Content-Based Social Recommendation with Poisson Matrix Factorization

Eliezer de Souza da Silva[1], Helge Langseth[1], and Heri Ramampiaro[1]

Norwegian University of Science and Technology (NTNU)
Department of Computer Science
NO-7491 Trondheim, Norway
{eliezer.souza.silva, helge.langseth, heri}@ntnu.no

Abstract. We introduce Poisson Matrix Factorization with Content and Social trust information (PoissonMF-CS), a latent variable probabilistic model for recommender systems with the objective of jointly modeling social trust, item content and user's preference using Poisson matrix factorization framework. This probabilistic model is equivalent to collectively factorizing a non-negative user–item interaction matrix and a non-negative item–content matrix. The user–item matrix consists of sparse implicit (or explicit) interactions counts between user and item, and the item–content matrix consists of words or tags counts per item. The model imposes additional constraints given by the social ties between users, and the homophily effect on social networks – the tendency of people with similar preferences to be socially connected. Using this model we can account for and fine-tune the weight of content-based and social-based factors in the user preference. We develop approximate *variational* inference algorithm and perform experiments comparing PoissonMF-CS with competing models. The experimental evaluation indicates that PoissonMF-CS achieves superior predictive performance on held-out data for the top-M recommendations task. Also, we observe that PoissonMF-CS generates compact latent representations when compared with alternative models while maintaining superior predictive performance.

Keywords: Probabilistic Matrix Factorization, Non-negative Matrix Factorization, Hybrid Recommender Systems, Poisson matrix factorization

1 Introduction

Recommender systems have proven to be a valuable component in many applications of personalization and Internet economy. Traditional recommender systems try to estimate a score function mapping each pair of user and item to a scalar value using the information of previous items already rated or interacted by the user [1]. Recent methods have been successful in integrating side information as content of the item, user context, social network, item topics, etc. For this purpose a variety of features should be taken into consideration, such as the routine, the geolocation, spatial correlation of certain preferences, mood and sentiment

analysis, as well as social relationships such as "friendship" to others users or "belonging" to a community in a social network [2]. In particular, a rich area of research has explored the integration of topic models and collaborative filtering approaches using principled probabilistic models [3–5]. Another group of models has been developed to integrate social network information into recommender systems using user–item ratings with extra dependencies [6] or constraining and regularizing directly the user latent factors with social features [7, 8]. Finally, some models have focused on the collective learning of both social features and content features, constructing hybrid recommender systems [5, 9, 10].

Our contribution is situated within all these three groups of efforts: we propose a probabilistic model that generalizes both previous models by jointly modeling content and social factors in the preference model applying Poisson-Gamma latent variable models to model the non-negativeness of the user–item ratings and induce sparse non-negative latent representation. Using this joint model we can generate recommendations based on the estimated score of non-observed items. In this article, we formulate the problem (Section 1.1), describe the proposed model (Section 3), present the variational inference algorithm (Section 4) and discuss the empirical results (Section 5). Our results indicate improved performance when compared to state-of-the-art methods including Correlated Topic Regression with Social Matrix Factorization (CTR-SMF) [5].

1.1 Problem formulation

Consider that given a set of observations of user–item interactions $R_{\text{train}} = \{(u, d, R_{ud})\}$, with $|R_{\text{train}}| = N_{\text{obs}} \ll U \times D$ (U is the number of users and D the number of documents), using additional item content information and user social network, we aim to learn a function f that estimates the value of each user–item interactions for all pairs of user and items $R_{\text{complete}} = \{(u, d, f(u, d))\}$. In general to solve this problem we assume that users have a set of preferences, and (using matrix factorization) we model these preferences using latent vectors.

Therefore, we have the documents (or items) set $\mathcal{D}$ of size $|\mathcal{D}| = D$, vocabulary set $\mathcal{V}$ of size $|\mathcal{V}| = V$, users set $\mathcal{U}$ of size $|\mathcal{U}| = U$, the social network given by the set of neighbors for each user $\{N(u)\}_{u \in \mathcal{U}}$. So, given the partially observed user–item matrix with integer ratings or implicit counts $\boldsymbol{R} = (R_{ud}) \in \mathbb{N}^{U \times D}$, the observed document–word count matrix $\boldsymbol{W} = (W_{dv}) \in \mathbb{N}^{D \times V}$, and the user social network $\{N(u)\}_{u \in \mathcal{U}}$, we need to estimate a matrix $\widetilde{\boldsymbol{R}} \in \mathbb{N}^{U \times D}$ to complete the user–item matrix $\boldsymbol{R}$. Finally, with the estimated matrix we can rank the unseen items for each user and make recommendations.

2 Related work

Collaborative Topic Regression (CTR): CTR [3] is a probabilistic model combining topic modeling (using Latent Dirichlet Allocation) and probabilistic matrix factorization (using Gaussian likelihood). Collaborative Topic Regression

with Social Matrix Factorization (CTR-SMF) [5] builds upon CTR adding social matrix factorization, creating a joint model Gaussian factorization model with content and social side information. Limited Attention Collaborative Topic Regression (LA-CTR) [9], is another approach with which the authors propose a joint model based on CTR integrating behavioral mechanism of attention. In this case, the amount of attention the user has invested in the social network is limited, and there is a measure of influence implying that the user may favor some friends more than others. In [10], the authors propose a CTR model seamlessly integrated item–tags, item content and social network information. All the models mentioned above combine in some degree LDA with Gaussian based matrix factorization for recommendations. Thus the time complexity for training those models is dominated by LDA complexity, making them difficult to scale. Also, the combination of LDA and Gaussian matrix factorization in CTR is a non-conjugate model that is hard to fit and difficult to work with sparse data.

Poisson Factorization: the basic Poisson factorization is a probabilistic model for non-negative matrix factorization based on the assumption that each user–item interaction R_{ui} can be modelled as a inner product of a user K dimensional latent vector $\boldsymbol{U}_u$ and item latent vector $\boldsymbol{V}_i$ representing the unobserved user preferences and item attributes [11], so that $R_{ui} \sim \text{Poisson}(\boldsymbol{U}_{\boldsymbol{u}}^T \boldsymbol{V}_{\boldsymbol{i}})$. Poisson factorization models for recommender systems have the advantage of principled modeling of implicit feedback, generating sparse latent representations, fast approximate inference with sparse matrix (the likelihood depends only on the consumed items) and improved empirical results compared with the Gaussian-based models [12, 11]. Nonparametric Poisson factorization model (BNPPF) [12] extends basic Poisson factorization by drawing user weights from a *Gamma process*. The latent dimensionality in this model is estimated from the data, effectively avoiding the *ad hoc* process of choosing the latent space dimensionality K. Social Poisson factorization (SPF) [6] extends basic Poisson factorization to accommodate preference and social based recommendations, adding a degree of trust variable and making all user–item interaction conditionally dependent on the user friends. With collaborative topic Poisson factorization (CTPF) [4], shared latent factors are utilized to fuse recommendation with topic model using Poisson likelihood and Gamma variables for both.

Non-negative matrix and tensor factorization using Poisson models: Poisson models are also successfully utilized in more general models such as tensor factorization and relational learning, particularly where it can use count data and non-negative factors. In [13], the authors propose a generic Bayesian non-negative tensor factorization model for count data and binary data. In [14], the authors explore the idea of adding constraints between the model variables using side information with hierarchical information, while the approach in [15] uses graph side information jointly modeled with topic modeling with Gamma process – a joint non-parametric model of network and documents.

3 Poisson Matrix Factorization with Content and Social trust information (PoissonMF-CS)

The proposed model PoissonMF-CS (see Figure 1) is an extension and generalization of previous Poisson models, combining social factorization model (social Poisson factorization – SPF) [6], and topic based factorization (collaborative topic Poisson factorization – CTPF) [4].

The main idea is to employ shared latent Gamma factors for topical preference and trust weight variables in the user social network, combining all factors in the rate of a Poisson likelihood of the user–item interaction. We model both sources of information having an additive effect on the observed user–item interactions and add two global multiplicative weights for each group of latent factors. The intuition behind the additive effect of social trust is that users tend to interact with items presented by their peers, so we can imagine a mechanism of "peer pressure" operating, where items offered through the social network have a positive (or neutral) influence on the user. In other words, we believe there is a positive social bias more than an anti-social bias, and we factor this in PoissonMF-CS model.

In the case of Poisson models, this non-negative constraint results in sparseness in the latent factors and can help avoid over-fitting (in comparison the Gaussian-based models[11, 12]). Gamma priors on the latent factors, and the fact that the latent factors can only have a positive or a zero effect on the final prediction, induce sparse latent representations in the model. Hence, in the inference process we adjust a factor that decreases the model likelihood by making its value closer to zero.

3.1 Generative model

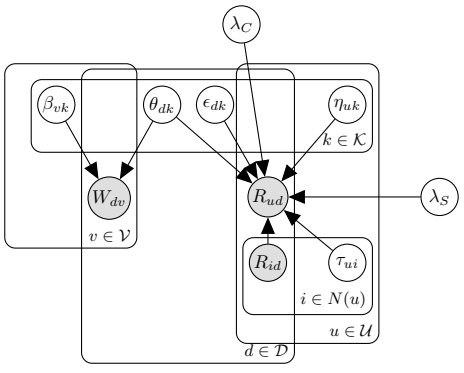


Fig. 1. Plate diagram for PoissonMF-CS model

In this model, W_{dv} is a counting variable for the number of times word v appears in document d, $\boldsymbol{\beta}_v$ is a latent vector capturing topic distribution of word v and $\boldsymbol{\theta}_d$ is the document–topic intensity vector, both with dimensionality K. Count variable W_{dv} is parametrized by the linear combination of these two latent factors $\boldsymbol{\theta}_{\boldsymbol{d}}^T \boldsymbol{\beta}_{\boldsymbol{v}}$. The document–topic latent factor $\boldsymbol{\theta}_d$ influences also the user–document rating variable R_{ud}. Each user has a latent vector $\boldsymbol{\eta}_u$ representing the user–topic propensity, which interacts with the document topic intensity factor $\boldsymbol{\theta}_d$ and document topic offset factor $\boldsymbol{\epsilon}_d$, resulting in the term $\boldsymbol{\eta}_u^T \boldsymbol{\theta}_d + \boldsymbol{\eta}_u^T \boldsymbol{\epsilon}_d$. Here, $\boldsymbol{\eta}_u^T \boldsymbol{\epsilon}_d$ captures the baseline matrix factorization, while $\boldsymbol{\eta}_u^T \boldsymbol{\theta}_d$ connects the rating variable with the content-based part of the model (word–document variable W_{dv}). The trust factor τ_{ui} between user u to user i is equal to zero for all users that are not connected in the social network ($\tau_{ui} > 0 \Leftrightarrow i \in N(u)$). This trust factor adds dependency between social connected users: the user–document rating R_{ud} is influenced by the average rating to item d given by friends of user u in the social network, weighted by the trust user u assigns to his friends ($\sum_{i \in N(u)} \tau_{ui} R_{id}$). We model this social dependency using a conditional specified model, as in [6]. The latent variables λ_C and λ_S are weight variables added in the model to capture and control the general weight of the content and social factors. These variables allow us to infer the importance of content and social factors according to the dataset or domain of usage. Also, instead of estimating these weights from the observed data, we may set λ_C and λ_S to constant values, thus controlling the importance of content and social parts of the model. Specifically if we set $\lambda_C = 0$ and $\lambda_S = 1$ we obtain the SPF model, while setting $\lambda_C = 1$ and $\lambda_S = 0$ result in CTPF, and $\lambda_C = 0$ and $\lambda_S = 0$ is equivalent to the simple Poisson matrix factorization without any side information [11].

Now we present the complete generative model assuming documents (or items) set $\mathcal{D}$ of size $|\mathcal{D}| = D$, vocabulary set $\mathcal{V}$ of size $|\mathcal{V}| = V$, users set $\mathcal{U}$ of size $|\mathcal{U}| = U$, the user social network given by the set of neighbors for each user $\{N(u)\}_{u \in \mathcal{U}}$ D documents, and K latent factors (topics) (with an index set $\mathcal{K}$).

1. Latent parameter distributions:
   (a) for all topics $k \in \mathcal{K}$:
      – for all words $v \in \mathcal{V}$: $\beta_{vk} \sim \text{Gamma}(a_\beta^0, b_\beta^0)$
      – for all documents $d \in \mathcal{D}$: $\theta_{dk} \sim \text{Gamma}(a_\theta^0, b_\theta^0)$ and $\epsilon_{dk} \sim \text{Gamma}(a_\epsilon^0, b_\epsilon^0)$
      – for all users $u \in \mathcal{U}$: $\eta_{uk} \sim \text{Gamma}(a_\eta^0, b_\eta^0)$
         • for all user $i \in N(u)$: $\tau_{ui} \sim \text{Gamma}(a_\tau^0, b_\tau^0)$
   (b) Content weight: $\lambda_C \sim \text{Gamma}(a_C^0, b_C^0)$
   (c) Social weight: $\lambda_S \sim \text{Gamma}(a_S^0, b_S^0)$
2. Observations probability distribution:
   (a) for all observed document–word pairs dv :

$$W_{dv} | \boldsymbol{\beta}_v, \boldsymbol{\theta}_d \sim \text{Poisson}(\boldsymbol{\beta}_{\boldsymbol{v}}^T \boldsymbol{\theta}_{\boldsymbol{d}})$$

   (b) for all observed user–document pairs ud :

$$R_{ud} | \boldsymbol{R}_{\boldsymbol{N(u),d}}, \boldsymbol{\eta_u}, \boldsymbol{\epsilon_d}, \boldsymbol{\theta_d} \sim \text{Poisson}(\lambda_C \boldsymbol{\eta}_u^T \boldsymbol{\theta}_d + \boldsymbol{\eta}_u^T \boldsymbol{\epsilon}_d + \lambda_S \sum_{i \in N(u)} \tau_{ui} R_{id})$$

4 Inference

First, we add a set of auxiliary latent Poisson variables to facilitate the posterior inference of the model. By doing so, the extended model will be complete conjugate, and consequently have analytical equations for the complete conditionals and variational updates [16]. In Appendix A we show that those auxiliary variables can be seen as by-product of a lower bound on the expected value of the log sum of the latent random variables. Variable $Y_{dv,k}$ represent a topic specific latent count for a word–document pair, so that the observed word–document counts is a sum of the latent counts (a property of the Poisson distribution) [1]. We can perform a similar modification for the user–item counts, splitting the latent terms of R_{ud} rate into two groups of topic specific latent count allocation variables: $Z^M_{ud,k}$ for the item content part, $Z^N_{ud,k}$ for the collaborative filtering part and $Z^S_{ud,i}$ for the social trust part (for this part, the intuitive explanation for the latent dimension is the idea of friend specific allocation of trust). The sum of all those latent counts is the observed user–item interaction count variable R_{ud}.

$$\begin{aligned} Y_{dv,k}|\beta_{vk},\theta_{dk} &\sim \text{Poisson}(\beta_{vk}\theta_{dk}) \\ Z^M_{ud,k}|\lambda_C,\eta_{uk},\theta_{dk} &\sim \text{Poisson}(\lambda_C\eta_{uk}\theta_{dk}) \\ Z^N_{ud,k}|\eta_{uk},\epsilon_{dk} &\sim \text{Poisson}(\eta_{uk}\epsilon_{dk}) \\ Z^S_{ud,i}|\lambda_S,\tau_{ui},R_{id} &\sim \text{Poisson}(\lambda_S\tau_{ui}R_{id}) \end{aligned} \tag{1}$$

$$\text{with } \sum_k Y_{dv,k} = W_{dv}, \text{ and } \sum_k Z^M_{ud,k} + Z^N_{ud,k} + \sum_{i\in N(U)} Z^S_{ud,i} = R_{ud}.$$

The inference problem consists on the estimation of the posterior distribution of the latent variables given the observed rating $\boldsymbol{R}$, the observed document–word counts $\boldsymbol{W}$, and the user social network $\{N(u)\}_{u\in\mathcal{U}}$, in other words, computing

$$p(\Theta|\boldsymbol{R},\boldsymbol{W},\{N(u)\}_{u\in\mathcal{U}}),$$

where $\Theta = \{\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\epsilon},\boldsymbol{\tau},\boldsymbol{y},\boldsymbol{z},\lambda_C,\lambda_S\}$ is the set of all latent variables. The exact computation of this posterior probability is intractable for any practical scenario, so we need approximation techniques for efficient parameter learning. In our case, we apply variational techniques to derive the learning algorithm. As an intermediate step towards the variational inference algorithm, we also derive the full conditional distribution for each latent variable. The full conditional distribution of each latent variable is also useful as update equations for Gibbs sampling, meaning that we could use the resulting equations to implement a sampling-based approximation. However, sampling methods are hard to scale and usually requires more memory, so as a design choice for the implementation of the learning algorithm, we refrained from applying the Gibbs sampling method and focus on the variational inference.

[1] The change consist in assigning a new Poisson variable to each sum-term in the latent rate of the Poisson likelihood, so if $S \sim \text{Poisson}(\sum_i X_i)$, we add variables $S_i \sim \text{Poisson}(X_i)$, and by the sum property of Poisson random variable $S = \sum_i S_i \sim \text{Poisson}(\sum_i X_i)$

In the next sections, we present the full conditional distribution of each of the latent variables in Section 4.1, and show the resulting update equation for the variational parameters in Section 4.2.

4.1 Full conditional distribution

The full conditional distribution of each of the latent variables is the distribution of a variable given all the other variables in the model, except the variable that we are considering. Given a set of indexed random variables X_k, we use the notation $p(X_k|X_{-k})$ (where X_{-k} means all the variables X_i such that $i \neq k$) to represent the full conditional distribution. Given the factorized structure of the model we can simplify the conditional set to the Markov blanket of the node we are considering (children nodes and co-parents nodes)[2] [16]. For conciseness, we show the derivations only for one Gamma latent variables and one Poisson latent count variable.

– **Gamma distributed variables:** We demonstrate how to obtain the full conditional distribution for Gamma distributed variable θ_{dk}, for the remaining Gamma distributed variables we only present the end result without the intermediate steps.

$$\begin{aligned} p(\theta_{dk}|*) &= p(\theta_{dk}|\text{MarkovBlanket}(\theta_{dk})) \\ &\propto p(\theta_{dk}) \prod_{v=1}^{V} p(Y_{dv,k}|\beta_{vk}, \theta_{dk}) \prod_{u=1}^{U} p(Z^M_{ud,k}|\lambda_C, \eta_{uk}, \theta_{dk}) \\ &\propto \theta_{dk}^{a^0_\theta - 1} e^{-b^0_\theta \theta_{dk}} \prod_v \theta_{dk}^{Y_{dv,k}} e^{-\beta_{vk}\theta_{dk}} \prod_u \theta_{dk}^{Z^M_{ud,k}} e^{-\lambda_C \eta_{uk} \theta_{dk}} \\ &\propto \theta_{dk}^{a^0_\theta + \sum_v Y_{dv,k} + \sum_u Z^M_{ud,k} - 1} e^{-\theta_{dk}(b^0_\theta + \sum_v \beta_{vk} + \lambda_C \sum_u \eta_{uk})} \end{aligned} \tag{2}$$

Normalizing equation Eq. 2 over θ_{dk} we obtain the pdf of a Gamma variable with shape $a^0_\theta + \sum_v Y_{dv,k} + \sum_u Z^M_{ud,k}$ and rate $b^0_\theta + \sum_v \beta_{vk} + \lambda_C \sum_u \eta_{uk}$. The final solution is written in Eq. 3. Also, notice that because of the way the model is structured all other Gamma latent variable have similar equations, the difference being the set of variables in the Markov blanket.

$$\begin{aligned} \theta_{dk}|* &\sim \text{Gamma}(a^0_\theta + \textstyle\sum_v Y_{dv,k} + \sum_u Z^M_{ud,k}, b^0_\theta + \sum_v \beta_{vk} + \lambda_C \sum_u \eta_{uk}) \\ \beta_{vk}|* &\sim \text{Gamma}(a^0_\beta + \textstyle\sum_d Y_{dv,k}, b^0_\beta + \sum_d \theta_{dk}) \\ \eta_{uk}|* &\sim \text{Gamma}(a^0_\eta + \textstyle\sum_d Z^M_{ud,k} + Z^N_{ud,k}, b^0_\eta + \lambda_C \sum_d \theta_{dk} + \sum_d \epsilon_{dk}) \\ \epsilon_{dk}|* &\sim \text{Gamma}(a^0_\epsilon + \textstyle\sum_u -Z^N_{ud,k}, b^0_\epsilon + \sum_u \eta_{uk}) \\ \tau_{ui}|* &\sim \text{Gamma}(a^0_\tau + \textstyle\sum_d Z^S_{ud,i}, b^0_\tau + \lambda_S \sum_d R_{id}) \\ \lambda_C|* &\sim \text{Gamma}(a_C + \textstyle\sum_{u,d,k} Z^M_{ud,k}, b_C + \sum_{u,d,k} \eta_{uk}\theta_{dk}) \\ \lambda_S|* &\sim \text{Gamma}(a_S + \textstyle\sum_{u,d,i} Z^S_{ud,i}, b_S + \sum_{u,d,i} \tau_{ui} R_{id}) \end{aligned} \tag{3}$$

– **Multinomial distributed (auxiliary) variables:** looking at the Markov blanket of $\boldsymbol{Y}_{dv}$ we obtain:

$$\begin{aligned} p(\boldsymbol{Y}_{dv}|*) &\propto \textstyle\prod_{k=1}^{K} p(Y_{dv,k}|\beta_{vk}, \theta_{dk}) = \prod_{k=1}^{K} \text{Poisson}(Y_{dv,k}|\beta_{vk}\theta_{dk}) \\ &\propto \textstyle\prod_{k=1}^{K} \frac{(\beta_{vk}\theta_{dk})^{Y_{dv,k}}}{Y_{dv,k}!} \end{aligned} \tag{4}$$

[2] We use the notation MarkovBlanket(X) to denote the Markov blanket of a variable X – the set of children and co-parents nodes of variable X in the graphical model

Given that we know that $\sum_k Y_{dv,k} = W_{dv}$, this functional form is equivalent to the pdf of a Multinomial distribution with parameter probabilities proportional to $\beta_{vk}\theta_{dk}$.

$$\boldsymbol{Y}_{dv}|* \sim \text{Mult}(W_{dv}; \boldsymbol{\phi_{dv}}) \qquad \text{with } \phi_{dv,k} = \frac{\beta_{vk}\theta_{dk}}{\sum_k \beta_{vk}\theta_{dk}} \tag{5}$$

Similarly, $\boldsymbol{Z}_{ud}$ is a Multinomial with parameters proportional to the parent nodes of $\boldsymbol{Z}_{ud}$. For convenience in the previous section, we split $\boldsymbol{Z}_{ud}$ in three blocks of variables and parameters $\boldsymbol{Z}_{ud} = [\boldsymbol{Z}^M_{ud}, \boldsymbol{Z}^N_{ud}, \boldsymbol{Z}^S_{ud}]$ representing the different high-level parts of our model. The dimensionality of the first two blocks is the K, while for the last block is U, resulting that $\boldsymbol{Z}_{ud}$ has dimensionality $2K + U$. Similarly the parameters of the $\boldsymbol{Z}_{ud}$ full conditional Multinomial have a block structure $\boldsymbol{\xi}_{ud} = [\boldsymbol{\xi}^M_{ud}, \boldsymbol{\xi}^N_{ud}, \boldsymbol{\xi}^S_{ud}]$.

$$Z_{ud}|* \sim \text{Mult}(R_{ud}; \boldsymbol{\xi_{ud}})$$

$$\text{with } \xi_{ud,k} = \begin{cases} \xi^M_{ud,k} = \frac{\lambda_C \eta_{uk}\theta_{dk}}{\sum_k \eta_{uk}(\lambda_C\theta_{dk}+\epsilon_{dk})+\lambda_S \sum_{i\in N(u)} \tau_{ui} R_{id}} \\ \xi^N_{ud,k} = \frac{\eta_{uk}\epsilon_{dk}}{\sum_k \eta_{uk}(\lambda_C\theta_{dk}+\epsilon_{dk})+\lambda_S \sum_{i\in N(u)} \tau_{ui} R_{id}} \\ \xi^S_{ud,i} = \frac{\lambda_S \tau_{ui} R_{id}}{\sum_k \eta_{uk}(\lambda_C\theta_{dk}+\epsilon_{dk})+\lambda_S \sum_{i\in N(u)} \tau_{ui} R_{id}} \end{cases}$$

We present in next section how to use these equations to derive a deterministic optimization algorithm for approximate inference using the *variational* method.

4.2 Variational inference

Given a family of surrogate distributions $q(\Theta|\Psi)$ for the unobserved variables (latent terms) parametrized by variational parameters Ψ, we want to find an assignment of the variational parameters that minimize the KL-divergence between $q(\Theta|\Psi)$ and $p(\Theta|\boldsymbol{R}, \boldsymbol{W})$ [3],

$$\underset{\Psi}{\text{argmin}}\, \text{KL}\{q(\Theta|\Psi), p(\Theta|\boldsymbol{R}, \boldsymbol{W})\}.$$

Then, the optimal surrogate distribution can be used as an approximation the true posterior. However, the optimization problem using directly the KL divergence is not tractable, since it depends on the computation of the evidence $\log p(\boldsymbol{R}, \boldsymbol{W})$. This can be accomplished using Jensen inequality to get lower bounds on the evidence and changing the optimization objective to this lower bound – the Evidence Lower BOund (ELBO):

$$\underset{\Psi}{\text{argmin}}\, L(\Psi) = \text{E}_q[\log p(\boldsymbol{R}, \boldsymbol{W}, \Theta) - \log q(\Theta|\Psi)]$$

[3] To simplify the notation, we use the short-handed $p(\Theta|\boldsymbol{R}, \boldsymbol{W})$ to denote the posterior distribution $p(\Theta|\boldsymbol{R}, \boldsymbol{W}, \{N(u)\}_{u\in\mathcal{U}})$. Also, we drop the explicitly notation indicating the dependency on the social network

Another ingredient in this approximation is the mean field assumption. It consists in assuming that all variables in the variational distribution $q(\Theta|\Psi)$ are mutually independent. As a result the variational surrogate distribution can be expressed as a factorized distribution of each latent factor (Eq. 7). Another implication is that we can compute the updates for each variational X_i factor using the complete conditional of the latent factor [16]. Finally, the inference algorithm consists in iterative updating of variational parameters of each factorized distribution until convergence is reached, resulting in the *coordinate ascent variational inference* algorithm based on the following equation:

$$q(X_i) \propto \exp\{\mathrm{E}_q[\log p(X_i|*)]\} \tag{6}$$

Using Eq. 6, we can take each complete conditional variable that we described in the previous section and create a respective proposal distribution for the variational inference. This proposal distribution is in the same family as the full conditional distribution of the latent variables, meaning that we have a group of Gamma and Multinomial variables. As long as we update the parameters of the variational distribution using Eq. 6, it is guaranteed to minimize the KL divergence between the surrogate variational distribution (Eq. 7) over the latent variables and the posterior distribution of the model.

$$\begin{aligned} q(\Theta|\Psi) = {} & q(\lambda_C|a_{\lambda_C}, b_{\lambda_C}) q(\lambda_S|a_{\lambda_S}, b_{\lambda_S}) \textstyle\prod_{u,k,i} q(\tau_{ui}|a_{\tau_{ui}}, b_{\tau_{ui}}) q(\eta_{uk}|a_{\eta_{uk}}, b_{\eta_{uk}}) \\ & \times \textstyle\prod_{d,v,k} q(\epsilon_{dk}|a_{\epsilon_{dk}}, b_{\epsilon_{dk}}) q(\theta_{dk}|a_{\theta_{dk}}, b_{\theta_{dk}}) q(\beta_{vk}|a_{\beta_{vk}}, b_{\beta_{vk}}) \\ & \times \textstyle\prod_{d,v,u} q(\boldsymbol{Z}_{dv}|\boldsymbol{\phi}^*_{dv}) q(\boldsymbol{Y}_{ud}|\boldsymbol{\xi}^{M*}_{ud}, \boldsymbol{\xi}^{N*}_{ud}, \boldsymbol{\xi}^{S*}_{ud}) \end{aligned} \tag{7}$$

After applying Eq. 6 together with the expected value properties for each latent variable[4], we obtain the following update equations for the variational parameters.

– **Content and social weights**:

$$\begin{aligned} a_{\lambda_C} &= a_C + \textstyle\sum_{u,d,k} R_{ud} \xi^{M*}_{ud,k}, & b_{\lambda_C} &= b_C + \textstyle\sum_{u,d,k} \frac{a_{\eta_{uk}}}{b_{\eta_{uk}}} \frac{a_{\theta_{dk}}}{b_{\theta_{dk}}} \\ a_{\lambda_S} &= a_S + \textstyle\sum_u R_{ud} \xi^{M*}_{ud,k} + \sum_v W_{dv} \phi^*_{dv,k}, & b_{\lambda_S} &= b_S + \textstyle\sum_{u,d,i} R_{id} \frac{a_{\tau_{ui}}}{b_{\tau_{ui}}} \end{aligned}$$

– **Content v (topic/tags/etc) parameters**:

$$a_{\beta_{vk}} = a^0_\beta + \textstyle\sum_d W_{dv} \phi^*_{dv,k}, \; b_{\beta_{vk}} = b^0_\beta + \sum_d \frac{a_{\theta_{dk}}}{b_{\theta_{dk}}}$$

– **Item d parameters**:

$$\begin{aligned} a_{\epsilon_{dk}} &= a^0_\epsilon + \textstyle\sum_u R_{ud} \xi^{N*}_{ud,k}, & b_{\epsilon_{dk}} &= b^0_\epsilon + \textstyle\sum_u \frac{a_{\eta_{uk}}}{b_{\eta_{uk}}} \\ a_{\theta_{dk}} &= a^0_\theta + \textstyle\sum_u R_{ud} \xi^{M*}_{ud,k} + \sum_v W_{dv} \phi^*_{dv,k}, & b_{\theta_{dk}} &= b^0_\theta + \mathrm{E}_q[\lambda_C] \textstyle\sum_u \frac{a_{\eta_{uk}}}{b_{\eta_{uk}}} + \sum_v \frac{a_{\beta_{vk}}}{b_{\beta_{vk}}} \end{aligned}$$

– **User u parameters**:

[4] Notice that, if $q(X) = \mathrm{Gamma}(X|a_X, b_X)$ (parameterized by shape and rate) , then $\mathrm{E}_q[X] = \frac{a_X}{b_X}$ and $\mathrm{E}_q[\log X] = \Psi(a_X) - \log(b_X)$, where $\Psi(.)$ is the Digamma function. If $q(\boldsymbol{X}) = \mathrm{Mult}(R|\boldsymbol{p})$, then $\mathrm{E}_q[X_i] = Rp_i$.

$$a_{\eta_{uk}} = a^0_\eta + \sum_d R_{ud}(\xi^{M*}_{ud,k} + \xi^{N*}_{ud,k}),\; b_{\eta_{uk}} = b^0_\eta + \sum_d \mathrm{E}_q[\lambda_C]\frac{a_{\theta_{dk}}}{b_{\theta_{dk}}} + \frac{a_{\epsilon_{dk}}}{b_{\epsilon_{dk}}}$$
$$a_{\tau_{ui}} = a^0_\tau + \sum_d R_{ud}\xi^{S*}_{ud,i}, \qquad b_{\tau_{ui}} = b^0_\tau + \mathrm{E}_q[\lambda_S]\sum_d R_{id}$$

– **item–content** dv **parameters**:

$$\phi^*_{dv,k} \propto \frac{e^{\Psi\left(a_{\beta_{vk}}\right)}}{b_{\beta_{vk}}}\frac{e^{\Psi\left(a_{\theta_{dk}}\right)}}{b_{\theta_{dk}}} \text{ with } \sum_k \phi_{dv,k} = 1$$

– **user–item** ud **parameters**:

$$\xi^{M*}_{ud,k} \propto e^{E_q[\log\lambda_C]}\frac{e^{\Psi\left(a_{\eta_{uk}}\right)}}{b_{\eta_{uk}}}\frac{e^{\Psi\left(a_{\theta_{dk}}\right)}}{b_{\theta_{dk}}} \quad \xi^{N*}_{ud,k} \propto \frac{e^{\Psi\left(a_{\eta_{uk}}\right)}}{b_{\eta_{uk}}}\frac{e^{\Psi\left(a_{\epsilon_{dk}}\right)}}{b_{\epsilon_{dk}}}$$
$$\xi^{S*}_{ud,i} \propto e^{E_q[\log\lambda_S]}\frac{e^{\Psi(a_{\tau_{ui}})}}{b_{\tau_{ui}}}R_{id} \qquad \text{with } \sum_k \xi^{M*}_{ud,k} + \xi^{N*}_{ud,k} + \sum_i \xi^{S*}_{ud,i} = 1$$

Computing the ELBO: The variational updates calculated in the previous sections are guaranteed to non-decrease the ELBO. However, we still need to calculate this lower bound after each iteration to evaluate a stopping condition for the optimization algorithm. We briefly describe a particular lower-bounding for the ELBO involving the log-sum present in the Poisson rate.

Note also that the surrogate distribution is factorized using the mean field assumptions (Eq. 7), so we have a sum of terms corresponding to the expected log probability over the surrogate distribution. The terms comprising the log-probabilities of the Poisson likelihood display a expected value over a sum of logarithms of latent variables (for example $\mathrm{E}_q[\log(\sum_k \beta_{vk}\theta_{dk})]$), this is a challenging computation, but we can apply another lower-bound[5] and simplify it to Eq. 8.

$$\begin{aligned}\mathrm{E}_q[\log(\textstyle\sum_k \beta_{vk}\theta_{dk})] \geq \textstyle\sum_k \phi^*_{dv,k}\left(\mathrm{E}_q[\log\beta_{vk}] + \mathrm{E}_q[\log\theta_{dk}]\right]) \\ - \textstyle\sum_k \phi^*_{dv,k}\log\phi^*_{dv,k}\end{aligned} \tag{8}$$

This same simplification can be done to all Poisson terms independently because of the mean field assumptions. It is equivalent to using the auxiliary latent counts. So, for example, using the latent variable $Z_{dv,k}$, β_{vk} and θ_{dk}, the Poisson term in the ELBO results in Eq. 9.

$$\begin{aligned}\mathrm{E}_q\left[\log\tfrac{p(Z_{dv})}{q(Z_{dv})}\right] = \textstyle\sum_k W_{dv}\phi^*_{dv,k}\,\mathrm{E}_q[\log(\beta_{vk}\theta_{dk})] \\ - \mathrm{E}_q[\beta_{vk}\theta_{dk}] - W_{dv}\phi^*_{dv,k}\log(\phi^*_{dv,k}) - \log(W_{dv}!)\end{aligned} \tag{9}$$

For the Gamma terms, the calculations are a direct application of ELBO formula for the appropriate variable. For example, Eq. 10 describes the resulting terms for β_{vk}.

$$\begin{aligned}\mathrm{E}_q\left[\log\tfrac{p(\beta_{vk})}{q(\beta_{vk})}\right] = \log\tfrac{\Gamma(a_{\beta_{vk}})}{\Gamma(a)} + a\log b + a_{\beta_{vk}}(1 - \log b_{\beta_{vk}}) \\ + (a - a_{\beta_{vk}})\,\mathrm{E}_q[\log\beta_{vk}] - b\,\mathrm{E}_q[\beta_{vk}]\end{aligned} \tag{10}$$

[5] this lower bound is valid for any $\phi^*_{dv,k}$, with $\sum_k \phi^*_{dv,k} = 1$, check Eq. 13 in Appendix A for details

Recommendations: Once we learn the latent factors of the model from the observations we can infer the user preference over the set of items using the expected value of the user–item rating $\mathrm{E}[R_{ud}]$. The recommendation algorithm ranks the unobserved items for each user according to $\mathrm{E}[R_{ud}]$ and recommend to top-M items. We utilize the variational distribution to efficiently compute $\mathrm{E}[R_{ud}]$ as defined in Eq. 11. This value can be broken down into three non-negative scores: $\mathrm{E}_q[\eta_u]^T \mathrm{E}_q[\epsilon_d]$, representing the "classic" collaborative filtering matching of users preferences and items features, $\mathrm{E}_q[\lambda_C] \mathrm{E}_q[\eta_u]^T \mathrm{E}_q[\theta_d]$ representing the content factors contribution and $\mathrm{E}_q[\lambda_S] \sum_{i \in N(u)} \mathrm{E}_q[\tau_{ui}] R_{id}$ the social influence contribution.

$$\mathrm{E}[R_{ud}] \approx \mathrm{E}_q[\eta_u]^T (\mathrm{E}_q[\lambda_C] \mathrm{E}_q[\theta_d] + \mathrm{E}_q[\epsilon_d]) + \mathrm{E}_q[\lambda_S] \sum_{i \in N(u)} \mathrm{E}_q[\tau_{ui}] R_{id} \qquad (11)$$

Complexity and convergence: the complexity of each iteration of the variational inference algorithm is linear on the number of latent factors K, non-zero ratings nR, non-zero word-document counts nW, users U, items D, vocabulary set W and neighbors for each user nS, in other words $O(K(nW + nR + nS + U + D + W))$. We have shown that we can obtain closed-form updates for the inference algorithm, which stems from the fact that the model is fully conjugate and in the exponential family of distributions. In this setting variational inference is guaranteed to converge, and we observed in the experiments the algorithm converging after 20 to 40 iterations.

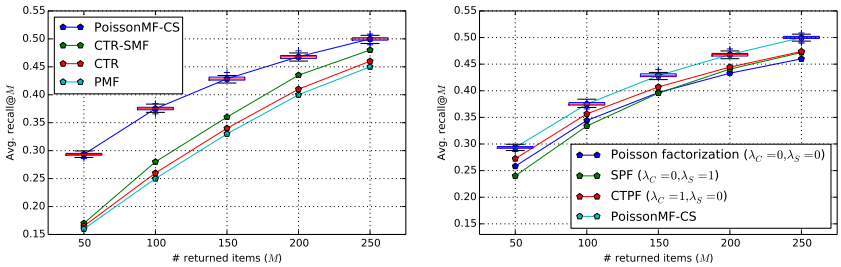
5 Evaluation

In this section, we analyze the predictive power of the proposed model with a real world dataset and compare it with state of the art methods.[6]

Datasets. to be able to compare with the state-of-art method Correlated Topic Regression with Social Matrix Factorization [5], we conducted experiments using the *hetrec2011-lastfm-2k* (Last.fm) dataset [17]. This dataset consists of a set of user–artists weighted interactions ("artists" is item set), a set of user–artists-tags and a set of user–user relations[7]. We process the dataset to create an artist–tags matrix by summing up all the tags given by all users to a given artist, this matrix is the item–content matrix in our model. Also, we discard the user–artists weight, considering a "1" for all observed cases. After the preprocessing, we sample 85% of the user–artists observation for training, and kept 15% held-out for predictive evaluation, selecting only users with more than 5 item ratings for the training part of the split.

[6] Our C++ implementation of PoissonMF-CS with some of the experiments will be available this repository `https://github.com/zehsilva/poissonmf_cs`

[7] The statistics for the dataset are: 1892 users, 17632 artists, 11946 tags, 25434 user–user connections, 92834 user–items interactions, and 186479 user–tag–items entries



(a) PoissonMF-CS (K=10) and Gaussian based models

(b) PoissonMF-CS (K=10) and other PF models

Fig. 2. Comparison of PoissonMF-CS with alternative models. Each subplot is the result of running the PoissonMF-CS recommendation algorithm over 30 random splits of the *Hetrec2011-lastfm* dataset for a fixed number of latent features K (in this case, $K = \{10, 15\}$). The values for CTR-SMF, CTR, and PMF was taken from [5], and according to the reported results, they are the best values found after a grid search.

Metric: Given the random splits of training and test, we train our model and use the estimated latent factors to predict the entries in the testing datasets. In this setting zero ratings can not be necessarily interpreted as negative, making it problematic to use the precision metric. Instead, we focus on recall metric to be comparable with previous work [5] and because the relevant items are available. Specifically, we calculate the recall at the top M items (recall@M) for a user, defined as:

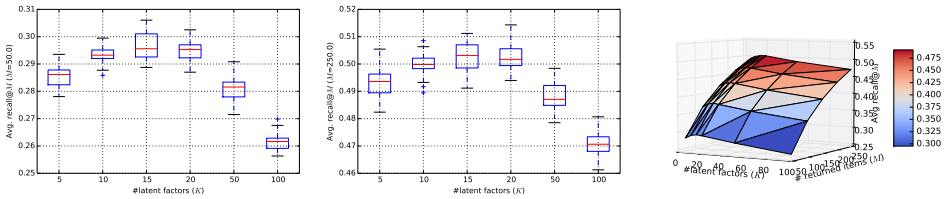
$$\text{recall@}M = \frac{\text{number of items the user likes in Top } M}{\text{total number of items the user likes}} \tag{12}$$

Recall@M from Eq. 12 is calculated for each user, to obtain a single measure for the whole dataset we average it over all the users obtaining the Avg. Recall@M.

5.1 Experiments

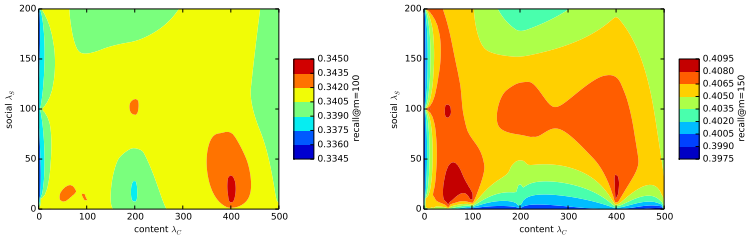
Initially we set all the Gamma hyperparameters to the same values a_{all}[8] and b_{all}[9] equal to 0.1, while varying the latent dimensionality K. For each value of K we ran the experiments on 30 multiple random splits of the dataset in order to be able to generate boxplots of the final recommendation recall. We compare our results with the reported results in [5] for the same dataset and with optimal parameters. In this first experiment we let the algorithm estimate the optimal content weight λ_C and social weight λ_S. It is possible to see in Fig 2 that PoissonMF-CS is consistently outperforming by large margin CTR-SMF and CTR (Fig. 2a), while outperforming other Poisson factorization methods

[8] $a_{all} = a^0_\beta = a^0_\eta = a^0_\theta = a^0_\epsilon = a^0_\tau = a_C = a_S = 0.1$
[9] $b_{all} = b^0_\beta = b^0_\eta = b^0_\theta = b^0_\epsilon = b^0_\tau = b_C = b_S = 0.1$



(a) M=50 (b) M=250 (c) 3D visualization

Fig. 3. Impact of the number of latent variables (K) parameter on the Av. Recall@M metric for different number of returned items (M). Each subplot is the result of running the PoissonMF-CS recommendation algorithm over 30 random splits of the dataset with K varying in (5,10,15,20,50,100)

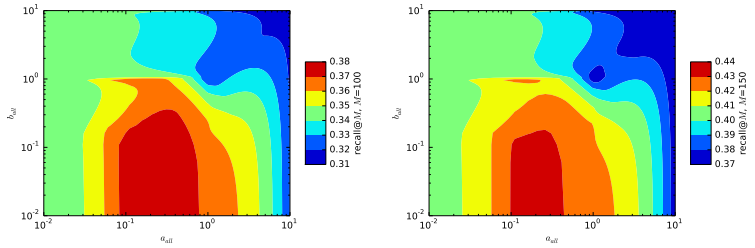


(a) M=100 (b) M=150

Fig. 4. Evaluation of the impact of content and social weight parameters (in all experiments in this figure $K = 10$)

(Fig. 2b) by a significant margin ($p \leq 1 \cdot 10^{-6}$ in Wilcoxon paired test for each M). . This may be indicative that both the choice of Poisson likelihood with non-negative latent factors and the modelling of content and social weights have positive impact in the predictive power of the model.

Model selection. Fig. 3 shows the resulting predictive performance of PoissonMF-CS with different values of number of latent factors K in *Hetrec2011-lastfm* dataset. We concluded that the optimal choice for K is 15. This result is important, indicating that the model is generating compact latent representations, given that the optimal choice of K reported for CTR-SMF in the same dataset is 200. In Fig. 5 we show the results for the latent variable hyperparameters. We ran one experiment varying the hyperparameters a_{all} and b_{all} to understand the impact of these hyperparameters in the final recommendation. We noticed that the optimal values for different values of M for both hyperparameters are between 0.1 and 1, a result consistent with the recommendations in the literature [12, 4, 6] and with the statistical intuition that Poisson likelihood with Gamma prior with shape parameter $a < 1$ favour sparse latent representation.



(a) M=100 (b) M=150

Fig. 5. Evaluation of the impact of latent Gamma hyperpriors on the recall (in all experiments in this figure $K = 10$)

The next experiment was to set the content weight and social weight to fixed values and evaluate the impact of these weights on the result. In Fig 4 we can see that the resulting pattern for different values of M is not evident, but indicates that the resulting recall is less sensitive to change in the content and social weights parameters than on the hyperparameters a_{all} and b_{all}. This is also indicative that the importance of social and content factors is not the same at different points of the ranked list of recommendations.

6 Conclusion

This article describes PoissonMF-CS, a joint Bayesian model for recommendations integrating three sources of information: item textual content, user social network, and user–item interactions. It generalizes existent Poisson factorization models for recommendations by adding both content and social features. Our experiment shows that the proposed model consistently outperforms previous Poisson models (SPF and CTPF) and alternative joint models based on Gaussian probabilistic factorization and LDA (CTR-SMF and CTR) on a dataset containing both content and social side information. These results demonstrate that joint modeling of social and content features using Poisson models improves the recommendations, can have scalable inference and generates more compact latent features. Although the batch variational inference algorithm is already efficient [10], one future improvement will be the design of Stochastic Variational Inference algorithm for very large scale inference.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. **17**(6) (June 2005) 734–749

[10] For example, it takes 12 minutes to train the best performing model in a desktop machine with the *Hetrec2011-lastfm* dataset in a single core without any parallelism

2. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Social Network Analysis and Mining **3**(4) (2013) 1113–1133
3. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011. (2011) 448–456
4. Gopalan, P., Charlin, L., Blei, D.M.: Content-based recommendations with poisson factorization. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. (2014) 3176–3184
5. Purushotham, S., Liu, Y.: Collaborative topic regression with social matrix factorization for recommendation systems. In: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012, icml.cc / Omnipress (2012)
6. Chaney, A.J., Blei, D.M., Eliassi-Rad, T.: A probabilistic model for using social networks in personalized item recommendation. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015. (2015) 43–50
7. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011. (2011) 287–296
8. Yuan, Q., Chen, L., Zhao, S.: Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation. In: Proceedings of the Fifth ACM Conference on Recommender Systems. RecSys '11, New York, NY, USA, ACM (2011) 245–252
9. Kang, J., Lerman, K.: LA-CTR: A limited attention collaborative topic regression for social media. In: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA. (2013)
10. Wang, H., Chen, B., Li, W.: Collaborative topic regression with social regularization for tag recommendation. In: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. (2013) 2719–2725
11. Gopalan, P., Hofman, J.M., Blei, D.M.: Scalable recommendation with hierarchical poisson factorization. In Meila, M., Heskes, T., eds.: Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands, AUAI Press (2015) 326–335
12. Gopalan, P., Ruiz, F.J.R., Ranganath, R., Blei, D.M.: Bayesian nonparametric poisson factorization for recommendation systems. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014. Volume 33 of JMLR Workshop and Conference Proceedings., JMLR.org (2014) 275–283
13. Hu, C., Rai, P., Chen, C., Harding, M., Carin, L.: Scalable bayesian non-negative tensor factorization for massive count data. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II. (2015) 53–70
14. Hu, C., Rai, P., Carin, L.: Non-negative matrix factorization for discrete data with hierarchical side-information. In: Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016. (2016) 1124–1132

15. Acharya, A., Teffer, D., Henderson, J., Tyler, M., Zhou, M., Ghosh, J.: Gamma process poisson factorization for joint modeling of network and documents. In: Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I. (2015) 283–299
16. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
17. Cantador, I., Brusilovsky, P., Kuflik, T.: 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In: Proceedings of the 5th ACM conference on Recommender systems. RecSys 2011, New York, NY, USA, ACM (2011)

A A lower bound for $\mathbf{E}_{\boldsymbol{q}}[\log \sum_{\boldsymbol{k}} \boldsymbol{X_k}]$

The function $\log(\cdot)$ is a concave, meaning that:

$$\begin{aligned}&\log(px_1 + (1-p)x_2) \geq p \log x_1 + (1-p)\log x_2\\ &\forall p : p \geq 0\end{aligned}$$

By induction this property can be generalized to any convex combination of x_k ($\sum_k p_k x_k$ with $\sum_k p_k = 1$ and $p_k \geq 0$): $\log \sum_k p_k x_k \geq \sum_k p_k \log x_k$ Now using random variables we can create a similar convex combination by multiplying and dividing each random variable X_k by $p_k > 0$ and apply the sum of of expectation property:

$$\begin{aligned}\mathrm{E}_q[\log \textstyle\sum_k X_k] &= \mathrm{E}_q[\textstyle\sum_k \log p_k \frac{X_k}{p_k}]\\ \log \textstyle\sum_k p_k \frac{X_k}{p_k} &\geq \textstyle\sum_k p_k \log \frac{X_k}{p_k}\\ \Rightarrow \mathrm{E}_q[\log \textstyle\sum_k p_k \frac{X_k}{p_k}] &\geq \textstyle\sum_k p_k \mathrm{E}_q[\log \frac{X_k}{p_k}]\\ \Rightarrow \mathrm{E}_q[\log \textstyle\sum_k X_k] &\geq \textstyle\sum_k p_k \mathrm{E}_q[\log X_k] - p_k \log p_k\end{aligned} \tag{13}$$

The lower bound of Eq. 13 is applied in Eq. 8 and it is a general lower bound useful for the log–sum terms in the ELBO computation. If we want a tight lower bound, we should use Lagrange multipliers to choose the set of p_k that maximize the lower-bound given that they sum to 1.

$$\begin{aligned}L(p_1, \ldots, p_K) &= (\textstyle\sum_k p_k \mathrm{E}_q[\log X_k] - p_k \log p_k) + \lambda\,(1 - \textstyle\sum_k p_k)\\ \tfrac{\partial L}{\partial p_k} &= \mathrm{E}_q[\log X_k] - \log p_k - 1 - \lambda = 0\\ \tfrac{\partial L}{\partial \lambda} &= 1 - \textstyle\sum_k p_k = 0\\ \Rightarrow \mathrm{E}_q[\log X_k] &= \log p_k + 1 + \lambda\\ \Rightarrow \exp \mathrm{E}_q[\log X_k] &= p_k \exp(1+\lambda)\\ \Rightarrow \textstyle\sum_k \exp \mathrm{E}_q[\log X_k] &= \underbrace{\textstyle\sum_k p_k}_{=1} \exp(1+\lambda)\\ \Rightarrow p_k &= \tfrac{\exp\{\mathrm{E}_q[\log X_k]\}}{\sum_k \exp\{\mathrm{E}_q[\log X_k]\}}\end{aligned} \tag{14}$$

The final formula for p_k in Eq. 14 is exactly the same that we can find for the parameters of the Multinomial distribution of the auxiliary variables in the Poisson model with sum of Gamma distributed latent variables, which demonstrates that the choice of distribution for the auxiliary variables is optimal for this lower-bound.