Deep graphical models and inference strategies for the analysis of networks comprising textual edges

Rémi Boutin

Université Paris Cité

École doctorale de Sciences Mathématiques de Paris-Centre (ED 386)
Laboratoire Mathématiques Appliquées à Paris 5, CNRS, UMR 8145

Deep graphical models and inference strategies for the analysis of networks comprising textual edges

par Rémi BOUTIN

Thèse de doctorat de mathématiques appliquées

Dirigée par
Pierre LATOUCHE et Charles BOUVEYRON

Présentée et soutenue publiquement le 14 décembre 2023
devant un Jury composé de :

Sophie DONNET Directrice de recherche, INRAE, Université Paris Saclay	Rapporteuse
Brendan MURPHY Professeur, University College Dublin	Rapporteur
Etienne CÔME Chargé de recherche, Université Gustave Eiffel	Examinateur
Tabea REBAFKA Maître de conférence, Sorbonne Université	Examinatrice
Cinzia VIROLI Professeure, Università di Bologna	Examinatrice
Charles BOUVEYRON Professeur, Université Côte d'Azur	Co-directeur
Pierre LATOUCHE Professeur, Université Clermont-Auvergne	Directeur

Abstract

In this manuscript, we shall develop new methodologies to cluster nodes of networks, possibly holding textual edges. We aim at providing an end-to-end modelling, capable of using the texts exchanged between the nodes as well as the network topology to extract salient information at the core of the dataset. This work is motivated by questions arising in different fields such as social sciences. Gathering and understanding large datasets from social media may help researchers to answer questions, regarding the way a policy may be perceived, for instance. We adopt a probabilistic modelling framework to classify nodes and analyse texts. Among other things, these models provide information on the uncertainty of our estimates as well as a framework that has proven to be robust historically. Furthermore, in order to benefit from the efficiency of deep neural networks to encode complex types of data, our methodologies strive to include them within a probabilistic framework. Several analyses of real data are provided. In particular, during several months preceding the 2017 French presidential election, each publication of one social media, as well as their re-publications, involved with one of the candidates were gathered to form a data base. Our methodology helps understanding the groups present on the social media as well as the way interactions were taking place during this particular time period. Python implementations associated with the methodologies developed in this manuscript have been made public.

Keywords Statistical network analysis, topic modelling, model-based node clustering, variational inference, Bayesian variational inference, variational graph autoencoder

Résumé

Dans ce manuscrit, nous développerons de nouvelles méthodologies pour regrouper les nœuds de réseaux comportant, éventuellement, des arêtes textuelles. Notre objectif est de fournir une modélisation de bout-en-bout, capable d'utiliser les textes échangés entre les nœuds ainsi que la topologie du réseau pour extraire les motifs à l'origine de ce jeu de données. Ce travail est motivé par des questions qui se posent dans différents domaines comme en sciences sociales par exemple. La collecte et la compréhension de gros volumes de données provenant des réseaux sociaux peuvent, par exemple, aider les chercheurs à répondre à des questions concernant la manière dont une politique est perçue. Nous adoptons un cadre de modélisation probabiliste pour classifier les nœuds et analyser les textes. Entre autres, ces modèles renseignent sur l'incertitude de nos estimations et fournissent un cadre qui s'est avéré robuste historiquement. De plus, afin de bénéficier de l'efficacité des réseaux de neurones profonds pour encoder des types de données complexes, nos méthodologies combinent les modèles probabilistes avec les derniers avancement dans ce domaine. Plusieurs analyses de données réelles sont fournies. En particulier, durant plusieurs mois précédant l'élection présidentielle française de 2017, chaque publication d'un média social, ainsi que leurs rediffusions, impliquant l'un des candidats ont été rassemblées dans une base de données. Notre méthodologie permet de comprendre les groupes présents sur les réseaux sociaux ainsi que la manière dont les interactions se sont établies au cours de cette période particulière. Les implémentations Python associées aux méthodologies développées dans ce manuscrit ont été rendues publiques.

Mots-clés Analyse statistique des réseaux, modélisation de thèmes, classification de nœuds basée sur des modèles, méthodes variationelle pour l'inférence, Méthodes variationelle pour l'inférence bayésienne, autoencodeur de graphe variationnel

Contributions

De nombreuses interactions impliquent des textes, comme dans les réseaux d'auteurs/co-auteurs, les réseaux sociaux ou les emails par exemple. Lorsque le nombre d'interactions augmente, il devient rapidement difficile de comprendre et de tirer du sens de ces données. Pour les représenter et les analyser, une structure très générale et complexe, appelée *réseau*, a été développée au XXème siècle. Un réseau est composé de *nœuds*, connectés ou non, tel que deux nœuds sont connectés s'ils sont liés par une *arête* ou arc dans le cas orienté. Les arcs, ou plus généralement connexions, peuvent parfois être associées à l'échange d'un texte dans certaines applications, comme dans les exemples cités précédemment. La capacité de stockage n'ayant de cesse d'augmenter, les réseaux représentant des échanges textuelles deviennent de plus en plus fréquents et volumineux. Afin de rendre de tels réseaux intelligibles pour les humains, il apparaît nécessaire de développer des méthodologies pour obtenir des informations sur les textes échangés entre les nœuds ainsi que sur la structure des connexions. Des exemples concrets d'utilisation de l'analyse de réseaux en sciences humaines sont présentés dans Borgatti et al. (2009). Citons l'utilisation des réseaux pour étudier l'impacte de la structure d'un groupe de travail sur la vitesse de transmission de l'information au sein de celui-ci. Contre-intuitivement, bien que le plus court-chemin moyen puisse être plus petit dans un réseaux très connecté, un groupe hiérarchique avec, en son milieu, une personne identifiée comme *au centre* par tous les nœuds, pourra faire passer l'information plus vite dans le réseau. Ainsi, plusieurs questions se posent lorsque l'on souhaite analyser et comprendre ce type de données. Par exemple, dans le cadre de réseaux échangeant des textes, il est légitime de se demander quels sont les thèmes abordés dans ces textes ? Y a-t-il des thèmes récurrents autour desquels se forment les échanges ? Y a-t-il des comportements et des discussions faisant émerger une structure sous-jacente, organisant le réseau ? Dans ce manuscrit, nous proposons trois nouvelles méthodologies pour comprendre la formation d'un réseau que nous présentons ci-dessous.

Comprendre la structure à l'origine d'interactions impliquant du texte

Pour commencer, nous proposons une nouvelle méthodologie repoussant les performances des modèles existants. La rencontre de l'analyse de réseaux et de la modélisation thé-

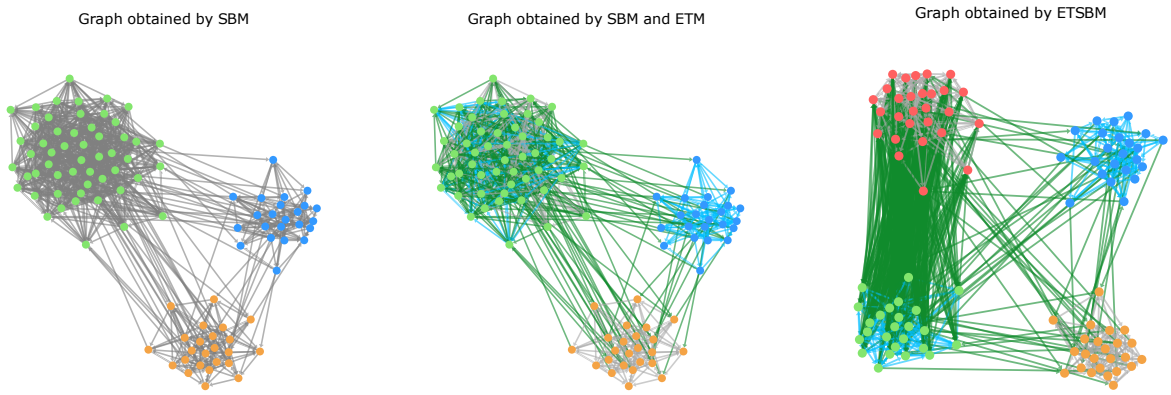


Figure 1: Exemple de l'amélioration apportée par ETSBM en incluant les thèmes découvert à l'aide d'un modèle thématique profond (correspondants à la couleur des arcs), dans la détection des clusters des nœuds (correspondants à la couleur des noœuds).

matique est récente et les méthodes proposées reposaient jusqu'alors, sur la fréquence des mots présents dans les documents, sans incorporer de sens sémantique. Notre première contribution, le *embedded topic in the stochastic block model (ETSBM)*, pallie ce manque en s'appuyant sur les avancées de la modélisation de thèmes. Elle a pour objet de classifier les nœuds, et d'analyser simultanément les textes échangés, en intégrant des représentations de mot pré-entraînées. Ce sont ces représentations qui, possiblement pré-entraînées sur un large corpus, peuvent incorporer le sens sémantique des mots. Par conséquent, notre méthode fournit des clusters de nœuds, en utilisant les connexions et les thèmes abordés dans les documents échangés. Un autre aspect important de cette méthodologie est sa flexibilité pour modéliser tout type de structure de connexion. Pour éclairer ce besoin, nous présentons l'exemple d'un réseau social organisé autour de quelques comptes centraux, et des comptes périphériques, faiblement connectés entre eux mais très connectés aux autre comptes centraux. Ce type de structure de connexion ne peut pas être détecté par ce qu'on appelle des méthodes de *détection de communautés*. En effet, nous réserverons le terme de *communauté* à des groupes de nœuds densément connectés entre eux mais faiblement connectés au reste du graphe. Au contraire, le terme de *cluster* désignera un groupe de nœuds partageant un modèle de connectivité similaire, pouvant être différent du concept de communauté. Par exemple, contrairement aux communautés, un motif en étoile est défini par deux clusters avec de faibles probabilités d'intra-connexion et de grandes probabilités d'inter-connexion (Latouche, Birmele, Ambroise, 2012). Ce type de cluster ne peut pas être déceler par les méthodes de détection de communautés. La figure 1 illustre la nécessité de combiner le clustering de graphes et la modélisation de sujets afin de distinguer les quatre clusters et d'obtenir des sujets plus significatifs pour chaque cluster. Pour modéliser les sujets échangés entre les nœuds, les documents sont encodés avec un réseau de neurones profond pour bénéficier de leur flexibilité. Le décodeur est constitué de vecteurs représentants les mots et les thèmes, comme dans Dieng, Ruiz, Blei (2020), et nous proposons une auto-agrégation des documents au niveau du clus-

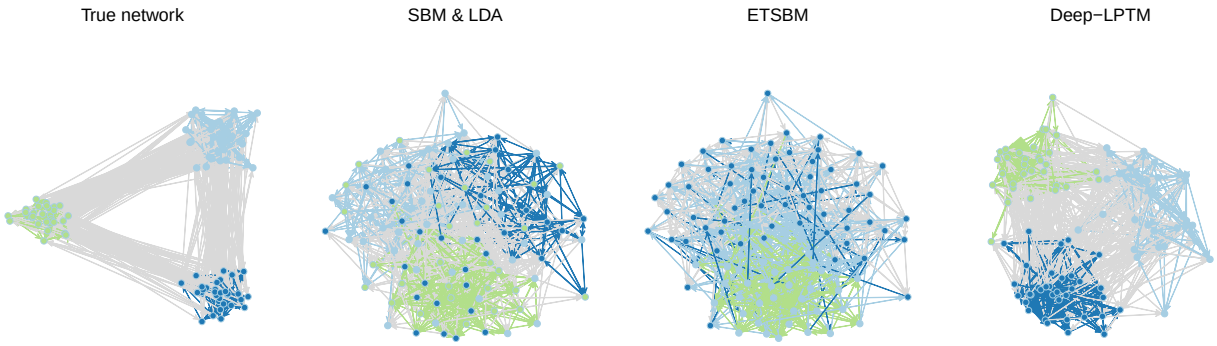


Figure 2: Illustration de l'avantage procuré par Deep-LPTM pour représenter un réseaux comportant des textes sur les arcs avec des algorithmes utilisés à posteriori de l'analyse. La couleur des nœuds indique son cluster et la couleur des arcs correspond au thème principal du document correspondant.

ter, en Q^2 méta-documents avec Q le nombre de clusters. En particulier, notre stratégie d'inférence est capable d'optimiser directement la construction des méta-documents via la procédure d'inférence.

Représentation de réseaux comportant du textes

Dans un second temps, nous proposons le deep latent topic model (Deep-LPTM), une méthodologie capable de produire une représentation de graphes incorporant l'analyse des thèmes des documents et la modélisation des communautés pour déterminer la position des nœuds. Le modèle que nous proposons est le premier à regrouper simultanément les nœuds d'un réseau, à découvrir les sujets dans les textes échangés entre les nœuds et à produire une représentation à la fois des sujets et des arcs dans un espace euclidien, en se basant sur une approche probabiliste. Dans ce but, nous proposons un modèle génératif supposant que chaque nœud et chaque arcs sont représentés dans un espace latent par un mélange de densités gaussiennes. Ce faisant, nous incorporons le clustering dans le processus génératif à deux niveaux. D'un côté, les nœuds sont représentés par un mélange dont chaque composante modélise un cluster. De l'autre, les documents sont également modélisés par un mélange de sorte que chaque composante représente les thèmes abordé entre une paire de clusters. De plus, notre modèle se distingue des méthodes précédentes en permettant à chaque nœud ou arc d'être représenté par une position latente et pas seulement par le groupe auquel il appartient. Ceci est illustré dans la figure 2. Un algorithme externe, à savoir l'algorithme de Fruchterman-Reingold (Fruchterman, Reingold, 1991), considérant la présence de connexions uniquement dans le réseau, a dû être utilisé pour la représentation graphique des réseaux non estimés par Deep-LPTM. La figure de droite présente les résultats Deep-LPTM. Contrairement aux méthodes précédentes, Deep-LPTM est capable de rassembler des informations sur la structure du réseau ainsi que les documents échangés dans les positions des nœuds tout en trouvant la véritable partition et les véritables sujets des nœuds. Une représentation du graphique est directe-

ment obtenue par la procédure d'estimation de sorte qu'aucun algorithme de représentation graphique externe n'est nécessaire. Comme nous le verrons, les positions des nœuds sont calculées en considérant à la fois les connexions et le contenu des documents correspondants. Nous dérivons un algorithme E-M variationnelle (VEM) en deux étapes pour estimer les paramètres du modèle ainsi que les paramètres a posteriori des positions latentes. La première étape s'appuie sur des formules analytiques pour mettre à jour les probabilités d'appartenance aux clusters ainsi que les paramètres de mélange. La deuxième étape utilise un algorithme de descente de gradient stochastique pour maximiser la borne inférieure de la vraisemblance, en optimisant par rapport aux paramètres de l'autoencodeur de graphe variationnel (VGAE) et aux paramètres de l'autoencoder variationnel pour la modélisation de thèmes. En particulier, la modélisation des thèmes est capable de tirer parti des représentations pré-entraînées. Par conséquent, introduire un sens sémantique dans les représentations de mots est possible ainsi qu'apprendre la représentation de zéro. Afin de choisir les nombres pertinents de clusters et les dimensions de l'espace latent, nous introduisons la vraisemblance classifiante et latente intégrée (IC2L) pour effectuer la sélection de modèle. Ce critère étend la vraisemblance classifiante intégrée (ICL), conçu pour les modèles de mélange, en prenant en compte les représentations latentes des nœuds et des arc. La pertinence du critère est fortement confirmée par l'évaluation sur les données synthétiques ainsi que par le cas d'utilisation réel fourni. De plus, en sélectionnant de faibles dimensions concernant l'espace des représentations des nœuds, IC2L favorise les modèles dotés d'une capacité de représentation forte et directe.

Généralisation de la représentation de réseaux via une modélisation par bloc

Notre dernier travail se concentre sur l'analyse de réseaux, sans texte, dans l'optique de combiner les autoencoders de graphes variationnels avec une modélisation par bloc. Dans certains domaines, les experts peuvent fournir des connaissances sur la structure de la connectivité d'un réseau, mais il est souvent nécessaire de la déduire des données. Cette flexibilité a été apportée par des approches de modélisation par bloc, permettant d'analyser tout type de structure de connectivité au sein du réseau. Toutefois, cette flexibilité se fait au détriment de la représentation. En effet, ces méthodes ne fournissent pas de représentations continues des nœuds, mais uniquement des *méta-réseaux*, où les clusters sont représentés par des nœuds, la taille des clusters par la taille des nœuds et le nombre de connexions entre clusters par la largeur des arcs. Cette méta-représentation peut cacher certaines propriétés de nœuds spécifiques. Par exemple, un nœud d'un cluster peut également être connecté à un autre groupe plus intensément que le reste de son cluster. Il est souhaitable que cela transparaisse dans la représentation estimée, ce nœud pouvant être crucial dans le réseau, précisément parce qu'il se situe entre deux clusters. Pour représenter le réseau, les méthodologies basées sur la position, c'est à dire sur une représentation continue des nœuds, s'appuient sur la similarité entre les représentations latentes des

nœuds pour évaluer leur probabilité de connexion. Malheureusement, ces méthodes se concentrent sur l'estimation de communautés et impliquent une propriété de transitivité dans le réseau. Pour palier ce manque, nous proposons une marginalisation d'un modèle par bloc stochastique, encodé par un réseau de neurones adapté. En notant $\Pi \in \mathcal{M}_{Q\times Q}((0,1))$ la matrice de probabilité de connexion, $\mathbf{A} \in \mathcal{M}_{N\times N}(\{0,1\})$ la matrice d'adjacence binaire, et $\boldsymbol{\eta} = (\eta_i)_{i=1,\dots,N} \in \mathcal{M}_{N\times,Q}((0,1))$, chacune tirée selon une loi logistique normale, nous supposons que la probabilité de connexion est donnée par :

$$p(\mathbf{A} \mid \boldsymbol{\eta}, \Pi) = \prod_{i\neq j} (\eta_i^\top \Pi \eta_j)^{A_{ij}} (1 - \eta_i^\top \Pi \eta_j)^{1-A_{ij}}.$$

L'estimation de ce modèle repose sur une approche variationnelle combinée à des formes analytiques pour mettre à jour les paramètres. Des premiers résultats sont présentés sur les représentations obtenues pour différents types de structures, telles que des étoiles, des hub ou des communautés.

Nos contributions ont été partagées à travers une publication dans une revue internationale à comité de lecture,

- **Embedded topics in the stochastic block model**, joint work with Pierre Latouche and Charles Bouveyron, *Statistics and Computing (2023)*,

une autre a été soumise à un journal,

- **The Deep Latent Position Topic Model for Clustering and Representation of Networks with Textual Edges**, joint work with Pierre Latouche and Charles Bouveyron, *Pre-print hal-04068665 (2023)*,

et notre dernier travail est en cours de finalisation en vue de la soumission à un journal scientifique,

- **The Deep Latent Position Block Model**, joint work with Pierre Latouche and Charles Bouveyron.

Nos contributions sont accompagnées d'implémentations en Python, la première est disponible à l'adresse https://plmlab.math.cnrs.fr/rboutin/etsbm_package, et la deuxième à l'adresse https://plmlab.math.cnrs.fr/rboutin/deeplptm_package.

Remerciements

Ces dernières lignes sonnent la fin de trois années riches en découvertes. Ces découvertes, si nombreuses soient-elles, ont eu lieu grâce à de nombreuses personnes. Ces quelques mots viennent les remercier. Je ne pourrais citer tout le monde, j'espère que vous ne m'en tiendrez pas rigueur.

Je voudrais commencer par adresser des remerciements sincères et chaleureux à mes directeurs de thèse. Charles, malgré la distance entre Nice et Paris, puis Clermont, nous avons réussi à nous voir, à de nombreuses reprises. Tu as su entretenir un encadrement bienveillant et positif, et tu m'as redonné confiance dans les moments de doute. Tu es resté sincère lors des prises de décisions importantes, comme aux JDS à Lyon, en m'exposant les différentes possibilités qui s'offraient à moi et en me laissant choisir. Tout cela a grandement contribué au travail accompli durant ces trois années et à la réalisation de ce manuscrit, sois-en remercié.

Pierre, après trois ans à se côtoyer quotidiennement, je peux maintenant te remercier de m'avoir considéré d'égal à égal scientifiquement et humainement, et ce dès le début de la thèse (quand bien même nos discussions et mes erreurs me rappelaient rapidement que j'étais encore loin de cette égalité). Tu as su me donner le goût pour les modèles graphiques, leur capacité de modélisation, et leurs intérêts pour d'autres disciplines. Tu as été compréhensif lors de moments difficiles, ce qui m'a permis de mener cette thèse dans les meilleures conditions. Pour tout cela, je te remercie. Mais je suis bien obligé de te remercier aussi pour m'avoir permis de découvrir Clermont-Ferrand, ses volcans et son festival du court-métrage ! À tous les deux, j'espère que nous nous retrouverons, que ce soit pour des collaborations futures, à des conférences ou simplement pour discuter.

Je tiens également à remercier, Sophie Donnet et Brendan Murphy, pour avoir accepté de rapporter ma thèse. Que ce soit à Cargèse en 2021 pour Brendan, ou au workshop "Bayesian Methods for the Social Sciences" à l'Institut Henri Poincaré pour Sophie, vos présentations ont été précieuses, et m'ont conforté dans mes envies de recherche. Vos retours sur mon manuscrit ont permis d'en améliorer la qualité. Je tiens également à remercier Étienne Côme, Tabea Rebafka ainsi que Cinzia Viroli pour avoir accepté d'être examinateur et examinatrices de ma soutenance. Je suis également reconnaissant envers Julien Chiquet et Olivier Bouaziz de m'avoir encouragé lors de mes comités de suivis individualisés durant ces trois années.

Au MAP5, je tiens à remercier tous.tes les membres du laboratoire, permanent.e.s, doctorant.e.s, post-doc, et invité.e.s, avec qui j'ai partagé des repas sur la terrasse mythique

du labo. Anne, je te remercie d'avoir toujours été disponible et accueillante en tant que directrice de laboratoire. Les méandres de l'administration en auraient perdu plus d'un.e sans ton écoute, ta gentillesse et ton efficacité Marie-Hélène, sois-en ici remerciée. C'est au tour des doctorant.e.s, vous êtes trop nombreux pour que je fasse un paragraphe sur chacun d'entre vous, bien que ce serait amplement mérité, alors cette courte ligne vient (trop) rapidement vous remercier pour ces nombreux rires, échanges mathématiques (parfois) et amicaux (toujours). Merci à vous Alessandro, Alexander, Antoine M, Antoine S, Antòn, Apolline, Arthur, Cécile, Chabane, Charles, Charlie, Diala, Eloi, Florian, Guillaume, Herb, Julianna, Keanu, Martin (le seul qui rentre dans la section Paris et Clermont!), Laurent, Léonard, Loïc, Lucia, Marie, Mariem, Mehdi, Safa, Sergio, Thaïs, Yen.

Je me dois de faire une mention spéciale à Zoé, qui a presque réussi à me faire regarder Top Chef et Koh Lanta et qui m'a aidé lors des étapes administratives de la fin de thèse. Ariane, ce GIF, avec l'autre Rémi, dans le bureau du 8ème étage reste l'un de mes plus gros fous rires de ces trois dernières années. Sonia, merci pour cette entrée « en fanfare » lors d'un verre rue de Buci. Ousmane, je suis désolé. Désolé que tu croies encore à une victoire de ManU en LdC ! J'espère que l'on se recroisera vite. Pierre-Louis, quand je suis triste, il me suffit de penser à ton imitation phare pour éclaircir ma journée.

Je ne sais pas où te remercier Rémi, car tu ne rentres dans aucune case. Ou peut-être dans toutes. J'aimerais mettre mille anecdotes ("the law says...", le fameux GIF, ta tête face à l'imitation de Pilou), mais je tiens surtout à te remercier de m'avoir soutenu dans les moments difficiles, et de continuer à être présent, même depuis Copenhague.

Ensuite, à Clermont-Ferrand, il y a du monde à remercier. Tout d'abord Valérie, qui a été très accueillante, a pris le temps pour m'aider dans les processus administratifs, rendant mes jours à Clermont-Ferrand bien plus aisés. Émilien, j'espère avoir ton assurance lorsque l'on me demandera la période et l'endroit que j'aimerais avoir connu. L'entrée sans laquelle le bureau 258 ne serait pas le même, « Eh bonjour» Léo ! Merci à Hoa. Sue, tu es l'une des rares qui arrive à entrer en contact avec les fantômes dans un labo de maths, c'est surprenant la première fois. Tristan, je suis encore impressionné par ton « sommaire body » (enroulé dans un plaid et une écharpe). Le Vincent, il a une manière de toquer des plus agréables (toc toc toc toc toc ...) et une qualité d'imitateur telle que « si tu prends mon frère qui imite mon père qui imite Jacques Chirac, t'as [Vincent] qui imite Nikos » comme dirait Benjamin Tranié. Clément, merci pour cette boîte de cassoulet de Castelnaudary que j'ouvre en écrivant ces lignes. Martin A., nos verres et sorties nocturnes ont été de beaux moments en ces terres clermontoises, merci à toi ! Enfin il y a Julian et sa passion pour le cacao. Mes mollets me disent d'écrire des non-remerciements, mais ils oublient combien j'ai été heureux de rencontrer un féru de cinéma, assez fou pour aller voir *Retribution* avec moi, ou de s'extasier devant *La mouche*, avant une bière au Hops'n'Roll. Merci beaucoup.

En outre, je vous remercie de m'avoir fait vivre la recherche non pas comme un défi en solitaire, mais comme une aventure enrichie par les interactions.

Cette thèse aurait été bien différente sans ceux qui ont été présent.e.s à l'extérieur du monde académique.

Alors merci à toi, Cyril ! Nous savons combien tu m'as aidé à différentes étapes de la

thèse. Victor, Maël, Quentin, Aurélie nos retrouvailles ont été des bouffées d'air qui ont ponctué ces trois années.

Je tiens également à te remercier Flora, car tu as supporté une partie du poids de cette thèse. Évidemment, s'il y en a une, l'autre n'est pas bien loin. Alors merci Zoé G, pour avoir passé un noël « covidés » ensemble (pas sûr que ça te ravisse comme remerciement), et de m'avoir prêté ce qui est devenu mon livre préféré.

À ceux que j'ai rencontrés lors de mes études et qui continuent de m'accompagner, Hassan, Lo Po, Jean-Maël, Apollinaire, je vous dis un grand merci. Hass, on en a rêvé de ces moments avec un chocolat et un muffin en haut des marches de Tolbiac. Lo Po, merci d'avoir été là et promis j'arrête de te suivre (le 11ème, l'Auvergne)... jusqu'au prochain déménagement.

Est-ce que ce manuscrit aurait vu le jour, si tu ne m'avais pas soutenu pendant ces semaines d'écritures ? Rien n'est moins sûr... Merci Marie pour m'avoir fait découvrir le lac d'Aydat, où le manuscrit pesait beaucoup moins lourd.

Merci à ma famille pour son soutien inconditionnel. Céline, tu as su me montrer que l'abnégation, dans le temps long, pouvait réserver de bien belles surprises. Audrey, je pense souvent au courage qu'il faut pour suivre ses envies personnelles et professionnelles et tracer son chemin, merci de ton exemple. Papa, Maman, vous m'avez toujours poussé à « faire ce que je voulais », ce qui paraissait bien naturel à l'époque, et qui semble aujourd'hui une chance inestimable. Je ne vous remercierai jamais assez de m'avoir donné goût au sport, au cinéma, aux sciences, à la musique et de nombreuses autres choses.

Contents

List of Figures

List of Tables

Notations

Sets and variables

N	The number of nodes.
M	The number of edges.
Q	The number of clusters.
K	The number of topics.
V	The size of the vocabulary.
$\mathbf{A} \in \mathcal{M}_{N\times N}(\{0,1\})$	The binary adjacency matrix.
$\mathbf{C} \in \mathcal{M}_{N\times Q}(\{0,1\})$	The node cluster memberships.
$\mathbf{W} \in \mathcal{M}_{M\times V}(\mathbb{N})$	The bag of words.
Δ_K	The simplex of dimension K.
$\mathbf{I}_n$	The identity matrix of size n.

Models

Deep-LPBM	The deep latent position block model.
Deep-LPM	The deep latent position model.
Deep-LPTM	The deep latent position topic model.
ETM	The embedded topic model.
LDA	The latent Dirichlet allocation model.
LPM	The latent position model.
LPCM	The latent position cluster model.
SBM	The stochastic block model.
VAE	The variational autoencoder.
VGAE	The variational graph autoencoder.

Abbreviations

CAVI	Coordinate ascent variational inference.
ELBO	Expected lower-bound.
GCN	Graph convolutional network.
MLE	Maximum likelihood estimate.

Mathematical notations

$\mathcal{C}(\theta)$	A distribution $\mathcal{C}$ with parameters θ.
$\mathcal{C}(x;\theta)$	The density of a distribution $\mathcal{C}$ with parameters θ evaluated at x.
$\mathcal{D}ir_d(p)$	The d-dimensional Dirichlet distribution with parameter p.
$\mathcal{M}_d(n,p)$	The multinomial distribution for d categories, with n independent draws and a probability vector $p = (p_1, \ldots, p_d) \in \Delta_d$.
$\mathcal{N}_d(\mu,\Sigma)$	The d-dimensional multivariate normal distribution with mean μ and variance-covariance matrix Σ.
$b_n = O_p(a_n)$	For any $\epsilon > 0$ there exists M and N such that $p(\lvert b_n\rvert \leq M\lvert a_n\rvert) > 1-\epsilon$, for all $n > N$.
$\mathrm{softmax}(x)$	$= \left(\sum_{i=1}^d e^{x_i}\right)^{-1} (e^{x_1}, \ldots, e^{x_d})$, for all $x \in \mathbb{R}^d$.
$\mathrm{KL}(r \mid\mid p)$	$= \int_{\mathcal{X}} r(x)\log(r(x)/p(x))dx$, the Kullback-Leibler divergence between two distributions r and q over a space $\mathcal{X}$.

Chapter 1

Introduction

Numerous interactions imply the exchange of texts, as in co-authors networks, or emails. Since both the storage capacity as well as the number of interactions grow quickly, understanding the dataset at hand is a difficult task. The models developed in this manuscript aim to make these datasets intelligible to human beings. Before diving into the mathematical requirements to analyse such data in the next chapter, we start by introducing two types of data that will be the main focus of this thesis, namely the corpus of texts and the networks, or graphs. These types of data are encountered on a daily basis, through social media for instance, or for specific tasks such as social sciences studies. Nonetheless, both of these data types are very challenging and require the development of a specific methodology to extract interesting patterns. This introduction will serve to present an

example of textual corpus, with the arXiv abstract dataset and an example of a network with the Sampson monastery dataset. Those examples will serve to illustrate common questions related to those data types, such as how is structured the network, or what are the main threads in the texts and examples of answers will be provided. Eventually, we will discuss some limitations of those methods and we will highlight the differences between modelling and *describing* the data.

1.1 Textual data, corpus, and topics

1.1.1 Representation of documents

Retrieving information from a large amount of documents is crucial in different fields, such as social sciences, and literature, or even to provide features for another analysis. Whether it is through emails, messages, books or articles, for instance, sources providing texts are spiking, in number and volume, and with them have grown the need to quickly obtain information about those corpus. When the documents are too long, and/or the number of documents is too large, reading all of the material quickly becomes impossible, as well as summarising the content of the documents. For instance, we might not be interested in sports articles but very interested in articles dealing with the economy. Reading all of the articles in a journal to determine which ones belong to a topic of interest is highly inefficient. Thus, we would like to find a way to determine which articles deal with the topics we are interested in. Before going further, it is necessary to transform the data so that the texts can be represented as a mathematical object. One of the most basic approaches is called the *bag of words*, which corresponds to the creation of a *document-term matrix* to represent the texts. Essentially, it consists in representing each document by a vector of the same size as the vocabulary, with each coordinate corresponding to the number of occurrences of the word in the document, as presented in Table 1.1. Notice that this representation of the documents discards the order of the words, which may be crucial information for some applications. This type of data is referred to as *count data* and is often modelled with a Poisson distribution.

Document-term matrix								
Documents \ Vocabulary	What	is	your	name	My	John	a	beautiful
What is your name	1	1	1	1	0	0	0	0
John, my name is John	0	1	0	1	1	2	0	0
What a beautiful name	1	0	0	1	0	0	1	1

Table 1.1: Example of a document term matrix $W \in \mathcal{M}_{3\times 8}(\mathbb{R})$, using a bag-of-words representation, corresponding to the three documents on the left-hand side.

1.1.2 Desired properties of topics

While the mathematical formulation of a topic will be given in the next chapter, this section aims to give insights about the desired properties of a topic. The first interesting property for a topic would be to distinguish two semantic meanings of the same word. For instance, the word *"apple"* may refer to a computer brand as well as a fruit. Therefore, we would like the topic dealing with fruit as well as the topic dealing with computers to identify the word *"apple"* as important. In other words, a topic should be able to deal with polysemy. Another interesting feature for a topic would be to deal with synonymy. Indeed, considering that the words *"performance"* and *"show"* may be used similarly, we would like a topic using one to describe an exhibition to also be able to identify the other as relevant. This has been theorised in Deerwester et al. (1990), where the authors proposed to find a latent semantic meaning of the documents by looking for models sharing properties including: *"adjustable representational richness"*, that is a model with sufficient power to represent the semantic structure of the documents and *"explicit representation of both terms and documents"* such that the simultaneous representation of both terms and documents permits to retrieve documents closed to a topic.

To summarise, the topics are built to capture semantic information of the documents. Even though two words are not in a document, they may have a close semantic meaning. Hence, we would like to represent words and documents in a same subspace to be able to measure their similarity, as we shall develop in the next chapter. The next section presents an example of the results we may expect when looking for the topics shaping a corpus.

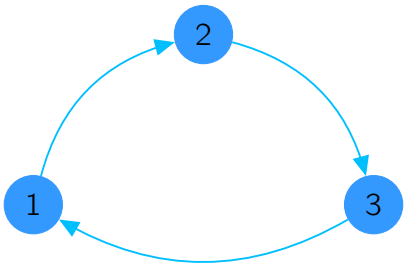
1.1.3 An example with the arXiv abstract dataset

In this section, we illustrate how topics help to get information from a corpus of texts. To do so, we used the arXiv corpora composed of 1.7 million papers (Clement et al., 2019), from which we kept almost 110,000 papers related to statistics. To capture, the different subfields in statistics, we analysed the abstracts of these papers using the latent Dirichlet allocation model (Blei, Ng, Jordan, 2003). The results are presented in Figure 1.2. From a qualitative point-of-view, it is interesting to see some traditional fields, such as the statistical test theory (Topic 5) or Bayesian statistics (Topic 7) present as well as more recent fields such as reinforcement learning (Topic 1) or deep neural networks (Topic 9), very well represented in the papers present on arXiv.

Some other questions arise naturally, such as the evolution of the papers published over the years, or when did a new topic appear. This is precisely the type of information relevant to a taxonomy, and that may help obtain a better understanding of the different threads shaping a field. For more information about time-based topic models, see Vayansky, Kumar (2020).

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
bounds	learning	model	analysis	test
bound	algorithm	time	machine	distribution
optimal	algorithms	causal	paper	sample
problem	optimization	treatment	learning	testing
lower	problem	effects	results	tests
regret	gradient	effect	research	power
algorithm	performance	models	representations	based
convex	training	using	al	detection
linear	methods	study	statistical	value
log	method	prediction	work	hypothesis
Topic 6	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
time	distribution	model	neural	model
process	bayesian	method	networks	data
processes	distributions	estimation	network	information
series	model	regression	learning	based
function	posterior	proposed	models	systems
kernel	inference	estimator	model	models
stochastic	clustering	methods	deep	using
functions	matrix	models	graph	approach
non	prior	data	training	different
gaussian	models	based	classification	domain

Table 1.2: Top-10 most probable words estimated by a latent Dirichlet allocation using 10 topics on ArXiv abstract dataset, restricted to the statistical field.



		Receiver		
		1	2	3
Sender	1	0	1	0
	2	0	0	1
	3	1	0	0

Figure 1.1: Example of a network on the left-hand side with its associated binary adjacency matrix on the right-hand side.

1.2 Networks

To analyse *interactions* between a set of agents, called nodes or vertices, the common approach is to use a network defined as a set of connections, or edges, between some of the nodes. Before diving into the mathematical representation of networks in the next chapter, we use this introduction to provide an illustration of a simple network. We move on and give three real-life examples of interactions from different fields to show the variety of situations represented by networks. Then, we will dive into some details concerning the representation of this type of data, using Sampson's monk dataset. Several difficulties will arise, and we will present some methods developed to solve them as well as their limitations.

1.2.1 A vanilla network

Before giving real-life examples of applications, it is important to precise the meaning of a network and the reasons for its use. Figure 1.1 represents the connections between three nodes. Namely, the node 1 is connected to node 2, 2 to 3 and 3 to 1. On the

1.2.2 Three different examples of interactions

We begin those examples with the challenge of understanding gene co-expression patterns. From a biological point-of-view, as well as for medical purposes, discerning the genes involved in the creation of a protein is crucial. For instance, assuming that the relationships between genes and proteins were known, discovering abnormalities among genes may allow to prevent diseases. One way to represent the co-expression of genes is to use a *co-expression similarity*, a function that measures the similarity of expression between two genes, and to assume that genes expressing similarly are interacting together. Then, it is possible to create a network of interactions between the nodes and to find patterns on that network using the available methodologies in the network literature, see Zhang, Horvath (2005) for more details.

To stress the variety of applications relying on graph-structured data, we now present an application to ecological networks. For instance, we can consider species and assume inter-

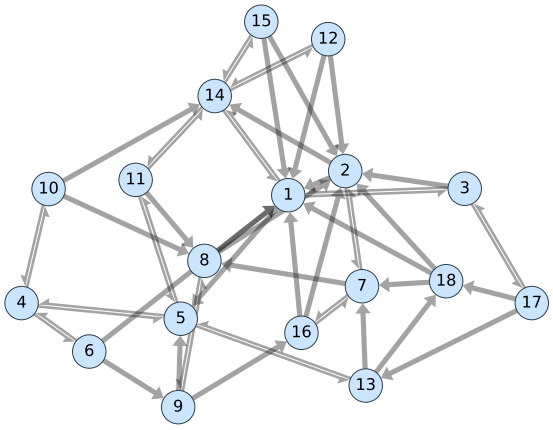
actions representing "who eats who?", as well as interactions between "plants/pollinators". In Thébault, Fontaine (2010), the authors described how the very different structures of these networks impact the stability of the system, and the species coexistence.

Eventually, we end our examples with a very different field in the name of social sciences. Among the many applications in this domain, we mention the *strength and weak ties theory* (Granovetter, 1973). Granovetter proposed to analyse how the topology of a social relationship network may impact the social life of the persons composing the network. In a nutshell, Granovetter assumed that strong ties between people, e.g. close friends, and family members, often bring redundant information in a relationship, while weak ties can easily be unconnected with the rest of someone's relations, making it more likely to bring novel information. This methodology has been extended to the idea of social capital: the set of persons someone is connected to, as well as how those persons are connected together, enables them to access resources, that may lead to better jobs and faster promotions. Here again, the structure of the network is assumed to play a role in social life. Other examples can be found in Borgatti et al. (2009).

Those three questions, as different as they may seem at first sight, can be gathered into a unified framework. Let us represent the individuals, e.g. the genes, the plants/insects, or the persons respectively, by *nodes* and the interactions between the nodes, e.g. similar genes, one is eaten by the other, or two persons are connected respectively, by *edges*. This corresponds to the definition of a network and has been studied extensively to answer some of the questions raised before or to evaluate the likelihood of generative assumptions regarding the network. We present Sampson's monk dataset in the next section. We will use it, first, to provide some illustrations of a network, second, to show how important graphical representation may be to understand the data and finally, to present some difficulties to obtain these figures. We hope to convince the reader that the network framework is both natural and very general.

1.2.3 Sampson monastery dataset

In his PhD dissertation, Sampson reported his observations about the social interactions that took place during his stay in a cloister composed of 18 monks. He classified the monks into four categories: the "loyal opposition", the "young Turks", the "outcasts" and the "waverers". After a political "crisis in the cloister", four monks were expelled, including the leader of the "young Turks". This led five other "young Turks" to leave, followed by three "waverers". Eventually, with two other monks leaving, only four of the 18 original monks were still in the monastery at the end. Several datasets have been created by Sampson, from the information he gathered. We will focus on the first affect relations network, which was built by asking each monk to name the three persons he had the most positive relationships with. Hence, a directed edge is created from monk A to monk B if A named B among his top choices. While the data is much richer, for instance, Sampson gathered that information at different time steps to study the evolution of the relationships, we chose to focus on this dataset for clarity's sake. The network is composed



(a) Sampson's monastery Network.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
2	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0
6	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
8	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
10	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0
12	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
13	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1
14	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
15	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
16	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
18	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

(b) Adjacency matrix with 1 in coordinate (i, j) if i is connected to j and 0 otherwise. For instance, node 10 is connected to node 4 and $A_{10,4} = 1$.

Figure 1.2: Network of the affect relationship in the cloister observed by Sampson.

of 18 vertices and 55 edges.

In Figure 1.2, we display the row adjacency matrix as well as a possible representation of the network obtained by using the Fruchterman-Reingold algorithm (Fruchterman, Reingold, 1991). Other node features gathered by Sampson are available to obtain a more informative representation. For instance, for each monk, Sampson provided information about the group to which he belonged, as well as an indicator concerning his attendance to the minor seminary of "Cloisterville" before coming to the monastery.

One possibility to get a better understanding of Sampson's network is to find groups of close monks, which comes down to splitting the nodes into groups highly connected together but poorly to the nodes in the other groups. This type of group is called a *community* and many community detection methods have been developed, such as the modularity (Newman, 2006), or more advanced statistical network analysis model as the latent position model (LPM Hoff, Raftery, Handcock, 2002), as we shall see later on. As stated in Girvan, Newman (2002), one of the core motivations for the development of these methods is that many networks share the property of transitivity, which is the property that two vertices that are both neighbours of the same third vertex have a heightened probability of also being neighbours of one another. However, in many cases, a more complex structure may be responsible for the observed network. We will give a deeper treatment on the necessity to use model-based methodologies at the end of this introduction. Also, we will introduce some of the core methods enabling the discovery of latent structures that may

arise in complex real-life networks and may not be distinguished by community detection methods.

1.2.4 A first limitation of community detection methods: an example with some latent structures

Since looking for communities may be too restrictive and may prevent from capturing more complex underlying patterns, this section aims to provide some examples of a few other structures.

This consideration is not just theoretical but is observed in real-life datasets. For instance, let us mention the case of social networks. In many cases, they hold few nodes with a high degree, that is nodes with a large number of connections and many nodes with low degrees. This property is referred to as the "the small world property" in social sciences (Watts, Strogatz, 1998). To give an example, many Twitter accounts follow media accounts, politician accounts and so on, which may lead to hundreds of thousands of connections while a random node would only have hundreds of connections or much less on average. This type of pattern is not a community. Other patterns, described below, may also appear.

In Figure 1.3, we start by presenting the community structure on a simulated network, as well as three other latent structures that cannot be uncovered by traditional community detection methods. The first one is called a bipartite graph and is used for networks with two types of nodes, such as clients and servers for instance, and illustrates how clients are connected to servers. Thus, the bipartite graphs are composed of two groups with nodes only connected to nodes in the other group. The second one is composed of a hub corresponding to the blue nodes. This is a group of nodes highly connected to several groups of the graph and/or a group to which several groups are connected to, as well as two other communities. The third one, named a star, is composed of two groups, namely a star, where one node (the blue node), is connected to the other nodes of a group (the green nodes), but where the nodes in the other groups are very poorly connected together.

In each of these three topologies, the property of *transitivity* does not hold, making the community detection methods fail. For instance, in the hub structure, a node from cluster green and a node from cluster red may share a common neighbour in cluster blue but it does not imply that they also have a higher probability to be connected. This is also true for the bipartite network as well as the star network. This first limitation is part of a broader limitation of traditional heuristic-based methods. In the next section, we give a deeper treatment of this aspect, as well as a motivation for probabilistic modelling, which will be formally presented in the next chapter.

1.2.5 An introduction to networks with textual edges

Before diving into the limitations of the descriptive methods, we give a brief example of networks with textual edges and motivate the development of tailored methodologies. Go-

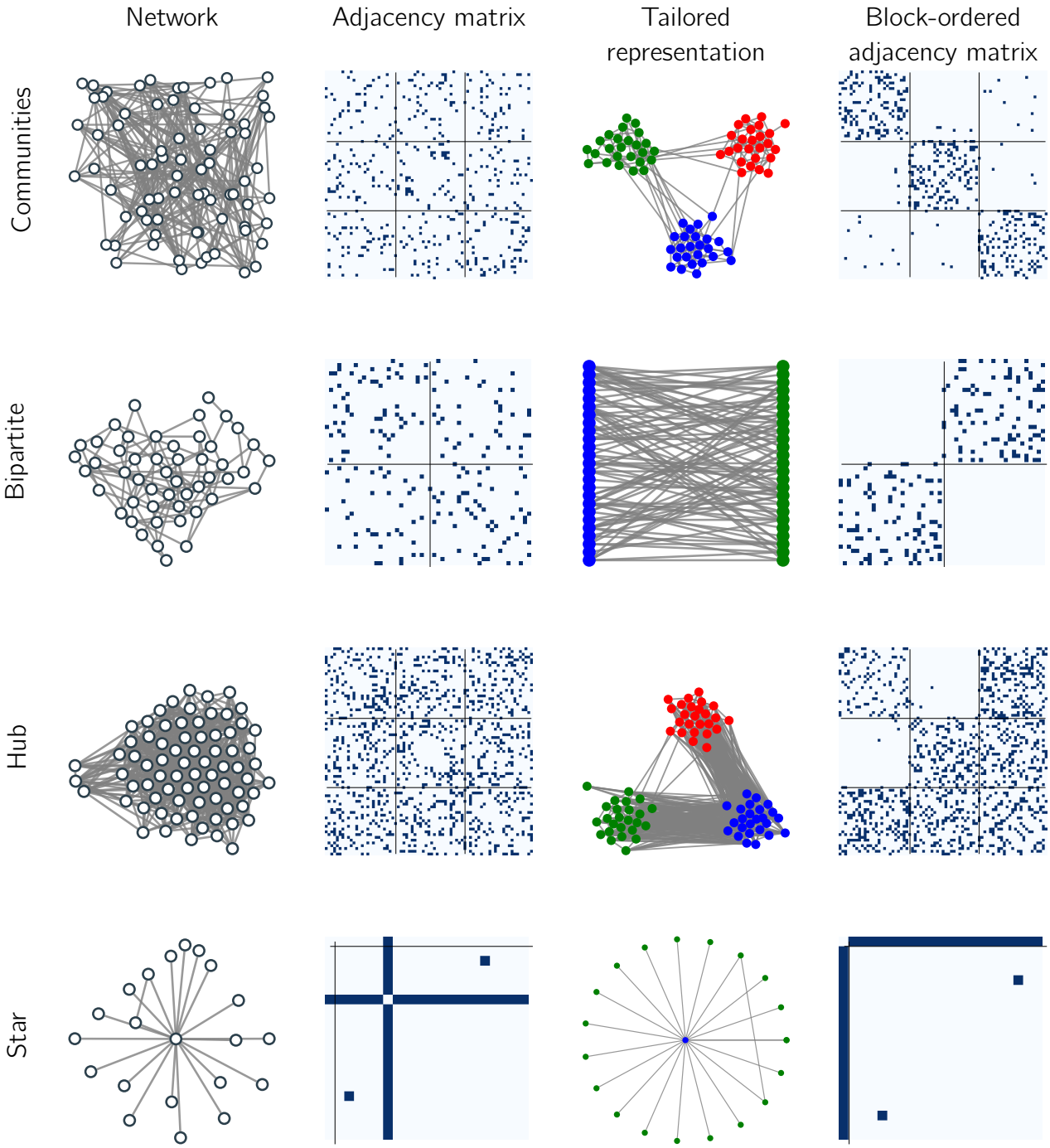


Table 1.3: Example of four networks with a latent structure composed of (top to bottom) three communities, a bipartite graph, two communities with a hub and a star. The networks on the left-hand side are obtained by using only the connection without any suitable method to represent them. On the contrary, a tailored representation is displayed in the third column, taking the true structure to highlight what we could expect. The group of a node is indicated by its colour. The second and fourth columns represent the adjacency matrix unordered and ordered by cluster, or block, respectively. The dark square corresponding to coordinates (i, j) indicates that node i is connected to node j.

ing back to Sampson's monastery network, all the information provided about the monks, as well as the different time periods at which the data was gathered and other information, could not be incorporated into the analysis since the methods are not able to include them. This may induce a loss of crucial information and thus, degrade the results. Therefore, in this manuscript, we focus on networks with textual edges and aim at incorporating the content of the texts exchanged between the nodes, to improve the quality of the results. A network with textual edges is a network in which each edge models an interaction involving a document, as depicted in Figure 1.3.

In Figure 1.3, the same network is displayed three times. The first network represents the data without any analysis, the second network presents the clustering results obtained with community detection methods. The last network presents the results obtained with a specifically designed methodology, capable of including the texts as well as the network connectivity to obtain relevant node clusters. This is one of the core motivations for the development of such methodologies, that we will pursue in this manuscript.

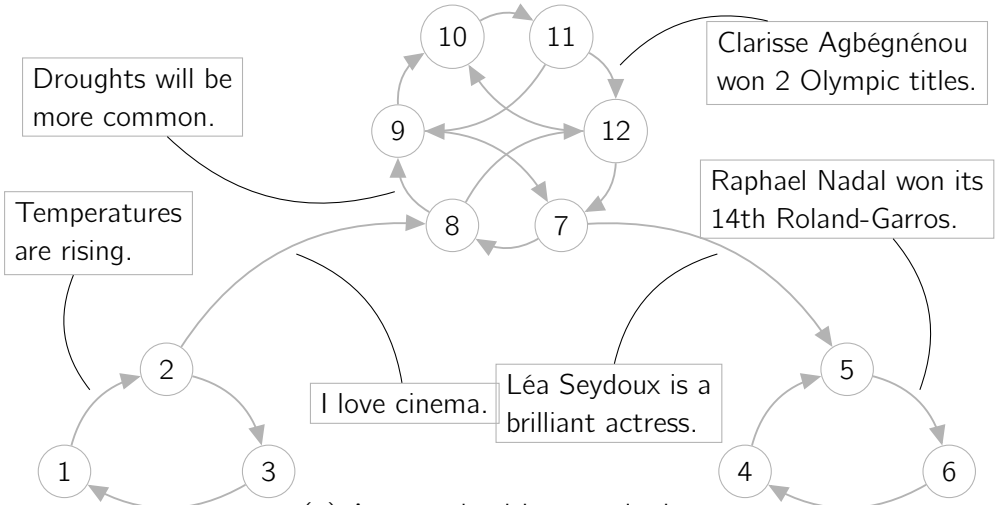
1.3 Descriptive methods and their limitations

In this section, we first introduce *descriptive methods*, as described in Peixoto (2021). We will give examples of some of the most-used methods, in the name of the min-cut/max-cuts, as well as the modularity maximisation. Eventually, we will present some of the limitations of descriptive methods. In particular, we will show that they may perform poorly in some settings and that they are not able to *understand* patterns in the data but only to describe them.

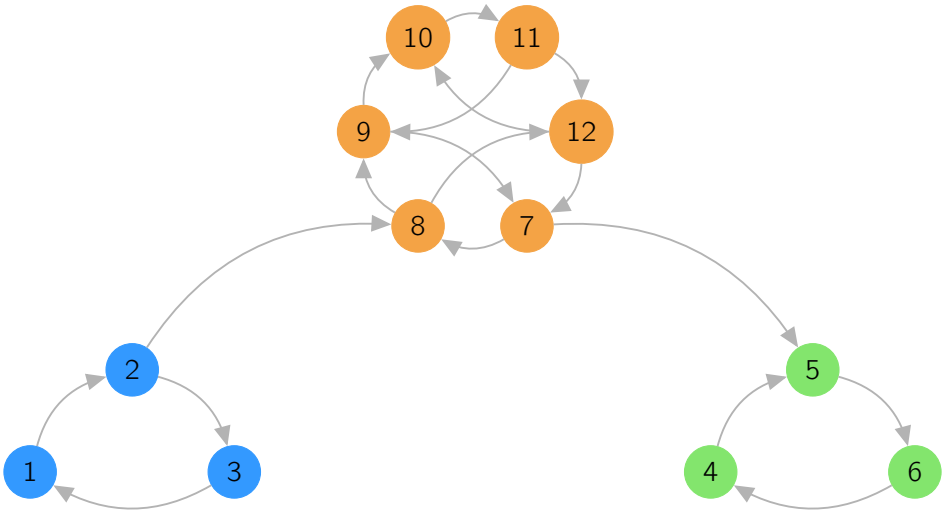
1.3.1 Descriptive methods

In this manuscript, we call *descriptive methods* a heuristic-based method, relying on an objective function that is designed to answer a specific purpose. In particular, the algorithms do not try to understand how the data was generated, or the rules responsible for the observed patterns. They are only intended to *detect* motifs related to the objective function defining the corresponding algorithm, not to explain the data. One of the weaknesses of such a method comes from the dependence of the notion of community on the algorithm itself, and its objective function.

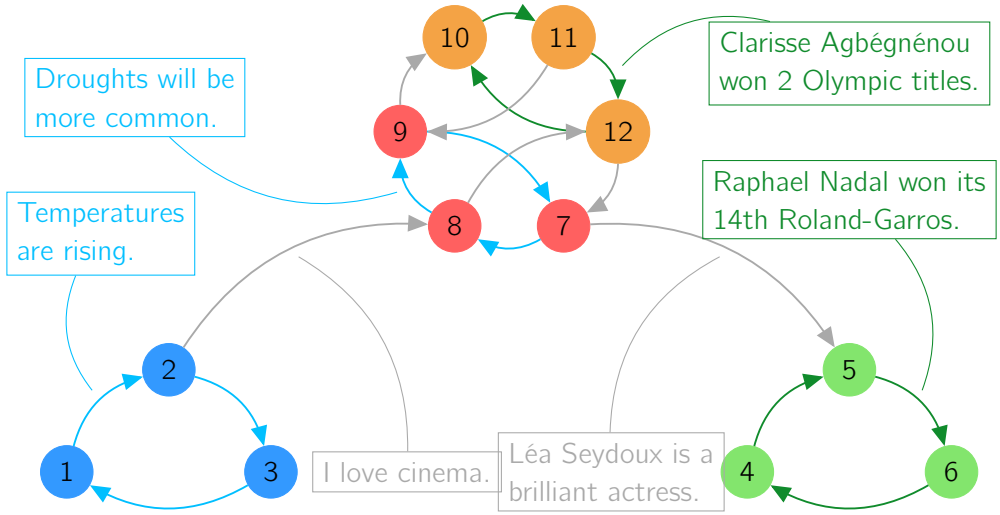
We want to emphasise that understanding the data is not always necessary in practice and having a good description can be well-suited for certain applications. For instance, Peixoto mentioned the design of very large-scale integrated circuit (VLSI), that aims at combining millions of transistors into a microprocessor chip in an efficient way. Thus, finding a good partition to obtain smaller modules with few connections between them, providing an optimal positioning of the transistors, is sufficient to build the chip. Let us mention some descriptive algorithms, including the modularity algorithm which is arguably the most used technique to detect communities in a network.



(a) A network with textual edges.



(b) The community detection results on the network alone. Since these methods cannot deal with textual data, the documents are discarded from the analysis.



(c) The results obtained with a methodology analysing simultaneously the texts and the network connectivity.

Figure 1.3: Example of a simple network with textual edges. The node (edge respectively) colours denote the node cluster memberships (the edge majority topics).

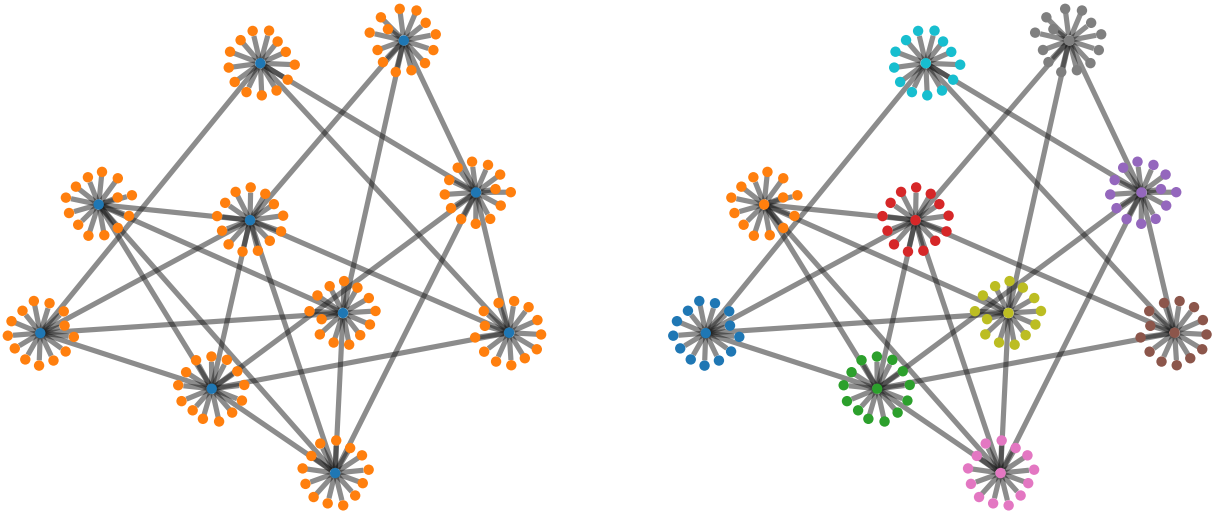


Figure 1.4: The partitions obtained with a stochastic block model on the left-hand side, and a Louvain algorithm for community detection on the right-hand side. The node colours indicate the cluster memberships.

Min cut/ max cut methods An intuitive idea to discover communities is to look for groups of nodes minimising the number of edges between the groups. Many implementations of this algorithm have been developed, see Stoer, Wagner (1997) for instance.

Modularity based techniques In cases where the cluster sizes are not fixed beforehand, an easy solution would be to place all nodes in a single cluster and therefore to have no edge between the two clusters. To avoid this trivial, yet uninformative solution, a constraint can be placed on the size of the clusters. This can be seen as quite arbitrary. To solve those shortcomings, the modularity-based algorithm has been proposed by Newman (2006). In this paper, the author stressed that taking this problem as just a minimisation of the number of edges between groups is not the right way to look at this. Instead, the aim should be to look for evidence of a structure, by comparing the observed number of edges between clusters with the number of expected edges obtained based on a random chance. For instance, the Louvain algorithm (Blondel et al., 2008) is an efficient modularity-based algorithm to perform community detection on a graph.

1.3.2 Limitations of such methods

These methods are created to *describe* the data and not to *understand how the network was generated*. In Figure 1.4, we provide an example suggested in Peixoto (2021), where detecting the communities, in the sense of the modularity, is very different from understanding the structure behind the generation of the network.

The network presented in Figure 1.4 was simulated by first drawing 10 nodes among 140, called central nodes, and drawing an edge between them with 0.45 probability. Then, for each star, 13 nodes were drawn at random and connected to the corresponding central

node. At first sight, the network on the right-hand side of Figure 1.4, obtained using the Louvain algorithm, may seem to better recover the communities in the network than the stochastic block model, located on the left-hand side. First, note that those are not communities but stars, which is already misleading and highlights the problem of the definition of a community in the Louvain algorithm. Let us now describe the generative process responsible for this network. We picked 10 nodes among 140 at random, the centre of each star, and for each star, we picked 13 nodes at random that were connected to the central node of the cluster. Clearly, in the generative process, the latent structure of the network is twofold. First, the central nodes are picked, and then the nodes around each centre. The stochastic block model, which will be presented in the next chapter, should be looked at as an example of a generative model for now, precisely to capture these patterns. This example displays the main difference between describing and explaining the data.

Another limitation of descriptive methods is the absence of quantification of the uncertainty. On the contrary, statistical network analysis, may provide estimates of the variance of quantities and ensure it is reasonable enough to use the results for decision-making, which can be decisive in industrial processes for instance.

Eventually, we also mention some drawbacks in terms of the performance of descriptive methods. In settings with very little data, or with a lot of noise, these methods may tend to overfit. In Figure 1.5, we fitted a stochastic block model on networks composed of 100 nodes, with three communities, allocated to a group with equiprobability. This corresponds to the network at the top of Table 1.3. Two nodes of different groups are connected with a 0.01 probability while two nodes of the same community have a probability ρ to be connected, varying along the abscissa axis. The higher ρ is, the more separated are the communities, and the easier it becomes to retrieve them. Conversely, for ρ lower than 0.05, the graph is very close to an Erdős–Rényi random graph, that is a graph in which any two nodes have a probability p to be connected, hence does not have any community structure. Figure 1.5 highlights the efficiency of SBM in most settings with a structure and to decay very quickly when the structure becomes evasive if not inexistent. This avoids being overconfident in settings with no real structure. On the contrary, Louvain's algorithm performances are not as good as the SBM on structured networks but still report community structure for very low probabilities of connection. Another performance limitation of the modularity-based method is the well-known resolution limit depending on the size of the network and the interconnectedness of the communities, preventing small-sized communities from being discovered (Fortunato, Barthelemy, 2007). The next section presents the framework developed to overcome the limitations described in the previous sections.

1.3.3 Motivation behind Generative modelling

To end this chapter, we introduce the notion of generative modelling, a methodology that answers the limitations raised above. Before explaining this notion, we recall a central idea behind modelling a phenomenon, often named Occam's razor. This property has

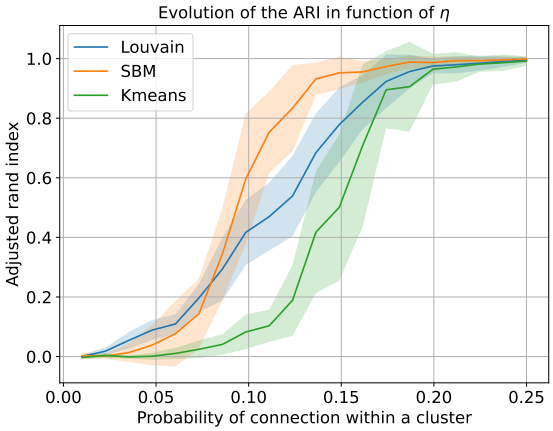


Figure 1.5: Evolution of the adjusted rand index (ARI) in function of the noise for three different methods. The methods are averaged over 20 networks, composed of three communities, with an intra-connection probability ρ and an inter-connection probability $\eta = 0.01$. The noise is modelled by the probability of connection within a community ρ such that the lower it is, the less structured the graph and the more difficult it is to find the communities. An ARI of 1 means a perfect recovery of the partition of the nodes while an ARI of 0 means that is as good as a random guess. The K-means corresponds to naively applying the K-means algorithm on the rows of the adjacency matrix.

been formulated in many ways, for instance by Aristotle: *"We may assume the superiority [other things being equal] of the demonstration which derives from fewer postulates or hypotheses."*. In other words, among two competing theories, or models, leading to the same results, one should favour the simplest one (the one with the fewest hypothesis). This statement can be retrieved in Bayesian statistics where each assumption comes with a prior probability impacting the posterior probability on the data, favouring parsimonious model, see Jefferys, Berger (1991) for more details.

Relying not only on the evidence of the data assuming that the assumptions hold but also evaluating the assumptions is a core idea in statistics. We will take our time to describe the assumptions regarding the generation of the data in each one of our models. We emphasise that this is indeed the main difference with descriptive methods and one of the biggest strengths of probabilistic models. This is highlighted by the following considerations.

In statistical network analysis, we are often interested in the probability of our data $\mathbf{A}$ given the partition, or node cluster memberships, $\mathbf{C}$, denoted $P(\mathbf{A} \mid \mathbf{C})$, that we would like to maximise. Conversely, Bayesian modelling aims at computing the probability of $\mathbf{C}$, when observing $\mathbf{A}$ using Bayes' rule:

$$P(\mathbf{C} \mid \mathbf{A}) = \frac{P(\mathbf{A} \mid \mathbf{C})P(\mathbf{C})}{P(\mathbf{A})}.$$

It has been linked to the idea of compression, see Grünwald, Roos (2019) for more details, such that the description length $\Sigma(\mathbf{A}, \mathbf{C})$, corresponding to the length, in bits, necessary

to describe the network $\mathbf{A}$ with the partition $\mathbf{C}$ through the following relationship:

$$P(\mathbf{A} \mid \mathbf{C})P(\mathbf{C}) = 2^{-\Sigma(\mathbf{A},\mathbf{C})}. \tag{1.1}$$

The Equation (1.1) implicates the equivalence between inferring the partition $\mathbf{C}$ and compressing the network using this partition. Note that the description length can be decomposed into two terms

$$\Sigma(\mathbf{A}, \mathbf{C}) = L(\mathbf{A} \mid \mathbf{C}) + L(\mathbf{C}),$$

where $L(\mathbf{A} \mid \mathbf{C})$ is the number of bits required to encode the network given the parameters and the partition, and $L(\mathbf{C})$ the amount of bits required to encode the parameters of the model and the partition. Hence, a partition decreasing the number of bits necessary to encode the network but increasing the number of bits necessary to encode the parameters will be relevant if the sum of the two diminishes. This property is essential to understand what drives the compression. In addition, this is a way of understanding how compression, as well as inference, avoids overfitting thanks to the famous Shannon's theorem. Indeed, since any distribution has a minimal description length equal to its entropy, it bounds the compression possibility.

Going back to description methods, the same framework gives important insights on the difference between the two approaches. Let us consider the objective function of the description method $W(\mathbf{A}, \mathbf{C})$ that we would like to maximise, for instance, the modularity. Thus, using a Gibbs measure, we can write this description method as an inference problem such that

$$P(\mathbf{C} \mid \mathbf{A}) = \frac{e^{\beta W(\mathbf{A},\mathbf{C})}}{\kappa(\mathbf{A})},$$

where $\kappa(\mathbf{A}) = \sum_{\mathbf{C}} e^{\beta W(\mathbf{A},\mathbf{C})}$. Moreover, denoting $\kappa(\mathbf{C}) = \sum_{\mathbf{A}} e^{\beta W(\mathbf{A},\mathbf{C})}$ as well as $\kappa = \sum_{\mathbf{A},\mathbf{C}} e^{\beta W(\mathbf{A},\mathbf{C})}$, and making the assumption that

$$P(\mathbf{A}) = \frac{\kappa(\mathbf{A})}{\kappa} \text{ and } P(\mathbf{C}) = \frac{\kappa(\mathbf{C})}{\kappa},$$

we obtain the following likelihood of the model:

$$P(\mathbf{A} \mid \mathbf{C}) = \frac{e^{\beta W(\mathbf{A},\mathbf{C})}}{\kappa(\mathbf{C})}.$$

Thus, the length description in nats units, that is using the natural logarithm in the definition of the information of a random variable, is given by:

$$\begin{aligned}
\Sigma(\mathbf{A}, \mathbf{C}) &= -\ln\left(P(\mathbf{A} \mid \mathbf{C})\right) - \ln\left(P(\mathbf{C})\right) \\
&= -\ln\left(P(\mathbf{A} \mid \mathbf{C})\right) + \ln\left(\kappa\right) \\
&= -\beta W(\mathbf{A}, \mathbf{C}) + \ln\left(\sum_{\mathbf{A}',\mathbf{C}'} e^{\beta W(\mathbf{A}',\mathbf{C}')}\right).
\end{aligned}$$

Thus, any description methods rely on a model that is not explicit and can be associated with the inference of the corresponding model. Saying otherwise is a misuse of language, that may be encountered in this manuscript. Indeed, since the underlying assumptions are not known, they may be poorly chosen, and not have any impact on the compression of the data, as stated in Peixoto (2021): *"Although we mentioned [...] that inference and compression are equivalent, the compression achieved when considering a particular generative model is constrained by the assumptions encoded in its likelihood and prior. If these are poorly chosen, no actual compression might be achieved, for example when comparing to the one obtained with a fully random model. This is precisely what happens with descriptive community detection methods: they overfit because their implicit modelling assumptions do not accommodate the possibility that a network may be fully random, or contain a balanced mixture of structure and randomness."*

Chapter 2

Foundations

In the following chapter, we present the mathematical formalism of the generative modelling used in this manuscript. This will lay the ground for a presentation of the inference tools underlying the developed models. We will also present some of the core models

in topic modelling as well as statistical network analysis, taking great care of exposing their generative assumptions. Eventually, a brief introduction to graph neural networks is provided.

2.1 Introduction to probabilistic modelling

2.1.1 Graphical models

For a statistical model to be useful, it is often necessary to obtain interpretable results. For instance, in industrial processes, it is common for experts to require a deep understanding of the results as well as a causality between the variables and the output. Graphical models are specifically designed to make assumptions regarding the relationships responsible for the observed data. Let $X_1, \dots, X_p$ be p random variables, such that the joint distribution is denoted $p(X_1, \dots, X_p)$, a *graphical model* is a set of assumptions concerning the variables and their dependencies allowing to factorise the joint distribution. This can be represented by a graph, as illustrated in Example 2.1, giving the name to these models. In this manuscript, we will only consider *directed* graphical models. In directed graphical models, the set of assumptions represents the hypothesis made on the causalities between variables, or in a probabilistic language, induces conditional independences between the variables. This allows us to factorise the joint distribution using Bayes rules.

Definition 1. *A directed graphical model is a set of assumptions on the relationships between the variables $X_1, \dots, X_p$, inducing a set of ancestors $\pi(X_l)$ for any $l = 1, \dots, p$, such that the joint distribution can be factorised as*

$$p(X_1, \dots, X_p) = \prod_{l=1}^{p} p(X_l \mid \pi(X_l)).$$

Remark 1. *Any joint distribution can be factorised using Bayes' rule iteratively as*

$$\begin{aligned} p(X_1, \dots, X_p) &= p(X_1 \mid X_2, \dots, X_p) p(X_2, \dots, X_p) \\ &= p(X_1 \mid X_2, \dots, X_p) p(X_2 \mid X_3, \dots, X_p) \dots p(X_p) \end{aligned} \tag{2.1}$$

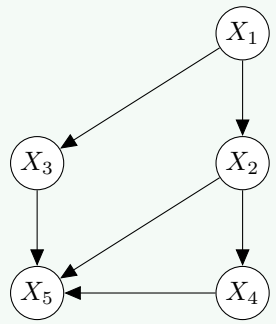
Therefore, Definition 1 may seem futile at first sight. But Equation (2.1) *does not require any assumption and is not informative. Indeed, it is possible that all variables depend on each other. On the contrary, a graphical model implies that assumptions are made to characterise the dependencies between the variables and provides an informative factorisation of the joint distribution as in Example 2.1.*

In some cases, the variables involved in the generative model are not observed, they are called *latent variables*. A large literature has focused on models with latent variables. Indeed, those models benefit from interesting properties, such as the expressiveness of the model, as well as the uncertainty measures they can produce. In the last 20 years, new inference strategies have been developed allowing the analysis of very large datasets. In

the next sections, we shall detail both what we mean by latent variables as well as some core inference strategies used in this manuscript.

Example 2.1: A directed graphical model

Figure 2.1: A directed graphical model



This graphical model corresponds to the following factorisation of the joint distribution

$$p(X_1, \dots, X_5) = p(X_5 \mid X_4, X_3, X_2)p(X_4 \mid X_2) \\ p(X_3 \mid X_1)p(X_2 \mid X_1)p(X_1).$$

Additional assumptions can be made on each conditional distribution to specify the family (for instance Gaussian, Poisson, binomial ...). If all conditional distributions are specified and each distribution can be sampled, it is possible to generate data from the model distribution.

2.1.2 Latent variable models and inference

Let θ be a collection of parameters, $\boldsymbol{\eta}$ a random vector that is not observed, called a *latent variable*, or *hidden variable*, and $\mathbf{Y} = (Y_1, \dots, Y_N)$, N observed variables. A latent variable model makes assumptions about the joint-distribution of $\boldsymbol{\eta}$ and $\mathbf{Y}$, such that the *marginal likelihood* is given by integrating over the latent variable,

$$p(\mathbf{Y} \mid \theta) = \int_{\eta} p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) d\boldsymbol{\eta}, \tag{2.2}$$

where $p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)$ is the *complete data likelihood*.

In many cases, the integral in Equation (2.2) is not tractable and an adapted inference strategy needs to be deployed to tackle this issue. We will briefly present some of the common inference strategies to solve this problem and will detail the E-M algorithm as well as the variational inference methods which are the main ingredients of the works presented in this manuscript.

Example 2.2: Probabilistic PCA and factor analysis

The factor analysis model as well as the probabilistic principal component analysis (PPCA), assume that an observed random vector $Y \in \mathbb{R}^P$ is a linear transformation of a latent (not observed) variable $X \sim \mathcal{N}_K(0, \mathbf{I}_K)$, with $K < P$, combined with an additive noise $\varepsilon \sim \mathcal{N}_P(0, \Psi)$,

$$Y = WX + \mu + \varepsilon,$$

where $W \in \mathcal{M}_{P \times K}(\mathbb{R})$ and $\mu \in \mathbb{R}^P$. On the one hand, factor analysis does not make any assumption regarding Ψ. On the other hand, PPCA assumes that $\Psi =$

$\sigma^2\mathbf{I}_P$ which induces the matrix W to model the covariances between variables. The graphical representations of those two basic models are given below, with the factor analysis model on the left-hand side and the PPCA on the right-hand side. An empty circle denotes a latent variable, and a shaded circle refers to an observed variable. The parameters are not in a circle and the plate indicates the factorisation of the distribution due to conditional independencies.

Figure 2.2: Factor analysis

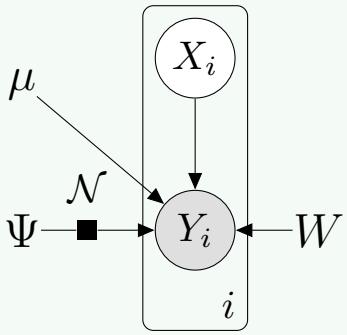
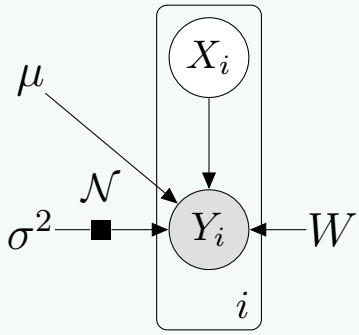


Figure 2.3: PPCA



To estimate the parameters θ, the common approach is to obtain **maximum-likelihood estimates (MLE)**. However, the integral in Equation (2.2) is often not tractable. Among the most used methods to tackle this issue, we mention the approximations of the integral, e.g. a Laplace approximation, the Monte Carlo and Markov Chains (MCMC) methods, for instance using a Gibbs sampler or a Metropolis-Hastings algorithm to approximate the posterior distribution of the parameters θ (Chapter 7 and 8, Robert, Casella, Casella, 1999), the gradient-based methods (Kingma, Welling, 2014; Rezende, Mohamed, Wierstra, 2014) as well as the E-M algorithm (Dempster, Laird, Rubin, 1977) or the variational inference (Chapter 10, Bishop, 2006). The last two methods will be detailed in the next sections.

2.1.3 The E-M algorithm

The E-M algorithm surged after the 70's, as one of the most natural ways to deal with missing values. Indeed, to estimate the parameters of a model with missing values, it is quite natural to first evaluate the most likely value of the unobserved variable using the most likely parameters as well as the data (E-step) and then, to find the parameters maximising the complete likelihood, with the missing value estimated in the E-step (M-step). In this section, we present this algorithm in a unified framework, as intended in the classical paper of Dempster, Laird, Rubin (1977).

Example 2.3: Gaussian mixture model

One of the most used latent variable models is the Gaussian mixture model (GMM). It relies on the assumption that the data is composed of several groups, each one sampled from a specific Gaussian distribution. To keep this example concise, we assume that the variables are one-dimensional. Formally, the observed variables $\mathbf{Y} = (Y_1, \dots, Y_n)$ are independent, and the cluster membership variables $\mathbf{C} = (C_1, \dots, C_n)$ are not observed, and $C_{iq} = 1$ if i is in cluster q and 0 otherwise. We will denote μ_q (σ_q respectively) the mean (standard-variation) of group q, $\alpha = (\alpha_1, \dots, \alpha_Q) \in \Delta_Q$ the probability of cluster memberships and $\theta = ((\mu_q, \sigma_q)_q, \alpha)$ the collection of all the model parameters. Therefore, the density of observation i is obtained by marginalising out the cluster membership variable

$$p(Y_i \mid \theta) = \sum_{q=1}^{Q} \alpha_q \mathcal{N}(Y_i \mid \mu_q, \sigma_q^2).$$

Hence, the *complete log-likelihood* is given by

$$\begin{aligned} \log p(\mathbf{Y}, \mathbf{C} \mid \theta) &= \sum_{i=1}^{n} \log p(Y_i, C_i \mid \theta) \\ &= \sum_{i=1}^{n} \sum_{q=1}^{Q} C_{iq} \log\big(\alpha_q \mathcal{N}(Y_i; \mu_q, \sigma_q^2)\big), \end{aligned} \tag{2.3}$$

while the *marginal log-likelihood* is obtained as

$$\log p(\mathbf{Y} \mid \theta) = \sum_{i=1}^{n} \log \left(\sum_{q=1}^{Q} \alpha_q \mathcal{N}(Y_i; \mu_q, \sigma_q^2) \right). \tag{2.4}$$

Those two expressions highlight the need for a tailored inference. Indeed, while the likelihood of the complete-data (2.3) could be optimised with respect to the parameters, the first-order conditions of the observed likelihood (2.4) does not provide closed-form updates.

The general algorithm Denoting the marginal log-likelihood $\ell(\mathbf{Y}, \theta) = \log p(\mathbf{Y} \mid \theta)$, we are interested in finding the maximum likelihood estimate (MLE)

$$\theta^\star \in \arg\max_{\theta \in \Theta} \quad \ell(\mathbf{Y}, \theta).$$

The E-M algorithm aims at finding maximum likelihood estimates in latent variable models. Since the marginal log-likelihood cannot be computed, the E-M algorithm relies on a distribution r to approach the posterior distribution of the latent variables $p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)$ and then maximises the (approximated) expected complete log-likelihood. For GMM described in Example 2.3, $\boldsymbol{\eta}$ corresponds to the cluster memberships $\mathbf{C}$. First, let us establish the

basics that will also be useful for the variational inference section. Let $\mathcal{R}$ be a family distribution over $\boldsymbol{\eta}$ and $r \in \mathcal{R}$ a distribution, the following algebra allows to split the marginal log-likelihood into two terms to perform the inference:

$$\begin{aligned}
\ell(\mathbf{Y}, \theta) &= \mathbb{E}_{r(\boldsymbol{\eta})}\left[\log p(\mathbf{Y} \mid \theta)\right] \\
&= \mathbb{E}_{r(\boldsymbol{\eta})}\left[\log \frac{p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)}{p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)}\right] && \text{applying Bayes' rule} \\
&= \mathbb{E}_{r(\boldsymbol{\eta})}\left[\log \left(\frac{p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)}{r(\boldsymbol{\eta})} \frac{r(\boldsymbol{\eta})}{p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)}\right)\right] \\
&= \mathscr{L}(\mathbf{Y}, \theta; r) + \mathrm{KL}\big[r(\boldsymbol{\eta}) \,||\, p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)\big], && (2.5)
\end{aligned}$$

where $\mathrm{KL}\big[r \,||\, p\big] = \int_x r(x) \log\left(r(x)/p(x)\right) dx$. Moreover,

$$\begin{aligned}
\mathscr{L}(\mathbf{Y}, \theta; r) &= \mathbb{E}_{r(\boldsymbol{\eta})}\left[\log \frac{p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)}{r(\boldsymbol{\eta})}\right] \\
&= \mathbb{E}_{r(\eta)}\left[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\right] + \mathbb{H}\left[r(\boldsymbol{\eta})\right],
\end{aligned}$$

where $\mathbb{H}\left[r(\boldsymbol{\eta})\right] = -\int_{\boldsymbol{\eta}} r(\boldsymbol{\eta}) \log r(\boldsymbol{\eta}) d\boldsymbol{\eta}$ corresponds to the entropy of the distribution r.

Remark 2. *For any distribution r over $\boldsymbol{\eta}$ and any parameter $\theta \in \Theta$, the following equality holds true*

$$\mathscr{L}(\mathbf{Y}, \theta; r) = \ell(\mathbf{Y}, \theta) - \mathrm{KL}\big[r(\boldsymbol{\eta}) \,||\, p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)\big].$$

Since $\mathrm{KL} \geq 0$, $\mathscr{L}(\mathbf{Y}, \theta; r)$ is a lower bound of the marginal log-likelihood and is named the expected lower bound (ELBO). The difference between the ELBO and the marginal log-likelihood corresponds to the Kullback-Leibler divergence between the variational distribution and the latent posterior distribution. Indeed, in the case where $r(\boldsymbol{\eta}) = p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)$, we obtain exactly $\mathscr{L}(\mathbf{Y}, \theta; r) = \ell(\mathbf{Y}, \theta)$.

The E-M algorithm comes down to iterate between estimating the best variational distribution r (E-step) and maximising the expected complete likelihood with respect to the parameters θ. This is summarised in Algorithm 1.

Algorithm 1: General E-M algorithm.

Input: $t = 0$; Initialise $\theta^{(0)}$ and $r^{(0)}(\eta)$;
while $\mathscr{L}(\mathbf{Y}, \theta^{(t)}; r^{(t)})$ *has not converged* **do**

(E-step) $$r^{(t+1)} \in \arg\max_{r} \mathscr{L}(\mathbf{Y}, \theta^{(t)}; r) \qquad (2.6)$$

(M-step) $$\theta^{(t+1)} \in \arg\max_{\theta} \mathscr{L}(\mathbf{Y}, \theta; r^{(t+1)}) \qquad (2.7)$$

$t = t + 1$;
end

Remark 3. *It is possible to present the E-M algorithm without introducing the surrogate distribution $r(\cdot)$ and simply considering the posterior distribution $p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)$, in the case*

where this quantity can be easily computed, see for instance the GMM case in Example 2.4. Denoting $\theta^{(t)}$ the parameter at the t-th iteration, and denoting

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{p(\boldsymbol{\eta}|\mathbf{Y},\theta^{(t)})}\left[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\right],$$

the two steps of the E-M algorithm can be rewritten as

Algorithm 2: E-M algorithm with explicit posterior $p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)$.

Input: $t = 0$; *Initialise* $\theta^{(0)}$;
while $Q(\theta \mid \theta^{(t)})$ *has not converged* **do**
  (E-step) *Compute* $p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta^{(t)})$ *and* $Q(\theta \mid \theta^{(t)})$;
  (M-step) *Compute* $\theta^{(t+1)} \in \underset{\theta \in \Theta}{\arg\max}\, Q(\theta \mid \theta^{(t)})$.
  $t = t + 1$;
end

Those notations allow us to understand the two difficulties of the problem. First, since the marginal likelihood cannot be maximised straightforwardly, an expectation of the complete likelihood is used instead (E-step). Second, the term obtained in the E-step can be maximised with respect to the parameter (M-step) and hopefully will tend toward a MLE.

This algorithm increases the value of both the ELBO and the marginal log-likelihood at each iteration since,

$$\begin{aligned}
\ell(\mathbf{Y}, \theta^{(t)}) &= \mathscr{L}(\mathbf{Y}, \theta^{(t)}; r^{(t+1)}) \\
&\leq \mathscr{L}(\mathbf{Y}, \theta^{(t+1)}; r^{(t+1)}) && \text{(M-step)} \\
&\leq \mathscr{L}(\mathbf{Y}, \theta^{(t+1)}; r^{(t+2)}) = \ell(\mathbf{Y}, \theta^{(t+1)}). && \text{(E-step)}
\end{aligned}$$

Hence, the sequence $(\ell(\mathbf{Y}, \theta^{(t)}))_t$ is non-decreasing in the sense that the ELBO will never get worse. However, this does not provide any result on the convergence toward a local or global maximum if it exists, and, more importantly, if the parameters $(\theta^{(t)})_t$ converge toward a MLE. While the first convergence properties of the E-M algorithm are discussed in Dempster, Laird, Rubin (1977), Wu (1983) is the first to provide a rigorous proof of the convergence of the parameters. In particular, the author showed that any limit of $(\theta^t)_t$ is a stationary point of the likelihood and, in unimodal cases, with some assumptions regarding the differentiability of the likelihood, $(\theta^t)_t$ converges toward the unique MLE θ^*. For instance, any distribution in the exponential family meets those sufficient conditions and gives closed-form updates that ensure convergence toward a MLE. For more details about the E-M algorithm for distributions in the exponential family, see Wainwright, Jordan (2008), or more generally in Bishop (2006).

Unfortunately, in some cases, the E-step cannot be solved because of the the posterior distribution $p(\eta \mid Y, \theta)$ that cannot be computed directly, as in the stochastic block model

for instance. The next section presents an assumption on the family distribution $\mathcal{R}$ giving rise to an approximation of the E-step making the computations tractable.

Example 2.4: Gaussian mixture model with an E-M based inference

To obtain an estimate of the MLE, we would like to use the first-order conditions with respect to the parameters on Equation (2.4). Unfortunately, this does not provide closed-form updates, see Bishop (2006) Chapter 9 Section 2. Therefore, we apply the E-M algorithm and use the explicit posterior $p(\mathbf{Y}, \mathbf{C} \mid \theta)$, as in Remark 3

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{p(\mathbf{C}|\mathbf{Y},\theta^{(t)})} \left[\log p(\mathbf{Y}, \mathbf{C} \mid \theta)\right].$$

First, let us compute the quantity $p(\mathbf{C} \mid \mathbf{Y}, \theta)$

$$\begin{aligned} p(\mathbf{C} \mid \mathbf{Y}, \theta) &\propto p(\mathbf{Y} \mid \mathbf{C}, \theta) p(\mathbf{C} \mid \theta) \\ &\propto \prod_{i=1}^{N} \prod_{q=1}^{Q} \left(\alpha_q \mathcal{N}(Y_i; \mu_q, \sigma_q^2)\right)^{C_{iq}}. \end{aligned}$$

Since C_{iq} is a binary variable with zeros everywhere except on the coordinate q corresponding to the cluster membership of i, we obtain

$$p(C_{iq} = 1 \mid Y_i, \theta) = \frac{\alpha_q \mathcal{N}(Y_i; \mu_q, \sigma_q^2)}{\sum_{l=1}^{Q} \alpha_l \mathcal{N}(Y_i; \mu_l, \sigma_l^2)} = \gamma_{iq}(\theta),$$

where $\gamma_{iq}(\theta)$ is the posterior probability for i to belong to cluster q. Therefore,

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= \mathbb{E}_{p(\mathbf{C}|\mathbf{Y},\theta^{(t)})} \left[\sum_{i=1}^{n} \sum_{q=1}^{Q} C_{iq} \log\left(\alpha_q \mathcal{N}(Y_i; \mu_q, \sigma_q^2)\right)\right] \\ &= \sum_{i=1}^{n} \sum_{q=1}^{Q} \gamma_{iq}(\theta^{(t)}) \log(\alpha_q) + \gamma_{iq}(\theta^{(t)}) \log(\mathcal{N}(Y_i; \mu_q, \sigma_q^2)). \end{aligned} \tag{2.8}$$

Adding the term $\lambda(1 - \sum_{q=1}^{Q} \alpha_q)$, with $\lambda \in \mathbb{R}$, associated with the constraint $\alpha \in \Delta_Q$ to $Q(\theta \mid \theta^{(t)})$ and computing the first-order conditions gives the following updates:

$$\begin{aligned} \mu_q^{(t+1)} &= \frac{1}{N_q^{(t)}} \sum_{i=1}^{N} \gamma_{iq}(\theta^{(t)}) Y_i, \\ \sigma_q^{2(t+1)} &= \frac{1}{N_q^{(t)}} \sum_{i=1}^{N} \gamma_{iq}(\theta^{(t)}) \|Y_i - \mu_q^{(t+1)}\|^2, \\ \alpha_q^{(t+1)} &= \frac{N_q^{(t)}}{N}, \end{aligned}$$

with $N_q^{(t)} = \sum_{i=1}^{N} \gamma_{iq}(\theta^{(t)})$, the expected size of cluster q. The mean (standard deviation, respectively) of group q corresponds to the classical mean (standard deviation) with each term weighted by the observation contribution to the group q, corresponding to the a posteriori probability membership of the observation to the group $\gamma_{iq}(\theta)$.

2.1.4 The mean-field variational inference

In some cases, the latent posterior distribution $p(\boldsymbol{\eta} \mid \mathbf{Y}, \theta)$ cannot be computed, making the E-step intractable. One way to tackle this issue is to restrain the family distribution $\mathcal{R}$ to cases where the computations can be done. The most common set of distributions that makes the E-step tractable is the *mean-field assumption*. This comes down to assuming that all latent variables are independent one from another. For instance, if $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_p\}$ is a set of p random variables, the mean-field assumptions states that

$$r(\boldsymbol{\eta}) = \prod_{l=1}^{p} r_l(\eta_l),$$

where r_l is a distribution over η_l for all $l \in \{1, \ldots, p\}$. For the sake of brevity, the subscript under the distribution is not specified in the rest of this manuscript and $r_l(\eta_l)$ becomes $r(\eta_l)$. The previous E-step can now be decomposed into p iterative computations, as in a Gibbs sampler, updating each distribution iteratively. Indeed, let us denote $r(\eta_{-l}) = \prod_{k \neq l} r(\eta_k)$, we can write the ELBO with respect to η_l

$$\begin{aligned}
\mathscr{L}(\mathbf{Y}, \theta; r) =& \mathbb{E}_{r(\boldsymbol{\eta})}\left[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\right] - \mathbb{E}_{r(\boldsymbol{\eta})}\left[\log r(\eta)\right] \\
=& \int_{\eta} \log\big(p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\big) \prod_{k=1}^{p} r(\eta_k) d\eta_k - \int_{\boldsymbol{\eta}} \log\left(\prod_{k=1}^{p} r(\eta_k)\right) \prod_{k=1}^{p} r(\eta_k) d\eta_k \\
=& \int_{\eta_l} \left\{ \int_{\eta_{-l}} \log\big(p(\mathbf{Y}, \eta_l, \eta_{-l} \mid \theta)\big) \prod_{k \neq l} r(\eta_k) d\eta_k \right\} r(\eta_l) d\eta_l \\
& - \sum_{k=1}^{p} \int_{\eta_k} \log\big(r(\eta_k)\big) r(\eta_k) d\eta_k \\
=& \int_{\eta_l} \mathbb{E}_{r(\eta_{-l})}\big[\log p(\mathbf{Y}, \eta_l, \eta_{-l} \mid \theta)\big] r(\eta_l) d\eta_l \\
& - \int_{\eta_l} \log\left(r(\eta_l)\right) r(\eta_l) d\eta_l + cst.
\end{aligned}$$

Following Bishop (2006), Chapter 10, Section 1, we can define the quantity

$$\log \tilde{p}(\mathbf{Y}, \eta_l) = \mathbb{E}_{r(\eta_{-l})}\big[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\big] + cst,$$

such that, the ELBO can be written

$$\begin{aligned}
\mathscr{L}(\mathbf{Y}, \theta; r) &= \int_{\eta_l} \log\left(\frac{\tilde{p}(\mathbf{Y}, \eta_l)}{r(\eta_l)}\right) r(\eta_l) d\eta_l + const \\
&= -\operatorname{KL}\left(r(\eta_l) \,||\, \tilde{p}(\mathbf{Y}, \eta_l)\right) + const.
\end{aligned}$$

Hence, the variational distribution $r^\star(\eta_l)$ minimising the ELBO is given by

$$r^\star(\eta_l) \propto \exp\left\{\mathbb{E}_{r(\eta_{-l})}\big[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\big]\right\}. \tag{2.9}$$

At this stage, it is important to note that the expectation on the right-hand side of Equation (2.9) does not involve η_l, thanks to the mean-field assumption. Thus, computing the best variational distribution requires to compute Equation (2.9) for all $l = 1, \dots, L$. Let us remark it is closely related to the updates of a Gibbs sampler, where each variable distribution is updated, keeping the others fixed. In the context of the variational inference, this algorithm is often referred to as *coordinates ascent variational inference* (CAVI, Blei, Kucukelbir, McAuliffe, 2017). The E step of the E-M algorithm becomes the evaluation of each distribution $r^\star(\eta_l)$, that is tractable in many cases, even when $\mathbb{E}_{p(\boldsymbol{\eta}|\mathbf{Y},\theta)}\left[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\right]$ is not.

Algorithm 3: Coordinate ascent variational inference

Input: $t = 0$; Initialise $\theta^{(0)}$ and $r^{(0)}(\eta)$;
while $\mathscr{L}(\mathbf{Y}, \theta^{(t)}; r^{(t)})$ *has not converged* **do**
  for $l = 1, \dots, L$ **do**
    $r^{(t+1)}(\eta_l) \propto \exp\left\{\mathbb{E}_{r^{(t)}(\eta_{-l})}\left[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta^{(t)})\right]\right\}$;
  end
  $\theta^{(t+1)} \in \arg\max_{\theta} \mathscr{L}(\mathbf{Y}, \theta; r^{(t+1)})$;
  $t = t + 1$;
end

The *mean-field* assumption is arguably one of the standard choices in the literature. However, getting rid of this assumption and considering more general variational distributions is still an active research field. For instance, in Saul, Jordan (1995), the authors present a variational inference for a hidden Markov model applied to networks with binary stochastic units. They add to the classical mean-field Boltzmann distribution a term to account for interactions between groups of units. This line of work has grown with Ghahramani (1997) that extended the results of Saul, Jordan (1995) to variational distribution from the exponential family with a polynomial sufficient statistic of arbitrary order. In Hoffman, Blei (2015), the authors restored dependencies between the variables by assuming that the variational distribution is composed of local hidden variables and global parameters. Thus, the variational distribution is factorised conditionally on the global parameter. Taking advantage of the developments in automatic differentiation, and the reparametrisation trick Kingma, Welling (2014); Rezende, Mohamed, Wierstra (2014), Rezende, Mohamed (2015) proposed the normalising flows as smooth invertible transformation, used to parametrised the variational distribution and with easy-to-compute Jacobian matrix to perform efficient learning. Ranganath, Tran, Blei (2016) used a prior distribution on the variational parameters, that are assumed to be dependent, and introduce efficient inference, for instance by using a normalising flow to parametrise the prior distribution as well as a hierarchical ELBO, which is a lower bound of the ELBO.

2.1.5 Optimising the ELBO with gradient-based algorithms

At this point, a major drawback of variational inference is its computational complexity. Indeed, for the sake of clarity, we omitted the subscript denoting the observations, but

assuming that there is a *one-to-one* relationship between the observed variable Y_i and the latent vector η_i, using the notation $r(\boldsymbol{\eta}_{-l}) = \prod_{i=1}^N r(\eta_{i,-l}) = \prod_{i=1}^N \prod_{v \neq l} r(\eta_{i,v})$, the E-step requires to compute for each $l = 1, \dots, L$:

$$\mathbb{E}_{r^{(t)}(\eta_{-l})}\big[\log p(Y, \eta \mid \theta^{(t)})\big] = \sum_{i=1}^N \mathbb{E}_{r^{(t)}(\eta_{-l})}\big[\log p(Y_i, \eta_i \mid \theta^{(t)})\big].$$

Therefore, the E-step requires the computation of NL terms at each iteration, which is intractable for datasets with hundreds of thousands or even millions of observations. This bottleneck prevents variational inference from scaling to large datasets. One way to overcome this issue is to estimate the expectation, using mini-batches. Moreover, the advancements in gradient descent allow to perform optimisation (towards local optimum) without deriving closed-form equations. While this is called *black-box variational inference*, we want to emphasise that the only thing that is unknown are the closed-form updates but the model may be completely explicit.

Here, we propose an algorithm that can be applied to general parametric variational distributions, parametrised by ϕ. Therefore, the ELBO will be denoted $\mathscr{L}(\mathbf{Y}, \theta; \phi)$ instead of $\mathscr{L}(\mathbf{Y}, \theta; r_\phi)$. To begin with, it is worth noticing that obtaining an estimate of the parameter θ is straightforward since

$$\begin{aligned}
\nabla_\theta \mathscr{L}(\mathbf{Y}, \theta; \phi) =& \nabla_\theta \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\big[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\big] \\
=& \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\big[\nabla_\theta \{\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\}\big] \\
=& \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\big[\nabla_\theta \log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta)\big].
\end{aligned}$$

Hence, a classical Monte-Carlo estimate can be obtained by sampling S variables $\boldsymbol{\eta}^s \overset{i.i.d}{\sim} r_\phi(\boldsymbol{\eta})$ and computing

$$\nabla_\theta \hat{\mathscr{L}}(\mathbf{Y}, \theta; \phi) \approx \frac{1}{S} \sum_{s=1}^S \nabla_\theta \log p(\mathbf{Y}, \boldsymbol{\eta}^s \mid \theta). \tag{2.10}$$

However, the same derivation with respect to ϕ fails short because the expectation is taken with respect to $r_\phi(\boldsymbol{\eta})$ that depends on ϕ,

$$\begin{aligned}
\nabla_\phi \mathscr{L}(\mathbf{Y}, \theta; \phi) =& \nabla_\phi \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\big[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\big] \\
\neq& \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\big[\nabla_\phi \{\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\}\big].
\end{aligned}$$

In the following, we introduce two methods addressing this issue, the Reinforce algorithm and the reparametrisation trick.

Reinforce algorithm The reinforce gradients have been introduced for variational inference in Ranganath, Gerrish, Blei (2014) as a mean to estimate gradients of the ELBO with respect to the variational parameters, using Monte-Carlo estimates. In some cases, for instance, if the distributions are in the conjugate-exponential family, closed-form equations may appear for the gradients, simplifying the computations, see Hoffman, Blei, et

al. (2013). Here, no restriction is made on the parametric variational distribution. The gradient of the ELBO with respect to ϕ is given by

$$\begin{aligned}
\nabla_\phi \mathscr{L}(\mathbf{Y}, \theta; \phi) =& \nabla_\phi \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\left[\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\right] \\
=& \int_{\boldsymbol{\eta}} \nabla_\phi \left\{ r_\phi(\boldsymbol{\eta}) \left(\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\right)\right\} d\boldsymbol{\eta} \\
=& \int_{\boldsymbol{\eta}} \nabla_\phi r_\phi(\boldsymbol{\eta}) \left(\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\right) d\boldsymbol{\eta} \\
& - \int_{\boldsymbol{\eta}} r_\phi(\boldsymbol{\eta}) \nabla_\phi \log r_\phi(\boldsymbol{\eta}) \\
=& \int_{\boldsymbol{\eta}} \nabla_\phi r_\phi(\boldsymbol{\eta}) \left(\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\right) d\boldsymbol{\eta} \\
=& \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\left[\nabla_\phi \log r_\phi(\boldsymbol{\eta}) \left(\log p(\mathbf{Y}, \boldsymbol{\eta} \mid \theta) - \log r_\phi(\boldsymbol{\eta})\right)\right],
\end{aligned}$$

where we used that the score function $\nabla_\phi \log r_\phi(\boldsymbol{\eta}) = \nabla_\phi r_\phi(\boldsymbol{\eta}) / r_\phi(\boldsymbol{\eta})$ is centred since $\int_{\boldsymbol{\eta}} r_\phi(\boldsymbol{\eta}) \left(\nabla_\phi r_\phi(\boldsymbol{\eta}) / r_\phi(\boldsymbol{\eta})\right) d\boldsymbol{\eta} = \nabla_\phi \int_{\boldsymbol{\eta}} r_\phi(\boldsymbol{\eta}) d\boldsymbol{\eta} = \nabla_\phi 1 = 0$. Therefore, we managed to express the gradient of the ELBO as an expectation, allowing to use Monte-Carlo methods to approximate this quantity as:

$$\nabla_\phi \hat{\mathscr{L}}(\mathbf{Y}, \theta; \phi) \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log r_\phi(\boldsymbol{\eta}^s) \left(\log p(\mathbf{Y}, \boldsymbol{\eta}^s \mid \theta) - \log r_\phi(\boldsymbol{\eta}^s)\right),$$

with S the number of samples, and $\boldsymbol{\eta}^s \overset{i.i.d}{\sim} r_\phi(\boldsymbol{\eta})$. A major drawback of this method is the large variance of those estimates, which have been addressed using Rao-Blackwelisation and control-variates (Ranganath, Gerrish, Blei, 2014). This method has been observed to give higher variance gradient estimates than the reparametrisation trick presented in the next section.

Reparametrisation trick In Kingma, Welling (2014) and Rezende, Mohamed, Wierstra (2014), the authors proposed to transform the variational distribution using a reparametrisation. Let g be an invertible and differentiable function such that

$$\boldsymbol{\eta} = g(\epsilon, \phi, \mathbf{Y}), \tag{2.11}$$

with ϵ a random variable independent of ϕ and $\mathbf{Y}$. For any integrable function f, the *law of the unconscious statistician*, gives that:

$$\mathbb{E}_{r_\phi(\boldsymbol{\eta})}\left[f(\boldsymbol{\eta})\right] = \mathbb{E}_{p(\epsilon)}\left[f(g(\epsilon, \phi, \mathbf{Y}))\right],$$

and the gradient with respect to ϕ can be computed as

$$\begin{aligned}
\nabla_\phi \mathbb{E}_{r_\phi(\boldsymbol{\eta})}\left[f(\boldsymbol{\eta})\right] &= \nabla_\phi \mathbb{E}_{p(\epsilon)}\left[f(g(\epsilon, \phi, \mathbf{Y}))\right] \\
&= \mathbb{E}_{p(\epsilon)}\left[\nabla_\phi f(g(\epsilon, \phi, \mathbf{Y}))\right] \\
&\approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi f(g(\epsilon^s, \phi, \mathbf{Y})),
\end{aligned}$$

with S the number of ϵ sampled from $p(\epsilon)$.

To end this section, we introduce two core model selection criteria allowing to choose the best model among several, making it possible, for instance, to select the dimension of a space, the number of mixture components or the number of topics, which are crucial questions in many latent variable models, such as mixture models.

2.1.6 Selection model criteria

In this section, we present two selection model criteria, used to choose the *best* model among several. In mixture models, this may be useful to select the number of mixture components.

Bayesian criterion information The data is denoted $\mathbf{Y} = (Y_1, \dots Y_N)$, with $(Y_i)_i$, N independent variables of the same distribution, of unknown density f. Let $\mathcal{M}_m$ be a parametric model, that is a set of densities $\mathcal{M}_m = \{g_{\mathcal{M}_m}(\cdot, \theta), \theta \in \Theta_m\}$ with $g_{\mathcal{M}_m}(\cdot, \theta)$ the density under the model m and parameter $\theta \in \Theta_m$, and $\mathcal{M}_1, \dots, \mathcal{M}_M$, M models that we wish to compare. The Bayesian information criterion (BIC) selects the model with the largest a posteriori probability:

$$\mathcal{M}_{BIC} = \arg\max_{\mathcal{M}_m} p(\mathcal{M}_m \mid \mathbf{Y}). \tag{2.12}$$

Using Bayes' rule, the probability in Equation (2.12) can be written as

$$p(\mathcal{M}_m \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid \mathcal{M}_m)p(\mathcal{M}_m)}{p(\mathbf{Y})}.$$

A non-informative prior on $\mathcal{M}_m$ is then chosen. Hence, finding the best model according to the BIC criterion only requires computing $P(\mathbf{Y} \mid \mathcal{M}_m)$, involving an integral over the parameter θ:

$$p(\mathbf{Y} \mid \mathcal{M}_m) = \int_\theta p(\mathbf{Y}, \theta \mid \mathcal{M}_m) d\theta = \int_\theta g_{\mathcal{M}_m}(\mathbf{Y}, \theta) p(\theta \mid \mathcal{M}_m) d\theta. \tag{2.13}$$

where $g_{\mathcal{M}_m}(\mathbf{Y}, \theta) = \prod_{i=1}^n g_{\mathcal{M}_M}(Y_i, \theta)$.

Lemma 1 (Laplace approximation)**.** *Let $f \colon \mathbb{R}^d \to \mathbb{R}$ be a mapping three times differentiable, that admits a unique maximum at $\theta^\star$, then*

$$\int_\theta e^{nf(\theta)} d\theta = e^{nf(\theta^\star)} \left(\frac{2\pi}{n}\right)^{\frac{d}{2}} | - f''(\theta^\star)|^{-\frac{1}{2}} + O(n^{-1}).$$

Proof. A proof of this lemma can be found in De Bruijn (1981) in Section 4.4 or in Lebarbier, Mary-Huard (2006) in Appendix A. □

Remark 4. *This can be interpreted as assuming that the function is picked at its maximum and decays quickly, as a Gauss curve, implying that the integral, or the surface under the curve, can be well approximated using only the maximum and the Hessian matrix. This lemma gives an approximation of the error and its demonstration relies on a Taylor expansion of the function around its maximum.*

In order to apply the Laplace approximation on the quantity $p(\mathbf{Y} \mid \mathcal{M}_m)$, we use the following transformation

$$g_{\mathcal{M}_m}(\theta) = \log\left(p(\theta \mid \mathcal{M}_m) g_{\mathcal{M}_m}(\mathbf{Y}, \theta)\right) = \log p(\theta \mid \mathcal{M}_m) + \sum_{i=1}^{N} \log g_{\mathcal{M}_m}(Y_i, \theta).$$

Denoting $L_{N,\mathcal{M}_m}(\theta) = \frac{g_{\mathcal{M}_m}(\theta)}{N}$, we can apply the Laplace approximation on $L_{N,\mathcal{M}_m}(\theta)$ such that

$$p(\mathbf{Y} \mid \mathcal{M}_m) = \int_\theta e^{N L_{N,\mathcal{M}_m}(\theta)} d\theta = e^{g_{\mathcal{M}_m}(\theta^\star)} \left(\frac{2\pi}{N}\right)^{\frac{\nu_Y}{2}} |A_{\mathcal{M}_m}(\theta^\star)|^{-\frac{1}{2}} + O_p(N^{-1}),$$

where ν_Y is the number of parameters with respect to $\mathbf{Y}$ (here, the number of parameters in θ) and

$$A_{\mathcal{M}_m}(\theta^\star) = \left(-\frac{\partial^2 L_{N,\mathcal{M}_m}}{\partial\theta_j \partial\theta_l}(\theta^\star)\right)_{1 \le j,l \le d_i}.$$

This can be written as

$$\log p(\mathbf{Y} \mid \mathcal{M}_m) = \log g_{\mathcal{M}_m}(\mathbf{Y}, \theta^\star) + \log p(\theta^\star \mid \mathcal{M}_m) + \frac{\nu_Y}{2} \log\left(\frac{2\pi}{N}\right) - \frac{1}{2} \log\left(|A_{\mathcal{M}_m}(\theta^\star)|\right) + O_p(N^{-1}). \tag{2.14}$$

Unfortunately, the *best* parameter $\theta^\star$ is unknown. Hence, we want to estimate it and plug it into $e^{g_{\mathcal{M}_m}(\theta^\star)}$ as well as in $A_{\mathcal{M}_m}(\theta^\star)$. This can be achieved using the asymptotic convergence of the MLE towards $\theta^\star$, with the MLE given by

$$\hat{\theta}^{MLE} = \arg\max_{\theta \in \Theta} g_{\mathcal{M}_m}(\mathbf{Y}, \theta).$$

Moreover, for any $\theta \in \Theta_i$,

$$L_{N,\mathcal{M}_m}(\theta) = \underbrace{\frac{1}{N} \sum_{i=1}^{N} \log g_{\mathcal{M}_m}(Y_i, \theta)}_{O_p(1)} + \underbrace{\frac{1}{N} \log p(\theta \mid \mathcal{M}_m)}_{O_p(N^{-1})} \underset{N \to \infty}{\longrightarrow} \mathbb{E}\left[\log g_{\mathcal{M}_m}(Y, \theta)\right].$$

Thus, the matrix $A_{\mathcal{M}_m}(\theta^\star)$ can be asymptotically estimated by the Fisher information matrix evaluated for $\hat{\theta}^{MLE}$,

$$I_{\mathcal{M}_m}(\hat{\theta}) = -\mathbb{E}\left[\left(\frac{\partial^2 \log g_{\mathcal{M}_m}(Y, \hat{\theta}^{MLE})}{\partial\theta_j \partial\theta_l}\right)_{1 \le j,l \le d_i}\right]. \tag{2.15}$$

Replacing the true parameter with its estimate introduces an error term of the order of $N^{-1/2}$. Eventually, replacing $\theta^\star$ with $\hat{\theta}^{MLE}$ and assembling together Equations (2.14) and (2.15), we obtain

$$\begin{aligned}
\log p(\mathbf{Y} \mid \mathcal{M}_m) &= \log g_{\mathcal{M}_m}(\mathbf{Y}, \hat{\theta}^{MLE}) - \frac{\nu_Y}{2} \log(N) \\
&\quad + \log p(\hat{\theta}^{MLE} \mid \mathcal{M}_m) + \frac{\nu_Y}{2} \log(2\pi) - \frac{1}{2} \log\left(|I_{\mathcal{M}_m}(\hat{\theta}^{MLE})|\right) + O_p(N^{-1/2}) \\
&\approx \log g_{\mathcal{M}_m}(\mathbf{Y}, \hat{\theta}^{MLE}) - \frac{\nu_Y}{2} \log(N).
\end{aligned} \tag{2.16}$$

The last approximation discards the terms that are constant or decreasing when taking the absolute value, with N. Note that those terms emerge from the assumptions made regarding the prior distributions of the models. As remarked in Raftery (1995), those terms can in practice be much smaller than $O(1)$ by using a reasonable prior, allowing the terms to be of the order of $O(n^{-1/2})$ instead. For instance, this order is recovered with a Gaussian centred around the MLE and with a covariance matrix equal to the inverse of the expected Fisher information matrix. In the end, multiplying by -2 (for historical reasons, as in the AIC criterion), gives the BIC criterion:

$$\mathcal{M}_{BIC} = \arg\min_{\mathcal{M}_m} \; -2\log\left(\sum_{i=1}^{N} g_{\mathcal{M}_m}(Y_i, \hat{\theta}^{MLE})\right) + \nu_Y \log(N).$$

Interested readers may refer to the very good introduction to the BIC criterion proposed by Lebarbier, Mary-Huard (2006) that inspired most of this section, as well as Raftery (1995).

ICL In the context of a finite mixture model, that is a model with observations sampled from Q different distributions, the BIC-like approximation is no longer theoretically backed, in particular if the true number of groups $Q' < Q$, it implies that $Q - Q'$ parameters will tend to zero and be on the boundary space, which prevents from using the BIC criterion (Biernacki, Celeux, Govaert, 2000). Moreover, the BIC is not designed for latent variable models, and the Q groups are not observed, thus the clustering purpose is not incorporated into the criterion (this holds true for other types of latent variables). Indeed, the BIC objective of Equation (2.12) becomes

$$\log p(\mathbf{Y}, \mathbf{C} \mid \mathcal{M}_m) = \int_{\theta} \log p(\mathbf{Y}, \mathbf{C} \mid \theta, \mathcal{M}_m) p(\theta \mid \mathcal{M}_m) d\theta. \tag{2.17}$$

Assuming an a priori distribution on θ factorising as $p(\theta) = p(\mu, \sigma)p(\alpha)$ allows to split Equation (2.17) in two terms:

$$\log p(\mathbf{Y}, \mathbf{C} \mid \mathcal{M}_m) = \log p(\mathbf{Y} \mid \mathbf{C}, \mathcal{M}_m) + \log p(\mathbf{C} \mid \mathcal{M}_m),$$

with the first term that can be decomposed using a BIC-like approximation and the second term that can be computed explicitly, see Lemma 3.1 in Biernacki, Celeux, Govaert (2000). Indeed, since each C_i is a binary variable with zeros everywhere except on the coordinate q corresponding to the cluster membership of i, we have that for any $i \in \{1, \ldots, N\}$, $C_i \sim \mathcal{D}(\delta_1, \ldots, \delta_Q)$, with $\delta = (\delta_1, \ldots, \delta_Q)$. Using Equation (2.16), we obtain that,

$$\log p(\mathbf{Y}, \mathbf{C} \mid \mathcal{M}_m) \approx \max_{\mu,\sigma} \log p(\mathbf{Y} \mid \mathbf{C}, \mu, \sigma, \mathcal{M}_m) - \frac{\nu_Y}{2}\log(N) + \log p(\mathbf{C} \mid \mathcal{M}_m),$$

with the term $\log p(\mathbf{C} \mid \mathcal{M}_m)$ equal to

$$\begin{aligned} p(\mathbf{C} \mid \mathcal{M}_m, Q) &= \int p(\mathbf{C} \mid \delta', \mathcal{M}_m, Q) p(\delta') d\delta' \\ &= \frac{\Gamma\left(\sum_{q=1}^{Q} \delta_q\right)}{\prod_{q=1}^{Q} \Gamma\left(\delta_q\right)} \frac{\prod_{q=1}^{Q} \Gamma(n_q + \delta_q)}{\Gamma\left(\sum_{q=1}^{Q} n_q + \delta_q\right)}, \end{aligned}$$

where $n_q := \sum_{i=1}^{N} C_{iq}$. To obtain a simpler criterion, one can use a BIC-like approximation on $\log p(\mathbf{C} \mid \mathcal{M})$, assuming that all the n_q are large enough, which comes down to

$$\begin{aligned} \log p(\mathbf{Y}, \mathbf{C} \mid \mathcal{M}_m) \approx & \max_{\mu,\sigma} \log p(\mathbf{Y} \mid \mathbf{C}, \mu, \sigma, \mathcal{M}_m) - \frac{\nu_Y}{2} \log(N) \\ & + \max_{\delta} \log p(\mathbf{C} \mid \delta, \mathcal{M}_m) - \frac{Q-1}{2} \log(N). \end{aligned} \tag{2.18}$$

To match the BIC criterion, we can multiply the last equation by -2 and obtain

$$\begin{aligned} \mathcal{M}_{ICL} = \arg\min_{\mathcal{M}_m} & -2 \log p(\mathbf{Y} \mid \mathbf{C}, \hat{\mu}, \hat{\sigma}, \mathcal{M}_m) + \nu_Y \log(N) \\ & -2 \log p(\mathbf{C} \mid \hat{\delta}, \mathcal{M}_m) + (Q-1) \log(N), \end{aligned} \tag{2.19}$$

where $\hat{\mu}, \hat{\sigma}$ and $\hat{\delta}$ are solutions of the maximisation problem in Equation (2.18).

Remark 5. *While the ICL criterion has been developed for mixture models, we extended it to the more general case of hierarchical latent distributions combining continuous and discrete latent variables as we shall see in Chapter 4.*

2.2 Deep generative models

During the last decade, probabilistic modelling has known new developments with the emergence of deep neural networks as an efficient way to encode and decode data. Before diving into the deep probabilistic modelling details, we give a quick reminder on deep neural networks. Then, a summary concerning deep probabilistic models as well as an associated inference strategy are presented. This section will end with the variational autoencoder, presented in Kingma, Welling (2014); Rezende, Mohamed, Wierstra (2014), that made possible the scaling to large datasets.

2.2.1 Supervised deep learning in a nutshell

This section aims at presenting the basics of deep learning, in the supervised setting. To begin with, let us consider a d-dimensional random variable X and Y a target random variable such that the two are assumed to be linked through an unknown function $f^\star$ as

$$Y = f^\star(X) + \epsilon,$$

with ϵ a 0 noise variable. This setting raises two major questions.

First, given a function f, it is necessary to evaluate the goodness of this function to approximate Y given X. To do so, we introduce a *loss function* L to compare $f(X)$ with Y. Eventually, we want to minimize the *risk*, associated with L and the joint-distribution $p(X, Y)$

$$R(f(X), Y) = \mathbb{E}_{p(X,Y)} \left[L(f(X), Y) \right]. \tag{2.20}$$

Unfortunately, the distribution $p(X, Y)$ is unknown, making the computation of (2.20) intractable. Thus, using the observed data $(\mathbf{X}, \mathbf{Y}) = ((X_1, \ldots, X_N), (Y_1, \ldots, Y_N))$, the *empirical risk* $\hat{R}_N$ is used to estimate (2.20):

$$\hat{R}_N(f(\mathbf{X}), \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} L(f(X_i), Y_i). \tag{2.21}$$

As an example, let us mention two of the most used loss functions, namely the Euclidean distance for continuous variables, $L(f(X), Y) = \|f(X) - Y\|^2$, and the standard cross-entropy for categorical data $L(f(X), Y) = -\sum_{q=1}^{Q} Y_q \log f(X)_q$.

Second, it is necessary to restrict the search of f to a class of function $\mathcal{F}$ for the estimation to be tractable. While famous family comes to mind (the linear model for regression, the logistic function for binary classification), more complex types of functions have been studied during the past fifty years. In particular, *neural networks* have raised a lot of attention in the past thirty years, due to excellent prediction results and state-of-the-art models in numerous domains (vision, object detection, natural language processing, and many others). We give a brief presentation of the feed-forward neural network, arguably the simplest deep neural network architecture, that will be the core of the deep neural networks used later on.

Feed-forward neural network

We would like to define a class of functions that contains good approximations of f. A feed-forward neural network is a mapping f_θ defined as

$$f_\theta : \left\{ \begin{array}{ccc} \mathbb{R}^d & \longrightarrow & \mathbb{R}^{d_{out}} \\ x & \mapsto & h_L \circ \cdots \circ h_1(\mathbf{X}), \end{array} \right. \tag{2.22}$$

where $\theta = (\mathbf{W}_l, \mathbf{b}_l)_l$, and the result of each function $h_l(\cdot)$ is called a *layer*. It is defined as a linear operation followed by an activation function, often non-linear,

$$h_l : \left\{ \begin{array}{ccc} \mathbb{R}^{d_l} & \longrightarrow & \mathbb{R}^{d_{l+1}} \\ x & \mapsto & \sigma_l \left(\mathbf{W}_l x + \mathbf{b}_l \right), \end{array} \right. \tag{2.23}$$

with $\mathbf{W}_l \in \mathcal{M}_{d_l \times d_{l+1}}(\mathbb{R})$, $\mathbf{b}_l \in \mathbb{R}^{d_{l+1}}$ and $\sigma_l(\cdot)$ an activation function applied element-wise. The dimension d_l denotes the number of *units* in layer l. When $L = 1$, this function is called a *shallow neural network*. On the contrary, when $L > 1$, which will always be the case in this manuscript, such a function is called a multilayer perceptron (MLP). This class of function will be denoted $\mathcal{F}_{\text{MLP}} = \{f_\theta, \theta \in \Theta\}$. The parameters $\mathbf{W}_l$ and $\mathbf{b}_l$ need to be estimated in order to obtain the *best* function $\hat{f}_\theta$ for our problem

$$\hat{f}_\theta = \underset{f_\theta \in \mathcal{F}_{\text{MLP}}}{\arg\min}\ \hat{R}_n(f(\mathbf{X}), \mathbf{Y}).$$

Using non-linear activation functions permits to capture non-linear features of the data. Indeed, in Hornik, Stinchcombe, White (1989), the authors showed that any continuous

function can be approximated as well as wanted, given enough units. However, in practice, the number of units required to obtain a *good* approximation is not known. In Montufar et al. (2014), the authors showed that the number of linear regions carved out by deep neural networks using rectified linear units as activation functions increases exponentially with the depth of the network. Moreover, empirical works suggest that greater depth results in better generalisation (Bengio et al., 2006) and advocate that a deeper network can be seen as a useful prior over the space of functions.

Back-propagation algorithm, learning and stochastic gradient descent

The next section presents the back-propagation combined with the stochastic gradient descent algorithm, to be able to optimise quickly the parameters of the deep neural network using very large datasets. This capacity of scaling to high dimensional data and a large number of observations plays a central role in the spread of these techniques in real-life applications.

In order to optimise the parameters of the neural network, we would like to be able to compute the gradient of the loss function with respect to each parameter to be able to use a gradient descent algorithm. For certain well-chosen loss functions and activation functions, the differentiation of each layer with respect to its parameters is straightforward and has a closed form (no approximation is required). Therefore, the gradients of the objective function with respect to the parameters of the d-th units of layer l can be computed analytically by using the chain rule, as detailed in Chapter 6, Section 5 in the book by Goodfellow, Bengio, Courville (2016). It makes it possible to build an efficient learning algorithm (LeCun et al., 1998). In particular, the *automatic differentiation*, implemented in computer science packages, such as Pytorch Paszke et al. (2019), builds on the GPU's efficiency to parallelise computations when performing matrix multiplications and the construction of a graph to efficiently back-propagate during the gradient descent algorithm. This is crucial to quickly compute gradients, which is one of the building bricks to obtain efficient optimisation algorithms for deep neural networks. Thus, we need to compute the gradients of our objective function at each step, such that,

$$\hat{\nabla}_\theta \hat{R}_N(f_\theta(\mathbf{X}), \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^{N} \hat{\nabla}_\theta L(f_\theta(X_i), Y_i).$$

Since this computation has to be done for each observation, it becomes intractable or even inefficient for datasets with hundreds of thousands or even millions of observations N, as pointed out in Bottou, Bousquet (2007) where the authors compare different optimisation algorithms and their performances. To reduce the computational cost, sampling B observations from the datasets to form the mini-batch $\mathcal{B}$, with $B << N$ can drastically diminish the computation, and provide faster good approximations of the gradient. A Monte-Carlo estimate of the gradient using only the mini-batch $\mathcal{B}$ is obtained by computing:

$$\nabla_\theta R(f_\theta(X), Y) \approx \nabla_\theta \hat{R}_B(f_\theta(\mathbf{X}_\mathcal{B}), \mathbf{Y}_\mathcal{B}) = \frac{1}{B} \sum_{(X_i, Y_i) \in \mathcal{B}} \nabla_\theta L(f_\theta(X_i), Y_i).$$

Repeating this scheme iteratively gives Algorithm 4.

Algorithm 4: The stochastic gradient algorithm

Input: θ_0, f, $\mathbf{X}$, $\mathbf{Y}$

while θ_t *has not converged* **do**

  Sample the mini-batch $\mathcal{B}$;

  Compute the gradient estimate $\hat{g}_t = \frac{1}{B} \sum_{(X_b, Y_b) \in \mathcal{B}} \hat{\nabla}_\theta L(f_\theta(X_b), Y_b)$;

  Update the parameter $\theta_{t+1} = \theta_t - a_t \hat{g}_t$;

  $t = t + 1$;

end

More information about deep learning theory can be found in the book by Goodfellow, Bengio, Courville (2016).

2.2.2 Unsupervised deep learning with latent representation of the data

Among the important tasks solved by DNN, one is known as *representation learning*. Contrary to the previous setting, we are no longer performing *supervised learning*, with an observation X and a target variable Y, but *unsupervised learning* instead. Let $\mathbf{Y} = (Y_1, \dots, Y_N)$ be N independent samples of a distribution p. We would like to find a homogeneous representation h of the heterogeneous data Y, using a smooth function f. For instance, the principal component analysis (PCA) learns a latent representation $h \in \mathbb{R}^d$ of the data $Y \in \mathbb{R}^p$, with $d < p$, using a linear mapping such that $h = f(Y) = \mathbf{W}^\top Y + \mathbf{b}$. This can be extended to non-linear functions, to extract non-linear features. In particular, the expressiveness of deep neural networks has played a major role in the development of the field of representation learning. We present two core models, the autoencoder and the variational autoencoder, that are central to this manuscript.

Autoencoder

An autoencoder is a function composed of an encoder, or feature-extracting mapping, $h = \text{Encoder}_\phi(Y)$, and a decoder, or reconstruction mapping $\hat{Y} = \text{Decoder}_\theta(h)$, aiming at copying its input, while learning a meaningful hidden layer, or representation, h. The objective function of this problem can be written as

$$\hat{\phi}, \hat{\theta} = \arg\min_{\phi,\theta} \hat{L}_N(\mathbf{Y}; \theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} L(Y_i, \text{Decoder}_\theta(\text{Encoder}_\phi(Y_i))).$$

with $L(\cdot, \cdot)$ a loss function (for instance the Euclidean distance for continuous data). This function can then be optimised using (stochastic) gradient descent algorithm with respect to ϕ, θ. When the dimension of h is less than the dimension of Y, the autoencoder is called *undercomplete* and is forced to learn the most salient features of the data to be able to

replicate Y. For instance, if we assume that the decoder is linear and the loss function is the MSE, an under complete autoencoder will learn to span the same subspace as the one obtained with a PCA. Using a non-linear decoder will permit to obtain more complex features than the standard linear PCA. However, if the encoder and decoder are complex enough, they may be able to encode and decode each data point. For instance, Y_i could be encoded as i and i be decoded as Y_i, without learning any aspect of the data distribution. To prevent this and to obtain a meaningful latent space, different regularisations have been investigated. In particular, a sparsity constraint using a lasso-like penalty has been used Ranzato et al. (2006). This can be seen as using a Laplace prior over the model distribution over the latent variable h. More details can be found Chapter 14, Section 2.2 of the book by Goodfellow, Bengio, Courville (2016).

Applications: words embeddings

One of the advantages of learning representation is the ability to represent discrete data in a continuous space. In Mikolov, Chen, et al. (2013); Mikolov, Sutskever, et al. (2013), the authors proposed the skipgram model as well as the continuous-bag of words model (CBOW) to efficiently learn continuous representations of words from a large corpus, using the context of a word, which corresponds to its closest neighbours. Let $\mathbf{W} = (W_1, \dots, W_D)$ be a corpus of text, with D documents. A window size c is chosen such that the documents are now decomposed into sequences of $2c+1$ words, centred around a word w_t. The document indicator is dropped from the notations since all sequences are treated as i.i.d observations. Then, each word is one-hot encoded, such that the t-th word $w_t \in \{0,1\}^V$, with 0 everywhere excepted on the coordinate corresponding to its index in the vocabulary. We denote the neighbours of the t-th word $\mathbf{n}_t = (w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c})$, the indices of the neighbours $i(t) = \{t-c, \dots, t-1, t+1, \dots, t+C\}$, the input weights are denoted $\rho^I = (\rho_1, \dots, \rho_V) \in \mathcal{M}_{L\times V}$ and the output weights, corresponding to the word embeddings, $\rho^O = (\rho'_1, \dots, \rho'_V)^\top \in \mathcal{M}_{V\times L}$. The skipgram model estimates for any $t' \in i(t)$, the *distribution of a word to be in the context given the centred word of the sequence* w_t as,

$$p(w_{t',v'} = 1 \mid w_{t,v} = 1) = \frac{\exp({\rho'_{v'}}^\top \rho_v)}{\sum_{k=1}^V \exp({\rho'_k}^\top \rho_v)}.$$

Eventually, the matrix ρ^O can be used as embeddings of the words to perform operations such as measuring similarities or distance between words in a continuous space.

On the contrary, CBOW aims at predicting the word at the centre using its context. The notation v_t is used to refer to the vocabulary index of the t-th word such that $w_{t,v_t} = 1$. The estimated *probability of a word given the context* is here defined as

$$p(w_{t,v} = 1 \mid \mathbf{h}_t) = \text{softmax}\left(\frac{1}{C}\sum_{t'\in i(t)} {\rho_{v_{t'}}}^\top \rho'_v\right)_v.$$

Eventually, the cross-entropy is used as the loss function to evaluate the estimated probabilities. The model can be trained using a stochastic gradient descent algorithm to

optimise the loss function with respect to the *word embeddings* ρ^O and the input weights ρ^I. Since we only wish to provide an example of an autoencoder, we only give the models and architecture of the networks. The implementation details are out of the scope of this manuscript. However, the popularity of those models is due to their ability to scale to very large datasets, up to millions of samples in the case of the CBOW, thanks to the hierarchical softmax and the negative sampling, as described in the original paper (Mikolov, Sutskever, et al., 2013).

2.2.3 Variational autoencoder

One of the drawbacks of autoencoders is the necessity to choose a loss function. In addition, the latent space obtained using an autoencoder may lack of regularity. Some regularisation techniques have been proposed for autoencoders, as we already mentioned. Remember that for classical autoencoders, $\boldsymbol{\eta}$ does not incorporate any randomness and is considered as a determinist transformation of $\mathbf{Y}$, such that we can define the distribution of the data as

$$p_\theta(\mathbf{Y}, \boldsymbol{\eta}) = p(\mathbf{Y} \mid \text{Decoder}_\theta(\boldsymbol{\eta}))p(\boldsymbol{\eta}).$$

While classical autoencoders directly maximise the term $p(\mathbf{Y} \mid \text{Decoder}_\theta(\boldsymbol{\eta}))$ where $\boldsymbol{\eta} = \text{Encoder}_\phi(\mathbf{Y})$, with an external regularity term, the variational autoencoder assumes that $\boldsymbol{\eta}$ incorporates randomness and is therefore treated as in a latent variable model. Hence, for computational efficiency, the variational inference is combined with the reparametrisation trick. This unleashes the possibility to use (deep) parametric functions to *encode* the observed data $\mathbf{Y}$ into the parameters of the variational distribution. While this may seem restrictive (since it prevents the parameter from being free), the choice of the function as well as the use of the same function for all observations, called amortised inference Gershman, Goodman (2014), induces a modelling choice. Moreover, instead of increasing the number of parameters linearly with the number of latent variables, this can be seen as learning a more general approximation, in the sense that it should generalise on unseen data. Let the variational distribution be

$$r_\phi(\boldsymbol{\eta} \mid \mathbf{Y}) = r(\boldsymbol{\eta} \mid \text{Encoder}_\phi(\mathbf{Y})).$$

Contrary to what we have presented so far, the variational distribution is conditioned on the observation. This gives the following ELBO

$$\begin{aligned}\mathscr{L}(\theta, \phi) &= \mathbb{E}_{r_\phi}\left[\log p_\theta(\mathbf{Y}, \boldsymbol{\eta})\right] - \mathbb{E}_{r_\phi}\left[\log r_\phi(\boldsymbol{\eta})\right] \\ &= \underbrace{\mathbb{E}_{r_\phi}\left[\log p(\mathbf{Y} \mid \text{Decoder}_\theta(\boldsymbol{\eta}))\right]}_{\text{Reconstruction error}} - \underbrace{\text{KL}\left(r(\boldsymbol{\eta}; \text{Encoder}_\phi(\mathbf{Y})) \,||\, p(\boldsymbol{\eta} \mid \mathbf{Y})\right)}_{\text{Regularisation term}}.\end{aligned}$$

Interestingly, we retrieve the objective of the autoencoder, the reconstruction term, penalised by the Kullback-Leibler divergence term, which can be seen as a regularising term

arising from the modelling choices, in particular, in the architecture of the encoder. The standard choice for the variational distribution, when dealing with continuous data, is the Gaussian distribution, such that:

$$\begin{aligned}(\mu_\phi(\mathbf{Y}), \sigma_\phi(\mathbf{Y})) &= \text{Encoder}_\phi(\mathbf{Y}),\\ r_\phi(\boldsymbol{\eta} \mid \mathbf{Y}) &= \mathcal{N}(\boldsymbol{\eta}; \mu_\phi(\mathbf{Y}), \sigma_\phi(\mathbf{Y})^2).\end{aligned}$$

To be able to estimate the parameters of the neural network ϕ, we use the reparametrisation trick

$$g(\epsilon_i, \phi, \mathbf{Y}) = \mu_\phi(\mathbf{Y})_i + \sigma_\phi(\mathbf{Y})_i \epsilon,$$

and $\epsilon = (\epsilon_i)_i$. Hence, the ELBO can be written as

$$\mathscr{L}(\theta, \phi) = \mathbb{E}_{p(\epsilon)}\left[\log p_\theta(\mathbf{Y} \mid \boldsymbol{\eta})\right] + \mathbb{E}_{p(\epsilon)}\left[\log p_\theta(\boldsymbol{\eta})\right] - \mathbb{E}_{p(\epsilon)}\left[\log r_\phi(\boldsymbol{\eta})\right],$$

where $\eta_i = g(\epsilon_i, \phi, \mathbf{Y})$. Using Equation (2.10) with $\eta_i = g(\epsilon_i, \phi, \mathbf{Y})$, we can directly obtain Monte Carlo estimates of the gradient and optimise $\mathscr{L}(\theta, \phi)$ as described in Algorithm 5.

Algorithm 5: Inference of a variational autoencoder.

Input: $\phi^{(0)}$, $\theta^{(0)}$, $t = 0$
while *ELBO has not converged* **do**
  Sample a minibatch $\mathcal{B}$;
  Sample $\epsilon_i \sim p(\epsilon)$ for every datapoints in $\mathcal{B}$;
  Compute the approximation of the ELBO $\hat{\mathscr{L}}(\theta^{(t)}, \phi^{(t)}, \epsilon)$ and the gradients $\nabla_{\theta,\phi}\hat{\mathscr{L}}(\theta^{(t)}, \phi^{(t)}, \epsilon)$;
  Update $\theta^{(t)}$ and $\phi^{(t)}$ using a stochastic gradient descent optimiser;
end

Although we only presented the VAE for Gaussian distributions, other distributions can be considered. In particular, approximations of discrete variational distributions have emerged over the year, Nalisnick, Smyth (2016); Maddison, Mnih, Teh (2017). More complex distribution has also been considered by transforming the parameters with invertible and differentiable functions (Rezende, Mohamed, 2015). For more details regarding variational autoencoders, the reader can refer to Kingma, Welling, et al. (2019).

To summarise this section, let us insist on the possibilities that offer deep probabilistic models. In particular, variational autoencoders permit to use complex hierarchical variational distributions, which in turn allows for meaningful latent representations. In addition, deep neural networks can be used to encode the data into the latent space without increasing the number of parameters with the size of the samples. This differs from traditional variational inference and translates the capacity of the encoder to represent well the data into the latent space. Neural networks can also be used to parametrise the model distribution in the decoder part. Using the ELBO as the objective function, efficient stochastic gradient descent algorithms, such as Adam (Kingma, Ba, 2014), showed very good results in practice.

2.3 Topic Modelling

This section starts by presenting the rise of probabilistic models in topic modelling. In particular, we will present the latent Dirichlet allocation, arguably one of the most central models in the field. Eventually, we will present new developments, relying on variational autoencoders introduced previously.

2.3.1 The rise of generative models in topic modelling

The statistical analysis of topics emerged in the late 90s with Papadimitriou et al. (1998), developing statistical results for the latent semantic indexing (LSI), first proposed by Deerwester et al. (1990). LSI relies on a spectral analysis of the "term frequency-inverse document frequency" and successfully captures synonymy between words. To overcome the lack of probabilistic foundations of LSI, Hofmann (1999) introduced the probabilistic latent semantic index (pLSI) which models each word distribution as a mixture model such that each mixture component corresponds to a "topic". The topic membership of each word is modelled by a multinomial random variable in pLSI. Even though the topic membership of the words depends on the document, a major drawback of pLSI is the absence of a model at the document level. This was overcome by Blei, Ng, Jordan (2003) with the latent Dirichlet allocation (LDA).

2.3.2 The latent Dirichlet allocation

In this manuscript, the documents will be denoted $\mathbf{W} = (\mathbf{W}_1, \dots, \mathbf{W}_D)$, with D the number of documents, and $\mathbf{W}_d = (w_{dn}^v)_{n,v} \in \mathcal{M}_{N_d \times V}(\{0,1\})$ such that $w_{dn}^v = 1$ if the n-th word of document d is the v-th word of the vocabulary, and 0 otherwise. The size of the vocabulary is denoted V. Moreover, the word *topic* will be used to refer to a distribution over the vocabulary. In other words, a topic $k \in \{1, \dots, K\}$, where K denotes the number of topics, is represented by a vector $\beta_k \in \Delta_V$, where Δ_V denotes the V-dimensional simplex and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$ denotes the topic matrix. Let $\mathbf{T} = (\mathbf{T}_1, \dots, \mathbf{T}_d)$ be the topic memberships, with $\mathbf{T}_d = (t_{d1}, \dots, t_{dN_d})^\top \in \mathcal{M}_{N_d \times K}(\{0,1\})$ the topic memberships of words in document d, such that $t_{dn}^k = 1$ if the n-th word of document d is from topic k and 0 otherwise. The latent Dirichlet allocation assumes that each document is sampled from a mixture of topics with its own proportion denoted θ_d, and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_D)$. The generative model is summarised in Generative model 1.

Generative model 1: Generation of the corpus under the latent Dirichlet allocation model.

Parameters: $\boldsymbol{\beta}$,θ_0
for $d = 1, \dots, D$ **do**
  Draw the mixture proportions associated with document d:
  $\theta_d \sim Dir_K(\theta_0)$;
  for $n = 1, \dots, n_d$ **do**
    Draw the topic of n-th word: $t_{dn} \sim \mathcal{M}_K(1; \theta_d)$;
    Draw the n-th word: $w_{dn} \mid t_{dn}^k = 1 \sim \mathcal{M}_V(1; \beta_k)$;
  end
end

Using the generative model, the complete distribution of document d can be factorised as

$$\begin{aligned} p(\mathbf{W}_d, \mathbf{T}_d, \theta_d \mid \theta_0, \boldsymbol{\beta}) &= p(\theta_d \mid \theta_0) p(\mathbf{T}_d \mid \theta_d) p(\mathbf{W}_d \mid \mathbf{T}_d, \boldsymbol{\beta}) \\ &= p(\theta_d \mid \theta_0) \prod_{n=1}^{N_d} p(t_{dn} \mid \theta_d) p(W_{dn} \mid t_{dn}, \boldsymbol{\beta}). \end{aligned}$$

Hence, the likelihood of document d requires to marginalise on θ_d and t_d,

$$p(\mathbf{W}_d \mid \theta_0, \boldsymbol{\beta}) = \int_{\theta_d} p(\theta_d \mid \theta_0) \left(\prod_{n=1}^{N_d} \sum_{t_{dn}} p(w_{dn} \mid t_{dn}, \boldsymbol{\beta}) p(t_{dn} \mid \theta_d) \right) d\theta_d.$$

Given the parameters β and θ_0, the distribution can be factorised over the documents, resulting in the following observed likelihood:

$$p(\mathbf{W} \mid \theta_0, \boldsymbol{\beta}) = \prod_{d=1}^{D} \int_{\theta_d} p(\theta_d \mid \theta_0) \left(\prod_{n=1}^{N_d} \sum_{t_{dn}} p(w_{dn} \mid t_{dn}, \boldsymbol{\beta}) p(t_{dn} \mid \theta_d) \right) d\theta_d.$$

Since the integral is not tractable, we can rely on the variational mean-field inference introduced in Section 2.1.4, with a variational distribution factorising as

$$r_\phi(\mathbf{T}, \boldsymbol{\theta}) = \prod_{d=1}^{D} \left(r(\theta_d; \tilde{\theta}_d) \prod_{n=1}^{N_d} r(t_{dn}; \tilde{t}_{dn}) \right),$$

with $\phi = ((\tilde{\theta}_d)_d, (\tilde{t}_{dn})_{d,n})$. Hence, the ELBO is obtained as

$$\mathscr{L}(\theta_0, \boldsymbol{\beta}; r_\phi) = \mathbb{E}_{r_\phi} \left[\log p(\mathbf{W}, \mathbf{T}, \boldsymbol{\theta} \mid \theta_0, \boldsymbol{\beta}) \right] - \mathbb{E}_{r_\phi} \left[\log r_\phi(\mathbf{T}, \boldsymbol{\theta}) \right].$$

Using the first-order conditions, as well as the conjugacy properties on the exponential family (see Section 4 of Blei, Kucukelbir, McAuliffe, 2017), of the ELBO with respect to the variational parameters (E-step), we obtain the following approximated posterior distribution of the latent variables

$$\begin{aligned} r(t_{dn}; \tilde{t}_{dn}^\star) &= \mathcal{M}_K(t_{nd}; 1, \tilde{t}_{dn}^\star), \\ r(\theta_d; \tilde{\theta}_d^\star) &= \mathcal{D}_K \left(\theta_d; \tilde{\theta}_d^\star \right), \end{aligned}$$

where

$$\tilde{t}_{dn}^{\star k} \propto \beta_{k,w_{dn}} \exp\left(\mathbb{E}_{r_\phi}\left[\log \theta_{dk}\right]\right),$$
$$\tilde{\theta}_d^{\star} = \theta_0 + \sum_{n=1}^{n_d} \tilde{t}_{dn}^{\star},$$

and $\mathbb{E}_{r_\phi}\left[\log\left(\theta_{dk}\right)\right] = \Psi\left(\tilde{\theta}_k\right) - \Psi\left(\sum_{l=1}^{K} \tilde{\theta}_l\right)$, (see Appendix A.1 in Blei, Ng, Jordan, 2003).

Next, optimising the ELBO with respect to parameter β (M-step) using first-order conditions provides the following update,

$$\beta_{kv} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \tilde{t}_{dni}^{\star} w_{dn}^{j}.$$

An online version of this algorithm has been proposed in Hoffman, Bach, Blei (2010). However, the Dirichlet distribution makes the topics almost uncorrelated and does not directly model correlation. Blei, Lafferty (2006) then proposed to use a normal-logistic prior instead of a Dirichlet prior on the topic proportion to directly model the correlations. All these models require to derive the equations for any new generative model. In Srivastava, Sutton (2017), they bridged the gap between topic modelling and autoencoders, taking full advantage of gradient descent for those models. Nevertheless, all the former approaches do not incorporate semantic meaning to the words, as embeddings such as the CBOW or skipgram presented in Section 2.2.2. Indeed, since the model is only based on the document term-frequency matrix, they lose the information provided by the order of the words. The embedded topic model (ETM), Dieng, Ruiz, Blei (2020) used the strength of word embeddings, such as the continuous bag of words (CBOW) or skipgram (Mikolov, Chen, et al., 2013) as a part of the decoder of a variational autoencoder. We give a brief overview of this model, that can be considered as an extension of LDA

2.3.3 The embedded topic model and other neural topic models

To benefit from the neural network representational power, let each word $v \in \{1, \ldots, V\}$, where V denotes the vocabulary size, be represented by a vector $\rho_v \in \mathbb{R}^L$, called embedding in literature, such that $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_V) \in \mathcal{M}_{L\times V}(\mathbb{R})$ is the embedding matrix. Similarly, each topic k is represented in the same latent space by a vector $\alpha_k \in \mathbb{R}^L$, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \in \mathcal{M}_{L\times K}(\mathbb{R})$ such that the distribution associated with topic k is obtained as

$$\beta_k = \text{softmax}\left(\boldsymbol{\rho}^\top \alpha_k\right),$$

and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)^\top \in \mathcal{M}_{K\times V}$. In addition, the generative model marginalises over the word-topic membership variables as described in Generative model 2.

Generative model 2: Generation of the corpus under the embedded topic model.

Parameters: $\boldsymbol{\rho}$, α, $\beta_k = \text{softmax}\left(\boldsymbol{\rho}^\top \alpha_k\right)$, $k = 1, \dots, K$
for $d = 1, \dots, D$ **do**
  Draw the variable associated with the topic proportions:
  $\delta_d \sim \mathcal{N}_K(0, I)$;
  Compute the topic proportions:
  $\theta_d = \text{softmax}(\delta_d)$;
  for $n = 1, \dots, n_d$ **do**
    Draw the word n: $w_{dn} \sim \mathcal{M}_V(1; \theta_d^\top \beta)$;
  end
end

As for the LDA, the likelihood can be obtained by marginalising over the latent variable such that

$$p(\mathbf{W} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}) = \int_\delta p(\mathbf{W} \mid \boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\rho}) p(\boldsymbol{\delta}) d\boldsymbol{\delta}.$$

Unfortunately, the softmax function prevents from obtaining any closed form. Therefore, as in the LDA, we rely on a variational inference strategy. However, in order to fully take advantage of the variational autoencoder, the variational distribution is now parametrised using a neural network such that

$$r(\boldsymbol{\delta} \mid \mathbf{W}) = \prod_{d=1}^{D} \mathcal{N}_K(\delta_d; \mu_\phi(\mathbf{W})_d, \text{diag}(\sigma_\phi^2(\mathbf{W})_d)).$$

where ϕ denotes the neural network encoder parameters. We can directly obtain the ELBO as

$$\mathscr{L}(\boldsymbol{\alpha}, \boldsymbol{\rho}; r_\phi) = \mathbb{E}_{r_\phi}\left[\log p(\mathbf{W}, \boldsymbol{\delta} \mid \boldsymbol{\alpha}, \boldsymbol{\rho})\right] - \mathbb{E}_{r_\phi}\left[\log r_\phi(\boldsymbol{\delta})\right].$$

Using the reparametrisation trick to approximate the ELBO and the gradients, the optimisation can be carried out with a stochastic gradient descent algorithm, such as Adam, and an automatic differentiation package such as Pytorch. In practice, ETM allows to initialise the matrix ρ with embeddings already pre-trained on very large corpora, with CBOW or skipgram model for instance. In practice, using pre-trained embeddings showed a significant improvement in the metrics. Other neural topic models have been developed during the last decade (Meng et al., 2020; Zhao, Phung, Huynh, Le, et al., 2021). In particular, in Wu et al. (2023), the authors managed to overcome one of the main limits of ETM, named *topic collapsing*. Indeed, Zipf's law (Piantadosi, 2014) states that few words have a high frequency and many have a low frequency, such that the word distribution tends to be long-tailed in real-life corpora. According to Wu et al. (2023), this tends to bias topics into favouring words with high frequency, ending in some topics being very similar one to another. To tackle this phenomenon, the authors combined the topic model approach with an embedding clustering regularisation showing significant improvement in the results. For

a review of classical methods relying exclusively on the document term frequency matrix, the reader may refer to Vayansky, Kumar (2020), while the reader interested in neural topic models may refer to Zhao, Phung, Huynh, Jin, et al. (2021).

2.4 Statistical network analysis

This section aims to give a brief overview of the statistical network analysis field. In particular, as in topic modelling, the first developed methods were heuristic-based. The uncertainty measure appeared later on thanks to probabilistic modelling. Two core models to represent complex networks will be presented, namely the stochastic block model as well as the latent position cluster model. The introduction of deep neural networks in the field will be treated in the last section of this chapter.

Notations

In this manuscript, a network, or graph, is defined as a couple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is a set of N vertices and $\mathcal{E} = \{(i,j) : i,j \in \{1,\dots,N\}, i \rightsquigarrow j\}$ a set of M edges. The notation $i \rightsquigarrow j$ indicates that i is connected to j. The connections, or edges, are represented by the binary adjacency matrix $\mathbf{A} \in \mathcal{M}_{N\times N}(\{0,1\})$ such that i is connected to j, or $(i,j) \in \mathcal{E}$, if and only if $A_{ij} = 1$. The number of clusters is denoted Q.

2.4.1 From heuristic-based methods to probabilistic modelling

Statistical network analysis first started with random graph theory, initiated by Erdos, Rényi, et al. (1960). They studied probabilistic properties of graphs with binary connections and a unique probability for any connection to exist. However, real-life datasets do not show such regularity. Therefore, more complex and realistic graph structures have been considered. Here, a structure designates a partition of the nodes such that nodes in a cluster present a homogeneous connectivity pattern. For example, a community is a group of nodes highly connected one to another but with few connections to the rest of the graph. If the graph is only composed of communities, reordering the adjacency matrix by group would output a block diagonal matrix. Another direction emerged with Fienberg, Wasserman (1981) who first introduced a probabilistic model assuming that the probability for two nodes to be connected only depends on the group they belong to and applied it to Sampson's monastery dataset (Sampson, 1969). Introducing a latent representation of the nodes then became popular. For instance, the latent position model (LPM, Hoff, Raftery, Handcock (2002)) obtains an informative representation of the network in a Euclidean space by assuming that each node can be modelled by a Gaussian variable in a low dimensional space. In Handcock, Raftery, Tantrum (2007), the authors proposed an extension, the latent position cluster model, that we shall detail in the next section.

2.4.2 The latent position cluster model

LPM was extended to the latent position cluster model (LPCM Handcock, Raftery, Tantrum, 2007) to incorporate clustering in the generative model. The authors assumed that the latent node positions are sampled from a mixture of normal distributions. This extension incorporates the clustering within the model and therefore combines the clustering task with the network representation.

A cluster membership vector C_i is associated with each node $i \in \{1, \dots, N\}$ such that:

$$C_i \overset{\text{i.i.d}}{\sim} \mathcal{M}_Q(1, \gamma), \tag{2.24}$$

with $\gamma \in \Delta_Q$ and $C_i \in \{0,1\}^Q$ being one hot encoded so that $C_{iq} = 1$ if node i belongs to cluster q and $C_{iq} = 0$ otherwise. Thus, denoting $\mathbf{C} = (C_1, \dots, C_N)^T \in \mathcal{M}_{N\times Q}(\{0,1\})$ the cluster membership matrix, we have:

$$p(\mathbf{C} \mid \gamma) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \gamma_q^{C_{iq}}. \tag{2.25}$$

Moreover, given its cluster membership, the node i is assumed to be represented by a Gaussian vector Z_i in a p-dimensional latent space,

$$Z_i \mid C_{iq} = 1 \sim \mathcal{N}\left(\mu_q, \sigma_q^2 \mathbf{I}_p\right). \tag{2.26}$$

Eventually, the connection between two nodes is assumed to depend on the closeness of the node representations in the latent space. Therefore, denoting $\eta_{ij} := \kappa - \omega_0^\top x_{ij} - \omega_1 \|Z_i - Z_j\|$, where x_{ij} is an edge vector feature, the probability for node i to be connected to node j is:

$$P(A_{ij} = 1 \mid Z_i, Z_j, \kappa, \omega_0, \omega_1) = \frac{1}{1 + e^{-\eta_{ij}}}, \tag{2.27}$$

where a logistic function is used as a link function. For the sake of brevity, we will denote $p_{ij} = (1 + e^{-\eta_{ij}})^{-1}$ and $\mathbf{Z} = (Z_1, \dots, Z_N)$. Finally, the joint distribution of the adjacency matrix, the latent node vectors, as well as the cluster memberships can be factorised as follows:

$$p(\mathbf{A}, \mathbf{Z}, \mathbf{C} \mid \kappa, \omega_0, \omega_1, \boldsymbol{\mu}, \boldsymbol{\sigma}, \gamma) = p(\mathbf{A} \mid \mathbf{Z}, \kappa) p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(\mathbf{C} \mid \gamma).$$

where $\boldsymbol{\mu} = (\mu_q)_q$ and $\boldsymbol{\sigma} = (\sigma_q)_q$.

Generative model 3: Generative model of the latent position cluster model

Parameters: $\kappa, \mu, \sigma, \gamma$, ω_0, ω_1
for *each node* n **do**
  Draw a cluster membership $C_i \sim \mathcal{M}_Q(1, \gamma)$;
  Draw a node embedding $Z_i \mid C_{iq} = 1 \sim \mathcal{N}_p\left(\mu_q, \sigma_q^2 \mathbf{I}_p\right)$;
end
for *each pair of nodes* $i \neq j$ **do**
  Compute $p_{ij} = (1 + e^{-\eta_{ij}})^{-1}$ where $\eta_{ij} = \kappa - \omega_0^\top x_{ij} - \omega_1 \|Z_i - Z_j\|$;
  Draw an edge $A_{ij} \mid Z_i, Z_j, \kappa \sim \mathcal{B}(p_{ij})$;
end

Inference

In the original paper, the authors proposed two different inference strategies. The first consists of a two-stage MLE estimation. First, the MLE of the LPM parameters are computed. Then, considering the LPM parameters fixed, the MLE of the cluster memberships parameters can be derived. However, as raised by the authors, this method does not allow the position of the nodes to take into account the cluster memberships. Therefore, the authors proposed a Bayesian inference strategy, combining a Gibbs sampler with Metropolis-Hastings steps giving better results at the expense of a computationally intensive algorithm.

LPCM gives representations of the network in a Euclidean space, but can only analyse networks with communities, that is groups where the nodes are highly connected with nodes in the same group but poorly connected to the rest of the graph. Block models are able to model any connectivity pattern between groups, making it a more flexible framework for complex networks as we shall see.

2.4.3 The stochastic block model

The stochastic block model (SBM) is another core model for statistical network analysis that relies on latent variables denoting node cluster memberships. As for the LPCM, the node cluster membership vectors C_i are modelled by a multinomial distribution,

$$p(\mathbf{C} \mid \gamma) = \prod_{i=1}^{C} \prod_{q=1}^{Q} \gamma_q^{C_{iq}}. \tag{2.28}$$

The notations are the same as in Section 2.4.2.

Besides, given the cluster memberships of the nodes, the connections are assumed to be independent. In particular, if node i is in cluster r and node j in cluster q, the nodes are connected with probability π_{qr}. Thus, given $\mathbf{C}$ and the probability matrix $\mathbf{\Pi} = (\pi_{qr})$, the probability of the node connections is

$$p(\mathbf{A} \mid \mathbf{C}, \mathbf{\Pi}) = \prod_{i \neq j}^{N} \prod_{q,r}^{Q} \left(\pi_{qr}^{A_{ij}} (1 - \pi_{qr})^{(1 - A_{ij})} \right)^{C_{iq} C_{jr}}. \tag{2.29}$$

Eventually, the joint-probability of the adjacency matrix $\mathbf{A}$, and the cluster memberships vector $\mathbf{C}$, is obtained by multiplying Equations (2.28) and (2.29),

$$p(\mathbf{A}, \mathbf{C} \mid \mathbf{\Pi}, \gamma) = p(\mathbf{A} \mid \mathbf{C}, \mathbf{\Pi}) p(\mathbf{C} \mid \gamma). \tag{2.30}$$

Generative model 4: Generative model of the stochastic block model

Parameters: $\gamma, \mathbf{\Pi}$
for *each node* i **do**
  Draw a cluster membership $C_i \sim \mathcal{M}_Q(1, \gamma)$;
end
for *each pair of nodes* $i \neq j$ **do**
  Draw an edge $A_{ij} \mid C_{iq}C_{jr} = 1, \pi_{qr} \sim \mathcal{B}(\pi_{qr})$;
end

Inference

To estimate the parameters, the classical MLE would be obtained by maximising the following log-likelihood:

$$\log p(\mathbf{A} \mid \gamma, \mathbf{\Pi}) = \log \left(\sum_{\mathbf{C}} p(\mathbf{A} \mid \mathbf{C}, \mathbf{\Pi}) p(\mathbf{C} \mid \gamma) \right).$$

Unfortunately, this requires computing Q^N terms, which is exponential in the number of nodes and quickly becomes intractable for networks with hundreds of nodes. Indeed, Snijders, Nowicki (1997) mentioned that the E-M algorithm can only be used for small graphs because of this limitation. Therefore, they proposed to use a Bayesian inference strategy relying on a Gibbs sampler to infer the posterior distribution of the cluster memberships and the probability matrix. In Daudin, Picard, Robin (2008), the authors proposed another strategy, based on a mean-field variational inference strategy, allowing to analyse larger graphs, and with simple updates. Since we will use this strategy, let us give some insights about the computation. First, since $\mathbf{C}$ are discrete latent vectors, the mean-field variational distribution can be written, just with the full factorisation hypothesis, as

$$r_\tau(\mathbf{C}) = \prod_{i=1}^{N} \mathcal{M}_Q(C_i; 1, \tau_i) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \tau_{iq}^{C_{iq}}.$$

Hence, the ELBO can be written

$$\mathscr{L}(\gamma, \mathbf{\Pi}; \boldsymbol{\tau}) = \mathbb{E}_{r_\tau} \left[\log p(\mathbf{A}, \mathbf{C} \mid \gamma, \mathbf{\Pi}) \right] - \mathbb{E}_{r_\tau} \left[\log r_\tau(\mathbf{C}) \right]. \tag{2.31}$$

The following propositions give the updates of the parameters.

Proposition 1. *Denoting* $b(\pi_{qr}, A_{ij}) = \pi_{qr}^{A_{ij}}(1 - \pi_{qr})^{1 - A_{ij}}$, *the updates of the parameters of the mean-field variational inference, for the SBM are given by*

$$\begin{aligned}
\tau_{iq} &\propto \gamma_q \prod_{j \neq i} \prod_{r=1}^{Q} \Big(b(\pi_{qr}, A_{ij}) b(\pi_{rq}, A_{ji}) \Big)^{\tau_{jr}}, \\
\gamma_q &= \frac{1}{N} \sum_{i=1} \tau_{iq}, \\
\pi_{qr} &= \frac{\sum_{i \neq j} \tau_{iq} \tau_{jr} A_{ij}}{\sum_{i \neq j} \tau_{iq} \tau_{jr}}.
\end{aligned}$$

Proof. The proof is given in Appendix 7.1.1 □

This was extended to the Bayesian setting in Latouche, Birmele, Ambroise (2012), where the authors proposed a variational Bayes expectation maximisation (VBEM). The classification can either be deduced from the latent variable distribution or be incorporated in the optimisation strategy with a hard clustering, for instance using the classification variational expectation maximisation (CVEM) algorithm (Bouveyron, Latouche, Zreik, 2018). The choice of the number of cluster Q can either be done through a model selection criterion (Daudin, Picard, Robin, 2008; Latouche, Birmele, Ambroise, 2012), through a greedy search (Côme, Latouche, 2015) or through a non-parametric scheme (Kemp et al., 2006). Many extensions have been developed to incorporate valued edges, as in Mariadassou, Robin, Vacher (2010), as well as categorical edges in Jernite et al. (2014) or to add prior information in Zanghi, Volant, Ambroise (2010). Some developments also focused on looking for overlapping clusters (Airoldi et al., 2008; Latouche, Birmelé, Ambroise, 2011) as well as dynamic networks (Matias, Miele, 2017; Zreik, Latouche, Bouveyron, 2017; Corneli, Latouche, Rossi, 2016). For more insights about SBM developments, see Lee, Wilkinson (2019). For reviews on statistical network modelling, we also relate to Snijders (2011) and Matias, Robin (2014).

Unification of positional and block modelling We end this section with an approach unifying the two previous types of modelling, namely block modelling and positional learning. This unification, proposed in Hoff (2007), uses a new framework to describe the probability of connection between nodes i and j. Indeed, the author proposed to create a similarity matrix, using node representations Z_i and a similarity function $\alpha(Z_i, Z_j)$ and to model the probability of connection using a probit function based on the output of the similarity function. Assuming that $Z_i \in \mathbb{R}^d$ and $\alpha(Z_i, Z_j) = -\|Z_i - Z_j\|^2$ would provide the same distribution as LPM while considering u_i as a one hot encoded cluster membership and taking $\alpha(Z_i, Z_j) = m_{Z_i, Z_j}$ would provide the same modelling as the stochastic block model. Using this similarity-based formulation of the model, the author introduced a more general similarity function, encapsulating both previous examples, using a continuous node latent representation $Z_i \in \mathbb{R}^d$ and $\alpha(Z_i, Z_j) = Z_i^\top \Lambda Z_j$ where Λ is a diagonal matrix. However, this method relies on a Monte-Carlo Markov Chain algorithm, preventing it from scaling to large networks. In addition, the continuous latent representation prevents from clustering the nodes of the network with the flexibility of traditional block modelling approaches.

Daudin, Pierre, Vacher (2010) also proposed a way to combine continuous node representations with a block modelling approach. Instead of assuming that the mixture is at the network level and that each node belongs to a single cluster, they proposed a model based on a mixture at the node level, such that each node is a mixture of extremal nodes, and the probability of connection between node i and node j is given by $Z_i^\top \mathbf{\Pi} Z_j$ where the $Z_i \in \Delta_Q$ and $\mathbf{\Pi} \in \mathcal{M}_{Q \times Q}([0, 1])$.

2.5 Joint analysis of texts and networks

While both topic modelling and statistical network analysis gave rise to many publications over the last 20 years, only a few works have combined the two approaches. The community-user topic model (CUT, Zhou et al. (2006)) added a latent variable to the author-topic model (AT, Rosen-Zvi et al. (2004) to represent the communities either as a set of co-authors or as a set of topics. Thereafter, the community-author-recipient-topic model (CART, Pathak et al. (2008)) used communities both at the document generation level and at the author and recipient generation level which corresponds to the network generation. However, the high number of parameters combined with the inference based on a Gibbs sampler does not allow to scale those models to large datasets. The topic-link LDA, presented in Liu, Niculescu-Mizil, Gryc (2009), also offers a joint analysis of texts and links in a unified framework by conditioning the generation of a link on both the topics within the documents and the community of authors. The inference relies on a variational E-M algorithm to scale the approach to large datasets. However, this method only deals with undirected networks. Finally, the topic-user-community model (TUCM) was introduced in Sachan et al. (2012) and was able to discover topic-meaningful communities. The main feature of this model is its capacity to incorporate different types of interactions, well-suited for social network applications. The inference relied on Gibbs sampling approach which can be limiting when dealing with large datasets. More recently, the stochastic topic block model (STBM) presented in Bouveyron, Latouche, Zreik (2018) was the first model to handle the simultaneous clustering of nodes and edges while keeping the inference tractable to large datasets thanks to a variational classification EM-based inference. This model was extended in Bergé et al. (2019) for the simultaneous clustering of bipartite networks with textual edges. It was also adapted for dynamic networks in Corneli, Bouveyron, et al. (2019).

2.6 Deep neural networks on graphs

To end this chapter, let us present a deep probabilistic model adapted to graph-structured data. We focus on the most central architecture of our work, which leads to the convolutional graph neural networks. This line of work relies on the symmetric normalised Laplacian matrix to represent the graph, given by $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ with $\mathbf{D}$ the diagonal matrix of node degrees, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. Since $\mathbf{L}$ is a real symmetric matrix (see for instance Von Luxburg (2007) for more properties on Laplacian matrices), the matrix can be decomposed in the orthogonal group such that there exists $\mathbf{U} \in \mathcal{M}_{N\times N}(\mathbb{R})$, comprising the eigenvectors as columns, ordered by eigenvalue, such that

$$\mathbf{L} = \mathbf{U}\Delta\mathbf{U}^\top,$$

where $\Delta = \mathrm{diag}(\lambda_0, \ldots, \lambda_{N-1})$ is the diagonal matrix of the ordered eigenvalues.

2.6.1 Graph signal processing

Let us consider a signal $x \in \mathbb{R}^N$ on the nodes of the graph. As in signal processing, it is possible to define the *graph Fourier transform*,

$$\mathcal{F}(x) = \mathbf{U}^\top x.$$

This transformation maps a signal belonging to the graph domain to the Fourier domain. Since $\mathbf{U}$ is orthogonal, that is $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_N$, the *inverse graph Fourier transform* is naturally given by $\mathcal{F}^{-1}(\hat{x}) = \mathbf{U}\hat{x}$ for $\hat{x} \in \mathbb{R}^N$. As a result, it is possible to define a *graph convolution of the signal x with a filter $f \in \mathbb{R}^N$* as a product in the Fourier domain

$$\begin{aligned} x \star_G f &= \mathcal{F}^{-1}(\mathcal{F}(x) \odot \mathcal{F}(f)) \\ &= \mathbf{U}(\mathbf{U}^\top x \odot \mathbf{U}^\top f). \end{aligned} \tag{2.32}$$

Let us remark that, denoting $\tilde{f} = \mathrm{diag}(\mathbf{U}^\top f)$, Equation (2.32) can be rewritten as

$$x \star_G \tilde{f} = \mathbf{U}\tilde{f}\mathbf{U}^\top x.$$

In the rest of this manuscript, this equation will be used as the definition of the graph convolution of signal x with filter f.

2.6.2 From spectral-based convolutional graph neural networks to the variational graph autoencoder

All the spectral-based convolutional graph neural networks follow this definition. However, they distinguish one from another by their respective filter $\tilde{f}$. For instance, Bruna et al. (2013) considered the filter as a set of learnable parameters and composed their neural network architecture as the aggregation of multiple channels and hidden layers to gather local information iteratively. The ChebNet architecture, proposed in Defferrard, Bresson, Vandergheynst (2016), relies on some approximations concerning the filtering, allowing an important gain regarding the complexity (from $O(N^3)$ to $O(M)$). In particular, the authors proposed to approximate the filter $\tilde{f}$ with Chebyshev polynomial $f_\theta = \sum_{l=0}^{L} \theta_l T_l(\tilde{\Delta})$ where $\tilde{\Delta} = (2\lambda_{max}^{-1}\Delta) - \mathbf{I}_N \in \mathcal{M}_{N\times N}([-1,1])$ and the Chebyshev polynomial are defined as $T_l(x) = 2xT_{l-1}(x) - T_{l-2}(x)$, $T_0(x) = 1$ and $T_1(x) = x$. This gives the following form

$$\begin{aligned} x \star_G f_\theta &= \mathbf{U}\left(\sum_{l=0}^{L} \theta_l T_l(\tilde{\Delta})\right)\mathbf{U}^\top x \\ &= \sum_{l=0}^{L} \theta_l \mathbf{U} T_l(\tilde{\Delta})\mathbf{U}^\top x \\ &= \sum_{l=0}^{L} \theta_l T_l(\tilde{L})x, \end{aligned} \tag{2.33}$$

where $\tilde{L} = 2\lambda_{max}L - \mathbf{I}_N$. The equality $T_l(\tilde{L}) = \mathbf{U}T_l(\tilde{\Delta})\mathbf{U}^\top$ for any $l \in \{0, \dots, L\}$ can be proofed by induction. Eventually, Kipf, Welling (2017) relied on a first-order approximation of the ChebNet filter to obtain an architecture easier to implement. Indeed, assuming that $L = 1$ and $\lambda_{max} = 2$ (expecting the neural network to adapt to this change of scale), Equation 2.33 can be written as

$$x \star_G f_\theta = \theta_0 x - \theta_1 \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}x.$$

Assuming that $\theta_0 = -\theta_1$ (since limiting the number of parameters might address overfitting) leads to the following definition of the convolution for GCN

$$x \star_G f_\theta = \theta(\mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})x. \tag{2.34}$$

However, $\mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ has eigenvalues in the range $[0, 2]$. Therefore, repeating this operation might lead to numerical instabilities. Therefore, for numerical stability reasons, $\mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is replaced with $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ and $\tilde{\mathbf{D}}$ is a diagonal matrix, where $\tilde{\mathbf{D}}_{ii} = \sum_{j=1}^{N} \tilde{\mathbf{A}}_{ij}$. This gives the following new definition of a graph convolution

$$x \star_G f_\theta = \theta\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}x. \tag{2.35}$$

Until now, we assumed that the input signal was 1-dimensional for each node, and restricted to the case of 1 filter being used. Kipf, Welling (2017) extended the definition of graph convolution to C-dimensional signals $\mathbf{X} \in \mathcal{M}_{N\times C}(\mathbb{R})$, and F filters $\Theta \in \mathcal{M}_{N\times F}(\mathbb{R})$ by considering

$$\mathbf{X} \star_G \mathbf{F}_\Theta = \hat{\mathbf{A}}\mathbf{X}\Theta, \tag{2.36}$$

This multi-dimensional convolution is at the core of the graph convolutional network, which builds upon Equation (2.36) to construct layers on top of each other. In particular, Kipf, Welling (2017) proposed the following architecture, referred to as graph convolutional network (GCN) in the rest of this manuscript,

$$\mathbf{Z} = \operatorname{softmax}\left(\hat{\mathbf{A}}\,\mathrm{ReLU}\left(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right), \tag{2.37}$$

where $\mathbf{W}^{(0)} \in \mathcal{M}_{C\times H_0}$ and $\mathbf{W}^{(1)}(\mathbb{R}) \in \mathcal{M}_{H_0\times H_1}(\mathbb{R})$.

This graph convolutional network has been used as an encoder in a variational autoencoder, named the variational graph autoencoder (VGAE) (Kipf, Welling, 2016). Many extensions are built upon the variational graph autoencoder. In Pan et al. (2018), the authors regularise VGAE by using an adversarial inference strategy to improve the results. Unfortunately, those models rely on external methods, such as K-means, on the posterior node representations, to achieve node clustering. As an answer to this limit, Mehta, Duke, Rai (2019) presented an extension of the overlapping stochastic block model (Latouche, Birmelé, Ambroise, 2011). They proposed to encode the node embeddings with a neural network. Lately, Liang et al. (2022) combined the two and introduced the deep latent position model (Deep-LPM). Indeed, the model relies on VGAE and assumes that the latent representations are distributed according to a mixture of Gaussians, depending on the node cluster memberships. Their results advocate the incorporation of the clustering into the latent representations.

Chapter 3

The Embedded Topics for the Stochastic Block Model

. . .

Many real-life interactions induce the exchange of texts, as in co-authorship networks, social networks or emails for instance. Since the storage capacity keeps increasing, networks with textual data on the edges become even more frequent. To make such networks, called communication networks, intelligible to humans, it is of great interest to gather information about the texts exchanged between the nodes and to summarise the connectivity structure. While those two questions have been studied independently, the work we propose aims at bridging the gap between the two by modelling the joint distribution of texts and edges. To the best of our knowledge, the interest in making the two disciplines of topic modelling, when texts are present on the edges, and model-based graph clustering meets is recent and the methods that have been proposed only rely on the frequency of words within the documents without incorporating semantic meaning. In this paper, we propose to take advantage of pre-trained word embeddings in the topic-model as presented in Dieng, Ruiz, Blei (2020) in order to incorporate semantic meaning of the words and to obtain topic-meaningful clusters. An introduction to the data analysed in this chapter as well as our contribution are exhibited in Section 3.1. The embedded topics for the stochastic block model (ETSBM) is presented in Section 3.2. The inference and the model selection are presented in Section 3.3. Eventually, the model is evaluated against state-of-the-art algorithms on synthetic data and we present results for a real word example built from tweets during the last French presidential election in Sections 3.4 and 3.5, respectively. Section 3.6 presents some concluding remarks and further work.

3.1 Introduction and contribution

3.1.1 Introduction

Many real-life interactions induce the exchange of texts, as in co-authorship networks, social networks or emails for instance. Since the storage capacity keeps increasing, networks with textual data on the edges become even more frequent. To make such networks, called communication networks, intelligible to humans, it is of great interest to gather information about the texts exchanged between the nodes and to summarise the connectivity structure. While those two questions have been studied independently, the work we propose aims at bridging the gap between the two by modelling the joint distribution of texts and edges. To the best of our knowledge, the interest in making the two disciplines of topic modelling, when texts are present on the edges, and model-based graph clustering meets is recent and the methods that have been proposed only rely on the frequency of words within the documents without incorporating semantic meaning. In this chapter, we propose to take advantage of pre-trained word embeddings in the topic-model as presented in Dieng, Ruiz, Blei (2020) in order to incorporate semantic meaning of the words and to obtain topic-meaningful clusters.

3.1.2 Our contribution

In this chapter, we propose a new methodology called the embedded topics in the stochastic block model (ETSBM), to look for node partitions incorporating the connectivity patterns as well as the topics exchanged between the nodes. We will reserve the term *community* to groups of nodes that are densely connected together but poorly connected to the rest of the graph. In the block model literature, the term *cluster* denotes a group of nodes that share a similar connectivity pattern which goes beyond the concept of community. For instance, contrary to communities, a star pattern is defined by two clusters with low intra-connection and large inter-connection probabilities (Latouche, Birmele, Ambroise, 2012). Such a pattern is particularly common in social networks. This type of cluster cannot be retrieved by community detection methods. In this chapter, we will also assume that the nodes of the same cluster share a similar use of topic proportions. To find clusters complying with this definition, **(i)** we propose a generative model assuming that each node belongs to a cluster and that the probability of connection between two nodes, as well as the topic proportions of a document, only depend on the clusters of the corresponding nodes. Figure 3.1 illustrates the necessity to combine graph clustering and topic modelling in order to distinguish all four clusters and to obtain more meaningful topics for each cluster. **(ii)** To model the topics exchanged between the nodes, the documents are encoded with a deep neural network to benefit from their flexibility. **(iii)** The decoder is made of word and topic embeddings, as in Dieng, Ruiz, Blei (2020). **(iv)** In this chapter, the documents are aggregated at the cluster level, into Q^2 meta-documents with Q the number of clusters. The meta-documents are obtained by weighting each document with the cluster membership probabilities of the corresponding nodes. In particular, our inference strategy is able to directly optimise the construction of the meta-documents through the inference procedure.

3.2 The ETSBM Model

3.2.1 Notations

In this chapter, we focus on data represented by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, such that $\mathcal{V} = \{1, \dots, M\}$ denotes the set of nodes and $\mathcal{E} := \{(i,j) : i,j \in \{1,\dots,M\}, i \rightsquigarrow j\}$ the set of edges, where $i \rightsquigarrow j$ indicates that i is connected to j. The connections, or edges, are represented by a binary matrix $\mathbf{A} \in \mathcal{M}_{M\times M}(\{0,1\})$ such that i is connected to j, or $(i,j) \in \mathcal{E}$, if and only if $A_{ij} = 1$. In the applications we consider, this implies that node i sent textual information to j such as one or a series of emails for instance. These texts are denoted $W_{ij} = \{W_{ij}^1, \dots, W_{ij}^{D_{ij}}\}$ with D_{ij} the number of documents sent from i to j and are gathered in the collection $\mathbf{W} = \{W_{ij}, (i,j) \in \mathcal{E}\}$. Each document d in W_{ij} is a collection of words of size N_{ij}^d, i.e $W_{ij}^d = \{w_{ij}^{d1}, \dots, w_{ij}^{dN_{ij}^d}\}$. The size of the vocabulary is denoted V and the words are identified by their index in the vocabulary: each word w is in $\{1, \dots, V\}$. Finally, only graphs without self-loops are considered in this chapter, therefore

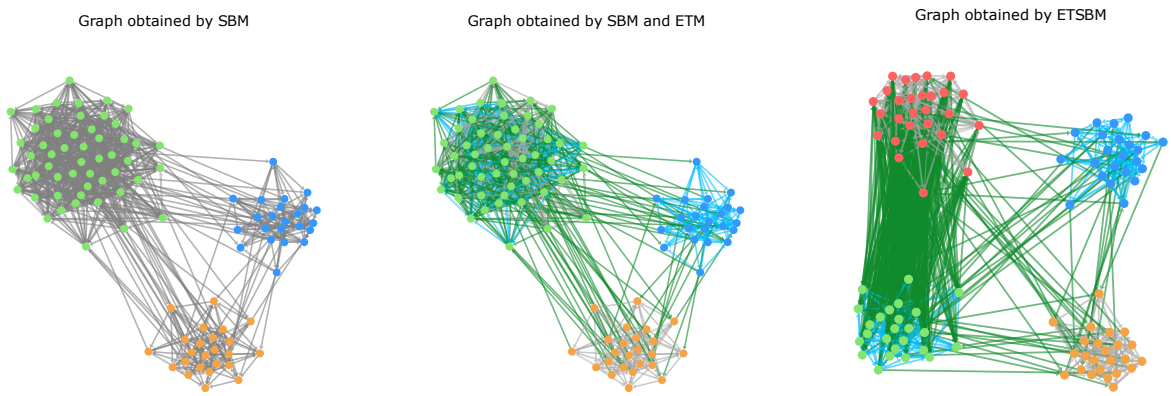


Figure 3.1: Comparison of results on a simulated network with the use of SBM on the left, SBM and ETM in the middle and ETSBM on the right. The colours of the nodes indicate the cluster of the vertices. The colours of the edges indicate the most-used topic in the corresponding documents. Note that SBM alone does not provide edge information. Thus, the left network only has a single-edge colour. On the left-hand side, SBM clustering results uncover 3 clusters. Again, in the middle, SBM is used and uncovers 3 clusters of nodes. ETM edge information is added to the network through the 3 edge colours green, grey and blue. On the right-hand side, ETSBM clustering results uncover 4 clusters. The cluster coloured in green, in the middle of the figure, is split into two clusters on the right-hand side, the green one and the red one, each discussing a different topic, the blue and grey topic respectively. The clusters of nodes of the figure on the right-hand side are coherent both in terms of topology and topics of discussion contrary to the figure in the middle.

$A_{ii} = 0$ for all $i \in \mathcal{V}$. Notice that all the present work can be extended to undirected networks using $W_{ij} = W_{ji}$ for all pairs (i, j) such that $A_{ij} = A_{ji} = 1$. The directed case is more adequate for messages sent from i to j while the undirected case is better suited for co-authorship networks for instance.

The notation $\mathcal{M}_{d\times p}(\mathbb{F})$ will be used to denote the space of $d \times p$ matrices with coefficients in $\mathbb{F}$ while the notation $\mathcal{M}_d(N, \omega)$ will be used to denote the multinomial distribution with parameters $N \in \mathbb{N}$ and $\omega \in \Delta_d$ where

$$\Delta_d \ =: \ \left\{ x \in \mathbb{R}^d : \forall i \in \{1, \dots, d\}, x_i \geq 0, \sum_{i=1}^{d} x_i = 1 \right\}.$$

3.2.2 Modelling the interactions

In this chapter, we assume that each node belongs to a single cluster. Moreover, we assume that the connexion probability between two nodes only depends on the cluster memberships. Indeed, let C_i denotes the cluster membership of node i for any $i \in \{1, \dots, M\}$. All C_i are assumed to follow a multinomial distribution and to be independent and identically distributed (*i.i.d*), given the cluster proportions $\gamma \in \Delta_Q$, lying in the simplex of dimension Q,

$$C_i \mid \gamma \overset{\text{i.i.d}}{\sim} \mathcal{M}_Q(1, \gamma).$$

Thus, each node i is associated with cluster q with probability γ_q. Then, we define the cluster membership matrix C by stacking the node cluster membership vectors $(C_i)_i$ together such that $\mathbf{C} = (C_1 \cdots C_M)^\top \in \mathcal{M}_{M\times Q}(\{0, 1\})$. The probability of $\mathbf{C}$ is given by

$$p(\mathbf{C} \mid \gamma) = \prod_{i=1}^{M} \prod_{q=1}^{Q} \gamma_q^{C_{iq}}. \tag{3.1}$$

Besides, the connections between nodes are supposed to be independent given their cluster memberships. Moreover, if nodes i and j are respectively in clusters q and r, an edge is assumed to be present with probability π_{qr},

$$A_{ij} \mid C_{iq}C_{jr} = 1, \pi_{qr} \overset{\text{i.i.d}}{\sim} \mathcal{B}(\pi_{qr}), \tag{3.2}$$

where $\mathcal{B}(\mu)$ denotes the Bernoulli distribution with probability μ. Thus, given the cluster memberships of the nodes $\mathbf{C}$ and the probability matrix $\mathbf{\Pi} = (\pi_{qr})_{qr} \in \mathcal{M}_{Q\times Q}(bbr)$, the probability of all node connections is given by

$$p(\mathbf{A} \mid \mathbf{C}, \mathbf{\Pi}) = \prod_{i\neq j}^{M} \prod_{q,r}^{Q} \left(\pi_{qr}^{A_{ij}} (1 - \pi_{qr})^{(1-A_{ij})} \right)^{C_{iq}C_{jr}}. \tag{3.3}$$

Eventually, the joint-probability of the adjacency matrix A, and the cluster memberships vector C, is obtained by multiplying Equations (3.1) and (3.3),

$$p(\mathbf{A}, \mathbf{C} \mid \mathbf{\Pi}, \gamma) = p(\mathbf{A} \mid \mathbf{C}, \mathbf{\Pi}) p(\mathbf{C} \mid \gamma). \tag{3.4}$$

Combining Equations (3.1), (3.3), and (3.4), we retrieve the SBM distribution (Daudin, Picard, Robin, 2008).

3.2.3 Modelling the texts

Our approach extends ETM to capture information from groups of texts. Essentially, texts are assumed to be generated according to a mixture of topics with latent topic vectors only depending on node clusters. More precisely, a text sent from node i in cluster q to node j in cluster r is assumed to have a logistic-normal topic proportion vector $\theta_{qr} = (\theta_{qr1}, \dots, \theta_{qrK})^\top \in \Delta_K$, with the number of topics K fixed beforehand. It is obtained by applying the softmax function to a Gaussian random vector δ_{qr},

$$\begin{aligned} Y_{qr} &\sim \mathcal{N}(0_K, \mathbf{I}_K), \\ \theta_{qr} &= \mathrm{softmax}(Y_{qr}), \end{aligned}$$

where $\mathrm{softmax}(x) = \left(\sum_{k=1}^K e^{x_k}\right)^{-1} (e^{x_1}, \dots, e^{x_K})^\top$.

In the rest of this chapter, the notation $\boldsymbol{\theta} = (\theta_{qr})_{1 \leq q,r \leq Q}$ is used to refer to the topic proportions while $\mathbf{Y} = (Y_{qr})_{1 \leq q,r \leq Q}$ will refer to the sampling of the random variable. If two nodes i and j are connected and if they are respectively in cluster q and r, the words in document W_{ij} are assumed to be *i.i.d.* Indeed, the n-th word of the d-th documents is assumed to be distributed according to a mixture of topics conditionally on the node clusters,

$$W_{ij}^{dn} \mid C_{iq} C_{jr} A_{ij} = 1, \theta_{qr}, \boldsymbol{\alpha}, \boldsymbol{\rho} \sim \mathcal{M}_V(1, \theta_{qr}^\top \boldsymbol{\beta}), \tag{3.5}$$

where the matrix $\boldsymbol{\beta} = (\beta_1 \cdots \beta_K)^\top \in \mathcal{M}_{K \times V}(\mathbb{R})$ corresponds to the distribution over the vocabulary for each topic such that $\beta_k = \mathrm{softmax}\left(\boldsymbol{\rho}^\top \alpha_k\right)$ for any $k \in \{1, \dots, K\}$. The matrix $\boldsymbol{\rho} \in \mathcal{M}_{L \times V}(\mathbb{R})$ corresponds to the matrix of the vocabulary embedded into an L-dimensional vector space, and $\boldsymbol{\alpha} = (\alpha_1 \cdots \alpha_K) \in \mathcal{M}_{L \times K}(\mathbb{R})$ the matrix of topics represented into the same vector space.

Therefore, the probability of texts can be computed as follows:

$$\begin{aligned} p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\rho}) &= \prod_{i \neq j}^{M} \prod_{d=1}^{D_{ij}} p(W_{ij} \mid C_i, C_j, A_{ij} = 1, \boldsymbol{\theta}, \boldsymbol{\alpha}, \boldsymbol{\rho}) \\ &= \prod_{i \neq j}^{M} \prod_{d=1}^{D_{ij}} \prod_{n=1}^{N_{ij}^d} \prod_{q,r}^{Q} \prod_{v=1}^{V} \Big(\sum_{k=1}^{K} \theta_{qrk} \beta_{kv} \Big)^{W_{ij}^{dnv} A_{ij} C_{iq} C_{jr}} \\ &= \prod_{q,r}^{Q} \prod_{v=1}^{V} \Big(\sum_{k=1}^{K} \theta_{qrk} \beta_{kv} \Big)^{W_{qr}^v}. \end{aligned} \tag{3.6}$$

The number of time the word v of the dictionary is used in texts sent from cluster q to cluster r is denoted $W_{qr}^v = \sum_{i \neq j}^{M} \sum_{d=1}^{D_{ij}} \sum_{n=1}^{N_{ij}^d} W_{ij}^{dnv} A_{ij} C_{iq} C_{jr}$. Here, $W_{qr} = (W_{qr}^1, \dots, W_{qr}^V)^\top \in \mathbb{N}^V$ shall be designated as meta-document (q, r). Moreover, we shall use the bag of words notations such that for any connected pair of nodes $(i, j) \in \mathcal{E}$, $W_{ij} = (W_{ij}^1, \dots, W_{ij}^V)^\top \in \mathbb{N}^V$ with for any $v \in \{1, \dots, V\}$, W_{ij}^v represents the total count of word v for all documents sent from i to j. The model is represented in Figure 3.2.

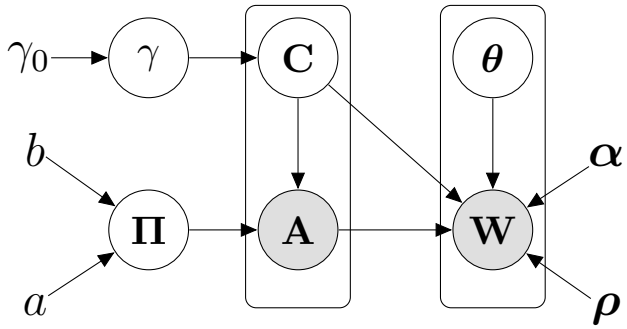


Figure 3.2: Graphical representation of the model.

3.2.4 Distribution of the model and links with SBM and ETM.

Given a cluster configuration Y, the joint probability of the model is obtained using Equations (3.3) and (3.6)

$$p(\mathbf{A}, \mathbf{W} \mid \mathbf{C}, \boldsymbol{\alpha}, \boldsymbol{\rho}) = p(\mathbf{W} \mid \mathbf{C}, \mathbf{A}, \boldsymbol{\alpha}, \boldsymbol{\rho}) p(\mathbf{A} \mid \mathbf{C}, \mathbf{\Pi}). \tag{3.7}$$

At this point, we emphasise that meta-documents between pairs of clusters of nodes are constructed using the cluster memberships $\mathbf{C}$ and the node connections $\mathbf{A}$. Assuming that the cluster membership $\mathbf{C}$ is available as well as all the network information held by $\mathbf{\Pi}$ and γ, the model we propose would simply correspond to ETM applied on the meta-documents $(W_{qr})_{1 \leq q,r, \leq Q}$, computed with the available $\mathbf{C}$.

On the other hand, if no texts are exchanged between nodes or the texts are not available, the distribution would reduce to the second term of Equation 3.7. In that case, the conditional distribution of a standard SBM (Daudin, Picard, Robin, 2008) is recovered. It is also worth noticing that if a Dirichlet prior is assumed on the topic proportion instead of a logistic-normal, and no factorisation in an embedded latent space is considered, the model corresponds to STBM. By construction, ETSBM generalises SBM and ETM to incorporate both textual data and network information.

3.3 Inference

This section presents the Bayesian framework considered for inference. It also describes the variational-bayes E-M algorithm used to maximise the integrated joint likelihood.

3.3.1 Bayesian framework for the graph modelling part

First, a Dirichlet distribution is assumed as a prior distribution on the proportions γ of nodes in each cluster,

$$\gamma \sim \mathcal{D}ir_Q(\gamma_0). \tag{3.8}$$

where γ_0 is set to $(1, \ldots, 1) \in \mathbb{R}^Q$, which corresponds to a uniform prior on the simplex. Moreover, each coefficient of the probability matrix $\pi \in \mathcal{M}_{Q \times Q}(\mathbb{R})$, is assumed to be

sampled from a Beta distribution, such that for any pair $(q, r) \in \{1, \dots, Q\}^2$,

$$\pi_{qr} \overset{\text{i.i.d}}{\sim} \mathcal{B}eta(a, b).$$

In particular, a and b are set to 1. Thus, the Beta prior corresponds to a Uniform distribution between 0 and 1.

3.3.2 Variational inference

Eventually, the integrated joint log-likelihood is given by:

$$\log p(\mathbf{A}, \mathbf{W} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}) = \log \left(\sum_C \int_{\mathbf{Y}} \int_{\gamma} \int_{\mathbf{\Pi}} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{\Pi}, \gamma, \mathbf{Y} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}) d\mathbf{\Pi} d\mathbf{Y} d\gamma \right). \quad (3.9)$$

Unfortunately, this quantity is intractable since it requires computing it for the Q^M configurations of $\mathbf{C}$, which is naturally computationally too demanding. Moreover, the integral with respect to $\mathbf{Y}$ is not tractable either because of the $\mathrm{softmax}$ function. Thus, it cannot be optimised directly. However, it is possible to overcome this issue using a variational-bayes expectation-maximisation algorithm (VBEM) proposed in Attias (1999). This comes handy as it makes the inference scalable to large datasets.

The variational approach consists in splitting Equation (3.9) in two terms using a surrogate distribution on $\mathbf{C}, \mathbf{\Pi}, \gamma$ and $\mathbf{Y}$, denoted $R(, \mathbf{\Pi}, \gamma, \mathbf{Y})$.

Proposition 2. *Denoting $R(\cdot)$, a distribution on $\mathbf{C}, \mathbf{\Pi}, \gamma$ and $\mathbf{Y}$, the integrated joint log-likelihood can be decomposed as follow:*

$$\log p(\mathbf{A}, \mathbf{W} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}) = \mathscr{L}(R(\cdot); \boldsymbol{\alpha}, \boldsymbol{\rho}) + \mathrm{KL}(R(\cdot) || p(\mathbf{C}, \mathbf{\Pi}, \gamma, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\rho})),$$

where

$$\mathscr{L}(R(\cdot); \boldsymbol{\alpha}, \boldsymbol{\rho}) = \sum_C \int_{\mathbf{\Pi}, \gamma, \mathbf{Y}} R(\mathbf{C}, \mathbf{\Pi}, \gamma, \mathbf{Y}) \log \frac{p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{\Pi}, \gamma, \mathbf{Y} \mid \boldsymbol{\alpha}, \boldsymbol{\rho})}{R(\mathbf{C}, \mathbf{\Pi}, \gamma, \mathbf{Y})} d\mathbf{\Pi} d\mathbf{Y} d\gamma.$$

Proof. The proof is provided in 7.2.1. □

To make $\mathscr{L}(R(\cdot); \boldsymbol{\alpha}, \boldsymbol{\rho})$ tractable, we use the following mean-field assumption:

$$R(\mathbf{C}, \mathbf{\Pi}, \gamma, \mathbf{Y}) = R(\mathbf{C}) R(\mathbf{\Pi}) R(\gamma) R(\mathbf{Y}). \quad (3.10)$$

Following the optimality results of Latouche, Birmele, Ambroise (2012), we impose the following variational distributions:

$$\begin{aligned} R(\mathbf{C}) &= \prod_{i=1}^{M} R(C_i) = \prod_{i=1}^{M} \mathcal{M}_Q(C_i; 1, \tau_i), \\ R(\mathbf{\Pi}) &= \prod_{q,r=1}^{Q} R(\pi_{qr}) = \prod_{q,r=1}^{Q} \mathcal{B}eta(\pi_{qr}; \tilde{\pi}_{qr1}, \tilde{\pi}_{qr2}), \\ R(\gamma) &= \mathcal{D}ir_Q(\gamma; \tilde{\gamma}). \end{aligned} \quad (3.11)$$

Each vector τ_i is of size Q and encodes the (approximate) posterior probabilities for node i to be in each cluster. Given $\boldsymbol{\tau} = (\tau_i)_i$, the set of posterior cluster membership probabilities, for any pair (q, r) the corresponding expected meta-document can be computed as

$$\tilde{W}_{qr} = \sum_{i \neq j} \tau_{iq} \tau_{jr} W_{ij}. \tag{3.12}$$

By construction, the v-th element of vector $\tilde{W}_{qr}$ is the expected pseudo count of word v for all documents sent from nodes in cluster q to nodes in cluster r. Finally, the variational distribution on latent topic proportions is assumed to be:

$$R(\mathbf{Y}) = \prod_{q,r=1}^{Q} R(Y_{qr}) = \prod_{q,r=1}^{Q} \mathcal{N}\big(Y_{qr}; \mu_{qr}(\boldsymbol{\tau}, \nu), \mathrm{diag}(\sigma^2_{qr}(\boldsymbol{\tau}, \nu))\big), \tag{3.13}$$

with $(\mu_{qr}(\boldsymbol{\tau}, \nu), \sigma_{qr}(\boldsymbol{\tau}, \nu))^\top = f(\tilde{W}^{norm}_{qr}(\boldsymbol{\tau}); \nu)$ the output of a parametric function, typically a (deep) neural network, with parameters denoted ν. Hereafter, the ETM encoder will be used as the function f parametrised by ν. The normalised expected meta-documents $\tilde{W}^{norm}_{qr}(\boldsymbol{\tau}) = \left(\sum_{v=1}^{V} \tilde{W}^{v}_{qr}(\boldsymbol{\tau})\right)^{-1} \tilde{W}_{qr}(\boldsymbol{\tau}) \in \mathbb{R}^V$ are then given to the encoder which outputs the mean and variance vectors $(\mu_{qr}(\boldsymbol{\tau}, \nu), \sigma_{qr}(\boldsymbol{\tau}, \nu))^\top$ of the posterior distribution. Our inference strategy is inspired by Dieng, Ruiz, Blei (2020) and finds its roots in the original work of Kingma, Welling (2014) for classical data. However, as we shall see, a critical property of our methodology is that the (approximate) posterior allocation probabilities $\boldsymbol{\tau}$ will change through the updates and so will the inputs of the encoder. In all experiments we carried out, we used a 3-layer architecture with 800 units for the hidden layers, as originally proposed in Dieng, Ruiz, Blei (2020). In order not to increase the number of parameters ν linearly with the number of pairs of groups, the amortised inference is used as advocated in Gershman, Goodman (2014) or Kingma, Welling (2014).

Proposition 3. *Using the assumptions described in Equations* (3.10), (3.11) *and* (3.13), *the ELBO, which is a functional of the variational distribution, reduces to a function of the variational parameters and can be split into two terms associated with the network and with the texts respectively:*

$$\mathscr{L}(R(\cdot); \boldsymbol{\alpha}, \boldsymbol{\rho}) = \mathscr{L}(\boldsymbol{\tau}, \tilde{\pi}_1, \tilde{\pi}_2, \tilde{\gamma}, \nu; \boldsymbol{\alpha}, \boldsymbol{\rho}) \tag{3.14}$$

$$= \mathscr{L}^{net}(\boldsymbol{\tau}, \tilde{\pi}_1, \tilde{\pi}_2, \tilde{\gamma}; \boldsymbol{\alpha}, \boldsymbol{\rho}) + \mathscr{L}^{texts}(\boldsymbol{\tau}, \nu; \boldsymbol{\alpha}, \boldsymbol{\rho}), \tag{3.15}$$

where $\tilde{\pi}_1 = (\tilde{\pi}_{qr1})_{qr}$, $\tilde{\pi}_2 = (\tilde{\pi}_{qr2})_{qr}$.

Proof. The proof and the exact value of the ELBO are detailed in 7.2.1 □

3.3.3 Optimisation and Algorithm

We now aim at maximising the ELBO with respect to the variational parameters $\tilde{\pi}, \tilde{\gamma}, \tau$ and ν and to the parameters $\boldsymbol{\rho}$ and α. On the one hand, following Latouche, Birmele,

Ambroise (2012), the variational parameters $\tilde{\pi}$ and $\tilde{\gamma}$ only depend on $\boldsymbol{\tau}$ and are updated as follow:

$$\tilde{\gamma_q} = \gamma_{0q} + \sum_{i=1}^{M} \tau_{iq}$$
$$\tilde{\pi}_{qr1} = \pi^0_{qr1} + \sum_{i \neq j}^{M} \tau_{iq}\tau_{jr}A_{ij}, \quad \tilde{\pi}_{qr2} = \pi^0_{qr2} + \sum_{i \neq j}^{M} \tau_{iq}\tau_{jr}(1 - A_{ij}). \tag{3.16}$$

On the other hand, ν, as well as $\boldsymbol{\rho}$ and α are optimised by a stochastic gradient descent algorithm using Pytorch automatic differentiation (Paszke et al., 2019) and the Adam optimiser (Kingma, Ba, 2014) with a learning rate of 10^{-4}. Once both parts are done, we only need to update $\boldsymbol{\tau}$ using the already up-to-date parameters. To do so, we switch from $\boldsymbol{\tau}$ lying on the simplex Δ_Q to the unconstrained space $\mathbb{R}^{Q-1}$ using for any $i \in \mathcal{V}$ and $q \in \{1, \dots, Q-1\}$:

$$\xi_{iq} = \ln(\tau_{iq}) - \ln(\tau_{iQ}).$$

We then use the automatic differentiation and the Adam optimiser with a learning rate of 0.55 to maximise the ELBO with respect to ξ. It is worth emphasising that the ELBO is optimised over the whole set of allocation probability vectors $\boldsymbol{\tau} = (\tau_i)_i$ contrary to STBM which looks for a hard allocation of nodes to clusters, one allocation being optimised at a time, all the others being fixed. Moreover, by optimising the entry of the encoder through $\boldsymbol{\tau}$, thus looking for an optimal allocation of documents to pairs of clusters, the moves in $\boldsymbol{\tau}$ aim at uncovering the optimal direction in the posterior distribution in $(\theta_{qr})_{qr}$ maximising the ELBO. In that regard, ETSBM has links with the quasi-branching bound algorithm of Jouvin et al. (2021) for document clustering. Considering a unique core for illustration, on an Intel(R) Core(TM) i7-10875H 2.30 GHz CPU and a Nvidia GeForce RTX 2080 Super 8 Go GPU, it takes about 15 seconds to analyse a dataset with 100 nodes and 1,000 documents. Moreover, studying a dataset with more than 200,000 documents, and characterising all the connections between 1,500 nodes is done in approximately 6 minutes. In practice, we emphasise that the running time can be reduced even more by considering extensive parallelisation, as well as stochastic variational inference strategies adapted for networks as in Gopalan, Blei (2013). The Python implementation of the complete methodology we propose is available at https://plmlab.math.cnrs.fr/rboutin/etsbm_package.

3.3.4 Model selection

Finally, the selection of the number of clusters Q is performed using the ELBO. It is useful to remind that the model aims to select the number of clusters providing the most meaningful results. Therefore, relying on Latouche, Birmele, Ambroise (2012), we take advantage of the Bayesian framework that automatically penalises the complexity of the model with respect to Q. The best number of clusters Q is then selected by estimating the parameters for models with different numbers of clusters Q and keeping the one with

the highest ELBO. Our experiment Section 3.2 confirms that this procedure provides a relevant model selection criterion. In this chapter, the number of topics K is not selected. Indeed, we choose to keep a high K as advocated in Dieng, Ruiz, Blei (2020). In practice, once the inference of the topics is done, a classical approach consists in focusing the interpretation on the results associated with the most frequent topics. As we shall see, in the experiment section, provided that the value of K chosen is large enough, the proposed procedure provides an accurate estimate of Q.

3.4 Numerical experiments

In this section, a series of experiments is presented to assess the proposed methodology. First, three scenarios used for benchmarking are described. Second, an illustration of the results provided by ETSBM on a simulated dataset from one of the scenarios is given. Then, results from experiments to evaluate the model selection criterion on the three scenarios considered are brought. Moreover, various strategies to initialise ETSBM are compared. Finally, an extensive set of experiments on the three scenarios with three levels of difficulty is carried out to evaluate the clustering performances of ETSBM against competitive algorithms.

3.4.1 Simulation setup

The networks with textual edges are generated following three scenarios A, B, C, as originally introduced in Bouveyron, Latouche, Zreik (2018).

Sampling networks with textual edges

- Scenario A is composed of three communities, each defining a cluster, and four topics. By definition, a community is defined such that more connections are present between nodes of the same community. For each cluster, a specific topic is employed to sample all the documents associated with the corresponding intra-cluster connections. Besides, an extra topic is considered to model documents exchanged between nodes from different clusters. Thus, by construction, the clustering structure can be retrieved either using the network or the texts only.

- Scenario B is made of a single community and three topics. Thus, all nodes connect with the same probability. Then, the community is split into two clusters with their respective topics. An extra topic is used to model documents exchanged between the two clusters. Therefore, in such a scenario, the network itself is not sufficient to find the two clusters but the documents are.

- Scenario C is composed of three communities and three topics. Two of the communities are associated with their respective topics, say t_1 and t_2. Moreover, following

the previous scenario, the third community is split into two clusters, one being associated with topic t_1 and the other with t_2. Thus, considering both texts and topology, each network is made of four node clusters. Fundamentally, both textual data and the network itself are necessary to uncover the clusters. This scenario will be of major interest in this experiment section since it allows to ensure that the two sources of information are correctly used to retrieve partitions.

The edges holding the documents are constructed by sampling words from four BBC articles, focusing each on a given topic. The first topic deals with the UK monarchy, the second with cancer treatments, and the third with the political landscape in the UK. The last topic deals with astronomy. In the general setting, for all scenarios, the average text length for the documents is set to 150 words. The parameters used to sample data from the three scenarios are given in Table 3.1. Moreover, three examples of networks generated from A, B and C are presented in Figure 3.3.

Clustering performance evaluation The main criterion used in the following to evaluate the clustering performances of the different strategies is the adjusted random index (ARI). ARI measures how close two partitions are. The closer ARI is to 1, the better the results are. A random cluster assignment leads to an ARI of 0, while a perfect retrieval of the cluster memberships gives an ARI of 1.

	Scenario A	Scenario B	Scenario C
Q (clusters)	3	2	4
K (topics)	4	3	3
Communities	3	1	3
π_{qr} (connection probabilities) $\eta = 0.25$, $\epsilon = 0.01$	$\begin{pmatrix} \eta & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon \\ \epsilon & \epsilon & \eta \end{pmatrix}$	$\begin{pmatrix} \eta & \eta \\ \eta & \eta \end{pmatrix}$	$\begin{pmatrix} \eta & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \eta & \eta \\ \epsilon & \epsilon & \eta & \eta \end{pmatrix}$
Topics between pairs of clusters (q, r)	$\begin{pmatrix} t_1 & t_4 & t_4 \\ t_4 & t_2 & t_4 \\ t_4 & t_4 & t_3 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 \\ t_3 & t_2 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 & t_3 & t_3 \\ t_3 & t_2 & t_3 & t_3 \\ t_3 & t_3 & t_1 & t_3 \\ t_3 & t_3 & t_3 & t_2 \end{pmatrix}$
Sufficient information to uncover the clusters	Network	Topics	Network & Topics

Table 3.1: Details of the three simulation scenarios to evaluate our model.

Different levels of difficulties To evaluate ETSBM against state-of-the-art STBM in Sections 3.4.3 and 3.4.5, two levels of difficulty are introduced. The first one, named *Hard 1*, makes it particularly hard to distinguish connectivity patterns by using an intra-cluster

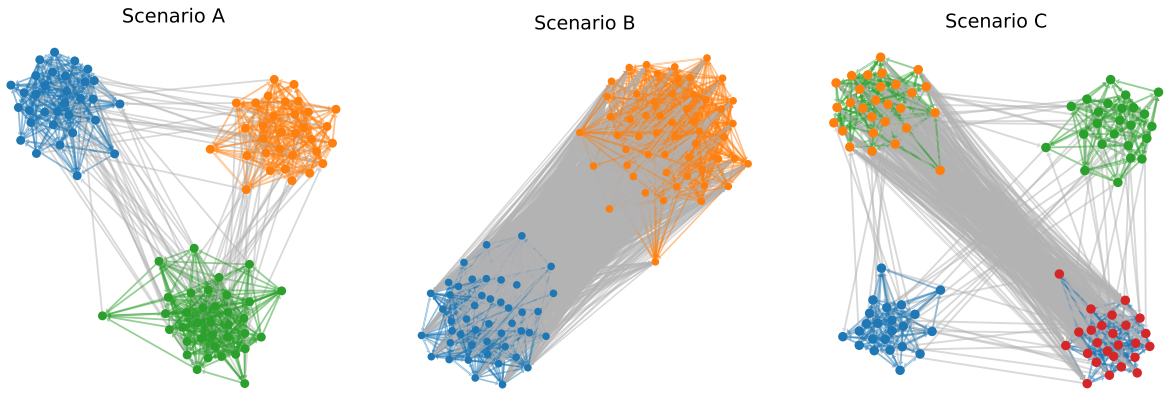


Figure 3.3: An example of each scenario is presented. The node colours denote the cluster memberships and the edge colours denote the most-used topic within the corresponding documents. The scenarios A, B and C are composed of 3, 1 and 3 communities respectively.

connectivity probability of 0.2. In Table 3.1, it corresponds to $\epsilon = 0.2$ instead of 0.01. The second one, named *Hard 2*, introduces difficulty on the text part by using smaller texts of 110 words on average instead of 150 and by adding noise. In our case, this translates into fixing:

$$\theta_{qr} = (1-\zeta)\theta^{\star}_{qr} + \zeta * \left(\frac{1}{K}, \ldots, \frac{1}{K}\right)^{\top}, \tag{3.17}$$

with $\zeta = 0.7$. Thus, for each pair of clusters (q, r), the texts are sampled according to a mixture between a multinomial distribution with probability 1 on the corresponding topic and a uniform distribution over all topics considered. Finally, the intra-cluster connection probability is decreased from 0.2 to $\eta = 0.1$.

3.4.2 An introductory example

A first glimpse at the ETSBM results on a single network simulated with Scenario C is presented here. In Figure 3.4, the evolution of the ELBO and ARI values are monitored at each iteration of the inference of ETSM applied on this single simulated network. As we can see, both the ELBO and the ARI increase after each iteration. In particular, starting from the clustering initialisation with an ARI value of 0.62, the algorithm converges to a value of 1, characterising a perfect cluster recovery. This figure illustrates the ability of the methodology proposed to retrieve the true node partition, by combining the textual and network data.

In addition, Figure 3.5 provides representations for the expected posterior estimates $\hat{\pi}$ and $\hat{\gamma}$ computed as follows $\hat{\pi}_{qr} = \tilde{\pi}_{qr1}/(\tilde{\pi}_{qr1} + \tilde{\pi}_{qr2})$ and $\hat{\gamma}_q = \tilde{\gamma}_q/(\sum_{r=1}^{Q} \tilde{\gamma}_r)$. We emphasise that the matrix characterises the connexion probabilities between clusters with a 10^{-2} rounding. It matches the expected connectivity structure described in Table 3.1.

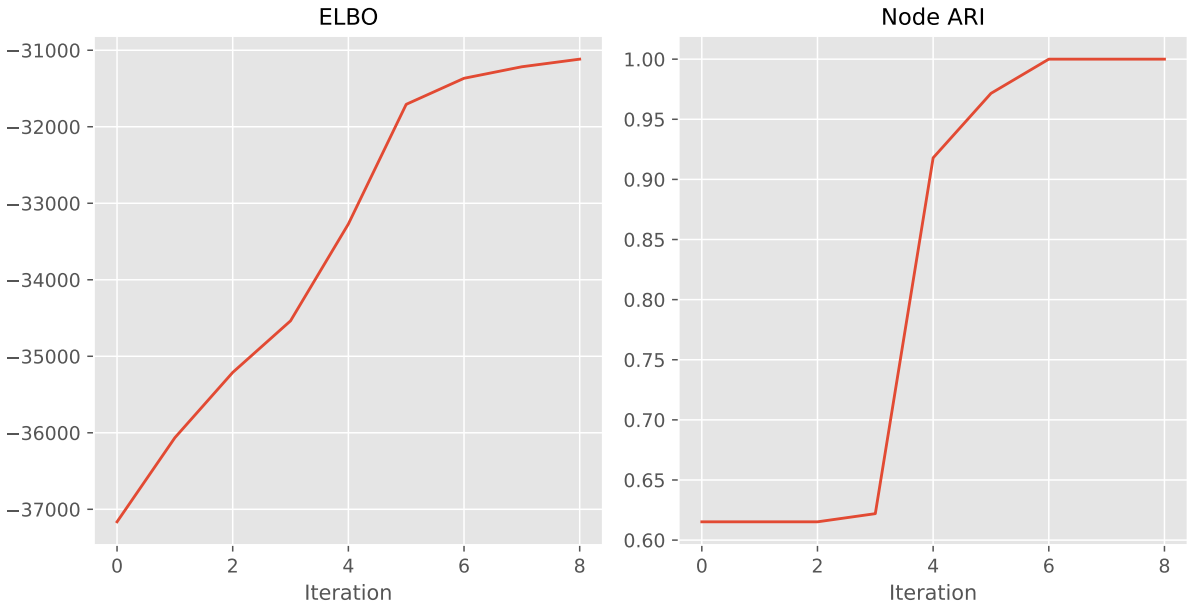


Figure 3.4: Evolution of ETSBM ELBO and ARI (y-axis) at each iteration (x-axis) on the Scenario C after initialising τ with the K-means algorithm.

Eventually, the topics learnt as well as the clustering results on the network are presented in Figure 3.6. In the network representation, the node colours correspond to the cluster memberships while the edge colours indicate the most used topic in the corresponding documents. Moreover, for each topic t_k with $k \in \{1, 2, 3\}$, the 10 words with the highest probabilities, according to the corresponding topic vector β_k, are displayed. The three topics presented are well-separated and can be identified as the topics dealing respectively with astronomy, the political landscape in the UK, and the UK monarchy, as expected. In addition, four clusters have been retrieved and the edge topics, or colours, match the description of the Scenario C setup. To conclude, ETSBM successfully renders both the network topology and the edge topics.

Finally, Figure 3.7 provides a high-level representation of the results. On the one hand, the "meta-nodes" represent ETSBM clusters and their size is proportional to the number of nodes assigned to the corresponding clusters. Moreover, the "meta-node" colours are consistent with the colours in Figure 3.6. On the other hand, the edges represent the meta-documents. We recall that they correspond to the expected posterior estimate of a document for a given pair of clusters. The edge colours correspond to the most used topic within the meta-document. The edge widths are determined by the posterior probabilities of connections between pairs of clusters. This figure underlines ETSBM capability to produce an intelligible and accurate data summary. We emphasise that graphs with thousands of edges, which sometimes cannot be represented because of memory issues, are able to be summarised in easy-to-read meta-graphs.

To conclude, this introductory example showed the ETSBM capacity to render meaningful summaries by combining both network and text information. It is worth reminding

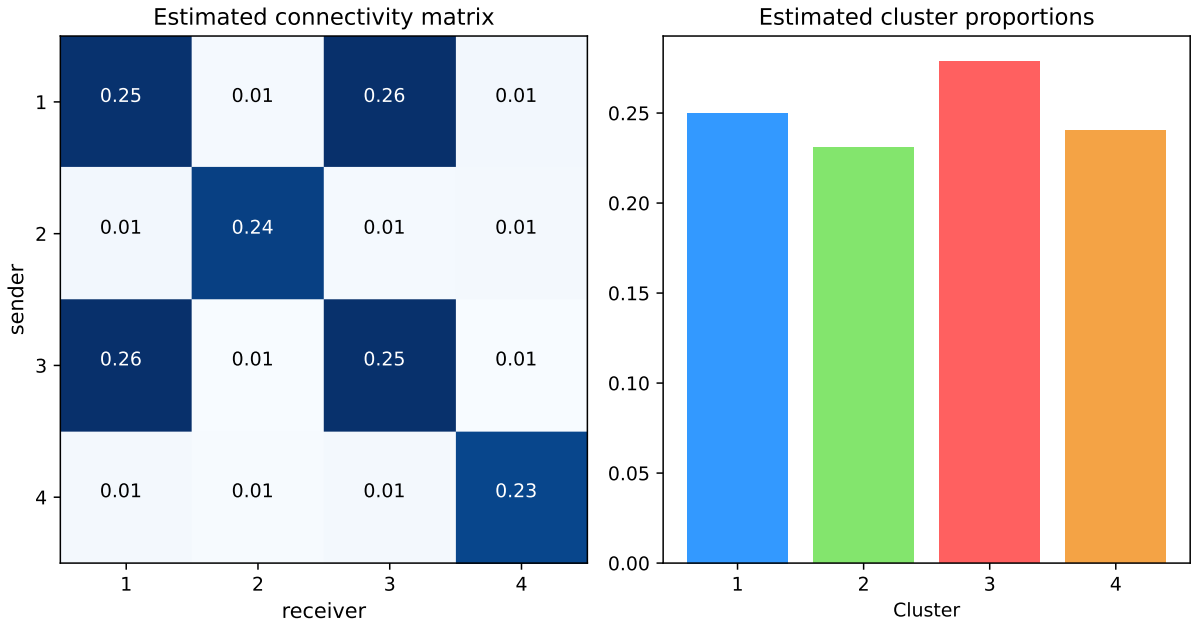


Figure 3.5: On the left-hand side, the expected posterior estimate of the connectivity matrix $\mathbf{\Pi}$ provided by ETSBM. On the right-hand side, the expected posterior estimate of the cluster proportions γ. The graph was generated following Scenario C.

that, since it comes from Scenario C, those results could not have been retrieved with models handling only network or texts as SBM, LDA or ETM.

3.4.3 Effect of the initialisation

This experiment aims to evaluate the impact of the initialisation on the final performance of our methodology. The networks are generated according to the *Hard 2* difficulty, to easily visualise the differences between the tested configurations. Moreover, the experiment is performed on Scenario C to ensure both the network and textual data are used. Three different initialisations are compared: clusters may be randomly assigned to the nodes (random), or initial clusters can be determined by a K-Means algorithm fitted on the adjacency matrix $\mathbf{A}$. Finally, the dissimilarity procedure proposed in Bouveyron, Latouche, Zreik (2018) is evaluated as the last initialisation strategy (dissimilarity). It uses both network and textual information to build a similarity matrix based on the topics discussed between nodes. Then, a K-means algorithm is performed on this similarity matrix to find a cluster allocation for each node. This initialisation strategy requires providing the topic proportion of each edge. Thus, ETM is trained on the texts and the estimated topic proportions $(\theta_{ij})_{(i,j)\in\mathcal{E}}$ are used for the dissimilarity initialisation. Figure 3.8 presents the ARI results with, for each initialisation strategy, a boxplot of the raw initialisation and ETSBM clustering.

While the random initialisation is close to 0 for ARI, both the K-means and the dissimilarity initialisation fluctuate in terms of ARI, with no clear advantage for one of the

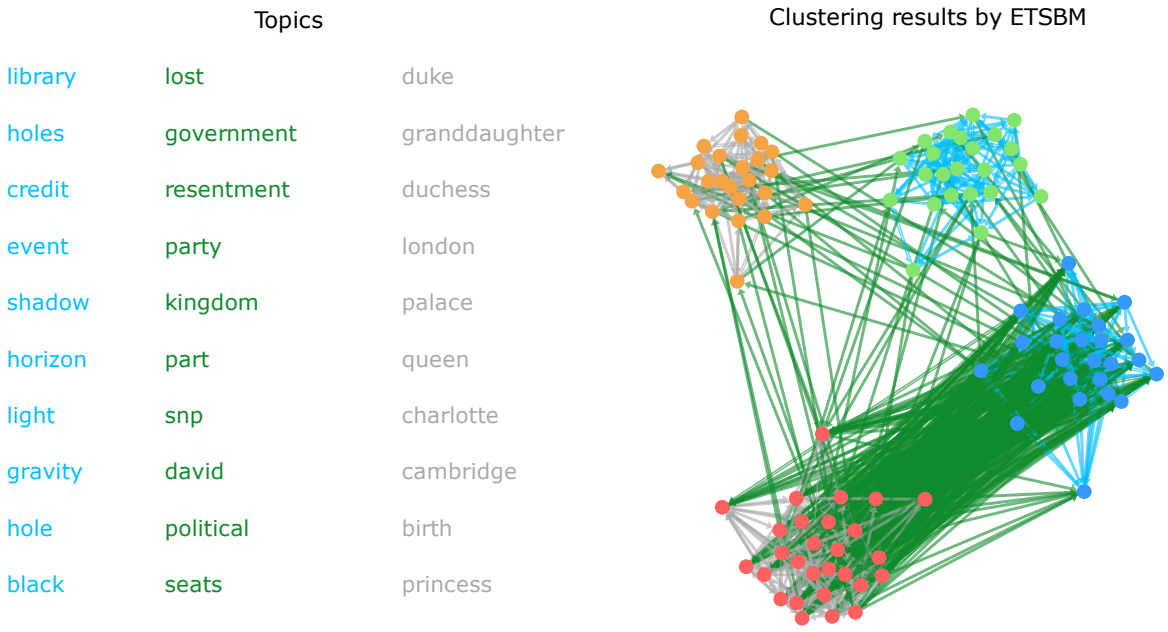


Figure 3.6: On the left-hand side, the top 10 words of each topic according to ETSBM results. Thus, for each topic t_k with $k \in \{1, 2, 3\}$, the 10 words with the highest probability values, according to the corresponding topic vector β_k, are displayed. On the right-hand side, ETSBM clustering result is illustrated. The node colours indicate the node clusters while the edge colours correspond to the most used topic within the document.

two strategies. However, ETSBM provides much better results with the dissimilarity initialisation than with K-means. It is also worth noticing that the gap between the random and K-Means initialisations has largely been closed by ETSBM algorithm. One possibility is that the model suffers the same flaws as SBM, which is for the ELBO to fall into local minimum. The use of texts in the dissimilarity initialisation may limit this effect. Therefore, we will only use the dissimilarity initialisation in the rest of the chapter as it provides the best results in most cases.

3.4.4 Model selection

This experiment aims to assess the efficiency of the model selection criterion, presented in Section 3.3.4. Let us remind that we do not aim at selecting the number of topics K since it is handled afterwards. As a consequence, the model selection criterion is evaluated for different values of K to ensure that the performances remain high, in all cases. For each scenario, 50 networks are sampled following the setup described in Section 3.4.1. For each network, ETSBM parameters are estimated taking the best initialisation out of 10. Table 3.2 presents the percentage of time a number Q is selected using the strategy proposed in Section 3.3.4 over the 50 networks, for each K value. It is worth noticing that the right model is selected more than 75% of the time, except for the Scenario B with $K = 5$, slightly below with 68%. In addition, as advocated before, for $K = 10$, the right

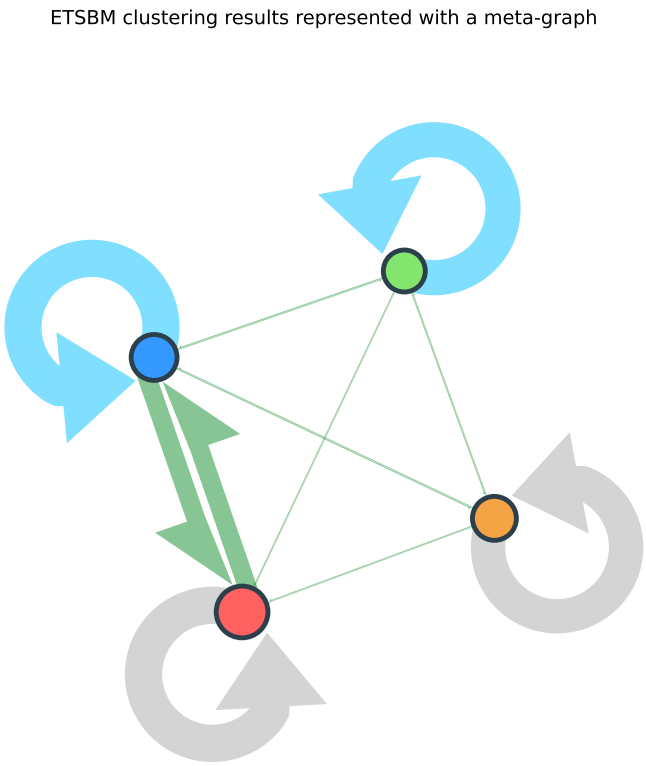


Figure 3.7: Meta representation of ETSBM results. On the one hand, the clusters are represented by the node colours, the node widths are proportional to the expected posterior estimate of the cluster proportions, and their colours correspond to the same cluster colours as in the network in Figure 3.6. On the other hand, the edges are coloured as the most used topic within the meta-document and the widths are proportional to the posterior probabilities of connections between clusters.

model is selected more than 80% of the time in each scenario. This experiment illustrates the capacity of the model selection criterion to retrieve the number of clusters. Moreover, keeping a high value of K is confirmed to be compatible with an efficient cluster number selection.

3.4.5 Benchmark study

To end this section, ETSBM is evaluated against state-of-the-art clustering algorithms for STBM. We recall that STBM is currently the only algorithm capable of simultaneously analysing the texts on the edges as well as the node connections to cluster the nodes. To provide baselines, we also give the results obtained with SBM as well as a spectral clustering algorithm (SC) presented in Shi, Malik (2000); Von Luxburg (2007), with a radial basis function as a kernel and a normalised symmetric Lagrangian. Those methods are evaluated on the three levels of difficulty presented in Section 3.4.1. Besides, results for LDA as well as ETM for text clustering are also provided. For each level of difficulty

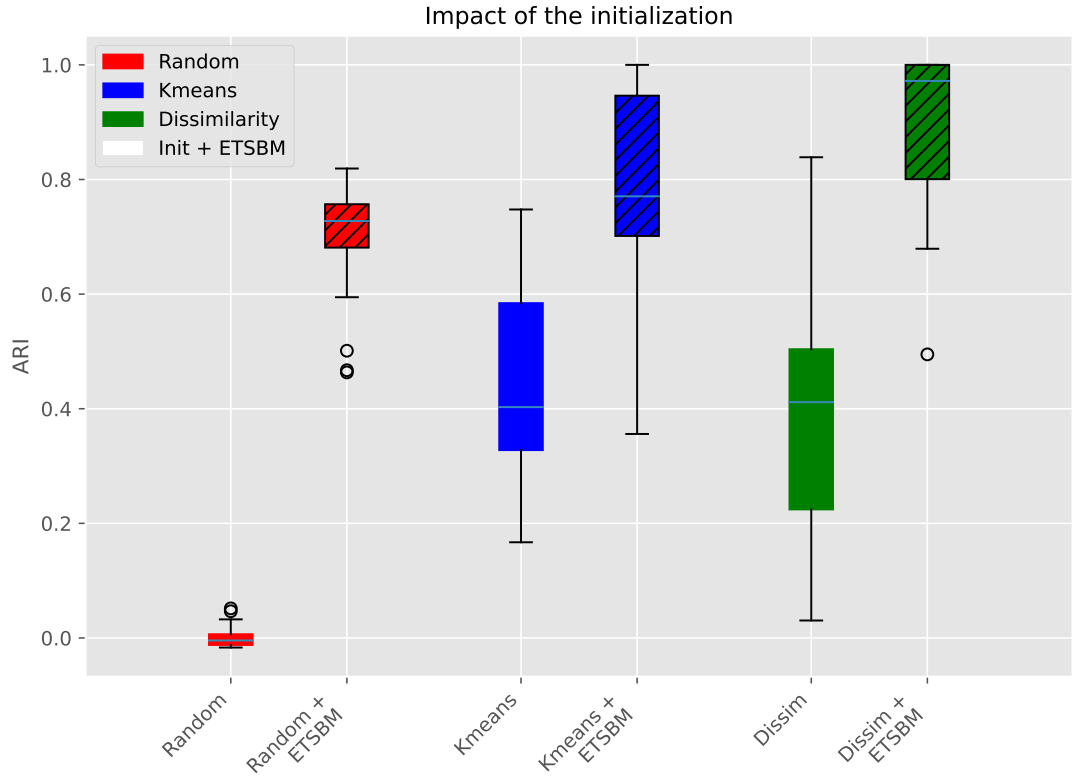


Figure 3.8: This figure displays the boxplots of the initialisation ARI (the boxplot without stripe) and of ETSBM clustering ARI with the same initialisation (the boxplot with stripes). This experiment was performed on 50 networks generated following Scenario C in the *Hard 2* setting.

and each scenario, Table 3.3 displays the mean and the standard deviation of the ARI values obtained over 50 graphs. Both the node and edge clusters ARI are provided but we recall that the main interest of this model concerns the node clustering performances. In the *Easy* and *Hard 1* settings, the ARI is always 1, which indicates that the true partitions are successfully retrieved by ETSBM and STBM. On the contrary, SBM and SC are not able to distinguish clusters in Scenario B since all nodes connect one another with the same probability. Identically, in Scenario C, SBM and SC alone cannot differentiate the nodes highly connected but discussing different topics. For instance, in the *Easy* case, this translates into an ARI of 0.01 and 0.69 respectively for SBM, and 0.00 and 0.63 respectively for SC. In the *Hard 2* setting, ETSBM node clustering significantly outperforms STBM. In particular in Scenario C, *Hard 2*, ETSBM results reach an ARI of 0.91 against 0.63 for STBM. Even though it is not the main focus of this model, the edge ARI is always higher than 0.84, which is satisfactory, and is competitive when not higher than STBM. These significant gaps in the noisy settings highlight ETSBM clustering improvement upon STBM. To conclude, our experiments strongly indicate that ETSBM node clustering performances are either the same or significantly better than STBM.

K \ Q	Scenario A					Scenario B					Scenario C				
	2	**3**	4	5	10	**2**	3	4	5	10	2	3	**4**	5	10
2	0	**94**	6	0	0	**74**	24	2	0	0	0	0	**92**	8	0
3	0	**90**	10	0	0	**78**	18	4	0	0	0	0	**90**	10	0
4	0	**78**	20	2	0	**76**	20	4	0	0	0	0	**94**	6	0
5	0	**86**	14	0	0	**68**	28	4	0	0	0	0	**84**	16	0
10	0	**88**	10	2	0	**82**	18	0	0	0	0	0	**86**	14	0

Table 3.2: This table presents the percentage of time a number of clusters have been selected on 50 simulated networks. The experiment is repeated for different values of K, and Scenarios A, B and C. For instance, in Scenario A with $K = 3$, the model with $Q = 3$ clusters was selected in 90% of cases.

3.5 Real World example: analysing the French presidential election with a Twitter dataset

In this section, we now consider the analysis of a real dataset. We start by describing the context of the study. The dataset is then presented and the results obtained with ETSBM are given. To complete this study, the results obtained with SBM and ETM employed independently are also provided. Finally, a comparison of these results with the ones obtained with ETSBM is performed.

3.5.1 Context

This section presents a use case on a Twitter dataset dealing with the French presidential election of 2022. The election resulted in Emmanuel Macron being re-elected as President of France. The objective is to use ETSBM to capture the global trends on Twitter before the first round of the French presidential election in April 2022. The network has been constructed using tweets collected by Linkfluence, a Meltwater company, during a collaboration between journalists of the French newspaper *Le Monde* and two authors of this article (Laurent, 2022). Newspapers such as *Le Monde* may be interested in having a good understanding of the global dynamics on social media during an electoral period, to understand the interest of the public opinion. Thus, interpretable topics and meaningful clusters may help them get a grasp on the core factors interesting the elector. During the last 50 years, the French political landscape has been split between two main parties, the left-democrat, mainly represented by the socialist party, and the right-liberal, represented by *Les Républicains* (formerly UMP). A shift occurred in 2017 when a three-way split between the far-left political families, the centrists, or liberals, and the far-right emerged. This analysis aims to capture the major topics discussed before the election. In addition, we want to understand the way those topics shape interactions between user groups. However, this study does not aim at making any form of prediction about the election.

Table 3.3: Benchmark of our model against STBM, SBM, SC and LDA. When a model does not provide information, a line is displayed instead of the result. For instance, SBM does not provide edge information.

		Scenario A		Scenario B		Scenario C	
		Node ARI	Edge ARI	Node ARI	Edge ARI	Node ARI	Edge ARI
Easy	ETSBM	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.99 \pm 0.03}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
	STBM	0.98 ± 0.04	0.98 ± 0.04	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$
	SBM	1.00 ± 0.00	$------$	0.01 ± 0.01	$------$	0.69 ± 0.07	$------$
	SC	0.97 ± 0.07	$------$	0.00 ± 0.01	$------$	0.63 ± 0.11	$------$
	LDA	$------$	0.97 ± 0.06	$------$	1.00 ± 0.00	$------$	1.00 ± 0.00
	ETM	$------$	0.96 ± 0.14	$------$	1.00 ± 0.00	$------$	1.00 ± 0.00
Hard 1	ETSBM	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.95 \pm 0.03}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	0.97 ± 0.04
	STBM	$\mathbf{1.00 \pm 0.00}$	0.90 ± 0.13	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.98 \pm 0.03}$
	SBM	0.01 ± 0.01	$------$	0.01 ± 0.01	$------$	0.01 ± 0.01	$------$
	SC	0.00 ± 0.02	$------$	-0.00 ± 0.01	$------$	-0.00 ± 0.01	$------$
	LDA	$------$	0.90 ± 0.17	$------$	1.00 ± 0.00	$------$	0.99 ± 0.01
	ETM	$------$	0.93 ± 0.07	$------$	1.00 ± 0.00	$------$	0.98 ± 0.03
Hard 2	ETSBM	$\mathbf{0.98 \pm 0.06}$	$\mathbf{0.83 \pm 0.07}$	$\mathbf{1.00 \pm 0.00}$	0.86 ± 0.03	$\mathbf{0.91 \pm 0.12}$	$\mathbf{0.84 \pm 0.12}$
	STBM	0.75 ± 0.27	0.82 ± 0.22	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	0.63 ± 0.19	0.77 ± 0.15
	SBM	0.96 ± 0.05	$------$	0.00 ± 0.00	$------$	0.63 ± 0.11	$------$
	SC	0.98 ± 0.08	$------$	-0.00 ± 0.01	$------$	0.60 ± 0.11	$------$
	LDA	$------$	0.77 ± 0.09	$------$	0.88 ± 0.02	$------$	0.84 ± 0.04
	ETM	$------$	0.83 ± 0.08	$------$	0.85 ± 0.03	$------$	0.86 ± 0.04

3.5.2 Dataset construction and method

In the collected data, each node represents a Twitter account. An account i is connected to j if the former retweeted the latter or if i "mentioned" j with an "@account_name" in a tweet. The texts on the edges are the tweets themselves. Our database has been created by saving any tweet talking about one of the twelve candidates. If several tweets appear from i to j, the edge (i, j) holds all those tweets stacked together. We only keep edges with text length greater than 100 characters. Then, a lemmatisation procedure is used to reduce the vocabulary size. The "stopwords", defined as non-informative words such as "and" or "it", are withdrawn, as well as numeric characters and words with a length inferior to 3 characters. In the end, we keep the largest connected component of this graph. Our dataset holds $2,730$ nodes and $403,768$ edges. This means that the graph is sparse at 94.58%. We emphasise that this level of sparsity is quite high and makes the data analysis particularly challenging. The number of topics is set to $K = 20$. Also, for each Q value, the model is trained for 10 different initialisations and the best result among those 10, ELBO-wise, is kept. Then, the number of clusters is selected using our model selection criterion. Figure 3.9 shows that the most appropriate model according to our criterion

corresponds to a number of clusters $Q = 5$.

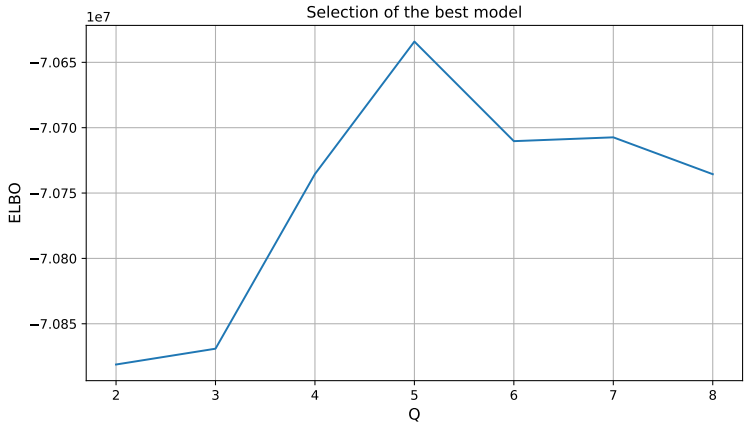


Figure 3.9: After running ETSBM with a different number of clusters Q, the ELBO suggest keeping five clusters.

3.5.3 Results

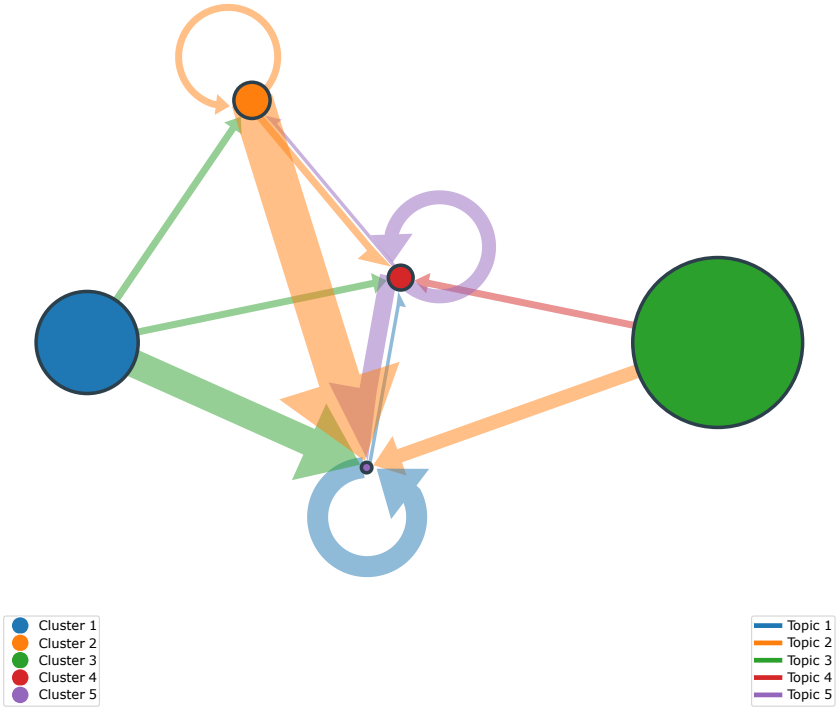
The meta-graph presented in Figure 3.10a is a high-level representation of the network. The "meta-nodes" correspond to ETSBM clusters and the edges to the meta-documents as defined in Equation (3.12). A translation of the top words is provided in Appendix 7.1. It is interesting to note the two types of clusters uncovered. In particular, Cluster 5 is composed of central accounts such as French politicians and their communication teams, for instance *Jean-Luc Mélenchon*, *Guillaume Peltier*, *En Marche #avecvous*, *les Républicains* or *Eléonore Lhéritier*. Some popular French media such as *BFMTV*, *Le Figaro*, *Valeurs actuelles*, *franceinfo* are also in this cluster. On average, the accounts in this cluster have been retweeted or mentioned 299 times against 12 times for the whole network. This cluster does not correspond to a political trend but to accounts with a high level of interactions with the rest of the graph. Despite the small size of this cluster, composed of 25 nodes, ETSBM is able to detect it and render its central function as a relay of information to other parts of the graph. This is stressed by Topic 1, the main topic discussed within Cluster 5. It regards the election as a democratic process: "round", "vote", "power", "president", and "first" which we assume stands for "first round". This core cluster is retweeted differently by the four other clusters which on the contrary hold clear political trends. Cluster 2 and Cluster 3 are interested in *Jean-Luc Mélenchon* (Topic 2) and left parties in general (Topic 4) but they seem to differ in terms of function. Cluster 2 relays information about *Jean-Luc Mélenchon* and is interacting with Cluster 4, interested in *Eric Zemmour*. On the contrary, Cluster 3 seems to only relegate content without being retweeted. Eventually, Cluster 4, interested in Eric Zemmour (Topic 5), appears to relegate content from the central accounts as well as sharing much of its own content. This dynamic differs from Cluster 1 interested in *Emmanuel Macron* (Topic 3),

which mainly retransmits information without many self-interactions. To conclude, the three-way split of the French political landscape is rightfully captured. ETSBM is also able to detect subtleties such as a split within the left wing, with the orange cluster interested only in *Jean-Luc Mélenchon* and the biggest one exchanging about different left-political front runners, *Jean-Luc Mélenchon*, *Yannick Jadot*, *Fabien Roussel* and *Anne Hidalgo*. ETSBM combines the connection information, for instance all clusters are connected to Cluster 5, and the topics information, for instance Cluster 2 and Cluster 3 should be separated, to provide relevant insights about the information organisation within the social network. This level of detail is promising and highlights how ETSBM gives a better comprehension of the complex dataset at our disposal.

3.5.4 Comparison with SBM and ETM fitted independently

Description of the results We now give the results obtained using SBM and ETM independently on the Twitter dataset in Figure 3.11. The number of topics is set to $K = 20$ again, but only the ones appearing in the meta-graph are presented. As in the previous section, we restrict the search of the number of clusters between 2 and 8 to keep the results easily interpretable and to provide a fair comparison with ETSBM. The ICL criterion selects a number of clusters $Q = 8$ which is the maximum value considered. SBM detects a central cluster in terms of connectivity of the graph (cluster 8), such that all other clusters are connected to it. It is composed of two accounts, the BMFTV account as well as Jean-Luc Mélenchon account. Most connections deal with Topic 2, which is very general but not informative.

Comparison with ETSBM results The topics in Figure 3.11 do not provide much information to understand the content of the connections in the network. In particular, Topic 2, which is general and not specific, is the most used in the meta-network. This can be explained by the independence between the construction of the clusters and the content of the tweets. Therefore, the meta-documents exchanged between clusters have no reason to be specific or to share a common topic. As a result, Topic 2 emerges as the most used topic between clusters. Compared to ETSBM results, the connections are not informative and the topics exchanged are too general to be considered for interpretation. We emphasise that among the 20 topics estimated by ETM, some are very informative but do not emerge in the meta-graph, backing the claim that the clusters are not meaningful. In addition, the number of clusters selected by the ICL (8), is higher than the number of clusters selected by ETSBM (5). Having a low number of clusters can help make the results easier to understand.



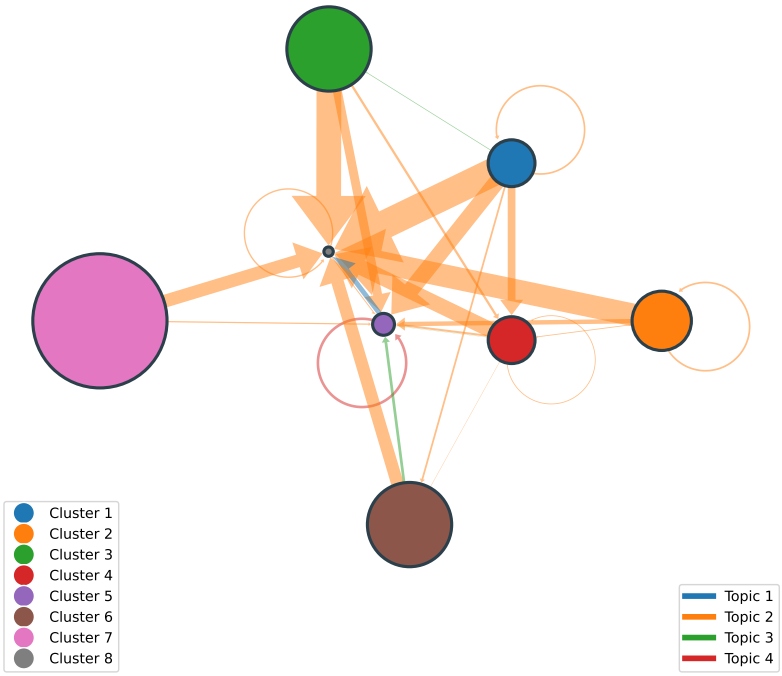
(a) Meta-network obtained with ETSBM. Each node corresponds to a cluster and the node widths are proportional to the posterior cluster proportions. On the other hand, the edges are coloured as the most used topics within the meta-documents and the widths are proportional to the posterior probabilities of connections between clusters.

Topics

tour	heure	macron	melenchon	zemmour
tout	melenchonvagagner	candidat	jadot	eric
faire	monde	emmanuel	jlm	jevotezemmour
aller	erepublique	campagne	roussel	soutenir
voter	melenchon	zemmour	voter	jevotezemmourle
pouvoir	meeting	presidentielle	gauche	hdelareconquete
vote	unionpopulaire	journaliste	vote	zemmourpresident
president	programme	debat	tour	partager
merci	marchepourla	direct	hidalgo	zemmourvsmacron
premier	melenchontf	via	droite	maintenant

(b) The most important words of the topics presented in the meta-graph above for ETSBM. A translation is provided in Figure 7.1 of the appendix.

Figure 3.10: ETSBM results on the Twitter dataset for $Q = 5$ clusters.



(a) Meta-network estimated with SBM. Each node corresponds to a cluster and the node widths are proportional to the cluster proportions. On the other hand, the edges are coloured as the most used topics of the documents exchanged between the pairs of clusters found by SBM alone. Such topics are obtained by applying ETM alone. The widths of the edges are proportional to the probabilities of connections between clusters.

Topics			
zemmour	faire	zemmour	melenchon
eric	tout	voter	faire
macron	dire	macron	heure
france	aller	faire	tout
francais	non	tout	tour
tout	bien	france	plus
faire	plus	plus	voter
hdelareconquete	voir	aller	programme
plus	pouvoir	mlp	aller
zemmourpresident	comme	seul	melenchonvagagner

(b) Meta-topics estimated with ETM on the Twitter dataset. A translation is provided in Figure 7.2 of the appendix.

Figure 3.11: SBM and ETM results on the Twitter dataset for $Q = 8$ clusters.

3.6 Conclusion and discussion

The embedded topics for the stochastic block model (ETSBM) is well suited to simultaneously find meaningful node and edge clusters. In addition, ETSBM provides an intelligible high-level representation of the graph. It can be used both on directed and undirected graphs and is suited for large datasets thanks to the variational inference. The numerical experiments showed that the ELBO is a relevant model selection criterion to estimate the number of node clusters Q in this Bayesian framework. Moreover, this criterion provides a good estimate of Q for a high number of topics K. In the end, a use case on a Twitter dataset proved the usefulness of the method. ETSBM clustering results were both meaningful and humanly intelligible. Further work may be directed toward the study of the theoretical foundations of the model selection criterion proposed. Adding temporal information concerning the connectivity patterns and the topics modelling could also contribute to obtain useful information on the data.

Chapter 4

The Deep Latent Position and Topic Model

Numerical interactions leading to users sharing textual content published by others are naturally represented by a network where the individuals are associated with the nodes and the exchanged texts with the edges. To understand those heterogeneous and complex data structures, clustering nodes into homogeneous groups as well as rendering a comprehensible visualisation of the data is mandatory. To address both issues, we introduce Deep-LPTM,

a model-based clustering strategy relying on a variational graph autoencoder approach as well as a probabilistic model to characterise the topics of discussion. Deep-LPTM allows to build a joint representation of the nodes and the edges in two embedding spaces. The parameters are inferred using a variational inference algorithm. We also introduce IC2L, a model selection criterion specifically designed to choose models with relevant clustering and visualisation properties. An extensive benchmark study on synthetic data is provided. In particular, we find that Deep-LPTM better recovers the partitions of the nodes than the state-of-the-art ETSBM and STBM. Eventually, the emails of the Enron company are analysed and visualisations of the results are presented, with meaningful highlights of the graph structure. The next section provides some motivating examples as well as our contributions. In Section 4.2, we present the assumptions concerning the generation of the data. The inference as well as the model selection criterion are presented in Section 4.3. In Section 4.4, Deep-LPTM and the impact of the initialisation are evaluated on synthetic data. An extensive benchmark study against state-of-the-art methods is also provided. Eventually, the emails of the Enron company are analysed with Deep-LPTM in Section 4.5. The results as well as the visualisations are presented to illustrate the ease of interpretation of the model outcomes.

4.1 Introduction

Numerical interactions between individuals often imply the creation of texts. For instance, on social media such as Twitter, it is possible to publish some content, a tweet or a post, that will in turn be republished, or re-twitted, by other accounts. Also, it is possible to mention another account directly in the publication. In the same way, the exchange of emails between collaborators can be seen as connections between accounts exchanging documents. Both examples can be represented by a network with the nodes corresponding to the accounts, and the edges to the exchanged texts. Such data structure is particularly difficult to apprehend, due to the heterogeneity and the volume of the data. One solution is to cluster homogeneous nodes into groups to obtain intelligible and useful information. However, very few methods performing node clustering are able to simultaneously exploit both the texts present on the edges and the connections.

In the following, we first present the advancements in the statistical network analysis field. We then review some of the core probabilistic models that can capture the main topics in a corpus of texts. Eventually, we close this section with models considering both texts and networks to cluster nodes, before introducing the contribution of the present work.

Main contributions of the chapter The model proposed in the present chapter is the first to simultaneously cluster the nodes of a network, uncover the topics in the texts exchanged between the nodes and output a representation of both the topics and the edges in a Euclidean space. To this aim, **(i)** we propose a generative model assuming that each node and each edge is represented in a latent space by a mixture of Gaussians. By doing so,

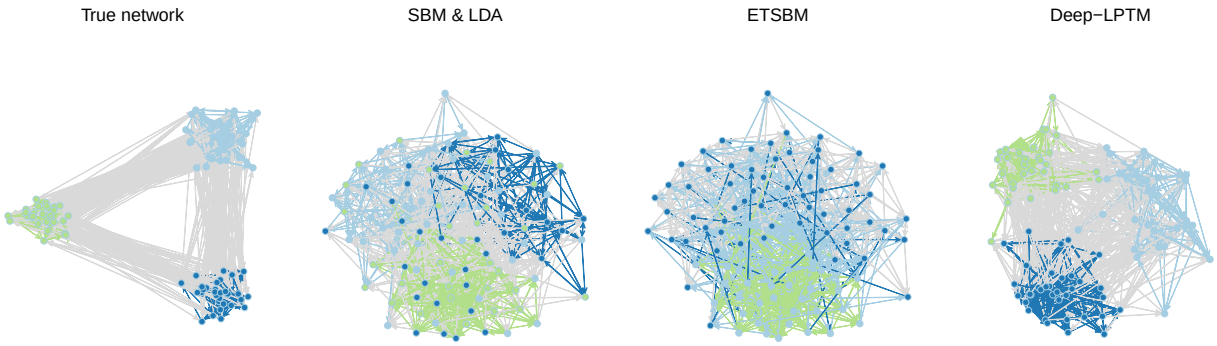


Figure 4.1: Illustration of Deep-LPTM main contributions on a synthetic network. The colours of the nodes and the edges denote the node clusters and the main topics of the corresponding documents respectively. The true node partition as well as the true topics of the documents are represented on the left-hand side. The three other figures are based on the respective results of SBM and LDA (second figure), ETSBM (third figure) and Deep-LPTM (fourth figure). Only Deep-LPTM is able to provide node positions incorporating information about the network structure as well as the document contents. The four figures were obtained with the *gplot* function from the *sna* library (Pavel N. Krivitsky et al., 2003). While the first network is plotted manually to highlight the generative structure, the two networks in the middle are based on the Fruchterman-Reingold algorithm while the fourth graph uses the node positions estimated by Deep-LPTM.

we incorporate the clustering in the generative process such that each mixture component models either a cluster of nodes or the documents exchanged between a pair of clusters of nodes. Moreover, our model distinguishes itself from former methods by allowing each node or edge to be represented by a latent position and not only by the group it belongs to. This is illustrated in Figure 4.1, where the first graph depicts the true node clusters and main topics of the corresponding documents of a simulated dataset. The second graph gives, for this dataset, the node clusters provided by SBM and the topics estimated by LDA. SBM does not retrieve the true node partition and does not provide node positions to apprehend the results. An external algorithm, namely the Fruchterman-Reingold algorithm (Fruchterman, Reingold, 1991), considering the presence of connections only in the network, had to be used for graph representation. The third graph is based on state-of-the-art ETSBM and does not recover the node partition either. Moreover, ETSBM is not able to render a comprehensive representation of the full network. Again, as for SBM, the Fruchterman-Reingold algorithm had to be used for graph representation Finally, the figure on the right-hand side presents the Deep-LPTM results. As opposed to the previous methods, Deep-LPTM is able to gather information about the network structure as well as the exchanged documents into the node positions while finding the true node partition and topics. A representation of the graph is directly obtained by the estimation procedure such that no external graph representation algorithm is needed. As we shall see, the node positions are computed by considering both the connections and the content of the corresponding documents. **(ii)** We derive a two-stage variational expectation-maximisation

(VEM) algorithm for estimating the model parameters as well as the posterior parameters of the latent positions. The first stage relies on analytical formulas to update the cluster probabilities as well as the mixture parameters. The second stage uses a stochastic gradient descent algorithm to update the expected lower bound with respect to the VGAE parameters and the deep topic model parameters. In particular, the deep topic model can make the best out of pre-trained embeddings. Thus, introducing semantic meaning into the word representations is possible as well as learning the representation from scratch. **(iii)** To choose the relevant numbers of clusters and latent space dimensions, we introduce the integrated classification and latent likelihood (IC2L) for model selection. It extends the integrated classification likelihood criterion, which was conceived for mixture models, to account for the latent representations of the nodes and of the edges. The criterion relevance is strongly upheld by the evaluation on synthetic data as well as the provided real-world use case. Moreover, by selecting a low dimension regarding the node embedding space, IC2L praises models with a strong and direct capacity of representation.

4.1.1 Notations

In this chapter, we are interested in data represented by a graph $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{1, \dots, N\}$ denotes the set of vertices. The set $\mathcal{E}$ denotes the edges between the nodes with $M = |\mathcal{E}|$ the number of edges. We focus on binary a adjacency matrix $\mathbf{A} = (A_{ij})_{ij} \in \mathcal{M}_{N\times N}(\{0,1\})$ such that A_{ij} equals 1 if $(i,j) \in \mathcal{E}$, and 0 otherwise. The graph is assumed to be directed and without any self-loop. Therefore $A_{ii} = 0$ for all $i \in \mathcal{V}$. Finally, Q denotes the number of clusters of nodes.

Each edge in the graph represents a textual document sent from one node to another. An edge from node i to node j exists or equivalently $(i,j) \in \mathcal{E}$, if and only if node i sent a textual document to node j, denoted W_{ij}. We use a bag-of-word representation of the texts where $W_{ij} = \left(W_{ij}^1, \dots, W_{ij}^V\right) \in \mathbb{N}^V$ denotes the vector of word occurrences in the document between nodes i and j such that W_{ij}^v is the number of times word v appears in the document, $M_{ij} = \sum_{v=1}^{V} W_{ij}^v$ is the total number of words in document W_{ij} and V the size of the vocabulary. The set of documents will be denoted $\mathbf{W} := (W_{ij})_{(i,j)\in\mathcal{E}}$ and the number of topics is denoted by K. Eventually, the simplex of dimension d will be denoted Δ_d.

4.2 Model

In the following, the assumptions about the graph generation as well as the hypothesis concerning the construction of the documents are presented.

4.2.1 Graph generation

Assuming that the number of clusters Q is fixed beforehand, each node i is assumed to belong to a cluster, represented by the cluster membership variable C_i. The variables C_i,

for any $i \in \mathcal{V}$, are assumed to be independent and identically distributed (i.i.d) according to a multinomial distribution such that for any node $i \in \{1, \dots, N\}$:

$$C_i \sim \mathcal{M}_Q(1, \gamma), \tag{4.1}$$

with $\gamma \in \Delta_Q$ and $C_i \in \{0,1\}^Q$ being one hot encoded so that $C_{iq} = 1$ if node i belongs to cluster q and $C_{iq} = 0$ otherwise. Thus, denoting $\mathbf{C} = (C_1, \dots, C_N)^T \in \mathcal{M}_{N \times Q}(\{0,1\})$ the cluster membership matrix, we have:

$$p(\mathbf{C} \mid \gamma) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \gamma_q^{C_{iq}}. \tag{4.2}$$

Moreover, given its cluster membership, the node i is assumed to be represented by a Gaussian vector Z_i in a p dimensional latent space such that:

$$Z_i \mid C_{iq} = 1 \sim \mathcal{N}\left(\mu_q, \sigma_q^2 \mathbf{I}_p\right). \tag{4.3}$$

Eventually, the connection between two nodes is assumed to depend on the closeness of the node representations in the latent space. Therefore, denoting $\eta_{ij} := \kappa - \|Z_i - Z_j\|$, the probability for node i to be connected to node j is:

$$P(A_{ij} = 1 \mid Z_i, Z_j, \kappa) = \frac{1}{1 + e^{-\eta_{ij}}}, \tag{4.4}$$

where a logistic function is used as a link function. For the sake of brevity, we will denote $p_{ij} = (1 + e^{-\eta_{ij}})^{-1}$. Finally, the joint-distribution of the adjacency matrix, the latent node vectors, as well as the cluster memberships can be factorised as follows:

$$p(\mathbf{A}, \mathbf{Z}, \mathbf{C} \mid \kappa, \boldsymbol{\mu}, \boldsymbol{\sigma}, \gamma) = p(\mathbf{A} \mid \mathbf{Z}, \kappa) p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(\mathbf{C} \mid \gamma). \tag{4.5}$$

where $\boldsymbol{\mu} = (\mu_q)_q$ and $\boldsymbol{\sigma} = (\sigma_q)_q$. It is worth noticing that the model described in Equations (4.1), (4.3) and (4.4) corresponds to the latent position cluster model Handcock, Raftery, Tantrum (2007). The fundamental difference with our approach for this part of the model will arise in the inference, as discussed in Section 4.3.

4.2.2 Generation of the texts on the edges

At the core of our approach is the motivation to be able to use textual data to obtain more homogeneous and meaningful clusters.

To begin with, we assume that each edge can be represented in a latent space by a Gaussian vector, depending only on the node cluster memberships. Thus, given $(C_i)_{i \in \mathcal{V}}$, the latent variables Y_{ij} are assumed to be i.i.d such that:

$$Y_{ij} \mid A_{ij} C_{iq} C_{jr} = 1 \sim \mathcal{N}(m_{qr}, s_{qr}^2 \mathbf{I}_K), \ \forall (i,j) \in \mathcal{E}, \tag{4.6}$$

where $m_{qr} \in \mathbb{R}^K$, $s_{qr} \in \mathbb{R}^+$ and $\mathbf{Y} = (Y_{ij})_{ij}$.

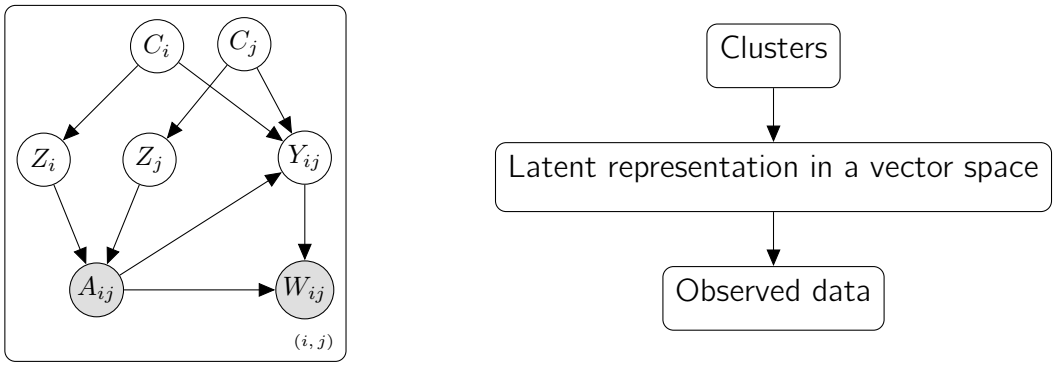


Figure 4.2: Graphical representation of the model without the parameters for the sake of clarity.

Moreover, we assume that the topic proportions of the document W_{ij}, denoted θ_{ij}, can be deduced from the latent variables such that:

$$\theta_{ij} = \text{softmax}(Y_{ij}), \tag{4.7}$$

where $\text{softmax}(x) = \left(\sum_{f=1}^{F} e^{x_f}\right)^{-1} (e^{x_1}, \dots, e^{x_F})^\top$ if $x \in \mathbb{R}^F$.

Hence, assuming that the documents are i.i.d given their corresponding topic proportions, we have for any edge $(i,j) \in \mathcal{E}$:

$$W_{ij} \mid A_{ij} = 1, \theta_{ij} \sim \mathcal{M}_V\left(M_{ij}, \boldsymbol{\beta}^\top \theta_{ij}\right), \tag{4.8}$$

where $\beta_k = \text{softmax}(\rho^\top \alpha_k) \in \mathbb{R}^V$, $\boldsymbol{\beta} = (\beta_1 \dots \beta_K)^\top \in \mathcal{M}_{K\times V}(\mathbb{R})$, $\boldsymbol{\rho} \in \mathcal{M}_{L\times V}(\mathbb{R})$, $\alpha_k \in \mathbb{R}^L$ and $\boldsymbol{\alpha} = (\alpha_1 \dots \alpha_K) \in \mathcal{M}_{L\times K}(\mathbb{R})$. Thus, denoting $\mathbf{m} = (m_{qr})_{qr}$, $\mathbf{s} = (s_{qr})_{qr}$, the joint likelihood of the texts, the latent representation of the documents, as well as the clusters memberships can be computed as:

$$p(\mathbf{W}, \mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \boldsymbol{\rho}, \boldsymbol{\alpha}, \mathbf{m}, \mathbf{s}) = p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \boldsymbol{\rho}, \boldsymbol{\alpha}) p(\mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \mathbf{m}, \mathbf{s}), \tag{4.9}$$

where $\mathbf{m} = (m_{qr})_{1\leq q,r\leq Q}$ and $\mathbf{s} = (s_{qr})_{1\leq q,r\leq Q}$.

Figure 4.2 gives a graphical representation of the generative assumptions presented above. We omitted the parameters for the sake of clarity but the full version can be found in Appendix 7.3. Interestingly enough, those assumptions can be linked to existing models as we shall see in the next section.

4.2.3 Link with other models

On the one hand, if the topic modelling alone is considered, restricting all topic proportions to be equal for all (i,j) such that $C_{iq}C_{jr} = 1$ corresponds to the text modelling in ETSBM (Boutin, Bouveyron, Latouche, 2023). In that sense, Deep-LPTM increases the freedom of each edge representation compared to ETSBM. Additionally, Deep-LPM corresponds to Deep-LPTM when the textual data present on the edges are disposed of (or LPCM if no GCN-based encoder is used in the inference strategy). Accordingly, Deep-LPTM prolongs

Deep-LPM and LPCM to networks with textual data. On the other hand, discarding the information provided by the graph and the clustering of the nodes would correspond to ETM applied to the observed documents. Therefore, Deep-LPTM extends ETM to texts with a connectivity structure to improve the topic modelling.

4.3 Inference

In the next section, the inference of the model is presented as well as the model selection criterion.

4.3.1 Likelihood

In this work, we consider the marginal likelihood of the network and the texts for parameter estimation. The latent variables are denoted by $\mathbf{C} = (C_i)_{i=1}^N$, $\mathbf{Z} = (Z_i)_{i=1}^N$ and $\mathbf{Y} = (Y_{ij})_{(i,j)\in\mathcal{E}}$, and the set of parameters is $\Theta = \{\gamma, \boldsymbol{\mu}, \boldsymbol{\sigma}, \kappa, \mathbf{m}, \mathbf{s}, \boldsymbol{\alpha}, \boldsymbol{\rho}\}$ .Thus, the marginal log-likelihood is given by:

$$\mathcal{L}(\Theta; \mathbf{A}, \mathbf{W}) = \log p(\mathbf{A}, \mathbf{W} \mid \Theta) = \log \left(\sum_{\mathbf{C}} \int_{\mathbf{Z}} \int_{\mathbf{Y}} p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta) d\mathbf{Z} d\mathbf{Y} \right). \quad (4.10)$$

Unfortunately, this quantity is not tractable since the sum over $\mathbf{C}$ requires to compute Q^N terms. Besides, it involves integrals that cannot be computed analytically. Therefore, we choose to rely on a variational inference approach for approximation purposes.

Decomposition of the marginal log-likelihood For any distribution $R(\mathbf{C}, \mathbf{Z}, \mathbf{Y})$, the following decomposition holds:

$$\mathcal{L}(\Theta; A, W) = \mathscr{L}(R(\cdot); \Theta) + \mathrm{KL}(R(\cdot) || p(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W})), \quad (4.11)$$

where

$$\mathscr{L}(R(\cdot); \Theta) = \mathbb{E}_R \left[\log \frac{p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \Theta)}{R(\mathbf{C}, \mathbf{Z}, \mathbf{Y})} \right]. \quad (4.12)$$

Since the Kullback-Leibler divergence is always positive in Equation (4.11), the ELBO $\mathscr{L}(R(\cdot); \Theta)$ is a lower bound of the marginal log-likelihood. Moreover, the closer $R(\cdot)$ is to the posterior distribution of the latent variables, in terms of Kullback-Leibler divergence, the closer the ELBO is to the marginal log-likelihood. Since the marginal log-likelihood does not depend on $R(\cdot)$, maximizing the ELBO with respect to $R(\cdot)$ is equivalent to minimizing the Kullback-Leibler divergence between $R(\cdot)$ and the posterior distribution. To make the ELBO tractable, we restrict the family of variational distributions by considering

a mean-field assumption as well as the following hypotheses:

$$R(\mathbf{C}, \mathbf{Z}, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = R(\mathbf{C})R(\mathbf{Z} \mid \mathbf{A})R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}), \tag{4.13}$$

$$R(\mathbf{C}) = \prod_{i=1}^{N} R_{\tau_i}(C_i) = \prod_{i=1}^{N} \mathcal{M}_Q(C_i; 1, \tau_i), \tag{4.14}$$

$$R(\mathbf{Z} \mid \mathbf{A}) = \prod_{i=1}^{N} R_{\phi_Z}(Z_i \mid \mathbf{A}) = \prod_{i=1}^{N} \mathcal{N}(Z_i; \mu_{\phi_Z}(\mathbf{A})_i, \sigma^2_{\phi_Z}(\mathbf{A})_i I_P), \tag{4.15}$$

$$R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W}) = \prod_{i \neq j} R_{\phi_Y}(Y_{ij} \mid W_{ij})^{A_{ij}} = \prod_{i \neq j} \mathcal{N}\left(Y_{ij}; \mu_{\phi_Y}(W_{ij}), \operatorname{diag}\left(\sigma^2_{\phi_Y}(W_{ij})\right)\right)^{A_{ij}}, \tag{4.16}$$

where $\boldsymbol{\tau} = (\tau_i)_{i=1}^N$ with $\forall i \in \{1, \dots, N\}$, $\tau_i \in \Delta_Q$. Moreover, in Equation 4.15, the mapping $\mu_{\phi_Z} : \mathcal{M}_{N\times N}(\mathbb{R}) \mapsto \mathcal{M}_{N\times P}(\mathbb{R})$ ($\sigma^2_{\phi_Z} : \mathcal{M}_{N\times N}(\mathbb{R}) \mapsto (\mathbb{R}^+)^N$ respectively) is the mapping normalising the adjacency matrix by its degree, $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{A}^{-1/2}$, and encoding the normalised adjacency matrix into the approximated posterior means (standard deviations) of the node latent positions. The diagonal matrix $\mathbf{D}$ is filled with D_{ii}, the degree of node i, for all nodes. The two mappings μ_{ϕ_Z} and $\sigma^2_{\phi_Z}$ rely on a GCN parametrised by ϕ_Z (Kipf, Welling, 2016). Similarly, in Equation 4.16, $\mu_{\phi_Y} : \mathcal{M}_{M\times V}(\mathbb{R}) \mapsto \mathcal{M}_{M\times K}(\mathbb{R})$ ($\sigma^2_{\phi_Y} : \mathcal{M}_{M\times V}(\mathbb{R}) \mapsto \mathcal{M}_{M\times K}(\mathbb{R}^+)$ respectively) encodes the documents into the approximated posterior means (standard deviations) of their corresponding latent vectors. However, the two functions rely on the ETM encoder with parameter ϕ_Y (Dieng, Ruiz, Blei, 2020). In practice, in all the experiments we carried out, we used a feed-forward neural network to encode the documents, with three layers and 800 units on each layer. The first two layers are shared to encode the variances and the means while the last layer is specific to each vector. Regarding the encoder of the adjacency matrix, we rely on Kipf, Welling (2016) such that, $\mu_{\phi_Z}(\mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{W}_0)\mathbf{W}_\mu$ and $\log \sigma^2_{\phi_Z}(\mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{W}_0)\mathbf{W}_\sigma$, where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\mathrm{ReLU}(x) = (\max(0, x_1), \dots, \max(0, x_F))$ if $x \in \mathbb{R}^F$. $\mu_{\phi_Z}(\cdot)$ and $\log \sigma^2_{\phi_Z}(\cdot)$ share the first-layer parameter $\mathbf{W}_0 \in \mathcal{M}_{N\times D}$ with $D = 10$ in all the experiments we carried out, and $\mathbf{W}_\mu, W_\sigma \in \mathcal{M}_{D\times P}$. For the sake of brevity, we will take the exponential of the encoder of the log variance and consider $\sigma^2_{\phi_Z}(\cdot)$.

Thus, the ELBO can be decomposed as follow :

$$\begin{aligned}\mathscr{L}(R(\cdot); \Theta) = {} & \mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right] + \mathbb{E}_R\left[\log p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right] + \mathbb{E}_R\left[\log p(\mathbf{C} \mid \gamma)\right] \\ & + \mathbb{E}_R\left[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] + \mathbb{E}_R\left[\log p(\mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \mathbf{m}, \mathbf{s})\right] - \mathbb{E}_R\left[\log R(\mathbf{C})\right] \\ & - \mathbb{E}_R\left[\log R(\mathbf{Z} \mid \mathbf{A})\right] - \mathbb{E}_R\left[\log R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W})\right].\end{aligned} \tag{4.17}$$

The computation of each term in Equation (4.17) is detailed in Appendix 7.3.2. To optimise the ELBO, we propose here to alternate between closed-form updates and stochastic gradient descent steps thanks to the results presented in the next section.

4.3.2 Optimisation

Analytical updates

Given a variational distribution $R(\cdot)$ complying with Equations (4.13) to (4.16), the model parameters, as well as $\boldsymbol{\tau}$, can be updated using Propositions (4) and (5).

Proposition 4. *Let $R(\cdot)$ be a variational distribution complying with Equations* (4.13) *to* (4.16)*. The parameters of the node embedding distributions maximising the ELBO are given by:*

$$\mu_q = \frac{1}{N_q} \sum_{i=1}^{N} \tau_{iq} \mu_{\phi_Z}(\mathbf{A})_i, \tag{4.18}$$

$$\sigma_q^2 = \frac{1}{pN_q} \sum_{i=1}^{N} \tau_{iq} \left(p\sigma_{\phi_Z}^2(\mathbf{A})_i + \|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2 \right), \tag{4.19}$$

where $N_q = \sum_{i=1}^{N} \tau_{iq}$ is the posterior mean of the number of nodes in cluster q.

Proof. The proof is given in Appendix 7.3.3. □

Interestingly, this proposition states that the μ_q are the weighted mean of the (approximated) posterior mean nodes positions μ_{ϕ_Z} provided by the DNN. It also indicates that each σ_q is updated as the sum of two terms: the first one corresponds to a weighted mean of the posterior variances while the second one is the intra-cluster variance weighted by the posterior clusters membership probabilities τ_i.

Proposition 5. *For a given variational distribution $R(\cdot)$ complying with Equations* (4.13) *to* (4.16)*, with parameters $\tau, \mu_{\phi_Y}, \sigma_{\phi_Y}$, the parameters of the edge embeddings distributions maximising the ELBO are given by:*

$$m_{qr} = \frac{1}{N_{qr}} \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \mu_{\phi_Y}(W_{ij}), \tag{4.20}$$

$$s_{qr}^2 = \frac{1}{KN_{qr}} \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \left[\sum_{k=1}^{K} \sigma_{\phi_Y}^2(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 \right], \tag{4.21}$$

where $N_{qr} = \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr}$ denotes the expected number of documents sent from cluster q to cluster r under the approximated posterior distribution.

Proof. The proof is given in Appendix 7.3.3 □

The interpretation following Proposition 4 can also be applied to Proposition 5 while a second DNN is used. The main difference lies in the weighting that here corresponds to the probability that the pair of nodes composing each edge belong to a pair of clusters. For instance, for the edge (i, j), the posterior probability that node i belongs to cluster q and node j to cluster r is given by $\tau_{iq}\tau_{jr}$.

Proposition 6. *For a given variational distribution $R(\cdot)$ complying with Equations (4.13) to (4.16), with parameters $\tau, \mu_{\phi_Y}, \sigma_{\phi_Y}$, the parameters τ_{iq} maximising the ELBO is given by:*

$$\tau_{iq} = \frac{\gamma_q e^{-\mathrm{KL}^Z_{iq} - \sum_{j\neq i}\sum_{r=1}^{Q}\left(A_{ij}\tau_{jr}K^Y_{ij,qr} + A_{ji}\tau_{jr}\,\mathrm{KL}^Y_{jirq}\right)}}{\sum_{l=1}^{Q}\gamma_l e^{-\mathrm{KL}^Z_{il} - \sum_{j\neq i}\sum_{l'=1}^{Q}\left(A_{ij}\tau_{jl'}K^Y_{ij,ll'} + A_{ji}\tau_{jl'}\,\mathrm{KL}^Y_{jil'l}\right)}}.$$

Proof. The proof is given in Appendix 7.3.3. □

The stochastic gradient descent

The other model parameters $\kappa, \boldsymbol{\rho}$ and $\boldsymbol{\alpha}$, and variational parameters ϕ_Z and ϕ_Y cannot be updated with analytical formulas because of the integral involving the variational distribution $R(\cdot)$ in the ELBO. In the following section, we aim at deriving estimates of the gradients of the ELBO with respect to these parameters, to perform stochastic gradient descent.

Model parameters The partial derivatives of the ELBO with respect to each parameter κ, ρ and α are obtained thanks to Monte-Carlo estimates. For instance, the gradient of the ELBO with respect to κ is estimated by:

$$\begin{aligned}\frac{\partial}{\partial\kappa}\mathscr{L}(R(\cdot);\Theta) &= \frac{\partial}{\partial\kappa}\mathbb{E}_R\left[\log p(\mathbf{A}\mid\mathbf{Z},\kappa)\right]\\ &= \mathbb{E}_R\left[\frac{\partial}{\partial\kappa}\log p(\mathbf{A}\mid\mathbf{Z},\kappa)\right]\\ &\approx \frac{1}{S}\sum_{s=1}^{S}\frac{\partial}{\partial\kappa}\log p(\mathbf{A}\mid\mathbf{Z}^{(s)},\kappa),\end{aligned}$$

where $\mathbf{Z}^{(s)} = (Z_1^{(s)},\dots,Z_N^{(s)})$ and $Z_i^{(s)} \overset{i.i.d}{\sim} \mathcal{N}(\mu_{\phi_Z}(\mathbf{A})_i, \sigma^2_{\phi_Z}(\mathbf{A})_i\mathbf{I}_p)$. The same computations result in estimates for the partial derivatives of the ELBO with respect to $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$. In practice, we rely on the common practice in the field of VAE and set $S = 1$.

Variational parameters The last parameters to update are the variational parameters ϕ_Y and ϕ_Z. Ideally, we would like to use the same computation as in the previous section. For instance, we would like to compute the following partial derivates of the ELBO with respect to ϕ_Z:

$$\begin{aligned}\frac{\partial}{\partial\phi_Z}\mathscr{L}(R(\cdot);\Theta) =& \frac{\partial}{\partial\phi_Z}\mathbb{E}_R\left[\log p(\mathbf{A}\mid\mathbf{Z},\kappa) + \log p(\mathbf{Z}\mid\mathbf{C},\boldsymbol{\mu},\boldsymbol{\sigma}) - \log R(\mathbf{Z})\right]\\ =& \frac{\partial}{\partial\phi_Z}\mathbb{E}_R\left[\log p(\mathbf{A}\mid\mathbf{Z},\kappa)\right]]\\ &- \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\frac{\partial}{\partial\phi_Z}\,\mathrm{KL}^Z_{iq}\left(\mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}(\mathbf{A})_i, \mu_q, \sigma_q\right),\end{aligned} \tag{4.22}$$

with

$$\mathrm{KL}_{iq}^{Z}\left(\mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}(\mathbf{A})_i, \mu_q, \sigma_q\right) = \log\frac{\sigma_q^p}{\sigma_{\phi_Z}(\mathbf{A})_i^p} - \frac{p}{2} + \frac{p\sigma_{\phi_Z}^2(\mathbf{A})_i + \|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2}{2\sigma_q^2}.$$

See Appendix 7.4 for the computational details. Unfortunately, the expectation of the term $\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right]$, is taken with respect to the variational distribution which depends on ϕ_Z. Therefore, the derivation of this quantity is not straightforward. Thanks to Kingma, Welling (2014) and Rezende, Mohamed, Wierstra (2014), this difficulty can be tackled by using the reparametrisation trick. In particular, let ε_i be a centred, normalised and P-dimensional Gaussian vector. Hence, the vectors Z_i and $\mu_{\phi_Z}(\mathbf{A})_i \oplus [\sigma_{\phi_Z}(\mathbf{A})_i \odot \varepsilon_i]$ have the same distribution. Therefore, the expectation can be taken with respect to $\epsilon = (\varepsilon_i)_{i=1\ldots,N}$ which gives:

$$\frac{\partial}{\partial\phi_Z}\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right]] = \frac{\partial}{\partial\phi_Z}\mathbb{E}_\varepsilon\Big[\log p\left(\mathbf{A} \mid \mu_{\phi_Z}(\mathbf{A}) \oplus \left[\sigma_{\phi_Z}(\mathbf{A}) \odot \epsilon\right], \kappa\right)\Big],$$

where $\mu_{\phi_Z}(\mathbf{A}) \oplus [\sigma_{\phi_Z}(\mathbf{A}) \odot \epsilon] = \Big(\mu_{\phi_Z}(\mathbf{A})_i \oplus [\sigma_{\phi_Z}(\mathbf{A})_i \odot \varepsilon_i]\Big)_{i=1,\ldots,N}$, $\odot$ denotes the Hadamard product and $\oplus$ the element-wise sum. A Monte-Carlo estimate of this quantity is derived by sampling S centred and reduced P-dimensional Gaussian vectors $\epsilon_i^{(s)}$ with $s = 1, \ldots, S$, $i = 1, \ldots, N$ and with $\boldsymbol{\epsilon}^{(s)} = (\varepsilon_i^{(s)})_{i=1,\ldots,N}$. Plugging it back into (4.22) gives the following estimate:

$$\begin{aligned}\frac{\partial}{\partial\phi_Z}\mathscr{L}(R(\cdot);\Theta) \approx & \frac{1}{S}\sum_{s=1}^{S}\Big[\frac{\partial}{\partial\phi_Z}\log p\left(\mathbf{A} \mid \mu_{\phi_Z}(\mathbf{A}) \oplus \left[\sigma_{\phi_Z}(\mathbf{A}) \odot \boldsymbol{\epsilon}^{(s)}\right], \kappa\right)\Big] \\ & - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\frac{\partial}{\partial\phi_Z}\mathrm{KL}_{iq}^{Z}\left(\mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}^2(\mathbf{A})_i, \mu_q, \sigma_q^2\right).\end{aligned}$$

The same derivation steps lead to a similar estimate for the partial derivatives of the ELBO with respect to ϕ_Y. Thanks to the low variances of the gradients estimated with the reparametrisation trick and to avoid increasing the computations, we use a sample size $S = 1$, as advised in the VAE literature. In addition, the computation of the partial derivatives is implemented with Pytorch automatic differentiation framework (Paszke et al., 2019) to take full advantage of the computational efficiency of GPUs. Moreover, we rely on the Adam optimiser (Kingma, Ba, 2014) to carry out the stochastic gradient descent with a learning rate of 0.002 (0.005 respectively) for the optimiser of κ and ϕ_Z (ϕ_Y respectively).

4.3.3 Model selection

To complete the inference, we present IC2L, a new model selection criterion accounting for both the clustering partition as well as the latent representations. In all previous sections, we considered the number of clusters Q, the number of topics K and the dimension of the node latent space P fixed beforehand. In this section, we aim at selecting

the triplet (K, P, Q) that captures the most information out of the data without over-parametrisation. The integrated complete (or classification) likelihood (ICL) (Biernacki, Celeux, Govaert, 2000) was introduced for mixture models and is now a common model selection criterion in this context.

First, considering a mixture model $\mathcal{M}$, with observed data $\mathbf{X}$, latent cluster memberships $\mathbf{C}$, Q clusters and the distribution parameters in the set Θ, the complete likelihood refers to $p(\mathbf{X}, \mathbf{C} \mid \mathcal{M}, Q, \Theta)$. This quantity depends on both the clustering and the model parameters. Then, to account for the uncertainty over the set of parameters and to penalise the model complexity, Biernacki, Celeux, Govaert (2000) proposed to integrate over Θ and to evaluate the quantity $\log\ p(\mathbf{X}, \mathbf{C}\ \mid\ \mathcal{M}, Q) = \int_{\theta \in \Theta} \log p(\mathbf{X}, \mathbf{C} \mid \mathcal{M}, Q, \theta) d\theta$. Since the integral is not tractable for many statistical models, the authors relied on a BIC-like approximation of this quantity, as presented in Chapter 2.1.6. In the present section, we propose to extend ICL to include the evaluation of the node embeddings $\mathbf{Z}$, as well as the edge embeddings $\mathbf{Y}$. Denoting $\mathcal{M}$ the model presented in Sections 4.2.1 and 4.2.2, we are interested in:

$$\log p(\mathbf{A}, \mathbf{W}, \mathbf{Z}, \mathbf{Y}, \mathbf{C} \mid \mathcal{M}, Q, K, P) = \log \int_{\theta} p(\mathbf{A}, \mathbf{W}, \mathbf{Z}, \mathbf{Y}, \mathbf{C} \mid \theta, \mathcal{M}, Q, K, P) p(\theta) d\theta. \tag{4.23}$$

Since this quantity is not tractable, we derive an estimate in the following proposition.

Proposition 7. *Let us consider a model $\mathcal{M}$, as described in Section 4.2, with Q denoting the number of clusters, K the number of topics and P the dimension of the node latent space. In addition, let us assume that the prior distribution over the model parameters fully factorises, as $p(\kappa, \gamma, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{m}, \mathbf{s}, \boldsymbol{\rho}, \boldsymbol{\alpha}) = p(\kappa)p(\gamma)p(\boldsymbol{\mu})p(\boldsymbol{\sigma})p(\mathbf{m})p(\mathbf{s})p(\boldsymbol{\rho})p(\boldsymbol{\alpha})$. Then, the IC2L criterion is given by:*

$$\begin{aligned} IC2L(\mathcal{M}, Q, K, P, \hat{\mathbf{Z}}, \hat{\mathbf{Y}}, \hat{\mathbf{C}}) = &\max_{\theta} \log p(\mathbf{A}, \mathbf{W}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}}, \hat{\mathbf{C}} \mid \theta, \mathcal{M}, Q, K, P) \\ &- \Omega(\mathcal{M}, Q, K, P). \end{aligned}$$

Denoting this quantity $\hat{IC2L}(\mathcal{M}, Q, K, P)$ to avoid cumbersome notations, we obtain:

$$\begin{aligned} \widehat{IC2L}(\mathcal{M}, Q, K, P) = &\max_{\kappa} \log p(\mathbf{A} \mid \hat{\mathbf{Z}}, \kappa, \mathcal{M}) - \frac{1}{2}\log(N(N-1)) \\ &+ \max_{\mu, \sigma} \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{C}}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{M}, Q, P) - \frac{QP + Q}{2}\log(N) \\ &+ \max_{\boldsymbol{\rho}, \boldsymbol{\alpha}} \log p(\mathbf{W} \mid \mathbf{A}, \hat{\mathbf{Y}}, \boldsymbol{\rho}, \boldsymbol{\alpha}, \mathcal{M}) - \frac{VL + KL}{2}\log(M) \\ &+ \max_{\mathbf{m}, \mathbf{s}} \log p(\hat{\mathbf{Y}} \mid \mathbf{A}, \hat{\mathbf{C}}, \mathbf{m}, \mathbf{s}, \mathcal{M}, K) - \frac{Q^2K + Q^2}{2}\log(M) \\ &+ \max_{\gamma} \log p(\hat{\mathbf{C}} \mid \gamma, \mathcal{M}, Q) - \frac{Q-1}{2}\log(N), \end{aligned}$$

with $\hat{\mathbf{Z}}, \hat{\mathbf{Y}}$ and, $\hat{\mathbf{C}}$ the maximum-a-posteriori estimates, and

$$\begin{aligned} \Omega(\mathcal{M}, Q, K, P) = &\frac{1}{2}\log(N(N\ -\ 1)) \\ &+ \frac{Q(P+2)-1}{2}\log(N) + \frac{L(V+K) + Q^2(K+1)}{2}\log(M). \end{aligned}$$

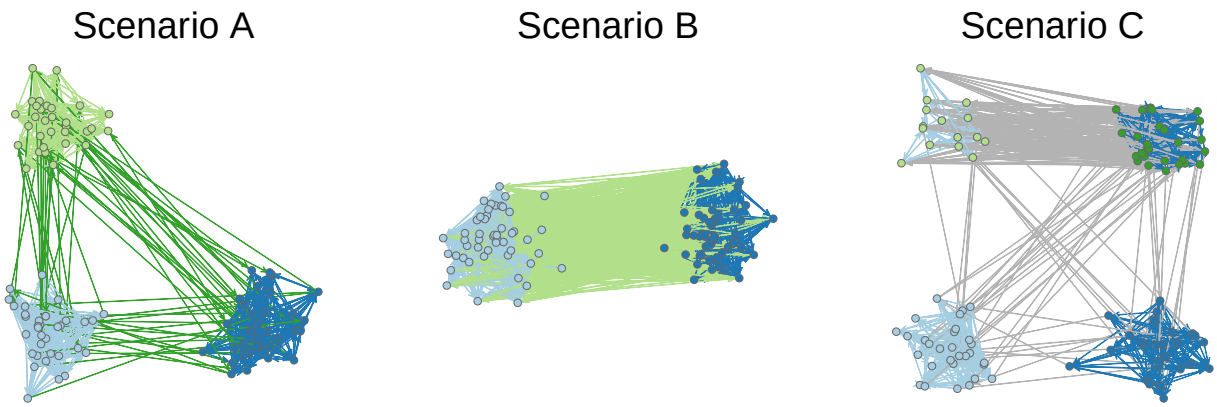


Figure 4.3: Networks sampled from each scenario. The node colours denote the node cluster memberships and the edge colours denote the majority topic in the corresponding documents.

Proof. See Appendix 7.3.3. □

Ultimately, the relevance of this criterion as well as the parameter estimations are assessed in the next section on synthetic data. Moreover, an extensive comparison with baseline methods is provided.

4.4 Numerical experiments

This section is dedicated to the assessment of the proposed methodology. We start with an introductory example to illustrate the results obtained with Deep-LPTM. We continue with an evaluation of the initialisation impact on our method. Then, we move on to Section 4.4.4 to provide numerical evidence of the robustness of IC2L against the dimensions of the parameter spaces. We close this section with a benchmark study to compare Deep-LPTM with the state-of-the-art ETSBM and STBM. Our code is available at https://plmlab.math.cnrs.fr/rboutin/deeplptm_package.

4.4.1 Simulation settings

To begin with, we introduce three simulation scenarios to be used for evaluating the methodology on different conditions detailed hereinafter.

Scenarios

- Scenario A is constituted of three communities, each defining a cluster, and four topics. By definition, a community is a group of nodes more densely connected together than with the rest of the network. For each cluster, a specific topic is employed to sample the documents associated with the intra-cluster connections. Besides, an extra topic is employed to model documents sent between nodes from

different clusters. Hence, by construction, the clustering structure can be retrieved either using the network or the texts only.

- Scenario B is made of a single community and three topics. Consequently, all nodes connect with the same probability. Then, the nodes are spread into two clusters using distinct topics. An extra topic is used to model documents exchanged between the two clusters. Accordingly, the network itself is not sufficient to find the two clusters but the documents are.

- Scenario C comprises three communities and three topics. Two of the communities are associated with their respective topics, say t_1 and t_2. Furthermore, following Scenario B, the third community is split into two clusters, one being associated with topic t_1 and the other with t_2. Thus, considering both texts and topology, each network is made up of four clusters of nodes. Consequently, both textual data and the network are necessary to uncover the clusters. This scenario will be of major interest in this experiment section since it ensures that the two sources of information are correctly used to uncover the node partition.

For all scenarios, networks with 100 nodes are sampled and the edges holding the documents are constructed by sampling words from four BBC articles, focusing each on a given topic. The first topic deals with the UK monarchy, the second with cancer treatments, the third with the political landscape in the UK and the last topic deals with astronomy. In the general setting, for all scenarios, the average text length for the documents is set to 150 words. The parameters used to sample data from the three scenarios are given in Table 4.1. Moreover, three examples of networks generated from A, B and C are presented in Figure 4.3. To summarise, the three proposed scenarios inspect different facets of the model. Scenario A ensures that the model rightfully uses the network structure, and Scenario B focuses on the usage of the topics to recover the node partition. Finally, Scenario C combines the two scenarios to guarantee that both sources of information are correctly utilised simultaneously.

Clustering performance evaluation The adjusted rand index (ARI) is used as a measure of the closeness between two partitions. In this chapter, ARI compares the true node labels with the node partition provided by a model. In particular, obtaining an ARI of 0 suggests that the clustering is as close to the true node labels as a random cluster assignment of the nodes. On the contrary, the closer the ARI is to 1, the better the results are. Ultimately, an ARI of 1 signifies that the true partition was perfectly recovered (up to a label permutation).

Level of difficulty To generate more situations from the three scenarios, we introduce the *Hard* difficulty to test the model robustness against two aspects. First, we want to test the model against documents using several topics. Thus, in the *Hard* difficulty, the documents are formed of multiple topics such that, for any edge (i, j) with node i in

	Scenario A	Scenario B	Scenario C
Q (clusters)	3	2	4
K (topics)	4	3	3
Communities	3	1	3
γ_{qr} (connection probabilities) $\eta = 0.25$, $\epsilon = 0.01$	$\begin{pmatrix} \eta & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon \\ \epsilon & \epsilon & \eta \end{pmatrix}$	$\begin{pmatrix} \eta & \eta \\ \eta & \eta \end{pmatrix}$	$\begin{pmatrix} \eta & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \eta & \eta \\ \epsilon & \epsilon & \eta & \eta \end{pmatrix}$
Topics matrix $\mathbf{T}$ between pairs of clusters (q, r)	$\begin{pmatrix} t_1 & t_4 & t_4 \\ t_4 & t_2 & t_4 \\ t_4 & t_4 & t_3 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 \\ t_3 & t_2 \end{pmatrix}$	$\begin{pmatrix} t_1 & t_3 & t_3 & t_3 \\ t_3 & t_2 & t_3 & t_3 \\ t_3 & t_3 & t_1 & t_3 \\ t_3 & t_3 & t_3 & t_2 \end{pmatrix}$

Table 4.1: Details of the three simulation scenarios used to evaluate our model.

cluster q and node j in cluster r, the topic proportions are computed as a ratio between the pure topic proportions $\theta^{\star}_{qr} \in \{0, 1\}^K$, with zeros everywhere except at the coordinate corresponding to the true topic, and between the uniform distribution over the topics. This combination is controlled by a parameter ζ such that $\zeta = 0$ corresponds to a pure topic case while $\zeta = 1$ leads to a uniform distribution over the topics. This translates into:

$$\theta_{qr} = (1 - \zeta)\theta^{\star}_{qr} + \zeta * \left(\frac{1}{K}, \dots, \frac{1}{K}\right)^{\top}, \tag{4.24}$$

with $\zeta = 0.7$ in the *Hard* setting. The second aspect tested by the *Hard* setting is the robustness in the presence of less connected communities. Consequently, the intra-cluster connection probability is decreased from $\eta = 0.25$ in the classical setting to $\eta = 0.1$ in the *Hard* one.

4.4.2 Main features of Deep-LPTM

This section gives an overview of the main features of Deep-LPTM on one network simulated according to Scenario A, with an intra-cluster connection probability η equal to 0.15, an inter-cluster connection probability ϵ fixed to 0.05 and the parameter controlling the topic proportions ζ set to 0.5. In addition, $\phi_Y, \boldsymbol{\rho}, \boldsymbol{\alpha}$, the parameters referring to the topic modelling, are pre-trained for only 5 epochs with ETM alone. Conversely, the parameters $\phi_Z, \kappa, \boldsymbol{\mu}, \boldsymbol{\sigma}$, related to network modelling, are randomly initialised without pre-training to illustrate the evolution of the node embeddings during the optimisation. In the rest of the chapter, those parameters will be pre-trained by running ETM and Deep-LPM independently beforehand.

On the one hand, the evolution of the ELBO as well as the ARI of the nodes and the edges are presented in Figure 4.4. The node ARI and the edge ARI increase to reach an ARI of 1 following the evolution of the ELBO. We only display the first 200 epochs for the sake of clarity, but the entire training is provided in the appendix.

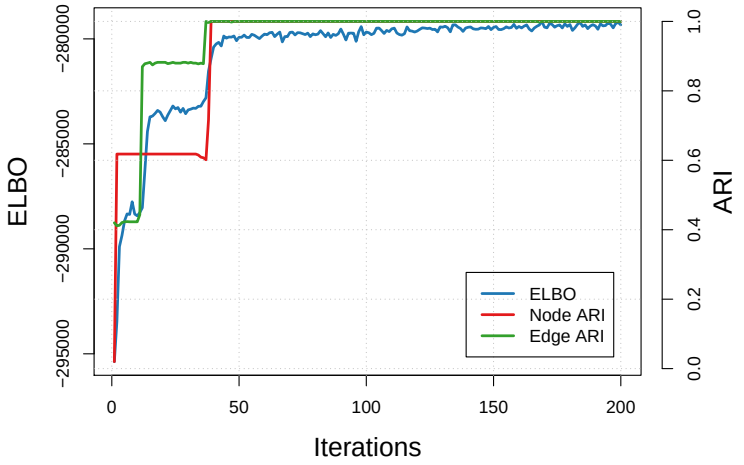


Figure 4.4: Evolution of the ELBO, as well as the node and edge ARI during the optimisation of Deep-LPTM.

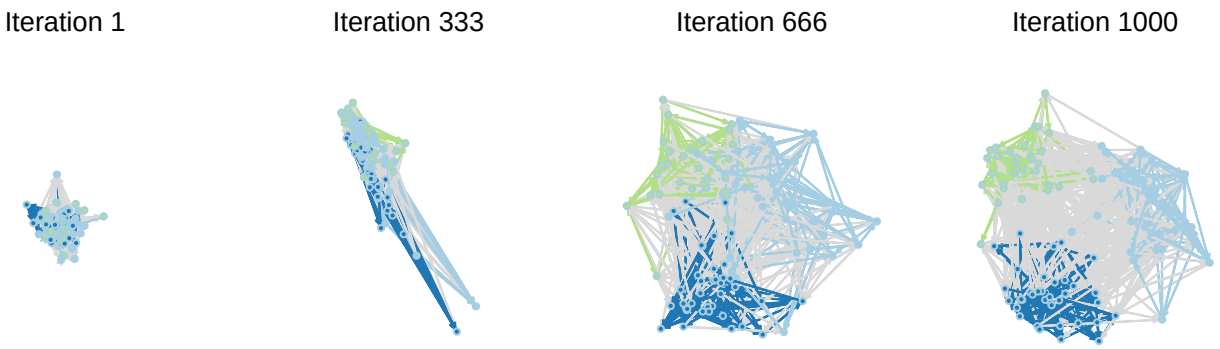


Figure 4.5: Evolution of the latent node positions during the training of Deep-LPTM.

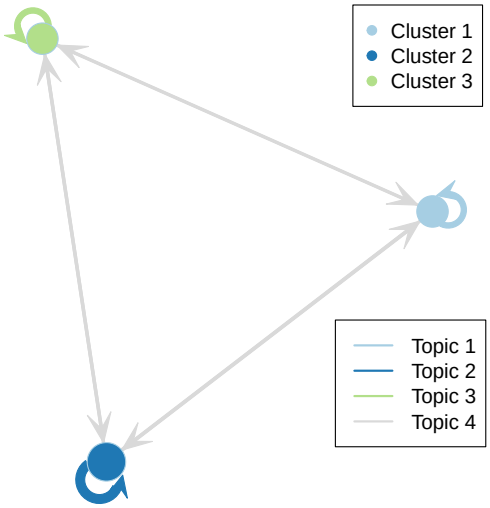


Figure 4.6: The meta-network is a representation of the network at the cluster level based on Deep-LPTM estimates. Each cluster is represented by a node, the node sizes depend on the number of nodes assigned to each cluster, the node positions as well as the major topics between two connected clusters (denoted by the colours of the edges) are estimated by the parameters $(\mu_q)_q$ as well as $(m_{qr})_{qr}$ respectively and the sizes of the edges depend on the number of connections between two clusters.

On the other hand, Figure 4.5 features the evolution of the node latent positions during the training. Interestingly enough, Deep-LPTM finds a meaningful representation of the network even with a random initialisation. This difficult problem requires training the model longer when no initialisation is provided. In the rest of the chapter, the GCN parameters as well as the topic model parameters are initialised beforehand. In addition, the node cluster membership probabilities are initialised with a similarity-based method between the topic proportions $\boldsymbol{\theta}$ of the node neighbours (Bouveyron, Latouche, Zreik, 2018).

The meta-network, represented in Figure 4.6, describes the connectivity at the cluster level. The node sizes depend on the number of nodes assigned to each cluster q and are given by $\sum_{i=1}^{N} \hat{C}_{iq}$, with $\hat{C}_{iq} = 1$ if $\arg\max_r \tau_{ir} = q$ and 0 otherwise. The cluster positions are estimated by $(\mu_q)_q$ and the major topics (denoted by the colours of the edges on the figure) between two clusters are estimated by the m_{qr} for all pairs of clusters (q, r). Finally, the sizes of the edges depend on the number of connections between clusters, given by $\sum_{i,j=1}^{N} \hat{C}_{iq}\hat{C}_{jr}$ for all pairs of clusters (q, r).

Eventually, Table 4.2 presents the topics obtained by Deep-LPTM. They are both very interpretable and distinguishable one from another which is crucial to understand complicated datasets. This will be stressed in the analysis of the Enron email dataset in Section 4.5.

4.4.3 Impact of the initialisation and the pre-trained embeddings

This section aims to evaluate the improvement of our method upon the initialisation, with a warm start and without. In this regard, Table 4.3 presents the ARI of a random initialisation

Table 4.2: Topics of the model in Scenario A Easy

	1	2	3	4
1	cancer	black	princess	seats
2	cell	hole	birth	david
3	occur	gravity	charlotte	political
4	genes	light	cambridge	lost
5	cancers	shadow	queen	kingdom
6	due	credit	granddaughter	black
7	mutations	event	duchess	party
8	radiation	disc	palace	part
9	princess	princess	london	resentment
10	include	horizon	great	united

as well as a dissimilarity initialisation (Bouveyron, Latouche, Zreik, 2018), denoted *random* and *dissimilarity* respectively, in Table 4.3. The initialisation alone (without any model name preceding it in the table) as well as the model with the initialisations (with the model preceding the initialisation) are provided. Moreover, the Deep-LPTM node clustering is evaluated with and without pre-trained skipgram embeddings (Mikolov, Chen, et al., 2013), denoted *PT* in the table. The results are obtained by averaging the ARI over 10 graphs for each scenario and difficulty and can be summarised in three points.

First, in all cases where the initialisation has not already reached an ARI of 1, Deep-LPTM improves the node clustering, even in difficult settings with no warm start. For instance, in Scenario A with the *Hard* setting, the model starts from an ARI of 0.31, with the dissimilarity initialisation, to reach 0.99 and 1.00 without and with pre-trained embeddings respectively.

Second, the improvement provided by the pre-trained embeddings depends on the scenario. On the one hand, Scenario A benefits from the usage of pre-trained embeddings which always improves the results. For instance, in the *Hard* setting with random initialisation, the ARI increases from 0.80 ± 21 to 0.95 ± 0.05. On the other hand, Scenario B and C always favour the results without pre-trained embeddings. As an example, with a random initialisation in the *Hard* setting, the ARI decreases from 0.95 ± 0.05 to 0.73 ± 0.30 and 0.47 ± 0.02 to 0.45 ± 0.04 in Scenario B and C respectively. The same deduction can be made with the dissimilarity initialisation. Since Scenario B and C are the ones evaluating the text contribution to the clustering, we advise not to use pre-trained embeddings.

Finally, the best ARI are all obtained with the dissimilarity initialisation. Consequently, Deep-LPTM will only be initialised with it in the rest of the chapter.

4.4.4 Model selection

In order to assess the model selection criterion, this section provides two experiments. First, we evaluate IC2L relevancy to select Q on all three scenarios. Second, we test IC2L

		Scenario A	Scenario B	Scenario C
Easy	Random init	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Dissimilarity init	0.97 ± 0.06	1.00 ± 0.00	0.98 ± 0.03
	Deep-LPTM random	1.00 ± 0.00	1.00 ± 0.01	0.63 ± 0.20
	Deep-LPTM dissim	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Deep-LPTM - PT random	1.00 ± 0.00	0.90 ± 0.30	0.55 ± 0.15
	Deep-LPTM - PT dissim	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Hard	Random init	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Dissimilarity init	0.31 ± 0.14	1.00 ± 0.00	0.38 ± 0.24
	Deep-LPTM random	0.80 ± 0.21	0.95 ± 0.05	0.47 ± 0.02
	Deep-LPTM dissim	0.99 ± 0.02	1.00 ± 0.00	0.89 ± 0.15
	Deep-LPTM - PT random	0.95 ± 0.05	0.73 ± 0.30	0.45 ± 0.04
	Deep-LPTM - PT dissim	1.00 ± 0.01	1.00 ± 0.00	0.85 ± 0.18

Table 4.3: Adjusted rand index (ARI) of the initialisations and the results of Deep-LPTM in terms of node clustering, without and with pre-trained embeddings (denoted PT in that case). ARI is averaged over 10 graphs, for each scenario and difficulty.

	Scenario A $Q^\star = 3$	Scenario B $Q^\star = 2$	Scenario C $Q^\star = 4$
$Q = 2$	0	10	0
$Q = 3$	10	0	0
$Q = 4$	0	0	10
$Q = 5$	0	0	0
$Q = 10$	0	0	0

Table 4.4: Number of times a value Q is selected by the IC2L criterion over 10 graphs with the true value of K and $P = 2$.

efficiency to select the triplet (K, P, Q) on Scenario C specifically.

Selection of Q with $P = 2$ and the true K Keeping P set to 2 and K fixed to its true value for the moment, Table 4.4 assesses the effectiveness of IC2L to select the number of clusters Q. In all three scenarios, IC2L selects the true model 10 times out of 10.

Selection of the triplet (P, K, Q) Let us now consider selecting the triplet (K, P, Q) simultaneously. Table 4.5 displays the number of times a triplet is selected by IC2L for 10 graphs simulated according to Scenario C (with $Q^\star = 4$ and $K^\star = 3$ the true values). The selected node embedding dimension is always $P = 2$, thus, we only provide K and Q in the table. First, it is satisfactory that IC2L always selects the true number of topics and clusters. Second, by always picking the dimension $P = 2$, IC2L favours models with a

	$K=2$	$\mathbf{K=3}$	$K=4$	$K=5$	$K=6$
$Q=2$	0	0	0	0	0
$Q=3$	0	0	0	0	0
$\mathbf{Q=4}$	0	10	0	0	0
$Q=5$	0	0	0	0	0
$Q=6$	0	0	0	0	0

Table 4.5: Number of times a triplet (K, P, Q) is associated with the highest IC2L over 10 graphs simulated according to Scenario C ($Q^\star = 4$ and $K^\star = 3$). All the models with the highest IC2L value correspond to $P = 2$. Therefore, only the table corresponding to this value is shown.

lower complexity. In our case, this translates into choosing models able to directly visualise the data in two dimensions, simply by plotting the latent vectors Z_i. This suits our purpose of building an explainable model.

4.4.5 Benchmark

We close this section with a benchmark study presented in Table 4.6 to compare Deep-LPTM with state-of-the-art ETSBM and STBM. We also provide SBM and Deep-LPM as baselines even though they are not able to take into account the text edges. As in the previous sections, the table presents the average of the ARI over 10 graphs. Each graph result is obtained by running each method with five different initialisations and by taking the one resulting in the highest ELBO. The table is presented for three different models, namely STBM, ETSBM and Deep-LPTM. The last two models are evaluated with and without pre-trained embedding. In all cases, Deep-LPTM is either as good as or better than other models. In particular, in Scenario C with difficulty *Hard*, the ARI of Deep-LPTM node clustering is higher than all other methods, by at least 0.15. Likewise, in Scenario A with difficulty *Hard*, Deep-LPTM always recover the true partition while STBM only reaches an ARI of 0.66 ± 0.18.

4.5 Application to the analysis of the Enron email network

Enron, formed in 1985, was an American company selling natural gas in North America. In 2001, the securities and exchange commission (SEC) opened an investigation on October, 31th, for fraud, while on August, 14th, the company was "probably in the strongest and best shape that it [had] probably ever been in" according to its CEO. In early December of the same year, the company filed for the largest bankruptcy at that time. We propose here to concentrate on the critical period from September, 1st to December, 31st leading to the downfall of the company to understand the organisation of the company during this

		Scenario A	Scenario B	Scenario C
Easy	SBM	1.00 ± 0.00	-0.00 ± 0.01	0.73 ± 0.05
	STBM	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01
	ETSBM	0.99 ± 0.03	1.00 ± 0.00	0.96 ± 0.04
	ETSBM - PT	1.00 ± 0.00	1.00 ± 0.00	0.96 ± 0.05
	Deep-LPTM	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	Deep-LPTM - PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Hard	SBM	0.97 ± 0.03	0.00 ± 0.00	0.62 ± 0.1
	STBM	0.63 ± 0.23	1.00 ± 0.00	0.66 ± 0.19
	ETSBM	0.96 ± 0.10	0.90 ± 0.30	0.72 ± 0.25
	ETSBM - PT	0.99 ± 0.01	1.00 ± 0.00	0.74 ± 0.21
	Deep-LPTM	0.99 ± 0.02	1.00 ± 0.00	0.89 ± 0.15
	Deep-LPTM - PT	1.00 ± 0.01	1.00 ± 0.00	0.85 ± 0.18

Table 4.6: ARI of the node clustering averaged over 10 graphs in all three scenarios for the two levels of difficulty Easy and Hard. Deep-LPTM, as well as ETSBM, are presented with and without pre-trained embeddings (denoted PT). Moreover, STBM and SBM are also provided as baselines.

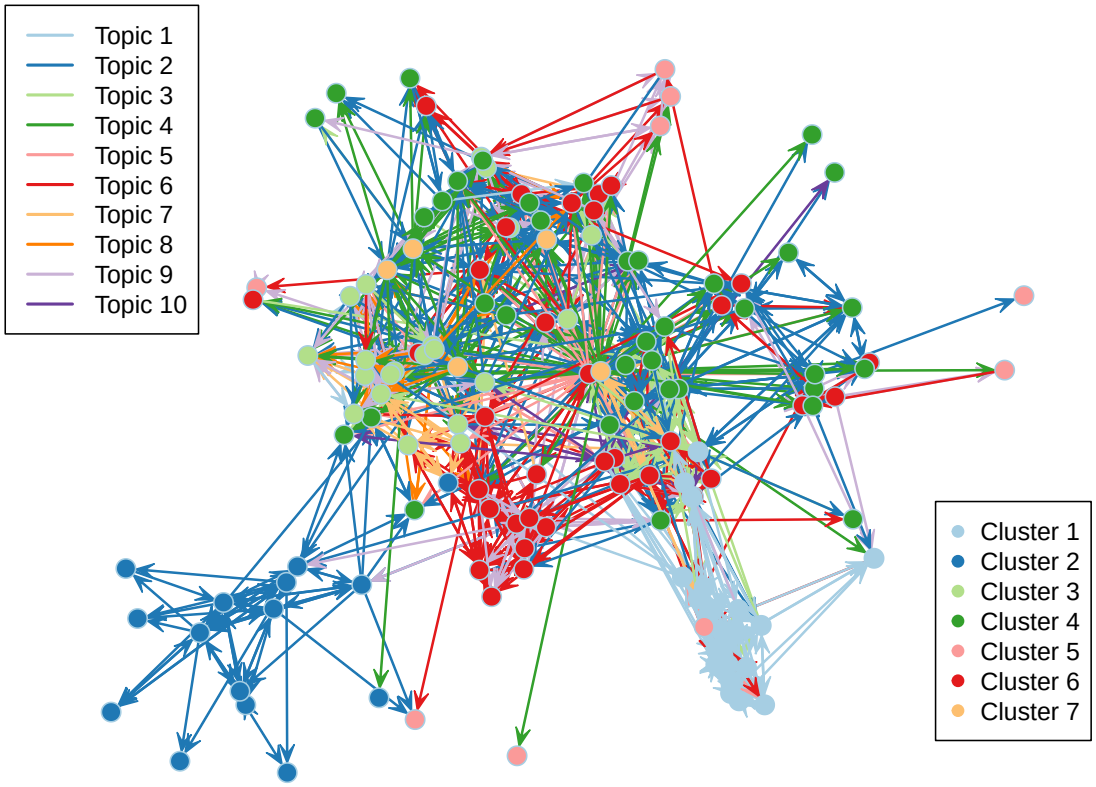


Figure 4.7: Deep-LPTM representation of Enron email network. The nodes positions, the node cluster memberships (denoted by the colour of the nodes) as well as the majority topic in the documents (denoted by the colour of the edges) are estimated by Deep-LPTM.

Figure 4.8: The 10 most probable words of each topic according to Deep-LPTM.

crucial period. Thanks to the decision of the federal energy regulatory commission (FERC), the dataset is publicly accessible and contains all 20,940 emails exchanged between 149 employees. All edges holding multiple messages were coerced into a single meta-message by stacking the documents together. As a result, the network holds 1,234 edges between the 149 employees. The dataset can be found at https://www.cs.cmu.edu/~./enron/.

The estimation of Deep-LPTM is conducted for all triplets (Q, K, P) where $Q \in \{5, 7, 10\}$, $K \in \{3, 5, 7, 10\}$ and $P \in \{2, 4, 8, 16\}$. The highest value of IC2L corresponds to the triplet $(Q, K, P) = (7, 10, 2)$. Deep-LPTM clustering results are displayed in Figure 4.7. The node cluster memberships as well as the edge majority topics are represented by their respective colours. In addition, the topics are interpreted by looking at their corresponding most probable words in Figure 4.8.

Topics analysis The topics can be depicted as follow:

- Topic 1 concerns Charles Watson, Dynegy CEO at the time, who negotiated a deal to finance Enron, involving the *transwestern* pipeline, and to merge the companies
- Topic 2 refers to regional energy
- Topic 3 deals with business operations
- Topic 4 is related to office supplies and day-to-day work
- Topic 5 mentions the energy usage and delivery
- Topic 6 is related to legal and strategical aspects of Enron's business, involving Sara Shakleton (vice president of Enron North America Corporation), and Debra Perlingiere, from the legal department
- Topic 7 is concerned with infrastructures and geographical projects
- Topic 8 corresponds to discussions about Enron activities in Afghanistan, which may be seen as sensitive given the American situation in 2001
- Topic 9 focuses on financial aspects
- Topic 10 mentions the California electricity crisis, which almost led to the bankruptcy of the Southern California Edison corporation

The topics as well as the visualisation provide significant information on the dataset. In particular, Deep-LPTM identifies different departments and cases of the company through the topics and successfully represents it in the graph structure as we shall detail.

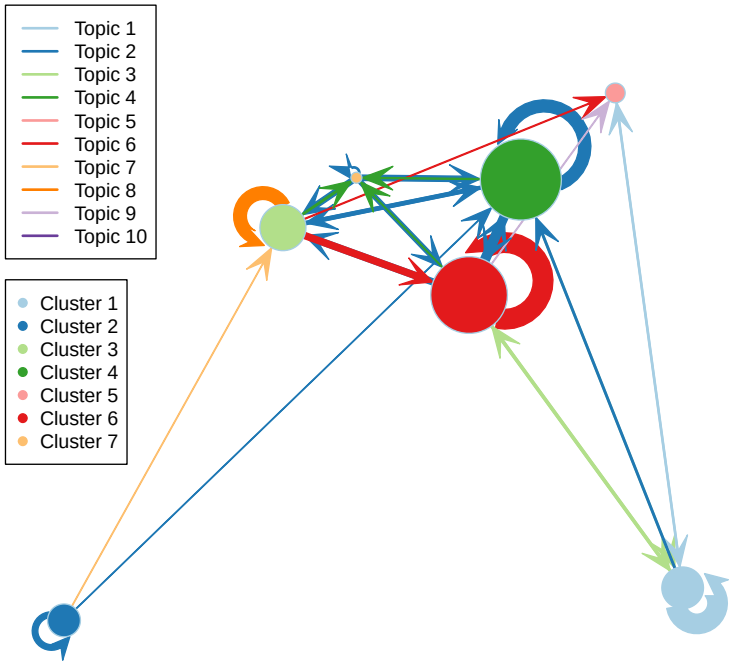


Figure 4.9: Enron email meta-network, based on Deep-LPTM estimates, represents the clusters and their interactions. The node sizes depend on the number of email accounts assigned to each cluster and the node positions are estimated by Deep-LPTM. The colours of the edges refer to the majority topic between the respective clusters and the sizes of the edges depend on the number of connections between the clusters. We keep only the edges corresponding to more than five connections for readability purposes.

Cluster analysis The meta-network in Figure 4.9 gives an overview of the results to analyse the clusters. We keep only the edges corresponding to more than five connections for readability purposes. However, important details provided by Figure 4.7 could not be uncovered with the meta-network only.

First, Clusters 1 and 2 are well separated from the rest of the graph. Moreover, each one is characterised by a specific topic. Cluster 1 is involved in discussions about a financial deal and a possible merger (Topic 1) and more than half of the employees in Cluster 1 have a managerial position or work in the legal department (vice presidents, directors, lawyers). Cluster 2 refers to the regional energy business (Topic 2) and nine out of the 15 nodes correspond to either employees or traders.

Second, Cluster 6 displays a high level of internal connectivity and is essentially featured by discussions involving the legal department (Topic 6) as well as the financial aspects of the company (Topic 9). The status of employees in that cluster are mainly composed of directors, vice presidents as well as presidents of the company. Interestingly, Deep-LPTM placed a president, corresponding to the node with the highest degree in the graph, at the centre of the network, and allocated it to the cluster of managers of the company. This graphical property, unique to Deep-LPTM, stresses the incorporation of the connections, as well as the topics in the emails, to obtain a meaningful representation of the network. Conversely, in Figure 4.9, the meta-network is not able to dissociate the connectivity of a node with a different connectivity than the rest of the cluster. Thus, Cluster 6 is central as a whole in the network in Figure 4.9.

Third, the topics involved in emails of Clusters 3 and 4 are the main drivers of the characterisation of the two clusters. For instance, Cluster 3 is highly connected to the graph and is involved in discussions about infrastructures and geographical aspects (Topic 7), as well as financial aspects (Topic 9). Emails related to Afghanistan were exchanged, mainly involving nodes from Cluster 3. It is worth noticing that nodes in cluster 3 correspond to people with a high position in the hierarchy of the company (almost two-thirds of the employees in the clusters are either managers, directors, vice presidents, presidents or CEO). In addition, although many nodes in Cluster 4 are involved in regional energy discussions (Topic 2) like Cluster 2, they also discuss day-to-day business questions (Topic 4), and are highly connected with the rest of the network, unlike Cluster 2.

Finally, Cluster 7 is composed of four nodes that are at the crossroads between several clusters and topics. We close this analysis with Cluster 5 which is composed of nodes mainly receipting emails and poorly connected to the graph, highlighted in the representation by the node positions being on the rim of the network.

4.6 Conclusion

We introduced a novel deep probabilistic model, named the deep latent position topic model (Deep-LPTM) to analyse networks with textual edges. Deep-LPTM allows to simultaneously clustering nodes, modelling the topics in the documents, and providing a network visualisation. To represent networks in an Euclidean space, most methodologies

rely either on heuristic method used a posteriori, such as K-Means, or only focus on the connectivity of the graph. On the contrary, this work tackles this problem by incorporating the network representation into the modelling, enabling the clustering as well as the topic modelling to be included in the calculation of the node positions. To benefit from the flexibility of deep neural networks, our methodology is based on a graph convolutional network (GCN) to encode the nodes into a vector space using the connectivity of the graph as well as a neural topic model to encode the documents and the topics into a vector space. Even though we focused on directed networks, the extension to undirected networks is straightforward. The applications of this methodology are numerous, including social sciences, journalism and social network analysis. The proposed methodology relies on a variational inference algorithm to maximise the marginal likelihood. The optimisation combines analytical formulas and stochastic gradient descent steps to estimate the parameters. In this chapter, we also derived the integrated classification and latent likelihood (IC2L) criterion to choose relevant numbers of clusters and topics as well as the dimension of the node latent space. Both the extensive benchmark study, as well as the Enron emails analysis, highlighted the visualisation power and the clustering efficiency of the proposed methodology.

Chapter 5

The Deep Latent Position Block Model

Network structures are able to represent any type of relations between objects. This key property makes networks very appealing to analyse complex datasets, such as gene co-expression networks or social networks. However, to capture meaningful information from a network data structure, it is necessary to identify the patterns responsible for the data generation, as well as be able to visualise them. In this chapter, we introduce the deep latent block model (Deep-LPBM). This model uses continuous node representations, obtained with a graph convolutional network, to encode the node cluster membership probabilities and a marginalised block model to estimate the existence of an edge between a pair of nodes. In particular, the method we propose offers a high flexibility by considering

a mixture model at the edge level. As a result, the node embeddings can be projected into a 2-dimensional space and therefore, provide a representation of the entire network incorporating the block modelling estimations. To estimate the parameters of the model, a variational inference strategy is used to obtain a lower bound of the intractable likelihood and is used as the objective function to maximise. The optimisation combines closed-form updates as well as a stochastic gradient descent algorithm. This is an ongoing work, an extensive benchmark will be provided later on as well as the analysis of a real-life dataset. A comparison of Deep-LPBM representations with the variational graph autoencoder as well as the deep latent position model on synthetic data is provided, as well as the assessment of the clustering efficiency of the model, on three distinctive network architectures. To the best of our knowledge, this is the first model able to combine continuous node representation with block modelling as well as a graph convolutional network.

The first section is dedicated to motivating this work as well as presenting our contribution. In Section 5.2, the assumptions regarding the generative models are exposed. Section 5.3 provides the inference strategy adopted to estimate the parameters of the model. Eventually, we compare Deep-LPBM clustering results with the VGAE and the deep latent position model in Section 5.4.

5.1 Introduction and contribution

5.1.1 Introduction

Networks are encountered in a variety of fields, ranging from social sciences, and biology to social networks. Among their attractive properties, their capacity to represent any type of object and relationship is well appreciated. However, they present difficulties to apprehend since non-observed features, such as node cluster memberships, may impact the observed network topology and engender specific connectivity patterns. This requires the development of specifically fashioned methods, models and inference strategies to capture information. For instance, for some specific types of groups, methods have been developed specially designed. To give an example, one of the most studied types of groups is called a community and corresponds to nodes highly connected to nodes of the same group but poorly connected to nodes from other groups. The community-detection methods are numerous but do not necessarily generalise to other types of structure such as stars (a group of nodes connected to nodes of different groups but not connected together). Therefore, being able to capture the structure underlying the data is essential to model any type of relationship, even without prior knowledge of the network topology. This flexibility was provided by block model approaches, enabling to model any type of connectivity patterns among the network. However, this flexibility comes at the cost of representation. Indeed, those approaches do not provide a direct representation of the network, but only a high-level depiction of the underlying patterns, where clusters are represented as nodes, cluster sizes as node sizes, and the number of connections between

clusters as edge widths. This meta representation may hide some node-specific properties of specific nodes. For instance, a node in a cluster may be more connected to another group than the other nodes in its cluster. Hence, this feature should appear in the network representation, since this node might play a crucial role in the network, precisely because it connects two clusters. To represent a network, positional approaches have been proposed by performing link prediction based on the similarity between estimated continuous node representations. Unfortunately, such methods only estimate communities. This chapter aims at combining block modelling with a positional approach, by incorporating graph neural network capacity to provide informative embeddings of the nodes. To this end, a marginalised block model is introduced, where a logistic-Gaussian distribution models the node cluster membership probabilities and is used for visualisation purposes.

5.1.2 Main contribution and notations

A new methodology bridging the gap between block modelling and network positional learning is introduced. Deep-LPBM takes advantage of a graph convolutional network combined with a new decoder to make the most of the block modelling flexibility. In particular, Deep-LPBM is able to analyse disassortative graphs as well as communities and provide meaningful representations and clustering. This is assessed through experiments led on varying network structures. In particular, it provides convincing results on the improvement of our method over competitors.

The work presented in this chapter is still under development and will come with a Python package.

Notations In this chapter, matrices and collections of vectors are denoted in bold cases $\mathbf{X}$, the space of $n \times m$ matrices with coefficient in E is denoted $\mathcal{M}_{n\times m}(E)$, and should not be confused with the multinomial distribution denoted $\mathcal{M}_n(m, p)$ where n is the dimension of the vector, m is the number of draws and $p = (p_1, \dots, p_n) \in \Delta_n$ is the probability vector. The n-dimensional simplex is denoted

$$\Delta_n := \left\{ p \in \mathbb{R}^n : \forall i, p_i \geq 0 \text{ and } \sum_{i=1}^{n} p_i = 1 \right\}.$$

Moreover, the network is denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, N\}$ denotes the set of vertices and $\mathcal{E}$ the set of edges.

5.2 Assumptions regarding the network generation

In this section, we present the assumptions concerning the graph generation. Assuming that the number of clusters Q is fixed beforehand, each node $i \in \mathcal{V}$ is assumed to belong to a cluster, represented by the cluster membership variable C_i. The variables $(C_i)_i$ are assumed to be independent and identically distributed (i.i.d) according to a multinomial

distribution such that:

$$C_i \overset{i.i.d}{\sim} \mathcal{M}_Q(1, \gamma),$$

with $\gamma \in \Delta_Q$ the vector of cluster proportions. The vectors $C_i \in \{0,1\}^Q$ are one hot encoded, with $C_{iq} = 1$ if node i belongs to cluster q and $C_{iq} = 0$ otherwise. Denoting $\mathbf{C} = (C_1, \dots, C_N)^T \in \mathcal{M}_{N \times Q}(\{0,1\})$ the cluster membership matrix, its probability is given by:

$$p(\mathbf{C} \mid \gamma) = \prod_{i=1}^{N} \prod_{q=1}^{Q} \gamma_q^{C_{iq}}. \tag{5.1}$$

Assuming that the cluster membership matrix $\mathbf{C}$ is given, the nodes are assumed to be independent, and each node i is assumed to be represented by a Gaussian vector Z_i in a $Q-1$ dimensional latent space:

$$Z_i \mid C_{iq} = 1 \overset{i.i.d}{\sim} \mathcal{N}_{Q-1}\left(\mu_q, \sigma_q^2 \mathbf{I}_{Q-1}\right). \tag{5.2}$$

The set of node embeddings is denoted $\mathbf{Z} = (Z_i)_i$ in the rest of the paper, and the set of means and variances are denoted $\boldsymbol{\mu} = (\mu_q)_q$ and $\boldsymbol{\sigma}^2 = (\sigma_q^2)_q$ respectively.

To link the the latent representations of the nodes Z_i, with the block modelling, we rely on the *bijective softmax* transformation, as presented in Xu, Ke, Wang (2014), $h\colon Z_i \in \mathbb{R}^{Q-1} \mapsto \eta_i \in \Delta_Q$ where:

$$\eta_{iq} = \begin{cases} e^{Z_{iq}} / \left(1 + \sum_{r=1}^{Q-1} e^{Z_{ir}}\right) & \text{if } q \in \{1, \dots, Q-1\} \\ 1 / \left(1 + \sum_{r=1}^{Q-1} e^{Z_{ir}}\right) & \text{if } q = Q \end{cases}, \tag{5.3}$$

and we denote $\boldsymbol{\eta} = (\eta_1, \dots, \eta_N)^\top \in \mathcal{M}_{N \times Q}\big((0,1)\big)$. The mapping h aims at encoding the $(Z_i)_i$ into cluster membership probabilities. Eventually, the probability of connection between two nodes is assumed to depend on their respective cluster membership probabilities η_i. Eventually, the probability of connection follows a Bernoulli distribution with parameters depending on η such that:

$$\begin{aligned} p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi}) &= \prod_{i \neq j} p(A_{ij} \mid Z_i, Z_j, \mathbf{\Pi}) \\ &= \prod_{i \neq j} \left(\eta_i^\top \mathbf{\Pi} \eta_j\right)^{A_{ij}} \left(1 - \eta_i^\top \mathbf{\Pi} \eta_j\right)^{1 - A_{ij}}, \end{aligned} \tag{5.4}$$

where $\mathbf{\Pi} = (\pi_{qr})_{1 \leq q,r \leq Q} \in \mathcal{M}_{Q \times Q}\big((0,1)\big)$ is the matrix of probability of connection between clusters.

Finally, the joint distribution of the adjacency matrix, the latent node representations, as well as the cluster memberships can be computed with Equations (5.1) (5.2) and (5.4)

$$p(\mathbf{A}, \mathbf{Z}, \mathbf{C} \mid \mathbf{\Pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}) = p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi}) p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(\mathbf{C} \mid \boldsymbol{\gamma}). \tag{5.5}$$

5.2.1 Link with other models

The mixed-membership stochastic block model The mixed-membership stochastic block model (MMSBM, Airoldi et al., 2008) assumes that for any edge from node i to j, each node plays a specific role identified by a membership indicator $U_{ij} \sim \mathcal{M}_Q(1; \eta_i)$ for the initiator and $V_{ij} \sim \mathcal{M}_Q(1; \eta_j)$ for the receiver. The probabilities are sampled according to $\eta_i \sim \mathcal{D}_Q(\gamma)$. While the authors aimed at maximising the quantity $p(\mathbf{A} \mid U, V, \mathbf{\Pi})$, Deep-LPBM focuses on a marginalisation over U, V of this quantity. Indeed, doing so would result in the probability of connection provided in Equation (5.4). In addition, contrary to Deep-LPBM, this model does not incorporate a node cluster membership variable $\mathbf{C}$.

The latent variable modelling of relational data The latent variable model of relational data (Hoff, 2007) is also related to mixture models introduced at the node level by considering a probability of connection between two nodes as a probit function of $\eta_i \mathbf{\Pi} \eta_i$, where η_i is a vector of free parameters in $\mathbb{R}^Q$ and $\mathbf{\Pi} \in \mathcal{M}_{Q \times Q}(\mathbb{R})$ is a diagonal matrix with entries that may be positives or negatives. First, Deep-LPBM does not assume a specific form of the matrix $\mathbf{\Pi}$ but constrained its values between 0 and 1. Second, this model does not assume a generative assumption for each variable nor introduce a node cluster membership variable allowing to incorporate the clustering within the probabilistic model.

The extremal vertices model for random graph The extremal vertices model for random graph (EVMRG, Daudin, Pierre, Vacher, 2010) also relies on a marginalisation of MMSBM, but does not directly incorporate a node cluster membership variable in the generative model. Moreover, η is not considered as a variable but as a parameter to estimate.

5.2.2 Identifiability of the model

In Daudin, Pierre, Vacher (2010), the authors proposed a model closely related to ours with two major distinctions. First, they assumed that η_i was a parameter to estimate. Second, they do not incorporate a cluster variable in the generative model but instead use η_i as a vector of cluster memberships probabilities and deduce the most likely cluster for each node. Indeed, they rely on Q *extremal hypothetical vertices* as representations of each cluster.

The authors showed that the model introduced above is not identifiable and proposed a way to circumvent this issue. First, the non-identifiability of the model may be illustrated by considering a matrix $\mathbf{H} \in \mathcal{M}_{Q \times Q}(\mathbb{R})$ such that $\mathbf{H}^{-1}$ exists and with its columns summing to one $\mathbf{H}\mathbf{1}_Q = \mathbf{1}_Q$, where $\mathbf{1}_d$ is a d-dimensional vector filled with ones. Denoting $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}\mathbf{H}$ and $\tilde{\mathbf{\Pi}} = \mathbf{H}^{-1}\mathbf{\Pi}(\mathbf{H}^\top)^{-1}$, and assuming that $\tilde{\boldsymbol{\eta}} \geq 0$ and $\tilde{\mathbf{\Pi}} \in \mathcal{M}_{Q \times Q}([0, 1])$, we obtain:

$$\begin{cases} \tilde{\boldsymbol{\eta}}\tilde{\mathbf{\Pi}}\tilde{\boldsymbol{\eta}}^\top = \boldsymbol{\eta}\mathbf{H}\mathbf{H}^{-1}\mathbf{\Pi}(\mathbf{H}^\top)^{-1}\mathbf{H}^\top\boldsymbol{\eta}^\top = \boldsymbol{\eta}\mathbf{\Pi}\boldsymbol{\eta}^\top, \\ \tilde{\boldsymbol{\eta}}\mathbf{1}_Q = \boldsymbol{\eta}\mathbf{H}\mathbf{1}_Q = \boldsymbol{\eta}\mathbf{1}_Q = \mathbf{1}_N. \end{cases}$$

Therefore, $(\tilde{\boldsymbol{\Pi}}, \tilde{\boldsymbol{\eta}})$ and $(\boldsymbol{\Pi}, \boldsymbol{\eta})$ are equivalent since they correspond to the same model (see Daudin, Pierre, Vacher (2010) for the existence of $\mathbf{H}$). As a consequence, for any $(\boldsymbol{\Pi}, \boldsymbol{\eta})$ such that $\boldsymbol{\Pi} \in \mathcal{M}_{Q\times Q}([0,1])$ and $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_N)^\top$ with $\eta_i \in \Delta_Q$ for any i, we consider the classes of equivalence

$$[\boldsymbol{\Pi}, \boldsymbol{\eta}] = \left\{(\tilde{\boldsymbol{\Pi}}, \tilde{\boldsymbol{\eta}}) \in \mathcal{M}_{Q\times Q}([0,1]) \times \mathcal{M}_{N\times Q}(\mathbb{R}^+), \tilde{\boldsymbol{\eta}}\tilde{\boldsymbol{\Pi}}\tilde{\boldsymbol{\eta}}^\top = \boldsymbol{\eta}\boldsymbol{\Pi}\boldsymbol{\eta}^\top \text{ and } \tilde{\boldsymbol{\eta}}\mathbf{1}_Q = \mathbf{1}_N\right\}.$$

The objective is now to first, infer the best equivalence class and second, to choose a set of parameters among the best class of equivalence. For the second step, Daudin, Pierre, Vacher (2010) proposed to choose the parameters $(\boldsymbol{\Pi}^\star, \boldsymbol{\eta}^\star)$ maximising $\sum_{i=1}^N \|\eta_i\|_2^2$ within the class of equivalence. This may be interpreted as favouring the configurations resulting in spiked node cluster membership probabilities. This corresponds to nodes belonging to a single cluster instead of belonging to several. In other words, this may be seen as favouring more interpretable models. Although the generative model we propose is close to the model in Daudin, Pierre, Vacher (2010), we consider $\boldsymbol{\eta}$ as random variables and not as parameters. Hence, we are currently investigating the identifiability of our model. In practice, no difficulty has been encountered.

5.3 Inference

The next section presents the inference as well as the optimisation.

5.3.1 Likelihood

In this chapter, we consider the marginal likelihood of the network for parameter estimation, with latent variables $\mathbf{C}$ and $\mathbf{Z}$, and the set of parameters $\Theta = \{\boldsymbol{\Pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}\}$. From Equations (5.1), (5.2) and (5.4), we can deduce that the marginal log-likelihood is given by:

$$\mathcal{L}(\Theta; \mathbf{A}) = \log p(\mathbf{A} \mid \Theta) = \log\left(\sum_{\mathbf{C}} \int_{\mathbf{Z}} p(\mathbf{A}, \mathbf{C}, \mathbf{Z} \mid \Theta) d\mathbf{Z}\right). \tag{5.6}$$

Unfortunately, this quantity is not tractable since the sum over $\mathbf{C}$ requires to compute Q^N terms. Besides, it involves integrals that cannot be computed analytically because of the softmax function. Therefore, we choose to rely on a variational inference strategy for approximation purposes.

Decomposition of the marginal log-likelihood For any distribution $R(\mathbf{C}, \mathbf{Z})$, the following decomposition holds:

$$\mathcal{L}(\Theta; \mathbf{A}) = \mathscr{L}(R(\cdot); \Theta) + \mathrm{KL}\left(R(\cdot) || p(\mathbf{C}, \mathbf{Z} \mid \mathbf{A})\right), \tag{5.7}$$

where

$$\mathscr{L}(R(\cdot); \Theta) = \mathbb{E}_R\left[\log \frac{p(\mathbf{A}, \mathbf{C}, \mathbf{Z} \mid \Theta)}{R(\mathbf{C}, \mathbf{Z})}\right]. \tag{5.8}$$

Since the Kullback-Leibler divergence is always positive in Equation (5.7), the ELBO $\mathscr{L}(R(\cdot);\Theta)$ is a lower bound of the marginal log-likelihood. Moreover, the closer $R(\cdot)$ is to the posterior distribution of the latent variables, in terms of Kullback-Leibler divergence, the tighter the lower bound. Since the marginal log-likelihood does not depend on $R(\cdot)$, maximizing the ELBO with respect to $R(\cdot)$ is equivalent to minimizing the Kullback-Leibler divergence between $R(\cdot)$ and the posterior distribution. To make the ELBO tractable, we restrict the family of variational distributions by considering a mean-field assumption as well as the following hypotheses:

$$R(\mathbf{C}, \mathbf{Z} \mid \mathbf{A}) = R(\mathbf{C})R(\mathbf{Z} \mid \mathbf{A}), \tag{5.9}$$

$$R(\mathbf{C}) = \prod_{i=1}^{N} R_{\tau_i}(C_i) = \prod_{i=1}^{N} \mathcal{M}_Q(C_i; 1, \tau_i), \tag{5.10}$$

$$R(\mathbf{Z} \mid \mathbf{A}) = \prod_{i=1}^{N} R_\phi(Z_i \mid \mathbf{A}) = \prod_{i=1}^{N} \mathcal{N}_{Q-1}(Z_i; \mu_\phi(\mathbf{A})_i, \sigma^2_\phi(\mathbf{A})_i \mathbf{I}_{Q-1}), \tag{5.11}$$

where $\boldsymbol{\tau} = (\tau_i)_{i=1}^N$ with $\forall i \in \{1, \dots, N\}$, $\tau_i \in \Delta_Q$. Moreover, in Equation 5.11, the mapping $\mu_\phi : \mathcal{M}_{N\times N}(\mathbb{R}) \mapsto \mathcal{M}_{N\times(Q-1)}(\mathbb{R})$ ($\sigma^2_\phi : \mathcal{M}_{N\times N}(\mathbb{R}) \mapsto (\mathbb{R}^+)^N$ respectively) is the mapping normalising the adjacency matrix (with 1 on its diagonal for numeric stability) by its degree, $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I}_N)\mathbf{D}^{-1/2}$, and encoding the normalised adjacency matrix into the approximated posterior means (standard deviations) of the node latent positions. The diagonal matrix $\mathbf{D}$ is filled with $D_{ii} = \sum_{j=1}^{N}(\mathbf{A} + \mathbf{I}_N)_{ij}$. The two mappings μ_ϕ and σ^2_ϕ rely on a GCN parametrised by ϕ. Regarding the encoder of the adjacency matrix, we based our neural network architecture on Kipf, Welling (2016) such that, $\mu_\phi(\mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{W}_0)\mathbf{W}_\mu$ and $\log \sigma^2_\phi(\mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{W}_0)\mathbf{W}_\sigma$, where $\mathrm{ReLU}(x) = (\max(0, x_1), \dots, \max(0, x_F))$ if $x \in \mathbb{R}^F$. The mappings $\mu_\phi(\cdot)$ and $\log \sigma^2_\phi(\cdot)$ share the parameters of the first layer $\mathbf{W}_0 \in \mathcal{M}_{N\times D}(\mathbb{R})$ with $D = 64$ in all the experiments we carried out, and $\mathbf{W}_\mu, \mathbf{W}_\sigma \in \mathcal{M}_{D\times(Q-1)}(\mathbb{R})$. For the sake of brevity, we will take the exponential of the encoder of the log variance and consider $\sigma^2_\phi(\cdot)$. Thus, the ELBO can be decomposed as follows:

$$\begin{aligned}
\mathscr{L}(R(\cdot);\Theta) =& \mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi})\right] \\
&+ \mathbb{E}_R\left[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] - \mathbb{E}_R\left[\log R(\mathbf{Z} \mid \mathbf{A})\right] \\
&+ \mathbb{E}_R\left[\log p(\mathbf{C} \mid \boldsymbol{\gamma})\right] - \mathbb{E}_R\left[\log R(\mathbf{C})\right]. \\
=& \underbrace{\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi})\right]}_{\text{Reconstruction loss}} \underbrace{- \mathrm{KL}(R(\mathbf{Z} \mid \mathbf{A}) || p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})) - \mathrm{KL}(R(\mathbf{C}) || p(\mathbf{C} \mid \boldsymbol{\gamma}))}_{\text{Regularising term}}.
\end{aligned} \tag{5.12}$$

Hence, the ELBO is given by:

$$\begin{aligned}\mathscr{L}(R(\cdot);\Theta) = &\sum_{i,j=1}^{N}\left\{A_{ij}\mathbb{E}_R\left[\log \eta_i^\top \mathbf{\Pi}\eta_j\right] + (1-A_{ij})\mathbb{E}_R\left[\log\left(1-\eta_i^\top\mathbf{\Pi}\eta_j\right)\right]\right\}\\ &- \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\,\mathrm{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right)\\ &- \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\log\frac{\tau_{iq}}{\gamma_q},\end{aligned} \tag{5.13}$$

where

$$\mathrm{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right) = \log\frac{\sigma_q^{(Q-1)}}{\sigma_\phi(\mathbf{A})_i^{(Q-1)}} - \frac{Q-1}{2} + \frac{(Q-1)\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2}{2\sigma_q^2}.$$

The computation of each term in Equation (5.12) is detailed in Appendix 7.4.1. To optimise the ELBO, we propose here to alternate between closed-form updates and stochastic gradient descent steps thanks to the results presented in the next section.

5.3.2 Optimisation

We begin this section by providing analytical updates for the parameters $\boldsymbol{\tau}$, γ, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

Analytical updates with respect to the variational parameter $\boldsymbol{\tau}$ Using the first-order condition, it is possible to derive an analytical update for the variational parameters $\boldsymbol{\tau}$, as detailed in the following proposition.

Proposition 8. *Let $\mathscr{L}(R(\cdot);\Theta)$ denote the ELBO, as describe in Equation* (5.12)*. The first-order conditions with respect to $\boldsymbol{\tau} = (\tau_i)_i$ give the following updates for any node i and cluster q*

$$\tau_{iq} = \frac{\gamma_q e^{-\mathrm{KL}_{iq}}}{\sum_{r=1}^{Q}\gamma_r e^{-\mathrm{KL}_{ir}}}. \tag{5.14}$$

Proof. See Appendix 7.4.2. □

One way to interpret Equation (5.14) is to note that for a given set of parameters and a node i, the optimal probability of node cluster membership with respect to cluster q decreases exponentially fast with the Kullback-Leibler divergence between the variational distribution of Z_i and the distribution of the representation of cluster q. Each term is scaled by the proportion of the corresponding cluster.

Analytical updates with respect to the model parameters γ, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ The first-order conditions applied to the ELBO with respect to the model parameters give closed-form updates stated in the following proposition.

Proposition 9. *Let $\mathscr{L}(R(\cdot);\Theta)$ denote the ELBO described in Equation* (5.12)*. The first-order conditions with respect to γ, $(\mu_q)_q$ and $(\sigma_q)_q$ give the following updates:*

$$\gamma_q = \frac{1}{N}\sum_{i=1}^{N}\tau_{iq}, \tag{5.15}$$

$$\mu_q = \left(\sum_{i=1}^{N}\tau_{iq}\right)^{-1}\sum_{i=1}^{N}\tau_{iq}\mu_\phi(\mathbf{A})_i, \tag{5.16}$$

$$\sigma_q^2 = \left((Q-1)\sum_{i=1}^{N}\tau_{iq}\right)^{-1}\sum_{i=1}^{N}\tau_{iq}\left((Q-1)\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2\right). \tag{5.17}$$

Proof. See Appendices 7.4.2. □

The update of γ corresponds to the approximated posterior expectation of the cluster proportions in the network. On the one hand, the optimal μ_q is given by the posterior means of $(Z_i)_i$ weighted by each node contribution to the corresponding cluster. On the other hand, the optimal variance σ_q^2 is given by the sum of two terms. The first one corresponds to the weighted mean of the posterior variances of the nodes. The second one corresponds to the weighted mean of the squared Euclidean distances between the node posterior means and μ_q. In other words, the variances incorporate both the uncertainty about the posterior variance, illustrated by the $\sigma_\phi^2(A)_i$ terms, as well as the variance of node embeddings, corresponding to the $\|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2$ terms.

Stochastic gradient descent One of the core difficulties in this model is the estimation of the parameters $\boldsymbol{\Pi}$ and the variational parameters ϕ. This is due to the impossibility to compute $\mathbb{E}_R\left[\log p(A \mid \mathbf{Z}, \boldsymbol{\Pi})\right]$ because of the intractable expectations $\mathbb{E}_R\left[\log(\sum_{q,r}\eta_{i,q}\eta_{j,r}\pi_{qr})\right]$. To overcome this issue, we rely on a stochastic gradient descent algorithm using the reparametrisation trick (Kingma, Welling, 2014; Rezende, Mohamed, Wierstra, 2014) enabling easy computations of the gradient estimates with low variances.

Ideally, we would like to use the same computations as in the previous section. For instance, we would like to compute the following partial derivates of the ELBO with respect to ϕ:

$$\begin{aligned}\frac{\partial}{\partial\phi}\mathscr{L}(R(\cdot);\Theta) =& \frac{\partial}{\partial\phi}\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\Pi}) + \log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma}) - \log R(\mathbf{Z} \mid \mathbf{A})\right] \\ =& \frac{\partial}{\partial\phi}\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\Pi})\right] - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\frac{\partial}{\partial\phi}\mathrm{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right),\end{aligned} \tag{5.18}$$

Unfortunately, the expectation of the term on the left-hand side of Equation (5.18) is taken with respect to $R(\cdot)$, which depends on ϕ. Therefore, the derivation of this quantity is not straightforward. Thanks to Kingma, Welling (2014) and Rezende, Mohamed, Wierstra (2014), this difficulty can be tackled by using the reparametrisation trick. In particular, let ε_i be a centred, normalised and $(Q-1)$ dimensional Gaussian vector. Hence, the vectors Z_i and $\mu_\phi(\mathbf{A})_i \oplus [\sigma_\phi(\mathbf{A})_i \odot \varepsilon_i]$ have the same distribution. Therefore, the expectation can be taken with respect to $\boldsymbol{\epsilon} = (\varepsilon_i)_{i=1\dots,N}$ which gives:

$$\frac{\partial}{\partial\phi}\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Pi})\right] = \frac{\partial}{\partial\phi}\mathbb{E}_\varepsilon\Big[\log p\left(\mathbf{A} \mid \mu_\phi(\mathbf{A}) \oplus [\sigma_\phi(\mathbf{A}) \odot \boldsymbol{\epsilon}], \boldsymbol{\Pi}\right)\Big],$$

where $\mu_\phi(\mathbf{A}) \oplus [\sigma_\phi(\mathbf{A}) \odot \boldsymbol{\epsilon}] = \Big(\mu_\phi(\mathbf{A})_i \oplus [\sigma_\phi(\mathbf{A})_i \varepsilon_i]\Big)_{i=1,\dots,N}$, $\odot$ denotes the Hadamard product and $\oplus$ the element-wise sum. A Monte-Carlo estimate of this quantity is derived by sampling S centred and reduced $(Q-1)$-dimensional Gaussian vectors $\epsilon_i^{(s)}$ with $s = 1, \dots, S$, $i = 1, \dots, N$ and with $\boldsymbol{\epsilon}^{(s)} = (\varepsilon_i^{(s)})_{i=1,\dots,N}$. Plugging it back into (5.18) gives the following estimate:

$$\begin{aligned}\frac{\partial}{\partial\phi}\mathscr{L}(R(\cdot);\Theta) \approx & \frac{1}{S}\sum_{s=1}^{S}\Big[\frac{\partial}{\partial\phi}\log p\left(\mathbf{A} \mid \mu_\phi(\mathbf{A}) \oplus \left[\sigma_\phi(\mathbf{A}) \odot \boldsymbol{\epsilon}^{(s)}\right], \boldsymbol{\Pi}\right)\Big] \\ & - \sum_{i=1}^{N}\sum_{q=1}^{Q}\tau_{iq}\frac{\partial}{\partial\phi}\,\mathrm{KL}_{iq}^{Z}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi^2(\mathbf{A})_i, \mu_q, \sigma_q^2\right).\end{aligned}$$

Thanks to the low variances of the gradients estimated with the reparametrisation trick and to avoid increasing the computations, we use a sample size $S = 1$, as advised in the VAE literature. In addition, the computation of the partial derivatives is implemented with Pytorch automatic differentiation framework (Paszke et al., 2019) to take full advantage of the computational efficiency of GPUs. Moreover, we rely on the Adam optimiser (Kingma, Ba, 2014) to carry out the stochastic gradient descent with a learning rate of 0.002 and a weight decay of 0.01 for the optimiser of $\boldsymbol{\Pi}$ and ϕ. Since π_{qr} lies between 0 and 1, we cannot perform gradient descent directly on this parameter. To solve this, we use the bijective transformation $f\colon x \in (0,1) \mapsto \tan(\pi(x-0.5)) \in \mathbb{R}$ and the inverse function $f^{-1}\colon y \in \mathbb{R} \mapsto 0.5 + \pi^{-1}\arctan(y) \in (0,1)$ to perform the gradient descent on the unconstrained variable $\tilde{\pi}_{qr} = f(\pi_{qr}) \in \mathbb{R}$ and switch back to $\pi_{qr} = f^{-1}(\tilde{\pi}_{qr})$ to compute the ELBO.

5.3.3 Initialisation strategy

We advocate for a tailored initialisation to avoid the regularising term to prevent discovering patterns at the beginning of the estimation, when the clusters have not been estimated yet, nor the encoder has been initialised. Indeed, we experienced in practice a collapse of the clusters toward a single cluster assignment when using the regularising term from the start of the optimisation in our objective function. Therefore, we first run a stochastic block model to obtain an estimation of $\boldsymbol{\Pi}$. Then, the GCN parameters are estimated to

minimise the reconstruction loss, with $\mathbf{\Pi}$ fixed, such as to obtain an encoder matching with $\mathbf{\Pi}$. During this step, we use a learning rate of 0.1. Since we optimised only with respect to the reconstruction loss, the cluster parameters have not been initialised at this stage. The value of the parameters related to the clustering is set after the GCN initialisation, using the updates described in Proposition 9.

Algorithm 6: Algorithm summarising the optimisation steps to estimate Deep-LPBM parameters, $\theta^{(t)} = \{\gamma^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}\}$

Input: $t = 0$;
Initialise $\mathbf{\Pi}^{(0)}$ and $\tau^{(0)}$ with SBM;
Initialise $\phi^{(0)}$ by maximising the reconstruction loss with $\mathbf{\Pi}^{(0)}$ fixed;
Initialise $\gamma^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{(0)}$ with Proposition 9;
while $\mathscr{L}(\theta^{(t)}; R^{(t)})$ *and* $\mu_\phi^{(t)}$ *have not converged* **do**
  Update $\phi^{(t)}$ by minimising $-\mathscr{L}(\theta^{(t+1)}; R_{\tau^{(t+1)}}, R_\phi)$ with a SGD algorithm;
  Update $\gamma^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}$ with Proposition 9;
  Update $\tau^{(t)}$ with Proposition 8;
  Compute $\mathscr{L}(\theta^{(t+1)}; R^{(t+1)})$;
  $t = t + 1$;
end

5.4 Evaluation on synthetic datasets

In real-life datasets, quantifying the relevance of a network representation as well as node partitions is a challenging task since no partition of the node exists. Therefore, to assess Deep-LPBM ability to cluster the data, it is necessary to compare its clustering results with a ground truth on synthetic data. First, we present the network structures used in this section to evaluate our methodology. Second, we illustrate the information provided by Deep-LPBM results on the challenging disassortative structures. Third, a comparison of Deep-LPBM representational capacity with VGAE and Deep-LPM on the three proposed connectivity structures is exposed. To end this section, we give a quantitative assessment of the clustering results on the community, the hub and the disassortative structures and compare our results with Deep-LPM clustering and the K-means algorithm applied on VGAE embeddings.

5.4.1 Presentation of network structures and Deep-LPBM results

To assess Deep-LPBM capacity to represent different network topologies, we evaluate our methodology on three different structures, all composed of 100 nodes and 5 clusters, such that:

- **the community structure** is composed of nodes divided into 5 clusters with sizes sampled according to a multinomial distribution with 100 draws and an equiprobability for each cluster to be drawn, such that two nodes of the same cluster are

connected with probability 0.5, while two nodes from different clusters are connected with probability 0.01

- **the disassortative structure** splits the nodes into 5 clusters with sizes sampled according to a multinomial distribution with 100 draws and an equiprobability for each cluster to be drawn. Two nodes from the same cluster have a probability of 0.01 to be connected while two nodes from different clusters are connected with a probability of 0.5

- **the hub structure** contains 4 communities, and 1 hub, with sizes sampled according to a multinomial distribution with 100 draws and a probability two times superior to be sampled from a community than from the hub. An edge between a node in the hub and any other node exists with probability 0.5.

We use those specific connectivity patterns to inspect different properties of our algorithm. First, the community structure tests the ability to group together nodes having a high probability of being connected, corresponding to the main focus of the standard positional models. Second, the disassortative graph evaluates the capability to group together, in the latent space, non-connected nodes, which cannot be captured by canonical positional models. Third, the hub structure appraises the relevance of the obtained representation when dealing with a cluster connected to the entire network.

Learning node representations

These two sections highlight the flexibility of Deep-LPBM by analysing both the block modelling estimates as well as the network representation. We start with the latter, with the evolution of the representation during the optimisation presented in Figure 5.1.

The results provided in this section are obtained by fitting Deep-LPBM on a disassortative network and projecting the estimated embeddings in $\mathbb{R}^2$ with a t-sne algorithm (Van der Maaten, Hinton, 2008). First, we observe an efficient separation of the clusters which cannot be obtained with a similarity-based decoder since the probability of connection would increase with the correlation of the node embeddings. Therefore, the latent space would not be able to show any structure, as depicted in Figure 5.3. On the contrary, the model we propose is capable of imposing a structure on the variational distribution such as to obtain a node latent space matching with the connectivity patterns of the network.

Block modelling information

Conversely to the previous section, Figure 5.2 can only be obtained by a block modelling strategy. The connectivity structure of the graph, captured by the matrix $\mathbf{\Pi}$, is displayed in Figure 5.2 as well as the associated meta-graph. A meta-graph is a network composed of nodes representing the estimated clusters with a size proportional to the estimated cluster proportions γ. The edge widths of the meta-graph are proportional to $\mathbf{\Pi}$ and inform us about the probability of connection between the corresponding clusters. In large networks,

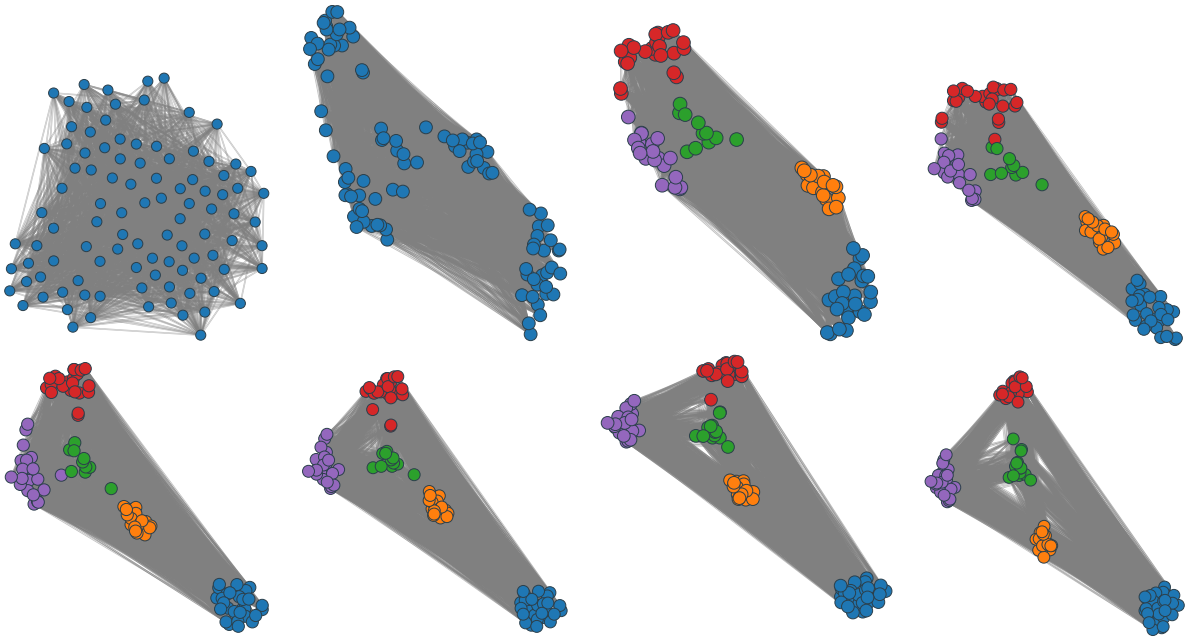


Figure 5.1: Evolution of the node embeddings during the estimation of Deep-LPBM on a disassortative network. The networks at the top (at the bottom respectively) correspond, from left to right, to the embeddings at the start of the GCN initialisation, the end of the GCN initialisation, iteration 100 and iteration 200 of Deep-LPBM (iteration 500, 1000, 1500, 2000 respectively). The embeddings were projected in $\mathbb{R}^2$ using the t-sne algorithm.

	1	2	3	4	5
1	0.01	0.5	0.51	0.52	0.49
2	0.5	0.0	0.48	0.54	0.52
3	0.51	0.48	0.0	0.49	0.52
4	0.52	0.54	0.49	0.0	0.5
5	0.49	0.52	0.52	0.5	0.02

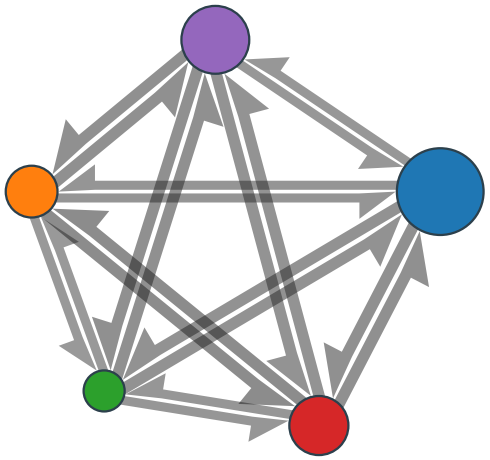


Figure 5.2: Meta-network based on Deep-LPBM results with an underlying disassortative structure. On the left-hand side, we provide the estimation of the connection probability matrix $\mathbf{\Pi}$. On the right-hand side, the meta-network is composed of nodes representing the clusters, their size is proportional to the corresponding estimated cluster proportion γ and the edge widths are proportional to $\mathbf{\Pi}$. A threshold has been set such that edges corresponding to a probability of connection lesser or equal to 0.02 are not displayed.

this type of representation presents the advantage of being easily interpretable as well as focusing on the generative understanding of the network. Here, it is clear that the clusters are highly connected one to another but nodes from the same cluster are poorly connected together. This summary of information is not straightforward in canonical positional models

5.4.2 Representational power on three network structures

In Figure 5.3, both the VGAE and the Deep-LPBM efficiently render a latent space capturing the community structure. In addition, the hub is also well depicted by both models. Eventually, the structure presenting the biggest trouble is the disassortative graph. Let us recall that the decoder of a variational graph autoencoder models the probability of connection between two nodes with a sigmoid function applied to the cosine similarity between their respective latent positions. Therefore, it necessarily fails to capture the disassortative structure of the network. Hence, two nodes in the same cluster, that have a low probability of connection cannot be represented similarly, and thus cannot be close in the latent space. Conversely, Deep-LPBM is able to translate the connectivity pattern into the position of the nodes, such that nodes of the same cluster, poorly connected together, are close in the latent space.

5.4.3 Clustering evaluation on synthetic data

In this section, we use the same network structures as the one described in the previous section. We aim to assess the clustering performance of our methodology and compare it with the deep latent position model (Deep-LPM) that relies on node embedding similarity as a decoder. We also provide a baseline with the variational graph autoencoder used to estimate the node embeddings and a K-means algorithm fitted on these embeddings to obtain partitions of the nodes. The benchmark is performed on networks with 100 nodes and $Q = 5$ clusters. The results are displayed in Table 5.1. First, we note the efficiency of Deep-LPBM on communities and hubs, reaching an ARI of 1 in both cases. It does not degrade the good performance of its competitors on these architectures, with ARIs of 1 and 0.97 for the VGAE and K-means, and an ARI of 1 on both structures for Deep-LPM. However, these competitors are not able to represent any connectivity structure in the disassortative case. In particular, as shown in Figure 5.3 for the VGAE, they cannot find relevant node clusters in this setting and end up with an ARI of 0.02 and 0. On the contrary, our methodology reaches an ARI of 0.8, much higher than its competitors.

5.5 Conclusions

This chapter introduced a new methodology combining a block model with a deep latent position model. By modifying the edge distribution and marginalising over a bijective transformation of the node latent representations, we managed to use the node embed-

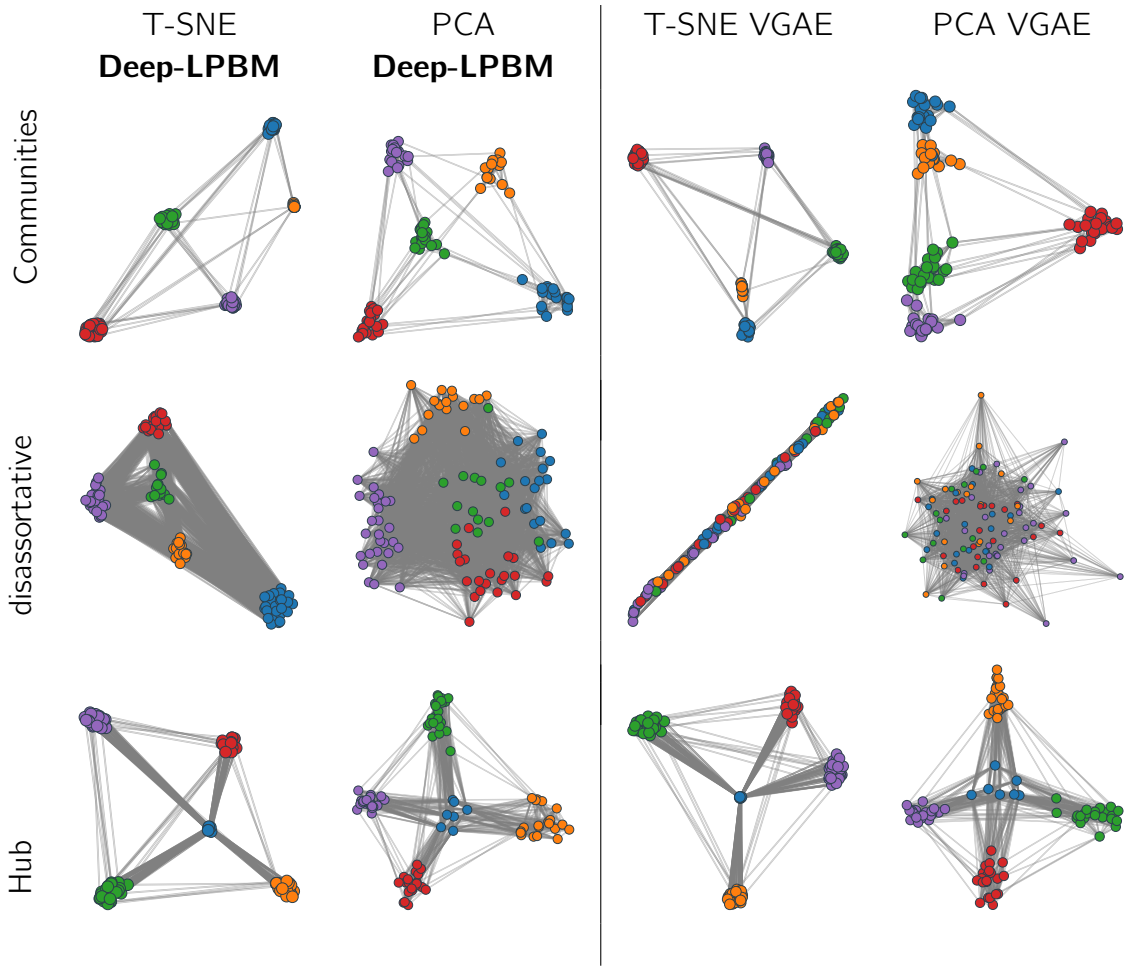


Figure 5.3: Example of three networks with a latent structure composed of (top to bottom) five communities, a disassortative network with five clusters and four communities with a hub. On the left-hand side, the networks represent the results obtained fitting Deep-LPBM and using a t-sne projection (left) as well as a PCA (right). On the right-hand side, the variational graph auto encoder is fitted and projected with the same two methods. Each node colour corresponds to its true cluster membership.

	Communities	disassortative	Hub
VGAE + Kmeans	1.00 ± 0.01	0.02 ± 0.02	0.97 ± 0.06
Deep LPM	1.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
Deep LPBM	1.00 ± 0.00	0.80 ± 0.08	1.00 ± 0.00

Table 5.1: Comparison of the partition obtained by a K-means algorithm applied on the VGAE node embeddings, the Deep-LPM and Deep-LPBM partitions. For each network, each model was run 10 times and the one corresponding to the highest ELBO was kept as a result. The mean and standard deviations were obtained over 10 networks for each graph structure.

dings as cluster probability memberships. This framework bridges the gap between deep block modelling and deep latent position model. Consequently, we obtain richer results, simultaneously providing a high-level meta-network, summarising the information, as well as a full network representation, allowing node specificity possibly to be incorporated as they may hold crucial detail. Our methodology is based on the encoder of a graph variational autoencoder combined with a novel block model decoder. A variable incorporates the clustering within the model instead of performing the clustering a posteriori on the node embeddings. This allows to obtain node cluster probability memberships. The optimisation of our model combines analytical formulas as well as stochastic gradient descent. On the network structures studied in this section, namely the community, the hub and the disassortative topologies, our methodology rightfully translated the network salient information into the latent space. In addition, the clustering results are competitive with the state-of-the-art Deep-LPM. This work is currently under study. In particular, a more extensive benchmark will be provided in a later work as well as an analysis of a real-world dataset.

Chapter 6

Conclusion and perspectives

...

6.1 Summary of the contributions

To conclude this manuscript, we provide a summary of the contributions exposed in the previous chapters.

Chapter 1 is dedicated to the introduction of the notions of network and textual data. We advocated for the necessity to understand the patterns responsible for the generation of the network as well as the texts. In particular, we stressed the relevance of probabilistic modelling to meet this goal.

Chapter 2 provided the mathematical background necessary for the rest of the manuscript.

Chapters 3 and 4 presented two models aiming at explaining the patterns responsible for the generation of networks with textual edges, each model focusing on a specific aspect. In Chapter 3, the focus was set on finding any type of connectivity patterns, by introducing the embedded topic in the stochastic block model (ESTBM), allowing an auto-aggregation of documents. This allowed us to simultaneously perform node clustering, based on a variational inference strategy, as well as topic modelling, by relying on a neural topic model able to incorporate pre-trained embeddings holding semantic meaning. To assess the relevance of the proposed model selection criterion and ETSBM clustering performances, we carried out an extensive benchmark on synthetic data as well as an analysis of a Twitter network dataset.

In Chapter 4, we concentrated on the possibility of obtaining visualisations of the networks incorporating textual information by presenting the deep latent topic model (Deep-LPTM). This methodology is based on a variational graph autoencoder combined with a neural topic model to cluster the nodes of a network as well as render a meaningful representation of the graph. We derived a model selection criterion, namely the integrated classification and latent likelihood, and provided a thorough evaluation of our methodology on synthetic data as well as the Enron emails dataset.

Eventually, in Chapter 5, we studied networks without textual edges. We provided an approach capable of performing node clustering, using a block modelling approach, as well as rendering a meaningful visualisation of the entire network. The variational distribution is based on graph neural networks, allowing us to obtain complex node representations, and to impose a structure on the variational distribution to capture relevant connectivity patterns, thanks to the novel decoder proposed in our work.

All the models presented in this manuscript incorporated the latest advancement in topic modelling for the embedding topics in the stochastic block model, and the latest advancement in graph neural network for the deep latent position topic model, and the deep latent position block model. Interestingly, new architectures of graph neural networks can be used without modification of the model. All these approaches share the property to combine stochastic gradient descent, which allows to consider complex variational distributions, as well as closed-form updates. We showed that such complexity resulted in improvements in clustering performances and ameliorations of visualisation and comprehension of the results.

Our first contribution was shared with the community through a publication in an international peer-reviewed journal

- **Embedded topics in the stochastic block model**, joint work with Pierre Latouche and Charles Bouveyron, *Statistics and Computing (2023)*

the second contribution is under revision,

- **The Deep Latent Position Topic Model for Clustering and Representation of Networks with Textual Edges**, joint work with Pierre Latouche and Charles Bouveyron, *Pre-print hal-04068665 (2023)*.

and our latest work is on-going

- **The Deep Latent Position Block Model**, joint work with Pierre Latouche and Charles Bouveyron.

The Python implementations of our methodologies have been made public. ETSBM can be found at https://plmlab.math.cnrs.fr/rboutin/etsbm_package, and Deep-LPTM at https://plmlab.math.cnrs.fr/rboutin/deeplptm_package.

6.2 Perspective

This last section presents some of the perspectives of research that we find interesting and we believe are worth studying.

6.2.1 Mini-batch training in network with textual edges

We would like to speed up the optimisation procedure of Deep-LPTM to be able to scale to large networks. Several approaches have been proposed in the literature to apply stochastic gradient descent to estimate parameters of network probabilistic models, such as in Gopalan, Blei (2013). Two main differences appear with our procedure. First, one comes from the model itself since Deep-LPTM also analyses textual data and as such, requires additional care to ensure that the parameters ρ and α, as well as ϕ_Y, are correctly optimised. Second, we relied on analytical formulas to update the parameters $\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{m}, \mathbf{s}$. In a full-batch setting, it seemed beneficial as it permitted to make larger steps with regard to these parameters, and hopefully optimise the ELBO faster. In a mini-batch setting, using analytical formulas would imply that at each iteration, the new values of the parameters are based only on the subsampled graph. We believe that this would hurt the optimisation. Therefore, we propose to adopt a fully gradient-based optimisation. We begin by recalling the ELBO expression:

$$\begin{aligned}\mathscr{L}(R(\cdot);\Theta) =& \mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right] + \mathbb{E}_R\left[\log p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right] + \mathbb{E}_R\left[\log p(\mathbf{C} \mid \gamma)\right] \\ &+ \mathbb{E}_R\left[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] + \mathbb{E}_R\left[\log p(\mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \mathbf{m}, \mathbf{s})\right] - \mathbb{E}_R\left[\log R(\mathbf{C})\right] \\ &- \mathbb{E}_R\left[\log R(\mathbf{Z} \mid \mathbf{A})\right] - \mathbb{E}_R\left[\log R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W})\right]. \end{aligned} \tag{6.1}$$

Instead of using the entire graph to compute the ELBO, it may be possible to sample nodes, edges, or a combination of both to obtain an efficient optimisation as well as memory gain. To formalise this, we base our notations on Gopalan, Blei (2013). Hence, the sampling density of an edge is denoted $g(i,j)$ for any pair of nodes (i,j). Contrary to Gopalan, Blei (2013) that only deals with a network without edge information, we also have to incorporate the sampling strategy into the topic modelling terms. Sampling a pair of nodes (i,j) according to g, we have:

$$\begin{aligned}\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right] &= \mathbb{E}_g\left[\frac{1}{g(i,j)}\mathbb{E}_R\left[\log p(A_{i,j} \mid Z_i, Z_j, \kappa)\right]\right], \\ \mathbb{E}_R\left[\log p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right] &= \mathbb{E}_g\left[\frac{1}{g(i,j)}\mathbb{E}_R\left[\log p(W_{i,j} \mid A_{i,j}Y_{i,j}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right]\right].\end{aligned}$$

Hence, denoting (I,J) a random vector with density g, we can reorder the terms of the ELBO to make explicit the expectation with regard to the sampling density $g(I,J)$,

and obtain an unbiased expectation of the ELBO:

$$\begin{aligned}\mathscr{L}(R(\cdot);\Theta) = &\sum_{i=1}^{N} \mathbb{E}_R\left[\log p(C_i \mid \gamma)\right] - \mathbb{E}_R\left[\log R(C_i)\right] \\ &+ \sum_{i=1}^{N} \mathbb{E}_R\left[\log p(Z_i \mid C_i, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] - \mathbb{E}_R\left[\log R(Z_i \mid \mathbf{A})\right] \\ &+ \mathbb{E}_g\left[\frac{1}{g(I,J)}\left(\mathbb{E}_R\left[\log p(A_{I,J} \mid Z_i, Z_j, \kappa)\right] + \mathbb{E}_R\left[\log p(W_{I,J} \mid A_{I,J}, Y_{I,J}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right]\right)\right] \\ &+ \mathbb{E}_g\left[\frac{1}{g(I,J)}\left(\mathbb{E}_R\left[\log p(Y_{I,J} \mid \mathbf{A}, C_I, C_J, \mathbf{m}, \mathbf{s})\right] - \mathbb{E}_R\left[\log R(Y_{I,J} \mid \mathbf{A}, W_{I,J})\right]\right)\right].\end{aligned} \tag{6.2}$$

Therefore, we can compute an unbiased estimator of the gradient with respect to m_{qr}. To avoid cumbersome notations, the last term in Equation (6.2) will be denoted:

$$\mathrm{KL}_{IJ}^{qr} = \mathrm{KL}\Big(\mathcal{N}_K\left(\mu_{\phi_Y}(W_{I,J} \mid \mathbf{A}), \sigma_{\phi_Y}(W_{I,J} \mid \mathbf{A})\right) \,||\, \mathcal{N}_K\left(m_{qr}, s_{qr}^2 \mathbf{I}_K\right)\Big),$$

and therefore,

$$\mathbb{E}_R\left[\log \frac{p(Y_{I,J} \mid \mathbf{A}, C_I, C_J, \mathbf{m}, \mathbf{s})}{R(Y_{I,J} \mid \mathbf{A}, W_{I,J})}\right] = -\sum_{q,r=1}^{Q} \tau_{Iq}\tau_{Jr}A_{I,J}\,\mathrm{KL}_{IJ}^{qr}\,. \tag{6.3}$$

To illustrate the new optimisation strategy, we detail the computations to update the parameter m_{qr}. Noting that the only term depending on m_{qr} in Equation (6.2) is the one detailed in Equation (6.3), the partial derivative of the ELBO with respect to m_{qr} is given by:

$$\frac{\partial}{\partial m_{qr}}\mathscr{L}(R(\cdot);\Theta) = -\mathbb{E}_g\left[\frac{\tau_{Iq}\tau_{Jr}A_{I,J}}{g(I,J)}\frac{\partial}{\partial m_{qr}}\mathrm{KL}_{IJ}^{qr}\right] = -\mathbb{E}_g\left[\frac{\tau_{Iq}\tau_{Jr}A_{I,J}}{g(I,J)}\frac{1}{\sigma_{qr}^2}(m_{qr} - \mu_{\phi_Y}(W_{I,J}))\right].$$

The last expectation can easily be estimated by sampling edges according to g. However, a more efficient strategy, taking into account the sparsity of the graph as well as the neighbours of the nodes to pass local information, may favour the optimisation. The next section presents the strategy advised in Gopalan, Blei (2013) and adapts it to our case.

Sampling sets of edges for better and faster optimisation

Instead of sampling edges at random, more advanced sampling strategies may be beneficial to gather information provided by node neighbours while keeping each iteration fast enough. To this aim, Gopalan, Blei (2013) proposed four sampling strategies and advocated for the use of the *stratified random node sampling*. It consists in defining, for each node i, the *"link set"*, composed of edges corresponding to the connections to which the node is connected, and the *"non-link sets"*, split into m subsets since the number of non-links may be very high in sparse networks. At each iteration, a node is sampled at random.

Then, either the *"link-set"* or *"non-link set"* is sampled at random. If the *"non-link set"* is selected, one of the m subsets is sampled uniformly. Consequently, the density of the proposed sampling scheme is given by:

$$h(S) = \begin{cases} \frac{1}{2N} & \text{if the } \textit{link set} \text{ is sampled,} \\ \frac{1}{2mN} & \text{if one of the } m \textit{ non-link set} \text{ is sampled.} \end{cases}$$

Therefore, a non-biased estimate of the gradient can be computed as:

$$\frac{\partial}{\partial m_{qr}} \widehat{\mathscr{L}}(R^{(t)}(\cdot); \Theta^{(t)}) = -\frac{1}{h(S_t)} \sum_{(i,j)\in\mathcal{S}^{(t)}} \frac{\tau_{iq}^{(t)} \tau_{jr}^{(t)} A_{ij}}{\sigma_{qr}^{(t)2}} (m_{qr}^{(t)} - \mu_{\phi_Y^{(t)}}(W_{ij})).$$

Remark If we consider a non-directed graph, each edge is present in 2 sets. Therefore, it is necessary to multiply the expected term of the ELBO with respect to h by $1/2$, see Gopalan, Blei (2013).

From there, plugging this estimate into a gradient descent algorithm with an adaptative stepsize υ_t provides a stochastic gradient descent algorithm with update at the t-th iteration:

$$m_{qr}^{(t+1)} = m_{qr}^{(t)} + \upsilon_t \frac{\partial}{\partial m_{qr}} \widehat{\mathscr{L}}(R^{(t)}(\cdot); \Theta^{(t)}),$$

with for all t, $\upsilon_t > 0$, $\sum_{t\geq 0} \upsilon_t = \infty$ and $\sum_{t\geq 0} \upsilon_t^2 < \infty$ to ensure convergence (Robbins, Monro, 1951). A common choice in variational inference is to set $\upsilon_t = (t + \upsilon_0)^{-\kappa}$, with $\kappa \in (0.5, 1]$ (Hoffman, Blei, et al., 2013). This can be applied to other parameters. For instance, parameters $\boldsymbol{\tau}$ depend on all other nodes, as stated in Proposition 6. Therefore, we advocate for also using a stochastic gradient descent algorithm to optimise it, as described above, to avoid basing the value of $\boldsymbol{\tau}$ only on the sampled sub-graph.

6.2.2 Implementation of fast and flexible packages

As stated above, the implementations accompanying our contributions have been made publicly accessible. Since advancements in graph neural networks are thriving, we advocate for the development of flexible implementations, allowing to switch from one graph architecture to another. To this end, we emphasise the possibilities offered by *Pytorch Geometric* Python package (Fey, Lenssen, 2019). It provides a unifying framework to compare new graph neural network encoders and decoders, as well as Python class to make it easy to tweak the architecture of the graph neural network. From a statistical perspective, it offers the possibility to easily change the parametrisation of the variational distribution without necessarily modifying the generation assumptions of the graph. Going even further, this may motivate a deeper investigation, for a given methodology, about the amount of information captured that is due to the modelling assumptions and the amount that can be attributed to the encoding hypothesis, or in other words, to the variational distribution.

6.2.3 Advancements in Deep-LPBM

To end this section concerning our perspectives of research, let us present improvements under consideration concerning Deep-LPBM. First, a model selection criterion adapted to our objective of simultaneously obtaining a network visualisation, as well as performing node clustering, is presented. Second, an extensive benchmark is proposed, with a set of experiments on synthetic data as well as a real-world use-case.

Selection model criterion for Deep-LPBM

A selection model criterion needs to be assessed for Deep-LPBM. In Liang et al. (2022), the authors suggest to use the ELBO as VAEs have been known to be self-regularising. However, Chapter 4 has highlighted the benefit of incorporating the latent variables in the model selection criterion. Another natural candidate would be to evaluate the quantity

$$\log p(\mathbf{A}, \mathbf{Z}, \mathbf{C} \mid \mathcal{M}, Q).$$

Since $\mathbf{Z}, \mathbf{C}$ are not observed and this quantity is intractable, we can rely on BIC-like approximation to estimate this quantity with $IC2L(\mathcal{M}, Q, \hat{\mathbf{Z}}, \hat{\mathbf{C}})$ such that:

$$IC2L(\mathcal{M}, Q, \hat{\mathbf{Z}}, \hat{\mathbf{C}}) = \max_{\theta} \log p(\mathbf{A}, \hat{\mathbf{Z}}, \hat{\mathbf{C}} \mid \theta, \mathcal{M}, Q) - \Omega(\mathcal{M}, Q), \tag{6.4}$$

where $\Omega(\mathcal{M}, Q, K, P) = \frac{1}{2}\log(N(N-1)) + \frac{QP+Q}{2}\log(N) + \frac{Q-1}{2}\log(N)$. A proof based on the same arguments as in Appendix 7.3.3 may be derived.

Eventually, a comparison between the ELBO, the ICL (Daudin, Picard, Robin, 2008), and the IC2L, proposed in this manuscript, would help to understand the differences between these quantities. In particular, it would be interesting to establish which one of those criterion best capture the structure of the latent space and thus select the model with the best representational power.

Evaluation of Deep-LPBM clustering robustness to noise

We recall that Deep-LPBM aims at clustering nodes using a block modelling strategy combined with a latent position approach. To make sure of the efficiency of Deep-LPBM to cluster nodes, we propose to compare the ARI obtained by fitting Deep-LPBM against both a VGAE followed by a K-means algorithm on the node embeddings as well as Deep-LPM clustering results. To this aim, noise can be added to the structures proposed in Section 5.4.1, namely the community structure, the disassortative one as well as the hub. In particular, the underlying topology can be more difficult to detect using a noise control variable ϵ as presented in Table 6.1.

Communities	Disassortative	Hub
$\begin{pmatrix} \eta & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \eta & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \eta & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \eta & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \eta \end{pmatrix}$	$\begin{pmatrix} \epsilon & \eta & \eta & \eta & \eta \\ \eta & \epsilon & \eta & \eta & \eta \\ \eta & \eta & \epsilon & \eta & \eta \\ \eta & \eta & \eta & \epsilon & \eta \\ \eta & \eta & \eta & \eta & \epsilon \end{pmatrix}$	$\begin{pmatrix} \eta & \eta & \eta & \eta & \eta \\ \eta & \eta & \epsilon & \epsilon & \epsilon \\ \eta & \epsilon & \eta & \epsilon & \epsilon \\ \eta & \epsilon & \epsilon & \eta & \epsilon \\ \eta & \epsilon & \epsilon & \epsilon & \eta \end{pmatrix}$

Table 6.1: Example of connection probability matrices $\mathbf{\Pi}$ for the communities, the disassortative and the hub structures. The parameters ϵ would be set to 0.01 for instance, while η would vary from 0.5 (topological structure) to 0.01 (noisy structure).

Real world dataset To illustrate the performance of Deep-LPBM over other methods, it would be interesting to analyse a well-known dataset, in particular a network that has already been studied by canonical positional models such as the Karate Club dataset as a small network example (34 nodes and 156 edges) as well as the Cora dataset for an evaluation on a larger network ($2,708$ nodes and $10,556$ edges).

6. Conclusion

Chapter 7

Appendix

This chapter is dedicated to provide additional materials, proofs and technical details associated with the manuscript.

7.1 Appendix of Foundations

In this section, we provide some additional material to Chapter 2.

7.1.1 SBM derivations

The ELBO is given by the following expressions

$$\begin{aligned}\mathscr{L}(\gamma, \mathbf{\Pi}; \boldsymbol{\tau}) &= \mathbb{E}_r\left[\log p(\mathbf{A}, \mathbf{C} \mid \boldsymbol{\gamma}, \boldsymbol{\pi})\right] - \mathbb{E}_r\left[r(\mathbf{C}; \tau)\right] \\ &= \sum_{i \neq j} \sum_{q,r=1}^{Q} \mathbb{E}_r\left[C_{iq} C_{jr} \log b(A_{ij}, \pi_{qr})\right] \\ &+ \sum_{i=1}^{N} \sum_{q=1}^{Q} \mathbb{E}_r\left[C_{iq} \log(\gamma_q)\right] - \sum_{i=1}^{N} \sum_{q=1}^{Q} \mathbb{E}_r\left[C_{iq} \log(\tau_{iq})\right] \\ &= \sum_{i \neq j} \sum_{q,r=1}^{Q} \tau_{iq} \tau_{jr} \log b(A_{ij}, \pi_{qr}) + \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log\left(\frac{\gamma_q}{\tau_{iq}}\right).\end{aligned} \tag{7.1}$$

$\boldsymbol{\tau}$ update

Note that we did not add any constraint on τ to keep it in the simplex and that will be taken care of afterwards. The derivative of the ELBO, detailed in Equation (7.1), with respect to τ_{ml} is given by

$$\frac{\partial \mathscr{L}}{\partial \tau_{ml}}(\gamma, \pi; \tau) = \sum_{n \neq m} \sum_{k=1}^{Q} \tau_{nk} \left(\log b(A_{mn}, \pi_{lk}) + \log b(A_{nm}, \pi_{kl})\right) + \log(\gamma_q) - \log(\tau_{iq}) - 1$$

Setting this derivative to zero gives the following update

$$\log(\tau_{iq}) = \sum_{n \neq m} \sum_{k=1}^{Q} \tau_{nk} \left(\log b(A_{mn}, \pi_{lk}) + \log b(A_{nm}, \pi_{kl})\right) + \log(\gamma_q) + const$$

which can be written as

$$\tau_{ml} \propto \gamma_l \prod_{n \neq m} \prod_{k=1}^{Q} \left(b(A_{mn}, \pi_{lk}) b(A_{nm}, \pi_{kl})\right)^{\tau_{nk}}$$

γ update

Since $\gamma \in \Delta_Q$, we add the constraint $\lambda(1 - \sum_{q=1}^{Q} \gamma_q)$ to the Equation (7.1), giving the Lagrangian function that we shall optimise, using the first-order conditions. The derivative of the Lagrangian with respect to γ_q is given by

$$\frac{\partial \mathscr{L}}{\partial \gamma_q}(\gamma, \mathbf{\Pi}; \boldsymbol{\tau}) = \sum_{i=1}^{N} \tau_{iq} \log(\gamma_q) - \lambda.$$

Setting this to zero gives that $\lambda = \frac{1}{\gamma_q} \sum_{i=1}^{N} \tau_{iq}$ which can be written as $\gamma_q \lambda = \sum_{i=1}^{N} \tau_{iq}$. Summing over q on both sides gives that $\lambda = N$. Plugging it back into the previous equation gives

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

$\boldsymbol{\Pi}$ update

The partial derivative of the ELBO with respect to π_{lk} gives

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \pi_{qr}}(\gamma, \pi; \tau) &= \sum_{i \neq j} \tau_{iq}\tau_{jr}\left(A_{ij}\frac{1}{\pi_{qr}} - (1 - A_{ij})\frac{1}{1-\pi_{qr}}\right) \\ &= \sum_{i \neq j}\left\{\tau_{iq}\tau_{jr}A_{ij}\left(\frac{1}{\pi_{qr}} + \frac{1}{1-\pi_{qr}}\right)\right\} - \sum_{i \neq j}\left\{\tau_{iq}\tau_{jr}\frac{1}{1-\pi_{qr}}\right\} \\ &= \frac{1}{\pi_{qr}(1-\pi_{qr})}\sum_{i \neq j}\{\tau_{iq}\tau_{jr}A_{ij}\} - \frac{1}{1-\pi_{qr}}\sum_{i \neq j}\tau_{iq}\tau_{jr}.\end{aligned}$$

Setting the derivative to zero and multiplying by $1 - \pi_{qr}$ gives

$$\sum_{i \neq j}\tau_{iq}\tau_{jr} = \frac{1}{\pi_{qr}}\sum_{i \neq j}\tau_{iq}\tau_{jr}A_{ij}.$$

Therefore, the update of π_{qr} is given by

$$\pi_{qr} = \frac{\sum_{i \neq j}\tau_{iq}\tau_{jr}A_{ij}}{\sum_{i \neq j}\tau_{iq}\tau_{jr}}.$$

7.2 Appendix of Chapter 1

7.2.1 Inference

Proof of Proposition 2. The ELBO can be decomposed as follow:

$$\begin{aligned}\log p(\mathbf{A}, \mathbf{W} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}) &= \mathbb{E}_R\left[\log p(\mathbf{A}, \mathbf{W} \mid \boldsymbol{\alpha}, \boldsymbol{\rho})\right] \\ &= \mathbb{E}_R\left[\log \frac{p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y} \mid \boldsymbol{\alpha}, \boldsymbol{\rho})}{p(\mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\rho})}\right] \qquad \text{applying Bayes rule} \\ &= \mathbb{E}_R\left[\log \frac{p(\mathbf{A}, \mathbf{W}, \mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y} \mid \boldsymbol{\alpha}, \boldsymbol{\rho})}{R(\mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y})} + \log \frac{R(\mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y})}{p(\mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\rho})}\right] \\ &= \mathscr{L}(R(\cdot); \boldsymbol{\alpha}, \boldsymbol{\rho}) + \mathrm{KL}(R(\cdot) || p(\mathbf{C}, \boldsymbol{\Pi}, \gamma, \mathbf{Y} \mid \mathbf{A}, \mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\rho})).\end{aligned}$$

□

Proof of Proposition 3.

$$
\begin{aligned}
\mathscr{L}(R(\cdot);\boldsymbol{\alpha},\boldsymbol{\rho}) =& \overbrace{\mathbb{E}_R\left[\log \frac{p(\mathbf{W}\mid \mathbf{C},\mathbf{A},\mathbf{Y},\boldsymbol{\alpha},\boldsymbol{\rho})p(\mathbf{Y})}{R(\mathbf{Y})}\right]}^{\mathscr{L}^{texts}(\boldsymbol{\tau},\tilde{\pi}_{qr1},\tilde{\pi}_{qr2}\tilde{\gamma};\boldsymbol{\alpha},\boldsymbol{\rho}):=} \\
&+ \overbrace{\mathbb{E}_R\left[\log \frac{p(\mathbf{A}\mid \mathbf{C},\boldsymbol{\Pi})p(\mathbf{C}\mid\gamma)p(\boldsymbol{\Pi})p(\gamma)}{R(\mathbf{C})R(\boldsymbol{\Pi})R(\gamma)}\right]}^{\mathscr{L}^{net}(\boldsymbol{\tau},\nu;\boldsymbol{\alpha},\boldsymbol{\rho}):=} \\
=& \mathbb{E}_R\left[\log p(\mathbf{W}\mid \mathbf{C},\mathbf{A},\mathbf{Y},\boldsymbol{\alpha},\boldsymbol{\rho})\right] \\
&+ \mathbb{E}_R\left[\log p(\mathbf{Y})\right] - \mathbb{E}_R\left[\log R(\mathbf{Y})\right] \\
&+ \mathbb{E}_R\left[\log p(\mathbf{A}\mid \mathbf{C},\boldsymbol{\Pi})\right] \\
&+ \mathbb{E}_R\left[\log p(\mathbf{C}\mid\gamma)\right] - \mathbb{E}_R\left[\log R(\mathbf{C})\right] \\
&+ \mathbb{E}_R\left[\log p(\boldsymbol{\Pi})\right] - \mathbb{E}_R\left[\log R(\boldsymbol{\Pi})\right] \\
&+ \mathbb{E}_R\left[\log p(\gamma)\right] - \mathbb{E}_R\left[\log R(\gamma)\right] \\
=& \sum_{i\neq j}^{M}\sum_{q,r}^{Q} A_{ij}\tau_{iq}\tau_{jr}\mathbb{E}_R\left[\underbrace{\log p(w_{ij}\mid Y_{qr},\boldsymbol{\alpha},\boldsymbol{\rho})}_{T_{ij}^{Y_{qr}}}\right] \\
&- \sum_{q,r}\mathrm{KL}(\mathcal{N}(\mu_{qr}(\tau,\nu),\sigma_{qr}(\tau,\nu))||\mathcal{N}(0,I)) \\
&+ \sum_{i\neq j}^{M}\sum_{q,r}^{Q}\tau_{iq}\tau_{jr}A_{ij}\left(\psi(\kappa_{qr1})-\psi(\kappa_{qr2})\right) \\
&+ \sum_{i\neq j}^{M}\sum_{q,r}^{Q}\tau_{iq}\tau_{jr}(\psi(\kappa_{qr2})-\psi(\kappa_{qr1}+\kappa_{qr2})) \\
&+ \sum_{i=1}^{M}\sum_{q=1}^{Q}\tau_{iq}\left(\psi(\gamma_q)-\psi\left(\sum_q\gamma_q\right)\right) + \log\mathcal{B}(1_Q) + \log(\mathcal{B}(a,b)) \\
&- \sum_{i=1}^{M}\sum_{q=1}^{Q}\tau_{iq}\log(\tau_{iq}) - \sum_{q,r}\log\mathcal{B}(\kappa_{qr1},\kappa_{qr2}) - \log\mathcal{B}(\gamma). \qquad (7.2)
\end{aligned}
$$

where,

$$
T_{ij}^{Y_{qr}} = \sum_{d=1}^{D_{ij}}\sum_{n=1}^{N_{id}^d}\sum_{v=1}^{V} w_{ij}^{dnv}\log\left(\sum_{k=1}^{K}\theta_{qrk}\beta_{kv}\right). \qquad (7.3)
$$

and $\theta_{qr} = \mu_{qr}(\tau,\nu) + \sigma_{qr}(\tau,\nu)\epsilon$, $\epsilon\sim\mathcal{N}(0_K,\mathrm{I}_K)$.

The Kullback-Leibler divergence between two Gaussian variables has a close form and is easy to compute. All the terms can be computed except for the expectation of $T_{ij}^{Y_{qr}}$ that can be approximated using a Monte-Carlo estimator, by drawing S samples for each pair (q,r), such that:

$$
\epsilon^s\sim\mathcal{N}(0,\mathbf{I}_K),\quad Y_{qr}^s = \mu_{qr}(\tau,\nu)+\sigma_{qr}(\tau,\nu)\odot\epsilon^s,\quad \theta_{qr}^s = \mathrm{softmax}(Y_{qr}^s).
$$

with $\odot$ denoting the Hadamard product. Thus, for each pair of nodes (i, j) and pair of clusters (q, r), the estimate is given by:

$$\hat{T}_{ij}^{qr} = S^{-1} \sum_{s=1}^{S} T_{ij}^{Y_{qr}^s}.$$

Plugging $\hat{T}_{ij}^{qr}$ in the Equation (7.2) gives the final estimator of the ELBO.

□

Figure 7.1 provides a translation of topics found by ETSBM on the real dataset and appearing in the meta-network.

7.2.2 Real data

Topics

round	hour	macron	melenchon	zemmour
all	melenchonwillwin	candidat	jadot	eric
make	world	emmanuel	jlm	ivotezemmour
go	erepublic	campaign	roussel	support
to vote	melenchon	zemmour	to vote	ivotezemmourthe
power	meeting	presidential	left	hofthereconquest
vote	popularunion	journalist	vote	zemmourpresident
president	program	debate	round	share
thanks	walkforthe	live	hidalgo	zemmourvsmacron
first	melenchontf	via	right	now

Figure 7.1: The most important words of each topic present in the meta-graph translated in English.

Figure 7.2 provides a translation of topics found by ETM on the real dataset and appearing in the meta-network.

Topics

zemmour	make	zemmour	melenchon
eric	all	to vote	make
macron	say	macron	hour
france	go	make	all
french	no	all	round
all	good	france	more
make	more	more	to vote
hofreconquest	see	go	program
more	power	mlp	go
zemmourpresident	as	alone	melenchonwillwin

Figure 7.2: The most important words of each topic present in the meta-graph translated in English.

7.3 Appendix of Chapter 2

We provide some material accompanying Chapter 3.

7.3.1 Graphical Model

Figure 7.3 provides the graphical representation of the model with its parameters.

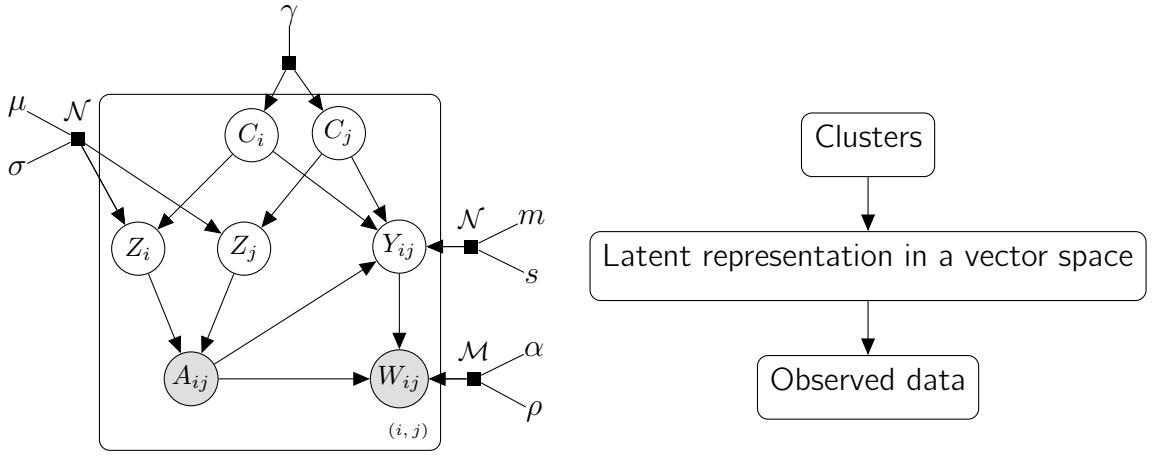


Figure 7.3: Graphical model with the parameters of Deep-LPTM where the Z_is denote the latent node vectors, Y_{ij}s the latent document vector, C_is the node cluster memberships, $\mathbf{A} = (A_{ij})_{ij} \in \mathcal{M}_{N\times N}(\{0,1\})$ the binary adjacency matrix and W_{ij} the document sent by node i to node j.

7.3.2 Computation of the ELBO terms

In this section, computational details regarding the ELBO are provided term by term. First, $\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right]$ is given by:

$$\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \kappa)\right] = \sum_{i,j=1}^{N} \left\{ A_{ij}\mathbb{E}_R\left[\log p_{ij}\right] + (1 - A_{ij})\mathbb{E}_R\left[\log\big(1 - p_{ij}\big)\right]\right\},$$

where $p_{ij} = (1 + e^{-\eta_{ij}})^{-1}$ and $\eta_{ij} := \kappa - \|Z_i - Z_j\|$.

Second, denoting $\beta_k = \mathrm{softmax}(\rho^\top \alpha_k) \in \mathbb{R}^V$, $\boldsymbol{\beta} = (\beta_1 \dots \beta_K)^\top \in \mathcal{M}_{K\times V}(\mathbb{R})$ and $w_{ij}^v = (\beta^\top \theta_{ij})_v$, the probability for the word v to appear in document W_{ij} ,for any $v \in \{1, \dots, V\}$, $\mathbb{E}_R\left[\log p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right]$ is:

$$\begin{aligned}\mathbb{E}_R\left[\log p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \boldsymbol{\rho}, \boldsymbol{\alpha})\right] &= \sum_{i,j=1}^{N} A_{ij}\mathbb{E}_R\left[\log \mathcal{M}_V\left(W_{ij}; M_{ij}, \boldsymbol{\beta}^\top \mathrm{softmax}(Y_{ij})\right)\right] \\ &= \sum_{i,j} A_{ij}\mathbb{E}_R\left[\log \frac{M_{ij}!}{\prod_{v=1}^V (W_{ij}^v)!} \prod_{v=1}^V (w_{ij}^v)^{W_{ij}^v}\right].\end{aligned}$$

The difference between the terms related to the cluster memberships, $\mathbb{E}_R[\log p(\mathbf{C} \mid \gamma)]$ and $\mathbb{E}_R[\log R(\mathbf{C})]$, gives the following:

$$\begin{aligned}\mathbb{E}_R[\log p(\mathbf{C} \mid \gamma)] - \mathbb{E}_R[\log R(\mathbf{C})] &= \sum_{i=1}^{M}\sum_{q=1}^{Q} \mathbb{E}_R[C_{iq}\log\gamma_q] - \mathbb{E}_R[C_{iq}\log\tau_{iq}] \\ &= \sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq}\log\frac{\gamma_q}{\tau_{iq}}.\end{aligned}$$

The difference between the generative distribution of the node positions term $\mathbb{E}_R[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})]$ and the one related to the variational distribution of the node positions $\mathbb{E}_R[\log R(\mathbf{Z} \mid \mathbf{A})]$ gives:

$$\begin{aligned}&\mathbb{E}_R[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})] - \mathbb{E}_R[\log R(\mathbf{Z} \mid \mathbf{A})] \\ &= \sum_{i=1}^{N}\sum_{q=1}^{Q} \mathbb{E}_R\left[C_{iq}\log\mathcal{N}\left(Z_i; \mu_q, \sigma_q^2\mathbf{I}_p\right)\right] - \sum_{i=1}^{N} \mathbb{E}_R\left[\log\mathcal{N}\left(Z_i; \mu_{\phi_Z}(\mathbf{A})_i, \sigma^2_{\phi_Z}(\mathbf{A})_i\mathbf{I}_p\right)\right] \\ &= -\sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq}\,\mathrm{KL}\left(\mathcal{N}\left(\mu_{\phi_Z}(\mathbf{A})_i, \sigma^2_{\phi_Z}(\mathbf{A})_i\mathbf{I}_p\right) \,||\, \mathcal{N}\left(\mu_q, \sigma_q^2\mathbf{I}_p\right)\right) \\ &= -\sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq} \underbrace{\left[\log\frac{\sigma_q^p}{\sigma_{\phi_Z}(\mathbf{A})_i^p} - \frac{p}{2} + \frac{p\sigma^2_{\phi_Z}(\mathbf{A})_i + \|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2}{2\sigma_q^2}\right]}_{\mathrm{KL}^Z_{iq}\left(\mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}(\mathbf{A})_i, \mu_q, \sigma_q\right)} \\ &= -\sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq}\,\mathrm{KL}^Z_{iq}\left(\mu_{\phi_Z}(\mathbf{A})_i, \sigma_{\phi_Z}(\mathbf{A})_i, \mu_q, \sigma_q\right).\end{aligned}$$

Symmetrically, the term regarding the edge positions is obtained as follow:

$$\begin{aligned}&\mathbb{E}_R[\log p(\mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \mathbf{m}, \mathbf{s})] - \mathbb{E}_R[\log R(\mathbf{Y} \mid \mathbf{A}, \mathbf{W})] \\ &= \sum_{i,j=1}^{N}\sum_{q,r=1}^{Q} \mathbb{E}_R\left[A_{ij}C_{iq}C_{jr}\log\mathcal{N}\left(Y_{ij}; m_{qr}, s_{qr}^2\mathbf{I}_K\right)\right] \\ &\quad - \sum_{i,j=1}^{N} \mathbb{E}_R\left[A_{ij}\log\mathcal{N}\left(Y_{ij}; \mu_{\phi_Y}(W_{ij}), \mathrm{diag}\left(\sigma^2_{\phi_Y}(W_{ij})\right)\right)\right] \\ &= -\sum_{i,j=1}^{N}\sum_{q,r=1}^{Q} A_{ij}\tau_{iq}\tau_{jr}\,\mathrm{KL}\left(\mathcal{N}\left(\mu_{\phi_Y}(W_{ij}), \mathrm{diag}\left(\sigma^2_{\phi_Y}(W_{ij})\right)\right) \,||\, \mathcal{N}\left(m_{qr}, s_{qr}^2\mathbf{I}_K\right)\right)\end{aligned} \tag{7.4}$$

Moreover, denoting $\mathrm{KL}^Y_{ijqr}\left(\mu_{\phi_Y}(W_{ij}), \sigma_{\phi_Y}(W_{ij}), m_{qr}, s_{qr}\right)$ the Kullback-Leibler term in Equation 7.4, we have:

$$\begin{aligned}&\mathrm{KL}^Y_{ijqr}\left(\mu_{\phi_Y}(W_{ij}), \sigma_{\phi_Y}(W_{ij}), m_{qr}, s_{qr}\right) \\ &= K\log s_{qr} - \sum_{k=1}^{K}\log\sigma_{\phi_Y}(W_{ij})_k - \frac{K}{2} + \frac{\sum_{k=1}^{K}\sigma^2_{\phi_Y}(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2}{2s_{qr}^2}\end{aligned}$$

7.3.3 Inference

Optimisation

In this section, we provide the optimisation steps to recover the parameters maximising the ELBO. To begin with, let us recall the ELBO expression:

$$
\begin{aligned}
&\mathscr{L}(\boldsymbol{\tau}, \phi_Z, \phi_Y; \gamma, \boldsymbol{\mu}, \boldsymbol{\sigma}, \kappa, \mathbf{m}, \mathbf{s}, \boldsymbol{\alpha}, \boldsymbol{\rho}) := \mathscr{L}(R(\cdot), \Theta) \\
&= \sum_{i,j} \left\{ A_{ij} \mathbb{E}_R \left[\log p_{ij} \right] + (1 - A_{ij}) \mathbb{E}_R \left[\log\left(1 - p_{ij}\right) \right] \right\} \\
&\quad + \sum_{i,j} A_{ij} \mathbb{E}_R \left[\log \frac{M_{ij}!}{\prod_{v=1}^{V} (W_{ij}^v)!} \prod_{v=1}^{V} (w_{ij}^v)^{W_{ij}^v} \right] \\
&\quad - \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \, \mathrm{KL}_{iq}^{Z} \left(\mu_{\phi_Z}(A)_i, \sigma^2_{\phi_Z}(A)_i \mathbf{I}_p, \mu_q, \sigma_q^2 \mathbf{I}_p \right) \\
&\quad - \sum_{i,j=1}^{N} \sum_{q,r=1}^{Q} A_{ij} \tau_{iq} \tau_{jr} \, \mathrm{KL}_{ijqr}^{Y} \left(\mu_{\phi_Y}(W_{ij}), \mathrm{diag}\left(\sigma^2_{\phi_Y}(W_{ij}) \right), m_{qr}, s_{qr}^2 \mathbf{I}_K \right) \\
&\quad + \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \frac{\gamma_q}{\tau_{iq}}.
\end{aligned}
\tag{7.5}
$$

Update of $\boldsymbol{\tau}$ First, we optimise the ELBO with respect to τ_{iq}. Since $\tau_i \in \Delta_{Q-1}$, the term $c_i(1 - \sum_{q=1}^{Q} \tau_{iq})$ is added to the ELBO, giving the Lagrangian of the function. Thus, the derivative of the Lagrangian with respect to τ_{iq} gives:

$$
\frac{\partial}{\partial \tau_{iq}} \mathcal{L}(R(\cdot); \Theta) = - \mathrm{KL}_{iq}^{Z} - \sum_{j=1}^{N} \sum_{r=1}^{Q} \left\{ A_{ij} \tau_{jr} \, \mathrm{KL}_{ijqr}^{Y} + A_{ji} \tau_{jr} \, \mathrm{KL}_{jirq}^{Y} \right\} + \log \frac{\gamma_q}{\tau_{iq}} - 1 - c_i.
$$

Setting this partial derivative to zero gives:

$$
\log \tau_{iq} = \underbrace{- \mathrm{KL}_{iq}^{Z} - \sum_{j=1}^{N} \sum_{r=1}^{Q} \left\{ A_{ij} \tau_{jr} \, \mathrm{KL}_{ijqr}^{Y} + A_{ji} \tau_{jr} \, \mathrm{KL}_{jirq}^{Y} \right\} + \log \gamma_q - 1}_{T_{iq}} - c_i.
$$

Thus $\tau_{iq} = e^{T_{iq}} e^{-c_i}$. Moreover, since $\sum_{q=1}^{Q} \tau_{iq} = 1$, we have $e^{c_i} = \sum_{q=1}^{Q} e^{T_{iq}}$. Therefore, $\tau_{iq} = e^{T_{iq}} / (\sum_{q=1}^{Q} e^{T_{iq}})$. The complete form is:

$$
\tau_{iq} = \frac{\gamma_q e^{- \mathrm{KL}_{iq}^{Z} - \sum_{j \neq i} \sum_{r=1}^{Q} \left(A_{ij} \tau_{jr} K_{ij,qr}^{Y} + A_{ji} \tau_{jr} \, \mathrm{KL}_{jirq}^{Y} \right)}}{\sum_{l=1}^{Q} \gamma_l e^{- \mathrm{KL}_{il}^{Z} - \sum_{j \neq i} \sum_{l'=1}^{Q} \left(A_{ij} \tau_{jl'} K_{ij,ll'}^{Y} + A_{ji} \tau_{jl'} \, \mathrm{KL}_{jil'l}^{Y} \right)}}.
\tag{7.6}
$$

Update of γ Since $\gamma \in \Delta_{Q-1}$, this constraint is added to the function to obtain the lagrangian. This corresponds to adding the term $c\left(1 - \sum_{q=1}^{Q} \gamma_q\right)$ to the ELBO. Thus, the

partial derivative of the Lagrangian $\mathcal{L}$ with respect to γ_q of the lagrangian is:

$$\frac{\partial}{\partial \gamma_q}\mathcal{L}(R(\cdot);\Theta,c) = \sum_{i=1}^{N}\frac{\tau_{iq}}{\gamma_q} - c.$$

Setting this to zero gives:

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

Multiplying by γ_q and summing over q gives that $c = N$. Therefore, after plugging it back into the previous expression, the following holds:

$$\gamma_q = \frac{1}{N}\sum_{i=1}^{N}\tau_{iq}. \tag{7.7}$$

Updates of μ_q and σ_q Taking the partial derivative of the ELBO with respect to μ_q gives the following:

$$\frac{\partial}{\partial \mu_q}\mathscr{L}(R(\cdot);\Theta) = -\sum_{i=1}^{N}\frac{\tau_{iq}}{2\sigma_q^2}\left(2\mu_q - 2\mu_{\phi_Z}(\mathbf{A})_i\right).$$

Therefore, setting this quantity to zero gives the following update for μ_q:

$$\mu_q = \left(\sum_{i=1}^{N}\tau_{iq}\right)^{-1}\sum_{i=1}^{N}\tau_{iq}\mu_{\phi_Z}(\mathbf{A})_i. \tag{7.8}$$

The partial derivate of the ELBO with respect to σ_q is:

$$\frac{\partial}{\partial \sigma_q}\mathscr{L}(R(\cdot);\Theta) = -\sum_{i=1}^{N}\tau_{iq}\left(\frac{p}{\sigma_q} - \frac{p\sigma^2_{\phi_Z}(\mathbf{A})_i + \|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2}{2}\frac{2\sigma_q}{\sigma_q^4}\right).$$

Thus, the first-order condition on σ_q gives the following:

$$\begin{aligned}\frac{p}{\sigma_q}\sum_{i=1}^{N}\tau_{iq} &= \frac{1}{\sigma_q^3}\sum_{i=1}^{N}\tau_{iq}\left(p\sigma^2_{\phi_Z}(\mathbf{A})_i + \|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2\right)\\ \sigma_q^2 &= \left(p\sum_{i=1}^{N}\tau_{iq}\right)^{-1}\sum_{i=1}^{N}\tau_{iq}\left(p\sigma^2_{\phi_Z}(\mathbf{A})_i + \|\mu_{\phi_Z}(\mathbf{A})_i - \mu_q\|_2^2\right).\end{aligned} \tag{7.9}$$

Updates of $\mathbf{m}$ and $\mathbf{s}$ As in the previous sections, we optimise the ELBO with respect to $\mathbf{m}$ and $\mathbf{s}$ with the first-order conditions. The partial derivate of $\mathscr{L}$ with respect to m_{qr} is:

$$\frac{\partial}{\partial m_{qr}}\mathscr{L}(R(\cdot);\Theta) = -\frac{1}{2s_{qr}^2}\sum_{i,j=1}^{N}A_{ij}\tau_{iq}\tau_{jr}\big(2m_{qr} - 2\mu_{\phi_Y}(W_{ij})\big).$$

Therefore, setting this expression to zero gives the following update for m_{qr}:

$$m_{qr} = \left(\sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \right)^{-1} \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \mu_{\phi_Y}(W_{ij}). \tag{7.10}$$

The partial derivate of the ELBO with respect to s_{qr} is:

$$\frac{\partial}{\partial s_{qr}} \mathscr{L}(R(\cdot); \Theta) = -\sum_{i=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \left(\frac{K}{s_{qr}} - \frac{\sum_{k=1}^{K} \sigma^2_{\phi_Y}(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2}{2} \frac{2 s_{qr}}{s_{qr}^4} \right).$$

Thus, the first-order condition on s_{qr} gives the following:

$$\begin{aligned} s_{qr}^2 K \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} &= \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \left[\sum_{k=1}^{K} \sigma^2_{\phi_Y}(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 \right] \\ s_{qr}^2 &= \left(K \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \right)^{-1} \sum_{i,j=1}^{N} A_{ij} \tau_{iq} \tau_{jr} \left[\sum_{k=1}^{K} \sigma^2_{\phi_Y}(W_{ij})_k + \|\mu_{\phi_Y}(W_{ij}) - m_{qr}\|_2^2 \right]. \end{aligned} \tag{7.11}$$

Derivation of the selection model criterion

Proof of Proposition 7. Assuming a fully factorised prior distribution such that $p(\kappa, \gamma, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{m}, \mathbf{s}, \boldsymbol{\rho}, \boldsymbol{\alpha}) = p(\kappa)p(\gamma)p(\boldsymbol{\mu})p(\boldsymbol{\sigma})p(\mathbf{m})p(\mathbf{s})p(\boldsymbol{\rho})p(\boldsymbol{\alpha})$, Lemma 3.1 in Biernacki, Celeux, Govaert (2000) can be directly extended to our case to decompose the integral in (4.23):

$$\begin{aligned} \log p(\mathbf{A}, \mathbf{W}, \mathbf{Z}, \mathbf{Y}, \mathbf{C} \mid \mathcal{M}, Q, K, P) = &\log p(\mathbf{A} \mid \mathbf{Z}, \mathcal{M}) + \log p(\mathbf{Z} \mid \mathbf{C}, \mathcal{M}, Q, P) \\ &+ \log p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \mathcal{M}) + \log p(\mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \mathcal{M}, K) \\ &+ \log p(\mathbf{C} \mid \mathcal{M}, Q). \end{aligned} \tag{7.12}$$

Unfortunately, this expression cannot be computed since each term requires an integral with respect to the corresponding parameter. For instance, $p(\mathbf{A} \mid \mathbf{Z}, \mathcal{M})$ cannot be integrated analytically because of the logistic link function. Fortunately, a BIC-like approximation can be derived for $p(\mathbf{A} \mid \mathbf{Z}, \mathcal{M})$, $p(\mathbf{Z} \mid \mathbf{C}, \mathcal{M}, Q, P)$, $p(\mathbf{Y} \mid \mathbf{A}, \mathbf{C}, \mathcal{M}, K)$ and $p(\mathbf{W} \mid \mathbf{A}, \mathbf{Y}, \mathcal{M})$. For instance, $p(\mathbf{A} \mid \mathbf{Z}, \mathcal{M})$ can be approximated by:

$$\begin{aligned} \log p(\mathbf{A} \mid \mathbf{Z}, \mathcal{M}) &= \log \int_{\kappa} p(\mathbf{A} \mid \kappa, \mathbf{Z}, \mathcal{M}) p(\kappa) d\kappa \\ &\approx \max_{\kappa} \log p(\mathbf{A} \mid \mathbf{Z}, \kappa, \mathcal{M}) - \frac{\nu_{A,\mathcal{M}}}{2} \log(n_A), \end{aligned}$$

where $\nu_{A,\mathcal{M}} = 1$ denotes the number of free components in κ and $n_A = N(N-1)$ denotes the number of observations in A. This can be applied to all terms except $p(\mathbf{C} \mid \mathcal{M}, Q)$ since the posterior cluster memberships probabilities τ_i can be on the boundary of the

parameter space. Fortunately, this term can be computed analytically. By assuming a Dirichlet prior $\mathcal{D}_Q(\delta_1, \ldots, \delta_Q)$ on the topic proportions γ:

$$\begin{aligned} p(\mathbf{C} \mid \mathcal{M}, Q) &= \int p(\mathbf{C} \mid \gamma, \mathcal{M}, Q) p(\gamma) d\gamma \\ &= \frac{\Gamma\left(\sum_{q=1}^{Q} \delta_q\right)}{\prod_{q=1}^{Q} \Gamma\left(\delta_q\right)} \frac{\prod_{q=1}^{Q} \Gamma(n_q + \delta_q)}{\Gamma\left(\sum_{q=1}^{Q} n_q + \delta_q\right)}, \end{aligned}$$

where $n_q := \sum_{i=1}^{N} C_{iq}$. In this paper, we consider the non-informative Jeffreys prior distribution ($\delta_q = 1/2$), as in Biernacki, Celeux, Govaert (2000) and Daudin, Picard, Robin (2008). Moreover, since C_i is not available, we replace it with its maximum-a-posteriori estimate $\hat{C}_i$ where $\hat{C}_{iq} = 1$ if $q = \arg\max(\tau_{i1}, \ldots, \tau_{iQ})$, and 0 otherwise, which in turn gives $\hat{n}_q := \sum_{i=1}^{N} \hat{C}_{iq}$. Using Stirling formula to approximate the Gamma function for a large value of N, we obtain:

$$p(\hat{\mathbf{C}} \mid \mathcal{M}, Q) \approx p(\hat{\mathbf{C}} \mid \hat{\gamma}, \mathcal{M}, Q) - \frac{Q-1}{2} \log(N). \tag{7.13}$$

To conclude, since $\mathbf{Z}$ and $\mathbf{Y}$ are not available, we replace the missing data with their maximum-a-posteriori estimates $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$. Denoting $\widehat{ICL}(\mathcal{M}, Q, K, P)$ the quantity $\log p(\mathbf{A}, \mathbf{W}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}}, \hat{\mathbf{C}} \mid \mathcal{M}, Q, K, P)$, we have:

$$\begin{aligned} \widehat{ICL}(\mathcal{M}, Q, K, P) =& \max_{\Theta} \log p(\mathbf{A}, \mathbf{W}, \hat{\mathbf{Z}}, \hat{\mathbf{Y}}, \hat{\mathbf{C}} \mid \Theta, \mathcal{M}, Q, K, P) \\ & - \Omega(\mathcal{M}, Q, K, P) \\ =& \max_{\kappa} \log p(\mathbf{A} \mid \hat{\mathbf{Z}}, \kappa, \mathcal{M}) - \frac{1}{2} \log(N(N-1)) \\ & + \max_{\boldsymbol{\mu}, \boldsymbol{\sigma}} \log p(\hat{\mathbf{Z}} \mid \hat{\mathbf{C}}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{M}, Q, P) - \frac{QP + Q}{2} \log(N) \\ & + \max_{\boldsymbol{\rho}, \boldsymbol{\alpha}} \log p(\mathbf{W} \mid \mathbf{A}, \hat{\mathbf{Y}}, \boldsymbol{\rho}, \boldsymbol{\alpha}, \mathcal{M}) - \frac{VL + KL}{2} \log(M) \\ & + \max_{\mathbf{m}, \mathbf{s}} \log p(\hat{\mathbf{Y}} \mid \mathbf{A}, \hat{\mathbf{C}}, \mathbf{m}, \mathbf{s}, \mathcal{M}, K) - \frac{Q^2 K + Q^2}{2} \log(M) \\ & + \max_{\gamma} \log p(\hat{\mathbf{C}} \mid \gamma, \mathcal{M}, Q) - \frac{Q-1}{2} \log(N), \end{aligned}$$

where

$$\begin{aligned} \Omega(\mathcal{M}, Q, K, P) =& \frac{1}{2} \log(N(N-1)) \\ & + \frac{Q(P+2) - 1}{2} \log(N) \\ & + \frac{L(V+K) + Q^2(K+1)}{2} \log(M). \end{aligned}$$

□

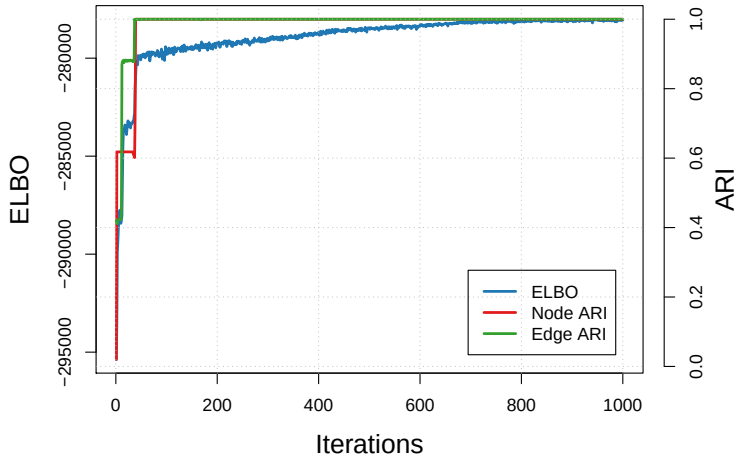


Figure 7.4: Evolution of the ELBO, as well as the nodes ARI during optimisation for 1000 iterations

7.3.4 Numerical experiments

This section provides the entire ELBO evolution as well as the evolutions of the node ARI and the edge ARI, corresponding to the example in Section 4.4.2.

7.4 Appendix of Chapter 3

7.4.1 Computation of the ELBO terms

In this section, computational details regarding the ELBO are provided term by term. First, $\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi})\right]$ is given by:

$$\mathbb{E}_R\left[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{\Pi})\right] = \sum_{i,j=1}^{N} \left\{ A_{ij} \mathbb{E}_R\left[\log \gamma_i^\top \mathbf{\Pi} \gamma_j\right] + (1 - A_{ij}) \mathbb{E}_R\left[\log 1 - \gamma_i^\top \mathbf{\Pi} \gamma_j\right] \right\}.$$

The difference between the terms related to the cluster memberships gives the following:

$$\begin{aligned} \mathbb{E}_R\left[\log p(\mathbf{C} \mid \gamma)\right] - \mathbb{E}_R\left[\log R(\mathbf{C})\right] &= \sum_{i=1}^{M} \sum_{q=1}^{Q} \mathbb{E}_R\left[C_{iq} \log \gamma_q\right] - \mathbb{E}_R\left[C_{iq} \log \tau_{iq}\right] \\ &= \sum_{i=1}^{N} \sum_{q=1}^{Q} \tau_{iq} \log \frac{\gamma_q}{\tau_{iq}}. \end{aligned}$$

The difference between the generative distribution of the node positions term $\mathbb{E}_R\left[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})\right]$ and the one related to the variational distribution of the node positions $\mathbb{E}_R\left[\log R(\mathbf{Z} \mid \mathbf{A})\right]$

gives:

$$
\begin{aligned}
&\mathbb{E}_R\left[\log p(\mathbf{Z} \mid \mathbf{C}, \boldsymbol{\mu}, \boldsymbol{\sigma})\right] - \mathbb{E}_R\left[\log R(\mathbf{Z} \mid \mathbf{A})\right] \\
&= \sum_{i=1}^{N}\sum_{q=1}^{Q} \mathbb{E}_R\left[C_{iq} \log \mathcal{N}\left(Z_i; \mu_q, \sigma_q^2 \mathbf{I}_p\right)\right] - \sum_{i=1}^{N} \mathbb{E}_R\left[\log \mathcal{N}\left(Z_i; \mu_\phi(\mathbf{A})_i, \sigma_\phi^2(\mathbf{A})_i \mathbf{I}_p\right)\right] \\
&= -\sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq} \operatorname{KL}\left(\mathcal{N}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi^2(\mathbf{A})_i \mathbf{I}_p\right) \,||\, \mathcal{N}\left(\mu_q, \sigma_q^2 \mathbf{I}_p\right)\right) \\
&= -\sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq} \underbrace{\left[\log \frac{\sigma_q^p}{\sigma_\phi(\mathbf{A})_i^p} - \frac{p}{2} + \frac{p\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2}{2\sigma_q^2}\right]}_{\operatorname{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right)} \\
&= -\sum_{i=1}^{N}\sum_{q=1}^{Q} \tau_{iq} \operatorname{KL}_{iq}\left(\mu_\phi(\mathbf{A})_i, \sigma_\phi(\mathbf{A})_i, \mu_q, \sigma_q\right). \qquad (7.14)
\end{aligned}
$$

7.4.2 Optimisation

Update of $\boldsymbol{\tau}$

First, we optimise the ELBO with respect to τ_{iq}. Since $\tau_i \in \Delta_Q$, the term $\lambda_i(1 - \sum_{q=1}^{Q} \tau_{iq})$ is added to the ELBO, giving the Lagrangian of the function. Thus, the derivative of the Lagrangian with respect to τ_{iq} gives:

$$
\frac{\partial}{\partial \tau_{iq}} \mathcal{L}(R(\cdot); \Theta) = -\operatorname{KL}_{iq} - \log \frac{\tau_{iq}}{\gamma_q} - 1 - \lambda_i.
$$

Setting this partial derivative to zero gives:

$$
\begin{aligned}
\log \tau_{iq} &= -\operatorname{KL}_{iq} + \log \gamma_q - 1 - \lambda_i \\
&= -\operatorname{KL}_{iq} + \log \gamma_q + c_i
\end{aligned}
$$

with the constraint that $\sum_{q=1}^{Q} \tau_{iq} = 1$ which translates into $c_i = -\log(\sum_{q=1}^{Q} \gamma_q \operatorname{KL}_{iq})$. Therefore,

$$
\tau_{iq}^{\star} = \frac{\gamma_q e^{-\operatorname{KL}_{iq}}}{\sum_{r=1}^{Q} \gamma_r e^{-\operatorname{KL}_{ir}}}. \qquad (7.15)
$$

Update of γ

Since $\gamma \in \Delta_Q$, the term $c\left(1 - \sum_{q=1}^{Q} \gamma_q\right)$ is added to the ELBO and gives the Lagrangian function, that incorporates the constraint on γ. Thus, the partial derivative of the Lagrangian $\mathcal{L}$ with respect to γ_q of the Lagrangian is:

$$
\frac{\partial}{\partial \gamma_q} \mathcal{L}(R(\cdot); \Theta, c) = \sum_{i=1}^{N} \frac{\tau_{iq}}{\gamma_q} - c.
$$

Setting this to zero gives:

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

Multiplying by γ_q and summing over q gives that $c = N$. Therefore, after plugging it back into the previous expression, the following holds:

$$\gamma_q^\star = \frac{1}{N}\sum_{i=1}^{N}\tau_{iq}. \tag{7.16}$$

Updates of μ_q

Taking the partial derivative of the ELBO with respect to μ_q gives the following:

$$\frac{\partial}{\partial\mu_q}\mathscr{L}(R(\cdot);\Theta) = -\sum_{i=1}^{N}\frac{\tau_{iq}}{2\sigma_q^2}\left(2\mu_q - 2\mu_{\phi_Z}(\mathbf{A})_i\right).$$

Therefore, setting this quantity to zero gives the following update for μ_q:

$$\mu_q = \left(\sum_{i=1}^{N}\tau_{iq}\right)^{-1}\sum_{i=1}^{N}\tau_{iq}\mu_{\phi_Z}(\mathbf{A})_i. \tag{7.17}$$

Updates of σ_q

The partial derivative of the ELBO with respect to σ_q is:

$$\frac{\partial}{\partial\sigma_q}\mathscr{L}(R(\cdot);\Theta) = -\sum_{i=1}^{N}\tau_{iq}\left(\frac{d}{\sigma_q} - \frac{d\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2}{2}\frac{2\sigma_q}{\sigma_q^4}\right).$$

Thus, the first-order condition on σ_q gives the following:

$$\begin{aligned}\frac{d}{\sigma_q}\sum_{i=1}^{N}\tau_{iq} &= \frac{1}{\sigma_q^3}\sum_{i=1}^{N}\tau_{iq}\left(d\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2\right)\\ {\sigma_q^2}^\star &= \left(d\sum_{i=1}^{N}\tau_{iq}\right)^{-1}\sum_{i=1}^{N}\tau_{iq}\left(d\sigma_\phi^2(\mathbf{A})_i + \|\mu_\phi(\mathbf{A})_i - \mu_q\|_2^2\right).\end{aligned} \tag{7.18}$$

Bibliography

...

Airoldi, Blei, Fienberg, Xing (2008). *Mixed Membership Stochastic Blockmodels*. In: Journal of Machine Learning Research, Vol. 9, No. 65, pp. 1981–2014.

Attias (1999). *A variational baysian framework for graphical models*. In: Advances in neural information processing systems, Vol. 12, pp. 209–215.

Bengio, Lamblin, Popovici, Larochelle (2006). *Greedy layer-wise training of deep networks*. In: Advances in neural information processing systems, Vol. 19, pp. 153–160.

Bergé, Bouveyron, Corneli, Latouche (2019). *The latent topic block model for the co-clustering of textual interaction data*. In: Computational Statistics & Data Analysis, Vol. 137, pp. 247–270.

Biernacki, Celeux, Govaert (2000). *Assessing a mixture model for clustering with the integrated completed likelihood*. In: IEEE transactions on pattern analysis and machine intelligence, Vol. 22, No. 7, pp. 719–725.

Bishop (2006). *Pattern recognition and machine learning*. Springer.

Blei, Kucukelbir, McAuliffe (2017). *Variational inference: A review for statisticians*. In: Journal of the American statistical Association, Vol. 112, No. 518, pp. 859–877.

Blei, Lafferty (2006). *Correlated topic models*. In: Advances in neural information processing systems, Vol. 18, p. 147.

Blei, Ng, Jordan (2003). *Latent dirichlet allocation*. In: the Journal of machine Learning research, Vol. 3, pp. 993–1022.

Blondel, Guillaume, Lambiotte, Lefebvre (2008). *Fast unfolding of communities in large networks*. In: Journal of statistical mechanics: theory and experiment, Vol. 2008, No. 10, p. 10008.

Borgatti, Mehra, Brass, Labianca (2009). *Network analysis in the social sciences*. In: science, Vol. 323, No. 5916, pp. 892–895.

Bottou, Bousquet (2007). *The tradeoffs of large scale learning*. In: Advances in neural information processing systems, Vol. 20, pp. 161–168.

Boutin, Bouveyron, Latouche (2023). *Embedded topics in the stochastic block model*. In: Statistics and Computing, Vol. 33, No. 5, pp. 1–20.

Bouveyron, Latouche, Zreik (2018). *The stochastic topic block model for the clustering of vertices in networks with textual edges*. In: Statistics and Computing, Vol. 28, No. 1, pp. 11–31.

Bruna, Zaremba, Szlam, LeCun (2013). *Spectral networks and locally connected networks on graphs*. arXiv: 1312.6203 [cs.LG].

Clement, Bierbaum, O'Keeffe, Alemi (2019). *On the use of arxiv as a dataset*. arXiv: 1905.00075 [cs.IR].

Côme, Latouche (2015). *Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood*. In: Statistical Modelling, Vol. 15, No. 6, pp. 564–589.

Corneli, Bouveyron, Latouche, Rossi (2019). *The dynamic stochastic topic block model for dynamic networks with textual edges*. In: Statistics and Computing, Vol. 29, No. 4, pp. 677–695.

Corneli, Latouche, Rossi (2016). *Block modelling in dynamic networks with non-homogeneous poisson processes and exact ICL*. In: Social Network Analysis and Mining, Vol. 6, No. 1, pp. 1–14.

Daudin, Picard, Robin (2008). *A mixture model for random graphs*. In: Statistics and computing, Vol. 18, No. 2, pp. 173–183.

Daudin, Pierre, Vacher (2010). *Model for heterogeneous random networks using continuous latent variables and an application to a tree–fungus network*. In: Biometrics, Vol. 66, No. 4, pp. 1043–1051.

De Bruijn (1981). *Asymptotic methods in analysis*. Vol. 4. Courier Corporation.

Deerwester, Dumais, Furnas, Landauer, Harshman (1990). *Indexing by latent semantic analysis*. In: Journal of the American society for information science, Vol. 41, No. 6, pp. 391–407.

Defferrard, Bresson, Vandergheynst (2016). *Convolutional neural networks on graphs with fast localized spectral filtering*. In: Advances in neural information processing systems, Vol. 29, pp. 3844–3852.

Dempster, Laird, Rubin (1977). *Maximum likelihood from incomplete data via the EM algorithm*. In: Journal of the royal statistical society: series B (methodological), Vol. 39, No. 1, pp. 1–22.

Dieng, Ruiz, Blei (2020). *Topic modeling in embedding spaces*. In: Transactions of the Association for Computational Linguistics, Vol. 8, pp. 439–453.

Erdos, Rényi, et al. (1960). *On the evolution of random graphs*. In: Publ. Math. Inst. Hung. Acad. Sci, Vol. 5, No. 1, pp. 17–60.

Fey, Lenssen (2019). *Fast graph representation learning with PyTorch Geometric*. arXiv: 1903.02428 [cs.LG].

Fienberg, Wasserman (1981). *Categorical data analysis of single sociometric relations*. In: Sociological methodology, Vol. 12, pp. 156–192.

Fortunato, Barthelemy (2007). *Resolution limit in community detection*. In: Proceedings of the national academy of sciences, Vol. 104, No. 1, pp. 36–41.

Fruchterman, Reingold (1991). *Graph drawing by force-directed placement*. In: Software: Practice and experience, Vol. 21, No. 11, pp. 1129–1164.

Gershman, Goodman (2014). *Amortized inference in probabilistic reasoning*. In: Proceedings of the annual meeting of the cognitive science society. Vol. 36.

Ghahramani (1997). *On structured variational approximations*. In: University of Toronto Technical Report, CRG-TR-97-1. Citeseer.

Girvan, Newman (2002). *Community structure in social and biological networks*. In: Proceedings of the national academy of sciences, Vol. 99, No. 12, pp. 7821–7826.

Goodfellow, Bengio, Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Gopalan, Blei (2013). *Efficient discovery of overlapping communities in massive networks*. In: Proceedings of the National Academy of Sciences, Vol. 110, No. 36, pp. 14534–14539.

Granovetter (1973). *The strength of weak ties*. In: American journal of sociology, Vol. 78, No. 6, pp. 1360–1380.

Grünwald, Roos (2019). *Minimum description length revisited*. In: International journal of mathematics for industry, Vol. 11, No. 01, p. 1930001.

Handcock, Raftery, Tantrum (2007). *Model-based clustering for social networks*. In: Journal of the Royal Statistical Society: Series A (Statistics in Society), Vol. 170, No. 2, pp. 301–354.

Hoff (2007). *Modeling homophily and stochastic equivalence in symmetric relational data*. In: Advances in neural information processing systems, Vol. 20, pp. 657–664.

Hoff, Raftery, Handcock (2002). *Latent space approaches to social network analysis*. In: Journal of the american Statistical association, Vol. 97, No. 460, pp. 1090–1098.

Hoffman, Blei, Wang, Paisley (2013). *Stochastic Variational Inference*. arXiv: 1206.7051 [stat.ML].

Hoffman, Bach, Blei (2010). *Online learning for latent dirichlet allocation*. In: Advances in neural information processing systems, Vol. 23, pp. 856–864.

Hoffman, Blei (2015). *Structured stochastic variational inference*. In: Artificial Intelligence and Statistics, pp. 361–369.

Hofmann (1999). *Probabilistic latent semantic analysis*. In: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pp. 289–296.

Hornik, Stinchcombe, White (1989). *Multilayer feedforward networks are universal approximators*. In: Neural networks, Vol. 2, No. 5, pp. 359–366.

Jefferys, Berger (1991). *Sharpening Occam's Razor on a Bayesian strop.* In: Bulletin of the Astronomical Society, Vol. 23, No. 3, p. 1259.

Jernite, Latouche, Bouveyron, Rivera, Jegou, Lamassé (2014). *The random subgraph model for the analysis of an ecclesiastical network in Merovingian Gaul*. In: The Annals of Applied Statistics, Vol. 8, No. 1, pp. 377–405.

Jouvin, Latouche, Bouveyron, Bataillon, Livartowski (2021). *Greedy clustering of count data through a mixture of multinomial PCA*. In: Computational Statistics, Vol. 36, No. 1, pp. 1–33.

Kemp, Tenenbaum, Griffiths, Yamada, Ueda (2006). *Learning systems of concepts with an infinite relational model*. In: AAAI. Vol. 3, p. 5.

Kingma, Ba (2014). *Adam: A method for stochastic optimization*. arXiv: 1412.6980 [cs.LG].

Kingma, Welling (2014). *Auto-Encoding Variational Bayes*. arXiv: 1312.6114 [stat.ML].

Kingma, Welling, et al. (2019). *An introduction to variational autoencoders*. In: Foundations and Trends® in Machine Learning, Vol. 12, No. 4, pp. 307–392.

Kipf, Welling (2016). *Variational graph auto-encoders*. arXiv: 1611.07308 [stat.ML].

Kipf, Welling (2017). *Semi-Supervised Classification with Graph Convolutional Networks*. In: International Conference on Learning Representations.

Latouche, Birmele, Ambroise (2012). *Variational Bayesian inference and complexity control for stochastic block models*. In: Statistical Modelling, Vol. 12, No. 1, pp. 93–115.

Latouche, Birmelé, Ambroise (2011). *Overlapping stochastic block models with application to the french political blogosphere*. In: The Annals of Applied Statistics, Vol. 5, No. 1, pp. 309–336.

Laurent (2022). *Comment la gauche sociale-démocrate a perdu la bataille des réseaux sociaux*. In: Le Monde.

Lebarbier, Mary-Huard (2006). *Une introduction au critère BIC: fondements théoriques et interprétation*. In: Journal de la Société française de statistique, Vol. 147, No. 1, pp. 39–57.

LeCun, Bottou, Bengio, Haffner (1998). *Gradient-based learning applied to document recognition*. In: Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324.

Lee, Wilkinson (2019). *A review of stochastic block models and extensions for graph clustering*. In: Applied Network Science, Vol. 4, No. 1, pp. 1–50.

Liang, Corneli, Bouveyron, Latouche (2022). *Deep latent position model for node clustering in graphs*. In: The 30th European Symposium on Artificial Neural Networks (ESANN 2022).

Liu, Niculescu-Mizil, Gryc (2009). *Topic-link LDA: joint models of topic and author community*. In: proceedings of the 26th annual international conference on machine learning, pp. 665–672.

Maddison, Mnih, Teh (2017). *The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables*. In: International Conference on Learning Representations.

Mariadassou, Robin, Vacher (2010). *Uncovering latent structure in valued graphs: a variational approach*. In: The Annals of Applied Statistics, Vol. 4, No. 2, pp. 715–742.

Matias, Miele (2017). *Statistical clustering of temporal networks through a dynamic stochastic block model*. In: Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 79, No. 4, pp. 1119–1141.

Matias, Robin (2014). *Modeling heterogeneity in random graphs through latent space models: a selective review*. In: ESAIM: Proceedings and Surveys, Vol. 47, pp. 55–74.

Mehta, Duke, Rai (2019). *Stochastic blockmodels meet graph neural networks*. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 4466–4474.

Meng, Huang, Wang, Wang, Zhang, Zhang, Han (2020). *Discriminative topic mining via category-name guided text embedding*. In: Proceedings of The Web Conference 2020, pp. 2121–2132.

Mikolov, Chen, Corrado, Dean (2013). *Efficient estimation of word representations in vector space*. arXiv: 1301.3781 [cs.CL].

Mikolov, Sutskever, Chen, Corrado, Dean (2013). *Distributed representations of words and phrases and their compositionality*. In: Advances in neural information processing systems, Vol. 26, pp. 3111–3119.

Montufar, Pascanu, Cho, Bengio (2014). *On the number of linear regions of deep neural networks*. In: Advances in neural information processing systems, Vol. 27, pp. 2924–2932.

Nalisnick, Smyth (2016). *Stick-Breaking Variational Autoencoders*. In: International Conference on Learning Representations.

Newman (2006). *Modularity and community structure in networks*. In: Proceedings of the national academy of sciences, Vol. 103, No. 23, pp. 8577–8582.

Pan, Hu, Long, Jiang, Yao, Zhang (2018). *Adversarially Regularized Graph Autoencoder for Graph Embedding*. In: International Joint Conference on Artificial Intelligence. IJCAI'18. AAAI Press: Stockholm, Sweden, pp. 2609–2615.

Papadimitriou, Raghavan, Tamaki, Vempala (1998). *Latent Semantic Indexing: A Probabilistic Analysis*. In: ACM press, pp. 159–168.

Paszke, Gross, Massa, Lerer, Bradbury, Chanan, Killeen, Lin, Gimelshein, Antiga, Desmaison, Kopf, Yang, DeVito, Raison, Tejani, Chilamkurthy, Steiner, Fang, Bai, Chintala (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett. Curran Associates, Inc., pp. 8024–8035.

Pathak, Delong, Erickson, Banerjee (2008). *Social topic models for community extraction*.

Pavel N. Krivitsky, Hunter, Butts, Klumb, Goodreau, Morris (2003–2022). *Statnet: Tools for the Statistical Modeling of Network Data*. Statnet Development Team.

Peixoto (2021). *Descriptive vs. inferential community detection in networks: pitfalls, myths, and half-truths*. arXiv: 2112.00183 [physics.soc-ph].

Piantadosi (2014). *Zipf's word frequency law in natural language: A critical review and future directions*. In: Psychonomic bulletin & review, Vol. 21, pp. 1112–1130.

Raftery (1995). *Bayesian model selection in social research*. In: Sociological methodology, Vol. 25, pp. 111–163.

Ranganath, Gerrish, Blei (2014). *Black box variational inference*. In: Artificial intelligence and statistics. Proceedings of Machine Learning Research, pp. 814–822.

Ranganath, Tran, Blei (2016). *Hierarchical variational models*. In: International conference on machine learning. Proceedings of Machine Learning Research, pp. 324–333.

Ranzato, Poultney, Chopra, Cun (2006). *Efficient learning of sparse representations with an energy-based model*. In: Advances in neural information processing systems, Vol. 19, pp. 1137–1144.

Rezende, Mohamed (2015). *Variational inference with normalizing flows*. In: International conference on machine learning. Proceedings of Machine Learning Research, pp. 1530–1538.

Rezende, Mohamed, Wierstra (2014). *Stochastic backpropagation and approximate inference in deep generative models*. In: International conference on machine learning. Proceedings of Machine Learning Research, pp. 1278–1286.

Robbins, Monro (1951). *A stochastic approximation method*. In: The annals of mathematical statistics, Vol. 22, No. 3, pp. 400–407.

Robert, Casella, Casella (1999). *Monte Carlo statistical methods*. Vol. 2. Springer.

Rosen-Zvi, Griffiths, Steyvers, Smyth (2004). *The Author-Topic Model for Authors and Documents*. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence. UAI '04. AUAI Press, pp. 487–494.

Sachan, Contractor, Faruquie, Subramaniam (2012). *Using content and interactions for discovering communities in social networks*. In: Proceedings of the 21st international conference on World Wide Web, pp. 331–340.

Sampson (1969). "Crisis in a cloister". PhD thesis. Ph. D. Thesis. Cornell University, Ithaca.

Saul, Jordan (1995). *Exploiting tractable substructures in intractable networks*. In: Advances in neural information processing systems, Vol. 8, pp. 486–492.

Shi, Malik (2000). *Normalized cuts and image segmentation*. In: IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 8, pp. 888–905.

Snijders (2011). *Statistical models for social networks*. In: Annual review of sociology, Vol. 37, pp. 131–153.

Snijders, Nowicki (1997). *Estimation and prediction for stochastic blockmodels for graphs with latent block structure*. In: Journal of classification, Vol. 14, No. 1, pp. 75–100.

Srivastava, Sutton (2017). *Autoencoding Variational Inference For Topic Models*. In: International conference on machine learning.

Stoer, Wagner (1997). *A simple min-cut algorithm*. In: Journal of the ACM, Vol. 44, No. 4, pp. 585–591.

Thébault, Fontaine (2010). *Stability of ecological communities and the architecture of mutualistic and trophic networks*. In: Science, Vol. 329, No. 5993, pp. 853–856.

Van der Maaten, Hinton (2008). *Visualizing data using t-SNE*. In: Journal of machine learning research, Vol. 9, No. 86, pp. 2579–2605.

Vayansky, Kumar (2020). *A review of topic modeling methods*. In: Information Systems, Vol. 94, p. 101582.

Von Luxburg (2007). *A tutorial on spectral clustering*. In: Statistics and computing, Vol. 17, No. 4, pp. 395–416.

Wainwright, Jordan (2008). *Graphical models, exponential families, and variational inference*. In: Foundations and Trends® in Machine Learning, Vol. 1, No. 1–2, pp. 1–305.

Watts, Strogatz (1998). *Collective dynamics of 'small-world' networks*. In: Nature, Vol. 393, No. 6684, pp. 440–442.

Wu (1983). *On the convergence properties of the EM algorithm*. In: The Annals of statistics, Vol. 11, No. 1, pp. 95–103.

Wu, Dong, Nguyen, Luu (2023). *Effective neural topic modeling with embedding clustering regularization*. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 37335–37357.

Xu, Ke, Wang (2014). *A Fast Inference Algorithm for Stochastic Blockmodel*. In: 2014 IEEE International Conference on Data Mining, pp. 620–629.

Zanghi, Volant, Ambroise (July 2010). *Clustering based on Random Graph Model embedding Vertex Features*. In: Pattern Recognition Letters, Vol. 31, pp. 830–836.

Zhang, Horvath (2005). *A general framework for weighted gene co-expression network analysis*. In: Statistical applications in genetics and molecular biology, Vol. 4, No. 1, pp. 1–45.

Zhao, Phung, Huynh, Jin, Du, Buntine (2021). *Topic modelling meets deep neural networks: A survey*. arXiv: 2103.00498 [cs.LG].

Zhao, Phung, Huynh, Le, Buntine (2021). *Neural Topic Model via Optimal Transport*. In: International Conference on Learning Representations.

Zhou, Manavoglu, Li, Giles, Zha (2006). *Probabilistic models for discovering e-communities*. In: Proceedings of the 15th international conference on World Wide Web, pp. 173–182.

Zreik, Latouche, Bouveyron (2017). *The dynamic random subgraph model for the clustering of evolving networks*. In: Computational Statistics, Vol. 32, No. 2, pp. 501–533.