Overlapping Communities on Large-Scale Networks:

Benchmark Generation and Learning via Adaptive Stochastic Optimization

Alessandro Antonio Grande

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2023

Abstract

Overlapping Communities on Large-Scale Networks: Benchmark Generation and Learning via Adaptive Stochastic Optimization

Alessandro Antonio Grande

This dissertation builds on two lines of research that are related to the task of community detection on large-scale network data.

Our first contribution is a novel generator for large-scale networks with overlapping communities. Synthetic generators are essential for algorithm testing and simulation studies for networks, as these data are scarce and constantly evolving. We propose a generator based on a flexible random graph model that allows for the control of two complementary measures of centrality – the degree centrality and the eigencentrality. For an arbitrary centrality target and community structure, we study the problem of recovering the model parameters that enforce such targets in expectation. We find that this problem always admits a solution in the parameter space, which is also unique for large graphs. We propose to recover this solution via a properly initialized multivariate-Newton Raphson algorithm. The resulting benchmark generator is able to simulate networks with a billion edges and hundreds of millions of nodes in 30 seconds, while reproducing a wide spectrum of network topologies – including assortative mixing and power-law centrality distributions.

Our second contribution involves variance reduction techniques for stochastic variational inference (SVI). SVI scales approximate inference to large-scale data – including massive networks – via stochastic optimization. SVI is efficient because, at each iteration, it only uses a random minibatch of the data to produce a noisy estimate of the gradient. However, such estimates can suffer from high variance, which slows down convergence. One strategy to reduce the variance of the gradient is to use importance sampling, biasing the distribution of data for each minibatch towards

the data points that are most influential to the inference at hand. Here, we develop an importance sampling strategy for SVI. Our adaptive stochastic variational inference algorithm (AdaSVI) reweights the sampling distribution to minimize the variance of the stochastic natural gradient. We couple the importance sampling strategy with an adaptive learning rate providing a parameter-free stochastic optimization algorithm where the only user input required is the minibatch size. We study AdaSVI on a matrix factorization model and find that it significantly improves SVI, leading to faster convergence on synthetic data.

Table of Contents

List of Figures

Acknowledgements

My first *grazie* goes to my advisor Dave Blei for always being supportive of my work and for creating a safe space for research to flourish. Dave's passion for research is contagious and resulted in many inspiring conversations that I greatly enjoyed. I look forward to those that are yet to come. My second *grazie* goes to Yang Feng, who advised me in the first years of my Ph.D. I thank Yang for his infinite patience with my early research struggles and for encouraging me to pursue different approaches whenever I got stuck on a problem. I also thank the other members of my committee – Sumit Mukherjee, John Cunningham, and John Paisley – for their helpful comments on this dissertation.

Of all the faculty and staff that I met during these years at Columbia, a special thanks goes to Mark Brown, Michael Sobel, Dood Kalicharan, and Anthony Cruz. I would also like to thank the GSAS Writing Studio and fellow dissertation writers Anna Waller and Alexander Ekserdjian for making the task of writing this dissertation somewhat less daunting.

To acknowledge the friends that supported me during my Ph.D., I decided to distribute my gratitude in the form of awards. Miguel Ángel Garrido wins the *best overall Ph.D. companion* award for embodying the definition of a true friend and for his *tortilla de patatas*. William Michael Eull gets the *best roommate* award despite the many atrocities that he perpetrated against Italian food in our kitchen. Keyon Vafa wins the *best movie critic* award for all the great movies we watched together, such as *A Marriage Story*, a devastating drama on divorce that Keyon made me watch as soon as I was back from my honeymoon. Owen Ward and Andrew Davison get the *best deskmate* and *best custodian of the stats lounge* awards, respectively, for the many great conversations we

shared in the department. I also want to thank Paolo Baudissone, Cecilia Balocchi, Luca Perdoni, Valerio Proietti, Matteo Sordello, Niccolò Dalmasso, and Cecilia Ferrando for their support in different stages of my Ph.D.

During my many trips to Italy, my family made sure that I stocked up on food, love, and wine. For that, I thank: my grandmothers Anna and Nilde; my grandfathers Antonio and Franco; my aunts Cinzia, Isabella, and Rosella; my uncles Fabio, Gianluca, and Vincenzo; and my cousins Anna Camilla, Giuseppe Paolo, Matteo, and Arianna Luce. As if I hadn't enough support already, I was lucky to acquire a second family during my Ph.D. I would then like to thank my *Famiglia di Serie B*: Maria Rosaria, Bruno, Cristina, Harry, Matilde, Laura, and Lucia.

This dissertation would not be possible without the continuous support of my parents. A big *grazie* goes to them: to my mum, for overcoming her fear of flying every time she visited me across the ocean; to my father, for being genuinely enthusiastic about every bit of my work.

Lastly, the most special *grazie* goes to my wife Laura, who sustained me in this journey from start to finish. Thank you for bringing levity when I took myself too seriously; for joining our countless walks through Riverside Park; for going great lengths to find the best *sachertorte* in town; for letting me buy far too many wine bottles; for proofreading every line of this dissertation; for forgiving me for all the things I forgot to thank you for; and for being my home.

Ai miei nonni Anna e Antonio, Nilde e Franco

A mia moglie Laura

Chapter 1: Preliminaries

In this chapter, we review variational inference methods and discuss the Poisson factorization model for overlapping communities on networks. This random graph model is the underlying engine of the network generators in Chapter 2 and 3. The rest of this chapter serves as background for the stochastic variational inference algorithm in Chapter 4.

We proceed as follows. We first provide an overview of variational inference methods. Next, we define the variational objective (Section 1.2) and discuss a coordinate ascent algorithm for a large family of models, called conditionally conjugate models (Section 1.3). In Section 1.4, we review stochastic variational inference. Finally, in Section 1.5, we apply the variational paradigm to the Poisson factorization model for overlapping community detection on large-scale networks.

1.1 An overview of variational inference methods

A central problem in statistics for large data is that of approximating hard to compute densities. For example, in Bayesian statistics, any inference task is framed in terms of the posterior distribution. As the complexity of models increases, exact posterior inference becomes intractable: on large datasets, computing the exact posterior amounts to integrating over millions of latent variables.

This is where variational inference methods come to aid. The main idea (Jordan et al., 1999; Wainwright and Jordan, 2008) is that of introducing a parametric approximation that minimizes the Kullback–Leibler (KL) divergence from the target distribution. While this problem is still intractable, it can be rewritten up to an added constant in terms of an alternative objective, named the evidence lower bound (ELBO). The variational problem thus reduces to finding the approximant that maximizes the ELBO (or equivalently, minimizes the KL divergence).

When the full conditionals belong to the exponential family, the variational problem can be tackled efficiently via a coordinate ascent algorithm (also known as coordinate ascent variational inference, CAVI). A particular instance of this class is given by conditionally conjugate models, where some latent variables are tied to single data points while other – the model parameters – govern the whole data set. For this family, CAVI updates can be derived in closed form. This result is of great interest since it makes VI readily available for a large family of models that find application in many disciplines, such as computational biology, computational neuroscience, and natural language processing.

Recent advances in the past decade have made VI scalable to larger datasets and applicable to a wider class of models. In terms of scalability, while efficient, CAVI requires a full pass through the whole dataset at each iteration. As this is unfeasible for massive datasets, inference can be further scaled up via stochastic optimization (Stochastic variational inference, Hoffman et al., 2013). In terms of applicability, many complicated models fall outside the class of conditionally conjugate models. The resulting ELBO presents intractable expectations that prevent analytical solutions. Black-box variational inference (BBVI, Ranganath, Gerrish, and Blei, 2014) circumnavigates this problem by rewriting the gradient of the ELBO as an expectation, which can be then estimated via Monte Carlo methods, thus making VI applicable to many complicated models. This line of research falls outside the scope of this work.

1.2 Variational inference

Consider a probabilistic model where a set of latent factors drive the data generation process. Let $\boldsymbol{x} = (x_i)_{i\in[n]}$ denote the data and let $\boldsymbol{z} = (z_j)_{j\in[m]}$ denote such a set of latent variables. The joint density $p(\boldsymbol{x}, \boldsymbol{z})$ univocally defines the model. The inference goal is to compute the conditional distribution of the latent factors given the data:

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x})}. \tag{1.1}$$

The Poisson factorization random graph model falls under this setting. The latent propensities of nodes to connect through some communities drive the edge generation between nodes in the network. The conditional distribution of the propensities given the observed edges defines the community assignments of the nodes.

The normalizing constant in Eq. (1.1), also known as the evidence, is the marginal distribution of the data and it can be obtained by intergrating out the latent variables form the joint density, $p(\boldsymbol{x}) = \int_z p(\boldsymbol{x}, z)\mathrm{d}z$. In practice, this integral is intractable, in that it does not always admit a closed-form or it takes exponential time to compute.

1.2.1 The evidence lower bound

Variational inference methods seek to approximate the conditional $p(z \mid \boldsymbol{x})$ by minimizing the Kullback-Leibler (KL) divergence between $p(z \mid \boldsymbol{x})$ and a family of approximant distributions. Let $\mathcal{Q}$ denote such a family and q an arbitrary member of it. The inference problem then amounts to:

$$\min_{q \in \mathcal{Q}} \mathrm{KL}(q||p(\cdot \mid \boldsymbol{x})) = \min_{q \in \mathcal{Q}} \mathbb{E}_q(\log(q(z))) - \mathbb{E}_q \log(p(z \mid \boldsymbol{x})). \tag{1.2}$$

The KL divergence is asymmetric, non-negative, and it is minimized if and only if $q \equiv p(\cdot \mid \boldsymbol{x})$. However, th objective still depends on the conditional $p(z\cdot \mid \boldsymbol{x})$, which in turn depends on the intractable evidence.

By expanding the conditional $p(\boldsymbol{x} \mid z)$, it follows that the problem in Eq. (1.2) reduces up to an additive constant to:

$$\max_{q \in \mathcal{Q}} \mathbb{E}_q(\log p(\boldsymbol{x}, z)) - \mathbb{E}_q \log q(z, \boldsymbol{\beta}). \tag{1.3}$$

The objective in Eq. (1.3) is known as the Evidence Lower Bound (ELBO). Minimizing the KL divergence is tantamount to maximizing the ELBO. While the problems in Eq.s (1.2) and (1.3) are equivalent, the ELBO does not directly target the conditional $p(\cdot \mid \boldsymbol{x})$ and it is therefore tractable.

1.2.2 Coordinate ascent over the mean-field variational family

The complexity of the problem in Eq. (1.3) depends on the family of approximants $\mathcal{Q}$. This family has to be simple enough to allow for efficient optimization, yet flexible enough to approximate the conditional distribution of the latent variables. A popular choice of $\mathcal{Q}$ is the mean-field family, under which the latent variables are mutually independent and the distribution of each z_j can be chosen freely. More specifically, an arbitrary approximant $q \in \mathcal{Q}$ from the mean-field family takes the following form:

$$q(\boldsymbol{z}) = \prod_{j=1}^{m} q_j(z_j), \tag{1.4}$$

where the variational density q_j encodes the distribution of the latent variable z_j.
Under the mean-field family, the optimization in Eq. (1.3) can be conveniently tackled by a coordinate-ascent algorithm. Coordinate-ascent variational inference (CAVI) sequentially maximizes the ELBO with respect to a given factor while keeping the others fixed. To derive the updates note that, after replacing Eq. (1.4) in Eq. (1.3) for q, the only term of the ELBO that depends on $q_j(z_j)$ is:

$$\text{ELBO}(q_j) \propto \mathbb{E}_{q_j} \mathbb{E}_{q_{-j}} \log p(\boldsymbol{x}, \boldsymbol{z}) - \mathbb{E}_{q_j} \log q_j(z_j), \tag{1.5}$$

where $q_{-j} = \prod_{\ell \neq j} q_\ell$. By defining the variational density $q^*(z_j)$ such that:

$$q_j^*(z_j) \propto \exp\left(\mathbb{E}_{q_{-j}} \log p(\boldsymbol{x}, \boldsymbol{z})\right), \tag{1.6}$$

it follows that Eq. (1.5) can be rewritten as:

$$\text{ELBO}(q_j) \propto \mathbb{E}_{q_j} \log q^*(z_j) - \mathbb{E}_{q_j} \log q_j(z_j) = \text{KL}(q_j^* || q_j). \tag{1.7}$$

This shows that Eq. (1.6) is the optimal choice of the variational density and thus Eq. (1.6) defines the coordinate updates. The optimal factor q_j^* is given by the exponentiated expected log of the joint density $p(\boldsymbol{x}, \boldsymbol{z})$, where the expectation is taken with respect to all the other factors $\ell \neq j$. The resulting algorithm is stated in Algorithm 1.1.

Algorithm 1.1 Coordinate-Ascent Variational Inference

Input: model $p(\boldsymbol{x}, \boldsymbol{z})$, data $\boldsymbol{x}$.

Output: a mean-field variational density q.

Initialize $q = \prod_{j=1}^{m} q_j$.
while *ELBO has not converged* **do**
  for $j = 1, \ldots, m$ **do**
    Compute the update $q(z_j) \propto \exp\left(\mathbb{E} q_{-j} \log p(\boldsymbol{x}, \boldsymbol{z})\right)$.
  Compute $ELBO(q) = \exp\left(\mathbb{E}_{q_j} \mathbb{E}_{q_{-j}} \log p(\boldsymbol{x}, \boldsymbol{z})\right)$.

1.3 Conditional conjugate models

We define the class of conditional conjugate models and then state the resulting CAVI updates.

1.3.1 Setting

Let $\boldsymbol{x} = (x_i)_{i \in [n]}$ denote the data, $\boldsymbol{\beta}$ denote a vector of global latent variables, and $\boldsymbol{z} = (z_i)_{i \in [n]}$ denote a vector of local latent variables. While $\boldsymbol{\beta}$ may parametrize the distribution of any of the data, the latent variable z_i controls only the local "context" of the i-th data point x_i. The model reads:

$$p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{i=1}^{n} p(x_i, z_i \mid \boldsymbol{\beta}). \tag{1.8}$$

Under conditional conjugacy, the distributions in Eq. (1.8) are set so that the complete conditionals of the latent variables are an exponential family. The density of the i-th local context (z_i, x_i) given

$\boldsymbol{\beta}$ is then defined as:

$$p(z_i, x_i \mid \boldsymbol{\beta}) = h(z_i, x_i) \exp\left\{\eta(\boldsymbol{\beta})^\top t(z_i, x_i) - a(\boldsymbol{\beta})\right\}, \tag{1.9}$$

where h is the base measure, t the sufficient statistic, η the natural parameter, and a the log-normalizer. The prior $p(\boldsymbol{\beta})$ is set to be the conjugate prior of the global latent variables with hyperparameter $\boldsymbol{\alpha}$:

$$p(\boldsymbol{\beta}) = h(\boldsymbol{\beta}) \exp\left\{\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -a(\boldsymbol{\beta})] - a(\boldsymbol{\alpha})\right\}. \tag{1.10}$$

Finally, the complete conditional of the local latent variables z_i is also an exponential family:

$$p(z_i \mid x_i, \boldsymbol{\beta}, \boldsymbol{z}_{-i}, \boldsymbol{x}_{-i}) = h(z_i) \exp\left\{\eta(\boldsymbol{\beta}, x_i)^\top z_i - a(\eta(\boldsymbol{\beta}, x_i))\right\}. \tag{1.11}$$

1.3.2 CAVI updates

In conditional conjugate models, the coordinate ascent updates can be readily derived. Consider a mean-field family $\boldsymbol{Q}$ over the latent variables $(\boldsymbol{\beta}, \boldsymbol{z})$. It follows from Equations (1.6) and (1.11) that:

$$\begin{aligned} q^*(z_i) &\propto \exp\left(\mathbb{E}_{\boldsymbol{\beta}, \boldsymbol{z}_{-i}} \log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta})\right) \\ &\propto \exp\left(\mathbb{E}_{\boldsymbol{\beta}, \boldsymbol{z}_{-i}} \log p(\boldsymbol{z}_i \mid \boldsymbol{x}, \boldsymbol{z}_{-i}, \boldsymbol{\beta})\right) \\ &= \exp\left\{\mathbb{E}_{\boldsymbol{\beta}}\left(\log h(z_i) + \eta(\boldsymbol{\beta}, x_i)^\top z_i - a(\eta(\boldsymbol{\beta}, x_i))\right)\right\} \\ &\propto h(z_i) \exp\left(\mathbb{E}_{\boldsymbol{\beta}}\left[\eta(\boldsymbol{\beta}, x_i)\right]^\top z_i\right). \end{aligned} \tag{1.12}$$

This means that the optimal variational factors belong to the same exponential family of the corresponding complete conditionals, with the same base measure h and log normalizer a. Also, if we let $\boldsymbol{\lambda}$ and $\boldsymbol{\varphi}$ parametrize the factors of the global and local latent variables respectively, we further

have:

$$\varphi_i^* = \mathbb{E}_{\lambda}\eta(\boldsymbol{\beta}, x_i). \tag{1.13}$$

In other words, the variational parameter of the local latent factor $q_i(z_i)$ is given by the expected natural parameter of the corresponding complete conditional. In CAVI for conditional conjugate models, the optimal local variational parameters is then a function of the data from the i-th local context x_i and of the current instance of the global variational parameters $\boldsymbol{\lambda}$.

As the same derivations apply to the global latent factors, analogues updates can be derived for the global variational parameters, too. Under the prior in Equation (1.10), the complete conditional of the global latent variables is an exponential family with the following natural parameter:

$$\boldsymbol{\alpha} + \left[\sum_{i=1}^{n} t(z_i, x_i), n\right]. \tag{1.14}$$

It follows that the CAVI updates for global variational parameters are:

$$\boldsymbol{\lambda}^* = \boldsymbol{\alpha} + \left[\sum_{i=1}^{n} \mathbb{E}_{\varphi_i} t(z_i, x_i), n\right]. \tag{1.15}$$

1.4 Stochastic variational inference

Stochastic variational inference (SVI, Hoffman et al., 2013) scales CAVI to large-scale datasets for conditionally conjugate models via stochastic optimization. This method optimizes the ELBO with respect to the global variational parameters. It does so by using, at each iteration, a random minibatch of the data to produce a noisy estimate of the natural gradient. Before presenting the main algorithm, we briefly review the natural gradient and stochastic optimization.

1.4.1 The natural gradient

Parameter spaces are non Euclidean, in that the classical notion of Euclidean distance does not capture their structure. To this end, natural gradient methods model the parameter space as a Riemannian space, introducing a more suitable metric. If one wants to run gradient ascent under this new geometry, what direction should they follow? The natural gradient is the steepest ascent direction of a function in a Riemannian space.

More specifically, consider a parameter space $\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^D\}$ on which a function $f(\boldsymbol{\theta})$ is defined. If Θ is Riemannian, the squared length of a small incremental vector $d\boldsymbol{\theta}$ from $\boldsymbol{\theta}$ to $\boldsymbol{\theta} + d\boldsymbol{\theta}$ is given by the quadratic form:

$$|d\boldsymbol{\theta}|^2 = d\boldsymbol{\theta}^\top T(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{1.16}$$

for some D-by-D matrix T, which is called the Riemannian metric tensor and depends in general on $\boldsymbol{\theta}$[1]. The steepest ascent direction of $f(\boldsymbol{\theta})$ is defined by the incremental vector $d\boldsymbol{\theta}^*$ that maximizes $f(\boldsymbol{\theta} + d\boldsymbol{\theta})$, or equivalently:

$$d\boldsymbol{\theta}^*(\boldsymbol{\theta}) = \underset{d\boldsymbol{\theta}:|d\boldsymbol{\theta}|^2=\varepsilon^2}{\arg\max}\ f(\boldsymbol{\theta} + d\boldsymbol{\theta}), \tag{1.17}$$

where ε is a sufficiently small constant. It can be shown that the steepest direction of $f(\boldsymbol{\theta})$ in a Riemannian space is:

$$\widetilde{\nabla} f(\boldsymbol{\theta}) = T^{-1}(\boldsymbol{\theta}) \cdot \nabla f(\boldsymbol{\theta}), \tag{1.18}$$

where ∇f denotes the gradient of f. We call $\widetilde{\nabla} f$ the natural gradient of f.

As firstly discussed by Rao (1945), a suitable Riemannian metric for parameter spaces is the one

[1] In particular, if Θ is Euclidean, T is the identity matrix and equation (1.16) reduces to:

$$|d\boldsymbol{\theta}|^2 = d\boldsymbol{\theta}^\top d\boldsymbol{\theta}.$$

for which T is set to be equal to Fisher information matrix. Under such a choice of T, the natural gradient points in the same direction of the following incremental vector:

$$\arg\max_{d\boldsymbol{\theta}} f(\boldsymbol{\theta} + d\boldsymbol{\theta}) \qquad \text{subject to } \mathrm{KL}_{\mathrm{sym}}(\boldsymbol{\theta}, \boldsymbol{\theta} + d\boldsymbol{\theta}) < \varepsilon. \tag{1.19}$$

Here, $\mathrm{KL}_{\mathrm{sym}}$ denotes the symmetrized KL divergence, i.e. $\mathrm{KL}_{\mathrm{sym}}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathrm{KL}(\boldsymbol{\theta}||\boldsymbol{\theta}') + \mathrm{KL}(\boldsymbol{\theta}'||\boldsymbol{\theta})$. In other words, the natural gradient points in the direction of steepest ascent in the parameter space where the local distance between two distributions is defined by the symmetrized KL metric.

1.4.2 Stochastic optimization

Stochastic optimization climbs an objective by following noisy estimates of its gradient. Such estimates are usually cheaper to compute than the true gradient and their randomness can prevent algorithms from getting stuck in shallow local optima. If the step size is decreasing and it satisfies certain conditions, then convergence to an optimum is guaranteed.

More specifically, let f denote the objective, G an unbiased estimator of its gradient ∇f. At each iteration t, we observe a realization $g^{(t)}$ of G and make a step ρ_t in that direction:

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} + \rho_t \cdot g^{(t)}. \tag{1.20}$$

As shown in the seminal paper by Robbins and Monro (1951), if the step size schedule $(\rho_t)_t$ is such that:

$$\sum_t \rho_t = \infty, \quad \sum \rho_t^2 < \infty, \tag{1.21}$$

then $\boldsymbol{\theta}^{(t)}$ will converge the to global optimum if f is convex or to a local optimum if f is not convex. A further result by Bottou et al. (1998) shows that the above holds also when we multiply G by a sequence of positive-definite matrices $(T_t^{-1})_t$ with bounded eigenvalues, which leads to the

following update rule:

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} + \rho_t \cdot T_t^{-1} g^{(t)}. \tag{1.22}$$

In particular, if T is set to be equal to Fisher information matrix, the noisy gradient G in Eq. (1.20) can be replaced with a noisy estimator $T_t^{-1}G$ of the natural gradient of f.

1.4.3 The algorithm

SVI optimizes the global variational parameters with stochastic natural gradient ascent of the ELBO. At each iteration, one samples a minibatch of the data and computes the CAVI updates of the corresponding local parameters. These parameters are in turn used to build a noisy estimate of the natural gradient, which allows to run stochastic optimization on the global parameters.
For conditional conjugate models, the natural gradient can be readily computed. As shown in (Hoffman et al., 2013), the classical Euclidean gradient of the ELBO with respect to the global variational parameters $\boldsymbol{\lambda}$ can be written as:

$$\nabla_{\boldsymbol{\lambda}} \text{ELBO} = a''(\boldsymbol{\lambda}) \, (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}) \,, \tag{1.23}$$

where $\boldsymbol{\lambda}^*$ is the optimal global variational parameter defined by the coordinate update Eq. (1.15). Because in exponential families Fisher information matrix corresponds to the second derivative of the log-normalizer, it follows from Eq.s (1.18) and (1.15) that the natural gradient of the ELBO w.r.t. $\boldsymbol{\lambda}$ reduces to:

$$\widetilde{\nabla}_{\boldsymbol{\lambda}} \text{ELBO} = \boldsymbol{\lambda}^* - \boldsymbol{\lambda} = \boldsymbol{\alpha} + \left[\sum_{i=1}^{n} \mathbb{E}_{\varphi_i} t(z_i, x_i), n \right] - \boldsymbol{\lambda}. \tag{1.24}$$

The natural gradient is just given by the difference between the coordinate updates of $\boldsymbol{\lambda}$ and its current instance. Hence, for conditionally conjugate models, the natural gradient does not require any additional derivations with respect to the classical CAVI updates. However, the gradient in

Eq. (1.24) is still too expensive to compute, as it requires a pass through the whole dataset (similarly to the CAVI updates). We thus turn to stochastic optimization.

One can estimate the natural gradient on a random subset (or "minibatch") of the data by computing only the contributions of the points that have been sampled. After reweighting each contribution by its inverse sampling probability, the resulting quantity forms an unbiased estimate of the natural gradient. In detail, let S denote a random minibatch of the data of size m sampled uniformly at random. Then:

$$\boldsymbol{\alpha} + \frac{n}{m}\left[\sum_{i \in S} \mathbb{E}_{\varphi_i} t(z_i, x_i), n\right] - \boldsymbol{\lambda} \tag{1.25}$$

is an unbiased estimator of the gradient in Eq. (1.24).

The SVI algorithm proceeds as follows: first, sample a minibatch of the data and update the corresponding local parameters via the CAVI Eq. (1.13); using these parameters, compute the noisy gradient in Eq. (1.25); run stochastic natural gradient ascent of the ELBO on $\boldsymbol{\lambda}$ with the noisy gradient. The algorithm is fully stated below.

Algorithm 1.2 Stochastic variational inference

Input: model $p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta})$, data $\boldsymbol{x}$, minibatch size m, and step size sequence $(\varepsilon_t)_{t\in\mathbb{N}}$.

Output: global variational densities $q_{\boldsymbol{\lambda}}(\boldsymbol{\beta})$.

while *Noisy ELBO has not converged* **do**

  if *first iteration* **then**

    Set $S = [n]$.

  else

    Sample with replacement a minibatch S of size m according to the weights $\widetilde{\boldsymbol{p}}$.

  Optimize the local variational parameters φ_i^* for any $i \in S$:

$$\varphi_i^* = \mathbb{E}_{\boldsymbol{\lambda}}\eta(\boldsymbol{\beta}, x_i).$$

  Compute the coordinate update:

$$\hat{\boldsymbol{\lambda}} = \boldsymbol{\alpha} + \left[\frac{n}{m}\sum_{i\in S}\mathbb{E}_{\varphi_i^*}t(z_i, x_i), n\right].$$

  Update the global variational parameters:

$$\boldsymbol{\lambda}_{t+1} = (1-\varepsilon_t)\boldsymbol{\lambda}_t + \varepsilon_t\hat{\boldsymbol{\lambda}}.$$

1.5 Overlapping community detection via Poisson factorization

To illustrate the variational inference setting that we have presented so far, we derive the CAVI and SVI algorithms for the Poisson-factorization (PF) random graph model. These methods can be used to infer the community assignments of nodes in a network.

We first recall the model formulation. Next, we discuss a latent variable augmentation that makes the model conditionally conjugate and state the resulting CAVI and SVI algorithms.

1.5.1 The model

Consider a network with n nodes over K overlapping communities and let $(A_{ij})_{i,j\in[n]}$ denote its adjacency matrix. A_{ij} is the number of edges that node i and j share; if $A_{ij} = 0$, then node i and j are not connected. Under Poisson factorization, each node i is given a vector of latent propensities $\boldsymbol{\theta}_i = (\theta_{ik})_{i\in[n],k\in[K]}$ that encode their preferences to connect via the K communities. The mean number of edges between a node pair is then given by the joint propensities of the nodes to connect via a single or multiple communities. More specifically, the generative model reads:

- Draw the latent propensities:

$$\theta_{ik} \sim \text{Gamma}(\alpha, \beta), \tag{1.26}$$

  for each node $i \in [n]$ and community $k \in [N]$;

- Draw edges:

$$A_{ij} \sim \text{Poisson}(\sum_{k\in[K]} \theta_{ik}\theta_{jk}), \tag{1.27}$$

  for each node pair (i, j) with $i \neq j$.

Here, communities are seen as group of edges rather than nodes. Under this edge-centric perspective, overlapping communities arise because nodes connect through different types of edges. For example, in social networks, people connect because they share activities, interests, familiar ties, and so forth.

This model is a Bayesian extension of the random graph model in Ball, Karrer, and Newman (2011), who were the first to propose a non-negative matrix factorization model with Poisson likelihood for network data. In our Bayesian setting, we equip the latent propensities with a Gamma prior. More recent advances include Bayesian nonparametric extensions (Zhou, 2015; Ayed and Caron, 2021) that treat the number of communities as unknown. For simplicity, here we assume

that K is given.

We note that the model allows for multi-edges. Since most real-world network data come in the form of simple graphs ($A_{ij} \in \{0, 1\}$), this model may seem unrealistic. However, as pointed in Newman (2010), the error introduced by multiedges tend to vanish as the network size n increases, typically as $1/n$ in the limit.

1.5.2 CAVI and SVI via latent variable augmentation.

The Poisson factorization model in Eq.s (1.26)–(1.27) is not conditional conjugate. However, one can introduce a set of auxiliary latent variables that make the model conditional conjugate while preserving its original formulation. Latent variable augmentation is a common technique in the variational inference literature: recent notable applications include Gaussian processes (Hensman, Fusi, and Lawrence, 2013) and logistic models (Durante and Rigon, 2019).

Following Gopalan, Hofman, and Blei (2015), we augment the model by introducing the unobserved variable z_{ijk}, which denotes the latent number of edges between nodes i and j that are formed via community k. Under this augmentation, the model in Eq.s (1.26)–(1.27) reads:

$$\theta_{ik} \sim \text{Gamma}(\alpha, \beta), \quad z_{ijk} \mid \boldsymbol{\theta} \sim \text{Poisson}(\theta_{ik}\theta_{jk}), \quad A_{ij} = \sum_{k \in [K]} z_{ijk}. \tag{1.28}$$

Since the sum of independent Poisson random variables is also Poisson, the new model formulation preserves the original distributional assumption in Eq. (1.27). Also, this formulation nicely fits into the framework of conditional conjugate models with local and global latent variables. The node-dependent propensities $\boldsymbol{\theta}$ are the global latent variables, while the edge-dependent unobserved counts $\boldsymbol{z}$ are the local latent variables.

Under this new formulation, the complete conditionals of the latent variables belong to the exponential family. In particular, the complete conditional of the latent propensities follows a Gamma distribution, which is due to the Gamma-Poisson conjugacy. The complete conditional of the auxiliary latent variables turns out to be multinomial: this follows from the fact that the conditional

distribution of a vector of independent Poisson given their sum is multinomial. A full derivation of these distributions is presented in the next section.

CAVI It follows from Equation (1.6) that the optimal mean-field variational family for this model takes the following form:

$$z_{ij\cdot} \overset{q}{\sim} \text{Multinomial}(A_{ij}, \boldsymbol{\varphi}_{ij\cdot}), \tag{1.29}$$

$$\theta_{ik} \overset{q}{\sim} \text{Gamma}(\lambda_{ik}, \gamma_{ik}), \tag{1.30}$$

where $\boldsymbol{\varphi}_{ij\cdot}$ denotes the vector of multinomial probabilities parameterizing the variational factors of the latent edge counts $\boldsymbol{z}$, and $(\lambda_{ik}, \gamma_{ik})$ denote the Gamma scale and rate parameters for the factor of the latent propensities $\boldsymbol{\theta}$. Eq.s (1.13) and (1.15) yield the following CAVI updates:

$$\lambda_{ik} = \alpha + \sum_{j \neq i} A_{ij} \varphi_{ijk}\,, \tag{1.31}$$

$$\gamma_{ik} = \beta_i + \sum_{j \neq i} \frac{\lambda_{jk}}{\gamma_{jk}}\,, \tag{1.32}$$

$$\varphi_{ijk} \propto \exp\left\{\psi(\lambda_{ik}) - \log \gamma_{ik} + \psi(\lambda_{jk}) - \log \gamma_{jk}\right\}. \tag{1.33}$$

CAVI iterates through the updates in Eq.s (1.31)–(1.33) until convergence. The resulting variational parameters univocally define a mean-field approximant of the posterior distribution, which can then be used for Bayesian inference.

SVI The SVI updates readily follow from Eq.s (1.25) and (1.31)–(1.33). After sampling a mini-batch S with m edges, the corresponding local (i.e. edge-dependent) variational parameters $\boldsymbol{\varphi}$ are optimized via Eq. (1.33). This set of parameters is in turn used to assemble the noisy gradient for $\boldsymbol{\lambda}$:

$$\tilde{\nabla}_{\lambda_{ik}} ELBO = \alpha + \frac{n}{m} \sum_{(i,j) \in S} A_{ij} \varphi^*_{ijk} - \lambda_{ik}. \tag{1.34}$$

The gradient step is then:

$$\boldsymbol{\lambda}_{t+1} = (1 - \varepsilon_t)\boldsymbol{\lambda}_t + \varepsilon_t \left(\alpha + \frac{n}{m} \sum_{(i,j) \in S} A_{ij} \varphi^*_{ijk} \right)_{i \in [n], k \in [K]} . \tag{1.35}$$

As for the global rate parameters $\boldsymbol{\gamma}$, note that the update in Eq. (1.32) does not depend on the local parameters $\boldsymbol{\varphi}$. Hence, the natural gradient of the ELBO with respect to $\boldsymbol{\gamma}$ can be computed efficiently and the corresponding gradient ascent step will not be stochastic. The natural gradient reads:

$$\nabla_{\gamma_{ik}} ELBO = \beta_i + \sum_{j \neq i} \frac{\lambda_{ik}}{\gamma_{ik}} - \gamma_{ik}. \tag{1.36}$$

The gradient step is then:

$$\boldsymbol{\gamma}_{t+1} = (1 - \varepsilon_t)\boldsymbol{\gamma}_t + \varepsilon_t \left(\beta_i + \sum_{j \neq i} \frac{\lambda_{ik}}{\gamma_{ik}} \right)_{i \in [n], k \in [K]} . \tag{1.37}$$

Community memberships The main inference task is that of recovering communities. We note that the Poisson factorization model does not come with a precise definition of community memberships, which then need to be defined. Rather than setting a threshold for the propensities (whose scale is hard to interpret), we turn to the expected number of edges. A natural criterion to assign nodes to communities is to estimate their k-degree. The k-degree of a node i, which we denote by d_{ik}, is the number of edges that a node forms via community k. We then say that node i belongs to community k if its expected k-degree is greater than 1 – or equivalently, if node i forms at least one edge via community k in expectation. The k-degree can be readily estimated by approximating its posterior mean via CAVI:

$$\mathbb{E}(d_{ik} \mid A) = \sum_{i \neq j} \mathbb{E}(z_{ijk} \mid A_{ij}) \approx \sum_{j \neq i} A_{ij} \varphi^*_{ijk} = \lambda^*_{ik} - \alpha, \tag{1.38}$$

where $\boldsymbol{\varphi}^*$ and $\boldsymbol{\lambda}^*$ denote the CAVI variational parameters at convergence.

1.6 Derivations

ELBO

$$\mathcal{L}(\lambda_{i,k}, \gamma_{i,k}, \boldsymbol{\phi}_{i,j}) \propto \sum_{i \neq j,k} \Bigg[A_{i,j} \cdot \phi_{i,j,k} \cdot \big(\psi(\lambda_{i,k}) - \log \gamma_{i,k} + \psi(\lambda_{j,k}) - \log \gamma_{j,k} - \log \phi_{i,j,k}\big) - \frac{\lambda_{i,k}}{\gamma_{i,k}} \cdot \frac{\lambda_{j,k}}{\gamma_{j,k}} \Bigg] + \sum_{i,k} \Bigg[\alpha \cdot \big(\psi(\lambda_{i,k}) - \log \gamma_{i,k}\big) - \lambda_{i,k} \cdot \psi(\lambda_{i,k}) - (\beta_i - \gamma_{i,k}) \cdot \frac{\lambda_{i,k}}{\gamma_{i,k}} + \log \Gamma(\lambda_{i,k}) + \alpha \log \beta_i - \log \Gamma(\alpha) \Bigg]. \tag{1.39}$$

CAVI updates The complete conditionals of the latent propensities are:

$$p(\theta_{i,k} \mid \boldsymbol{z}_{i,.,k}, \boldsymbol{\theta}_{-i,k}) \propto p(\theta_{i,k}, \boldsymbol{z}_{i,.,k}, \boldsymbol{\theta}_{-i,k}) \tag{1.40}$$

$$\propto \theta_{i,k}^{\alpha_{i,k}-1} e^{-\beta_{i,k}\theta_{i,k}} \prod_j \theta_{i,k}^{z_{i,j,k}} e^{-\theta_{i,k}\theta_{j,k}} \tag{1.41}$$

$$\propto \text{Gamma}(\alpha_{i,k} + \sum_j z_{i,j,k}, \beta_{i,k} + \sum_{j \neq i} \theta_{j,k}). \tag{1.42}$$

As for the auxiliary latent variables, the complete conditionals read:

$$p(z_{i,j,k} \mid \theta_{i,k}\theta_{j,k}, A_{i,j}, z_{i,j,k'}) \propto p(z_{i,j,k}, \theta_{i,k}\theta_{j,k}, A_{i,j}, z_{i,j,k'}) \tag{1.43}$$

$$\propto (\theta_{i,k}\theta_{j,k})^{z_{i,j,k}} (z_{i,j,k}!)^{-1} \mathbb{1}_{\{\sum_k z_{i,j,k} = A_{i,k}\}}. \tag{1.44}$$

Hence, $\boldsymbol{z}_{i,j,.} \sim \text{Multinomial}(A_{i,j}, \boldsymbol{\varphi}_{i,j})$, where $\varphi_{i,j,k} = \frac{\theta_{i,k}\theta_{j,k}}{\sum_k \theta_{i,k}\theta_{j,k}}$.

Since both distributions belong to the exponential family, the model is complete conjugate, and then the variational updates can be easily derived as described in Blei, Kucukelbir, and McAuliffe (2017).

A memory-efficient implementation of CAVI The algorithm presented in Section 2 can be further refined so as to make it more computationally efficient. This can be achieved by pre-computing the terms that are needed to update the variational probabilities $\boldsymbol{\phi}$. This variation is memory-free as we still have to store $3 \cdot N \cdot K$ entries at each iteration.

Algorithm 1.3 Fast and memory-efficient CAVI for the Poisson factorization random graph model

Randomly initialize $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$.

while *ELBO has not converged* **do**

  Compute $\xi_{i,k} = \exp\{\psi(\lambda_{i,k}) - \log \gamma_{i,k}\}$ for all i, k.

  Set $\lambda_{i,k} = \alpha$ for all i, k.

  for *each edge* (i, j) **do**

    For all k, compute:

$$\phi_k = \xi_{i,k} \cdot \xi_{j,k}.$$

    Compute $s = \sum_k \phi_k$ and then set $\phi_k = \phi_k / s$ for all k.

    For all k, set:

$$\hat{\lambda}_{i,k} = \lambda_{i,k} + \phi_k,$$

$$\hat{\lambda}_{j,k} = \lambda_{j,k} + \phi_k.$$

  Set $\gamma_{i,k} = \beta_i + \sum_{j \neq i} \frac{\lambda_{j,k}}{\gamma_{j,k}}$ for all i, k.

  Return $\hat{d}_{i,k} = \lambda_{i,k} - \alpha$ for all i, k.

Variational EM One can derive coordinate-ascent updates for the hyperparameters α and $\boldsymbol{\beta}$, too. This approach is called Variational EM. To this end, we first isolate the terms of the ELBO that depend on α and $\boldsymbol{\beta}$ and then study the first order conditions. We have:

$$\text{ELBO}_{\alpha,\beta} = \sum_{i,k} \left((\alpha - 1)\mathbb{E}_q(\log \theta_{i,k}) - \beta_i \frac{\lambda_{i,k}}{\gamma_{i,k}} + \alpha \log \beta_i - \log \Gamma(\alpha) \right), \tag{1.45}$$

where $\mathbb{E}_q(\log \theta_{i,k}) = \psi(\lambda_{i,k}) - \log \gamma_{i,k}$. Then:

$$\frac{\partial}{\partial \alpha} \text{ELBO}_{\alpha,\beta} = \sum_{i,k} \left(\mathbb{E}_q(\log \theta_{i,k}) + \log \beta_i - \psi(\alpha) \right). \tag{1.46}$$

Hence, the coordinate update for α is:

$$\alpha^* = \psi^{-1} \left(\frac{\sum_{i,k} \mathbb{E}_q(\log \theta_{i,k}) + K \sum_i \log \beta_i}{N \cdot K} \right). \tag{1.47}$$

Also:

$$\frac{\partial}{\partial \beta_i} \text{ELBO}_{\alpha,\beta} = \sum_k -\frac{\lambda_{i,k}}{\gamma_{i,k}} + \frac{\alpha}{\beta_i}. \tag{1.48}$$

Hence, the coordinate update for β_i is:

$$\beta_i^* = \alpha \cdot K \cdot \left(\sum_k \frac{\lambda_{i,k}}{\gamma_{i,k}} \right)^{-1}. \tag{1.49}$$

The second order conditions can be easily verified. We have:

$$\frac{\partial^2}{\partial \alpha^2} \text{ELBO}_{\alpha,\beta} = -N \cdot K \cdot \psi'(\alpha) \leq 0, \tag{1.50}$$

since $\psi' \geq 0$. Also:

$$\frac{\partial^2}{\partial \beta_i^2} \text{ELBO}_{\alpha,\beta} = -\frac{K \cdot \alpha}{\beta_i^2} \leq 0. \tag{1.51}$$

Chapter 2: Generating networks with target degree centralities

2.1 Introduction

Any estimation task on networks benefits from a reliable generator of synthetic data. Real-world networks are often hard to access because of privacy and cost concerns, and may be constantly evolving. Such shortage of data makes synthetic networks essential for testing algorithms under different structures, complexities, and scales. This applies in particular to community detection methods. Because real-world data rarely come with community memberships, testing has to be carried out on synthetic networks with built-in communities, also known as benchmark networks.

A valid benchmark generator should be equipped with some key features. First of all, while early efforts at community detection have focused on disjoint communities, generators should allow for an overlapping built-in community structure. This is because in social and biological applications communities have been shown to overlap (Palla et al., 2005). Also, generators should replicate structural properties of interest of real-world networks, such as power-law degree distributions. Finally, with the advent of large-scale datasets, more and more attention has been given to the development of community detection algorithms for massive networks (Gopalan and Blei, 2013). To properly test these methods, generators need to be scalable, too.

In Chapter 2 and 3, we address the following question:

How can we generate large-scale networks with overlapping communities and a target structure of interest?

Most efforts in the literature provide sequential algorithms that target prescribed degree centrality sequences. For example, the de facto standard in the field (Lancichinetti, Fortunato, and Radicchi, 2008) and its overlapping extension (Lancichinetti and Fortunato, 2009) employ an edge switching

algorithm to produce networks with node degrees and community sizes that follow a power-law distribution. Even more recent works based on random graph models (Slota et al., 2019; Kamiński, Prałat, and Théberge, 2021) remain algorithmic in their nature, in that their probabilistic representations are posed to solve the task at hand – that of reproducing target degree distributions – rather than to capture a realistic community generation process.

To this end, consider a non-Bayesian version of the Poisson factorization model introduced in the previous chapter, where the n-by-n random adjacency matrix A is generated as follows:

$$A_{ij} \sim \text{Poisson}\left(\sum_{1 \le k \le K} \lambda_{ik}\lambda_{jk}\right), \tag{2.1}$$

independently for $i < j$. Recall that the parameter λ_{ik} can be interpreted as the propensity of node i to connect via community k. This formulation, originally proposed by Ball, Karrer, and Newman (2011), gives rise to K overlapping communities.

While simple, this is a principled, flexible, and scalable random graph model. First, the idea of link communities finds correspondence in the sociological literature with the social foci theory of Feld (1981), for which individuals connect around social entities known as foci. Secondly, as pointed out by Yang and Leskovec (2014), nodes that reside in overlaps of communities may be more densely connected than those that do not. While many models assume sparse community overlaps, the model formulation in Eq. (2.1) is flexible enough to capture dense overlaps, too. Finally, in contraposition to the sequential design of most benchmark generators, here edges are conditionally independent given propensities and thus their generation is *pleasingly parallel*.

We want to employ the model in Eq. (2.1) to generate networks that comply with a target structure of interest. Let $g : \mathbb{R}^{n \times n} \to \mathbb{R}^d$ denote a map that encodes such a structure. It takes as input the adjacency matrix and returns a d-dimensional statistic that captures the target structure. A natural way to formulate the network generation problem is to set $g(\cdot)$ equal to a given goal $\hat{\boldsymbol{g}} \in \mathbb{R}^d$ in

expectation:

$$\mathbb{E}_{\lambda} g(A) = \hat{\boldsymbol{g}}. \tag{2.2}$$

The subscript in the expectation emphasizes the dependence of the left-hand side above on the model parameters $\boldsymbol{\lambda}$. The task of generating networks under the constraint in Eq. (2.2) amounts to recovering at least one set of parameters $\boldsymbol{\lambda}^*$ that complies with Eq. (2.2), which can then be used to draw random adjacency matrices from the model in Eq. (2.1). Hence, for a given target structure g and goal $\hat{\boldsymbol{g}}$, we may wonder: does the problem in Eq. (2.2) admit a solution in the parameter space? If such a solution exists, is it unique? If it exists and is unique, can we recover it?
The focus of this chapter is to address such questions when setting g equal to the degree centrality.

2.1.1 Overview of results

We fix a vector $\hat{\boldsymbol{g}} \in \mathbb{R}^n$ of target degree centralities for the nodes and seek to to recover the model parameters $\boldsymbol{\lambda}$ that secure $\hat{\boldsymbol{g}} \in \mathbb{R}^n$ in expectation. Under the Poisson factorization model in Eq. (2.1), the system of equations in Eq. (2.2) reduces to a system of real multivariate polynomials of degree 2 in $\boldsymbol{\lambda}$. Since the parameters are constrained to be positive, we want to establish the existence of solutions that live in the positive orthant. To this end, we borrow some recent results by Bihan, Dickenstein, and Giaroli (2020) from real algebraic geometry. After establishing existence, we turn to uniqueness. We are able to show that, for large graphs, the set of solutions shrinks to a singleton. More precisely, we study the following asymptotic regime: first, we partition the network into sets of stochastically equivalent nodes; then, we let the cardinalities of these subsets grow in size.
Our results can be summarized as follows:

Under mild conditions, the degree centrality problem always admits a solution

in the parameter space that is also unique for large graphs.

We propose to recover the solutions of the problem via a multivariate Newton-Raphson algorithm. Given the sensitivity of these class of methods to their initialization, we bound the roots of the system in terms of the system coefficients and randomly sample points within such bounds to provide a valid initialization of the algorithm. Extensive numerical experiments demonstrate the effectiveness of the algorithm in recovering the solutions under different specifications. Also, we show how we can initialize the centrality targets and the community structure so as to reproduce assortative mixing and power-law centrality distributions. Our efforts result in a novel generator that is able produce networks with a billion edges and hundreds of millions of nodes in 30 seconds, while recreating a wide spectrum of network topologies. Our generator is implemented in the R package genet[1].

2.1.2 Related works

Benchmark generators for community detection The first generative model with built-in community structure dates back to the work of Girvan and Newman (2002), who employ the Erdös-Rényi model to produce networks with 128 nodes evenly split between four communities. The seminal Lancichinetti-Fortunato-Radicchi (LFR) generator by Lancichinetti, Fortunato, and Radicchi (2008) is the first to produce networks with heterogenous degrees and community sizes, sampled from power-law distributions. This has later been extended to weighted and directed graphs with overlapping communities (Lancichinetti and Fortunato, 2009). As the extensive rewiring procedure required by LFR generators makes them hard to scale, Hamann et al. (2018) have proposed an external memory algorithm that can produce LFR graphs with more than 10^{10} edges in less than a day with modest hardware. More recent efforts have focused on more interpretable (Kamiński, Prałat, and Théberge, 2021), realistic (Fagnan et al., 2018), or scalable methods (Kolda et al., 2014; Slota et al., 2019). As mentioned in the previous section, this body of literature is predominantly algorithmic and focuses mostly on reproducing degree distributions.

[1] github.com/aagrande/genet.

Poisson factorization Our work builds on the random graph model by Ball, Karrer, and Newman (2011), which is based on Poisson factorization (PF) (Dunson and Herring, 2005). PF and its generalizations have been used in a wide range of applications, including signal processing (Virtanen, Cemgil, and Godsill, 2008; Cemgil, 2008), music tagging (Liang, Paisley, Ellis, et al., 2014), topic modelling (Paisley, Blei, and Jordan, 2014), and recommender systems (Gopalan, Ruiz, et al., 2014; Gopalan, Hofman, and Blei, 2015). In the field of community detection, notable examples include Bayesian non-parametric extensions (Zhou, 2015; Ayed and Caron, 2021) that allow for an unknown number of communities to be estimated from the data

Real algebraic geometry. Under Poisson factorization, both the degree and the relaxed eigencentrality problems result in a system of real multivariate polynomials in the model parameters, which are constrained to be positive. The study of solutions of real polynomial systems falls under the domain of real algebraic geometry (Bochnak, Coste, and Roy, 2013). We employ some recent results by Bihan, Dickenstein, and Giaroli (2020), who provide sufficient sign conditions on the coefficient and exponent matrices of a real polynomial system that guarantee the existence of a positive solution.

Matrix perturbation analysis In the eigencentrality problem, we bound the distance between the sample eigencentrality and its population counterpart using a Davis-Kahan theorem. We use a variant (Yu, Wang, and Samworth, 2015) of the original theorem that depends only on the population eigenvalues. Also, to bound the operator norm between the sample and the expected adjacency matrix, we extend the spectral bound by Lei and Rinaldo (2015) to our Poisson multiedge setting.

2.1.3 Setting and notation

Throughout this chapter we assume that the number of nodes N and the number of overlapping communities K of the network we want to generate are given.

Node types To simplify the problem complexity while preserving heterogeneity within the network, we partition the nodes into L types. Nodes within the same type share the same propensities; in other words, they are stochastically equivalent. More specifically, if we let the map $\sigma : [N] \to [L]$ return the type of a node, we have:

$$\sigma(i) = \sigma(j) \implies \lambda_{ik} = \lambda_{jk}, \quad k \in [K]. \tag{2.3}$$

We assume that each node type-ℓ has cardinality $n_\ell > 1$. Without loss of generality, we assume that nodes 1 to n_1 belong to the first node type, nodes $n_1 + 1$ to $n_1 + n_2$ to the second type, and so forth. Clearly, $\sum_{\ell \in [L]} n_\ell = N$.

Membership assignments Each node type can belong to one or multiple communities. We say that node type ℓ belongs to community k if its membership assignment $z_{\ell,k}$ is positive. To allow for varying node preferences over communities, we allow the membership assignments to take any value in $[0, 1]$ and require they have unitary norm, $\sum_{k \in [K]} z_{\ell,k}^2 = 1$ for each ℓ. We encode this information in a membership matrix $Z = (z_{\ell k})_{\ell \in [L], k \in [K]}$ and define the type interaction matrix $H = ZZ^\top \in \mathbb{R}^{L \times L}$. Note that $H_{\ell,\ell'} > 0$ if and only if the node types ℓ and ℓ' share at least one community. For our theoretical analysis, we will assume that the type interaction matrix is given. In Section 2.3, we discuss how it can be generated so as to enforce assortative mixing.

Degree correction terms We relate the propensities to membership assignments as follows:

$$\lambda_{ik} = \theta_{\sigma(i)} z_{\sigma(i)k}, \quad i \in [N], k \in [K], \tag{2.4}$$

where $\theta_{\sigma(i)}$ is a positive degree correction term for node type $\sigma(i)$. Eq. 2.4 verifies Eq. (2.3). We split propensities into two components: the membership assignments allocate the normalized preferences of node types over communities; the degree correction terms allow for degree heterogeneity among types. While the type interaction matrix is assumed to be given, $\boldsymbol{\theta}$ is assumed to be

unknown. Recovering the degree correction terms $\boldsymbol{\theta} = (\theta_\ell)_{\ell\in[L]}$ will be the object of interest in the next sections.

Recalling that $H = ZZ^\top$, it follows from (2.4) that:

$$\mathbb{E}A_{ij} = \sum_{k\in[K]} \lambda_{ik}\lambda_{jk} = \theta_{\sigma(i)}\theta_{\sigma(j)} \sum_{k\in[K]} z_{\sigma(i)k} z_{\sigma(j)k} = \theta_{\sigma(i)}\theta_{\sigma(j)} H_{\sigma(i)\sigma(j)}, \quad i \neq j. \tag{2.5}$$

Hence, two nodes i and j share edges if and only if their types interact (i.e. $H_{\sigma(i)\sigma(j)} > 0$), or equivalently, if and only if their types have at least one community in common.

Finally, in contrast to the original model by Ball, Karrer, and Newman (2011), we do not allow self-loops, and set $\mathbb{E}A_{ii} = 0$. The expected adjacency matrix is univocally defined by σ (or equivalently, by the type cardinalities $\boldsymbol{n} = (n_\ell)_{\ell\in[L]}$), Z (or equivalently, H), and $\boldsymbol{\theta}$. We then denote it as $\mathbb{E}A(\boldsymbol{\theta}; H, \boldsymbol{n})$.

2.1.4 Outline

In Section 2.2, we state the degree centrality problem and show the existence and uniqueness of its solutions. In Section 2.3, we present our generator and show how to initialize the affiliation graph and the centrality targets so as to reproduce a wide spectrum of network topologies. Finally, in Section 2.4, we investigate the efficacy of our generator with extensive numerical experiments. All proofs are deferred to the Appendix.

2.2 The degree centrality problem

The degree centrality problem amounts to recovering the degree correction terms that secure (in expectation) a target degree sequence.

Definition 2.2.1 (Degree centrality problem)**.** *Let $\hat{\boldsymbol{d}} \in \mathbb{R}_+^L$, $\hat{\boldsymbol{d}}_\sigma = (\hat{d}_{\sigma(i)})_{i\in[N]}$, and let $\mathbf{1}_L$ denote a vector of ones of size L. We define the degree centrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ as the problem of*

finding a solution $\boldsymbol{\theta} \in \mathbb{R}_+^L$ *of the following system:*

$$\mathbb{E}A(\boldsymbol{\theta}; H, \boldsymbol{n})\mathbf{1}_L = \hat{\boldsymbol{d}}_\sigma. \tag{2.6}$$

The system defining the degree centrality problem arises from equating the expected degree of each node to its corresponding target, which depends on the node's type. Using the representation in Eq. (2.5), the system in Eq. (2.6) reduces to:

$$(n_\ell - 1)H_{\ell\ell}\theta_\ell^2 + \left(\sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'}\theta_{\ell'}\right)\theta_\ell - \hat{d}_\ell = 0, \quad \ell \in [L]. \tag{2.7}$$

This is a system of L sparse multivariate polynomials of degree 2 in the L variables $\boldsymbol{\theta}$. The first term in Eq. (2.7) results from the fact that a type-ℓ node can connect with all n_ℓ type-ℓ nodes but itself.

2.2.1 Existence of a solution

From a theoretical standpoint, the first question of interest is whether Eq. (2.7) admits a solution in the parameter space $\mathbb{R}_+^L$. For a particular choice of $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$, this can be assessed with effective quantifier elimination (Basu, Pollack, and Roy, 2006) via, for example, the algebra software `SINGULAR` (Decker et al., 2021). Our goal, however, is to study existence for the family of systems arising from any choice of the parameters $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$. To this end, we employ some recent results by Bihan, Dickenstein, and Giaroli (2020) that provide sufficient conditions on the coefficients of the system. Before presenting their main idea, we need to introduce yet another representation of the system in Eq. (2.7).

Fix an exponent set $\mathcal{E} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_P\} \subseteq \mathbb{R}^L$ of cardinality P and a coefficient matrix $C \in \mathbb{R}^{L \times P}$. The resulting system of L multivariate polynomials into the L variables in $\boldsymbol{\theta}$ reads:

$$\sum_{p \in [P]} c_{\ell p} \boldsymbol{\theta}^{\boldsymbol{e}_p} = 0, \quad \ell \in [L], \tag{2.8}$$

where $\boldsymbol{\theta}^{\boldsymbol{e}_p} \stackrel{\text{def}}{=} \prod_{\ell \in [L]} \theta_\ell^{e_{p\ell}}$. Our system in Eq. (2.7) is a special case of Eq. (2.8) with $C = C(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ and $P \leq L(L+1)/2 + 1$.

Example 2.2.1. *To clarify the notation, we consider a toy setting with $K = 3$ communities, $L = 3$ node types, and $\boldsymbol{n} = (2, 2, 3)^\top$. We set the goal $\hat{\boldsymbol{d}} = (1.5, 2.5, 5)^\top$. Also, we assume that the first two node types belong exclusively to one community, while the third to all three, as encoded in the following membership assignment matrix:*

$$Z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \end{pmatrix}. \tag{2.9}$$

The degree centrality problem amounts to finding a solution to the following system:

$$\theta_1^2 + \frac{3}{\sqrt{6}}\theta_1\theta_3 - 1.5 = 0, \tag{2.10}$$

$$\theta_2^2 + \sqrt{6}\theta_2\theta_3 - 2.5 = 0, \tag{2.11}$$

$$\frac{2}{\sqrt{6}}\theta_1\theta_3 + \frac{4}{\sqrt{6}}\theta_2\theta_3 + 2\theta_3^2 - 5 = 0. \tag{2.12}$$

This system of three polynomials in $(\theta_1, \theta_2, \theta_3)$ can be formulated as in Eq. (2.8) *by choosing the exponent set:*

$$\mathcal{E} = \{(2,0,0), (1,1,0), (1,0,1), (0,2,0), (0,1,1), (0,0,2), (0,0,0)\}, \tag{2.13}$$

and the following coefficient matrix:

$$C = \begin{pmatrix} 1 & 0 & 3/\sqrt{6} & 0 & 0 & 0 & -1.5 \\ 0 & 0 & 0 & 1 & \sqrt{6} & 0 & -2.5 \\ 0 & 0 & 2/\sqrt{6} & 0 & 4/\sqrt{6} & 2 & -5 \end{pmatrix}. \tag{2.14}$$

The main idea in Bihan, Dickenstein, and Giaroli (2020) is to study a dual system to that in

Eq. (2.8) whose solutions are in a one-to-one correspondence with the solutions of our original system, if any. While the latter live in the positive orthant $\mathbb{R}_+^L$, the solutions of the dual system are restrained in a convex and bounded polytope, and their existence can then be investigated via degree theory. The dual system is obtained via the Gale dual matrices[2] of the coefficient matrix and of a matrix based on the exponents in $\mathcal{E}$. This approach results in a set of sign conditions on the coefficient and exponent matrix of the original system that are sufficient for the existence of at least one positive solution.

To show that our original system in Eq. (2.7) admits a positive solution, we then reformulate it under the representation in Eq. (2.8) and derive a closed form of the Gale duals of the resulting coefficient and exponent matrices. This allows us to check the conditions by Bihan, Dickenstein, and Giaroli (2020) for any choice of the system parameters. In Section 2.5, we introduce the full framework in Bihan, Dickenstein, and Giaroli (2020) and present the proof of the following result.

Theorem 2.2.1 (Existence of a solution of the degree centrality problem)**.** *Let Z be a membership matrix that results in a type interaction matrix $H = ZZ^\top$ with strictly positive entries. Let $\boldsymbol{n} = (n_\ell)_{\ell\in[L]}$ be a set of node type cardinalities that are all strictly greater than 1. Then the degree centrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ admits a solution for any target $\hat{\boldsymbol{d}} \in \mathbb{R}_+^L$.*

Theorem 2.2.1 provides a theoretical validation of the flexibility of the model, in that it guarantees the existence of a solution for any arbitrary centrality target. In practice, the remarkable implication of the result is that we can reproduce any target degree centrality sequence, provided that we can recover the degree correction terms that support it.

The technical assumption of strict positivity of the entries of the type interaction matrix H may seem restrictive, as it forces any pair of node types to connect in expectation. However, since the assumption involves the sign of the entries of H rather than their magnitude, we can approximate any overlapping community structure by a perturbed version of the resulting type interaction matrix. To see this, consider the setting of Example 2.2.1 in which the first two node types do not have

[2]We recall that a Gale dual matrix of a matrix $A \in \mathbb{R}^{m\times n}$ with maximal rank m is any matrix $B \in \mathbb{R}^{n\times(n-m)}$ of maximal rank whose columns form a basis of the kernel of A.

any communities in common, and thus $H_{12} = H_{21} = 0$. Theorem 2.2.1 still applies if we introduce an arbitrarily small perturbation $\varepsilon > 0$ in the type interaction matrix H yielding:

$$\tilde{H} = \begin{pmatrix} 1 & \varepsilon & 1/\sqrt{6} \\ \varepsilon & 1 & 2/\sqrt{6} \\ 1/\sqrt{6} & 2/\sqrt{6} & 1 \end{pmatrix}. \tag{2.15}$$

2.2.2 Asymptotic uniqueness

Now that we have shown that the degree centrality problem always admits a solution, we may wonder whether such solution is unique. It is important to note that, in order to generate networks with a prescribed degree sequence and overlapping community structure, it suffices to recover at least one solution in the parameter space. In this sense, uniqueness is somewhat marginal to our main goal, especially if compared to existence. However, if uniqueness holds, then the equations behind the degree centrality problem fully specify the model, in that each desired structure corresponds to one and only one point in the parameter space.

We are able to show that, if we let the network size grow while keeping the proportion of nodes for each type constant, the set of solutions of the degree centrality problem will converge to a singleton. We refer to this property as asymptotic uniqueness. Also, we can show that the limit is given by any solution of the relaxation of the degree centrality problem that allows for self-loops, and that convergence is uniform. This further implies that the problem with self-loops admits one and only one solution.

We use the simplified setting of Example 2.2.1 to show the intuition behind our result, which is stated in Theorem 2.2.2.

Example 2.2.2. *Recall the setting from Example 2.2.1 with $L = 3$ node types and $K = 3$ communities. We fix some initial cardinalities $\boldsymbol{n}(1)$ and a centrality target $\hat{\boldsymbol{d}}(1)$. Assume that we multiply the size of the network by some factor $t \in \mathbb{N}$ uniformly over the node types, so that the resulting*

cardinalities are $\boldsymbol{n}(t) = t\boldsymbol{n}(1)$ *and the centrality target is* $\hat{\boldsymbol{d}}(t) = t\hat{\boldsymbol{d}}(1)$*.We want to find the degree correction terms* $(\theta_1(t), \theta_2(t), \theta_3(t))$ *that solve:*

$$(n_1(t) - 1)\theta_1^2(t) + \frac{1}{\sqrt{6}} n_3(t)\theta_1(t)\theta_3(t) - \hat{d}_1(t) = 0, \tag{2.16}$$

$$(n_2(t) - 1)\theta_2^2(t) + \frac{2}{\sqrt{6}} n_3(t)\theta_2(t)\theta_3(t) - \hat{d}_2(t) = 0, \tag{2.17}$$

$$\frac{1}{\sqrt{6}} n_1(t)\theta_1(t)\theta_3(t) + \frac{2}{\sqrt{6}} n_2(t)\theta_2(t)\theta_3(t) + (n_3(t) - 1)\theta_3^2(t) - \hat{d}_3(t) = 0. \tag{2.18}$$

We make two simple remarks. First, the coefficients of the new system in Eq.s (2.16)–(2.18) *are multiples of those of the initial system* ($t = 1$) *except for the coefficients that correspond to the monomials in* $(\theta_\ell^2)_{\ell\in[3]}$*, which equal* $tn_\ell(1) - 1$*. As a result, the solutions of the initial and new system differ. Secondly, for the equations – and thus the solutions – of the system not to change as we multiply in size the cardinalities, we should allow self-loops. However, if we divide both sides of the system of Eq.s* (2.16)–(2.18) *by* t*, the coefficients of the monomials in* $(\theta_\ell^2)_{\ell\in[3]}$ *would differ from those of the problem with self-loops by* $1/t$*. Hence, as* $t \to \infty$*, we expect the set of solutions of the degree centrality problem to converge to that of the problem with self-loops.*

We defer the formal argument behind the remarks in Example 2.2.2 to the next chapter.

Theorem 2.2.2 (Asymptotic uniqueness for the degree centrality problem)**.** *Let* H *and* $\boldsymbol{n}$ *satisfy the assumptions of Theorem 2.2.1. For* $t \in \mathbb{N}$*, let* $\mathcal{S}(t, H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ *denote the set of solutions of the degree centrality problem for* $(H, t\boldsymbol{n}, t\hat{\boldsymbol{d}})$*. The solutions of the degree centrality problem for* $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ *are asymptotically unique, in the sense that there exists* $\boldsymbol{\theta}^*$ *such that for any* $\varepsilon > 0$ *there exists* $T_\varepsilon > 0$ *such that:*

$$\sup_{\boldsymbol{\theta}\in\mathcal{S}(t;H,\boldsymbol{n},\hat{\boldsymbol{d}})} ||\boldsymbol{\theta} - \boldsymbol{\theta}^*|| < \varepsilon \quad \textit{for all} \quad t \geq T_\varepsilon. \tag{2.19}$$

Moreover, $\boldsymbol{\theta}^*$ *is the solution of the degree centrality problem for* $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ *with self-loops.*

2.3 Network generation via the multivariate Newton-Raphson method

Under our framework, generating networks with target expected centralities involves two steps. First, we need to recover a set of parameters that secures such targets; next, given the parameters, we independently sample edges from the underlying Poisson random graph model.

In the previous sections we have shown that, for both the degree centrality and the eigencentrality, the first task amounts to solve a system of multivariate polynomials of degree 2 in the degree correction terms. More specifically, if we let $f : \mathbb{R}_+^L \to \mathbb{R}_+^L$ denote the map encoding the polynomial system, we want to find $\boldsymbol{\theta} \in \mathbb{R}_L^+$ such that:

$$f(\boldsymbol{\theta}) = \mathbf{0}. \tag{2.20}$$

Our theoretical results guarantee that Eq. (2.20) always admits a solution (Theorem 2.2.1) and that, for large graphs, such solution is unique (Theorem 2.2.2). We can recover the degree correction terms with a multivariate Newton-Raphson (MNR) algorithm. The MNR update rule reads:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - J^{-1}\left(\boldsymbol{\theta}^{(t)}\right) f\left(\boldsymbol{\theta}^{(t)}\right), \tag{2.21}$$

where $J : \mathbb{R}^L \to \mathbb{R}^{L\times L}$ denotes the Jacobian of f. The updates can be stopped once the algorithm has converged in a neighborhood of the solution, or equivalently if $||\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}|| + ||f(\boldsymbol{\theta}^{(t+1)})|| < \varepsilon$. The effectiveness of the algorithm to recover a solution of Eq. (2.20) is known to depend on the validity of the initial guess $\boldsymbol{\theta}^{(0)}$. If we start close enough to a solution, the MNR algorithm provides quadratic convergence (Ortega, 1968). Conversely, if initialized too far from a root, the update rule could result in a large step that makes the estimate worse. To this end, we bound the solutions of our two problems of interest in terms of their coefficients. These bounds allow us to provide a valid initialization to the MNR algorithm.

Theorem 2.3.1. *Let $\boldsymbol{\theta} \in \mathbb{R}_+^L$ denote the solution of the degree centrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$. For each node type $\ell \in [L]$, there exist some constants $\underline{\theta}_\ell, \bar{\theta}_\ell > 0$ that depend on $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ such that*

$\underline{\theta}_\ell \leq \theta_\ell \leq \bar{\theta}_\ell$.

After recovering the solution $\boldsymbol{\theta}^*$ to Eq. (2.20), we generate networks from the model under the propensities induced by $\boldsymbol{\theta}^*$. This step can be executed efficiently under Poisson factorization. Rather than generating each entry of the adjacency matrix A, we can first sample the total number of multiedges M and then sample independently their $2M$ endpoints proportionally to the joint propensities. Also, since edges are sampled independently, the edge generation step can be easily scaled up via parallel computing.

A general version of our generator is stated below.

Algorithm 2.1 Graph generator for an arbitrary community structure and centrality target

Input: type interaction matrix $H \in \mathbb{R}^{L\times L}$, cardinalities $\boldsymbol{n} \in \mathbb{N}^L$, centrality target $\hat{\boldsymbol{g}} \in \mathbb{R}^L$

Output: two-column matrix E listing edges

Initialize the degree correction terms $\boldsymbol{\theta}^{(0)}$ by sampling each term independently and uniformly over the intervals generated by the bounds in Theorem 2.3.1

Solve (2.20) for $\boldsymbol{\theta}$ with MNR initialized at $\boldsymbol{\theta}^{(0)}$

for (ℓ, ℓ') such that $\ell \leq \ell', H_{\ell\ell'} > 0$ **do**

    if $\ell = \ell'$ **then**

        $\rho \leftarrow (n_\ell - 1) n_\ell H_{\ell\ell} \theta_\ell^2$

    else

        $\rho \leftarrow n_\ell n_{\ell'} H_{\ell\ell'} \theta_\ell \theta_{\ell'}$

    Sample $m_{\ell,\ell'} \sim \text{Poisson}(\rho)$

Initialize 2-col. int. matrix E with $\sum_{\ell,\ell'} m_{\ell,\ell'}$ rows

for (ℓ, ℓ') such that $\ell \leq \ell', H_{\ell\ell'} > 0$ **do**

    Randomly sample with replacement $m_{\ell,\ell'}$ node pairs from $\{(i, j) : i \neq j,\ i \text{ is type-}\ell,\ j \text{ is type-}\ell'\}$ and save them in E

Return E

2.3.1 The MNR algorithm

We derive the multivariate Newton-Raphson (MNR) algorithm for the degree centrality problem in detail. The problem reduces to a system of multivariate polynomials of degree 2 in L variables and L equations, whose coefficients depend on the type interaction matrix H, the type cardinalities $\boldsymbol{n}$, and the centrality targets. We first state the system of equations, derive the Jacobian, and then state the MNR algorithm.

The system behind the degree centrality problem for $(H, \boldsymbol{n}, \boldsymbol{d})$ reads:

$$(n_\ell - 1)H_{\ell\ell}\theta_\ell^2 + \left(\sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'}\theta_{\ell'}\right)\theta_\ell - d_\ell = 0, \qquad \ell \in [L]. \tag{2.22}$$

We encode the coefficients of the above system in the matrix of total interactions $T \in \mathbb{R}^{L\times L}$. We set:

$$T_{\ell\ell'} \doteq \begin{cases} 2(n_\ell - 1)H_{\ell\ell} & \ell' = \ell \\ n_{\ell'} H_{\ell\ell'} & \ell' \neq \ell \end{cases} \tag{2.23}$$

We note that T is a sparse, non-Hermitian matrix that has as many entries as H.

Solving the system in Eq.(2.22) amounts to finding the roots of a map $f : \mathbb{R}_+^{L\times L} \to \mathbb{R}_+^L$ such that:

$$f_\ell(\boldsymbol{\theta}) = \frac{T_{\ell\ell}}{2}\theta_\ell^2 + \left(\sum_{\ell' \neq \ell} T_{\ell\ell'}\theta_{\ell'}\right)\theta_\ell - d_\ell, \qquad \ell \in [L], \tag{2.24}$$

where $\boldsymbol{d}$ denotes an arbitrary degree sequence. The Jacobian of Eq. (2.24) is given by:

$$J_{\ell\ell'}(\boldsymbol{\theta}) = \frac{\partial f_\ell}{\partial \theta_{\ell'}}(\boldsymbol{\theta}) = \begin{cases} \sum_{\ell' \in [L]} T_{\ell\ell'}\theta_{\ell'} & \ell' = \ell \\ T_{\ell\ell'}\theta_\ell & \ell' : T_{\ell\ell'} > 0 \cdot \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

The MNR algorithm requires computing the Jacobian at each iteration. However, because T depends only on the system coefficients, this matrix can be precomputed and stored in memory before running the MNR updates. After computing the Jacobian, the MNR updates reduce to solving a linear system of equations at each iteration. More specifically, at the t-th iteration, we have:

$$J(\boldsymbol{\theta})\left(\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\right) = -f\left(\boldsymbol{\theta}^{(t)}\right), \tag{2.26}$$

where f and J are given by Eq.s (2.24) and (2.25), respectively, and $\boldsymbol{\theta}^{(t)}$ denotes the current guess of a root of f. We want to solve Eq. (2.26) for $\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}$. Similarly to the total interaction matrix T, we note that J is a square, non-Hermitian matrix with as many entries as H. We can then solve (2.26) via a LU factorization. To this end, we use the sparse LU-based solver provided in the C++ library Eigen (Guennebaud, Jacob, et al., 2010), which we interface to R via the R package RcppEigen (Bates and Eddelbuettel, 2013). This efficient solver analyzes the sparsity pattern of the coefficient matrix and reorders its nonzero entries so as to minimize the number of fill-ins. Because the sparsity pattern of the Jacobian in Eq. (2.25) does not change across the MNR iterations, we can execute this step only once before running the updates.

The resulting MNR algorithm is stated below.

Algorithm 2.2 Multivariate Newton Raphson for the degree centrality problem

Input: intial root $\boldsymbol{\theta}^{(0)}$, sparse matrix of total interactions $T \in \mathbb{R}^{L \times L}$, centrality target $\boldsymbol{d}$, error tolerance ε

Output: root $\boldsymbol{\theta}$

Compute J from T and $\boldsymbol{\theta}^{(0)}$ as in (2.25)
Analyze pattern of J
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
not_converged $\leftarrow true$
while not_converged **do**
    $\boldsymbol{b} \leftarrow f(\boldsymbol{\theta}, \boldsymbol{d})$ as in (2.24)
    Compute J from T and $\boldsymbol{\theta}$ as in (2.25)
    Factorize J and solve $J\Delta\boldsymbol{\theta} = \boldsymbol{b}$ for $\Delta\boldsymbol{\theta}$
    if $||\boldsymbol{b}|| + ||\Delta\boldsymbol{\theta}|| < \varepsilon$ **then**
        not_converged $\leftarrow false$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$
Return $\boldsymbol{\theta}$

2.3.2 Initializing the community structure and centrality targets

To illustrate the flexibility of our generator, we show how to initialize the centrality targets and the community structure in order to reproduce an arbitrary topology of interest. For example, one may want to investigate the dynamics of an epidemic process on a modular network of contacts with a power-law degree distribution and with high degree assortativity – two properties that are known to occur in social networks (Newman and Park, 2003; Barabási and Albert, 1999). With the general aim of capturing a wide spectrum of network topologies, we operate at two levels: on a global scale, we reproduce an arbitrary distribution for the degree centrality; on a local scale, we control the assortativity of the network – how likely nodes are to connect based on how similar they are. In particular, we propose a method to initialize the cardinalities and centrality targets so

as to reproduce an arbitrary centrality distribution, and then we discuss how to build a community structure that supports assortative mixing.

Initializing centrality targets and cardinalities Assume that we want to generate a network with an arbitrary degree distribution $\mathcal{P}$, with probability mass function p and finite support $D \subseteq \mathbb{N}$. The empirical degree distribution of the network will depend on both the centrality target $\hat{\boldsymbol{d}}$ and the type cardinalities $\boldsymbol{n}$. Hence, we need to choose $\hat{\boldsymbol{d}}$ and $\boldsymbol{n}$ so that the empirical degree distribution is as close as possible to p. To this end, we set as centrality target the vector of $(L+1)$-quantiles of $\mathcal{P}$, and then adjust the cardinalities so that the empirical distribution complies with p. More specifically, for all $\ell \in [L]$, we let $\hat{d}_\ell$ be the quantile of $\mathcal{P}$ corresponding to the probability $\ell/(L+1)$, and then set:

$$n_\ell \sim \text{nint}\left(2\frac{p(\hat{d}_\ell)}{\min_{\ell' \in [L]} p(\hat{d}_{\ell'})}\right), \tag{2.27}$$

where $\text{nint}(\cdot)$ denotes the nearest integer function. In Eq. (2.27), we adjust the cardinalities while enforcing the constraint that they must be integers greater than 1. Clearly, the greater the number of types, the more closely the empirical distribution will resemble $\mathcal{P}$. In practice, since $\mathcal{P}$ is discrete, some centrality targets may be identical, especially if the number of types is large. Under the cardinality scaling in Eq. (2.27), identical targets introduce bias. To avoid this, if any two consecutive targets $\hat{d}_\ell$ and $\hat{d}_{\ell+1}$ are such that $\hat{d}_\ell \geq \hat{d}_{\ell+1}$, we set $\hat{d}_{\ell+1} = \hat{d}_\ell + 1$; we repeat this correction iteratively until all targets are distinct.

One limitation of this approach is that, while it can be applied to any arbitrary distribution $\mathcal{P}$, highly concentrated distributions will result in massive network sizes. This is not the case, however, for fat-tailed distributions, such as power-law distributions, which are known to capture the empirical distribution of centralities in real-world networks.

Initializing the community structure via assortative mixing We can control the network assortativity via the type interaction matrix H. While the entries of H are positive real numbers, we

can model their signs as Bernoulli random variables with expectation proportional to the similarities of types, as measured by a similarity function of our choice. In this chapter, we focus on the degree assortativity and show how we can control the latter via the survival function of a Beta distribution. More specifically, denoting by $\hat{d}_\ell$ the target degree for type ℓ, we let:

$$\operatorname{sign}(H_{\ell,\ell'}) \sim \text{Bernoulli}\left(\bar{I}_{\alpha,\beta}\left(\frac{|\hat{d}_\ell - \hat{d}_{\ell'}|}{\max_\ell \hat{d}_\ell}\right)\right), \qquad \ell \neq \ell', \tag{2.28}$$

where $\bar{I}_{\alpha,\beta}(\cdot)$ denotes the survival function of a Beta random variable with shape parameters α and β. Beta survival functions are a natural choice for assigning degree assortativity scores, as they form a rich family of parametric curves that are bounded and strictly decreasing in $[0, 1]$.

After generating the type interactions via Eq. (2.28), we need to build some community assignments $Z_{\ell k}$'s that are compatible with $\operatorname{sign}(H)$. Note that if two types ℓ and ℓ' share edges (i.e. $\operatorname{sign}(H_{\ell\ell'}) = 1$), they have at least one community in common supporting their interaction. Any other node type ℓ'' belonging to that community must share edges with both type-ℓ and type-ℓ' nodes. Hence, if we let G_H denote the undirected graph of type interactions with adjacency matrix $\operatorname{sign}(H)$, we can assign to each maximal clique in G_H a separate community that supports the type interactions within that clique. This approach will yield the minimum number of communities needed to enforce the type compatibilities generated in Eq. (2.28). The resulting number of communities K will be given by the number of maximal cliques plus all the single edges in G_H that belong to no clique. This procedure will determine whether or not a type belongs to a community, i.e. the sign of the community assignments; their weight can be then assigned arbitrarily.

We may wonder whether Eq. (2.28) could be used to generate disassortative networks, for example by adopting an increasing score function (such as the Beta cumulative distribution function) for disassortativity scores. However, because nodes of the same type are connected with high probability ($H_{\ell\ell} = 1$ by definition), the networks produced by our generator are likely to display some level of assortativity. For this reason, our generator is not suitable for applications that are known to present disassortative mixing, such as protein-to-protein networks (Maslov and Sneppen, 2002).

2.4 Numerical experiments

We perform extensive numerical experiments[3] to illustrate the ability of our generator to capture target degree centrality distributions, reproduce varying levels of degree assortativity, and generate massive networks.

Power-law degree distributions Following the procedure discussed in Section 2.3.2, we initialize the cardinalities and the degree centrality target so as to reproduce the right tail of a power-law degree distribution. Under this distribution, the probability that a node has degree d is proportional to $d^{-\alpha}$. In our experiments, we set the minimum degree $d_{\min} = 3$, the maximum degree $d_{\max} = 500$, and $\alpha = 1.94$, resulting in a mean degree of approximately 15. For simplicity, we assume that each node type is assigned with equal weights to 15 out of 100 total communities. We also ensure that no types have identical assignments, all communities have at least one type, and the resulting graph of type interactions is connected.

Figure 2.1 compares the right tail of the target power-law distribution with that of the empirical degree distribution of 1,000 networks produced by our generator, for different number of types. The empirical degree distributions of the synthetic networks comply with the target, illustrating the efficacy of our generator. Under the initialization in Eq. (2.27), as the number of types increases, so does the number of distinct q-quantiles that we use to pinpoint the target distribution. As a result, the empirical distribution more closely resembles the underlying power-law distribution.

[3]The R package is available at github.com/aagrande/genet. All experiments are single-threaded and run in RAM on a Intel® Core™ i7-1068NG7 CPU @2.3GHz.

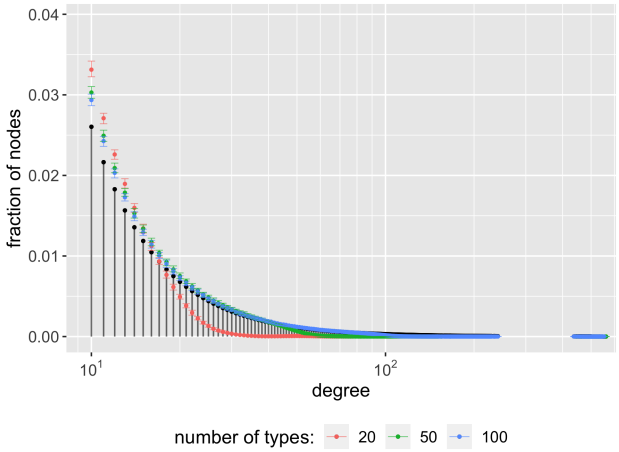


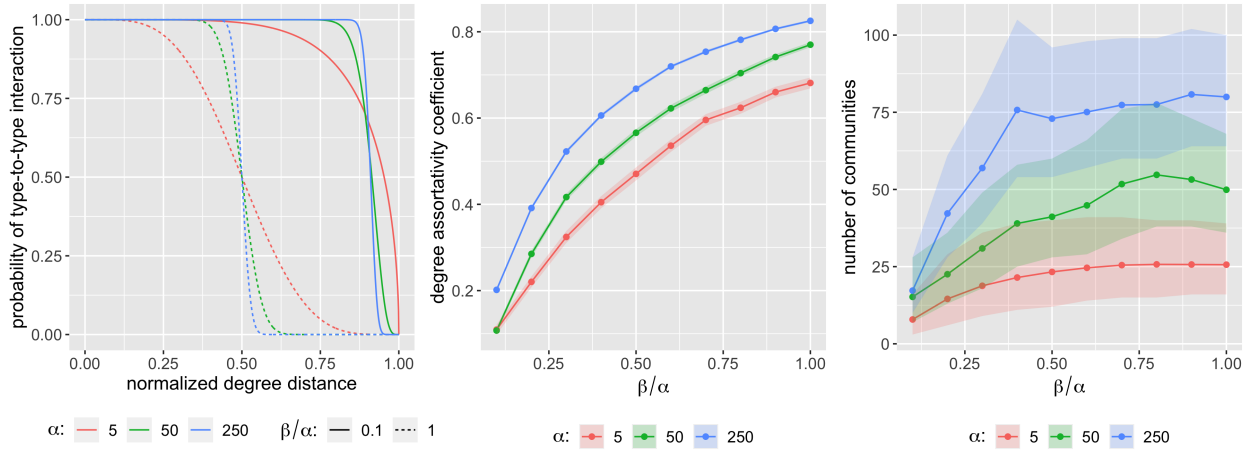
Figure 2.1: Monte Carlo estimates and 95% confidence intervals of the degree distribution (1,000 networks per point) for different node types. The black lines represent the probability mass function of the target power-law distribution ($d_{\min} = 3, d_{\max} = 500, \alpha = 1.938, \bar{d} \approx 15$).

Assortative mixing We demonstrate the ability of our generator to produce networks that vary in terms of degree assortativity. Following the initialization recipe presented in Eq. (2.27), we investigate how the assortative score function determines the mean degree assortativity coefficient (Newman, 2002) and the number of communities.

We consider three different network sizes, as indexed by the number of types $L \in \{20, 50, 100\}$. The degree centrality targets are drawn from a power-law degree distribution ($d_{\min} = 3, d_{\max} = 500$, and $\alpha = 1.938$ so that $\bar{d} \approx 15$), which results in networks with $\sim 10^3$ ($L = 20$), $\sim 10^4$ ($L = 50$), and $\sim 10^5$ ($L = 100$) nodes. In order to generate the type interaction matrices via (2.28), we must choose the Beta shape parameters of the assortative score function. We proceed as follows. First, we fix a different value of α for each network size and let β range from $\alpha/10$ to α. Next, we investigate via numerical experiments how to choose α so as to (a) enforce different levels of degree assortativity while (b) keeping the number of communities under control.

Figure 2.2 illustrates this two-step procedure. Figure 2.2(a) presents some assortativity score functions based on the Beta survival functions. When $\alpha > 1$ and $\beta > 1$, the ratio β/α determines the location of the inflection point of the Beta survival function. The larger this ratio, the closer to

the origin the inflection point, and thus the less likely an interaction between types with dissimilar degrees is. We can then think of β/α as a rough measure of the level of degree assortativity that we want to enforce. This is reflected in the second plot, Figure 2.2(b), which presents Monte Carlo estimates (10^2 type interaction matrices per point, 10^3 networks per interaction matrix) of the degree assortativity coefficient resulting from different parameter ratios. As we vary the ratio β/α, our generator is able to produce a wide spectrum of degree assortative topologies for each network size, ranging from mildly assortative to highly assortative networks. Finally, for a given ratio β/α, the shape parameter α influences the steepness of the survival function. As the number of types increases, so does the number of potential interactions; a steeper survival function will result in a sparser type interaction matrix, thus limiting the number of communities needed to support a given degree assortativity regime. This is shown in Figure 2.2(c), which illustrates the ability of our generator to produce a reasonable number of communities for each network size.



(a) (b) (c)

Figure 2.2: (a) Beta survival functions for different combinations of shape parameters. (b) Monte Carlo estimates (10^3 networks $\times$ 10^3 type interaction matrices per point) and 95% confidence intervals of the mean degree assortativity coefficient for three network sizes ($L \in \{20, 50, 100\}$ resulting in 10^3–10^5 nodes) under different parametric regimes. (c) Monte Carlo estimates (10^4 type interaction matrices per point) and 95% confidence intervals of the number of communities for three network sizes resulting from different parametric regimes. The communities are recovered via the maximum clique procedure discussed in Section 2.3.2.

Runtimes We investigate the performance of our generator on massive networks with sizes ranging from 10^6 to 10^8 nodes. We choose an arbitrary degree centrality target so that the average degree is 20 and, as a result, the expected number of edges ranges from 10^7 to 10^9. We measure the runtimes of our generator as we vary the number of types and the sparsity of the type interaction matrix, which can be seen as a proxy for the number of communities. Because types interact through communities, the larger the number of active communities, the denser the interactions between types.

As shown in Figure 2.3, our generator produces networks with 1 billion edges and a hundred million nodes in less than 30 seconds. This is orders of magnitude faster than the other benchmark generators in the literature. For example, the recent and scalable Artificial Benchmark for Community Detection (ABCD; Kamiński, Prałat, and Théberge 2021) generates networks with 10^7 nodes and average degree 25 in 1.2 to 3.8 minutes. Kamiński, Prałat, and Théberge (2021) also show that an efficient implementation of the LFR generator from the NetworKit package (Staudt, Sazonovs, and Meyerhenke, 2016) takes between 15.4 and 20 minutes to generate a network of the same scale. We remark that the runtimes in Figure 2.3 refer to our single-threaded implementation. Because our algorithm samples edges independently, the edge generation step is *pleasingly parallel*. In the most demanding experiments ($N = 10^8$ and $L = 10^3$), this step amounted to at least 95% of the running time, indicating the large potential for scalability of our generator.

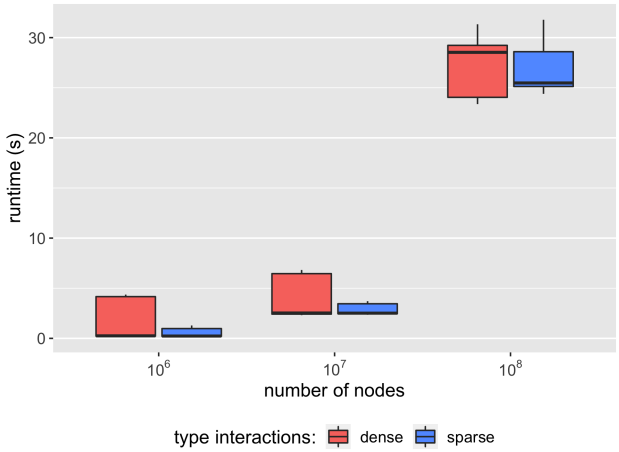


Figure 2.3: Runtimes of our generator for three different network sizes. For each size, we generated 100 networks with $L \in \{10, 100, 1000\}$ node types. Each type is set to interact with $4L \log L$ types in the sparse interaction regime, and with $0.5 \cdot \binom{L}{2}$ types under the dense interaction regime. The runtimes refer to the eigencentrality problem with a balanced eigencentrality target and average degree equal to 20.

2.5 Proofs

Existence of a positive solution Here we prove the existence of a solution for the degree centrality problems. Our results follow from the main result in Bihan, Dickenstein, and Giaroli (2020), a simplified version of which is stated in Theorem 2.5.1. Before proving our results, we recall the framework in Bihan, Dickenstein, and Giaroli (2020) and state the results that are relevant to our application without proof.

Existence of a positive solution of sparse polynomial systems

For a fixed exponent set $\mathcal{E} = \{e_1, \ldots, e_n\} \subset \mathbb{R}^d$ and given matrix of coefficients $C \in \mathbb{R}^{d \times n}$, consider the following system of d multivariate polynomials in d variables $x = (x_1, \ldots, x_d)$:

$$\sum_{j=1}^{n} c_{ij} x^{e_j} = 0, \quad i = 1, \ldots, d, \tag{2.29}$$

where $x^{e_j} \stackrel{\text{def}}{=} \prod_{i=1}^{d} x_i^{e_{ji}}$. We are interested in the existence of solutions that live in the positive orthant $\mathbb{R}^d_{>0}$. To tackle this problem, Bihan, Dickenstein, and Giaroli (2020) introduce a Gale dual system and a convex and bounded polytope such that the solutions of the dual system in this polytope are in a one-to-one correspondence with the real and positive solutions of the initial system Eq. (2.29). As opposed to the initial system, the solutions of the dual problem live in a convex and bounded polytope, and we can then investigate their existence via degree theory.

Note that the number of positive solutions of the initial system is an affine invariant of the exponent set $\mathcal{E}$. We thus group the exponents in $\mathcal{E}$ into the matrix $E \in \mathbb{R}^{(d+1)\times n}$ by combining the vectors $(1, e_i)$ for $i = 1, \ldots, n$ column-wisely, and we let $k = n - d - 1$ denote the codimension of E. The initial system is univocally identified by the pair (E, C). We assume that E and C are of maximal rank, and we let B and D denote some Gale dual matrices[4] of E and C, respectively. We number the columns of B from 1 to k and those of D from 0 to k. Also, we let $C_1, \ldots, C_n$ denote the columns of the coefficient matrix C, and $P_1, \ldots, P_n$ the rows of its Gale dual D. Finally, we let:

$$C_C = \mathbb{R}_{>0} C_1 + \cdots + \mathbb{R}_{>0} C_n, \tag{2.30}$$

denote the positive cone generated by $C_1, \ldots, C_n$; similarly, we let C_P denote the positive cone generated by $P_1, \ldots, P_n$.

The Gale dual system of Eq. (2.29) associated to the Gale dual matrices B and D is given by:

$$\prod_{i=1}^{n} \langle P_i, y \rangle^{b_{ij}} = 1, \quad j = 1, \ldots, k. \tag{2.31}$$

This is a homogenous system with k equations in $k + 1$ variables $y = (y_0, y_1, \ldots, y_k)$. We look for the solutions of Eq. (2.31) over the convex polytope:

$$\Delta_P = \{y \in \mathbb{R}^{k+1} : \langle P_i, y \rangle > 0,\ i = 1, \ldots, n\} \cap \{y \in \mathbb{R}^{k+1} : y_0 = 1\}. \tag{2.32}$$

[4]For a given matrix $M \in \mathbb{R}^{r\times s}$ of maximal rank r, a Gale dual matrix of M is any matrix $G \in \mathbb{R}^{s\times(s-r)}$ of maximal rank whose columns form a basis of the kernel of M.

The following lemma provides necessary and sufficient conditions for Δ_P to be fully dimensional and bounded.

Lemma 2.5.1. *Assume that C has maximal rank, $0 \in C_C$ and let D be a Gale dual matrix of C. Then $(1, 0, \ldots, 0) \in C_{\mathcal{P}}$ if and only if Δ_P has dimension k and is bounded.*

We can now establish the relationship between our initial problem and its Gale dual.

Lemma 2.5.2. *There is a bijection between the positive solutions of the initial system Eq.* (2.29)*) and the solutions of the Gale dual system Eq.* (2.31)*) in Δ_P when $(1, 0, \ldots, 0) \in \overline{C}_P$.*

To study the solutions of Eq. (2.31) in Δ_P, we define the Gale map $g = (g_1, \ldots, g_k) : \mathbb{R}^k \to \mathbb{R}^k$, where:

$$g_j(y) = \prod_{b_{ij}>0} \langle P_i, (1, y)\rangle^{b_{ij}} - \prod_{b_{ij}<0} \langle P_i, (1, y)\rangle^{-b_{ij}}, \quad j = 1, \ldots, k. \tag{2.33}$$

Note that if $y \in \mathbb{R}^k$ is a zero of g such that $(1, y) \in \Delta_P$, then $(1, y)$ is a root of Eq. (2.31). We can assess the existence of such zeros by studying the sign of g on the faces of the polytope Δ_P. For our purposes, it suffices to know that this reduces to some sign conditions on the Gale dual matrices B and D, stated in the next result. Before that, note that the facets of Δ_P are supported on the hyperplanes $\{y \in \mathbb{R}^{k+1} : \langle P_i, y\rangle = 0\}$ for all the vectors P_i that lie on the rays of C_P; for each P_i, we denote by F_i the corresponding facet of Δ_P. Also, we let $I_C \subset \{1, \ldots, n\}$ be the set of indices of the vectors P_i that belong to the boundary of Δ_P. Finally, for any proper face F of Δ_P, we define $L(F) = \{i \in I_C : F \subset F_i\}$ and $\mathcal{F}(\Delta_P) = \{L(F) : F \text{ is a proper face of } \Delta_P\}$.

Theorem 2.5.1. *Consider the system in Eq.*(2.29)*. Assume that E and C have maximal rank, $0 \in C_C$, and Δ_P is a full dimensional bounded polytope. Also, assume that:*

(*i*) *For any $L \in \mathcal{F}(\Delta_P)$, the submatrix of B given by the rows with indices in L has at least one nonnegative or nonpositive column with at least one non-zero entry.*

(*ii*) *For any $i \in I_C$, the following holds:*

- $b_{ij} \cdot d_{ij} \geq 0$ *for* $j = 1, \ldots, k$*;*
- *there exists* $j \in \{1, \ldots, k\}$ *such that* $b_{ij} \cdot d_{ij} > 0$*;*
- *for all* $j \in \{1, \ldots, k\}$*, if* $b_{ij} = 0$ *then* d_{ij}*=0.*

Then the system has at least one real and positive solution.

Existence of a solution for the degree centrality problem

Recall that the degree centrality problem amounts to solving the following system of polynomials in the degree correction terms $\boldsymbol{\theta} \in \mathbb{R}_+^L$:

$$(n_\ell - 1)H_{\ell\ell}\theta_\ell^2 + \sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'}\theta_{\ell'}\theta_\ell - \hat{d}_\ell = 0, \quad \ell \in [L]. \tag{2.34}$$

If H has strictly positive entries, we can show that this problem always admits a (real and positive) solution. Our result is a direct application of Theorem 2.5.1. In what follows, we first introduce the coefficient and exponent matrices resulting from Eq. (2.34), and then derive their Gale duals (Lemma 2.5.3 and 2.5.4). Finally, we show that the degree centrality problems verifies the sign conditions in Theorem 2.5.1.

We can think of H as the adjacency matrix of a weighted undirected graph of type-to-type interactions. The number of monomials that appear in the degree centrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$, which we denote by Q, is given by the number of edges of the type-to-type interaction graph plus one for the constant term, i. e. $Q = \sum_{\ell' \geq \ell} \mathbb{1}_{\{H_{\ell\ell'} > 0\}} + 1$. The assumption of strict positivity of the entries of H implies that the interaction graph is complete; hence, $Q = L(L+1)/2$. We order the monomials lexicographically and define the resulting map $\kappa : \{(\ell, \ell') \in [L] \times [L] : H_{\ell,\ell'} > 0\} \to [Q-1]$ that returns the column index associated to the monomial $\theta_\ell \theta_{\ell'}$ for both the coefficient and exponent

matrices:

$$\kappa(\ell,\ell') = \begin{cases} \kappa(\ell,\ell) + \sum_{\ell' \geq \ell'' > \ell} \mathbb{1}_{\{H_{\ell\ell''}>0\}} & \ell' > \ell, \\ \sum_{\ell''<\ell} \sum_{\ell'''} \mathbb{1}_{\{H_{\ell''\ell'''}>0\}} + 1 & \ell' = \ell, \\ \kappa(\ell',\ell) & \ell' < \ell. \end{cases} \tag{2.35}$$

The coefficient and exponent matrices $C \in \mathbb{R}^{L\times Q}$ and $E \in \mathbb{R}^{(L+1)\times Q}$ that arise from the degree centrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ are:

$$C_{\ell q} = \begin{cases} n_{\ell'} H_{\ell\ell'} & q = \kappa(\ell,\ell') \text{ for some } \ell' \neq \ell, \\ (n_\ell - 1) H_{\ell\ell} & q = \kappa(\ell,\ell), \\ -\hat{d}_\ell & q = Q, \end{cases} \tag{2.36}$$

and:

$$E_{\ell+1,q} = \begin{cases} 1 & \ell = 0, \\ 1 & q = \kappa(\ell,\ell') \text{ for some } \ell' \neq \ell, \\ 2 & q = \kappa(\ell,\ell), \\ 0 & \text{otherwise.} \end{cases} \tag{2.37}$$

Lemma 2.5.3 (Gale dual to the exponent matrix of the degree centrality problem)**.** *Define the matrix $B \in \mathbb{R}^{Q\times(Q-L-1)}$ such that:*

$$B = \begin{pmatrix} \widetilde{B} \\ \widetilde{I}_B \end{pmatrix}, \tag{2.38}$$

where $\widetilde{B} \in \mathbb{R}^{L\times(Q-L-1)}$*:*

$$\widetilde{B}_{\ell j} = \begin{cases} 1 & \ell = 1, \\ -1 & \ell > 1, j = \kappa(\ell,\ell') - L \text{ for some } \ell' \in \{2,\dots,L\} \setminus \{\ell\}, \\ -2 & \ell > 1,\ j = \kappa(\ell,\ell) - L, \\ 0 & \text{otherwise}, \end{cases} \tag{2.39}$$

and $\widetilde{I}_B \in \mathbb{R}^{(Q-L)\times(Q-L-1)}$ *is the matrix given by the identity matrix of size* $Q - L - 1$ *to which an empty row has been appended from below. Then* B *is a Gale dual matrix of* E*.*

Proof. Note that the columns of B are linearly independent. Also, for $\boldsymbol{y} \in \mathbb{R}^Q$, we have $E\boldsymbol{y} = 0$ if and only if $\sum_{i=1}^{Q} y_i = 0$ and:

$$2y_{\kappa(\ell,\ell)} + \sum_{\ell'\neq\ell} y_{\kappa(\ell,\ell')} = 0, \quad \ell \in [L]. \tag{2.40}$$

It follows from Eq. (2.40) that:

$$\begin{aligned} \sum_{\ell'\geq\ell} y_{\kappa(\ell,\ell')} &= \sum_{\ell} y_{\kappa(\ell,\ell)} + \sum_{\ell}\sum_{\ell'>\ell} y_{\kappa(\ell,\ell')} \\ &= -\frac{1}{2}\sum_{\ell}\sum_{\ell'\neq\ell} y_{\kappa(\ell,\ell')} + \sum_{\ell}\sum_{\ell'>\ell} y_{\kappa(\ell,\ell')} \\ &= 0. \end{aligned} \tag{2.41}$$

Since the entries of $\boldsymbol{y}$ sum to zero, we then have:

$$y_Q = 0. \tag{2.42}$$

We will now express the set of equations Eq. (2.40) in the free variables $(y_{\kappa(\ell,\ell')})_{1<\ell\leq\ell'}$. For all

$\ell > 1$ in $[L]$:

$$y_{\kappa(1,\ell)} = -2y_{\kappa(\ell,\ell)} - \sum_{\ell' \neq \ell,1} y_{\kappa(\ell,\ell')}. \tag{2.43}$$

This in turn implies:

$$y_{\kappa(1,1)} = -\frac{1}{2}\sum_{\ell \neq 1} y_{\kappa(1,\ell)} = \sum_{i>1}\sum_{\ell'>\ell} y_{\kappa(\ell,\ell')}. \tag{2.44}$$

The set of equations in Eq.s (2.42), (2.43), and (2.44) in the free variables $(y_{\kappa(\ell,\ell')})_{1<\ell\leq\ell'\leq L}$ defines the kernel of E. It follows that the columns of B span the kernel of E.

□

Lemma 2.5.4 (Gale dual to the coefficient matrix of the degree centrality problem)**.** *Define the matrix $D \in \mathbb{R}^{Q\times(Q-L)}$ given by:*

$$D = \begin{pmatrix} \widetilde{D} \\ \widetilde{I}_D \end{pmatrix}, \tag{2.45}$$

where $\widetilde{D} \in \mathbb{R}^{L\times(Q-L)}$ is defined as follows:

$$\widetilde{D}_{ij} = \begin{cases} \hat{d}_1/(n_1 H_{11}) - \sum_{\ell>1} n_\ell \hat{d}_\ell/(n_1^2 H_{11}) & i = 1, j = 0, \\ (n_\ell - 1)n_\ell H_{\ell\ell}/(n_1^2 H_{11}) & i = 1, j = \kappa(\ell,\ell) - L \text{ for some } \ell \in \{2,\ldots,L\}, \\ 2n_\ell n_{\ell'} H_{\ell\ell'}/(n_1^2 H_{11}) & i = 1, j = \kappa(\ell,\ell') - L \text{ for some } \ell \neq \ell' \text{ in } \{2,\ldots,L\}, \\ \hat{d}_i/(n_1 H_{1i}) & i > 1, j = 0, \\ -(n_i - 1)H_{ii}/(n_1 H_{1i}) & i > 1, j = \kappa(i,i) - L, \\ -n_\ell H_{i\ell}/(n_1 H_{1i}) & i > 1, j = \kappa(i,\ell) - L \text{ for some } \ell \in \{2,\ldots,L\}\setminus\{i\}, \\ 0 & \text{otherwise}, \end{cases},$$

and $\widetilde{I}_D$ is given by the identity matrix of size $Q-L$ after moving its $(Q-L)$-th column before the remaining $Q-L-1$ columns. Then D is a Gale dual matrix of C.

Proof. Note that the columns of D are linearly independent. Also, for $\boldsymbol{y} \in \mathbb{R}^Q$, we have $C\boldsymbol{y} = \mathbf{0}$ if and only if $(n_\ell - 1)H_{\ell\ell}y_{\kappa(\ell,\ell)} + \sum_{\ell'} n'_\ell H_{\ell\ell'}y_{\kappa(\ell,\ell')} - \hat{d}_\ell y_Q = 0$ for all $\ell \in [L]$. For all $\ell > 1$:

$$y_{\kappa(1,\ell)} = -\frac{1}{n_1 H_{1\ell}}\left[\sum_{\ell'>\ell} n_{\ell'}H_{\ell\ell'}y_{\kappa(\ell,\ell')} + (n_\ell - 1)H_{\ell\ell}y_{\kappa(\ell,\ell)} - \hat{d}_\ell y_Q\right]. \tag{2.46}$$

Hence:

$$\begin{aligned} y_{\kappa(1,1)} &= -\frac{1}{n_1 H_{11}}\left[\sum_{\ell>1} n_\ell H_{1\ell}y_{\kappa(1,\ell)} - \hat{d}_1 y_Q\right] \\ &= \sum_{\ell>1}\frac{(n_\ell - 1)n_\ell H_{\ell\ell}}{n_1^2 H_{11}}y_{\kappa(\ell,\ell)} + \sum_{\ell>1}\sum_{\ell'>\ell}\frac{2n_\ell n_{\ell'}H_{\ell\ell'}}{n_1^2 H_{11}}y_{\kappa(\ell,\ell')} \\ &\quad + \left(\frac{\hat{d}_1}{n_1 H_{11}} - \sum_{\ell>1}\frac{n_\ell \hat{d}_\ell}{n_1^2 H_{11}}\right)y_Q. \end{aligned} \tag{2.47}$$

The set of equations in Eq.s(2.46) and (2.47) in the free variables $(y_{\kappa(\ell,\ell')})_{1<\ell\leq\ell'\leq L}$ and y_Q defines the kernel of C, which, as a result, is spanned by the columns of D. □

Theorem 2.5.2 (Existence of a solution for the degree centrality problem)**.** *Let Z be a membership matrix that results in a type-to-type interaction matrix $H = ZZ^\top$ with strictly positive entries. Let $\boldsymbol{n} = (n_\ell)_{\ell\in[L]}$ be a set of node type cardinalities that are all strictly greater than 1. Then the degree centrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{d}})$ admits a solution for any target $\hat{\boldsymbol{d}} \in \mathbb{R}_+^L$.*

Proof. The result is a direct application of Theorem 2.5.1. Note that C and E have maximal rank; all the other conditions of Theorem 2.5.1 are verified below.

$\underline{\mathbf{0} \in C_C}$

Let $\tilde{\boldsymbol{y}} \in \mathbb{R}^Q$. If we choose $\tilde{y}_Q = 1$ and $\tilde{y}_{\kappa(\ell,\ell')} = \min_\ell \hat{d}_\ell/(2\max_\ell \sum_{\ell'\neq\ell} n_{\ell'}H_{\ell\ell'})$ for all $\ell \neq \ell'$, then:

$$\tilde{y}_{\kappa(\ell,\ell)} = \frac{\hat{d}_\ell - \sum_{\ell'\neq\ell} n_{\ell'}H_{\ell\ell'}\tilde{y}_{\kappa(\ell,\ell')}}{(n_\ell - 1)H_{\ell\ell}} > 0 \tag{2.48}$$

It can be easily verified that $C\tilde{\boldsymbol{y}} = \mathbf{0}$.

Δ_P is a full dimensional bounded polytope

By Lemma 2.5.1, it suffices to show that $(1, 0, \ldots, 0) \in \mathbb{R}^Q$ belongs to C_P. Let $\tilde{\boldsymbol{y}} \in \mathbb{R}^Q$. It follows from Lemma 2.5.4 that $\tilde{y}D = (1, 0, \ldots, 0)$ if and only if:

$$\tilde{y}_Q = 1 + \left(\sum_{\ell>1} \frac{n_\ell \hat{d}_\ell}{n_1^2 H_{11}} - \frac{\hat{d}_1}{n_1 H_{11}} \right) \tilde{y}_{\kappa(1,1)} - \sum_{\ell>1} \frac{\hat{d}_\ell}{n_1 H_{1\ell}} \tilde{y}_{\kappa(1,\ell)}, \tag{2.49}$$

$$\tilde{y}_{\kappa(\ell,\ell)} = \frac{(n_\ell - 1)H_{\ell\ell}}{n_1 H_{1\ell}} \tilde{y}_{\kappa(1,\ell)} - \frac{(n_\ell - 1)n_\ell H_{\ell\ell}}{n_1^2 H_{11}} \tilde{y}_{\kappa(1,1)}, \tag{2.50}$$

$$\tilde{y}_{\kappa(\ell,\ell')} = \frac{n_{\ell'} H_{\ell\ell'}}{n_1 H_{1\ell}} \tilde{y}_{\kappa(1,\ell)} + \frac{n_\ell H_{\ell\ell'}}{n_1 H_{1\ell'}} \tilde{y}_{\kappa(1,\ell')} - \frac{2 n_\ell n_{\ell'} H_{\ell\ell'}}{n_1^2 H_{11}} \tilde{y}_{\kappa(1,1)}. \tag{2.51}$$

For all $\ell, \ell' > 1$, if we set:

$$\tilde{y}_{\kappa(1,\ell)} = \left(\frac{n_\ell H_{1\ell}}{n_1 H_{11}} + \varepsilon \right) \tilde{y}_{\kappa(1,1)}, \tag{2.52}$$

then we can check by means of Eqs. (2.50) and (2.51) that $\tilde{y}_{\kappa(\ell,\ell')} > 0$ and $\tilde{y}_{\kappa(\ell,\ell)} > 0$ for any choice of $\tilde{y}_{\kappa(1,1)} > 0$. We can then use Eq. (2.49) to set $\tilde{y}_{\kappa(1,1)} > 0$ small enough for $\tilde{y}_Q$ to be positive.

Assumption (i)

Recall the Gale dual matrix B of the coefficient matrix E in Lemma 2.5.4. Let B_S denote the submatrix of B formed by the rows of B with indices in S, and assume that S is such that B_S has mixed-sign columns[5]. We want to show that $\cap_{i \in S} F_i$ is empty.

First, assume that $1 \in S$. Then, for the $(\kappa(i,i) - L)$-th column of B_S to be mixed-sign, we must have $i \in S$, for all $i = 2, \ldots, L$. Hence, $[L] \subseteq S$. Now assume to the contrary that there exists $(1, \boldsymbol{y}) \in \mathbb{R}^{(L-1)L/2+1} \in \cap_{i \in [L]} F_i$.Because $(P_i)_{i \in S}$ belong to the boundary of Δ_P, we must have:

$$\langle \sum_{i \in S} a_i P_i, (1, \boldsymbol{y}) \rangle = 0. \tag{2.53}$$

[5]We say that a column is mixed-sign if it has at least two non-zero entries with opposite sign.

for any vector of scalars $(a_i)_{i\in S} \in \mathbb{R}^{|S|}$. However, if we were to remove the first column from the matrix D, its first L rows would be linearly dependent. To see this, note that by Lemma 2.5.4, we have:

$$\begin{aligned} -\sum_{i=2}^{L} n_i H_{1i} D_{i,\kappa(\ell,\ell')-L} &= -n_\ell H_{1\ell} D_{\ell,\kappa(\ell,\ell')-L} - \mathbb{1}_{\{\ell\neq\ell'\}} n_{\ell'} H_{1\ell'} D_{\ell',\kappa(\ell,\ell')-L} \\ &= n_1 H_{11} D_{1,\kappa(\ell,\ell')-L}. \end{aligned} \tag{2.54}$$

Hence:

$$\langle \sum_{i=1}^{L} n_i H_{1i} P_i, (1, \boldsymbol{y})\rangle = \sum_{i=1}^{L} n_i H_{1i} D_{i0} = \hat{d}_1 > 0, \tag{2.55}$$

which contradicts Eq. (2.53).

Next, assume that $1 \notin S$. B_S must contain at least one of the rows with indices in $\{2, \ldots, L\}$, since they are the only rows with some negative entries. Let $i \in \{2, \ldots, L\} \cap S$ be the index of such row. The negative entries of the i-th row of B are located on the columns with indices $\{\kappa(i, j) - L : j \in [L]\}$. Excluding the first row, the only positive entries on those columns are located on the rows of B with indices in $\{\kappa(i, j) : j \in [L]\}$. Hence, since $1 \notin S$, for B_S to have mixed-sign columns we must have $\{\kappa(i, j) : j \in [L]\} \subseteq S$. However, $F_i \cap (\cap_{j\in[L]} F_{\kappa(i,j)})$ is empty. To see this, assume to the contrary that there exists $(1, \boldsymbol{y}) \in \cap_{j\in[L]} F_{\kappa(i,j)}$. Because the only non-zero entry of row $P_{\kappa(i,j)}$ is the $(\kappa(i, j) - L)$-th entry, we have $y_{\kappa(i,j)-L} = 0$ for all $j \in [L]$, which in turn implies that $\langle P_i, (1, \boldsymbol{y})\rangle = D_{i0} = \hat{d}_i / n_1 H_{1i} > 0$.

Assumption (ii)

Let D_{-0} denote the submatrix of D obtained by removing its first column. By Lemma 2.5.3 and Lemma 2.5.4, $\text{sign}((D_{-0})_{ij}) = \text{sign}(B_{ij})$ for all i and j, so that second assumption in Theorem 2.5.1 is immediately verified. □

Bounding the roots We use the following bounds to initialize the multivariate Newton-Raphson algorithm.

Lemma 2.5.5 (Bounds for the solutions of the degree centrality problem)**.** *Let $\boldsymbol{\theta}$ denote the solution of the degree centrality problem for $(H, \boldsymbol{n}, \boldsymbol{d})$. Let $\bar{\theta}_\ell = \sqrt{\frac{d_\ell}{(n_\ell - 1)H_{\ell\ell}}}$ and $b_\ell = \sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} \bar{\theta}_{\ell'}$. Then:*

$$\frac{\sqrt{b_\ell^2 + 4(n_\ell - 1)H_{\ell\ell} d_\ell} - b_\ell}{2(n_\ell - 1)H_{\ell\ell}} \leq \theta_\ell \leq \bar{\theta}_\ell, \qquad \ell \in [L]. \tag{2.56}$$

Proof. The polynomial system of the degree centrality problem reads:

$$\sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} \theta_{\ell'} \theta_\ell + (n_\ell - 1) H_{\ell\ell} \theta_\ell^2 = \hat{d}_\ell. \tag{2.57}$$

The upper bound follows from the fact that all the terms in the LHS of Eq. (2.57) are non-negative, and thus $(n_\ell - 1)H_{\ell\ell}\theta_\ell^2 \leq \hat{d}_\ell$ for any $\ell \in [L]$.

Because $\theta_\ell \leq \bar{\theta}_\ell$, we have:

$$\sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} \bar{\theta}_{\ell'} \theta_\ell + (n_\ell - 1) H_{\ell\ell} \theta_\ell^2 - \hat{d}_\ell \geq 0. \tag{2.58}$$

The LHS of Eq. (2.58) is an upward parabola in θ_ℓ whose vertex falls to the left of the positive vertical semi-axis. The parabola crosses this semi-axis at $\underline{\theta}_\ell$. □

Lemma 2.5.6 (Bounds for the solutions of the eigencentrality problem)**.** *Let $\boldsymbol{\theta}$ denote the solution of the relaxed eigencentrality problem for $(H, \boldsymbol{n}, \boldsymbol{x}, d)$. Let $\bar{\theta}_\ell = \sqrt{\frac{\sum_{\ell > 1} n_\ell d}{(n_\ell - 1)H_{\ell\ell}}}$, $b_\ell = \frac{1}{x_\ell} \sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} x_{\ell'} \bar{\theta}_{\ell'}$, $b = \sum_\ell \sum_{\ell' > \ell} n_\ell n_{\ell'} H_{\ell\ell'} \bar{\theta}_{\ell'}$, and:*

$$\underline{\lambda} = \frac{\sqrt{b^2 + 2\sum_{\ell > 1} n_\ell (n_\ell - 1) H_{\ell\ell} \sum_{\ell > 1} n_\ell d} - b}{2\sum_{\ell > 1} n_\ell (n_\ell - 1) H_{\ell\ell}} \cdot \min_{\ell \in [L]} (n_\ell - 1) H_{\ell\ell}. \tag{2.59}$$

Then:

$$\frac{\sqrt{b_\ell^2 + 4(n_\ell - 1)H_{\ell\ell}\underline{\lambda}} - b_\ell}{2(n_\ell - 1)H_{\ell\ell}} \leq \theta_\ell \leq \bar{\theta}_\ell, \qquad \ell \in [L]. \tag{2.60}$$

Proof. Let $\ell^* = \arg\max_{\ell \in [L]} x_\ell$. Then, starting from the ℓ^*-th equation of the polynomial system behind the relaxed eigencentrality problem, we have:

$$\begin{aligned}
\lambda &= (n_{\ell^*} - 1) H_{\ell^*\ell^*} \theta_{\ell^*}^2 + \sum_{\ell' \neq \ell^*} n_{\ell'} H_{\ell'\ell^*} \frac{x_{\ell'}}{x_{\ell^*}} \theta_{\ell'} \theta_{\ell^*} \\
&\leq (n_{\ell^*} - 1) H_{\ell^*\ell^*} \theta_{\ell^*}^2 + \sum_{\ell' \neq \ell^*} n_{\ell'} H_{\ell'\ell^*} \theta_{\ell'} \theta_{\ell^*} \\
&\leq \left(\sum_{\ell \geq 1} n_\ell \right) d,
\end{aligned} \tag{2.61}$$

where in the last step we used the average degree constraint. The upper bound for θ_ℓ immediately follows after noting that $\theta_\ell \leq \sqrt{\frac{\lambda}{(n_\ell - 1) H_{\ell\ell}}}$.

Let $\hat{\ell} = \arg\max_{\ell \in [L]} x_\ell$. From the average degree constraint we have:

$$\left(\sum_{\ell \geq 1} (n_\ell - 1) n_\ell H_{\ell\ell} \right) \theta_{\hat{\ell}}^2 + \left(\sum_{\ell, \ell' > \ell} n_\ell n_{\ell'} H_{\ell\ell'} \bar{\theta}_{\ell'} \right) \theta_{\hat{\ell}} - \left(\sum_{\ell \geq 1} n_\ell \right) d \geq 0. \tag{2.62}$$

Using the same argument as in Eq. (2.58), Eq. (2.62) results in a lower bound for $\theta_{\hat{\ell}}$. Also, since $\lambda \geq \min_\ell (n_\ell - 1) H_{\ell\ell} \theta_{\hat{\ell}}^2$, then $\lambda \geq \underline{\lambda}$. This further implies that:

$$(n_\ell - 1) \theta_\ell^2 + \frac{1}{x_\ell} \left(\sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} x_{\ell'} \bar{\theta}_{\ell'} \right) \theta_\ell - \underline{\lambda} \geq 0. \tag{2.63}$$

Using once again the same argument as in Eq.s (2.58) and (2.62) yields the lower bound in Eq. (2.60). □

Chapter 3: Generating networks with target eigencentrality

3.1 Introduction and overview of results

Recall the network generation problem where we want to simulate networks that comply with a target structure. More specifically, we want to generate random adjacency matrices $A \in \mathbb{R}^{n\times n}$ from a Poisson factorization model under the constraint that:

$$\mathbb{E}_{\lambda} g(A) = \hat{\boldsymbol{g}}, \tag{3.1}$$

where $g : \mathbb{R}^{n\times n} \to \mathbb{R}^d$ is a map that encodes a target structure and $\hat{\boldsymbol{g}} \in \mathbb{R}^d$ is a fixed goal to achieve in expectation. While in Chapter 2 we set g to be the degree centrality, here we turn to an opposing measure of centrality – the eigencentrality.

The eigencentrality is defined as the principal eigenvector of the adjacency matrix and it captures the number of connections that a node shares with highly influential nodes. As such, the eigencentrality is a relative measure of the influence of nodes in a network and it represents a natural counterpart of the degree centrality problem. While most generators in the literature have focused on targeting the degree centrality, to the best of our knowledge this is the first work that presents a generative model for network data that allows for the control of the eigencentrality.

Before presenting our results, we summarize our contributions. Under the eigenvector centrality, the problem of recovering the parameters that solve Eq. (3.1) does not reduce to a tractable system of equations. Hence, rather than directly addressing the expected eigencentrality (i.e. the expected principal eigenvector of the adjacency matrix), we study a relaxation involving the principal eigenvector of the expected adjacency matrix. Under this much more tractable setting, the problem reduces to a system of multivariate polynomials of degree 2, similarly to the degree centrality case.

We can then extend the previous arguments to show that:

Under mild conditions, the relaxed eigencentrality problem always admits a solution in the parameter space that is unique for large graphs.

To assess the validity of such a relaxation, we show – via a Davis-Kahan argument (Yu, Wang, and Samworth, 2015) and under the same asymptotic regime as in Chapter 2 – that the relaxed eigencentrality converges in norm to the expected eigencentrality. In other words:

For large graphs, the relaxed eigencentrality problem is a valid approximation of the eigencentrality problem.

Similarly to the degree centrality case, we recover the solutions of the (relaxed) eigencentrality problem via a properly initialized multivariate Newton-Raphson algorithm.

3.2 The eigencentrality problem

Similarly to the degree centrality problem, in the eigencentrality problem we set a target eigencentrality for each node type and then aim to recover the degree correction terms which secure that target in expectation. Let the map $\boldsymbol{x} : \mathbb{R}^{n\times n} \to \mathbb{R}^n_+$ return the top eigenvector (with unitary norm) of an adjacency matrix (i.e. its eigencentrality).

Definition 3.2.1 (Eigencentrality problem)**.** *Let $\hat{\boldsymbol{x}} \in \mathbb{R}^L_+$ such that $\sum_\ell n_\ell \hat{x}^2_\ell = 1$ and let $\hat{\boldsymbol{x}}_\sigma = (\hat{x}_{\sigma(i)})_{i\in[N]}$; let $\hat{d} \in \mathbb{R}_+$. The eigencentrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d})$ amounts to finding a solution $\boldsymbol{\theta} \in \mathbb{R}^L_+$ of the system defined by the following set of equations:*

$$\mathbb{E}_{\boldsymbol{\theta}} \boldsymbol{x}(A(H, \boldsymbol{n})) = \hat{\boldsymbol{x}}_\sigma, \tag{3.2}$$

$$\mathbb{1}_L^\top \mathbb{E} A(\boldsymbol{\theta}; H, \boldsymbol{n}) \mathbb{1}_L = N\hat{d}. \tag{3.3}$$

In the eigencentrality problem, we pose the additional constraint Eq. (3.3) that targets the expected average degree of the graph. This is because the eigencentrality is a global – rather than local – measure of centrality and, as such, it does not control the sparsity of the network.
Ideally, we would like to have a tractable expression of the expected eigencentrality $\mathbb{E}\boldsymbol{x}(A)$ as a function of $(\boldsymbol{\theta}, H, \boldsymbol{n})$, but this is hard to achieve. We thus turn to a relaxed version of the eigencentrality problem.

Definition 3.2.2 (Relaxed eigencentrality problem)**.** *The relaxed eigencentrality problem for* $(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d})$ *amounts to finding a solution* $\boldsymbol{\theta} \in \mathbb{R}_+^L$ *of the system defined by the following set of equations:*

$$\boldsymbol{x}(\mathbb{E}A(\boldsymbol{\theta}; H, \boldsymbol{n})) = \hat{\boldsymbol{x}}_\sigma, \tag{3.4}$$

$$\mathbb{1}_L^\top \mathbb{E}A(\boldsymbol{\theta}; H, \boldsymbol{n})\mathbb{1}_L = N\hat{d}. \tag{3.5}$$

The relaxed eigencentrality problem targets the top eigenvector of the expected adjacency matrix $\boldsymbol{x}(\mathbb{E}A)$ rather than the expected eigencentrality $\mathbb{E}\boldsymbol{x}(A)$. The rationale behind this relaxation is bifaceted. First, $\boldsymbol{x}(\mathbb{E}A)$ is much more tractable than $\mathbb{E}\boldsymbol{x}(A)$. While we are not able to provide a closed form expression for $\boldsymbol{x}(\mathbb{E}A)$ in terms of $(\boldsymbol{\theta}, H, \boldsymbol{n})$, we can reformulate Eq. (3.4) using the eigenequations, as shown in the next section. Secondly, we can employ recent results from matrix perturbation analysis to control the distance between $\boldsymbol{x}(\mathbb{E}A)$ and $\mathbb{E}\boldsymbol{x}(A)$. By doing so, we are able to show that, for a fixed number of node types L, as we increase the network size N such distance vanishes at a rate that is proportional to $\sqrt{N}$. This is to say that, for large graphs, $\boldsymbol{x}(\mathbb{E}A)$ is a valid approximation of $\mathbb{E}\boldsymbol{x}(A)$.

3.2.1 Existence and asymptotic uniqueness

The target eigencentrality constraint in Eq. (3.4) can be reframed in terms of the eigenequations:

$$\mathbb{E}A(\boldsymbol{\theta}; H, \boldsymbol{n})\hat{\boldsymbol{x}}_\sigma = \lambda_1 \hat{\boldsymbol{x}}_\sigma, \tag{3.6}$$

for some $\lambda_1 \in \mathbb{R}_+$. The positivity of the top eigenvalue λ_1 is ensured by the Perron-Frobenius theorem.

As done with the degree centrality problem, we reduce the eigenequations and the sparsity requirement to a system of $(L+1)$ multivariate polynomials of degree 2 in the $(L+1)$ variables $(\boldsymbol{\theta}, \lambda_1) \in \mathbb{R}_+^{L+1}$. The resulting system is stated in Section 3.3. We can then employ the same arguments that we used for the degree centrality problem to show that there exists a solution to the relaxed eigencentrality problem and that, if there exist more than one, they are asymptotically unique.

Theorem 3.2.1 (Existence and asymptotic uniqueness of solutions of the relaxed eigencentrality problem)**.** *Let H and $\boldsymbol{n}$ satisfy the assumptions of Theorem 2.2.1. The relaxed eigencentrality problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d})$ admits a solution for any choice of the targets $(\hat{\boldsymbol{x}}, \hat{d}) \in \mathbb{R}_+^{L+1}$ such that $\sum_\ell n_\ell \hat{x}_\ell^2 = 1$. Moreover, the solutions of the problem are asymptotically unique and they converge to the solution of the problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d})$ with self-loops, which is unique.*

3.2.2 Perturbation analysis

Fix an affiliation graph with L node types and K communities, and set an eigencentrality goal $\hat{\boldsymbol{x}} = (\hat{x}_\ell)_{\ell \in L}$ and an average expected degree goal $\hat{d}$. For given $(\hat{\boldsymbol{x}}, \hat{d})$, we want to study how $\boldsymbol{x}(\mathbb{E}A)$ and $\mathbb{E}\boldsymbol{x}(A)$ differ as we increase the network size N.

We study the same asymptotic regime as in Section 2.2.2, where the proportion of nodes that belong to a node type stays constant as the cardinalities grow. More precisely, let $\boldsymbol{n}(1)$ denote an initial cardinality profile and set $\boldsymbol{n}(t) = t\boldsymbol{n}(1)$ for $t \in \mathbb{N}$; as $t \to \infty$, the network size $N(t) \overset{\text{not.}}{=} \sum_{\ell \in [L]} n_\ell(t) \to \infty$. Accordingly, we let $\hat{d}(1)$ denote an initial target average degree and set $\hat{d}(t) = t\hat{d}(1)$.

Our main result is based on the variant of the Davis-Kahan theorem introduced by Yu, Wang, and Samworth (2015), which, compared to the original formulation, involves the population eigenvalues rather than their sample counterpart. The theorem statement simplifies considerably when we are only interested in a single eigenvector, like in our case.

Lemma 3.2.1 (Yu, Wang, and Samworth (2015))**.**

$$||\boldsymbol{x}(A) - \boldsymbol{x}(\mathbb{E}A)|| \leq \frac{2^{3/2}||A - \mathbb{E}A||_{op}}{\lambda_1(\mathbb{E}A) - \lambda_2(\mathbb{E}A)}, \tag{3.7}$$

where $||\cdot||_{op}$ *denotes the matrix operator norm, and the map* $\lambda_i : \mathbb{R}^{N\times N} \to \mathbb{R}$ *returns the i-th largest eigenvalue of a N-by-N matrix with real entries.*

We are able to show that, as we increase the network size N, the operator norm $||A - \mathbb{E}A||_{\text{op}}$ grows on the scale of $\sqrt{N}$, while the eigengap $\lambda_1(\mathbb{E}A) - \lambda_2(\mathbb{E}A)$ scales linearly with N. It follows from Lemma 3.2.1 that for large graphs the sample eigencentrality $\boldsymbol{x}(A)$ is close to its population counterpart $\boldsymbol{x}(\mathbb{E}A)$. This is in line with previous works from the literature that focused on simpler generative models (see, for example, Lei and Rinaldo (2015)). Our asymptotic analysis, however, comes with the additional difficulty that the model parameters vary as we increase the network size while targeting a fixed eigencentrality goals $\hat{\boldsymbol{x}}$. Hence, as we increase the network size N, not only will the matrix $\mathbb{E}A \in \mathbb{R}^{N\times N}$ multiply in size, but its entries will also change non-trivially.
To control the eigengap in the denominator in Eq. (3.7), we then relate the original relaxed eigencentrality problem – whose solutions vary as $t \to \infty$ – with its equivalent version with self-loops – whose solutions stay constant as $t \to \infty$. Let $\boldsymbol{\theta}_P(t)$ and $\boldsymbol{\theta}(t)$ denote the solutions of the relaxed eigencentrality problem for $(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t))$ with and without self-loops, respectively. Also, we set $\mathbb{E}A(t) \overset{\text{not.}}{=} \mathbb{E}A(\boldsymbol{\theta}(t); H, \boldsymbol{n}(t))$ and let P denote the expected adjacency matrix given by $\mathbb{E}A$ plus the diagonal entries resulting from self-loops:

$$P(t) \overset{\text{not.}}{=} \mathbb{E}A(\boldsymbol{\theta}_P(t); H, \boldsymbol{n}(t)) + \text{diag}((\theta^2_{P,1}(t))_{1,\dots,n_1(t)}, \dots, (\theta^2_{P,L}(t))_{1,\dots,n_L(t)}). \tag{3.8}$$

Since $\boldsymbol{\theta}_P(t) = \boldsymbol{\theta}_P(1)$, it follows that the eigengap $\lambda_1(P(t)) - \lambda_2(P(t))$ is linear in t. We can then relate the eigengap of $\mathbb{E}A$ with that of P via Weyl's inequality:

$$\lambda_1(\mathbb{E}A(t)) - \lambda_2(\mathbb{E}A(t)) \geq t\,[\lambda_1(P(1)) - \lambda_2(P(1))] - 2C|\lambda_1(\mathbb{E}A(t)) - \lambda_1(P(t))|, \tag{3.9}$$

where $C > 0$ is some constant that depends on $\hat{\boldsymbol{x}}$. It turns out that we can control the gap between the top eigenvalues of the problem with and without self-loops.

Lemma 3.2.2. *As* $t \to \infty$, $|\lambda_1(\mathbb{E}A(t)) - \lambda_1(P(t))| = O(1)$.

It immediately follows that Eq. (3.9) results in the following asymptotic lower bound for our eigengap of interest.

Lemma 3.2.3. *As* $t \to \infty$, $\lambda_1(\mathbb{E}A(t)) - \lambda_2(\mathbb{E}A(t)) = \Omega(t)$.

To control the operator norm $||A - \mathbb{E}A||_{\text{op}}$, we extend the spectral bound for random binary matrices by Lei and Rinaldo (2015) to our multiedge Poisson setting.

Theorem 3.2.2 (Spectral bound for random matrices with independent Poisson entries)**.** *Let A be the adjacency matrix of an undirected random graph on N nodes in which edges occur independently according to the Poisson distribution. Assume that* $N \cdot \max_{ij} \mathbb{E}A_{ij} \leq d$ *for* $d \geq c_0 \log N$ *and* $c_0 > 0$*. Then, for any* $r > 0$*, there exists a constant* $C = C(c_0, r)$ *such that:*

$$||A - \mathbb{E}A||_{op} \leq C\sqrt{d}, \tag{3.10}$$

with probability at least $1 - N^{-r}$.

Huang and Feng (2018) derive a similar result but further require that $\max_{ij} \mathbb{E}A_{ij} < 1$. This is because the original argument for binary entries by Lei and Rinaldo (2015) makes extensive use of Bernstein's inequality, which under the Poisson distribution requires bounding the rates to control second moments. However, this sparsity assumption does not necessarily apply to our setting. To this end, we turn to Cramér-Chernoff bounds. The full proof is given in the Appendix.
Piecing together Theorem 3.2.2 and Lemma 3.2.3 by means of Lemma 3.2.1, we get the central result of this section.

Theorem 3.2.3. *For any* $r > 0$*, there exists* $C = C(\boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d}, r) > 0$ *such that:*

$$||\boldsymbol{x}(A(t)) - \hat{\boldsymbol{x}}|| \leq \frac{2^{3/2}C}{\sqrt{t}[\lambda_1(P(1)) - \lambda_2(P(1))] - O(1/\sqrt{t})} \quad \text{as } t \to \infty, \tag{3.11}$$

with probability at least $1 - N^{-r}$.

In words, if we fix the number of node types, as we let the cardinalities multiply in size and the average degree grow accordingly, the sample eigencentrality converges to the target at a $\sqrt{t}$-rate, with high probability.

3.3 The generator

To generate networks, we first need to recover the model parameters that reproduce the target eigencentrality in expectation and then use them to draw adjacency matrices from the Poisson factorization model. As with the degree centrality, the problem reduces to a system of multivariate polynomials of degree 2 in L variables and L equations, whose coefficients depend on the type interaction matrix H, the type cardinalities $\boldsymbol{n}$, and the eigenvector centrality targets. We recover the solutions of the problem via a multivariate Newton-Raphson algorithm.

Rather than tackling the problem given by the average degree constraint Eq. (3.5) and the eigencentrality constraint Eq. (3.6), we fix the leading eigenvalue $\lambda_1 = 1$ and tackle the eigencentrality constraint while momentarily ignoring the average degree constraint. The resulting system of polynomials reads:

$$(n_\ell - 1)H_{\ell\ell}x_\ell\theta_\ell^2 + \left(\sum_{\ell' \neq \ell} n_{\ell'}H_{\ell\ell'}\theta_{\ell'}x_{\ell'}\right)\theta_\ell - x_\ell = 0, \qquad \ell \in [L]. \tag{3.12}$$

Any solution $\boldsymbol{\theta}^*$ of Eq. (3.12) can be then rescaled so as to enforce the average degree constraint. More specifically, if the target average expected degree is d_0, then the solution of the relaxed eigencentrality problem is given by $\boldsymbol{\theta}^* d(\boldsymbol{\theta}^*)/\sqrt{d_0}$, where $d(\boldsymbol{\theta}^*)$ denotes the average expected degree resulting from $\boldsymbol{\theta}^*$. As done with the degree centrality case, we encode the coefficients of the system in the matrix of total interactions $T \in \mathbb{R}^{L\times L}$. We set:

$$T_{\ell\ell'} \doteq \begin{cases} 2(n_\ell - 1)H_{\ell\ell}x_\ell & \ell' = \ell \\ n_{\ell'}H_{\ell\ell'}x'_\ell & \ell' \neq \ell \end{cases}. \tag{3.13}$$

The map $f : \mathbb{R}_+^{L\times L} \to \mathbb{R}_+^L$ encoding the system in Eq. (3.12) and its Jacobian matrix J are defined exactly as with the degree centrality problem (see Eq.s 2.24 and 2.25).

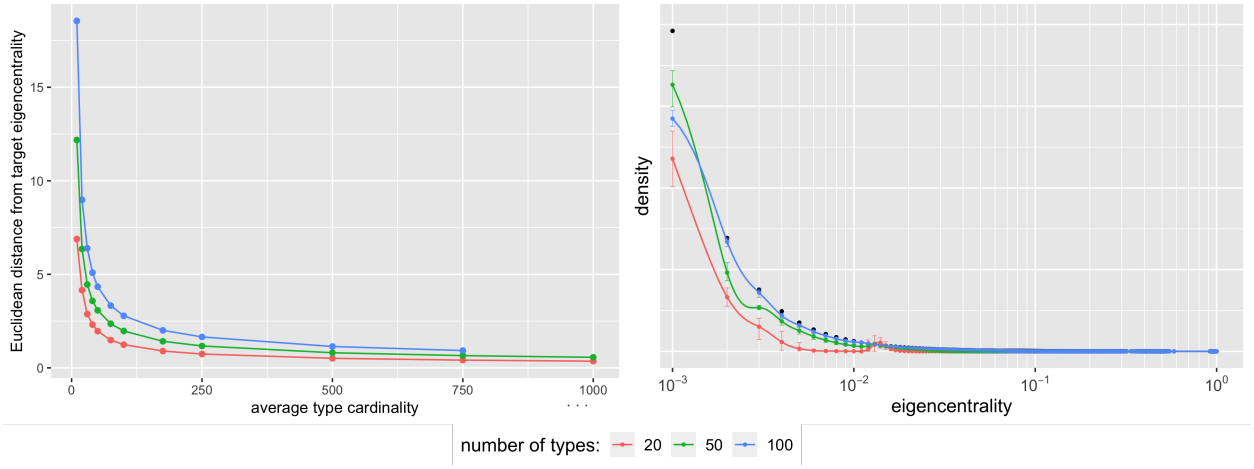
The full MNR algorithm for the relaxed eigencentrality problem and the resulting benchmark generator follow immediately from Algorithms 2.1 and 2.2 from Section 2.3. To initialize the MNR algorithm, we sample each parameter from the region defined by the bounds in Lemma 3.5.2 from Appendix 3.5.2.

3.4 Numerical experiments

We first provide a numerical illustration of our result on perturbation analysis (Theorem 3.2.3), and then examine the ability of our generator to reproduce a power-law eigencentrality distribution. From Theorem 3.2.3 we know that, as the networks grow in size and the average degree increases accordingly, the error introduced by our relaxation exhibits an inverse square-root decay. To investigate this result, we set a perfectly balanced eigencentrality target (i.e. $x_\ell = x_{\ell'}$ for all $\ell, \ell' \in [L]$) and cardinality profile (i.e. $n_\ell = n_{\ell'}$). We generate the type interaction matrix as in the experiments from Chapter 2. First, we assume that each node type is assigned with equal weights to 15 out of 100 total communities. Next, we ensure that no types have identical assignments, all communities have at least one type, and the resulting graph of type interactions is connected. Finally, we fix an initial average degree of 2 and an initial cardinality of 10 nodes for each type, and then gradually multiply them in size.

Figure 3.1 (a) presents Monte Carlo estimates of the expected distance between the sample eigencentrality and the target versus the average type cardinality, for different number of types. As expected, as the networks expands, their eigencentralities approach the target. At the same time, as the number of types increases, so does the average cardinality needed to secure a valid approximation. We may then wonder to what extent this effect impacts the precision of our method in reproducing a target distribution.

We investigate this in Figure 3.1 (b), which compares the empirical degree distribution of our networks against a target power-law distribution. As shown in the plot, as we increase the number



(a) (b)

Figure 3.1: (a) Monte Carlo estimates of the expected distance between the sample eigencentrality and the target versus the average type cardinality, for different number of types. (b) Monte Carlo estimates and 95% confidence intervals of the eigencentrality distribution (rounded to the nearest thousandth) for different node types. The black lines represent the target power-law distribution ($x_{\min} = 0.001, x_{\max} = 1, \alpha = 1.5, \bar{d} = 20$). In both plots, the estimates are computed over 100 networks per point for a single randomly generated interaction matrix.

of types, the empirical distribution better captures the target distribution even if the sizes of the networks are comparable for all types (~150,000 nodes). This is because, as discussed with the degree centrality case, the greater the number of types, the greater the number of q-quantiles used to pinpoint the target, and thus the greater the precision of the generator. This effect dominates the loss in precision of the eigencentrality relaxation due to the increased complexity of the problem. These experiments demonstrate that our initialization is effective for the relaxed eigencentrality problem, too.

3.5 Proofs

3.5.1 Existence and asymptotic uniqueness

Existence of a positive solution Here we prove the existence of a solution for the eigencentrality problems. The proof is slightly more convoluted than – but still along the lines of – that for the

degree centrality problem.

Recall that the system resulting from the relaxed eigencentrality for $(H, \boldsymbol{n}, \boldsymbol{x}, d)$ problem reads:

$$(n_\ell - 1)H_{\ell\ell}x_\ell\theta_\ell^2 + \sum_{\ell' \neq \ell} n_{\ell'}H_{\ell\ell'}x_{\ell'}\theta_{\ell'}\theta_\ell - \lambda x_\ell = 0, \qquad \ell \in [L], \tag{3.14}$$

$$\sum_\ell n_\ell(n_\ell - 1)H_{\ell\ell}\theta_\ell^2 + \sum_\ell \sum_{\ell' \neq \ell} n_\ell n_{\ell'}H_{\ell\ell'}\theta_{\ell'}\theta_\ell = N\hat{d}. \tag{3.15}$$

There is a bijection between the solutions of the relaxed eigencentrality problem and the solutions of Eq. (3.14) with $\lambda = 1$. To see this, assume there exists a solution $\boldsymbol{\theta}^*$ to Eq. (3.14) with $\lambda = 1$, and let $d(\boldsymbol{\theta}^*)$ denote the average expected degree resulting from $\boldsymbol{\theta}^*$. We can rescale $\boldsymbol{\theta}^*$ such that $\boldsymbol{\theta} = \boldsymbol{\theta}^*\sqrt{\hat{d}}/\sqrt{d(\boldsymbol{\theta}^*)}$ solves Eq. (3.15); as a result, $\boldsymbol{\theta}$ is also a root of Eq. (3.14) with $\lambda = \hat{d}/d(\boldsymbol{\theta}^*)$. This means that $(\boldsymbol{\theta}, \hat{d}/d(\boldsymbol{\theta}^*))$ is a solution of the relaxed eigencentrality problem. Hence, to assess existence of a root $(\boldsymbol{\theta}, \lambda)$ to Eq.s (3.14)–(3.15), it suffices to show that the system in Eq. (3.14) with $\lambda = 1$ admits a positive root.

The matrix of coefficients resulting from Eq. (3.14) reads:

$$C_{\ell q} = \begin{cases} n_{\ell'}H_{\ell\ell'}x'_\ell & q = \kappa(\ell, \ell') \text{ for some } \ell' \neq \ell \\ (n_\ell - 1)H_{\ell\ell}x_\ell & q = \kappa(\ell, \ell) \\ -x_\ell & q = Q \end{cases}. \tag{3.16}$$

The matrix of exponents for Eq. (3.14) is the same as that stated in Eq. (2.37) for the degree centrality problem. As Lemma 2.5.3 still applies, we just need to derive a Gale dual matrix of C.

Lemma 3.5.1 (Gale dual to the coefficient matrix in Eq. (3.16))**.** *Define the matrix* $D \in \mathbb{R}^{Q\times(Q-L)}$ *given by:*

$$D = \begin{pmatrix} \widetilde{D} \\ \widetilde{I}_D \end{pmatrix}, \tag{3.17}$$

where $\widetilde{D} \in \mathbb{R}^{L\times(Q-L)}$ *is defined as follows:*

$$\widetilde{D}_{ij} = \begin{cases} \left(n_1 x_1^2 - \sum_{\ell>1} n_\ell x_\ell^2\right)/n_1^2 H_{11} x_1^2 & i = 1, j = 0 \\ (n_\ell - 1) n_\ell H_{\ell\ell} x_\ell^2/(n_1^2 H_{11} x_1^2) & i = 1, j = \kappa(\ell,\ell) - L \text{ for some } \ell \in \{2,\dots,L\} \\ 2n_\ell n_{\ell'} H_{\ell\ell'} x_\ell x_{\ell'}/(n_1^2 H_{11}) & i = 1, j = \kappa(\ell,\ell') - L \text{ for some } \ell \neq \ell' \text{ in } \{2,\dots,L\} \\ x_i/(n_1 H_{1i} x_1) & i > 1, j = 0 \\ -(n_i - 1) H_{ii} x_i/(n_1 H_{1i} x_1) & i > 1, j = \kappa(i,i) - L \\ -n_\ell H_{i\ell} x_\ell/(n_1 H_{1i} x_1) & i > 1, j = \kappa(i,\ell) - L \text{ for some } \ell \in \{2,\dots,L\} \setminus \{i\} \\ 0 & \text{otherwise} \end{cases}, \tag{3.18}$$

and $\widetilde{I}_D$ *is given by the identity matrix of size* $Q - L$ *after moving its last column in front of the others. Then* D *is a Gale dual matrix of the coefficient matrix* C *in Eq.* (3.16)*.*

Proof. The result can be obtained along the lines of Lemma 2.5.4. □

Theorem 3.5.1 (Existence of a solution for the relaxed eigencentrality problem)**.** *Let* Z *be a membership matrix that results in a type-to-type interaction matrix* $H = ZZ^\top$ *with strictly positive entries. Let* $\boldsymbol{n} = (n_\ell)_{\ell\in[L]}$ *be a set of node type cardinalities that are all strictly greater than 1. Then the relaxed eigencentrality problem for* $(H, \boldsymbol{n}, \boldsymbol{x}, d)$ *admits a solution for any centrality target* $\boldsymbol{x} \in \mathbb{R}_+^L$ *and average expected degree target* $d \in \mathbb{R}_+$*.*

Proof. The existence of a root to Eq. (3.14) with $\lambda = 1$ follows from Lemma 2.5.3, Lemma 3.5.1, and Theorem 2.5.1. The argument is along the lines of that of Theorem 2.5.2. Because the set of such roots is in one-to-one correspondence with the solutions of the relaxed eigencentrality problem, the relaxed eigencentrality problem admits a solution, too. □

Asymptotic uniqueness We present the asymptotic uniqueness for the relaxed eigencentrality problem. Recall that, in our asymptotic analysis, we let the cardinalities and the target average

degree multiply in size by a constant factor $t \in \mathbb{N}$, and then let $t \to \infty$. We set $\boldsymbol{n}(t) = t\boldsymbol{n}(1)$, $\hat{d}(t) = t\hat{d}(1)$, and fix an initial cardinality profile $\boldsymbol{n}(1) \in \mathbb{R}_+^L$ and initial target average degree $\hat{d}(1) \in \mathbb{R}_+$.

Recall that the eigencentrality constraint of the relaxed problem for $(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d})$ reads:

$$(n_\ell - 1)H_{\ell\ell}\hat{x}_\ell\theta_\ell^2 + \sum_{\ell' \neq \ell} n_{\ell'}H_{\ell\ell'}\hat{x}_{\ell'}\theta_{\ell'}\theta_\ell - \lambda\hat{x}_\ell = 0, \quad \ell \in [L], \tag{3.19}$$

If we allowed for self-loops, the coefficient of the monomial θ_ℓ^2 would be replaced by $n_\ell H_{\ell\ell}x_\ell$. We can reformulate both systems – the one with and the one without self-loops – by recalling the representation in Eq. (2.29). Any system of L multivariate polynomials in the L variables in $\boldsymbol{\theta}$ can be represented as:

$$\sum_{q \in [Q]} c_{\ell q}\boldsymbol{\theta}^{\boldsymbol{e}_q} = 0, \quad \ell \in [L], \tag{3.20}$$

for some exponents $(\boldsymbol{e}_p)_{p\in[Q]} \in \mathbb{R}^L$ and matrix of real coefficients $C = (c_{\ell q})_{l\in[L],q\in[Q]}$, and where $\boldsymbol{\theta}^{\boldsymbol{e}_q} = \prod_{\ell\in[L]} \theta_\ell^{e_{q\ell}}$. We order the exponents (and thus the columns of C, too) based on the lexicographic order they impose on the monomials $(\boldsymbol{\theta}^{\boldsymbol{e}_q})_{q\in[Q]}$. In what follows, we let $C_P(t)$ denote the matrix resulting from omitting the last column of the coefficient matrix in Eq. (3.20) of the problem with self-loops for $(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d}(t))$. (This column refers to the constant term of the system, which has the vector of zeros $\mathbf{0} \in \mathbb{R}^L$ as exponent.) We denote by $\mathcal{S}_P(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t)) \in \mathbb{R}_+^{L+1}$ the set of solutions $(\boldsymbol{\theta}_P(t), \lambda_P(t))$ of the problem with self-loops. For the original problem without self-loops, we define $C_{\mathbb{E}A}(t)$ and $\mathcal{S}_{\mathbb{E}A}(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t))$ in a similar fashion. To avoid any confusion between the solutions of the two problems, we will also denote by $(\boldsymbol{\theta}_{\mathbb{E}A}(t), \lambda_{\mathbb{E}A}(t))$ the solutions of the original problem.

Note that, $\boldsymbol{\theta}_P(t) = \boldsymbol{\theta}_P(1)$, $\lambda_P(t) = t\lambda_P(1)$, and $C_P(t) = tC_P(1)$. It remains to understand the asymptotic behavior of a solution $(\boldsymbol{\theta}_{\mathbb{E}A}(t), \lambda_{\mathbb{E}A}(t))$, which is clarified by the following result.

Theorem 3.5.2. *The solution $(\boldsymbol{\theta}_P, \lambda_P)$ of the relaxed eigencentrality problem with self-loops for*

$(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \hat{d})$ *is unique. Moreover,* $|\lambda_{\mathbb{E}A}(t) - t\lambda_P(1)| = O(1)$ *and* $||\boldsymbol{\theta}_{\mathbb{E}A}(t) - \boldsymbol{\theta}_P|| = O(1/t)$ *as* $t \to \infty$.

Proof.

Uniqueness of $(\boldsymbol{\theta}_P, \lambda_P)$

Fix $(\boldsymbol{\theta}_P, \lambda_P(t)) \in \mathcal{S}_P(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t))$. Let $\tilde{\mathcal{S}}(t, \lambda_P(t))$ denote[1] the set of real and positive solutions $\tilde{\boldsymbol{\theta}}(t)$ of the following polynomial system:

$$\mathbb{E}A(\boldsymbol{n}(t), \tilde{\boldsymbol{\theta}}(t))\hat{\boldsymbol{x}} = \lambda_P(t)\hat{\boldsymbol{x}}. \tag{3.21}$$

Note that, because $\mathbb{E}A(\boldsymbol{n}, \boldsymbol{\theta})$ is homogenous of degree 2 in $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{\mathbb{E}A}(t)\sqrt{\frac{\lambda_P(t)}{\lambda_{\mathbb{E}A}(t)}} \in \tilde{\mathcal{S}}(t, \lambda_P(t)), \tag{3.22}$$

for any $(\boldsymbol{\theta}_{\mathbb{E}A}(t), \lambda_{\mathbb{E}A}(t)) \in \mathcal{S}_{\mathbb{E}A}(\boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t))$. Hence, for any t, $\tilde{\mathcal{S}}(t, \lambda_P(t))$ is non-empty.

We can rewrite (3.21) in vectorized form as:

$$(C_P(t) + \Delta C_P)(g(\boldsymbol{\theta}_P) + \Delta g(t)) = \lambda_P(t)\hat{\boldsymbol{x}}, \tag{3.23}$$

where $\Delta C_P = C_{\mathbb{E}A}(t) - C_P(t)$ does not depend on t, and $\Delta g(t) = g(\tilde{\boldsymbol{\theta}}(t)) - g(\boldsymbol{\theta}_P)$.

Since $(\boldsymbol{\theta}_P, \lambda_P(t)) \in \mathcal{S}_P(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t))$, we have:

$$\Delta C_P\,(g(\boldsymbol{\theta}_P) + \Delta g(t)) = -C_P(t)\Delta g(t). \tag{3.24}$$

If we apply the Frobenius norm to both sides, we can bound the LHS from below by noting that $|C_P(t)\Delta g(t)| \geq \rho_L(C_P(t))||\Delta g(t)||$, where $\rho_L(C_P(t))$ denotes the smallest singular value of $C_P(t)$.

[1] To simplify the notation, we omit the dependency of $\tilde{\mathcal{S}}$ on $(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}})$.

Since $C_P(t) = tC_P(1)$, we have $\rho_L(C_P(t)) = t\rho_L(C_P(1))$. Also:

$$\rho_L(C_P(1)) > 0. \tag{3.25}$$

To see this, note that if $\boldsymbol{y}^\top C_P(1) = \mathbf{0}$ then $y_\ell n_\ell(1) H_{\ell\ell}\hat{x}_\ell = 0$ for all $\ell \in [L]$. Hence, the left-null space of $C_P(1)$ is trivial for any choice of $(\boldsymbol{n}(1), H, \hat{\boldsymbol{x}})$.

It follows from the triangle inequality that:

$$||\Delta C_P||_F \left(||g(\boldsymbol{\theta}_P)|| + ||\Delta g(t)||\right) \geq t\rho_L(C_P(1))||\Delta g(t)||, \tag{3.26}$$

for any $\tilde{\boldsymbol{\theta}}(t) \in \tilde{\mathcal{S}}(t, \lambda_P(t))$. We can control the norm of $g(\tilde{\boldsymbol{\theta}}(t))$ as follows:

$$\begin{aligned} ||g(\tilde{\boldsymbol{\theta}}(t))|| &= \sqrt{\sum_{i\leq j} \tilde{\theta}_i^2(t)\tilde{\theta}_j^2(t)} \\ &\leq \sqrt{\sum_{i\leq j} \frac{\lambda_P^2(t)}{H_{ii}(tn_i(1)-1)H_{jj}(tn_j(1)-1)}} \\ &\leq \lambda_P(1)\left(\sum_{i\leq j} \frac{1}{H_{ii}(n_i(1)-1)H_{jj}(n_j(1)-1)}\right)^{\frac{1}{2}}, \end{aligned} \tag{3.27}$$

where in the last step we used the fact that $\lambda_P(t) = t\lambda_P(1)$.

Combining Eq.s (3.25), (3.26), and (3.27), we conclude that there exists a constant $c(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \lambda_P(1)) > 0$ such that:

$$\sup_{\tilde{\boldsymbol{\theta}}(t)\in\tilde{\mathcal{S}}(t,\lambda_P(t))} ||\Delta g(t)|| \leq \frac{||\Delta C_P||_F \left(2||g(\boldsymbol{\theta}_P)|| + ||g(\tilde{\boldsymbol{\theta}}(t))||\right)}{t\rho_L(C_P(1))} \leq \frac{c(H, \boldsymbol{n}, \hat{\boldsymbol{x}}, \lambda_P(1))}{t\rho_L(C_P(1))}. \tag{3.28}$$

Now, assume to the contrary that there exists a solution $(\boldsymbol{\theta}'_P, \lambda'_P(t)) \in \mathcal{S}_P(H, \boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d})$ that differs from $(\boldsymbol{\theta}_P, \lambda_P(t))$.

It follows from Eq.s (3.22)) and (3.28) that:

$$\sqrt{\frac{\lambda_P(1)}{\lambda'_P(1)}} = \frac{\theta_{P,i}}{\theta'_{P,i}}, \qquad i = 1, \ldots, L. \tag{3.29}$$

For the two solutions to be distinct, we must have $\lambda'_P(1) \neq \lambda_P(1)$, and thus $\boldsymbol{\theta}_P$ and $\boldsymbol{\theta}'_P$ must be proportional, which contradicts the fact that the graphs induced by $P(\boldsymbol{n}(1), \boldsymbol{\theta}_P)$ and $P(\boldsymbol{n}(1), \boldsymbol{\theta}'_P)$ have the same average degree $\hat{d}(1)$.

$||\boldsymbol{\theta}_{\mathbb{E}A}(t) - \boldsymbol{\theta}_P|| = O(1/t)$ as $t \to \infty$

Take $(H, \theta_{\mathbb{E}A}(t), \lambda_{\mathbb{E}A}(t)) \in \mathcal{S}_{\mathbb{E}A}(\boldsymbol{n}(t), \hat{\boldsymbol{x}}, \hat{d}(t))$ and let $\tilde{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}_{\mathbb{E}A}(t)\sqrt{\frac{\lambda_P(t)}{\lambda_{\mathbb{E}A}(t)}}$. By the average degree constraint on $\boldsymbol{\theta}_{\mathbb{E}A}(t)$:

$$\mathbf{1}^\top \mathbb{E}A(\boldsymbol{n}(t), \tilde{\boldsymbol{\theta}}(t))\mathbf{1} = \frac{\lambda_P(t)}{\lambda_{\mathbb{E}A}(t)} \mathbf{1}^\top \mathbb{E}A(\boldsymbol{n}(t), \boldsymbol{\theta}_{\mathbb{E}A}(t))\mathbf{1} = \frac{\lambda_P(t)}{\lambda_{\mathbb{E}A}(t)} t^2 \sum_i n_i(1)\hat{d}(1). \tag{3.30}$$

The LHS can be expanded as follows:

$$\mathbf{1}^\top \mathbb{E}A(\boldsymbol{n}(t), \tilde{\boldsymbol{\theta}}(t))\mathbf{1} = \mathbf{1}^\top P(\boldsymbol{n}(t), \tilde{\boldsymbol{\theta}}(t))\mathbf{1} - t \sum_i H_{ii} n_i(1) \tilde{\theta}_i^2(t). \tag{3.31}$$

Combining Eq. (3.30) and Eq. (3.31) yields:

$$\frac{\lambda_P(t)}{\lambda_{\mathbb{E}A}(t)} = \frac{1}{\hat{d}} \cdot \frac{\mathbf{1}^\top P(\boldsymbol{n}(1), \tilde{\boldsymbol{\theta}}(t))\mathbf{1} - \frac{1}{t}\sum_i H_{ii} n_i(1) \tilde{\theta}_i^2(t)}{\sum_i n_i(1)}. \tag{3.32}$$

By Eq. (3.28) and the average degree constraint on $\boldsymbol{\theta}_P$, we know that $\mathbf{1}^\top P(\boldsymbol{n}(1), \tilde{\boldsymbol{\theta}}(t))\mathbf{1}/\sum_i n_i(1) \to \hat{d}$. Since $\tilde{\theta}_i(t) = O(1)$, we can deduce from Eq. (3.32) that $|\lambda_{\mathbb{E}A}(t) - \lambda_P(t)| = O(1)$. Since $\lambda_P(t)/\lambda_{\mathbb{E}A}(t) \to 1$, we further conclude by Eq. (3.22) and Eq. (3.28) that $||\boldsymbol{\theta}_{\mathbb{E}A}(t) - \boldsymbol{\theta}_P|| = O(1/t)$. □

3.5.2 Bounding the roots

We use the following bounds to initialize the multivariate Newton-Raphson algorithm.

Lemma 3.5.2 (Bounds for the solutions of the eigencentrality problem)**.** *Let $\boldsymbol{\theta}$ denote the solution of the relaxed eigencentrality problem for $(H, \boldsymbol{n}, \boldsymbol{x}, d)$. Let $\bar{\theta}_\ell = \sqrt{\frac{\sum_{\ell>1} n_\ell d}{(n_\ell - 1)H_{\ell\ell}}}$, $b_\ell = \frac{1}{x_\ell} \sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} x_{\ell'} \bar{\theta}_{\ell'}$, $b = \sum_\ell \sum_{\ell'>\ell} n_\ell n_{\ell'} H_{\ell\ell'} \bar{\theta}_{\ell'}$, and:*

$$\underline{\lambda} = \frac{\sqrt{b^2 + 2\sum_{\ell>1} n_\ell (n_\ell - 1) H_{\ell\ell} \sum_{\ell>1} n_\ell d} - b}{2\sum_{\ell>1} n_\ell (n_\ell - 1) H_{\ell\ell}} \cdot \min_{\ell \in [L]} (n_\ell - 1) H_{\ell\ell} \tag{3.33}$$

Then:

$$\frac{\sqrt{b_\ell^2 + 4(n_\ell - 1)H_{\ell\ell}\underline{\lambda}} - b_\ell}{2(n_\ell - 1)H_{\ell\ell}} \leq \theta_\ell \leq \bar{\theta}_\ell, \qquad \ell \in [L]. \tag{3.34}$$

Proof. Let $\ell^* = \arg\max_{\ell \in [L]} x_\ell$. Then, starting from the ℓ^*-th equation of the polynomial system behind the relaxed eigencentrality problem, we have:

$$\begin{aligned}
\lambda &= (n_{\ell^*} - 1) H_{\ell^*\ell^*} \theta_{\ell^*}^2 + \sum_{\ell' \neq \ell^*} n_{\ell'} H_{\ell'\ell^*} \frac{x_{\ell'}}{x_{\ell^*}} \theta_{\ell'} \theta_{\ell^*} \\
&\leq (n_{\ell^*} - 1) H_{\ell^*\ell^*} \theta_{\ell^*}^2 + \sum_{\ell' \neq \ell^*} n_{\ell'} H_{\ell'\ell^*} \theta_{\ell'} \theta_{\ell^*} \\
&\leq \left(\sum_{\ell \geq 1} n_\ell \right) d,
\end{aligned} \tag{3.35}$$

where in the last step we used the average degree constraint. The upper bound for θ_ℓ immediately follows after noting that $\theta_\ell \leq \sqrt{\frac{\lambda}{(n_\ell - 1)H_{\ell\ell}}}$.

Let $\hat{\ell} = \arg\max_{\ell \in [L]} x_\ell$. From the average degree constraint we have:

$$\left(\sum_{\ell \geq 1} (n_\ell - 1) n_\ell H_{\ell\ell} \right) \theta_{\hat{\ell}}^2 + \left(\sum_{\ell, \ell' > \ell} n_\ell n_{\ell'} H_{\ell\ell'} \bar{\theta}_{\ell'} \right) \theta_{\hat{\ell}} - \left(\sum_{\ell \geq 1} n_\ell \right) d \geq 0. \tag{3.36}$$

Using the same argument as with the degree centrality case, we note that Eq. (3.36) results in a

lower bound for $\theta_{\hat{\ell}}$. Also, since $\lambda \geq \min_\ell (n_\ell - 1) H_{\ell\ell} \theta^2_{\hat{\ell}}$, then $\lambda \geq \underline{\lambda}$. This further implies that:

$$(n_\ell - 1)\theta^2_\ell + \frac{1}{x_\ell} \left(\sum_{\ell' \neq \ell} n_{\ell'} H_{\ell\ell'} x_{\ell'} \bar{\theta}_{\ell'} \right) \theta_\ell - \underline{\lambda} \geq 0. \tag{3.37}$$

Using once again the same argument as in Eq.s (2.58) and (3.36) yields the lower bound in Eq. (3.34). □

3.5.3 Perturbation analysis

Let A be the adjacency matrix of an undirected random graph on n nodes in which edges occur independently according to the Poisson distribution. Our goal is to extend the spectral bound for binary symmetric random matrices by Lei and Rinaldo (2015) to the random matrix A.

The original argument is based on a discretization of the n-dimensional unit sphere into a finite grid of vector pairs over which we maximize the Rayleigh quotient. The entries of these pairs are split into "light" and "heavy" entry pairs depending on their size, and their contribution to the Rayleigh quotient is then controlled separately. In both cases, the argument makes extensive use of Bernstein's inequality; however, under the Poisson distribution, such bounds would require additional assumptions on the rate to control second moments. Hence, to extend the original result to our multi-edge setting, we turn to Cramér-Chernoff bounds.

We proceed as follows. After stating the extended spectral bound and presenting the aforementioned discretization, we show how to bound the contribution of light and heavy entry pairs. For the proof of Lemma 3.5.3 and the combinatorial argument behind Lemma 3.5.5, we refer the reader to the supplemental material to Lei and Rinaldo (2015).

Main result and proof overview

Theorem 3.5.3 (Spectral bound for symmetric random matrices with independent Poisson entries)**.** *Let A be the adjacency matrix of an undirected random graph on n nodes in which edges occur independently according to the Poisson distribution. Set $\mathbb{E}A = \Lambda = (\lambda_{ij})_{i,j=1,\dots,n}$ and assume that*

$n \cdot \max_{i,j} \lambda_{i,j} \leq d$ *for* $d \geq c_0 \log n$ *and* $c_o > 0$. *Then, for any* $r > 0$, *there exists a constant* $C = C(c_0, r)$ *such that:*

$$||A - \Lambda|| \leq C\sqrt{d}, \tag{3.38}$$

with probability at least $1 - n^{-r}$.

Our goal is to bound:

$$\sup_{x \in \mathbb{R}^n : ||x||_2 \leq 1} |x^\top (A - \Lambda) x|. \tag{3.39}$$

This optimization can be carried over a finite grid of pairs rather than on the unit sphere in $\mathbb{R}^n$, at the cost of a small loss of accuracy that is quantified in the following result.

Lemma 3.5.3. *Fix* $\delta \in (0, 1)$ *and let:*

$$T = \left\{ x = (x_1, \ldots, x_n) : ||x|| \leq 1 \wedge \sqrt{n} x_i / \delta \in \mathbb{Z}, \forall i \right\}, \tag{3.40}$$

where $\mathbb{Z}$ *denotes the set of all integers. Then for all* $W \in \mathbb{R}^{n \times n}$*:*

$$||W||_2 \leq (1 - \delta)^{-2} \sup_{x,y \in T} |x^\top W y|. \tag{3.41}$$

For any $x, y \in T$, we define the set of light entry pairs as:

$$\mathcal{L}(x, y) = \left\{ (i, j) : |x_i y_j| \leq \sqrt{d}/n \right\}. \tag{3.42}$$

We refer to the pairs in $\bar{\mathcal{L}}(x, y)$ as heavy entry pairs. The contribution of (x, y) to $x^\top (A - \Lambda) y$ is then split into two parts, one corresponding to the light pairs and the other to the heavy pairs:

$$x^\top (A - \Lambda) y = \sum_{(i,j) \in \mathcal{L}(x,y)} x_i (A - \Lambda) y_j + \sum_{(i,j) \in \bar{\mathcal{L}}(x,y)} x_i (A - \Lambda) y_j. \tag{3.43}$$

We bound these two addends separately.

Bounding the contribution of light pairs

Lemma 3.5.4 (Light pair bound)**.**

$$P\left(\sup_{x,y\in T}\left|\sum_{(i,j)\in\mathcal{L}(x,y)} x_i y_j (A_{i,j}-\lambda_{i,j})\right| \geq c\sqrt{d}\right) \leq \exp\left\{-\left(\frac{c-1}{4} - 2\log\left(\frac{7}{\delta}\right)\right)n\right\}. \tag{3.44}$$

Proof. Let $u_{i,j} = x_i y_j \mathbb{1}(|x_i y_j| \leq \sqrt{d}/n) + x_j y_i \mathbb{1}(|x_j y_i| \leq \sqrt{d}/n)$ for $i, j = 1, \ldots, n$. Using the independence of the $A_{i,j}$'s and Markov's inequality, we have:

$$\begin{aligned} P\left(\left|\sum_{i<j} u_{i,j}(A_{i,j}-\lambda_{i,j})\right| \geq c\sqrt{d}\right) &\leq 2P\left(e^{t\sum_{i<j} u_{i,j}(A_{i,j}-\lambda_{i,j})} > e^{tc\sqrt{d}}\right) \\ &\leq 2e^{-tc\sqrt{d}} \prod_{i<j} \mathbb{E} e^{tu_{i,j}(A_{i,j}-\lambda_{i,j})} \\ &= 2\exp\left\{\sum_{i<j} \lambda_{i,j}(e^{tu_{i,j}} - tu_{i,j} - 1) - tc\sqrt{d}\right\}, \end{aligned} \tag{3.45}$$

for $t > 0$. To bound this further we follow the approach in Feige and Ofek (2005) and make use of the inequality $e^x - x - 1 \leq 2x^2$, which holds for $|x| \leq 1/2$. Since $|u_{i,j}| \leq 2\sqrt{d}/n$, if we pick $t = n/(4\sqrt{d})$, we have $|tu_{i,j}| \leq 1/2$. Then:

$$\sum_{i<j} \lambda_{i,j}(e^{tu_{i,j}} - tu_{i,j} - 1) \leq 2t^2 \sum_{i<j} \lambda_{i,j} u_{i,j}^2 \leq 2t^2 \frac{d}{n} \sum_{i<j} u_{i,j}^2, \tag{3.46}$$

where the second inequality follows from the fact that $d \geq n \max \lambda_{i,j}$. Finally, note that $\sum_{i<j} u_{i,j}^2 \leq 2\sum_{i<j}(x_i^2 y_j^2 + x_j^2 y_i^2) \leq 2\sum_i x_i^2 \sum_j y_j^2 = 2$. We then have:

$$\begin{aligned} P\left(\left|\sum_{i<j} u_{i,j}(A_{i,j}-\lambda_{i,j})\right| \geq c\sqrt{d}\right) &\leq 2\exp\left\{4t^2\frac{d}{n} - tc\sqrt{d}\right\} \\ &= 2\exp\left\{-\frac{c-1}{4}n\right\}. \end{aligned} \tag{3.47}$$

Finally, as discussed in Lei and Rinaldo (2015), we have $|T| \leq e^{n \log(7/\delta)}$ by a standard volume argument. This result combined with the union bound concludes the proof. □

Bounding the contribution of heavy pairs

Lemma 3.5.5 (Heavy pair bound)**.** *For any $c > 0$, there exists a constant $C = C(c)$ such that:*

$$\sup_{x,y \in T} \left| \sum_{(i,j) \in \bar{\mathcal{L}}(x,y)} x_i (A_{i,j} - \Lambda_{i,j}) y_j \right| \leq C\sqrt{d}, \tag{3.48}$$

with probability at least $1 - 2n^{-c}$.

The full combinatorial argument behind Lemma 3.5.5 can be found in the supplemental material to Lei and Rinaldo (2015). For this bound to hold in our Poisson setting, it suffices to show that the following two properties apply.

Lemma 3.5.6 (Bounded degree for Poisson multi-edge networks)**.** *Let d_i denote the degree of node i. Assume that $n \max_{i,j} \lambda_{i,j} \leq d$ for $d \geq c_0 \log n$ and $c_0 > 0$. Then, for $c > 0$, there exists a constant $c_1 = c_1(c)$ such that with probability at least $1 - n^{-c}$, $d_i \leq c_1 d$ for all i.*

Proof. We have:

$$\begin{aligned}
P(d_i \geq c_1 d) &\leq P\left(\sum_j A_{i,j} - \lambda_{i,j} \geq (c_1 - 1)d \right) \\
&\leq \inf_{t \geq 0} e^{-t(c_1-1)d} \prod_j \mathbb{E} e^{t(A_{i,j} - \lambda_{i,j})} \\
&= \exp\left\{ \inf_{t \geq 0} \sum_j \lambda_{i,j} \left(e^t - t - 1 \right) - t(c_1 - 1)d \right\} \\
&\leq \exp\left\{ \inf_{t \geq 0} \left[e^t - t - 1 - t(c_1 - 1) \right] d \right\} \\
&\leq n^{-c_0[(\log c_1 - 1)c_1 + 1]},
\end{aligned} \tag{3.49}$$

where the first and the second to last inequalities follows from the fact that $d \geq n \max_{i,j} \lambda_{i,j}$, and the last inequality from $d \geq c_0 \log n$ and the optimal choice $t = \log c_1$. □

Lemma 3.5.7 (Bounded discrepancy for Poisson multi-edge networks)**.** *Let $I, J \subseteq [n]$ be any two node subsets. Let $e(I, J)$ denote the number of distinct edges between I and J, and let $\bar{\mu}(I, J) = |I||J|d/n$.*

For $c > 0$, there exist constants $c_2 = c_2(c)$ and $c_3 = c_3(c)$, both larger than 1, such that with probability at least $1 - 2n^{-c}$, for any $I, J \subseteq [n]$ with $|I| \leq |J|$, at least one of the following holds:

1. $\frac{e(I,J)}{\bar{\mu}(I,J)} \leq ec_2$;
2. $e(I, J) \log \frac{e(I,J)}{\bar{\mu}(I,J)} \leq c_3|J| \log \frac{n}{|J|}$.

The bounded discrepancy property requires the number of edges between any two subgraphs not to deviate much from its expectation. To show this result, we employ a Cramér-Chernoff bound for centered Poisson random variables.

Lemma 3.5.8. *Let $X_1, \ldots, X_n$ be n independent Poisson random variables with mean $\mathbb{E}X_i = \lambda_i > 0$. Then, for any $x > 0$,*

$$P\left(\sum_{i=1}^{n} X_i - \lambda_i > x\right) \leq \exp\left\{x - \left(\sum_{i=1}^{n} \lambda_i + x\right) \log\left(\frac{x}{\sum_{i=1}^{n} \lambda_i} + 1\right)\right\}. \tag{3.50}$$

Proof. The result follows by a standard application of Cramér-Chernoff bounds (see, for example, Boucheron, Lugosi, and Massart, 2013). □

Proof sketch of Lemma 3.5.7. If $|J| \geq n/e$, Lemma 3.5.6 implies that $\frac{e(I,J)}{d|I||J|/n} \leq c_1 e$. If $|J| < n/e$, let $s(I, J)$ denotes the set of distinct edges between I and J, and let $k > 1$ be chosen later. Then:

$$\begin{aligned} P\left(e(I, J) \geq k\bar{\mu}(I, J)\right) &\leq P\left(\sum_{(i,j)\in s(I,J)} a_{i,j} - \lambda_{i,j} \geq k\bar{\mu}(I, J) - \sum_{(i,j)\in s(I,J)} \lambda_{i,j}\right) \\ &\leq P\left(\sum_{(i,j)\in s(I,J)} a_{i,j} - \lambda_{i,j} \geq (k-1)\bar{\mu}(I, J)\right). \end{aligned} \tag{3.51}$$

This can be further bounded using Lemma 3.5.8 by:

$$\exp\left\{(k-1)\bar{\mu}(I,J) - \left(\sum_{(i,j)\in s(I,J)} \lambda_{i,j} + (k-1)\bar{\mu}(I,J)\right)\log\left(\frac{(k-1)\bar{\mu}(I,J)}{\sum_{(i,j)\in s(I,J)}\lambda_{i,j}} + 1\right)\right\}. \tag{3.52}$$

Since $x \mapsto (x+b)\log(b/x+1)$ with $b > 0$ is monotonically decreasing on $(0,\infty)$, we have:

$$\begin{aligned} P\left(e(I,J) \geq k\bar{\mu}(I,J)\right) &\leq \exp\left\{(k-1)\bar{\mu}(I,J) - k\bar{\mu}(I,J)\log k\right\} \\ &\leq \exp\left\{\frac{1}{2}(k\log k)\bar{\mu}(I,J)\right\}, \end{aligned} \tag{3.53}$$

where the last inequality holds for $k \geq 8$.

The rest of the proof in Lei and Rinaldo, 2015 applies. For given $c_3 > 0$, let $t(I,J)$ denote the unique number such that $t(I,J)\log t(I,J) = \frac{c_3|J|}{\bar{\mu}(I,J)}\log\frac{n}{|J|}$, and set $k(I,J) = \max\{8, t(I,J)\}$. Then:

$$\begin{aligned} P\left(e(I,J) \geq k(I,J)\bar{\mu}(I,J)\right) &\leq \exp\left\{\frac{1}{2}k(I,J)\log k(I,J)\bar{\mu}(I,J)\right\} \\ &\leq \exp\left\{\frac{c_3|J|}{\bar{\mu}(I,J)}\log\frac{n}{|J|}\right\}. \end{aligned} \tag{3.54}$$

From this, it can be shown that:

$$P\left[\exists(I,J) : |I| \leq |J| \leq n/e, e(I,J) \geq k(I,J)\bar{\mu}(I,J)\right] \leq n^{-\frac{1}{2}(c_3-12)}. \tag{3.55}$$

Finally, we split the (I,J) pairs according to the value of $k(I,J)$. If $k(I,J) = 8$, then $e(I,J) \leq 8\bar{\mu}(I,J)$, with probability at least $1 - n^{-\frac{1}{2}(c_3-12)}$. If $k(I,J) = t(I,J) > 8$, $e(I,J)/\bar{\mu}(I,J) \leq t(I,J)$ and thus $\frac{e(I,J)}{\bar{\mu}(I,J)}\log\frac{e(I,J)}{\bar{\mu}(I,J)} \leq \frac{c_3|J|}{\bar{\mu}(I,J)}\log\frac{n}{|J|}$ with the same probability. To conclude the proof, we set $c_2 = \max\{c_1, 8\}$ and $c_3 = 2c + 12$. □

Chapter 4: Adaptive stochastic variational inference

4.1 Introduction

Stochastic variational inference (SVI, Hoffman et al., 2013) scales up traditional variational inference with stochastic natural gradient ascent. At each iteration of the ascent, SVI samples a random subset (or "minibatch") of the data to form noisy estimates of the gradient. SVI is readily used in conditionally conjugate models, latent-variable models for which each complete conditional is in the exponential family.

While noisy gradients enable efficient inference, their variance can slow convergence. In stochastic optimization, researchers have tried to alleviate this problem with non-uniform draws of the minibatch (Zhao and Zhang, 2014). In that work, the ideal minibatch distribution is often defined as one that maximizes the expected progress (Stich, Raj, and Jaggi, 2017) or, equivalently, the trace of the variance-covariance matrix of the gradient (Zhao and Zhang, 2015). In practice, however, the resulting distribution is infeasible as it requires iteration-dependent updates that are too expensive to compute.

This chapter develops adaptive minibatch sampling for SVI (AdaSVI). We show that, in the special case of SVI in a conditionally conjugate model, we can effectively approximate the optimal minibatch distribution. In detail, we make the following contributions:

- We derive a general closed form of the optimal minibatch distribution for the stochastic natural gradient of the variational objective. This distribution can be efficiently approximated, preserving the computational efficiency of SVI.

- We develop the specific form of the optimal minibatch distribution for latent Dirichlet allocation (Blei, Ng, and Jordan, 2003) and Poisson factorization (Gopalan, Hofman, and Blei,

2015). We show how to interpret these distributions, explaining which observations in the data are most pivotal for stochastic variational inference.

- We couple the optimal minibatch distribution with the adaptive learning rate by Ranganath, Gerrish, and Blei (2014). The resulting algorithm, AdaSVI, is a parameter-free version of SVI that only requires the minibatch size as input.

- We study AdaSVI on the Poisson factorization model. We find it significantly improves SVI, leading to faster convergence on synthetic data.

We emphasize that AdaSVI can be readily derived for any conditionally conjugate model. Examples include: Bayesian mixtures of exponential families, hierarchical regression models, matrix factorization models, and mixed-membership models (see Blei, Kucukelbir, and McAuliffe, 2017, for a review). Even some exceptions to this class, such as Gaussian processes and logistic models, can be expressed as conditional-conjugate with augmentations of the space of latent variables (Hensman, Fusi, and Lawrence, 2013; Durante and Rigon, 2019).

The rest of this chapter is organized as follows. In Section 4.2, we briefly recall SVI for conditionally conjugate models. In Section 4.3, we develop non-uniform minibatch sampling for SVI in general. In Section 4.4, we discuss the adaptive learning rate in Ranganath, Gerrish, and Blei (2014). In Section 4.5, we combine the results from the previous sections and present AdaSVI. Finally, in Section 4.6, we study the performance of AdaSVI on synthetic data from the Poisson factorization model. Section 4.7 concludes.

Related work AdaSVI contributes to the field of variance reduction for stochastic optimization. In different settings, researchers have proposed several techniques, such as control variates (Paisley, Blei, and Jordan, 2012; Ross, 2013) and Rao-Blackwellization (Casella and Robert, 1996).

This work builds on efforts to reduce the variance by non-uniform subsampling of the minibatch (Zhao and Zhang, 2014). As discussed, the optimal minibatch distribution is, in general, inefficient to compute. Consequently, recent efforts focus on building suboptimal, yet efficient,

sampling strategies. These include: optimizing an iteration-independent upper bound of the objective (Needell, Ward, and Srebro, 2014; Zhao and Zhang, 2015); partitioning the data into clusters and then sampling from them (Zhao and Zhang, 2014; Fu and Zhang, 2017; Csiba and Richtárik, 2018); using repulsive point processes to diversify the minibatch (Zhang, Kjellstrom, and Mandt, 2017); and adaptive yet efficient sampling distributions (Csiba, Qu, and Richtarik, 2015; Papa, Bianchi, and Clémençon, 2015; Perekrestenko, Cevher, and Jaggi, 2017; Stich, Raj, and Jaggi, 2017). These works mainly focus on stochastic gradient descent under the setting of convex objectives with smooth gradients. In contrast, SVI optimizes a non-convex objective.

In the context of SVI, Gopalan and Blei (2013) propose non-uniform minibatch sampling strategies for community detection on networks. Their approach maintains unbiasedness and reduces variance, but it is not designed to be optimal. Our work generalizes this method to any model amenable to SVI, and provides a provably optimal minibatch distribution.

Our work also builds on previous research on adaptive learning rates. For SVI, Ranganath, Wang, et al. (2013) propose a learning rate that minimizes the expected distance between the stochastic minibatch update and the full batch update. While inefficient to compute, the optimal learning rate can be estimated via exponentially decaying averages, in the same spirit of Schaul, Zhang, and LeCun (2013). AdaSVI couples this learning rate with importance minibatch sampling by jointly optimizing the same objective in Ranganath, Wang, et al. (2013) over both the learning rate and the sampling distribution.

While Ranganath, Wang, et al. (2013) provide a global learning rate, other works propose parameter-specific stepsizes. For example, AdaGrad (Duchi, Hazan, and Singer, 2011) scales the learning rate of each parameter by the root of the cumulative sum of past squared gradients. This correction term normalizes the gradient step by taking small steps towards large gradients and large steps towards small gradients. However, as AdaGrad progresses, the correction term accumulates in the denominator leading to fast decay. To address this flaw, AdaDelta (Zeiler, 2012) and RMSProp (Hinton, Srivastava, and Swersky, 2012) replace the cumulative correction term in AdaGrad with an exponentially decaying average. These adaptive learning rates are used in popular optimizers such as

Adam (Kingma and Ba, 2014). Inspired by this line of research, we further extend the adaptive stepsize of Ranganath, Gerrish, and Blei (2014) so that every variational parameter is updated with its own learning rate.

4.2 Background

We consider a conditional conjugate model with local and global latent variables. Let $\boldsymbol{x} = (x_i)_{i\in[n]}$ denote the data, $\boldsymbol{\beta}$ denote a vector of global latent variables, and $\boldsymbol{z} = (z_i)_{i\in[n]}$ denote a vector of local latent variables. The model reads:

$$p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{i=1}^{n} p(x_i, z_i \mid \boldsymbol{\beta}). \tag{4.1}$$

Variational inference (VI) approximates the posterior distribution of the latent variables in (4.1) by optimizing the evidence lower bound (ELBO):

$$\max_{q\in\boldsymbol{Q}} \text{ELBO}(q) = \max_{q\in\boldsymbol{Q}} \mathbb{E}_q(\log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta})) - \mathbb{E}_q \log q(\boldsymbol{z}, \boldsymbol{\beta}). \tag{4.2}$$

Solving Eq. (4.2) is tantamount to minimizing the Kullback-Leibler divergence of q to the exact posterior. We assume that $\boldsymbol{Q}$ is a mean-field family of approximate distributions, so that the latent variables are mutually independent under any $q \in Q$. We let $\boldsymbol{\varphi} = (\varphi_i)_{i\in[n]}$ and $\boldsymbol{\lambda}$ parametrize the densities in $\boldsymbol{Q}$ of the local and global latent factors, respectively.

Coordinate-ascent variational inference (CAVI) tackles the optimization problem in Eq. (4.2) via a coordinate ascent algorithm. For conditional conjugate models, the optimal variational factors are in the same exponential family of the complete conditionals and the corresponding updates for the variational parameters $(\boldsymbol{\varphi}, \boldsymbol{\lambda})$ can be derived in closed form. However, updating the global parameters $\boldsymbol{\lambda}$ is inefficient for large-scale data, as it requires a single pass through the whole dataset at each iteration (Hoffman et al., 2013).

One way to address this issue is to turn to stochastic optimization. Stochastic variational inference runs stochastic gradient ascent on the natural gradient of the ELBO with respect to the global

variational parameters $\boldsymbol{\lambda}$. As shown in Blei, Kucukelbir, and McAuliffe (2017), the natural gradient of the ELBO with respect to $\boldsymbol{\lambda}$ is:

$$g(\boldsymbol{\lambda}) = \boldsymbol{\alpha} + \left[\sum_{i=1}^{n} \mathbb{E}_{\varphi_i^*} t(z_i, x_i), n \right] - \boldsymbol{\lambda}, \tag{4.3}$$

where we recall that $t(\cdot, \cdot)$ is the sufficient statistic of the joint distribution of (z_i, x_i) given $\boldsymbol{\beta}$. At each iteration of the ascent, we estimate the natural gradient with a noisy gradient computed over a random minibatch $S \subset [n]$ of size m. We produce such an estimate with the current local variational parameters $\boldsymbol{\varphi}^*$, which are first optimized with the usual CAVI updates:

$$\varphi_i^* = \mathbb{E}_{\lambda} \left[\eta(\boldsymbol{\beta}, x_i) \right]. \tag{4.4}$$

Her η denotes the natural parameter of the complete conditional of the local variable z_i, which belongs to the exponential family.

4.3 Adaptive minibatch sampling

We present adaptive minibatch sampling for SVI. First, we introduce a framework for non-uniform minibatch sampling and derive the optimal sampling distribution, which we refer to as the oracle. The resulting sampling probabilities explain which observations are the most pivotal for inference. We thus derive and interpret the oracle distribution for latent Dirichlet allocation and Poisson factorization model. While informative, the oracle is infeasible to compute. In the last section, we present two strategies to approximate it.

4.3.1 An optimal sampling distribution

In the original SVI formulation, the minibatch is sampled uniformly at random. In practice, any distribution over the data can be used, provided that the resulting noisy gradient is an unbiased estimate of the natural gradient.

Non-uniform random sampling We introduce a more general framework where data points are sampled with replacement into the minibatch according to a vector of probabilities $\boldsymbol{p} = (p_i)_{i\in[n]}$. An unbiased estimate of the natural gradient can be assembled by reweighting each contribution from the i-th local context by its inverse sampling probability p_i. More specifically, consider the following noisy gradient:

$$\hat{g}_{\boldsymbol{p}}(\boldsymbol{\lambda}) = \boldsymbol{\alpha} + \left[\frac{1}{m}\sum_{i\in S} p_i^{-1}\mathbb{E}_{\varphi_i^*} t(z_i, x_i), n\right] - \boldsymbol{\lambda}, \tag{4.5}$$

where we recall again that t is the sufficient statistic of the joint density of (z_i, x_i) given $\boldsymbol{\beta}$, and φ_i^* is the local variational parameter optimized via Eq. (4.4) with the current instance of $\boldsymbol{\lambda}$.
This gradient is an unbiased estimator of the natural gradient in Eq. (4.3). See Section 4.8 for the proof.

Optimal sampling distribution Under the noisy gradient in Eq. (4.5), any choice of $\boldsymbol{p}$ leads to a valid stochastic optimization procedure. For example, setting $p_i = 1/n$ yields the uniform random sampling formulation in Hoffman et al. (2013). However, some minibatch sampling distributions may result in very noisy gradient estimates, impairing the convergence of the algorithm. Of all possible sampling distributions $\boldsymbol{p}$, the optimal choice is the one that minimizes the variance of its corresponding noisy natural gradient.
Following Zhao and Zhang (2015), we quantify the variance of the noisy gradient through the trace of the variance-covariance matrix:

$$V(\hat{g}_{\boldsymbol{p}}(\boldsymbol{\lambda})) = \mathbb{E}_{\boldsymbol{p}}\left\|\hat{g}_{\boldsymbol{p}}(\boldsymbol{\lambda}) - g(\boldsymbol{\lambda})\right\|^2, \tag{4.6}$$

where $||\cdot||$ denotes the $L2$ norm and where the expectation is taken with respect to the distribution of the random minibatch S, drawn from $\boldsymbol{p}$. Under sampling with replacement, minimizing the

variance of the gradient is equivalent to the following:

$$\min_{\boldsymbol{p}\in\Delta_{n-1}} V(\hat{g}_{\boldsymbol{p}}(\boldsymbol{\lambda})) \iff \min_{\boldsymbol{p}\in\Delta_{n-1}} \sum_{i=1}^{n} p_i^{-1} \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|^2, \tag{4.7}$$

where Δ_{n-1} denotes the $(n-1)$-simplex. From Eq. (4.7), an application of Cauchy-Schwarz inequality yields the optimal sampling probabilities:

$$p_i^* \propto \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|. \tag{4.8}$$

See Section 4.8 for details on how to derive Eq.s (4.6) and (4.8).

We refer to the distribution $\boldsymbol{p}^* = (p_i^*)_{i\in[n]}$ as the oracle distribution, and to each p_i^* as the optimal sampling probability or *importance score* of the i-th local context. We note that the optimal probability is iteration-dependent, as it is a function of the current instance of the local variational parameters. This makes the optimal sampling strategy adaptive: at each iteration, its weights are dynamically adjusted as a result of the update of the variational parameters.

The unnormalized importance scores only depend on the i-th local context, namely on the local variational parameter φ_i^* and on the data x_i. Hence, once the local variational parameters are optimized, the unnormalized importance scores are cheap to compute. In practice, Eq. (4.8) can be seen as an additional variational update that comes with no computational cost. However, the full oracle distribution cannot be maintained efficiently, as it would require a single pass over the whole dataset at each iteration. In Section 4.3.3, we discuss two methods to approximate the oracle.

4.3.2 Importance scores for discrete factorization models

We discuss non-uniform minibatch sampling for two different high-dimensional factorization models: topic discovery via Latent Dirichlet Allocation (LDA) and recommender systems via Poisson factorization. We derive and interpret the optimal importance scores for both settings, characterizing the observations that are the most pivotal for inference.

Latent Dirichlet Allocation LDA (Blei, Ng, and Jordan, 2003) is a Bayesian topic model that uses latent variables to recover topics from large collections of documents. LDA represents the original motivation for SVI: while early efforts based on CAVI could handle collections of tens of thousands of documents, SVI allows for topic discovery on corpora of millions of documents (Hoffman et al., 2013).

Under this generative model, topics are modeled as distributions over a vocabulary of words, while documents exhibit such topics with different proportions. We let K, D, and V denote the number of topics, documents, and words in the vocabulary, respectively. The generative model is:

1. Draw topics $(\boldsymbol{\beta}_k)_{k\in[K]} \sim \text{Dirichlet}_V(\boldsymbol{\eta})$.
2. For each document $d \in [D]$:
   - Draw the topic proportions $\boldsymbol{\theta}_d \sim \text{Dirichlet}_K(\boldsymbol{\alpha})$.
   - For each word in document d:
     - Draw its topic assignments $\boldsymbol{z}_{dn} \sim \text{Multinomial}(1, \boldsymbol{\theta}_d)$.
     - Draw the word $\boldsymbol{w}_{dn} \sim \text{Multinomial}(1, \boldsymbol{\beta} \cdot \boldsymbol{z}_{dn})$.

The complete conditionals of both global topics and local topic proportions turn out to be Dirichlet, while the complete conditional of local topic assignments is multinomial (Hoffman et al., 2013). Hence, under the optimal variational mean-field family, if $w_{dnv} = 1$ then $z_{dn} \overset{q}{\sim} \text{Multinomial}(1, \boldsymbol{\varphi}_{dv})$ where $(\boldsymbol{\varphi}_{dv})_{d\in[D],v\in V}$ denotes the local per-word variational parameters.

The joint distribution of the local context $(\boldsymbol{w}_d, \boldsymbol{z}_d, \boldsymbol{\theta}_d)$ given the global topics $\boldsymbol{\beta}$ is:

$$p(\boldsymbol{w}_d, \boldsymbol{z}_d, \boldsymbol{\theta}_d \mid \boldsymbol{\beta}) \propto h(\boldsymbol{z}_d, \boldsymbol{\theta}_d) \exp\left\{\sum_{v,k} m_{dvk}(\boldsymbol{w}_d, \boldsymbol{z}_d) \log \beta_{vk}\right\}, \tag{4.9}$$

where $m_{dvk}(\boldsymbol{w}_d, \boldsymbol{z}_d) = \sum_n \mathbb{1}_{\{w_{dnv}=1\}} \mathbb{1}_{\{z_{dnk}=1\}}$ is the number of times word v from document d is assigned to topic k. The sufficient statistic of this distribution is then $t(\boldsymbol{w}_d, \boldsymbol{z}_d) = \boldsymbol{m}_d$. It follows

from Eq. (4.8) that the optimal sampling probability of document d is:

$$p_d^* \propto \sqrt{\sum_v n_{dv} \left\|\boldsymbol{\varphi}_{dv}\right\|^2}, \tag{4.10}$$

where n_{dv} is the number of occurrences of word v in document d. The squared optimal sampling probability of a document is a linear combination of the squared norms of the per-word parameters $\boldsymbol{\varphi}_{dv}$, whose weights depend on the frequency of word occurrences.

Note that $\boldsymbol{\varphi}_{dv}$ is a vector with non-negative entries that sum to one: for the norm $||\boldsymbol{\varphi}_{dv}||$ to be maximized, word v must be assigned to one and only one topic in document d. This coincides with the intuition that distinctive terms define topics; Eq. (4.10) states that such terms are the most pivotal to our estimation process. Between two documents of the same length, the one with the most topic-defining terms is more likely to be visited by the oracle.

Poisson Factorization Poisson factorization (Dunson and Herring, 2005) is a probabilistic matrix factorization method that uses the Poisson distribution to model the data. The flexibility of this model to capture sparse factors and its efficiency on sparse matrices have proved useful in a wide range of applications, including, for example, signal processing (Virtanen, Cemgil, and Godsill, 2008; Cemgil, 2008), genomics (Levitin et al., 2019), and recommendation systems (Gopalan, Hofman, and Blei, 2015).

We use the language of recommendation systems. We observe a matrix of user-behavior data over a set of items. Each user is given a vector of latent preferences; each item has a latent vector of attributes. When the attributes of items meet the preferences of users, a user-item interaction is generated. The generative model reads:

1. Sample the preferences $\theta_{uk} \sim \text{Gamma}(a, b)$ of each user u, for all components $k \in [K]$.
2. Sample the attributes $\beta_{ik} \sim \text{Gamma}(c, d)$ of each item i, $k \in [K]$.
3. For each user u and item i:

- sample the user-item contribution $z_{uik} \sim \text{Poisson}(\boldsymbol{\beta}_i^\top \boldsymbol{\theta}_u)$ from component k, $k \in [K]$;
- set $y_{ui} = \sum_{k \in [K]} z_{uik}$.

In the formulation above, the model is augmented with the local latent factors $\boldsymbol{z}_{ui}$, which make it conditional conjugate (Gopalan, Hofman, and Blei, 2015). The complete conditionals of the global variables $\boldsymbol{\theta}_u$ and $\boldsymbol{\beta}_i$ follow a Gamma distribution, due to the Gamma-Poisson conjugacy. As for the local latent variables $\boldsymbol{z}_{ui}$, the complete conditionals follow a multinomial distribution, which arises from conditioning independent Poisson random variables on their sum.

The joint density of the i-th user-item interaction given the latent preferences and attributes is:

$$p(y_{ui}, \boldsymbol{z}_{ui} \mid \boldsymbol{\theta}_u, \boldsymbol{\beta}_i) = h(y_{ui}, \boldsymbol{z}_{ui}) \exp\left\{ \sum_{k \in [K]} z_{uik} \log \theta_{uk} \beta_{ik} - \sum_{k \in [K]} \theta_{uk} \beta_{uk} \right\}. \tag{4.11}$$

The sufficient statistic of this distribution is $t(y_{ui}, \boldsymbol{z}_{ui}) = \boldsymbol{z}_{ui}$. Recalling that $\boldsymbol{z}_{ui} \overset{q}{\sim} \text{Multinomial}(y_{ui}, \boldsymbol{\varphi}_{ui})$, it follows that the optimal sampling probability of a user-item interaction is:

$$p^*_{ui} \propto y_{ui} \left\| \boldsymbol{\varphi}_{ui} \right\|. \tag{4.12}$$

The importance score of a user-item interaction is driven by two components: its magnitude and how aligned user preferences and item attributes are. As with LDA, the norm $\left\| \boldsymbol{\varphi}_{ui} \right\|$ is maximized when user u interacts with item i exclusively via attribute k. This means that, among interactions of the same magnitude, those that match k-enthusiast users with k-defining items have the highest importance score.

In practice, for our stochastic optimization algorithm, we may want to sample by row (i.e. by user) rather than by cell (i.e. by user-item interaction). The user importance scores can be derived similarly by treating all the user variational parameters as local. Let $(\gamma^{\text{shape}}_{uk}, \gamma^{\text{rate}}_{uk})_{k \in [K]}$ denote the

Gamma variational parameters for the users. The user importance scores read:

$$p_u^* \propto \sqrt{\sum_i y_{ui}^2 \left\|\boldsymbol{\varphi}_{ui\cdot}\right\|^2 + \left\|\frac{\boldsymbol{\gamma}_{u\cdot}^{\text{shape}}}{\boldsymbol{\gamma}_{u\cdot}^{\text{rate}}}\right\|^2}, \tag{4.13}$$

where the ratio of vectors in Eq. (4.13) is component-wise. Recall that the ratio $\lambda_{uk}^{\text{shape}}/\lambda_{uk}^{\text{rate}}$ represents the mean preference of user u for attribute k under the variational distribution. The norm $||\boldsymbol{\gamma}_{u\cdot}^{\text{shape}}/\boldsymbol{\gamma}_{u\cdot}^{\text{rate}}||$ can be then seen as an indicator of the user activity level across all attributes. Hence, similarly to user-item interactions, the user importance scores are driven by two components: how specialized user preferences are and how much users consume across all attributed.

Poisson Factorization for network data Below we study the specific instance of Poisson factorization for community detection on networks that we discussed in the previous chapters.
In detail, consider a network with n nodes distributed over K overlapping communities. Let y_{ij} denote the number of edges between two nodes i and j. The model is:

$$\theta_{ik} \overset{\text{i.i.d.}}{\sim} \text{Gamma}(\alpha_i, \beta_i), \quad z_{ijk} \overset{\text{i.i.d.}}{\sim} \text{Poisson}(\theta_{ik}\theta_{jk}), \tag{4.14}$$

and $y_{ij} = \sum_k z_{ijk}$. The model in Eq. (4.14) is a specific instance of the Poisson factorization model above, where rows and columns share the same latent space. The global latent variables θ_{ik} represent the propensity of a node i to belong to community k; the local latent factors z_{ijk} capture the number of edges between i and j via community k.
The importance score of an edge between two nodes i and j is in (4.12). If we only observe simple graphs (i.e. $y_{ij} \in \{0, 1\}$ for all i, j), the optimal sampling weights are proportional to $\left\|\boldsymbol{\varphi}_{ij}\right\|$, where we recall that $\boldsymbol{\varphi}$ is the set of local variational parameters. Also, $\boldsymbol{\varphi}_{ij} = 1$ if and only if edge (i, j) is formed via a single community. Such an edge can only be generated by two nodes that belong exclusively to that community. Hence, in simple graphs, the edges with the highest importance score will be those between non-overlapping nodes that belong to the same community. Conversely, $\left\|\boldsymbol{\varphi}_{ij}\right\|$ is minimized when edge (i, j) is equally likely to be formed via

any community, suggesting that edges that sit at the overlap of communities are less likely to be visited by the oracle.

4.3.3 Approximating the optimal sampling distribution

We propose two methods to approximate the optimal sampling distribution in Eq. (4.8).

Iteration-independent importance scores The first approach is to approximate the optimal importance scores with scores that only depend on the data $\boldsymbol{x}$ and the hyperparameters $\boldsymbol{\alpha}$. Letting $\boldsymbol{p}^e = (p_i^e)_{i\in[n]}$ denote such scores, we want:

$$p_i^*(\varphi_i, x_i, \alpha_i) \approx p_i^e(x_i, \alpha_i). \tag{4.15}$$

Iteration-independent importance scores come with two computational advantages. The first is that they do not need to be updated at each iteration, as they do not depend on the current state of learning. The second advantage is that sampling with replacement from a static distribution is very efficient. In general, while uniform random sampling of a minibatch of size m requires $O(m)$ operations, weighted sampling is more computationally burdensome. Efficient algorithms for weighted sampling usually require building a data structure that supports fast sampling, a task that depends at least loglinearly on n. (See, e.g., Walker's alias method by Walker, 1977 or the binary-heap method in Wong and Easton, 1980.) However, for static distributions, this preprocessing step is run only at the first iteration and the subsequent sampling can be done in $O(m)$.

We have yet to discuss how to construct the iteration-independent scores in Eq. (4.15). A common approach in the convex optimization literature assumes the Lipschitz-continuity of gradients and employs the Lipschitz constants to form an iteration-independent upper bound of the variance of the noisy gradient. The resulting variance minimization problem yields importance scores that do not depend on the current state of learning. This approach is not feasible under our setting as the ELBO is not necessarily Lipschitz continuous. However, we can still bound the variance of the noisy gradient.

The main idea is to use the CAVI updates to bound the variational parameters in terms of the model hyperparameters and the data. Next, we employ such bounds to control the variance of the noisy gradient with iteration-independent quantities. To illustrate this approach, consider the Poisson factorization model stated in Section 4.3.2. The CAVI updates for the user Gamma shape and rate parameters $(\gamma_{uk}^{\text{shape}}, \gamma_{uk}^{\text{rate}})$ read:

$$\gamma_{uk}^{\text{shape}} = \alpha_u^{\text{shape}} + \sum_i y_{ui} \varphi_{uik}, \tag{4.16}$$

$$\gamma_{uk}^{\text{rate}} = \alpha_u^{\text{rate}} + \sum_i \frac{\delta_{ik}^{\text{shape}}}{\delta_{ik}^{\text{rate}}}, \tag{4.17}$$

where we recall that: α_u^{shape} denotes the Gamma shape hyperparameter of user u; y_{ui} is the number of user-item interactions between user u and item i; φ_{uik} is the variational multinomial probability that user-item interaction (u, i) is via attribute k; and $(\delta_{ik}^{\text{shape}}, \delta_{ik}^{\text{rate}})$ are the variational Gamma parameters for item i.

Since $\varphi_{uik} \leq 1$, it follows from Eq. (4.16) that:

$$\gamma_{uk}^{\text{shape}} \leq \alpha_u^{\text{shape}} + \sum_i y_{ui}. \tag{4.18}$$

Similarly, the item Gamma variational parameters are positive, it follows from Eq. (4.17) that:

$$\gamma_{uk}^{\text{rate}} \geq \alpha_u^{\text{rate}}. \tag{4.19}$$

While crude, these bounds allow us to control the user importance scores via the hyperparameters $(\alpha_u^{\text{shape}}, \alpha_u^{\text{rate}})$ and the data $\boldsymbol{y}$. To see this, recall that the user importance scores read:

$$p_u^* \propto \sqrt{\sum_i y_{ui}^2 ||\boldsymbol{\varphi}_{ui\cdot}||^2 + \left\| \frac{\boldsymbol{\gamma}_{u\cdot}^{\text{shape}}}{\boldsymbol{\gamma}_{u\cdot}^{\text{rate}}} \right\|^2}. \tag{4.20}$$

where the ratio of vectors in Eq. (4.20) is component-wise. Using Eq.s (4.18) and (4.19), we can bound from above the RHS of Eq. (4.20) by:

$$\sqrt{\sum_i y_{ui}^2 + K\left[\frac{\alpha_u^{\text{shape}} + \sum_i y_{ui}}{\alpha_u^{\text{rate}}}\right]^2}. \tag{4.21}$$

We can then define the iteration-independent importance score $\boldsymbol{p}_u^e = \boldsymbol{p}_u^e(\alpha_u^{\text{shape}}, \alpha_u^{\text{rate}}, \boldsymbol{y}_{u\cdot})$ for user u as the normalized scores resulting from Eq. (4.21).

Adaptive minibatch sampling via rejection sampling Iteration-independent importance scores lack one important feature of optimal importance scores – their adaptivity. The oracle in Eq. (4.8) dynamically reweights the scores so as to minimize the variance of the noisy gradient at the current state of learning. This is in contrast with iteration-independent scores, which are only based on the prior information (encoded by the hyperparameters) and on the data $\boldsymbol{x}$.

To implement a fully adaptive minibatch sampling strategy, we turn to rejection sampling. The iteration-independent sampling distribution $\boldsymbol{p}^e$ introduced in the previous paragraph comes in handy as a natural candidate for a proposal distribution. Throughout the rest of this chapter we then refer to $\boldsymbol{p}^e$ as the envelope sampling distribution, meaning that we use (a multiple of) $\boldsymbol{p}^e$ as the envelope for rejection sampling of the oracle. We denote by $\tilde{p}^*$ and $\tilde{p}^e$ the unnormalized oracle and envelope scores, respectively. We set $\tilde{p}_i^* \leq \tilde{p}_i^e$ for all points i.

To assemble a minibatch under rejection sampling, we sample a point i from the envelope and compute its unnormalized oracle score $\tilde{p}_i^*$ as in Eq. (4.8). We then include the candidate in the minibatch with probability $\tilde{p}_i^*/\tilde{p}_i^e$ and repeat these steps until the full minibatch is formed.

Rejection sampling allows us to sample directly from the oracle at little computational cost. Rather than calculating the whole set of probabilities, we compute only the unnormalized scores of the candidate points sampled from the envelope. The computational complexity of the algorithm depends on how closely the envelope matches the oracle: the closer the envelope, the lower the rejection rate, and the more efficient the sampling.

Even if we can sample from the oracle distribution to assemble the minibatch, we still need to estimate the inverse probability weights that make the noisy gradient unbiased. To do that, we not only require the unnormalized scores (which we have already computed) but also the normalizing constant, $\sum_{i\in[n]} p_i^*$. This constant cannot be computed, as it would require a full pass through the dataset at each iteration.

With the goal of estimating the normalizing constant, we use the unnormalized scores of all candidate points that have been sampled from the envelope (including those that were rejected). We combine these scores with the inverse envelope probabilities so as to form an unbiased estimate of the normalizing constant. Interestingly, the resulting probability estimates (i.e. unnormalized scores over the estimated normalizing constant) are not unbiased estimates of the oracle probabilities. This is because the normalizing constant sits at the denominator. However, the noisy gradient is linear in the inverse probability weights, which are in turn linear in the normalizing constant. Hence, as the estimate of the normalizing constant is unbiased, our algorithm preserves the unbiasedness of the noisy gradient. The full algorithm is stated below.

The additional estimation step required for the oracle normalizing constant may slow down the convergence of the algorithm. While sampling from the oracle minimizes the variance of the noisy gradient, estimating the normalizing constant adds extra noise. We investigate this trade-off in the numerical studies in Section 4.6.

Algorithm 4.1 Adaptive minibatch sampling via rejection-sampling

Input: envelope $\tilde{\boldsymbol{p}}^e$, data $\boldsymbol{x}$, minibatch size m, current global variational parameters $\boldsymbol{\lambda}$.

Output: minibatch S, noisy gradient g.

Initialize: minibatch $S = \emptyset$, candidate set $E = \emptyset$.

while $|S| < m$ **do**

  Sample i from the normalized envelope and set $E = E \cup \{i\}$.

  Optimize the local variational parameters $\boldsymbol{\varphi}_i^* = \boldsymbol{\varphi}_i^*(\boldsymbol{\lambda})$.

  Compute the unnormalized oracle score $\tilde{p}_i^*$ from Eq. (4.8).

  Sample $u \sim \text{Uniform}([0, 1])$.

  if $u < \tilde{p}_i^* / \tilde{p}_i^e$ **then**

    Set $S = S \cup \{i\}$.

  Compute estimate of the oracle normalizing constant:

$$c = \frac{\sum_j \tilde{p}_j^e}{|E|} \sum_{i \in E} \tilde{p}_i^* / \tilde{p}_i^e. \tag{4.22}$$

  Compute the noisy gradient:

$$g = \boldsymbol{\alpha} + \left[\frac{c}{m} \sum_{i \in S} (\tilde{p}_i^*)^{-1} \mathbb{E}_{\varphi_i^*} t(z_i, x_i), n \right] - \boldsymbol{\lambda}. \tag{4.23}$$

4.4 Adaptive learning rate

Stochastic optimization methods are sensitive to learning rates. If the learning rate is too small, the optimizer may move too slowly; if the learning rate is too large, the optimizer may oscillate too much. A common approach in the literature is to set up a decreasing learning rate schedule, based on Robbins-Monro conditions (see Section 1.4.2). Under this setting, the optimizer is sensitive to the choice of the decay rate. A fast decay may slow down convergence, while a slow decay may

lead to erratic progress.

To this end, adaptive learning rates dynamically adjust the learning step size depending on how noisy the current gradient estimate is. They increase the step size toward the noisy gradient when the variance is low, while they reduce it when the noise is large. Clearly, the higher the variance, the less reliable the direction pointed by the noisy gradient.

For locally quadratic objectives, Schaul, Zhang, and LeCun (2013) propose to set the learning rate adaptively by minimizing the expected objective at each iteration. While the ELBO is not necessarily locally quadratic, one may adopt a similar approach by optimizing its Taylor approximation. However, Ranganath, Gerrish, and Blei (2014) found that the resulting adaptive learning rate behaves erratically. They argue that this is due to the fact that the Taylor approximation is imprecise when the step size is large and that the Hessian of the ELBO may not be negative definite.

Rather than tackling directly the ELBO, the same authors propose to choose the learning rate so as to minimize the squared distance between the noisy update and its counterpart – the full batch update. This defines an alternative notion of optimal learning rate which can be estimated efficiently at each iteration. In the rest of this section we review this approach. Following Ranganath, Gerrish, and Blei (2014), we first present the optimal learning rate arising from the distance-minimization problem and then state an efficient algorithm to estimate it. Finally, we derive a parameter-specific extension that assigns a different learning rate to every variational parameter.

4.4.1 An optimal adaptive learning rate

Recall from Chapter 1 that, for conditionally conjugate models, the CAVI updates for the global variational parameters are:

$$\boldsymbol{\lambda}_t^* = \boldsymbol{\alpha} + \sum_{i=1}^{n} \mathbb{E}_{\varphi_{t,i}} t(z_i, x_i). \tag{4.24}$$

Since the CAVI update requires visiting the full dataset, we refer to it as the full batch update. Conversely, the stochastic natural ascent update in SVI is only based on a subset of the data S and

it reads:

$$\boldsymbol{\lambda}_{t+1} = (1 - \rho_t)\boldsymbol{\lambda}_t + \rho_t \hat{\boldsymbol{\lambda}}_t, \tag{4.25}$$

where by $\hat{\lambda}_t$ we denote the intermediate minibatch-based update:

$$\hat{\boldsymbol{\lambda}}_t = \boldsymbol{\alpha} + \sum_{i \in S}^{n} \mathbb{E}_{\varphi_{t,i}} t(z_i, x_i). \tag{4.26}$$

The minibatch update in Eq. (4.25) depends on the learning rate. The main idea in Ranganath et al. Ranganath, Gerrish, and Blei (2014) is to choose the learning rate ρ_t so as to minimize the expected squared distance between the noisy update in Eq. (4.25) and the full batch update in Eq. (4.24). The problem of interest is then:

$$\min_{\rho_t} \mathbb{E} \left\| \boldsymbol{\lambda}_{t+1}(\rho_t) - \boldsymbol{\lambda}_t^* \right\|^2, \tag{4.27}$$

where the expectation is taken with respect to the sampling distribution of the minibatch, which is the only stochastic component of the algorithm. The solution of the problem is:

$$\rho_t^* = \frac{\left\| \boldsymbol{\lambda}_t - \boldsymbol{\lambda}_t^* \right\|^2}{\left\| \boldsymbol{\lambda}_t - \boldsymbol{\lambda}_t^* \right\|^2 + tr(\Sigma)}, \tag{4.28}$$

where Σ denotes the variance-covariance matrix of the intermediate global parameter $\hat{\boldsymbol{\lambda}}_t$.

The optimal learning rate in Eq. (4.28) is driven by two components: the distance between the full batch update $\boldsymbol{\lambda}_t^*$ and the current estimate $\boldsymbol{\lambda}_t$, and the trace of the variance-covariance matrix of the intermediate global parameter $\hat{\boldsymbol{\lambda}}_t$. If the current estimate is far from the batch update, the optimizer takes a large step towards the direction pointed by the noisy gradient. At the same time, as the noise of the intermediate parameter grows, the learning rate decreases.

While adaptive, the optimal learning rate in Eq. (4.28) cannot be computed. The full batch update is not readily available, as it requires a single pass through the whole dataset at each iteration. The

same applies to the variance-covariance matrix of the intermediate parameters. However, both the numerator and the denominator in Eq. (4.28) can be estimated while preserving the computational efficiency of the stochastic variational inference algorithm, as shown next.

4.4.2 Estimating the learning rate

To estimate the optimal learning rate, Ranganath, Wang, et al. (2013) propose to rewrite its numerator and denominator in terms of expectations of (some transformations of) the noisy gradient. One could then estimate such expectations with Monte Carlo draws. However, estimating different noisy gradients by drawing multiple minibatches would affect the computational efficiency of the stochastic variational inference algorithm. This is why the authors turn to exponential moving averages, in the same spirit of Schaul, Zhang, and LeCun (2013).

Starting from the terms in the numerator, note that:

$$\mathbb{E}g_t = \mathbb{E}\hat{\boldsymbol{\lambda}}_t - \boldsymbol{\lambda}_t = \boldsymbol{\lambda}_t^* - \boldsymbol{\lambda}_t, \tag{4.29}$$

which follows from the fact that the intermediate parameter is an unbiased estimator of the full batch update. (Recall that the expectation in Eq. (4.29) is defined with respect to the sampling distribution of the minibatch.)

As for the denominator, note that:

$$\mathbb{E}g_t^\top g_t = (\mathbb{E}g_t)^\top \mathbb{E}g_t + \mathrm{tr}\Sigma = \left\|\boldsymbol{\lambda}_t^* - \boldsymbol{\lambda}_t\right\|^2 + \mathrm{tr}\Sigma. \tag{4.30}$$

It follows from Eq.s (4.29) and (4.30) that the optimal learning rate can be rewritten as:

$$\rho_t^* = \frac{(\mathbb{E}g_t)^\top \mathbb{E}g_t}{\mathbb{E}g_t^\top g_t}. \tag{4.31}$$

The optimal learning rate can be then expressed in terms of expectations of the noisy natural gradient g_t. In particular, to estimate the denominator, one needs not to estimate the full covariance

matrix Σ, but just an unidimensional summary of the noisy gradient.

As previously mentioned, an efficient way to estimate the expectations in Eq. (4.31) is by using moving averages across iterations. At a given iteration t, let $\bar{g}_t$ and $\bar{h}_t$ denote the current averages for $\mathbb{E}g_t$ and $\mathbb{E}g_t^\top g_t$, respectively, and let τ_t denote the current window size of the exponential moving average. Then the estimates are updated as follows:

$$\bar{\boldsymbol{g}}_t = (1 - \tau_t^{-1})\bar{g}_{t-1} + \tau_t^{-1}\hat{\boldsymbol{g}}_t, \tag{4.32}$$

$$\bar{h}_t = (1 - \tau_t^{-1})\bar{h}_{t-1} + \tau_t^{-1}\hat{\boldsymbol{g}}_t^\top \hat{\boldsymbol{g}}_t. \tag{4.33}$$

The resulting estimate $\bar{\rho}_t$ of the learning rate ρ_t^* at time t is:

$$\bar{\rho}_t = \frac{\bar{g}_t^\top \bar{g}_t}{\bar{h}_t}. \tag{4.34}$$

The learning rate $r\bar{h}o_t$ is not an unbiased estimate of the optimal learning rate ρ_t^*.

The window size can be updated as follows:

$$\tau_{t+1} = \tau_t(1 - \bar{\rho}_t) + 1. \tag{4.35}$$

The update in Eq. (4.35) reflects the fact that, when the optimizer takes a large step away from the previous instance of the parameter, the current moving average is less accurate of an estimate. As a result, the window size shrinks.

To initialize the algorithm, Ranganath, Gerrish, and Blei (2014) propose to estimate the noisy gradient at the initial global parameter $\boldsymbol{\lambda}_0$ with Monte Carlo draws. The number of samples drawn to initialize the moving averages can be used to set the initial window size τ_0.

4.4.3 Parameter-specific learning rates

The adaptive learning rate in Ranganath, Gerrish, and Blei (2014) is a single rate shared by all variational parameters. However, when the gradient magnitudes differ significantly across param-

eters, faster convergence may be achieved with parameter-specific learning rates (Duchi, Hazan, and Singer, 2011; Schaul, Zhang, and LeCun, 2013). The results reviewed in the previous paragraphs can be easily extended to this multidimensional setting as the problem is separable in the learning rates.

To see this, let ρ_{td} denote the learning rate of the global variational parameters (λ_{td}) at iteration t, and let $\boldsymbol{\rho}_t = (\rho_{td})_{d\in[D]}$. The learning rates in $\boldsymbol{\rho}_t$ are chosen so as to minimize the expected squared distance of the stochastic update $\boldsymbol{\lambda}_{t+1}$ from the full batch update $\boldsymbol{\lambda}_t^*$:

$$\min_{\boldsymbol{\rho}_t} \mathbb{E} \left\| \boldsymbol{\lambda}_{t+1}(\boldsymbol{\rho}_t) - \boldsymbol{\lambda}_t^* \right\|^2, \tag{4.36}$$

where we recall that the expectation is taken with respect to the sampling distribution of the mini-batch. It can be shown that the problem in Eq. (4.36) reduces to:

$$\min_{\boldsymbol{\rho}_t} \sum_{d\in[D]} (1-\rho_{td})^2 \mathbb{E}\left(\hat{\lambda}_{td} - \lambda_{td}^*\right)^2 + \rho_{td}^2 (\lambda_{td} - \lambda_{td}^*)^2, \tag{4.37}$$

which is separable in ρ_{td}. The solution is:

$$\rho_{td}^* = \frac{(\lambda_{td} - \lambda_{td}^*)^2}{(\lambda_{td} - \lambda_{td}^*)^2 + \Sigma_{dd}}, \tag{4.38}$$

where Σ_{dd} is the d-th entry of the diagonal of Σ, i.e. the variance of the intermediate global parameter $\hat{\lambda}_{td}$. The optimal parameter-specific learning rate ρ_{td}^* can be expressed as the ratio between the squared mean of the noisy gradient and its second moment, which can be both estimated via exponentially decaying averages as shown in the previous sections.

The resulting adaptive learning rate relates to other parameter-specific rates such as AdaDelta (Duchi, Hazan, and Singer, 2011) and RMSProp (Hinton, Srivastava, and Swersky, 2012). In particular, it uses exponentially decaying averages (like AdaDelta and RMSProp) and it is unit-free (like AdaDelta). However, it does not involve the root mean square gradient. Also, it is theoretically justified by the optimization problem in Eq. (4.36).

4.5 Adaptive stochastic variational inference

In this section, we couple the minibatch sampling distribution with the learning rate. The resulting algorithm, which we name adaptive stochastic variational inference (AdaSVI), is a parameter-free version of SVI with the minibatch size being the only user input required.

4.5.1 Coupling the sampling distribution with the learning rate

As we have shown, adaptive sampling distributions aim at reducing the variance of the noisy gradient by biasing the minibatch sampling towards the observations that have a higher signal-to-noise ratio. To capitalize on these variance gains, the optimizer requires an adaptive step-size method that adjusts the learning rate depending on how noisy the gradient is. We may then wonder whether one can couple the optimal minibatch sampling distribution in Section 4.3 with the optimal learning rate from Section 4.4.

A natural place to start is the following joint optimization problem:

$$\min_{\rho_t, \boldsymbol{p}_t} \mathbb{E}_{\boldsymbol{p}_t} \left\| \boldsymbol{\lambda}_{t+1}(\rho_t, \boldsymbol{p}_t) - \boldsymbol{\lambda}_t^* \right\|^2 . \tag{4.39}$$

In Eq. (4.39), the learning rate $\boldsymbol{\rho}_t$ and the sampling distribution $\boldsymbol{p}_t$ are chosen so as to jointly minimize the distance between the SVI update $\boldsymbol{\lambda}_{t+1}$ and its full-batch counterpart $\boldsymbol{\lambda}_t^*$.

It can be shown that the problem in Eq. (4.39) can be rewritten as:

$$\min_{\rho_t} \left\{ (1-\rho_t)^2 \left\| \boldsymbol{\lambda}_t - \boldsymbol{\lambda}_t^* \right\|^2 + \rho_t^2 \min_{\boldsymbol{p}_t} \mathbb{E}_{\boldsymbol{p}_t} \, ||\hat{\boldsymbol{\lambda}}_t(\boldsymbol{p}_t) - \boldsymbol{\lambda}_t^*||^2 \right\} . \tag{4.40}$$

It follows from Eq. (4.40) that the joint problem can be separated in two subproblems, one of which depends only on the sampling distribution $\boldsymbol{p}$. To solve Eq. (4.39), we optimize the subproblem in the sampling distribution first and then recover the optimal learning rate.

The solution to Eq. (4.40) is:

$$\boldsymbol{p}_t^* = \frac{\left(||\mathbb{E}_{\boldsymbol{\varphi}_i^*} t(x_i, z_i)||\right)_{i\in[n]}}{\sum_{i\in[n]} ||\mathbb{E}_{\boldsymbol{\varphi}_i^*} t(x_i, z_i)||}, \tag{4.41}$$

$$\rho_t^* = \frac{\left\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_t^*\right\|^2}{\left\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_t^*\right\|^2 + \text{tr}\ \Sigma^*}, \tag{4.42}$$

where $\Sigma^* = \mathbb{E}||\hat{\boldsymbol{g}}_t(\boldsymbol{p}_t^*) - \boldsymbol{g}_t^*||^2$.

Eq.s (4.41) and (4.42) couple the optimal minibatch sampling distribution with the optimal learning rate. Together, they provide a recipe for a parameter-free stochastic optimization, where the only decision required by the user is the minibatch size.

The optimal sampling distribution in Eq. (4.41) is the same as that arising from the variance minimization problem in Section 4.3. This comes as no surprise after noting that minimizing the distance between the intermediate update $\hat{\boldsymbol{\lambda}}$ and the full batch update $\boldsymbol{\lambda}^*$ is equivalent to minimizing the variance of the noisy gradient, when optimizing with respect to $\boldsymbol{p}$. More specifically, we have:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{p}_t}||\hat{\boldsymbol{\lambda}}_t(\boldsymbol{p}_t) - \boldsymbol{\lambda}_t^*||^2 &= \mathbb{E}_{\boldsymbol{p}_t}||\hat{\boldsymbol{\lambda}}_t(\boldsymbol{p}_t) - \boldsymbol{\lambda}_t - (\boldsymbol{\lambda}_t^* - \boldsymbol{\lambda}_t)||^2 \\
&= \mathbb{E}_{\boldsymbol{p}_t}||\hat{g}_t(\boldsymbol{\lambda}_t; \boldsymbol{p}_t) - g_t(\boldsymbol{\lambda}_t)||^2 \\
&= V(\hat{g}_t(\boldsymbol{\lambda}_t; \boldsymbol{p}_t)).
\end{aligned} \tag{4.43}$$

4.5.2 The algorithm

We present adaptive SVI (AdaSVI). This method combines the importance sampling techniques in Section 4.3 with the adaptive learning rate from Section 4.4. As shown in the previous section, the learning rate and the minibatch sampling distribution are chosen so as to minimize the distance of the stochastic update from the full batch update.

Since we presented two different importance sampling strategies, we consider two versions of AdaSVI. One approximates the oracle distribution with an iteration-independent envelope; the

other samples directly from the oracle via rejection sampling but requires estimating the oracle normalizing constant. Both strategies employ the adaptive learning rate.

AdaSVI proceeds as follows. At each iteration, we sample a minibatch with replacement from either the envelope or the oracle itself via rejection sampling. We update the local variational parameters for the points in the minibatch and compute the importance-weighted noisy gradient in Eq. (4.5). We update the moving averages of the gradient and of its second moment, which we then use to compute the adaptive learning rate. We update the global variational parameters. The full algorithm for envelope-based AdaSVI is stated in Algorithm 4.2. For the rejection-sampling based version of the algorithm, we replace lines 2-4 of Algorithm 4.2 with the sampling procedure stated in Algorithm 4.1.

4.6 Numerical studies

We study AdaSVI on the Poisson factorization model stated in Section 4.3.2.

Our goal is to compare envelope-sampling and rejection-sampling AdaSVI to SVI in terms of speed of convergence and inference.

We find that envelope-based AdaSVI converges faster than the other methods. Surprisingly, AdaSVI with rejection sampling results in even slower convergence than SVI. We argue that this is due to the additional noise introduced by estimating the oracle normalizing constant. The experiment details, results, and plots follow below.

Data We simulate data from the Poisson factorization model. Recall from the discussion on importance scores from Section 4.3.2 that user (/item) importance scores are driven by two factors: the user specialization and total activity level.

Algorithm 4.2 AdaSVI with envelope minibatch sampling

Input: normalized envelope $\boldsymbol{p}^e$, model $p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\beta})$, data $\boldsymbol{x}$, minibatch size m.

Output: global variational densities $q_{\boldsymbol{\lambda}}(\boldsymbol{\beta})$.

Initialize: global variational parameters $\boldsymbol{\lambda}_0$; moving average $\bar{g}_0$ for the natural gradient and $\bar{h}_0$ for its second moment; window size τ_0 of moving average (see Section 4.4 for details).

1: **for** $t = 1, 2, \ldots$ **do**

2: Sample with replacement a minibatch S of size m from the envelope.

3: Optimize the local variational parameters φ_i^* for any $i \in S$.

4: Compute the importance-weighted noisy gradient:

$$\hat{\boldsymbol{g}}_t = \boldsymbol{\alpha} + \left[\frac{1}{m} \sum_{i \in S} (p_i^e)^{-1} \mathbb{E}_{\varphi_i^*} t(z_i, x_i), n \right] - \boldsymbol{\lambda}.$$

5: Update the moving averages:

$$\bar{\boldsymbol{g}}_t = (1 - \tau_t^{-1}) \bar{g}_{t-1} + \tau_t^{-1} \hat{\boldsymbol{g}}_t,$$
$$\bar{h}_t = (1 - \tau_t^{-1}) \bar{h}_{t-1} + \tau_t^{-1} \hat{\boldsymbol{g}}_t^\top \hat{\boldsymbol{g}}_t.$$

6: Compute the learning rate:

$$\rho_t = \bar{\boldsymbol{g}}_t^\top \bar{\boldsymbol{g}}_t / \bar{h}_t.$$

7: Update the window size:

$$\tau_{t+1} = \tau_t (1 - \rho_t) + 1.$$

8: Update the global variational parameters:

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \rho_t \hat{\boldsymbol{g}}_t.$$

To generate heterogenous data, we then divide the users into five classes with varying levels of specialization over $K = 50$ attributes. To this end, we fix the users activity level to $10K$ and generate their attribute assignments from a symmetric Dirichlet distribution:

$$\frac{\boldsymbol{\beta}_u}{10K} \sim \text{Symmetric Dirichlet}_K(\eta_{\sigma(u)}), \quad u = 1, \ldots, 10,000, \tag{4.44}$$

where $\sigma(u) \in \{1, 2, \ldots, 5\}$ denotes the specialization class of user u and where we let $\eta_c = 10^{c-2}$, $c = 1, \ldots, 5$. By varying the concentration parameter η_c, we generate different levels of specialization: when $\eta = 0.1$ users consume items almost exclusively via a single attribute; when $\eta = 100$ users have no preferences over the attributes. We generate 100,000 users. Finally, to preserve the identifiability of the model, we generate $1,000$ items with perfect specialization. We randomly assign them to the 50 attributes and set their total activity levels to 1.

Experiment details We run the four following versions of SVI, each with a different minibatch sampling distribution: oracle sampling; envelope sampling; rejection sampling; and uniform sampling (i.e. the benchmark vanilla SVI by Hoffman et al. 2013).

As discussed in the previous sections, to benefit from the variance gains introduced by importance sampling, the weighted minibatch sampling distributions (first three methods) were paired with the adaptive learning rate from Section 4.4. For the original SVI formulation, we adopted an exponentially decaying learning rate satisfying Robbins-Monro conditions:

$$\rho_t = (t + T)^{-f}. \tag{4.45}$$

The forgetting rate $f \in (0.5, 1]$ and the delay T were chosen with a grid search, resulting in $f = 0.51$ and $T = 1,000$.

We set the Gamma shape and rate hyperparameters equal to 1 for all users and items. We initialized the user and item variational parameters from the Gamma priors. For all methods, we set the batch size to 1%.

Figure 4.1 and 4.2 track the progress of the ELBO during training. Figure 4.1 shows the ELBO by iteration. While not meaningful for computational purposes, this plots shows the information efficiency of each method and how they compare to the oracle. Figure 4.2 plots the ELBO by runtime, illustrating the actual speed of convergence of each method.

The results in Figure 4.1 refer to a a smaller version of the data with only 1,000 users, 10 items, and 10 attributes. The limited size of the experiment is required to fit the model via AdaSVI with oracle sampling, which is computationally prohibitive.

Figure 4.3 shows the progress of AdaSVI with oracle sampling on each of the five subsets of the data.

Results AdaSVI with envelope sampling converges faster than SVI (Figure 4.2). Also, in terms of progress per iteration, sampling from the envelope is comparable to the oracle. As shown in Figure 4.2, the speed of convergence of these two methods is comparable when measured by iteration. This similarity in performance is surprising for the envelope sampling probabilities do not account for the users specialization levels, but only adjust for the user activity levels. It indicates that accounting for the activity levels is enough to benefit from the convergence gains introduced by the oracle.

AdaSVI with rejection sampling results in much slower convergence than the other methods (Figure 4.1). Recall that the only difference between AdaSVI with rejection sampling and the oracle is that the former method requires estimating the oracle normalizing constant. The results in Figure 4.1 show that the noise introduced by this additional estimation step exceeds the variance savings of the oracle. This extra noise negatively impacts the speed of convergence leading to even slower progress than uniform random sampling.

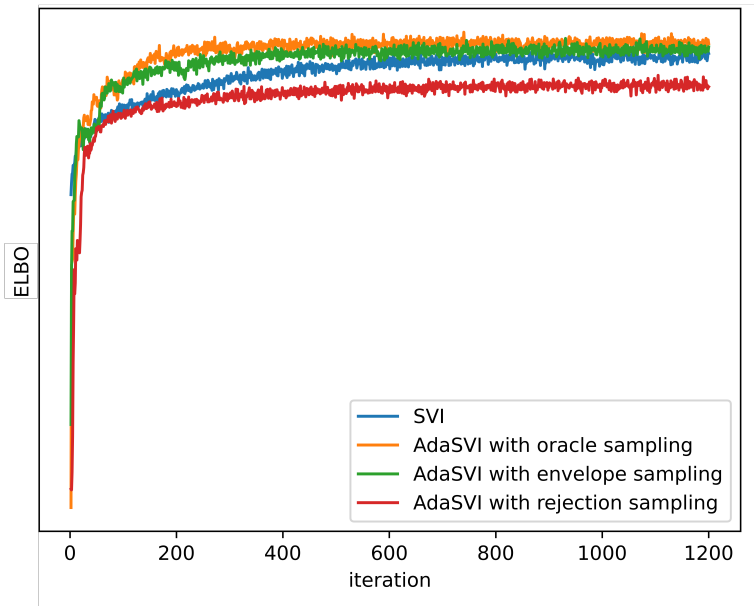


Figure 4.1: ELBO vs. iteration. The progress of AdaSVI with envelope sampling is comparable to the oracle. Conversely, the added noise in the rejection-sampling version makes AdaSVI even less efficient than the benchmark vanilla SVI.

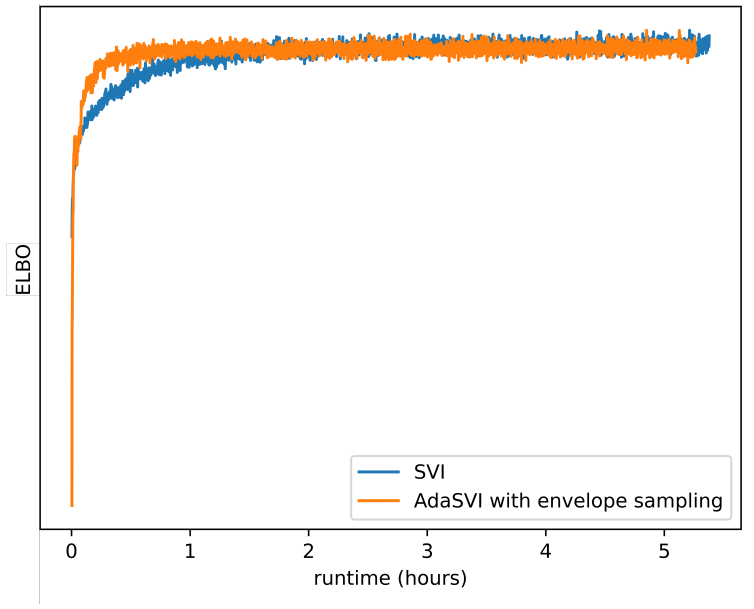


Figure 4.2: ELBO vs. runtime. AdaSVI with envelope sampling converges faster than the benchmark vanilla SVI.

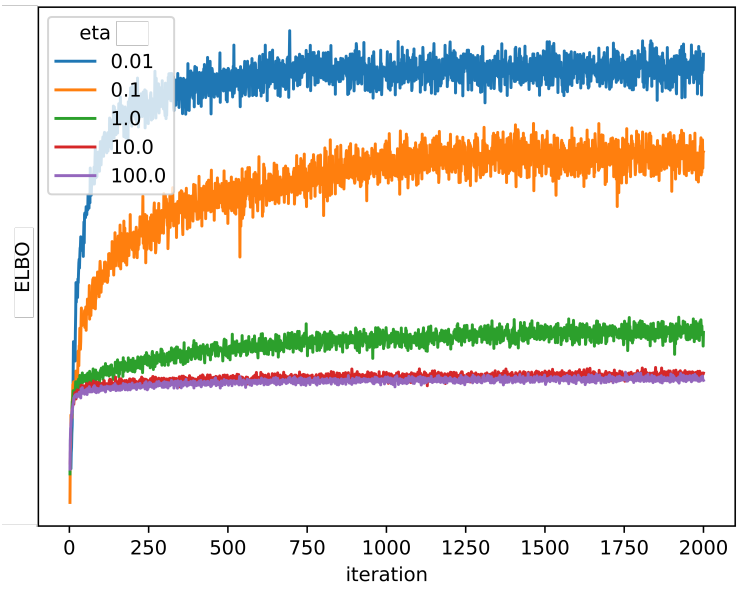


Figure 4.3: Training progress of AdaSVI with oracle sampling (ELBO vs. iteration) for different concentration parameters. Convergence is faster when users are more specialized (lower concentration).

4.7 Discussion

We developed AdaSVI, a stochastic variational inference algorithm with non-uniform mini-batch sampling paired with an adaptive learning rate. At each iteration, AdaSVI biases the mini-batch sampling distribution toward the observations with a higher signal-to-noise ratio. AdaSVI navigates through large-scale datasets more efficiently, achieving faster convergence on synthetic data.

While AdaSVI can be applied to any conditional conjugate model, it does not yet extend to other successful applications of VI, such as black-box variational inference methods (Ranganath, Gerrish, and Blei, 2014). A natural continuation of this work is to assess up to what extent similar techniques can be applied to variational methods for non-conjugate models (Ruiz, Titsias, and Blei, 2016; Agrawal, Sheldon, and Domke, 2020).

4.8 Proofs

First, we show that the noisy gradient is unbiased and derive an expression for the trace of its variance-covariance matrix (Eq. 4.6). Next, we show that, under sampling with replacement, the variance minimization problem simplifies significantly (Eq. 4.7). Finally, we provide a closed form of the resulting optimal sampling weights (Eq. 4.8). We follow Zhao and Zhang (2015), who derive the optimal sampling weights for SGD under a regularized loss and with a minibatch of size 1. We apply this framework to SVI and extend it to a minibatch of size m under sampling with replacement.

Notation Recall that t denotes the sufficient statistic of the conditional joint distribution of the i-th local context (x_i, z_i) given the global parameters $\boldsymbol{\beta}$. We assume that t is D-dimensional. Also, we let $\mathcal{S}$ denote the sample space of all possible minibatches of size m, and let $\mathcal{P}$ be a probability distribution over $\mathcal{S}$. We define a sampling strategy as a pair $(\mathcal{S}, \mathcal{P})$. For example, under sampling with replacement, $\mathcal{S}$ is the set of all the possible permutations of the data $(x_i)_{i\in[n]}$ of length m and $\mathcal{P}$ is univocally defined by the sampling weights $(p_i)_{i\in[n]}$.

Proof of Eq. (4.6)**.** We have:

$$\begin{aligned}
\mathbb{E}_S\left[\frac{1}{m}\sum_{i\in S} p_i^{-1}\mathbb{E}_{\varphi_i^*}t(z_i,x_i)\right] &= \frac{1}{m}\sum_{S\in\mathcal{S}}\mathcal{P}(S)\sum_{i\in S} p_i^{-1}\mathbb{E}_{\varphi_i^*}t(z_i,x_i) \\
&= \frac{1}{m}\sum_{i}\left(\sum_{S:i\in S} M_i(S)\mathcal{P}(S)\right) p_i^{-1}\mathbb{E}_{\varphi_i^*}t(z_i,x_i) \\
&= \frac{1}{m}\sum_{i}\mathbb{E}_S M_i\, p_i^{-1}\mathbb{E}_{\varphi_i^*}t(z_i,x_i), \qquad (4.46)
\end{aligned}$$

where we define the random variable M_i as the number of times i is sampled in the minibatch. Under sampling with replacement with probabilities $\boldsymbol{p}$, $(M_i)_{i\in[n]} \sim \text{Multinomial}(m, \boldsymbol{p})$ and thus $\mathbb{E}_S M_i = mp_i$. Plugging this into Eq. (4.46) yields that $\mathbb{E}_S\hat{g} = g$, and so $\hat{g}$ is an unbiased estimator

of g. As a result, the variance of the noisy gradient reduces to its expected squared norm:

$$V(\hat{g})(\boldsymbol{\lambda}) = \mathbb{E}_S \left|\left|\hat{g}(\boldsymbol{\lambda}) - g(\boldsymbol{\lambda})\right|\right|^2 = \sum_d \mathbb{E}_S \left(\hat{g}_d(\boldsymbol{\lambda}) - g_d(\boldsymbol{\lambda})\right)^2 = \mathbb{E}_S \left|\left|\hat{g}_d(\boldsymbol{\lambda})\right|\right|^2 . \tag{4.47}$$

Proof of Eq.s (4.7) **and** (4.8)**.**

$$\begin{aligned}
\mathbb{E}_S \left|\left|\hat{g}_d(\boldsymbol{\lambda})\right|\right|^2 &= \frac{1}{m^2}\mathbb{E}_S \left[\sum_{i\in S} p_i^{-2} \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|^2 + \sum_d \sum_{i,j\in S: i\neq j} (p_i p_j)^{-1} \mathbb{E}_{\varphi_i^*} t_d(z_i, x_i) \mathbb{E}_{\varphi_j^*} t_d(z_j, x_j) \right] \\
&= \frac{1}{m^2}\Bigg[\sum_{i\in[n]} (\mathbb{E}_S M_i) p_i^{-2} \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|^2 + \\
&\qquad \sum_{i,j: i\neq j} (\mathbb{E}_S M_i M_j)(p_i p_j)^{-1} \sum_d \mathbb{E}_{\varphi_i^*} t_d(z_i, x_i) \mathbb{E}_{\varphi_j^*} t_d(z_j, x_j) \Bigg].
\end{aligned} \tag{4.48}$$

Since $\mathbb{E}M_i = mp_i$ and $\mathbb{E}M_i M_j = m(m-1)p_i p_j$, Eq. (4.48) yields:

$$\min_{\boldsymbol{p}} V(\hat{g}) \iff \min_{\boldsymbol{p}} \frac{1}{m} \sum_i p_i^{-1} \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|^2 . \tag{4.49}$$

An application of Cauchy-Schwarz inequality yields:

$$\begin{aligned}
\min_{\boldsymbol{p}} \sum_i p_i^{-1} \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|^2 &= \min_{\boldsymbol{p}} \sum_i p_i \sum_i p_i^{-1} \left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|^2 \\
&\geq \sum_i \left\| \mathbb{E}_{\varphi_i^*} t_d(z_i, x_i) \right\|,
\end{aligned} \tag{4.50}$$

which does not depend on $\boldsymbol{p}$. The equality holds if and only if $\boldsymbol{p}$ is proportional to $(\left\| \mathbb{E}_{\varphi_i^*} t(z_i, x_i) \right\|)_{i\in[n]}$.

Conclusions

We conclude by summarizing the contributions of this dissertation, its limitations, and directions of future work.

We first presented a novel generator for massive networks with overlapping community structure. While the predominant approach in the literature has been algorithmic and it has mostly focused on reproducing power-law degree distributions, our generator builds on a principled random graph model (Ball, Karrer, and Newman, 2011) and it can be used to target a local or a global centrality measure – namely the degree centrality and the eigencentrality. The flexibility of our method is certified by our theoretical results, which guarantee that under some mild conditions there always exist some model parameters that enforce an arbitrary centrality target. In practice, our generator can produce a wide spectrum of network topologies with an underlying overlapping community structure.

A fundamental assumption behind our generator is that of node types. From a computational perspective, partitioning the networks into groups of stochastically equivalent nodes reduces the complexity of the problem. This allows us to recover the solutions of the degree and the eigencentrality problem via a multivariate Newton-Raphson (MNR) algorithm, which proves effective in solving sparse systems with up to 10^4 variables. However, as the complexity of networks scales with their size, if we were to extend our generator to even larger settings (e.g. terascale networks), we may want to solve multivariate systems of quadratic polynomials with hundreds of thousands – if not millions – of variables. While such systems are not feasible for MNR, they could be tackled with quasi-Newton methods. To further scale up our generator, a parallel implementation of the edge generation step is required. We leave these directions of research to future work.

Next, we focused on large-scale approximate inference and developed AdaSVI, a stochastic variational inference algorithm that combines an adaptive learning rate with weighted minibatch sampling. AdaSVI reweights the sampling distribution of the minibatch to minimize the variance of the natural gradient. AdaSVI navigates through large-scale datasets more efficiently, achieving faster convergence and better predictions.

The only user input required by AdaSVI is the minibatch size. However, this parameter could also be chosen adaptively. Recent works in the stochastic optimization literature propose increasing batch-size schedules to further facilitate convergence (Devarakonda, Naumov, and Garland, 2017; Zaheer et al., 2018). A natural continuation of this work is to extend these results to the stochastic variational objective.

Another direction of future research concerns the applicability of the algorithm. While AdaSVI can be applied to any conditional conjugate model, it does not yet extend to other successful applications of VI, such as black-box variational inference methods (Ranganath, Gerrish, and Blei, 2014). In future work, we seek to assess up to what extent similar techniques can be applied to variational methods for non-conjugate models (Ruiz, Titsias, and Blei, 2016; Agrawal, Sheldon, and Domke, 2020).

References

Agrawal, Abhinav, Daniel R Sheldon, and Justin Domke (2020). "Advances in Black-Box VI: Normalizing Flows, Importance Weighting, and Optimization". In: Advances in Neural Information Processing Systems. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 17358–17369. URL: https://proceedings.neurips.cc/paper/2020/file/c91e3483cf4f90057d02aa492d2b25b1-Paper.pdf.

Ayed, Fadhel and François Caron (2021). "Nonnegative Bayesian nonparametric factor models with completely random measures". In: Statistics and Computing 31.5, pp. 1–24.

Ball, Brian, Brian Karrer, and M. E. J. Newman (Sept. 2011). "Efficient and principled method for detecting communities in networks". In: Physical Review E 84 (3), p. 036103. DOI: 10.1103/PhysRevE.84.036103. URL: https://link.aps.org/doi/10.1103/PhysRevE.84.036103.

Barabási, Albert-László and Réka Albert (1999). "Emergence of scaling in random networks". In: Science 286.5439, pp. 509–512.

Basu, Saugata, Richard Pollack, and Marie-Françoise Roy (2006). Algorithms in Real Algebraic Geometry. Vol. 19. Springer-Verlag Berlin Heidelberg, pp. 22–29.

Bates, Douglas and Dirk Eddelbuettel (2013). "Fast and elegant numerical linear algebra using the RcppEigen package". In: Journal of Statistical Software 52, pp. 1–24.

Bihan, Frédéric, Alicia Dickenstein, and Magalı Giaroli (2020). "Sign conditions for the existence of at least one positive solution of a sparse polynomial system". In: Advances in Mathematics 375, p. 107412.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (Apr. 2017). "Variational Inference: A Review for Statisticians". In: Journal of the American Statistical Association 112 (518), pp. 859–877. ISSN: 1537274X. DOI: 10.1080/01621459.2017.1285773/SUPPL_FILE/UASA_A_1285773_SM4869.PDF. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.2017.1285773.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent Dirichlet Allocation". In: Journal of Machine Learning Research 3, pp. 993–1022. DOI: 10.5555/944919.

Bochnak, Jacek, Michel Coste, and Marie-Françoise Roy (2013). Real algebraic geometry. Vol. 36. Springer Science & Business Media.

Bottou, Léon et al. (1998). “Online learning and stochastic approximations”. In: On-line learning in neural networks 17.9, p. 142.

Boucheron, Stéphane, Gábor Lugosi, and Pascal Massart (2013). Concentration inequalities: A nonasymptotic theory of independence. Oxford University Press.

Casella, George and Christian P Robert (1996). “Rao-Blackwellisation of sampling schemes”. In: Biometrika 83.1, pp. 81–94.

Cemgil, Ali Taylan (2008). “Bayesian inference for nonnegative matrix factorisation models”. In: Computational Intelligence and Neuroscience 2009.

Csiba, Dominik, Zheng Qu, and Peter Richtarik (July 2015). “Stochastic Dual Coordinate Ascent with Adaptive Probabilities”. In: Proceedings of the 32nd International Conference on Machine Learning. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: Proceedings of Machine Learning Research, pp. 674–683. URL: https://proceedings.mlr.press/v37/csiba15.html.

Csiba, Dominik and Peter Richtárik (Jan. 2018). “Importance Sampling for Minibatches”. In: Journal of Machine Learning Research 19.1, pp. 962–982. ISSN: 1532-4435.

Decker, Wolfram et al. (2021). SINGULAR 4-2-1 — A computer algebra system for polynomial computations. https://www.singular.uni-kl.de.

Devarakonda, Aditya, Maxim Naumov, and Michael Garland (2017). “Adabatch: Adaptive batch sizes for training deep neural networks”. In: arXiv preprint arXiv:1712.02029.

Duchi, John, Elad Hazan, and Yoram Singer (2011). “Adaptive subgradient methods for online learning and stochastic optimization.” In: Journal of Machine Learning Research 12.7.

Dunson, David B and Amy H Herring (2005). “Bayesian latent variable models for mixed discrete outcomes”. In: Biostatistics 6.1, pp. 11–25.

Durante, Daniele and Tommaso Rigon (2019). “Conditionally Conjugate Mean-Field Variational Bayes for Logistic Models”. In: Statistical Science 34.3, pp. 472–485. DOI: 10.1214/19-STS712. URL: https://doi.org/10.1214/19-STS712.

Fagnan, Justin et al. (2018). “Modular Networks for Validating Community Detection Algorithms”. In: arXiv preprint arXiv:1801.01229.

Feige, Uriel and Eran Ofek (2005). “Spectral techniques applied to sparse random graphs”. In: Random Structures & Algorithms 27.2, pp. 251–275.

Feld, Scott L (1981). “The focused organization of social ties”. In: American Journal of Sociology 86.5, pp. 1015–1035.

Fu, Tianfan and Zhihua Zhang (20–22 Apr 2017). “CPSG-MCMC: Clustering-Based Preprocessing method for Stochastic Gradient MCMC”. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. Proceedings of Machine Learning Research, pp. 841–850. URL: https://proceedings.mlr.press/v54/fu17a.html.

Girvan, Michelle and Mark EJ Newman (2002). “Community structure in social and biological networks”. In: Proceedings of the National Academy of Sciences 99.12, pp. 7821–7826.

Gopalan, Prem, Jake M Hofman, and David M Blei (2015). “Scalable Recommendation with Hierarchical Poisson Factorization.” In: Uncertainty in Artificial Intelligence, pp. 326–335.

Gopalan, Prem, Francisco J Ruiz, et al. (2014). “Bayesian nonparametric Poisson factorization for recommendation systems”. In: Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, pp. 275–283.

Gopalan, Prem K. and David M. Blei (2013). “Efficient discovery of overlapping communities in massive networks”. In: Proceedings of the National Academy of Sciences 110.36, pp. 14534–14539. ISSN: 0027-8424. DOI: 10.1073/pnas.1221839110. eprint: https://www.pnas.org/content/110/36/14534.full.pdf. URL: https://www.pnas.org/content/110/36/14534.

Guennebaud, Gaël, Benoît Jacob, et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

Hamann, Michael et al. (2018). “I/O-efficient generation of massive graphs following the LFR benchmark”. In: The ACM Journal of Experimental Algorithmics 23, pp. 1–33.

Hensman, James, Nicolò Fusi, and Neil D. Lawrence (2013). “Gaussian Processes for Big Data”. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence. UAI’13. Bellevue, WA: AUAI Press, pp. 282–290.

Hinton, Geoffrey, Nitish Srivastava, and Kevin Swersky (2012). “Lecture 6a: overview of mini-batch gradient descent”. In: Lectures on neural networks for machine learning.

Hoffman, Matthew D. et al. (2013). “Stochastic Variational Inference”. In: Journal of Machine Learning Research 14.4, pp. 1303–1347. URL: http://jmlr.org/papers/v14/hoffman13a.html.

Huang, Sihan and Yang Feng (July 2018). “Pairwise Covariates-adjusted Block Model for Community Detection”. In: arXiv preprint arXiv:1807.03469. DOI:

10.48550/arxiv.1807.03469. URL: https://arxiv.org/abs/1807.03469v2.

Jordan, Michael I et al. (1999). "An introduction to variational methods for graphical models". In: Machine learning 37.2, pp. 183–233.

Kamiński, Bogumił, Paweł Prałat, and François Théberge (2021). "Artificial Benchmark for Community Detection (ABCD)–Fast random graph model with community structure". In: Network Science, pp. 1–26.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: arXiv preprint arXiv:1412.6980.

Kolda, Tamara G et al. (2014). "A scalable generative graph model with community structure". In: SIAM Journal on Scientific Computing 36.5, pp. C424–C452.

Lancichinetti, Andrea and Santo Fortunato (July 2009). "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities". In: Physical Review E 80 (1), p. 016118. DOI: 10.1103/PhysRevE.80.016118. URL: https://link.aps.org/doi/10.1103/PhysRevE.80.016118.

Lancichinetti, Andrea, Santo Fortunato, and Filippo Radicchi (2008). "Benchmark graphs for testing community detection algorithms". In: Physical Review E 78.4, p. 046110.

Lei, Jing and Alessandro Rinaldo (Feb. 2015). "Consistency of spectral clustering in stochastic block models". In: Annals of Statistics 43.1, pp. 215–237. DOI: 10.1214/14-AOS1274. URL: https://doi.org/10.1214/14-AOS1274.

Levitin, Hanna Mendes et al. (2019). "De novo gene signature identification from single-cell RNA-seq with hierarchical Poisson factorization". In: Molecular systems biology 15.2, e8557.

Liang, Dawen, John W Paisley, Dan Ellis, et al. (2014). "Codebook-based Scalable Music Tagging with Poisson Matrix Factorization." In: The International Society for Music Information Retrieval Conference, pp. 167–172.

Maslov, Sergei and Kim Sneppen (2002). "Specificity and stability in topology of protein networks". In: Science 296.5569, pp. 910–913.

Needell, Deanna, Rachel Ward, and Nati Srebro (2014). "Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm". In: Advances in Neural Information Processing Systems. Vol. 27. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/file/f29c21d4897f78948b91f03172341b7b-Paper.pdf.

Newman, M. E. J. (2010). Networks: an Introduction. Oxford University Press. ISBN: 199206650.

Newman, Mark EJ (2002). “Assortative mixing in networks”. In: Physical Review Letters 89.20, p. 208701.

Newman, Mark EJ and Juyong Park (2003). “Why social networks are different from other types of networks”. In: Physical review E 68.3, p. 036122.

Ortega, James M (1968). “The Newton-Kantorovich theorem”. In: The American Mathematical Monthly 75.6, pp. 658–660.

Paisley, John, David M. Blei, and Michael I. Jordan (2012). “Variational Bayesian Inference with Stochastic Search”. In: Proceedings of the 29th International Coference on International Conference on Machine Learning. ICML’12. Edinburgh, Scotland: Omnipress, pp. 1363–1370. ISBN: 9781450312851.

Paisley, John W, David M Blei, and Michael I Jordan (2014). “Bayesian Nonnegative Matrix Factorization with Stochastic Variational Inference.” In: Handbook of Mixed Membership Models and Their Applications. Chapman & Hall/CRC.

Palla, Gergely et al. (2005). “Uncovering the overlapping community structure of complex networks in nature and society”. In: Nature 435.7043, pp. 814–818.

Papa, Guillaume, Pascal Bianchi, and Stéphan Clémençon (2015). “Adaptive Sampling for Incremental Optimization Using Stochastic Gradient Descent”. In: Algorithmic Learning Theory.

Perekrestenko, Dmytro, Volkan Cevher, and Martin Jaggi (20–22 Apr 2017). “Faster Coordinate Descent via Adaptive Importance Sampling”. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. Proceedings of Machine Learning Research, pp. 869–877. URL: https://proceedings.mlr.press/v54/perekrestenko17a.html.

Ranganath, Rajesh, Sean Gerrish, and David Blei (2014). “Black Box Variational Inference”. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics. Ed. by Samuel Kaski and Jukka Corander. Vol. 33. Proceedings of Machine Learning Research. Reykjavik, Iceland: Proceedings of Machine Learning Research, pp. 814–822. URL: https://proceedings.mlr.press/v33/ranganath14.html.

Ranganath, Rajesh, Chong Wang, et al. (2013). “An adaptive learning rate for Stochastic Variational Inference”. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 298–306.

Rao, Calyampudi R. (1945). “Information and Accuracy Attainable in the Estimation of Statistical Parameters”. In: Bulletin of the Calcutta Mathematical Society 37, pp. 81–91.

Robbins, Herbert and Sutton Monro (Sept. 1951). "A Stochastic Approximation Method". In: The Annals of Mathematical Statistics 22.3, pp. 400–407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

Ross, Sheldon (2013). Simulation. 5th Edition. Academic Press. ISBN: 978-0-12-415825-2. DOI: https://doi.org/10.1016/B978-0-12-415825-2.00015-2. URL: https://www.sciencedirect.com/science/article/pii/B9780124158252000152.

Ruiz, Francisco J. R., Michalis K. Titsias, and David M. Blei (2016). "Overdispersed Black-Box Variational Inference". In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence. UAI'16. Jersey City, New Jersey, USA: AUAI Press, pp. 647–656. ISBN: 9780996643115.

Schaul, Tom, Sixin Zhang, and Yann LeCun (2013). "No more pesky learning rates". In: International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 343–351.

Slota, George M et al. (2019). "Scalable generation of graphs for benchmarking HPC community-detection algorithms". In: Proceedings of the International Conference for High Performance Computing, pp. 1–14.

Staudt, Christian L, Aleksejs Sazonovs, and Henning Meyerhenke (2016). "NetworKit: A tool suite for large-scale complex network analysis". In: Network Science 4.4, pp. 508–530.

Stich, Sebastian U, Anant Raj, and Martin Jaggi (2017). "Safe Adaptive Importance Sampling". In: Advances in Neural Information Processing Systems. Vol. 30. URL: https://proceedings.neurips.cc/paper/2017/file/1177967c7957072da3dc1db4ceb30e7a-Paper.pdf.

Virtanen, Tuomas, A Taylan Cemgil, and Simon Godsill (2008). "Bayesian extensions to non-negative matrix factorisation for audio signal modelling". In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1825–1828.

Wainwright, Martin J and Michael I Jordan (2008). "Graphical models, exponential families, and variational inference". In: Foundations and Trends® in Machine Learning 1.1–2, pp. 1–305.

Walker, Alastair J. (Sept. 1977). "An Efficient Method for Generating Discrete Random Variables with General Distributions". In: ACM Transactions on Mathematical Software 3.3, pp. 253–256. ISSN: 0098-3500. DOI: 10.1145/355744.355749. URL: https://doi.org/10.1145/355744.355749.

Wong, C. K. and M. C. Easton (1980). “An Efficient Method for Weighted Sampling without Replacement”. In: SIAM Journal on Computing 9.1, pp. 111–113. DOI: 10.1137/0209009. eprint: https://doi.org/10.1137/0209009. URL: https://doi.org/10.1137/0209009.

Yang, Jaewon and Jure Leskovec (Apr. 2014). “Structure and Overlaps of Ground-Truth Communities in Networks”. In: ACM Transactions on Intelligent Systems and Technology 5.2. ISSN: 2157-6904. DOI: 10.1145/2594454. URL: https://doi.org/10.1145/2594454.

Yu, Yi, Tengyao Wang, and Richard J Samworth (2015). “A useful variant of the Davis–Kahan theorem for statisticians”. In: Biometrika 102.2, pp. 315–323.

Zaheer, Manzil et al. (2018). “Adaptive Methods for Nonconvex Optimization”. In: Advances in Neural Information Processing Systems. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf.

Zeiler, Matthew D (2012). “Adadelta: an adaptive learning rate method”. In: arXiv preprint arXiv:1212.5701.

Zhang, Cheng, Hedvig Kjellstrom, and Stephan Mandt (2017). “Determinantal Point Processes for Mini-Batch Diversification”. In: Proceedings of the Thirthy-Third Conference on Uncertainty in Artificial Intelligence. UAI’17. DOI: 10.48550/ARXIV.1705.00607. URL: https://arxiv.org/abs/1705.00607.

Zhao, Peilin and Tong Zhang (2014). “Accelerating Minibatch Stochastic Gradient Descent using Stratified Sampling”. In: arXiv preprint arXiv:1405.3080. DOI: 10.48550/ARXIV.1405.3080. URL: https://arxiv.org/abs/1405.3080.

— (2015). “Stochastic optimization with importance sampling for regularized loss minimization”. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, pp. 1–9.

Zhou, Mingyuan (2015). “Infinite edge partition models for overlapping community detection and link prediction”. In: Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, pp. 1135–1143.