Exploiting Ratings, Reviews and Relationships for Item Recommendations in Topic Based Social Networks

Pengfei Li
Zhejiang University, China
pfl@zju.edu.cn

Hua Lu
Aalborg University, Denmark
luhua@cs.aau.dk

Gang Zheng
Zhejiang University, China
gangzheng@zju.edu.cn

Qian Zheng
Nanyang Technological University, Singapore
csqianzheng@gmail.com

Long Yang
Zhejiang University, China
yanglong@zju.edu.cn

Gang Pan
Zhejiang University, China
gpan@zju.edu.cn

ABSTRACT

Many e-commerce platforms today allow users to give their rating scores and reviews on items as well as to establish social relationships with other users. As a result, such platforms accumulate heterogeneous data including numeric scores, short textual reviews, and social relationships. However, many recommender systems only consider historical user feedbacks in modeling user preferences. More specifically, most existing recommendation approaches only use rating scores but ignore reviews and social relationships in the user-generated data. In this paper, we propose TSNPF—a latent factor model to effectively capture user preferences and item features. Employing Poisson factorization, TSNPF fully exploits the wealth of information in rating scores, review text and social relationships altogether. It extracts topics of items and users from the review text and makes use of similarities between user pairs with social relationships, which results in a comprehensive understanding of user preferences. Experimental results on real-world datasets demonstrate that our TSNPF approach is highly effective at recommending items to users.

KEYWORDS

Recommender System, Graphical Model, Variational Inference

ACM Reference Format:
Pengfei Li, Hua Lu, Gang Zheng, Qian Zheng, Long Yang, and Gang Pan. 2019. Exploiting Ratings, Reviews and Relationships for Item Recommendations in Topic Based Social Networks. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313473

1 INTRODUCTION

Recommender systems are the core of today's personalized online e-commerce platforms like Amazon and Taobao. Such systems make use of user activities (e.g., ratings and reviews) and/or social relationships to generate features of items and identify user preferences.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313473

The rating matrix based collaborative filtering [38] makes use of user activity to identify latent user-item factors to suggest products to a user. A representative realization of latent factor models is based on Matrix Factorization [27]. However, matrix factorization is confronted with a big challenge—computation complexity. When there are large numbers of users or items, the matrix factorization methods inevitably incur very high cost of latent factor identification from a huge user-item rating matrix. With the intensive mathematical computation required, such methods cannot handle large datasets. Some variants, e.g., Probabilistic Matrix Factorization [37], attempt to solve this problem. However, like most collaborative filtering methods, such variants often fail to identify the latent factors when a user or an item is associated with too few ratings. Consequently, the computation for recommendation falls apart due to data sparsity—too many zero ratings in a rating matrix. In contrast, the Hierarchical Poisson Factorization (HPF) [6, 15] only utilizes non-zero ratings to make scalable recommendations on large and sparse data.

In addition to user ratings on items, modern e-commerce platforms also generate other kinds of useful data such as social relationships and user reviews. By combining a rating matrix with additional data, better recommendations are expected. On the one hand, user textual reviews contain large amounts of information that often reflect concrete consumer experiences and many item attributes related to product quality. Some studies have utilized review text information for recommendation. By integrating ratings with review text or item contents, works [2, 9, 14, 28, 43] combine factors in ratings with topics in item reviews. On the other hand, the abundant social relationships provide an independent data source for recommendation and another possibility to improve recommended result quality. E.g., existing works [16, 31, 41] factorize rating matrix and social matrix simultaneously to improve recommendation performance.

In general, methods utilizing heterogeneous data tend to perform better than those using a single data source [45]. However, few studies take a rating matrix, social relationships and review text altogether into consideration. Work [20] is such an effort. However, it is simply a linear combination of two methods [2, 41]; it does not leverage the comprehensive implications of the latent factors in the completeness of a rating matrix, review text and social relationships.

In this paper, we propose an approach called Topic Soical Network Poisson Factorization (TSNPF for short). It integrates a rating matrix, review text and social relationships altogether in order

to enable high-quality recommendations. Overall, TSNPF aims to find item attributes and user preferences by exploiting all available heterogeneous data. Specifically, TSNPF merges all reviews on an item/user to generate the *document* for this item/user, in the hope of extracting topic intensities from such documents. When the topics of an item and a user are similar, the rating score of the user on that item tends to be high. However, item attributes and user preferences are not fully parameterized by their documents. To this end, we associate an attribute annex to each item and a preference annex to each user. As a result, the summation of topic intensities and corresponding annexes capture item attributes and user preferences to a greater extent. Rating of a user on an item is generated from the user's preferences and the item's attributes. In addition, most existing recommender systems that use social information assume that friends have similar preferences, which is however unrealistic in real life. In contrast, we utilize social relationships in a different way. Specifically, by assuming the similarity between two friends are generated from both users' preferences, we model the similarity with respect to the items they both have rated. The intuition behind is that a pair of friends who share similar (different) ratings on the same items naturally tend to have similar (different) preferences.

Our TSNPF exploits observations of multiple kinds, namely the ratings, the frequencies of words in the documents of items and users, and the similarity between a pair of friends on the items which they have both rated. In our model, each observation is represented as non-negative integers and generated from a Poisson distribution [25] that is in exponential family over non-negative integers. The latent item/user topic intensity, the item attribute annex and the user preference annex are all sampled from Gamma distributions—another exponential family distribution with shape and rate parameters. TSNPF only scans non-zero ratings, word frequencies and similarities, and factorizes them based on their posterior Gamma distributions. Therefore, TSNPF is able to capture features from sparse observations. By augmenting TSNPF with some auxiliary variables, we make TSNPF conditionally conjugate such that we can apply particular techniques to infer the parameters for it. To fit our TSNPF model, we use variational inference. It is an optimization-based strategy for efficiently approximating posterior distributions in large-scale complex probabilistic models [26, 42]. We propose a simple mean-field [36] variational inference method to update the parameters of TSNPF.

Our contributions are summarized as follows:

- We propose a method based on Gamma-Poisson distribution to extract the topic intensities of items and users from user-generated textual reviews. Compared to previous techniques, our method is able to address the usual problem of data scarcity.
- We propose TSNPF, a conjugate graphical model based on Poisson factorization which only models non-zero observations in ratings, reviews and social relations simultaneously via interpretable user preferences and item attributes. In addition, we propose a closed form mean-field variational inference method to train TSNPF.
- We evaluate the performance of TSNPF using three publicly available real datasets. The results show that TSNPF outperforms state-of-the-art alternatives.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 formulates the research problem and introduces the notations used throughout this paper. Section 4 describes our TSNPF model. Sections 5 details the variational inference for our TSNPF model. Experimental results are shown in Section 6. Section 7 concludes the paper and points out future work directions.

2 RELETED WORK

In this section, we mainly discuss the works which utilize heterogeneous data for recommendations.

Many existing works attempt to extract topics and features from user-generated reviews and learn latent factors from ratings using matrix factorization (MF) methods [10, 11, 18, 33, 40, 47, 52]. A general idea of these methods is to extract latent topics from reviews using topic models [10, 28, 33, 40, 47]. In particular, RMR [28] uses topic models on reviews to learns items' features and models rating using a maxiture of Gaussian. RBLT [40] and ITLFM [47] linearly combine the latent topics and latent factors to form the latent representations for users and items. TopicMF [2] and HPF [33] define a transform action function to link the latent topics and latent factors. He *et al.* [18] model the user-item-aspect relation with a tripartite graph to extract latent topics from reviews. JMARS [11] is an integrated graphical model that makes use of ratings and sentiments together for movie recommendation. However, with the rapid growth of user numbers, the increasing sparsity of the data becomes a critical concern [30]. Recently, neural networks are widely used to process reviews in recommender systems [7, 48, 49, 52]. DeepCoNN [52] first uses two CNNs to process reviews to learn users' and items' representations which are in turn concatenated and passed into a regression layer for rating predictions. However, neural network based methods often have the efficiency problem and their performance decrease greatly when reviews are scarce or unavailable in the testing phase [7].

There are also works [34, 35, 50] analyzing users' sentiments in reviews to improve the quality of recommendations. These works usually rely on the external NLP tools for sentiment analysis and thus are not self-contained.

Besides, item descriptions are also useful data for recommendations. Personalized articles recommendation methods such as CTR [43] and CTPF [14] assume that latent factors of items depend on the latent topic distributions from article contents. CTR [43] combines topic modeling using LDA [5] with Gaussian matrix factorization for one-class collaborative filtering [21]. CTPF [14] models both reader behavior and article texts with Poisson distributions, connecting the latent topics that represent the texts with the latent preferences that represent the readers. However, CTR is not conditionally conjugate and its inference algorithm depends on numerical optimization of topic intensities. Also, CTR and CTPF demand that the items themselves are articles or associated with descriptions. Thus, it is difficult to apply them in other scenarios.

In addition to textual data, social relations is another kind of heterogeneous data often used in recommender systems. LOCABAL [41] takes advantage of both local and global social contexts for recommendation. Ma *et al.* [31] propose a factor analysis approach based on probabilistic matrix factorization called "SoRec". It addresses the data sparsity and poor prediction accuracy problems

by employing both users' social network information and rating records. This idea is also applied in other works [24, 32]. Trust-based approach is another way to utilize social relations to make recommendation [23, 51]. This approach assumes a trust network among users and makes recommendations based on the ratings of the users that are directly or indirectly trusted by a user. Combining TopicMF [2] and LOCABAL [41], synthetic approach MR3 [20] utilizes ratings, social relations and reviews together for recommendation. However, MR3 is essentially a linear combination and it performs worse than our proposed approach according to our experimental results.

Moreover, Zhang *et. al* [46] and He *et al.* [17] consider the effect of visual content such as images when making recommendations. Bao *et. al* [1] and Do *et. al* [12] integrate metadata into their approaches.

3 NOTATIONS AND PROBLEM FORMULATION

Traditional recommender systems usually only use the rating matrix and ignore review text and social relationships. However, it is beneficial to take both of these two aspects into consideration [20]. On the one hand, social relationships often have impacts on a user's impression of items. On the other hand, a rating score can tell if a user likes or dislikes an item, but it cannot tell the reason behind the preference. In contrast, if this rating score is associated with some review text, the chance is better for us to understand why the user likes or dislikes the item. In addition, reviews offer abundant information about items' attributes. We intend to exploit the full combination of the rating matrix, the review text and the social relationships. We fuse three heterogeneous data types in one comprehensive data model to make high quality recommendations. The main notations used throughout the paper are shown in Table 1.

Table 1: Notations

R_{ui}	User u's rating on item i
D_i^I	Document of item i
D_u^U	Document of user u
C_{iw}^I	Count of word w in D_i^I
C_{uw}^U	Count of word w in D_u^U
S_{uvi}	Rating similarity between users u and v on item i
ε_w	latent topic rate for word w
ϵ_w	Latent topic intensities for word w
ϖ_i	latent topic rate for item i
π_i	Latent topic intensities for item i
α_i	latent popularity for item i
β_i	Latent attribute annex for item i
ς_u	Latent topic rate for user u
σ_u	Latent topic intensities for user u
ϑ_u	Latent activity for user u
θ_u	Latent preference annex for user u

Suppose there are U users numbered from 1 to U and I items numbered from 1 to I. The matrix of ratings given by users to items is denoted by $R \in \mathbb{R}^{U\times I}$, where R_{ui} is the rating by user u on item i. Each R_{ui} is an integer satisfying $0 \le R_{ui} \le M$, where M is the maximum rating score and 0 means currently the rating is unavailable. In addition to the rating matrix, the observed review on item i written by user u is denoted as D_{ui}, which is along with a rating score R_{ui}. For every item i, we merge all users' review comments together, generating a *document* D_i^I, i.e., $D_i^I = \bigcup_u D_{ui}$. Likewise, D_u^U denotes the document of a user u, i.e., $D_u^U = \bigcup_i D_{ui}$. We use D^I and D^U to denote the set of all documents of items and users, respectively. Furthermore, we use $G \in \mathbb{R}^{U\times U}$ to denote the user social network where $G_{uv} = 1$ means users u and v are friends and $G_{uv} = 0$ otherwise. Typically, R, D^I, D^U and G are all highly sparse, i.e., having many zero values.

For the recommendation purpose, we need to judge if a user will like an item that he/she has never consumed before. The problem statement for our research is as follows:

Problem Formulation: Given the rating matrix R, the document sets of items and users, i.e., D^I and D^U, and the social network G, estimate user u's rating on item i, i.e., R_{ui} that currently is 0.

4 TOPIC SOCIAL NETWORK POISSON FACTORIZATION

In this section, we describe the topic social network poisson factorization model (TSNPF). We model each user and each item by exploiting the rating matrix R, the documents sets D^I and D^U, and the social network G comprehensively.

Extracting topic intensities from documents. In TSNPF, there is a collection of K topics $\epsilon_{1:V,1:K}$ where each topic $\epsilon_{:,k}$ is composed of a vector of word intensities on the vocabulary and V is the size of the vocabulary. An item i is partly parameterized by the intensities of these topics denoted by $\pi_{i,1:K}$. The document D_i^I is generated by $\epsilon_{1:V}$ and π_i with C_{iw}^I, where C_{iw}^I is the count of word w in D_i and it is modeled by the inner product of ϵ_w and π_i, i.e., $C_{iw}^I \sim \text{Poisson}(\pi_i^\top \epsilon_w)$. Suppose $\sigma_{u,1:K}$ models the topics of user u, and C_{uw}^U is the word count of w in the document D_u^U. Likewise, C_{uw}^U is modeled using $\text{Poisson}(\sigma_u^\top \epsilon_w)$. When topic intensities of an item and a user are similar, the corresponding rating tends to be high. Using word intensities as topics is an widely-used technique [5]. Compared to existing work, our method of extracting topics is built on Poisson factorization which can take advantage of natural sparsity of user/item documents and is more efficient than Gaussian factorization based approaches [14].

Modeling ratings. The documents of items or users can not fully parameterize the items or users [14]. To this end, we make TSNPFassociate K latent attribute annexes $\beta_{i,1:K}$ to each item i and K latent preference annexes $\sigma_{u,1:K}$ to each user u. Such annexes capture the items' and users' deviations from their topic intensities. The rating R_{ui} is modeled by the inner product of the item attributes and user preferences, i.e., $R_{ui} \sim \text{Poisson}\Big((\sigma_u + \theta_u)^\top (\pi_i + \beta_i)\Big)$.

Modeling similarities. In addition, it is reasonable to believe that a user u's ratings on items are influenced by u's friends. When a pair of friends u and v have similar preferences, they are likely to give similar ratings on items. To incorporate this, we define the similarity between u and v on each item i on which they both give ratings, i.e., $S_{uvi} = M - |R_{ui} - R_{vi}|$ where M is the maximum rating score. In TSNPF, we model S_{uvi} using the inner product of users u and v's preferences, i.e., $S_{uv,i} \sim \text{Poisson}\Big((\sigma_u + \theta_u)^\top (\sigma_v + \theta_v)\Big)$. In this way, the distance of a pair of friends' preference vectors tends to be small when the users have similar ratings.

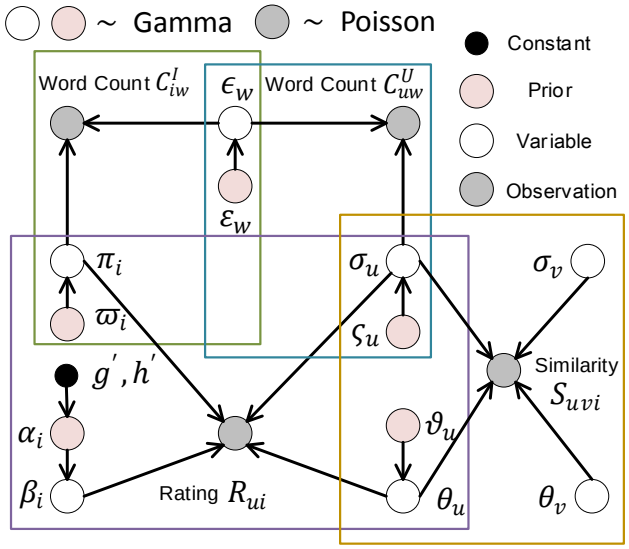

Figure 1: The directed graphical model representing TSNPF.

To deal with the sparsity of users and items, TSNPF also preserves the heterogeneity in the features by placing Gamma priors on ϵ_w, π_i, σ_u, β_i and θ_u [15]. The graphical model of TSNPF is illustrated in Figure 1. To avoid ambiguity, we only draw one constant node (g', h') in the example. More specifically, TSNPF's generative process is described as follows:

1. Generating documents of items and users
   1). For each word w:
      (a) Sample latent topic rate $\varepsilon_w \sim \text{Gamma}(a', b')$.
      (b) Sample word topic intensities $\epsilon_{wk} \sim \text{Gamma}(a, \varepsilon_w)$.
   2). For each item i:
      (a) Sample topic rate $\varpi_i \sim \text{Gamma}(c', d')$.
      (b) Sample document topic intensities $\pi_{ik} \sim \text{Gamma}(c, \varpi_i)$.
      (c) For every word w in D^I_i, sample word count
      $$C^I_{iw} \sim \text{Poisson}(\pi_i^\top \epsilon_w).$$
   3). For each user u:
      (a) Sample topic rate $\varsigma_u \sim \text{Gamma}(e', f')$.
      (b) Sample document topic intensities $\sigma_{uk} \sim \text{Gamma}(e, \varsigma_u)$.
      (c) For every word w in D^U_u, sample word count
      $$C^U_{uw} \sim \text{Poisson}(\sigma_u^\top \epsilon_w).$$
2. Generating ratings
   1). For each item i:
      (a) Sample latent popularity $\alpha_i \sim \text{Gamma}(g', h')$.
      (b) Sample attribute annex for each component k:
      $$\beta_{ik} \sim \text{Gamma}(g, \alpha_i).$$
   2). For each user u:
      (a) Sample latent activity $\vartheta_u \sim \text{Gamma}(m', n')$.
      (b) Sample preference annex for each component k:
      $$\theta_{uk} \sim \text{Gamma}(m, \vartheta_u)$$
   3). For each user u and item i, sample rating
      $$R_{ui} \sim \text{Poisson}\Big((\sigma_u + \theta_u)^\top (\pi_i + \beta_i)\Big).$$
3. Generating rating similarities between friends
   For a pair of users u and v with $G_{uv} = 1$, for every item they both have rated, sample rating similarity:
   $$S_{uvi} \sim \text{Poisson}\Big((\sigma_u + \theta_u)^\top (\sigma_v + \theta_v)\Big).$$

Given the observed rating matrix R, users similarity set S and word count sets C^I and C^U generated by document sets D^I and D^U, respectively, our goal is to infer the topics $\epsilon_{w,1:K}$, the item and user topic intensities $\pi_{i,1;K}$ and $\sigma_{u,1;K}$, the item attribute annex $\beta_{i,1:K}$, the user preference annex $\theta_{u,1:K}$ and their priors, i.e., to estimate the posterior distribution $p(\epsilon, \varepsilon, \pi, \varpi, \sigma, \varsigma, \beta, \alpha, \theta, \vartheta, |R, S, C^I, C^U)$ of these variables. Once the posterior is fit, TSNPF can be used to recommend items to users by ranking users' unconsumed items according to their scores based on the posterior expected Poisson parameters. Such a score for user u and item i is defined as follow:

$$\text{score}(u, i) = \mathbb{E}[(\sigma_u + \theta_u)^\top (\pi_i + \beta_i) | R, S, C^I, C^U]$$

5 INFERENCE FOR TSNPF

It is intractable to compute the exact posterior distribution $p(\epsilon, \varepsilon, \pi, \varpi, \sigma, \varsigma, \beta, \alpha, \theta, \vartheta | R, S, C^I, C^U)$ of TSNPF directly as variables are dependent on each other in p. We use variational inference [26] to approximate it. A family of distributions over latent variables indexed by a set of free parameters is introduced in variational inference. Those parameters are optimized to find the member of the family that is closest to the posterior. Previous studies [39, 44] have proved that the combination of simple distributions (*e.g.*, Poisson and Gaussian distribution) is able to approximate a very complex distribution. Thus, variational inference is reasonable. We use a distribution $q(\epsilon, \varepsilon, \pi, \varpi, \sigma, \varsigma, \beta, \alpha, \theta, \vartheta)$ in mean-field family [36], the simplest variational family, to approximate the exact posterior distribution p.

5.1 Complete Conditionals for TSNPF

We first augment TSNPF with auxiliary variables to make it conditionally conjugate, which means the complete conditional of each latent variable in p is in the exponential family and is in the same family as its prior [13]. A complete conditional is the conditional distribution of a latent variable given the observations and other latent variables. Following previous work [15], for word counts C^I_{iw} and C^U_{uw}, we add K latent variables $x^I_{iw,k} \sim \text{Poisson}(\epsilon_{wk}\pi_{ik})$ and K latent variables $x^U_{iw,k} \sim \text{Poisson}(\epsilon_{wk}\sigma_{uk})$, respectively, where $C^I_{iw} = \sum_k x^I_{iw,k}$ and $C^U_{uw} = \sum_k x^U_{uw,k}$. For rating similarity S_{uvi} that is non-zero, we add $2K$ latent variables $z_{uvi,k}$ which include two parts: $z^\sigma_{uvi,k}$ and $z^\theta_{uvi,k}$ such that $z^\sigma_{uvi,k} \sim \text{Poisson}(\sigma_{uk}\sigma_{vk})$ and $z^\theta_{uvi,k} \sim \textit{Poisson}(\theta_{uk}\theta_{vk})$, where $S_{uvi} = \sum_k (z^\sigma_{uvi,k} + z^\theta_{uvi,k})$ for every item i. For rating R_{ui}, $4K$ latent variables $y^1_{ui,k} \sim \text{Poisson}(\pi_{ik}\sigma_{uk})$, $y^2_{ui,k} \sim \text{Poisson}(\pi_{ik}\theta_{uk})$, $y^3_{ui,k} \sim \text{Poisson}(\beta_{ik}\sigma_{uk})$ and $y^4_{ui,k} \sim \text{Poisson}(\beta_{ik}\theta_{uk})$ are introduced such that $R_{ui} = \sum_k (y^1_{ui,k} + y^2_{ui,k} + y^3_{ui,k} + y^4_{ui,k})$. As a sum of independent Poisson random variables is itself a Poisson random variable with rate equal to the sum of the rates, the new latent variables preserve the marginal distribution of the observations. With the additional auxiliary variables, TSNPF is conditionally conjugate. We define the mean-field family that considers the latent variables

Table 2: TSNPF: latent variables, complete conditionals and variational parameters

Variable	Type	Complete Conditional	Variational Params
ε_w	Gamma	$a' + Ka, b' + \sum_k \epsilon_{wk}$	$\iota_w^{shp}, \iota_w^{rte}$
ϵ_{wk}	Gamma	$a + \sum_i x_{iw,k}^I + \sum_u x_{uw,k}^U, \varepsilon_w + \sum_i \pi_{ik} + \sum_u \sigma_{uk}$	$\mu_{wk}^{shp}, \mu_{wk}^{rte}$
ϖ_i	Gamma	$c' + Kc, d' + \sum_k \pi_{ik}$	$\zeta_i^{shp}, \zeta_i^{rte}$
π_{ik}	Gamma	$c + \sum_w x_{iw,k}^I + \sum_u (y_{ui,k}^1 + y_{ui,k}^2), \varpi_i + \sum_w \epsilon_{wk} + \sum_u (\sigma_{uk} + \theta_{uk})$	$\rho_{ik}^{shp}, \rho_{ik}^{rte}$
α_i	Gamma	$g' + Kc, h' + \sum_k \beta_{ik}$	$\tau_i^{shp}, \tau_i^{rte}$
β_{ik}	Gamma	$g + \sum_u (y_{ui,k}^3 + y_{ui,k}^4), \alpha_i + \sum_u (\sigma_{uk} + \theta_{uk})$	$\lambda_{ik}^{shp}, \lambda_{ik}^{rte}$
ς_u	Gamma	$e' + Ke, f' + \sum_k \sigma_{uk}$	ξ_u^{shp}, ξ_u^{rte}
σ_{uk}	Gamma	$e + \sum_w x_{iw,k}^U + \sum_i (y_{ui,k}^1 + y_{ui,k}^3) + \sum_{v,i,S_{uvi}\neq 0} z_{uvi,k}^{\sigma}$, $\varsigma_u + \sum_w \epsilon_{wk} + \sum_i (\pi_{ik} + \beta_{ik}) + \sum_{v,i,S_{uvi}\neq 0} \sigma_{v,k}$	$\nu_{uk}^{shp}, \nu_{uk}^{rte}$
ϑ_u	Gamma	$m' + Km, n' + \sum_k \theta_{uk}$	$\kappa_u^{shp}, \kappa_u^{rte}$
θ_{uk}	Gamma	$m + \sum_i (y_{ui,k}^2 + y_{ui,k}^4) + \sum_{v,i,S_{uvi}\neq 0} z_{uv,k}^{\theta}, \vartheta_u + \sum_i (\pi_{ik} + \beta_{ik}) + \sum_{v,i,S_{uvi}\neq 0} \theta_{vk}$	$\gamma_{uk}^{shp}, \gamma_{uk}^{rte}$
x_{iw}^I	Mult	$\log \pi_{ik} + \log \epsilon_{wk}$	χ_{iw}^I
x_{uw}^U	Mult	$\log \sigma_{uk} + \log \epsilon_{wk}$	χ_{iw}^U
z_{uvi}	Mult	$\begin{cases} \log \sigma_{uk} + \log \sigma_{vk}, & \text{case } z_{uvi}^{\sigma} \\ \log \theta_{uk} + \log \theta_{vk}, & \text{case } z_{uvi}^{\theta} \end{cases}$	$\varphi_{uv}^{\sigma}, \varphi_{uv}^{\theta}$
y_{ui}	Mult	$\begin{cases} \log \pi_{ik} + \log \sigma_{uk}, & \text{case } y_{ui}^1 \\ \log \pi_{ik} + \log \theta_{uk}, & \text{case } y_{ui}^2 \\ \log \beta_{ik} + \log \sigma_{uk}, & \text{case } y_{ui}^3 \\ \log \beta_{ik} + \log \theta_{uk}, & \text{case } y_{ui}^4 \end{cases}$	$\phi_{ui}^1, \phi_{ui}^2, \phi_{ui}^3, \phi_{ui}^4$

to be independent and each governed by its own distributions.

$$
\begin{aligned}
q(\varepsilon, \epsilon, \varpi, \pi, \alpha, \beta, \varsigma, \sigma, \vartheta, \theta, x^I, x^U, z^{\sigma}, z^{\theta}, y) = & \prod_{w,k} q(\epsilon_{wk}|\mu_{wk}) \\
& \prod_w q(\varepsilon_w|\iota_w) \prod_{i,k} q(\pi_{ik}|\rho_{ik}) q(\beta_{ik}|\lambda_{ik}) \prod_i q(\varpi_i|\zeta_i) q(\alpha_i|\tau_i) \\
& \prod_{u,k} q(\sigma_{uk}|\nu_{uk}) q(\varsigma_u|\xi_u) q(\theta_{uk}|\gamma_{uk}) \prod_u q(\vartheta_u|\kappa_u) \\
& \prod_{i,w} q(x_{iw}^I|\chi_{iw}^I) \prod_{u,w} q(x_{uw}^U|\chi_{iw}^U) \prod_{u,v,i} q(z_{uvi}^{\sigma}, z_{uvi}^{\theta}|\varphi_{uv}^{\sigma}, \varphi_{uv}^{\theta}) \\
& \prod_{u,i} q(y_{ui}^1, y_{ui}^2, y_{ui}^3, y_{ui}^4|\phi_{ui}^1, \phi_{ui}^2, \phi_{ui}^3, \phi_{ui}^4) \qquad (1)
\end{aligned}
$$

The distributions of the variational variables (ε_w, ϵ_{wk}, ϖ_i, π_{ik}, α_i, β_{ik}, ς_u, σ_{uk}, ϑ_u and θ_{uk}) and the auxiliary variables (x^I, x^U, z and y) are all the same as their conditional distributions in p. The complete conditionals of all variables and the variational parameters that govern these variables in q are shown in Table 2. For more detailed derivation, please refer to Appendix A.1

The complete conditionals of ε_w, ϵ_{wk}, ϖ_i, π_{ik}, α_i, β_{ik}, ς_u, σ_{uk}, ϑ_u and θ_{uk} are all Gamma distributions, which are in the exponential family, with shape and rate variational parameters. We denote shape with superscript 'shp' and rate with 'rte'. For example, the variational distribution for the user preference θ_{uk} is Gamma($\gamma_{uk}^{shp}, \gamma_{uk}^{rte}$). For auxiliary variables x_{iw}^I, x_{uw}^U, z_{uvi} and y_{ui}, the complete conditionals are all free multinomials [25]. Specifically, χ_{iw}^I and χ_{uw}^U are all K-vectors that sum to one; $\varphi_{uv} = (\varphi_{uv}^{\sigma}, \varphi_{uv}^{\theta})$ and $\phi_{ui} = (\phi_{ui}^1, \phi_{ui}^2, \phi_{ui}^3, \phi_{ui}^4)$ are points in the $2K$ and $4K$-simplex, respectively.

5.2 Closed Form Variational Parameters Update

In this section, we give an coordinate ascent algorithm [4, 19] to update every variational parameter by holding all other parameters fixed. The detailed derivation can be found in Appendix A.2.

For all words and items, we initialize the word topic intensities ϵ_w using LDA [5]. These operations can be implemented by setting the point-wise ratio μ_w^{shp}/μ_w^{rte} to be ϵ_w. Subsequently, for all users and items, we initialize the user activity rate κ_u^{rte}, user preference γ_u, item popularity rate τ_i^{rte} and item feature λ_i with a small positive noise. We initialize the item topic intensities and attribute annexes i.e., ζ_i^{shp} and τ_i^{shp}, and the user topic intensities and preference annexes shapes, i.e., ξ_u^{shp} and κ_u^{shp}, according to the following equations:

$$
\begin{aligned}
\zeta_i^{shp} &= c' + Kc; \ \tau_i^{shp} = g' + Kg \\
\xi_u^{shp} &= e' + Ke; \ \kappa_u^{shp} = m' + Km
\end{aligned}
$$

Furthermore, the following steps are repeated until convergence:

1. Let

$$
f(r, s) = \exp\left(\Psi(r^{shp}) - \log r^{rte} + \Psi(s^{shp}) - \log s^{rte}\right)
$$

We use $\Psi(\cdot)$ to denote the digamma function. For each word/item such that $C_{iw}^I > 0$, each word/user such that $C_{uw}^U > 0$, each user pairs such that $S_{uvi} > 0$ for some items and each

user/item such that $R_{ui} > 0$, update the multinomials:

$$\hat{\chi}^I_{iw,k} = f(\rho_{ik}, \mu_{wk}) \; ; \; \chi^I_{iw,k} = \frac{\hat{\chi}^I_{iw,k}}{\sum_k \hat{\chi}^I_{iw,k}}$$

$$\hat{\chi}^U_{uw,k} = f(\sigma_{uk}, \mu_{wk}) \; ; \; \chi^U_{uw,k} = \frac{\hat{\chi}^U_{uw,k}}{\sum_k (\hat{\chi}^U_{iw,k})}$$

$$\hat{\varphi}^\sigma_{uv,k} = f(\nu_{uk}, \nu_{vk}) \; ; \; \hat{\varphi}^\theta_{uv,k} = f(\gamma_{uk}, \gamma_{vk})$$

$$\varphi^\sigma_{uv,k} = \frac{\hat{\varphi}^\sigma_{uv,k}}{\sum_k (\hat{\varphi}^\sigma_{uv,k} + \hat{\varphi}^\theta_{uv,k})} \; ; \; \varphi^\theta_{uv,k} = \frac{\hat{\varphi}^\theta_{uv,k}}{\sum_k (\hat{\varphi}^\sigma_{uv,k} + \hat{\varphi}^\theta_{uv,k})}$$

$$\hat{\phi}^1_{ui,k} = f(\rho_{ik}, \nu_{uk}) \; ; \; \hat{\phi}^2_{ui,k} = f(\rho_{ik}, \gamma_{uk})$$

$$\hat{\phi}^3_{ui,k} = f(\lambda_{ik}, \nu_{uk}) \; ; \; \hat{\phi}^4_{ui,k} = f(\lambda_{ik}, \gamma_{uk})$$

$$\phi^j_{ui,k} = \frac{\hat{\phi}^j_{ui,k}}{\sum_k (\hat{\phi}^1_{ui,k} + \hat{\phi}^2_{ui,k} + \hat{\phi}^3_{ui,k} + \hat{\phi}^4_{ui,k})} \; , \; j = 1, 2, 3, 4$$

2. For each item, update the topic prior and intensities, attribute prior and annex parameters:

$$\rho^{shp}_{ik} = c + \sum_w C^I_{iw} \chi^I_{iw,k} + \sum_u R_{ui} (\phi^1_{ui,k} + \phi^2_{ui,k})$$

$$\rho^{rte}_{ik} = \frac{\zeta^{shp}_i}{\zeta^{rte}_i} + \sum_w \frac{\mu^{shp}_{wk}}{\mu^{rte}_{wk}} + \sum_u (\frac{\nu^{shp}_{uk}}{\nu^{rte}_{uk}} + \frac{\gamma^{shp}_{uk}}{\gamma^{rte}_{uk}})$$

$$\lambda^{shp}_{ik} = g + \sum_u R_{ui} (\phi^3_{ui,k} + \phi^4_{ui,k})$$

$$\lambda^{rte}_{ik} = \frac{\tau^{shp}_i}{\tau^{rte}_i} + \sum_u (\frac{\nu^{shp}_{uk}}{\nu^{rte}_{uk}} + \frac{\gamma^{shp}_{uk}}{\gamma^{rte}_{uk}})$$

$$\zeta^{rte}_i = d' + \sum_k \frac{\rho^{shp}_{ik}}{\rho^{rte}_{ik}} ; \tau^{rte}_i = h' + \sum_k \frac{\lambda^{shp}_{ik}}{\lambda^{rte}_{ik}}$$

3. For each user, update the topic prior and intensities, preference prior and annex parameters:

$$\nu^{shp}_{uk} = e + \sum_w C^U_{uw} \chi^U_{uw,k} + \sum_i R_{ui} (\phi^1_{ui,k} + \phi^3_{ui,k}) + \sum_{v,i,S_{uvi} \neq 0} S_{uvi} \varphi^\sigma_{uv,k}$$

$$\nu^{rte}_{uk} = \frac{\xi^{shp}_u}{\xi^{rte}_u} + \sum_w \frac{\mu^{shp}_{wk}}{\mu^{rte}_{wk}} + \sum_i (\frac{\rho^{shp}_{ik}}{\rho^{rte}_{ik}} + \frac{\lambda^{shp}_{ik}}{\lambda^{rte}_{ik}}) + \sum_{v,i,S_{uvi} \neq 0} \frac{\nu^{shp}_{vk}}{\nu^{rte}_{vk}}$$

$$\gamma^{shp}_{uk} = m + \sum_i R_{ui} (\phi^2_{ui,k} + \phi^4_{ui,k}) + \sum_{v,i,S_{uvi} \neq 0} S_{uvi} \varphi^\theta_{uv,k}$$

$$\gamma^{rte}_{uk} = \frac{\kappa^{shp}_u}{\kappa^{rte}_u} + \sum_i (\frac{\rho^{shp}_{ik}}{\rho^{rte}_{ik}} + \frac{\lambda^{shp}_{ik}}{\lambda^{rte}_{ik}}) + \sum_{v,i,S_{uvi} \neq 0} \frac{\gamma^{shp}_{vk}}{\gamma^{rte}_{vk}}$$

$$\xi^{rte}_u = f' + \sum_k \frac{\nu^{shp}_{uk}}{\nu^{shp}_{uk}} ; \kappa^{rte}_u = n' + \sum_k \frac{\gamma^{shp}_{uk}}{\gamma^{shp}_{uk}}$$

4. For each word, update the word topic parameters:

$$\mu^{shp}_{wk} = a + \sum_i C^I_{iw} \chi^I_{iw,k} + \sum_u C^U_{uw} \chi^U_{uw,k}$$

$$\mu^{rte}_{wk} = \frac{\iota^{shp}_w}{\iota^{rte}_w} + \sum_i \frac{\rho^{shp}_{ik}}{\rho^{shp}_{ik}} + \sum_u \frac{\nu^{shp}_{uk}}{\nu^{shp}_{uk}} ; \iota^{rte}_w = b' + \sum_k \frac{\mu^{shp}_{wk}}{\mu^{shp}_{wk}}$$

Specifically, this algorithm only needs the non-zero observations in R, S, C^I and C^U. In step 1 above, we only need to update the variational parameters of the multinomials χ_{iw}, φ_{uv} and ϕ_{ui} for the non-zero word count, user similarity and rating observations, respectively. In steps 2, 3 and 4, all of the sums also only consider the non-zero observations. To judge if the distribution converges, we calculate the log probability of generating a validation rating matrix $\tilde{R}$ at every iteration:

$$\log p(\tilde{R} | \theta, \pi, \beta, \sigma, \theta) = \sum_{u,i,\tilde{R}_{ui} \neq 0} \log p(\tilde{R}_{ui} | \theta_u, \pi_i, \beta_i, \sigma_u, \theta_u)$$

$$= \sum_{u,i,\tilde{R}_{ui} \neq 0} \left(\tilde{R}_{ui} \log \left((\sigma_u + \theta_u)^\top (\pi_i + \beta_i) \right) - \log \tilde{R}_{ui}! \right)$$

$$- (\sum_u \sigma_u + \theta_u)^\top (\sum_i \pi_i + \beta_i)$$

When the change in the log probability is less than a very small threshold, we decide that the distribution converges and terminate the algorithm. The equation above implies that non-zero observations are not necessary when calculating the log probability. Therefore, this algorithm is able to handle sparse data.

6 EXPERIMENTS

In this section, we report on experimental studies that compare our TSNPF with existing methods and investigate the factors having impacts on the performance of TSNPF. In Section 6.1, we describe the settings of our experiments. Section 6.2 details the comparison results between our TSNPF and alternative methods. Section 6.3 reports on the impacts of reviews and social relations on the performance of TSNPF. Finally, Section 6.4 investigates how TSNPF performs as user activeness varies.

6.1 Experimental Settings

Datasets. We use the following open real datasets that are popular and used in the experiments of many recent works [20, 29, 33]:

- **PH-Restaurants** includes 204,887 users, 17,213 restaurants and 728,948 friend pairs. There are on average 1,180 words in the document of each restaurant. It is a part of the dataset of full Yelp data challenge[1] related to Phoenix.
- **LV-Restaurants** is also from Yelp data challenge. It contains 506,278 users, 26,809 restaurants located in Las Vegas and 3,109,068 friend pairs. The average word count of every document is 2,231.
- **Ciao** [2] is a dataset from a knowledge sharing and review website, on which users can rate items, give reviews and connect to others. This dataset includes 9,100 users, 23,432 items and 223,522 friend pairs. Each document contains 1,959 words on average.

[1] https://www.yelp.com/dataset/challenge
[2] https://www.cse.msu.edu/~tangjili/trust.html

The actual ratings on these datasets are all integers ranging from 1 to 5. For every dataset, we only keep top 20,000 words that appear most frequently .

Competing approaches. We compare our TSNPF with the following three typical approaches. They are the most recent representative approaches related to our TSNPF:

- **HPF** [15] factorizes the rating matrix based on Gamma-Poisson distribution. It is the first model in recommender systems that uses Poisson factorization.
- **CuPF** [12] is short for Coupled User Poisson Factorization that learns the couplings between users in the process of Poisson factorization. It is an improvement of HPF.
- **MR3** [20] is a synthetic approach combining ratings, social relationships and reviews. It integrates two other previous works [2, 41].

Evaluation Metrics. We randomly select 20% of the ratings with reviews in each dataset as testing set. Additionally, we set aside 1% of the training ratings and reviews as a validation set which is used to determine the algorithms convergence and to tune variational parameters. We define the relevant and recommended items for user u as those items on which the actual and predicted ratings by u are larger than 3.5, respectively. The following three metrics are used to evaluate the performance of the approaches in our experiments:

- **Normalized Mean Recall** (NMR) is a variant of recall-at-N which adjusts the denominator N'_u for a user u to be the $\min(N, I_u)$ where I_u denotes the most number of items the user u has rated in testing set. It is defined as

$$\text{NMR} = \frac{\sum_u \#\text{ of recommended and relevant items@}N'_u}{\sum_u \#\text{ of relevant items@}N'_u}$$

- **Normalized Mean Precision** (NMP) is a variant of precision-at-N with the denominator defined in normalized mean recall. Likewise,

$$\text{NMP} = \frac{\sum_u \#\text{ of recommended and relevant items@}N'_u}{\sum_u \#\text{ of recommended items@}N'_u}$$

- **Root-Mean-Square Error** (RMSE) is defined as

$$\text{RMSE} = \sqrt{\sum_{u,i}(R_{ui} - \hat{R}_{ui})^2/\mathcal{T}}$$

  Above, $\hat{R}_{ui}$ is the predicted rating of user u on item i and $\mathcal{T}$ is the number of ratings in testing set.

Parameter settings In our settings, the number of latent variables K is set to 100. The parameters a', b', a, c', d', c, e', f', e, g', h', g, m', n' and m in Section 4 are all set to 0.3 [15].

6.2 Overall Results and Analysis

We run top-20 recommendations using all the four approaches on all of the three datasets. Figure 2 reports the average results of the NMR, NMP and RMSE achieved by the four approaches. Overall, TSNPF outperforms the others on all three datasets. First, our TSNPF achieves considerably higher NMR than others, especially HPF and CuPF. This indicates that, compared to the items recommended by other approaches, a much higher fraction of all relevant items are indeed recommended by TSNPF to the respective users. Next, the NMP is high for all approaches and that of TSNPF is the highest. This indicates that all approaches are capable of returning high ratios of relevant items in their recommendations to users and TSNPF is overall the most effective one. Furthermore, the RMSE of TSNPF is also the best (smallest). Compared with HPF and CuPF, TSNPF gains clear RMSE improvements on all of the three datasets. MR3 outperforms HPF and CuPF, but is slightly better than TSNPF only in terms of NMP and RMSE on the Ciao dataset.

As recommender systems are more concerned about recommending more relevant items, i.e., achieving higher NMR, TSNPF performs best among all the four approaches in comparison. These experimental results demonstrate that TSNPF yields higher-quality recommendations.

Both HPF and CuPF use rating matrix only. More concretely, HPF only utilizes the ratings by individual users on items to generate user preferences and item attributes. CuPF attemps to capture the coupling relations between users and rating popularity. To predict the rating of user u on item i, CuPF requires that at least another user v and another item j exist in the training rating matrix such that both items i and j have been rated by user v. This is very demanding. According to our statistics, more than half of the testing ratings cannot be predicted by CuPF in all of the three datasets [3]. This indicates that CuPF is unsuitable for sparse data. Although MR3 utilizes all three types of data, it is simply a linear combination of two existing methods without sophisticated designs. The performance gain of TSNPF implies that jointly modeling the three types of data results in clearly better item recommendations.

6.3 Effects of Reviews and Social Relationships

Note that it is unnecessary for TSNPF that every rating is associated with a review, since we model the topic intensities of users/items based on all reviews related with users/items. Even if no review of a user/item is available, TSNPF is still able to process the relevant data. In case reviews are missing, there is no process of generating topic intensities. As a result, for an item, the ratings on it dominates its attributes. A user's preference is dominated by the ratings he/she post and his/her friends' preferences. To investigate the effect of each data type on TSNPF, we eliminate either reviews or social relationships, or both of them from TSNPF:

- **TSNPF\R** eliminates the effect of reviews by removing step 1 in the generative process (Section 4).
- **TSNPF\S** eliminates the effect of social relationships by removing step 3 in the generative process (Section 4).
- **TSNPF\R\S** eliminates the effect of both reviews and social relationships by removing steps 1 and 3 in the generative process (Section 4). As a result, it becomes HPF.

The recommendation results of TSNPF and its three variants are shown in Figure 3. TSNPF\R and TSNPF\S performs better than TSNPF\R\S on all three datasets, which suggests that both reviews and social relationships contain useful information for item recommendations. However, compared with TSNPF, the performance of a variant degrades when either reviews or social relationships are eliminated. The overall results indicate that TSNPF can make very good use of the heterogeneous information in reviews and social relationships.

[3] For testing data, we randomly decide the ratings from 1 to M.

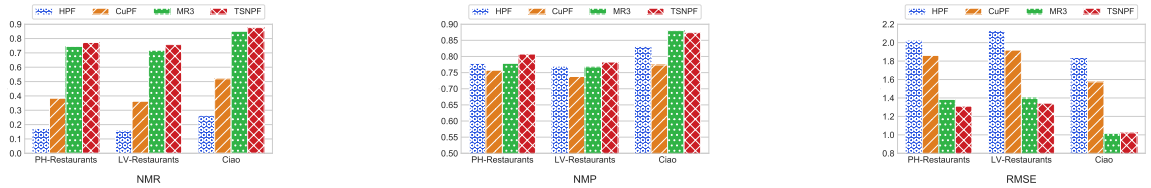

Figure 2: Performance of four approaches on three datasets

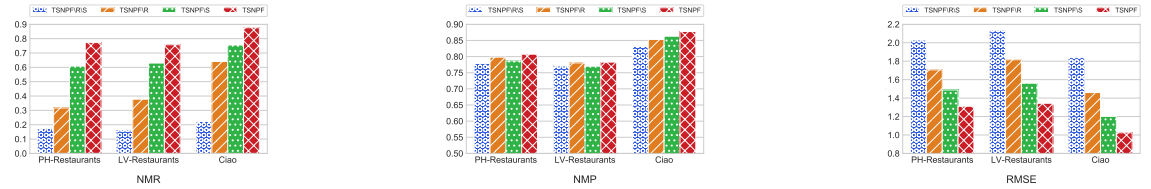

Figure 3: Effects of reviews and social relationships

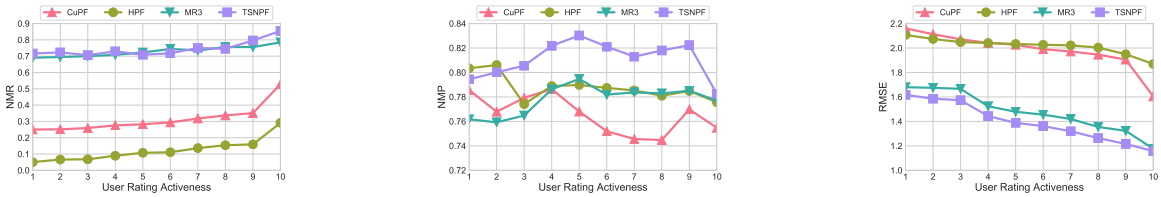

Figure 4: Effect of user rating activeness

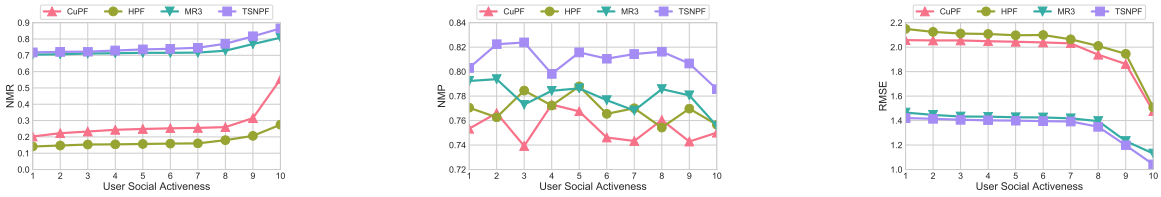

Figure 5: Effect of user social activeness

6.4 The Sensitivity of User Activeness

We treat NMR, NMP and RMSE as a function of user activeness and investigate how the recommendation performance varies across users of different types. We split all users in each of the three datasets into 10 groups ranked by user activeness. The first group is the bottom 10% users who are the least active, i.e., having rated fewest items. The 10th group contains the top 10% most active users. Figure 4 shows the performance results of NMR and NMP at top-20 recommendations and RMSE with different user types on the PH-Restaurants dataset. Due to the page limit, we omit the similar results on other datasets. The results show that all approaches tend to perform better on more active users as such users generate more data that can be used in the models. Again, we see that TSNPF outperforms the others in all but a few cases. This implies that TSNPF exploits larger amounts of user-generated data more effectively.

We also study the effect of the number of users' friends. Similarly, we split all users in each of the three datasets into 10 groups. The first group is the bottom 10% users who are the least sociable, i.e., having fewest friends. The 10th group contains the top 10% most sociable users. Figure 5 shows the performance results of NMR, NMP at top-20 recommendations and RMSE with different user types on the PH-Restaurants dataset.

The results show that all approaches tend to perform better on more sociable users as such users have more social relationships with other users that can be exploited by the models. Again, TSNPF outperforms the others.

7 CONCLUSIONS AND FUTURE WORK

We propose the Topic Social Network Poisson Factorization (TSNPF) to jointly model rating matrix, review texts and social relationships in a comprehensive manner. TSNPF extracts topics of users and

items reasonably as well as supports measuring of user similarities naturally. To address the data sparsity in high-quality recommendations, we use the Gamma-Poission generative process which only uses non-zero observations to model item attributes and user preferences. Experimental results show that TSNPF significantly outperforms alternative methods in item recommendation.

There are other kinds of heterogeneous data in addition to ratings, social relations and review, *e.g.*, metadata and visual content. A potential future work is to integrate them into our model to make even better recommendations. In addition, TSNPF only iterates over the non-zero observations. Thus, the coordinate ascent algorithm in our TSNPF is efficient. However, it is still difficult to fit TSNPF to very large datasets and therefore a stochastic variational inference algorithm [19] is needed in such cases. Furthermore, our approach uses a directed graphical model (Bayesian network) to capture user preferences and item features. Nevertheless, it is interesting to use a neural network instead of the graphical model, as studies show that neural networks can approximate arbitrarily complex functions [3] and thus have more powerful expression ability.

A APPENDIX

A.1 Derivation of the Complete Conditionals

To facilitate inference, we want to make TSNPF conditionally conjugate. To achieve this, we introduce some auxiliary vector variables (x^I_{iw}, x^U_{uw}, z_{uvi} and y_{ui}) in which each element is sampled from a Poisson distribution. Due to the given Poisson observations and Gamma priors of variables, the complete conditionals of these variables are still Gamma distributions. Hence, we use these auxiliary variables instead of actual observations to derive the complete conditionals of Gamma variables. Take ϵ_{wk} as an example, as described in the generative process, the conjugate prior of ϵ_{wk} is

$$\hat{p}(\epsilon_{wk}) = \frac{\varepsilon_w^a}{\Gamma(a)}\epsilon_{wk}^a \exp(-\varepsilon_w\epsilon_{wk})$$
$$= \overbrace{\hat{h}(\epsilon_{wk})}^{\text{Base measure}} \exp\Big((a-1,-\varepsilon_w)^\top(\log\epsilon_{wk},\epsilon_{wk}) - \overbrace{A\big((a-1,-\varepsilon_w)\big)}^{\text{Log-normalizer}}\Big).$$

Here, $(a-1,-\varepsilon_w)$ and $(\log\epsilon_{wk},\epsilon_{wk})$ are the natural parameters and sufficient statistics of this prior, respectively. Given the auxiliary variables x^I and x^U and other variables $\pi_{:,k}$ and $\sigma_{:,k}$, the conditional distribution is described as follows:

$$\begin{aligned}
&p(\epsilon_{wk}|x^I,x^U,\pi_{:,k},\sigma_{:,k})\\
&\propto \hat{p}(\epsilon_{wk})\prod_i p(x^I_{iw,k}|\epsilon_{wk},\pi_{ik})\prod_u p(x^U_{uw,k}|\epsilon_{wk},\sigma_{uk})\\
&\propto \hat{p}(\epsilon_{wk})\prod_i \frac{(\epsilon_{wk}\sigma_{uk})^{x^U_{uw,k}}\exp(-\epsilon_{wk}\sigma_{uk})}{x^U_{uw,k}!}\prod_u p(x^U_{uw,k})\\
&\propto \hat{p}(\epsilon_{wk})\prod_i\Big(\frac{1}{x^I_{iw,k}!}\exp\Big((\log\epsilon_{wk}+\log\sigma_{uk})x^U_{uw,k}-\epsilon_{wk}\sigma_{uk}\Big)\Big)\\
&\prod_u p(x^U_{uw,k})\\
&\propto \hat{h}(\epsilon_{wk})\prod_i\frac{1}{x^I_{iw,k}!}\exp\Big(\overbrace{\Big(a+\sum_i x^I_{iw,k}-1,-(\varepsilon_w+\sum_i\pi_{ik})\Big)}^{\eta}{}^\top\\
&(\log\epsilon_{wk},\epsilon_{wk})-A(\eta)\Big)\prod_u p(x^U_{uw,k})\\
&\propto \overbrace{(\hat{h}(\epsilon_{wk})\prod_i\frac{1}{x^I_{iw,k}!}\prod_u\frac{1}{x^U_{uw,k}!})}^{\text{New base measure } h(\epsilon_{wk})}\\
&\exp\Big(\overbrace{\Big(a+\sum_i x^I_{iw,k}+\sum_u x^U_{uw,k}-1,-(\varepsilon_w+\sum_i\pi_{ik}+\sum_u\sigma_{uk})\Big)}^{\eta}{}^\top\\
&(\log\epsilon_{wk},\epsilon_{wk})-\overbrace{A(\eta)}^{\text{New log-normalizer}}\Big)
\end{aligned}$$

Subsequently, $\big(a+\sum_i x^I_{iw,k}+\sum_u x^U_{uw,k}-1,-(\varepsilon_w+\sum_i\pi_{ik}+\sum_u\sigma_{uk})\big)$ are the natural parameters of the conditional distribution of $p(\epsilon_{wk}|x^I,x^U,\pi_{:,k},\sigma_{:,k})$. Thus,

$$\epsilon_{wk}|x^I,x^U,\epsilon_{:,k},\pi_{:,k},\sigma_{:,k} \sim \text{Gamma}(a+\sum_i x^I_{iw,k}+\sum_u x^U_{uw,k},\varepsilon_w+\sum_i\pi_{ik}+\sum_u\sigma_{uk}) \quad (2)$$

For conditional distribution of ε_w, the latent topic rate,

$$\begin{aligned}
p(\varepsilon_w|\epsilon_{:,k}) &\propto \hat{h}(\varepsilon_w)\exp\Big((a'-1,-b)^\top(\log\varepsilon_w,\varepsilon_w)-A(a',b')\Big)\\
&\prod_k\frac{(\epsilon_{wk})^{a-1}}{\Gamma(a)}\exp\Big((a,-\epsilon_{wk})^\top(\log\varepsilon_w,\varepsilon_w)\Big)\\
&\propto \Big(\hat{h}(\varepsilon_w)\prod_k\frac{(\epsilon_{wk})^{a-1}}{\Gamma(a)}\Big)\exp\Big(\Big(a'+Ka-1,-(b+\sum_k\epsilon_{wk})\Big)^\top\\
&(\log\varepsilon_w,\varepsilon_w)\Big)\\
&\Rightarrow\ \varepsilon_w|\epsilon_{:,k}\sim\text{Gamma}(a'+Ka,b'+\sum_k\epsilon_{wk})
\end{aligned}$$

The variational parameters in the conditional distributions of other variables (π_{ik}, ϖ_i, β_{ik}, α_i, σ_{uk}, ς_u, θ_{uk} and ϑ_u) are derived likewise.

The final latent variables are the auxiliary variables. Recall that each x^I_{iw} or x^U_{uw} is a K-vector of Poisson counts that sum to the observations C^I_{iw} or C^U_{uw}, respectively; each z_{uvi}, in the $2K$ simplex,

or y_{ui}, in the $4K$ simplex, is a vector of Poisson counts that sum to the observations S_{uvi} and R_{ui}, respectively. It is proved that the conditional distribution of a set of Poisson variables, given their sum, is a multinomial for which the parameters are their normalized set of rates [8, 25]. Thus, the complete conditionals for these vectors are

$$x^I_{iw}|\pi_i, \epsilon_w, C^I_{iw} \sim \text{Mult}(C^I_{iw}, \frac{\pi_i \epsilon_w}{\sum_k \pi_{ik}\epsilon_{wk}}) \tag{3}$$

$$x^U_{uw}|\sigma_u, \epsilon_w, C^U_{uw} \sim \text{Mult}(C^U_{uw}, \frac{\sigma_u \epsilon_w}{\sum_k \sigma_{uk}\epsilon_{wk}})$$

$$z_{uvi}|\sigma_u, \theta_u, \sigma_v, \theta_v, S_{uvi} \sim \text{Mult}\Big(S_{uvi}, \frac{(\sigma_u, \theta_u) \odot (\sigma_v, \theta_v)}{\sum_k (\sigma_{uk}\sigma_{vk} + \theta_{uk}\theta_{vk})}\Big)$$

$$y_{ui}|\pi_i, \beta_i, \sigma_u, \theta_u, R_{ui} \sim \text{Mult}\Big(R_{ui}, \frac{(\pi_i, \pi_i, \beta_i, \beta_i) \odot (\sigma_u, \theta_u, \sigma_u, \theta_u)}{\sum_k (\pi_{ik}\sigma_{uk} + \pi_{ik}\theta_{uk} + \beta_{ik}\sigma_{uk} + \beta_{ik}\theta_{uk})}\Big)$$

Above, $\odot$ denotes the element-wise multiplication operation.

A.2 Derivation of the Parameters Update

The objective of variational inference is to minimize the KL divergence between an exponential family member q and the posterior p. For similarity, we use $\boldsymbol{X}$ and $\boldsymbol{V}$ to denote the set of all observations and the set of all variational variables, respectively. Suppose $\boldsymbol{\Lambda}$ is the parameters of the distribution $q(\boldsymbol{V})$ that governs $\boldsymbol{V}$, and the distribution $q(\boldsymbol{V})$ and the conditional distribution $p(\boldsymbol{V}|\boldsymbol{X})$ are both in the exponential families. In this case, we let $\boldsymbol{\Lambda}$ be the natural parameters of $q(\boldsymbol{V})$. Thus,

$$\begin{aligned} p(\boldsymbol{V}|\boldsymbol{X}) &= h(\boldsymbol{V}) \exp\Big(\eta(\boldsymbol{V}|\boldsymbol{X}, p)^\top T(\boldsymbol{X}) - A(\eta(\boldsymbol{V}|\boldsymbol{X}, p))\Big) \\ q(\boldsymbol{V}) &= h(\boldsymbol{V}) \exp\Big(\boldsymbol{\Lambda}^\top T(\boldsymbol{V}) - A(\boldsymbol{\Lambda})\Big) \end{aligned}$$

Above, $\eta(\cdot)$, $T(\cdot)$, $h(\cdot)$ and $A(\cdot)$ are the functions of natural parameters, sufficient statistics, base measure and log-normalizer, respectively. Thus, $\boldsymbol{\Lambda} = \eta(\boldsymbol{V}, q)$. In practice, we often transfer this minimization into maximization of a lower bound on the logarithm of the marginal probability of the observations $\log p(\boldsymbol{X}, \boldsymbol{V})$ called the evidence lower bound (ELBO) [22] of q. It is defined as $\mathcal{L}(q) = \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{V})] - \mathbb{E}_q[\log q(\boldsymbol{V})]$. The KL divergence is equal to the negative ELBO up to $\log p(\boldsymbol{X})$ which is a constant as it does not depend on q:

$$\begin{aligned} \text{KL}\Big(q(\boldsymbol{V})||p(\boldsymbol{V}|\boldsymbol{X})\Big) &= \mathbb{E}_q[\log q(\boldsymbol{V}) - \mathbb{E}_q[\log p(\boldsymbol{V}|\boldsymbol{X})] \\ &= \mathbb{E}_q[\log q(\boldsymbol{V})] - \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{V})] + \log p(\boldsymbol{X}) \\ &= -\mathcal{L}(q) + \text{const} \end{aligned}$$

Besides,

$$\begin{aligned} \mathcal{L}(q) &= \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{V})] - \mathbb{E}_q[\log q(\boldsymbol{V})] \\ &= \mathbb{E}_q[\log p(\boldsymbol{V}|\boldsymbol{X})] - \mathbb{E}_q[\log q(\boldsymbol{V})] + \text{const} \\ &= \mathbb{E}_q[\eta(\boldsymbol{V}|\boldsymbol{X}, p)^\top \nabla_{\boldsymbol{\Lambda}} A(\boldsymbol{\Lambda})] - \boldsymbol{\Lambda}^\top \nabla_{\boldsymbol{\Lambda}} A(\boldsymbol{\Lambda}) + A(\boldsymbol{\Lambda}) + \text{const} \\ &\quad \Big(\text{Because } \mathbb{E}_q[T(\boldsymbol{V})] = \nabla_{\boldsymbol{\Lambda}} A(\boldsymbol{\Lambda})\Big) \\ &\quad \Rightarrow \nabla_{\boldsymbol{\Lambda}} \mathcal{L} = \nabla^2_{\boldsymbol{\Lambda}} A(\boldsymbol{\Lambda}) \Big(\mathbb{E}_q[\eta(\boldsymbol{V}|\boldsymbol{X}, p)] - \boldsymbol{\Lambda}\Big) \end{aligned}$$

Setting the gradient $\nabla_{\boldsymbol{\Lambda}} \mathcal{L}$ to be zero, we get the closed form parameters update: $\boldsymbol{\Lambda} = \mathbb{E}_q[\eta(\boldsymbol{V}|\boldsymbol{X}, p)]$.

When $q(\boldsymbol{V})$ is a mean-field family member, the natural parameters of $q(\boldsymbol{V})$ are just the Cartesian product of its corresponding elements, respectively. Suppose $\boldsymbol{V} = \{v_1, ..., v_{|V|}\}$ and $\lambda_1, ...\lambda_V$ are the natural parameters of the distributions $q(v_1), ..., q(v_{|V|})$, respectively. Then $\boldsymbol{\Lambda} = (\lambda_1, ..., \lambda_{|V|})$. In addition, directly getting $\eta(\boldsymbol{V}|\boldsymbol{X}, p)$ is difficult as it involves all of the variable and observations. Thus we usually iteratively optimize every natural parameter λ_i for $1 \le i \le |V|$ using coordinate ascent algorithm by holding the parameters of all other variables fixed. Note that in this case these fixed variables become observations and the conditional distribution of a variable becomes the corresponding complete conditional. Each λ_i equals to the expected natural parameter (under q) of the complete conditional of v_i, i.e.,

$$\lambda_i = \mathbb{E}_q[\eta(v_i|\boldsymbol{X}, \boldsymbol{V}\backslash v_i, p)]$$

More details about coordinate ascent algorithm can be found in [4, 19].

By applying the derivations above into our TSNPF, we can easily derive the closed form variational parameters update in the main body of the paper. For simplicity, here we only give two examples: 1). μ_{wk}, the variational parameters of ϵ_{wk} which is sampled from a Gamma distribution and 2). χ^I_{iw}, the variational parameters of x^I_{iw} of which every element $x^I_{iw,k}$ is sampled from a Poisson distribution but the conditional distribution given other observations and variables is a multinomial. The updates for the variational parameters of the other Gamma and multinomial latent variables are similarly derived.

From Eq.(2), the natural parameters of the conditional distribution $p(\epsilon_{wk}|x^I, x^U, \pi_{:,k}, \sigma_{:,k}))$ and the distribution $q(\epsilon_{wk}|\mu_w k)$ are $\Big(a + \sum_i x^I_{iw,k} + \sum_u x^U_{uw,k} - 1, -(\varepsilon_w + \sum_i \pi_{ik} + \sum_u \sigma_{uk})\Big)$ and $(\mu^{shp}_{wk} - 1, -\mu^{rte}_{wk})$, respectively. As the natural parameters $(\mu^{shp}_{wk} - 1, -\mu^{rte}_{wk})$ are the affine transformation of shape and rate parameters—μ^{shp}_{wk} and μ^{rte}_{wk}, the update of variational parameters of gamma variable ϵ_{wk} are just the expectations of the shape and rate parameters of the complete conditional distributions under q, i.e.,

$$\begin{aligned} \mu^{shp}_{wk} &= \mathbb{E}_q[a + \sum_i x^I_{iw,k} + \sum_u x^U_{uw,k}] \\ &= a + \sum_i C^I_{iw}\chi^I_{iw,k} + \sum_u C^U_{uw}\chi^U_{uw,k} \\ \mu^{rte}_{wk} &= \mathbb{E}_q[\varepsilon_w + \sum_i \pi_{ik} + \sum_u \sigma_{uk}] \\ &= \frac{\iota^{shp}_w}{\iota^{rte}_w} + \sum_i \frac{\rho^{shp}_{ik}}{\rho^{shp}_{ik}} + \sum_u \frac{\nu^{shp}_{uk}}{\nu^{shp}_{uk}} \end{aligned}$$

Notice that the expectation of $x^I_{iw,k}$ equals to k-th probability of the multinomial $\chi^I_{iw,k}$ times the word count C^I_{iw}, i.e., $\mathbb{E}_q[x^I_{iw,k}] = C^I_{iw}\chi^I_{iw,k}$.

The natural parameters of $x^I_{iw,k}$ is $\log \chi^I_{iw,k}$. Due to the assumption of mean-field family, the natural parameters of x^I_{iw} are just the Cartesian product of every $\log \chi^I_{iw,k}$, i.e., $\eta(x^I_{iw}) = (\log \chi^I_{iw,1}, ..., \log \chi^I_{iw,K})$. From Eq. (3), the conditional distribution $p(x^I_{iw}|\pi_i, \epsilon_w, C^I_{iw})$ is a multinomial. The natural parameters of this conditional distribution are the logarithms of event probabilities, i.e., $\eta(x^I_{iw}|\pi_i, \epsilon_w, C^I_{iw})$ $= \Big((\log \pi_{i1} + \log \epsilon_{w1}) - r, ..., (\log \pi_{iK} + \log \epsilon_{wK}) - r\Big)$ where $r =$

$\log(\sum_k \pi_{ik}\epsilon_{wk})$. Thus,

$$\begin{aligned}\log \chi^I_{iw,k} &= \mathbb{E}_q[\log \pi_{ik} + \log \epsilon_{wk} - r] \Rightarrow \\ \chi^I_{iw,k} &= \exp(\mathbb{E}_q[\log \pi_{ik} + \log \epsilon_{wk} - r]) \\ &\propto \exp\left(\Psi(\rho^{shp}_{ik}) - \log \rho^{rte}_{ik} + \Psi(\mu^{shp}_{wk}) - \log \mu^{rte}_{wk}\right)\end{aligned}$$

Above, $\Psi(\cdot)$ is the digamma function. This update comes from the expectation of the log of a Gamma variable, e.g., $\mathbb{E}_q[\log \pi_{ik}] = \Psi(\rho^{shp}_{ik}) - \log \rho^{rte}_{ik}$. Let

$$\hat{\chi}^I_{iw,k} = \exp\left(\Psi(\rho^{shp}_{ik}) - \log \rho^{rte}_{ik} + \Psi(\mu^{shp}_{wk}) - \log \mu^{rte}_{wk}\right)$$

Then

$$\chi^I_{iw,k} = \frac{\hat{\chi}^I_{iw,k}}{\sum_k \hat{\chi}^I_{iw,k}}. \tag{4}$$

Eq. (4) guarantees that χ^I_{iw} is a K-vector whose elements sum to one.

ACKNOWLEDGMENTS

REFERENCES

[1] Xinlong Bao, Lawrence Bergman, and Rich Thompson. 2009. Stacking recommendation engines with additional meta-features. In *RecSys*. 109–116.

[2] Yang Bao, Hui Fang, and Jie Zhang. 2014. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation. In *AAAI*. 2–8.

[3] Andrew R. Barron. 1994. Approximation and Estimation Bounds for Artificial Neural Networks. *Machine Learning* 14, 1 (1994), 115–133.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* 3 (2003), 993–1022.

[6] John F. Canny. 2004. GaP: a factor model for discrete data. In *SIGIR*. 122–129.

[7] Rose Catherine and William W. Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*. 288–296.

[8] Ali Taylan Cemgil. 2009. Bayesian inference for nonnegative matrix factorisation models. *Computational intelligence and neuroscience* 2009 (2009).

[9] Muthusamy Chelliah and Sudeshna Sarkar. 2017. Product Recommendations Enhanced with Reviews. In *RecSys*. 398–399.

[10] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan S. Kankanhalli. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *WWW*. 639–648.

[11] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *KDD*. 193–202.

[12] Trong Dinh Thac Do and Longbing Cao. 2018. Coupled Poisson Factorization Integrated With User/Item Metadata for Modeling Popular and Sparse Ratings in Scalable Recommendation. In *AAAI*.

[13] Andrew Gelman et al. 2006. Prior distributions for variance parameters in hierarchical models. *Bayesian analysis* 1, 3 (2006), 515–534.

[14] Prem Gopalan, Laurent Charlin, and David M. Blei. 2014. Content-based recommendations with Poisson factorization. In *NIPS*. 3176–3184.

[15] Prem Gopalan, Jake M. Hofman, and David M. Blei. 2015. Scalable Recommendation with Hierarchical Poisson Factorization. In *UAI*. 326–335.

[16] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. In *AAAI*. 123–129.

[17] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517.

[18] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *CIKM*. 1661–1670.

[19] Matthew D. Hoffman, David M. Blei, Chong Wang, and John William Paisley. 2013. Stochastic variational inference. *JMLR* 14, 1 (2013), 1303–1347.

[20] Guang-Neng Hu, Xin-Yu Dai, Yunya Song, Shujian Huang, and Jiajun Chen. 2015. A Synthetic Approach for Recommendation: Combining Ratings, Social Relations, and Reviews. In *IJCAI*. 1756–1762.

[21] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. 263–272.

[22] Tommi S. Jaakkola and Michael I. Jordan. 1996. Computing upper and lower bounds on likelihoods in intractable networks. In *UAI*. 340–348.

[23] Mohsen Jamali and Martin Ester. 2009. *TrustWalker*: a random walk model for combining trust-based and item-based recommendation. In *KDD*. 397–406.

[24] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2012. Social contextual recommendation. In *CIKM*. 45–54.

[25] Norman L Johnson, Adrienne W Kemp, and Samuel Kotz. 2005. *Univariate discrete distributions*. Vol. 444. John Wiley & Sons.

[26] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning* 37, 2 (1999), 183–233.

[27] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[28] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *RecSys*. 105–112.

[29] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining Quality Phrases from Massive Text Corpora. In *SIGMOD*. 1729–1744.

[30] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Coevolutionary Recommendation Model: Mutual Learning between Ratings and Reviews. In *WWW*. 773–782.

[31] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: social recommendation using probabilistic matrix factorization. In *CIKM*. 931–940.

[32] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. 287–296.

[33] Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*. 165–172.

[34] Nikolaos Pappas and Andrei Popescu-Belis. 2013. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *SIGIR*. 773–776.

[35] Stefan Pero and Tomás Horváth. 2013. Opinion-Driven Matrix Factorization for Rating Prediction. In *UMAP*. 1–13.

[36] Carsten Peterson and James R. Anderson. 1987. A Mean Field Theory Learning Algorithm for Neural Networks. *Complex Systems* 1, 5 (1987), 995–1019.

[37] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NIPS*. 1257–1264.

[38] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[39] Brajendra C Sutradhar and Zhende Qu. 1998. On approximate likelihood inference in a Poisson mixed model. *Canadian Journal of Statistics* 26, 1 (1998), 169–186.

[40] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews. In *IJCAI*. 2640–2646.

[41] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Exploiting Local and Global Social Context for Recommendation. In *IJCAI*. 2712–2718.

[42] Martin J. Wainwright and Michael I. Jordan. 2008. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning* 1, 1-2 (2008), 1–305.

[43] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *KDD*. 448–456.

[44] Mike West. 1993. Approximating posterior distributions by mixture. *Journal of the Royal Statistical Society. Series B (Methodological)* (1993), 409–422.

[45] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. 2014. Data Mining with Big Data. *IEEE Trans. Knowl. Data Eng.* 26, 1 (2014), 97–107.

[46] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *KDD*. 353–362.

[47] Wei Zhang and Jianyong Wang. 2016. Integrating Topic and Latent Factors for Scalable Personalized Review-based Rating Prediction. *IEEE Trans. Knowl. Data Eng.* 28, 11 (2016), 3013–3027.

[48] Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative Multi-Level Embedding Learning from Reviews for Rating Prediction. In *IJCAI*. 2986–2992.

[49] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. 2017. Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources. In *CIKM*. 1449–1458.

[50] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*. 83–92.

[51] Tong Zhao, Chunping Li, Mengya Li, Qiang Ding, and Li Li. 2013. Social recommendation incorporating topic mining and social trust analysis. In *CIKM*. 1643–1648.

[52] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*. 425–434.