# Joint Factorizational Topic Models for Cross-City Recommendation

Lin Xiao[1] and Zhang Yongfeng[2]

[1] Institute of Interdisciplinary Information Sciences, Tsinghua University, Beijing, China, jackielinxiao@gmail.com;
[2] College of Information and Computer Science, University of Massachusetts Amherst, MA 01003, USA, yongfeng@cs.umass.edu

**Abstract.** The research of personalized recommendation techniques today has mostly parted into two mainstream directions, namely, the factorization-based approaches and topic models. Practically, they aim to benefit from the numerical ratings and textual reviews, correspondingly, which compose two major information sources in various real-world systems, including Amazon, Yelp, eBay, Netflix, and many others.

However, although the two approaches are supposed to be correlated for their same goal of accurate recommendation, there still lacks a clear theoretical understanding of how their objective functions can be mathematically bridged to leverage the numerical ratings and textual reviews collectively, and why such a bridge is intuitively reasonable to match up their learning procedures for the rating prediction and top-N recommendation tasks, respectively.

In this work, we exposit with mathematical analysis that, the vector-level randomization functions to harmonize the optimization objectives of factorizational and topic models unfortunately do not exist at all, although they are usually pre-assumed and intuitively designed in the literature.

Fortunately, we also point out that one can simply avoid the seeking of such a randomization function by optimizing a Joint Factorizational Topic (JFT) model directly. We further apply our JFT model to the cross-city Point of Interest (POI) recommendation tasks for performance validation, which is an extremely difficult task for its inherent cold-start nature. Experimental results on real-world datasets verified the appealing performance of our approach against previous methods with pre-assumed randomization functions in terms of both rating prediction and top-N recommendation tasks.

Keywords: Topic Model, Recommendation, Factorizational Model

## 1 Introduction

The vast amount of items in various web-based applications has made it an essential task to construct reliable Personalized Recommender Systems (PRS) [25]. With the ability to leverage the wisdom of crowds, the Collaborative Filtering (CF)-based [29,16] approaches have achieved significant success and wide application, especially for those Latent Factor Models (LFM) [12] based on Matrix Factorization (MF) [30] techniques, which attempt to model the preferences of users and items collectively through multivariate hidden factors, so as to make recommendations based on numerical star rating predictions.

Recently, researchers have been putting attention on another important information source in many online systems, namely, the textual user reviews. Usually, the ratings and reviews come in pairs in many typical applications, *e.g.*, Amazon and Yelp. While the ratings act as integrated indicators of user attitudes towards products, the reviews serve as more detailed explanations of what aspects users care about and why the corresponding rating is made [34,33].

As such, the application of Topic Models [4] has gained attention to leverage the textual reviews for personalized recommendation, especially the frequently used Latent Dirichlet Allocation (LDA) [5] technique and its variants, for their ability to extract latent topics/aspects from reviews, which represent the inherently actual factors that users care about when making numerical ratings [21,20]. This further leads to the recent research direction to bridge the LFM and LDA models, which makes use of the ratings and reviews collectively for personalized recommendation [21,20,2,32,23].

However, without a clear mathematical understanding of how the objective functions of LFM and LDA interact with each other when bridged for unified model learning, current approaches have to base themselves on unvalidated and pre-assumed designations to bridge the inherently heterogenous objective functions. For example, McAuley et al [21] transform the latent factors in LFM to topic distributions in LDA through a manually designed randomization function based on logistic normalization, while Ling et al [20] let the factors and topics be the same by assuming them to be sampled from mixture Gaussian distributions.

In this work, we investigate the mathematical relations between the probability of recommending an item to a user and the estimated user-item correlations by LFM or LDA models. Based on this, we prove that a multiplicatively monotonic randomization function that transforms latent factors in LFM to topic distributions in LDA actually does not exist at all. As a result, although some normalization-based transformations seem to be intuitional in previous work [21], they actually make the objective functions of LFM and LDA conflict with each other during optimization procedure, where a higher value of log-likelihood in the LDA component may force a lower rating prediction in the LFM component, which is not favoured when bridging the two models.

Fortunately, we further find that instead of transforming a latent factor to a topic distribution separately, we can simply transform the product of latent factors in LFM to the corresponding product of topic distributions in LDA as a whole, so as to avoid the seeking of a theoretically nonexistent randomization function. This is because what we really care about in practice is the final product of the user/item latent factors (in LFM) or topic distributions (in LDA), where the former accounts for the predicted user-item ratings, and the latter affects the log-likelihood of the observed reviews. Based on these findings, we propose the Joint Factorizational Topic (JFT) model to bridge LFM and LDA, so as to adopt the numerical ratings and textual reviews collectively, and at the same time guarantee the inner-model consistency between the LFM and LDA components.

## 2 Related Work

With the continuous growth of various online items across a vast range of the Web, Personalized Recommender Systems (PRS) [25] have set their missions to save users from information overload [14], and they have been widely integrated into various online applications in the forms of, for example, product recommendation in e-commerce

[19], friend recommendation in social networks [3], news article recommendation in web portals [7], and video recommendation in video sharing websites [8], etc.

Early systems of personalized recommendations rely on content-based approaches [22], which construct the user/item content profiles and make recommendation by paring users with the contently similar items. Content-based approaches usually gain good accuracy but functionally lack the ability of providing recommendations with novelty, serendipity, and flexibility. Besides, they usually require a large amount of expensive human annotations [25]. This further leads to the prospering of Collaborative Filtering (CF)-based recommendation algorithms [16,29] that leverage the wisdom of the crowds. Typically, they construct the partially observed user-item rating matrix and conduct missing rating prediction based on the historical records of a user, as well as those of the others.

With widely recognized performance in rating prediction, scalability, and computational efficiency, the Latent Factor Models (LFM) [12] based on Matrix Factorization (MF) [30] techniques for CF have been extensively investigated by the research community, and widely applied in practical systems. Perhaps the most early and representative formalization of LFM for recommendation dates back to Koren et al [15], and other variants for personalization include Non-negative Matrix Factorization (NMF) [17], Probabilistic Matrix Factorization (PMF) [27,26], and Maximum Margin Matrix Factorization (MMMF) [28], etc. Despite the important success in rating prediction, the CF approaches based solely on the numerical ratings suffer from the problems of explainability [34], cold-start [18], and the difficulty to provide more specific recommendations that meet targeted item aspects [13]. Besides, related research results show that the performance on numerical rating prediction does not necessarily relate to the performance on practical top-N recommendations [6], and that the numerical star ratings may not always be a reliable indicator of users' attitudes towards items [35].

To alleviate these problems, researchers have been investigating the incorporation of textual reviews for recommendation, which is another important information source beyond the star ratings in many systems [31]. Early approaches rely on manually extracted item aspects from reviews for more informed recommendation [1,13] and rating prediction [11,9], which improved the performance but also required extensive human participations. As a results, researchers recently have begun to investigate the possibility of integrating the automatic topic modeling techniques on textual reviews and the latent factor modeling approach on numerical ratings for boosted recommendation, and have achieved appealing results [21,2,32].

However, without a clear mathematical exposition of the relationships between latent factor models and topic modeling, current approaches have to base themselves on manually designed randomization functions or probabilistic distributions. In this work, however, we attempt to make an exposition on the relationships between the two types of objective functions, and further bridge the inherently heterogenous models in a harmonious way for recommendation with the power of both numerical ratings and textual reviews.

## 3 Preliminaries and Definitions

### 3.1 Latent Factor Models (LFM)

Latent Factor Models (LFM) [16] attempt to encode user and item preferences in a latent factor space so as to estimate the user-item relations for rating prediction, which

account for many of the frequently used Matrix Factorization (MF) [30] techniques. Among those, a 'standard' and representative formalization [15] predicts the user-item ratings $r_{u,i}$ with user/item biases and latent factors by,

$$rate(u,i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i \tag{1}$$

where $\alpha$ is the global offset, $\beta_u$ and $\beta_i$ are user and item biases, $\gamma_u$ and $\gamma_i$ are the $K$-dimensional latent factors of user $u$ and item $i$, respectively, and "$\cdot$" denotes vector multiplication. Intuitively, $\gamma_u$ can be interpreted as the preference of user $u$ to some latent factors, while $\gamma_i$ is the property embedding of item $i$ on those latent factors. Based on a set of observed training records $\mathcal{R}$, the model is typically targeted with the goal of providing accurate rating predictions, where we determine the parameter set $\Theta = \{\alpha, \beta_u, \beta_i, \gamma_u, \gamma_i\}$ with the following minimization problem,

$$\Theta = \underset{\Theta}{argmin} \sum_{r_{u,i} \in \mathcal{R}} \big(rate(u,i) - r_{u,i}\big)^2 + \lambda \Omega(\Theta) \tag{2}$$

and $\Omega(\Theta)$ is a regularization term. A variety of methods exist to minimize Eq.(2), for example, Stochastic Gradient Descent (SGD) or Alternating Least Squares (ALS) [15]. However, this model merely takes into account the numerical ratings and leaves out the textual reviews, which is information-rich and may well help to provide better recommendations.

### 3.2 Latent Dirichlet Allocation (LDA)

Different from LFM, the LDA model attempts to learn a number of $K$ latent topics from documents (textual reviews in this work), where each word $w$ is assigned to a topic $z_w$, and each topic $z$ is associated with a word distribution $\phi_z$. Based on this, each document $d \in \mathcal{D}$ is represented with a $K$-dimensional topic distribution $\theta_d$, where the $j$-th word $w_{d,j}$ in document $d$ discusses its corresponding topic $z_{d,j}$ with probability $\theta_{d,z_{d,j}}$. It is usually convenient to also define the word distribution $\phi_{z,w}$, which is the probability that word $w$ is used for topic $z$ in the whole corpus $\mathcal{D}$. The final model conducts parameter learning by maximizing the likelihood of observing the whole $\mathcal{D}$:

$$P(\mathcal{D}|\theta, \phi, z) = \prod_{d \in \mathcal{D}} \prod_{j=1}^{L_d} \theta_{d,z_{d,j}} \phi_{z_{d,j},w_{d,j}} \tag{3}$$

where $L_d$ is the length (number of words) of document $d$. Intuitionally, we are multiplying the probability of seeing a particular topic in $\theta_d$ with the likelihood of seeing a particular word given the topic to estimate the likelihood of seeing the whole corpus.

### 3.3 Randomization Function

Let $\gamma \in \mathbb{R}^K$ be an arbitrary vector and $\theta \in [0,1]^K$ be a stochastic vector, where their dimensions are the same $K$ as the latent factors $\gamma_u, \gamma_i$ and latent topics $\theta_d$ in the previous subsections. According to the definition, we have $0 \leq \theta_k \leq 1$ and $\|\theta\|_1 = \sum_{k=1}^{K} \theta_k = 1$. The target of a randomization function $f : \mathbb{R}^K \to \mathbb{R}^K$ is to convert an arbitrary vector $\gamma$ to a probabilistic distribution $\theta = f(\gamma)$. The inherent nature of a randomization function is the key component to bridge the gap between LFM and LDA models, which links the latent factors $\gamma$ in LFM to the topic distributions $\theta$ in LDA,

and thus makes it possible to model the numerical ratings and textual reviews in a joint manner.

In the background of personalized recommendation, a desired randomization function is expected to be *monotonic* in the sense that it preserves the orderings, so that the largest value of $\gamma$ should also correspond to the largest value in $\theta$, thus the dimensions of the LFM model and the LDA model are inherently aligned during the model learning process to express the user-item relations in a shared feature space. As a result, the basic properties of a randomization function $f(\cdot)$ can be summarized as follows:

$$\begin{cases} 0 \leq f(\gamma)_i \leq 1, \|f(\gamma)\|_1 = 1 \\ \gamma_i < \gamma_j \rightarrow f(\gamma)_i < f(\gamma)_j \end{cases}, \forall \gamma \in \mathbb{R}^K, 1 \leq i, j \leq K \tag{4}$$

For example, in [21] the authors designed a randomization function as:

$$\theta_k = f(\gamma)_k = \frac{\exp(\kappa\gamma_k)}{\sum_{k'} \exp(\kappa\gamma_{k'})} \tag{5}$$

which conducts logistic normalization on a latent factor. In the following, we investigate the relationship between the objective functions of LFM and LDA, and further point out the properties required on a randomization function to harmonize the models when bridging the two different functions.

## 4 Bridging Factors and Topics

### 4.1 Probability of Item Recommendation

In the Latent Factor Model (LFM), a recommendation list is constructed in descending order of the predicted ratings $rate(u, i)$ for a given user, which means that an item $i$ with a higher rating prediction on user $u$ also gains a higher probability of being recommended $P(i|u)$. As a result,

$$P_{\text{LFM}}(i|u) \propto rate(u, i) \propto \gamma_u \cdot \gamma_i \tag{6}$$

where $\propto$ denotes a positive correlation, and we leave out the parameters $\alpha, \beta_u$ and $\beta_i$ because they are constants given a user and an item in the model learning process [15].

In Latent Dirichlet Allocation (LDA) for personalized recommendation, each user or item is represented by its corresponding set of textual reviews $d_u$ or $d_i$, and the underlying intuition models the topical correlation between them by estimating the potential review $d_{u,i}$ that a user may write on an item, based on the topical distributions $\theta_{d_u}$ and $\theta_{d_i}$. To simplify the notations, we use $u, i$, and $k$ to denote the user or item document representations $d_u, d_i$ and the $k$-th latent topic $z_k$ interchangeably, and we thus have:

$$P(k|u) = \theta_{u,k} \text{ and } P(k|i) = \theta_{i,k} \tag{7}$$

The LDA model conducts likelihood maximization on each observed review $d_{u,i}$ given the corresponding user $u$ and item $i$, and the embedded topical distribution represents the probability of observing each topic $k$, which is:

$$P(k|u, i) = \theta_{d,k} \tag{8}$$

LDA applies an indirect causal effect from users to items via latent topics [5], which means that user $u$ and item $i$ are conditionally independent given topic $k$, i.e., $P(u|i,k) = P(u|k)$, and this further gives us the following:

$$P(u,i,k) = \frac{P(u,k)P(i,k)}{P(k)} \tag{9}$$

By applying Eq.(9) to Eq.(8), we decompose the topical distribution of a review into the topical representations of the corresponding user and item:

$$\theta_{d,k} = P(k|u,i) = P(k|u)P(k|i)\frac{P(u)P(i)}{P(k)P(u,i)} \propto P(k|u)P(k|i) = \theta_{u,k}\theta_{i,k} \tag{10}$$

where $P(u), P(i)$ and $P(u,i)$ are constants in the LDA procedure, and the latent topics $z_k$ are identically independent from each other, giving us constant and equal valued $P(k)$'s over the $K$ topics. As a result, we have the following conditional recommendation probability for LDA models:

$$\begin{aligned} P_{\text{LDA}}(i|u) &= \sum_{k=1}^{K} P(k|u)P(i|k) = \sum_{k=1}^{K} P(k|u)P(k|i)\frac{P(i)}{P(k)} \\ &\propto \sum_{k=1}^{K} P(k|u)P(k|i) = \sum_{k=1}^{K} \theta_{u,k}\theta_{i,k} = \theta_u \cdot \theta_i \end{aligned} \tag{11}$$

and this result conforms with Eq.(10) in that, the probability of recommending an item given a user is positively correlated to the sum of topic probabilities that a user may textually review on an item.

### 4.2 Bridging the Objective Functions

According to the conditional item recommendation probabilities given a target user specified in Eq.(6) and Eq.(11) for LFM and LDA models, respectively, a favoured approach to bridge the two models to leverage the power of both ratings and reviews should harmonize their objective functions, so that a higher value of $\gamma_u \cdot \gamma_i$ also corresponds to a higher value in $\theta_u \cdot \theta_i$. More precisely, except for the monotonic property defined in Eq.(4) on a vector itself, the randomization function $f(\cdot)$ from $\gamma$ to $\theta$ is also required to be monotonic for vector multiplications:

$$\gamma_1 \cdot \gamma_2 < \gamma_3 \cdot \gamma_4 \rightarrow f(\gamma_1) \cdot f(\gamma_2) < f(\gamma_3) \cdot f(\gamma_4),\ \forall \gamma_1, \gamma_2, \gamma_3, \gamma_4 \tag{12}$$

In this way, the LFM and LDA components in a bridged objective function would not conflict with each other during the model learning process, because both of them increase/decease the recommendation probability $P(i|u)$ at the same time for each single iteration.

Previous work intuitionally assumes that a randomization function satisfying the vector-level monotonic property in Eq.(4) will also be monotonic on product-level as Eq.(12). Frequently used examples are the normalization-based randomization functions, which normalize the elements of $\gamma$ to construct $\theta$ so that they sum to one [21,20,2,32]. In [21] for example, a logistic normalization randomization function as in Eq.(5) is applied so as to minimize the following joint objective function to bridge the LFM and LDA models:

$$\mathcal{O} = \sum_{r_{u,i} \in \mathcal{R}} \underbrace{\big(rate(u,i) - r_{u,i}\big)^2}_{\text{LFM component}} - \lambda \underbrace{\mathcal{L}\big(\mathcal{D}|\theta,\phi,z\big)}_{\text{LDA component}} \tag{13}$$

where the LFM component still minimizes the error in predicted ratings, while the LDA component is the log-likelihood of the probability for the review corpus in Eq.(3).

Previous designations do seem intuitional and reasonable, and they indeed improve the performance of personalized recommendation in many cases. However, we would like to point out in this work that the vector-level monotonic property does not necessarily guarantee the monotonic property on a product-level. Actually, we prove that such a randomization function that satisfies both vector- and product-level monotonic properties does not exist at all, and the proof is omitted due to the page limit (details can be found in the supplementary file).

For this reason, forcing a vector-level randomization function on the latent factors to bridge the LFM and LDA models will result in a conflict between the two components during the procedure of objective optimization, i.e., while the LFM component gains a higher probability of item recommendation with a larger value of $\gamma_u \cdot \gamma_i$, the LDA component may reversely force a lower recommendation probability with $\theta_u \cdot \theta_i$ just because of the mathematical property of the randomization function, which is not favoured in model learning process. This further explains the observation that the prediction accuracy of Eq.(13) tends to fluctuate drastically during optimization, although the overall performance generally tends to increase along with the iterations.

### 4.3 Direct Product-Level Randomization

Despite that a randomization function with product-level monotonic property does not exist, we shall notice a simple fact that the de facto components that we need to consider so as to preserve the orderings of $P_{\text{LFM}}(i|u)$ and $P_{\text{LDA}}(i|u)$, are the final product of the latent factors or latent topics as a whole, i.e., $\gamma_u \cdot \gamma_i$ and $\theta_u \cdot \theta_i$, rather than each latent factor $\gamma$ to a latent topic distribution $\theta$ separately.

More precisely, what we really need in the LDA model of Eq.(3) is the topic distribution of each document $\theta_d$, where we have $\theta_{d,k} \propto \theta_{u,k}\theta_{i,k}$ by Eq.(10). As a result, we can apply a randomization function $f(\cdot)$ to the product of latent factors $\gamma_{u,k}\gamma_{i,k}$ directly, so as to obtain the product of latent topic distributions $\theta_{u,k}\theta_{i,k}$ as a whole, which is further positively correlated to $\theta_{d,k}$ that will finally be adopted by the LDA component for model learning.

A lot of normalization-based randomization functions guarantee the product-level monotonic property when applied to the product of latent factors directly. In this work, we adopt the logistic-normalization function to enforce $\theta_{u,k}\theta_{i,k}$ (and thus $\theta_{d,k}$) to be positive and sum to one:

$$\theta_{d,k} \propto \theta_{u,k}\theta_{i,k} = f(\gamma_{u,k}\gamma_{i,k}) \doteq \frac{\exp(\gamma_{u,k}\gamma_{i,k})}{\sum_{k'} \exp(\gamma_{u,k'}\gamma_{i,k'})} \tag{14}$$

which preserves the orderings of the dimensions from $\gamma_u \cdot \gamma_i$ to $\theta_u \cdot \theta_i$, and thus guarantees the positive correlation between $P_{\text{LFM}}(i|u)$ and $P_{\text{LDA}}(i|u)$ according to Eq.(6) and Eq.(11). Based on this direct product-level randomization, we are fortunately able to bridge the LFM and LDA models to leverage the power of ratings and reviews collectively, and meanwhile make the two components harmonize with each other for model learning, which improves both the performance and stability of personalized recommendation.

In the following, we describe our Joint Factorizational Topic (JFT) model, as well as its application in the practical scenario of (cross-city) restaurant recommendation.

# 5 The Model

## 5.1 Joint Factorizational Topic Model (JFT)

The basic Joint Factorizational Topic (JFT) model bridges the LFM component as Eq.(1) and the LDA component in Eq.(3) according to the product-level randomization. Specifically, let $\theta_{d,k} = \frac{\exp(\gamma_{u,k}\gamma_{i,k})}{\sum_{k'}\exp(\gamma_{u,k'}\gamma_{i,k'})}$ in Eq.(3), the JFT model attempts to optimize the following objective function:

$$F(\Theta, \phi, z) = \sum_{r_{u,i}\in\mathcal{R}} \big(rate(u,i) - r_{u,i}\big)^2 - \lambda_l \mathcal{L}(\mathcal{D}|\theta, \phi, z) + \lambda_p \Omega(\Theta) \tag{15}$$

where $\Theta = (\alpha, \beta_u, \beta_i, \gamma_u, \gamma_i)$ is the parameter set of the LFM component, $\mathcal{L}(\mathcal{D}|\theta, \phi, z)$ is the log-likelihood of the whole corpus whose document distributions $\theta$ come from the latent factors $\gamma$, and $\Omega(\Theta) = \sum_{u,i}(\beta_u^2 + \beta_i^2 + \|\gamma_u\|_2^2 + \|\gamma_i\|_2^2)$ is the $\ell_2$-norm regularizer for the latent parameters.

Intuitionally, the LDA component $\mathcal{L}(\cdot)$ serves as another regularization term besides the traditional $\ell_2$-norm regularizer $\Omega(\cdot)$ for numerical rating prediction, and we trade off between them two with $\lambda_l$ and $\lambda_p$, respectively. In this way, the JFT model attempts to minimize the error in rating prediction, and meanwhile maximizes the likelihood of observing the corresponding textual reviews.

Most importantly, the LFM and LDA components are designed to be consistent with each other by product-level randomization in model learning, in that a smaller prediction error in the LFM component functionally invokes a larger likelihood of observing the corresponding textual review, which makes the two components collaborate rather than violate with each other when optimizing the objective function.

## 5.2 Incorporate City Factors and Novelty-Seeking

The problem of cross-city recommendation finds its fundamental importance in many Location-Based Services (LBS) like Foursquare and Yelp, where it is usual for users to expect personalized recommendations from the application when he is travelling outside the hometown in a new city.

However, previous approaches for point-of-interest recommendation (especially for LFM and its variants) encounter serious cold-start problems [18] in the application scenario of cross-city recommendation, where we may have only a few or even none historical rating records of a user who is traveling in a new city, although he/she may have made quite a number of ratings in his/her home city.

Our basic JFT model helps to alleviate the cross-city cold-start problem by bridging the textual reviews with numerical ratings, because the topic embeddings that we learn from the reviews of a new restaurant, may be similar to the embeddings that we can learn for the restaurants that a user previously liked in his/her hometown, which may help to provide personalized cross-city recommendations.

Nevertheless, the assumption of recommending similar items may not always be true in different scenarios, although it is one of the most basic assumptions that inherently drives the intuition of most personalization models. This is because the preferences of a user travailing in a new city may well diverge from his/her historical preferences in hometown. For example, some users may prefer to try new flavours of local features when in a new city, while others may still like to keep to their previous favourites.

As a result, we further introduce the city factors into the basic JFT model so as to learn the variant of user preferences for cross-city recommendation. To do so, we first model the user-item rating as:

$$rate'(u,i) = \alpha + \beta_u + \beta_i + (1-\tau_u)(\gamma_u \cdot \gamma_i) + \tau_u(\gamma_i \cdot \gamma_c) \tag{16}$$

where $\gamma_u \cdot \gamma_i$ estimates the similarity between a user and a targeted item as with Eq.(1), while $\gamma_i \cdot \gamma_c$ models the similarity between the targeted item and its corresponding city, and the novel-seeking parameter $0 \leq \tau_u \leq 1$ indicates the degree that a user prefers to try local flavours.

The intuition here (which will later be verified in the experiments) lies in that, a user $u$ with a high preference of novelty-seeking $\tau_u$ would put more interest on those restaurants whose factor representations $\gamma_i$ are similar to that of the whole city $\gamma_c$ (i.e., local flavour), while a user who prefers flavours she previously liked would be attracted more by those restaurants with similar factors of herself $\gamma_u$. We leave out the consideration of $\gamma_u \cdot \gamma_c$ because the factors $\gamma_u$ and $\gamma_c$ would be fixed parameters in model learning and when making recommendation given a user $u$ and a targeted city $c$, as a result, this component would not make a difference in learning and recommendation procedures.

Correspondingly, we re-parameterize the topic distribution $\theta_d$ of each review document $d$ from user $u$ to item $i$ by product-level randomization of the item-city factors $\gamma_{i,k}\gamma_{c,k}$:

$$\theta'_{d,k} = \frac{\exp(\gamma_{i,k}\gamma_{c,k})}{\sum_{k'} \exp(\gamma_{i,k'}\gamma_{c,k'})} \tag{17}$$

where each city $c$ is similarly represented as the set of reviews $d_c$ corresponding to the restaurants located therein. In this way, we reformulate the likelihood of the review corpus with weighted geometric mean of the user-item randomization $\theta_d$ in Eq.(14) and item-city randomization $\theta'_d$ in Eq.(17):

$$P'(\mathcal{D}|\theta,\theta',\phi,z) = \prod_{d\in\mathcal{D}} \prod_{j=1}^{L_d} \left(\theta_{d,z_{d,j}}\right)^{1-\tau_u} \left(\theta'_{d,z_{d,j}}\right)^{\tau_u} \phi_{z_{d,j},w_{d,j}} \tag{18}$$

Based on this, our JFT model for cross-city recommendation attempts to minimize the following objective function:

$$F'(\Theta',\phi,z) = \sum\nolimits_{r_{u,i}\in\mathcal{R}} \left(rate'(u,i) - r_{u,i}\right)^2 - \lambda_l \mathcal{L}'(\mathcal{D}|\theta,\theta',\phi,z) + \lambda_p \Omega(\Theta') \tag{19}$$

Similar to the basic JFT model, $\Theta' = (\alpha, \beta_u, \beta_i, \gamma_u, \gamma_i, \gamma_c, \tau_u)$ is the parameter set of the LFM component, and $\mathcal{L}'(\mathcal{D}|\theta,\theta',\phi,z)$ is the log-likelihood of the corpus probability in Eq.(18).

### 5.3 Fitting the Model

We introduce the algorithm for model fitting in this subsection. For notational simplicity and also without loss of generality, we use $F(\Theta,\phi,z)$ to denote the objective function of both the basic and the cross-city JFT model, where the parameter set for the LFM component are $\Theta = (\alpha, \beta_u, \beta_i, \gamma_u, \gamma_i)$ and $\Theta = (\alpha, \beta_u, \beta_i, \gamma_u, \gamma_i, \gamma_c, \tau_u)$, respectively. The learning procedure for them are similar.

Typically, the LFM component (with $\ell_2$-regularizer) can be easily fit with gradient descent, while the log-likelihood LDA component is usually optimized with Gibbs

sampling. As our model jointly includes the two inherently heterogenous components, we construct a learning procedure that optimizes the two components alternatively:

$$Step1 : \{\Theta^t, \phi^t\} \leftarrow \underset{\Theta,\phi}{argmin}\, F(\Theta, \phi, z^{t-1}) \; by \; gradient \; descent$$
$$Step2 : Logistic \; normalization \; on \; each \; topic \; vector \; \phi_k^t$$
$$Step3 : Sample \; z_{d,j}^t \; with \; probability \; P(z_{d,j}^t = k) = \theta_{d,k}^t \phi_{k,w_{d,j}}^t$$

In the first step, we fix the topic $z$ of each word in each document, and further compute the gradient of each parameter in $\{\Theta, \phi\}$ while fixing the others. Based on these gradients, the parameters in $\{\Theta, \phi\}$ are updated one by one, where the step size for each parameter is determined by linear search.

Specifically, we should note that the parameter $\tau_u$ in the cross-city JFT model represents the probability that user $u$ attempts to try new flavours different from his/her historical preferences. As a result, we only adopt those review records in $\mathcal{D}$ that user $u$ visited a restaurant outside of his/her home city to construct document $d_u$ and to update parameter $\tau_u$ ($\tau_u$ is kept stable if $d_u = \emptyset$), while the other parameters are updated with all their corresponding reviews.

However, the gradient descent procedure would not guarantee the word distribution $\phi$ of latent topics to be stochastic vectors. As a result, we conduct logistic normalization for each topic $\phi_k$ $(1 \leq k \leq K)$ in the second step, where each dimension of $\phi_k$ is normalized as $\phi_{k,w} = \frac{\exp(\phi_{k,w})}{\sum_{w'} \exp(\phi_{k,w'})}$.

In the last step, we preserve the results from the previous steps, and update the topic assignment for each word in each document. Similar to LDA, which assigns each word to the $k$-th topic according to the likelihood of the word discussing topic $k$, we set $z_{d,j} = k$ with probability proportional to $\theta_{d,k}\phi_{k,w_{d,j}}$, where the indices pair $\{d, j\}$ denotes the $j$-th word of document $d$, $\theta_d$ is the the topic distribution of document $d$, and $\phi_k$ is the word distribution of topic $k$.

The major difference between LDA and the last step of our JFT model is that, the topic distributions $\theta_d$ are determined base on the product-level randomization from latent factors $\gamma_u, \gamma_i$ and $\gamma_c$ in our model, instead of sampling from a Dirichlet distribution in LDA. As a result, we only need to sample the topic assignments $z$ in each iteration of our JFT model. The probabilistic interpretation of our approach and its inherent relationship with the LFM component have been exposited in the previous sections.

Finally, these steps are repeated iteratively until convergence, i.e., the $\ell_2$-difference in $\Theta$ is sufficiently small between two consecutive iterations, or that an overfitting is observed in the validation set.

### 5.4 Top-N Recommendation

In this subsection, we further adapt our basic and cross-city JFT model to provide more practical personalized top-N recommendation lists beyond numerical rating predictions.

It is known that a good performance on rating prediction does not necessarily guarantee a satisfactory performance of top-N recommendation by ranking the items in descending order of the predicted ratings [6]. This is partly because of the contradiction between the goal of recommending items that users would potentially visit and the data (ratings) that we use for model training, i.e., users actually indeed visited the items in

**Algorithm 1** TOP-N RECOMMENDATION

Input: $\mathcal{R}, \mathcal{D}, N$ Recommendation list of length $N$

1: $\mathcal{R}^+ \leftarrow \mathcal{R}$ with all ratings reset to be 1
2: Initialize model parameters $\alpha, \beta, \gamma, \phi, z$ randomly
3: Initialize $\tau_u \leftarrow 0.5$ for all users, $t \leftarrow 0$ **while** Not Convergence **or** $t < T$
4: $t \leftarrow t + 1, \mathcal{R}^- \leftarrow \emptyset$ **for** $(u, i) \in \mathcal{R}^+$
5: Sample item $j$ from the same city of item $i$ randomly, where $(u, j) \notin \mathcal{R}^+$
6: $\mathcal{R}^- \leftarrow \mathcal{R}^- \cup (u, j)$ with rating 0
7: Update model by $Step1 \sim 3$ with $\{\mathcal{R}^+ \cup \mathcal{R}^-, \mathcal{D}\}$
8: Rank items in descending order of rating prediction
9: Top-N Recommended items for each user

the dataset, no matter what numerical ratings they eventually made on them. Intuitionally, a relatively low predicted rating does not necessarily mean that the user would not be attracted by the item at all, because of the many items with low ratings yet visited by the users.

As a result, we train our JFT model (and also the baseline approaches) for top-N recommendation in a different way from the task of rating prediction. Specifically, we feed the learning procedure with binary inputs, where the observed records in $\mathcal{R}$ are all treated as positive cases (rating=1), and the negative cases (rating=0) are sampled from the unobserved user-item pairs in a $1:1$ negative sampling manner. For clarity, we exposit the sampling, learning, and recommendation produce in Algorithm 5.3.

# 6 Experiments

## 6.1 Experimental Setup

We collected user reviews from a major restaurant review website Dianping.com in China, including 253,749 reviews from 32,529 users towards 8,026 restaurants located in 194 cities, where each user made 20 or more reviews, including intro- and cross-city cases. We set the home city of a user according to the registration information in his/her profile.

Of the 253,749 reviews in the whole corpus, 233,802 records fall into intro-city reviews, and the remaining 19,947 records are cross-city reviews, where the ratio between intro-city and cross-city records is 11.72. Each review in the corpus consists of an integer rating ranging from 1 to 5 stars and a piece of textual comment, where the user expresses his/her opinions on the corresponding restaurant. The average length of textual comments is 41.5 words. To feed the LDA component with high quality textual inputs, we conduct part-of-speech tagging and stop word removing for each review with the widely used Stanford NLP toolkit[3].

We initialize $\tau_u = 0.5$ for the cross-city JFT model, and the eventual value of $\tau_u$ for each user is automatically determined by the model learning process. After careful tuning with grid search, we set the hyper-parameters $\lambda_l = 0.01$ and $\lambda_p = 0.001$, and five-fold cross-validation was conducted in performance evaluation for all methods.

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`

### 6.2 Performance on Rating Prediction

In this section, we investigate the performance on rating prediction of our basic and cross-city JFT model, which are denoted as JFT and JFTC in the following. We also adopt the following baseline methods for performance comparison.

**LFM**: The basic LFM approach denoted in Eq.(2), which takes no advantage of the textual reviews.

**EFM**: The Explicit Factor Model presented in [34], which is the state-of-the-art recommendation approach based on textual reviews by phrase-level sentiment analysis.

**HFT**: The Hidden Factors and Topics model in [21], which also takes advantages of both LFM and LDA, but applies a vector-level randomization (Eq.(5)) on the latent factors.

We adopt Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for evaluation, and the results with the number of topics/factors $K = 10$ are shown in Table 1. The standard deviations in five-fold cross-validation for each method and metric are $\leq 0.002$.

**Table 1.** RMSE and MAE when $K = 10$. Standard deviations for each method are $\leq 0.002$.

| Method | LFM | EFM | HFT | JFT | JFTC |
|---|---|---|---|---|---|
| RMSE | 0.6688 | 0.6529 | 0.6532 | 0.6456 | **0.6386** |
| MAE | 0.5309 | 0.5283 | 0.5280 | 0.5213 | **0.5128** |

We find that all the other approaches gain better performance against LFM, which means that taking advantage of the textual reviews helps to make better rating predictions. Besides, our basic JFT model achieves better performance than both the EFM and HFT models. On considering that a major difference between our basic JFT model and HFT is the product- and vector-level randomization, this experimental result verifies our theoretical analysis to bridge the LFM and LDA components in Section 4. Finally, by incorporating user preferences in novelty-seeking in a cross-city scenario, our JFTC approach achieves the best performance.

To exhibit a clearer view of the performance on cross-city scenarios, we further take out the cross-city rating cases from the test set in each of the 5 folds, and conduct performance evaluation under different choices of the number of latent factors and topics $K$ from 10 through 100. Results for RMSE and MAE are shown in Figure 1. We see that our cross-city JFT approach outperforms all other baselines on all choices of topic numbers, which validates the superior performance when we consider the local features of a city and the user preference of novelty-seeking in cross-city scenarios, where it would be easy for other approaches to encounter the problem of cold-start.

### 6.3 Top-N Recommendation

In this subsection, we explore the performance of our approach in more practical top-N recommendation tasks. We adopt our JFTC for top-N recommendation algorithm with binary inputs and negative sampling described Section 5.4, and make comparison with the following baseline methods:

**WRMF**: Weighted Regularized Matrix Factorization described in [10], which is similar to LFM but applies weighted negative sampling to benefit top-N recommendations.

**BPRMF**: Bayesian Personalized Ranking (BPR) for MF presented in [24], which is the state-of-the-art algorithm for top-N recommendation based only on numerical ratings.

**HFT**: The original HFT method achieves poor top-N performance in our settings. As a result, we optimize the HFT method with the same binary inputs and negative sampling approach as in our model for fair comparison.

To evaluate, we randomly hold out 5 records for each user, and provide top-5 recommendation list for each user, as with most practical applications. We adopt the measures of Precision@5 and NDCG@5, where the latter takes the positions of recommended items into consideration, and the results are shown in Table 2.

**Table 2.** Prec@5 and NDCG@5 with $K = 10$. Standard deviations for each method are $\leq 0.0006$.

| Method | WRMF | BPRMF | HFT | JFTC |
|---|---|---|---|---|
| Precision@5 | 0.0060 | 0.0057 | 0.0065 | **0.0079** |
| NDCG@5 | 0.1616 | 0.1632 | 0.1653 | **0.1790** |

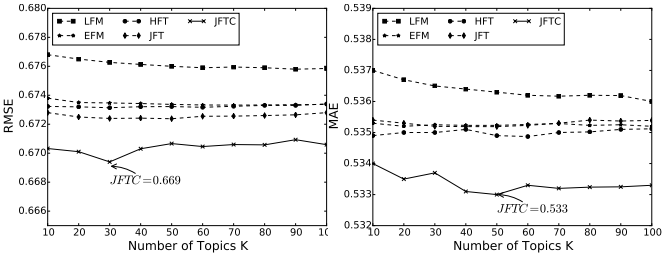
We see that both our JFTC approach and the HFT method (which make use of textual reviews) gain better performance than WRMF and BPRMF (which only make use of ratings). Further more, our JFTC method gains a 22% improvement against HFT in terms of precision, and 8.3% on NDCG, which is a superior achievement for practical applications.

Similar to the task of rating prediction, we also evaluate the top-N performance in cross-city settings. To do so, we select those user-city pairs that a user has at least 5 records in a city beyond his/her home city, and this results into 4,021 pairs corresponding to 1,739 users. We thus randomly hold out 5 records for a user in a corresponding city, and construct the recommendation list using the restaurants from that city, which gives us 4,021 lists in total for evaluation. We also conduct 5-fold cross-validation, and the standard deviations for both Precision and NDCG are $\leq 0.005$. Figure 2 shows the results against the number of latent factors/topics $K$.

We see that the performance of our JFTC model is better than the baselines on nearly all choices of $K$, except that the NDCG of HFT is slightly better when $K = 50$, which means that our model sometimes may not rank the right items to the top, though with a much better precision. However, our approach still beats the baselines for nearly all the cases. Interestingly, we find that the overall cross-city performance is a magnitude better than that on the whole dataset. This means that user behaviours can be more predictable in cross-city settings, where users do visit local attractions beyond their historical preferences. This further validates the underlying intuition of our novelty-seeking component in the JFTC model for cross-city recommendations.

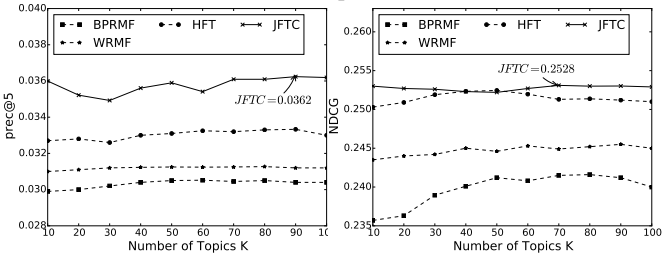
## 7 Conclusions

In this paper, we propose the Joint Factorizational Topic model that leverages the ratings and reviews in a collective manner for cross-city recommendation. For the first time, we examine the mathematical relationship between the LFM and LDA approaches for personalized recommendation, and we prove that vector-level randomization functions that




(a) RMSE vs Topic Number (b) MAE vs Topic Number

**Fig. 1.** RMSE and MAE vs the number of topics or latent factors $K$ in cross-city settings.




(a) Precision vs Topic Number (b) NDCG vs Topic Number

**Fig. 2.** Precision@5 and NDCG@5 vs the number of topics or latent factors $K$ in cross-city settings.

are multiplicatively monotonic actually do not exist at all, although they are frequently used to bridge the LFM and LDA components in previous work. Fortunately, we also find that a direct product-level randomization approach can be used to bridge the two components and harmonize their behavior for model learning. Extensive experimental results on case studies, rating prediction, top-N recommendation, and inner-model analysis verified both the intuitional reasonability, theoretical basis, and the quantitative performance of our approach.

## References

1. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed Recommender: Basing Recommendations on Consumer Product Reviews. Intelligent Systems 22(3), 39–47 (2007)
2. Agarwal, D., Chen, B.C.: fLDA: Matrix Factorization through Latent Dirichlet Allocation. WSDM (2010)
3. Baatarjav, E.A., Phithakkitnukoon, S., Dantu, R.: Group Recommendation System for Facebook. On the Move to Meaningful Internet Systems: OTM 2008 Workshops pp. 211–219 (2008)
4. Blei, D.M.: Probabilistic Topic Models. Communications of the ACM 55(4), 77–84 (2012)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. JMLR 2003(3), 993–1022 (2003)
6. Cremonesi, P., Koren, Y., Turrin, R.: Performance of Recommender Algorithms on Top-N Recommendation Tasks. RecSys pp. 39–46 (2010)
7. Das, A., Datar, M., Garg, A., Rajaram, S.: Google News Personalization: Scalable Online Collaborative Filtering. WWW pp. 271–280 (2007)
8. Davidson, J., Liebald, B., Liu, J., et al.: The YouTube Video Recommendation System. RecSys pp. 293–296 (2010)

9. Ganu, G., Elhadad, N., Marian, A.: Beyond the Stars: Improving Rating Predictions using Review Text Content. WebDB (2009)
10. Hu, Y., Koren, Y., Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. ICDM (2008)
11. Jakob, N., Weber, S.H., Mller, M.C., et al.: Beyond the Stars: Exploiting Free-Text User Reviews to Improve the Accuracy of Movie Recommendations. TSA (2009)
12. Knott, M., Bartholomew, D.: Latent Variable Models and Factor Analysis. Kendall's Library of Statistics 2 (1999)
13. Ko, M., Kim, H.W., Yi, M.Y., Song, J., Liu, Y.: MovieCommenter: Aspect-Based Collaborative Filtering by Utilizing User Comments. CollaborateCom (2011)
14. Konstan, J.A., Riedl, J.: Recommender Systems: from Algorithms to User Experience. User Modeling and User-Adapted Interaction 22(1-2), 101–123 (2012)
15. Koren, Y., Bell, R., Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer (2009)
16. Koren, Y., Bell, R.: Advances in Collaborative Filtering. Recommender Systems Handbook pp. 145–186 (2011)
17. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. Proc. NIPS (2001)
18. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the Cold Start Problem in Recommender Systems. Expert Systems with Applications 41(4), 2065–2073 (2014)
19. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-item Collaborative Filtering. Internet Computing 7(1), 76–80 (2003)
20. Ling, G., Lyu, M.R., King, I.: Ratings meet Reviews: a Combined Approach to Recommend. RecSys (2014)
21. McAuley, J., Leskovec, J.: Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. RecSys pp. 165–172 (2013)
22. Pazzani, M., Billsus, D.: Content-Based Recommendation Systems. The Adaptive Web LNCS pp. 325–341 (2007)
23. Purushotham, S., Liu, Y., Kuo, C.C.J.: Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems. ICML (2012)
24. Rendle, S., Freudenthaler, C., Gantner, Z., Thieme, L.S.: BPR: Bayesian Personalized Ranking from Implicit Feedback. UAI (2009)
25. Ricci, F., Rokach, L., Shapira, B.: Introduction to Recommender Systems Handbook. Springer US (2011)
26. Salakhutdinov, R., Mnih, A.: Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo. Proc. ICML (2008)
27. Salakhutdinov, R., Mnih, A.: Probabilistic Matrix Factorization. Proc. NIPS (2008)
28. Srebro, N., Rennie, J.D.M., Jaakkola, T.S.: Maximum-Margin Matrix Factorization. NIPS (2005)
29. Su, X., Khoshgoftaar, T.M.: A Survey of Collaborative Filtering Techniques. Advances in AI 4 (2009)
30. Takacs, G., Pilaszy, I., Nemeth, B., Tikk, D.: Investigation of Various Matrix Factorization Methods for Large Recommender Systems. Proc. ICDM (2008)
31. Terzi, M., Ferrario, M.A., Whittle, J.: Free text in user reviews: Their role in recommender systems. RecSys (2011)
32. Wang, C., Blei, D.M.: Collaborative Topic Modeling for Recommending Scientific Articles. KDD (2011)
33. Xu, X., Datta, A., Dutta, K.: Using Adjective Features from User Reviews to Generate Higher Quality and Explainable Recommendations. IFIP Advances in Info. and Com. Tech. 389, 18–34 (2012)
34. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. SIGIR (2014)
35. Zhang, Y., Zhang, H., Zhang, M., Liu, Y., et al.: Do Users Rate or Review? Boost Phrase-level Sentiment Labeling with Review-level Sentiment Classification. SIGIR (2014)