Extended Stochastic Block Models for Recommendations

Lorenzo Costa

18 July 2025

Acknowledgements

Ci tengo a ringraziare innanzitutto il Prof. Daniele Durante per la grande disponibilità dimostrata, per i consigli, l'attenzione e per avermi trasmesso la passione per la materia.

Vorrei dedicare questa tesi a tutta la mia famiglia. First and foremost, a mamma e papà per avermi supportato in ogni cosa, per non avermi mai fatto mancare nulla e soprattutto per avermi sempre amato incondizionatamente. Tutto ciò che ho lo devo (quite literally) a voi. A ricky e bubi, la vostra presenza è stata fondamentale per spingermi a migliorarmi ogni giorno. Al nonno e la nonna per esserci sempre e per il vostro entusiasmo per qualsiasi traguardo raggiunto (spesso anche più grande del mio).

Un ringraziamento enorme va agli splendidi 🧚. A Calu, Cami, Clari, Diana, Elliot, Let, Lollo, Marco e Mari per questi (quasi) 10 anni insieme. Per avermi kept sane, per le conversazioni più profonde e per quelle su argomenti random, per avermi insegnato a dare peso a ciò che conta veramente, per essere la mia valvola di sfogo, per avermi sempre fatto sentire bene, per tutto il divertimento e le risate. Sapendo di poter contare su di voi non mi sono mai sentito solo.

Last but not least, un ringraziamento speciale va a Angelo, Ema (1 e 2), Fabio, Fede, Leo e Mapi per aver condiviso con me il divertimento (e la sofferenza) di questi ultimi due anni in università, per avermi spinto con il vostro entusiamo a impegnarmi ogni giorni e aspirare a traguardi sempre più grandi.

Grazie :)

Contents

Chapter 1

Introduction

Consumers in the 21st century are overwhelmed with choices. Whether it's picking a movie to watch, a product to buy or a song to listen to we have at our disposal a seemingly endless amount of different options. While such variety can be a positive thing it also presents significant challenges: how can individuals efficiently find items that match their preferences ? How can we avoid spending hours browsing the whole Netflix catalogue to find the next TV show to watch?

This is where recommender systems come into play. Recommender systems are a class of algorithms created to predict which items a user would consume next based on their past behaviours, the behaviour of similar users or stated preferences. They act as personalized guides, helping individuals navigate large catalogues of options by highlighting the content most relevant to them. From e-commerce platforms to music streaming services, from social media feeds to online education portals, recommender systems have become a fundamental part of the digital experience.

Historically, these systems have been used primarily by companies, especially content-hosting platforms (e.g. Netflix, Spotify, Pinterest) and e-commerce services (e.g. Amazon, eBay), to both increase revenues and improve the user's experience. For business purposes, effective recommendation strategies can drive user engagement, increase

sales, and foster customer loyalty. For users, they can transform overwhelming environments into tailored, enjoyable experiences, uncovering products or content they might not have discovered otherwise. Recently however these systems are being incorporated in new fields such as for personalised education (Khanal et al., 2020) or in the healthcare industry (Etemadi et al., 2023).

This thesis aims at exploring a promising perspective to develop such systems: Stochastic Block Models. Indeed, one powerful way to tackle the problem of recommending items is thinking in terms of networks. We can represent users and items as points (or "nodes") connected by lines (or "edges") whenever there is an interaction, like a user rating a movie. These user-item networks can reveal hidden patterns and groupings that are not obvious at first glance. To make sense of these structures, researchers have developed mathematical tools called Stochastic Block Models (SBMs). These models assume that there are underlying "communities" groups of users and items that behave similarly and try to uncover them by looking at the networks overall structure.

Building on this idea, we focus on nonparametric SBMs, which generalize classical SBMs to account for uncertainty in the number of communities, within the framework of Extended Stochastic Block Models (ESBMs) introduced by Legramanti et al. (2022). While ESBMs have been developed for undirected, unweighted graphs, this work extends the formulation to weighted, bipartite graphs, which better reflect the structure and requirements of recommender systems. This adaptation allows for modelling interaction strengths (e.g., ratings) and the distinct roles of users and items.

In addition, we address a key limitation of classical SBMs and ESBMs that is their inability to model degree heterogeneity within communities. This can be problematic in real-world recommendation settings, where some users are much more active than others, or certain items are far more popular, regardless of which group they belong to. To address this, we introduce a degree-corrected Extended Stochastic Block Model. This version of the model incorporates node-specific parameters that separate a nodes level of activity or popularity from its community assignment. This modification retains the community-based structure of the model while allowing for more realistic degree

distributions, possibly improving its ability to capture real-world interaction patterns in recommendation data.

The remainder is structured as follows: in Chapter 2 we present the basis of Stochastic Block Models with a focus on the framework of Extended Stochastic Block Models, in Chapter 3 we present the adaptation of ESBM to the specific needs of recommender systems , including the proposed degree-corrected extension, and in Chapter 4 we apply the models to simulated and real-world data and compare them with other popular methods.

Chapter 2

Stochastic Block Models

2.1 Network-structured data

Network data are a common type of data encountered in many fields. They are used to model relationships between entities and can be represented as graphs.
More formally, we define a graph as a pair $G = (V, E)$, where V is a set of nodes and E is a set of edges. A typical representation of a graph is the adjacency matrix $\mathbf{Y}$, where the entries y_{uv} represent the relationship between vertices u and v.

Starting from this general definition, there are many possible ways to represent a network. In a broad formulation, the entries of the adjacency matrix $\mathbf{Y}$ can take values in a set $\mathcal{A} = \{\alpha_0, \ldots, \alpha_n\}$, which encodes the types of relations between nodes. For example:

- $\mathcal{A} = \{0, 1\}$ corresponds to binary relations (e.g., presence or absence of a link),
- $\mathcal{A} = \mathbb{R}$ corresponds to weighted relations,
- $\mathcal{A} = \{0, 1, 2\}$ might represent ternary relations, and so on.

In addition to choosing the type of relation, we can impose structural constraints on the network. One common choice is whether to allow self-loops (i.e. edges of the form $u \longrightarrow v$) indicating a node having a relation with itself. Another key distinction arises from the symmetry of the adjacency matrix. If the matrix $\mathbf{Y}$ is symmetric, meaning that $y_{vu} = y_{uv}$ for all pairs v, u, the network is called an *undirected graph*, as the relation is mutual. On the other hand, if symmetry is not enforced (i.e. $y_{vu} \neq y_{uv}$) is allowed, the network is said to be a *directed graph*, where relations have a defined direction from one node to another.

In the rest of this thesis, we will focus on models where $\mathcal{A} = \{0, 1\}$ or $\mathcal{A} = \{0, \ldots, \alpha_n\}$ $\alpha_n \in \mathbb{N}$, the adjacency matrix is symmetric with no entries on the diagonal (no self-loops). We note however that most results can be easily generalized to other cases.

2.2 Classical SBM

A popular task in network analysis is *community detection*, which involves identifying a grouping structure from an observed network. A widely used model to address this problem is the *Stochastic Block Model* (SBM).
This model was first developed in the context of social sciences by Holland et al. (1983) as a combination of two powerful approaches to the study of relational data: block models (White et al., 1976) and stochastic models for digraphs (Holland and Leinhardt, 1981). In particular, Holland et al. (1983) generalized the concept of *structural equivalence* (White et al., 1976) by introducing its probabilistic counterpart, called *stochastic equivalence*.

Structural equivalence defines an equivalence relation over the nodes in a network such that two nodes belong to the same class (or block) if they relate to all other nodes in the same way (Snijders and Nowicki, 1997). The probabilistic generalization proposed by Holland et al. (1983) states that two nodes v and u are stochastically equivalent if and only if the probability of any event involving the random graph is unchanged by interchanging nodes v and u.

This concept is fundamental for the notion of SBMs. In fact, Wasserman and Anderson (1987) define nodes to be stochastically equivalent if they belong to the same block in a stochastic block model.

A formal definition of SBMs can be given as:

Definition 2.2.1. *A Stochastic Block Model is a random graph $G = (V, E)$ with V nodes and H communities such that the probability of an edge between two nodes u and v is given by*

$$\mathbb{P}(y_{uv} = \alpha_j) = f(\alpha_j, z_u, z_v)$$

where z_u and z_v are the community assignments of u and v, respectively.

The role of stochastic equivalence is immediate from this definition: in an SBM, the probability distribution of the entire graph is invariant under permutations of nodes within the same block.
A key distinction when working with SBMs is whether the block structure is known (in which case the model is referred to as *a priori block modelling*), or must be inferred from data (referred to as *a posteriori block modelling*). The latter case has been extensively studied in the literature, with key contributions from Nowicki and and (2001) and Snijders and Nowicki (1997), who analysed the case of a posteriori SBM with a fixed number of communities.

Consider the case of an unweighted graph (i.e., edges are either 0 or 1) with V nodes. A natural choice is to model each entry of the adjacency matrix using a Bernoulli distribution. To complete the model, define:

- a vector $\mathbf{z} = \{z_1, \ldots, z_V\}$, which specifies the community assignment of each node (i.e., if $z_u = 1$, node u belongs to the first group),
- a matrix $\Theta = \{\theta_{hk}\}$, where each entry specifies the probability of having an edge between two groups:

$$\theta_{hk} = \mathbb{P}(y_{uv} = 1 \mid z_u = h, z_v = k)$$

We can then model the adjacency matrix $\mathbf{Y}$ as:

$$y_{uv} \mid \mathbf{z}, \Theta \overset{\text{iid}}{\sim} \text{Bernoulli}(\theta_{z_u, z_v}) \quad \text{for } u, v = 1, \ldots, V$$

We also assume that $z_u \in \{1, \ldots, \bar{H}\}$, where $\bar{H}$ is the maximum number of non-empty communities. Note that there is a distinction between the *actual* number of non-empty clusters H and the *maximum* number of non-empty clusters $\bar{H}$.

Following Schmidt and Morup (2013), we can obtain a hierarchical formulation of the model by assuming that the vector $\mathbf{z}$ is drawn from a multinomial distribution with a Dirichlet prior, and by imposing a Beta prior on the entries of Θ:

$$\begin{aligned}
\boldsymbol{\pi} \mid \gamma &\sim \text{Dirichlet}(\gamma) \\
z_u \mid \boldsymbol{\pi} &\sim \text{Categorical}_{\bar{H}}(\boldsymbol{\pi}) \\
\theta_{hk} \mid a, b &\sim \text{Beta}(a, b) \\
y_{uv} \mid \mathbf{z}, \Theta &\sim \text{Bernoulli}(\theta_{z_u, z_v})
\end{aligned}$$

Note that this formulation can be easily generalized to a network with weighted edges by exploiting the Poisson-Gamma conjugacy:

$$\begin{aligned}
\boldsymbol{\pi} \mid \gamma &\sim \text{Dirichlet}(\gamma) \\
z_u \mid \boldsymbol{\pi} &\sim \text{Categorical}_{\bar{H}}(\boldsymbol{\pi}) \\
\theta_{hk} \mid a, b &\sim \text{Gamma}(a, b) \\
y_{uv} \mid \mathbf{z}, \Theta &\sim \text{Poisson}(\theta_{z_u, z_v})
\end{aligned}$$

2.3 Generalisation of SBM

Starting from the foundational formulation presented in Schmidt and Morup (2013) numerous extensions of the model have been proposed aimed at capturing different structural patterns in the data. In the remainder of this section we present a selective, though not exhaustive, overview of some of the most popular developments in this line of research.

2.3.1 Mixed Membership Stochastic Block Model

One possible extension is to move away from a hard clustering approach and allow nodes to belong to multiple communities. This model, developed by Airoldi et al. (2008), is known as the *Mixed Membership Stochastic Block Model* (MMSBM).

As in the classical SBM, we have a matrix $\Theta \in \mathbb{R}^{H \times H}$ that contains the probabilities of edges between any pair of communities. Each node is associated with a vector $\boldsymbol{\pi} \in \mathbb{R}^{H}$ representing the probability distribution over the H communities on which we impose a Dirichlet prior. For each pair of nodes (u, v) we draw two membership indicators $z_{u,v}, z_{v,u}$ and then draw the entry of the matrix from a Bernoulli (when the network is undirected) with mean $\theta_{z_{u,v},z_{v,u}}$.

The model can be written in hierarchical form as:

$$
\begin{aligned}
\boldsymbol{\pi}_u \mid \gamma &\sim \text{Dirichlet}(\gamma) \\
z_{u,v} \mid \boldsymbol{\pi}_u &\sim \text{Categorical}_{\bar{H}}(\boldsymbol{\pi}_u) \\
z_{v,u} \mid \boldsymbol{\pi}_v &\sim \text{Categorical}_{\bar{H}}(\boldsymbol{\pi}_v) \\
\theta_{hk} \mid a, b &\sim \text{Beta}(a, b) \\
y_{uv} \mid z_{u,v}, z_{v,u}, \Theta &\sim \text{Bernoulli}(\theta_{z_{u,v},z_{v,u}})
\end{aligned}
$$

The use of two membership indicators makes group membership *context-dependent*, meaning that a node's group assignment can vary depending on the interaction partner.

2.3.2 Non-parametric Stochastic Block Model

A clear limitation of the models presented so far is that the number of communities must be fixed or known a priori. A natural extension is to move to a non-parametric setting, where $\bar{H}$ is assumed to be infinite. The simplest non-parametric version of the SBM is the *Infinite Relational Model* (Kemp et al., 2006), which uses the distribution over partitions induced by the Chinese Restaurant Process (CRP) (Pitman, 2006).

The hierarchical formulation of the model is:

$$\begin{aligned}
\mathbf{z} \mid \alpha &\sim \text{CRP}(\alpha), \\
\theta_{hk} \mid a, b &\sim \text{Beta}(a, b), \\
y_{uv} \mid \mathbf{z}, \Theta &\sim \text{Bernoulli}(\theta_{z_u, z_v}),
\end{aligned}$$

where z_u, z_v are the group assignments for nodes u and v.

2.3.3 Degree-Corrected Stochastic Block Model

A known limitation of the standard SBM is that nodes within the same community are assumed to have the same expected degree. As pointed out by Karrer and Newman (2011), this degree homogeneity can lead the model to group nodes based on degree rather than actual community structure, possibly clustering high- and low-degree nodes. To address this issue, Karrer and Newman (2011) propose the *Degree-Corrected Stochastic Block Model* (DCSBM), which introduces node-specific parameters to account for degree heterogeneity within communities.

Building on this idea, Herlau et al. (2014) develop a Bayesian non-parametric version called the *Infinite Degree-Corrected Stochastic Block Model* (IDCSBM). For each node u such that $z_u = h$, we introduce a node-specific weight $\eta_u = n_h \phi_{h,u}$, with ϕ_h drawn from a Dirichlet distribution and n_h the number of nodes in group h.

The model can be written hierarchically as:

$$\begin{aligned}
\mathbf{z} \mid \alpha &\sim \text{CRP}(\alpha), \\
\phi_h \mid \gamma, \mathbf{z} &\sim \text{Dirichlet}(\gamma \mathbf{1}_{n_h}), \\
\eta_u &= n_{z_u} \phi_{z_u, u}, \\
\theta_{hk} \mid a, b &\sim \text{Gamma}(a, b), \\
y_{uv} \mid \eta_u, \eta_v, \mathbf{z}, \Theta &\sim \text{Poisson}(\eta_v \theta_{z_u, z_v} \eta_u),
\end{aligned}$$

where $\mathbf{1}_n$ denotes a vector of ones of size n.

2.4 Extended Stochastic Block Models

A unifying framework for the models presented so far is the *Extended Stochastic Block Model* (ESBM), proposed in Legramanti et al. (2022). This model provides two key advantages:

- it relies on Gibbs-type priors for the community assignment mechanism, allowing greater flexibility,
- by exploiting the connection between Gibbs-type priors and product partition models, it allows the introduction of covariates in the partition process.

2.4.1 Model formulation

Given an adjacency matrix $\mathbf{Y}$, which is assumed to represent an undirected, unweighted graph, the model is similar to the IRM (Kemp et al., 2006), but it replaces the Chinese Restaurant Process with the more general formulation of a Gibbs-type prior:

$$\begin{aligned} \mathbf{z} &\sim \text{Gibbs-type prior} \\ \theta_{hk} \mid a, b &\sim \text{Beta}(a, b) \\ y_{uv} \mid \mathbf{z}, \Theta &\sim \text{Bernoulli}(\theta_{z_u, z_v}) \end{aligned}$$

where:

- the vector $\mathbf{z}$ represents the cluster assignment (i.e., if $z_u = h$, then node u is assigned to group h),
- θ_{hk} is the probability of an edge between two nodes in groups h and k, respectively.

We can write the likelihood of the model as:

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{z}, \Theta) &\propto \prod_{u,v} y_{uv}^{\theta_{z_u, z_v}} (1 - y_{uv})^{1 - \theta_{z_u, z_v}} \\ &\propto \prod_{h=1}^{H} \prod_{k=1}^{K} \theta_{hk}^{m_{hk}} (1 - \theta_{hk})^{\bar{m}_{hk}} \end{aligned}$$

where m_{hk} is the number of edges and $\bar{m}_{hk}$ is the number of non-edges between nodes in groups h and k.

2.4.2 Prior specification

Gibbs-type priors are a class of discrete nonparametric priors in which, as shown in De Blasi et al. (2015), the probability of generating a new value (in our case, the probability of being assigned to an empty cluster) depends on the number of data points n, the number of non-empty clusters H and a finite-dimensional vector of parameters possibly entering the prior specification.

Being discrete nonparametric priors, Gibbs-type distributions can be written as:

$$\tilde{p} = \sum_{j=1}^{\infty} \tilde{p}_j \delta_{x_j}$$

where δ_x is the point mass at x, $(\tilde{p}_j)_{j\geq 1}$ is a sequence of non-negative random variables such that $\sum_j \tilde{p}_j = 1$, and $(x_j)_{j\geq 1}$ is a sequence of i.i.d. random variables from P^*, a diffuse probability measure. This property is key for clustering because it means Gibbs-type distributions induce a partition of the observations, defined by the ties observed almost surely in the data.

We can then characterise any Gibbs-type prior using the *Exchangeable Partition Probability Function* (EPPF) (Pitman, 2006) defined as:

$$p_H^{(V)}(n_1, \ldots, n_H) = \int_{\mathbb{X}^k} \mathbb{E}(\tilde{p}^{n_1}(dx_1), \ldots, \tilde{p}^{n_H}(dx_H))$$

which corresponds to the probability, induced by $\tilde{p}$, of observing a sample of size V, with H distinct observations having frequencies $n_1, \ldots, n_H$.

The EPPF for Gibbs-type priors can be written (De Blasi et al., 2015) as:

$$p_H^{(V)}(n_1, \ldots, n_H) = \mathcal{V}_{V,H} \prod_{i=1}^{H} (1-\sigma)_{n_i - 1}$$

where:

- $V \geq 1$, $h \leq V$, $\sum_{i=1}^{H} n_i = V$, $\sigma < 1$,
- $(x)_n$ denotes the rising factorial defined as

$$(x)_n = x(x+1)\cdots(x+n-1),$$

- $\mathcal{V}_{V,H}$ is a collection of non-negative weights such that

$$\mathcal{V}_{V,H} = (n - \sigma H)\mathcal{V}_{V+1,H} + \mathcal{V}_{V+1,H+1}.$$

  with $V_{1,1} = 1$.

In the ESBM case, we can use this formulation of the EPPF to write the probability distribution function of $p(\mathbf{z})$ as:

$$p(\mathbf{z}) = \mathcal{V}_{V,H} \prod_{h=1}^{H} (1 - \sigma)_{n_h - 1}.$$

The advantage of Gibbs-type priors is that they represent a broad class of distributions in which the group assignment process can be expressed as[1]:

$$p(z_{V+1} = h \mid \mathbf{z}) \propto \begin{cases} \mathcal{V}_{V+1,H}(n_h - \sigma) & \text{for } h = 1, \ldots, H, \\ \mathcal{V}_{V+1,H+1} & \text{for } h = H + 1. \end{cases} \tag{2.1}$$

This creates an interpretable *seating mechanism*, akin to the Chinese Restaurant Process, where the probability of being assigned to group h increases with the current size of that cluster n_h, discounted by a global parameter σ, and scaled by a weight $\mathcal{V}_{V+1,H}$, which depends on the number of nodes V and the number of non-empty clusters H.

For applications to clustering, it is particularly interesting to examine the role of the parameter σ. As shown by De Blasi et al. (2015), if $\sigma > 0$, a reinforcement mechanism occurs: the sampling procedure tends to favour clusters with higher frequency. If $\sigma < 0$, the inverse happens: the probability of being assigned to cluster h decreases as n_h grows. This latter case is more appropriate when the number of clusters is assumed to be finite.

The parameter σ also determines the rate at which the number of occupied clusters H grows with the data. Indeed, as shown in De Blasi et al. (2015), we can define:

$$c_V(\sigma) = \begin{cases} 1 & \sigma < 0, \\ \log V & \sigma = 0, \\ V^{\sigma} & \sigma \in (0, 1), \end{cases}$$

[1]Derivations can be found in the Appendix 5

and for any $V \geq 1$:

$$\frac{H}{c_V(\sigma)} \xrightarrow{\text{a.s.}} S_\sigma,$$

where S_σ is a limiting random variable called the *σ-diversity*. This implies that a larger σ leads to a faster growth rate of H.

Gibbs-type priors include many commonly used priors for SBMs, such as the Dirichlet-Multinomial model, the Dirichlet Process (e.g., Kemp et al., 2006), the PitmanYor process (e.g., Lee and Battiston, 2023), and the Gnedin process (e.g., Legramanti et al., 2022). We present next these 4 popular priors as well as remarks on when to use each of them in the context of ESBM.

Dirichlet–Multinomial model

The Dirichlet-Multinomial model arises when adopting the parametric formulation of Nowicki and and (2001) (presented in Section 2.2) where the maximum number of non-empty clusters is assumed to be fixed at $\bar{H}$ and the cluster assignment is chosen from a multinomial distribution with a Dirichlet prior.
We can recover this model from the Gibbs-type prior formulation by setting $\sigma < 0$ and considering the collection of weights

$$\mathcal{V}_{V,H} = \frac{\beta^{V+1}}{(\beta\bar{H}+1)_{H-1}} \prod_{h=1}^{H} (\bar{H}-h)\mathbb{1}(H \leq \bar{H})$$

These weights lead to the sampling scheme:

$$\mathbb{P}(z_{V+1} = h \mid \mathbf{z}) \propto \begin{cases} n_h + \beta & \text{for } h = 1, \dots, H \\ \beta(\bar{H}-H)\mathbb{1}(H \leq \bar{H}) & \text{for } h = H+1 \end{cases}$$

By using a Dirichlet-Multinomial model, we assume that the number of non-empty clusters is finite and upper bounded by a fixed quantity $\bar{H}$, which is specified a priori. This assumption holds even as the number of observations tends to infinity. Thus a Dirichlet-Multinomial model is appropriate when we are willing to assume that there

exists a true maximum number of clusters and that a reasonable upper bound for it can be identified in advance.

Dirichlet process (DP)

Introduced by Ferguson (1973) the Dirichlet process is a very popular Bayesian non-parametric prior. We can give a formal definition as:

Definition 2.4.1. *Let α be a positive real number and P_0 a non-null measure on $(\mathbb{X}, \mathcal{X})$. A random probability measure is said to be a Dirichlet Process, denoted as $\tilde{P} \sim DP(\alpha, P_0)$ if for every finite measurable partition $A_1, \ldots, A_H$ of $\mathbb{X}$ we have:*

$$(\tilde{P}(A_1), \ldots, \tilde{P}(A_H)) \sim Dir(\alpha P_0(A_1), \ldots, \alpha P_0(A_H))$$

In this formulation α is called the concentration parameter and P_0 the base distribution. The predictive distribution, that is the distribution of X_{V+1} given the observations up to V, can be written in terms of the number of unique values H and their multiplicity n_h as:

$$\mathbb{P}(X_{V+1} \in A \mid X_1, \ldots, X_V) = \frac{\alpha}{\alpha + V} P_0(A) + \frac{V}{\alpha + 1} \frac{1}{V} \sum_{h=1}^{H} n_h \delta_{X_h^*}(A)$$

We can find the DP from Gibbs-type priors for $\sigma = 0$ and setting $\mathcal{V}_{V,H} = \alpha^H / (\alpha)_V$. This leads to the well-known sampling scheme called *Chinese Restaurant Process* (Aldous, 1985):

$$\mathbb{P}(z_{V+1} = h \mid \mathbf{z}) \propto \begin{cases} n_h & \text{for } h = 1, \ldots, H \\ \alpha & \text{for } h = H + 1 \end{cases}$$

By using a Dirichlet process prior, we assume that the number of non-empty clusters, H, increases as the number of observations grows. However, H grows at a relatively slow rate, specifically $O(\log V)$. Therefore, this model is appropriate when we expect the number of clusters to increase with the number of nodes, but only at a logarithmic rate.

Pitman-Yor process (PY)

The two-parameter Poisson-Dirichlet process, introduced in Perman et al. (1992), and termed Pitman-Yor process in Ishwaran and and (2001) is a discrete random probability measure which generalises the DP by introducing a parameter σ controlling the growth rate of the number of distinct groups H.

The predictive distribution can be written as:

$$\mathbb{P}(X_{V+1} \in A \mid X_1, \ldots, X_V) = \frac{\alpha + \sigma H}{\alpha + V} P_0(A) + \frac{V - \sigma H}{\alpha + V} \frac{1}{V - \sigma K} \sum_{h=1}^{H} (n_h - \sigma) \delta_{X_h^*}(A)$$

where admissible values for σ are $\sigma \in [0, 1)$ with $\alpha > -\sigma$ or $\sigma < 0$ with $\alpha = m|\sigma|$ for some positive integer m (De Blasi et al., 2015). From this formulation it is clear that we can recover the DP by setting $\sigma = 0$.

We can find the PY from Gibbs-type distributions for $\sigma \in [0, 1)$ and by setting:

$$\mathcal{V}_{V,H} = \frac{\prod_{h=1}^{H} (\alpha + h\sigma)}{(\alpha + 1)_{V-1}}$$

which leads to the sampling scheme:

$$\mathbb{P}(z_{V+1} = h \mid \mathbf{z}) \propto \begin{cases} n_h - \sigma & \text{for } h = 1, \ldots, H \\ \alpha + H\sigma & \text{for } h = H + 1 \end{cases}$$

As with the Dirichlet process, using the two–parameter Poisson-Dirichlet (Pitman-Yor) process implies that the number of occupied clusters diverges as the network grows. However, unlike the DP, the PY process offers greater flexibility in controlling the rate of growth. Specifically, the number of clusters H grows at a rate of $O(V^\sigma)$. By setting $\sigma \approx 1$, we can model much faster growth compared to the logarithmic rate of the Dirichlet process. This prior is thus suitable when we believe the true number of clusters is infinite in the limit $V \to \infty$, and the growth rate is not logarithmic.

Gnedin process (GN)

The Gnedin process can be recovered from a Dirichlet-Multinomial formulation by adding a prior on the maximum number of groups $\bar{H}$ (Legramanti et al., 2022) of the form:

$$\mathbb{P}(\bar{H} = h) = \frac{\gamma(1-\gamma)_{h-1}}{h!}$$

for $\gamma \in (0, 1)$ which has mode at 1, heavy tail and infinite expectation (Gnedin, 2010).

The sampling scheme generated by this distribution is

$$\mathbb{P}(z_{V+1} = h \mid \mathbf{z}) \propto \begin{cases} (n_h + 1)(V - H + \gamma) & \text{for } h = 1, \dots, H \\ H^2 - H\gamma & \text{for } h = H + 1 \end{cases}$$

An interesting feature of the Gnedin process is that, similarly to the Dirichlet-Multinomial model, it allows the number of occupied clusters to remain finite even as V diverges. Indeed, unlike the DP and PY priors, the Gnedin process defines a proper prior distribution over the number of clusters as the network size increases. Specifically the asymptotic prior is:

$$p_{GN}(H = h) = \binom{V}{h} \frac{(1-\gamma)_{h-1}(\gamma)_{V-h}}{(1+\gamma)_{V-1}} \quad \text{for } h = 1, \dots, V$$

This formulation allows us to compute the expected value of H as:

$$E(H) = \sum_{h=1}^{V} h \cdot p(H = h)$$

which can be used as a guide to select the hyperparameters. Moreover, unlike the DP and PY processes, when the data are generated from a model with a finite true number of clusters (i.e. $\bar{H} < \infty$) a GN prior leads to consistent estimates for $\bar{H}$ (Geng et al., 2019).

Given these properties, the Gnedin process is a suitable choice when we believe that the true number of clusters remains finite as the number of nodes grows, but still want to treat this quantity as random.

2.4.3 Connection with product partition models

As shown by De Blasi et al. (2015), Gibbs-type priors also exhibit an interesting connection with product partition models (PPMs), a class of models (Hartigan, 1990) where the probability distribution of a partition $\mathbf{z}$ can be written as:

$$p(\mathbf{z}) \propto \prod_{h=1}^{H} \rho(Z_h)$$

where $\{Z_1, \ldots, Z_H\}$ is the partition induced by $\mathbf{z}$, and $\rho(\cdot)$ is a *cohesion function* measuring homogeneity within each cluster. In particular, Gibbs-type priors are a specific case of PPMs where the cohesion function depends only on the cardinality of Z_h (De Blasi et al., 2015).

The product partition model with covariates (PPMx), introduced by Müller et al. (2011), extends the PPM formulation by allowing the distribution of each partition to depend also on a *similarity function* $g(\cdot)$, which measures the homogeneity of each cluster in terms of some external covariates $\mathbf{X}$. The probability of a specific partition, conditional on the covariates, can then be written as:

$$p(\mathbf{z} \mid \mathbf{X}) \propto \prod_{h=1}^{H} g(\mathbf{X}_h) \cdot \rho(Z_h)$$

where $\mathbf{X}_h = \{\mathbf{x}_i : i \in Z_h\}$ denotes the sub-matrix of covariates corresponding to nodes in cluster h.

There are many possible choices for the similarity function. These include functions based on empirical cluster variance or entropy (Page and Quintana, 2018), Gower dissimilarity (Gower, 1971), or an auxiliary model (Quintana et al., 2015). A common and practical choice, proposed by Müller et al. (2011), is to define $g(\mathbf{X}_h)$ via the marginal probability induced by an auxiliary model:

$$g(\mathbf{X}_h) = \int \prod_{u \in Z_h} q(x_u \mid \xi_h) q(\xi_h) \, d\xi_h$$

Note that this approach can also be used when the covariates are not considered random; in that case, the probability distribution is simply a convenient computational device.

Depending on the type of covariates, the similarity function takes different forms:

- for *categorical* variables, a common choice is a Dirichlet-Multinomial model, resulting in:

$$p(\mathbf{X}_h) \propto \frac{1}{\Gamma(n_h + \alpha_0)} \prod_{c=1}^{C} \Gamma(n_{hc} + \alpha_c)$$

  where n_{hc} is the number of nodes in cluster h with covariate category c, $\{\alpha_c\}$ is a vector of Dirichlet prior parameters, and $\alpha_0 = \sum_c \alpha_c$,

- for *count* covariates, a common choice is a mixture of Poisson distributions with a conjugate Gamma prior:

$$\begin{aligned} p(\mathbf{X}_h) &= \frac{1}{\prod_{i:z_i=h} x_i!} \int \xi_h^{\sum_{i:z_i=h} x_i} \exp(-n_h \xi_h)\, dq(\xi_h) \\ &\propto \int \xi_h^{\sum_{i:z_i=h} x_i} \xi_h^{a-1} e^{-n_h \xi_h} e^{-b\xi_h}\, d\xi_h \\ &\propto \frac{\Gamma\left(a + \sum_{i:z_i=h} x_i\right)}{(b + n_h)^{a+\sum_{i:z_i=h} x_i}} \end{aligned}$$

- for *continuous multivariate* covariates, a possible choice is the Gaussian-Inverse Wishart model (Ghidini et al., 2023), which yields:

$$\begin{aligned} g(\mathbf{X}_h) = \frac{2^{p(\nu_0+n_h)/2}}{\sqrt{k_0 + n_h}} \Gamma_p\left(\frac{\nu_0 + n_h}{2}\right) \cdot \\ \cdot \det\left\{\Sigma_0^{-1} + \frac{k_0 n_h}{k_0 + n_h}(\bar{x}_h - \mu_0)^T(\bar{x}_h - \mu_0) + \sum_{v:z_v=h} (x_v - \bar{x}_h)^T(x_v - \bar{x}_h)\right\}^{-\frac{\nu_0+n_h}{2}} \end{aligned}$$

  where $\Gamma_p(x) = \pi^{(p-1)/2}\Gamma_{p-1}(x - 1/2)$ with $\Gamma_1(\cdot) = \Gamma(\cdot)$ being the standard Gamma function.

2.4.4 Posterior computation

The core of Bayesian inference is to obtain the posterior distribution of the parameters given the data and study the properties of this distribution. For ESBM, however, the posterior is not available in closed form. A common solution in these cases is to

approximate it using *Markov Chain Monte Carlo* (MCMC) methods, which are a class of algorithms used to draw samples from a probability distribution by constructing a Markov chain that has the target distribution as its stationary distribution. Among MCMC algorithms, a widely used method is the *Gibbs sampler* (Geman and Geman, 1984), which is a special case of MCMC where we sample from the full conditional distribution of each parameter given the other parameters and the data.

Legramanti et al. (2022) propose a *collapsed Gibbs sampler* to approximate the posterior of the ESBM, which is a version of Gibbs sampling where we *collapse* (i.e. integrate out) some of the variables by integrating them out from the posterior. The advantage of this method is that it reduces the computational cost, since we need to sample a smaller number of parameters. In this case, we collapse the block probabilities parameter Θ, and, exploiting beta-binomial conjugacy, we get:

$$p(\mathbf{Y} \mid \mathbf{z}) = \prod_{h=1}^{H} \prod_{k=1}^{K} \frac{B(a + m_{hk}, b + \bar{m}_{hk})}{B(a, b)}$$

where $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ is the Beta function.

Having marginalised out the block probabilities, at each iteration we sample the node assignment z_u from the full conditional distribution, which is the distribution given the adjacency matrix $\mathbf{Y}$ and the vector $\mathbf{z}_{-u}$ (i.e., cluster assignments excluding node u). These full conditionals follow a Categorical distribution with probabilities:

$$\begin{aligned} \mathbb{P}(z_u = h \mid \mathbf{Y}, \mathbf{X}, \mathbf{z}_{-u}) &\propto \mathbb{P}(z_u = h \mid \mathbf{X}, \mathbf{z}_{-u}) \frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u})} \propto \\ &\propto \mathbb{P}(z_u = h \mid \mathbf{z}_{-u}) \frac{p(\mathbf{X}_h)}{p(\mathbf{X}_{h,-u})} \frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u})} \end{aligned} \tag{2.2}$$

where $\mathbf{Y}_{-u}$ is the adjacency matrix excluding the u-th row and column, and $\mathbf{X}_{h,-u}$ represents the covariates for nodes in group h excluding node u.

We can compute the first term of 2.2 from the urn scheme described in Section 2.4.2 (equation 2.1):

$$\mathbb{P}(z_u = h \mid \mathbf{z}_{-u}) \propto \begin{cases} \mathcal{V}_{V,H^-}(n_h^- - \sigma) & \text{for } h \leq H^- \\ \mathcal{V}_{V,H^-+1} & \text{for } h = H^- + 1 \end{cases}$$

where H^- is the number of clusters without item v, and n_h^- is the cardinality of cluster h excluding node u. The last term instead depends on the form of the likelihood. When using the Beta-Bernoulli model, we obtain (Legramanti et al., 2022):

$$\frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u})} = \prod_{k=1}^{H} \frac{B(a + m_{hk}^- + y_{uk},\ b + \bar{m}_{hk}^- + \bar{y}_{uk})}{B(a + m_{hk}^-,\ b + \bar{m}_{hk}^-)}$$

where quantities with the superscript $^-$ are computed excluding node u, and y_{uk}, $\bar{y}_{uk}$ denote the number of edges and non-edges between node u and the nodes in cluster k.

When including covariates, we also need to compute the ratio $p(\mathbf{X}_h)/p(\mathbf{X}_{h,-u})$. This depends on the type of covariate. For instance:

- for categorical (or binary) covariates, this takes the form[2]:

$$\frac{p(\mathbf{X}_h)}{p(\mathbf{X}_{h,-u})} \propto \frac{n_{hc}^- + \alpha_c}{n_h^- + \alpha_0}$$

- for count-type covariates, this takes the form:

$$\frac{p(\mathbf{X}_h)}{p(\mathbf{X}_{h,-u})} = \frac{\Gamma(a + \sum_{i:z_i=h} x_i)}{(b + n_h)^{a+\sum_{i:z_i=h} x_i}} \cdot \frac{(b + n_h^-)^{a+\sum_{i\neq u:z_i=h} x_i}}{\Gamma(a + \sum_{i\neq u:z_i=h} x_i)}$$

2.4.5 Cluster assignment estimation

After obtaining samples from the posterior via the Gibbs sampling algorithm described in Section 2.4.4, we must estimate the cluster assignments. A key advantage of the ESBM, and of Bayesian methods in general, over algorithmic strategies is that we have access to the full posterior distribution. This is beneficial because it allows us to construct principled point estimates and perform inference. However, it also presents a challenge: we need a way to summarize the information contained in the posterior.

In practice, MCMC methods yield a large number of posterior samples. Due to the vast size of the partition space and the fact that many of the sampled partitions are very similar, the posterior mass tends to be spread over a wide range of distinct partitions.

[2]Derivation in Appendix 5

Since listing all sampled partitions would be infeasible, we must adopt a principled approach to summarization.

A straightforward solution is to report the posterior mode, that is the partition most frequently visited by the chain. However, in high-dimensional partition spaces, frequency counts can be unreliable, as it is common for the chain to visit most partitions only once. Among the various strategies proposed to address this issue, we adopt the approach introduced by Wade and Ghahramani (2018). Within the framework of decision theory, we define a loss function $\mathcal{L}$ over the space of partitions and then look for a clustering that minimizes this loss:

$$\hat{\mathbf{z}} = \arg\min_{\hat{\mathbf{z}}} \mathbb{E}[\mathcal{L}(\mathbf{c}, \hat{\mathbf{c}}) \mid y_{1:n}]$$

The loss function proposed in Wade and Ghahramani (2018) is the *variation of information* (VI) which compares the information in two clustering with the information shared between them.

Having two clusterings $\mathbf{z}$ and $\hat{\mathbf{z}}$, define n_{ij} the number of points that are labelled as i in the first and j in the second and set $n_{i+} = \sum_j n_{ij}$. We define the *entropy* (H) and *mutual information* (I) between $\mathbf{z}$ and $\hat{\mathbf{z}}$ as:

$$H(\mathbf{z}) = \sum_{i=1}^{H} \frac{n_{i+}}{n} \log_2\left(\frac{n_{i+}}{n}\right) \quad I(\mathbf{z}, \hat{\mathbf{z}}) = \sum_{i=1}^{H} \sum_{j=1}^{\hat{H}} \frac{n_{ij}}{n} \log_2\left(\frac{n_{ij}}{n}\right)$$

A formal definition of the VI metric, which ranges from 0 to $\log(V)$, can be given as:

$$VI(\mathbf{z}, \hat{\mathbf{z}}) = H(\mathbf{z}) + H(\hat{\mathbf{z}}) - 2I(\mathbf{z}, \hat{\mathbf{z}})$$

Having this loss function between two clusterings a Bayesian estimate of $\mathbf{z}$ and $\mathbf{q}$ can be computed minimizing the expected VI distance as:

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}'} \mathbb{E}_{\mathbf{z}}[VI(\mathbf{z}, \mathbf{z}') \mid \mathbf{Y}]$$

Solving this optimization requires computing the VI distance for all possible partitions $\mathbf{z}'$. Given the enormous size of the partition space, this is infeasible in most practical settings. To address this, Wade and Ghahramani (2018) propose a greedy search

algorithm that, starting from a given clustering $\mathbf{z}$, explores a set of partitions in its neighbourhood and moves in the direction of the one that minimizes the expected VI distance $\mathbb{E}_{\mathbf{z}}[VI(\mathbf{z}, \mathbf{z}') \mid \mathbf{Y}]$.

A notable advantage of this method is that it can identify partitions not present in the MCMC samples. As the authors observe, in most examples the selected partition, that is the one minimizing the expected loss, does not appear among those visited by the chain. In our application in Chapter 4, we follow the implementation provided in the R package `mcclust.ext` (Wade and Ghahramani, 2018).

Chapter 3

ESBM for recommender systems

3.1 Overview of recommender systems

Recommender systems are a subclass of information filtering systems used to suggest items that a user might be interested in based on user's past consumption, the behaviour of similar users and, depending on the case, some other external information about the user and items (Raza et al., 2025). The main goal of recommender systems is to enhance the user experience, increase engagement and facilitate decision-making. Historically the main use has been for content-hosting platforms (e.g. Netflix, Spotify, Pinterest) or e-commerce services (e.g. Amazon, eBay), however recently these systems are being incorporated in new fields such as for personalised education (Khanal et al., 2020) or in the healthcare industry (Etemadi et al., 2023).

There are many methods that fall inside this category. A broad classification can be given depending on the type of information used to make recommendations:

- *Content-Based Filtering* (CBF) where we create a profile of each user or item to characterise its nature. The data used to create these profiles are usually external

information. For instance a user's profile could include demographic information and data gathered from survey,

- *Collaborative Filtering* (CF) where recommendations are made based on patterns of behaviour of many users. The core idea is that if users u and i have liked similar items in the past, the next item rated positively by u is likely to be relevant for i as well. The information leveraged by these algorithms are then only the user-item interactions,
- *hybrid approaches* that combine the strength of CF and CBF to get more accurate recommendations and balance different interests (e.g. accuracy, novelty, fairness).

A significant appeal of CF methods is that they are largely *domain free* and tend to be more accurate than content-based models (Hu et al., 2008). Among CF methods a popular class of that of *Matrix Factorization* (MF). MF is a subset of a large class called *latent factor model* whose aim is to characterize both users and items using a small number of factors inferred from rating patterns. This is done by decomposing the matrix collecting user-item interactions using lower dimensional matrices as:

$$\mathbf{Y} = \mathbf{QP}$$

If we assume that we are using a latent space of dimension H then we have that the matrix $\mathbf{Q}$ is made of vectors $q_i \in \mathbb{R}^H$, one for each item, and similarly the matrix $\mathbf{P}$ is made of vectors $p_u \in \mathbb{R}^H$, one for each user. From this representation is it clear that the rating for each pair (u, i) is computed as the inner product of q_i and p_u.
The vectors q_i and p_u can be learned in a variety of different ways. In its most basic this is done by minimising the mean squared error with a L2 penalty (Koren et al., 2009):

$$\min_{q,p} \sum_{(u,i)} (y_{ui} - q_i^T p_u)^2 \lambda_q \|q_i\|^2 + \lambda_p \|p_u\|^2 \tag{3.1}$$

There are several extension and variations of this basic model such as Non-Negative Matrix factorization (Paatero and Tapper, 1994; Lee and Seung, 1999) where we assume the entries of $\mathbf{Q}$ and $\mathbf{P}$ are non negative or Probabilistic Matrix factorization which is a probabilistic version of MF based on a linear model with Gaussian observation noise (Salakhutdinov and Mnih, 2007)

3.2 Why SBM

After having described the theoretical foundations of Stochastic Block Models (SBMs) and introduced the basics of recommender systems, we now turn to the rationale for adopting SBMs in the context of recommendations.

As mentioned in the Introduction 1, the primary motivation comes from the natural representation of user-item interactions as a graph structure. Specifically, we can model these data as a bipartite network, where one set of nodes corresponds to users and the other to items.

Definition 3.2.1. *A bipartite graph* (V, E) *is a graph in which the set of nodes* V *is partitioned into two disjoint subsets* U *and* I *such that every edge connects a node from* U *to a node from* I *or vice versa.*

SBMs are designed to capture the latent group structure in such graphs making them particularly suitable to uncover the latent communities in both users and items based on their patterns of interactions. This aligns well with the intuition behind many recommender systems: users with similar preferences often interact in a similar way with groups of items.

Another compelling reason for employing SBMs comes from their theoretical connection to Matrix Factorization (MF) techniques. As shown by Zhang et al. (2018), likelihood maximization in SBMs is equivalent to a constrained form of non-negative matrix factorization (NMF). Moreover, these results extend naturally to more sophisticated variants such as bipartite and degree-corrected SBMs.

3.3 Model extension

A straightforward extension of the model presented so far is to have two vectors specifying the community assignments: one for the users $\mathbf{z}$ and one for the items $\mathbf{q}$:

$$\mathbf{z} \sim \text{Gibbs-type}$$
$$\mathbf{q} \sim \text{Gibbs-type}$$

Moreover in recommender systems we typically encounter two types of data:

- *explicit feedback* data: users provide a rating for each item they have consumed. The observations are thus count data,
- *implicit feedback* data: we observe only whether a user has consumed a certain item. The assumption in this case is that users only consume items they "like".

When dealing with *explicit feedback* data, we must modify the likelihood to account for the fact that $y_{ui} \in \{1, 2, \dots\}$. The remainder of this chapter assumes that we are in the explicit feedback setting, but the models can also accommodate implicit feedback data without changing the overall formulation.

When the entries of the adjacency matrix are integers, we resort to a Poisson likelihood and a Gamma prior on the rate of the Poisson distribution. The choice of a Gamma prior has two advantages:

- it simplifies computations thanks to the conjugacy of the Gamma-Poisson model,
- it allows us to impose sparsity on the model, a characteristic that reflects real-world recommendation data.

This gives us the following hierarchical model:

$$\begin{aligned}
\mathbf{z} &\sim \text{Gibbs-type}, \\
\mathbf{q} &\sim \text{Gibbs-type}, \\
\theta_{hk} \mid a, b &\sim \text{Gamma}(a, b), \\
y_{ui} \mid \Theta, \mathbf{z}, \mathbf{q} &\sim \text{Poisson}(\theta_{z_u q_i}).
\end{aligned}$$

and we can write the likelihood as:

$$p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}, \boldsymbol{\Theta}) = \prod_{u,i} \frac{\theta_{z_u q_i}^{y_{ui}} e^{-\theta_{z_u q_i}}}{y_{ui}!} \propto \prod_{h=1}^{H} \prod_{k=1}^{K} \theta_{hk}^{m_{hk}} e^{-n_h n_k \theta_{hk}}.$$

As in Section 2.4.4, we marginalize out $\boldsymbol{\Theta}$. Exploiting the properties of the Gamma-Poisson model, we can write the conditional for the entire matrix, in terms of the clustering structure, as[1]:

$$p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}) = \left(\frac{b^a}{\Gamma(a)}\right)^{HK} \prod_{h=1}^{H} \prod_{k=1}^{K} \left[\frac{1}{\prod_{(u,i):z_u=h,q_i=k} y_{ui}!} \cdot \frac{\Gamma(m_{hk}+a)}{(n_h n_k + b)^{m_{hk}+a}}\right].$$

3.3.1 Posterior computation

As in the ESBM, we use a Gibbs sampling algorithm, which relies on the full conditional distributions for the node assignments. Note that we now have two distinct expressions: one for users and one for items:

$$\mathbb{P}(z_u = h \mid \mathbf{Y}, \mathbf{X}, \mathbf{z}_{-u}, \mathbf{q}) \propto \mathbb{P}(z_u = h \mid \mathbf{X}, \mathbf{z}_{-u}) \cdot \frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})},$$
$$\mathbb{P}(q_i = k \mid \mathbf{Y}, \mathbf{X}, \mathbf{q}_{-i}, \mathbf{z}) \propto \mathbb{P}(q_i = k \mid \mathbf{X}, \mathbf{q}_{-i}) \cdot \frac{p(\mathbf{Y} \mid q_i = k, \mathbf{q}_{-i}, \mathbf{z})}{p(\mathbf{Y}_{-i} \mid \mathbf{q}_{-i}, \mathbf{z})}.$$

The first term can be found from the urn scheme of the prior as in Chapter 2. The second term can be computed from the conditional $p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q})$ as[2]:

$$\frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})} \propto \prod_{k=1}^{K} \left[\frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)}{\Gamma(m_{hk}^{-u} + a)} \cdot \frac{(b + (n_h - 1)n_k)^{m_{hk}^{-u}+a}}{(b + n_h n_k)^{m_{hk}^{-u}+y_{uk}+a}}\right],$$
$$\frac{p(\mathbf{Y} \mid q_i = k, \mathbf{q}_{-i}, \mathbf{z})}{p(\mathbf{Y}_{-i} \mid \mathbf{q}_{-i}, \mathbf{z})} \propto \prod_{h=1}^{H} \left[\frac{\Gamma(m_{hk}^{-i} + y_{hi} + a)}{\Gamma(m_{hk}^{-i} + a)} \cdot \frac{(b + (n_k - 1)n_h)^{m_{hk}^{-i}+a}}{(b + n_k n_h)^{m_{hk}^{-i}+y_{hi}+a}}\right],$$

where

$$y_{uk} = \sum_{i:q_i=k} y_{ui}, \quad y_{hi} = \sum_{u:z_u=h} y_{ui}, \quad m_{hk} = \sum_{u,i} y_{ui} \mathbb{1}(z_u = h, q_i = k).$$

[1] Derivation in Appendix 5

[2] Derivation in Appendix 5

3.3.2 Rating prediction

After estimating the posterior distribution of the model by running the Gibbs sampler, we may want to predict the rating for a pair (u, i) such that user u has not consumed item i. The first step is to estimate the cluster assignments, which can be done using the method specified in Section 2.4.5.

To find an estimate for Θ, we first note that the conditional distribution, given the estimated cluster assignments, is:

$$\theta_{hk} \mid \mathbf{Y}, \mathbf{z}, \mathbf{q} \sim \text{Gamma}(a + m_{hk},\ b + n_h \cdot n_k)$$

We can then obtain a point estimate for the ratings using the posterior mean:

$$\hat{y}_{ui} = \mathbb{E}[y_{ui} \mid \mathbf{Y}, z_u = h, q_i = k, \Theta] = \mathbb{E}[\theta_{hk} \mid \mathbf{Y}, \mathbf{z}, \mathbf{q}] = \frac{a + m_{hk}}{b + n_h \cdot n_k} \tag{3.2}$$

3.3.3 Generating recommendations

After having estimated the community structure of the network we may want to generate recommendations for a user u. Suppose we want to recommend n items to u, a natural way to rank the items would be to sort them by θ_{z_u,q_i}. The issue is that, by properties of SBM, each item such that $q_i = k$ would have the same value of θ_{z_u,q_i} and the size of block k may be more than n.

The simplest way to overcome this problem is finding the community k (in the items) with highest $\theta_{z_u,k}$ and take a random sample of the items in that community. We note however that there are many other heuristics that could be used at this step. For instance we could select, out of the items in group k, the n most popular ones. We may also select other criteria aimed at balancing different needs of the recommender systems such as to improve diversity or freshness in the recommendations.

3.4 Degree–Corrected ESBM

As pointed out in Section 2.3.3, a limitation of the SBM is that, by construction, nodes in the same group are assumed to have the same degree. When dealing with recommender systems, this can be problematic because, intuitively, users with similar preferences may have very different levels of "activity", and items in the same group may vary in terms of "popularity".

One way to address this is to use a *Degree Corrected SBM*. In particular, we consider a non-parametric formulation, similar to the *Infinite Degree Corrected Stochastic Block Model* (IDCSBM) (Herlau et al., 2014), leveraging the ESBM framework.

As in Section 3.3, we use a Poisson likelihood and two community assignment vectors, one for the users and one for the items. Following the formulation of the IDCSBM, we introduce a Dirichlet-distributed vector ϕ for every cluster, and a degree correction parameter η for each user and each item. The ratings are assumed to be drawn from a Poisson distribution with a rate computed as the cluster parameter θ_{z_u,q_i}, multiplied by the user-specific and item-specific parameters.

We can write down the model in hierarchical form as:

$$\begin{gathered}
\mathbf{z} \sim \text{Gibbs-type} \quad \mathbf{q} \sim \text{Gibbs-type} \\
\phi_h \mid \gamma, \mathbf{z} \sim \text{Dirichlet}(\gamma \mathbf{1}_{n_h}) \quad \phi_k \mid \gamma, \mathbf{q} \sim \text{Dirichlet}(\gamma \mathbf{1}_{n_k}) \\
\eta_u = n_{z_u} \phi_{z_u,u} \quad \eta_i = n_{q_i} \phi_{q_i,i} \\
\theta_{hk} \mid a, b \sim \text{Gamma}(a, b) \\
y_{ui} \mid \Theta, \mathbf{z}, \mathbf{q}, \eta_u, \eta_i \sim \text{Poisson}(\eta_i \theta_{z_u q_i} \eta_u)
\end{gathered}$$

As pointed out by Herlau et al. (2014) the parameter γ plays an important role. Indeed if $\gamma \to \infty$ we will have $\phi_{h,u} \to 1/n_h$ that is the model becomes independent of the degree and very close to the ESBM. On the other hand if $\gamma \to 0$ we will have that within each group a single node will have $\phi_{h,u} \approx 1$ and the network becomes entirely dominated by a few nodes.

Note that, by construction, we have $\sum_u \eta_u \mathbb{1}(z_u = h) = n_h$ for all clusters. With this observation we can write the likelihood for the model as[3]:

$$p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}, \eta, \Theta) \propto \prod_{h,k} \theta_{hk}^{m_{hk}} \exp(-\theta_{hk} n_h n_k) \prod_{u:z_u=h} \eta_u^{d_u} \prod_{i:q_i=k} \eta_i^{d_i}$$

where $d_u = \sum_i y_{ui}$ that is the degree of node u.

As we have done in section 3.3 we obtain the conditional of the matrix given the cluster assignment vectors by integrating out block parameters $\theta_{h,k}$ as well as the relative degree proportions ϕ_h and ϕ_k. The resulting expression is[4]:

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}) \propto \prod_{h,k} \frac{\Gamma(m_{hk} + a)}{(n_h^2 n_k^2 + b)^{\theta_{m_{hk}}+a}} \prod_h \frac{B(\gamma \mathbf{1}_{n_h} + (d_u)_{u:z_u=h}))}{B(\gamma \mathbf{1}_{n_h})} n_h^{\sum_{u:z_u=h} d_u} \\ \prod_k \frac{B(\gamma \mathbf{1}_{n_k} + (d_i)_{i:q_i=k}))}{B(\gamma \mathbf{1}_{n_k})} n_k^{\sum_{i:q_i=k} d_i} \end{aligned} \tag{3.3}$$

where $B(\gamma \mathbf{1}_n) = \frac{\prod_{i=1}^n \Gamma(\gamma)}{\Gamma(\sum_{i=1}^n \gamma)}$

3.4.1 Posterior computation

The Gibbs sampling algorithm has similar form as the one in for the plain ESBM with a modification to the full conditional term. In particular, we get that the probability of user u being allocated to group h is proportional to[5]:

$$\begin{aligned} \mathbb{P}(z_u = h \mid \mathbf{Y}, \mathbf{X}, \mathbf{z}_{-u}, \mathbf{q}) \propto & \mathbb{P}(z_u = h \mid \mathbf{X}, \mathbf{z}_{-u}, \mathbf{q}) \cdot \frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})}, \propto \\ & \propto \mathbb{P}(z_u = h \mid \mathbf{X}, \mathbf{z}_{-u}, \mathbf{q}) \cdot \frac{\Gamma((n_h - 1)\gamma + d_h^{-u})}{\Gamma((n_h + 1)\gamma + d_h)} \frac{\Gamma(n_h \gamma)}{\Gamma((n_h - 1)\gamma)} \cdot \\ & \frac{n_h^{d_h}}{(n_h - 1)^{d_h^{-u}}} \prod_{k=1}^K \frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)}{\Gamma(m_{hk}^{-u} + a)} \cdot \frac{(b + (n_h - 1)n_k)^{m_{hk}^{-u}+a}}{(b + n_h n_k)^{m_{hk}^{-u}+y_{uk}+a}} \end{aligned}$$

and a similarly for the items replacing the appropriate quantities[6].

[3]Derivation in Appendix 5

[4]Derivation in Appendix 5

[5]Derivation in Appendix 5

[6]See Appendix 5

3.4.2 Rating prediction

As in the previous section we estimate the cluster assignment using the method specified in Section 2.4.5. The next step is finding an estimate for the block parameters $\theta_{h,k}$ and the relative degree vectors ϕ_h, ϕ_k. Exploiting conjugacy of both the Gamma and Dirichlet distribution in this model we find:

$$\phi_h \mid \mathbf{Y}, \mathbf{z}, \gamma \sim \text{Dirichlet}(\gamma \mathbf{1}_{n_h} + (d_u)_{u:z_u=h})$$
$$\phi_k \mid \mathbf{Y}, \mathbf{q}, \gamma \sim \text{Dirichlet}(\gamma \mathbf{1}_{n_k} + (d_i)_{i:q_i=k})$$
$$\theta_{hk} \mid \mathbf{Y}, \mathbf{z}, \mathbf{q}, a, b \sim \text{Gamma}(a + m_{hk}, b + n_h \cdot n_k)$$

We can then obtain a point estimate for the ratings using the posterior mean to get:

$$\hat{y}_{ui} = \mathbb{E}[y_{ui} \mid \mathbf{Y}, z_u = h, q_i = k, \Theta, \phi_h, \phi_k] = \frac{a + m_{hk}}{b + n_h \cdot n_k} n_h \hat{\phi}_{h,u} n_k \hat{\phi}_{k,i} \tag{3.4}$$

for $\hat{\phi}_{h,u} = \frac{\gamma + d_u}{\sum_{u:z_u=h} \gamma + d_u}$ and $\hat{\phi}_{k,i} = \frac{\gamma + d_i}{\sum_{i:q_i=k} \gamma_i + d_i}$.

From this formula the role of the parameter γ becomes clear, in the limit as $\gamma \to \infty$ the degree-correction parameter dominates the numerator and we get $\hat{\phi}_{h,u} \to 1/n_h$ which makes the score in equation 3.4 equal to that of the plain ESBM (equation 3.2). We can visualise this behaviour in Figure 3.1. The plot shows the variability in the values of $\hat{\phi}_{h,u}$ within a single cluster as the value of γ increases. In particular this example uses the first users' cluster for the dataset presented in 4.2 and labelled *DC network*. We can easily see that, as the degree correction parameter increases, the values converge to $1/n_h$ (in this case ≈ 0.007).

3.4.3 Generating recommendations

Suppose we want to recommend n of items to u, unlike in Section 3.3.3 we have a way to sort also items belonging to the same community. Indeed thanks to the node correction factors η_u and η_i we can sort the items according to:

$$\mathbb{E}[y_{ui} \mid \mathbf{Y}, \eta_i, \eta_u, \mathbf{q}, \mathbf{z}, \Theta] = \eta_i \eta_u \theta_{z_u q_i}$$

and select the n with the highest score.

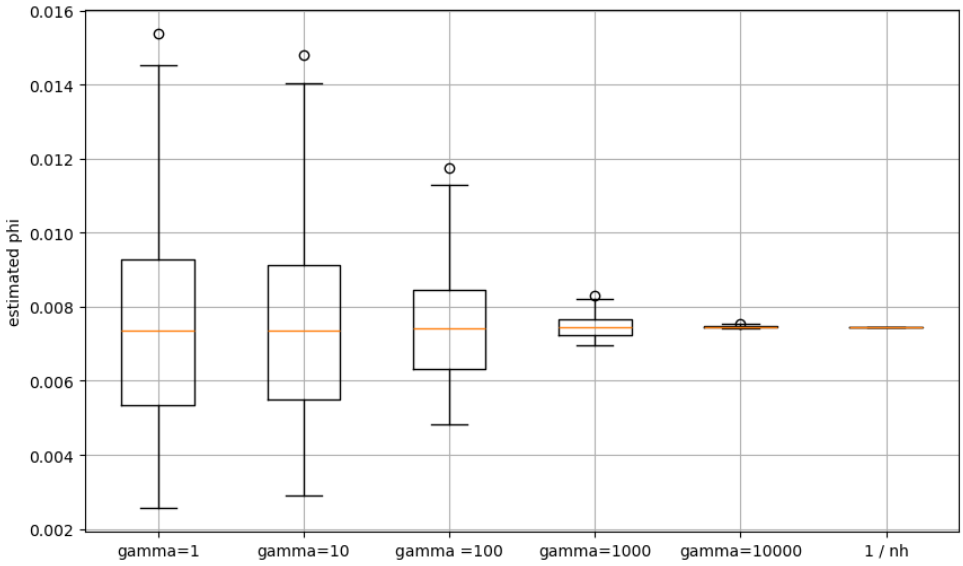

Figure 3.1: Boxplot of the different values of ϕ for a single cluster as the value of γ increases

3.5 Other methods

In the literature there are a number of different models to generate recommendations and a number of different approaches. In this section we will present some of these methods which we will use as benchmark to compare the performance of the ESBM and degree-corrected ESBM presented in Sections 3.3 and 3.4.

As mentioned in section 3.1 a popular class is that of *Matrix factorization* methods. When generating recommendations a widely used method is that of Non-Negative Matrix Factorization (NMF) (Paatero and Tapper, 1994; Lee and Seung, 1999) where we decompose the user-item interaction matrix into two lower dimensional matrices $\mathbf{Q}$, $\mathbf{P}$ and assume all their entries are non-negative. There are a number of different ways to fit this model, we follow the approach presented in Luo et al. (2014) and implemented in the Python package `surprise` (Hug, 2020). Here the loss is defined as in equation 3.1, with an additional non-negativity constraint, and the optimization problem is solved using a computationally efficient variation of the Stochastic Gradient Descent method.

An alternative formulation is that of *Poisson Matrix Factorization*. Indeed it can be shown that the loss function in NMF is equivalent to a factorised Poisson likelihood (Cemgil, 2009). A model that builds on this connection is that of *Hierarchical Poisson Factorization* (HPF) presented in Gopalan et al. (2015). In this framework the ratings y_{ui} are assumed to be Poisson-distributed with mean equal to the inner product of a user-specific vector $\theta_u \in \mathbb{R}^H$ and an item-specific vector $\theta_i \in \mathbb{R}^H$ where H is a model hyperparameter representing the latent dimensionality. The component of the latent vectors θ_u and θ_i are assumed to be independent and drawn from Gamma distributions. To introduce additional flexibility and capture heterogeneity across users and items, the authors place Gamma-distributed hyperpriors on the rate parameter of these Gamma distributions. These hyperparameters, denoted η_u for each user and η_i for each item, can be interpreted as user-specific "activity" and item-specific "popularity" (similar to the degree-correction terms introduced in Section 2.3.3).

We can write the model as:

$$
\begin{aligned}
\eta_u \mid a', b' &\sim \text{Gamma}(a', a'/c') \\
\eta_i, \mid c', d' &\sim \text{Gamma}(b', b'/d') \\
\theta_{uh} \mid a, \eta_u &\sim \text{Gamma}(a, \eta_u) \\
\theta_{ih} \mid b, \eta_i &\sim \text{Gamma}(b, \eta_i) \\
y_{ui} \mid \theta_u, \theta_i &\sim \text{Poisson}(\theta_u^T \theta_i)
\end{aligned}
$$

The posterior is approximated using Variational Inference (Blei et al., 2017) and recommendations are generated by scoring unconsumed items according to:

$$
\text{score}_{ui} = \mathbb{E}\left[\theta_u^T \theta_i \mid \mathbf{Y}\right]
$$

A ready-to-use implementation of this model can be found in the Python package `LensKit` (Ekstrand, 2020).

An alternative to matrix factorization is using memory-based methods. These are a class of algorithms that use the entire user-item data to compare the similarity between users or items. The main idea is to identify a suitable notion of similarity between users

or items, find a neighbourhood and use an algorithm to aggregate the information of the neighbours to generate predictions. From this high-level description it is clear that there are many possible choices for this algorithm such as whether to compute the similarity on users or items, the choice of similarity metric or the algorithms to aggregate the neighbours. Popular choices for the similarity function are the *cosine similarity* and *Pearson correlation* (for users u and v):

$$\text{cosine}(u, u') = \frac{\sum_i y_{ui} \cdot y_{u'i}}{\sqrt{\sum_i y_{ui}^2} \cdot \sqrt{\sum_i y_{u'i}^2}} \quad \text{pearson}(u, u') = \frac{\sum_i (y_{ui} - \mu_u) \cdot (y_{u'i} - \mu_{u'})}{\sqrt{\sum_i (y_{ui} - \mu_u)^2} \sqrt{\sum_i (y_{u'i} - \mu_{u'})^2}}$$

with μ_u being the mean for user u. A simple choice to aggregate the information of the neighbours is to use the K-Nearest Neighbours algorithms which, for a user u, finds the K-nearest users to u according to some similarity metric and then computes prediction for the items as:

$$y_{ui} = \frac{\sum_{u' \in N_i^K(u)} \text{sim}(u, u') \cdot y_{u'i}}{\sum_{u' \in N_i^K(u)} \text{sim}(u, u')}$$

where $N_i^K(u)$ denotes the K-nearest users to u. A ready-to-use implementation of this model can be found in the Python package `surprise`.

More recently, a variety of approaches have leveraged Deep Learning techniques for recommendation tasks. One particularly notable model is *Neural Matrix Factorization* (NeuMF), proposed by He et al. (2017). NeuMF extends the classical Matrix Factorization framework by incorporating a deep neural network architecture. In traditional MF, predictions are made by computing the inner product of latent user and item vectors. NeuMF generalizes this approach: it first learns the latent representations through embedding layers and then combines them using both linear and non-linear interactions.

Figure 3.2 illustrates the overall architecture of the NeuMF model. The input consists of user–item pairs and their corresponding ratings. Each user and item is represented via one-hot encoding and passed through fully connected embedding layers, which project these sparse vectors into dense latent spaces. Let $\mathbf{P}$ and $\mathbf{Q}$ denote the embedding matrices for users and items, respectively. Then, the latent vectors used in Equation 3.1 are obtained by multiplying the one-hot vectors with $\mathbf{P}$ and $\mathbf{Q}$, respectively.

The output of the embedding layers is duplicated and processed along two separate paths:

1. *Multi-Layer Perceptron* (MLP) path: the user and item embeddings are concatenated and passed through several fully connected layers to capture non-linear interactions.
2. *Generalized Matrix Factorization* (GMF) path: the embeddings are combined element-wise (Hadamard product), and a weighted sum is applied:

$$\hat{y}_{ui} = a_{\text{out}}\left(\mathbf{h}^T(\mathbf{p}_u \odot \mathbf{q}_i)\right)$$

where a_{out} is the activation function applied at the output layer. This formulation generalizes the classical inner product; for instance, setting a_{out} to the identity and $\mathbf{h}$ to a vector of ones recovers the standard inner product.

The outputs from the MLP and GMF branches are then concatenated and passed to the final prediction layer. In is worth noting that, while NeuMF was originally developed for implicit feedback and trained with binary cross-entropy loss, it can be adapted for explicit feedback tasks by modifying the output activation function and replacing the loss function accordingly.

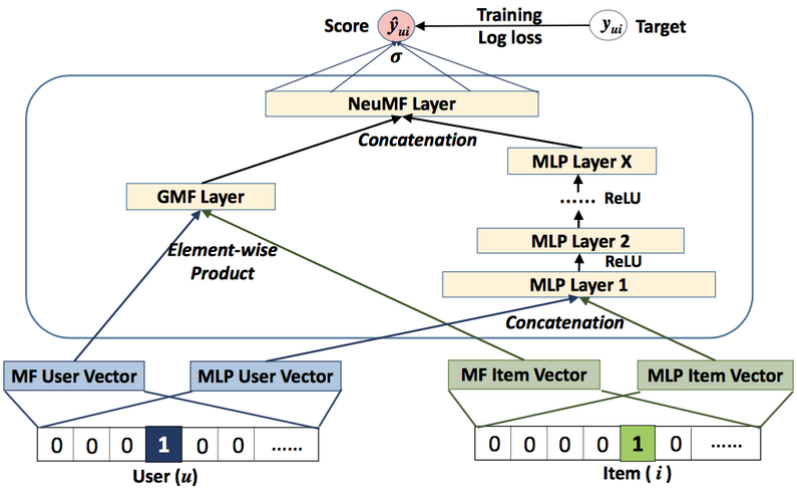

Figure 3.2: Architecture of the Neural Matrix Factorization (NeuMF) model. Source (He et al., 2017)

Chapter 4

Applications

4.1 Simulations

To assess model performance, we generate 10 synthetic datasets, each containing 200 users and 200 items, with $H = K = 4$ clusters[1]. These datasets are drawn from two different model specifications:

- the first 10 networks (hereafter referred to as the *ESBM networks*) are generated using the ESBM formulation, with θ_{hk} sampled from a Gamma$(1, 1)$. Note that, being generated from a classical SBM formulation, the nodes in each group tend to have, on average, the same degree (i.e., the same "activity" for users and "popularity" for items),
- the other 10 network (hereafter referred to as the *DC networks*) are generated from a degree-corrected ESBM, with θ_{hk} also sampled from a Gamma$(1, 1)$. As pointed out in Section 3.4, by setting a high value of γ, we can ensure that no nodes entirely dominate the network. In particular, we set $\gamma = 10$, and observe in Figure 4.1 that despite the adjacency matrix contains nodes with higher activity it still appearing fairly homogeneous.

[1]Code to replicate all simulations is available at https://github.com/lorenzo-costa/REC-ESBM

Alongside both networks, we generate a binary covariate that mirrors, with some noise, the clustering structure. Figure 4.1 illustrates an example dataset: the heatmap colours represent the rating values, and the sidebars indicate the covariate values.

Before running the Gibbs sampler, we remove 10% of the edges to be used as a held-out set for evaluating the models performance on recommendations.

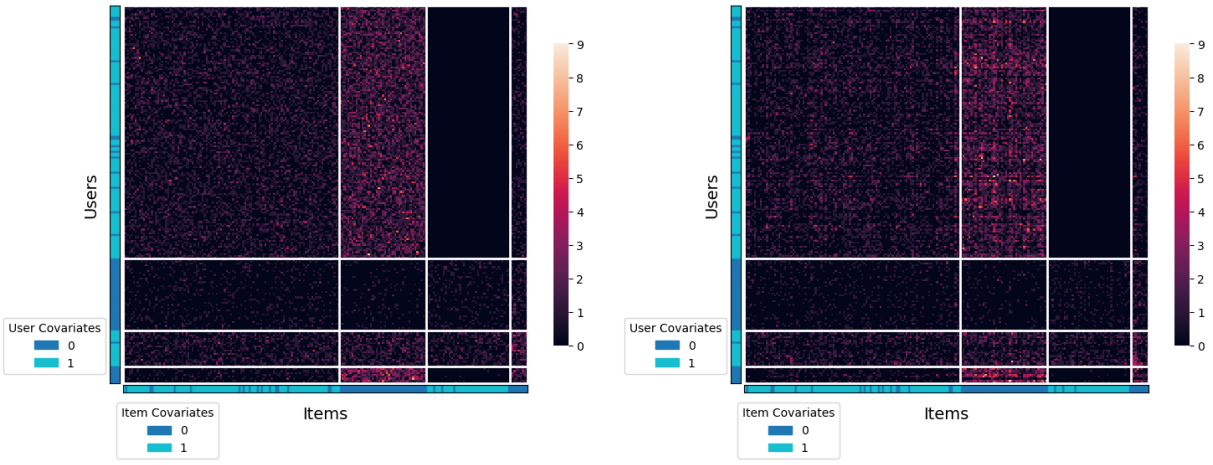

Figure 4.1: Heatmap of the synthetic networks generated from the ESBM (left) and degree-corrected ESBM (right)

We run the MCMC algorithm for 8,000 iterations, discarding the first 4,000 as burn-in. To evaluate the robustness of the models, we test them under different prior specifications: the Dirichlet, PitmanYor, and Gnedin processes. Each model is also fit both with and without covariates to assess the impact of covariate information on recovering the underlying community structure.

The models performance is assessed using the following metrics:

- clustering accuracy is measured by computing the VI distance between the estimated and true partitions,
- rating prediction accuracy is measured computing the Mean Absolute Error (MAE) and Mean Squared Error (MSE) between the point predictions generated by the model (as discussed in Sections 3.3.2 and 3.4.2) and the true values in the held-out set.

While MAE and MSE are useful for evaluating prediction accuracy, they are not sufficient for assessing the quality of recommendations. In recommender systems, the primary objective is to identify items a user is likely to be interested in, not just to minimize prediction error.
For this reason, we employ two metrics called *precision-at-k* ($P@k$) and *recall-at-k* ($R@k$) which are specifically tailored for recommender systems. These are constructed by having the model recommend k items to a user and then computing:

$$P@k(u) = \frac{\#\text{ of recommended items relevant for } u}{k}$$
$$R@k(u) = \frac{\#\text{ of recommended items relevant for } u}{\#\text{ of all possible relevant items for } u}$$

As mentioned in Section 3.5 we compare the performance of ESBM and degree-corrected ESBM to those of three difference methods: a user-based KNN, Non-negative Matrix Factorization, Hierarchical Poisson Factorization and Neural Matrix Factorization. All these methods have a number of hyperparameters that were fine-tuned on the relative datasets using cross-validation.

4.1.1 DC network

By looking at the plot of the adjacency matrix (Figure 4.1) we expect, a priori, that the node degree will have an influence on the community structure. For this reason we set the parameter γ of the degree-corrected model to a relatively low value, allowing the degree-correction factors to play a significant role. Knowing in advance that the true number of community is $K = H = 4$ we set the hyperparameters such that the expected number of communities, a priori, is close to 10. This is done to test whether the model can overcome a partially incorrect parameter specification. In practice we set $\alpha = 2$ for DP, σ=0.1 and $\alpha = 1.3$ for PY, and $\gamma = 0.55$ for GN.

The results for the simulation are shown in Table 4.1. It is evident by considering the VI distance between the true and estimated partitions that the degree-corrected model recovers the true community structure more effectively than the ESBM, particularly when covariates are included.

Results on DC network								
DC	**Cov**	**Prior**	**VI users**	**VI items**	**MAE**	**MSE**	**P@10**	**R@10**
No	No	DP	1.145	1.225	**0.657**	**1.062**	0.178	0.618
		PY	**1.142**	1.225	**0.657**	1.063	0.179	0.625
		GN	1.148	**1.219**	**0.657**	1.066	**0.181**	**0.637**
Yes	No	DP	0.172	**0.216**	0.679	1.178	0.201	**0.661**
		PY	0.182	0.267	0.676	1.127	0.201	0.660
		GN	**0.106**	0.263	**0.672**	**1.101**	**0.202**	0.659
No	Yes	DP	**1.025**	**1.153**	**0.661**	**1.078**	0.179	0.634
		PY	1.037	1.161	**0.661**	1.080	**0.180**	**0.644**
		GN	1.516	1.222	**0.661**	**1.078**	0.179	0.628
Yes	Yes	DP	0.101	0.252	**0.673**	1.123	0.202	**0.666**
		PY	0.114	0.213	0.671	1.119	0.202	0.665
		GN	**0.070**	**0.195**	0.667	**1.079**	**0.203**	0.665

Table 4.1: Performance comparison of different priors averaged over the 10 DC networks, indicating use of covariates (Cov) and degree correction (DC)

It is worth noting that the high value of VI for the ESBM model is due to its tendency to cluster nodes based on degree rather than community structure. Figure 4.2 display the estimated community partition using an ESBM model with DP prior. Comparing this with the true partition (left panel in Figure 4.1) we can see that it identifies the communities but tends to split them based on the degree of the nodes. This explanation is strengthen by the fact that the point predictions and performance on recommendations are similar between the two model specifications.

Another interesting insight is that the scores for point predictions and recommendation quality are similar when using the three different priors and the best prior varies between model specifications. This suggests that the model is relatively insensitive to the choice of the prior type and prior hyperparameters when applied to moderately large datasets (e.g., 200×200).

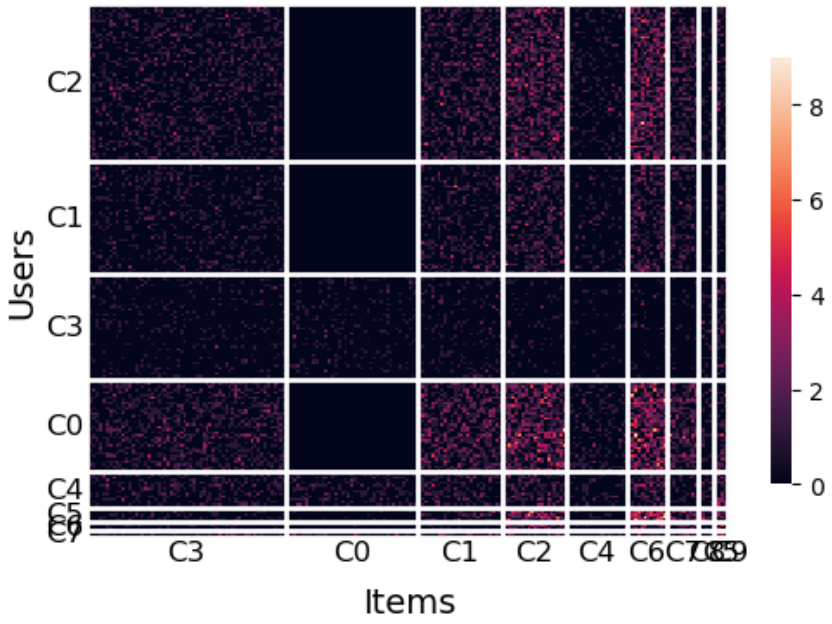

Figure 4.2: Heatmap reordered following the clustering partition estimated by plain ESBM with DP prior on DC network

Table 4.2 displays the performance of competing methods on this dataset. Among the three models, Hierarchical Poisson Factorization (HPF) performs best, achieving the lowest MAE and MSE, as well as the highest $P@10$ and $R@10$. Neural Matrix Factorization ranks second. Notably, the main difference in recommendation performance lies in the lower recall. Since its precision is close to that of HPF, this may indicate that the model tends to recommend often the most popular items (which are relevant to many users) but fails to make recommendations specifically tailored to individual user preferences.

Overall, these results demonstrate that both ESBM and its degree-corrected variant outperform the other popular methods considered on this dataset. This is both in terms of point prediction accuracy, as reflected by lower MAE and MSE, but also in terms of recommendation relevance, as indicated by higher $P@10$ and $R@10$. This suggests that the proposed model can be more effective at capturing the underlying structure of user-item interactions, especially when relevant covariates are included.

	MAE	MSE	P@10	R@10
KNN	0.989	1.778	0.159	0.103
NMF	0.967	1.776	0.171	0.114
HPF	**0.741**	**1.521**	**0.174**	**0.555**
NeuMF	0.753	1.620	0.172	0.243

Table 4.2: Performance of competing methods on DC network, averaged over 10 datasets

	MAE	MSE	P@10	R@10
KNN	0.961	1.590	0.122	0.073
NMF	0.955	1.524	0.119	0.080
HPF	**0.749**	**1.362**	**0.167**	**0.480**
NeuMF	0.751	1.481	0.157	0.342

Table 4.3: Performance of competing methods on ESBM network, averaged over 10 dataset

4.1.2 ESBM network

An advantage of the degree-corrected model is that through the parameter γ we can make the model arbitrarily close to a standard ESBM. Indeed looking at the heatmap in Figure 4.1 we expect the node to have fairly homogeneous degree and therefore we set γ such that the degree correction effect is small. The remaining hyperparameter are set as in Section 4.1.1 that is $\alpha = 2$ for DP, σ=0.1 and $\alpha = 1.3$ for PY, and $\gamma = 0.55$ for GN.

Table 4.4 shows the performance of the methods. Unsurprisingly ESBM recovers the true community structure better than its degree-corrected version since the data are generated with homogeneous degree for nodes in the same cluster. As expected, setting a high value of γ (for these simulations $\gamma = 20$) makes the degree-corrected version able to learn the community structure well as shown by the low VI distance for both users and items. It is worth noting that, for the plain ESBM, the partition is easy enough to learn that including covariates does not yield massive performance improvements except a slightly higher R@10. On the contrary, for the degree-corrected version including covariates makes the model able to identify the community structure better and more consistently across the different prior specifications.

From Table 4.3 we see that, consistent with the findings in Section 4.1.1, Hierarchical Poisson Factorization emerges as the best performing among the competing methods, followed by Neural Matrix Factorization. However, both ESBM and its degree-corrected

variant achieve superior performance across all settings, outperforming the competing methods in terms of both recommendation relevance and point prediction accuracy, regardless of the inclusion of covariates or the choice of priors.

Results on ESBM network								
DC	**Cov**	**Prior**	**VI users**	**VI items**	**MAE**	**MSE**	**P@10**	**R@10**
No	No	DP	0.113	0.194	0.667	0.976	0.188	0.571
		PY	**0.066**	**0.141**	**0.665**	**0.965**	**0.189**	**0.572**
		GN	0.113	0.179	0.667	0.976	0.188	0.569
Yes	No	DP	0.356	0.450	**0.676**	**1.010**	**0.187**	0.553
		PY	**0.299**	**0.436**	0.677	1.012	**0.187**	0.554
		GN	0.320	0.438	0.678	1.020	0.186	**0.557**
No	Yes	DP	0.035	0.101	0.665	0.963	0.188	0.563
		PY	0.035	0.101	0.665	0.963	0.188	0.563
		GN	**0.025**	**0.092**	**0.665**	**0.963**	**0.188**	**0.563**
Yes	Yes	DP	**0.143**	**0.222**	0.672	**0.994**	0.187	0.563
		PY	**0.143**	**0.222**	0.672	**0.994**	0.187	0.563
		GN	0.149	0.240	**0.671**	0.993	**0.188**	**0.564**

Table 4.4: Performance comparison of different priors averaged over the 10 ESBM networks, indicating use of covariates (Cov) and degree correction (DC)

4.2 Application to book recommendations

In this section we apply the models proposed in Chapter 3 to some real-world data. The dataset we selected contains data scraped from the book reviews website *Goodreads* and was collected by Wan and McAuley (2018). *Goodreads* is a popular online platform where users can rate, review, and recommend books. The feedback shared by users contains both a text review and a numeric rating between 1 and 5.We focus on the numeric rating and therefore we are in the context of *explicit feedback* data where users provide information on how much they liked the items consumed.

The goal is to test our models in providing users recommendations on which book they should read next based on their previous review as well as the pattern of consumption of other (similar) users. Considering the nature of the data we believe it is reasonable to assume the presence of an underlying block structure. Users often exhibit similar preferences and behaviours, forming latent communities based on shared interests. At the same time, books naturally cluster by subgenre, theme, or readership. This structure provides a motivation for the use of Stochastic Block Models which naturally capture these block structures in a bipartite network. Employing an ESBM or its degree-corrected version provides a principled way to model the heterogeneity in user-book interactions and to exploit community structure for more accurate recommendations.

4.2.1 The data

The dataset is very extensive containing more than 100 million ratings for 1.5 million books and 876k users. To ease the computational burden and conduct more extensive experiments we select only the 1000 most active users and 1000 more popular books resulting in approximately 105k ratings. We can see a plot of these data in Figure 4.3.

Along with observations of the ratings the creators of the dataset provide also information about the genre of the books which we will use as covariates. Since some of these genres are barely represented in our dataset (less than 5 items) we keep side information only for the top 5 genres: biography, fiction, historical, classics and romance. Each book can belong to more than one genre hence we pass these information to the model as a series of binary variables. For example the book *To Kill a Mockingbird* by Harper Lee is tagged as both historical and a classic hence we would have that both the binary vector for "historical" and "classic" contain a 1.

Similarly to what we have done in Section 4.1 we remove 10% of the ratings and use them as validation set to test the model performance on recommendations.

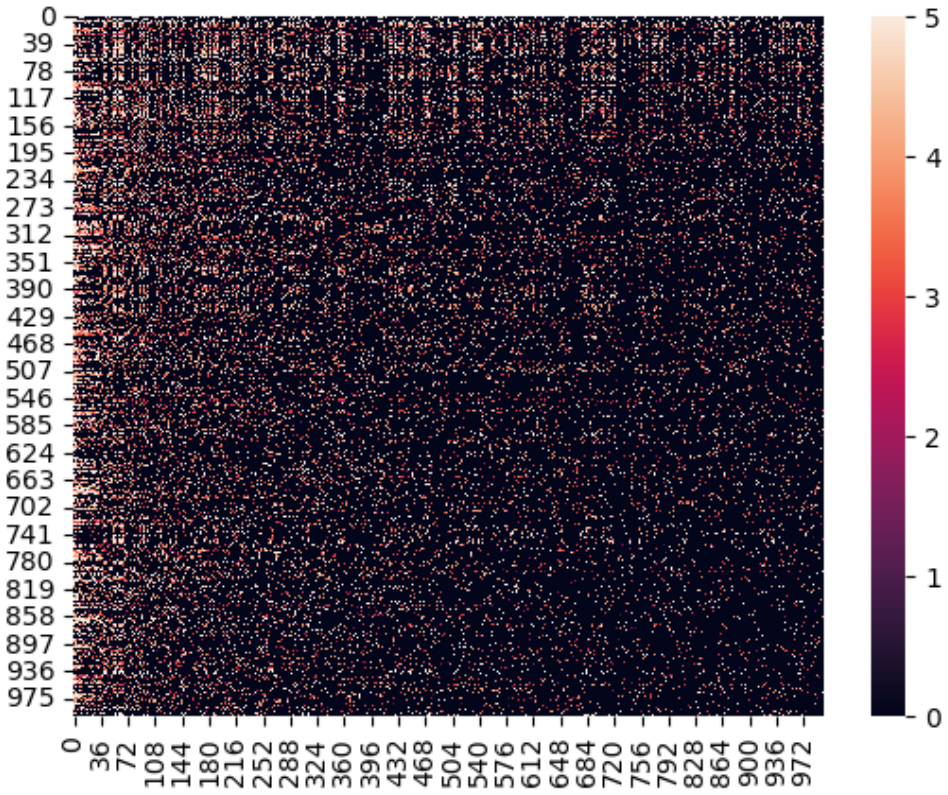

Figure 4.3: Heatmap of the *Goodreads* dataset

4.2.2 Model setup

As we can see from Figure 4.4 the distribution of the users and books degree are skewed toward the left and this effect is stronger for the items. This indicates that there are some items significantly more popular and users that are more active than others and/or tend to give higher ratings. This suggests that when using the degree-corrected model, it makes sense to set a relatively low value for γ to have a strong degree-correction factors.

In our experiments we take into account these observations and set the degree correction parameter to 20 for the users and 10 for the items. As in Section 4.2 we set the parameters of the gamma priors to $a = b = 1$ to encourage sparsity. The remaining hyperparameters are set to have an expected number of cluster, a priori, equal to 25 thus

we have $\alpha = 5$ for DP, $\sigma = 0.5$ and $\alpha = -0.2$ for PY and $\gamma = 0.55$ for GN. To estimate the posterior we run the Gibbs sampler for 1500 iterations discarding the first 500 as burn-in.

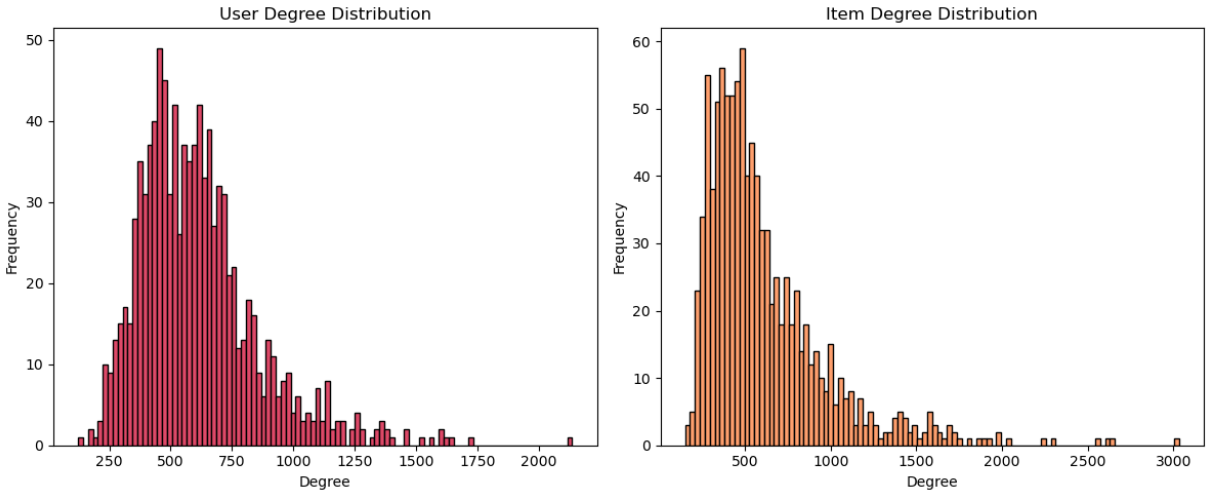

Figure 4.4: Degree distribution of users (left) and items (right) for the training set of the *Goodreads* network

We compare the ESBM and degree-corrected ESBM with the models described in Section 3.5 that is user-based KNN, Non-Negative Matrix Factorization, Hierarchical Poisson Factorization and Neural Matrix Factorization. All these methods require careful fine-tuning of the hyperparameters which we carry out using cross-validation.

4.2.3 Results

The models are evaluated both in terms of point prediction accuracy, using MAE and MSE , and their ability to generate relevant recommendations, assessed through P@10 and R@10. In particular, to compute P@10 and R@10, we use the held-out test set considering ratings above 3 as "relevant" for each user u. This threshold, while somewhat arbitrary, reflects the reasonable assumption that ratings below 3 indicate user dissatisfaction whereas ratings of 3 or above suggest that the user found the book satisfactory or enjoyable.

Performance of ESBM methods on the *Goodreads* dataset						
DC	**Cov**	**Prior**	**MAE**	**MSE**	**P@10**	**R@10**
No	No	DP	0.723	1.653	0.220	0.072
		PY	0.724	1.658	0.205	0.069
		GN	**0.693**	**1.581**	**0.243**	**0.075**
Yes	No	DP	0.702	1.610	0.246	0.084
		PY	0.698	1.589	0.253	**0.085**
		GN	**0.677**	**1.544**	**0.307**	0.051
No	Yes	DP	0.740	1.706	0.221	0.075
		PY	0.755	1.745	0.197	0.069
		GN	**0.702**	**1.601**	**0.250**	**0.082**
Yes	Yes	DP	0.705	**1.602**	0.356	0.122
		PY	**0.704**	1.608	0.359	**0.124**
		GN	0.706	1.610	**0.360**	**0.124**

Table 4.5: Performance of ESBM methods with different priors on the *Goodreads* dataset, indicating use of covariates (Cov) and degree correction (DC)

Table 4.5 summarizes the performance, computed on the held-out set, of the plain ESBM and degree-corrected ESBM models, while Table 4.6 presents results for the methods described in Section 3.5. From the reported metrics, it appears that ESBM-based models outperform competitors in terms of MAE and MSE, indicating superior point prediction capabilities. Notably, the degree-corrected variants (especially with a GN prior) achieve the best overall scores in both error metrics.

When it comes to recommendation quality, the SBM-based models demonstrate competitive performance. The degree-corrected ESBM with a GN prior attains $P@10$ and $R@10$ values that are comparable to those of HPF and even exceed them when covariates are included. This supports the conclusion that ESBM is not only accurate for point predictions but also effective for recommendation tasks. Furthermore, ESBM models provide advantages in terms of interpretability and transparency, which are typically lacking in more complex approaches such as HPF and NeuMF.

Interestingly, including covariates degrades accuracy in point predictions but improves recommendation quality. A possible explanation for this is that, while covariates provide a strong signal for a user's relative preferences, they also introduce additional complexity which increases the chance of the Markov chain getting stuck in a local optima. Point prediction metrics like MAE and MSE may be more sensitive to this issue hurting its ability to predict exact rating values. On the other hand, the task of recommendation is about relative ordering, which is more robust to small prediction errors. The model is then able to exploit better the signal from the book genre and learn which item a user will prefer even if the individual score predictions are less precise

These results suggest that community-based models like ESBM offer a promising balance between performance and explainability, making them well-suited for applications where we care about interpretability.

	MAE	**MSE**	**P@10**	**R@10**
KNN	1.304	2.345	0.187	0.009
NMF	1.223	1.877	0.043	0.003
HPF	**0.743**	**1.683**	**0.320**	**0.108**
NeuMF	0.753	1.826	0.189	0.048

Table 4.6: Performance of competing methods on *Goodreads* network

Chapter 5

Conclusion

In this thesis, we investigated the use of Stochastic Block Models (SBMs), and more specifically their non-parametric formulation in the framework of Extended Stochastic Block Models (ESBMs), as a tool for building recommender systems. By framing the recommendation task as a problem of uncovering latent structure in bipartite user-item networks, we highlighted how community detection techniques can offer interpretable approaches to personalized recommendations. To further improve the flexibility of ESBMs, we proposed a degree-corrected extension designed to better handle degree heterogeneity among users and items, a feature commonly observed in real-world data.

Through empirical applications, we showed that these models are capable of capturing meaningful patterns in user-item interactions and that they can provide competitive performance when compared to popular methods. Moreover, their generative grounding in network theory offers attractive interpretability benefits, often lacking in black-box models.

Despite their advantages, ESBM-based models suffer from a major drawback: limited computational efficiency. The Gibbs sampling algorithms they rely on are computationally intensive and do not scale well to large datasets. As a result, training these models on real-world data can be time-consuming, which limits their usefulness in scenarios

involving many users or a large number of items.

In light of this finding, one natural direction for future research is to address the computational bottlenecks associated with the inference of ESBMs and their degree-corrected variants. In particular, exploring faster and more scalable alternatives to Gibbs sampling could significantly expand the usability of these models in practical recommender system. Promising directions include approximate inference techniques such as variational inference, which can offer substantial speed-ups by turning the inference problem into an optimization task rather than relying on sampling-based methods.

In conclusion, while non-parametric Stochastic Block Models show substantial promise as interpretable and theoretically grounded tools for recommendation, making them practically viable at scale will require further advances in modelling, inference and computational strategies.

Bibliography

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Aldous, D. J. (1985). Exchangeability and related topics. In Hennequin, P. L., editor, *École d'Été de Probabilités de Saint-Flour XIII*, pages 1–198. Springer Berlin Heidelberg.

Blei, D. M., Kucukelbir, A., and and, J. D. M. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Cemgil, A. T. (2009). Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009:785152.

De Blasi, P., Favaro, S., Lijoi, A., Mena, R. H., Prünster, I., and Ruggiero, M. (2015). Are gibbs-type priors the most natural generalization of the dirichlet process? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):212–229.

Ekstrand, M. D. (2020). Lenskit for python: Next-generation software for recommender systems experiments. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pages 2999–3006. Association for Computing Machinery.

Etemadi, M., Bazzaz Abkenar, S., Ahmadzadeh, A., Haghi Kashani, M., Asghari, P., Akbari, M., and Mahdipour, E. (2023). A systematic review of healthcare recommender systems: Open issues, challenges, and techniques. *Expert Systems with Applications*, 213:118823.

Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209 – 230.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Geng, J., Bhattacharya, A., and and, D. P. (2019). Probabilistic community detection with unknown number of communities. *Journal of the American Statistical Association*, 114(526):893–905.

Ghidini, V., Legramanti, S., and Argiento, R. (2023). Extended stochastic block model with spatial covariates for weighted brain networks. In Avalos-Pacheco, A., De Vito, R., and Maire, F., editors, *Bayesian Statistics, New Generations New Approaches*, pages 47–56. Springer International Publishing.

Gnedin, A. (2010). A Species Sampling Model with Finitely Many Types. *Electronic Communications in Probability*, 15(none):79 – 88.

Gopalan, P., Hofman, J. M., and Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 326–335. AUAI Press.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871.

Hartigan, J. A. (1990). Partition models. *Communications in Statistics - Theory and Methods*, 19(8):2745–2756.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182. International World Wide Web Conferences Steering Committee.

Herlau, T., Schmidt, M. N., and Mørup, M. (2014). Infinite-degree-corrected stochastic block model. *Phys. Rev. E*, 90:032819.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272.

Hug, N. (2020). Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174.

Ishwaran, H. and and, L. F. J. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173.

Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., and Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 381–388. AAAI Press.

Khanal, S. S., Prasad, P., Alsadoon, A., and Maag, A. (2020). A systematic review: machine learning based recommendation systems for e-learning. *Education and Information Technologies*, 25(4):2635–2664.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Lee, C. and Battiston, M. (2023). A bayesian nonparametric stochastic block model for directed acyclic graphs. *arXiv preprint arXiv:2301.07513*.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Legramanti, S., Rigon, T., Durante, D., and Dunson, D. B. (2022). Extended stochastic block models with application to criminal networks. *The Annals of Applied Statistics*, 16(4):2369–2395.

Luo, X., Zhou, M., Xia, Y., and Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284.

Müller, P., Quintana, F., and Rosner, G. L. (2011). A product partition model with regression on covariates. *Journal of Computational and Graphical Statistics*, 20(1):260–278.

Nowicki, K. and and, T. A. B. S. (2001). Estimation and prediction for stochastic block-structures. *Journal of the American Statistical Association*, 96(455):1077–1087.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

Page, G. L. and Quintana, F. A. (2018). Calibrating covariate informed product partition models. *Statistics and Computing*, 28(5):1009–1031.

Perman, M., Pitman, J., and Yor, M. (1992). Size-biased sampling of poisson point processes and excursions. *Probability Theory and Related Fields*, 92(1):21–39.

Pitman, J. (2006). *Sequential constructions of random partitions*, pages 55–75. Springer Berlin Heidelberg.

Quintana, F. A., Müller, P., and Papoila, A. L. (2015). Cluster-specific variable selection for product partition models. *Scandinavian Journal of Statistics*, 42(4):1065–1077.

Raza, S., Rahman, M., Kamawal, S., Toroghi, A., Raval, A., Navah, F., and Kazemeini, A. (2025). A comprehensive review of recommender systems: Transitioning from theory to practice. *arXiv preprint arXiv.2407.13699*.

Salakhutdinov, R. and Mnih, A. (2007). Probabilistic matrix factorization. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'07, pages 1257–1264. Curran Associates Inc.

Schmidt, M. N. and Morup, M. (2013). Nonparametric bayesian modeling of complex networks: an introduction. *IEEE Signal Processing Magazine*, 30(3):110–128.

Snijders, T. A. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100.

Wade, S. and Ghahramani, Z. (2018). Bayesian Cluster Analysis: Point Estimation and Credible Balls (with Discussion). *Bayesian Analysis*, 13(2):559 – 626.

Wan, M. and McAuley, J. (2018). Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 86–94. Association for Computing Machinery.

Wasserman, S. and Anderson, C. (1987). Stochastic a posteriori blockmodels: construction and assessment. *Social Networks*, 9(1):1–36.

White, H. C., Boorman, S. A., and Breiger, R. L. (1976). Social structure from multiple networks: blockmodels of roles and positions. *American Journal of Sociology*, 81(4):730–780.

Zhang, Z.-Y., Gai, Y., Wang, Y.-F., Cheng, H.-M., and Liu, X. (2018). On equivalence of likelihood maximization of stochastic block model and constrained nonnegative matrix factorization. *Physica A: Statistical Mechanics and its Applications*, 503:687–697.

Appendix

Derivation of seating mechanism for Gibbs-type priors

We can derive this urn scheme by noting that

$$
\begin{aligned}
p(z_{V+1} = h \mid \mathbf{z}) =& \frac{p(\mathbf{z}, z_{V+1} = h)}{p(\mathbf{z})} = \\
&= \begin{cases} \frac{\mathcal{V}_{V+1,H} \prod_{l=1}^{H} (1-\sigma)_{n_l - 1}}{\mathcal{V}_{V,H} \prod_{l=1}^{H} (1-\sigma)_{n_l^- - 1}} & h = 1, \dots, H \\ \frac{\mathcal{V}_{V+1,H+1} \prod_{l=1}^{H+1} (1-\sigma)_{n_l - 1}}{\mathcal{V}_{V,H} \prod_{l=1}^{H} (1-\sigma)_{n_l^- - 1}} & h = H+1 \end{cases} = \\
&= \begin{cases} \frac{\mathcal{V}_{V+1,H}}{\mathcal{V}_{V,H}} (n_h - \sigma) & h = 1, \dots, H \\ \frac{\mathcal{V}_{V+1,H+1}}{\mathcal{V}_{V,H}} (1-\sigma)_{1-1} & h = H+1 \end{cases} = \\
&= \begin{cases} \frac{\mathcal{V}_{V+1,H}}{\mathcal{V}_{V,H}} (n_h - \sigma) & h = 1, \dots, H \\ \frac{\mathcal{V}_{V+1,H+1}}{\mathcal{V}_{V,H}} & h = H+1 \end{cases} =
\end{aligned}
$$

where the last equality follows from the definition of $(1-\sigma)_{n_h - 1 + 1} / (1-\sigma)_{n_h - 1} = \frac{(1-\sigma) \cdots (n_h - \sigma)}{(1-\sigma) \cdots (n_h - \sigma - 1)}$ and $(1-\sigma)_0 = 1$ by definition

Derivation Gibbs update for categorical covariates

We can derive this urn scheme by noting that Assuming that $X_u = c$ we can derive:

$$
\frac{p(\mathbf{X}_h)}{p(\mathbf{X}_{h,-u})} \propto \frac{n_{hc}^- + \alpha_c}{n_h^- + \alpha_0}
$$

Using the property of the gamma function $\Gamma(x) = (x-1)!$ when x is an integer:

$$\frac{p(\mathbf{X}_h)}{p(\mathbf{X}_{h,-u})} \propto \frac{\Gamma(n_h^- + \alpha_0)}{\Gamma(n_h^- + 1 + \alpha_0)} \frac{\Gamma(n_{h,c}^- + 1 + \alpha_{h,c})}{\Gamma(n_{h,c}^- + \alpha_{h,c})} \propto$$
$$\propto \frac{(n_h^- + \alpha_0 - 1)!\,(n_{h,c}^- + \alpha_{h,c})!}{(n_h^- + \alpha_0)!\,(n_{h,c}^- + \alpha_{h,c} - 1)!} \propto \frac{n_{hc}^- + \alpha_c}{n_h^- + \alpha_0}$$

Derivation of marginal for the adjacency matrix with Poisson likelihood in ESBM

The goal is writing the distribution for the adjacency matrix in terms of the clustering structure:

$$p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}) = \left(\frac{b^a}{\Gamma(a)}\right)^{HK} \prod_{h=1}^{H} \prod_{k=1}^{K} \left[\frac{1}{\prod_{(u,i):z_u=h,q_i=k} y_{ui}!} \frac{\Gamma(m_{hk}+a)}{(n_h n_k + b)^{m_{hk}+a}} \right]$$

We first write the distribution for each combination of two blocks h, k by exploiting Gamma-Poisson conjugacy:

$$p(\mathbf{Y}_{hk} \mid \mathbf{z}, \mathbf{q}) = \int_0^\infty \frac{b^a}{\Gamma(a)} \theta_{hk}^{a-1} e^{-b\theta_{hk}} \prod_{(u,i):z_u=h,q_i=k} \frac{\theta_{hk}^{y_{ui}} e^{-\theta_{hk}}}{y_{ui}!} =$$
$$= \frac{b^a}{\Gamma(a) \prod_{(u,i):z_u=h,q_i=k} y_{ui}!} \int_0^\infty \theta_{hk}^{a-1+m_{hk}} e^{-(n_h n_k + b)\theta_{hk}} =$$
$$= \frac{b^a}{\Gamma(a) \prod_{(u,i):z_u=h,q_i=k} y_{ui}!} \frac{\Gamma(m_{hk}+a)}{(n_h n_k + b)^{m_{hk}+a}}$$

then the marginal of the full matrix is simply the product of the blocks.

$$p(\mathbf{Y}_{hk} \mid \mathbf{z}, \mathbf{q}) = \prod_{h=1}^{H} \prod_{k=1}^{K} p(\mathbf{Y}_{hk} \mid \mathbf{z}, \mathbf{q}) =$$
$$= \left(\frac{b^a}{\Gamma(a)}\right)^{HK} \prod_{h=1}^{H} \prod_{k=1}^{K} \left[\frac{1}{\prod_{(u,i):z_u=h,q_i=k} y_{ui}!} \frac{\Gamma(m_{hk}+a)}{(n_h n_k + b)^{m_{hk}+a}} \right]$$

Derivation of Gibbs update for Poisson likelihood in ESBM

We want to derive the expression for:

$$\frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})}$$

Note that this can be thought of as the change in likelihood when adding user u with assignment $z_u = h$. The first thing we note is that everything that is not related to user u would cancel out, thus we need to consider only what happens to the block Y_{hk} of the matrix. Before adding user u the likelihood would be:

$$\begin{aligned} p(Y_{hk}^{-u} \mid \mathbf{z}_{-u}, \mathbf{q}) &= \frac{b^a}{\Gamma(a)} \frac{\Gamma(m_{hk} - \sum_{i:q_i=k} y_{ui} + a)}{(b + (n_h - 1)n_k)^{m_{hk} - \sum_{i:q_i=k} y_{ui} + a}} \frac{1}{\prod_{(u,i):z_u=h,q_i=k,u' \neq u} y_{u'i}!} \\ &= \frac{b^a}{\Gamma(a)} \frac{\Gamma(m_{hk}^{-u} + a)}{(b + (n_h - 1)n_k)^{m_{hk}^{-u} + a}} \frac{1}{\prod_{(u,i):z_u=h,q_i=k,u' \neq u} y_{u'i}!} \end{aligned}$$

If we denote $y_{u,k} = \sum_{i:q_i=k} y_{ui}$ we get:

$$p(Y_{hk} \mid \mathbf{z}, \mathbf{q}) = \frac{b^a}{\Gamma(a)} \frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)}{(b + n_h n_k)^{m_{hk}^{-u} + y_{uk} + a}} \frac{1}{\prod_{(u,i):z_u=h,q_i=k,u' \neq u} y_{u'i}! \prod_{i:q_i=k} y_{ui}!}$$

and the ratio becomes:

$$\begin{aligned} \frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})} &= \prod_{k=1}^{K} \frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)(b + (n_h - 1)n_k)^{m_{hk}^{-u} + a}}{\Gamma(m_{hk}^{-u} + a)(b + n_h n_k)^{m_{hk}^{-u} + y_{uk} + a}} \frac{\prod_{(u,i):z_u=h,q_i=k,u' \neq u} y_{u'i}!}{\prod_{(u,i):z_u=h,q_i=k,u' \neq u} y_{u'i}! \prod_{i:q_i=k} y_{ui}!} = \\ &= \prod_{k=1}^{K} \frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)}{\Gamma(m_{hk}^{-u} + a)} \cdot \frac{(b + (n_h - 1)n_k)^{m_{hk}^{-u} + a}}{(b + n_h n_k)^{m_{hk}^{-u} + y_{uk} + a}} \cdot \frac{1}{\prod_{i:q_i=k} y_{ui}!} \end{aligned}$$

The update for the user can be derived in the same way substituting the appropriate quantities

Derivation of marginal for the adjacency matrix with Poisson likelihood in ESBM

We first show how to write the conditional distribution of the adjacency matrix in the degree-corrected ESBM. First note that, since ϕ_i and ϕ_u are drawn from a Dirichlet

distribution their sum is equal to one. With this we have that $\sum_{u:z_u=h} \eta_u = n_h$ and $\sum_{i:q_i=k} \eta_i = n_k$.

Then calling d_i the degree of node i (i.e. $\sum_j y_{ij}$) and $m_{hk} = \sum_{i:z_i=h, j:z_j=k} y_{ui}$ we get:

$$
\begin{aligned}
p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}, \eta, \Theta) &\propto \prod_{ui} \exp(-\eta_u, \eta_i \theta_{z_u, q_i})(\eta_u, \eta_i \theta_{h,k})^{y_{ui}} \propto \\
&\propto \prod_{h,k} \theta_{h,k}^{m_{hk}} \exp\left(- \sum_{u:z_u=h, i:q_i=k} \theta_{h,k} \sum_{u:z_u=h} \eta_u \sum_{i:q_i=k} \eta_i \right) \prod_{u,i} \eta_u^{y_{ui}} \prod_{u,i} \eta_i^{y_{ui}} \propto \\
&\propto \prod_{h,k} \theta_{h,k}^{m_{hk}} \exp(-\theta_{h,k} n_h n_k) \prod_{u:z_u=h} \eta_u^{d_u} \prod_{i:q_i=k} \eta_i^{d_i}
\end{aligned}
$$

Then we want to integrate out the vector probabilities $\theta_{h,k}$ and degree proportions ϕ_h and ϕ_k. Looking at the likelihood it is clear that we can integrate them separately. For blocks h, k we get:

$$
\begin{aligned}
p(\mathbf{Y}_{hk} \mid \mathbf{z}, \mathbf{q}) &\propto \int \theta_{h,k}^{m_{h,k}} \exp(-\theta_{h,k} n_h n_k) \mathrm{d}p(\theta_{h,k}) \int \prod_{u:z_u=h} (n_h \phi_{h,u})^{d_u} \, \mathrm{d}p(\phi_h) \int \prod_{i:z_i=k} (n_k \phi_{k,i})^{d_i} \mathrm{d}p(\phi_k) \\
&\propto \int \theta_{h,k}^{m_{hk}} \exp(-\theta_{h,k} n_h n_k) \frac{b^a}{\Gamma(a)} \theta_{h,k}^{a-1} e^{-b\theta_{hk}} \, \mathrm{d}\theta_{h,k} \int \frac{\Gamma\left(\sum_{u:z_u=h} \gamma\right)}{\prod_{u:z_u=h} \Gamma(\gamma)} \prod_{u:z_u=h} (n_h \phi_{h,u})^{d_u} \phi_{h,u}^{\gamma-1} \, \mathrm{d}\phi_h \\
&\quad \int \frac{\Gamma\left(\sum_{i:q_i=k} \gamma\right)}{\prod_{i:q_i=k} \Gamma(\gamma)} \prod_{i:q_i=k} (n_k \phi_i)^{d_i} \phi_{k,i}^{\gamma-1} \, \mathrm{d}\phi_k \\
&\propto \int \theta_{h,k}^{m_{hk}+a-1} \exp(-\theta_{h,k}(n_h n_k + b)) \frac{b^a}{\Gamma(a)} \, \mathrm{d}\theta_{h,k} \cdot n_h^{\sum_{u:z_u=h} d_u} \frac{\Gamma(\sum_{u:z_u=h} \gamma)}{\prod_{u:z_u=h} \Gamma(\gamma)} \int \prod_{u:z_u=h} \phi_{h,u}^{d_u+\gamma-1} \, \mathrm{d}\phi_h \\
&\quad n_k^{\sum_{i:q_i=k} d_i} \frac{\Gamma\left(\sum_{i:q_i=k} \gamma\right)}{\prod_{i:q_i=k} \Gamma(\gamma)} \int \prod_{i:q_i=k} \phi_i^{d_i+\gamma-1} \, \mathrm{d}\phi_k \\
&\propto \frac{b^a}{\Gamma(a)} \frac{\Gamma(m_{hk}+a)}{(n_h n_k + b)^{\theta_{m_{hk}}+a}} \frac{B(\gamma \mathbf{1}_{n_h} + (d_u)_{u:z_u=h}))}{B(\gamma \mathbf{1}_{n_h})} \frac{B(\gamma \mathbf{1}_{n_k} + (d_i)_{u:q_i=k}))}{B(\gamma \mathbf{1}_{n_k})} n_h^{\sum_{u:z_u=h} d_u} n_k^{\sum_{i:q_i=k} d_i}
\end{aligned}
$$

The since the block are independent, we get that the distribution for the whole matrix is:

$$
\begin{aligned}
p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}) \propto & \frac{b^a}{\Gamma(a)} \prod_{h,k} \frac{\Gamma(m_{hk}+a)}{(n_h n_k + b)^{\theta_{m_{hk}}+a}} \prod_h \frac{B(\gamma \mathbf{1}_{n_h} + (d_u)_{u:z_u=h}))}{B(\gamma \mathbf{1}_{n_h})} n_h^{\sum_{u:z_u=h} d_u} \\
& \prod_k \frac{B(\gamma \mathbf{1}_{n_k} + (d_i)_{i:q_i=k}))}{B(\gamma \mathbf{1}_{n_k})} n_k^{\sum_{i:q_i=k} d_i}
\end{aligned}
$$

Derivation of Gibbs for Poisson likelihood in degree-corrected ESBM

To derive the updates for the Gibbs sampler we need to find:

$$\frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})}$$

Looking at the conditional in equation 3.3 we see that it can be split into three independent part:

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{z}, \mathbf{q}) \propto & \prod_{h,k} \frac{\Gamma(m_{hk} + a)}{(n_h n_k + b)^{\theta m_{hk} + a}} \prod_h \frac{B(\gamma \mathbf{1}_{n_h} + (d_u)_{u:z_u=h}))}{B(\gamma \mathbf{1}_{n_h})} n_h^{\sum_{u:z_u=h} d_u} \\ & \prod_k \frac{B(\gamma \mathbf{1}_{n_k} + (d_i)_{i:q_i=k}))}{B(\gamma \mathbf{1}_{n_k})} n_k^{\sum_{i:q_i=k} d_i} \\ \propto & g(\mathbf{Y}, \mathbf{z}, \mathbf{q}) f(\mathbf{Y}, \mathbf{z}, \mathbf{q}) \end{aligned}$$

The first piece $g(\mathbf{Y}, \mathbf{z}, \mathbf{q}) = \prod_{h,k} \frac{\Gamma(m_{hk}+a)}{(n_h n_k+b)^{\theta m_{hk}+a}}$ is the same as the plain ESBM and so we can get the update for this piece as:

$$\frac{g(\mathbf{Y}, z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{g(\mathbf{Y}_{-u}, \mathbf{z}_{-u}, \mathbf{q})} \propto \prod_{k=1}^{K} \frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)}{\Gamma(m_{hk}^{-u} + a)} \cdot \frac{(b + (n_h - 1)n_k)^{m_{hk}^{-u}+a}}{(b + n_h n_k)^{m_{hk}^{-u}+y_{uk}+a}}$$

We now need to take care of the degree-correction part:

$$f(\mathbf{Y}, \mathbf{z}, \mathbf{q}) = \prod_h \frac{B(\gamma \mathbf{1}_{n_h} + (d_u)_{u:z_u=h}))}{B(\gamma \mathbf{1}_{n_h})} n_h^{\sum_{u:z_u=h} d_u} \prod_k \frac{B(\gamma \mathbf{1}_{n_k} + (d_i)_{i:q_i=k}))}{B(\gamma \mathbf{1}_{n_k})} n_k^{\sum_{i:q_i=k} d_i}$$

As before we can interpret this as the change in the likelihood before adding element u. This can be written as:

$$\begin{aligned} f(\mathbf{Y}_{-u}, \mathbf{z}_{-u}, \mathbf{q}) = & \prod_{l \neq h} \frac{\prod_{u':z'_u=l} \Gamma(\gamma + d_{u'})}{\Gamma\left(\sum_{u':z_{u'}=l} (\gamma + d_{u'})\right)} \frac{\Gamma\left(\sum_{u':z_{u'}=l} \gamma\right)}{\prod_{u':z_{u'}=l} \Gamma(\gamma)} n_l^{\sum_{u':z_{u'}=l} d_{u'}} \\ & \frac{\prod_{u' \neq u:z_{u'}=h} \Gamma(\gamma + d_{u'})}{\Gamma\left(\sum_{u' \neq u:z_{u'}=h} \gamma + d_{u'}\right)} \frac{\Gamma\left(\sum_{u' \neq u:z_{u'}=h} \gamma\right)}{\prod_{u' \neq u:z_{u'}=h} \Gamma(\gamma)} (n_h - 1)^{\sum_{u' \neq u:z_{u'}=h} d_{u'}} \\ \propto & \prod_{l \neq h} \frac{\prod_{u':z_{u'}=l} \Gamma(\gamma + d_{u'})}{\Gamma(n_l \gamma + d_l^{-u'})} \frac{\Gamma(n_l \gamma)}{\prod_{u':z_{u'}=l} \Gamma(\gamma)} n_l^{d_l} \\ & \frac{\prod_{u' \neq u:z_{u'}=h} \Gamma(\gamma + d_{u'})}{\Gamma((n_h - 1)\gamma + d_h^{-u'})} \frac{\Gamma((n_h - 1)\gamma)}{\prod_{u' \neq u:z_{u'}=h} \Gamma(\gamma)} (n_h - 1)^{d_h^{-u'}} \end{aligned}$$

Where $d_h = \sum_{u:z_u=h} d_u$ and d_h^{-u} denotes the same quantity excluding user u. When adding element u the function becomes the following:

$$f(\mathbf{Y}, \mathbf{z}_{-u}, z_u = h, \mathbf{q}) = \prod_{l \neq h} \frac{\prod_{u':z_{u'}=l} \Gamma(\gamma + d_{u'})}{\Gamma(n_l\gamma + d_l^{-u'})} \frac{\Gamma(n_l\gamma)}{\prod_{u':z_{u'}=l} \Gamma(\gamma)} n_l^{d_l}$$
$$\frac{\Gamma(\gamma + d_u) \prod_{u' \neq u:z_{u'}=h} \Gamma(\gamma + d_{u'})}{\Gamma((n_h + 1)\gamma + d_h)} \frac{\Gamma(n_h\gamma)}{\Gamma(\gamma) \prod_{u' \neq u:z_{u'}=h} \Gamma(\gamma)} (n_h + 1)^{d_h}$$

Taking the ratio of the two parts we get:

$$\frac{f(\mathbf{Y}, z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{f(\mathbf{Y}_{-u}, \mathbf{z}_{-u}, \mathbf{q})} \propto \frac{\Gamma((n_h - 1)\gamma + d_h^{-u})}{\Gamma((n_h + 1)\gamma + d_h)} \frac{\Gamma(n_h\gamma)}{\Gamma((n_h - 1)\gamma)\Gamma(\gamma)} \frac{n_h^{d_h}}{(n_h - 1)^{d_h^{-u}}}$$

The update rules follows the following form:

$$\frac{p(\mathbf{Y} \mid z_u = h, \mathbf{z}_{-u}, \mathbf{q})}{p(\mathbf{Y}_{-u} \mid \mathbf{z}_{-u}, \mathbf{q})} \propto \frac{\Gamma((n_h - 1)\gamma + d_h^{-u})}{\Gamma((n_h + 1)\gamma + d_h)} \frac{\Gamma(n_h\gamma)}{\Gamma((n_h - 1)\gamma)} \frac{n_h^{d_h}}{(n_h - 1)^{d_h^{-u}}}$$
$$\prod_{k=1}^{K} \frac{\Gamma(m_{hk}^{-u} + y_{uk} + a)}{\Gamma(m_{hk}^{-u} + a)} \cdot \frac{(b + (n_h - 1)n_k)^{m_{hk}^{-u}+a}}{(b + n_h n_k)^{m_{hk}^{-u}+y_{uk}+a}}$$

Using a similar process we can derive the update for the items as:

$$\frac{p(\mathbf{Y} \mid q_i = k, \mathbf{q}_{-i}, \mathbf{z})}{p(\mathbf{Y}_{-i} \mid \mathbf{q}_{-i}, \mathbf{z})} \propto \frac{\Gamma((n_k - 1)\gamma + d_k^{-i})}{\Gamma((n_k + 1)\gamma + d_k)} \frac{\Gamma(n_k\gamma)}{\Gamma((n_k - 1)\gamma)} \frac{n_k^{d_k}}{(n_k - 1)^{d_k^{-i}}}$$
$$\prod_{h=1}^{H} \left[\frac{\Gamma(m_{hk}^{-i} + y_{ih} + a)}{\Gamma(m_{hk}^{-i} + a)} \times \frac{(b + (n_k - 1)n_h)^{m_{hk}^{-i}+a}}{(b + n_k n_h)^{m_{hk}^{-i}+y_{ih}+a}} \right]$$