est étudié, on s'attend à ce qu'un nombre conséquent d'individus appartiennent à plusieurs groupes ou communautés.

Dans cette thèse, notre contribution est la suivante:

- Nous proposons un nouvel algorithme de classification pour SBM ainsi qu'un nouveau critère de sélection de modèle. Ces travaux sont implémentés dans le package R "mixer" qui est disponible sur le CRAN: `http://cran.r-project.org/web/packages/mixer`.

- Nous introduisons un nouveau modèle de graphe aléatoire que nous appelons modèle à blocs stochastiques chevauchants, Overlapping Stochastic Block Model (OSBM) en anglais. Il autorise les nœuds d'un réseau à appartenir à plusieurs groupes simultanément et peut prendre en compte des topologies de connexion très différentes. La classification et l'estimation des paramètres de OSBM sont réalisées à partir d'un algorithme basé sur des techniques variationnelles.

- Nous présentons une nouvelle approche d'inférence pour OSBM ainsi qu'un nouveau critère de sélection de modèle qui permet d'estimer le nombre de classes, éventuellement chevauchantes, dans les réseaux. Ces travaux sont implémentés dans le package R "OSBM" qui sera très prochainement disponible sur le CRAN.

Le *premier chapitre* s'attache à présenter les principaux concepts statistiques et les méthodes existantes sur lesquels se fondent nos travaux. Nous introduisons notamment les modèles de mélange ainsi que les techniques d'inférence de type EM et variationnel. Plusieurs critères de sélection de modèle sont également discutés. Enfin, il est fait état des méthodes les plus connues pour la classification des nœuds dans les réseaux.

Dans le *deuxième chapitre*, le modèle à blocs stochastiques est décrit dans un cadre Bayésien et une nouvelle procédure d'inférence est proposée. Elle offre la possibilité d'approcher la loi *a posteriori* des paramètres et des variables latentes. Cette approche permet également d'obtenir un critère non asymptotique de sélection de modèle, basé sur une approximation de la vraissemblance marginale.

Le *troisième chapitre* introduit le modèle à blocs stochastiques chevauchants. Nous montrons que le modèle est identifiable dans des classes d'équivalence. De plus, un algorithme basé sur des techniques variationnelles locales et globales est proposé. Il permet le clustering de nœuds dans des classes chevauchantes et l'estimation des paramètres du modèle.

Enfin, le *quatrième chapitre* considère à nouveau un cadre Bayésien. Des lois *a priori* conjuguées sont utilisées pour caractériser les paramètres du modèle à blocs stochastiques chevauchants. Une procédure d'inférence permet alors d'approcher la loi *a posteriori* des paramètres et des variables latentes. Elle donne naissance à un critère de sélection de modèle basé sur une nouvelle approximation de la vraissemblance marginale.

Cette thèse a fait l'objet de deux papiers et d'un chapitre de livre: Latouche et al. (2009; 2010a;b).

Abstract

Networks are used in many scientific fields to represent the interactions between objects of interest. For instance, in Biology, regulatory networks describe the regulation of genes with transcriptional factors while metabolic networks focus on representing pathways of biochemical reactions. In social sciences, networks are commonly used to represent the interactions between actors.

In this thesis, we consider unsupervised methods which aim at clustering the vertices of a network depending on their connection profiles. There has been a wealth of literature on the topic which goes back to the earlier work of Moreno in 1934. It appears that available techniques can be grouped into three significant categories. First, some models look for community structure, where vertices are partitioned into classes such that vertices of a class are mostly connected to vertices of the same class. They are particularly suitable for the analysis of affiliation networks. Other models look for disassortative mixing in which vertices mostly connect to vertices of different classes. They are commonly used to analyze bipartite networks. Finally, a few procedures look for heterogeneous structure where vertices can have different types of connection profiles. In particular, they can be used to uncover both community structure and disassortative mixing. The starting point of this thesis is the Stochastic Block Model (SBM) which belongs to this later category of approaches.

SBM is a mixture model for graphs which was originally developed in social sciences. It assumes that the vertices of a network are spread into different classes such that the probability of an edge between two vertices only depends on the classes they belong to. No assumption is made on these probabilities of connection such that SBM can take very different topological structures into account. In particular, the model can characterize the presence of hubs which make networks locally dense. Moreover and to some extent, it generalizes many of the existing graph clustering techniques.

The clustering of vertices as well as the estimation of SBM parameters have been subject to previous work and numerous inference strategies have been proposed. However, SBM still suffers from a lack of criteria to estimate the number of components in the mixture. To our knowledge, only one model based criterion has been derived for SBM in the literature. Unfortunately, it tends to be too conservative in the case of small networks.

Besides, almost all graph clustering models, such as SBM, partition the vertices into disjoint clusters. However, recent studies have shown that most existing networks contained overlapping clusters. For instance, many proteins, so-called *moonlighting proteins*, are known to have several functions in the cells, and actors might belong to several groups of inter-

ests. Thus, a graph clustering method should be able to uncover overlapping clusters.

In this thesis, our contributions are the following:

- We propose a new graph clustering algorithm for SBM as well as a new model selection criterion. The R package "mixer" implementing this work is available at `http://cran.r-project.org/web/packages/mixer`.

- We introduce a new random graph model, so-called, Overlapping Stochastic Block Model (OSBM). It allows the vertices of a network to belong to multiple classes and can take very different topological structures into account. A variational algorithm, based on global and local variational techniques, is considered for clustering and estimation purposes.

- We present a new inference procedure for OSBM as well as a model selection criterion which can estimate the number of overlapping clusters in networks. A R package "OSBM" implementing this work will be soon available on the CRAN.

The *first chapter* introduces the main concepts and existing work this thesis builds on. In particular, we review mixture models as well as inference techniques such as the EM algorithm and the variational EM algorithm. We also focus on some model selection criteria to estimate the number of classes from the data. Finally, we describe some of the most widely used graph clustering methods for network analysis and focus mainly on model based approaches.

The *second chapter* illustrates how SBM can be described in a Bayesian framework. A new inference procedure is proposed to approximate the full posterior distribution over the model parameters and latent variables, given the observed data. This approach leads to a new model selection criterion based on a non asymptotic approximation of the marginal likelihood.

The *third chapter* presents OSBM. We show that the model is identifiable within classes of equivalence. Moreover, an algorithm is proposed, based on global and variational techniques. It allows the model parameters to be estimated and the vertices to be classified into overlapping clusters.

Finally, in *chapter 4*, conjugate prior distributions are proposed for the OSBM model parameters. Then, we show how an inference procedure can be used to obtain an approximation of the full posterior distribution over the model parameters and latent variables. This framework leads to a model selection criterion based on new approximation of the marginal likelihood.

During this thesis, two papers and a book chapter were published: Latouche et al. (2009; 2010a;b).

Context

1

Contents

This preliminary chapter introduces the main concepts and existing work this thesis builds on. In Section 1.1, we consider a simple example, *i.e.* the Gaussian mixture model, to illustrate the use of the Expectation Maximization (EM) algorithm for estimation and clustering purposes in mixture models. We also focus on some model selection criteria to estimate the number of classes from the data. We then give in Section 1.2 a variational treatment of EM and describe a more general framework called variational Bayes EM which can lead to an approximation of the full posterior distribution over the model parameters and latent variables. In Section 1.3, we present some of the most widely used graph clustering methods for network analysis and focus mainly on model based approaches. Finally, we bring some insights into the stochastic block model which is considered in Chapter 2 and extended in Chapters 3 and 4 to allow overlapping clusters in networks.

1.1 Mixture models and EM

In this section, we introduce some algorithms which are commonly used to classify the points of a data set. Existing techniques can be distinguished depending on the structure of the classification they produce. Here we consider methods which look for a partition $\mathbf{P}$ of the data only. Approaches looking for hierarchies or fuzzy partitions will be briefly mentioned in the rest of the thesis while others producing overlapping clusters will be described in more details in Chapters 3 and 4.

For a few decades, there has been a wealth of literature on mixture models and the associated Expectation Maximization (EM) algorithm to uncover the classes of a partition $\mathbf{P}$ in a data set. Mixture models first assume that the observations are spread into Q hidden classes such that observation $\mathbf{x}_i$ is drawn from distribution f_q if its belongs to class q. If the posterior distribution over the latent variables takes an analytical form, the EM algorithm can then be applied to estimate the mixture model parameters and classify the observations. In the following, we consider a simple example, *i.e.* the Gaussian mixture model, for illustration purposes. However, we emphasize that EM has a much broader applicability. In particular, it can be used for every distributions f_q of the exponential family (McLachlan and Peel 2000).

We start by introducing the kmeans and kmedoids algorithms. After having described EM, we show how it is related to kmeans in the case of Gaussian mixture models. Finally, we describe some model selection criteria to estimate the number Q of classes from the data in a mixture context.

1.1.1 Kmeans

Let us consider a data set $\mathbf{E} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of N observations. Each vector $\mathbf{x}_i$ is in $\mathbb{R}^d$ and therefore the data set can be represented by a matrix $\mathbf{X}$ in $\mathbb{R}^{N \times d}$ where the ith row equals $\mathbf{x}_i^\intercal$. The goal is to cluster the observations into Q disjoint clusters $\{\mathbf{P}_1, \ldots, \mathbf{P}_Q\}$ of a partition $\mathbf{P}$ such that $\mathbf{P}_q \bigcap \mathbf{P}_l = \emptyset, \forall q \neq l$ and $\bigcup_{q=1}^{Q} \mathbf{P}_q = \mathbf{E}$. We will see in Section 1.1.4 how Q can be estimated from the data but for now, Q is held fixed. A common approach consists in looking for clusters where data points of the same cluster have a smaller inter-point distance than points of different clusters. To formalize this notion, we introduce a d-dimensional vector $\boldsymbol{\mu}_q$, called prototype, for each of the Q clusters. The kmeans algorithm aims at minimizing the sum of squares of the distances between each point and its closest prototype. Formally, this involves minimizing an objective function, called distortion function:

$$J = \sum_{i=1}^{N} \sum_{q=1}^{Q} Z_{iq} || \mathbf{x}_i - \boldsymbol{\mu}_q ||^2, \tag{1.1}$$

where Z_{iq} equals 1 if $\mathbf{x}_i$ is assigned to cluster q and 0 otherwise. In the following, we will denote $\mathbf{Z}_i$ the corresponding Q-dimensional binary vector of cluster assignments. It satisfies $Z_{iq} \in \{0, 1\}, \forall (i, q)$ and $\sum_{q=1}^{Q} Z_{iq} = 1$. From the N binary vectors, we also build a $N \times Q$ binary matrix $\mathbf{Z}$ whose ith row is equal to $\mathbf{Z}_i^\intercal$. In the following, we will show that minimizing

(1.1) is equivalent to looking for a partition $\mathbf{P}$ with the smaller *intra* class inertia denoted I_{intra} and given by:

$$I_{intra} = \sum_{q=1}^{Q} \sum_{i \in \mathbf{P}_q} ||\, \mathbf{x}_i - \bar{\mathbf{x}}_{\mathbf{P}_q}||^2,$$

where $\bar{\mathbf{x}}_{\mathbf{P}_q}$ is the center of mass of $\mathbf{P}_q$:

$$\bar{\mathbf{x}}_{\mathbf{P}_q} = \frac{1}{\text{card}(\mathbf{P}_q)} \sum_{i \in \mathbf{P}_q} \mathbf{x}_i,$$

and $\text{card}(\mathbf{P}_q)$ is the cardinality of $\mathbf{P}_q$. It is known that the total inertia I_{total} can be decomposed into the sum of the *intra* class inertia I_{intra} and the *inter* class inertia I_{inter}:

$$I_{total} = I_{intra} + I_{inter},$$

where

$$I_{inter} = \sum_{q=1}^{Q} \text{card}(\mathbf{P}_q)||\bar{\mathbf{x}}_{\mathbf{P}_q} - \bar{\mathbf{x}}||^2,$$

and $\bar{\mathbf{x}}$ is the center of mass of the entire data set. Since I_{total} does not depend on the partition $\mathbf{P}$, minimizing I_{intra} is equivalent to maximizing I_{inter}. Thus, kmeans looks for clusters as separated as possible.

The algorithm starts with a set of initial prototypes usually sampled from a Gaussian distribution. It then optimizes (1.1) with respect to $\mathbf{Z}$ and the set $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_Q\}$ in turn. First, the $\boldsymbol{\mu}_q$s are kept fixed while J is minimized with respect to $\mathbf{Z}$. Second, J is minimized with respect to the $\boldsymbol{\mu}_q$s keeping $\mathbf{Z}$ fixed. This two stage optimization procedure is repeated until the absolute distance between two successive values of J is smaller than a threshold *eps*. In Section 1.1.3, we will see how these two steps are related to the E and M steps of the Expectation Maximization (EM) algorithm in the case of Gaussian mixture models.

Since J is a linear function of the Z_{iq}s, the optimization of J with respect to $\mathbf{Z}$ leads to a closed form solution. Indeed, the terms indexed by i are all independent, and therefore, optimizing J involves optimizing each of the objective functions:

$$J_i = \sum_{q=1}^{Q} Z_{iq}||\, \mathbf{x}_i - \boldsymbol{\mu}_q\, ||^2, \forall i \in \{1, \ldots, Q\}.$$

The function J_i is minimized when Z_{iq} equals 1 for whichever value of q gives the minimum value of $||\, \mathbf{x}_i - \boldsymbol{\mu}_q\, ||^2$. In other words, Z_{iq} equals 1 if $q = \text{argmin}_l||\, \mathbf{x}_i - \boldsymbol{\mu}_l\, ||^2$ and 0 otherwise.

Finally, in order to optimize J with respect $\boldsymbol{\mu}_q$, we set the gradient of (1.1) to zero:

$$\bigtriangledown_{\boldsymbol{\mu}_q} J = 2 \sum_{i=1}^{N} Z_{iq}(\mathbf{x}_i - \boldsymbol{\mu}_q) = 0.$$

This implies:

$$\boldsymbol{\mu}_q = \frac{\sum_{i=1}^{N} Z_{iq}\, \mathbf{x}_i}{\sum_{i=1}^{N} Z_{iq}}. \tag{1.2}$$

Thus, the prototype $\boldsymbol{\mu}_q$ is defined as the mean of all the points $\mathbf{x}_i$ assigned to cluster q.

Kmeans (Algorithm 1) was studied by MacQueen (1967). Because each step is guaranteed to reduce the objective function, convergence of the algorithm is assured. However, it may converge to local rather than global minimum.

Algorithm 1: The kmeans algorithm.

```
// INITIALIZATION
  Initialize μ_q^0, ∀q
  μ_q^new ← μ_q^0, ∀q
// OPTIMIZATION
  repeat
    μ_q^old ← μ_q^new, ∀q
    // Step 1
    for i ∈ 1 : N do
      Find q = argmin_l || x_i − μ_l^old ||^2
      Z_iq ← 1
      Z_il ← 0, ∀l ≠ q
    end
    // Step 2
    μ_q^new ← (Σ_{i=1}^N Z_iq x_i) / (Σ_{i=1}^N Z_iq), ∀q
  until J converges
```

Numerous methods have been proposed to speed up the kmeans algorithm. Some of them precompute a tree such that nearby points are in the same sub-tree (Rmasubramanian and Paliwal 1990). Others limit the number of distances $|| \mathbf{x}_i - \boldsymbol{\mu}_q ||^2$ to compute using the triangle inequality (Hodgson 1998, Moore 2000, Elkan 2003).

Because the kmeans algorithm is based on the Euclidean distance, it is very sensitive to outliers. Moreover, depending on the type of data considered, the Euclidean distance might not be an appropriate choice as a measure of dissimilarity between data points and prototypes. Therefore, it can be generalized by introducing any dissimilarity measure $ds(\cdot, \cdot)$. The goal is now to optimize the objective function J^* rather than J, where:

$$J^* = \sum_{i=1}^{N} \sum_{q=1}^{Q} Z_{iq} ds(\mathbf{x}_i, \boldsymbol{\mu}_q).$$

This gives rise to the kmedoids algorithm. As kmeans, the kmedoids algorithm relies on a two stage optimization procedure. First, each data point $\mathbf{x}_i$ is assigned to the prototype $\boldsymbol{\mu}_q$ for which the corresponding dissimilarity measure $ds(\mathbf{x}_i, \boldsymbol{\mu}_q)$ is minimized. Second, each cluster prototype is set equal to one of the data points assigned to that cluster.

A drawback of the kmeans and the kmedoids algorithms is that, at each iteration, each data point is assigned to one and only one cluster. In practice, while some points are much closer to one of the mean vectors $\boldsymbol{\mu}_q$, some others often lie midway between cluster centers. In this

case, the hard assignment of points to the nearest cluster may not be the most appropriate choice. In Section 1.1.3, we will see how a probabilistic framework can lead to soft assignments reflecting the uncertainty to which cluster each data point belongs.

1.1.2 Gaussian mixture models

We denote by $\boldsymbol{\kappa}$ a set of Q mean vectors $\boldsymbol{\mu}_q$ and covariance matrices $\boldsymbol{\Sigma}_q$. Moreover, let us consider a vector $\boldsymbol{\alpha}$ of class proportions which satisfies $\alpha_q > 0, \forall q$ and $\sum_{q=1}^{Q} \alpha_q = 1$. A Gaussian mixture distribution is then given by:

$$p(\mathbf{x}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) = \sum_{q=1}^{Q} \alpha_q \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \tag{1.3}$$

where

$$\mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = \frac{1}{(2\pi)^{d/2} |\, \boldsymbol{\Sigma}_q \,|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_q)^\intercal \, \boldsymbol{\Sigma}_q^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_q) \right).$$

As shown in the following, a mixture distribution can be seen as the result of a marginalization over a latent variable.

First, let us assume that a binary vector $\mathbf{Z}_i$ is sampled from a multinomial distribution:

$$\begin{aligned} p(\mathbf{Z}_i \mid \boldsymbol{\alpha}) &= \mathcal{M}\left(\mathbf{Z}_i;\, 1, \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_Q)\right) \\ &= \prod_{q=1}^{Q} \alpha_q^{Z_{iq}}. \end{aligned} \tag{1.4}$$

As in the previous section, $\mathbf{Z}_i$ sees all its components set to zero except one such that $Z_{iq} = 1$ if observation i belongs to class q. The vector $\mathbf{x}_i$ is then sampled from a Gaussian distribution:

$$p(\mathbf{x}_i \mid Z_{iq} = 1, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) = \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \tag{1.5}$$

with mean vector $\boldsymbol{\mu}_q$ and covariance matrix $\boldsymbol{\Sigma}_q$. The full conditional distribution is given by:

$$\begin{aligned} p(\mathbf{x}_i \mid \mathbf{Z}_i, \boldsymbol{\kappa}) &= \prod_{q=1}^{Q} p(\mathbf{x}_i \mid Z_{iq} = 1, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)^{Z_{iq}} \\ &= \prod_{q=1}^{Q} \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)^{Z_{iq}}. \end{aligned} \tag{1.6}$$

Therefore, the marginalization of $p(\mathbf{x}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa})$ over all possible vectors $\mathbf{Z}_i$ leads to:

$$\begin{aligned} p(\mathbf{x}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) &= \sum_{\mathbf{Z}_i} p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \\ &= \sum_{\mathbf{Z}_i} p(\mathbf{x}_i \mid \mathbf{Z}_i, \boldsymbol{\kappa}) p(\mathbf{Z}_i \mid \boldsymbol{\alpha}) \\ &= \sum_{\mathbf{Z}_i} \prod_{q=1}^{Q} \left(\alpha_q \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \right)^{Z_{iq}}. \end{aligned} \tag{1.7}$$

Note that (1.7) is equivalent to (1.3). However, we now have an explicit formulation of the Gaussian mixture model in terms of the latent vector $\mathbf{Z}_i$ which will play a key role in Section 1.1.3.

Maximum likelihood

We are given a data set $\mathbf{X}$ of N observations which are assumed to be independent and drawn from a Gaussian mixture distribution with Q classes. In a frequentist setting, the goal is to maximize the observed-data log-likelihood $\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa})$ with respect to $\boldsymbol{\kappa}$ and $\boldsymbol{\alpha}$:

$$\begin{aligned} \log p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) &= \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \\ &= \sum_{i=1}^{N} \log \left(\sum_{q=1}^{Q} \alpha_q \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \right). \end{aligned} \tag{1.8}$$

One approach consists in relying directly on gradient based algorithms for the optimization procedure (Fletcher 1987, Nocedal and Wright 1999). Although this approach is feasible, it is rarely used in practice and the EM algorithm described in Section 1.1.3 is usually preferred.

Bayesian approach

In the Bayesian framework, some distributions, called priors, are introduced to characterize *a priori* information about the model parameters. For instance, in the regression context, a Gaussian prior distribution over the weight vector $\boldsymbol{\beta}$ is often used for regularization in order to avoid overfitting (Wahba 1975, Berger 1985, Gull 1989, MacKay 1992, Bernardo and Smith 1994, Gelman et al. 2004). While a Gaussian prior leads to a L_2-norm regularization (ridge regression) (Hoerl and Kennard 1970, Hastie et al. 2001), a Laplace prior gives rise to a L_1-norm regularization (Lasso) (Tibshirani 1996, Park and Casella 2008).

In the case of Gaussian mixture model, the Bayesian framework can be used to deal with the singularities of the observed-data log-likelihood (1.8). For simplicity, consider that the covariance matrices of the different components are such that $\boldsymbol{\Sigma}_q = \sigma_q^2 \mathbf{I}$ where $\mathbf{I}$ denotes the identity matrix. If one component has its mean vector equal to one of the points $\mathbf{x}_i$ in the data set, *i.e.* $\boldsymbol{\mu}_q = \mathbf{x}_i$, then:

$$\begin{aligned} \mathcal{N}(\mathbf{x}_i;\, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) &= \mathcal{N}(\mathbf{x}_i;\, \mathbf{x}_i, \boldsymbol{\Sigma}_q) \\ &= \frac{1}{(2\pi)^{d/2} \sigma_q^d}. \end{aligned} \tag{1.9}$$

If we now consider the limit $\sigma_q \to 0$, then (1.9) goes to infinity and so does (1.8). Therefore, optimizing the observed-data log-likelihood is not a well defined problem since singularities can arise whenever a component of the Gaussian mixture model collapses to a single point. To tackle this issue, a common strategy consists in introducing conjugate priors[1] for the model parameters. Thus, a Dirichlet distribution is chosen for the vector $\boldsymbol{\alpha}$:

$$p(\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\alpha};\, \mathbf{n}^0 = (n_1^0, \ldots, n_Q^0)),$$

[1] In a Bayesian framework, given a data set $\mathbf{X}$ as well as a model parameter $\boldsymbol{\theta}$, if a posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{X})$ is in the same family of distributions as the prior $p(\boldsymbol{\theta})$, then the prior and posterior are said to be conjugate

where $n_q^0 = 1/2, \forall q$. This Dirichlet distribution corresponds to a non-informative Jeffreys prior distribution which is known to be proper (Jeffreys 1946). It is also possible to consider a uniform distribution on the $Q-1$ dimensional simplex by fixing $n_q^0 = 1, \forall q$. As for $\boldsymbol{\kappa}$, independent Gaussian-Wishart priors can be used:

$$p(\boldsymbol{\kappa}) = \prod_{q=1}^{Q} \mathcal{N}(\boldsymbol{\mu}_q;\, \mathbf{m}_0, \beta_0\, \boldsymbol{\Delta}_q^{-1}) \mathcal{W}(\boldsymbol{\Delta}_q;\, \mathbf{W}_0, \lambda_0),$$

where $\boldsymbol{\Delta}_q$ denotes the precision matrix of the qth component, that is the inverse matrix of $\boldsymbol{\Sigma}_q$. Considering a quadratic loss function (see Dang 1998), the goal is now to maximize the log-posterior distribution of the model parameters, given the observed data set $\mathbf{X}$. Using Bayes rule, we have:

$$p(\boldsymbol{\alpha}, \boldsymbol{\kappa} \,|\, \mathbf{X}) \propto p(\mathbf{X} \,|\, \boldsymbol{\alpha}, \boldsymbol{\kappa}) p(\boldsymbol{\alpha}) p(\boldsymbol{\kappa}).$$

Therefore:

$$\log p(\boldsymbol{\alpha}, \boldsymbol{\kappa} \,|\, \mathbf{X}) = \log p(\mathbf{X} \,|\, \boldsymbol{\alpha}, \boldsymbol{\kappa}) + \log p(\boldsymbol{\alpha}) + \log p(\boldsymbol{\kappa}) + \text{const.} \qquad (1.10)$$

Note that the first term on the right hand side of (1.10) is the observed-data log-likelihood. As in the frequentist setting, it is possible to rely directly on gradient based algorithms for the optimization procedure. However, as mentioned above, the standard approach in the machine learning and statistical community, to obtain point estimates of mixture model parameters, is the EM algorithm (see Section 1.1.3). The estimates are called *Maximum A Posteriori* (MAP) estimates because they maximize the posterior distribution rather than the observed-data likelihood. As shown in Bishop (2006), the singularities are absent in this Bayesian framework, simply by introducing the priors and relying on the MAP estimates.

In this section, we made a first step towards a full Bayesian treatment of the data and motivated the use of prior distributions to deal with singularities. However, the Bayesian framework brings much more powerful features as illustrated in Chapters 2 and 4. For instance, we will see how an estimation of the full posterior distribution over the model parameters can be performed using variational techniques. This will naturally lead to new model selection criteria to estimate the number Q of classes in networks.

1.1.3 The EM algorithm

The Expectation Maximization (EM) algorithm (Dempster et al. 1977, McLachlan and Krishnan 1997) was originally developed in order to find maximum likelihood solutions for models with missing data, although it can also be applied to find MAP estimates. The two corresponding functions to be maximized will be denoted $\mathcal{Q}_{ML}(\cdot,\cdot)$ and $\mathcal{Q}_{MAP}(\cdot,\cdot)$ respectively. EM has a broad applicability and will be first described in a general setting. We shall then go back to our example in this chapter, *i.e.* the Gaussian mixture model. Note that EM can be used in the case of continuous latent variables but in the following, we concentrate on discrete mixture models only.

General EM

Let us consider a data set $\mathbf{X}$ of N observations and $\mathbf{Z}$ the corresponding matrix of class assignments introduced in Section 1.1.1 . The ith row of $\mathbf{X}$ and $\mathbf{Z}$ represent the vectors $\mathbf{x}_i^\intercal$ and $\mathbf{Z}_i^\intercal$ respectively. Moreover, we assume that the observations are drawn from a mixture distribution with parameter $\boldsymbol{\theta}$, *i.e.* $\boldsymbol{\theta} = (\boldsymbol{\kappa}, \boldsymbol{\alpha})$ in the Gaussian mixture model. If the observations are independent, then:

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \boldsymbol{\theta}).$$

Following (1.7), the marginalization of each distribution $p(\mathbf{x}_i \mid \boldsymbol{\theta})$ over all possible vectors $\mathbf{Z}_i$ leads to:

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log \left(\sum_{\mathbf{Z}_i} p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\theta}) \right).$$

A more general expression, which stands also in the non independent case, can be obtained by directly marginalizing over all possible matrices $\mathbf{Z}$:

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \log \left(\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \right), \tag{1.11}$$

where $p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$ is called complete-data likelihood. In practice, given a data set $\mathbf{X}$, the matrix $\mathbf{Z}$ is unknown and has to be inferred. The state of knowledge of $\mathbf{Z}$ is given only by the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$. Therefore, in order to estimate $\boldsymbol{\theta}$, the EM algorithm considers the maximization of the expected value of the complete-data log-likelihood, under the posterior distribution.

If we denote $\boldsymbol{\theta}^{old}$ the current value of the model parameters, then during the E step, the algorithm computes $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})$ and the expectation $\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$:

$$\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}). \tag{1.12}$$

During the M step, (1.12) is then maximized with respect to $\boldsymbol{\theta}$. This leads to a new estimate $\boldsymbol{\theta}^{new}$ of the model parameters. Thus, the algorithm starts with an initial value $\boldsymbol{\theta}_0$. The E and M steps are then repeated until the absolute difference between two successive values of $\boldsymbol{\theta}$ are smaller than a threshold *eps*. For now, the use of the expectation may seem arbitrary, but we will show in Section 1.2.1 how EM (Algorithm 2) is guaranteed to maximize the observed-data log-likelihood (1.11).

As mentioned in Section 1.1.2, the EM algorithm can also be used to find MAP estimates when the goal is to maximize $\log p(\boldsymbol{\theta} \mid \mathbf{X})$:

$$\log p(\boldsymbol{\theta} \mid \mathbf{X}) = \log \left(\sum_{\mathbf{Z}} p(\boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{X}) \right).$$

Algorithm 2: General EM algorithm for mixture models. $\mathcal{Q}(\cdot,\cdot)$ either denotes $\mathcal{Q}_{ML}(\cdot,\cdot)$ or $\mathcal{Q}_{MAP}(\cdot,\cdot)$.

```
// INITIALIZATION
  Initialize θ_0
  θ^new ← θ_0
// OPTIMIZATION
  repeat
    θ^old ← θ^new
    // E-step
    Compute p(Z | X, θ^old)
    Compute Q(θ, θ^old)
    // M-step
    Find θ^new = argmax_θ Q(θ, θ^old)
  until θ converges
```

During the E step, the algorithm computes $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})$ and the expectation $\mathcal{Q}_{MAP}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$:

$$\begin{aligned}
\mathcal{Q}_{MAP}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{X}) \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{X})} \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) + \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\boldsymbol{\theta}) \\
&\quad - \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{X}) \\
&= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) + \text{const.}
\end{aligned} \tag{1.13}$$

Again, $\mathcal{Q}_{MAP}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ is maximized with respect to $\boldsymbol{\theta}$ during the M step. Note that the first term on the right hand side of (1.13) corresponds exactly to (1.12).

As pointed out by Bishop (2006), among many others, the EM algorithm can converge to local rather than global maxima.

EM for Gaussian mixtures

In order to perform maximum likelihood inference in the case of Gaussian mixture models, we need an expression for the complete-data log-likelihood. The observations are assumed to be independent and therefore:

$$\begin{aligned}
\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) &= \log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\kappa}) + \log p(\mathbf{Z} \mid \boldsymbol{\alpha}) \\
&= \sum_{i=1}^{N} \left(\log p(\mathbf{x}_i \mid \mathbf{Z}_i, \boldsymbol{\kappa}) + \log p(\mathbf{Z}_i \mid \boldsymbol{\alpha}) \right).
\end{aligned}$$

Using (1.4) and (1.6) leads to:

$$\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) = \sum_{i=1}^{N} \sum_{q=1}^{Q} Z_{iq} \left(\log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) + \log \alpha_q \right).$$

Moreover, from Bayes rule and (1.3), the posterior distribution over all the latent variables takes a factorized form:

$$\begin{aligned}
p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\kappa}) &= \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa})}{p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa})} \\
&= \frac{\prod_{i=1}^{N} p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa})}{\prod_{i=1}^{N} p(\mathbf{x}_i \mid \boldsymbol{\alpha}, \boldsymbol{\kappa})} \\
&= \frac{\prod_{i=1}^{N} \prod_{q=1}^{Q} \left(\alpha_q \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \right)^{Z_{iq}}}{\prod_{i=1}^{N} \left(\sum_{l=1}^{Q} \alpha_l \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \right)} \\
&= \frac{\prod_{i=1}^{N} \prod_{q=1}^{Q} \left(\alpha_q \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \right)^{Z_{iq}}}{\prod_{i=1}^{N} \prod_{q=1}^{Q} \left(\sum_{l=1}^{Q} \alpha_l \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \right)^{Z_{iq}}} \\
&= \prod_{i=1}^{N} \prod_{q=1}^{Q} \left(\frac{\alpha_q \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}{\sum_{l=1}^{Q} \alpha_l \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \right)^{Z_{iq}} \\
&= \prod_{i=1}^{N} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{iQ})\right),
\end{aligned}$$

where

$$\tau_{iq} = \frac{\alpha_q \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}{\sum_{l=1}^{Q} \alpha_l \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \tag{1.14}$$

Each variable τ_{iq} denotes the posterior probability, sometimes called responsibility, of observation i to belong to class q. Note that for each observation, both the prior (1.4) and the posterior are multinomial distributions. During the M step, $\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ is then maximized with respect to all the model parameters:

$$\begin{aligned}
\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) &= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}^{old}, \boldsymbol{\kappa}^{old}) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\kappa}) \\
&= \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \left(\log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) + \log \alpha_q \right).
\end{aligned} \tag{1.15}$$

Setting the gradient of (1.15), with respect to $\boldsymbol{\mu}_q$, to zero, leads to:

$$\bigtriangledown_{\boldsymbol{\mu}_q} \mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{i=1}^{N} \tau_{iq} (\mathbf{x}_i - \boldsymbol{\mu}_q) = 0. \tag{1.16}$$

Therefore

$$\boldsymbol{\mu}_q = \frac{\sum_{i=1}^{N} \tau_{iq} \mathbf{x}_i}{\sum_{i=1}^{N} \tau_{iq}}. \tag{1.17}$$

Equation (1.17) must be compared with (1.2). Indeed, contrary to the kmeans algorithm, $\boldsymbol{\mu}_q$ is now a weighted mean of all the points in the data

set, the weighting factors being the posterior probabilities of the points to belong to class q. The gradient with respect to $\mathbf{\Sigma}_q$ also takes a simple form:

$$\nabla_{\mathbf{\Sigma}_q} \mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{i=1}^{N} \tau_{iq} \left(\frac{\mathbf{\Sigma}_q}{2} - \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_q)(\mathbf{x}_i - \boldsymbol{\mu}_q)^\intercal \right) = 0.$$

Thus

$$\mathbf{\Sigma}_q = \frac{\sum_{i=1}^{Q} \tau_{iq}(\mathbf{x}_i - \boldsymbol{\mu}_q)(\mathbf{x}_i - \boldsymbol{\mu}_q)^\intercal}{\sum_{i=1}^{N} \tau_{iq}}.$$

Again, each data point is weighted by τ_{iq}, the posterior probability that class q was responsible for generating $\mathbf{x}_i$. Finally, caution must be taken when maximizing $\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ with respect to α_q. Indeed, the optimization is subject to the constraint $\sum_{q=1}^{Q} \alpha_q = 1$. This is achieved using a Lagrange multiplier and maximizing:

$$\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) + \lambda(\sum_{q=1}^{Q} \alpha_q - 1). \tag{1.18}$$

Setting the gradient of (1.18), with respect to α_q, to zero, gives:

$$\frac{\sum_{i=1}^{N} \tau_{iq}}{\alpha_q} + \lambda = 0. \tag{1.19}$$

Then, after multiplying (1.19) by α_q as well as summing over q, we find $\lambda = -N$ and therefore:

$$\alpha_q = \frac{\sum_{i=1}^{N} \tau_{iq}}{N}.$$

The algorithm is initialized for instance using a few iterations of the kmeans algorithm (see Section 1.1.1), the E and M step are then repeated until convergence. As mentioned already in a general setting, EM (Algorithm 3) is not guaranteed to converge to a global maximum.

Relation to kmeans

As mentioned in Section 1.1.1, the two stages of the kmeans algorithm are related to the E and M steps of the EM algorithm in the case of Gaussian mixture models. While kmeans relies on hard assignments, we showed in Section 1.1.3 that EM performs soft assignments of points in a data set. In fact, the kmeans algorithm can be seen as a particular limit of EM for Gaussian mixtures (Bishop 2006). Thus, consider that each component of the mixture has a covariance matrix given by $\mathbf{\Sigma}_q = \sigma^2 \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix:

$$\begin{aligned} p(\mathbf{x}_i \,|\, \boldsymbol{\mu}_q, \sigma^2) &= \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q, \sigma^2) \\ &= \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2} || \mathbf{x}_i - \boldsymbol{\mu}_q ||^2 \right). \end{aligned}$$

The variance parameter σ^2 is shared by all the components and held fixed for now. If the EM algorithm is applied for maximum likelihood estimation of this Gaussian mixture model, then the posterior probabilities τ_{iq}s

Algorithm 3: The EM algorithm for maximum likelihood estimation in Gaussian mixture models.

// INITIALIZATION

Initialize $\alpha_q^0, \boldsymbol{\mu}_q^0, \boldsymbol{\Sigma}_q^0, \forall q$

$\alpha_q^{new} \leftarrow \alpha_q^0, \forall q$

$\boldsymbol{\mu}_q^{new} \leftarrow \boldsymbol{\mu}_q^0, \forall q$

$\boldsymbol{\Sigma}_q^{new} \leftarrow \boldsymbol{\Sigma}_q^0, \forall q$

// OPTIMIZATION

repeat

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

$\boldsymbol{\mu}_q^{old} \leftarrow \boldsymbol{\mu}_q^{new}, \forall q$

$\boldsymbol{\Sigma}_q^{old} \leftarrow \boldsymbol{\Sigma}_q^{new}, \forall q$

// E-step

$\tau_{iq} \leftarrow \frac{\alpha_q^{old} \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_q^{old}, \boldsymbol{\Sigma}_q^{old})}{\sum_{l=1}^{Q} \alpha_l^{old} \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_l^{old}, \boldsymbol{\Sigma}_l^{old})}, \forall i, q$

// M-step

$\alpha_q^{new} \leftarrow \frac{\sum_{i=1}^{N} \tau_{iq}}{N}, \forall q$

$\boldsymbol{\mu}_q^{new} \leftarrow \frac{\sum_{i=1}^{N} \tau_{iq} \mathbf{x}_i}{\sum_{i=1}^{N} \tau_{iq}}, \forall q$

$\boldsymbol{\Sigma}_q^{new} \leftarrow \frac{\sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu}_q^{new})(\mathbf{x}_i - \boldsymbol{\mu}_q^{new})^\intercal}{\sum_{i=1}^{N} \tau_{iq}}, \forall q$

until $(\alpha_q, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q), \forall q$ *converges*

given by (1.14) become:

$$\tau_{iq} = \frac{\alpha_q \exp\left(-\frac{1}{2\sigma^2} || \mathbf{x}_i - \boldsymbol{\mu}_q ||^2\right)}{\sum_{l=1}^{Q} \alpha_l \exp\left(-\frac{1}{2\sigma^2} || \mathbf{x}_i - \boldsymbol{\mu}_l ||^2\right)}. \tag{1.20}$$

If we now consider the limit $\sigma^2 \to 0$ in (1.20), the smallest term $|| \mathbf{x}_i - \boldsymbol{\mu}_l ||^2$ in the denominator will go to zero most slowly. Therefore, the probabilities τ_{iq} for observation $\mathbf{x}_i$ will all go to zero except for the corresponding τ_{il} which will go to unity. Thus, when the variance parameter of the components tends to zero, the EM algorithm leads to a hard assignment of the data as in kmeans, and each observation is assigned to the class having the nearest mean vector. The expectation $\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ in (1.15) then becomes:

$$\lim_{\sigma^2 \to 0} \mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \lim_{\sigma^2 \to 0} -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} || \mathbf{x}_i - \boldsymbol{\mu}_q ||^2 + \text{const.} \tag{1.21}$$

Since the observations are hard assigned to the classes, we can denote $\tau_{iq} = Z_{iq}$ as in kmeans and therefore maximizing (1.21) is equivalent to minimizing the kmeans objective function (1.1).

1.1.4 Model selection

For a fixed number Q of classes, we have seen how the EM algorithm could be used for both the estimation of mixture model parameters and

the classification of observations in a data set. We now describe some of the existing model selection criteria which aim at estimating Q from the data. For an extensive discussion on model selection in mixture models, we refer to McLachlan and Peel (2000). In this section, $\hat{\boldsymbol{\theta}}$ either denotes the maximum likelihood ($\boldsymbol{\theta}_{ML}$) or the MAP estimate ($\boldsymbol{\theta}_{MAP}$) as described in Section 1.1.3.

Given a set of values of Q, the goal is to select Q^* such that a given criterion is maximized. Because all the criteria described in this section rely on $\hat{\boldsymbol{\theta}}$, the mixture model parameters have to be estimated for each value of Q in the set.

Akaike's information criterion

Let us consider an observed data set $\mathbf{X}$ and a fixed number Q of classes. We also denote by $\mathbf{Y}$ any data set of the same size of $\mathbf{X}$. If the observations are assumed to be drawn from a mixture model with parameter $\boldsymbol{\theta}$, we have seen that the EM algorithm can be used to maximize $\log p(\mathbf{X} \mid \boldsymbol{\theta})$ or $\log p(\boldsymbol{\theta} \mid \mathbf{X})$. This leads to an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$. In practice, the value of Q considered might not be the most appropriate to model the data. Moreover, the EM algorithm can converge to local rather than global maximum. In any case, the distribution $p(\mathbf{Y} \mid \hat{\boldsymbol{\theta}})$ can only be seen as an approximation of the true distribution $p(\mathbf{Y})$ which generated $\mathbf{X}$.

Model selection can then be approached in terms of the Kullback-Leibler (KL) divergence between $p(\mathbf{Y})$ and $p(\mathbf{Y} \mid \hat{\boldsymbol{\theta}})$:

$$\begin{aligned} \mathrm{KL}\left(p(\cdot)||p(\,\cdot\,|\hat{\boldsymbol{\theta}}(\mathbf{X}))\right) &= -\int p(\mathbf{Y}) \log \left\{ \frac{p\left(\mathbf{Y} \mid \hat{\boldsymbol{\theta}}(\mathbf{X})\right)}{p(\mathbf{Y})} \right\} d\,\mathbf{Y} \\ &= \int p(\mathbf{Y}) \log p(\mathbf{Y}) d\,\mathbf{Y} - \int p(\mathbf{Y}) \log p\left(\mathbf{Y} \mid \hat{\boldsymbol{\theta}}(\mathbf{X})\right) d\,\mathbf{Y}, \end{aligned} \tag{1.22}$$

where we have denoted $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\mathbf{X})$ to emphasize that $\hat{\boldsymbol{\theta}}$ is estimated from $\mathbf{X}$. In information theory, (1.22) is seen as the information which is lost by approximating the true model $p(\mathbf{Y})$ with a fitted model $p(\mathbf{Y} \mid \hat{\boldsymbol{\theta}})$. The goal is then to select the model for which the corresponding information loss is minimum. As the first term on the right hand side of (1.22) does not depend on the fitted model, only the second term is relevant. It can be expressed as:

$$\begin{aligned} \eta(\mathbf{X}) &= \int p(\mathbf{Y}) \log p\left(\mathbf{Y} \mid \hat{\boldsymbol{\theta}}(\mathbf{X})\right) d\,\mathbf{Y} \\ &= \mathrm{E}_{\mathbf{Y}}[\log p\left(\mathbf{Y} \mid \hat{\boldsymbol{\theta}}(\mathbf{X})\right)], \end{aligned} \tag{1.23}$$

where the expectation is taken according to $p(\mathbf{Y})$. Unfortunately, because $p(\mathbf{Y})$ is unknown, (1.23) cannot be computed analytically. However, if we take the expectation of $\eta(\mathbf{X})$ over every possible data set $\mathbf{X}$, we obtain:

$$\mathrm{E}_{\mathbf{X}}[\eta(\mathbf{X})] = \mathrm{E}_{\mathbf{X},\mathbf{Y}}[\log p\left(\mathbf{Y} \mid \hat{\boldsymbol{\theta}}(\mathbf{X})\right)], \tag{1.24}$$

which can be estimated (McLachlan and Peel 2000). In practice, only a single data set $\mathbf{X}$ is given and Akaike (1973; 1974) showed that (1.24) is asymptotically equal to:

$$\log p(\mathbf{X} \mid \hat{\boldsymbol{\theta}}) - K, \tag{1.25}$$

where K is the total number of parameters in the model. The approximation (1.25) corresponds to the Akaike's Information Criterion (AIC).

The AIC criterion has been widely used to assess the order of a mixture model (Bozdogan and Sclove 1984, Sclove 1987). However, many authors observed that AIC is order inconsistent and tends to overfit models (Koehler and Murphee 1988, Soromenho 1993, Celeux and Soromenho 1996). In other words, AIC tends to overestimate the number of components in the mixture context.

Bayesian information criterion

The Bayesian Information Criterion (BIC) relies on an asymptotic approximation of the marginal log-likelihood, also called integrated observed-data log-likelihood, given by:

$$\begin{aligned} \log p(\mathbf{X}) &= \log \left\{ \int p(\mathbf{X}, \boldsymbol{\theta}) d\,\boldsymbol{\theta} \right\} \\ &= \log \left\{ \int p(\mathbf{X} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\,\boldsymbol{\theta} \right\}, \end{aligned} \tag{1.26}$$

where $p(\boldsymbol{\theta})$ is a prior distribution over the mixture model parameters. Since (1.26) involves integrating over all possible values of $\boldsymbol{\theta}$, it is generally not tractable. To approximate the integral, the integrand is expanded using a second order Taylor series about the point $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$:

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

where $\mathbf{H}$ is the negative hessian matrix of $\log p(\mathbf{X}, \boldsymbol{\theta})$ at $\hat{\boldsymbol{\theta}}$. Note that this approximation is relevant if the integrand is highly concentrated around $\hat{\boldsymbol{\theta}}$. If we set $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{MAP}$, we have:

$$\begin{aligned} \bigtriangledown_{\boldsymbol{\theta}=\boldsymbol{\theta}_{MAP}} \log p(\mathbf{X}, \boldsymbol{\theta}) &= \bigtriangledown_{\boldsymbol{\theta}=\boldsymbol{\theta}_{MAP}} \log \left(p(\boldsymbol{\theta} \mid \mathbf{X}) p(\mathbf{X}) \right) \\ &= \bigtriangledown_{\boldsymbol{\theta}=\boldsymbol{\theta}_{MAP}} \log p(\boldsymbol{\theta} \mid \mathbf{X}) + \bigtriangledown_{\boldsymbol{\theta}=\boldsymbol{\theta}_{MAP}} \log p(\mathbf{X}) \\ &= \bigtriangledown_{\boldsymbol{\theta}=\boldsymbol{\theta}_{MAP}} \log p(\boldsymbol{\theta} \mid \mathbf{X}) \\ &= 0, \end{aligned}$$

since $\boldsymbol{\theta}_{MAP}$ maximizes $\log p(\boldsymbol{\theta} \mid \mathbf{X})$ and $\log p(\mathbf{X})$ does not depend on $\boldsymbol{\theta}$. Therefore:

$$\begin{aligned} \log p(\mathbf{X}, \boldsymbol{\theta}) &\approx \log p(\mathbf{X}, \hat{\boldsymbol{\theta}}) - \frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\intercal} \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \\ &\approx \log p(\mathbf{X} \mid \hat{\boldsymbol{\theta}}) + \log p(\hat{\boldsymbol{\theta}}) - \frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\intercal} \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}). \end{aligned} \tag{1.27}$$

Using (1.27) in (1.26) leads to:

$$\begin{aligned} \log p(\mathbf{X}) &\approx \log \left\{ \int p(\mathbf{X} \mid \hat{\boldsymbol{\theta}}) p(\hat{\boldsymbol{\theta}}) \exp \left(-\frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\intercal} \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \right) d\,\boldsymbol{\theta} \right\} \\ &\approx \log p(\mathbf{X} \mid \hat{\boldsymbol{\theta}}) + \log p(\hat{\boldsymbol{\theta}}) + \log \left\{ \int \underbrace{\exp \left(-\frac{1}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\intercal} \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \right)}_{\text{Gaussian functional form}} d\,\boldsymbol{\theta} \right\}. \end{aligned} \tag{1.28}$$

The functional form of the integrand in (1.28) corresponds to a Gaussian distribution with mean vector $\hat{\boldsymbol{\theta}}$ and covariance matrix $\mathbf{H}^{-1}$. Thus, the integral takes a simple form:

$$\int \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^{\intercal}\mathbf{H}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})\right) d\,\boldsymbol{\theta} = (2\pi)^{d/2}|\mathbf{H}|^{-\frac{1}{2}},$$

and

$$\log p(\mathbf{X}) \approx \log p(\mathbf{X}\,|\hat{\boldsymbol{\theta}}) + \underbrace{\log p(\hat{\boldsymbol{\theta}}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{H}|}_{\text{Occam factor}}. \tag{1.29}$$

This approximation is known as Laplace's method for integral. The second part of the right hand side of (1.29) penalizes the model complexity and is sometimes called Occam factor. As shown in Kass and Raftery (1995) and Raftery (1995), for large samples, $\boldsymbol{\theta}_{ML} \approx \boldsymbol{\theta}_{MAP}$ and $\mathbf{H} \approx \mathbf{J}$, where $\mathbf{J}$ is the expected Fisher information matrix of the observed data. This leads to:

$$\log p(\mathbf{X}) \approx \log p(\mathbf{X}\,|\hat{\boldsymbol{\theta}}) + \log p(\hat{\boldsymbol{\theta}}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{J}|. \tag{1.30}$$

This strong approximation assumes that the prior is very flat such that its effect can be ignored (Ripley 1996). Finally, the BIC criterion, sometimes called Schwarz criterion (Schwarz 1978), is obtained by ignoring the terms in $O(1)$ in (1.30) and noting that:

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

to give

$$\log p(\mathbf{X}) \approx \log p(\mathbf{X}\,|\hat{\boldsymbol{\theta}}) - \frac{K}{2}\log N.$$

Leroux (1992) showed that BIC does not underestimate the true number of classes, asymptotically. Moreover, simulation studies as well as experiments on real data have been carried out to assess the performances of BIC and they have reported encouraging results (Roeder and Wasserman 1997, Campbell et al. 1997, Dasgupta and Raftery 1998).

For $\log N > 2$, that is $N > 8$, it can be easily verified that BIC penalizes the model complexity more heavily than AIC. Therefore, it reduces the tendency of AIC to fit too many components. On the other hand, Celeux and Soromenho (1996) showed that BIC fits too few components when the sample size is limited and the model for the component densities is valid. Conversely, if the model for the component densities is not valid, then Biernacki et al. (2000) found that BIC tends to fit too many components.

Finally, as mentioned by McLachlan and Peel (2000), we emphasize that not only BIC can be used to estimate Q, but it can also help deciding which model to adopt for the component densities (Biernacki et al. 1999).

Classification likelihood criterion

The complete-data log-likelihood, also called classification log-likelihood, is given by:

$$\log p(\mathbf{X}, \mathbf{Z}\,|\,\boldsymbol{\theta}) = \log p(\mathbf{X}\,|\,\boldsymbol{\theta}) + \log p(\mathbf{Z}\,|\,\mathbf{X}, \boldsymbol{\theta}). \tag{1.31}$$

For many mixture models (see for instance Section 1.1.3), the posterior distribution over the latent variables can be factorized and computed analytically. The functional form of the prior is conserved and we obtain a product of multinomial distributions:

$$\begin{aligned} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}) &= \prod_{i=1}^{n} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i) \\ &= \prod_{i=1}^{n} \prod_{q=1}^{Q} \tau_{iq}^{Z_{iq}}, \end{aligned} \tag{1.32}$$

where τ_{iq} is the posterior probability that observation i belongs to class q. As mentioned by Hathaway (1986), using (1.32) in (1.31) leads to:

$$\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) = \log p(\mathbf{X} \mid \boldsymbol{\theta}) + \sum_{i=1}^{n} \sum_{q=1}^{Q} Z_{iq} \log \tau_{iq}. \tag{1.33}$$

If we now set $\mathbf{Z} = \boldsymbol{\tau}$ and $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ in (1.33), we obtain:

$$\log p(\mathbf{X}, \boldsymbol{\tau} \,|\hat{\boldsymbol{\theta}}) = \log p(\mathbf{X} \,|\hat{\boldsymbol{\theta}}) - \mathrm{EN}(\boldsymbol{\tau}), \tag{1.34}$$

where

$$\mathrm{EN}(\boldsymbol{\tau}) = -\sum_{i=1}^{n} \sum_{q=1}^{Q} \tau_{iq} \log \tau_{iq},$$

is the entropy of the fuzzy classification matrix. If the mixture components are well separated, it will be close to zero. Otherwise, it will have a large value. Biernacki and Govaert (1997) originally proposed (1.34) as a model selection criterion for Gaussian mixture models although it has a broader applicability. Indeed, it only requires that the posterior distribution over the latent variables takes the factorized form (1.32). This criterion is referred as the Classification Likelihood Criterion (CLC). Biernacki and Govaert (1997) suggested to use the EM algorithm to estimate both $\boldsymbol{\tau}$ and $\hat{\boldsymbol{\theta}}$ from the data. According to Biernacki et al. (1999), CLC works well when the class proportions are restricted to being equal. On the other hand, if they are different, because CLC does not penalize the number K of mixture model parameters, it tends to overestimate the correct number of classes.

Integrated classification likelihood criterion

The Integrated Classification Likelihood (ICL) criterion was introduced by Biernacki et al. (2000). It relies on an asymptotic approximation of the integrated complete-data log-likelihood $\log p(\mathbf{X}, \mathbf{Z})$. So far, to keep the notations uncluttered, we have denoted $\boldsymbol{\theta}$ the set of all the mixture model parameters. However, to sketch the development of ICL, we now need to go back to the notations used in Section 1.1.2. Thus, $\boldsymbol{\kappa}$ denotes all the parameters which describe the component densities (*i.e.* the mean vectors and covariance matrices in Gaussian mixture models) and $\boldsymbol{\alpha}$ the class proportions. If the prior $p(\boldsymbol{\theta})$ is factorized such that:

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\kappa}) p(\boldsymbol{\alpha}),$$

then using (Appendix A.1), $\log p(\mathbf{X}, \mathbf{Z})$ is given by:

$$\log p(\mathbf{X}, \mathbf{Z}) = \log p(\mathbf{X} \mid \mathbf{Z}) + \log p(\mathbf{Z}). \tag{1.35}$$

A BIC-like approximation (see Section 1.1.4) applied on the first term of the right hand side of (1.35) leads to:

$$\log p(\mathbf{X} \mid \mathbf{Z}) \approx \log p(\mathbf{X} \mid \mathbf{Z}, \hat{\boldsymbol{\kappa}}) - \frac{K_1}{2} \log N, \tag{1.36}$$

where K_1 is the number of parameters in $\boldsymbol{\kappa}$ and $\hat{\boldsymbol{\kappa}}$ is a maximum likelihood or MAP estimate of $\boldsymbol{\kappa}$. Using a Jeffreys non informative prior distribution for the class proportions $\boldsymbol{\alpha}$, Biernacki et al. (2000) obtained an analytical expression for $\log p(\mathbf{Z})$:

$$\log p(\mathbf{Z}) = \log \Gamma(\frac{Q}{2}) + \sum_{q=1}^{Q} \log \Gamma(\frac{1}{2} + n_q) - Q \log \Gamma(\frac{1}{2}) - \log \Gamma(N + \frac{Q}{2}), \tag{1.37}$$

where $n_q = \sum_i^N Z_{iq}, \forall q$ and $\Gamma(\cdot)$ is the Gamma function. For more details, see (Appendix A.2). Assuming that N and the n_qs take large values (namely the α_qs are far from 0), Biernacki et al. (2000) relied on the Stirling formulae (Appendix A.3) to obtain an approximation of (1.37):

$$\log p(\mathbf{Z}) \approx \log p(\mathbf{Z} \mid \hat{\boldsymbol{\alpha}}) - \frac{Q-1}{2} \log N, \tag{1.38}$$

where $\hat{\boldsymbol{\alpha}} = \max_{\boldsymbol{\alpha}} \log p(\mathbf{Z} \mid \boldsymbol{\alpha})$. Finally, using (1.36) and (1.38) in (1.35) leads to:

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{Z}) &\approx \log p(\mathbf{X} \mid \mathbf{Z}, \hat{\boldsymbol{\kappa}}) - \frac{K_1}{2} \log N + \log p(\mathbf{Z} \mid \hat{\boldsymbol{\alpha}}) - \frac{Q-1}{2} \log N \\ &= \log p(\mathbf{X}, \mathbf{Z} \mid \hat{\boldsymbol{\theta}}) - \frac{K_1 + Q - 1}{2} \log N. \end{aligned}$$

Therefore, if we fix $\mathbf{Z} = \boldsymbol{\tau}$ as in Section 1.1.4, an (asymptotic) ICL criterion is given by:

$$\begin{aligned} ICL &= \log p(\mathbf{X}, \boldsymbol{\tau} \mid \hat{\boldsymbol{\theta}}) - \frac{K_1 + Q - 1}{2} \log N \\ &= \log p(\mathbf{X} \mid \hat{\boldsymbol{\theta}}) - \mathrm{EN}(\boldsymbol{\tau}) - \frac{K_1 + Q - 1}{2} \log N \\ &= CLC - \frac{K_1 + Q - 1}{2} \log N \\ &= CLC - \frac{K}{2} \log N, \end{aligned} \tag{1.39}$$

since K, the total number of unknown parameters in $\boldsymbol{\theta}$, is given by $K = K_1 + Q - 1$. The ICL criterion was proposed in order to favor models with well separated components (as in CLC) as well as penalizing the model complexity by the number of unknown mixture model parameters (as in BIC).

1.2 Variational Inference

In this section, we introduce some variational techniques which lie at the core of the main inference strategies for networks developed in this thesis. We first describe the variational EM algorithm which generalizes EM. We then go to a more general framework, called variational Bayes EM, which is used in Chapters 2 and 4 to obtain an approximation of the full posterior distribution over the model parameters and latent variables. This will naturally lead to two new model selection criteria to estimate the number of classes in networks.

1.2.1 Variational EM

As mentioned in Section 1.1.3, the EM algorithm can be used to find maximum likelihood estimates for models with latent variables. Here, we give a variational treatment of EM which will prove that the algorithm is guaranteed to maximize the observed-data log-likelihood (Csiszar and Tusnady 1984, Hathaway 1986, Neal and Hinton 1998).

Let us consider a data set $\mathbf{X}$ and the corresponding classification matrix $\mathbf{Z}$. We aim at maximizing the observed-data log-likelihood:

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \log \left(\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \right). \tag{1.40}$$

Note that (1.40) involves a summation over every possible matrix $\mathbf{Z}$ because we consider discrete mixture models. However, this analysis goes through unchanged if the latent variables are continuous, simply by replacing the summation with an integration. For any distribution $q(\mathbf{Z})$ over the latent variables, the following decomposition holds:

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \mathcal{L}_{ML}(q;\, \boldsymbol{\theta}) + \mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})\right),$$

where

$$\mathcal{L}_{ML}(q;\, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log\{ \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{q(\mathbf{Z})} \}, \tag{1.41}$$

and

$$\mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})\right) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log\{ \frac{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \}. \tag{1.42}$$

Note that $\mathcal{L}_{ML}(q;\, \boldsymbol{\theta})$ in (1.41) is a functional of the distribution $q(\mathbf{Z})$ as well as a function of the parameter $\boldsymbol{\theta}$. In (1.42), KL denotes the Kullback-Leibler divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$. It satisfies

$$\mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})\right) \geq 0,$$

with equality if, and only if, $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$. Therefore, $\mathcal{L}_{ML}$ is a lower bound of $\log p(\mathbf{X} \mid \boldsymbol{\theta})$:

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) \geq \mathcal{L}_{ML}(q;\, \boldsymbol{\theta}).$$

Suppose that the current value of the parameters is $\boldsymbol{\theta}^{old}$. During the E step, $\mathcal{L}_{ML}(q;\, \boldsymbol{\theta}^{old})$ is maximized with respect to $q(\mathbf{Z})$. The solution to this

optimization step occurs when the KL divergence vanishes that is when $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})$. The observed-data log-likelihood is then equal to its lower bound given by:

$$\begin{aligned}\mathcal{L}_{ML}(q;\, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \\ &= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) + \text{const},\end{aligned} \tag{1.43}$$

where all the terms that do not depend on $\boldsymbol{\theta}$ have been absorbed into the constant. Note that the first term on the right hand side of (1.43) is the expectation $\mathcal{Q}_{ML}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ of the complete-data log-likelihood (see Section 1.1.3). During the M step, $q(\mathbf{Z})$ is held fixed while $\mathcal{L}_{ML}(q;\, \boldsymbol{\theta})$ is maximized with respect to $\boldsymbol{\theta}$ to obtain a new estimate $\boldsymbol{\theta}^{new}$. This causes the lower bound to increase which will necessarily cause the corresponding observed-data log-likelihood to increase.

As already mentioned, the EM algorithm can also be used to find MAP estimates when the goal is to maximize $\log p(\boldsymbol{\theta} \mid \mathbf{X})$. The corresponding variational decomposition is given by:

$$\log p(\boldsymbol{\theta} \mid \mathbf{X}) = \mathcal{L}_{MAP}(q;\, \boldsymbol{\theta}) + \mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})\right),$$

where

$$\mathcal{L}_{MAP}(q;\, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log\{\frac{p(\boldsymbol{\theta}, \mathbf{Z} \mid \mathbf{X})}{q(\mathbf{Z})}\},$$

and KL is again the Kullback-Leibler divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$.

So far, we have assumed that the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$ could be computed analytically. However, for some mixture models it is not tractable and therefore variational approximations are required. This gives rise to the Variational EM (VEM) algorithm (Algorithm 4). During the variational E step, the lower bound $\mathcal{L}(q;\, \boldsymbol{\theta}^{old})$ is maximized with respect to $q(\mathbf{Z})$, where $\mathcal{L}$ either denotes $\mathcal{L}_{ML}$ or $\mathcal{L}_{MAP}$. This maximization induces a minimization of the KL divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})$. To obtain a tractable algorithm, it is often assumed that $q(\mathbf{Z})$ can be factorized such that:

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

The solution to this optimization procedure is an approximation $q(\mathbf{Z})$ of $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})$. During the variational M step, this approximation is used to compute the lower bound $\mathcal{L}(q;\, \boldsymbol{\theta})$ which is then maximized with respect to $\boldsymbol{\theta}$. These two step are repeated until convergence.

1.2.2 Variational Bayes EM

In the previous sections, we have seen how EM strategies could be used to obtain point estimates of mixture model parameters. Moreover, we motivated the use of a Bayesian treatment of the data in order to deal with the singularities of Gaussian mixture models which arise in the maximum likelihood setting. However, the Bayesian framework brings much more powerful features. Indeed, rather than looking for point estimates of the

Algorithm 4: Variational EM algorithm for mixture models. $\mathcal{L}$ either denotes $\mathcal{L}_{ML}$ or $\mathcal{L}_{MAP}$.

```
// INITIALIZATION
```
Initialize $\boldsymbol{\theta}_0$
$\boldsymbol{\theta}^{new} \leftarrow \boldsymbol{\theta}_0$

```
// OPTIMIZATION
```
repeat
  $\boldsymbol{\theta}^{old} \leftarrow \boldsymbol{\theta}^{new}$
  `// Variational E-step`
  Find $q(\mathbf{Z}) = \text{argmax}_{q(\mathbf{Z})}\mathcal{L}(q;\, \boldsymbol{\theta}^{old})$
  `// Variational M-step`
  Find $\boldsymbol{\theta}^{new} = \text{argmax}_{\boldsymbol{\theta}}\mathcal{L}(q;\, \boldsymbol{\theta})$
until $\boldsymbol{\theta}$ *converges*

model parameters, we are now going to see how an approximation of the full posterior distribution over the model parameters and latent variables can be obtained. Such approximation is particularly relevant when studying the variability of the MAP estimates. This also gives rise to new model selection criteria.

In a Bayesian framework, all the model parameters are regarded as random variables drawn from a prior distribution $p(\boldsymbol{\theta})$. If we denote $\mathbf{X}$ an observed data set and $\mathbf{Z}$ the corresponding classification matrix, the goal is to estimate $p(\mathbf{Z}, \boldsymbol{\theta} \,|\, \mathbf{X})$. The marginal log-likelihood, also called integrated observed-data log-likelihood, is given by:

$$\begin{aligned} \log p(\mathbf{X}) &= \log \left\{ \int p(\mathbf{X}, \boldsymbol{\theta}) d\,\boldsymbol{\theta} \right\} \\ &= \log \left\{ \sum_{\mathbf{Z}} \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta}) d\,\boldsymbol{\theta} \right\}. \end{aligned} \tag{1.44}$$

For any distribution $q(\mathbf{Z}, \boldsymbol{\theta})$, the following decomposition holds:

$$\log p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}\left(q(\cdot) \,||\, p(\cdot \,|\, \mathbf{X})\right),$$

where

$$\mathcal{L}(q) = \sum_{\mathbf{Z}} \int q(\mathbf{Z}, \boldsymbol{\theta}) \log\{\frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\theta})}{q(\mathbf{Z}, \boldsymbol{\theta})}\} d\,\boldsymbol{\theta}, \tag{1.45}$$

and

$$\text{KL}\left(q(\cdot) \,||\, p(\cdot \,|\, \mathbf{X})\right) = -\sum_{\mathbf{Z}} \int q(\mathbf{Z}, \boldsymbol{\theta}) \log\{\frac{p(\mathbf{Z}, \boldsymbol{\theta} \,|\, \mathbf{X})}{q(\mathbf{Z}, \boldsymbol{\theta})}\} d\,\boldsymbol{\theta}.$$

The functional $\mathcal{L}$ defined in (1.45) is a lower bound of the marginal log-likelihood, that is $\log p(\mathbf{X}) \geq \mathcal{L}(q)$, with equality iff $q(\mathbf{Z}, \boldsymbol{\theta}) = p(\mathbf{Z}, \boldsymbol{\theta} \,|\, \mathbf{X})$. However, we shall suppose that working with the true posterior distribution is intractable. To obtain an approximation of the posterior, we restrict our search to a family of factorized distributions. In practice, it is often

assumed that:

$$\begin{aligned} q(\mathbf{Z}, \boldsymbol{\theta}) &= q(\mathbf{Z})q(\boldsymbol{\theta}) \\ &= q(\boldsymbol{\theta}) \prod_{i=1}^{N} q(\mathbf{Z}_i). \end{aligned} \tag{1.46}$$

More generally, the hidden variables $(\mathbf{Z}, \boldsymbol{\theta})$ can be classified into P disjoint groups $\{\mathbf{g}_1, \ldots, \mathbf{g}_P\}$ of a partition $\mathbf{G}$ such that:

$$q(\mathbf{Z}, \boldsymbol{\theta}) = q(\mathbf{G}) = \prod_{i=1}^{P} q_i(\mathbf{g}_i). \tag{1.47}$$

In the following, we denote $\mathbf{G}^{\setminus j}$ all the groups of $\mathbf{G}$ except $\mathbf{g}_j$. Using (1.47) in (1.45), the lower bound becomes:

$$\begin{aligned} \mathcal{L}(q) &= \int q(\mathbf{G}) \log \frac{p(\mathbf{X}, \mathbf{G})}{q(\mathbf{G})} \, d\,\mathbf{G} \\ &= \int \prod_i q_i(\mathbf{g}_i) \left(\log p(\mathbf{X}, \mathbf{G}) - \sum_i \log q_i(\mathbf{g}_i) \right) d\,\mathbf{G} \\ &= \int \prod_i q_i(\mathbf{g}_i) \log p(\mathbf{X}, \mathbf{G}) \, d\,\mathbf{G} - \int \prod_i q_i(\mathbf{g}_i) \sum_i \log q_i(\mathbf{g}_i) \, d\,\mathbf{G} \\ &= \int q_j(\mathbf{g}_j) \left(\int \prod_{i \neq j} q_i(\mathbf{g}_i) \log p(\mathbf{X}, \mathbf{G}) \, d\,\mathbf{G}^{\setminus j} \right) d\,\mathbf{g}_j - \int \prod_i q_i(\mathbf{g}_i) \log q_j(\mathbf{g}_j) \, d\,\mathbf{G} \\ &\qquad - \int \prod_i q_i(\mathbf{g}_i) \sum_{i \neq j} \log q_i(\mathbf{g}_i) \, d\,\mathbf{G} \\ &= \int q_j(\mathbf{g}_j) \mathrm{E}_{\mathbf{G}^{\setminus j}}[\log p(\mathbf{X}, \mathbf{G})] \, d\,\mathbf{g}_j - \int q_j(\mathbf{g}_j) \log q_j(\mathbf{g}_j) \, d\,\mathbf{g}_j \\ &\qquad - \sum_{i \neq j} \int q_i(\mathbf{g}_i) \log q_i(\mathbf{g}_i) \, d\,\mathbf{g}_i \\ &= \int q_j(\mathbf{g}_j) \Big(\log \hat{p}(\mathbf{X}, \mathbf{g}_j) - \text{const} \Big) \, d\,\mathbf{g}_j - \int q_j(\mathbf{g}_j) \log q_j(\mathbf{g}_j) \, d\,\mathbf{g}_j \\ &\qquad + \sum_{i \neq j} \mathrm{H}[\mathbf{g}_i] \\ &= -\mathrm{KL}\left(q_j(\cdot) || \hat{p}(\cdot) \right) + \sum_{i \neq j} \mathrm{H}[\mathbf{g}_i] - \text{const}, \end{aligned} \tag{1.48}$$

where

$$\begin{aligned} \log \hat{p}(\mathbf{X}, \mathbf{g}_j) &= \mathrm{E}_{\mathbf{G}^{\setminus j}}[\log p(\mathbf{X}, \mathbf{G})] + \text{const} \\ &= \int \prod_{i \neq j} q_i(\mathbf{g}_i) \log p(\mathbf{X}, \mathbf{G}) \, d\,\mathbf{G}^{\setminus j}, \end{aligned}$$

and $\mathrm{H}[bg_i] = -\int q_i(\mathbf{g}_i) \log q_i(\mathbf{g}_i) d\,\mathbf{g}_i$ is the entropy of $\mathbf{g}_i(\cdot)$. To simplify the notations, we have used some integrations in formulating the lower bound. In fact, we emphasize that the latent variables we consider are discrete and therefore, depending on the groups of $\mathbf{G}$, some integrations should be replaced by summations as required. If the factors in (1.47) are assumed to be fixed except the distribution $q_j(\mathbf{g}_j)$, then according to (1.48), maximizing $\mathcal{L}(q)$ with respect to $q_j(\mathbf{g}_j)$ is equivalent to minimizing

the Kullback-Leibler divergence between $q_j(\mathbf{g}_j)$ and $\hat{p}(\mathbf{X}, \mathbf{g}_i)$. Therefore, the optimal approximation for the j-th factor is:

$$\log q_j(\mathbf{g}_j) = \log \hat{p}(\mathbf{X}, \mathbf{g}_j) = \mathrm{E}_{\mathbf{G}^{\setminus j}}[\log p(\mathbf{X}, \mathbf{G})]) + \text{const},$$

where the constant can be determined by normalizing $q_j(\mathbf{g}_j)$. Thus, we have obtained a set of P coupled equations. Indeed, each of the factor is optimized by computing an expectation with respect to all the other factors. As a consequence, the factors are first initialized and then by cycling through the equations and replacing each factor with its new approximation, we obtain a consistent solution for the variational optimization procedure. Convergence is guaranteed because the lower bound is convex with respect to each factor.

Algorithm 5: Variational Bayes EM algorithm for mixture models.

```
// INITIALIZATION
```

Initialize $q_i^0(\mathbf{g}_i), \forall i$
$q_i^{new}(\mathbf{g}_i) \leftarrow q_i^0(\mathbf{g}_i), \forall i$

```
// OPTIMIZATION
```

repeat
  $q_i^{old}(\mathbf{g}_i) \leftarrow q_i^{new}(\mathbf{g}_i), \forall i$
  $q_i^{new}(\mathbf{g}_i) \leftarrow \exp\left(\int \prod_{j \neq i} q_j^{old}(\mathbf{g}_j) \log p(\mathbf{X}, \mathbf{G})\, d\,\mathbf{G}^{\setminus i}\right), \forall i$
  Normalize $q_i^{new}(\mathbf{g}_i), \forall i$
until $\mathcal{L}(q)$ *converges*

For the factorization (1.46), some optimization equations involve the distributions $q(\mathbf{Z}_i)$ over the latent variables while others focus on the factor $q(\boldsymbol{\theta})$. These two steps are related to the E and M steps of EM strategies. Therefore, the optimization algorithm described in this section is usually referred as the Variational Bayes EM (VBEM) algorithm (Algorithm 5). It can be seen that VEM is a limiting case of VBEM in which the distribution over the model parameters $q(\boldsymbol{\theta})$ is collapsed to point estimates at the mode of the distribution (Hofman and Wiggins 2008).

1.3 Graph clustering

In the previous sections, we introduced mixture models and described numerous EM strategies for estimation and clustering purposes. We now concentrate on the classification of vertices in networks depending on their connection profiles. There has been a wealth of literature on the topic which goes back to the earlier work of Moreno (1934). As shown in Newman and Leicht (2007), it appears that available methods can be grouped into three significant categories. First, some models look for community structure, also called assortative mixing (Newman 2003, Danon et al. 2005), where vertices are partitioned into classes such that vertices of a class are mostly connected to vertices of the same class. They are particularly suitable for the analysis of affiliation networks. Other models look for disassortative mixing in which vertices mostly connect to vertices of different

classes. They are commonly used to analyze bipartite networks (Estrada and Rodriguez-Velazquez 2005). These models are not considered in this thesis. Finally, a few procedures look for heterogeneous structure where vertices can have different types of connection profiles. In particular, they can be used to uncover both community structure and disassortative mixing.

In this section, we first review some basic definitions of graph theory and give some examples of real networks. We then describe some of the most widely used graph clustering methods. Note that many model free approaches exist (Fortunato 2010). However, except for the algorithmic approach presented in Section 1.3.2, we concentrate in the following on methods which rely on statistical models only, as in Goldenberg et al. (2010).

1.3.1 Graph theory and real networks

Networks are used in many scientific fields to represent the interactions between objects of interest. For instance, in Biology, regulatory networks can describe the regulation of genes with transcriptional factors (Milo et al. 2002), while metabolic networks focus on representing pathways of biochemical reactions (Lacroix et al. 2006). Besides, the binding procedures of proteins are often described as protein-protein interaction networks (Albert and Barabási 2002, Barabási and Oltvai 2004). In social sciences, networks are widely used to represent relational ties between actors (Snijders and Nowicki 1997, Nowicki and Snijders 2001, Palla et al. 2007). Other examples of networks are powergrids (Watts and Strogatz 1998) and the Internet (Zanghi et al. 2008).

A network is commonly represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of N vertices and $\mathcal{E}$ is a set of edges between pairs of vertices. The graph is said to be directed (Figure 1.1) if the pairs (u, v) in $\mathcal{E}$ are ordered. Conversely, unordered pairs form an undirected graph (Figures 1.2 and 1.3). Note that the edges can be weighted by a function $w : \mathcal{E} \to \mathbb{F}$ for any set $\mathbb{F}$. However, in this thesis, we will concentrate only on binary graphs, that is $\mathbb{F} = \{0, 1\}$. The size of $\mathcal{G}$ is then given through the edge count $m = |\mathcal{E}|$. The graph is said to be dense if m is close to the maximal number M of edges whereas a low value of m leads to a sparse graph. To characterize the density of $\mathcal{G}$, a criterion $\delta(\mathcal{G})$ is often used. It is defined as the ratio of the number m of existing edges over the number M of potential edges:

$$\delta(\mathcal{G}) = \frac{m}{M}.$$

For a directed graph, $M = N^2$ while $M = N(N+1)/2$ otherwise. If $\mathcal{G}$ does not contain any self loop, that is an edge from a vertex to itself, then $M = N(N-1)$ for a directed graph and $M = N(N-1)/2$ otherwise.

The neighbourhood $N_{\mathcal{G}}(u)$ of vertex u is defined as the set of all the vertices connected to u. Its degree $d(u)$ is equal to its number of incident edges. Finally, a path from a vertex u to a vertex v is a sequence of edges in $\mathcal{E}$ starting at vertex $v_0 = u$ and ending at vertex $v_{k+1} = v$:

$$\{u, v_1\}, \{v_1, v_2\}, \ldots, \{v_k, v\}.$$

If there exists at least one path between every pair of vertices then the graph is said to be connected. For instance, the graph in Figure 1.1 is connected contrary to the graphs in Figures 1.2 and 1.3 which have some isolated vertices.

So far, we have denoted $\mathbf{X}$ a data matrix whose ith row represents observation $\mathbf{x}_i$. Because we considered N observations in $\mathbb{R}^d$, $\mathbf{X}$ was in $\mathbb{R}^{N\times d}$. The matrix $\mathbf{X}$ is now an adjacency matrix which describes the presence or absence of an edge in a graph. As mentioned already, we focus on binary graphs and therefore $\mathbf{X}$ is in $\{0,1\}^{N\times N}$. Thus, if there exists an edge from vertex i to vertex j then X_{ij} equals 1 and 0 otherwise. At this point, it is crucial to emphasize that N, which denotes the number of vertices, is no longer the number of observations in the data. Indeed, when considering graphs, the information about the data distribution comes from the presence or absence of edges. Therefore, the total number of observations is in $O(N^2)$. As we shall see shortly, for many graph clustering models, the edges are not independent and so approximation techniques are required for estimation and classification purposes.

Properties of real networks

Very interestingly, most real networks have been shown to share some properties (Albert et al. 1999, Broder et al. 2000, Dorogovtsev et al. 2000, Amaral et al. 2000, Strogatz 2001) that we briefly recall in the following.

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

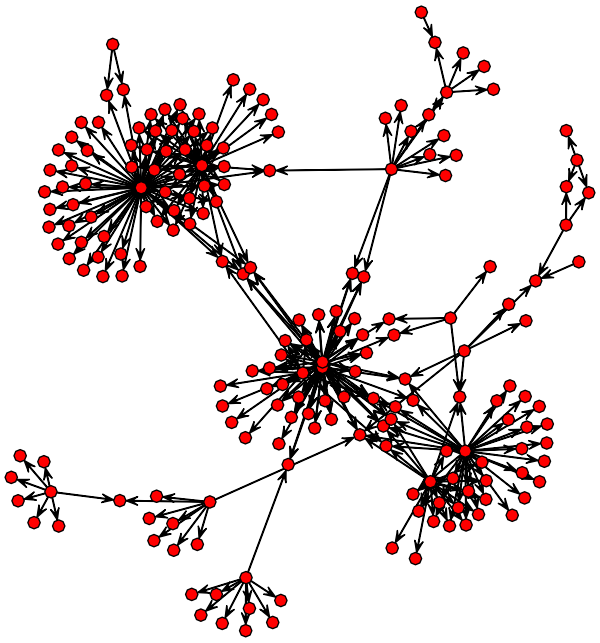

Figure 1.1 – *Subset of the yeast transcriptional regulatory network (Milo et al. 2002). Nodes of the directed network correspond to genes, and two genes are linked if one gene encodes a transcriptional factor that directly regulates the other gene.*

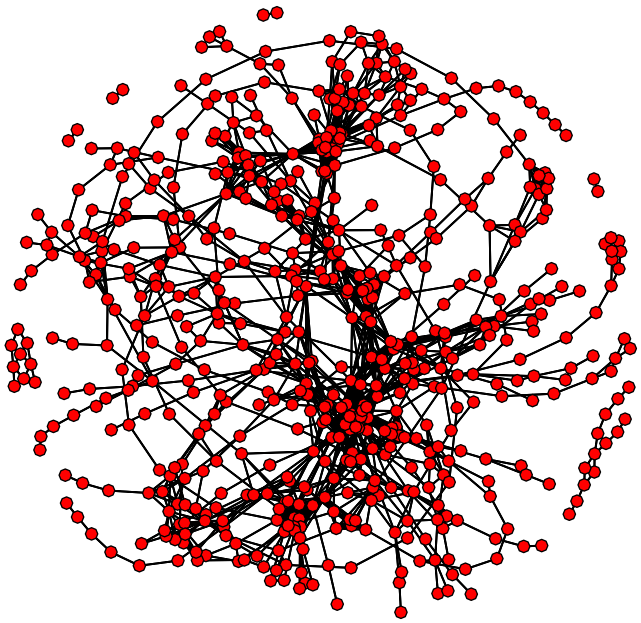

Figure 1.2 – *The metabolic network of bacteria* Escherichia coli *(Lacroix et al. 2006). Nodes of the undirected network correspond to biochemical reactions, and two reactions are connected if a compound produced by the first one is a part of the second one (or vice-versa).*

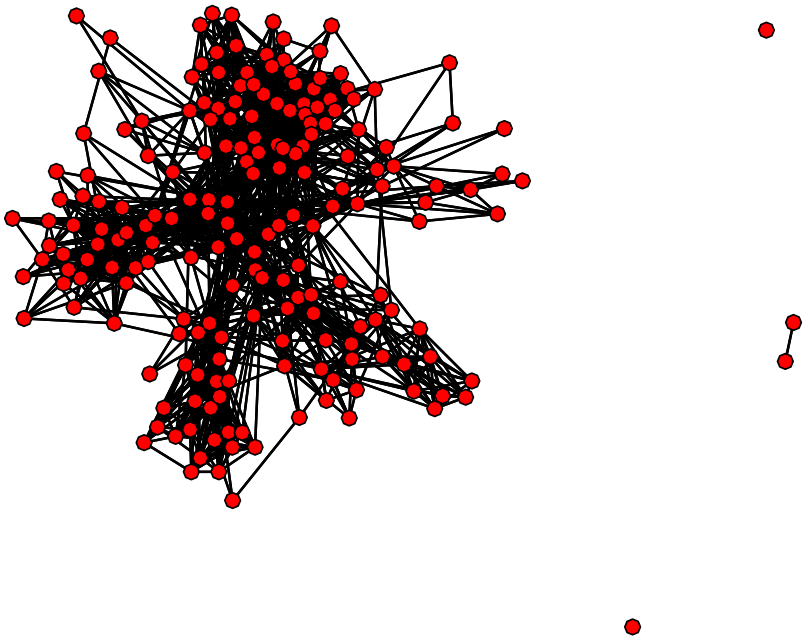

Figure 1.3 – *Subset of the french political blogosphere network. The data consists of a single day snapshot of political blogs automatically extracted on 14th october 2006 and manually classified by the "Observatoire Présidentielle project" (Zanghi et al. 2008). Nodes correspond to hostnames and there is an edge between two nodes if there is a known hyperlink from one hostname to another (or vice-versa).*

1.3.2 Community structure

Many graph clustering methods aim at detecting community structure, also called assortative mixing, meaning the appearance of densely connected groups of vertices, with only sparser connections between groups (Figure 1.4). Most of them rely on the modularity score of Newman and Girvan (2004). However, we point out the recent work of Bickel and Chen (2009) who showed that these algorithms are (asymptotically) biased and that using modularity scores could lead to the discovery of an incorrect community structure, even for large graphs.

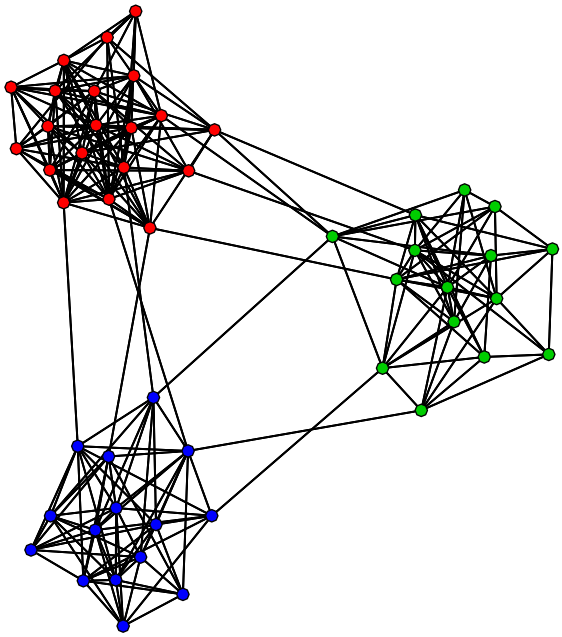

Figure 1.4 – *Example of an undirected affiliation network with 50 vertices. The network is made of three communities represented in red, blue, and green. Vertices connect mainly to vertices of the same community.*

Modularity score

Girvan and Newman (2002), Newman and Girvan (2004) proposed several intuitive community detection algorithms which involve iterative removal of edges from the network to split it into communities. Edges to be removed are identified using one of a number of possible betweenness measures. All of them are based on the same idea. If two communities are joined by only a few *inter* community edges, then all paths from vertices in one community to vertices in the other must pass along one of those

few edges. Therefore, given a suitable set of paths, we expect the number of paths that go along an edge to be largest for *inter* community edges.

First, they introduced the edge betweenness which is a generalization to edges of the (vertex) betweenness measure of Freeman (1977). The edge betweenness of an edge is defined as the number of shortest paths between all pairs of vertices in the network that run along that edge. Second, they considered the random walk betweenness. The expected number of times a random walk between a particular pair of vertices will pass down a particular edge is calculated. This expected value is then summed over all pairs of vertices to obtain the random walk betweenness of the edge. As shown in Newman and Girvan (2004), other scores can obviously be considered to obtain algorithms that may be more appropriate for some applications. However, it appears that the choice of measure does not highly influence the result of the algorithms. On the other hand, the recalculation step after each edge removal is crucial (see Algorithm 6).

All these algorithms produce a dendrogram (Figure 1.5) which represents an entirely nested hierarchy of possible community divisions for the network. In order to select one of these divisions, Newman and Girvan (2004) proposed a modularity criterion. Consider a particular division with Q communities and let us denote e_{ql} the fraction of all edges in the network that link vertices in community q to vertices in community l. Moreover, consider the fraction $a_q = \sum_{l=1}^{Q} e_{ql}$ of edges that connect to vertices of community q. The modularity criterion is then given by:

$$mod = \sum_{q=1}^{Q} (e_{qq} - a_q^2). \tag{1.49}$$

The criterion is computed for all the divisions, and a division is chosen such that the modularity is maximized.

A limiting factor of these community detection algorithms is their poor scaling with the number m of edges and the number N of vertices in the network. For instance calculating the shortest paths between a particular pair of vertices can be done in $O(m)$ (Ahuja et al. 1993, Cormen et al. 2001). Because they are $O(N^2)$ vertex pairs, the computational cost to compute all the edge betweenness scores is in $O(mN^2)$. This complexity was improved independently by Newman (2001) and Brandes (2001) finding all betweennesses in $O(mN)$. Since this calculation has to be repeated for the removal of each edge, the entire algorithm runs in worst-case time $O(m^2N)$. In other words, for dense networks, where m is in $O(N^2)$, it runs in $O(N^5)$ while it scales in $O(N^3)$ for sparse networks, where m is linear in N.

Algorithm 6: Example of a community structure detection algorithm with a betweenness score.

repeat
  Calculate betweenness scores for all edges
  Remove the edge with the highest score
until *No edges remain*

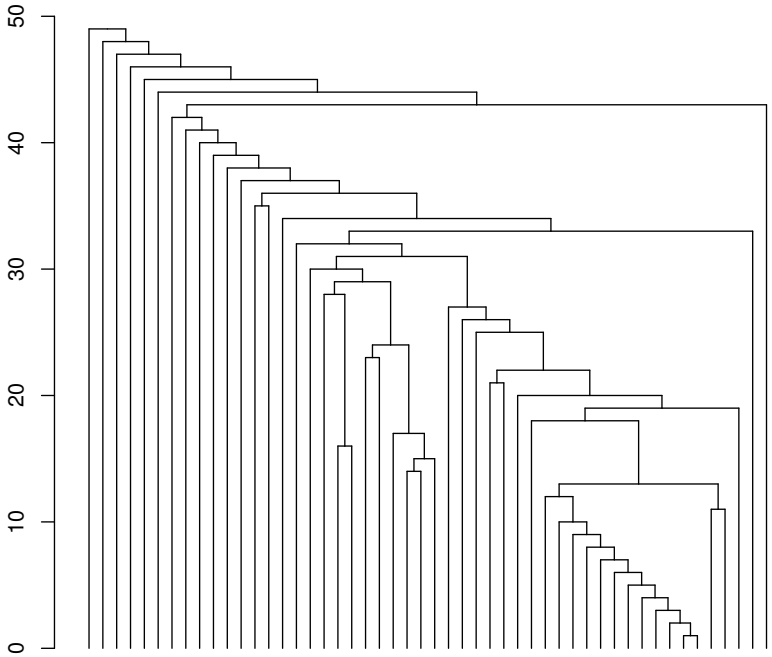

Figure 1.5 – *Dendrogram of a network with* 50 *vertices for the community detection algorithm with edge betweenness. It should be read from top to bottom. The algorithm starts with a single community which contains all the vertices. Edges with the highest edge betweenness are then removed iteratively splitting the network into several communities. After convergence, each vertex, represented by a leaf of the tree, is a sole member of one of the* 50 *communities.*

Rather than building the complete dendrogram (with edge removals) and then choosing the optimal division using the modularity criterion, Newman (2004) suggested to focus directly on the optimization of the modularity. Thus, he proposed an algorithm which falls in the general category of agglomerative hierarchical clustering methods (Everitt 1974, Scott 2000). Starting with a configuration in which each vertex is the sole member of one of N communities, the communities are iteratively joined together in pairs, choosing at each step the join that results in the greatest increase (or smallest decrease) in *mod* (1.49). Again, this leads to a dendrogram for which the best cut is chosen by looking for the maximal value of the modularity. The computational cost of the entire algorithm is in $O((m+N)N)$, or $O(N^3)$ for dense networks and $O(N^2)$ for sparse networks. It was shown to be capable of handling a collaboration network with 50000 vertices in Newman (2004).

All the algorithms described in this section are implemented in the R package "igraph" which is available on the CRAN:
`http://cran.r-project.org/web/packages/igraph`

Latent position cluster model

An alternative approach for community detection in networks is the Latent Position Cluster Model (LPCM) of Handcock et al. (2007). Consider a $N \times N$ binary adjacency matrix $\mathbf{X}$ such that X_{ij} equals 1 if there is an edge from vertex i to vertex j, and 0 otherwise. Moreover, let us define $\mathbf{Y}$ a covariate information where $\mathbf{Y}_{ij}$ denotes some observed characteristics about the pair (i,j) of vertices. This might represent for instance the traffic information of users from blog i to blog j in a blogosphere network (see Figure 1.3). Several characteristics can possibly be observed for each pair of vertices and therefore $\mathbf{Y}_{ij}$ can be vector valued. Note that a few other random graph models have been proposed in the literature to take covariates into account (see for instance Zanghi et al. 2010, Mariadassou et al. 2010). They will not be considered in this thesis where the goal is to cluster the vertices by using the network topology only. Here, we describe LPCM in a general setting, as in Handcock et al. (2007), and emphasize that the algorithm can also be used if $\mathbf{Y}$ is not available, simply by removing the terms in $\mathbf{Y}_{ij}$ in the following expressions.

LPCM assumes that the network does not contain any self loop while both directed and undirected relations can be analyzed. It is assumed that each vertex, usually called actor in social sciences, has an unobserved position in a d dimensional Euclidean latent space as in Hoff et al. (2002). Given the latent positions and the covariate information, the edges are assumed to be drawn from a Bernoulli distribution:

$$X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij} \sim \mathcal{B}\left(g(a_{\mathbf{Z}_i,\mathbf{Z}_j,\mathbf{Y}_{ij}})\right).$$

The function $g(x) = 1/(1+e^{-x})$ is the logistic sigmoid function. Moreover $a_{\mathbf{Z}_i,\mathbf{Z}_j,\mathbf{Y}_{ij}}$ is given by:

$$a_{\mathbf{Z}_i,\mathbf{Z}_j,\mathbf{Y}_{ij}} = \mathbf{Y}_{ij}^{\mathsf{T}} \boldsymbol{\beta}_0 - \beta_1 | \, \mathbf{Z}_i - \mathbf{Z}_j \, |, \tag{1.50}$$

where $\boldsymbol{\beta}_0$ as the same dimensionality as $\mathbf{Y}_{ij}$ and β_1 is a scalar. Both $\boldsymbol{\beta}_0$ and β_1 are unknown parameters to be estimated. To represent clustering, the

positions are assumed to be drawn from a finite mixture of Q multivariate normal distributions (see Section 1.1.2), each one representing a different class of vertices. Each multivariate distribution has its own mean vector as well as spherical covariance matrix:

$$\mathbf{Z}_i \sim \sum_{q=1}^{Q} \alpha_q \mathcal{N}(\boldsymbol{\mu}_q, \sigma_q^2 \mathbf{I}),$$

and $\boldsymbol{\alpha}$ denotes a vector of class proportions which satisfies $\alpha_q > 0, \forall q$ and $\sum_{q=1}^{Q} \alpha_q = 1$. Finally, according to LPCM, the latent positions $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$ are iid and given this latent structure, all the edges are supposed to be independent. Consider now the second term on the right hand side of (1.50). By construction, if β_1 is positive, we expect the L_1 distance $| \mathbf{Z}_i - \mathbf{Z}_j |$ to be smaller if vertices i and j are in the same class. In other words, the probability $g(a_{\mathbf{Z}_i, \mathbf{Z}_j, \mathbf{Y}_{ij}})$ of an edge between i and j is supposed to be higher for vertices sharing the same class. Note that this corresponds exactly to the definition of a community.

Handcock et al. (2007) proposed a two-stage maximum likelihood approach and a Bayesian algorithm, as well as a BIC criterion to estimate the number of latent classes. The two-stage maximum likelihood approach first maps the vertices in the latent space and then uses a mixture model to cluster the resulting positions. In practice, this procedure converges more quickly but looses some information by not estimating the positions and the cluster model at the same time. Conversely, the Bayesian algorithm (see Figure 1.6), based on Markov Chain Monte Carlo, estimates both the latent positions and the mixture model parameters simultaneously. It gives better results but is time consuming. Both the maximum likelihood and the Bayesian approach are limited in the sense that they can handle networks with a few hundreds of vertices only. They are implemented in the R package "latentnet" (Krivitsky and Handcock 2009) which is available on the CRAN:
`http://cran.r-project.org/web/packages/latentnet`

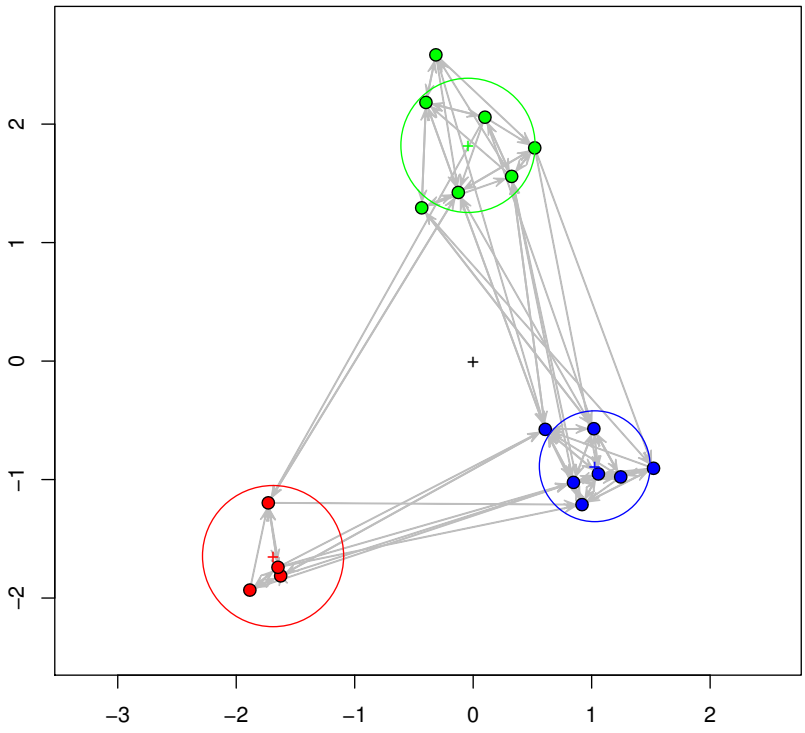

Figure 1.6 – *Directed network of social relations between 18 monks in an isolated American monastery (Sampson 1969, White et al. 1976). Sampson collected sociometric information using interviews, experiments, and observations. This network focus on the relation of "liking". A monk is said to have a social relation of "like" to another monk if he ranked that monk in the top three monks for positive affection in any of the three interviews given. The positions of the vertices in the two data dimensional latent space have been calculated using the Bayesian approach for LPCM. The position of the three class centers found are indicated as well as circles with radius equal to the square root of the class variances estimated.*

1.3.3 Heterogeneous structure

So far, we have seen some algorithms to uncover communities. We now present some other approaches which can look for heterogeneous structure in networks where vertices can have different types of connection profiles.

Hofman and Wiggins

Let us consider a binary adjacency matrix $\mathbf{X}$ representing a network $\mathcal{G}$. The model of Hofman and Wiggins (2008) associates to each vertex of the network a latent variable $\mathbf{Z}_i$ drawn from a multinomial distribution:

$$\mathbf{Z}_i \sim \mathcal{M}\left(1, \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_Q)\right). \tag{1.51}$$

As in other standard mixture models (see Section 1.1.2), the vector $\mathbf{Z}_i$ has all its components set to zero except one such that Z_{iq} equals 1 if vertex i belong to class q. The edges are then assumed to be drawn from a Bernoulli distribution:

$$X_{ij} \sim \mathcal{B}(\lambda),$$

if vertices i and j are in the same class, that is $\mathbf{Z}_i = \mathbf{Z}_j$, and

$$X_{ij} \sim \mathcal{B}(\epsilon),$$

otherwise. Thus, the model can take both community structure ($\lambda > \epsilon$) (Figure 1.4) and disassortative mixing ($\lambda < \epsilon$) (Figure 1.7) into account. As in the previous section, given the latent variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$, all the edges are supposed to be independent. In order to estimate the posterior distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \lambda, \epsilon | \mathbf{X})$ over the latent variables and model parameters, Hofman and Wiggins (2008) used a variational Bayes EM algorithm (see Section 1.2.2) with a factorized distribution:

$$q(\mathbf{Z}, \boldsymbol{\alpha}, \lambda, \epsilon) = q(\boldsymbol{\alpha}) q(\lambda) q(\epsilon) \prod_{i=1}^{N} q(\mathbf{Z}_i).$$

Moreover, they proposed a model selection criterion to estimate the number of latent classes in networks. It relies on a variational approximation of the marginal log-likelihood $\log p(\mathbf{X})$ and has shown promising results. This criterion is experimented in Chapter 2.

The computational cost of the variational Bayes algorithm is $O(N^2 Q)$ such that the algorithm can deal with networks with thousands of vertices. It is implemented in a Matlab package "VBMOD" available at:
`http://vbmod.sourceforge.net`

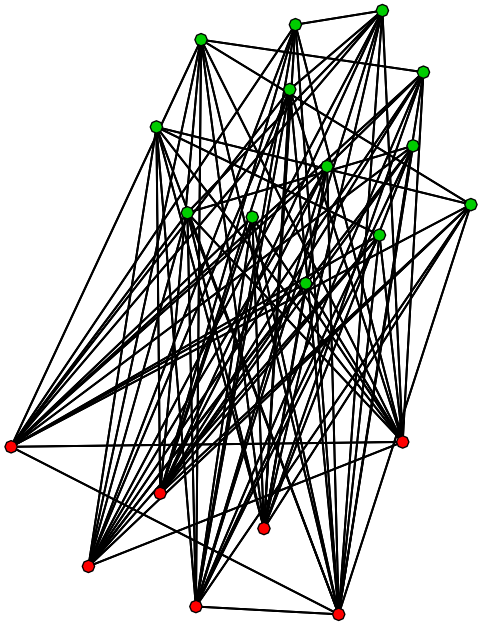

Figure 1.7 – *Example of an undirected network with 20 vertices. The connection probabilities between the two classes in red and green are higher than the* intra *class probabilities. Vertices connect mainly to vertices of a different class.*

Stochastic block models

Originally developed in social sciences, the Stochastic Block Model (SBM) is a probabilistic generalization (Fienberg and Wasserman 1981, Holland et al. 1983) of the method described in White et al. (1976). Given a network, it assumes that each vertex belongs to a hidden class among Q classes, and uses a matrix $\mathbf{\Pi}$ to describe the *intra* and *inter* connection probabilities (Frank and Harary 1982). No assumption is made on the form of the connectivity matrix such that very different structures can be taken into account. In particular, SBM can characterize the presence of hubs which make networks locally dense (Daudin et al. 2008). Moreover and to some extent, it generalizes many of the existing graph clustering techniques, as shown in Newman and Leicht (2007). For instance, the model of Hofman and Wiggins (2008) can be seen as a constrained SBM where the diagonal of $\mathbf{\Pi}$ is set to λ and all the other elements to ϵ.

Formally, SBM considers a latent variable $\mathbf{Z}_i$, drawn from a multinomial distribution (1.51), for each vertex in the network. Thus, each vertex belongs to a single class, and that class is q if Z_{iq} equals 1. The edges are then assumed to be drawn from a Bernoulli distribution:

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

where $\mathbf{\Pi}$ is a $Q \times Q$ matrix of connection probabilities. Again, given all the latent variables, the edges are supposed to be independent. Note that SBM was originally described in a more general setting (Nowicki and Snijders 2001), allowing any discrete relational data. However, as explained in Section 1.3.1, we concentrate in the following on binary edges only.

SBM is related to the infinite block model of Kemp et al. (2004) although the number Q of classes is fixed. Moreover, contrary to the mixed membership stochastic block model of Airoldi et al. (2008) which captures partial membership and allows each vertex to have a distribution over a set of classes, SBM assumes that each vertex of a network belongs to a single class. The identifiability of the parameters in SBM was studied by Allman et al. (2009; 2010), who showed that the model is generically identifiable up to a permutation of the classes. In other words, except in a set of parameters which has a null Lebesgue's measure, two parameters imply the same random graph model if and only if they differ only by the ordering of the classes. Many inference strategies as well as a model selection criterion have been proposed for SBM. They will be described in Chapter 2.

SBM is the starting point of this thesis. A new Bayesian inference procedure is proposed for this model in Chapter 2. It is then extended to allow overlapping clusters in Chapters 3 and 4.

1.4 Phase transition in stochastic block models

Contrary to all the previous sections of this chapter where we described existing work, we now present some new properties that we found which bring some insights into the SBM model. The goal is to show that, for a specific choice of connectivity matrix, the model can be seen as an instance of the inhomogeneous random graph model. We then use the results of

Bollobás et al. (2005) to characterize the critical point of phase transition in SBM, where a giant component appears. This work was done in collaboration with C. Matias.

Let us start by considering a SBM model with Q classes and a kernel[2] $\kappa(x, y)$ on a finite metric space $S = \{1, \dots, Q\}$. We denote $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_Q)$ the vector of class proportions and introduce a probability measure μ on S such that $\mu(q) = \alpha_q, \forall q \in S$. Moreover, let us define a $Q \times Q$ matrix $\mathbf{\Pi}$ of connection probabilities which depend on the number N of vertices:

$$\pi_{ql} = \frac{\kappa(q, l)}{N}, \forall (q, l) \in S \times S.$$

If an undirected graph without self loop is considered, the expected number of edges $e(N, \kappa)$ is given by:

$$\begin{aligned}
\mathrm{E}[e(N,\kappa)] &= \mathrm{E}[\sum_{i<j} X_{ij}] \\
&= \sum_{i<j} \mathrm{E}[X_{ij}] \\
&= \sum_{i<j} p(X_{ij} = 1 | \, \boldsymbol{\alpha}, \mathbf{\Pi}) \\
&= \sum_{i<j} \sum_{\mathbf{Z}_i} \sum_{\mathbf{Z}_j} p(X_{ij} = 1 | \, \mathbf{Z}_i, \mathbf{Z}_j, \mathbf{\Pi}) p(\mathbf{Z}_i \, | \, \boldsymbol{\alpha}) p(\mathbf{Z}_j \, | \, \boldsymbol{\alpha}) \\
&= \sum_{i<j} \sum_{q,l} \alpha_q \alpha_l \pi_{ql} \\
&= \frac{N(N-1)}{2} \sum_{q,l} \alpha_q \alpha_l \frac{\kappa(q,l)}{N} \\
&= \frac{N-1}{2} \sum_{q,l} \alpha_q \alpha_l \kappa(q,l).
\end{aligned} \tag{1.52}$$

Thus, $\mathrm{E}[e(N, \kappa)]$ is linear in N. Obviously, this is also the case if the network contains self loops and/or it is directed. Following Bollobás et al. (2005), the model considered is an inhomogeneous random graph model if the kernel κ is graphical, that is:

$$\sum_{q=1}^{Q} \sum_{l=1}^{Q} \alpha_q \alpha_k |\kappa(q, l)| < \infty,$$

and

$$\lim_{N \to \infty} \frac{1}{N} \mathrm{E}[e(N, \kappa)] = \frac{1}{2} \sum_{q,l}^{Q} \alpha_q \alpha_l \kappa(q, l).$$

The kernel is bounded and using (1.52), these two properties are verified.

1.4.1 The phase transition

In order to study the phase transition, also called percolation transition, we use the operator introduced by Bollobás:

$$(T_\kappa \, \mathbf{v})(q) = \sum_{l=1}^{Q} \kappa(q, l) v(l) \alpha_l, \forall q. \tag{1.53}$$

[2]Here we define a kernel as a symmetric non-negative function

Note that both the input $\mathbf{v}$ and the output $T_\kappa \mathbf{v}$ of the operator are Q dimensional vectors. Consider now the norms given by:

$$|| \mathbf{u} ||_2 = \left(\sum_{q=1}^{Q} \alpha_q u_q^2 \right)^{\frac{1}{2}},$$

for any Q dimensional vector $\mathbf{u}$ and

$$||T_\kappa|| = sup\{||T_\kappa \mathbf{v}||_2 : v_q \geq 0, \forall q, ||\mathbf{v}||_2 \leq 1\} < \infty.$$

The kernel κ is said to be *subcritical* if $||T_\kappa|| < 1$, *critical* if $||T_\kappa|| = 1$, and *supercritical* if $||T_\kappa|| > 1$. If $||T_\kappa|| < 1$, the size of the biggest connected component is $C_1(G) = O(\log N)$, whereas $C_1(G) = O(N)$ if $||T_\kappa|| > 1$. In our case, we aim at characterizing the connectivity matrix $\mathbf{\Pi}$ for which $||T_\kappa|| = 1$. Using matrix notations, (1.53) can be written:

$$T_\kappa \mathbf{v} = \mathbf{K} \operatorname{diag}(\alpha_1, \ldots, \alpha_Q) \mathbf{v},$$

where $\mathbf{K}$ is a $Q \times Q$ matrix such that $(\mathbf{K})_{ql} = \kappa(q, l)$ and $\operatorname{diag}(\alpha_1, \ldots, \alpha_Q)$ is a diagonal matrix with diagonal equal to the vector $\boldsymbol{\alpha}$. If the vector $\mathbf{v}$ is now decomposed on the basis $\{e_1, \ldots, e_Q\}$ of eigenvectors of $\mathbf{K} \operatorname{diag}(\alpha_1, \ldots, \alpha_Q)$, we obtain:

$$\mathbf{v} = \sum_{q=1}^{Q} v_q e_q,$$

and

$$T_\kappa \mathbf{v} = \mathbf{K} \operatorname{diag}(\alpha_1, \ldots, \alpha_Q) \sum_{q=1}^{Q} v_q e_q = \sum_{q=1}^{Q} v_k \lambda_q e_q,$$

where $\{\lambda_1, \ldots, \lambda_Q\}$ are the corresponding eigenvalues of $\{e_1, \ldots, e_Q\}$. Thus

$$||T_\kappa \mathbf{v}||_2^2 = \sum_{q=1}^{Q} \alpha_q v_q^2 \lambda_q^2 \leq (\max_q \lambda_q^2) \sum_{q=1}^{Q} \alpha_q v_q^2 = (\max_q \lambda_q^2) ||\mathbf{v}||_2^2.$$

Subject to the constraint $||\mathbf{v}||_2 \leq 1$, the maximum $\max_q |\lambda_q|$ of $||T_\kappa \mathbf{v}||_2$ is reached when $||\mathbf{v}||_2 = 1$. Therefore, $||T_\kappa|| = \max_q |\lambda_q|$. In other words, given the matrices $\mathbf{K}$ and $\operatorname{diag}(\alpha_1, \ldots, \alpha_q)$, a giant component appears in the graph, that is $C_1 = O(N)$, if $\max_q |\lambda_q| \geq 1$.

1.4.2 Experiments

In this section, some experiments are carried out to verify the properties found previously. A SBM model is considered with a connectivity matrix $\mathbf{\Pi}$ such that $\pi_{ql} = \frac{\kappa(q,l)}{N}$ and

$$\mathbf{K} = \begin{pmatrix} a & b & \ldots & b \\ b & a & & \vdots \\ \vdots & & \ddots & b \\ b & \ldots & b & a \end{pmatrix},$$

to limit the number of free parameters.

We start by fixing the number of classes $Q = 2$, the corresponding proportions $\{\alpha_1 = 0.6, \alpha_2 = 0.4\}$, and the number of vertices $N = 5000$. We then generate some graphs by changing the values of a and b. The parameter b is set to $a/8$ and a varies from 0.5 to 50. For each pair (a, b), both the proportion N^* of vertices in the biggest connected component and the highest eigenvalue $\max_q |\lambda_q|$ of $\mathbf{K}\,\mathrm{diag}(\alpha_1, \ldots, \alpha_Q)$ are computed. In order to obtain smoother results, each experiment is repeated 30 times and the resulting proportions N^* are averaged. The results are presented in Figure 1.8.

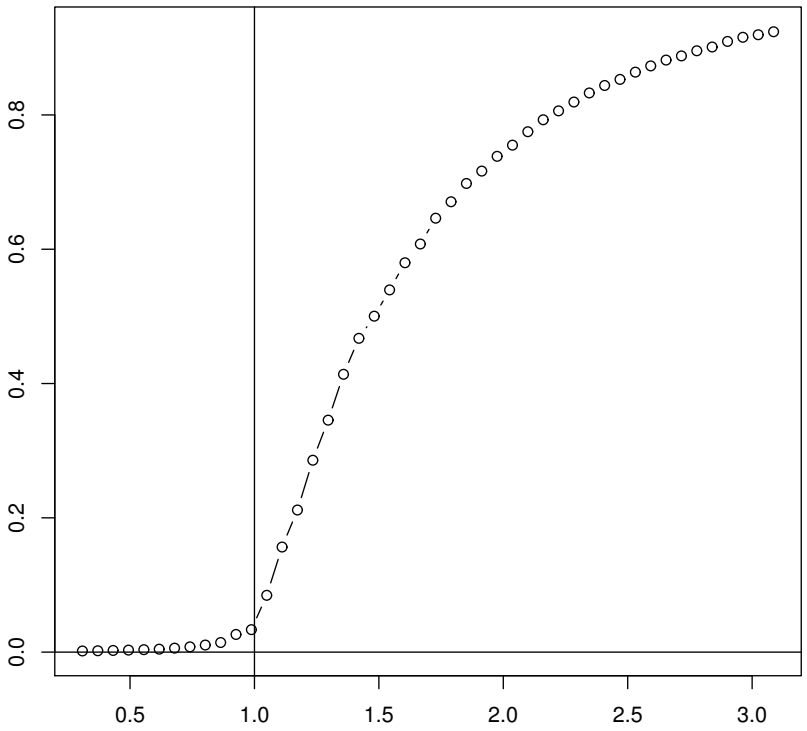

Figure 1.8 – *Several graphs with 5000 vertices are generated using SBM with various connectivity matrices. The x axis represents the values of* $\max_q |\lambda_q|$ *while the proportions* N^* *of vertices in the biggest connected component are given in the y axis. The critical point of phase transition occurs when* $\max_q |\lambda_q| \geq 1$.

We verify that the phase transition occurs when the highest eigenvalue $\max_k |\lambda_k|$ is equal to 1. Indeed, when $\max_q |\lambda_q| < 1$, the proportion N^* of vertices in the biggest connected component is close to zero whereas it converges to 1 when $\max_q |\lambda_q| \geq 1$.

Conclusion

In this chapter, we reviewed mixture models as well as inference techniques such as the EM algorithm and the variational EM algorithm. We also focused on some model selection criteria to estimate the number of components from the data. Finally, we described some of the most widely graph clustering algorithms and concentrated mainly on model based approaches.

2 Variational Bayesian inference and complexity control for stochastic block models

Contents

The clustering of vertices as well as the estimation of the Stochastic Block Model (SBM) parameters have been subject to previous work and numerous inference strategies such as variational Expectation Maximization (EM) and classification EM have been proposed. However, SBM still suffers from a lack of criteria to estimate the number of components in the mixture. To our knowledge, only one model based criterion, ICL, has been derived for SBM in the literature. It relies on an asymptotic approximation of the Integrated Complete-data Likelihood and recent studies have shown that it tends to be too conservative in the case of small networks. To tackle this issue, we propose a new criterion that we call ILvb, based on a non asymptotic approximation of the marginal likelihood. We describe how the criterion can be computed through a variational Bayes EM algorithm.

2.1 Introduction

Many methods have been proposed in the literature to jointly estimate SBM model parameters and cluster the vertices of a network. They all face the same difficulty. Indeed, contrary to many mixture models, the conditional distribution of all the latent variables $\mathbf{Z}$ and model parameters, given the observed data $\mathbf{X}$, can not be factorized due to conditional dependency (for more details, see Daudin et al. 2008). Therefore, optimization techniques such as the EM algorithm can not be used directly. In the case of SBM, Nowicki and Snijders (2001) proposed a Bayesian probabilistic approach. They introduced some prior Dirichlet distributions for the model parameters and used Gibbs sampling to approximate the posterior distribution over the model parameters and posterior predictive distribution. Their algorithm is implemented in the software BLOCKS, which is part of the package StoCNET (Boer et al. 2006). It gives accurate a posteriori estimates but can not handle networks with more than 200 vertices. Daudin et al. (2008) proposed a frequentist variational EM approach for SBM which can handle much larger networks. Online strategies have also been developed (Zanghi et al. 2008).

While many inference strategies have been proposed for estimation and clustering purpose, SBM still suffers from a lack of criteria to estimate the number of classes in networks. Indeed, many criteria, such as the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC) (Burnham and Anderson 2004) are based on the likelihood $p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi})$ of the observed data $\mathbf{X}$, which is intractable here. To tackle this issue, Mariadassou et al. (2010) and Daudin et al. (2008) used a criterion, so-called ICL, based on an asymptotic approximation of the integrated *complete-data* likelihood. This criterion relies on the joint distribution $p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi})$ rather than $p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi})$ and can be easily computed, even in the case of SBM. ICL was originally proposed by Biernacki et al. (2000) for model selection in Gaussian mixture models, and is known to be particularly suitable for cluster analysis view since it favors well separated clusters. However, because it relies on an asymptotic approximation, Biernacki et al. (2010) showed, in the case of mixtures of multivariate multinomial distributions, that it may fail to detect interesting structures present in the data, for small sample sizes. Mariadassou et al. (2010) obtained similar results when analyzing networks generated using SBM. They found that this asymptotic criterion tends to underestimate the number of classes when dealing with small networks. We emphasize that, to our knowledge, ICL is currently the only *model based criterion* developed for SBM.

Our main concern in this chapter is to propose a new criterion for SBM, based on the marginal likelihood $p(\mathbf{X})$, also called integrated *observed-data* likelihood. The marginal likelihood is known to focus on density estimation view and is expected to provide a consistent estimation of the distribution of the data. For a more detailed overview of the differences between integrated *complete-data* likelihood and integrated *observed-data* likelihood, we refer to Biernacki et al. (2010). In the case of SBM, the marginal likelihood is not tractable and we describe in this chapter how a non asymptotic approximation can be obtained through a variational Bayes EM algorithm.

In Section 2.2, we describe SBM and we introduce some non informative conjugate prior distributions for the model parameters. The variational Bayes EM algorithm is then presented in Section 2.3. We show in Section 2.4 how it naturally leads to a new model selection criterion that we call ILvb. Finally, in Section 2.5, we carry out some experiments using simulated data sets and the metabolic network of *Escherichia coli*, to assess ILvb.
The R package "mixer" implementing this work is available from the following web site: `http://cran.r-project.org/web/packages/mixer`

2.2 A mixture model for graphs

The data we model consists of a $N \times N$ binary matrix $\mathbf{X}$, with entries X_{ij} describing the presence or absence of an edge from vertex i to vertex j. Both directed and undirected relations can be analyzed but in the following, we focus on undirected relations. Therefore $\mathbf{X}$ is symmetric.

2.2.1 Model and notations

As mentioned in Section 1.3.3, the Stochastic Block Model (SBM) introduced by Nowicki and Snijders (2001) associates to each vertex of a network a latent variable $\mathbf{Z}_i$ drawn from a multinomial distribution, such that $Z_{iq} = 1$ if vertex i belongs to class q:

$$\mathbf{Z}_i \sim \mathcal{M}\Big(1, \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_Q)\Big).$$

The vector $\boldsymbol{\alpha}$ denotes the vector of class proportions. The edges are then drawn from a Bernoulli distribution:

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

where $\boldsymbol{\Pi}$ is a $Q \times Q$ matrix of connection probabilities. According to this model, the latent variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$ are iid and given this latent structure, all the edges are supposed to be independent.

Thus, when considering an undirected graph without self loops, this leads to:

$$p(\mathbf{Z} \mid \boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}},$$

and

$$\begin{aligned} p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\Pi}) &= \prod_{i<j} p(X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \boldsymbol{\Pi}) \\ &= \prod_{i<j} \prod_{q,l} \mathcal{B}(X_{ij} | \pi_{ql})^{Z_{iq} Z_{jl}} \\ &= \prod_{i<j} \prod_{q,l} \left(\pi_{ql}^{X_{ij}} (1 - \pi_{ql})^{1 - X_{ij}} \right)^{Z_{iq} Z_{jl}}. \end{aligned}$$

In the case of a directed graph, the products over $i < j$ must be replaced by products over $i \neq j$. The edges X_{ii} must also be taken into account if the graph contains self-loops.

2.2.2 A Bayesian Stochastic Block Model

SBM can be described in a full Bayesian framework where it can be considered as a generalisation of the affiliation model proposed by Hofman and Wiggins (2008). Indeed, the Bayesian model of Hofman and Wiggins (2008) considers a simple structure where vertices of the same class connect with probability λ and with probability ϵ otherwise (see Section 1.3.3). Therefore, it can be seen as a constrained SBM where the diagonal of $\boldsymbol{\Pi}$ is set to λ and all the other elements to ϵ.

To extend the SBM frequentist model, we first specify some non informative conjugate priors for the model parameters. Since $p(\mathbf{Z}_i \mid \boldsymbol{\alpha})$ is a multinomial distribution, we consider a Dirichlet distribution for the mixing coefficients:

$$p\Big(\boldsymbol{\alpha} \mid \mathbf{n}^0 = \{n_1^0, \ldots, n_Q^0\} \Big) = \mathrm{Dir}(\boldsymbol{\alpha}; \mathbf{n}^0),$$

where $n_q^0 = 1/2$, $\forall q$. This Dirichlet distribution corresponds to a non-informative Jeffreys prior distribution which is known to be proper (Jeffreys 1946). It is also possible to consider a uniform distribution on the $Q - 1$ dimensional simplex by fixing $n_q^0 = 1, \forall q$.

Since $p(X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \boldsymbol{\Pi})$ is a Bernoulli distribution, we use independent Beta priors to model the connectivity matrix $\boldsymbol{\Pi}$:

$$p\Big(\boldsymbol{\Pi} \mid \boldsymbol{\eta}^0 = (\eta_{ql}^0), \boldsymbol{\zeta}^0 = (\zeta_{ql}^0) \Big) = \prod_{q \leq l} \mathrm{Beta}(\pi_{ql}; \eta_{ql}^0, \zeta_{ql}^0),$$

with $\eta_{ql}^0 = \zeta_{ql}^0 = 1/2, \forall q$. This corresponds to a product of non-informative Jeffreys prior distributions. Note that if the graph is directed, the products over $q \leq l$, must be replaced by products over q, l since $\boldsymbol{\Pi}$ is no longer symmetric.

Thus, the model parameters are now seen as random variables (see Figure 2.1) whose distributions depend on the hyperparameters $\mathbf{n}^0$, $\boldsymbol{\eta}^0$, and $\boldsymbol{\zeta}^0$. In the following, since these hyperparameters are fixed and in order to keep the notations simple, they will not be shown explicitly in the conditional distributions.

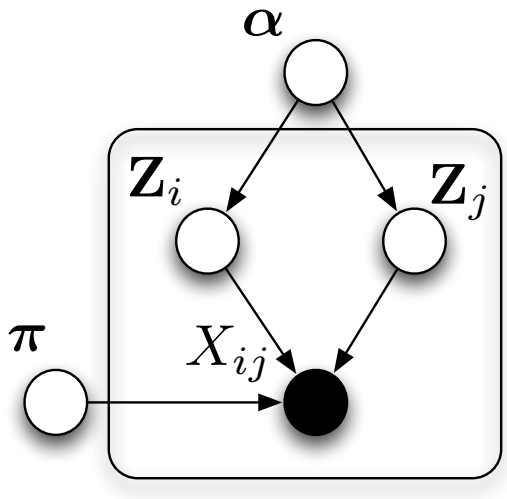

Figure 2.1 – *Directed acyclic graph representing the Bayesian view of SBM. Nodes represent random variables, which are shaded when they are observed and edges represent conditional dependencies.*

2.3 Estimation

In this section, we first describe the variational EM algorithm used by Daudin et al. (2008) to jointly estimate SBM model parameters and cluster the vertices of a network. We then propose a new variational Bayes EM algorithm for SBM which approximates the full posterior distribution of the model parameters and latent variables, given the observed data $\mathbf{X}$. This procedure relies on a lower bound which will be later used, in Section 2.4, as a non asymptotic approximation of the marginal log-likelihood $\log p(\mathbf{X})$.

2.3.1 Variational EM

The likelihood $p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi})$ of the observed data $\mathbf{X}$ can be obtained through the marginalization $p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi})$. This summation involves Q^N terms and quickly becomes intractable. To tackle such problem, the well known EM algorithm (see Section 1.1.3) has been applied with success on a large variety of mixture models. As shown in Section 1.2.1, this two stage estimation approach (Hathaway 1986, Neal and Hinton 1998) can be described in a variational inference framework. Unfortunately, EM relies on the distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$ which is not tractable in the case of SBM and therefore variational approximations are required.

Thus, given a distribution $q(\mathbf{Z})$ over the latent variables, the log-likelihood of the observed data is decomposed into two terms:

$$\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi}) = \mathcal{L}_{ML}(q; \boldsymbol{\alpha}, \boldsymbol{\Pi}) + \mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})\right), \tag{2.1}$$

where

$$\mathcal{L}_{ML}(q; \boldsymbol{\alpha}, \boldsymbol{\Pi}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \boldsymbol{\Pi})}{q(\mathbf{Z})} \right\}, \tag{2.2}$$

and

$$\mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \,|\, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})\right) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log\left\{\frac{p(\mathbf{Z} \,|\, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})}{q(\mathbf{Z})}\right\}. \tag{2.3}$$

In (2.1) and (2.3), KL denotes the Kullback-Leibler divergence between the distribution $q(\mathbf{Z})$ and the distribution $p(\mathbf{Z} \,|\, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$. It can be easily verified that minimizing (2.3) with respect to $q(\mathbf{Z})$ is equivalent to maximizing the lower bound (2.2) of (2.1) with respect to $q(\mathbf{Z})$. To obtain a tractable algorithm, Daudin et al. (2008) assumed that the distribution $q(\mathbf{Z})$ can be factorized such that:

$$q(\mathbf{Z}) = \prod_{i=1}^{N} q(\mathbf{Z}_i) = \prod_{i=1}^{N} \mathcal{M}(\mathbf{Z}_i;\, 1, \boldsymbol{\tau}_i),$$

where τ_{iq} is a variational parameter denoting the probability of node i to belong to class q. This gives rise to a so-called variational EM procedure. During the variational E-step, the model parameters are fixed and, by maximizing (2.2) with respect to $q(\mathbf{Z})$, the algorithm looks for an approximation of the conditional distribution of the latent variables. Conversely, during the variational M-step, the approximation $q(\mathbf{Z})$ is fixed and the lower bound is maximized with respect to the model parameters. This procedure is repeated until convergence and was proposed by Daudin et al. (2008) for SBM.

2.3.2 Variational Bayes EM

In the context of mixture models, the conditional distribution $p(\mathbf{Z} \,|\, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$ can generally be computed and therefore Bayesian inference strategies focus on estimating the posterior distribution $p(\boldsymbol{\alpha}, \boldsymbol{\Pi} \,|\, \mathbf{X})$. The distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi} \,|\, \mathbf{X})$ is then simply given by a byproduct. However, when considering SBM, the distribution $p(\mathbf{Z} \,|\, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$ is intractable and so we propose to approximate the full distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi} \,|\, \mathbf{X})$. We follow the work of Attias (1999), Corduneanu and Bishop (2001), Svensén and Bishop (2004) on Bayesian mixture modelling and Bayesian model selection. Thus, the marginal log-likelihood, also called integrated *observed-data* log-likelihood, can be decomposed into two terms:

$$\log p(\mathbf{X}) = \mathcal{L}\left(q(\cdot)\right) + \mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \,|\, \mathbf{X})\right), \tag{2.4}$$

where

$$\mathcal{L}(q) = \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi}) \log\left\{\frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})}\right\} d\,\boldsymbol{\alpha}\, d\,\boldsymbol{\Pi}, \tag{2.5}$$

and

$$\begin{aligned}&\mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \,|\, \mathbf{X})\right)\\ &\qquad = -\sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi}) \log\left\{\frac{p(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi} \,|\, \mathbf{X})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})}\right\} d\,\boldsymbol{\alpha}\, d\,\boldsymbol{\Pi}.\end{aligned} \tag{2.6}$$

Again, as for the variational EM approach (Section 2.3.1), minimizing (2.6) with respect to $q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$ is equivalent to maximizing the lower bound

(2.5) of (2.4) with respect to $q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$. However, we now have a full variational optimization problem since the model parameters are random variables and we are looking for an approximation $q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$ of $p(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi} \,|\, \mathbf{X})$. To obtain a tractable algorithm, we assume that the distribution $q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})$ can be factorized such that:

$$q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi}) = q(\boldsymbol{\alpha}) q(\boldsymbol{\Pi}) q(\mathbf{Z}) = q(\boldsymbol{\alpha}) q(\boldsymbol{\Pi}) \prod_{i=1}^{N} q(\mathbf{Z}_i).$$

In the following, we use a variational Bayes EM algorithm (see Section 1.2.2). We call variational Bayes E-step, the optimization of each distribution $q(\mathbf{Z}_i)$ and variational Bayes M-step, the approximations of the remaining distributions $q(\boldsymbol{\alpha})$ and $q(\boldsymbol{\Pi})$. All the optimization equations, the lower bound, as well as proofs are given in the appendix.

We first initialize a matrix $\boldsymbol{\tau}^{old}$ with a hierarchical algorithm based on the classical Ward distance. The distance between vertices which is considered is simply the Euclidean distance $d(i,j) = \sum_{k=1}^{N} (X_{ik} - X_{jk})^2$ which takes the number of discordances between i and j into account. Given a number of classes Q, each vertex is assigned (hard assignment) to its nearest group. Second, the algorithm uses (B.4) and (B.6) to estimate the variational distributions over the model parameters $\boldsymbol{\alpha}$ as well as $\boldsymbol{\Pi}$. Finally, the variational distribution over the latent variables is estimated using (B.1). The algorithm cycles though the E and M steps until the absolute distance between two successive values of the lower bound (B.8) is smaller than a threshold *eps*. In the experiment section, we set $eps = 1e-6$. In practice, smaller values slow the convergence of the algorithm and do not lead to better estimates.

The computational costs of the frequentist approach of Daudin et al. (2008) and our variational Bayes algorithm are both equal to $\mathrm{O}(Q^2 N^2)$. Analyzing a sparse network takes about a second for $N = 200$ nodes and about a minute for $N = 1000$.

2.4 Model selection

So far, we have seen that the variational Bayes EM algorithm leads to an approximation of the posterior distribution of all the model parameters and latent variables, given the observed data. However, the problem of estimating the number Q of classes in the mixture has not been addressed yet. Given a set of values of Q, we aim at selecting Q^* which maximizes the marginal log-likelihood $\log p(\mathbf{X} \,|\, Q)$, also called integrated *observed-data* log-likelihood. The marginal likelihood is known to focus on density estimation view and is expected to provide a consistent estimation of the distribution of the data (Biernacki et al. 2010). Unfortunately, this quantity is not tractable, since for each value of Q, it involves integrating over all the model parameters and latent variables:

$$\log p(\mathbf{X} \,|\, Q) = \log \left\{ \sum_{\mathbf{Z}} \int \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi} \,|\, Q) \, d\boldsymbol{\alpha} \, d\boldsymbol{\Pi} \right\}.$$

To tackle this issue, we propose to replace the marginal log-likelihood with its variational Bayes approximation. Thus, given a value of Q, the algorithm introduced in Section 2.3.2 is used to maximize the lower bound

(2.5) with respect to $q(.)$. We recall that this maximization implies a minimization of the KL divergence (2.6) between $q(.)$ and the unknown posterior distribution. After convergence of the algorithm, according to (2.4), if the KL divergence is small, then the lower bound $\mathcal{L}(q(.))$ approximates the marginal log-likelihood. Obviously, this assumption can not be verified in practice since (2.6) can not be computed analytically. Moreover, we emphasize that there is no solid reason to believe that the KL divergence is close to zero and does not depend on the model complexity. Nevertheless, in order to obtain a tractable model selection criterion we rely on this approximation. After convergence of the algorithm, the lower bound takes a simple form and leads to a new criterion for SBM that we call ILvb

$$IL_{vb} = \log\left\{\frac{\Gamma(\sum_{q=1}^{Q} n_q^0)\prod_{q=1}^{Q}\Gamma(n_q)}{\Gamma(\sum_{q=1}^{Q} n_q)\prod_{q=1}^{Q}\Gamma(n_q^0)}\right\} + \sum_{q\leq l}^{Q}\log\left\{\frac{\Gamma(\eta_{ql}^0+\zeta_{ql}^0)\Gamma(\eta_{ql})\Gamma(\zeta_{ql})}{\Gamma(\eta_{ql}+\zeta_{ql})\Gamma(\eta_{ql}^0)\Gamma(\zeta_{ql}^0)}\right\} - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\log\tau_{iq},$$

where τ_{iq} is the estimated probability of vertex i to belong to class q and $(n_q)_q$, $(\eta_{ql})_{ql}$, $(\zeta_{ql})_{ql}$ are parameters given in the appendix. The gamma function is denoted by $\Gamma(\cdot)$. Contrary to the criterion proposed by Daudin et al. (2008), ILvb does not rely on an asymptotic approximation, sometimes called BIC-like approximation. In practice, given a network, the variational Bayes EM algorithm is run for the different values of Q considered and Q^* is chosen such that ILvb is maximized.

2.5 Experiments

We present some results of the experiments we carried out to assess the criterion we proposed in Section 2.4. Throughout our experiments, we chose to compare our approach to the work of Daudin et al. (2008) and Hofman and Wiggins (2008). Indeed, contrary to many other *model based techniques*, the corresponding algorithms can analyze networks with hundred of nodes in a reasonable amount of time (a few minutes on a dual core). We recall that Daudin et al. (2008) proposed a frequentist maximum likelihood approach (see Section 2.3.1) for SBM as well as an ICL criterion. On the other hand, Hofman and Wiggins (2008) presented a model for community structure detection and a Bayesian criterion that we will denote VBMOD. Thus, by using both synthetic data and the metabolic network of bacteria *Escherichia coli*, our aim is twofold. First, we illustrate the overall capacity of SBM to retrieve interesting structures in a large variety of networks. Second, we concentrate on comparing the two criteria ICL and ILvb developed for SBM.

2.5.1 Comparison of the criteria

In these experiments, we consider two types of networks. In Section 2.5.1, we generate affiliation networks, made of community structures, using the generative model of Hofman and Wiggins (2008). Therefore, vertices

of the same class connect with probability λ and with probability ϵ otherwise. This corresponds to a constrained SBM where the diagonal of the connectivity matrix is set to λ and all the other elements to ϵ:

$$\mathbf{\Pi} = \begin{pmatrix} \lambda & \epsilon & \dots & \epsilon \\ \epsilon & \lambda & & \vdots \\ \vdots & & \ddots & \epsilon \\ \epsilon & \dots & \epsilon & \lambda \end{pmatrix}.$$

In Section 2.5.1, we then draw networks with more complex topologies, made of both community structures and a class of hubs. The corresponding model is given by the connectivity matrix:

$$\mathbf{\Pi} = \begin{pmatrix} \lambda & \epsilon & \dots & \epsilon & \lambda \\ \epsilon & \lambda & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \lambda & \dots & \dots & \dots & \lambda \end{pmatrix},$$

where hubs connect with probability λ to any vertices in the network.

Following Mariadassou et al. (2010) who showed that ICL tends to underestimate the number of classes in the case of small graphs, we consider networks with only $N = 50$ vertices to analyze the robustness of our criterion. We set $(\lambda = 0.9, \epsilon = 0.1)$ and for each value of Q_{True} in the set $\{3, \dots, 7\}$, we then generate 100 networks with classes mixed in the same proportions $\alpha_1 = \dots = \alpha_{Q_{True}} = 1/Q_{True}$.

In order to estimate the number of classes in the latent structures, we applied the methods of Hofman and Wiggins (2008), Daudin et al. (2008), and our algorithm (Section 2.3.2) on each network, for various numbers of classes $Q \in \{1, \dots, 7\}$. Note that, we choose $n_q^0 = 1/2$, $\forall q \in \{1, \dots, Q\}$ for the Dirichlet prior and $\eta_{ql}^0 = \zeta_{ql}^0 = 1/2$, $\forall (q, l) \in \{1, \dots, Q\}^2$ for the Beta priors. We recall that such distributions correspond to non informative prior distributions. Like any optimization technique, the clustering methods we consider depend on the initialization. Thus, for each simulated network and each number of classes Q, we use five different initializations of $\boldsymbol{\tau}$. Finally, we select the best learnt models for which the corresponding criteria VBMOD, ICL, or ILvb were maximized.

Before comparing ICL and ILvb, it is crucial to recall that these two criteria were not conceived for the same purpose. ICL approximates the integrated *complete-data* likelihood and is known to focus on cluster analysis view since it favors well separated clusters. It realizes a compromise between the estimation of the data density and the evidence of data partitioning. Conversely, ILvb approximates the marginal likelihood which is known to focus on density estimation only. In the following experiments, since networks are generated using SBM, and because we evaluate the criteria through their capacity to retrieve the true number of classes, ILvb is expected to lead to better results. However, in other situations (which are not considered in this chapter), where the focus would be on the clustering of vertices, ICL might be of possible interest.

Affiliation networks

In Table 2.1, we observe that VBMOD outperforms both ICL and ILvb. For instance, when $Q_{True} = 5$, VBMOD correctly estimates the number of classes of the 100 generated networks, while ICL and ILvb have respectively a percentage of accuracy of 77 and 99. These differences increase when $Q_{True} = 6$ and $Q_{True} = 7$. Indeed, the higher Q_{True} is, the less vertices the classes contain, and therefore, the more difficult it is to retrieve and distinguish the community structures. Thus, when $Q_{True} = 7$, each class only contains on average $Q_{True}/N \approx 7.1$ vertices. VBMOD appears to be a very stable criterion for community structure detection. It has a percentage of accuracy of 84 while ICL never estimates the true number of classes.

All the affiliation networks were generated using the model of Hofman and Wiggins (2008) which explains the results of VBMOD presented above. Indeed, the corresponding model for community structure detection only estimates the parameters λ and ϵ whereas the frequentist and Bayesian approaches for SBM look for a full $Q \times Q$ matrix $\mathbf{\Pi}$ of connection probabilities. They are capable of handling networks with complex topologies, as shown in the following section, but they might miss some structures if the number of vertices is too limited.

We observe that ILvb leads to a better estimates of the true number of classes in networks than ICL. Thus, when $Q_{True} = 5$ and $Q_{True} = 6$, ILvb estimates correctly the number of classes of 99 and 73 networks while ICL has respectively a percentage of accuracy of 77 and 12.

	2	3	4	5	6	7
3	0	100	0	0	0	0
4	0	0	100	0	0	0
5	0	0	0	**100**	0	0
6	0	0	0	0	**97**	3
7	0	0	0	2	14	**84**

(a) $Q_{True} \backslash Q_{VBMOD}$

	2	3	4	5	6	7
3	0	100	0	0	0	0
4	0	0	100	0	0	0
5	0	0	23	**77**	0	0
6	0	1	28	59	**12**	0
7	0	8	49	42	1	**0**

(b) $Q_{True} \backslash Q_{ICL}$

	2	3	4	5	6	7
3	0	100	0	0	0	0
4	0	0	100	0	0	0
5	0	0	0	**99**	1	0
6	0	0	4	23	**73**	0
7	0	2	14	44	27	**13**

(c) $Q_{True} \backslash Q_{ILvb}$

Table 2.1 – *Confusion matrices for VBMOD, ICL and ILvb.* $\lambda = 0.9$, $\epsilon = 0.1$ *and* $Q_{True} \in \{3, \ldots, 7\}$. *Affiliation networks.*

Networks with community structures and hubs

Table 2.2 displays the results of the experiments on networks exhibiting community structures and hubs. The presence of hubs is a central property of so-called real real networks (Albert and Barabási 2002).

This slightly more complex and more realistic situation does heavily perturb the estimation of VBMOD. Most of the time, VBMOD fails to detect the class of hub and henceforth underestimates the number of classes. For example, when $Q_{True} = 3$ or $Q_{True} = 4$, VBMOD always misses a class. When the number of true classes grows over four, VBMOD's behaviour becomes more variable but keep the same heavy tendency to underestimate.

In this context, ICL and ILvb behaves more consistently than VBMOD. When Q_{True} is less or equal than four both strategies are comparable. But when the number of true classes increases, the performance of ICL dramatically deteriorates, whereas ILvb remains more stable.

In the context of small graph, when the focus is on the estimation of the data density, ILvb clearly provides a more reliable estimation of the number of classes than ICL. It also shows better performances that VBMOD when networks are made of classes with more complex topologies than communities.

	2	3	4	5	6	7
3	95	**0**	3	0	0	2
4	1	95	**4**	0	0	0
5	0	0	94	**6**	0	0
6	0	0	1	83	**16**	0
7	0	0	2	15	78	**5**

(a) $Q_{True} \backslash Q_{VBMOD}$

	2	3	4	5	6	7
3	0	**100**	0	0	0	0
4	0	0	**100**	0	0	0
5	0	0	12	**88**	0	0
6	0	0	19	59	**22**	0
7	0	3	29	56	12	**0**

(b) $Q_{True} \backslash Q_{ICL}$

	2	3	4	5	6	7
3	0	**100**	0	0	0	0
4	0	0	**100**	0	0	0
5	0	0	2	**98**	0	0
6	0	0	1	29	**70**	0
7	0	0	3	34	45	**18**

(c) $Q_{True} \backslash Q_{ILvb}$

Table 2.2 – *Confusion matrices for VBMOD, ICL and ILvb.* $\lambda = 0.9$, $\epsilon = 0.1$ *and* $Q_{True} \in \{3, \ldots, 7\}$. *Affiliation networks and a class of hubs.*

2.5.2 The metabolic network of *Escherichia coli*

We apply the methodology described in this chapter to the metabolic network of bacteria *Escherichia coli* (Lacroix et al. 2006) which was analyzed by Daudin et al. (2008) using SBM. In this network, there are 605 vertices which represent chemical reactions and a total number of 1782 edges. Two reactions are connected if a compound produced by the first one is a part of the second one (or vice-versa). As in the previous section, we consider non informative priors: we fixed $n_q^0 = 1/2$, $\forall q \in \{1, \ldots, Q\}$ for the Dirichlet prior and $\eta_{ql}^0 = \zeta_{ql}^0 = 1/2$, $\forall (q, l) \in \{1, \ldots, Q\}^2$ for the Beta priors.

Thus, for $Q \in \{1, \ldots, 40\}$, we apply the methods of Hofman and Wiggins (2008) as well as our approach on this network. We compute the corresponding criteria and we repeat such procedure 60 times, for different initializations of $\boldsymbol{\tau}$. Indeed, to speed up the initialization, we first run a kmeans algorithm with 40 classes and random initial centers. We then use the corresponding partitions as inputs of the hierarchical algorithm described in Section 2.3.2. The results for ILvb are presented as boxplots in Figure 2.2. The criterion finds its maximum for $Q_{ILvb} = 22$ classes, while Daudin et al. (2008) found $Q_{ICL} = 21$. Thus, for this particular large data set, both ILvb and ICL lead to almost the same estimates of the number of latent classes.

We also compared the learnt partitions in the Bayesian and in the frequentist approach. Figure 2.3 is a dot plot representation of the metabolic network after having applied the Bayesian algorithm for $Q_{VB} = 22$. Each

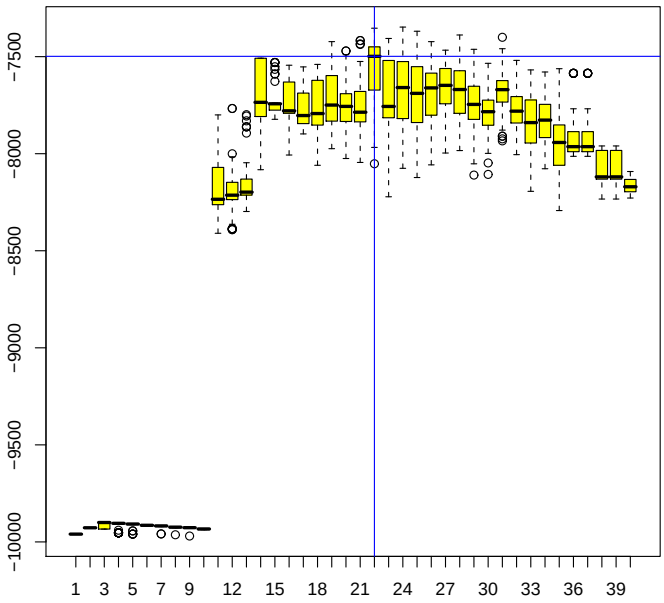

Figure 2.2 – *Boxplot representation (over 60 experiments) of ILvb for* $Q \in \{1, \ldots, 40\}$. *The maximum is reached at* $Q_{IL_{vb}} = 22$.

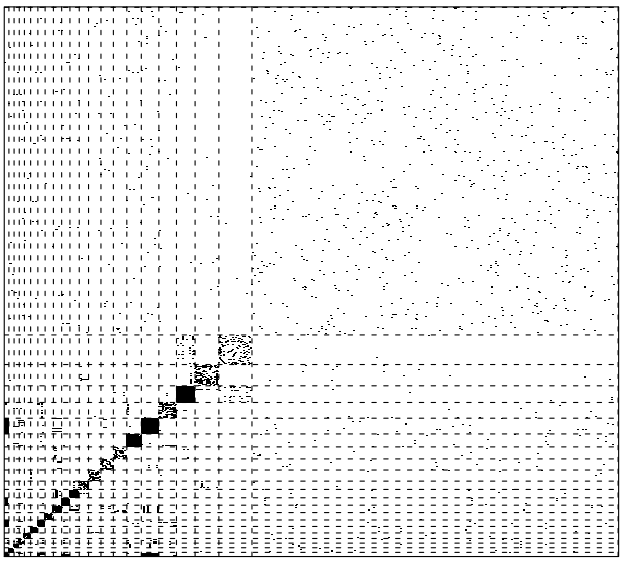

Figure 2.3 – *Dot plot representation of the metabolic network after classification of the vertices into $Q_{VB} = 22$ classes. The x-axis and y-axis correspond to the list of vertices in the network, from 1 to 605. Edges between pairs of vertices are represented by shaded dots.*

vertex i is classified into the class for which τ_{iq} is maximal (Maximum A Posteriori estimate). We observed very similar patterns in the frequentist approach. Among the classes, eight of them are cliques $\pi_{qq} = 1$ and six have within probability connectivity greater than 0.5. As shown by Daudin et al. (2008), these cliques or pseudo-cliques gather reactions involving a same compound. Thus, chorismate, pyruvate, L-aspartate, L-glutamate, D-glyceraldehyde-3-phosphate and ATP are all responsible for cliques. Moreover, as observed in Daudin et al. (2008), since the connection probability between class 1 and 17 is 1, they correspond to a single clique which is associated to pyruvate. However that clique is split into two sub-cliques because of their different connectivities with reactions of classes 7 and 10. The approach of Hofman and Wiggins (2008) can not retrieve such complex topologies, as shown in Section 2.5.1, and many classes such as class 1 and 17 were merged. We found $Q_{VBMOD} = 14$.

Conclusion

In this chapter, we showed how the Stochastic Block Model (SBM) could be described in a full Bayesian framework. We introduced some non informative conjugate priors over the model parameters and we described a variational Bayes EM algorithm which approximates the posterior distribution of all the latent variables and model parameters, given the observed data. Using this framework, we derived a non asymptotic model selection criterion, so-called ILvb, which approximates the marginal likelihood. By considering networks generated using SBM, we showed that ILvb focus on the estimation of the data density and provides a relevant estimation of the number of latent classes. We also illustrated the capacity of SBM to retrieve interesting structures in a large variety of networks.

Overlapping stochastic block models

3

Contents

Given a network, almost all graph clustering algorithms partition the vertices into *disjoint* clusters, according to their connection profile. However, recent studies have shown that these techniques were too restrictive and that most of the existing networks contained overlapping clusters. To tackle this issue, we present in this chapter the Overlapping Stochastic Block Model. Our approach allows the vertices to belong to multiple clusters, and, to some extent, generalizes the well known Stochastic Block Model (Nowicki and Snijders 2001). We show that the model is generically identifiable within classes of equivalence and we propose an approximate inference procedure, based on global and local variational techniques. Using toy data sets as well as the French Political Blogosphere network and the transcriptional network of *Saccharomyces cerevisiae*, we compare our work with other approaches.

3.1 Introduction

A drawback of existing graph clustering techniques is that they all partition the vertices into disjoint clusters, while lots of objects in real world applications typically belong to multiple groups or communities. For instance, many proteins, so-called *moonlighting proteins*, are known to have several functions in the cells (Jeffery 1999), and actors might belong to several groups of interests (Palla et al. 2005). Thus, a graph clustering method should be able to uncover overlapping clusters. This issue has received growing attention in the last few years, starting with an algorithmic approach based on small complete sub-graphs developed by Palla et al. (2005) and implemented in the software CFinder (Palla et al. 2006). They defined a k-clique community as a union of all k-cliques (complete sub-graphs of size k) that can be reached from each other through a series of adjacent[1] k-cliques. Given a network, their algorithm first locates all cliques and then identifies the communities using a clique-clique overlap matrix (Everett and Borgatti 1998). By construction, the resulting communities can overlap. In order to select the optimal value of k, the authors suggested a global criterion which looks for a community structure as highly connected as possible. Small values of k leads to a giant community which smears the details of a network by merging small communities. Conversely, when k increases, the communities tend to become smaller, more disintegrated, but also more cohesive. Therefore, they proposed a heuristic which consists in running their algorithm for various values of k and then to select the lowest value such that no giant community appears.

More recent work (Airoldi et al. 2008) proposed the Mixed Membership Stochastic Block model (MMSB) which has been used with success to analyze networks in many applications (Airoldi et al. 2007; 2006). They used variational techniques to estimate the model parameters and proposed a criterion to select the number of classes. As detailed in Heller et al. (2008), mixed membership models, as Latent Dirichlet Allocation (Blei et al. 2003), are flexible models which can capture partial membership (Griffiths and Ghahramani 2005, Heller and Ghahramani 2007), in the form of attribute-specific mixtures. In MMSB, a mixing weight vector $\boldsymbol{\pi}_i$ is drawn from a Dirichlet distribution for each vertex in the network, π_{iq} being the probability of vertex i to belong to class q. The edge probability from vertex i to vertex j is then given by $p_{ij} = \mathbf{Z}_{i\to j}^{\mathsf{T}} \mathbf{B} \mathbf{Z}_{i\leftarrow j}$, where $\mathbf{B}$ is a $Q \times Q$ matrix of connection probabilities similar to the $\boldsymbol{\Pi}$ matrix in SBM. The vector $\mathbf{Z}_{i\to j}$ is sampled from a multinomial distribution $\mathcal{M}(1, \boldsymbol{\pi}_i)$ and describes the class membership of vertex i in its relation towards vertex j. By symmetry, the vector $\mathbf{Z}_{i\leftarrow j}$ is drawn from a multinomial distribution $\mathcal{M}(1, \boldsymbol{\pi}_j)$ and represents the class membership of vertex j in its relation towards vertex i. Thus, depending on its relations with other vertices, each vertex can belong to different classes and therefore MMSB can be viewed as allowing overlapping clusters. However, the limit of MMSB is that it does not produce edges which are themselves influenced by the fact that some vertices belong to multiple clusters. Indeed, for every pair (i, j) of vertices, only a single draw of $\mathbf{Z}_{i\to j}$ and $\mathbf{Z}_{i\leftarrow j}$ determines the probability p_{ij} of an edge, all the other class memberships of vertex i and j towards other

[1] Two k-cliques are adjacent if they share $k-1$ vertices

vertices in the network do not play a part. In this chapter, we present a complementary approach which tackles this issue.

In Fu and Banerjee (2008), Fu and Banerjee model overlapping clusters on Q components by characterizing each individual i by a latent $\{0,1\}$ vector $\mathbf{Z}_i$ of length Q drawn from independent Bernoulli distributions. The i^{th} row of the data matrix then only depends on $\mathbf{Z}_i$. In the underlying clustering structure, i belongs to the components corresponding to a 1 in $\mathbf{Z}_i$. Nevertheless, the proposed model needs Q parameters for each individual and supposes independence between rows and columns of the data matrix, which is not the case when looking for network structures.

In this chapter, we propose a new model for generating networks, depending on $(Q+1)^2+Q$ parameters, where Q is the number of components in the mixture. A latent $\{0,1\}$-vector of length Q is assigned to each vertex, drawn from products of Bernoulli distributions whose parameters are not vertex-dependent. Each vertex may then belong to several components, allowing overlapping clusters, and each edge probability depends only on the components of its endpoints.

In Section 3.2, we recall the two constraints that the stochastic block model satisfies. In Section 3.3, we present the overlapping stochastic block model and we show in Section 3.4 that the model is identifiable within classes of equivalence. In Section 3.5, we propose an EM-like algorithm to infer the parameters of the model. Finally, in Section 3.6, we compare our work with other approaches using simulated data and two real networks. We show the efficiency of our model to detect overlapping clusters in networks.

3.2 The stochastic block model

In this chapter, we consider a directed binary random graph $\mathcal{G}$ represented by an $N \times N$ binary adjacency matrix $\mathbf{X}$. Each entry X_{ij} describes the presence or absence of an edge from vertex i to vertex j. We assume that $\mathcal{G}$ does not have any self loop, and therefore, the variables X_{ii} will not be taken into account. The Stochastic Block Model (SBM) associates to each vertex of a network a latent variable $\mathbf{Z}_i$ drawn from a multinomial distribution:

$$\mathbf{Z}_i \sim \mathcal{M}\Big(1,\, \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_Q)\Big),$$

where $\boldsymbol{\alpha}$ denotes the vector of class proportions. As in other standard mixture models, the vector $\mathbf{Z}_i$ sees all its components set to zero except one such that $Z_{iq} = 1$ if vertex i belongs to class q. The model then verifies:

$$\sum_{q=1}^{Q} Z_{iq} = 1,\ \forall i \in \{1, \ldots, N\}, \tag{3.1}$$

and

$$\sum_{q=1}^{Q} \alpha_q = 1. \tag{3.2}$$

3.3 The overlapping stochastic block model

In order to allow each vertex to belong to multiple classes, we relax the constraints (3.1) and (3.2). Thus, for each vertex i of the network, we introduce a latent vector $\mathbf{Z}_i$, of Q independent Boolean variables $Z_{iq} \in \{0,1\}$, drawn from a multivariate Bernoulli distribution:

$$\mathbf{Z}_i \sim \prod_{q=1}^{Q} \mathcal{B}(Z_{iq};\, \alpha_q) = \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} (1-\alpha_q)^{1-Z_{iq}}. \tag{3.3}$$

We point out that $\mathbf{Z}_i$ can also have all its components set to zero which is a useful feature in practice as described in Sections 3.3.2 and 3.6. The edge probabilities are then given by:

$$X_{ij} |\, \mathbf{Z}_i, \mathbf{Z}_j \sim \mathcal{B}\big(X_{ij};\, g(a_{\mathbf{Z}_i,\mathbf{Z}_j})\big) = e^{X_{ij} a_{\mathbf{Z}_i,\mathbf{Z}_j}} g(-a_{\mathbf{Z}_i,\mathbf{Z}_j}),$$

where

$$a_{\mathbf{Z}_i,\mathbf{Z}_j} = \mathbf{Z}_i^\intercal \, \mathbf{W} \, \mathbf{Z}_j + \mathbf{Z}_i^\intercal \, \mathbf{U} + \mathbf{V}^\intercal \, \mathbf{Z}_j + W^*, \tag{3.4}$$

and $g(x) = (1+e^{-x})^{-1}$ is the logistic sigmoid function. $\mathbf{W}$ is a $Q \times Q$ real matrix whereas $\mathbf{U}$ and $\mathbf{V}$ are Q-dimensional real vectors. The first term in the right-hand side of (3.4) describes the interactions between the vertices i and j. If i belongs only to class q and j only to class l, then only one interaction term remains ($\mathbf{Z}_i^\intercal \, \mathbf{W} \, \mathbf{Z}_j = W_{ql}$). However, as illustrated in table 3.1, the model can take more complex interactions into account if one or both of these two vertices belong to multiple classes (Figure 3.1). Note that the second term in (3.4) does not depend on $\mathbf{Z}_j$. It models the overall capacity of vertex i to connect to other vertices. By symmetry, the third term represents the global tendency of vertex j to receive an edge. These two parameters $\mathbf{U}$ and $\mathbf{V}$ are related to the sender/receiver effects δ_i and γ_j in the Latent Cluster Random Effects Model (LCREM) of Krivitsky et al. (2009). However, contrary to LCREM, $\delta_i = \mathbf{Z}_i^\intercal \, \mathbf{U}$ and $\gamma_j = \mathbf{V}^\intercal \, \mathbf{Z}_j$ depend on the classes. In other words, two different vertices sharing the same classes, will have exactly the same sender/receiver effects, which is not the case in LCREM. Finally, we use the scalar W^* as a bias, to model sparsity.

	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
$(0,0)$	W^*	$V_1 + W^*$	$V_2 + W^*$	$V_1 + V_2 + W^*$
$(1,0)$	$U_1 + W^*$	$W_{11} + U_1 + V_1 + W^*$	$W_{12} + U_1 + V_2 + W^*$	$W_{11} + W_{12} + U_1 + V_1 + V_2 + W^*$
$(0,1)$	$U_2 + W^*$	$W_{21} + U_2 + V_1 + W^*$	$W_{22} + U_2 + V_2 + W^*$	$W_{21} + W_{22} + U_2 + V_1 + V_2 + W^*$
$(1,1)$	$U_1 + U_2 + W^*$	$W_{11} + W_{21} + U_1 + U_2 + V_1 + W^*$	$W_{12} + W_{22} + U_1 + U_2 + V_2 + W^*$	$W_{11} + W_{12} + W_{21} + W_{22} + U_1 + U_2 + V_1 + V_2 + W^*$

Table 3.1 – *The values of $a_{\mathbf{Z}_i,\mathbf{Z}_j}$ in functions of $\mathbf{Z}_i$ (rows) and $\mathbf{Z}_j$ (columns) for an overlapping stochastic block model with $Q = 2$.*

If we associate to each latent variable $\mathbf{Z}_i$ a vector $\tilde{\mathbf{Z}}_i = \big(\mathbf{Z}_i, 1\big)^\intercal$, then (3.4) can be written:

$$a_{\mathbf{Z}_i,\mathbf{Z}_j} = \tilde{\mathbf{Z}}_i^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j, \tag{3.5}$$

where

$$\tilde{\mathbf{W}} = \begin{pmatrix} \mathbf{W} & \mathbf{U} \\ \mathbf{V}^\intercal & W^* \end{pmatrix}.$$

The $\tilde{Z}_{i(Q+1)}$s can be seen as random variables drawn from a Bernoulli distribution with probability $\alpha_{Q+1} = 1$. Thus, one way to think about the model is to consider that all the vertices in the graph belong to a $(Q+1)$-th cluster which is overlapped by all the other clusters. In the following, we will use (3.5) to simplify the notations.

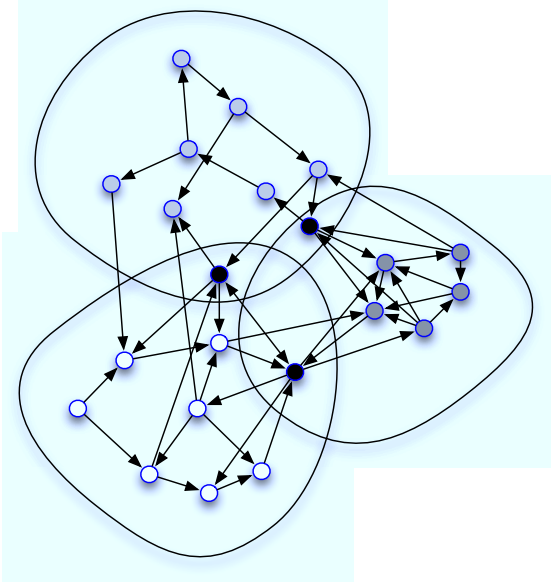

Figure 3.1 – *Example of a directed graph with three overlapping clusters.*

Finally, given the latent structure $\mathbf{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$, all the edges are supposed to be independent. Thus, when considering directed graphs without self-loop, the Overlapping Stochastic Block Model (OSBM) is defined through the following distributions:

$$p(\mathbf{Z} \mid \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} (1 - \alpha_q)^{1 - Z_{iq}}, \tag{3.6}$$

and

$$p(\mathbf{X} \mid \mathbf{Z}, \tilde{\mathbf{W}}) = \prod_{i \neq j}^{N} e^{X_{ij} a_{\mathbf{Z}_i, \mathbf{Z}_j}} g(-a_{\mathbf{Z}_i, \mathbf{Z}_j}).$$

The graphical model of OSBM is given in Figure 3.2.

3.3.1 Modeling sparsity

As explained in Airoldi et al. (2008), real networks are often sparse[2] and it is crucial to distinguish the two sources of non-interaction. Sparsity might be the result of the rarity of interactions in general but it might also indicate that some class (*intra* or *inter*) connection probabilities are close to zero. For instance, social networks (see Section 3.6.2) are often made of communities where vertices are mostly connected to vertices of the same

[2] the corresponding adjacency matrices contain mainly zeros

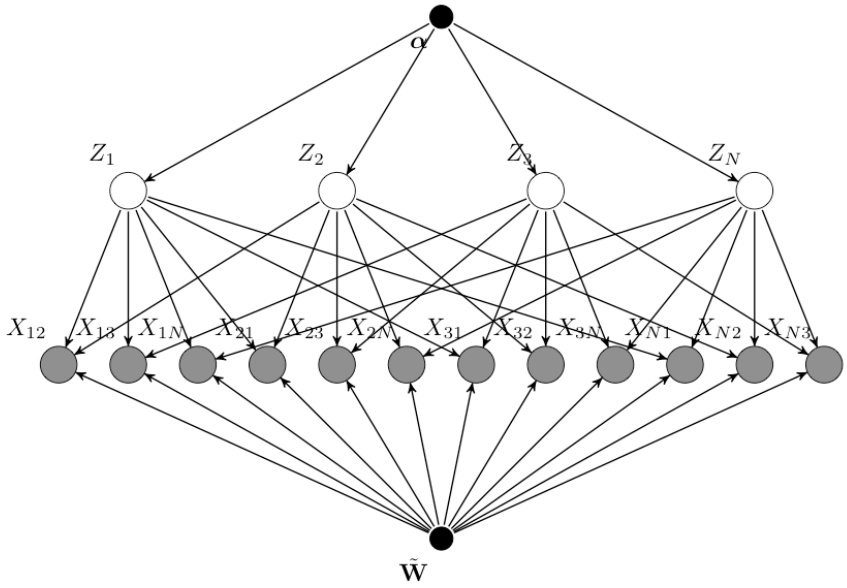

Figure 3.2 – *Directed acyclic graph representing the frequentist view of the overlapping stochastic block model. Nodes represent random variables, which are shaded when they are observed and edges represent conditional dependencies.*

community. This corresponds to classes with high *intra* connection probabilities and low *inter* connection probabilities. In (3.4), we can notice that W^* appears in $a_{\mathbf{Z}_i,\mathbf{Z}_j}$ for every pair of vertices. Therefore, W^* is a convenient parameter to model the two sources of sparsity. Indeed, low values of W^* result from the rarity of interactions in general, whereas high values signify that sparsity comes from the classes (parameters in $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{V}$).

3.3.2 Modeling outliers

When applied on real networks, graph clustering methods often lead to giant classes of vertices having low output and input degrees (Daudin et al. 2008, Latouche et al. 2009). These classes are usually discarded and the analysis of networks focus on more highly structured classes to extract useful information. The product of Bernoulli distributions (3.6) provides a natural way to encode these "outliers". Indeed, rather than using giant classes, OSBM uses the null component such that $\mathbf{Z}_i = \mathbf{0}$ if vertex i is an outlier and should not be classified in any class.

3.4 Identifiability

Before looking for an optimization procedure to estimate the model parameters, given a sample of observations (a network), it is crucial to verify whether OSBM is identifiable. A theorem of Allman et al. (2009) lies at the core of the results presented in this section.

If we denote, $\mathcal{F}(\Theta) = \{\mathbb{P}_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta\}$, a family of models we are interested in, the classical definition of identifiability requires that for any two

different values $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$, the corresponding probability distributions $\mathbb{P}_{\boldsymbol{\theta}}$ and $\mathbb{P}_{\boldsymbol{\theta}'}$ are different.

3.4.1 Correspondence with (non overlapping) stochastic block models

Let Θ_{OSBM} be the parameter space of the family of OSBMs with Q classes:

$$\Theta_{OSBM} = \{(\boldsymbol{\alpha}, \tilde{\mathbf{W}}) \in [0,1]^Q \times \mathbb{R}^{(Q+1)^2}\}.$$

Each $\boldsymbol{\theta}$ in Θ_{OSBM} corresponds to a random graph model which is defined by the distribution $p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})$. The aim of this section is to characterize whether there exists any relation between two different parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ in Θ_{OSBM}, leading to the same random graph model.

We consider the (non overlapping) Stochastic Block Model (SBM) introduced by Nowicki and Snijders (2001). The model is defined by a set of classes $\mathcal{C}$, a vector of class proportions $\boldsymbol{\gamma} = \{\gamma_{\mathbf{C}}\}_{\mathbf{C}\in\mathcal{C}}$ verifying $\sum_{\mathbf{C}\in\mathcal{C}} \gamma_{\mathbf{C}} = 1$, and a matrix of connection probabilities $\boldsymbol{\Pi} = \left(\Pi_{\mathbf{C},\mathbf{D}}\right)_{\mathbf{C},\mathbf{D}\in\mathcal{C}^2}$. Note that they are an infinite number of ways to represent and encode the classes. For simplicity, a common choice is to set $\mathcal{C} = \{1, \ldots, Q\}$ and possibly $\mathcal{C} = \left\{\mathbf{C} \in \{0,1\}^Q, \sum_{q=1}^Q C_q = 1\right\}$, for a model with Q classes. The random graphs are drawn as follows. First, the class of each vertex is sampled from a multinomial distribution with parameters $(1, \boldsymbol{\gamma})$. Thus, each vertex i belongs only to one class, and that class is $\mathbf{C}$ with probability $\gamma_{\mathbf{C}}$. Second, the edges are drawn independently from each other from Bernoulli distributions, the probability of an edge (i, j) being $\Pi_{\mathbf{C},\mathbf{D}}$, if i belongs to class $\mathbf{C}$ and j to class $\mathbf{D}$.

Let Θ_{SBM} be the parameter space of the family of SBMs with 2^Q classes:

$$\Theta_{SBM} = \{(\boldsymbol{\gamma}, \boldsymbol{\Pi}) \in [0,1]^{2^Q} \times [0,1]^{2^{2Q}}, \sum_{\mathbf{C}\in\mathcal{C}} \gamma_{\mathbf{C}} = 1\}.$$

Considering that each possible value of the vectors $\mathbf{Z}_i$s in an OSBM with Q classes encodes a class in a SBM with 2^Q classes (*i.e.* $\mathcal{C} = \{0,1\}^Q$), yields a natural function:

$$\phi : \begin{array}{ccc} \Theta_{OSBM} & \to & \Theta_{SBM} \\ (\boldsymbol{\alpha}, \tilde{\mathbf{W}}) & \to & (\boldsymbol{\gamma}, \boldsymbol{\Pi}) \end{array},$$

where

$$\gamma_{\mathbf{C}} = \prod_{q=1}^{Q} \alpha_q^{C_q} (1-\alpha_q)^{1-C_q}, \; \forall \mathbf{C} \in \{0,1\}^Q,$$

and

$$\Pi_{\mathbf{C},\mathbf{D}} = g(\mathbf{C}^\intercal \mathbf{W} \mathbf{D} + \mathbf{C}^\intercal \mathbf{U} + \mathbf{V}^\intercal \mathbf{D} + W^*), \; \forall (\mathbf{C}, \mathbf{D}) \in \{0,1\}^Q \times \{0,1\}^Q.$$

Let $\mathcal{G}_N$ denote the set of probability measures on the graphs of N vertices. The OSBM of parameter $\boldsymbol{\theta}$ in Θ_{OSBM} and the SBM of parameter $\phi(\boldsymbol{\theta})$ in Θ_{SBM} clearly induce the same measure μ in $\mathcal{G}_N$. Thus, denoting by $\psi(\boldsymbol{\gamma}, \boldsymbol{\Pi})$ the probability measure in $\mathcal{G}_N$ induced by the SBM of parameter $(\boldsymbol{\gamma}, \boldsymbol{\Pi})$, the problem of identifiability is to characterize the relations between parameters $\boldsymbol{\theta} \in \Theta_{OSBM}$ and $\boldsymbol{\theta}' \in \Theta_{OSBM}$ such that $\psi(\phi(\boldsymbol{\theta})) = \psi(\phi(\boldsymbol{\theta}'))$.

$$\begin{array}{ccccc} \Theta_{OSBM} & \rightarrow & \Theta_{SBM} & \rightarrow & \mathcal{G}_N \\ \boldsymbol{\theta} = (\boldsymbol{\alpha}, \tilde{\mathbf{W}}) & \stackrel{\phi}{\rightarrow} & (\boldsymbol{\gamma}, \boldsymbol{\Pi}) & \stackrel{\psi}{\rightarrow} & \mu \end{array}.$$

The identifiability of SBM was studied by Allman et al. (2009), who showed that the model is generically identifiable up to a permutation of the classes. In other words, except in a set of parameters which has a null Lebesgue's measure, two parameters imply the same random graph model if and only if they differ only by the ordering of the classes. Therefore, the main theorem of Allman et al. (2009) implies the following result:

Théorème 3.1 *There exists a set $\Theta_{SBM}^{bad} \subset \Theta_{SBM}$ of null Lebesgue's measure such that, for every $(\boldsymbol{\gamma}, \boldsymbol{\Pi})$ and $(\boldsymbol{\gamma}', \boldsymbol{\Pi}')$ not in Θ_{SBM}^{bad}, $\psi(\boldsymbol{\gamma}, \boldsymbol{\Pi}) = \psi(\boldsymbol{\gamma}', \boldsymbol{\Pi}')$ if and only if there exists a function P_ν such that $(\boldsymbol{\gamma}', \boldsymbol{\Pi}') = P_\nu\big((\boldsymbol{\gamma}, \boldsymbol{\Pi})\big)$, where:*

- *ν is a permutation on $\{0,1\}^Q$,*
- *$\boldsymbol{\gamma}'_{\mathbf{C}} = \boldsymbol{\gamma}_{\nu(\mathbf{C})}$, $\forall\, \mathbf{C} \in \{0,1\}^Q$,*
- *$\boldsymbol{\Pi}'_{\mathbf{C},\mathbf{D}} = \Pi_{\nu(\mathbf{C}),\nu(\mathbf{D})}$, $\forall (\mathbf{C}, \mathbf{D}) \in \{0,1\}^Q \times \{0,1\}^Q$.*

Thus, studying the generical identifiability of the OSBM is equivalent to characterizing the parameters of Θ_{OSBM} verifying $\phi(\boldsymbol{\theta}') = P_\nu(\phi(\boldsymbol{\theta}))$ for some permutation ν on $\{0,1\}^Q$.

3.4.2 Permutations and inversions

As in the case of the SBM, reordering the Q classes of the OSBM and doing the corresponding modification in $\boldsymbol{\alpha}$ and $\tilde{\mathbf{W}}$ does not change the generative random graph model. Indeed, let σ be a permutation on $\{1, \ldots, Q\}$ and let P_σ denote the function corresponding to the permutation σ of the classes. Then, $(\boldsymbol{\alpha}', \tilde{\mathbf{W}}') = P_\sigma(\boldsymbol{\alpha}, \tilde{\mathbf{W}})$ is defined by:

$$\alpha'_q = \alpha_{\sigma(q)},\ \forall q \in \{1, \ldots, Q\},$$

and

$$\tilde{\mathbf{W}}'_{q,l} = \tilde{\mathbf{W}}_{\sigma(q),\sigma(l)},\ \forall (q,l) \in \{1, \ldots, Q+1\}^2.$$

Now, let ν be the permutation of $\{0,1\}^Q$ defined by:

$$\nu(\mathbf{C}) = (C_{\sigma(1)}, \ldots, C_{\sigma(Q)}),\ \forall\, \mathbf{C} \in \{0,1\}^Q.$$

It is then straightforward to see that, for every parameter $\boldsymbol{\theta}$ in Θ_{OSBM} and every permutation σ, $\phi(P_\sigma(\boldsymbol{\theta})) = P_\nu(\phi(\boldsymbol{\theta}))$, where P_ν is defined in Theorem 3.1.

There is another family of operations in Θ_{OSBM} which does not change the generative random graph model, which we call inversions. They correspond to exchanging the labels 0 to 1 and vice versa on some of the coordinates of the Z_i's. To give an intuition, consider a parameter $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \tilde{\mathbf{W}})$ in Θ_{OSBM}. Let us generate graphs under the probability measure in $\mathcal{G}_N$ induced by $\boldsymbol{\theta}$ and consider only the first coordinate of the Z_i's. If we denote by "cluster 1" the vertices whose Z_i's have a 1 as first coordinate, the graph sampling procedure consists in sampling the set "cluster 1" and

then drawing the edges conditionally on that information. Note that it would be equivalent to sample the vertices which are not in "cluster 1" and to draw the edges conditionally on that information. Thus there exists an equivalent reparametrization where the 1's in the first coordinate correspond to the vertices which are not in "cluster 1". This is the parameter $\boldsymbol{\theta}'$ obtained from $\boldsymbol{\theta}$ by an inversion of the first coordinate.

Let $\mathbf{A}$ be any vector of $\{0,1\}^Q$. We define the $\mathbf{A}$-*inversion* $I_{\mathbf{A}}$ as follows:

$$I_{\mathbf{A}} : \begin{array}{ccc} \Theta_{OSBM} & \to & \Theta_{OSBM} \\ (\boldsymbol{\alpha}, \tilde{\mathbf{W}}) & \to & (\boldsymbol{\alpha}', \tilde{\mathbf{W}}') \end{array},$$

where

$$\alpha'_j = \begin{cases} 1-\alpha_j & \text{if } A_j = 1 \\ \alpha_j & \text{otherwise} \end{cases}, \forall j \in \{1,\dots,Q\},$$

and

$$\tilde{\mathbf{W}}' = \mathbf{M}_{\mathbf{A}}^{\intercal}\, \tilde{\mathbf{W}}\, \mathbf{M}_{\mathbf{A}}\,.$$

The matrix $\mathbf{M}_{\mathbf{A}}$ is defined by:

$$\mathbf{M}_{\mathbf{A}} = \begin{pmatrix} I - 2\mathrm{diag}(\mathbf{A}) & \mathbf{A} \\ 0\dots 0 & 1 \end{pmatrix},$$

with $\mathrm{diag}(\mathbf{A})$ being the $Q \times Q$ diagonal matrix whose diagonal is the vector $\mathbf{A}$.

Proposition 3.1 *For every $\mathbf{A} \in \{0,1\}^Q$, let ν be the permutation of $\{0,1\}^Q$ defined by:*

$$\forall\, \mathbf{C} \in \{0,1\}^Q, \nu(\mathbf{C})_i = \begin{cases} 1-C_i & \textit{if } A_i = 1 \\ C_i & \textit{otherwise} \end{cases}.$$

Then, for every $\boldsymbol{\theta}$ in Θ_{OSBM}:

$$\phi(I_{\mathbf{A}}(\boldsymbol{\theta})) = P_\nu(\phi(\boldsymbol{\theta})),$$

where P_ν is defined in theorem 3.1.

Proof. Consider $\boldsymbol{\theta} \in \Theta_{OSBM}$ and $\mathbf{A} \in \{0,1\}^Q$ and define $(\boldsymbol{\gamma}, \boldsymbol{\Pi}) = \phi(\boldsymbol{\theta})$ and $(\boldsymbol{\gamma}', \boldsymbol{\Pi}') = \phi(I_{\mathbf{A}}(\boldsymbol{\theta}))$. It is straightforward to verify that:

$$\gamma'_{\mathbf{C}} = \gamma_{\nu(\mathbf{C})},\ \forall\, \mathbf{C} \in \{0,1\}^Q.$$

Moreover, since $M_{\mathbf{A}} \binom{\mathbf{C}}{1} = \binom{\nu(\mathbf{C})}{1}$, it follows that:

$$\begin{aligned} \boldsymbol{\Pi}'_{\mathbf{C},\mathbf{D}} &= g\left(\begin{pmatrix}\mathbf{C}^{\intercal} & 1\end{pmatrix} \mathbf{M}_{\mathbf{A}}^{\intercal}\, \tilde{\mathbf{W}}\, \mathbf{M}_{\mathbf{A}} \binom{\mathbf{D}}{1}\right) \\ &= g\left(\begin{pmatrix}\nu(\mathbf{C})^{\intercal} & 1\end{pmatrix} \tilde{\mathbf{W}} \binom{\nu(\mathbf{D})}{1}\right) \\ &= \boldsymbol{\Pi}_{\nu(\mathbf{C}),\nu(\mathbf{D})}\,. \end{aligned}$$

Therefore, $\phi(I_A(\boldsymbol{\theta})) = P_\nu(\phi(\boldsymbol{\theta}))$.

3.4.3 Identifiability

Let us define the following equivalence relation:

$$\boldsymbol{\theta} \sim \boldsymbol{\theta}' \quad \text{if } \exists \sigma, \mathbf{A} \quad | \quad \boldsymbol{\theta}' = I_{\mathbf{A}}(P_\sigma(\boldsymbol{\theta})).$$

To be convinced that it is an equivalence relation, note that:

$$I_{\mathbf{A}} \circ P_\sigma = P_\sigma \circ I_{\sigma^{-1}(\mathbf{A})}.$$

Consider the set of equivalence classes for the relation $\sim$. It follows that:

- Two parameters in the same equivalence class induce the same measure in $\mathcal{G}_N$,
- Each equivalence class contains a parameter $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \tilde{\mathbf{W}})$ such that $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_Q \leq \frac{1}{2}$. Moreover, if the α_is are all distinct and strictly lower than $\frac{1}{2}$, there is a unique such parameter in the equivalence class.

We are now able to state our main theorem about identifiability, that is that the model is generically identifiable up to the equivalence relation $\sim$:

Théorème 3.2 *For every $\boldsymbol{\alpha} \in]0,1[^Q$, let $\beta \in \mathbb{R}^Q$ be the vector defined by $\beta_k = -\log(\frac{\alpha_k}{1-\alpha_k})$, for every k.*

Define Θ^{bad}_{OSBM} as the set of parameters $(\boldsymbol{\alpha}, \tilde{\mathbf{W}})$ such that one of the following conditions holds:

- *there exists $1 \leq k \leq Q$ such that $\alpha_k = 0$ or $\alpha_k = 1$ or $\alpha_k = \frac{1}{2}$,*
- *there exist $1 \leq k, l \leq Q$ such that $\alpha_k = \alpha_l$,*
- *there exist $\mathbf{C}, \mathbf{D} \in \{0,1\}^Q \times \{0,1\}^Q$ such that $\sum_k \beta_k C_k = \sum_k \beta_k D_k$,*
- *$\phi(\boldsymbol{\alpha}, \tilde{\mathbf{W}}) \in \Theta^{bad}_{SBM}$, set of null measure given by Theorem 3.1.*

Then Θ^{bad}_{OSBM} has a null Lebesgue's measure on Θ_{OSBM} and:

$$\forall\, \boldsymbol{\theta}, \boldsymbol{\theta}' \in (\Theta_{OSBM} \setminus \Theta^{bad}_{OSBM})^2, \quad \phi(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}') \Leftrightarrow \boldsymbol{\theta} \sim \boldsymbol{\theta}'\,.$$

Proof. Θ^{bad}_{OSBM} is the union of a finite number of hyperplanes or spaces which are isomorphic to hyperplanes. Therefore, $\mu(\Theta^{bad}_{OSBM}) = 0$.

Let $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \tilde{\mathbf{W}})$, $\boldsymbol{\theta}' = (\boldsymbol{\alpha}', \tilde{\mathbf{W}}')$, $\phi(\boldsymbol{\theta}) = (\boldsymbol{\gamma}, \boldsymbol{\Pi})$, and $\phi(\boldsymbol{\theta}') = (\boldsymbol{\gamma}', \boldsymbol{\Pi}')$. As ϕ is constant on each equivalence class and as $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are not in Θ^{bad}_{OSBM}, we can assume that $0 < \alpha_1 < \ldots < \alpha_k < \frac{1}{2}$ and $0 < \alpha'_1 < \ldots < \alpha'_k < \frac{1}{2}$. Proving the theorem is then equivalent to prove that $\boldsymbol{\theta} = \boldsymbol{\theta}'$.

As $\phi(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}')$, Theorem 3.1 ensures that there exists a permutation $\nu : \{0,1\}^Q \to \{0,1\}^Q$ such that:

$$\begin{cases} \gamma'_{\mathbf{C}} &= \gamma_{\nu(\mathbf{C})} & \forall\, \mathbf{C} \\ \Pi'_{\mathbf{C},\mathbf{D}} &= \Pi_{\psi(\mathbf{C}),\psi(\mathbf{D})} & \forall\, \mathbf{C}, \mathbf{D} \end{cases}.$$

Then, in particular:

$$\{\prod_k \alpha_k^{C_k}(1-\alpha_k)^{1-C_k}, \mathbf{C} \in \{0,1\}^Q\} = \{\prod_k (\alpha'_k)^{C_k}(1-\alpha'_k)^{1-C_k}, \mathbf{C} \in \{0,1\}^Q\}. \tag{3.7}$$

The minima of those two sets as well as the second lowest values are equal, that is:

$$\prod_k \alpha_k = \prod_k \alpha'_k \quad \text{and} \quad (\prod_{k\leq Q-1} \alpha_k)(1-\alpha_Q) = (\prod_{k\leq Q-1} \alpha'_k)(1-\alpha'_Q).$$

Dividing those equations term by term yields $\frac{\alpha_Q}{1-\alpha_Q} = \frac{\alpha'_Q}{1-\alpha'_Q}$ and finally $\alpha_Q = \alpha'_Q$. Dividing all terms by $\alpha_Q^{C_Q}(1-\alpha_Q)^{1-C_Q}$ in 3.7, by induction it follows that:

$$\alpha = \alpha'. \tag{3.8}$$

Now, for any $\mathbf{C} \in \{0,1\}^Q$, the fact that $\gamma'_{\mathbf{C}} = \gamma_{\nu(\mathbf{C})}$ can be written as:

$$\begin{aligned}
\prod_k \alpha_k^{C_k}(1-\alpha_k)^{1-C_k} &= \prod_k \alpha_k^{\nu(C)_k}(1-\alpha_k)^{1-\nu(C)_k} \\
\sum_k C_k \log(\frac{\alpha_k}{1-\alpha_k}) + \sum_k \log(1-\alpha_k) &= \sum_k \nu(C_k)\log(\frac{\alpha_k}{1-\alpha_k}) + \sum_k \log(1-\alpha_k) \\
\sum_k \beta_k C_k &= \sum_k \beta_k \nu(C)_k.
\end{aligned}$$

Since $\boldsymbol{\theta} \notin \Theta^{bad}_{OSBM}$, this implies that $\nu(\mathbf{C}) = \mathbf{C}$. As it is true for every $\mathbf{C}$, ν is in fact the identity function.

Therefore, for every $\mathbf{C}, \mathbf{D}$, $\Pi_{\mathbf{C},\mathbf{D}} = \Pi'_{\mathbf{C},\mathbf{D}}$, that is

$$\sum_{q,l} w_{ql}c_q d_l + \sum_q u_q c_q + \sum_l v_l d_l + w^* = \sum_{q,l} w'_{ql}c_q d_l + \sum_q u'_q c_q + \sum_l v'_l d_l + w'^*.$$

Applying it for $\mathbf{C} = \mathbf{D} = 0$ implies $W^* = W'^*$.

Applying it for $\mathbf{D} = 0$ and $\mathbf{C} = \boldsymbol{\delta}_q$, where $\boldsymbol{\delta}_q$ is the vector having a 1 on the q^{th} coordinate and 0's elsewhere yields $u_q + W^* = u'_q + W'^*$ and thus $u_q = u'_q$.

By symmetry, $\mathbf{C} = 0$ and $\mathbf{D} = \delta_l$ implies $v_l = v'_l$.

Finally, $\mathbf{C} = \delta_q$ and $\mathbf{D} = \delta_l$ gives $W_{ql} = W'_{ql}$.

Thus

$$\tilde{\mathbf{W}} = \tilde{\mathbf{W}}'. \tag{3.9}$$

By Equations 3.8 and 3.9, we have $\boldsymbol{\theta} = \boldsymbol{\theta}'$.

3.5 Statistical inference

Given a network, our aim in this section is to estimate the OSBM parameters.

The log-likelihood of the observed data set is defined through the marginalization: $p(\mathbf{X} \,|\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \,|\, \alpha, \tilde{\mathbf{W}})$. This summation involves 2^{NQ} terms and quickly becomes intractable. To tackle this issue,

the Expectation-Maximization (EM) algorithm has been applied on many mixture models. However, the E-step requires the calculation of the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ which cannot be factorized in the case of networks (see Daudin et al. 2008, for more details). In order to obtain a tractable procedure, we present some approximations based on global and local variational techniques.

3.5.1 The q-transformation

Given a distribution $q(\mathbf{Z})$, the log-likelihood of the observed data set can be decomposed using the Kullback-Leibler divergence $\mathrm{KL}(\cdot \,||\, \cdot)$:

$$\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) + \mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})\right), \tag{3.10}$$

where

$$\mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})}{q(\mathbf{Z})} \right\}, \tag{3.11}$$

and

$$\mathrm{KL}\left(q(\cdot) \,||\, p(\cdot \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})\right) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})}{q(\mathbf{Z})} \right\}. \tag{3.12}$$

The maximum $\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ of the lower bound $\mathcal{L}_{ML}$ (3.11) is reached when $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$. Thus, if the posterior distribution $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ was tractable, the optimizations of $\mathcal{L}_{ML}$ and $\log p(\mathbf{X} \mid \boldsymbol{\alpha}, \tilde{\mathbf{W}})$, with respect to $\boldsymbol{\alpha}$ and $\tilde{\mathbf{W}}$, would be equivalent. However, in the case of networks, $p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ cannot be calculated and $\mathcal{L}_{ML}$ cannot be optimized over the entire space of $q(\mathbf{Z})$ distributions. Thus, we restrict our search to the class of distributions which satisfy:

$$q(\mathbf{Z}) = \prod_{i=1}^{N} q(\mathbf{Z}_i), \tag{3.13}$$

with

$$\begin{aligned} q(\mathbf{Z}_i) &= \prod_{q=1}^{Q} \mathcal{B}(Z_{iq};\, \tau_{iq}) \\ &= \prod_{q=1}^{Q} \tau_{iq}^{Z_{iq}} (1 - \tau_{iq})^{1 - Z_{iq}}. \end{aligned}$$

Each τ_{iq} is a variational parameter which corresponds to the posterior probability of node i to belong to class q. As for the vector $\boldsymbol{\alpha}$, the vectors $\boldsymbol{\tau}_i = \{\tau_{i1}, \ldots, \tau_{iQ}\}$ are not constrained to lie in the $Q-1$ dimensional simplex.

Proposition 3.2 *(Proof in Appendix C.1) The lower bound of the observed data log-likelihood is*

given by:

$$\begin{aligned}\mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}}) = \sum_{i\neq j}^{N}\Big\{ X_{ij}\tilde{\boldsymbol{\tau}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j + \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})]\Big\} \\ + \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\alpha_q + (1-\tau_{iq})\log(1-\alpha_q)\right\} \\ - \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\tau_{iq} + (1-\tau_{iq})\log(1-\tau_{iq})\right\}.\end{aligned} \tag{3.14}$$

Unfortunately, since the logistic sigmoid function is non linear, $\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{\mathbf{Z}_i,\mathbf{Z}_j})]$ in (3.14) cannot be computed analytically. Thus, we need a second level of approximation to optimize the lower bound of the observed data set.

3.5.2 The ξ-transformation

Proposition 3.3 *(Proof in Appendix C.2) Given a variational parameter ξ_{ij}, $\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{\mathbf{Z}_i,\mathbf{Z}_j})]$ satisfies:*

$$\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})] \geq \log g(\xi_{ij}) - \frac{(\tilde{\boldsymbol{\tau}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j + \xi_{ij})}{2} - \lambda(\xi_{ij})\Big(\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j)^2] - \xi_{ij}^2\Big). \tag{3.15}$$

Eventually, a lower bound of the first lower bound can be computed:

$$\log p(\mathbf{X}\,|\,\boldsymbol{\alpha},\tilde{\mathbf{W}}) \geq \mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}}) \geq \mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}), \tag{3.16}$$

where

$$\begin{aligned}\mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}) = \sum_{i\neq j}^{N}\Bigg\{ \big(X_{ij}-\frac{1}{2}\big)\,\tilde{\boldsymbol{\tau}}_i^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j + \log g(\xi_{ij}) - \frac{\xi_{ij}}{2} \\ - \lambda(\xi_{ij})\left(\mathrm{Tr}\Big(\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\,\boldsymbol{\Sigma}_j\Big) + \tilde{\boldsymbol{\tau}}_j^{\intercal}\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j - \xi_{ij}^2\right)\Bigg\} \\ + \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\alpha_q + (1-\tau_{iq})\log(1-\alpha_q)\right\} \\ - \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\tau_{iq} + (1-\tau_{iq})\log(1-\tau_{iq})\right\}.\end{aligned}$$

The resulting variational EM algorithm (see Algorithm 7) alternatively computes the posterior probabilities $\boldsymbol{\tau}_i$ and the parameters $\boldsymbol{\alpha}$ and $\tilde{\mathbf{W}}$ maximizing

$$\max_{\boldsymbol{\xi}} \mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}).$$

The computational cost of the algorithm is equal to $O(N^2Q^4)$. For comparison the computational cost of the methods proposed by Daudin et al. (2008) and Latouche et al. (2009) for (non-overlapping) SBM is equal to $O(N^2Q^2)$. Analyzing a sparse network with 100 nodes takes about ten seconds on a dual core, and about a minute for dense networks.

Algorithm 7: Overlapping stochastic block model for directed graphs without self loop.

```
// INITIALIZATION
```

Initialize $\boldsymbol{\tau}$ with an Ascendant Hierarchical Classification algorithm
Sample $\tilde{\mathbf{W}}$ from a zero mean σ^2 spherical Gaussian distribution

```
// OPTIMIZATION
```

repeat

```
// ξ-transformation
```

$\xi_{ij} \leftarrow \sqrt{\mathrm{Tr}\left(\tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \boldsymbol{\Sigma}_j\right) + \tilde{\boldsymbol{\tau}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j}, \forall i \neq j$

```
// M-step
```

$\alpha_q \leftarrow \frac{\sum_{i=1}^{N} \tau_{iq}}{N}, \forall q$

Optimize $\mathcal{L}_{ML}\left(q; \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}\right)$ with respect to $\tilde{\mathbf{W}}$, with a gradient based optimization algorithm (e.g. quasi-Newton method of Broyden et al. 1970)

```
// E-step
```

repeat

for *i=1:N* **do**

Optimize $\mathcal{L}_{ML}\left(q; \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}\right)$ with respect to $\boldsymbol{\tau}_i$, with a box constrained ($\tau_{iq} \in [0, 1]$) gradient based optimization algorithm (e.g. Byrd method Byrd et al. 1995)

end

until $\boldsymbol{\tau}$ *converges*

until $\mathcal{L}_{ML}\left(q; \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}\right)$ *converges*

For all the experiments we present in the following section, set $\sigma^2 = 0.5$ and we used the Ascendant Hierarchical Classification algorithm implemented in the R package "mixer" which is available at: `http://cran.r-project.org/web/packages/mixer`.

3.6 Experiments

We present some results of the experiments we carried out to assess OSBM. Throughout our experiments, we compared our approach to SBM (the non-overlapping version of OSBM), the Mixed Membership Stochastic Block model (MMSB) of Airoldi et al. (2008), and the work of Palla et al. (2005), implemented in the software (Version 2.0.1) CFinder (Palla et al. 2006).

In order to perform inference in SBM, we used the variational Bayes algorithm of Latouche et al. (2009) which approximates the posterior distribution over the latent variables and model parameters, given the edges. We computed the Maximum A Posteriori (MAP) estimates and obtained the class membership vectors $\mathbf{Z}_i$. We recall that SBM assumes that each vertex belongs to a single class and therefore each vector $\mathbf{Z}_i$ has all its components set to zero except one, such that $Z_{iq} = 1$ if vertex i is classified into class q. For OSBM, we relied on the variational approximate inference procedure described in Section 3.5 and computed the MAP estimates. Contrary to SBM, each vertex can belong to multiple clusters and therefore the vectors $\mathbf{Z}_i$ can have multiple components set to one. As described in Section 3.1, MMSB can also be viewed as allowing overlapping clusters. For more details, we refer to Airoldi et al. (2008). In order to estimate the MMSB mixing weight vectors $\boldsymbol{\pi}_i$, we used the collapsed Gibbs sampling approach implemented in the R package lda (Chang 2010). We then converted each vector $\boldsymbol{\pi}_i$ into a binary membership vector $\mathbf{Z}_i$ using a threshold t. Thus, for $\pi_{iq} \geq t$, we set $Z_{iq} = 1$ and $Z_{iq} = 0$ otherwise. In all the experiments we carried out, we defined $t = 1/Q$ and we found that for higher values MMSB tended to behave like SBM. Finally, we considered CFinder which is a widely used algorithmic approach to uncover overlapping communities. As described in Section 3.1, CFinder looks for k-clique communities where each k-clique community is a union of all k-cliques (complete sub-graphs of size k) that can be reached from each other through a series of adjacent k-cliques. The algorithm first locates all cliques and then identifies the communities and overlaps between communities using a clique-clique overlap matrix (Everett and Borgatti 1998). Vertices that do not belong to any k-clique are seen as outliers and not classified.

Contrary to OSBM (and CFinder), SBM and MMSB cannot deal with outliers. Therefore, to obtain fair comparisons between the approaches, when OSBM was run with Q classes, SBM and MMSB were run with $Q+1$ classes and we identified the class of outliers. In practice, this can easily been done since this class contains most of the vertices of the network having low output and input degrees.

3.6.1 Simulations

In this set of experiments, we generated two types of networks using the OSBM generative model. In Section 3.6.1, we sampled networks with community structures (Figure 3.3), where vertices of a community are mostly connected to vertices of the same community. To limit the number of free parameters, we considered the $Q \times Q$ real matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} \lambda & -\epsilon & \dots & -\epsilon \\ -\epsilon & \lambda & & \vdots \\ \vdots & & \ddots & -\epsilon \\ -\epsilon & \dots & -\epsilon & \lambda \end{pmatrix}.$$

In Section 3.6.1, we generated networks with more complex topologies, using the matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} \lambda & \lambda & -\epsilon & \dots & \dots & \dots & -\epsilon \\ -\epsilon & -\lambda & -\epsilon & \dots & \dots & \dots & \vdots \\ \vdots & -\epsilon & \lambda & \lambda & -\epsilon & \dots & \vdots \\ \vdots & \vdots & -\epsilon & -\lambda & -\epsilon & \dots & \vdots \\ \vdots & \vdots & \vdots & -\epsilon & \ddots & -\epsilon & -\epsilon \\ \vdots & \vdots & \vdots & \vdots & -\epsilon & \lambda & \lambda \\ -\epsilon & \dots & \dots & \dots & \dots & -\epsilon & -\lambda \end{pmatrix}.$$

In these networks, if class i is a community and has therefore a high *intra* connection probability, then its vertices also highly connect to vertices of class $i+1$ which itself has a low *intra* connection probability. Such star patterns (Figure 3.4) often appear in transcription networks, as shown in Section 3.6.3, and protein-protein interaction networks.

For these two sets of experiments, we used the Q-dimensional real vectors $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{U} = \mathbf{V} = \begin{pmatrix} \epsilon & \dots & \epsilon \end{pmatrix},$$

and we set $Q = 4$, $\lambda = 4$, $\epsilon = 1$, $W^* = -5.5$. Moreover, for the vector $\boldsymbol{\alpha}$ of class probabilities, we set $\alpha_q = 0.25$, $\forall q \in \{1, \dots, Q\}$. We generated 100 networks with $N = 100$ vertices and for each of these networks, we clustered the vertices using CFinder, SBM, MMSB, and OSBM. Finally, we used a criterion similar to the one proposed by Heller and Ghahramani (2007), Heller et al. (2008) to compare the true $\mathbf{Z}$ and the estimated $\hat{\mathbf{Z}}$ clustering matrices. Thus, for each network and each method, we computed the L_2 distance $d(\mathbf{P}, \hat{\mathbf{P}})$ where $\mathbf{P} = \mathbf{Z}\,\mathbf{Z}^\intercal$ and $\hat{\mathbf{P}} = \hat{\mathbf{Z}}\hat{\mathbf{Z}}^\intercal$. These two $N \times N$ matrices are invariant to column permutations of $\mathbf{Z}$ and $\hat{\mathbf{Z}}$ and compute the number of shared clusters between each pair of vertices of a network. Therefore, $d(\mathbf{P}, \hat{\mathbf{P}})$ is a good measure to determine how well the underlying cluster assignment structure has been discovered. Since CFinder depends on a parameter k (size of the cliques), for each simulated network, we ran the software for various values of k and selected $\hat{k}$ for which the L_2 distance was minimized. Note that this choice of k tends to overestimate the performances of CFinder compared to the other approaches. Indeed, in practice,

when analyzing a real network, k needs to be estimated (see Section 3.6.2) while $\mathbf{P}$ is unknown. OSBM was run with Q classes whereas SBM and MMSB were run with $Q+1$ classes. For both SBM and MMSB, and each generated network, after having identified the class of outliers, we set the latent vectors of the corresponding vertices to zero (null component). The L_2 distance $d(\mathbf{P}, \hat{\mathbf{P}})$ was then computed exactly as described previously.

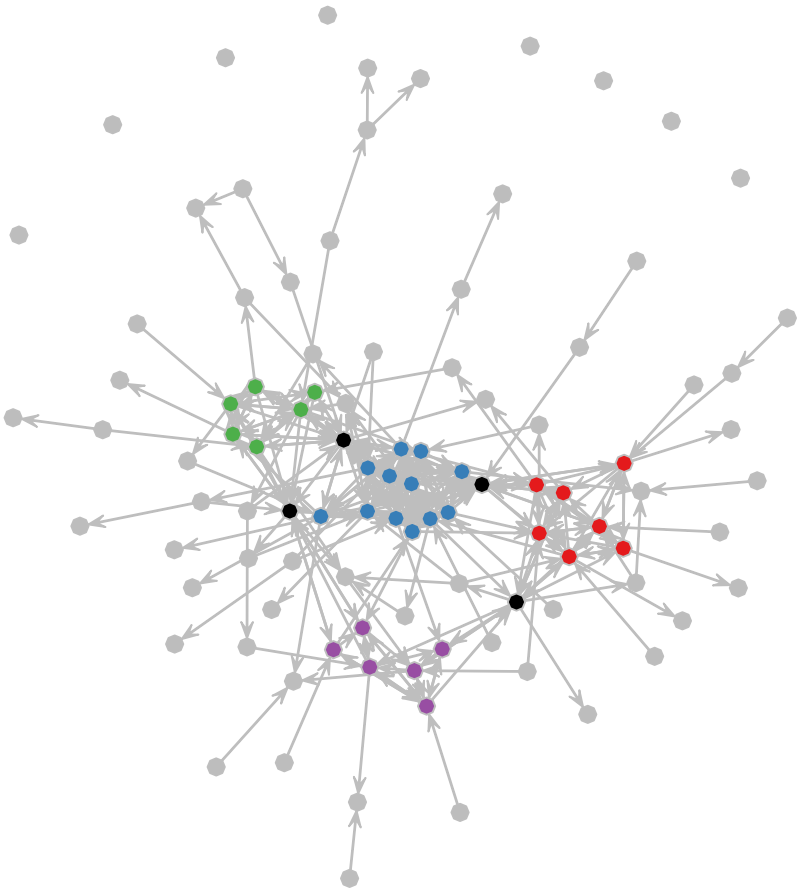

Figure 3.3 – *Example of a network with community structures. Overlaps are represented in black and outliers in gray.*

Networks with community structures

The results that we obtained are presented in Table 3.2 and in Figure 3.5. We can observe that CFinder, MMSB, and OSBM lead to very accurate estimates $\hat{\mathbf{Z}}$ of the true clustering matrix $\mathbf{Z}$. For most networks, they retrieve the clusters and overlaps perfectly although CFinder and MMSB appear to be slightly biased. Indeed, while the median of the L_2 distance $d(\mathbf{P}, \hat{\mathbf{P}})$ over the 100 samples is null for OSBM, it is equal to 22 for CFinder and 27.5 for MMSB. Since CFinder is an algorithmic approach, and not a probabilistic model, it does not classify a vertex v_i if it does not belong to any k-cliques of a k-clique community. Conversely, OSBM is more flexible and can take the random nature of the network into account. Indeed, the edges are assumed to be drawn randomly, and, given each pair of vertices, OSBM deciphers whether or not they are likely to belong to the same class, depending on their connection profiles. Therefore, OSBM can predict that

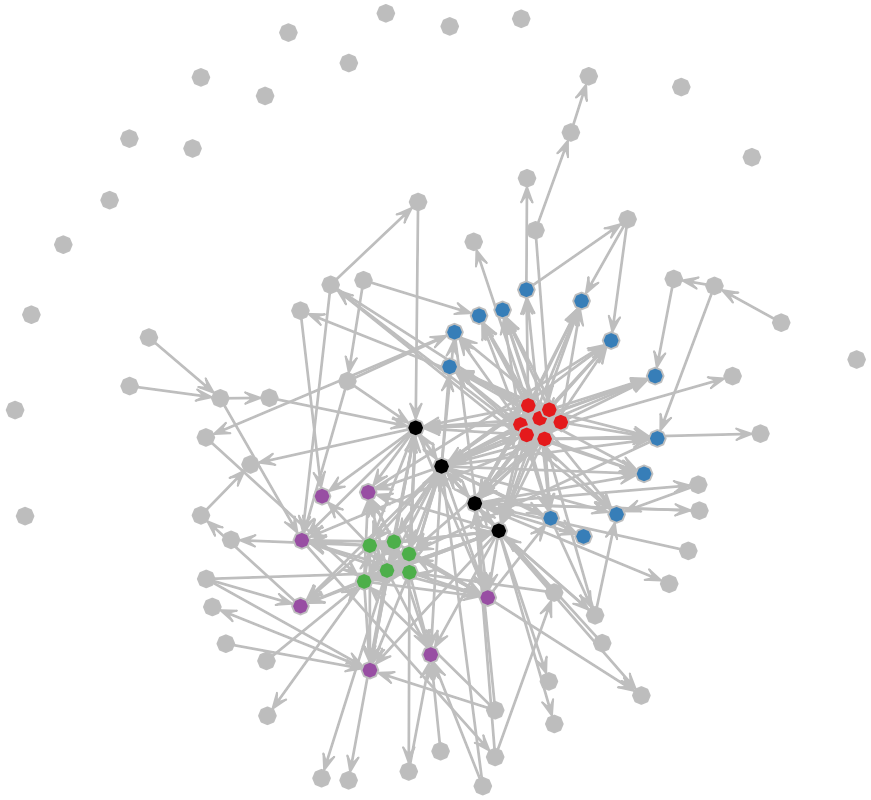

Figure 3.4 – *Example of a network with community structures and stars. Overlaps are represented in black and outliers in gray.*

v_i belongs to a class q although it does not belong to any k-cliques. Overall, we found that MMSB retrieves the clusters well but often misclassifies some of the overlaps. Thus, if a given vertex belongs to several clusters, it tends to be classified by MMSB into only one of them. Nevertheless, the results clearly illustrate that MMSB improves over SBM, which cannot retrieve any of the overlapping clusters. It should also be noted that CFinder has fewer outliers (Figure 3.5) than MMSB and OSBM and appears to be slightly more stable when looking for overlapping community structures in networks.

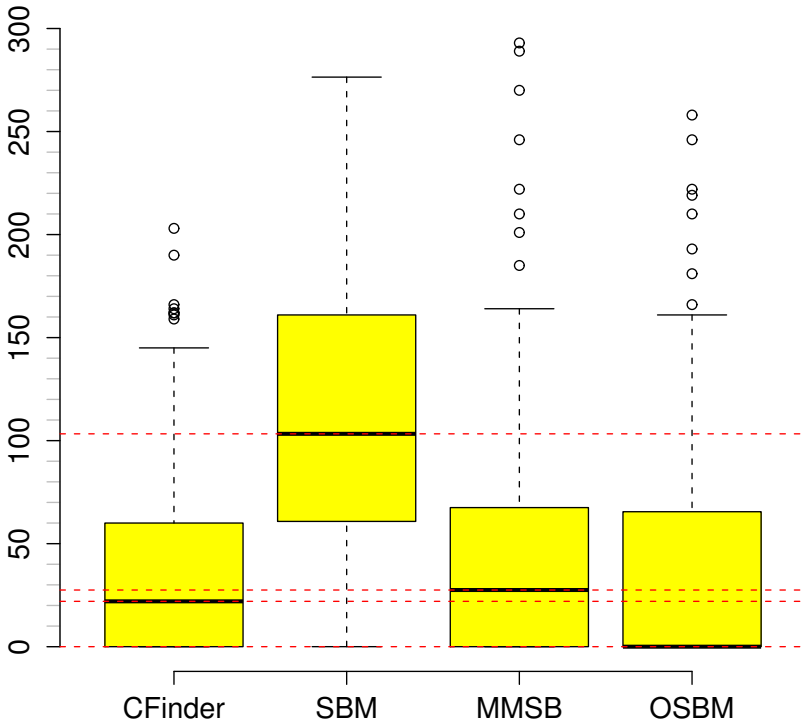

Figure 3.5 – *L_2 distance $d(\mathbf{P}, \hat{\mathbf{P}})$ over the* 100 *samples of networks with community structures, for CFinder and OSBM. Measures how well the underlying cluster assignment structure has been retrieved.*

	Mean	Median	Min	Max
CFinder	43.53	22	**0**	**203**
SBM	116.46	103.3	**0**	321
MMSB	53.76	27.5	**0**	293
OSBM	**41.83**	**0**	**0**	258

Table 3.2 – *Comparison of CFinder, SBM, and OSBM in terms of the L_2 distance $d(\mathbf{P}, \hat{\mathbf{P}})$ over the* 100 *samples of networks with community structures.*

Networks with community structures and stars

In this set of experiments, we considered networks with more complex topologies. As shown, in Table 3.3 and in Figure 3.6, the results of CFinder dramatically degrade while those of OSBM remain more stable. Indeed, the median of the L_2 distances $d(\mathbf{P}, \hat{\mathbf{P}})$ over the 100 samples is equal to 43 for OSBM, while it is equal to 354.5 for CFinder. This can be easily explained since CFinder only looks for community structures of adjacent k-cliques, and can not retrieve classes with low *intra* connection probabilities. Conversely, OSBM uses a $Q \times Q$ real matrix $\mathbf{W}$ and two real vectors $\mathbf{U}$ and $\mathbf{V}$ of size Q to model the *intra* and *inter* connection probabilities. No assumption is made on these matrix and vectors such that OSBM can take heterogeneous and complex topologies into account. As for CFinder, the results of MMSB degrades although they remain better than SBM. As for the previous section, MMSB retrieves the clusters well but misclassifies the overlaps more frequently when considering networks with community structures and stars.

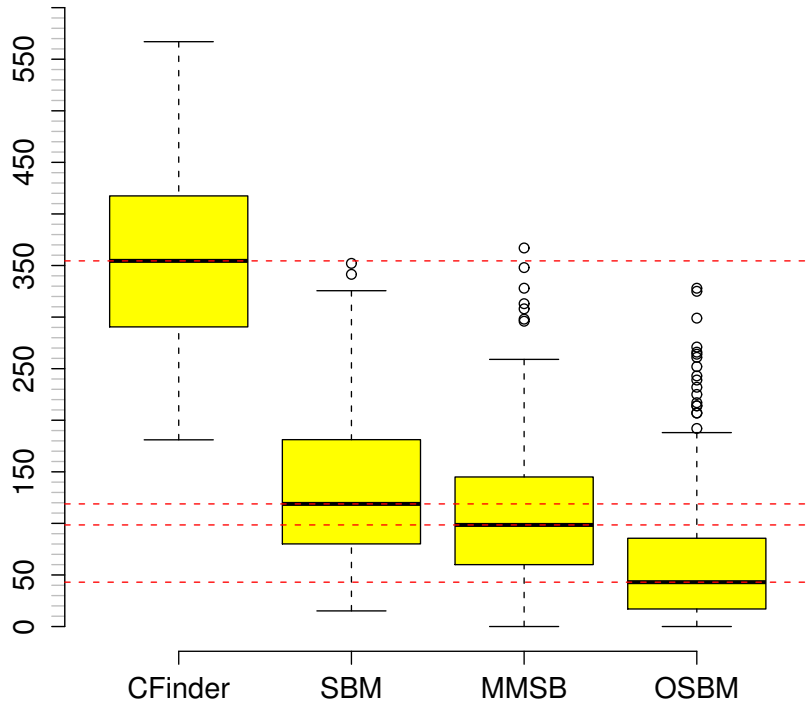

Figure 3.6 – L_2 *distance* $d(\mathbf{P}, \hat{\mathbf{P}})$ *over the* 100 *samples of networks with community structures and stars, for CFinder and OSBM. Measures how well the underlying cluster assignment structure has been retrieved.*

	Mean	Median	Min	Max
CFinder	362.07	354.5	181	567
SBM	134.68	118.87	15.14	352.09
MMSB	119.01	98.5	**0**	367
OSBM	**77**	**43**	**0**	**328**

Table 3.3 – *Comparison of CFinder, SBM, and OSBM in terms of the L_2 distance $d(\mathbf{P}, \hat{\mathbf{P}})$ over the* 100 *samples of networks with community structures and stars.*

3.6.2 French political blogosphere

We consider the French political blogosphere network and we focus on a subset of 196 vertices connected by 2864 edges. The data consists of a single day snapshot of political blogs automatically extracted on 14th october 2006 and manually classified by the "Observatoire Présidentielle project" (Zanghi et al. 2008). Nodes correspond to hostnames and there is an edge between two nodes if there is a known hyperlink from one hostname to another. The four main political parties which are present in the data set are the UMP (french "republican"), UDF ("moderate" party), liberal party (supporters of economic-liberalism), and PS (french "democrat"). Therefore, we applied our algorithm with $Q = 4$ clusters and we obtained the results presented in Figure 3.7 and in Table 3.4.

First, we notice that the clusters we found are highly homogeneous and correspond to the well known political parties. Thus, cluster 1 contains 35 blogs among which 33 are associated to UMP while cluster 2 contains 39 blogs among which 30 are related to UDF. Similarly, it follows that cluster 3 corresponds to the liberal party and cluster 4 to PS. We found nine overlaps. Thus, three blogs associated to UMP belong to both cluster 1 (UMP) and 2 (UDF). This is a result we expected since these two political parties are known to have some relational ties. Moreover, a blog associated to UDF belongs to both cluster 1 (UMP) and 4 (PS) while another UDF blog belongs to cluster 2 (UDF) and 4 (PS). This can be easily understood since UDF is a moderate party. Therefore, it is not surprising to find UDF blogs with links with the two biggest political parties in France, representing the left and right wings. Very interestingly, among the nine overlaps we found, four of them correspond to blogs of political analysts. Thus, a blog overlaps cluster 1 (UMP) and 4 (PS). Another one overlaps cluster 2 (UDF), 3 (liberal party), and 4 (PS). Finally, the two last blogs of political analysts overlap cluster 2 (UDF) and 4 (PS).

We ran CFinder and we used the criterion (Palla et al. 2005) they proposed to select k (see Section 3.1). Thus, we ran the software for various values of k and we found $\hat{k} = 7$. Lower values lead to giant components which smear the details of the network. Conversely, for higher values, the communities start disintegrating. Using $\hat{k}$, we uncovered 11 clusters which correspond to sub-clusters of the clusters we found using OSBM. For instance, cluster 3 (liberal party) was split into two clusters, whereas cluster 4 (PS) was split into three. Indeed, while OSBM predicted that the connection profiles of these sub-clusters were very similar and therefore should be merged, CFinder could not uncover any k-clique community, that is a union of *fully* connected sub-graphs of size k, containing these

sub-clusters. Note that using CFinder, we retrieved the overlaps uncovered by our algorithm. CFinder did not classify 95 blogs.

We also clustered the blogs of the network using MMSB and SBM. As previously, for both models, we used $Q + 1$ clusters and we identified the class of outliers. The results of MMSB are presented in Figure 3.8. Overall, we can notice that MMSB lead to similar clusters as OSBM, although cluster 4 is less homogeneous in MMSB than in OSBM. We found eight overlaps using MMSB and we emphasize that five of them correspond exactly to the one found with our approach. Thus, the model retrieved two among the three UMP blogs overlapping cluster 1 (UMP) and 2 (UDF). Moreover, MMSB uncovered the UDF blog overlapping cluster 1 (UMP) and 4 (PS), as well as the blog of political analysts overlapping cluster 2 (UDF), 3 (liberal party), and 4 (PS). It also retrieved the blog of political analysts overlapping cluster 1 (UMP) and 4 (PS). Finally, the results of SBM are presented in Figure 3.9. Again, the clusters found by this approach are very similar to the one uncovered by OSBM. However, because SBM does not allow each vertex to belong to multiple clusters, it misses a lot of information in the network. In particular, while some of the blogs of political analysts are viewed as overlaps by OSBM, because of their relational ties with the different political parties, they are all classified into a single cluster by SBM.

3.89	0.17	0.54	-0.70	-0.70
0.17	2.47	-0.40	-0.84	0.40
0.55	-0.40	4.43	-0.85	-0.38
-0.70	-0.84	-0.85	1.66	0.87
-0.70	0.40	-0.38	0.87	-3.60

Table 3.4 – *The estimated $\tilde{\mathbf{W}}$ matrix for the classification of the blogs into $Q = 4$ clusters using OSBM. The 4×4 matrix on the top left hand side represents the $\mathbf{W}$ matrix while the vectors on the top right hand side and bottom left hand side represent the vectors $\mathbf{U}$ and $\mathbf{V}^\intercal$ respectively. The remaining term corresponds to the bias. The diagonal of $\mathbf{W}$ indicates that blogs have a heavy tendency to connect to blogs of the same class. Blogs of cluster 1 (UMP) have also a positive tendency to connect to blogs of clusters 2 (UDF) and 3 (liberal party). Conversely, blogs of cluster 4 (PS), representing the left wing, are more isolated in the network.*

	UMP	UDF	liberal	PS	analysts	others
cluster 1	30 + 3	0 + 1	0	0	0 + 1	0
cluster 2	2 + 3	29 + 1	0	0	1 + 3	0
cluster 3	0	0	24	0	1 + 1	0
cluster 4	0	0 + 2	0	40	0 + 4	1
outliers	5	1	1	17	5	30

Figure 3.7 – *Classification of the blogs into $Q = 4$ clusters using OSBM. The entry (i, j) of the matrix describes the number of blogs associated to the j-th political party (column) and classified into cluster i (row). Each entry distinguishes blogs which belong to a unique cluster from overlaps (single membership blogs + overlaps). The last row corresponds to the null component.*

	UMP	UDF	liberal	PS	analysts	others
cluster 1	27 + 2	0 + 2	0	0	1 + 1	0
cluster 2	2 + 2	29 + 1	0	0 + 1	3 + 2	0
cluster 3	0	0	25	0	1 + 2	0
cluster 4	0	0 + 1	0	30 + 1	0 + 2	1
cluster 5	9	1	0	26	3	30

Figure 3.8 – *Classification of the blogs into $Q = 5$ clusters using MMSB. The entry (i, j) of the matrix describes the number of blogs associated to the j-th political party (column) and classified into cluster i (row). Each entry distinguishes blogs which belong to a unique cluster from overlaps (single membership blogs + overlaps). Cluster 5 corresponds to the class of outliers.*

	UMP	UDF	liberal	PS	analysts	others
cluster 1	37	0	1	0	0	2
cluster 2	1	31	0	0	1	0
cluster 3	0	0	24	0	1	0
cluster 4	0	0	0	26	0	0
cluster 5	2	1	0	31	9	29

Figure 3.9 – *Classification of the blogs into $Q = 5$ clusters using SBM. The entry (i, j) of the matrix describes the number of blogs associated to the j-th political party (column) and classified into cluster i (row). Cluster 5 corresponds to the class of outliers.*

3.6.3 Saccharomyces cerevisiae transcription network

We consider the *yeast* transcriptional regulatory network described in Milo et al. (2002) and we focus on a subset of 197 vertices connected by 303 edges. Nodes of the network correspond to operons, and two operons are linked if one operon encodes a transcriptional factor that directly regulates the other operon. The network is made of three regulation patterns, each one of them having its own regulators and regulated operons. Therefore, using $Q = 6$ clusters, we applied our algorithm and we obtained the results in Table 3.5.

First, we notice that clusters 1, 3, and 5 contain only two operons each. These operons correspond to hubs which regulate respectively the nodes of clusters 2, 4, and 6, all having a very low *intra* connection probability. To analyze our results, we used GOToolBox (Martin et al. 2004) on each cluster. This software aims at identifying statistically over-represented terms of the Gene Ontology (GO) in a gene data set. We found that the clusters correspond to well known biological functions. Thus, the nodes of cluster 2 are regulated by STE12 and TEC1 which are both involved in the response to glucose limitation, nitrogen limitation and abundant fermentable carbon source. Similarly, MSN4 and MSN2 regulate the nodes of cluster 4 in response to different stress such as freezing, hydrostatic pressure, and heat acclimation. Finally, the nodes of cluster 6 are regulated by YAP1 and SKN7 in the presence of oxygen stimulus. Our algorithm was able to uncover two overlapping clusters (operons in bold in Table. 3.5). Interestingly, contrary to the other operons of clusters 2, 4, and 6, which are all regulated by operons of a single cluster (cluster 1, 3, or 5), these overlaps correspond to co-regulated operons. Thus, SSA4 and TKL2 belong to cluster 2 and 4 since they are co-regulated by (STE12, TEC1) and (MSN4 and MSN2). Moreover, HSP78, CTT1, and PGM2 belong to cluster 4 and 6 since they are co-regulated by (MSN4, MSN2) and (YAP1, SKN7). It should also be noted that OSBM did not classify 112 operons which all have very low output and input degrees.

Because the network is sparse, we obtained very poor results with CFinder. Indeed, the network contains only one 3-clique and no k-clique for $k > 3$. Therefore, for $k = 2$, all the operons were classified into a single cluster and no biological information could be retrieved. For $k = 3$, only three operons were classified into a single class and for $k > 3$ no operon was classified.

As previously, we ran MMSB and SMB with $Q + 1$ clusters and we identified the class of outliers. Both approaches retrieved the six clusters found by OSBM. However, we emphasize that contrary to the political blogoshpere network, MMSB did not uncover any overlap in the *yeast* transcriptional regulatory network.

As in Section 3.6.1, these results clearly illustrate the capacity of OSBM to retrieve overlapping clusters in networks with complex topological structures. In particular, in situations were networks are not made of community structures, while the results of CFinder dramatically degrade or cannot even be interpreted, OSBM seems particularly promising.

cluster	size	operons
1	2	STE12 TEC1
2	33	YBR070C MID2 YEL033W SRD1 TSL1 RTS2 PRM5 YNL051W PST1 YJL142C **SSA4** YGR149W SPO12 YNL159C SFP1 YHR156C YPS1 YPL114W HTB2 MPT5 SRL1 DHH1 **TKL2** PGU1 YHL021C RTA1 WSC2 GAT4 YJL017W TOS11 YLR414C BNI5 YDL222C
3	2	MSN4 MSN2
4	32	CPH1 **TKL2** HSP12 SPS100 MDJ1 GRX1 SSA3 ALD2 GDH3 GRE3 HOR2 ALD3 SOD2 ARA1 HSP42 YNL077W **HSP78** GLK1 DOG2 HXK1 RAS2 **CTT1** HSP26 TPS1 TTR1 HSP104 GLO1 **SSA4** PNC1 MTC2 YGR086C **PGM2**
5	2	YAP1 SKN7
6	19	YMR318C **CTT1** TSA1 CYS3 ZWF1 HSP82 TRX2 GRE2 SOD1 AHP1 YNL134C **HSP78** CCP1 TAL1 DAK1 YDR453C TRR1 LYS20 **PGM2**

Table 3.5 – *Classification of the operons into $Q = 6$ clusters. Operons in bold belong to multiple clusters.*

Conclusion

In this chapter, we proposed a new random graph model, the Overlapping Stochastic Block Model, which can be used to retrieve overlapping clusters in networks. We used global and local variational techniques to obtain a tractable lower bound of the observed log-likelihood and we defined an EM like procedure which optimizes the model parameters in turn. We showed that the model is identifiable within classes of equivalence and we illustrated the efficiency of our approach compared to other methods, using simulated data and real networks. Since no assumption is made on the matrix $\mathbf{W}$ and vectors $\mathbf{U}$ and $\mathbf{V}$ used to characterize the connection probabilities, the model can take very different topological structures into account and seems particularly promising for the analysis of networks. In the experiment section, we set the number Q of classes using *a priori* information we had about the networks. However, we believe it is crucial to develop a model selection criterion to estimate the number of classes automatically from the topology.

4 Model selection in overlapping stochastic block models

Contents

In the previous chapter, we introduced the Overlapping Stochastic Block Model (OSBM), a new random graph model which allows the vertices of a network to belong to multiple classes. A variational EM algorithm was considered for estimation as well as clustering purposes and the model was shown to be able to take very different topological structures into account. However, no model selection criterion is yet available to estimate the number of components from the data. To tackle this issue, we consider a Bayesian framework as in Chapter 2. We then propose a criterion, that we call $ILvb_{osbm}$, based on a non asymptotic approximation of the marginal log-likelihood. We describe how $ILvb_{osbm}$ can be computed through a variational Bayes EM algorithm. Finally, experiments are carried out to assess the criterion using simulated data and the transcriptional regulatory network of *Saccharomyces cerevisiae*.

4.1 A Bayesian Overlapping Stochastic Block Model

The data we model consists of a $N \times N$ binary matrix $\mathbf{X}$ with entries X_{ij} describing the presence or absence of an edge from vertex i to vertex j. Both directed and undirected relations can be analyzed but in the following, we concentrate on directed relations. Moreover, we assume that the graph we consider does not contain any self loop. Therefore, the variables X_{ii} will not be taken into account.

As shown in the previous chapter, the Overlapping Stochastic Block Model (OSBM) associates to each vertex of a network a latent variable $\mathbf{Z}_i$ drawn from a multivariate Bernoulli distribution:

$$\mathbf{Z}_i \sim \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} (1-\alpha_q)^{1-Z_{iq}},$$

where Q denotes the number of classes considered. The edges are then assumed to be drawn from a Bernoulli distribution:

$$X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j \sim \mathcal{B}\big(X_{ij};\, g(a_{\mathbf{Z}_i,\mathbf{Z}_j})\big) = e^{X_{ij} a_{\mathbf{Z}_i,\mathbf{Z}_j}} g(-a_{\mathbf{Z}_i,\mathbf{Z}_j}),$$

where

$$a_{\mathbf{Z}_i,\mathbf{Z}_j} = \mathbf{Z}_i^\intercal \, \mathbf{W} \, \mathbf{Z}_j + \mathbf{Z}_i^\intercal \, \mathbf{U} + \mathbf{V}^\intercal \, \mathbf{Z}_j + W^*,$$

and $g(x) = (1+e^{-x})^{-1}$ is the logistic sigmoid function. According to OSBM, the latent variables $\mathbf{Z}_1, \ldots, \mathbf{Z}_N$ are iid and given this latent structures, all the edges are supposed to be independent.

Thus, when considering a directed graph without self loop, we recall that:

$$p(\mathbf{Z} \,|\, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \alpha_q^{Z_{iq}} (1-\alpha_q)^{1-Z_{iq}},$$

and

$$\begin{aligned} p(\mathbf{X} \,|\, \mathbf{Z}, \tilde{\mathbf{W}}) &= \prod_{i \neq j}^{N} p(X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \tilde{\mathbf{W}}) \\ &= \prod_{i \neq j}^{N} e^{X_{ij} a_{\mathbf{Z}_i,\mathbf{Z}_j}} g(-a_{\mathbf{Z}_i,\mathbf{Z}_j}). \end{aligned} \tag{4.1}$$

Finally, to keep the notations uncluttered, we will use the notations of the previous chapter, that is $\tilde{\mathbf{Z}}_i = \big(\mathbf{Z}_i, 1\big)^\intercal, \forall i$ and:

$$\tilde{\mathbf{W}} = \begin{pmatrix} \mathbf{W} & \mathbf{U} \\ \mathbf{V}^\intercal & W^* \end{pmatrix}.$$

As the (non overlapping) Stochastic Block Model (SBM), OSBM can be described in a full Bayesian framework by introducing some conjugate prior distributions for the model parameters. Since $p(\mathbf{Z}_i \,|\, \boldsymbol{\alpha})$ is a multivariate Bernoulli distribution, we consider independent Beta distributions for the class probabilities:

$$p(\boldsymbol{\alpha}) = \prod_{q=1}^{Q} \mathrm{Beta}(\alpha_q;\, \eta_q^0, \zeta_q^0),$$

where $\eta_q^0 = \zeta_q^0 = 1/2, \forall q$. As mentioned already, this corresponds to a product of non-informative Jeffreys prior distributions. A uniform distribution can also be chosen simply by fixing $\eta_q^0 = \zeta_q^0 = 1, \forall q$.

In order to model the $(Q+1) \times (Q+1)$ real matrix $\tilde{\mathbf{W}}$, we consider the vec operator which stacks the columns of a matrix into a vector. Thus, if $\mathbf{A}$ is a 2×2 matrix such that:

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

then

$$\mathbf{A}^{\text{vec}} = \begin{pmatrix} A_{11} \\ A_{21} \\ A_{12} \\ A_{22} \end{pmatrix}.$$

Following the work of Jaakkola and Jordan (2000) on Bayesian logistic regression, where a Gaussian distribution is used for the weight vector $\boldsymbol{\beta}$, we model the vector $\tilde{\mathbf{W}}^{\text{vec}}$ using a multivariate Gaussian prior distribution with mean vector $\tilde{\mathbf{W}}_0^{\text{vec}}$ and covariance matrix $\mathbf{S}_0$:

$$p(\tilde{\mathbf{W}}^{\text{vec}}) = \mathcal{N}(\tilde{\mathbf{W}}^{\text{vec}}; \tilde{\mathbf{W}}_0^{\text{vec}}, \mathbf{S}_0).$$

The inference procedure introduced in this chapter is described in a general setting, allowing any covariance matrix $\mathbf{S}_0$. In practice, we have used $\mathbf{S}_0 = \sigma^2 \mathbf{I}$ in all the experiments that we carried out, where $\mathbf{I}$ denotes the identity matrix.

4.2 Estimation

In this section, we propose a Variational Bayes EM (VBEM) algorithm, based on global and local variational techniques, which leads to an approximation of the full posterior distribution over the model parameters and latent variables, given the observed data $\mathbf{X}$. As for the VBEM algorithm described in Chapter 2, this procedure relies on a lower bound which will be later used as non asymptotic approximation of the marginal log-likelihood $\log p(\mathbf{X})$.

4.2.1 The q-transformation

The posterior distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}} \,|\, \mathbf{X})$ is intractable and so we propose to rely on variational approximations. Following the VBEM algorithm we introduced in Chapter 2, the marginal likelihood can be decomposed into two terms:

$$\log p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}\left((q(\cdot)||p(\cdot \,|\, \mathbf{X})\right),$$

where

$$\mathcal{L}(q) = \sum_{\mathbf{Z}} \int \int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})} \right\} d\,\boldsymbol{\alpha}\, d\,\tilde{\mathbf{W}}, \qquad (4.2)$$

and

$$\mathrm{KL}(q(\cdot)||p(\cdot|\,\mathbf{X})) = -\sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log\left\{\frac{p(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}} \,|\, \mathbf{X})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})}\right\} d\,\boldsymbol{\alpha}\, d\,\tilde{\mathbf{W}}\,. \tag{4.3}$$

$\mathcal{L}$ is a lower bound of $\log p(\mathbf{X})$ and $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence between the distribution $q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ and the distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}} \,|\, \mathbf{X})$. To obtain a tractable algorithm, we assume that $q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ can be factorized such that:

$$q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) = q(\boldsymbol{\alpha}) q(\tilde{\mathbf{W}}) q(\mathbf{Z}) = q(\boldsymbol{\alpha}) q(\tilde{\mathbf{W}}) \Big(\prod_{i=1}^{N} \prod_{q=1}^{Q} q(Z_{iq})\Big).$$

This factorization should be compared with (2.3.2). Indeed, contrary to SBM where we assumed that $q(\mathbf{Z}) = \prod_{i=1}^{N} q(\mathbf{Z}_i)$, we now consider a factorization over all the vertices and classes, that is $q(\mathbf{Z}) = \prod_{i=1}^{N} \prod_{q=1}^{Q} q(Z_{iq})$. Only with this later assumption did we obtain analytical expressions, as shown in Section 4.2.3.

At this point, the lower bound is still intractable due to the logistic function in the distribution $p(\mathbf{X} \,|\, \mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ (see 4.1 and 4.2). Therefore, a second level of approximation is required.

4.2.2 The ξ-transformation

As noted in Proposition 4.1, a tractable lower bound can be obtained using the work of Jaakkola and Jordan (2000).

Proposition 4.1 *(Proof in Appendix D.1) Given any $N \times N$ positive real matrix $\boldsymbol{\xi}$, a lower bound of the first lower bound is given by:*

$$\log p(\mathbf{X}) \geq \mathcal{L}(q) \geq \mathcal{L}(q;\, \boldsymbol{\xi}),$$

where

$$\mathcal{L}(q;\, \boldsymbol{\xi}) = \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log\Big(\frac{h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) p(\mathbf{Z} \,|\, \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\tilde{\mathbf{W}})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})}\Big) d\,\boldsymbol{\alpha}\, d\,\tilde{\mathbf{W}},$$

and

$$\log h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) = \sum_{i \neq j}^{N} \left\{ (X_{ij} - \frac{1}{2}) a_{\mathbf{Z}_i, \mathbf{Z}_j} - \frac{\xi_{ij}}{2} + \log g(\xi_{ij}) - \lambda(\xi_{ij})(a^2_{\mathbf{Z}_i, \mathbf{Z}_j} - \xi_{ij}^2) \right\}.$$

For now, $\boldsymbol{\xi}$ is held fixed but we will see in Section 4.2.4 how it can be estimated from the data.

4.2.3 Variational Bayes EM

In order to approximate the posterior distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}} \,|\, \mathbf{X})$ with a distribution $q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$, a VBEM algorithm is applied on the lower bound $\mathcal{L}(q;\, \boldsymbol{\xi})$. For a general description of the VBEM algorithm, we refer to Section 1.2.2. The algorithm starts with a matrix $\boldsymbol{\tau}$ initialized using a hierarchical classification algorithm, as in Chapter 3. Given a matrix $\boldsymbol{\xi}$, the

Propositions 4.2,4.3, and 4.4 are then used to maximize the lower bound $\mathcal{L}(q;\boldsymbol{\xi})$ with respect to $q(\mathbf{Z},\boldsymbol{\alpha},\tilde{\mathbf{W}})$. We call variational Bayes E step the optimization of the distributions $q(Z_{iq})$ and variational Bayes M-step the optimization of the remaining factors.

Proposition 4.2 *(Proof in Appendix D.2) VBEM leads to a distribution $q(\boldsymbol{\alpha})$ which takes the same functional form as the prior $p(\boldsymbol{\alpha})$:*

$$q(\boldsymbol{\alpha}) = \prod_{q=1}^{Q} \text{Beta}(\alpha_q;\, \eta_q^N, \zeta_q^N),$$

where

$$\eta_q^N = \eta_q^0 + \sum_{i=1}^{N} \tau_{iq},$$

and

$$\zeta_q^N = \zeta_q^0 + N - \sum_{i=1}^{N} \tau_{iq}.$$

Proposition 4.3 *(Proof in Appendix D.3) VBEM leads to a distribution $q(\tilde{\mathbf{W}})$ which takes the same functional form as the prior $p(\tilde{\mathbf{W}})$:*

$$q(\tilde{\mathbf{W}}^{vec}) = \mathcal{N}(\tilde{\mathbf{W}}^{\text{vec}};\, \tilde{\mathbf{W}}_N^{\text{vec}}, \mathbf{S}_N),$$

with

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + 2\sum_{i\neq j}^{N} \lambda(\xi_{ij})(\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i),$$

and

$$\tilde{\mathbf{W}}_N^{\text{vec}} = \mathbf{S}_N \left\{ \mathbf{S}_0^{-1}\, \tilde{\mathbf{W}}_0^{\text{vec}} + \sum_{i\neq j}^{N} (X_{ij} - \frac{1}{2})\, \tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i \right\}.$$

The symbol $\otimes$ denotes the Kronecker product. Moreover, each $(Q+1)\times(Q+1)$ probability matrix $\tilde{\mathbf{E}}_i$ satisfies:

$$\begin{aligned}\tilde{\mathbf{E}}_i &= \mathrm{E}_{\mathbf{Z}_i}[\tilde{\mathbf{Z}}_i\, \tilde{\mathbf{Z}}_i^\intercal] \\ &= \begin{pmatrix} \tau_{i1} & \tau_{i1}\tau_{i2} & \dots & \tau_{i1}\tau_{iQ} & \tau_{i1} \\ \tau_{i2}\tau_{i1} & \tau_{i2} & \dots & \tau_{i2}\tau_{iQ} & \tau_{i2} \\ \vdots & & & & \vdots \\ \tau_{iQ}\tau_{i1} & \tau_{iQ}\tau_{i2} & \dots & \tau_{iQ} & \tau_{iQ} \\ \tau_{i1} & \tau_{i2} & \dots & \tau_{iQ} & 1 \end{pmatrix}.\end{aligned}$$

Proposition 4.4 *(Proof in Appendix D.4) VBEM leads to a distribution $q(Z_{iq})$ with the same functional form as the prior:*

$$q(Z_{iq}) = \mathcal{B}(Z_{iq};\, \tau_{iq}),$$

where

$$\begin{aligned}\tau_{iq} = g\Big\{ &\psi(\eta_q^N) - \psi(\zeta_q^N) + \sum_{j\neq i}^{N} (X_{ij} - \frac{1}{2})\, \tilde{\boldsymbol{\tau}}_j^\intercal (\tilde{\mathbf{W}}_N^\intercal)_{\cdot q} + \sum_{j\neq i}^{N} (X_{ji} - \frac{1}{2})\, \tilde{\boldsymbol{\tau}}_j^\intercal (\tilde{\mathbf{W}}_N)_{\cdot q} \\ &- \mathrm{Tr}\Big(\big(\boldsymbol{\Sigma}'_{qq} + 2\sum_{l\neq q}^{Q+1} \tilde{\tau}_{il}\, \boldsymbol{\Sigma}'_{ql} \big) \big(\sum_{j\neq i}^{N} \lambda(\xi_{ij})\, \tilde{\mathbf{E}}_j \big) + \big(\boldsymbol{\Sigma}_{qq} + 2\sum_{l\neq q}^{Q+1} \tilde{\tau}_{il}\, \boldsymbol{\Sigma}_{ql} \big) \big(\sum_{j\neq i}^{N} \lambda(\xi_{ji})\, \tilde{\mathbf{E}}_j \big) \Big) \Big\},\end{aligned}$$

and $\boldsymbol{\Sigma}_{ql} = \mathrm{E}_{\tilde{\mathbf{W}}_q, \tilde{\mathbf{W}}_l}[\tilde{\mathbf{W}}_{\cdot q}\, \tilde{\mathbf{W}}_{\cdot l}^\intercal]$, $\boldsymbol{\Sigma}'_{ql} = \mathrm{E}_{\tilde{\mathbf{W}}_{q\cdot}, \tilde{\mathbf{W}}_{l\cdot}}[\tilde{\mathbf{W}}_{q\cdot}^\intercal\, \tilde{\mathbf{W}}_{l\cdot}]$.

4.2.4 Optimization of ξ

So far, we have seen how a VBEM algorithm could be used to obtain an approximation of the posterior distribution $p(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}} \mid \mathbf{X})$ for a given matrix $\boldsymbol{\xi}$. However, we have not addressed yet how $\boldsymbol{\xi}$ could be estimated from the data. We follow the work of Bishop and Svensén (2003) on Bayesian hierarchical mixture of experts. Thus, given a distribution $q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$, the lower bound $\mathcal{L}(q; \boldsymbol{\xi})$ is maximized with respect to each variable ξ_{ij} in order to obtain the tightest lower bound $\mathcal{L}(q; \boldsymbol{\xi})$ of $\mathcal{L}(q)$. As shown in Proposition 4.5 and Appendix D.5, this optimization leads to estimates $\hat{\xi_{ij}}$ of ξ_{ij}.

Proposition 4.5 *(Proof in Appendix D.5) An estimate $\hat{\xi_{ij}}$ of ξ_{ij} is given by:*

$$\hat{\xi}_{ij} = \sqrt{\mathrm{Tr}\Big(\big(\mathbf{S}_N + \tilde{\mathbf{W}}_N^{\mathrm{vec}} (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \big) (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big)}.$$

This gives rise to a three step optimization algorithm. Given a matrix $\boldsymbol{\xi}$, the variational Bayes E and M steps are used to approximate the posterior distribution over the model parameters and latent variables. The distribution $q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})$ is then held fixed while the lower bound $\mathcal{L}(q; \boldsymbol{\xi})$ is maximized with respect to $\boldsymbol{\xi}$. These three stages are repeated until convergence of the lower bound (see Algorithm 8).

For all the experiments that we carried out, we set $\mathbf{S}_0 = \sigma^2 \mathbf{I}$ with $\sigma^2 = 1$ and $\xi_{ij} = 0.001, \forall i \neq j$. As the VEM algorithm proposed in Chapter 3, the computational cost of the algorithm is equal to $O(N^2 Q^4)$. Note that a R package called "OSBM" will be soon available on the CRAN. For now, the code is available upon request.

Algorithm 8: Variational Bayes inference for overlapping stochastic block model when applied on a directed graph without self loop.

// INITIALIZATION

Initialize $\boldsymbol{\tau}$ with an Ascendant Hierarchical Classification algorithm
Initialize $\xi_{ij}, \forall i \neq j$

// OPTIMIZATION

repeat

$\tilde{\mathbf{E}}_i \leftarrow \mathrm{E}_{\mathbf{Z}_i}[\tilde{\mathbf{Z}}_i \tilde{\mathbf{Z}}_i^\intercal], \forall i$

// M-step

$\eta_q^N \leftarrow \eta_q^0 + \sum_{i=1}^N \tau_{iq}, \forall q$

$\zeta_q^N \leftarrow \zeta_q^0 + N - \sum_{i=1}^N \tau_{iq}, \forall q$

$\mathbf{S}_N^{-1} \leftarrow \mathbf{S}_0^{-1} + 2\sum_{i \neq j}^N \lambda(\xi_{ij})(\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i)$

$\tilde{\mathbf{W}}_N^{\text{vec}} \leftarrow \mathbf{S}_N \left\{ \mathbf{S}_0^{-1} \tilde{\mathbf{W}}_0^{\text{vec}} + \sum_{i \neq j}^N (X_{ij} - \frac{1}{2}) \tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i \right\}$

// Optimization of $\boldsymbol{\xi}$

$\xi_{ij} \leftarrow \sqrt{\mathrm{Tr}\Big(\big(\mathbf{S}_N + \tilde{\mathbf{W}}_N^{\text{vec}} (\tilde{\mathbf{W}}_N^{\text{vec}})^\intercal \big)(\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big)}, \forall i \neq j$

// E-step

repeat

Compute $\tau_{iq}, \forall (i, q)$ using Proposition 4.4

until $\boldsymbol{\tau}$ *converges*

until $\mathcal{L}(q; \boldsymbol{\xi})$ *converges*

4.3 Model Selection

Given a set of values of Q, we aim at selecting Q^* which maximizes the marginal log-likelihood $\log p(\mathbf{X}|Q)$, also called integrated observed-data log-likelihood. Unfortunately, this quantity is not tractable since for each value of Q, it involves integrating over all possible model parameters and latent variables:

$$\log p(\mathbf{X}|Q) = \log \left\{ \sum_{\mathbf{Z}} \int \int p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}|Q) d\boldsymbol{\alpha}\, d\tilde{\mathbf{W}} \right\}.$$

As in Chapter 2, we propose to replace the marginal log-likelihood with its variational approximation. Thus, for each value of Q considered, Algorithm 8 is applied in order to maximize $\mathcal{L}(q; \boldsymbol{\xi})$ with respect to $q(\cdot)$ and $\boldsymbol{\xi}$. After convergence, the lower bound is then used as an estimation of $\log p(\mathbf{X}|Q)$. Obviously, this approximation cannot be verified analytically because neither $\mathcal{L}(q)$ in (4.2) nor the Kullback-Leibler divergence in (4.3) are tractable. Nevertheless, we rely on this approximation to propose a tractable model selection criterion. After convergence of the algorithm, the lower bound takes a simple form and leads to the first criterion for

OSBM that we call IL_{osbm} (Proof in Appendix D.6):

$$IL_{osbm} = \sum_{i \neq j}^{N} \left\{ \log g(\xi_{ij}) - \frac{\xi_{ij}}{2} + \lambda(\xi_{ij})\xi_{ij}^2 \right\} + \sum_{q=1}^{Q} \log \left\{ \frac{\Gamma(\eta_q^0 + \zeta_q^0)\Gamma(\eta_q^N)\Gamma(\zeta_q^N)}{\Gamma(\eta_q^0)\Gamma(\zeta_q^0)\Gamma(\eta_q^N + \zeta_q^N)} \right\}$$
$$- \frac{1}{2}\log|\frac{\mathbf{S}_0}{\mathbf{S}_N}| - \frac{1}{2}(\tilde{\mathbf{W}}_0^{\text{vec}})^\intercal \, \mathbf{S}_0^{-1} \, \tilde{\mathbf{W}}_0^{\text{vec}} + \frac{1}{2}(\tilde{\mathbf{W}}_N^{\text{vec}})^\intercal \, \mathbf{S}_N^{-1} \, \tilde{\mathbf{W}}_N^{\intercal}$$
$$- \sum_{i=1}^{N}\sum_{q=1}^{Q} \left\{ \tau_{iq} \log \tau_{iq} + (1 - \tau_{iq}) \log(1 - \tau_{iq}) \right\}.$$

4.4 Experiment

We consider simulated data and the transcriptional regulatory network of *yeast* in order to assess the Bayesian inference procedure introduced in Section 4.2.4 as well as the model selection criterion IL_{osbm}.

4.4.1 Simulated data

OSBM is used in this set of experiments to generate networks with community structure, where vertices of a community are mostly connected to vertices of the same community. The networks are made of overlaps and therefore if we rely on a criterion for (non overlapping) SBM as IL_{vb} (see Chapter 2), the number of classes is highly overestimated. Indeed, such criterion uses many extra components to characterize the overlaps between classes. As in Section 3.6.1, to limit the number of free parameters, we consider the $Q \times Q$ real matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} \lambda & -\epsilon & \dots & -\epsilon \\ -\epsilon & \lambda & & \vdots \\ \vdots & & \ddots & -\epsilon \\ -\epsilon & \dots & -\epsilon & \lambda \end{pmatrix},$$

and the Q-dimensional real vectors $\mathbf{U}$ and $\mathbf{V}$:

$$\mathbf{U} = \mathbf{V} = \begin{pmatrix} \epsilon & \dots & \epsilon \end{pmatrix},$$

such that $\lambda = 6$, $\epsilon = 1$, and $W^* = -5.5$. For each value Q_{True} in the set $\{3, \dots, 7\}$, we generate 100 networks (see an example in Figure 4.1) with $N = 100$ vertices and classes mixed in the same proportions , that is $\alpha_1 = \cdots = \alpha_{Q_{True}} = 1/Q_{True}$. The VBEM algorithm is then applied on each network for various numbers of classes $Q \in \{2, \dots, 8\}$. Note that we choose $n_q^0 = 1/2, \forall q$ and $\mathbf{S}_0 = \sigma^2 \mathbf{I}$ with $\sigma^2 = 1$. Like any optimization method, the overlapping clustering algorithm we propose depends on the initialization. Thus, for each simulated network and each number of classes Q, we consider 20 initializations of $\boldsymbol{\tau}$. Finally, we select the best learnt model for which the criterion IL_{osbm} is maximized.

The results presented in Table 4.1 clearly illustrate that IL_{osbm} is a relevant criterion to estimate the number of overlapping classes in networks. For instance, when $Q_{True} = 3$ or $Q_{True} = 4$, IL_{osbm} correctly estimates the

	2	3	4	5	6	7	8
3	0	**99**	1	0	0	0	0
4	0	0	**99**	1	0	0	0
5	0	0	0	**93**	5	2	0
6	0	0	0	7	**64**	22	7
7	0	0	0	0	16	**47**	37

Table 4.1 – *Confusion matrix for* IL_{osbm}. $\lambda = 6, \epsilon = 1, W^* = -5.5, Q_{True} \in \{3, \dots, 7\}$ *and* $Q_{IL_{osbm}} \in \{2, \dots, 8\}$.

number of overlapping classes of 99 of the 100 networks generated. As Q increases, the performances of the criterion remain quite stable.

We now consider networks generated by changing the value of λ to $\lambda = 4$ which causes the *intra* class probabilities to decrease. The results are presented in Table 4.2. It appears that IL_{osbm} has a tendency to overestimate the number of components. Indeed, we found that the more difficult it is to distinguish the communities, the more the criterion tends to use extra classes to model the overlaps between classes.

	2	3	4	5	6	7	8
3	0	**99**	1	0	0	0	0
4	0	0	**85**	9	5	0	1
5	0	0	4	**53**	26	9	8
6	0	0	0	18	**34**	27	21
7	0	0	0	4	18	**30**	48

Table 4.2 – *Confusion matrix for* IL_{osbm}. $\lambda = 4, \epsilon = 1, W^* = -5.5, Q_{True} \in \{3, \dots, 7\}$ *and* $Q_{IL_{osbm}} \in \{2, \dots, 8\}$.

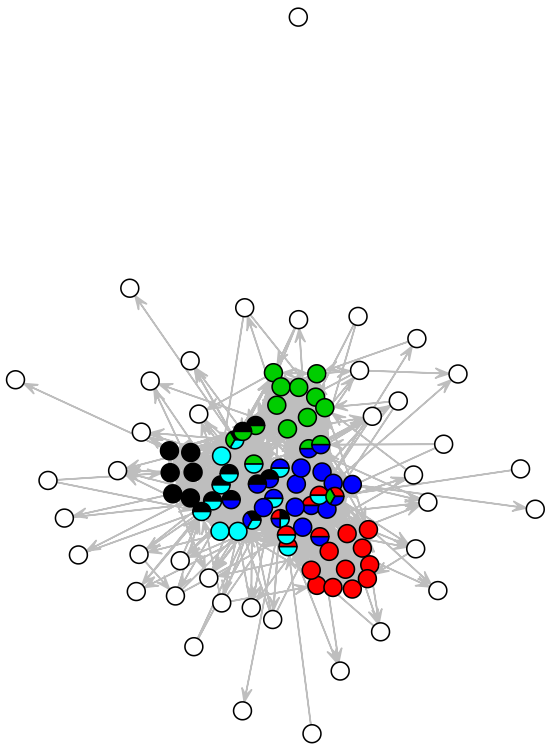

Figure 4.1 – *Figure produced by the R "OSBM" package. Example of a network generated using OSBM, with* $\lambda = 6$, $\epsilon = 1$, $W^* = -5.5$, *and* $Q = 5$ *classes. Overlaps are represented using pies and outliers are in white.*

4.4.2 Saccharomyces cerevisiae transcription network

Let us consider the *yeast* transcriptional regulatory network described in Chapter 4. The network is made of 197 vertices connected by 303 edges. We apply the VBEM algorithm for various number of classes $Q \in \{2, \dots, 8\}$. Moreover, for each value of Q, we consider 10 initializations of the matrix $\boldsymbol{\tau}$ and we select the best learnt model. The results of IL_{osbm} are presented in Figure 4.2. The criterion finds its maximum for $Q_{IL_{osbm}} = 6$ as expected. It retrieves the three regulations patterns (see Section 3.6.3) where each regulation pattern is made of a group of regulators and a group of regulated operons.

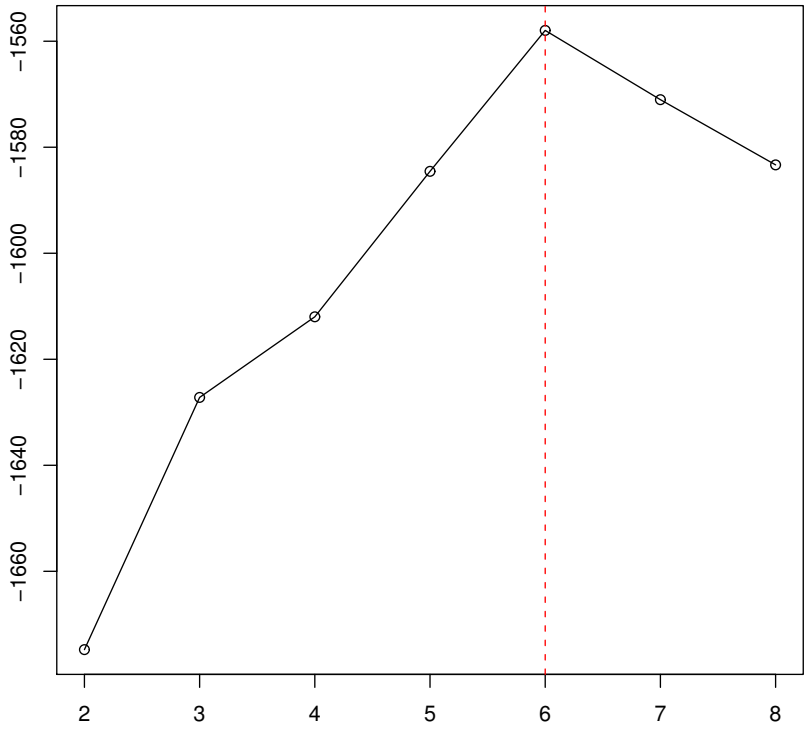

Figure 4.2 – *The IL_{osbm} criterion for $Q \in \{2, \dots, 8\}$. The maximum is reached at $Q_{IL_{osbm}} = 6$.*

Conclusion

In this chapter, we first introduced some conjugate prior distributions for the parameters of the overlapping stochastic block model. We then proposed a variational Bayes EM algorithm, based on global and local variational techniques. The algorithm can be used to approximate the posterior distribution over the model parameters and latent variables, given the observed data. In this framework, we derived a model selection criterion, so called IL_{osbm}, which is based on a non asymptotic approximation of the marginal log-likelihood. Using simulated data and a real network, we

showed that IL_{osbm} provides a relevant estimation of the number of overlapping clusters. In future work, we will assess the Bayesian confidence intervals of the posterior distributions found by the variational Bayes EM algorithm. Moreover, we are also interested in developing a method in order to be able to learn from a network whether the parameters $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ and W^* are zero or not.

Conclusion

As more and more network structured data sets are available, the statistical analysis of graphs has become common place. In this thesis, we considered unsupervised methods which aim at clustering vertices depending on their connection profiles. There is a long history of research on the topic, which goes back to the earlier work of Moreno in 1934, and many graph clustering algorithms have been proposed. We presented model based techniques and focused mainly on the Stochastic Block Model (SBM).

SBM is a mixture model which assumes that the vertices of a network are spread into different classes, so that the probability of an edge between two vertices only depends on the classes they belong to. No assumption is made on these probabilities such that very different topological structures can be taken into account. In particular, the model can characterize the presence of hubs which make networks locally dense. SBM was shown to generalize many of the existing graph clustering algorithms and so we considered it as the starting point of the thesis.

The clustering of vertices as well as the estimation of SBM parameters had been subject to previous work and numerous inference strategies had been proposed. However, the model still suffered from a lack of criteria to estimate the number of components in the mixture. Only one model based criterion had been derived for SBM in the literature. However, it tended to be too conservative in the case of small networks. In order to tackle this issue, we first illustrated how SBM could be described in a Bayesian framework. A new inference procedure was then proposed to estimate the posterior distribution over the model parameters and latent variables, given the observed data. In this framework, we derived a new model selection criterion, so-called ILvb, based on a non asymptotic approximation of the marginal likelihood. By considering networks generated using SBM as well as a real network, we showed that ILvb focus on the estimation of the data density and provides a relevant estimation of the number of latent classes.

Besides, almost all graph clustering models, such as SBM, partition the vertices into disjoint clusters. However, most real networks are known to be made of overlapping clusters. For instance, many proteins, so-called *moonlighting proteins*, are known to have several functions in the cells, and actors might belong to several groups of interest. Therefore, we introduced a new random graph model, the Overlapping Stochastic Block Model (OSBM). It allows the vertices to belong to multiple classes and can take very different topological structures into account. We proposed a variational EM algorithm, based on global and local variational techniques, to cluster the vertices of a network and estimate the model parameters. Using simulated data, the french political blogosphere network,

as well as the *yeast* transcriptional regulatory network, we showed that this procedure outperformed existing graph clustering algorithms.

Finally, we illustrated how OSBM could be described in a Bayesian framework by introducing some conjugate prior distribution for the model parameters. Then, we proposed an inference procedure to approximate the posterior distribution over the model parameters and latent variables, given the observed data. In this framework, we obtained the first model selection criterion for OSBM that we called $ILvbm$. It is based on a non asymptotic approximation of the marginal likelihood and has shown encouraging results.

Perspectives

In future work, we will investigate Markov chain Monte Carlo techniques as alternative approaches to estimate the posterior distribution of OSBM. These methods have already been considered for the (non overlapping) SBM model and have shown some interesting features. In particular, they appear to be less dependent on the initialization of the $\boldsymbol{\tau}$ matrix than variational approaches. Some other experiments on more challenging networks are also of interest in order to assess the $ILvbm$ criterion. Similarly, we believe it is crucial to investigate the Bayesian confidence intervals of the posterior distributions found by the variational Bayes EM algorithm. For now, we have only used the algorithm for model selection purposes and the posterior distributions should be better exploited. As mentioned already, it would be particularly relevant to develop a procedure to be able to learn from a network whether the parameters $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, and W^* are zero or not. Finally, recent studies have extended existing random graph models in order to take covariate information about the edges or vertices into account. Others have focused on modelling the dynamic of networks, that is the appearence of edges through time. We will investigate possible extensions of OSBM in order to take these features into account.

Mixture models

A

A.1 Factorization of the integrated complete-data likelihood

If the prior over the model parameters can be factorized, such that $p(\boldsymbol{\theta}) = p(\boldsymbol{\alpha})p(\boldsymbol{\kappa})$, then:

$$p(\mathbf{X}, \mathbf{Z}) = p(\mathbf{X} \mid \mathbf{Z})p(\mathbf{Z}).$$

Proof:

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}) &= \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\,\boldsymbol{\theta} \\ &= \int p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta})p(\mathbf{Z} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\,\boldsymbol{\theta} \\ &= \int p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\kappa})p(\mathbf{Z} \mid \boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\boldsymbol{\kappa})d\,\boldsymbol{\alpha}\,d\,\boldsymbol{\kappa} \\ &= \left(\int p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\kappa})p(\boldsymbol{\kappa})d\,\boldsymbol{\kappa}\right)\left(\int p(\mathbf{Z} \mid \boldsymbol{\alpha})p(\boldsymbol{\alpha})d\,\boldsymbol{\alpha}\right) \\ &= p(\mathbf{X} \mid \mathbf{Z})p(\mathbf{Z}). \end{aligned}$$

A.2 Exact expression of $\log p(\mathbf{Z})$

Using a Jeffreys non informative prior distribution for the class proportions $\boldsymbol{\alpha}$, Biernacki et al. (2000) showed that $\log p(\mathbf{Z})$ is given by:

$$\log p(\mathbf{Z}) = \log\Gamma(\frac{Q}{2}) + \sum_{q=1}^{Q} \log\Gamma(\frac{1}{2} + n_q) - Q\log\Gamma(\frac{1}{2}) - \log\Gamma(N + \frac{Q}{2}), \tag{A.1}$$

where $n_q = \sum_i^N Z_{iq}, \forall q$.

Proof: Let us first consider a Dirichlet prior distribution for the class proportions:

$$\begin{aligned} p(\boldsymbol{\alpha}) &= \mathrm{Dir}(\boldsymbol{\alpha};\, \mathbf{n}^0) \\ &= D(\mathbf{n}^{\,0}) \prod_{q=1}^{Q} \alpha_q^{n_q - 1}, \end{aligned}$$

with

$$D(\mathbf{n}^0) = \frac{\Gamma(\sum_q^Q n_q^0)}{\prod_{q=1}^Q \Gamma(n_q^0)}.$$

It leads to:

$$\begin{aligned}
\log p(\mathbf{Z}) &= \log\left(\int p(\mathbf{Z}\,|\,\boldsymbol{\alpha})p(\boldsymbol{\alpha})d\,\boldsymbol{\alpha}\right)\\
&= \log\left(\int \prod_{i=1}^{n}\prod_{q=1}^{Q}\alpha_q^{Z_{iq}} D(\mathbf{n}^0)\prod_{q=1}^{Q}\alpha_q^{n_q^0-1} d\,\boldsymbol{\alpha}\right)\\
&= \log\left(D(\mathbf{n}^0)\int \prod_{q=1}^{Q}\alpha_q^{\sum_{i=1}^{n} Z_{iq}}\prod_{q=1}^{Q}\alpha_q^{n_q^0-1} d\,\boldsymbol{\alpha}\right)\\
&= \log\left(D(\mathbf{n}^0)\int \prod_{q=1}^{Q}\alpha_q^{n_q^0+n_q-1} d\,\boldsymbol{\alpha}\right).
\end{aligned}$$

Therefore

$$\begin{aligned}
\log p(\mathbf{Z}) &= \log\left(D(\mathbf{n}^0)\int \prod_{q=1}^{Q}\alpha_q^{n_q^{new}-1} d\,\boldsymbol{\alpha}\right)\\
&= \log\left(\frac{D(\mathbf{n}^0)}{D(\mathbf{n}^{new})}\int D(\mathbf{n}^{new})\prod_{q=1}^{Q}\alpha_q^{n_q^{new}-1} d\,\boldsymbol{\alpha}\right)\\
&= \log\left(\frac{D(\mathbf{n}^0)}{D(\mathbf{n}^{new})}\underbrace{\int \mathrm{Dir}(\boldsymbol{\alpha};\,\mathbf{n}^{new})d\,\boldsymbol{\alpha}}_{1}\right) \qquad \text{(A.2)}\\
&= \log\frac{D(\mathbf{n}^0)}{D(\mathbf{n}^{new})}\\
&= \log\Gamma(\sum_{q=1}^{Q} n_q^0) + \sum_{q=1}^{Q}\log\Gamma(n_q^0+n_q) - \sum_{q=1}^{Q}\log\Gamma(n_q^0)\\
&\quad - \log\Gamma\left(\sum_{q=1}^{Q}(n_q^0+n_q)\right),
\end{aligned}$$

where $n_q = \sum_{i=1}^{n} Z_{iq}, \forall q$. In (A.2), $\mathbf{n}^{new}$ denotes the Q dimensional vector such that $n_q^{new} = n_q^0 + n_q, \forall q$. If we now consider a Jeffreys non informative prior distribution (Robert 1994) which corresponds to a Dirichlet distribution with $\mathbf{n}_q^0 = 1/2, \forall q$, we obtain (A.1).

A.3 Asymptotic approximation of $\log p(\mathbf{Z})$ using Stirling formulae

Assuming that N and the n_qs have large values (namely the α_q are far from 0), Biernacki et al. (2000) relied on the Stirling formulae to obtain an approximation of $\log p(\mathbf{Z})$:

$$\begin{aligned}
\log p(\mathbf{Z}) &\approx \max_{\boldsymbol{\alpha}} \log p(\mathbf{Z}\,|\,\boldsymbol{\alpha}) - \frac{Q-1}{2}\log N\\
&= \sum_{q=1}^{Q} n_q \log\frac{n_q}{N} - \frac{Q-1}{2}\log N,
\end{aligned}$$

where $n_q = 1/2, \forall q$.

Proof: For large values of s, the Stirling formula approximates the Gamma function:

$$\Gamma(s+1) \approx (2\pi)^{\frac{1}{2}} s^{s+\frac{1}{2}} \exp(-s). \tag{A.3}$$

Thus, using (A.3) in (A.1) and removing the terms in $O(1)$ (assuming $Q = O(N)$) leads to:

$$\begin{aligned}
\log p(\mathbf{Z}) &\approx \sum_{q=1}^{Q} \log \left((2\pi)^{\frac{1}{2}} (n_q - \frac{1}{2})^{n_q} \exp(-n_q + \frac{1}{2}) \right) \\
&\quad - \log \left((2\pi)^{\frac{1}{2}} (N + \frac{Q}{2} - 1)^{N + \frac{Q}{2} - \frac{1}{2}} \exp(-N - \frac{Q}{2} + \frac{1}{2}) \right) \\
&\approx \sum_{q=1}^{Q} \big(n_q \log n_q - n_q \big) \big) - N \log N - \frac{Q-1}{2} \log N + N \\
&= \sum_{q=1}^{Q} n_q \log n_q - N \log N - \frac{Q-1}{2} \log N \\
&= \sum_{q=1}^{Q} n_q \log \frac{n_q}{N} - \frac{Q-1}{2} \log N \\
&= \max_{\boldsymbol{\alpha}} \log p(\mathbf{Z} \,|\, \boldsymbol{\alpha}) - \frac{Q-1}{2} \log N.
\end{aligned}$$

Indeed, it is straightforward to see that the maximum of $\log p(\mathbf{Z} \,|\, \boldsymbol{\alpha})$ is reached when $\alpha_q = \frac{n_q}{N}, \forall q$. Therefore:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \log p(\mathbf{Z} \,|\, \boldsymbol{\alpha}) &= \sum_{i=1}^{N} \sum_{q=1}^{Q} Z_{iq} \log \frac{n_q}{N} \\
&= \sum_{q=1}^{Q} n_q \log \frac{n_q}{N}.
\end{aligned}$$

SBM

B

B.1 Optimization of $q(\mathbf{Z}_i)$

The optimal approximation at vertex i is:

$$q(\mathbf{Z}_i) = \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau_i} = \{\tau_{i1}, \ldots, \tau_{iQ}\}), \tag{B.1}$$

where τ_{iq} is the probability (responsability) of node i to belong to class q. It satisfies the relation:

$$\tau_{iq} \propto e^{\psi(n_q) - \psi(\sum_{l=1}^{Q} n_l)} \prod_{j \neq i}^{N} \prod_{l=1}^{Q} e^{\tau_{jl}\left(\psi(\zeta_{ql}) - \psi(\eta_{ql}+\zeta_{ql}) + X_{ij}\left(\psi(\eta_{ql}) - \psi(\zeta_{ql})\right)\right)}, \tag{B.2}$$

where $\psi(.)$ is the *digamma* function. In order to optimize the distribution $q(\mathbf{Z})$, we rely on a fixed point algorithm. Thus, given a matrix $\boldsymbol{\tau}^{old}$, the algorithm builds a new matrix $\boldsymbol{\tau}^{new}$ where each rows satisfies (B.2). After normalization, it then uses $\boldsymbol{\tau}^{new}$ to build a new matrix and so on. The algorithm stops when $\sum_{i=1}^{N}\sum_{q=1}^{Q} |\tau_{iq}^{old} - \tau_{iq}^{new}| < eps$. In the experiment section, we set $eps = 1e-6$.

Proof: According to variational Bayes, the optimal distribution $q(\mathbf{Z}_i)$ is given by:

$$\begin{aligned}
\log q(\mathbf{Z}_i) &= \mathrm{E}_{\mathbf{Z}^{\setminus i}, \boldsymbol{\alpha}, \boldsymbol{\Pi}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})] + \text{const} \\
&= \mathrm{E}_{\mathbf{Z}^{\setminus i}, \boldsymbol{\Pi}}[\log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\pi})] + \mathrm{E}_{\mathbf{Z}^{\setminus i}, \boldsymbol{\alpha}}[\log p(\mathbf{Z} \mid \boldsymbol{\alpha})] + \text{const} \\
&= \mathrm{E}_{\mathbf{Z}^{\setminus i}, \boldsymbol{\Pi}}[\sum_{i'<j} \sum_{q,l} Z_{i'q} Z_{jl} \Big(X_{i'j} \log \pi_{ql} + (1 - X_{i'j}) \log(1 - \pi_{ql})\Big)] \\
&\qquad + \mathrm{E}_{\mathbf{Z}^{\setminus i}, \boldsymbol{\alpha}}[\sum_{i'=1}^{N} \sum_{q=1}^{Q} Z_{i'q} \log \alpha_q] + \text{const} \\
&= \sum_{q=1}^{Q} Z_{iq} \Bigg(\mathrm{E}_{\alpha_q}[\log \alpha_q] + \sum_{j \neq i}^{N} \sum_{l=1}^{Q} \tau_{jl} \Big(X_{ij} \big(\mathrm{E}_{\pi_{ql}}[\log \pi_{ql}] - \mathrm{E}_{\pi_{ql}}[\log(1 - \pi_{ql})] \big) \\
&\qquad + \mathrm{E}_{\pi_{ql}}[\log(1 - \pi_{ql})] \Big) \Bigg) + \text{const} \\
&= \sum_{q=1}^{Q} Z_{iq} \Bigg(\psi(n_q) - \psi(\sum_{l=1}^{N} n_l) + \sum_{j \neq i}^{N} \sum_{l=1}^{Q} \tau_{jl} \Big(X_{ij} \big(\psi(\eta_{ql}) - \psi(\zeta_{ql}) \big) \\
&\qquad + \psi(\zeta_{ql}) - \psi(\eta_{ql} + \zeta_{ql}) \Big) \Bigg) + \text{const},
\end{aligned} \tag{B.3}$$

where $\mathbf{Z}^{\setminus i}$ denotes the class of all nodes except node i. We have used $\mathrm{E}_y[\log y] = \psi(a) - \psi(a+b)$ when $y \sim \mathrm{Beta}(y;\, a, b)$. Moreover, to simplify the calculations, the terms that do not depend on $\mathbf{Z}_i$ have been absorbed into the constant. Taking the exponential of (B.3) and after normalization, we obtain the multinomial distribution (B.1).

B.2 Optimization of $q(\boldsymbol{\alpha})$

The optimization of the lower bound with respect to $q(\boldsymbol{\alpha})$ produces a distribution with the same functional form as the prior $p(\boldsymbol{\alpha})$:

$$q(\boldsymbol{\alpha}) = \mathrm{Dir}(\alpha;\, \mathbf{n}), \tag{B.4}$$

where

$$n_q = n_q^0 + \sum_{i=1}^{N} \tau_{iq}.$$

Proof: According to variational Bayes, the optimal distribution $q(\boldsymbol{\alpha})$ is given by:

$$\begin{aligned}
\log q(\boldsymbol{\alpha}) &= \mathrm{E}_{\mathbf{Z},\boldsymbol{\Pi}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})] + \text{const} \\
&= \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{Z} \mid \boldsymbol{\alpha})] + \log p(\boldsymbol{\alpha}) + \text{const} \\
&= \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \alpha_q + \sum_{q=1}^{Q} (n_q^0 - 1) \log \alpha_q + \text{const} \\
&= \sum_{q=1}^{Q} \left(n_q^0 - 1 + \sum_{i=1}^{N} \tau_{iq} \right) \log \alpha_q + \text{const}.
\end{aligned} \tag{B.5}$$

Taking the exponential of (B.5) and after normalization, we obtain the Dirichlet distribution (B.4).

B.3 Optimization of $q(\boldsymbol{\Pi})$

Again, the functional form of the prior $p(\boldsymbol{\Pi})$ is conserved through the variational optimization:

$$q(\boldsymbol{\Pi}) = \prod_{q \le l}^{Q} \mathrm{Beta}(\pi_{ql};\, \eta_{ql}, \zeta_{ql}), \tag{B.6}$$

For $q \neq l$, the hyperparameter η_{ql} is given by:

$$\eta_{ql} = \eta_{ql}^0 + \sum_{i \neq j}^{N} X_{ij} \tau_{iq} \tau_{jl},$$

and $\forall q$

$$\eta_{qq} = \eta_{qq}^0 + \sum_{i < j}^{N} X_{ij} \tau_{iq} \tau_{jq}.$$

Moreover, for $q \neq l$, the hyperparameter ζ_{ql} is given by:

$$\zeta_{ql} = \zeta_{ql}^0 + \sum_{i \neq j}^{N} (1 - X_{ij}) \tau_{iq} \tau_{jl},$$

and $\forall q$

$$\zeta_{qq} = \zeta_{qq}^0 + \sum_{i<j}^{N} (1 - X_{ij}) \tau_{iq} \tau_{jq}.$$

Proof : According to variational Bayes, the optimal distribution $q(\mathbf{\Pi})$ is given by:

$$\begin{aligned}
\log q(\mathbf{\Pi}) &= \mathrm{E}_{\mathbf{Z},\boldsymbol{\alpha}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \mathbf{\Pi})] + \text{const} \\
&= \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{X} \mid \mathbf{Z}, \mathbf{\Pi})] + \log p(\mathbf{\Pi}) + \text{const} \\
&= \sum_{i<j}^{N} \sum_{q,l}^{Q} \tau_{iq} \tau_{jl} \Big(X_{ij} \log \pi_{ql} + (1 - X_{ij}) \log(1 - \pi_{ql}) \Big) \\
&\qquad + \sum_{q \leq l}^{Q} \Big((\eta_{ql}^0 - 1) \log \pi_{ql} + (\zeta_{ql}^0 - 1) \log(1 - \pi_{ql}) \Big) + \text{const} \\
&= \sum_{q<l}^{Q} \sum_{i \neq j}^{N} \tau_{iq} \tau_{jl} \Big(X_{ij} \log \pi_{ql} + (1 - X_{ij}) \log(1 - \pi_{ql}) \Big) \\
&\qquad + \sum_{q=1}^{Q} \sum_{i<j}^{N} \tau_{iq} \tau_{jq} \Big(X_{ij} \log \pi_{qq} + (1 - X_{ij}) \log(1 - \pi_{qq}) \Big) \\
&\qquad + \sum_{q \leq l}^{Q} \Big((\eta_{ql}^0 - 1) \log \pi_{ql} + (\zeta_{ql}^0 - 1) \log(1 - \pi_{ql}) \Big) + \text{const} \\
&= \sum_{q<l}^{Q} \left(\Big(\eta_{ql}^0 - 1 + \sum_{i \neq j}^{N} \tau_{iq} \tau_{jl} X_{ij} \Big) \log \pi_{ql} + \Big(\zeta_{ql}^0 - 1 + \sum_{i \neq j}^{N} \tau_{iq} \tau_{jl} (1 - X_{ij}) \Big) \log(1 - \pi_{ql}) \right) \\
&\quad + \sum_{q=1}^{Q} \left(\Big(\eta_{qq}^0 - 1 + \sum_{i<j}^{N} \tau_{iq} \tau_{jq} X_{ij} \Big) \log \pi_{qq} + \Big(\zeta_{qq}^0 - 1 + \sum_{i<j}^{N} \tau_{iq} \tau_{jq} (1 - X_{ij}) \Big) \log(1 - \pi_{qq}) \right).
\end{aligned} \tag{B.7}$$

Taking the exponential of (B.7) and after normalization, we obtain the product of Beta distribution (B.6).

B.4 Lower bound

The lower bound takes a simple form after the variational Bayes M-step. Indeed, it only depends on the posterior probabilities τ_{iq} as well as the normalizing constants of the Dirichlet and Beta distributions:

$$\mathcal{L}(q) = \log \left\{ \frac{\Gamma(\sum_{q=1}^{Q} n_q^0) \prod_{q=1}^{Q} \Gamma(n_q)}{\Gamma(\sum_{q=1}^{Q} n_q) \prod_{q=1}^{Q} \Gamma(n_q^0)} \right\} + \sum_{q \leq l}^{Q} \log \left\{ \frac{\Gamma(\eta_{ql}^0 + \zeta_{ql}^0) \Gamma(\eta_{ql}) \Gamma(\zeta_{ql})}{\Gamma(\eta_{ql} + \zeta_{ql}) \Gamma(\eta_{ql}^0) \Gamma(\zeta_{ql}^0)} \right\} - \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \tau_{iq}.$$

Proof : The lower bound is given by:

$$
\begin{aligned}
\mathcal{L}(q) &= \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi}) \log\{\frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})}\} d\,\boldsymbol{\alpha}\, d\,\boldsymbol{\pi} \\
&= \mathrm{E}_{\mathbf{Z},\boldsymbol{\alpha},\boldsymbol{\Pi}}[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})] - \mathrm{E}_{\mathbf{Z},\boldsymbol{\alpha},\boldsymbol{\Pi}}[\log q(\mathbf{Z}, \boldsymbol{\alpha}, \boldsymbol{\Pi})] \\
&= \mathrm{E}_{\mathbf{Z},\boldsymbol{\Pi}}[\log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\Pi})] + \mathrm{E}_{\mathbf{Z},\boldsymbol{\alpha}}[\log p(\mathbf{Z} \mid \boldsymbol{\alpha})] + \mathrm{E}_{\boldsymbol{\alpha}}[\log p(\boldsymbol{\alpha})] + \mathrm{E}_{\boldsymbol{\Pi}}[\log p(\boldsymbol{\Pi})] \\
&\qquad - \sum_{i=1}^{N} \mathrm{E}_{\mathbf{Z}_i}[\log q(\mathbf{Z}_i)] - \mathrm{E}_{\boldsymbol{\alpha}}[\log q(\boldsymbol{\alpha})] - \mathrm{E}_{\boldsymbol{\Pi}}[\log q(\boldsymbol{\pi})] \\
&= \sum_{i<j}^{N}\sum_{q,l}^{Q} \tau_{iq}\tau_{jl}\Bigg(X_{ij}\Big(\psi(\eta_{ql}) - \psi(\zeta_{ql})\Big) + \psi(\zeta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Bigg) \\
&\qquad + \sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq}\Big(\psi(n_q) - \psi(\sum_{l=1}^{Q} n_l)\Big) + \log\Gamma(\sum_{q=1}^{Q} n_q^0) - \sum_{q=1}^{Q}\log\Gamma(n_q^0) \\
&\qquad + \sum_{q=1}^{Q}\Big(n_q^0 - 1\Big)\Big(\psi(n_q) - \psi(\sum_{l=1}^{Q} n_l)\Big) + \sum_{q\le l}^{Q}\Bigg(\log\Gamma(\eta_{ql}^0 + \zeta_{ql}^0) \\
&\qquad - \log\Gamma(\eta_{ql}^0) - \log\Gamma(\zeta_{ql}^0) + (\eta_{ql}^0 - 1)\Big(\psi(\eta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Big) \\
&\qquad + (\zeta_{ql}^0 - 1)\Big(\psi(\zeta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Big)\Bigg) - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\log\tau_{iq} \\
&\qquad - \log\Gamma(\sum_{q=1}^{Q} n_q) + \sum_{q=1}^{Q}\log\Gamma(n_q) - \sum_{q=1}^{Q}\Big(n_q - 1\Big)\Big(\psi(n_q) - \psi(\sum_{l=1}^{Q} n_l)\Big) \\
&\qquad - \sum_{q\le l}^{Q}\Bigg(\log\Gamma(\eta_{ql} + \zeta_{ql}) - \log\Gamma(\eta_{ql}) - \log\Gamma(\zeta_{ql}) \\
&\qquad + (\eta_{ql} - 1)\Big(\psi(\eta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Big) + (\zeta_{ql} - 1)\Big(\psi(\zeta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Big)\Bigg) \\
&= \sum_{q<l}^{Q}\Bigg(\Big(\eta_{ql}^0 - \eta_{ql} + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}X_{ij}\Big)\Big(\psi(\eta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Big) \\
&\qquad + \Big(\zeta_{ql}^0 - \zeta_{ql} + \sum_{i\neq j}^{N}\tau_{iq}\tau_{jl}(1 - X_{ij})\Big)\Big(\psi(\zeta_{ql}) - \psi(\eta_{ql} + \zeta_{ql})\Big)\Bigg) \\
&\qquad + \sum_{q=1}^{Q}\Bigg(\Big(\eta_{qq}^0 - \eta_{qq} + \sum_{i<j}^{N}\tau_{iq}\tau_{jq}X_{ij}\Big)\Big(\psi(\eta_{qq}) - \psi(\eta_{qq} + \zeta_{qq})\Big) \\
&\qquad + \Big(\zeta_{qq}^0 - \zeta_{qq} + \sum_{i<j}^{N}\tau_{iq}\tau_{jq}(1 - X_{ij})\Big)\Big(\psi(\zeta_{qq}) - \psi(\eta_{qq} + \zeta_{qq})\Big)\Bigg) \\
&\qquad + \sum_{q=1}^{Q}\Big(n_q^0 - n_q + \sum_{i=1}^{N}\tau_{iq}\Big)\Big(\psi(n_q) - \psi(\sum_{l=1}^{Q} n_l)\Big) \\
&\qquad + \log\{\frac{\Gamma(\sum_{q=1}^{Q} n_q^0)\prod_{q=1}^{Q}\Gamma(n_q)}{\Gamma(\sum_{q=1}^{Q} n_q)\prod_{q=1}^{Q}\Gamma(n_q^0)}\} + \sum_{q\le l}^{Q}\log\{\frac{\Gamma(\eta_{ql}^0 + \zeta_{ql}^0)\Gamma(\eta_{ql})\Gamma(\zeta_{ql})}{\Gamma(\eta_{ql}) + \zeta_{ql}\Gamma(\eta_{ql}^0)\Gamma(\zeta_{ql}^0)}\} \\
&\qquad - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\log\tau_{iq}.
\end{aligned}
\tag{B.8}
$$

After the variational Bayes M-step, most of the terms in the lower bound vanish since:

- $\forall q \ : n_q = n_q^0 + \sum_{i=1}^{N} \tau_{iq}$.
- $\forall q \neq l \ : \eta_{ql} = \eta_{ql}^0 + \sum_{i \neq j}^{N} X_{ij} \tau_{iq} \tau_{jl}$,
- $\forall q \ : \eta_{qq} = \eta_{qq}^0 + \sum_{i<j}^{N} X_{ij} \tau_{iq} \tau_{jq}$.
- $\forall q \neq l \ : \zeta_{ql} = \zeta_{ql}^0 + \sum_{i \neq j}^{N} (1 - X_{ij}) \tau_{iq} \tau_{jl}$,
- $\forall q \ : \zeta_{qq} = \zeta_{qq}^0 + \sum_{i<j}^{N} (1 - X_{ij}) \tau_{iq} \tau_{jq}$.

Only the terms depending on the probabilities τ_{iq} and the normalizing constants of the Dirichlet and Beta distributions remain.

OSBM

C

C.1 First lower bound

The lower bound defined in (3.11) can be written:

$$\begin{aligned}\mathcal{L}_{ML}(q;\,\boldsymbol{\alpha},\tilde{\mathbf{W}}) = \sum_{i\neq j}^{N}\Big\{ X_{ij}\tilde{\boldsymbol{\tau}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j + \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})]\Big\} \\ + \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\alpha_q + (1-\tau_{iq})\log(1-\alpha_q)\right\} \\ - \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\tau_{iq} + (1-\tau_{iq})\log(1-\tau_{iq})\right\}.\end{aligned}$$

Proof: The lower bound can be decoposed into:

$$\begin{aligned}\mathcal{L}_{ML}(q;\,\boldsymbol{\alpha},\tilde{\mathbf{W}}) &= \sum_{\mathbf{Z}} q(\mathbf{Z})\log p(\mathbf{X},\mathbf{Z}\,|\,\boldsymbol{\alpha},\tilde{\mathbf{W}}) - \sum_{\mathbf{Z}} q(\mathbf{Z})\log q(\mathbf{Z}) \\ &= \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{X},\mathbf{Z}\,|\,\boldsymbol{\alpha},\tilde{\mathbf{W}})] - \mathrm{E}_{\mathbf{Z}}[\log q(\mathbf{Z})] \\ &= \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{X}\,|\,\mathbf{Z},\tilde{\mathbf{W}})] + \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{Z}\,|\,\boldsymbol{\alpha})] - \mathrm{E}_{\mathbf{Z}}[\log q(\mathbf{Z})],\end{aligned} \tag{C.1}$$

where the expectations are taken according to the distribution $q(\mathbf{Z})$ and the last term of (C.1) is an entropy term. Using (3.13), we obtain:

$$\begin{aligned}\mathcal{L}_{ML}(q;\,\boldsymbol{\alpha},\tilde{\mathbf{W}}) &= \sum_{i\neq j}^{N}\Big\{ X_{ij}\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a_{ij}] + \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})]\Big\} \\ &\quad + \sum_{i=1}^{N}\sum_{q=1}^{Q}\Big\{\mathrm{E}_{Z_{iq}}[Z_{iq}]\log\alpha_q + (1-\mathrm{E}_{Z_{iq}}[Z_{iq}])\log(1-\alpha_q)\Big\} \\ &\quad - \sum_{i=1}^{N}\sum_{q=1}^{Q}\Big\{\mathrm{E}_{Z_{iq}}[Z_{iq}]\log\tau_{iq} + (1-\mathrm{E}_{Z_{iq}}[Z_{iq}])\log(1-\tau_{iq})\Big\} \\ &= \sum_{i\neq j}^{N}\Big\{ X_{ij}\tilde{\boldsymbol{\tau}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j + \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})]\Big\} \\ &\quad + \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\alpha_q + (1-\tau_{iq})\log(1-\alpha_q)\right\} \\ &\quad - \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\tau_{iq} + (1-\tau_{iq})\log(1-\tau_{iq})\right\}.\end{aligned} \tag{C.2}$$

C.2 Second lower bound

Using local variational approximations, a tractable lower bound can be obtained:

$$\begin{aligned}\mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}) = \sum_{i\neq j}^{N}\Bigg\{ &(X_{ij}-\frac{1}{2})\,\tilde{\boldsymbol{\tau}}_i^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j + \log g(\xi_{ij}) - \frac{\xi_{ij}}{2} \\ &- \lambda(\xi_{ij})\left(\mathrm{Tr}\left(\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\,\boldsymbol{\Sigma}_j\right) + \tilde{\boldsymbol{\tau}}_j^{\intercal}\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j - \xi_{ij}^2\right)\Bigg\} \\ &+ \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\alpha_q + (1-\tau_{iq})\log(1-\alpha_q)\right\} \\ &- \sum_{i=1}^{N}\sum_{q=1}^{Q}\left\{\tau_{iq}\log\tau_{iq} + (1-\tau_{iq})\log(1-\tau_{iq})\right\},\end{aligned}$$

where $\tilde{\mathbf{E}}_i = \mathrm{E}_{\mathbf{Z}_i}[\tilde{\mathbf{Z}}_i\tilde{\mathbf{Z}}_i{}^{\intercal}] = \boldsymbol{\Sigma}_i + \tilde{\boldsymbol{\tau}}_i\tilde{\boldsymbol{\tau}}_i{}^{\intercal}$ and:

$$\boldsymbol{\Sigma}_i = \begin{pmatrix}\mathrm{var}(\mathbf{Z}_i) & \mathbf{0}\\ \mathbf{0} & 0\end{pmatrix},\ \forall i.$$

Proof: As noticed in Section 3.5 the first lower bound is a function of the expectations $\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})]$ which are untractable. In order to compute a second tractable lower bound, we consider the bound $\log g(x,\xi)$ on the log-logistic function:

$$\log g(x) \geq \log g(x,\xi) = \log g(\xi) + \frac{(x-\xi)}{2} - \lambda(\xi)(x^2-\xi^2),\ \ \forall x,\xi\in\mathbb{R}, \tag{C.3}$$

where $\lambda(\xi) = \frac{1}{4\xi}\tanh(\frac{\xi}{2}) = \frac{1}{2\xi}\left\{g(\xi)-\frac{1}{2}\right\}$ and ξ is a variational parameter. This bound was first introduced by Jaakkola and Jordan (2000), in the framework of Bayesian logistic regression, to obtain a tractable approximation of the marginal likelihood. It is based on symmetrization of the log-logistic function and a Taylor expansion in the variable x^2. It leads to:

$$\log g(-a_{ij}) = \log g(-\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j) \geq \log g(-\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j,\xi_{ij}),$$

where

$$\log g(-\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j,\xi_{ij}) = \log g(\xi_{ij}) - \frac{(\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j+\xi_{ij})}{2} - \lambda(\xi_{ij})\Big((\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j)^2 - \xi_{ij}^2\Big).$$

Therefore, we have:

$$\begin{aligned}\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\log g(-a_{ij})] &= \sum_{\mathbf{Z}_i,\mathbf{Z}_j\in\{0,1\}^Q}\log g(-a_{ij})q(\mathbf{Z}_i)q(\mathbf{Z}_j) \\ &\geq \sum_{\mathbf{Z}_i,\mathbf{Z}_j\in\{0,1\}^Q}\Big\{\log g(\xi_{ij}) - \frac{(\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j+\xi_{ij})}{2} - \lambda(\xi_{ij})\Big((\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j)^2 \\ &\qquad - \xi_{ij}^2\Big)\Big\}q(\mathbf{Z}_i)q(\mathbf{Z}_j) \\ &\geq \log g(\xi_{ij}) - \frac{(\tilde{\boldsymbol{\tau}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j+\xi_{ij})}{2} - \lambda(\xi_{ij})\Big(\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{Z}}_i{}^{\intercal}\tilde{\mathbf{W}}\tilde{\mathbf{Z}}_j)^2] - \xi_{ij}^2\Big).\end{aligned}$$

The expectation terms are now tractable:

$$\begin{aligned}
\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{Z}}_i{}^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j)^2] &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\tilde{\mathbf{Z}}_i{}^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j \tilde{\mathbf{Z}}_i{}^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j] \\
&= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\tilde{\mathbf{Z}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \tilde{\mathbf{Z}}_i \tilde{\mathbf{Z}}_i{}^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j] \\
&= \mathrm{E}_{\mathbf{Z}_j}[\tilde{\mathbf{Z}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \mathrm{E}_{\mathbf{Z}_i}[\tilde{\mathbf{Z}}_i \tilde{\mathbf{Z}}_i{}^\intercal] \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j] \\
&= \mathrm{E}_{\mathbf{Z}_j}[\tilde{\mathbf{Z}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \big(\boldsymbol{\Sigma}_i + \tilde{\boldsymbol{\tau}}_i \tilde{\boldsymbol{\tau}}_i{}^\intercal \big) \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j] \\
&= \mathrm{Tr}\Big(\tilde{\mathbf{W}}^\intercal \big(\boldsymbol{\Sigma}_i + \tilde{\boldsymbol{\tau}}_i \tilde{\boldsymbol{\tau}}_i{}^\intercal \big) \tilde{\mathbf{W}} \boldsymbol{\Sigma}_j \Big) + \tilde{\boldsymbol{\tau}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \big(\boldsymbol{\Sigma}_i + \tilde{\boldsymbol{\tau}}_i \tilde{\boldsymbol{\tau}}_i{}^\intercal \big) \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j,
\end{aligned}$$

where

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} \mathrm{var}(\mathbf{Z}_i) & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix}, \ \forall i.$$

We have used the property that $\forall\, \mathbf{A}$ a matrix,

$$\mathrm{E}[\tilde{\mathbf{Z}}_j{}^\intercal \, \mathbf{A} \, \tilde{\mathbf{Z}}_j] = \mathrm{Tr}\big(\mathbf{A} \, \mathrm{var}(\tilde{\mathbf{Z}}_j) \big) + \mathrm{E}[\tilde{\mathbf{Z}}_j]^\intercal \, \mathbf{A} \, \mathrm{E}[\tilde{\mathbf{Z}}_j].$$

In the following, and in order to simplify the notations, we denote:

$$\tilde{\mathbf{E}}_i = \mathrm{E}_{\mathbf{Z}_i}[\tilde{\mathbf{Z}}_i \tilde{\mathbf{Z}}_i{}^\intercal] = \boldsymbol{\Sigma}_i + \tilde{\boldsymbol{\tau}}_i \tilde{\boldsymbol{\tau}}_i{}^\intercal.$$

Thus:

$$\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{Z}}_i{}^\intercal \tilde{\mathbf{W}} \tilde{\mathbf{Z}}_j)^2] = \mathrm{Tr}\Big(\tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \boldsymbol{\Sigma}_j \Big) + \tilde{\boldsymbol{\tau}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j.$$

We eventually get the expression of a tractable second lower bound:

$$\begin{aligned}
\mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) = \sum_{i \neq j}^{N} \Bigg\{ & \big(X_{ij} - \frac{1}{2} \big) \tilde{\boldsymbol{\tau}}_i^\intercal \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j + \log g(\xi_{ij}) - \frac{\xi_{ij}}{2} \\
& - \lambda(\xi_{ij}) \Bigg(\mathrm{Tr}\Big(\tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \boldsymbol{\Sigma}_j \Big) + \tilde{\boldsymbol{\tau}}_j^\intercal \tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j - \xi_{ij}^2 \Bigg) \Bigg\} \\
& + \sum_{i=1}^{N} \sum_{q=1}^{Q} \big\{ \tau_{iq} \log \alpha_q + (1 - \tau_{iq}) \log(1 - \alpha_q) \big\} \\
& - \sum_{i=1}^{N} \sum_{q=1}^{Q} \big\{ \tau_{iq} \log \tau_{iq} + (1 - \tau_{iq}) \log(1 - \tau_{iq}) \big\}.
\end{aligned}$$

with

$$\log p(\mathbf{X} \,|\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \geq \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \geq \mathcal{L}_{ML}(q;\, \boldsymbol{\alpha}, \tilde{\mathbf{W}}, \boldsymbol{\xi}).$$

C.3 Optimization of ξ_{ij}

An estimate of ξ_{ij} is given by:

$$\hat{\xi_{ij}} = \sqrt{\Big(\mathrm{Tr}\Big(\tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \boldsymbol{\Sigma}_j \Big) + \tilde{\boldsymbol{\tau}}_j{}^\intercal \tilde{\mathbf{W}}^\intercal \tilde{\mathbf{E}}_i \tilde{\mathbf{W}} \tilde{\boldsymbol{\tau}}_j \Big)}.$$

Proof: The partial derivative of the lower bound, with respect to ξ_{ij}, is given by:

$$\begin{aligned}\frac{\partial \mathcal{L}_{ML}}{\partial \xi_{ij}}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}) &= g(-\xi_{ij}) - \frac{1}{2} - \lambda^{'}(\xi_{ij})\left(\mathrm{Tr}\Big(\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\boldsymbol{\Sigma}_j\Big) + \tilde{\boldsymbol{\tau}}_j{}^{\intercal}\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j - \xi_{ij}^2\right) \\ &\quad + 2\xi_{ij}\lambda(\xi_{ij}) \\ &= -\lambda^{'}(\xi_{ij})\left(\mathrm{Tr}\Big(\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\boldsymbol{\Sigma}_j\Big) + \tilde{\boldsymbol{\tau}}_j{}^{\intercal}\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j - \xi_{ij}^2\right),\end{aligned} \tag{C.4}$$

where we have used the property that $(\log g)^{'}(\xi_{ij}) = g(-\xi_{ij})$ and $g(\xi_{ij}) + g(-\xi_{ij}) = 1$. Since each bound $\log g(-a_{ij},\xi_{ij})$ is an even function with respect to ξ_{ij}, we can consider only positive values of ξ_{ij} without loss of generality. Therefore, we have $\lambda^{'}(\xi_{ij}) \neq 0$ since $\lambda(\xi_{ij})$ is a strictly decreasing function on this domain. Finally, if we set the derivative (C.4) of the lower bound to zero, we obtain:

$$\hat{\xi_{ij}}^2 = \mathrm{Tr}\Big(\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\boldsymbol{\Sigma}_j\Big) + \tilde{\boldsymbol{\tau}}_j{}^{\intercal}\tilde{\mathbf{W}}^{\intercal}\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j.$$

C.4 Optimization of α_q

An estimate of α_q is given by:

$$\hat{\alpha_q} = \frac{\sum_{i=1}^{N}\tau_{iq}}{N}.$$

Proof: If we set the partial derivative of the lower bound, with respect to α_q, to zero, we obtain:

$$\frac{\partial \mathcal{L}_{ML}}{\partial \alpha_q}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}) = \sum_{i=1}^{N}\{\frac{\tau_{iq}}{\alpha_q} - (\frac{1-\tau_{iq}}{1-\alpha_q})\} = 0.$$

Thus,

$$(1-\alpha_q)\sum_{i=1}^{N}\tau_{iq} = \alpha_q\sum_{i=1}^{N}(1-\tau_{iq}).$$

This leads to

$$\sum_{i=1}^{N}\tau_{iq} = \alpha_q N,$$

and

$$\hat{\alpha_q} = \frac{\sum_{i=1}^{N}\tau_{iq}}{N}.$$

C.5 Optimization of $\tilde{\mathbf{W}}$

An estimate of $\mathrm{vec}(\tilde{\mathbf{W}})$ is given by:

$$\mathrm{vec}(\tilde{\mathbf{W}}) = \left\{2\sum_{i\neq j}^{N}\lambda(\xi_{ij})\big(\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i\big)\right\}^{-1}\left\{\sum_{i\neq j}^{N}(X_{ij} - \frac{1}{2})\big(\tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i\big)\right\},$$

where vec denotes an operator which stacks the columns of a matrix into a vector.

Proof: The gradient of the lower bound with respect to the matrix $\tilde{\mathbf{W}}$ is given by:

$$\nabla_{\tilde{\mathbf{W}}}\mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}) = \sum_{i\neq j}^{N}\left\{(X_{ij}-\frac{1}{2})\tilde{\boldsymbol{\tau}}_i\tilde{\boldsymbol{\tau}}_j{}^{\mathsf{T}} - 2\lambda(\xi_{ij})\left(\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\boldsymbol{\Sigma}_j + \tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\boldsymbol{\tau}}_j\tilde{\boldsymbol{\tau}}_j{}^{\mathsf{T}}\right)\right\},$$

since $\forall\,\mathbf{B},\mathbf{C}$ symmetric matrices:

$$\nabla_{\tilde{\mathbf{W}}}\mathrm{Tr}(\tilde{\mathbf{W}}^{\mathsf{T}}\,\mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{C}) = \mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{C} + \mathbf{B}^{\mathsf{T}}\,\tilde{\mathbf{W}}\,\mathbf{C}^{\mathsf{T}} = 2\,\mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{C},$$

and $\forall\,\mathbf{b}$ a vector:

$$\nabla_{\tilde{\mathbf{W}}}\,\mathbf{b}^{\mathsf{T}}\,\tilde{\mathbf{W}}^{\mathsf{T}}\,\mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{b} = \mathbf{B}^{\mathsf{T}}\,\tilde{\mathbf{W}}\,\mathbf{b}\,\mathbf{b}^{\mathsf{T}} + \mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{b}\,\mathbf{b}^{\mathsf{T}} = 2\,\mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{b}\,\mathbf{b}^{\mathsf{T}}\,.$$

Finally, we obtain:

$$\nabla_{\tilde{\mathbf{W}}}\mathcal{L}_{ML}(q;\boldsymbol{\alpha},\tilde{\mathbf{W}},\boldsymbol{\xi}) = \sum_{i\neq j}^{N}\left\{(X_{ij}-\frac{1}{2})\tilde{\boldsymbol{\tau}}_i\tilde{\boldsymbol{\tau}}_j{}^{\mathsf{T}} - 2\lambda(\xi_{ij})\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\mathbf{E}}_j\right\}.$$

Therefore, the matrix $\tilde{\mathbf{W}}$ which maximizes the lower bound satisfies:

$$2\sum_{i\neq j}^{N}\left\{\lambda(\xi_{ij})\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\mathbf{E}}_j\right\} = \sum_{i\neq j}^{N}\left\{(X_{ij}-\frac{1}{2})\tilde{\boldsymbol{\tau}}_i\tilde{\boldsymbol{\tau}}_j{}^{\mathsf{T}}\right\}.$$

This implies:

$$\mathrm{vec}\Big\{2\sum_{i\neq j}^{N}\lambda(\xi_{ij})\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\mathbf{E}}_j\Big\} = \mathrm{vec}\Big\{\sum_{i\neq j}^{N}(X_{ij}-\frac{1}{2})\tilde{\boldsymbol{\tau}}_i\tilde{\boldsymbol{\tau}}_j{}^{\mathsf{T}}\Big\},$$

and

$$2\sum_{i\neq j}^{N}\lambda(\xi_{ij})\mathrm{vec}\big(\tilde{\mathbf{E}}_i\tilde{\mathbf{W}}\tilde{\mathbf{E}}_j\big) = \sum_{i\neq j}^{N}(X_{ij}-\frac{1}{2})\mathrm{vec}(\tilde{\boldsymbol{\tau}}_i\tilde{\boldsymbol{\tau}}_j{}^{\mathsf{T}}). \tag{C.5}$$

From (C.5), we obtain:

$$2\sum_{i\neq j}^{N}\lambda(\xi_{ij})\big(\tilde{\mathbf{E}}_j\otimes\tilde{\mathbf{E}}_i\big)\mathrm{vec}(\tilde{\mathbf{W}}) = \sum_{i\neq j}^{N}(X_{ij}-\frac{1}{2})\big(\tilde{\boldsymbol{\tau}}_j\otimes\tilde{\boldsymbol{\tau}}_i\big),$$

since $\tilde{\mathbf{E}}_j$ is a symmetric matrix and $\forall\,\mathbf{B},\mathbf{C}$ two matrices:

$$\mathrm{vec}\big(\,\mathbf{B}\,\tilde{\mathbf{W}}\,\mathbf{C}\,\big) = \big(\,\mathbf{C}^{\mathsf{T}}\otimes\mathbf{B}\,\big)\mathrm{vec}(\tilde{\mathbf{W}}).$$

Moreover $\forall\,\mathbf{b},\mathbf{c}$ two vectors:

$$\mathrm{vec}\big(\,\mathbf{c}\,\mathbf{b}^{\mathsf{T}}\,\big) = \mathbf{b}\otimes\mathbf{c}\,.$$

Therefore an estimate of $\mathrm{vec}(\tilde{\mathbf{W}})$ is given by:

$$\mathrm{vec}(\tilde{\mathbf{W}}) = \left\{2\sum_{i\neq j}^{N}\lambda(\xi_{ij})\big(\tilde{\mathbf{E}}_j\otimes\tilde{\mathbf{E}}_i\big)\right\}^{-1}\left\{\sum_{i\neq j}^{N}(X_{ij}-\frac{1}{2})\big(\tilde{\boldsymbol{\tau}}_j\otimes\tilde{\boldsymbol{\tau}}_i\big)\right\}.$$

Bayesian OSBM

D

D.1 Lower Bound

Given a $N \times N$ positive real matrix $\boldsymbol{\xi}$, a lower bound of the first lower bound can be computed:

$$\log p(\mathbf{X}) \geq \mathcal{L}(q) \geq \mathcal{L}(q; \boldsymbol{\xi}),$$

where

$$\mathcal{L}(q; \boldsymbol{\xi}) = \sum_{\mathbf{Z}} \int \int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log \left(\frac{h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) p(\mathbf{Z} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\tilde{\mathbf{W}})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})} \right) d\boldsymbol{\alpha}\, d\tilde{\mathbf{W}},$$

and

$$\log h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) = \sum_{i \neq j}^{N} \left\{ (X_{ij} - \frac{1}{2}) a_{\mathbf{Z}_i, \mathbf{Z}_j} - \frac{\xi_{ij}}{2} + \log g(\xi_{ij}) - \lambda(\xi_{ij})(a^2_{\mathbf{Z}_i, \mathbf{Z}_j} - \xi^2_{ij}) \right\}.$$

Proof: Let us start by showing that:

$$\log p(\mathbf{X} \mid \mathbf{Z}, \tilde{\mathbf{W}}) \geq \log h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}),$$

where $\boldsymbol{\xi}$ is an $N \times N$ positive real matrix. We use the bound on the log-logistic function introduced by Jaakkola and Jordan (2000):

$$\log g(x) \geq \log g(\xi) + \frac{x - \xi}{2} - \lambda(\xi)(x^2 - \xi^2), \forall (x, \xi) \in \mathbb{R} \times \mathbb{R}^+, \quad \text{(D.1)}$$

where $\lambda(\xi) = (g(\xi) - 1/2)/(2\xi)$. Note that (D.1) is an even function and therefore we can consider only positive values of x without loss of generality. Since

$$\log p(X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \tilde{\mathbf{W}}) = X_{ij} a_{\mathbf{Z}_i, \mathbf{Z}_j} + \log g(-a_{\mathbf{Z}_i, \mathbf{Z}_j}),$$

then

$$\begin{aligned} \log p(X_{ij} | \, \mathbf{Z}_i, \mathbf{Z}_j, \tilde{\mathbf{W}}) &\geq X_{ij} a_{\mathbf{Z}_i, \mathbf{Z}_j} + \log g(\xi_{ij}) - \frac{a_{\mathbf{Z}_i, \mathbf{Z}_j} + \xi_{ij}}{2} - \lambda(\xi_{ij})(a^2_{\mathbf{Z}_i, \mathbf{Z}_j} - \xi^2_{ij}) \\ &= (X_{ij} - \frac{1}{2}) a_{\mathbf{Z}_i, \mathbf{Z}_j} - \frac{\xi_{ij}}{2} + \log g(\xi_{ij}) - \lambda(\xi_{ij})(a^2_{\mathbf{Z}_i, \mathbf{Z}_j} - \xi^2_{ij}). \end{aligned} \quad \text{(D.2)}$$

Following (4.1):

$$\log p(\mathbf{X} \mid \mathbf{Z}, \tilde{\mathbf{W}}) = \sum_{i \neq j}^{N} \log p(\mathbf{X}_{ij} \mid \mathbf{Z}_i, \mathbf{Z}_j, \tilde{\mathbf{W}}).$$

Therefore

$$\log p(\mathbf{X} \mid \mathbf{Z}, \tilde{\mathbf{W}}) \geq \log h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}).$$

We recall that the lower bound $\mathcal{L}(q)$ is given by:

$$\begin{aligned}
\mathcal{L}(q) &= \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})} \right\} \\
&= \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log p(\mathbf{X} \mid \mathbf{Z}, \tilde{\mathbf{W}}) + \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log \left(p(\mathbf{Z} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\tilde{\mathbf{W}}) \right) \\
&\quad - q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \\
&\geq \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) + \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log \left(p(\mathbf{Z} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\tilde{\mathbf{W}}) \right) \\
&\quad - q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \\
&= \sum_{\mathbf{Z}} \int\int q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}}) \log \left(\frac{h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) p(\mathbf{Z} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\tilde{\mathbf{W}})}{q(\mathbf{Z}, \boldsymbol{\alpha}, \tilde{\mathbf{W}})} \right) d\boldsymbol{\alpha}\, d\tilde{\mathbf{W}} \\
&= \mathcal{L}(q; \boldsymbol{\xi}).
\end{aligned}$$

Finally

$$\log p(\mathbf{X}) \geq \mathcal{L}(q) \geq \mathcal{L}(q; \boldsymbol{\xi}).$$

D.2 Optimization of $q(\boldsymbol{\alpha})$

The optimization of the lower bound with respect to $q(\boldsymbol{\alpha})$ produces a distribution with the same functional form as the prior $p(\boldsymbol{\alpha})$:

$$q(\boldsymbol{\alpha}) = \prod_{q=1}^{Q} \text{Beta}(\alpha_q; \eta_q^N, \zeta_q^N),$$

where

$$\eta_q^N = \eta_q^0 + \sum_{i=1}^{N} \tau_{iq},$$

and

$$\zeta_q^N = \zeta_q^0 + N - \sum_{i=1}^{N} \tau_{iq}.$$

Proof: According to variational Bayes, the optimal distribution $q(\boldsymbol{\alpha})$ is given by:

$$\begin{aligned}
\log q(\boldsymbol{\alpha}) &= \mathrm{E}_{\mathbf{Z}, \tilde{\mathbf{W}}}[\log \left(h(\mathbf{Z}, \tilde{\mathbf{W}}, \boldsymbol{\xi}) p(\mathbf{Z} \mid \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\tilde{\mathbf{W}}) \right)] + \text{const} \\
&= \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{Z} \mid \boldsymbol{\alpha})] + \log p(\boldsymbol{\alpha}) + \text{const} \\
&= \sum_{i=1}^{N} \left\{ \tau_{iq} \log \alpha_q + (1 - \tau_{iq}) \log(1 - \alpha_q) \right\} + \sum_{q=1}^{Q} \left\{ (\eta_q - 1) \log \alpha_q + (\zeta_q - 1) \log(1 - \alpha_q) \right\} \\
&\quad + \text{const} \\
&= \sum_{q=1}^{Q} \left\{ (\eta_q^0 + \sum_{i=1}^{N} \tau_{iq} - 1) \log \alpha_q + (\zeta_q^0 + N - \sum_{i=1}^{N} \tau_{iq} - 1) \log(1 - \alpha_q) \right\} + \text{const.}
\end{aligned} \tag{D.3}$$

The functional form of (D.3) corresponds to the logarithm of a product of Beta distributions.

D.3 Optimization of $q(\tilde{\mathbf{W}})$

The optimization of the lower bound with respect to $q(\tilde{\mathbf{W}})$ produces a distribution with the same functional form as the prior $p(\tilde{\mathbf{W}})$:

$$q(\tilde{\mathbf{W}}^{vec}) = \mathcal{N}(\tilde{\mathbf{W}}^{\text{vec}};\, \tilde{\mathbf{W}}_N^{\text{vec}}, \mathbf{S}_N),$$

with

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + 2\sum_{i\neq j}^{N} \lambda(\xi_{ij})(\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i),$$

and

$$\tilde{\mathbf{W}}_N^{\text{vec}} = \mathbf{S}_N \left\{ \mathbf{S}_0^{-1}\, \tilde{\mathbf{W}}_0^{\text{vec}} + \sum_{i\neq j}^{N} (X_{ij} - \frac{1}{2})\, \tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i \right\}.$$

Each $(Q+1)\times(Q+1)$ probability matrix $\tilde{\mathbf{E}}_i$ satisfies:

$$\begin{aligned}\tilde{\mathbf{E}}_i &= \mathrm{E}_{\mathbf{Z}_i}[\tilde{\mathbf{Z}}_i\, \tilde{\mathbf{Z}}_i^{\intercal}] \\ &= \begin{pmatrix} \tau_{i1} & \tau_{i1}\tau_{i2} & \dots & \tau_{i1}\tau_{iQ} & \tau_{i1} \\ \tau_{i2}\tau_{i1} & \tau_{i2} & \dots & \tau_{i2}\tau_{iQ} & \tau_{i2} \\ \vdots & & & & \vdots \\ \tau_{iQ}\tau_{i1} & \tau_{iQ}\tau_{i2} & \dots & \tau_{iQ} & \tau_{iQ} \\ \tau_{i1} & \tau_{i2} & \dots & \tau_{iQ} & 1 \end{pmatrix}.\end{aligned}$$

Proof: According to variational Bayes, the optimal distribution $q(\tilde{\mathbf{W}})$ is given by:

$$\begin{aligned}\log q(\tilde{\mathbf{W}}^{\text{vec}}) &= \mathrm{E}_{\mathbf{Z},\boldsymbol{\alpha}}[\log\left(h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})p(\mathbf{Z}\,|\,\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\tilde{\mathbf{W}})\right)] + \text{const} \\ &= \mathrm{E}_{\mathbf{Z}}[\log h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})] + \log p(\tilde{\mathbf{W}}^{\text{vec}}) + \text{const} \\ &= \sum_{i\neq j}^{N} \left\{ (X_{ij} - \frac{1}{2})\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a_{\mathbf{Z}_i,\mathbf{Z}_j}] - \lambda(\xi_{ij})\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a^2_{\mathbf{Z}_i,\mathbf{Z}_j}] \right\} \\ &\quad + (\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}\, \mathbf{S}_0^{-1}\, \tilde{\mathbf{W}}_0^{\text{vec}} - \frac{1}{2}(\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}\, \mathbf{S}_0^{-1}\, \tilde{\mathbf{W}}^{\text{vec}} + \text{const}.\end{aligned} \tag{D.4}$$

$\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a_{\mathbf{Z}_i,\mathbf{Z}_j}]$ is given by:

$$\begin{aligned}\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a_{\mathbf{Z}_i,\mathbf{Z}_j}] &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\tilde{\mathbf{Z}}_i^{\intercal}\, \tilde{\mathbf{W}}\, \tilde{\mathbf{Z}}_j] \\ &= \tilde{\boldsymbol{\tau}}_i^{\intercal}\, \tilde{\mathbf{W}}\, \tilde{\boldsymbol{\tau}}_j \\ &= (\tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i)^{\intercal}\, \tilde{\mathbf{W}}^{\text{vec}} \\ &= (\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}(\tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i).\end{aligned} \tag{D.5}$$

$\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a^2_{\mathbf{Z}_i,\mathbf{Z}_j}]$ is given by:

$$\begin{aligned}\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[a^2_{\mathbf{Z}_i,\mathbf{Z}_j}] &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{Z}}_i^{\intercal}\, \tilde{\mathbf{W}}\, \tilde{\mathbf{Z}}_j)^2] \\ &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[\left((\tilde{\mathbf{Z}}_j \otimes \tilde{\mathbf{Z}}_i)^{\intercal}\, \tilde{\mathbf{W}}^{\text{vec}}\right)^2] \\ &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{Z}}_j \otimes \tilde{\mathbf{Z}}_i)^{\intercal}\, \tilde{\mathbf{W}}^{\text{vec}}(\tilde{\mathbf{Z}}_j \otimes \tilde{\mathbf{Z}}_i)^{\intercal}\, \tilde{\mathbf{W}}^{\text{vec}}] \\ &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}(\tilde{\mathbf{Z}}_j \otimes \tilde{\mathbf{Z}}_i)(\tilde{\mathbf{Z}}_j \otimes \tilde{\mathbf{Z}}_i)^{\intercal}\, \tilde{\mathbf{W}}^{\text{vec}}] \\ &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j}[(\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}\left((\tilde{\mathbf{Z}}_j\, \tilde{\mathbf{Z}}_j^{\intercal}) \otimes (\tilde{\mathbf{Z}}_i\, \tilde{\mathbf{Z}}_i^{\intercal})\right)\, \tilde{\mathbf{W}}^{\text{vec}}] \\ &= (\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}\left(\, \tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i\, \right)\, \tilde{\mathbf{W}}^{\text{vec}}.\end{aligned} \tag{D.6}$$

Using (D.5) and (D.6) in (D.4), we obtain:

$$\log q(\tilde{\mathbf{W}}^{\text{vec}}) = (\mathbf{W}^{\text{vec}})^{\intercal}\left\{\mathbf{S}_0^{-1}\,\tilde{\mathbf{W}}_0^{\text{vec}} + \sum_{i\neq j}^{N}(X_{ij}-\frac{1}{2})(\tilde{\boldsymbol{\tau}}_j\otimes\tilde{\boldsymbol{\tau}}_i)\right\} - (\tilde{\mathbf{W}}^{\text{vec}})^{\intercal}\left\{\frac{1}{2}\,\mathbf{S}_0^{-1} + \sum_{i\neq j}^{N}\lambda(\xi_{ij})\big(\,\tilde{\mathbf{E}}_j\otimes\tilde{\mathbf{E}}_i\,\big)\right\}\tilde{\mathbf{W}}^{\text{vec}} + \text{const.} \tag{D.7}$$

The fonctional form of (D.7) corresponds to the logarithm of a Gaussian distribution with mean $\tilde{\mathbf{W}}_N^{\text{vec}}$ and covariance matrix $\mathbf{S}_N$.

D.4 Optimization of $q(Z_{iq})$

The optimization of the lower bound with respect to $q(Z_{iq})$ produces a distribution with the same functional form as the prior $p(Z_{iq}|\,\boldsymbol{\alpha})$:

$$q(Z_{iq}) = \mathcal{B}(Z_{iq};\,\tau_{iq}),$$

where

$$\tau_{iq} = g\Big\{\psi(\eta_q^N) - \psi(\zeta_q^N) + \sum_{j\neq i}^{N}(X_{ij}-\frac{1}{2})\,\tilde{\boldsymbol{\tau}}_j^{\intercal}(\tilde{\mathbf{W}}_N^{\intercal})_{\cdot q} + \sum_{j\neq i}^{N}(X_{ji}-\frac{1}{2})\,\tilde{\boldsymbol{\tau}}_j^{\intercal}(\tilde{\mathbf{W}}_N)_{\cdot q} - \text{Tr}\Big(\big(\,\boldsymbol{\Sigma}'_{qq} + 2\sum_{l\neq q}^{Q+1}\tilde{\tau}_{il}\,\boldsymbol{\Sigma}'_{ql}\,\big)\big(\sum_{j\neq i}^{N}\lambda(\xi_{ij})\,\tilde{\mathbf{E}}_j\,\big) + \big(\,\boldsymbol{\Sigma}_{qq} + 2\sum_{l\neq q}^{Q+1}\tilde{\tau}_{il}\,\boldsymbol{\Sigma}_{ql}\,\big)\big(\sum_{j\neq i}^{N}\lambda(\xi_{ji})\,\tilde{\mathbf{E}}_j\,\big)\Big)\Big\},$$

and $\boldsymbol{\Sigma}_{ql} = \text{E}_{\tilde{\mathbf{W}}_q,\tilde{\mathbf{W}}_l}[\tilde{\mathbf{W}}_{\cdot q}\,\tilde{\mathbf{W}}_{\cdot l}^{\intercal}]$, $\boldsymbol{\Sigma}'_{ql} = \text{E}_{\tilde{\mathbf{W}}_{q\cdot},\tilde{\mathbf{W}}_{l\cdot}}[\tilde{\mathbf{W}}_{q\cdot}^{\intercal}\,\tilde{\mathbf{W}}_{l\cdot}]$.

Proof: According to variational Bayes, the optimal distribution $q(Z_{iq})$ is given by:

$$\log q(Z_{bc}) = \text{E}_{\mathbf{Z}^{\setminus bc},\boldsymbol{\alpha},\tilde{\mathbf{W}}}[\log\big(h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})\,p(\mathbf{Z}\,|\,\boldsymbol{\alpha})\,p(\boldsymbol{\alpha})\,p(\tilde{\mathbf{W}})\big)] + \text{const},$$

where $\mathbf{Z}^{\setminus bc}$ is the set of all class memberships except Z_{bc}.

$$\log q(Z_{bc}) = \text{E}_{\mathbf{Z}^{\setminus bc},\tilde{\mathbf{W}}}[\log h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})] + \text{E}_{\mathbf{Z}^{\setminus bc},\boldsymbol{\alpha}}[\log p(\mathbf{Z}\,|\,\boldsymbol{\alpha})] + \text{const}.$$

$\text{E}_{\mathbf{Z}^{\setminus bc},\boldsymbol{\alpha}}[\log p(\mathbf{Z}\,|\,\boldsymbol{\alpha})]$ is given by:

$$\begin{aligned}\text{E}_{\mathbf{Z}^{\setminus bc},\boldsymbol{\alpha}}[\log p(\mathbf{Z}\,|\,\boldsymbol{\alpha})] &= Z_{bc}\text{E}_{\alpha_c}[\log\alpha_c] + (1-Z_{bc})\text{E}_{\alpha_c}[\log(1-\alpha_c)] + \text{const}\\ &= Z_{bc}\big(\psi(\eta_c^N) - \psi(\eta_c^N+\zeta_c^N)\big) + (1-Z_{bc})\big(\psi(\zeta_c^N) - \psi(\eta_c^N+\zeta_c^N)\big) + \text{const}\\ &= Z_{bc}\big(\psi(\eta_c^N) - \psi(\zeta_c^N)\big) + \text{const},\end{aligned}$$

where $\psi(\cdot)$ is the digamma function (the logarithmic derivative of the gamma function $\Gamma(\cdot)$ which appears in the normalizaing constants of the Beta distributions).

$$
\begin{aligned}
\mathrm{E}_{\mathbf{Z}^{\setminus bc},\tilde{\mathbf{W}}}[\log h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})] &= \sum_{i\neq j}^{N}\Big\{(X_{ij}-\frac{1}{2})\mathrm{E}_{\mathbf{Z}^{\setminus bc},\tilde{\mathbf{W}}}[a_{\mathbf{Z}_i,\mathbf{Z}_j}] - \lambda(\xi_{ij})\mathrm{E}_{\mathbf{Z}^{\setminus bc},\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_i,\mathbf{Z}_j}]\Big\} + \text{const} \\
&= \sum_{j\neq b}^{N}\Big\{(X_{bj}-\frac{1}{2})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_b,\mathbf{Z}_j}] - \lambda(\xi_{bj})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_b,\mathbf{Z}_j}]\Big\} \\
&\quad + \sum_{i\neq b}^{N}\Big\{(X_{ib}-\frac{1}{2})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_i,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_i,\mathbf{Z}_b}] - \lambda(\xi_{ib})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_i,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_i,\mathbf{Z}_b}]\Big\} + \text{const} \\
&= \sum_{j\neq b}^{N}\Big\{(X_{bj}-\frac{1}{2})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_b,\mathbf{Z}_j}] + (X_{jb}-\frac{1}{2})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_j,\mathbf{Z}_b}] \\
&\quad - \lambda(\xi_{bj})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_b,\mathbf{Z}_j}] - \lambda(\xi_{jb})\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_j,\mathbf{Z}_b}]\Big\} + \text{const.}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_b,\mathbf{Z}_j}] &= \mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[\sum_{q,l}^{Q+1}\tilde{Z}_{bq}\,\tilde{W}_{ql}\,\tilde{Z}_{jl}] \\
&= Z_{bc}\sum_{l=1}^{Q+1}\mathrm{E}_{\tilde{W}_{cl}}[\tilde{W}_{cl}]\tilde{\tau}_{jl} + \text{const} \\
&= Z_{bc}\,\tilde{\boldsymbol{\tau}}_j^{\intercal}(\tilde{\mathbf{W}}_N^{\intercal})_{\cdot c} + \text{const.}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_j,\mathbf{Z}_b}] &= \mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[\sum_{q,l}^{Q+1}\tilde{Z}_{jq}\,\tilde{W}_{ql}\,\tilde{Z}_{bl}] \\
&= Z_{bc}\sum_{l=1}^{Q+1}\mathrm{E}_{\tilde{W}_{lc}}[\tilde{W}_{lc}]\tilde{\tau}_{jl} + \text{const} \\
&= Z_{bc}\,\tilde{\boldsymbol{\tau}}_j^{\intercal}(\tilde{\mathbf{W}}_N)_{\cdot c} + \text{const.}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_j,\mathbf{Z}_b}] &= \mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[(\sum_{q,l}^{Q+1}\tilde{Z}_{jq}\,\tilde{W}_{ql}\,\tilde{Z}_{bl})(\sum_{q,l}^{Q+1}\tilde{Z}_{jq}\,\tilde{W}_{ql}\,\tilde{Z}_{bl})] \\
&= \mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[\sum_{q,q',l,l'}^{Q+1}\tilde{Z}_{bl}\,\tilde{Z}_{bl'}\,\tilde{Z}_{jq}\,\tilde{W}_{ql}\,\tilde{W}_{q'l'}\,\tilde{Z}_{jq'}] \\
&= \mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[Z_{bc}\sum_{q,q'}^{Q+1}\tilde{Z}_{jq}\,\tilde{W}_{qc}\,\tilde{W}_{q'c}\,\tilde{Z}_{jq'} + 2Z_{bc}\sum_{q,q',l\neq c}^{Q+1}\tilde{Z}_{bl}\,\tilde{Z}_{jq}\,\tilde{W}_{qc}\,\tilde{W}_{q'l}\,\tilde{Z}_{jq'}] + \text{const} \\
&= Z_{bc}\left\{\mathrm{E}_{\mathbf{Z}_j,\tilde{\mathbf{W}}_{\cdot c}}[\tilde{\mathbf{W}}_{\cdot c}^{\intercal}\,\tilde{\mathbf{Z}}_j\,\tilde{\mathbf{Z}}_j^{\intercal}\,\tilde{\mathbf{W}}_{\cdot c}] + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\mathrm{E}_{\mathbf{Z}_j,\tilde{\mathbf{W}}_{\cdot c},\tilde{\mathbf{W}}_{\cdot l}}[\tilde{\mathbf{W}}_{\cdot c}^{\intercal}\,\tilde{\mathbf{Z}}_j\,\tilde{\mathbf{Z}}_j^{\intercal}\,\tilde{\mathbf{W}}_{\cdot l}]\right\} + \text{const} \\
&= Z_{bc}\left\{\mathrm{E}_{\tilde{\mathbf{W}}_{\cdot c}}[\tilde{\mathbf{W}}_{\cdot c}^{\intercal}\,\tilde{\mathbf{E}}_j\,\tilde{\mathbf{W}}_{\cdot c}] + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\mathrm{E}_{\tilde{\mathbf{W}}_{\cdot c},\tilde{\mathbf{W}}_{\cdot l}}[\tilde{\mathbf{W}}_{\cdot c}^{\intercal}\,\tilde{\mathbf{E}}_j\,\tilde{\mathbf{W}}_{\cdot l}]\right\} + \text{const} \\
&= Z_{bc}\left\{\mathrm{E}_{\tilde{\mathbf{W}}_{\cdot c}}[(\tilde{\mathbf{W}}_{\cdot c}\otimes\tilde{\mathbf{W}}_{\cdot c})^{\intercal}]\,\tilde{\mathbf{E}}_j^{\mathrm{vec}} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\mathrm{E}_{\tilde{\mathbf{W}}_{\cdot c},\tilde{\mathbf{W}}_{\cdot l}}[(\tilde{\mathbf{W}}_{\cdot l}\otimes\tilde{\mathbf{W}}_{\cdot c})^{\intercal}]\,\tilde{\mathbf{E}}_j^{\mathrm{vec}}\right\} + \text{const} \\
&= Z_{bc}\left\{\mathrm{E}_{\tilde{\mathbf{W}}_{\cdot c}}[((\tilde{\mathbf{W}}_{\cdot c}\,\tilde{\mathbf{W}}_{\cdot c}^{\intercal})^{\mathrm{vec}})^{\intercal}]\,\tilde{\mathbf{E}}_j^{\mathrm{vec}} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\mathrm{E}_{\tilde{\mathbf{W}}_{\cdot c},\tilde{\mathbf{W}}_{\cdot l}}[((\tilde{\mathbf{W}}_{\cdot c}\,\tilde{\mathbf{W}}_{\cdot l}^{\intercal})^{\mathrm{vec}})^{\intercal}]\,\tilde{\mathbf{E}}_j^{\mathrm{vec}}\right\} + \text{const} \\
&= Z_{bc}\left\{(\boldsymbol{\Sigma}_{cc}^{\mathrm{vec}})^{\intercal}\,\tilde{\mathbf{E}}_j^{\mathrm{vec}} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}(\boldsymbol{\Sigma}_{cl}^{\mathrm{vec}})^{\intercal}\,\tilde{\mathbf{E}}_j^{\mathrm{vec}}\right\} + \text{const} \\
&= Z_{bc}\mathrm{Tr}\Big((\,\boldsymbol{\Sigma}_{cc} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\,\boldsymbol{\Sigma}_{cl}\,)\,\tilde{\mathbf{E}}_j\Big) + \text{const},
\end{aligned}
$$

where $\boldsymbol{\Sigma}_{ql} = \mathrm{E}_{\tilde{\mathbf{W}}_q,\tilde{\mathbf{W}}_l}[\tilde{\mathbf{W}}_{\cdot q}\,\tilde{\mathbf{W}}_{\cdot l}^{\intercal}]$. Similarly, we have:

$$
\mathrm{E}_{\mathbf{Z}_b^{\setminus c},\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_b,\mathbf{Z}_j}] = Z_{bc}\mathrm{Tr}\Big((\,\boldsymbol{\Sigma}'_{cc} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\,\boldsymbol{\Sigma}'_{cl}\,)\,\tilde{\mathbf{E}}_j\Big) + \text{const},
$$

where $\boldsymbol{\Sigma}'_{ql} = \mathrm{E}_{\tilde{\mathbf{W}}_{q\cdot},\tilde{\mathbf{W}}_{l\cdot}}[\tilde{\mathbf{W}}_{q\cdot}^{\intercal}\,\tilde{\mathbf{W}}_{l\cdot}]$. Finally, we obtain:

$$
\begin{aligned}
\log q(Z_{bc}) = Z_{bc}\Big\{&\psi(\eta_c^N) - \psi(\zeta_c^N) + \sum_{j\neq b}^{N}(X_{bj}-\frac{1}{2})\,\tilde{\boldsymbol{\tau}}_j^{\intercal}(\tilde{\mathbf{W}}_N^{\intercal})_{\cdot c} + \sum_{j\neq b}^{N}(X_{jb}-\frac{1}{2})\,\tilde{\boldsymbol{\tau}}_j^{\intercal}(\tilde{\mathbf{W}}_N)_{\cdot c} \\
&- \mathrm{Tr}\Big((\,\boldsymbol{\Sigma}'_{cc} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\,\boldsymbol{\Sigma}'_{cl}\,)(\sum_{j\neq b}^{N}\lambda(\xi_{bj})\,\tilde{\mathbf{E}}_j\,) + (\,\boldsymbol{\Sigma}_{cc} + 2\sum_{l\neq c}^{Q+1}\tilde{\tau}_{bl}\,\boldsymbol{\Sigma}_{cl}\,)(\sum_{j\neq b}^{N}\lambda(\xi_{jb})\,\tilde{\mathbf{E}}_j\,)\Big)\Big\} \\
&+ \text{const}.
\end{aligned}
\tag{D.8}
$$

The fonctional form of (D.8) corresponds to the logarithm of a Bernoulli distribution with parameter τ_{bc}. Indeed:

$$
\begin{aligned}
\log\mathcal{B}(Z_{bc};\,\tau_{bc}) &= Z_{bc}\log\tau_{bc} + (1-Z_{bc})\log(1-\tau_{bc}) \\
&= Z_{bc}\log(\frac{\tau_{bc}}{1-\tau_{bc}}) + \text{const}.
\end{aligned}
$$

If we denote $p = \log\left(\tau_{bc}/(1-\tau_{bc})\right)$, then $\tau_{bc} = g(p)$.

D.5 Optimization of ξ

Setting the partial derivative of the lower bound with respect to ξ_{ij}, to zero, leads to an estimate $\hat{\xi_{ij}}$ of ξ_{ij}:

$$\hat{\xi}_{ij} = \sqrt{\mathrm{Tr}\Big(\big(\mathbf{S}_N + \tilde{\mathbf{W}}_N^{\mathrm{vec}} (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \big) (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big)}.$$

Proof: The partial derivative of the lower bound with respect to ξ_{ij} is given by:

$$\frac{\partial \mathcal{L}}{\partial \xi_{ij}}(q; \boldsymbol{\xi}) = -\frac{1}{2} + g(-\xi_{ij}) - \lambda^{'}(\xi_{ij}) \big(\mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j, \tilde{\mathbf{W}}} [a^2_{\mathbf{Z}_i, \mathbf{Z}_j}] - \xi_{ij}^2 \big) + 2\xi_{ij} \lambda(\xi_{ij}).$$

According to (D.6),

$$\mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j} [a^2_{\mathbf{Z}_i, \mathbf{Z}_j}] = (\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal \big(\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i \big) \tilde{\mathbf{W}}^{\mathrm{vec}},$$

therefore

$$\begin{aligned} \mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j, \tilde{\mathbf{W}}} [a^2_{\mathbf{Z}_i, \mathbf{Z}_j}] &= \mathrm{E}_{\tilde{\mathbf{W}}} [(\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \, \tilde{\mathbf{W}}^{\mathrm{vec}}] \\ &= \mathrm{E}_{\tilde{\mathbf{W}}} \Big[[\mathrm{Tr}\Big(\tilde{\mathbf{W}}^{\mathrm{vec}} (\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big) \Big] \\ &= \mathrm{Tr}\Big(\mathrm{E}_{\tilde{\mathbf{W}}} \big[\tilde{\mathbf{W}}^{\mathrm{vec}} (\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal \big] (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big) \\ &= \mathrm{Tr}\Big(\big(\mathbf{S}_N + \tilde{\mathbf{W}}_N^{\mathrm{vec}} (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \big) (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big). \end{aligned} \tag{D.9}$$

Moreover $(\log g)^{'}(\xi_{ij}) = g(-\xi_{ij})$ and $g(\xi_j) + g(-\xi_{ij}) = 1$. We obtain:

$$\frac{\partial \mathcal{L}}{\partial \xi_{ij}}(q; \boldsymbol{\xi}) = -\lambda^{'}(\xi_{ij}) \left\{ \mathrm{Tr}\Big(\big(\mathbf{S}_N + \tilde{\mathbf{W}}_N^{\mathrm{vec}} (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \big) (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big) - \xi_{ij}^2 \right\}.$$

Finally, $\lambda(\xi_{ij})$ is a strictly decreasing function for positive values of ξ_{ij}. Thus, $\lambda^{'}(\xi_{ij}) \neq 0$ and if we set the derivative of (D.5) to zero, it leads to:

$$\xi_{ij}^2 = \mathrm{Tr}\Big(\big(\mathbf{S}_N + \tilde{\mathbf{W}}_N^{\mathrm{vec}} (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \big) (\tilde{\mathbf{E}}_j \otimes \tilde{\mathbf{E}}_i) \Big).$$

D.6 Lower bound

After the variational Bayes M-step, most of the terms in the lower bound vanish:

$$\begin{aligned} \mathcal{L}(q; \boldsymbol{\xi}) = \sum_{i \neq j}^{N} \left\{ \log g(\xi_{ij}) - \frac{\xi_{ij}}{2} + \lambda(\xi_{ij}) \xi_{ij}^2 \right\} + \sum_{q=1}^{Q} \log \left\{ \frac{\Gamma(\eta_q^0 + \zeta_q^0) \Gamma(\eta_q^N) \Gamma(\zeta_q^N)}{\Gamma(\eta_q^0) \Gamma(\zeta_q^0) \Gamma(\eta_q^N + \zeta_q^N)} \right\} \\ - \frac{1}{2} \log |\frac{\mathbf{S}_0}{\mathbf{S}_N}| - \frac{1}{2} (\tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \, \mathbf{S}_0^{-1} \, \tilde{\mathbf{W}}_0^{\mathrm{vec}} + \frac{1}{2} (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \, \mathbf{S}_N^{-1} \, \tilde{\mathbf{W}}_N^{\intercal} \\ - \sum_{i=1}^{N} \sum_{q=1}^{Q} \Big\{ \tau_{iq} \log \tau_{iq} + (1 - \tau_{iq}) \log (1 - \tau_{iq}) \Big\}. \end{aligned} \tag{D.10}$$

Proof:

$$
\begin{aligned}
\mathcal{L}(q;\,\boldsymbol{\xi}) &= \sum_{\mathbf{Z}} \int\int q(\mathbf{Z},\boldsymbol{\alpha},\tilde{\mathbf{W}}) \log\big(\frac{h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})\,p(\mathbf{Z}\,|\,\boldsymbol{\alpha})\,p(\boldsymbol{\alpha})\,p(\tilde{\mathbf{W}})}{q(\mathbf{Z},\boldsymbol{\alpha},\tilde{\mathbf{W}})} d\,\boldsymbol{\alpha}\, d\,\tilde{\mathbf{W}} \\
&= \mathrm{E}_{\mathbf{Z},\tilde{\mathbf{W}}}[\log h(\mathbf{Z},\tilde{\mathbf{W}},\boldsymbol{\xi})] + \mathrm{E}_{\mathbf{Z},\boldsymbol{\alpha}}[\log p(\mathbf{Z}\,|\,\boldsymbol{\alpha})] + \mathrm{E}_{\boldsymbol{\alpha}}[\log p(\boldsymbol{\alpha})] + \mathrm{E}_{\tilde{\mathbf{W}}}[\log p(\tilde{\mathbf{W}})] \\
&\quad - \mathrm{E}_{\mathbf{Z}}[\log q(\mathbf{Z})] - \mathrm{E}_{\boldsymbol{\alpha}}[\log q(\boldsymbol{\alpha})] - \mathrm{E}_{\tilde{\mathbf{W}}}[\log q(\tilde{\mathbf{W}})] \\
&= \sum_{i\neq j}^{N} \Big\{ (X_{ij} - \frac{1}{2}) \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_i,\mathbf{Z}_j}] - \frac{\xi_{ij}}{2} + \log g(\xi_{ij}) - \lambda(\xi_{ij})\big(\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_i,\mathbf{Z}_j}] - \xi^2_{ij}\big) \Big\} \\
&\quad + \sum_{i=1}^{N}\sum_{q=1}^{Q} \Big\{ \tau_{iq}\big(\psi(\eta_q^N) - \psi(\eta_q^N + \zeta_q^N)\big) + (1-\tau_{iq})\big(\psi(\zeta_q^N) - \psi(\eta_q^N + \zeta_q^N)\big) \Big\} \\
&\quad + \sum_{q=1}^{Q} \Big\{ \log\big(\frac{\Gamma(\eta_q^0 + \zeta_q^0)}{\Gamma(\eta_q^0)\Gamma(\zeta_q^0)}\big) + (\eta_q^0 - 1)\big(\psi(\eta_q^N) - \psi(\eta_q^N + \zeta_q^N)\big) \\
&\quad + (\zeta_q^0 - 1)\big(\psi(\zeta_q^N) - \psi(\eta_q^N + \zeta_q^N)\big) \Big\} \\
&\quad + \mathrm{E}_{\tilde{\mathbf{W}}}[\log p(\tilde{\mathbf{W}})] - \sum_{i=1}^{N}\sum_{q=1}^{Q} \Big\{ \tau_{iq}\log\tau_{iq} + (1-\tau_{iq})\log(1-\tau_{iq}) \Big\} \\
&\quad - \sum_{q=1}^{Q} \Big\{ \log\big(\frac{\Gamma(\eta_q^N + \zeta_q^N)}{\Gamma(\eta_q^N)\Gamma(\zeta_q^N)}\big)(\eta_q^N - 1)\big(\psi(\eta_q^N) - \psi(\eta_q^N + \zeta_q^N)\big) \\
&\quad + (\zeta_q^N - 1)\big(\psi(\zeta_q^N) - \psi(\eta_q^N + \zeta_q^N)\big) \Big\} \\
&\quad - \mathrm{E}_{\tilde{\mathbf{W}}}[\log q(\tilde{\mathbf{W}})].
\end{aligned}
\tag{D.11}
$$

$\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_i,\mathbf{Z}_j}]$ is given by:

$$
\begin{aligned}
\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j,\tilde{\mathbf{W}}}[a_{\mathbf{Z}_i,\mathbf{Z}_j}] &= \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j,\tilde{\mathbf{W}}}[\tilde{\mathbf{Z}}_i^{\intercal}\,\tilde{\mathbf{W}}\,\tilde{\mathbf{Z}}_j] \\
&= \mathrm{E}_{\tilde{\mathbf{W}}}[\tilde{\boldsymbol{\tau}}_i^{\intercal}\,\tilde{\mathbf{W}}\,\tilde{\boldsymbol{\tau}}_j] \\
&= \mathrm{E}_{\tilde{\mathbf{W}}}[(\tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i)^{\intercal}\,\tilde{\mathbf{W}}^{\mathrm{vec}}] \\
&= \mathrm{E}_{\tilde{\mathbf{W}}}[(\tilde{\mathbf{W}}^{\mathrm{vec}})^{\intercal}(\tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i)] \\
&= (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^{\intercal}(\tilde{\boldsymbol{\tau}}_j \otimes \tilde{\boldsymbol{\tau}}_i).
\end{aligned}
\tag{D.12}
$$

$\mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j,\tilde{\mathbf{W}}}[a^2_{\mathbf{Z}_i,\mathbf{Z}_j}]$ is given by (D.9)

$\mathrm{E}_{\tilde{\mathbf{W}}}[\log p(\tilde{\mathbf{W}})]$ is given by:

$$\begin{aligned}
\mathrm{E}_{\tilde{\mathbf{W}}}[\log p(\tilde{\mathbf{W}})] &= \mathrm{E}_{\tilde{\mathbf{W}}}[-\frac{1}{2}(\tilde{\mathbf{W}}^{\mathrm{vec}} - \tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}(\tilde{\mathbf{W}}^{\mathrm{vec}} - \tilde{\mathbf{W}}_0^{\mathrm{vec}})] - \frac{1}{2}(Q+1)^2\log(2\pi) - \frac{1}{2}\log|\,\mathbf{S}_0\,| \\
&= -\frac{1}{2}\mathrm{E}_{\tilde{\mathbf{W}}}[(\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}^{\mathrm{vec}}] + \mathrm{E}_{\tilde{\mathbf{W}}}[(\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}}] - \frac{1}{2}(\tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} \\
&\quad - \frac{1}{2}(Q+1)^2\log(2\pi) - \frac{1}{2}\log|\,\mathbf{S}_0\,| \\
&= -\frac{1}{2}\mathrm{E}_{\tilde{\mathbf{W}}}[(\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}^{\mathrm{vec}}] + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} - \frac{1}{2}(\tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} \\
&\quad - \frac{1}{2}(Q+1)^2\log(2\pi) - \frac{1}{2}\log|\,\mathbf{S}_0\,| \\
&= -\frac{1}{2}\mathrm{Tr}\Big(\mathrm{E}_{\tilde{\mathbf{W}}}\big[(\tilde{\mathbf{W}}^{\mathrm{vec}})^\intercal \tilde{\mathbf{W}}^{\mathrm{vec}}\big]\,\mathbf{S}_0^{-1}\Big) + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} - \frac{1}{2}(\tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} \\
&\quad - \frac{1}{2}(Q+1)^2\log(2\pi) - \frac{1}{2}\log|\,\mathbf{S}_0\,| \\
&= -\frac{1}{2}\mathrm{Tr}\Big(\big(\,\mathbf{S}_N + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \tilde{\mathbf{W}}_N^{\mathrm{vec}}\,\big)\,\mathbf{S}_0^{-1}\Big) + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} - \frac{1}{2}(\tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} \\
&\quad - \frac{1}{2}(Q+1)^2\log(2\pi) - \frac{1}{2}\log|\,\mathbf{S}_0\,|.
\end{aligned} \tag{D.13}$$

Similary, we have:

$$\begin{aligned}
\mathrm{E}_{\tilde{\mathbf{W}}}[\log q(\tilde{\mathbf{W}})] &= -\frac{1}{2}\mathrm{Tr}\Big(\big(\,\mathbf{S}_N + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \tilde{\mathbf{W}}_N^{\mathrm{vec}}\,\big)\,\mathbf{S}_N^{-1}\Big) + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \mathbf{S}_N^{-1}\tilde{\mathbf{W}}_N^{\mathrm{vec}} - \frac{1}{2}(\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \mathbf{S}_N^{-1}\tilde{\mathbf{W}}_N^{\mathrm{vec}} \\
&\quad - \frac{1}{2}(Q+1)^2\log(2\pi) - \frac{1}{2}\log|\,\mathbf{S}_N\,|.
\end{aligned} \tag{D.14}$$

After rearranging the terms in (D.11) and using (D.9), (D.12), (D.13), as well as (D.14), we obtain:

$$\begin{aligned}
\mathcal{L}(q;\boldsymbol{\xi}) &= \sum_{i\neq j}^{N}\Big\{\log g(\xi_{ij}) - \frac{\xi_{ij}}{2} + \lambda(\xi_{ij})\xi_{ij}^2\Big\} + \sum_{q=1}^{Q}\log\Big\{\frac{\Gamma(\eta_q^0+\zeta_q^0)\Gamma(\eta_q^N)\Gamma(\zeta_q^N)}{\Gamma(\eta_q^0)\Gamma(\zeta_q^0)\Gamma(\eta_q^N+\zeta_q^N)}\Big\} \\
&\sum_{q=1}^{Q}\Big\{(\eta_q^0 + \sum_{i\neq j}^{N}\tau_{iq} - \eta_q^N)\,(\psi(\eta_q^N) - \psi(\eta_q^N+\zeta_q^N)) + (\zeta_q^0 + N - \sum_{i=1}^{N}\tau_{iq} - \zeta_q^N)\,(\psi(\zeta_q^N) - \psi(\eta_q^N+\zeta_q^N))\Big\} \\
&\quad - \frac{1}{2}\mathrm{Tr}\Big(\big(\,\mathbf{S}_N + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \tilde{\mathbf{W}}_N^{\mathrm{vec}}\,\big)\,\big(\,\mathbf{S}_0^{-1} + \sum_{i\neq j}^{N}2\lambda(\xi_{ij})(\tilde{\mathbf{E}}_j\otimes\tilde{\mathbf{E}}_i) - \mathbf{S}_N^{-1}\,\big)\Big) \\
&\quad + (\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal\Big(\,\mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} + \sum_{i\neq j}^{N}(X_{ij} - \frac{1}{2})(\tilde{\boldsymbol{\tau}}_j\otimes\tilde{\boldsymbol{\tau}}_i) - \mathbf{S}_N^{-1}\tilde{\mathbf{W}}_N^{\mathrm{vec}}\,\Big) \\
&\quad - \frac{1}{2}\log|\frac{\mathbf{S}_0}{\mathbf{S}_N}| - \frac{1}{2}(\tilde{\mathbf{W}}_0^{\mathrm{vec}})^\intercal \mathbf{S}_0^{-1}\tilde{\mathbf{W}}_0^{\mathrm{vec}} + \frac{1}{2}(\tilde{\mathbf{W}}_N^{\mathrm{vec}})^\intercal \mathbf{S}_N^{-1}\tilde{\mathbf{W}}_N^{\intercal} \\
&\quad - \Big\{\tau_{iq}\log\tau_{iq} + (1-\tau_{iq}\log(1-\tau_{iq})\Big\}.
\end{aligned} \tag{D.15}$$

Bibliography

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, Upper Saddle River, New Jersey, 1993. (Cited in page 36.)

E. Airoldi, D. Blei, E. Xing, and S. Fienberg. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the International Biometrics Society Annual Meeting*, 2006. (Cited in page 67.)

E. Airoldi, D. Blei, S. Fienberg, and E. Xing. Mixed membership analysis of high-throughput interaction studies: relational data. *ArXiv e-prints*, 2007. (Cited in page 67.)

E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9: 1981–2014, 2008. (Cited in pages 43, 67, 70, and 80.)

H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information Theory*, pages 267–281, 1973. (Cited in page 20.)

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974. (Cited in page 20.)

R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Modern Physics*, 74:47–97, 2002. (Cited in pages 30 and 60.)

R. Albert, H. Jeong, and A.L. Barabasi. Diameter of the world-wide web. *Nature*, 401:130–131, 1999. (Cited in page 31.)

E.S. Allman, C. Matias, and J.A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *Annals of Statistics*, 37(6A):3099–3132, 2009. URL `http://www.imstat.org/aos/`. (Cited in pages 43, 71, and 73.)

E.S. Allman, C. Matias, and J.A. Rhodes. Parameters identifiability in a class of random graph mixture models. *ArXiv e-prints*, 2010. (Cited in page 43.)

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 31.)

H. Attias. Inferring parameters and structure of latent variable models by variational bayes. In K.B. Laskey and H. Prade, editors, *Uncertainty in Artificial Intelligence : proceedings of the fifth conference*, pages 21–30. Morgan Kaufmann, 1999. (Cited in page 55.)

A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999. (Cited in page 31.)

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 30.)

J.O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985. (Cited in page 13.)

J.M. Bernardo and A.F.M. Smith. *Bayesian theory*. Wiley, 1994. (Cited in page 13.)

P.J. Bickel and A. Chen. A non parametric view of network models and newman-girvan and other modularities. In *Proceedings of the National Academy of Sciences*, volume 106, pages 21068–21073, 2009. (Cited in page 35.)

C. Biernacki and G. Govaert. Using the classification likelihood to choose the number of clusters. *Computing Science and Statistics*, 29:451–457, 1997. (Cited in page 23.)

C. Biernacki, G. Celeux, and G. Govaert. An improvement on the nec criterion for assessing the number of clusters in a mixture model. *Pattern Recognition Letters*, 20:267–272, 1999. (Cited in pages 22 and 23.)

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Machine Intel*, 7:719–725, 2000. (Cited in pages 22, 23, 24, 51, 109, and 110.)

C. Biernacki, G. Celeux, and G. Govaert. Exact and monte carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, 140:2991–3002, 2010. (Cited in pages 51 and 56.)

C.M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, 2006. (Cited in pages 14, 16, and 18.)

C.M. Bishop and M. Svensén. Bayesian hierarchical mixtures of experts. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 57–64. U. Kjaerulff and C. Meek, 2003. (Cited in page 99.)

D. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. (Cited in page 67.)

P. Boer, M. Huisman, T.A.B. Snijders, C.E.G. Steglich, L.H.Y Wichers, and E.P.H Zeggelink. *StOCNET : an open software system for the advanced statistical analysis of social networks*. Groningnen:ProGAMMA/ICS, 2006. Version 1.7. (Cited in page 51.)

B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures and Algorithms*, 2005. (Cited in page 44.)

H. Bozdogan and S.L. Sclove. Multi-sample cluster analysis using akaike's information criterion. *Annals of the Institute of Statistical Mathematics*, 36: 163–180, 1984. (Cited in page 21.)

U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001. (Cited in page 36.)

A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33:309–320, 2000. (Cited in page 31.)

C.G. Broyden, R. Fletcher, D. Goldfarb, and D.F. Shanno. Bfgs method. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970. (Cited in page 79.)

K.P. Burnham and D.R. Anderson. *Model selection and multi-model inference: a practical information-theoric approach*. Springer-Verlag, 2004. (Cited in page 51.)

R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *Journal on Scientific and Statistical Computing*, 16:1190–1208, 1995. (Cited in page 79.)

J.G. Campbell, C. Fraley, F. Murtagh, and A.E. Raftery. Linear flaw detection in woven textildes using model based clustering. *Pattern recognition Letters*, 18:1539–1548, 1997. (Cited in page 22.)

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 pages 21 and 22.)

J. Chang. *The lda package*, 2010. Version 1.2. (Cited in page 80.)

A. Corduneanu and C.M. Bishop. Variational bayesian model selection for mixture distributions. In T. Richardson and T. Jaakkola, editors, *Artificial Intelligence and Statistics : proceedings of the eighth conference*, pages 27–34. Morgan Kaufmann, 2001. (Cited in page 55.)

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, 2001. (Cited in page 36.)

I. Csiszar and G. Tusnady. Information geometry and alternating minimization procedures. *Statistics and Decisions*, 1(1):205–237, 1984. (Cited in page 25.)

V.M. Dang. *Classification de données spaciales: modèles probabilistes et critères de partitionnement*. PhD thesis, Université de Technologie de Compiègne, 1998. (Cited in page 14.)

L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *J Stat Mech*, 2005. (Cited in page 29.)

A. Dasgupta and A.E. Raftery. Detecting features in spacial point processes with clutter via model based clustering. *Journal of the American Statistical Association*, 93:294–302, 1998. (Cited in page 22.)

J. Daudin, F. Picard, and S. Robin. A mixture model for random graphs. *Statistics and Computing*, 18:1–36, 2008. (Cited in pages 2, 43, 51, 54, 55, 56, 57, 58, 61, 64, 71, 77, and 78.)

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood for incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977. (Cited in page 14.)

S.N. Dorogovtsev, J.F.F. Mendes, and A.N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letter*, 85:4633–4636, 2000. (Cited in page 31.)

C. Elkan. Using the triangle inequality to accelerate kmeans. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 147–153, 2003. (Cited in page 11.)

E. Estrada and J.A. Rodriguez-Velazquez. Spectral measures of bipartivity in complex networks. *Physical Review E*, 72:046105, 2005. (Cited in page 30.)

M.G. Everett and S.P. Borgatti. Analyzing clique overlap. *Connections*, 21: 49–61, 1998. (Cited in pages 67 and 80.)

B. Everitt. *Cluster analysis*. Wiley, 1974. (Cited in page 38.)

S.E. Fienberg and S. Wasserman. Categorical data analysis of single sociometric relations. *Sociological Methodology*, 12:156–192, 1981. (Cited in page 43.)

R. Fletcher. *Practical methods of optimization*. Wiley, 1987. (Cited in page 13.)

S. Fortunato. Community detection in graphs. *Physics Reports*, 3-5:75–174, 2010. (Cited in page 30.)

O. Frank and F. Harary. Cluster inference by using transitivity indices in empirical graphs. *Journal of the American Statistical Association*, 77:835–840, 1982. (Cited in page 43.)

L. Freeman. A set of measures of centrality based upon betweenness. *Sociometry*, 40:35–41, 1977. (Cited in page 36.)

Q. Fu and A. Banerjee. Multiplicative mixture models for overlapping clustering. In *Proceedings of the IEEE International Conference on Data Mining*, pages 791–796, 2008. (Cited in page 68.)

A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian data analysis*. Chapman and Hall, 2004. (Cited in page 13.)

M. Girvan and M.E.J. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*, volume 99, pages 7821–7826, 2002. (Cited in page 35.)

A. Goldenberg, A.X. Zheng, S.E. Fienberg, and E.M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2 (2):129–233, 2010. (Cited in page 30.)

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. In *Neural Information Processing Systems*, volume 18, pages 475–482, 2005. (Cited in page 67.)

S.F. Gull. Developpments in maximum entropy data analysis. In *Maximum Entropy and Bayesian Methods*, pages 53–71. Kluwer, 1989. (Cited in page 13.)

M.S. Handcock, A.E. Raftery, and J.M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society*, 170:1–22, 2007. (Cited in pages 38 and 39.)

T. Hastie, R. Tibshirani, and J. Friedman. *The element of statistical learning*. Springer, 2001. (Cited in page 13.)

R.J. Hathaway. Another interpretation of the em algorithm for mixture distributions. *Statistics & Probability Letters*, 4:53–56, 1986. (Cited in pages 23, 25, and 54.)

K. Heller and Z. Ghahramani. A nonparametric bayesian approach to modeling overlapping clusters. In *In Proceedings of The 11th International Conference On AI And Statistics*, 2007. (Cited in pages 67 and 81.)

K. Heller, S. Williamson, and Z. Ghahramani. Statistical models for partial membership. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 392–399, 2008. (Cited in pages 67 and 81.)

M.E. Hodgson. Reducing computational requirements of the minimum-distance classifier. *Remote Sensing of Environments*, 25:117–128, 1998. (Cited in page 11.)

A.E. Hoerl and R. Kennard. Ridge regression:biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970. (Cited in page 13.)

P.D. Hoff, A.E. Raftery, and M.S. Handcock. Latent space approaches to social network analysis. *Journal of the Royal Statistical Society*, 97:1090–1098, 2002. (Cited in page 38.)

J.M. Hofman and C.H. Wiggins. A bayesian approach to network modularity. *Physical Review Letters*, 100:258701, 2008. (Cited in pages 29, 41, 43, 53, 57, 58, 59, 61, and 64.)

P. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: some first steps. *Social Networks*, 5:109–137, 1983. (Cited in page 43.)

T.S. Jaakkola and M.I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000. (Cited in pages 96, 97, 120, and 125.)

C.J. Jeffery. Moonlighting proteins. *Trends in Biochemical Sciences*, 24:8–11, 1999. (Cited in page 67.)

H. Jeffreys. An invariant form for the prior probability in estimations problems. In *Proceedings of the Royal Society of London. Series A*, volume 186, pages 453–461, 1946. (Cited in pages 14 and 53.)

R. Kass and A.E. Raftery. Bayes factor. *Journal of the American Statistical Association*, 90:773–795, 1995. (Cited in page 22.)

C. Kemp, T.L. Griffiths, and J.B. Tenenbaum. Discovering latent classes in relational data. Technical report, MIT, 2004. (Cited in page 43.)

A.B. Koehler and E.H. Murphee. A comparison of the akaike and schwarz criteria for selecting model order. *Applied Statistics*, 37:187–195, 1988. (Cited in page 21.)

P.N. Krivitsky and M.S. Handcock. *The latentnet package*. Statnet project, 2009. Version 2.1-1. (Cited in page 39.)

P.N. Krivitsky, M.S. Handcock, A.E. Raftery, and P.D. Hoff. Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31:204–213, 2009. (Cited in page 69.)

V. Lacroix, C.G. Fernandes, and M.-F. Sagot. Motif search in graphs: pplication to metabolic networks. *Transactions in Computational Biology and Bioinformatics*, 3:360–368, 2006. (Cited in pages viii, 30, 33, and 61.)

P. Latouche, E. Birmelé, and C. Ambroise. *Bayesian methods for graph clustering*, pages 229–239. Springer, 2009. (Cited in pages 3, 6, 71, 78, and 80.)

P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block model with application to the french political blogosphere. *Annals of Applied Statistics*, 2010a. to appear. (Cited in pages 3 and 6.)

P. Latouche, E. Birmelé, and C. Ambroise. Variational bayes inference and complexity control for stochastic block models. *Statistical Modelling*, 2010b. to appear. (Cited in pages 3 and 6.)

B.G. Leroux. Consistent estimation of amixing distribution. *Annals of Statistics*, 20:1350–1360, 1992. (Cited in page 22.)

D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992. (Cited in page 13.)

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967. (Cited in page 11.)

M. Mariadassou, S. Robin, and C. Vacher. Uncovering latent structure in valued graphs: a variational approach. *Annals of Applied Statistics*, 4(2), 2010. (Cited in pages 38, 51, and 58.)

D. Martin, C. Brun, E. Remy, P. Mouren, D. Thieffry, and B. Jacq. Gotoolbox: functional analysis of gene datasets based on gene ontology. *Genome Biology*, 5(12), 2004. (Cited in page 91.)

G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. New York: John Wiley, 1997. (Cited in page 14.)

G McLachlan and D. Peel. *Finite mixture models*. New York: Jogn Wiley, 2000. (Cited in pages 9, 20, and 22.)

R. Milo, S. Shen-Orr, S. Itzkovitz, D. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002. (Cited in pages viii, 30, 32, and 91.)

A.W. Moore. The anchors hierarch: using the triangle inequality to survive high dimensional data. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 397–405, 2000. (Cited in page 11.)

J.L. Moreno. *Who shall survive?: a new approach to the problem of Human interrelations*. Nervous and Mental Disease Publishing, Washington DC, 1934. (Cited in page 29.)

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, Dordrecht, 1998. Kluwer. (Cited in pages 25 and 54.)

M. Newman and E. Leicht. Mixture models and exploratory analysis in networks. In *Proceedings of the National Academy of Sciences*, volume 104, pages 9564–9569, 2007. (Cited in pages 29 and 43.)

M.E.J Newman. Scientific collaboration networks: Ii. shortest paths, weighted networks, and centrality. *Physical Review E*, 64:016132, 2001. (Cited in page 36.)

M.E.J. Newman. Mixing patterns in networks. *Physical Review E*, 67:026126, 2003. (Cited in page 29.)

M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review Letter*, 69, 2004. (Cited in page 38.)

M.E.J Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004. (Cited in pages 35 and 36.)

J. Nocedal and S.J. Wright. *Numerical optimization*. Springer, 1999. (Cited in page 13.)

K. Nowicki and T.A.B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087, 2001. (Cited in pages 30, 43, 51, 52, 65, and 72.)

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 pages 67, 80, and 86.)

G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. *CFinder, the community cluster finding program*, 2006. Version 2.0.1. (Cited in pages 67 and 80.)

G. Palla, A.L Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, 2007. (Cited in page 30.)

T. Park and G. Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008. (Cited in page 13.)

A.E. Raftery. Bayesian model selection in social research (with discussion by andrew gelman, donald b. rubin, robert m. hauser, and a rejoinder), 1995. (Cited in page 22.)

B.D. Ripley. *Pattern recognition and neural networks*. Cambridge: Cambridge University Press, 1996. (Cited in page 22.)

V. Rmasubramanian and K.K. Paliwal. A generalized optimization of the k-d tree for fast nearest-neighbour search. In *Proceedings of the IEEE Region 10 International Conference*, pages 565–568, 1990. (Cited in page 11.)

C.P. Robert. *The Bayesian choice: a decision-theoretic motivation*. Springer-Verlag, 1994. (Cited in page 110.)

K. Roeder and L. Wasserman. Practical density estimation using mixture of normals. *Journal of the American Statistical Association*, 92:894–902, 1997. (Cited in page 22.)

S.F. Sampson. *Crisis in a cloister*. PhD thesis, Cornell University, 1969. (Cited in pages ix and 40.)

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6: 461–464, 1978. (Cited in page 22.)

S.L. Sclove. Application of model selection criteria to some problems in multivariate analysis. *Psychometrika*, 52:333–343, 1987. (Cited in page 21.)

J. Scott. *Social network analysis: a handbook*. Sage publications, 2000. (Cited in page 38.)

T.A.B. Snijders and K. Nowicki. Estimation and prediction for stochastic block-structures for graphs with latent block structure. *Journal of Classification*, 14:75–100, 1997. (Cited in page 30.)

G. Soromenho. Comparing approaches for testing the number of components in a finite mixture model. *Computational Statistics*, 9:65–78, 1993. (Cited in page 21.)

S.H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001. (Cited in page 31.)

M. Svensén and C.M. Bishop. Robust bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2004. (Cited in page 55.)

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, B*, 58:267–288, 1996. (Cited in page 13.)

G. Wahba. A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem. *Numerical Mathematics*, pages 383–393, 1975. (Cited in page 13.)

D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998. (Cited in page 30.)

H.C. White, S.A. Boorman, and R.L. Breiger. Social structure from multiple networks. i. blockmodels of roles and positions. *American Journal of Sociology*, 81:730–780, 1976. (Cited in pages ix, 40, and 43.)

H. Zanghi, C. Ambroise, and V. Miele. Fast online graph clustering via erdös renyi mixture. *Pattern Recognition*, 41(12):3592–3599, 2008. (Cited in pages viii, 30, 34, 51, and 86.)

H. Zanghi, S. Volant, and C. Ambroise. Clustering based on random graph model embedding vertex features. *Pattern Recognition Letters*, 31(9):830–836, 2010. (Cited in page 38.)

Ce document a été préparé à l'aide de l'éditeur de texte GNU Emacs et du logiciel de composition typographique LaTeX 2ε.

Titre Modèles de graphes aléatoires à structure cachée pour l'analyse des réseaux

Résumé Le résumé en français ($\approx$ 1000 caractères)

Mots-clés Les mots-clés en français

Title Le titre en anglais

Abstract Le résumé en anglais ($\approx$ 1000 caractères)

Keywords Les mots-clés en anglais