Journal of Machine Learning Research 24 (2023) 1-51 Submitted 6/21; Revised 7/22; Published 3/23

Prior Specification for Bayesian Matrix Factorization via Prior Predictive Matching

Eliezer de Souza da Silva ELIEZER.SOUZA.SILVA@NTNU.NO
Department of Computer Science
Norwegian University of Science and Technology, Trondheim, Norway
School of Applied Mathematics
Fundação Getulio Vargas, Rio de Janeiro, Brazil

Tomasz Kuśmierczyk TOMASZ.KUSMIERCZYK@HELSINKI.FI
Department of Computer Science
University of Helsinki, Helsinki, Finland

Marcelo Hartmann MARCELO.HARTMANN@HELSINKI.FI
Department of Computer Science
University of Helsinki, Helsinki, Finland

Arto Klami ARTO.KLAMI@HELSINKI.FI
Department of Computer Science
University of Helsinki, Helsinki, Finland

Editor: Matthias Seeger

Abstract

The behavior of many Bayesian models used in machine learning critically depends on the choice of prior distributions, controlled by some hyperparameters typically selected through Bayesian optimization or cross-validation. This requires repeated, costly, posterior inference. We provide an alternative for selecting good priors without carrying out posterior inference, building on the prior predictive distribution that marginalizes the model parameters. We estimate virtual statistics for data generated by the prior predictive distribution and then optimize over the hyperparameters to learn those for which the virtual statistics match the target values provided by the user or estimated from (a subset of) the observed data. We apply the principle for probabilistic matrix factorization, for which good solutions for prior selection have been missing. We show that for Poisson factorization models we can analytically determine the hyperparameters, including the number of factors, that best replicate the target statistics, and we empirically study the sensitivity of the approach for the model mismatch. We also present a model-independent procedure that determines the hyperparameters for general models by stochastic optimization and demonstrate this extension in the context of hierarchical matrix factorization models.

Keywords: Bayesian modeling, prior specification, hyperparameter search, probabilistic matrix factorization

1. Introduction

Bayesian machine learning (ML) builds on the idea of general-purpose hierarchical probabilistic models, such as mixture models, topic models or matrix factorization techniques, that can be used for various learning tasks. Often these models have a large number of latent variables, such as the cluster indicators for mixture models or the topic proportions for topic models, for which the prior distributions are chosen predominantly on the basis of computational convenience. The choice of the particular priors, as well as other hyperparameters, such as the number of factors or components, has notable impact on the overall performance of these models, but cannot be determined by classical statistical modeling principles: the priors may not have an intuitive interpretation and for complex hierarchical models the relationship between the priors and data is poorly understood, ruling out subjective prior knowledge. For example, the behavior of the hierarchical Poisson matrix factorization model of Gopalan et al. (2015) depends on seven hyperparameters (six for defining the priors and one for the number of latent factors) in a non-trivial manner.

The hyperparameters are typically chosen heuristically or through an iterative process that explicitly evaluates the quality of multiple choices. The search can be automated with Bayesian optimization (Snoek et al., 2012), but evaluating the quality is costly. The optimization is typically based on some proxy of the marginal likelihood, such as variational lower bound or leave-one-out cross validation (Vehtari et al., 2017), or directly on the performance in a downstream task, such as recommendation (Galuzzi et al., 2019). Both require carrying out posterior inference for every considered set of hyperparameters, which adds significant computational burden and increases the overall training time by orders of magnitude. Furthermore, the result is only optimal for the chosen measure and inference method, unnecessarily tying model specification with inference.

To overcome this, we turn our attention to the statistical literature on the *prior predictive distribution* (PPD) – the marginal distribution of observables before seeing any data. The PPD is routinely used in the statistical modeling pipeline in the form of prior predictive checks, to qualitatively assess whether the model and the priors are reasonable (Schad et al., 2019; Gabry et al., 2019; Gelman et al., 2020). PPD has also been used for prior elicitation, to convert the knowledge an expert has on the properties of data into prior distributions (Kadane et al., 1980; Akbarov, 2009; Hartmann et al., 2020; Mikkola et al., 2021). We turn those ideas into a tool for automatic learning of hyperparameters, by directly optimizing for a good match between *virtual statistics* of the PPD and statistics of the data[1]. The tool can be used in two ways: (1) the target statistics are provided by the expert (user) as prior knowledge on data, or (2) the target statistics are estimated from (a subset of) the actual data. The former is related using PPD for prior elicitation (Kadane et al., 1980), extended here for practical use with Bayesian ML models with a large number of latent variables, whereas the latter is similar to empirical Bayes (Casella, 1985).

The proposed *prior predictive matching* approach, described in Section 3, finds good hyperparameters without needing posterior inference. When the true data generating process is within the assumed model family, the approach provides hyperparameters that are optimal with respect to the selected statistics, and we show empirically that the method is robust to small model misspecification. If the data fits poorly the assumed model family, the approach may return unreasonable choices, which can be interpreted as sign of model mismatch and a need for remodeling. For instance, we show how priors not accounting for non-homogeneous row and column counts in recommender engine setups result in a clear

[1] In order to distinguish from statistics of the observed data, we use the term *virtual statistics* to refer to summary quantities calculated for hypothetical data sampled from PPD, inspired by the phrase *virtual counts* used sometimes for the hyperparameters in count-data models.

underestimation of the number of factors, which could be fixed by changing to a formulation that is consistent with the observed margins (Yıldırım et al., 2021).

The principle of prior predictive matching is generally applicable for all statistical models used in machine learning that specify a sampling distribution. In this work, we introduce prior predictive matching in the context of probabilistic matrix factorization models, focusing in particular on models for count data. This provides a tangible technical context for the work and enables showcasing how the specific instances of the general principle can provide particularly elegant solutions for the practical models of interest. First, we consider scenarios where certain moments of the PPD can be expressed analytically. We can then compute the virtual statistics in closed form and analytically solve for optimal hyperparameters. The method can be interpreted as a specific instance of the method of moments (Casella and Berger, 2001; Pawitan, 2001) since we determine the parameters by matching the moments, but applied for learning the hyperparameters as an intermediate step in the full modelling process, rather than directly as a means of approximate inference (once the hyperparameters are fixed, we still want to perform standard posterior inference). The approach facilitates computationally efficient selection of the hyperparameters, demonstrated in this work for count matrix factorization models for which we can set both the prior parameters and the number of factors in closed form. Previously, the latent dimensionality could be set automatically only for Gaussian matrix factorization, based on the analytic marginal likelihood by Bouveyron et al. (2019). With our method, this can be done also for count models, although based on selected statistics rather than the marginal likelihood.

Closed-form analytic expressions are computationally ideal, but their derivation is tedious already for fairly simple models. To address this, we also provide a model-independent stochastic algorithm that uses sampling to compute the virtual statistics, and stochastic gradient optimization to learn the hyperparameters. It only requires being able to sample from PPD and is applicable for all models with reparameterizable priors (Figurnov et al., 2018; Mohamed et al., 2020). The method is loosely related to several recent methods which directly learn a prior for flexible neural network-based models (Tomczak and Welling, 2018; Klushyn et al., 2019; Nalisnick and Smyth, 2018; Nalisnick et al., 2021; Du et al., 2019), but our goal is specifically to determine the hyperparameters of existing priors and we obtain the solution without considering the observed data. We also demonstrate the efficacy this approach in the context of Bayesian matrix factorization models, this time considering a more complex hierarchical model for which we do not have analytic expressions for the moments.

2. Motivation: Priors for Bayesian Matrix Factorization

Bayesian matrix factorization (BMF) is an important class of Bayesian ML models used, e.g., in recommender engines (Salakhutdinov and Mnih, 2007), for dimensionality reduction (Bai et al., 2013; Xu et al., 2003), in community detection (Psorakis et al., 2011), and modeling relationships between data modalities (Klami et al., 2013). Importantly, it is a family for which the prior distributions are difficult to specify, as will be clarified in Section 2.1. We start by characterizing two concrete models building on Poisson distribution, for which the effect is emphasized (Cemgil, 2009).

Poisson Matrix Factorization (PMF) (Cemgil, 2009; Gopalan et al., 2014) with latent dimensionality K specifies a generative model for a matrix $\mathbf{Y} = \{Y_{ij}\} \in \mathbb{R}^{N \times M}$, with each entry Y_{ij} following a Poisson distribution with rate $\theta_{ik}\beta_{jk}$, a product of latent factors θ_{ik} indexed by the rows and β_{jk} indexed by the columns.

Each latent variable follows a prior $F(\mu, \sigma^2)$, parameterized here using mean μ and standard deviation σ

$$\begin{aligned} &\theta_{ik} \overset{\text{iid}}{\sim} F(\mu_\theta, \sigma_\theta^2), \quad \beta_{jk} \overset{\text{iid}}{\sim} F(\mu_\beta, \sigma_\beta^2), \\ &Y_{ij} \overset{\text{iid}}{\sim} \text{Poisson}\left(\sum_{k=1}^{K} \theta_{ik}\beta_{jk}\right). \end{aligned} \tag{1}$$

The majority of the PMF literature assumes the priors to be gamma distributions (using shape-rate parameterization, $F(\mu_\theta, \sigma_\theta^2) = \text{Gamma}(a, b)$ and $F(\mu_\beta, \sigma_\beta^2) = \text{Gamma}(c, d)$, with $\mu_\theta = \frac{a}{b}$, $\sigma_\theta^2 = \frac{a}{b^2}$, $\mu_\beta = \frac{c}{d}$ and $\sigma_\beta^2 = \frac{c}{d^2}$) for efficient posterior inference, but we use the more general notation to extend the analysis to all scale-location priors.

The priors and the number of factors K control the *sparsity* and *magnitude* of the latent representation (Cemgil, 2009), via the expected mean and variance of the rates. However, these effects are hard to separate from each other and many practitioners are unaware of the implications of their choices. As a practical example, already the common choice of independent priors for θ and β (used also in our work) encodes the prior assumption that the rows (and, equivalently, the columns) are exchangeable and hence also that the margin counts are equal in expectation. Allowing for vastly varying row (and/or column) sums would hence require switching to a more general allocation models (Yıldırım et al., 2021; Cemgil et al., 2019) or more structured priors. In practice, the quality of the prior choices often becomes apparent for the practitioner only in the context of the observed data *a posteriori*.

Compound Poisson Matrix Factorization (CPMF) (Basbug and Engelhardt, 2016; Simsekli et al., 2013) extends PMF by incorporating an additive exponential dispersion model (EDM) (Jorgensen, 1987) in the observation model, while keeping the Poisson-Gamma factorization structure:

$$\begin{aligned} &\theta_{ik} \sim F(\mu_\theta, \sigma_\theta^2), \quad \beta_{jk} \sim F(\mu_\beta, \sigma_\beta^2) \\ &Y_{ij} \sim \text{ED}(w, \kappa N_{ij}), \quad N_{ij} \sim \text{Poisson}(\sum_{k=1}^{K} \theta_{ik}\beta_{jk}), \end{aligned} \tag{2}$$

where we have $p(Y_{ij}|N_{ij}; w, \kappa) = \exp(Y_{ij}w - \kappa N_{ij}\psi(w))h(Y_{ij}, \kappa N_{ij})$, $\mathbb{E}[Y_{ij}|N_{ij}; w, \kappa] = \kappa N_{ij}\psi'(w)$ and $\mathbb{V}[Y_{ij}|N_{ij}; w, \kappa] = \kappa N_{ij}\psi''(w)$, and n_{ui} is a Poisson distributed latent count. [2] $\text{ED}(w, \kappa N_{ij})$ represents a distribution from the family of exponential dispersion models, with the natural parameter given by w and the dispersion given by κN_{ij}, and the particular distribution determined by the base log-partition function $\psi(w)$ and base-measure $h(Y_{ij}, \kappa N_{ij})$. This model family includes Normal, Poisson, Gamma, Inverse-Gamma, and many other distributions (see Table 1 in Basbug and Engelhardt, 2016).

[2] We denote $\psi'(w) = \frac{d\psi}{dw}$, $\psi''(w) = \frac{d^2\psi}{dw^2}$

The data generating distribution is influenced by both the chosen EDM distribution and the hyperparameters, now including also κ and w, and a precise a priori reasoning about their joint effect is beyond feasible even for well-versed practitioners. An intuitive view of this model is that it allows us to decouple the sparsity or dispersion from the response model (controlled by the choice of distribution to be compounded). In this sense κ would give an indication about the variability of the responses, while w would be related to the natural parameterization of the response distribution. Determining specific values for these parameters to achieve desired or expected characteristics for the data is, however, difficult.

Generic Bayesian Matrix Factorization is a generalized template model, representing the family of matrix factorization models with distributions for the priors $F(\mu_\theta, \sigma_\theta^2)$ and $F(\mu_\beta, \sigma_\beta^2)$, and observation model F_Y.

$$\begin{aligned} &\theta_{ik} \sim F(\mu_\theta, \sigma_\theta^2), \quad \beta_{jk} \sim F(\mu_\beta, \sigma_\beta^2) \\ &Y_{ij} \sim F_Y(\sum_{k=1}^{K} \theta_{ik}\beta_{jk}), \text{ with } \mathbb{E}[Y_{ij}] = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}. \end{aligned} \tag{3}$$

The choice of a specific parametric distribution for each part of the model leads specific instantiations of matrix factorization, for example if parameters and observations are normally distributed, with $F_Y = \mathcal{N}$, $F(\mu_\theta, \sigma_\theta^2) = \mathcal{N}(\mu_\theta = 0, \sigma_\theta^2)$, and $F(\mu_\beta, \sigma_\beta^2) = \mathcal{N}(\mu_\beta = 0, \sigma_\beta^2)$, we obtain the classic probabilistic matrix factorization (Salakhutdinov and Mnih, 2007); with Poisson distributed observations and Gamma distributed latent variables, $F_Y =$ Poisson, $F(\mu_\theta, \sigma_\theta^2) = \text{Gamma}(a, b)$ and $F(\mu_\beta, \sigma_\beta^2) = \text{Gamma}(c, d)$, we obtain (PMF) (Cemgil, 2009; Gopalan et al., 2014). We use this generic formulation to obtain results that are valid for any choice of parametric distributions for prior and observation distributions.

2.1 On Priors for BMF

Despite the vast literature on BMF models and their inference algorithms, many of the papers developing or applying BMF models treat the prior choice lightly and resort to fairly heuristic choices. For example, Brouwer and Lio (2017) presents a compendium of BMF models with a wide range of likelihoods and priors, but despite focusing on small data applications still set the hyperparameters by either trying values from a regular grid or setting them to fixed values based on claims of insensitivity. Similarly, Gopalan et al. (2015) solved the problem of choosing the hyperparameters for hierarchical PMF by setting all of them to 1 or 0.3 to encourage sparsity, and Basbug and Engelhardt (2016) used a combination of empirically tested and heuristically chosen hyperparameter values for CPMF. Finally, Tan and Fevotte (2013) optimized for the latent dimensionality K, but used ah-hoc values for other hyperparameters. Importantly, we stress that these examples should not be seen as weakness in these particular works, but rather as examples of a common practice motivating our research – we have also published articles where we followed the same convention. We also point out that some works do take the prior choice much more seriously. In particular, Cemgil et al. (2019) discusses the prior choices in detail in the context of a broader class of count allocation models and provides valuable information not only for PMF but also for other models.

We argue that the common practice of heuristic choices is not because BMF models are particularly insensitive to the priors, but because selecting them is not easy. In many cases the PMF model is considered as a generic model family that is not necessarily motivated by a well-understood generative description for the data, which makes specifying subjective prior knowledge difficult. For example, often the only prior information for setting the variance parameters σ^2 of Eq. (1) would be based on what kind of values have worked before when applying the model on other data sets, rather than an expert being able to somehow quantify a real subjective knowledge based on domain knowledge. Furthermore, the hyperparameters are typically not even identifiable due to the latent variables relating to data only via the product $\theta_i^T \beta_j$. In general, this is a characteristic of hierarchical, high-dimensional and complex Bayesian models, where the interplay between prior specification and the final model properties is difficult to intuit aprioristically and can only be understood in connection to the likelihood and the predictions that come from it, as is argued by Gelman et al. (2017).

For machine learning applications, various hyperparameters are today often chosen using a global optimization routine, such as Bayesian optimization (Snoek et al., 2012) or a simple grid search. Global optimization for BMF, howerer, is hard because (1) training/validation split is non-trivial for structured data, (2) posterior inference is slow for large data, and (3) the optimization surface is difficult. Figure 1 illustrates the last point for the PMF model by evaluating the predictive quality of the mean-field variational approximation as measured by PSIS-LOO (Vehtari et al., 2017) for a range of hyperparameter choices (see Section 7.2 for details). There are large regions of inappropriate choices and the sharp border between those and the feasible region makes global optimization hard. Furthermore, as illustrated on the rightmost bottom plot in Figure 1 (an example of a 1D slice plotted using three different scalings for the y-axis), the optimization surface characteristics depend on the inspection scale; the small neighborhood with optimal scores is lost at coarser scales and is difficult to find with most optimization strategies. In particular, grid-based evaluation routines are likely to miss the good hyperparameters completely.

In Section 4, we will describe an analytic solution for learning the hyperparameters for this model by matching the virtual statistics of PPD with target values. The example already illustrates how this method, which here does not require any computation besides simple equations, finds a solution surface within the feasible region, selecting also the latent dimensionality automatically. It does not give the optimal hyperparameters for this specific evaluation metric and inference algorithm, nor should it as its intended purpose. Instead, the result captures essential properties of the data with the PPD, resulting in an appropriate starting point for posterior inference.

3. Prior Predictive Matching

Having illustrated the challenges with BMF, we now proceed to provide a new method for determining hyperparameters of any Bayesian machine learning model, building on the idea of matching virtual statistics of the prior predictive distribution as explained in detail below. The principle is presented from the perspective of general Bayesian models, but in this work it is demonstrated only in the context of matrix factorization.

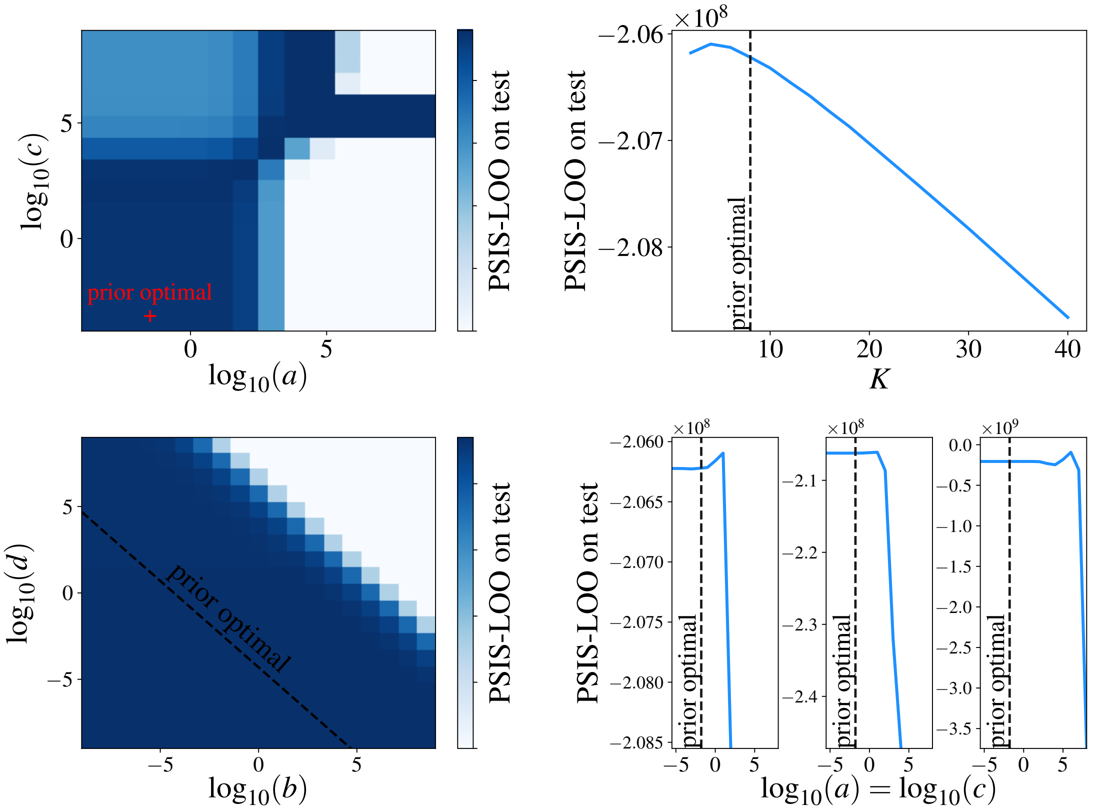

Figure 1: Illustration of difficulty of selecting good priors for Poisson matrix factorization, evaluated by predictive quality of a variational approximation on the `hetrec-lastfm` dataset. We show 2D (left) and 1D (right) slices of the loss surface in the five-dimensional hyperparameter space, with all other values fixed to prior optimal ones. The proposed prior predictive matching approach provides closed-form solution (indicated by ”prior optimal”), including the latent dimensionality K (top right), within the area of reasonable values.

3.1 General Idea

Our goal is to select good hyperparameters λ for a probabilistic model $p(Y, Z; \lambda)$, where Z denotes actual model parameters and latent variables collectively, without directly computing the posterior quality of any particular model fit. That is, we want to avoid costly and potentially difficult global optimization requiring the selection of a specific evaluation criterion for the quality of the final solution as well as training/validation split for the data. Instead, we prefer to optimize an overall match between the model and the data characteristics.

To achieve this, we consider the prior predictive distribution

$$p(Y; \lambda) = \int p(Y|Z; \lambda)p(Z; \lambda)dZ,$$

which integrates out the parameters, and we search for hyperparameters that matches the data distribution well. PPD is typically used for validating prior and modeling choices as part of the statistical modeling pipeline (Schad et al., 2019; Gelman et al., 2020), often by visual comparison of prior predictive samples and the data, e.g., so that a large deviation between the two is interpreted as an indication that the model should be modified (Gabry et al., 2019). We extend the idea to automate the prior choice, by optimizing for λ for which virtual statistics of PPD match sufficiently well with either prior knowledge of the user or empirical statistics for the available data. Our goal is specifically to find a point estimate for λ, to be used for defining the prior for subsequent posterior inference and additional processing steps. In this work we ignore any potential direct prior information on λ itself, assuming all choices are equally likely *a priori*.

Note that for some models, such as Gaussian processes with conjugate likelihoods, the PPD can be expressed analytically and the optimal solution is then obtained by directly maximizing the marginal likelihood for the observed data. This approach is, however, restricted to exponential family models with conjugate priors, or requires significant model-specific effort that does not generalize over even minor variants of the model; see Bouveyron et al. (2019) for the derivation of marginal likelihood for a specific Gaussian MF model. Our interest is in more general model classes where the PPD itself cannot be expressed in closed-form. Instead, we assume only that we can either (a) compute some low-order statistics of the PPD analytically, or (b) draw samples from the PPD. Our key contribution is providing an automatic process for selecting the hyperparameters for these broader classes of models in a computationally efficient manner. The practical method, described next, is related to the method of moments (see Section 6 for more details), but is applied here for the purpose of learning the hyperparameters as an intermediate step in the modelling workflow, rather than as an alternative for posterior inference.

3.2 Method

The gist of our proposed method is to search for λ such that the PPD $p(Y;\lambda)$ and the true data distribution $p^\dagger(Y)$ or user's prior beliefs about $p^\dagger(Y)$ (when following strictly the principles of Bayesian modeling framework) match as well as possible.[3] We quantify the match using a collection of statistics T that capture the essential properties of the data, for example in the form of central moments. The goal is to find λ such that the virtual statistics $\hat{\mathrm{T}}_\lambda$ of the PPD match some target statistics T^*. In the ideal case, we find the optimal match where $\hat{\mathrm{T}}_\lambda = \mathrm{T}^*$. We use the phrase *virtual statistic* for $\hat{\mathrm{T}}_\lambda$ to emphasize that it does not correspond to any particular observed data, but can instead be thought of as the corresponding statistic computed for a hypothetical – or virtual – data set sampled from PPD.

This general formulation depends on two elements: (1) the choice of the statistics T (and associated discrepancy measure) used for evaluating the match, and (2) the choice of the specific target statistic values T^*. Together they define the optimality. Importantly, these two objects are fundamentally linked with each other. On the one hand, a richer set of statistics T leads to the choice if hyperparameters likely to be good in the broader set

[3] We use † to denote to the true distribution for clarity, but in practice only work on empirical estimates of statistics computed from a sample Y or user-provided targets for the same statistics.

of applications, but at the same time implies the need to be more careful when providing the target values T^*, while often making computation more difficult as well. On the other hand, very simple statistics, such as only the mean of the data, are typically not sufficient for identifying a single optimal choice, but are already useful since they provide a surface of equally good choices and help ruling out nonsensical options. Finally, let us note that computational algorithms solving for λ are agnostic to how T^* were obtained, but to clarify the broad scope of the developed machinery we explain three common use-cases each with a different way of defining T^*:

1. **Principled statistician:** Following the strict Bayesian principle, the target statistics may be provided by a domain expert, in form of the expected values for the statistics. When used in this form, the proposed method essentially becomes a prior elicitation method; the expert provides subjective information on what is to be expected regarding the data, and this is used for indirectly defining the prior over the model parameters, similar to e.g. Kadane et al. (1980) and Hartmann et al. (2020). Importantly, the expert only needs to provide statistics of the data and not of the model parameters, not necessarily needing to understand the whole role of the model in detail.

2. **Held-out validation:** A somewhat more pragmatic approach is to use actual observed statistics T of a separate validation data as the target values T^*. For example, in the case of a recommender engine we might use a subset of the users and items to estimate the target statistics T^* and find λ for which the virtual statistics of the PPD best match the observed ones. After this, this data subset is discarded and the remaining data is used for posterior inference and possible further computation steps with the hyperparameters fixed to the selected ones.

3. **Automatic prior specification:** Finally, the method can also be used in a fashion where we use the observed statistics T of all available data Y as the targets T^*, loosely following the concept of empirical Bayes (Casella, 1985). While this breaks the fundamental idea of specifying the priors independent of the data, the statistics we use in practice are of low dimensionality and only characterize the data on a rough level. Consequently, we expect many practitioners to be comfortable in using the tool also in this manner. Prior predictive checks are routinely used for manually tuning the priors so that the support of the PPD roughly matches the data (Schad et al., 2019; Gabry et al., 2019; Gelman et al., 2020), and our approach can be interpreted as an automated procedure for this if using simple statistics like mean and variance. Note that this reasoning would no longer hold if using very rich statistics, in the extreme case directly using individual data entries so that $\mathrm{T}^* = Y$, but in this work we only consider problems where T consists of a few low-order moments.

Throughout this work we use moments such as mean and variance as the statistics T, since they lead to computationally efficient algorithms applicable for reasonably broad model families, but the method would work for other choices like quantiles or extreme values as well. In particular, we go through details of two practical algorithms for different scenarios, demonstrated in the context of Bayesian MF models. In Section 4, we first look at cases for which we can compute certain low-order moments of the PPD analytically and hence we can find a closed-form expression for λ corresponding to the optimal match $\hat{\mathrm{T}}_\lambda = \mathrm{T}^*$. This is ideal in terms of computation, but restricted in scope to specific models and statistics. Hence, in Section 5, we proceed to provide a general-purpose algorithm applicable to a broader family of models and statistics, formulated as explicit optimization of a discrepancy measure between the PPD and target statistics, using sampling-based estimates for the virtual statistics and stochastic gradient-descent (SGD) optimization. The method is applicable for all continuous hyperparameters and requires only ability to sample from PPD, but alternative forms of optimization could be considered to extend support also to discrete hyperparameters. We demonstrate this algorithm for hierarchical Bayesian MF models, but it could be also applied outside matrix factorizations.

4. Matching Moments for PMF and CPMF

PMF as specified in Eq. (1) allows us to obtain an analytic expression for certain moments of the PPD, to be used as virtual statistics. If we denote by Y a virtual data matrix following the PPD, we can compute the mean $\mathbb{E}[Y_{ij};\lambda]$, the variance $\mathbb{V}[Y_{ij};\lambda]$, and the correlations $\rho[Y_{ij}, Y_{tl};\lambda]$ in closed form. Here the hyperparameters are $\lambda = \{K, \mu_\theta, \sigma_\theta^2, \mu_\beta, \sigma_\beta^2\}$, and we drop the explicit depedency on λ, writing $\mathbb{E}[Y_{ij}] := \mathbb{E}[Y_{ij};\lambda]$. The detailed derivations, provided in the Appendix, build primarily on the laws of total expectation, variance and covariace for marginalizing out θ and β.

Proposition 1 *For any entry of the virtual data matrix $\mathbf{Y} = \{Y_{ij}\} \in \mathbb{R}^{N\times M}$, the mean and variance is given by:*

$$\mathbb{E}[Y_{ij}] = K\mu_\theta\mu_\beta \tag{4}$$

$$\mathbb{V}[Y_{ij}] = K[\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2] \tag{5}$$

Proposition 2 *For any pair of entries Y_{ij} and Y_{tl} of matrix $\mathbf{Y}$, their correlation is given by:*

$$\rho[Y_{ij}, Y_{tl}] = \begin{cases} 0, \textit{if } i \neq t \ \ \& \ \ j \neq l \\ 1, \textit{if } i = t \ \ \& \ \ j = l \\ \rho_1, \textit{if } i = t \ \ \& \ \ j \neq l \\ \rho_2, \textit{if } i \neq t \ \ \& \ \ j = l \end{cases} \tag{6}$$

With $\rho_1 = \frac{K(\mu_\beta\sigma_\theta)^2}{\mathbb{V}[Y_{ij}]}$ and $\rho_2 = \frac{K(\mu_\theta\sigma_\beta)^2}{\mathbb{V}[Y_{ij}]}$.

These results indicate exactly how the hyperparameters of the model directly affect the virtual statistics calculated from the PDD, here denotated by $\mathbb{E}[Y_{ij}]$, $\mathbb{V}[Y_{ij}]$ and $\rho[Y_{ij}, Y_{tl}]$. Given Propositions 1 and 2 and some target values for the moments, we can directly solve e.g. for the number of latent factors K. Denoting $\tau = 1 - (\rho_1 + \rho_2)$, we obtain:

$$K = \frac{\tau\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_1\rho_2}\left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2. \tag{7}$$

We also obtain formulas for relationships between the means and standard deviations of the priors that can be used to set e.g. the Gamma hyperparameters a, b, c and d.

$$a = \frac{\rho_2 \, \mathbb{V}[Y_{ij}]}{\tau \, \mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]} \tag{8}$$

$$c = \frac{\rho_1 \, \mathbb{V}[Y_{ij}]}{\tau \, \mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]} \tag{9}$$

$$bd = \frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]} \sqrt{\frac{ac}{\rho_1 \rho_2}}. \tag{10}$$

For the CPMF model we obtain the following result

$$K = \frac{\tau \, \mathbb{V}[Y_{ij}] - \left(\kappa \psi'(w) + \frac{\psi''(w)}{\psi'(w)}\right) \mathbb{E}[Y_{ij}]}{\rho_1 \rho_2} \left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2 \tag{11}$$

for the latent factors and the following relationships for other terms:

$$\begin{aligned}
\sigma_\theta \sigma_\beta &= \frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}] \kappa \psi'(w)} \sqrt{\rho_1 \rho_2} \\
\mathbb{E}[Y_{ij}] &= \kappa \psi'(w) K \mu_\theta \mu_\beta \\
\mathbb{V}[Y_{ij}] &= \kappa \psi''(w) K \mu_\theta \mu_\beta + [\kappa \psi'(w)]^2 K [\mu_\theta \mu_\beta + (\mu_\beta \sigma_\theta)^2 + (\mu_\theta \sigma_\beta)^2 + (\sigma_\theta \sigma_\beta)^2].
\end{aligned}$$

The derivations are provided in Appendix in Propositions 8, 9, 10 and 13.

The generic observation model $Y_{ij} \sim F_Y(\sum_{k=1}^K \theta_{ik} \beta_{jk})$, with $\mathbb{E}[Y_{ij}|\theta, \beta] = \sum_{k=1}^K \theta_{ik} \beta_{jk}$ can be analyzed using the same techniques. In this case we obtain the following equation for the dimensionality of the latent factors:

$$K = \frac{\tau \, \mathbb{V}[Y_{ij}] - \mathbb{E}[\mathbb{V}(Y_{ij}|\theta, \beta)]}{\rho_1 \rho_2} \left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2. \tag{12}$$

The term $\mathbb{E}[\mathbb{V}(Y_{ij}|\theta, \beta)]$ is model-dependent with each distinct choice of a distribution for the observation model F_Y having a different functional form for conditional variance $\mathbb{V}(Y_{ij}|\theta, \beta)$. For example, in the case of probabilistic matrix factorization model with a Gaussian observation model $Y_{ij} \sim \mathcal{N}(\sum_{k=1}^K \theta_{ik} \beta_{jk}, \sigma_Y^{-1})$, we obtain $\mathbb{E}[\mathbb{V}(Y_{ij}|\theta, \beta)] = \sigma_Y^2$; while in the case of Poisson matrix factorization with observation model $Y_{ij} \sim \text{Poisson}(\sum_{k=1}^K \theta_{ik} \beta_{jk})$, we would obtain $\mathbb{E}[\mathbb{V}(Y_{ij}|\theta, \beta)] = \mathbb{E}[Y_{ij}]$. Therefore, we obtain different analytical formulas based on Eq. 12 for each of these models. The analysis presented here can be applied to virtually any matrix factorization models, with any combination of distributions for the priors $F(\mu_\theta, \sigma_\theta^2)$ or $F(\mu_\beta, \sigma_\beta^2)$, and observation model F_Y (with finite mean and non-zero variance).

Observations: the above equations provide closed-form expressions that determine the priors in terms of the virtual statistics of PPD matching the target statistics. They can be computed instantaneously, bypassing the need for optimizing λ, and provide a computationally efficient way of automatically determining the number of factors for PMF and CPMF. The result is not necessarily optimal for any particular task, especially when the data does not follow the model well, but as illustrated experimentally in Section 7 tends to be a good choice when there is small model mismatch and for the dataset analyzed.

Algorithm 1: Empirical correlations $\widetilde{\rho_1}$ and $\widetilde{\rho_2}$ for a observed matrix $\mathbf{Y} = \{y_{ij}\} \in \mathbb{R}^{N \times M}$

```
input  : observed matrix $\mathbf{Y} = \{y_{i,j}\} \in \mathbb{R}^{N \times M}$, number of samples for the estimator
         $S$
output: $\widetilde{\rho_1}$ and $\widetilde{\rho_2}$
Initialize arrays $\mathbf{a} = \{a_{i,j}\} \in \mathbb{R}^{S \times 2}$ and $\mathbf{b} = \{b_{i,j}\} \in \mathbb{R}^{S \times 2}$;
for  $s \in \{1, \cdots, S\}$ do
    Sample $i \sim$ Unif($\{1, \cdots, N\}$);
    Sample $j_1 \sim$ Unif($\{1, \cdots, M\}$);
    Sample $j_2 \sim$ Unif($\{1, \cdots, M\} \setminus \{j_1\}$);
    $a_{s,1} \leftarrow y_{i,j_1}$;
    $a_{s,2} \leftarrow y_{i,j_2}$;
    Sample $j \sim$ Unif($\{1, \cdots, M\}$);
    Sample $i_1 \sim$ Unif($\{1, \cdots, N\}$);
    Sample $i_2 \sim$ Unif($\{1, \cdots, N\} \setminus \{i_1\}$);
    $b_{s,1} \leftarrow y_{i_1,j}$;
    $b_{s,2} \leftarrow y_{i_2,j}$;
end
Calculate and return the Pearson correlation of columns of a and b;
```

4.1 Empirical Estimates for the Moments

As explained in Section 3.2, one way of using the method is based on matching the virtual statistics with the true statistics of the observed data. For MF models, we only have a single matrix representing one (often partial) observation, and hence need to estimate the statistics by averaging over the rows and columns of one matrix instead of averaging over multiple matrices. The mean and variance can be easily estimated over the independent matrix entries, but how to compute the correlations is not immediate nor intuitive . One remark is that there are two values of correlations, one for rows and another one for columns, hence we can levarage this property of the model to make an estimator. To estimate the correlation we derived an estimator, described in Algorithm 1, that samples two elements from the same row or column and uses them to calculate the correlation of elements sharing a row or column index, independent of the specific index.

5. Gradient-based Approach

Deriving analytic expressions for even simple models and moments is tedious, error-prone, and often impossible. For users willing to give up the convenience and robustness of analytic expressions, we next provide an optimization-based alternative that does not require model-specific derivations. We will later demonstrate the effectiveness of this algorithm in the context of a hierarhical variant of Poisson MF by Gopalan et al. (2015), as a concrete example of a matrix factorization model for which analytic moments are not known.

5.1 Formulation as an Optimization Problem

Instead of directly equating the target and virtual statistics, we solve for λ that minimizes a discrepancy measuring the difference between the requested quantities T^* (either expert's prior expectation or empirical estimate) and the virtual statistics $\hat{\mathrm{T}}_\lambda$ of the PPD

$$\min_{\lambda} \mathrm{d}\left(\mathrm{T}^*, \hat{T}_\lambda\right), \tag{13}$$

where for brevity we write $\hat{\mathrm{T}}_\lambda$ instead of the complete $\hat{T}(\mathbb{E}[g(Y);\lambda])$. As before, T is a collection of statistics (e.g. central moments) defining which aspects are used for learning the hyperparameters. Intuitively, a richer set allows for a more accurate prior specification, but requires more careful choice of discrepancy as well. For example, to match an expected value E^* and variance V^* at the same time, we can use $\mathrm{d} := (\mathrm{E}^* - \mathbb{E}[Y])^2 + (\mathrm{V}^* - (\mathbb{E}[Y^2] - \mathbb{E}[Y]^2))^2$, where $g(Y) = (Y, Y^2)$ and $\hat{\mathrm{T}}(E_1, E_2) = (E_1, E_2 - E_1^2)$. The result of the optimization problem naturally depends on the choice of the discrepancy and – in the case of multiple target statistics – the weighting of the different statistics. However, when the data follows the assumed model and the target statistics are achievable, we can typically reach $\hat{T}_\lambda = T$ and the choice only influences the computational efficiency of the optimizer, not the result itself. For other cases, the user needs to select an appropriate measure and weighting that reflects their preferences.

The process builds on repeatedly drawing samples from PPD to estimate the virtual statistics $\hat{\mathrm{T}}_\lambda$, and solving for λ using some iterative algorithm. Importantly, this only requires the model code providing the samples and the target statistics T^*, and hence allows solving for the priors as part of the modeling pipeline without needing to consider any particular data.

5.2 Differentiable Moments' Estimators

We optimize Eq. (13) with stochastic gradient descent, using Monte Carlo approximation (Mohamed et al., 2020) for the prior predictive moments and automatic differentiation with reparameterization gradients (Figurnov et al., 2018) wherever available and REINFORCE (log derivative trick) (Williams, 1992) can be used elsewhere. For gradient-based optimization we require that $\mathrm{d}(\cdot)$ and $\hat{\mathrm{T}}$ are differentiable w.r.t their arguments, and that we can propagate gradient ∇_λ through $\mathbb{E}[g(Y)]$.

Even though our main interest is in BMF models, we directly derive the stochastic optimization for a somewhat more general family of hierarchical Bayesian models consisting of L layers of latent variables Z_l. The procedure is based on recursively applying the law of total expectation. The unconditional expectation of $g(Y)$ can be obtained by integrating out latent variables Z, but since an analytical form of it is not available, we proceed by performing a numerical approximation, where each of the integrals over latent variables $Z_1, \dots Z_l \dots Z_L$ is replaced by a sum over samples from respective (conditional) distributions. An estimate of the required gradient $\nabla_\lambda \mathbb{E}[g(Y)]$ is then obtained by propagating estimates of the gradients $\nabla_\lambda \mathbb{E}[g(Y)|z_l]$ and $\nabla_\lambda \log p(z_l|\dots;\lambda)$ backward through the computation graph.

The detailed description and derivation of the algorithm for estimating the gradients, which requires somewhat tedious notation and may not be of interest for readers primarily interested in BMF models, is provided in Supplement E. We then show in Supplement E.1 how both PMF and a hierarchical Poisson factorization (HPF) by Gopalan et al. (2015) can be expressed as instances of this general structure. For the HPF model we do not have analytic expressions for the moments and hence need this more general algorithm.

6. Related Work

The way we use the prior predictive distribution for hyperparameter optimization is, to the best of our knowledge, novel. The technical elements and the overall goal are, however, related to several seemingly distinct concepts, briefly outlined here.

Method of moments. Method of moments (MoM) refers to inferring model parameters by equating (in our terminology) virtual moments with sample moments, as an alternative to maximum likelihood or Bayesian inference (Casella and Berger, 2001; Pawitan, 2001; Knight, 1999). Even though our method involves similar expressions, the approaches are fundamentally different in their goals. MoM is typically used for learning a point estimate of the parameters (Z in our notation) of a probabilistic model based on observed data and the theoretical moments for the model, whereas our goal is to set values of hyperparameters λ and the formulation can be used for any statistics and targets that may not even be estimated from any data. Importantly, in our case we proceed to conduct standard Bayesian inference for Z with the chosen λ. The practical algorithms are also different. MoM is typically used either for simple models with easy analytic expression for the moments, or as an generalized method of moments relying on asymptotic normality (Casella and Berger, 2001; Pawitan, 2001). Our derivations for PMF and CPMF are contributions for the MoM literature in itself, providing equations for computing the virtual moments for a non-trivial model class, and the model-independent algorithm using stochastic gradient optimization may be useful also in other contexts.

Prior elicitation. By following the Bayesian paradigm, the choice of prior should be conducted independently from the observed data (Garthwaite et al., 2005; O'Hagan et al., 2006; Mikkola et al., 2021). When priors are difficult to specify, *prior elicitation* can be used to convert expert opinions expressed e.g. using graphical interfaces to prior distributions. This is often done by eliciting information about the parameters, but e.g. Kadane et al. (1980) and Akbarov (2009) have studied prior elicitation using prior predictive distributions. When the target statistics are provided by the user, our approach can be seen as prior elicitation following their principles but using an specific type of user-specified information. Furthermore, they could only elicit priors for over-simplistic models with Gaussian likelihoods, whereas we provide tools for more general models with a potentially large number of latent variables or hierarchical structure. More recently, Hartmann et al. (2020) followed similar lines as our paper by developing model-independent prior elicitation method based on PPD, but modelled directly the probability of the outcome data using a Dirichlet process model, to focus on accounting for the expert uncertainty in the numerical specification of the probability. The lack of sufficiently general algorithms for predictive elicitation is one of the main limitations preventing practical use of prior elicitation Mikkola et al. (2021), and our work may provide insights for improved general algorithms in that field.

Learning flexible priors. For highly flexible models, such as variational autoencoders (VAE) or Bayesian neural networks (BNN), the question of the prior choice is more poorly defined than for classic Bayesian models. Several authors have proposed directly learning a flexible prior for such cases, typically by optimizing the same objective that is used for inference. For instance, Tomczak and Welling (2018) and Klushyn et al. (2019) learnt mixture priors for VAEs by optimizing the variational objective over the prior as well, and Nalisnick and Smyth (2018) and Du et al. (2019) learnt parametric priors for BNNs to satisfy specific properties (e.g. rotational invariance or monotonicity) for the model predictions. The most closely related work in this direction is Nalisnick et al. (2021), who proposed *predictive complexity priors* to learn a prior such that the model output best matches that of a reference model. Their reference model can be interpreted as playing a role similar to our target statistic, providing an alternative way for the analyst to provide subjective information on how the model is expected to work. Their algorithm minimizing Kullback-Leibler divergence between the outputs shares some commonalities with our model-independent solution too.

Our approach has two main differences to these methods, which directly optimizing for the prior. First of all, our primary interest is in determining hyperparameters for a specific hierarhical Bayesian models with clearly defined priors, rather than learning an arbitrary prior distribution to optimize the predictive performance. The more fundamental difference, however, is that all of these approaches use the observed data itself while fitting the prior, often by explicitly maximizing the final learning objective, whereas we only consider the PPD of the model. Even if using the observed data to determine the target statistics T^*, the process for determining the priors still does not involve posterior inference, in contrast to these approaches.

Empirical Bayes. When the target statistics are computed directly from observed data, rather than provided by the user based on domain expertise, the method resembles *empirical Bayes* (EB) methods that use the data to form the prior (Casella, 1985). While this is misaligned with rigorous Bayesian principles, EB is commonly used for simplifying modeling tasks, e.g. by fixing hyperparameters to their maximum marginal likelihood values. Our approach shares the conceptual motivation and can be seen as practical extension of EB to models for which classical EB would be difficult. In context of BMFs, Wang and Stephens (2021) recently provided an EB solution for a variational approximation of the model. This approach uses flexible prior families, but is focused on Gaussian likelihoods.

Global optimization. Our approach provides an alternative to global optimization of hyper-parameters with techniques such as Bayesian optimization (Snoek et al., 2012). The core difference is that they directly optimize for some specific criterion and posterior inference algorithm, and the result will not be optimal for other choices. Our result naturally depends on the choice of the statistics (and the discrepancy measure), but is agnostic of the eventual inference algorithm and the utility of the downstream task – it can be used for any modeling task and thus better follows the Bayesian modeling paradigm. Global optimization also has a clear computational disadvantage in requiring repeated posterior analysis, although strategies for reducing the cost by using subsets of data have been developed (Klein et al., 2017a). For models with an analytic solution our approach is immediate, but also the model-independent algorithm in our experiments has been faster than any method relying on posterior inference.

Finally, we would like to emphasize that our method takes advantage of the fact that generative models define a sampling distribution, and hence is restricted to such models. That is, we can only carry out hyperparameter optimization for generative probabilistic models, in contrast to Bayesian optimization that can be applied for determining hyperparameters of arbitrary machine learning models. Relying on this additional property is also why we are able to outperform the more generic method.

Likelihood-free inference. Our approach also relates to likelihood-free inference (or Approximate Bayesian Computation) for posterior inference in the case of models for which the likelihood cannot be evaluated in closed form but can be sampled from (Marin et al., 2012; Lintusaari et al., 2017). The prior predictive distribution has these properties, which means our method can be interpreted as likelihood-free inference for the hyperparameters and we can borrow suggestions for discrepancy measures from the literature in that field (e.g. Beaumont 2010; Lintusaari et al. 2017). A core difference, however, is that we do not seek for a distribution over the hyperparameters but instead prefer a point estimate; the result will typically be used as input for subsequent modeling stages and we do not expect the user to be willing to specify further hierarchical priors for these parameters. This allows us to use direct gradient-based optimization, in constrast to model-based approximations (Gutmann and Corander, 2016) or inefficient MCMC samplers (Marjoram et al., 2003) used for likelihood-free inference.

Spectral methods. For various latent variable models, such as topic models, spectral methods find globally optimal parameters by matching low-order moments of data while integrating out the latent variables (Anandkumar et al., 2012). We also use low-order moments (or other statistics) as inputs and search for point estimates – for hyperparameters while marginalizing out the parameters – but otherwise the techniques are very different. Spectral methods require as inputs accurate estimates for very fine-grained moments (e.g. for topic models the joint probability of all possible triples of words, which already for a vocabulary of 10,000 words corresponds to 10^{12} values), whereas we use a small number of global moments – just four in case of PMF – that can be provided as crude estimates.

7. Experiments

Here we present two main experimental results demonstrating the potential of our method, followed by a set of technical examinations illustrating its properties and behavior. in Section 7.1 we first verify our approach by showing that for PMF the analytic solution can retrieve the true generating prior for artificial data. Then, in Section 7.2 we compare hyperparameters selected by prior predictive matching against the ones found by Bayesian optimization applied directly for a posterior quality measure. Finally, Section 7.3 presents several technical illustrations, including an analysis of sensitivity of the approach to model mismatch or various kinds and a convergence study for the gradient-based algorithm. Throughout the experiments we use capital letters to refer to different hyperparameter settings, defined in Tables 1, 2, 3 and 4 in Appendix. The code used for all experiments is available in a public repository[4].

[4] `https://github.com/zehsilva/prior-predictive-specification`

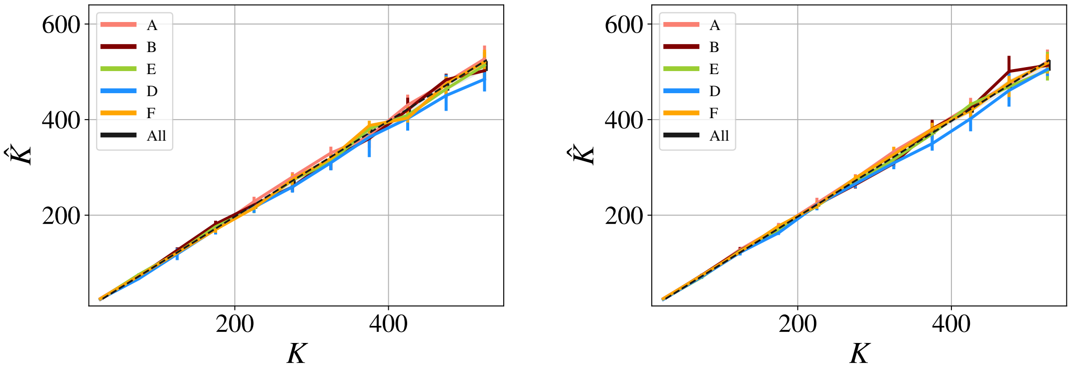

Figure 2: Prior predictive matching provides estimates $\hat{K}$ for each true latent factor dimensionality K and prior configurations (colored lines), as analytic expression of empirical moments for both Poisson MF (left) and Compound Poisson MF (right). The lines indicate the median over 30 replications and the shaded are corresponds to 95% confidence intervals.

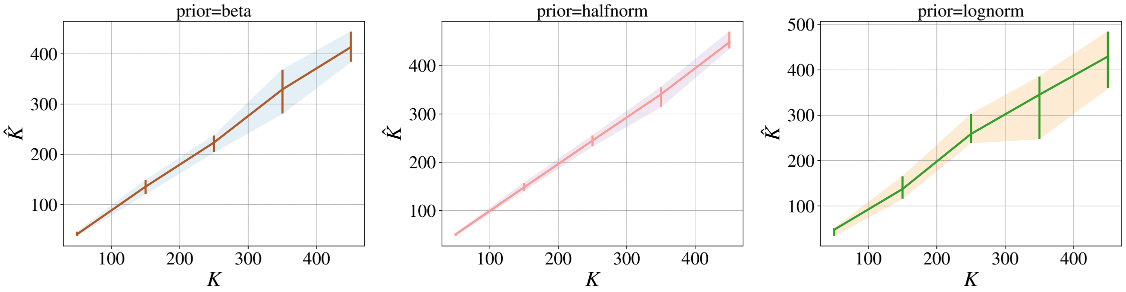

Figure 3: Comparison of the latent factor dimensionality estimator $\hat{K}$ for different choices of prior distribution for PMF: Beta (left), Half-Normal (center) and Log-Normal (right). The y-axis represent the estimated latent factor dimensionality $\hat{K}$ with 95% confidence intervals around the median, and the x-asis represent the true dimensionality K.

7.1 Analytic Solution for PMF and CPMF

7.1.1 Number of factors for count data

For PMF and CPMF, Eq. (7) and Eq. (11) provide analytic expressions for hyperparameters given the target moments. We demonstrate them in an empirical Bayes scenario, where estimates of the observed data are used as targets, to show that the results are robust to estimation errors. We sample a data matrix (of size $10^3 \times 10^3$) from the model for 30 scenarios where the true hyperparameters (denoted by λ^*) are set at different values. We repeat this for a range of values for the true K, and for each data compute the empirical

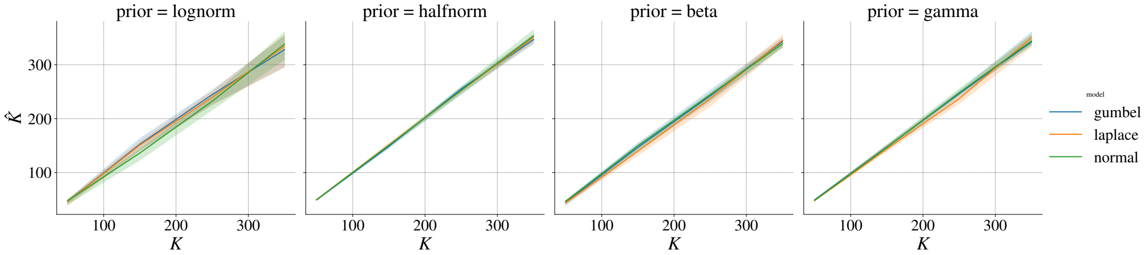

Figure 4: Comparison of the latent factor dimensionality estimator $\hat{K}$ for generic Bayesian matrix factorization models, validated using different choices of observation models (Gumbel, Laplace and Normal) and different choices of prior distribution: Log-Normal, Half-Normal, Beta and Gamma. The y-axis represent the estimated latent factor dimensionality $\hat{K}$ with 95% confidence intervals around the median, and the x-asis represent the true dimensionality K.

estimates required for estimating the number of factors using Eq. (7) and Eq. (11). Figure 2 shows the estimates accurately match the ground truth when the data follows the model, for both PMF and CPMF with observation model $Y_{ij} \sim \sum_{i=1}^{N_{ij}} \mathcal{N}(1,1)$. For both cases we used gamma distributions as the priors.

The formulation of the model in Eq. (1) allows for generic prior distributions with hyperparameters mean μ and variance σ^2, and the resultings equations are valid for multiple choices of prior distribution consistent with the model constraints (for example of non-negativity of the rate of the Poisson). We repeated the above analysis using different prior distributions for PMF: Beta, Half-Normal and Log-Normal. For each prior distribution we fixed a set of valid hyperparameters (the values used for each the configuration of prior distribution are in Table 3). Figure 3 shows the results for these choices of priors, verifying that the result holds irrespective of the prior.

7.1.2 Different observation models

To demonstrate the effectiveness of the method on other forms of observed data, we next evaluate the analytic solutions of Eq. (12) for the latent dimensionality of the generic MF model of Eq. (3). The empirical validation is performed using Gumbel, Laplace and Normal distributions for the observation model, with Log-Normal, Half-Normal, Beta and Gamma priors for each of observation distributions. The location parameters of the observation distributions were set such that the conditional mean is linear with respect to the latent factors $\mathbb{E}[Y_{ij}|\theta,\beta] = \eta_{ij} = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}$, while the expected conditional variance is a hyperparameter of the model that is related to the scale parameters of the distribution (details in Appendix A). For example, for the Gumbel distribution with location and scale parameters μ_{ij} and σ, we have $Y_{ij} \sim \text{Gumbel}(\mu_{ij},\sigma)$, and given that $E[Y_{ij}|\theta,\beta] = \eta_{ij} = \mu_{ij} + \gamma\sigma$, the location parameter is set to $\mu_{ij} = \eta_{ij} - \gamma\sigma$, and the resulting conditional expected variance is $\mathbb{E}[\mathbb{V}(Y_{ij}|\theta,\beta)] = \frac{\pi^2}{6}\sigma^2$. The constant $\gamma \approx 0.577$ used in the parameterization of the Gumbel distributions is the Euler–Mascheroni constant.

For each combination of observation and prior distributions, we vary the hyperparameters (details in Appendix A and Table 3) and the true dimensionality of the latent vectors $K \in \{50, 150, 250, 350\}$, with 20 runs for each unique configuration. We sample a data matrix (of size $10^3 \times 10^3$) for the each configuration of the experiment, calculating empirical estimates for the mean, variance and correlations, and using those estimates to estimate the latent dimensionality $\hat{K}$. Figure 4 shows the estimated dimensionality $\hat{K}$ (in the y-axis) for each true latent dimensionality K (aggregated over multiple runs and hyperparameters configurations for each prior and observation distribution). The conclusion is that the method retrieves the true latent dimensionality well irrespective of the likelihood and prior choices. Figure 5 shows the relative error $\frac{\hat{K}-K}{k}$ of the latent dimensionality estimator, which allows for a more detailed analysis of the effect of the prior and observation model choice. For some combinations the variance of the estimator is higher than for others, even though all have similar mean performance. In particular, the Log-Normal prior displays higher variability in general, while the variability of the relative error for the Beta prior decreases with the true latent dimensionality K.

7.2 Posterior Quality

The main use for the approach is as part of a modeling process where we eventually carry out posterior inference using the selected hyperparameters. The quality of the solution can hence only be evaluated by inspecting how the final model performs. We do this by fitting a PMF model to the `user-artists` data of the `hetrec-lastfm` dataset (Cantador et al., 2011)[5], using an efficient implementation of coordinate ascent variational inference[6] fitted to randomly selected 90% subset of the data. We evaluate the quality of the model using the PSIS-LOO criterion (Vehtari et al., 2017). PSIS-LOO is here considered as an example of a typical task-agnostic metric a practitioner would be likely to use, but the experiment is not sensitive to the specific choice.

Figure 1 already illustrated the quality surface via explicit enumeration of the hyperparameter choices in a regular grid, shown as slices of the five-dimensional surface where all remaining parameters were fixed to the optimal ones provided by our method. Some prior choices have better PSIS-LOO scores – for example, slightly smaller K would be better – but the crucial observation is that prior predictive matching provides a sufficiently good solution in an instant. It is also important to understand that these results depend on the specific inference algorithm and evaluation metric used, and the optimal solution would change – possibly by a lot – if the variational approximation was replaced, e.g., with MCMC and PSIS-LOO by a another metric. Hence, exactly matching whatever choice happens to be optimal here would not even be correct.

Figure 6 compares the solution with the alternative of running a global optimization algorithm to figure out the hyperparameters, for the same model, data, inference algorithm and evaluation metric as above. We optimize for λ using the robust Bayesian optimization (RoBO) tool (Klein et al., 2017b) (with default options for parameters specified below), and compare results of multiple runs with the one provided by our approach in an instant.

[5] `http://files.grouplens.org/datasets/hetrec2011`

[6] We extended `http://github.com/dawenl/stochastic_PMF` to support sparse data.

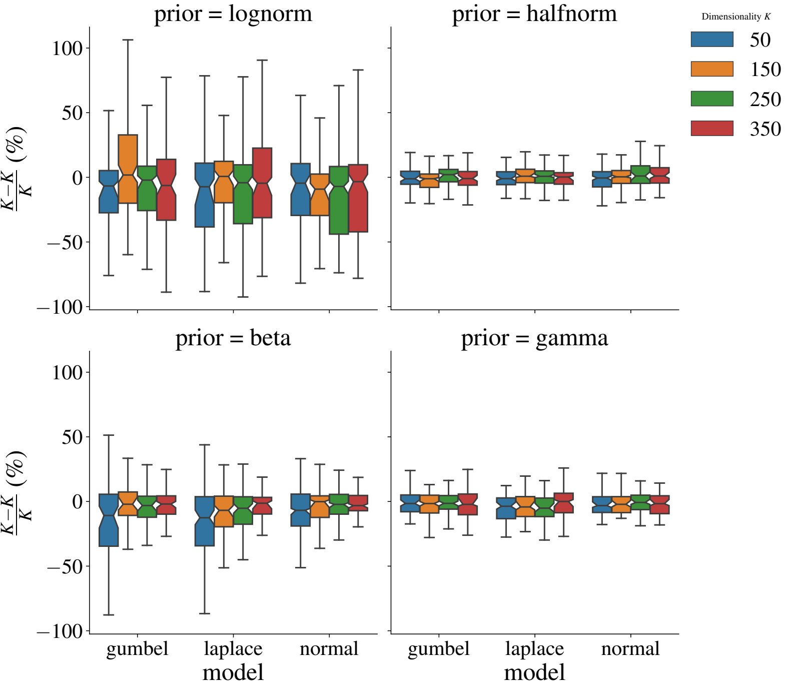

Figure 5: Boxplots of the relative error $\frac{\hat{K}-K}{K}$ analyzing the latent factor dimensionality estimator $\hat{K}$ for generic Bayesian matrix factorization models, validated using different choices of observation models (Gumbel, Laplace and Normal) and different choices of prior distribution: Log-Normal, Half-Normal, Beta and Gamma. The y-axis represent relative error, displayed in a collection of boxplots showing the median and quartiles, with color code for different true latent dimensionality $K \in \{50, 150, 250, 350\}$, observation model in the x-axis and organized in rows and colums for different priors (Log-Normal in top-left, Half-Normal in top-right, Beta in bottom-left and Gamma and bottom-right).

BO requires specifying a bounding box of feasible values, which is not trivial for PMF and has notable impact on the performance. For the fairly carefully selected search space of $\mu_\theta, \sigma_\beta, \sigma_\theta, \sigma_\beta \in [10^{-4}, 10^2]$ (optimized on logarithmic scale) and $K \in \{1..100\}$, BO is typically able to improve on the result of prior predictive matching. However, this takes on average roughly an hour even for this simplified scenario where efficient variational approximation can be fit to the fairly small data (on average) in a minute, and again we remind that BO is directly optimizing for this particular measure for the chosen inference algorithm. For more complex models taking e.g. hours for posterior inference, BO would be unfeasible. Even in the case of switching from analytic solutions to the model-independent algorithm for prior predictive matching, experiments in Section 7.3.2 indicate that our approach is still likely to be more efficient; for PMF the algorithm converged in seconds, which can be directly compared with the times reported here for BO. Finally, we remark that the prior optimal choice always outperforms the initial estimate of BO and BO typically requires tens

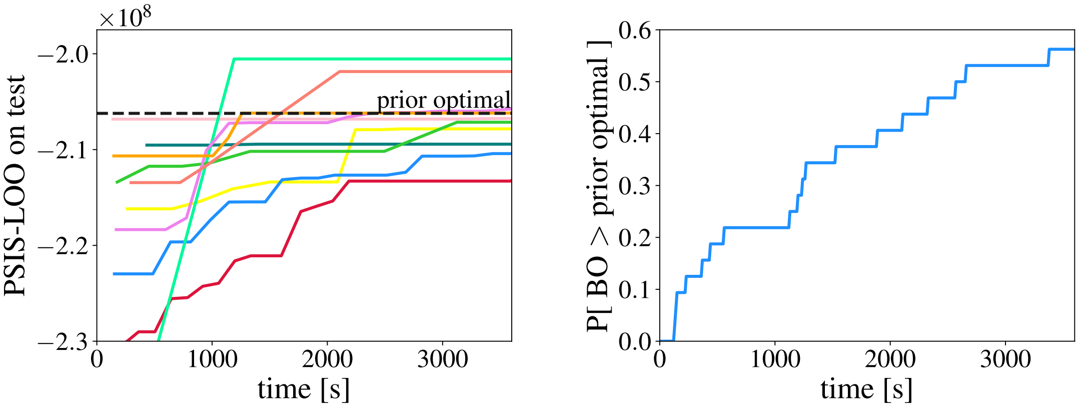

Figure 6: Comparison between Bayesian optimization runs (BO; colored lines) and the proposed method (dashed line) for optimization of five-dimensional hyperparameter for Poisson MF. BO eventually provides better PSIS-LOO (by directly optimizing for it), but even after an hour almost half of the runs are worse than our analytic, immediate, solution (right).

of iterations to reach its quality, which suggests that it is also likely to outperform various naive heuristics and default values for the hyperparameters.

7.3 Technical Validations

After demonstrating that our approach can be successfully applied for selecting hyperparameters and priors in practical scenarios, we proceed to technical experiments examining in detail the sensitivity of the approach to violation of the model assumptions (Section 7.3.1) and testing properties of the gradient-based algorithm (Sections 7.3.2). Additional technical experiments on estimators and parameterization of the prior are provided in Appendices F and G.

7.3.1 Sensitivity to Model Mismatch

In most applications, the data does not follow any model in the assumed model family. Since we compute the virtual statistics conditional on the model, it is unclear how well the approach works when the model mismatch is severe. We conducted three experiments to evaluate this, by controlling the amount of mismatch on artificial data while assuming the PMF model and using the analytic expression Eq. (7) for setting the number of factors K.

We first consider two forms of model mismatch that relate to the distribution of the individual entries. For both experiments we generated data with $K \in \{25, 50, 75, 100, 125, 150\}$ for two hyperameter configurations and generated 20 realizations for each configuration, and we summarize the results using relative error $\frac{\hat{K}-K}{K}$ of the estimates $\hat{K}$ for different true K.

Sparsity A typical mismatch with count-matrix models relates to sparsity; the data has more zeroes than expected under the model. To investigate the effect of this, we indepen-

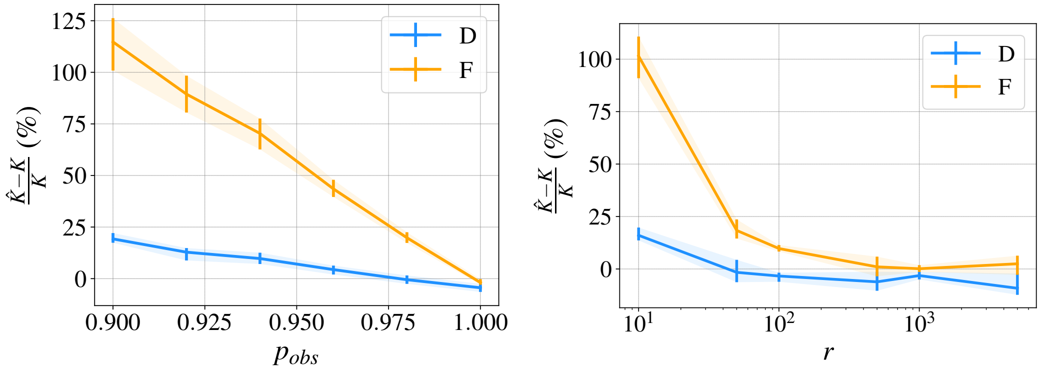

Figure 7: Sensitivity to model mismatch on zero-inflated (left) and overdispersed (right) data. For both cases increasing model mismatch (smaller p_{obs} or r) increases the error monotonically, implying the approach is robust for a small model mismatch but may give misleading results if the assumed model family fits the data very poorly. The y-axis represent the relative error $\frac{\hat{K}-K}{K}$ with 95% confidence intervals around the median, and D and F refer to two different true hyperparameter configurations.

dently sample for each entry Y_{ij} (sampled from PMF) a Bernoulli variable $X_{ij} \sim \text{Ber}(p_{obs})$ controlling whether the entry is observed, so that our final observation is given by $\tilde{Y}_{ij} = X_{ij} \times Y_{ij}$. Decreasing p_{obs} increases model mismatch, and Figure 7 (left) shows this also increases relative error in the estimate $\hat{K}$, but the decline is graceful. For some configurations (blue) the relative error stays below 25% even after dropping 10% of observations. Note that for this kind of model mismatch the retrieved hyperparameter is consistently larger than the true one; additional components are required to explain the increased variance caused by the excess zeroes.

Overdispersion The second experiment considers another prototypical model mismatch, a scenario where a too simple model class is used. We use PMF as the model, but generate the data so that the Poisson likelihood is replaced with negative binomial (NB) with varying rate of overdispersion that PMF cannot account for. More specifically, we sample data from NB distribution so that the conditional mean is given by the MF, but the variance is controlled with overdispersion parameter r, using the parameterization $\text{NB}(r, p)$ where r is the number of failures until the experiment is stopped and $1 - p$ is probability of failure. Using the notation of Eq. 1, our data is hence generated by $\eta_{ij} = \left(\sum_{k=1}^{K} \theta_{ik}\beta_{jk}\right)$ and $Y_{ij} \sim \text{NB}(r, \frac{\eta_{ij}}{\eta_{ij}+r})$. This implies that $\mathbb{E}[Y_{ij}|\eta_{ij}; r] = \eta_{ij}$ and $\mathbb{V}[Y_{ij}|\eta_{ij}; r] = \eta_{ij} + \frac{\eta_{ij}^2}{r}$, so the smaller the r, the more overdispersed the distribution is. We vary $r \in \{10, 50, 100, 500, 1000, 5000\}$, and Figure 7 (right) shows that again the procedure is relatively robust for the mismatch.

Both experiments indicate that the analytic expression for determining K is robust only for a small model mismatch, and that the relative error grows drastically as a function of the mismatch. Here both examples correspond to overdispersed data compared to the assumed model – which is often the case – which results in overestimating the number of factors.

In general, it is difficult to anticipate how specific kinds of model mismatches influence the results, but as a general rule we suggest interpreting odd hyperparameters as signs of potential mismatch worth investigating in more detail. For both of these examples the problem could be solved by switching to an appropriate model, explicitly modeling missing data in the first case and using negative binomial as likelihood in the latter. For both cases we would also need to switch to formulas, adapting the general result of Eq. 12.

Inconsistent margins As mentioned in Section 2, placing independent priors on the factors induces equal margin sums in expectation. More specifically, the expected row-sum is $\mathbb{E}[\sum_i Y_{ij}] = NK\mu_\beta\mu_\theta$ and the expected column-sum is $\mathbb{E}[\sum_j Y_{ij}] = MK\mu_\beta\mu_\theta$. Often the margins of real data sets are, however, quite non-uniform. We already showed in Section 7.2 that the overall configuration of the hyperparameters can be good even in the recommender engine setups that typically have very non-uniform margins, but it is worth studying the sensitivity to this specific form of model mismatch also in scenarios that can be analysed in more detail.

We consider a case where each row is consistently element-wise multiplied with a positive vector $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_M]$, with $\gamma_j \geq 1$, and again focus on the estimator for K. In this case, given $\boldsymbol{\eta} = \{\eta_{ij} = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}\}$, the conditional mean and variance is $\mathbb{E}[Y_{ij}|\theta\beta] = \mathbb{V}[Y_{ij}|\theta\beta] = \gamma_j\eta_{ij}$, and the marginal mean and variance $\mathbb{E}[Y_{ij}] = e_j = K\gamma_j\mu_\theta\mu_\beta$ and $\mathbb{V}[Y_{ij}] = v_j = e_j + \gamma_j^2\,\mathbb{V}[\eta_{ij}]$ with $\mathbb{V}[\eta_{ij}] = K((\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2)$. We observe that each column j have an expected sum controlled by γ_j, which also influences the degree of overdispersion, and the parameter γ_j can be chosen to enforce a desired value the expected column-sum. Furthermore, we can calculate the correlations between Y_{ij} and Y_{tl}, resulting in the following expressions for the non-trivial cases: for the fixed row $i = t$ and different columns $\rho(Y_{ij}, Y_{tl}) = \rho_{jl}^{(1)} = \frac{K\gamma_j\gamma_l}{\sqrt{v_j v_l}}(\mu_\beta\sigma_\theta)^2$, and for fixed columns $j = l$ and different rows $\rho(Y_{ij}, Y_{tl}) = \rho_j^{(2)} = \frac{K\gamma_j^2}{v_j}(\mu_\theta\sigma_\beta)^2$. In summary, the mismatch here corresponds to both larger expectation and variance, as well as a distinct correlation structure and column-wise variation which is not present in the PMF.

We perform an experimental validation by setting the scaling matrix $\boldsymbol{\gamma} = [1, \ldots, \gamma_{max}] \in \mathbb{R}_+^M$ and varying $\gamma_{max} \in \{1, \ldots, 8\}$. We use $K = 50$, executing 5 runs of the experiment for each prior (Beta, Gamma, Half-Normal and Log-Normal) with different configurations of their respective hyperparameters (detailed in appendix Table 3), while using the standard equations for PMF to estimate the necessary quantities. Figure 8 shows that the equation for K consistently underestimates the true value, with worse results as γ_{max} increases. This is expected, since we added multiple sources of deviation from the original model assumptions, resulting in equations for the moments that are varying with each column index for mean and variance, and each pair of columns for the correlations.

It is worth noting that this particular form of mismatch could be corrected for by deriving a new estimator for K, using the values for $\mathbb{E}[Y_{ij}]$, $\mathbb{V}[Y_{ij}]$ and $\rho(Y_{ij}, Y_{tl})$ described above, and new empirical estimates of those quantities. Alternatively, it is possible to recover the original terms of the equations derived for PMF by rescaling the observations with the formulas $\hat{Y}_{ij}^{(1)} = \frac{Y_{ij}}{\gamma_j}$ and $\hat{Y}_{ij}^{(2)} = \frac{Y_{ij}}{\gamma_j^2}$, and calculated adjusted empirical estimates for the mean $\mathbb{E}[\hat{Y}_{ij}^{(1)}]$, variance $\mathbb{V}[\hat{Y}_{ij}^{(1)}] - \mathbb{E}[\hat{Y}_{ij}^{(2)}] + \mathbb{E}[\hat{Y}_{ij}^{(1)}]$ and covariance $\mathrm{Cov}(Y_{ij}^{(1)}, Y_{tl}^{(1)})$. In

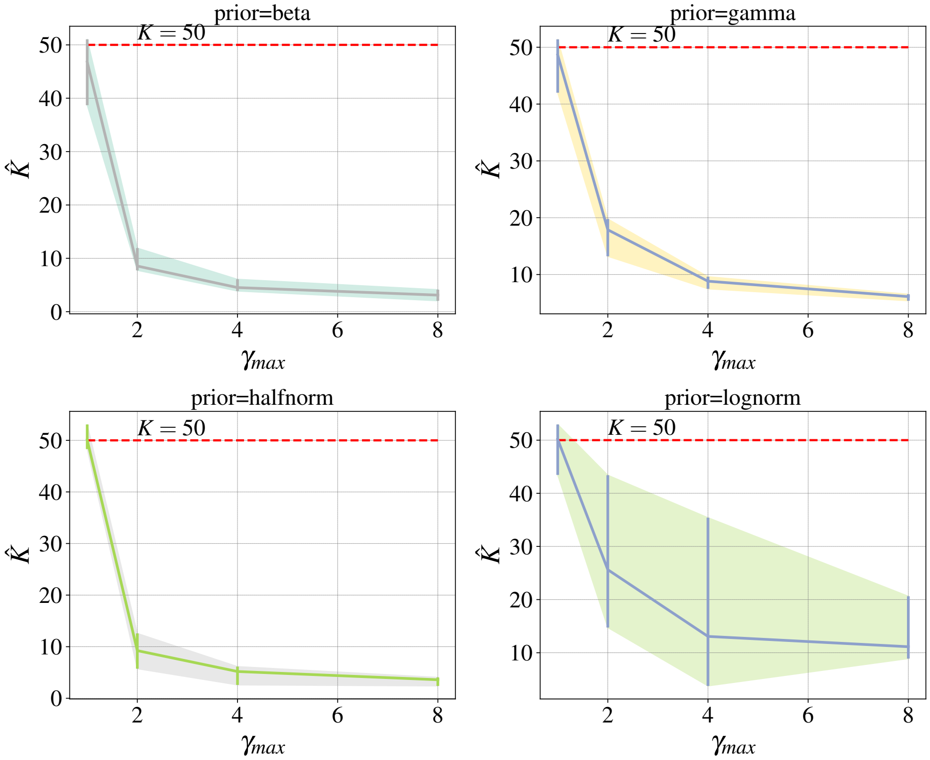

Figure 8: Sensitivity of the estimated latent dimensionality $\hat{K}$ when each row of the latent rate matrix $\boldsymbol{\eta} = \{\eta_{ij} = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}\} \in \mathbb{R}_{+}^{N\times M}$ is consistently multiplied element-wise by $[1, \ldots, \gamma_{max}]$. This procedure induces a resulting count-matrix with distinct column-sum counts, where higher column-index display higher values. Each graph was generated using a fixed true dimensionality $K = 50$, and different prior distributions: Beta (top left), Gamma (top right), Half-Normal (bottom left) and Log-Normal (bottom right). Results represent median over 5 runs for each prior configuration and its 95% confidence interval.

the Appendix H these estimates are discussed in more detail, and we show an example of adjusting for model mismatch in this case of column-wise multiplicative factors.

7.3.2 Stochastic Algorithm Convergence

The stochastic algorithm of Section 5 can be used for richer model families, but as an iterative algorithm, it requires more careful analysis. We study its properties via convergence speed and sensitivity to initial conditions for two models in the prior elicitation scenario (target values provided by the user), and illustrate an important property regarding scenarios where the target values cannot be satisfied.

Before demonstrating the method in practice, we note that the algorithm itself relies on Monte Carlo estimates for the virtual statistics. The statistics are not necessarily accurate. Hence, we empirically evaluate the bias and variance of estimators in Appendix G,

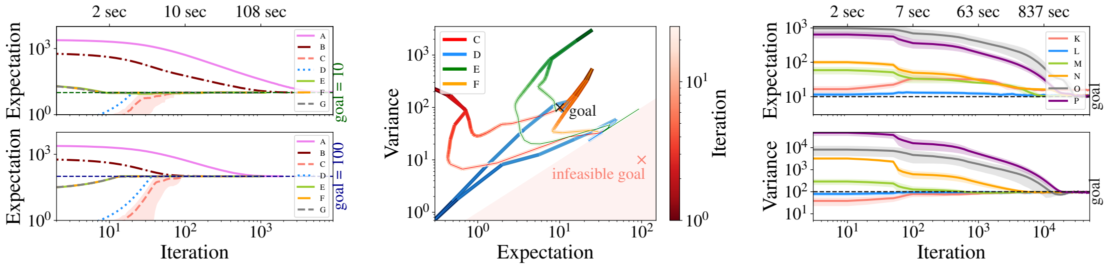

Figure 9: Convergence of the model independent algorithm for PMF (left and middle) and HPF (right) starting from different initializations.

showing that for estimating the mean the bias is typically negligible but that for the variance estimator the bias can be large for small values of $\mathbb{V}[Y]$. Furthermore, convergence of gradient-based optimization algorithms depends on how the optimization space is structured. Therefore, we compare two alternative parametrizations for PMF in Appendix F, showing that good model parametrization (e.g. in terms of means and variances) may simplify the solved optimization problem.

We first apply the algorithm for PMF, to demonstrate we can replicate the behavior of the analytic solution (but using fixed $K = 25$, since the algorithm currently does not support discrete hyperparameters). Figure 9 (left) shows convergence plots for two target values for the mean $\hat{\mathbb{E}}[Y]$ and seven initial values (Table 1 in the Appendix), demonstrating that we find the optimal solution typically in seconds and even for intentionally bad initializations (scenario "A" starts with orders of magnitude too large mean) in minutes. The optimization was carried out in location-scale parameterization, using the Adam optimizer with a learning rate of 0.1 and estimating $\hat{E}[Y]$ with $S_y = 3$ samples of the observed variable for each of $S_1 = 100$ samples of the latent variables.

Figure 9 (right) shows similar convergence curves for hierarchical PMF (Gopalan et al., 2015), optimizing for both mean and variance targets provided by the user with $\mathrm{d} = (\hat{\mathbb{E}}[Y] - 10)^2 + (\hat{\mathbb{V}}[Y] - 100)^2$, illustrating that the algorithm works for more complex MF models and discrepancies as well. Again, we find the solution irrespective of the initial value (Table 2 in the Appendix), but convergence is naturally slower due to higher dimensionality (2 more hyperparameters) and the hierarchical structure. The convergence of the optimizer is highly insensitive to the relative weighting of the two statistics, resulting only in minor speed differences (not shown here).

Finally, Figure 9 (middle) demonstrates the behavior of the algorithm for two PMF cases optimized for mean and variance, one with feasible and one with infeasible target moments. Besides users specifying contradicting goals in the prior elicitation case, the latter case can also happen in the empirical Bayes scenario if the data does not exactly match the model. For PMF we must have $\mathbb{V}[Y_{ij}] \geq \mathbb{E}[Y_{ij}]$ and hence here the target $(\mathrm{E}^*, \mathrm{V}^*) = (100, 10)$ cannot be reached. The algorithm correctly handles this, by converging to the border of the feasible region and returning a non-zero discrepancy, providing valuable feedback for the user on possible model mismatch or poorly specified targets. The weighting of the two statistics determines what kind of compromise the algorithm makes and allows freedom for the user to indicate e.g. reliability of their given targets.

8. Discussion and Conclusion

The hyperparameters of Bayesian machine learning models are often overlooked and chosen based on vague intuitions instead of more rigorous practices, e.g., gamma priors with small mean and large variance are used to induce sparsity. This is understandable as the strict Bayesian paradigm requires specifying the prior independently of the data (Garthwaite et al., 2005; O'Hagan et al., 2006), whereas standard optimization procedures require fitting the model on some data to evaluate the quality. Even if accepting that priors are to be adapted based on available data (which is admissible for latent-variable models whose priors do not really encode subjective prior knowledge), optimizing for the hyperparameters is challenging since the posterior inference is slow and the optimization surface can be difficult.

We provided a framework for selecting hyperparameters for Bayesian models that: (a) better fits within the Bayesian paradigm by only relying on simple data summaries or by entirely ignoring the data and only using expected moments provided by a user; (b) is efficient in contrast to the traditional alternatives by avoiding posterior inference when specifying priors. The framework builds on the prior predictive distribution, previously used for prior predictive elicitation (Kadane et al., 1980) and model checking (Schad et al., 2019; Gelman et al., 2020), converted here into a tool for ML practitioners working with large models. From another perspectice, the approach generalizes methods of moments and maximum marginal likelihood.

We demonstrated the effectiveness of the approach for Bayesian matrix factorization models, for which the choice of priors and hyperparameters is difficult. For Poisson MF (Cemgil, 2009) and Compound PMF (Basbug and Engelhardt, 2016) we derived analytic expressions that provide immediate solutions (i.e., hyperparameters optimal in terms of PPD matching the desired statistics) and, in particular, we now have an analytic expression for determining the number of latent factors. Finally, we presented a model-independent optimization algorithm that was here tested on a hierarchical MF model (Gopalan et al., 2015), but can be applied for a more general class of models. In this work, we only demonstrated the effectiveness of the approach in the context of matrix factorization to keep the scope of the paper more limited, but the algorithm applies to any probabilistic model as long as we can sample from the model in a differentiable manner. Empirical evaluation of the approach in context of other model families remains an interesting future direction.

The overall methodology is generic, supporting in principle an arbitrary choice of the statistics and discrepancy measures and not taking any stance on how the values for the target statistics are determined. Such a flexibility means the method requires some care when used. We believe that as long as the set of statistics are limited to a few low-order moments it is safe to estimate the target values directly from data, but cannot say in detail what would happen if using considerably higher order moments or other more detailed statistics. Our practical recommendation is to start with the obvious statistics of mean and variance, and then add additional ones if this is considered relevant for the problem or if the solution is still too undetermined. Besides correlation, which is considered in this work, we see potential in using kurtosis, skewness, extreme values and selected quantiles of the data distribution, already used in prior elicitation protocols.

Perhaps the most important aspect requiring expertise from the user concerns model mis-spefication. We empirically studied three different forms of mis-specification to highlight varying degrees of robustness. For instance, for PMF the approach is robust for minor over-dispersion and zero-inflation, but suggests a too low number of factors for cases where the margin counts do not follow the prior assumption of uniformity. However, it can still work well in practice even in such cases, as demonstrated by good the performance in posterior quality comparisons on real recommender engine data with clearly non-uniform margins. In general, the results for models that fit the data very poorly are not valid. Our best recommendation here is for practitioners to use this results as a starting point for the prior specification in the modeling workflow, but not to necessarily directly rely on the solution. For example, if our method suggests using only a few factors for a recommender engine task commonly modeled with dozens of factors, our recommendation is to carefully consider whether the specific probabilistic model matches the particular data at hand or whether, e.g., changing the likelihood or model structure would be in order. Alternatively, the solution can be used as an initial point for Bayesian optimization.

Our results demonstrate the practical feasibility of the idea, but further improvements would help make it a routine part of the Bayesian ML development pipeline. Promising directions for improving the model-independent algorithm include: (1) more careful analysis of the moment and discrepancy measure choices, possibly building on the existing work done for approximate Bayesian computation, (2) automatic choice of good parameterization for the hyperparameter space, possibly along the lines of Gorinova et al. (2020), to support generic probabilistic programs, (3) better empirical estimators for the gradients (Mohamed et al., 2020) to reduce optimization noise, (4) support for discrete hyperparameters, and (5) better means for identifying failure modes without relying on external validation or theoretical analysis.

Acknowledgments

The work was supported by Academy of Finland, under grant 1313125, as well as the Finnish Center for Artificial Intelligence (FCAI), a Flagship of the Academy of Finland. We also thank the Finnish Grid and Cloud Infrastructure (urn:nbn:fi:research-infras-2016072533) for providing computational resources. We also thank Norwegian Open AI Lab - NTNU for the support with computational resources and funding. We thank Helge Langseth, Heri Ramampiaro (NTNU/Trondheim) and António Góis (Mila/Montreal) for the useful discussions and critical review of the publication. We thank the anonymous reviewers for critical feedback and discussion. Finally, we thank Alina Karakanta (Leiden University) for the helpful feedback and review of the manuscript.

Appendix A. Setup of the experiments

Tables 1 and 2 contain the configurations used in experiments with respectively PMF and HPF. The legends in Figures 7, 11 and 12 refer to these letters.

Table 1: Considered sets of initial PMF hyperparameters.

	a	b	c	d	μ_θ	σ_θ	μ_β	σ_β	$\mathbb{E}[Y]$	$\mathbb{V}[Y]$
A	10	1	10	1	10.0	3.16	10.0	3.16	2500.00	55000.00
B	10	2	10	2	5.0	1.58	5.0	1.58	625.00	3906.25
C	0.001	0.01	0.01	0.1	0.1	3.16	0.1	1.0	0.25	253.00
D	0.1	1	0.1	1	0.1	0.32	0.1	0.32	0.25	0.55
E	0.1	0.1	0.1	0.1	1.0	3.16	1.0	3.16	25.00	3025.00
F	1	1	0.1	0.1	1.0	1.0	1.0	3.16	25.00	550.00
G	1000	1000	1000	1000	1.0	0.03	1.0	0.03	25.00	25.05

Table 2: Considered sets of initial HPF hyperparameters.

	a	a'	b'	c	c'	d'	$\mathbb{E}[Y]$	$\mathbb{V}[Y]$
K	1.0	100.0	10.0	1.0	100.0	10.0	0.26	0.26
L	0.1	100.0	1.0	1.0	100.0	1.0	2.55	8.25
M	50.0	5000.0	10.0	1.0	5000.0	1.0	125.05	781.42
N	1.0	100.0	1.0	10.0	10.0	1.0	280.57	15309.26
O	450.0	4500.0	100.0	10.0	400.0	1.0	1128.10	9833.55
P	50.0	50.0	1.0	1.0	50.0	1.0	1301.71	146074.01

Table 3 shows the configurations of the priors used in the experiments with multiple priors. For each prior distribution of the parameters, θ and β, hyperparam_1 and hyperparam_2 represent the location and scale, according to the standard parameterization used in the implementation of the `numpy`[7] library. Table 4 shows what is the location and scale parameterization of each distribution, corresponding to the values of hyperparam_1 and hyperparam_2, with its respective mean and variance. Table 4 includes also the distributions used in the experiment with different observations models, namely Gumbel, Laplace and Normal distribution. In this experiment the location parameter was set to the conditional mean $\mathbb{E}[Y_{ij}|\theta,\beta] = \sum_{k=1}^{K} \theta ik\beta_{jk}$ and the scale (for each distribution) with values $\{0.5, 1.0, \ldots, 3.5, 4.0, 4.5, \ldots, 7.5, 8.0\}$, leading to the expected conditional variance $\mathbb{E}[\mathbb{V}(Y_{ij}|\theta,\beta)]$ having values according to the formulas for the variance in Table 4 (for Gumbel, Laplace and Normal distributions).

Appendix B. Derivation of Analytic Solution for Poisson Matrix Factorization

We consider the model defined in Eq. (1). We reinstate that for the sake of notational convenience we will adopt the following notation $\lambda \overset{\text{def}}{=} \{\mu_\theta, \sigma_\theta^2, \mu_\beta, \sigma_\beta^2\}$. We are interested in computing $\mathbb{E}[Y_{ij};\lambda]$, $\mathbb{V}[Y_{ij};\lambda]$, $\text{Cov}[Y_{ij}, Y_{tl};\lambda]$ and $\rho[Y_{ij}, Y_{tl};\lambda]$ denotating here the virtual statistics derived from PDD. We want to marginalize out the latent variables to be able to calculate a mathematical relation between the hyperparameters and the virtual statistics.[8]

[7] `https://numpy.org/doc/stable/`

[8] For notation brevity assume the hyperparameter λ implicitily, for example $\mathbb{E}[Y_{ij}] \overset{\text{def}}{=} \mathbb{E}[Y_{ij};\lambda] = \int Y_{ij} p(Y_{ij}|\sum_k \theta_{ik}\beta_{jk}) \prod_k p(\theta_{ik};\lambda)p(\beta_{jk};\lambda) d\theta_{ik} d\beta_{jk} dY_{ij}$.

Table 3: Set of hyperparameters configurations for the experiments with multiple priors.

θ hyperparam_1	θ hyperparam_2	β hyperparam_1	β hyperparam_2	prior
1.0	1.0	1.0	1.0	gamma
10.0	1.0	10.0	1.0	gamma
1.0	10.0	1.0	10.0	gamma
0.1	1.0	1.0	0.1	gamma
1.0	1.0	1.0	1.0	beta
1.0	2.0	1.0	1.0	beta
1.0	2.0	1.0	2.0	beta
1.0	2.0	2.0	1.0	beta
0.0	1.0	0.0	1.0	halfnorm
0.0	2.0	0.0	2.0	halfnorm
0.5	2.0	1.0	2.0	halfnorm
0.5	0.5	2.0	1.0	halfnorm
1.0	1.0	0.0	1.0	lognorm
1.0	2.0	0.0	2.0	lognorm
1.0	2.0	0.5	0.5	lognorm
1.0	0.5	0.5	2.0	lognorm

Table 4: Parameterization of different distributions

distribution	location, scale	$\mathbb{E}$	$\mathbb{V}$
Gamma(a,b)	a,b (shape and scale)	ab	ab^2
Beta(a,b)	a,b	$\frac{a}{a+b}$	$\frac{ab}{(a+b)^2(a+b+1)}$
HalfNormal(μ,σ)	μ,σ	$\mu+\sigma\sqrt{\frac{2}{\pi}}$	$\sigma^2(1-\frac{2}{\pi})$
LogNormal(μ,σ)	μ,σ	$\exp(\mu+\frac{\sigma^2}{2})$	$(\exp(\sigma^2)-1)\exp(2\mu+\sigma^2)$
Gumbel(μ,σ)	μ,σ	$\mu+\gamma\sigma$	$\frac{\pi^2}{2}\sigma^2$
Laplace(μ,b)	μ,σ	μ	$\frac{\pi^2}{2}\sigma^2$
Normal(μ,σ^2)	μ,σ	μ	σ^2
Poisson(η)	η	η	η

B.1 Preliminaries

We will make use of Total Expectation Law, Total Variance Law and Total Covariance Law and the following propositions without proofs:

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}[\mathbb{E}[Y|X]] && (14)\\
\mathbb{V}[Y] &= \mathbb{E}[\mathbb{V}[Y|X]] + \mathbb{V}[\mathbb{E}[Y|X]] && (15)\\
\mathrm{Cov}[X,Y] &= \mathbb{E}[\mathrm{Cov}[X,Y|Z]] + \mathrm{Cov}[\mathbb{E}[X|Z],\mathbb{E}[Y|Z]] && (16)\\
\mathbb{V}[XY] &= \mathbb{E}[X^2]\,\mathbb{E}[Y^2] - \mathbb{E}[X]^2\,\mathbb{E}[Y]^2 \text{ if X and Y are independent} && (17)\\
\mathbb{V}[XY] &= \mathbb{E}[X]^2\,\mathbb{V}[Y] + \mathbb{E}[Y]^2\,\mathbb{V}[X] + \mathbb{V}[X]\,\mathbb{V}[Y] \text{ if X and Y are independent} && (18)\\
\mathbb{E}[X^2] &= \mathbb{V}[X] + \mathbb{E}[X]^2 && (19)\\
\mathbb{V}[\sum_k X_k] &= \sum_k \mathbb{V}[X_k] + 2\sum_{k<k'} \mathrm{Cov}[X_k, X_{k'}]. && (20)
\end{aligned}$$

These relations will be useful in the task of marginalizing out the latent parameters of the hierarchical model, as we shall see in the following section.

B.2 Intermediate results

We will start by providing some intermediate results that will be useful in different steps to obtain the final results.

Proposition 3 *For any combination of valid values for the indexes i,j,t and l, if the latent indexes $k \neq k'$, then $Cov[\theta_{ik}\beta_{jk}, \theta_{tk'}\beta_{lk'}] = 0$.*

Proof By definition of the covariance

$$\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{tk'}\beta_{lk'}] = \mathbb{E}[\theta_{ik}\beta_{jk}\theta_{tk'}\beta_{lk'}] - \mathbb{E}[\theta_{ik}\beta_{jk}]\,\mathbb{E}[\theta_{tk'}\beta_{lk'}].$$

Given that $k \neq k'$, this implies (for any combination of the other indices) $\mathbb{E}[\theta_{ik}\beta_{jk}\theta_{tk'}\beta_{lk'}] = \mathbb{E}[\theta_{ik}\beta_{jk}]\,\mathbb{E}[\theta_{tk'}\beta_{lk'}]$. ■

Proposition 4 *For any combination of valid values for the indexes i,j and k the following equations hold:*

1. $\mathbb{E}[\sum_k \theta_{ik}\beta_{jk}] = K\mu_\theta\mu_\beta$;
2. $\mathbb{V}[\sum_k \theta_{ik}\beta_{jk}] = K[(\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2]$;
3. $Cov[\sum_k \theta_{ik}\beta_{jk}, \sum_k \theta_{tk}\beta_{lk}] = K[\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\sigma_\theta\sigma_\beta)^2)]$.

Proof :

1. For the first equation we apply the summation property of the expected value and the fact that θ_{ik} and β_{jk} are independent.
2. For $\mathbb{V}[\sum_k \theta_{ik}\beta_{jk}]$, we start by using Eq. 20, thus resulting in $\mathbb{V}[\sum_k \theta_{ik}\beta_{jk}] = \sum_k \mathbb{V}[\theta_{ik}\beta_{jk}] + 2\sum_{k<k'} \mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{ik'}\beta_{jk'}]$. However, from Proposition 3 we know the covariance terms where the indexes k and k' are not the same should be zero, resulting in

$\mathbb{V}[\sum_k \theta_{ik}\beta_{jk}] = \sum_k \mathbb{V}[\theta_{ik}\beta_{jk}]$. Now using Eq. (18) for the variance of the product of random variables we obtain

$$\begin{aligned}
\mathbb{V}[\theta_{ik}\beta_{jk}] &= \mathbb{E}[\theta_{ik}]^2\,\mathbb{V}[\beta_{jk}] + \mathbb{E}[\beta_{jk}]^2\,\mathbb{V}[\theta_{ik}] + \mathbb{V}[\theta_{ik}]\,\mathbb{V}[\beta_{jk}] \\
&= \mu_\theta^2\sigma_\beta^2 + \mu_\beta^2\sigma_\theta^2 + \sigma_\beta^2\sigma_\theta^2 \\
\Rightarrow \mathbb{V}[\sum_k \theta_{ik}\beta_{jk}] &= K[(\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2].
\end{aligned}$$

3. For the last equation we start with the definition of covariance:

$$\begin{aligned}
\mathrm{Cov}[\sum_k \theta_{ik}\beta_{jk}, \sum_k \theta_{tk}\beta_{lk}] &= \mathbb{E}[\sum_{k,k'} \theta_{ik}\beta_{jk}\theta_{tk'}\beta_{lk'}] - \mathbb{E}[\sum_k \theta_{ik}\beta_{jk}]\,\mathbb{E}[\sum_{k'} \theta_{tk'}\beta_{lk'}] \\
&= \sum_{k,k'} \underbrace{\mathbb{E}[\theta_{ik}\beta_{jk}\theta_{tk'}\beta_{lk'}] - \mathbb{E}[\theta_{ik}]\,\mathbb{E}[\beta_{jk}]\,\mathbb{E}[\theta_{tk'}]\,\mathbb{E}[\beta_{lk'}]}_{\mathrm{Cov}[\theta_{ik}\beta_{jk},\theta_{tk'}\beta_{lk'}]}.
\end{aligned}$$

Considering Proposition 3, we know that only the shared indices k are non zero, thus simplyfiying to

$$\begin{aligned}
\mathrm{Cov}[\sum_k \theta_{ik}\beta_{jk}, \sum_k \theta_{tk}\beta_{lk}] &= \sum_k \mathbb{E}[\theta_{ik}\beta_{jk}\theta_{tk}\beta_{lk}] - \mathbb{E}[\theta_{ik}]\,\mathbb{E}[\beta_{jk}]\,\mathbb{E}[\theta_{tk}]\,\mathbb{E}[\beta_{lk}] \qquad (21) \\
&= \sum_k \mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{tk}\beta_{lk}].
\end{aligned}$$

Now, we can calculate $\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{tk}\beta_{lk}]$ for four different cases:

(a) if $i \neq t \;\&\; j \neq l$: because of independence of all variables, we obtain

$$\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{tk}\beta_{lk}] = 0;$$

(b) if $i = t \;\&\; j \neq l$:

$$\begin{aligned}
\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{ik}\beta_{lk}] &= \mathbb{E}[\theta_{ik}^2\beta_{jk}\beta_{lk}] - \mathbb{E}[\theta_{ik}]^2\,\mathbb{E}[\beta_{jk}]\,\mathbb{E}[\beta_{lk}] \\
&= \mathbb{E}[\theta_{ik}^2]\,\mathbb{E}[\beta_{jk}]E[\beta_{lk}] - \mathbb{E}[\theta_{ik}]^2\,\mathbb{E}[\beta_{jk}]\,\mathbb{E}[\beta_{lk}] \\
&= \mathbb{E}[\beta_{jk}]E[\beta_{lk}](\mathbb{E}[\theta_{ik}^2] - \mathbb{E}[\theta_{ik}]^2) = \mu_\beta^2\sigma_\theta^2;
\end{aligned}$$

(c) if $i \neq t \;\&\; j = l$:

$$\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{tk}\beta_{jk}] = \mathbb{E}[\beta_{jk}^2]\,\mathbb{E}[\theta_{ik}]E[\theta_{tk}] - \mathbb{E}[\beta_{jk}]^2\,\mathbb{E}[\theta_{ik}]E[\theta_{tk}] = \mu_\theta^2\sigma_\beta^2;$$

(d) if $i = t \;\&\; j = l$:

$$\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{ik}\beta_{jk}] = \mathbb{V}[\theta_{ik}\beta_{jk}] = (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2.$$

Putting all together using Kronecker delta for the indices in the different cases we obtain

$$\mathrm{Cov}[\theta_{ik}\beta_{jk}, \theta_{tk}\beta_{lk}] = \delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\sigma_\theta\sigma_\beta)^2). \qquad (22)$$

We obtain the final results combining Eq. (21) and Eq. (22)

$$\Rightarrow \mathrm{Cov}[\sum_k \theta_{ik}\beta_{jk}, \sum_k \theta_{tk}\beta_{lk}] = K[\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\sigma_\theta\sigma_\beta)^2)].$$

■

B.3 Expected values and variance

We can now proceed to calculate the prior predictive expected value and variance.

Proposition 5 *For any combination of valid values for the indexes i,j and the following equations hold:*

1. $\mathbb{E}[Y_{ij}] = K\mu_\theta\mu_\beta$;
2. $\mathbb{V}[Y_{ij}] = K[\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2]$.

Proof By the law of total expectation, $\mathbb{E}[Y_{ij}] = \mathbb{E}[\mathbb{E}[Y_{ij}|\sum_k \theta_{ik}\beta_{jk}]] = \mathbb{E}[\sum_k \theta_{ik}\beta_{jk}]$, already calculated in Proposition 4. For the second equation we use the law of total variance (Eq. (15)) $\mathbb{V}[Y_{ij}] = \mathbb{E}[\mathbb{V}[Y_{ij}|\sum_k \theta_{ik}\beta_{jk}]] + \mathbb{V}[\mathbb{E}[Y_{ij}|\sum_k \theta_{ik}\beta_{jk}]]$, and because we have a Poisson likelihood we know that $\mathbb{V}[Y_{ij}|\sum_k \theta_{ik}\beta_{jk}] = \mathbb{E}[Y_{ij}|\sum_k \theta_{ik}\beta_{jk}] = \sum_k \theta_{ik}\beta_{jk}$. Now putting both together and using Proposition 4 we obtain

$$\mathbb{V}[Y_{ij}] = \mathbb{E}[\sum_k \theta_{ik}\beta_{jk}] + \mathbb{V}[\sum_k \theta_{ik}\beta_{jk}] = K[\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2].$$

■

B.4 Covariance and correlation

Finally, combining the previous results we can obtain the covariance and correlation for the prior predictive distribution

Proposition 6 *The prior predictive covariance is given by*

$$Cov[Y_{ij}, Y_{tl}] = K[\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\mu_\theta\mu_\beta + (\sigma_\theta\sigma_\beta)^2)].$$

Proof Using the law of total covariance (Eq. (16)):

$$\begin{aligned}
\mathrm{Cov}[Y_{ij}, Y_{tl}] &= \mathbb{E}[\mathrm{Cov}[Y_{ij}, Y_{tl}|\theta_{i.}, \beta_{j.}, \theta_{t.}, \beta_{l.}]] + \mathrm{Cov}[\mathbb{E}[Y_{ij}|\theta_{i.}, \beta_{j.}], \mathbb{E}[Y_{tl}|\theta_{t.}, \beta_{l.}]] \\
&= \mathbb{E}[\delta_{it}\delta_{jl}\,\mathbb{V}[Y_{ij}|\theta_{i.}, \beta_{j.}]] + \mathrm{Cov}[\sum_k \theta_{ik}\beta_{jk}, \sum_k \theta_{tk}\beta_{lk}] \\
&= \mathbb{E}[\delta_{it}\delta_{jl}\sum_k \theta_{ik}\beta_{jk}] + K[\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\sigma_\theta\sigma_\beta)^2)] \\
&= K[\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\mu_\theta\mu_\beta + (\sigma_\theta\sigma_\beta)^2)].
\end{aligned}$$

■

Proposition 7 *The prior predictive correlation is given by*

$$\rho[Y_{ij}, Y_{tl}] = \frac{\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\mu_\theta\mu_\beta + (\sigma_\theta\sigma_\beta)^2)}{\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2}$$

or alternatively

$$\rho[Y_{ij}, Y_{tl}] = \begin{cases} 0, \textit{if } i \neq t \ \ \& \ \ j \neq l \\ 1, \textit{if } i = t \ \ \& \ \ j = l \\ \rho_1 = \frac{(\mu_\beta\sigma_\theta)^2}{\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2}, \textit{if } i = t \ \ \& \ \ j \neq l \\ \rho_2 = \frac{(\mu_\theta\sigma_\beta)^2}{\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2}, \textit{if } i \neq t \ \ \& \ \ j = l \end{cases}.$$

Proof From the definition of correlation we have

$$\begin{aligned} \rho[Y_{ij}, Y_{tl}] &= \frac{\operatorname{Cov}[Y_{ij}, Y_{tl}]}{\sqrt{\mathbb{V}[Y_{ij}]\,\mathbb{V}[y_{tlj}]}} = \frac{\operatorname{Cov}[Y_{ij}, Y_{tl}]}{\sqrt{\mathbb{V}[Y_{ij}]^2}} \\ &= \frac{\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\mu_\theta\mu_\beta + (\sigma_\theta\sigma_\beta)^2)}{\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2}. \end{aligned}$$

■

B.5 Finding the hyperparameters given the moments

Proposition 8 *Given that we know K, $\mathbb{E}[Y_{ij}]$, $\mathbb{V}[Y_{ij}]$, ρ_1 and ρ_2 the following equations hold and can be used for determining the hyperparameters:*

$$\sigma_\theta\sigma_\beta = \frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\sqrt{\rho_1\rho_2} \tag{23}$$

$$\left(\frac{\sigma_\beta}{\mu_\beta}\right)^2 = K\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]^2}\rho_2 \tag{24}$$

$$\left(\frac{\sigma_\theta}{\mu_\theta}\right)^2 = K\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]^2}\rho_1 \tag{25}$$

$$\rho_1\left(\frac{\sigma_\beta}{\mu_\beta}\right)^2 = \rho_2\left(\frac{\sigma_\theta}{\mu_\theta}\right)^2. \tag{26}$$

Proof We can rewrite the columns correlation ρ_1 and row correlation ρ_2 equations from Proposition 7 as

$$\rho_1\frac{\mathbb{V}[Y_{ij}]}{K} = (\mu_\beta\sigma_\theta)^2 \tag{27}$$

$$\rho_2\frac{\mathbb{V}[Y_{ij}]}{K} = (\mu_\theta\sigma_\beta)^2. \tag{28}$$

Multiplying them together we obtain

$$\rho_1\rho_2\left(\frac{\mathbb{V}[Y_{ij}]}{K}\right)^2 = \underbrace{(\mu_\beta\mu_\theta)}_{\frac{\mathbb{E}[Y_{ij}]}{K}}^2(\sigma_\beta\sigma_\theta)^2$$

$$\implies \rho_1\rho_2\left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2 = (\sigma_\beta\sigma_\theta)^2. \tag{29}$$

Taking the root of Eq. (29) completes the proof for Eq. (23).

Now, using Eq. (29), Eq. (27) and Eq. (28), we will obtain the value of $\frac{(\sigma_\beta\sigma_\theta)^2}{(\mu_\beta\sigma_\theta)^2}$ and $\frac{(\sigma_\beta\sigma_\theta)^2}{(\mu_\theta\sigma_\beta)^2}$:

$$\frac{(\sigma_\beta\sigma_\theta)^2}{(\mu_\beta\sigma_\theta)^2} = \frac{\sigma_\beta^2}{\mu_\beta^2} = \rho_1\rho_2\left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2\frac{K}{\rho_1\,\mathbb{V}[Y_{ij}]} = \rho_2 K\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]^2} \tag{30}$$

$$\frac{(\sigma_\beta\sigma_\theta)^2}{(\mu_\theta\sigma_\beta)^2} = \frac{\sigma_\theta^2}{\mu_\theta^2} = \rho_1\rho_2\left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2\frac{K}{\rho_2\,\mathbb{V}[Y_{ij}]} = \rho_1 K\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]^2}. \tag{31}$$

Finally, dividing Eq. (27) by Eq. (27) we obtain the last result that completes the proof.
■

Proposition 9 *Given that we know $\mathbb{E}[Y_{ij}]$, $\mathbb{V}[Y_{ij}]$, ρ_1 and ρ_2, we can obtain the number of latent factors K and coefficient of variation ($\frac{\sigma}{\mu}$) of the priors of the Poisson factorization model that would generate data to match those moments:*

$$K = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_1\rho_2}\left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2 \tag{32}$$

$$\left(\frac{\sigma_\theta}{\mu_\theta}\right)^2 = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_2\,\mathbb{V}[Y_{ij}]} \tag{33}$$

$$\left(\frac{\sigma_\beta}{\mu_\beta}\right)^2 = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_1\,\mathbb{V}[Y_{ij}]}. \tag{34}$$

Proof From Proposition 5, we can rewrite the expression for the variance as

$$\mathbb{V}[Y_{ij}] = \mathbb{E}[Y_{ij}] + K\underbrace{(\mu_\theta\sigma_\beta)}_{\rho_2\frac{\mathbb{V}[Y_{ij}]}{K}}^2 + K\underbrace{(\mu_\beta\sigma_\theta)}_{\rho_1\frac{\mathbb{V}[Y_{ij}]}{K}}^2 + K(\sigma_\theta\sigma_\beta).$$

Now, using Eq. (27) and Eq. (27) to substitute in the previous equation we obtain:

$$\mathbb{V}[Y_{ij}] = \mathbb{E}[Y_{ij}] + (\rho_1+\rho_2)\,\mathbb{V}[Y_{ij}] + K(\sigma_\theta\sigma_\beta)^2.$$

Using the squared version of Eq. (23) from Proposition 8, we know that $K(\sigma_\theta \sigma_\beta)^2 = K\left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2 \rho_1\rho_2$. This results in

$$\begin{aligned} K\left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2 \rho_1\rho_2 &= (1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}] \\ \implies K &= \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_1\rho_2}\left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2. \qquad (35) \end{aligned}$$

The remaining results are obtained by substituting Eq. (35) in Eq. (27) and Eq. (28). ■

B.5.1 Gamma priors

For gamma priors parameterized with shape (a,c) and rate (b,d) we have:

$$\mu_\theta = \frac{a}{b}; \sigma_\theta^2 = \frac{a}{b^2}; \mu_\beta = \frac{c}{d}; \sigma_\beta^2 = \frac{c}{d^2}$$

and the cofficient of variation is given by

$$\frac{\sigma_\theta^2}{\mu_\theta^2} = \frac{a}{b^2}\frac{b^2}{a^2} = \frac{1}{a} \qquad (36)$$

$$\frac{\sigma_\beta^2}{\mu_\beta^2} = \frac{d^2}{c^2}\frac{c}{d^2} = \frac{1}{c}. \qquad (37)$$

Thus, Eq. (36) and Eq. (36) establish a closed-form relationship between the shape hyperparameters of Gamma distributed latent variables in Poisson MF and moments of the marginal distribution of the data. This means that any assumption that the expert might have about those moments on the data can be readily translated into appropriate values for the prior specification.

In conclusion, given the chosen moments, the prior especification of Gamma-Poisson MF model reduces to one degree of freedom, given that the latent dimensionality, and shape hyperparameters are determined. The only two hyperparameters left are the rate/scale, although they would be restriced to be obey a relationship with functional form

$$b \propto \frac{1}{d}.$$

Proposition 10 *Given that we know the moments $\mathbb{E}[Y_{ij}]$, $\mathbb{V}[Y_{ij}]$, ρ_1 and ρ_2, we can obtain the scale parameters of the Gamma priors speficied as $F(\mu_\theta, \sigma_\theta^2) = Gamma(a,b)$ and $F(\mu_\beta, \sigma_\beta^2) = Gamma(c,d)$ in the Gamma-Poisson factorization model such that the prior predictive moments would match those given moments.*

$$\frac{1}{a} = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_2\,\mathbb{V}[Y_{ij}]} \qquad (38)$$

$$\frac{1}{c} = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[Y_{ij}]}{\rho_1\,\mathbb{V}[Y_{ij}]} \qquad (39)$$

Proof Immediate from Proposition 9 and the parameterization of Gamma distribution discussed. ■

Also we can rewrite Eq. 23 with Gamma parameterization to obtain

$$\begin{aligned} \frac{a}{b^2}\frac{c}{d^2} &= \sigma_\theta^2\sigma_\theta^2 = \left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2 \rho_1\rho_2 \\ \implies (bd)^2 &= \left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2 \frac{ac}{\rho_1\rho_2} \\ \implies bd &= \frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\sqrt{\frac{ac}{\rho_1\rho_2}}. \end{aligned} \tag{40}$$

Appendix C. Derivation of Analytic Solution for Compound Poisson Matrix Factorization

We will work with the Exponential Dispersion models (EDM) family of observation that makes Compound Poisson matrix factorization models. Keeping the same notation of the previous section, but adding variable N_{ui} as a Poisson distributed latent count factor of the ED model. With abuse of notation, for example this model allow for observations of the type $Y_{ij} = \sum_{i=1}^{N_{ij}} \mathcal{N}(1,1)$, where N_{ij} is a Poisson random variable, extending Poisson factorization to the domain of real valued observations. Also, from the additive properties Jorgensen (1987); Basbug and Engelhardt (2016) of EDM models, $Y_{ij} = \sum_{i=1}^{N_{ij}} Y_{ijk}$ with $Y_{ijk} \overset{\text{iid}}{\sim} \text{ED}(w,\kappa)$ is equivalent to $Y_{ij} \sim \text{ED}(w, \kappa N_{ij})$. Thus, the Compound Poisson Matrix Factorization (CPMF) model we use is defined as

$$\begin{aligned} Y_{ij} &\sim \text{ED}(w, \kappa N_{ij}) \\ N_{ij} &\sim \text{Poisson}(\sum_k \theta_{ik}\beta_{jk}) \\ \theta_{ik} &\sim F(\mu_\theta, \sigma_\theta^2) \\ \beta_{jk} &\sim F(\mu_\beta, \sigma_\beta^2), \end{aligned}$$

where $p(Y_{ij}|N_{ij}; w, \kappa) = \exp(Y_{ij}w - \kappa N_{ij}\psi(w))h(Y_{ij}, \kappa N_{ij})$, $\mathbb{E}[Y_{ij}|N_{ij}; w, \kappa] = \kappa N_{ij}\psi'(w)$ and $\mathbb{V}[Y_{ij}|N_{ij}; w, \kappa] = \kappa N_{ij}\psi''(w)$.

C.1 Mean, variance, covariance and correlation

Proposition 11 *For any combination of valid values for the indexes i,j, the following equations hold:*

1. $\mathbb{E}[Y_{ij}] = \kappa\psi'(w)K\mu_\theta\mu_\beta$;
2. $\mathbb{V}[Y_{ij}] = \kappa\psi''(w)K\mu_\theta\mu_\beta + (\kappa\psi'(w))^2 K[\mu_\theta\mu_\beta + (\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2]$.

Proof By the law of total expectation and the properties of the mean of ED family, $\mathbb{E}[Y_{ij}] = \mathbb{E}[\kappa\psi'(w)N_{ij}]$, which simplifies to $\mathbb{E}[Y_{ij}] = \kappa\psi'(w)\,\mathbb{E}[N_{ij}]$ and from Proposition 5 we know the expected value of the latent Poisson count N_{ij}, concluding that $\mathbb{E}[Y_{ij}] = \kappa\psi'(w)K\mu_\theta\mu_\beta$. Using the law of total variance $\mathbb{V}[Y_{ij}] = \mathbb{E}[\mathbb{V}[Y_{ij}|N_{ij}]] + \mathbb{V}[\mathbb{E}[Y_{ij}|N_{ij}]$, that simplifies to

$$\mathbb{V}[Y_{ij}] = \kappa\psi''(w)\,\mathbb{E}[N_{ij}] + [\kappa\psi'(w)]^2\,\mathbb{V}[N_{ij}],$$

and again substituting Proposition 5 completes the proof. ■

Proposition 12 *The prior predictive correlation is given by:*

$$\rho[Y_{ij}, Y_{tl}] = \begin{cases} 0, \textit{if } i \neq t \;\;\&\;\; j \neq l \\ 1, \textit{if } i = t \;\;\&\;\; j = l \\ \rho_1, \textit{if } i = t \;\;\&\;\; j \neq l \\ \rho_2, \textit{if } i \neq t \;\;\&\;\; j = l \end{cases},$$

with

$$\rho_1 = \frac{K[\kappa\psi'(w)]^2}{\mathbb{V}[Y_{ij}]}(\mu_\beta\sigma_\theta)^2$$
$$\rho_2 = \frac{K[\kappa\psi'(w)]^2}{\mathbb{V}[Y_{ij}]}(\mu_\theta\sigma_\beta)^2.$$

Proof Starting with the covariance, we apply the law of total covariance to obtain

$$\mathrm{Cov}[Y_{ij}, Y_{tl}] = \delta_{it}\delta_{jl}\kappa\psi''(w)\,\mathbb{E}[N_{ij}] + [\kappa\psi'(w)]^2\,\mathrm{Cov}[N_{ij}, N_{tl}]. \tag{41}$$

When all the indices coincide this will be equal to the variance, thus leading to a correlation of 1, when all the indices are different this will lead to correlation of zero. This means that the main difference between the prior predictive correlation structure of CPMF and PMF will be where there is rows and columns correlation, that we will be able to calculate because we know the covariance $\mathrm{Cov}[N_{ij}, N_{tl}]$ from Proposition 6, namely

$$\rho_1 = \frac{[\kappa\psi'(w)]^2\,\mathrm{Cov}[N_{ij}, N_{il}]}{\mathbb{V}[Y_{i,j}]} = \frac{K[\kappa\psi'(w)]^2(\mu_\beta\sigma_\theta^2)}{\mathbb{V}[Y_{i,j}]},$$
$$\rho_2 = \frac{[\kappa\psi'(w)]^2\,\mathrm{Cov}[N_{ij}, N_{tj}]}{\mathbb{V}[Y_{i,j}]} = \frac{K[\kappa\psi'(w)]^2(\mu_\theta\sigma_\beta^2)}{\mathbb{V}[Y_{i,j}]}.$$

■

C.2 Finding the hyperparameters given the moments

Proposition 13 *For Compound Poisson MF, given that we know $\mathbb{E}[Y_{ij}]$, $\mathbb{V}[Y_{ij}]$, ρ_1 and ρ_2, we can obtain the number of latent factors K of the model that would generate data to match those moments:*

$$K = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \left(\kappa\psi'(w) + \frac{\psi''(w)}{\psi'(w)}\right)\mathbb{E}[Y_{ij}]}{\rho_1\rho_2}\left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2. \tag{42}$$

Proof We will start by showing that

$$\sigma_\theta \sigma_\beta = \frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]\kappa\psi'(w)}\sqrt{\rho_1\rho_2}. \tag{43}$$

Take ρ_1 and ρ_2 and multiply them to obtain:

$$\rho_1\rho_2 = \left(\frac{K[\kappa\psi'(w)]^2}{\mathbb{V}[Y_{ij}]}\right)^2 (\mu_\theta\mu_\beta)^2(\sigma_\theta\sigma_\beta)^2.$$

From Proposition 11, we know $K\mu_\theta\mu_\beta = \frac{\mathbb{E}[Y_{ij}]}{\kappa\psi'(w)}$, so we can substitute that on the previous equation obtaining

$$\rho_1\rho_2 = \left(\frac{[\kappa\psi'(w)]^2}{\mathbb{V}[Y_{ij}]}\right)^2 \left(\frac{\mathbb{E}[Y_{ij}]}{\kappa\psi'(w)}\right)^2 (\sigma_\theta\sigma_\beta)^2 = [\kappa\psi'(w)]^2 \left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2 (\sigma_\theta\sigma_\beta)^2.$$

Now let us turn our attention to $\mathbb{V}[Y_{ij}]$ and re-write it using the previous results together with Proposition 12 for the correlations, and Proposition 11 for the mean:

$$\begin{aligned}
\mathbb{V}[Y_{ij}] &= \psi''(w)\underbrace{\kappa K\mu_\theta\mu_\beta}_{\frac{\mathbb{E}[Y_{ij}]}{\psi'(w)}} + \underbrace{K[\kappa\psi'(w)]^2\mu_\theta\mu_\beta}_{\kappa\psi'(w)\,\mathbb{E}[Y_{ij}]} + K[\kappa\psi'(w)]^2[(\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2] \\
&= \left(\frac{\psi''(w)}{\psi'(w)} + \kappa\psi'(w)\right)\mathbb{E}[Y_{ij}] + \underbrace{K[\kappa\psi'(w)]^2(\mu_\beta\sigma_\theta)^2}_{\rho_1\,\mathbb{V}[Y_{ij}]} \\
&+ \underbrace{K[\kappa\psi'(w)]^2(\mu_\theta\sigma_\beta)^2}_{\rho_2\,\mathbb{V}[Y_{ij}]} + K[\kappa\psi'(w)]^2(\sigma_\theta\sigma_\beta)^2 \\
&= \left(\frac{\psi''(w)}{\psi'(w)} + \kappa\psi'(w)\right)\mathbb{E}[Y_{ij}] + (\rho_1+\rho_2)\,\mathbb{V}[Y_{ij}] + K\underbrace{[\kappa\psi'(w)]^2(\sigma_\theta\sigma_\beta)^2}_{\rho_1\rho_2\left(\frac{\mathbb{V}[Y_{ij}]]}{\mathbb{E}[Y_{ij}}\right)^2} \\
&= \left(\frac{\psi''(w)}{\psi'(w)} + \kappa\psi'(w)\right)\mathbb{E}[Y_{ij}] + (\rho_1+\rho_2)\,\mathbb{V}[Y_{ij}] + K\rho_1\rho_2\left(\frac{\mathbb{V}[Y_{ij}]]}{\mathbb{E}[Y_{ij}}\right)^2.
\end{aligned}$$

Reorganizing the terms and isolating K we obtain the final formula

$$K = \frac{(1-(\rho_1+\rho_2))\,\mathbb{V}[Y_{ij}] - \left(\kappa\psi'(w) + \frac{\psi''(w)}{\psi'(w)}\right)\mathbb{E}[Y_{ij}]}{\rho_1\rho_2}\left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]}\right)^2.$$

■

Appendix D. Generic Probabilistic Matrix Factorization

Consider the model defined in Eq. 3, with priors $F(\mu_\theta, \sigma_\theta^2)$ and $F(\mu_\beta, \sigma_\beta^2)$, and observation model F_Y, defined as:

$$\theta_{ik} \sim F(\mu_\theta, \sigma_\theta^2), \quad \beta_{jk} \sim F(\mu_\beta, \sigma_\beta^2)$$
$$Y_{ij} \sim F_Y(\sum_{k=1}^{K} \theta_{ik}\beta_{jk}), \text{ with } \mathbb{E}[Y_{ij}] = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}.$$

Proposition 14 *For any entry of the matrix $\mathbf{Y} = \{Y_{ij}\}$, the mean and variance is given by:*

$$\begin{aligned} \mathbb{E}[Y_{ij}] &= K\mu_\theta\mu_\beta && (44) \\ \mathbb{V}[Y_{ij}] &= \mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]] \\ &\quad + K[(\mu_\beta\sigma_\theta)^2 + (\mu_\theta\sigma_\beta)^2 + (\sigma_\theta\sigma_\beta)^2] && (45) \end{aligned}$$

Proof The equation for $\mathbb{E}[Y_{ij}]$ is obtained from the same steps shown for the case PMF and CPMF, and the fact that $\mathbb{E}[Y_{ij}] = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}$ by definition of the model.

The equation for $\mathbb{V}[Y_{ij}]$ is obtained also from the same steps shown before, but with the additional restriction that $\mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]]$ is model dependent and can not be simplified further, once we apply the law of total variance. ■

Proposition 15 *For any pair of entries Y_{ij} and Y_{tl} of the matrix $\mathbf{Y}$, their correlation is given by:*

$$\rho[Y_{ij}, Y_{tl}] = \begin{cases} 0, \textit{if } i \neq t \ \& \ j \neq l \\ 1, \textit{if } i = t \ \& \ j = l \\ \rho_1, \textit{if } i = t \ \& \ j \neq l \\ \rho_2, \textit{if } i \neq t \ \& \ j = l \end{cases} \quad (46)$$

$$\begin{aligned} \textit{with } \rho_1 &= \frac{K(\mu_\beta\sigma_\theta)^2}{\mathbb{V}[Y_{ij}]} \\ \rho_2 &= \frac{K(\mu_\theta\sigma_\beta)^2}{\mathbb{V}[Y_{ij}]} \end{aligned}$$

Proof Using the law of total covariance:

$$\begin{aligned} \mathrm{Cov}[Y_{ij}, Y_{tl}] &= \mathbb{E}[\mathrm{Cov}[Y_{ij}, Y_{tl}|\theta,\beta]] + \mathrm{Cov}[\mathbb{E}[Y_{ij}|\theta,\beta], \mathbb{E}[Y_{tl}|\theta,\beta]] \\ &= \delta_{it}\delta_{jl}\,\mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]] + \mathrm{Cov}[\sum_k \theta_{ik}\beta_{jk}, \sum_k \theta_{tk}\beta_{lk}] \\ &= \delta_{it}\delta_{jl}\,\mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]] + K[\delta_{it}(\mu_\beta\sigma_\theta)^2 + \delta_{jl}(\mu_\theta\sigma_\beta)^2 + \delta_{it}\delta_{jl}(\sigma_\theta\sigma_\beta)^2]. \end{aligned}$$

Analyzing this expression for the different cases we obtain:

- $i \neq t \;\&\; j \neq l$: $\mathrm{Cov}[Y_{ij}, Y_{tl}] = 0$
- $i = t \;\&\; j = l$: $\mathrm{Cov}[Y_{ij}, Y_{tl}] = \mathrm{Cov}[Y_{ij}, Y_{ij}] = \mathbb{V}[Y_{ij}]$
- $i = t \;\&\; j \neq l$: $\mathrm{Cov}[Y_{ij}, Y_{tl}] = K(\mu_\beta \sigma_\theta)^2$
- $i \neq t \;\&\; j = l$: $\mathrm{Cov}[Y_{ij}, Y_{tl}] = K(\mu_\theta \sigma_\beta)^2$

Applying the definition of correlation $\rho[Y_{ij}, Y_{tl}] = \frac{\mathrm{Cov}[Y_{ij},Y_{tl}]}{\sqrt{\mathbb{V}[Y_{ij}]\,\mathbb{V}[Y_{tl}]}} = \frac{\mathrm{Cov}[Y_{ij},Y_{tl}]}{\mathbb{V}[Y_{ij}]}$ we obtain the final equation in the proposition. ■

Given Propositions 14 and 15 and some target values for the moments, we can directly solve for the number of latent factors K. Denoting $\tau = 1 - (\rho_1 + \rho_2)$, we obtain our main result

Theorem 16

$$K = \frac{\tau\,\mathbb{V}[Y_{ij}] - \mathbb{E}[\mathbb{V}(Y_{ij}|\theta,\beta)]}{\rho_1 \rho_2} \left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]} \right)^2 . \tag{47}$$

Proof We can rewrite the expression of variance using the terms ρ_1 and ρ_2 resulting in

$$\mathbb{V}[Y_{ij}] = \mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]] + (\rho_1 + \rho_2)\,\mathbb{V}[Y_{ij}] + K(\sigma_\theta \sigma_\beta)^2$$

And given that $K(\sigma_\theta\sigma_\beta)^2 == K\rho_1\rho_2 \left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]}\right)^2$ and $\tau = 1 - (\rho_1 + \rho_2)$ we obtain

$$\begin{aligned} K\rho_1\rho_2 \left(\frac{\mathbb{V}[Y_{ij}]}{\mathbb{E}[Y_{ij}]} \right)^2 &= (1 - (\rho_1 + \rho_2))\,\mathbb{V}[Y_{ij}] - \mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]] \\ \implies K &= \frac{\tau\,\mathbb{V}[Y_{ij}] - \mathbb{E}[\mathbb{V}[Y_{ij}|\theta,\beta]]}{\rho_1\rho_2} \left(\frac{\mathbb{E}[Y_{ij}]}{\mathbb{V}[Y_{ij}]} \right)^2 \end{aligned}$$

■

Appendix E. Differentiable Moment's Estimators for Hierarchical Bayesian Models

This appendix provides the details on how to compute the gradients for the gradient-based approach described in Section 5. To optimize Eq. (13) with stochastic gradient descent, we require that $\mathrm{d}(\cdot)$ and $\hat{\mathrm{T}}$ are differentiable w.r.t their arguments, and that we can propagate gradient ∇_λ through $\mathbb{E}[g(Y)]$. Next we show how this can be done for a rather general structure of hierarchical Bayesian models with outputs Y and latent variables Z. See Figure 10 for a conceptual illustration of the assumed model structure and the procedure for computing the gradient. The procedure is based on recursively applying the law of total expectation. The unconditional expectation of $g(Y)$ can be obtained by integrating out latent

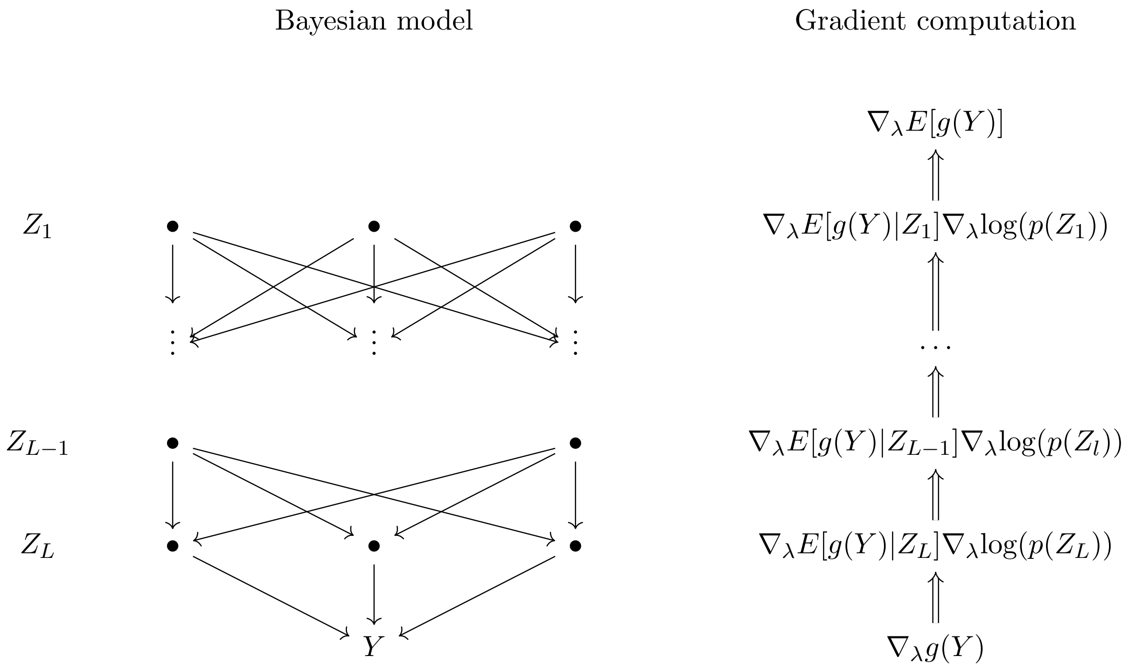
Bayesian model
Gradient computation
Z_1
Z_{L-1}
Z_L
Y
$\nabla_\lambda E[g(Y)]$
$\nabla_\lambda E[g(Y)|Z_1]\nabla_\lambda \log(p(Z_1))$
$\dots$
$\nabla_\lambda E[g(Y)|Z_{L-1}]\nabla_\lambda \log(p(Z_l))$
$\nabla_\lambda E[g(Y)|Z_L]\nabla_\lambda \log(p(Z_L))$
$\nabla_\lambda g(Y)$

Figure 10: Conceptual illustration of how the gradients for arbitrary moments can be estimated for Bayesian models with hierarchical structure of latent variables.

variables Z, but since an analytical form of it is not available, we proceed by performing a numerical approximation, where each of the integrals over latent variables $Z_1, \dots Z_l \dots Z_L$ is replaced by a sum over samples from respective (conditional) distributions. An estimate of the required gradient $\nabla_\lambda \mathbb{E}[g(Y)]$ is then obtained by propagating estimates of the gradients $\nabla_\lambda \mathbb{E}[g(Y)|z_l]$ and $\nabla_\lambda \log p(z_l|\dots;\lambda)$ backward through the computation graph.

Top level variables. Assuming all the remaining variables Z_{-1} integrated out, the expectation of $g(Y)$ can be expressed by conditioning only on variables Z_1 having no parents, i.e., latent variables being on top in the hierarchy of such a Bayesian model. For discrete Z_1

$$\mathbb{E}[g(Y)] = \sum_{z_1 \in \mathscr{Z}_1} \underbrace{\mathbb{E}[g(Y)|z_1]}_{f^1_\lambda(Y,z_1)} \cdot p(z_1;\lambda) \tag{48}$$

where we named $f^1_\lambda(Y,z_1) \equiv \mathbb{E}[g(Y)|z_1]$ to emphasize that the expectations here behave like ordinary functions. The gradient of the expectation is then given as

$$\nabla_\lambda \mathbb{E}[g(Y)] = \sum_{z_1 \in \mathscr{Z}_1} \nabla_\lambda f(Y,z_1) \cdot p(z_1;\lambda) + f(Y,z_1) \cdot \nabla_\lambda p(z_1;\lambda).$$

Whenever the number of possible discrete values, i.e., $|\mathscr{Z}_1|$, is too large or infinite (like for example, for Poisson distribution), the exact sum over all possible outcomes in Eq. (48) is replaced with a set of samples

$$\mathbb{E}[g(Y)] \approx \frac{1}{S_1} \sum_{z_1 \sim p(z_1;\lambda)} f^1_\lambda(Y, z_1) \cdot p(z_1; \lambda), \tag{49}$$

where S_1 denotes number of samples. Then, log derivative trick in (e.g., the DiCE incarnation by Foerster et al. 2018) is used to obtain unbiased estimates for the gradient

$$\nabla_\lambda \mathbb{E}[g(Y)] \approx \frac{1}{S_1} \sum_{z_1 \sim p(z_1;\lambda)} \nabla_\lambda f(Y, z_1) \cdot p(z_1; \lambda) + \underbrace{f(Y, z_1) \cdot \nabla_\lambda \log p(z_1; \lambda)}_{\text{log derivative trick}}, \tag{50}$$

The second term may incurr large variance of the gradient estimator but (even though we do not account for it in our experiments) this can be helped by variance reduction techniques (Mnih and Gregor, 2014; Mnih and Rezende, 2016; Tucker et al., 2017).

For continuous Z_1 the expectation is approximated using MC

$$\mathbb{E}[g(Y)] \approx \frac{1}{S_1} \sum_{z_1 \sim p(z_1;\lambda)} f^1_\lambda(Y, z_1) \tag{51}$$

and the gradient can be estimated by reparameterizing $z_1 := z_1(\epsilon_1, \lambda)$, i.e., expressing samples z_1 as a deterministic function of λ and another random variable ϵ_1 coming from a zero-parameter distribution p_0. The estimate is then

$$\nabla_\lambda \mathbb{E}[g(Y)] \approx \nabla_\lambda \left[\frac{1}{S_1} \sum_{\epsilon_1 \sim p_0(\epsilon_1)} f^1_\lambda \left(Y, z_1(\epsilon_1, \lambda)\right) \right] = \frac{1}{S_1} \sum_{\epsilon_1 \sim p_0(\epsilon_1)} \nabla_\lambda f^1_\lambda \left(Y, z_1(\epsilon_1, \lambda)\right) \tag{52}$$

where we omit the change-of-variables Jacobian for brevity. Even though the technical implementations of the path-wise (reparametrized) gradients (Figurnov et al., 2018) may differ from the conceptual presentation above, it does not affect our reasoning – usually the details are hidden by interface of an automatic-differentation library.

Inner latent variables. For both of the above cases (Z_1 discrete and continous) we need to be able to calculate the inner gradients of the conditional expectations $\nabla_\lambda f(Y, z_1) \equiv \nabla_\lambda \mathbb{E}[g(Y)|z_1]$, and in general, for any intermediate latent variables Z_l we need $\nabla_\lambda f(Y, z_l) \equiv \nabla_\lambda \mathbb{E}[g(Y)|z_l]$. It can be obtained by further expanding the expectations. For each l, we unwrap recursively the expectation by conditioning on variables' Z_l parents Z_{l-1}. In particular, for discrete Z_l we again have

$$\mathbb{E}[g(Y)|z_{l-1}] = \sum_{z_l \in \mathscr{Z}_l} \underbrace{\mathbb{E}[g(Y)|z_l]}_{f^l_\lambda(Y, z_l)} \cdot p(z_l|z_{l-1}; \lambda). \tag{53}$$

where Eq. (53) corresponds and has the same form as Eq. (48) and therefore the gradient $\nabla_\lambda \mathbb{E}[g(Y)|z_{l-1}]$ can be obtained similar as in Eq. (50) (with the exception that the probability mass function $p(z_l|z_{l-1}; \lambda)$ depends here also on samples z_{l-1} of the parents Z_{l-1}). Respectively, for continuous Z_l:

$$\mathbb{E}[g(Y)|z_{l-1}] \approx \frac{1}{S_l} \sum_{z_l \sim p(z_l;\lambda)} \underbrace{\mathbb{E}[g(Y)|z_l]}_{f^l_\lambda(Y, z_l)} = \frac{1}{S_l} \sum_{\epsilon_l \sim p_0} \mathbb{E}[g(Y)|z_l(\epsilon_l, \lambda, z_{l-1})], \tag{54}$$

where Eq. (54) corresponds and has the same form as Eq. (51) and allows obtaining its gradient similar as in Eq. (52). The only notable difference is again that the reparametrization $z_l := z_l(\epsilon_l, \lambda, z_{l-1})$ depends also on z_{l-1}.

Direct parents of observed variables. By recursively exploring the computation graph, we eventually arrive at the expectation of $g(Y)$ conditioned directly on its parents. For discrete Y it takes an exact form of

$$\mathbb{E}[g(Y)|z_L] = \sum_{y \in \mathscr{Y}} \underbrace{g(y)}_{f_\lambda^L(y)} \cdot p(y|z_L; \lambda), \tag{55}$$

where again we find the same structure as in Eq. (48) and Eq. (53), and the gradient can be obtained similar to Eq. (50). Note however that here $\nabla_\lambda f_\lambda^L(y) = \nabla_\lambda g(y)$ where we assume $\nabla_\lambda g(y)$ to be a known function. For continuous outputs Y, we again resort to the MC approximation

$$\mathbb{E}[g(Y)|z_L] \approx \frac{1}{S_y} \sum_{\epsilon_y \sim p_0} g(y(\epsilon_y, \lambda, z_L)), \tag{56}$$

where differentiability w.r.t λ is achieved by reparametrizing $y := y(\epsilon_y, \lambda, z_L)$. S_y is a number of MC samples and p_0 is a simple distribution free of hyperparameters λ. Finally, the reparametrized gradient similar to Eq. (52) is given as

$$\nabla_\lambda \mathbb{E}[g(Y)|z_L] \approx \frac{1}{S_y} \sum_{\epsilon_y \sim p_0} \nabla_\lambda g(y(\epsilon_y, \lambda, z_L)).$$

Mixed-type variables. If the set of variables denoted by Z_l consist of both continuous and discrete nodes, the above expressions need to be combined by summing over (or sampling from) the discrete variables and using MC approximation for the continuous ones.

E.1 Example: Derivation for PMF and HPF

To demonstrate the rather generic presentation above and to link it to the BMF use-case, we show as an example the MC estimate for the expectation $\mathbb{E}[Y_{ij}]$ of the PMF model

$$\mathbb{E}[Y] \approx \frac{1}{S_\theta \cdot S_\beta} \sum_{\epsilon_\theta \sim p_0} \sum_{\epsilon_\beta \sim p_0} \mathbb{E}[Y|\theta(\epsilon_\theta, \lambda)^T \beta(\epsilon_\beta, \lambda)], \tag{57}$$

where we reparametrize both β and θ. For clarity, we also dropped indices i and j in Y_{ij}, θ_i and β_j. The internal conditional expectation in Eq. (57) we expand as

$$\mathbb{E}[Y|\theta^T\beta] \approx \frac{1}{C} \sum_{y \sim \text{Poisson}(\theta^T\beta)} y \cdot \mathit{Poisson}(y|\theta^T\beta), \tag{58}$$

where we applied Eq. (49) and *Poisson* denotes Poisson probability mass distribution. Similarly, variance of Y we estimate with $\mathbb{V}[Y] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2$, where the estimator of $\mathbb{E}[Y^2]$ we obtain by substituting y with y^2 in Eq. (58).

To demonstrate the flexibility of the model-independent algorihm we also apply it on hierarchical Poisson factorization (HPF) model of Gopalan et al. (2015), for which we do

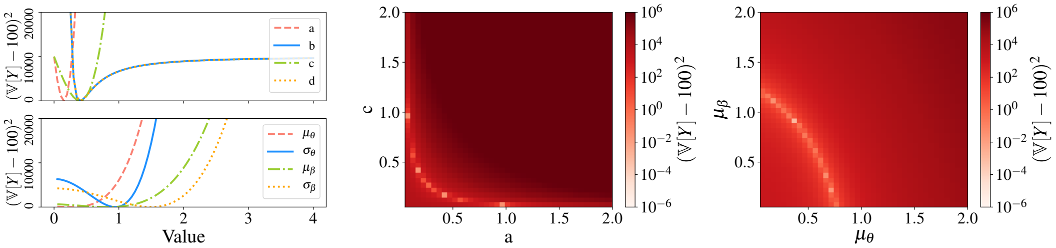

Figure 11: PMF ($K = 25$) parametrization using concentrations a, c and rates b, d vs. means μ_θ, μ_β and variances $\sigma_\theta^2, \sigma_\beta^2$: 1D and 2D projections of the optimization surface for matching $\mathbb{V}[Y] = 100$ in neighborhood of the optimal point ($a = 0.16$, $b = 0.4$, $c = 0.4$, $d = 0.4$).

not have closed-form expressions for the moments. This model adds one level of hierarchy to PMF, and hence matches our general formulation:

$$\begin{aligned} \theta &\sim \text{Gamma}(a, \xi), \quad & \xi &\sim \text{Gamma}(a', a'/b') \\ \beta &\sim \text{Gamma}(c, \eta), \quad & \eta &\sim \text{Gamma}(c', c'/d'), \end{aligned}$$

where the new continuous variables ξ, η we reparametrize (and sample) as follows:

$$\begin{aligned} \theta &:= \theta(\epsilon_\theta, a, \xi), \quad & \xi &:= \xi(\epsilon_\xi, a', a'/b') \\ \beta &:= \beta(\epsilon_\beta, c, \xi), \quad & \eta &:= \eta(\epsilon_\eta, c', c'/d'), \end{aligned}$$

Then, for HPF, Eq. (57) takes the form

$$\mathbb{E}[Y] \approx \frac{1}{S_\xi \cdot S_\eta} \sum_{\epsilon_\xi \sim p_0} \sum_{\epsilon_\eta \sim p_0} \underbrace{\frac{1}{S_\theta \cdot S_\beta} \sum_{\epsilon_\theta \sim p_0} \sum_{\epsilon_\beta \sim p_0} \mathbb{E}[Y|\theta(\epsilon_\theta, \lambda, \underline{\eta(\epsilon_\eta, \lambda)})^T \beta(\epsilon_\beta, \lambda, \underline{\eta(\epsilon_\eta, \lambda)})]}_{\mathbb{E}[Y|\xi,\eta]},$$

Appendix F. Parametrization of PMF

The performance of the stochastic optimization algorithm depends on the model parametrization, i.e., structure of the associated optimization space. Figure 11 compares two alternative parametrizations for PMF, one in terms of concentrations (a, c) and rates (b, d) and the other in terms of means (μ_θ, μ_β) and variances (σ_θ^2, σ_β^2). In this case, the latter results in diagonal Fisher information matrix and reduced correlations between the hyperparameters. This makes optimization easier, seen also as smoother optimization surface.

The question of optimal parameterization for general cases is clearly non-trivial problem, as it is for all optimization problems. One direction for improved parameterization of the prior distributions could build on the approach of Hartmann and Vanhatalo (2019), Tang and Ranganath (2019) and Gorinova et al. (2020), going towards a generic invariant to the parameterization of probabilistic programming models such as in Ollivier et al. (2017).

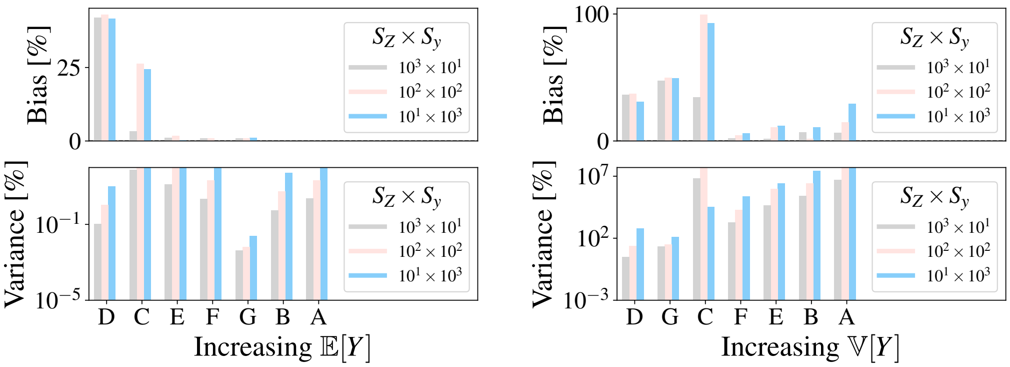

Figure 12: Biases and variances of the expected value (left) and variance (right) estimators for PMF ($K = 25$) with different hyperparameters' configurations (Table 1). S_Z denotes the number of latent variable samples and S_y is the number of samples of observed variable.

Appendix G. Bias and Variance of Generic Estimators

The model-independent stochastic algorithm relies on Monte Carlo estimates for the virtual statistics as explained in Section 5, instantiated for example cases in Eq. 19 and 20. Such estimates are not necessarily accurate, and hence we evaluate the bias and variance of the estimators against the closed-form solutions of Eq. 3 and 4, based on 10^3 independent runs.

Figure 12 compares variances and biases for the mean $\hat{\mathbb{E}}[Y]$ (top) and the variance $\hat{\mathbb{V}}[Y]$ (bottom) estimators for a range of hyperparameter configurations (Tables 1 and 2 in the Appendix), analyzing the effect of the number of samples used for estimating the statistics. We fix the total number of samples at $S_Z \times S_y = 10^4$, but vary the ratio between the number of samples for the latent variables (S_Z) and for the observed ones (S_y). Both variance and bias are minimized when more computational resources are spent on sampling the latent variables, suggesting the use of $S_Z \times S_y := 10^3 \times 10$.

The mean estimator has slight bias which is noticeable only for very small values ($\mathbb{E}[Y] < 1$), and the variance is usually lower than 50% of the mean estimate.

The variance estimator, however, may have significant bias and variance for values of $\mathbb{V}[Y]$, potentially disturbing the convergence for initializations C, D and G and raising a question regarding better estimators.

Appendix H. Adjustment for model mismatch

The model used in the experiment of Figure 8 in Section 7.3.1 has elements for each column controlled by a parameter, which is used to induce each columns to sum to a certain number. This model has a multiplicative factor γ_j for each latent rate that can be adjusted to induced the desired sum, which can be interprested as allocation factors controlling how much the latent rates are allocated to each column. The original equations for PMF are not applicable in this case, but it is possible to adjust the estimators for empirical mean, variance and covariance, such that we could still use the original equations obtained for PMF. In this section we show details about this result, as well as a simple experiment validating the approach.

Define $Y_{ij} \sim \text{Poisson}(\eta_{ij})$ with $\eta_{ij} = \sum_{k=1}^{K} \theta_{ik}\beta_{jk}$ as the original PMF matrix, and $\tilde{Y}_{ij} \sim \text{Poisson}(\gamma_j \eta_{ij})$ with $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_M]$ as the modified PMF model with the multiplicative factors. The conditional mean and variance of the modified PMF is $\mathbb{E}[\tilde{Y}_{ij}|\theta\beta] = \mathbb{V}[\tilde{Y}_{ij}|\theta\beta] = \gamma_j \eta_{ij} = \gamma_j\, \mathbb{E}[Y_{ij}|\theta\beta]$. The marginal expected and variance are obtained using the previous results resulting in $\mathbb{E}[\tilde{Y}_{ij}] = e_j = K\gamma_j\mu_\theta\mu_\beta = \gamma_j\, \mathbb{E}[Y_{ij}]$ and $\mathbb{V}[Y_{ij}] = v_j = e_j + \gamma_j^2\, \mathbb{V}[\eta_{ij}] = \gamma_j\, \mathbb{E}[Y_{ij}] + \gamma_j^2\, \mathbb{V}[\eta_{ij}]$.

Furthermore we can calculate the correlations between $\tilde{Y}_{ij}$ and $\tilde{Y}_{tl}$, and obtain a similar correlation structure that is zero when all the indices are different ($\rho(Y_{ij}, Y_{tl}) = 0$) and 1 when all indices are the same ($\rho(\tilde{Y}_{ij}, \tilde{Y}_{ij}) = 1$), but a column varying correlation for the remaining cases: for the fixed row $i = t$ and different columns $\rho(\tilde{Y}_{ij}, \tilde{Y}_{tl}) = \rho^{(1)}_{jl} = \frac{K\gamma_j\gamma_l}{\sqrt{v_j v_l}}(\mu_\beta\sigma_\theta)^2 = \gamma_j\gamma_l \frac{\text{Cov}(Y_{ij},Y_{tl})}{\sqrt{v_j v_l}}$, and for fixed columns $j = l$ and different rows $\rho(\tilde{Y}_{ij}, \tilde{Y}_{tl}) = \rho^{(2)}_{j} = \gamma_j^2 \frac{\text{Cov}(Y_{ij},Y_{tl})}{v_j}$. Making empirical estimates of these moments are challenging, specially if we intend to use a single observation matrix to estimate those quantities. It is possible to recover the original terms of the equations for PMF by observing the relationship between the obtained formulas and the moments for PMF.

Define $\hat{Y}^{(1)}_{ij} = \frac{\tilde{Y}_{ij}}{\gamma_j}$ and $\hat{Y}^{(2)}_{ij} = \frac{\tilde{Y}_{ij}}{\gamma_j^2}$. Now, $\mathbb{E}[\hat{Y}^{(1)}_{ij}] = \hat{e} = \mathbb{E}[Y_{ij}] = K\mu_\theta\mu_\beta$ which is the equation for the original PMF. Similarly we observe that for PMF, $\mathbb{V}[Y_{ij}] = \mathbb{E}[Y_{ij}] + \mathbb{V}[\eta_{ij}]$, while in the modified model we obtain $\mathbb{V}[\tilde{Y}_{ij}] = \gamma_j\, \mathbb{E}[Y_{ij}] + \gamma_j^2\, \mathbb{V}[\eta_{ij}]$, which implies that $\mathbb{V}[\hat{Y}^{(1)}_{ij}] = \frac{\mathbb{V}[\tilde{Y}_{ij}]}{\gamma_j^2} = \frac{\mathbb{E}[Y_{ij}]}{\gamma_j} + \mathbb{V}[\eta_{ij}]$. Since $\mathbb{E}[\hat{Y}^{(2)}_{ij}] = \frac{\mathbb{E}[Y_{ij}]}{\gamma_j}$, we can write $\mathbb{V}[\hat{Y}^{(1)}_{ij}] - \mathbb{E}[\hat{Y}^{(2)}_{ij}] = \mathbb{V}[\eta_{ij}]$, which is one of the terms in the original PMF equation for the variance, as well as being a term that does not vary with the column index. Furthermore, combining with the expected value $\mathbb{E}[\hat{Y}^{(1)}_{ij}]$ we obtain the result $\hat{v} = \mathbb{V}[\hat{Y}^{(1)}_{ij}] - \mathbb{E}[\hat{Y}^{(2)}_{ij}] + \mathbb{E}[\hat{Y}^{(1)}_{ij}] = \mathbb{E}[Y_{ij}] + \mathbb{V}[\eta_{ij}]$ which is precisely the equation for $\mathbb{V}[Y_{ij}]$ in the PMF model.

Given the bilinearity of the covariance, we have $\text{Cov}(Y^{(1)}_{ij}, Y^{(1)}_{tl}) = \frac{\text{Cov}(\tilde{Y}_{ij},\tilde{Y}_{tl})}{\gamma_j\gamma_l}$, which cancels with the terms $\gamma_j\gamma_l$ in the numerator when applied to the calculate the correlation coefficients. Finally, we obtain the original correlation of PMF using $\frac{\text{Cov}(Y^{(1)}_{ij},Y^{(1)}_{tl})}{\hat{v}}$. We can now use this equations to obtain empirical estimator using $\hat{Y}^{(1)}_{ij}$ and $\hat{Y}^{(2)}_{ij}$, and apply those in the same formula of the latent dimensionality K obtained for PMF.

The empirical validation setup consist of fixed priors Gamma$(1, 10)$ and Gamma$(1, 0.5)$, latent dimensionality $K \in \{100, 250, 500, 750, 1000\}$, and 10 repeated runs for each K. In each run, we sample a matrix $\{Y_{ij}\} \in \mathbb{R}^{1000\times1000}$ using the PMF model and another matrix $\{\tilde{Y}_{ij}\} \in \mathbb{R}^{1000\times1000}$ with the multiplicative factors $\boldsymbol{\gamma} = [1, \dots, \gamma_{max}]$ ($\gamma_{max} = 3.5$), using the same prior distribution (but resampling the latent factor for each case) and K. For the PMF matrix $\{Y_{ij}\}$ we can directly calculate empirical estimates of the mean, variance and correlations, and estimate the latent dimensionality $\hat{K}$ using Eq. (7). For the matrix $\{\tilde{Y}_{ij}\}$ of the modified model we calculate $\hat{Y}^{(1)}_{ij} = \frac{\tilde{Y}_{ij}}{\gamma_j}$ and $\hat{Y}^{(2)}_{ij} = \frac{\tilde{Y}_{ij}}{\gamma_j^2}$, and use empirical estimates of variance, expected value and covariance (that is used to calculate the correlations) according to the equations presented in the previous paragraph, finally calculating the estimated latent

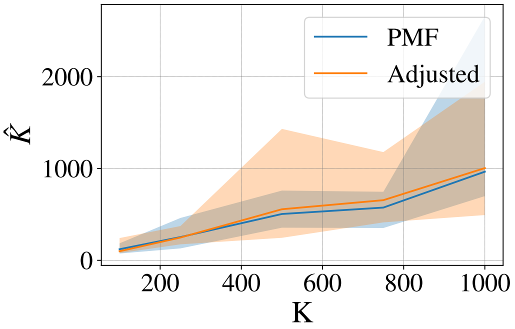

Figure 13: Comparing the estimated latent dimensionality $\hat{K}$ for PMF and the modified PMF with multiplicative factors, using the adjusted empirical estimates for variance, expected value and correlation (median over 10 runs, with CI=95%)

dimensionality $\hat{K}$ using the same Eq. (7). Figure 13 compares the results and shows that in both cases the estimate $\hat{K}$ recover the true K with similar degree of variability.

References

Artur Akbarov. *Probability elicitation: Predictive approach.* PhD thesis, University of Salford, 2009.

Anima Anandkumar, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, NeurIPS 2012, pages 917–925, 2012.

Lu Bai, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. Group sparse topical coding: From code to topic. In *ACM International Conference on Web Search and Data Mining*, WSDM 2013, page 315–324, 2013.

Mehmet Emin Basbug and Barbara E. Engelhardt. Hierarchical compound Poisson factorization. In *International Conference on Machine Learning*, ICML 2016, pages 1795–1803. PMLR, 2016.

Mark A Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406, 2010.

Charles Bouveyron, Pierre Latouche, and Pierre-Alexandre Mattei. Exact dimensionality selection for Bayesian PCA. *Scandinavian Journal of Statistics*, 2019.

Thomas Brouwer and Pietro Lio. Prior and likelihood choices for Bayesian matrix factorisation on small datasets. *arXiv preprint arXiv:1712.00288*, 2017.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In *ACM conference on Recommender systems*, RecSys 2011, 2011.

George Casella. An introduction to empirical bayes data analysis. *The American Statistician*, 39:83–87, 1985.

George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Press, 2 edition, 2001. ISBN 0534243126,9780534243128.

Ali Taylan Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Comp. Int. and Neurosc.*, 2009:785152:1–785152:17, 2009.

Ali Taylan Cemgil, Mehmet Burak Kurutmaz, Sinan Yildirim, Melih Barsbey, and Umut Simsekli. Bayesian allocation model: Inference by sequential monte carlo for nonnegative tensor factorizations and topic models using polya urns. *arXiv preprint arXiv:1903.04478*, 2019.

Jianzhun Du, Andrew Slavin Ross, Yonadav Shavit, and Finale Doshi-Velez. Controlled direct effect priors for bayesian neural networks. In *NeurIPS Bayesian deep learning workshop*, 2019.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, NeurIPS 2018, pages 439–450, 2018.

Jakob N. Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktaschel, Eric P. Xing, and Shimon Whiteson. Dice: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*, ICML 2018, pages 1524–1533. PMLR, 2018.

Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in Bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2):389–402, 2019.

Bruno Giovanni Galuzzi, Ilaria Giordani, Antonio Candelieri, Riccardo Perego, and Francesco Archetti. Bayesian optimization for recommender system. In *Optimization of Complex Systems: Theory, Models, Algorithms and Applications*, pages 751–760, 2019.

Paul H. Garthwaite, Joseph B. Kadane, and Anthony O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100: 680–701, 2005.

Andrew Gelman, Daniel Simpson, and Michael Betancourt. The prior can often only be understood in the context of the likelihood. *Entropy*, 19(10):555, 2017.

Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow. *arXiv preprint arXiv:2011.01808*, 2020.

Prem Gopalan, Laurent Charlin, and David M. Blei. Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems*, NeurIPS 2014, pages 3176–3184, 2014.

Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with hierarchical Poisson factorization. In *Conference on Uncertainty in Artificial Intelligence*, UAI 2015, pages 326–335. AUAI Press, 2015.

Maria I. Gorinova, Dave Moore, and Matthew D. Hoffman. Automatic reparameterisation of probabilistic programs. In *International Conference on Machine Learning*, ICML 2020, pages 3648–3657. PMLR, 2020.

Michael U. Gutmann and Jukka Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17 (125):1–47, 2016.

Marcelo Hartmann and Jarno Vanhatalo. Laplace approximation and natural gradient for Gaussian process regression with heteroscedastic Student-t model. *Statistics and Computing*, 29(4):753–773, Jul 2019.

Marcelo Hartmann, Georgi Agiashvili, Paul C. Bürkner, and Arto Klami. Flexible prior elicitation via the prior predictive distribution. In *Conference on Uncertainty in Artificial Intelligence*, UAI 2020, pages 1129–1138. AUAI Press, PMLR, 2020.

Bent Jorgensen. Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 49(2):127–162, 1987.

Joseph B. Kadane, James M. Dickey, Robert L. Winkler, Wayne S. Smith, and Stephen C. Peters. Interactive elicitation of opinion for a normal linear model. *Journal of the American Statistical Association*, 75:845–854, 1980.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.

Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, and Frank Hutter. Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *International Conference on Artificial Intelligence and Statistics*, AISTATS 2017. PMLR, 2017a.

Aaron Klein, Stefan Falkner, Numair Mansur, and Frank Hutter. RoBO: A flexible and robust Bayesian optimization framework in Python. In *NeurIPS 2017 Bayesian Optimization Workshop*, December 2017b.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning hierarchical priors in vaes. In *Advances in Neural Information Processing Systems*, NeurIPS 2019, 2019.

Keith Knight. *Mathematical Statistics*. CRC Texts in Statistical Science. Crc Pr Inc, 16 edition, 1999. ISBN 158488178X,9781584881780.

Jarno Lintusaari, Michael U Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Fundamentals and recent developments in approximate Bayesian computation. *Systematic biology*, 66(1):e66–e82, 2017.

Jean-Michel Marin, Pierre Pudlo, Christian P. Robert, and Robin J. Ryder. Approximate Bayesian computational methods. *Statistics and computing*, 22(6):1167–1180, 2012.

Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, 2003.

Petrus Mikkola, Osvaldo A. Martin, Suyog Chandramouli, Marcelo Hartmann, Oriol Abril Pla, Owen Thomas, Henri Pesonen, Jukka Corander, Aki Vehtari, Samuel Kaski, Paul-Christian Bürkner, and Arto Klami. Prior knowledge elicitation: The past, present, and future. *arXiv preprint arXiv:2112.01380*, 2021.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, ICML 2014, page II–1791–II–1799. PMLR, 2014.

Andriy Mnih and Danilo J. Rezende. Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, ICML 2016, page 2188–2196. PMLR, 2016.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132): 1–62, 2020.

Eric Nalisnick and Padhraic Smyth. Learning priors for invariance. In *International Conference on Artificial Intelligence and Statistics*, AISTATS 2018, pages 366–375. PMLR, 2018.

Eric Nalisnick, Jonathan Gordon, and José Miguel Hernández-Lobato. Predictive complexity priors. In *International Conference on Artificial Intelligence and Statistics*, AISTATS 2021, pages 694–702. PMLR, 2021.

Anthony O'Hagan, Caitlin E. Buck, Alireza Daneshkhah, J. Richard Eiser, Paul H. Garthwaite, David J. Jenkinson, Jeremy E. Oakley, Tim Rakow(auth.), Stephen Senn, Marian Scott, and Peter Bloomfield(eds.). *Uncertain Judgements: Eliciting Experts' Probabilities.* John Wiley & Sons, 2006.

Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

Yudi Pawitan. *In All Likelihood: Statistical Modelling and Inference Using Likelihood.* Oxford University Press, USA, 1 edition, 2001. ISBN 0198507658,9780198507659.

Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. Overlapping community detection using Bayesian non-negative matrix factorization. *Phys. Rev. E*, 83:066114, Jun 2011.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, NeurIPS 2007, page 1257–1264, 2007.

Daniel J. Schad, Michael Betancourt, and Shravan Vasishth. Toward a principled Bayesian workflow in cognitive science. *arXiv preprint arXiv:1904.12765*, 2019.

Umut Simsekli, Ali Taylan Cemgil, and Yusuf Kenan Yilmaz. Learning the beta-divergence in tweedie compound poisson matrix factorization models. In *International Conference on Machine Learning*, ICML 2013, pages 1409–1417. PMLR, 2013.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, NeurIPS 2012, pages 2960–2968, 2012.

Vincent Y. F. Tan and Cedric Fevotte. Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1592–1605, 2013.

Da Tang and Rajesh Ranganath. The variational predictive natural gradient. In *International Conference on Machine Learning*, ICML 2019, pages 6145–6154. PMLR, 2019.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, ICML 2018, pages 1214–1223. PMLR, 2018.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, NeurIPS 2017, pages 2627–2636, 2017.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, 2017.

Wei Wang and Matthew Stephens. Empirical bayes matrix factorization. *Journal of Machine Learning Research*, 22(120):1–40, 2021.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2003, page 267–273, 2003.

Sinan Yıldırım, M. Burak Kurutmaz, Melih Barsbey, Umut Şimşekli, and A. Taylan Cemgil. Bayesian allocation model: Marginal likelihood-based model selection for count tensors. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):560–573, 2021.