9-2017

Personalized microtopic recommendation on microblogs

Yang LI
Harbin Institute of Technology

Jing JIANG
Singapore Management University, jingjiang@smu.edu.sg

Ting LIU
Harbin Institute of Technology

Minghui QIU
Singapore Management University, minghui.qiu.2010@phdis.smu.edu.sg

Xiaofei SUN
Harbin Institute of Technology

DOI: https://doi.org/10.1145/2932192

Published in ACM Transactions on Intelligent Systems and Technology, 2017 September, Volume 8, Issue 6
http://doi.org/10.1145/2932192

Personalized Microtopic Recommendation on Microblogs

YANG LI, Harbin Institute of Technology
JING JIANG, Singapore Management University
TING LIU, Harbin Institute of Technology
MINGHUI QIU, Singapore Management University
XIAOFEI SUN, Harbin Institute of Technology

Microblogging services such as Sina Weibo and Twitter allow users to create tags explicitly indicated by the # symbol. In Sina Weibo, these tags are called *microtopics*, and in Twitter, they are called *hashtags*. In Sina Weibo, each microtopic has a designate page and can be directly visited or commented on. Recommending these microtopics to users based on their interests can help users efficiently acquire information. However, it is non-trivial to recommend microtopics to users to satisfy their information needs. In this article, we investigate the task of personalized microtopic recommendation, which exhibits two challenges. First, users usually do not give explicit ratings to microtopics. Second, there exists rich information about users and microtopics, for example, users' published content and biographical information, but it is not clear how to best utilize such information. To address the above two challenges, we propose a joint probabilistic latent factor model to integrate rich information into a matrix factorization-based solution to microtopic recommendation. Our model builds on top of collaborative filtering, content analysis, and feature regression. Using two real-world datasets, we evaluate our model with different kinds of content and contextual information. Experimental results show that our model significantly outperforms a few competitive baseline methods, especially in the circumstance where users have few adoption behaviors.

CCS Concepts: • **Information systems** → **Social networks**; **Personalization**; **Collaborative filtering**; *Social recommendation;*

Additional Key Words and Phrases: Microblogs, microtopic recommendation, topic model, collaborative filtering

ACM Reference Format:
Yang Li, Jing Jiang, Ting Liu, Minghui Qiu, and Xiaofei Sun. 2017. Personalized microtopic recommendation on microblogs. ACM Trans. Intell. Syst. Technol. 8, 6, Article 77 (August 2017), 21 pages.
DOI: http://dx.doi.org/10.1145/2932192

1. INTRODUCTION

Microblogging is a broadcasting medium that allows users to instantly post short messages on the Web to be shared with the public. Two microblogging services stand out among the many platforms in the world: Twitter and Sina Weibo. Twitter is the most popular service in most parts of the world. Sina Weibo, on the other hand, serves

Fig. 1. The main page of microtopic #冰桶挑战# (*Ice Bucket Challenge*) on Sina Weibo.

the majority of Chinese users. According to Sina Weibo's official statistics, by June 2014, Sina Weibo's daily active users and monthly active users reached 69.7 million and 156.5 million. This is of similar magnitude to Twitter, which has around 284 million monthly active users. Clearly, with such a large user base, there is a significant amount of information-seeking activities taking place in microblogging services [Arias et al. 2013].

Compared with other social media, microblogging offers a more diverse range of information-sharing mechanisms. In this article, we focus on a special mechanism, which we refer to as *microtopics*. The term *microtopic* is literally translated from its Chinese name 微话题. In Sina Weibo, a microtopic is represented as a word or phrase inside a pair of the # symbol. Each microtopic has its own designated page. Figure 1 shows the page of the microtopic #冰桶挑战# (*Ice Bucket Challenge*). A microtopic in Sina Weibo typically has a host, a short description, and rich attributes such as category and location. Users are encouraged to directly comment on the designated page of a microtopic, which can improve user experience and boost online interactions. In Twitter, we regard hashtags [Kwak et al. 2010] as microtopics, such as #iPhone6s. Twitter users can mention a hashtag when publishing a post. Although hashtags do not have explicit designated pages, we can treat all tweets mentioning the same hashtag collectively as the designated page for that hashtag. For the rest of this article, we use microtopics to refer to both the 微话题 (*microtopic*) in Sina Weibo and the hashtags in Twitter. Overall, microtopics differ from normal posts. In particular, in Sina Weibo, they are more like threads in discussion forums.

Microtopics cover a wide range of topics, including not only trending events such as #马航飞机失联# (*Malaysian Airlines flight missing*) and #世界杯# (*World Cup*) but also long-standing topics such as #深夜食堂# (*late night dining*) and #睡前阅读# (*bed-time reading*). Microtopics have been playing a very important role for users to better categorize and organize information by summarizing trending online topics. With the proliferation of microtopics, many users encounter the problem of information overload. It is important to help users easily browse microtopics and find those of their interests. For example, to a user who focuses on the new trends of IT technology, recommending

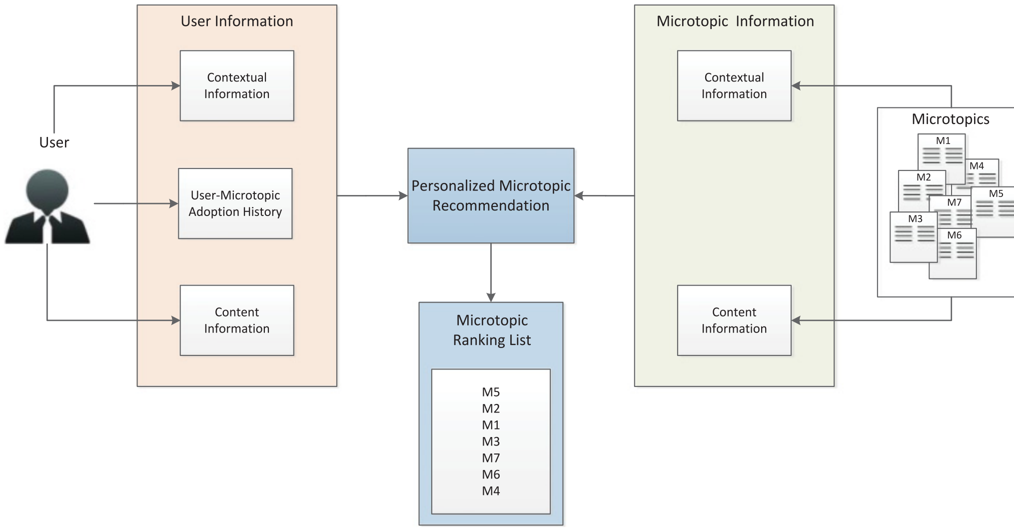

Fig. 2. The main framework of personalized microtopic recommendation.

microtopics such as #iPhone6s# is preferable than entertainment news that occurred during the same time period. On the other hand, it is also necessary to recommend generally interesting microtopics to fresh users who want to quickly catch the main idea of big current events or hot topics.

In this article, we investigate the task of personalized microtopic recommendation, which exhibits two challenges: (1) The users usually do not give explicit ratings to the microtopics, and (2) there exists rich information about users and microtopics, but it is not clear how to best make use of such information. While standard collaborative filtering-based methods [Koren et al. 2009; Mnih and Salakhutdinov 2007] can be directly applied to learn users' hidden interests via users' adoption history, they cannot easily deal with the users who have adopted very few microtopics. Moreover, these methods cannot take advantage of rich information on microblogs. For example, for each microtopic, we have comments on its designated page; for each user, we can also obtain her published posts. Such content information can presumably characterize properties of microtopics or indicate users' personal interests [Wang et al. 2014]. Furthermore, we observe that in microblog services, both users and microtopics have additional attributes such as the gender information of users and categories of microtopics. Similarly to texts, these types of contextual information can further connect similar users or similar microtopics. Intuitively, a joint model integrating all this rich information could help improve the recommendation performance. The main framework of personalized microtopic recommendation with rich information is illustrated in Figure 2.

However, even with the general framework, it is still not clear how to build an effective hybrid model for our problem. To this end, we propose a personalized microtopic recommendation model (MTRM) based on collaborative filtering and topic modeling, to seamlessly integrate user adoption behaviors, user microtopic content and contextual information. Unlike existing hybrid recommendation methods [Li et al. 2010; Fang and Si 2011], by deeply incorporating the content from users' published posts and comments on microtopics, our joint model gives interpretable representations of users and microtopics. By integrating both user and microtopic attributes, our model makes users or microtopics sharing the same attribute to have similar vectors in the latent factor space. Since a zero entry in the user-microtopic adoption matrix does not necessarily indicate that the user is not interested in the microtopic, we use a ranking optimization criterion to model users' preferences.

Our article makes the following contributions:

—We investigate the problem of personalized microtopic recommendation on two large real datasets (Sina Weibo and Twitter).
—We propose a novel probabilistic latent factor model effectively integrating user adoption behaviors, user microtopic content and contextual information.
—Through empirical evaluation, we find that both content and contextual information can help the recommendation task, and our model significantly outperforms the state-of-the-art methods.

2. MICROTOPIC RECOMMENDATION

In this section, we present our model for microtopic recommendation and explain the reasons behind the design of our model. We would like to consider several factors when designing our model. First, based on the idea of collaborative filtering, given a user and a new microtopic, to predict whether this user will be interested in this microtopic, we would like to make use of this user's as well as other users' historical records of microtopic adoption. Although a user's interest in a microtopic can be simply reflected by viewing the microtopic page, unfortunately such viewing history is not available. We then approach the problem by collecting user's implicit feedbacks, that is, we collect users who have published any posts on a microtopic page. We thus use these publishing records as indicators of users' interests in microtopics. Next, there are rich textual contents associated with both users and items. For a microtopic, we have the set of posts published on its microtopic page. For a user, similarly, we have her posts from her timeline. Furthermore, attributes such as a user's gender and location or a microtopic's category can presumably also be useful. Finally, we try to incorporate all this rich information into a ranking optimization criterion to infer users' preferences from the implicit feedbacks.

Our overall model is illustrated in Figure 3. The model mainly consists of three parts, namely, modeling user-microtopic adoptions, modeling user and microtopic content, and modeling user and microtopic attributes. In the rest of this section, we will present each of these three parts in detail. Table I shows the notation of the model. A basis of all three parts is that we assume there is a K-dimensional latent factor space. Each user and each microtopic is represented as a vector in this K-dimensional space (K is a hyperparameter to be pre-set). We use v_u to denote the vector for user u and v_i to denote the vector for microtopic i.

2.1. Modeling User-Microtopic Adoptions

The way we model microtopic adoption is similar to many existing latent factor models for recommendation. Given a user vector v_u and a microtopic vector v_i, we define an affinity score r_{ui} between user u and microtopic i as follows:

$$r_{ui} = v_u^\top v_i + b_u + b_i, \tag{1}$$

where r_{ui} models user u's preference to adopt microtopic i, and b_u and b_i are the user bias and the item bias to be learned.

Because microtopic recommendation belongs to the "one-class collaborative filtering problem" [Pan and Scholz 2009; Pan et al. 2008], where a zero entry in the adoption matrix indicates inaction rather than a negative rating, we adopt a ranking optimization criterion called Bayesian Personalized Ranking (BPR), which has been demonstrated

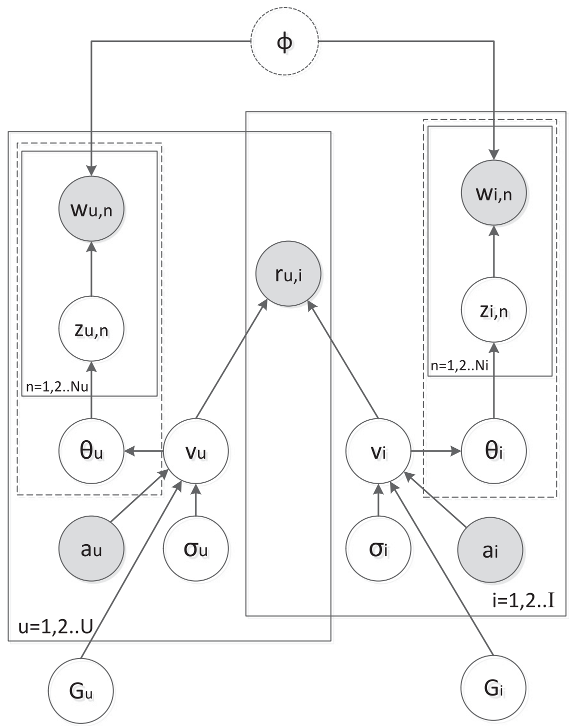

Fig. 3. Plate notation for our proposed microtopic recommendation model (MTRM). The dashed rectangles are optional parts. Hyperparameters are omitted for clarity.

Table I. Notation and Descriptions

	Description
K	total number of topics (latent factors)
V	total number of unique words
A_{U}	user attribute dimension
A_{I}	microtopic attribute dimension
r_{ui}	user u's preference score on microtopic i
θ_u	$\mathbb{R}^{K\times 1}$, user specific topic distribution
θ_i	$\mathbb{R}^{K\times 1}$, microtopic specific topic distribution
ϕ_k	$\mathbb{R}^{V\times 1}$, topic specific word distribution
v_u	$\mathbb{R}^{K\times 1}$, user vector
v_i	$\mathbb{R}^{K\times 1}$, microtopic vector
σ_u	$\mathbb{R}^{K\times 1}$, user's deviation vector
σ_i	$\mathbb{R}^{K\times 1}$, microtopic's deviation vector
a_u	$\mathbb{R}^{A_{\mathrm{U}}\times 1}$, user u's attributes
a_i	$\mathbb{R}^{A_{\mathrm{I}}\times 1}$, microtopic i's attributes
G_{U}	$\mathbb{R}^{A_{\mathrm{U}}\times K}$, regression coefficient matrix for users
G_{I}	$\mathbb{R}^{A_{\mathrm{I}}\times K}$, regression coefficient matrix for microtopics

effective in such tasks [Rendle et al. 2009; Shmueli et al. 2012]. In BPR, the goal is to rank items adopted by a user higher than items not adopted by her.

Without going into the details of the derivation, which can be found in Rendle et al. [2009], Equation (2) shows the objective function under the BPR criterion. Let r_{ui} denote the preference score computed from Equation (1). Let $\mathcal{P}$ denote a set of triplets $\langle u, i, j\rangle$ derived from the training data where user u has adopted microtopic i but not microtopic j. The BPR criterion tries to minimize the following function:

$$\min_{\Theta} \sum_{\langle u,i,j\rangle \in \mathcal{P}} \ln\left(1 + e^{-(r_{ui} - r_{uj})}\right), \tag{2}$$

where Θ denotes the set of model parameters, that is, the user and microtopic latent factor vectors and the bias terms. Here $\ln(1 + e^{-(r_{ui} - r_{uj})})$ can be considered the loss of ranking microtopic i higher than microtopic j for user u. We can see that the larger the value of $(r_{ui} - r_{uj})$, the smaller the loss. Thus, the objective function is trying to maximize the difference between r_{ui} and r_{uj} when we know that i has been adopted but j has not.

2.2. Modeling User and Microtopic Content

Previous studies have shown that hybrid approaches using both collaborative filtering and content-based filtering generally work better [Claypool et al. 1999; Fang and Si 2011; Hannon et al. 2010; Li et al. 2010; Tang et al. 2013; Wang and Blei 2011]. For our problem, we need to find suitable textual representations of users and microtopics to incorporate content into our model. For a user, we choose to use the latest 200 published posts. These posts should reflect this user's interests. For a microtopic, we opt to use the comments shown on its microtopic page. These are the posts containing the microtopic. They should reflect what this microtopic is about.

To incorporate the content into our model, we first combine a user's (or microtopic's) posts into a pseudo document. We then use Latent Dirichlet Allocation (LDA) to model the generation of these pseudo documents, as shown in Figure 3.[1] Inspired by recent work [McAuley and Leskovec 2013], we try to link LDA with collaborative filtering. Specifically, we assume that each of the K hidden factors that are used to represent users and microtopics has a corresponding multinomial word distribution, denoted by ϕ_k, that is, each hidden factor corresponds to a hidden topic in LDA. Each user (or microtopic) has a distribution over the K topics, which is derived from its hidden factor vector. Let θ_u denote user u's topic distribution. We have

$$\theta_{u,k} = \frac{\exp(\kappa v_{u,k})}{\sum_{k'} \exp(\kappa v_{u,k'})}. \tag{3}$$

Similarly, a microtopic i's topic distribution θ_i can be derived from v_i.

One may ask why we do not consider the practice of using v_u (or v_i) as parameters of a Dirichlet prior for the multinomial distribution parameterized by θ_u (or θ_i). This is because the parameters of a Dirichlet distribution have to be positive numbers, while here we do not place any constraint on v_u (or v_i). In other words, the values of v_u (or v_i) are real numbers that can be non-positive. Using the softmax function is also borrowed from the practice in McAuley and Leskovec [2013].

[1]Note that for Chinese texts, we can first segment them into words, which are more meaningful units than individual Chinese characters. Here we use the Chinese Word Segmentation (CWS) Tool provided by LTP-Cloud, a language technology platform for Chinese. http://www.ltp-cloud.com/.

Let w_u denote all the words in the pseudo document representing user u. Given the assumptions above, we can generate w_u using the following formula:

$$p(w_u \mid \theta_u, \phi) = \prod_n \sum_{k=1}^{K} \theta_{u,k}\phi_{k,w_{u,n}}. \tag{4}$$

The way the pseudo document for microtopic i is generated can be formulated similarly. With this, we can add the following term to the objective function (Equation (2)) that needs to be minimized:

$$-\left(\sum_u \ln p(w_u|\Theta) + \sum_i \ln p(w_i|\Theta)\right), \tag{5}$$

where Θ denotes all the model parameters.

2.3. Modeling User and Microtopic Attributes

Finally, we would like to incorporate additional attributes that characterize users and microtopics into our model. Intuitively, users or microtopics sharing the same attribute are likely to have similar vectors in the latent factor space. To this end, we consider a regression-based latent factorization method similar to the studies in Agarwal and Chen [2009] and Chen et al. [2012]. The idea is to embed a factor vector for each attribute value. Each user or microtopic is then profiled as an aggregation of the factor vectors of all its attributes.

Specifically, let a_u be an A_{U}-dimensional binary vector representing user u's attributes, where A_{U} is the total number of user attributes. $a_{u,t}$ is 1 if the attribute t is present in u and 0 otherwise. For example, one attribute may be *male* and another attribute may be *female*. We then model user latent factors v_u as follows:

$$v_u = G_{\mathrm{U}}^{\top} a_u + \sigma_u, \tag{6}$$

where $G_{\mathrm{U}} \in \mathbb{R}^{A_{\mathrm{U}} \times K}$ is a regression coefficient matrix and $\sigma_u \in \mathbb{R}^{K \times 1}$ is user u's deviation from the linear combination of the coefficients.

We profile each microtopic in a similar way. Let a_i, an A_{I}-dimensional binary vector, denote microtopic i's attributes. A_{I} refers to the number of microtopic attributes. We then model item vector v_i as

$$v_i = G_{\mathrm{I}}^{\top} a_i + \sigma_i, \tag{7}$$

where $G_{\mathrm{I}} \in \mathbb{R}^{A_{\mathrm{I}} \times K}$ is a regression coefficient matrix and $\sigma_i \in \mathbb{R}^{K \times 1}$ is an item-specific deviation. Note that we pose zero-mean Gaussian priors on G_{U}, G_{I}, σ_u, and σ_i.

2.4. Complete Model and Model Inference

We now present the complete model and model inference. In summary, we assume the following observations: For each user u, we observe a bag of words w_u and an attribute vector a_u. For each microtopic i, we also observe a bag of words w_i and an attribute vector a_i. We also have a set of triplets $\{\langle u, i, j\rangle\}$ indicating users' relative preferences between two microtopics. We have the following model parameters: For each latent factor (topic), there is a word distribution ϕ_k. For user attributes and microtopic attributes, there are two coefficient matrices G_{U} and G_{I}. Each user u has a user-specific latent factor vector σ_u and a bias term b_u. Similarly, each microtopic i also has a σ_i and a b_i. κ is the parameter which controls the transformation in Equation (3). We use Θ to denote all model parameters. We further use $R(\Theta)$ to denote a regularization function on Θ derived from the prior distributions of all the model parameters. Recall that all model

ALGORITHM 1: The Generative Process for Our Model

Draw two coefficient matrices, $G_{\mathrm{U}} \sim \mathcal{N}(0, \delta_g)$, and $G_{\mathrm{I}} \sim \mathcal{N}(0, \delta_{g'})$.
for each user u **do**
  Draw a user deviation vector $\sigma_u \sim \mathcal{N}(0, \delta_u)$,
  Set user latent factors $v_u = G_{\mathrm{U}}^{\top} a_u + \sigma_u$.
end for
for each microtopic i **do**
  Draw a microtopic deviation vector $\sigma_i \sim \mathcal{N}(0, \delta_i)$,
  Set microtopic latent factors $v_i = G_{\mathrm{I}}^{\top} a_i + \sigma_i$.
end for
for each rating $r_{u,i}$ **do**
  Draw $r_{ui} = v_u^{\top} v_i + b_u + b_i$.
end for
Set topic distributions $\theta_u \sim \mathrm{Softmax}(v_u)$, and $\theta_i \sim \mathrm{Softmax}(v_i)$ (Here $\mathrm{Softmax}(\cdot)$ is defined in Equation (3).)
Draw word distributions $\phi \sim \mathrm{Dir}(\beta)$.
for n-th word from user u **do**
  Draw $z_{u,n} \sim \mathrm{Multi}(\theta_u)$.
  Draw $w_{u,n} \sim \mathrm{Multi}(\phi_{z_{u,n}})$.
end for
for n-th word from microtopic i **do**
  Draw $z_{i,n} \sim \mathrm{Multi}(\theta_i)$.
  Draw $w_{i,n} \sim \mathrm{Multi}(\phi_{z_{i,n}})$.
end for

parameters have a zero-mean Gaussian prior except ϕ_k, which has a uniform Dirichlet prior. The generative story of our model is shown in Algorithm 1.

The overall objective function we try to minimize is thus defined as follows:

$$\min_{\Theta} \sum_{\langle u,i,j\rangle \in \mathcal{P}} \ln\left(1 + e^{-(r_{ui} - r_{uj})}\right) - \mu \left(\sum_u \ln p(w_u|\Theta) + \sum_i \ln p(w_i|\Theta) \right) + \lambda R(\Theta). \quad (8)$$

We can see that the objective function includes three parts. The first part is the ranking optimization, the second part is the log likelihood of generating the textual content, and the last part poses regularization on all the parameters. We use the commonly adopted L2-norm regularizer (i.e., the sum of the squared weight values). μ and λ are manually defined scalar values to balance the relative contributions of each part.

To learn the model parameters, we use Monte Carlo EM [Wallach 2006], an inference method that alternates between collapsed Gibbs sampling [Griffiths and Steyvers 2004] and gradient descent. It is similar to EM except that in the E-step we use sampling to get an estimation of the conditional probability needed for the M-step. So here in the E-step during the $(l+1)$th iteration, we fix all the parameters in Θ and use Gibbs sampling to obtain samples of the hidden variables Z (the set of all $z_{\cdot,\cdot}$). The samples of Z give us an estimation of the conditional probability $p(Z|W, \Theta^{(l)})$, where W is the set of all $w_{\cdot,\cdot}$ and $\Theta^{(l)}$ is the model parameters obtained in the lth iteration. Then in the M-step, we fix the latent topic labels Z and learn a new estimation of parameters $\Theta^{(l+1)}$ by maximizing the objective function.

2.4.1. E-step. In the E-step, we perform Gibbs sampling to learn the hidden variable $z_{u,n}$ by fixing all other parameters. In particular, we first compute θ_u from v_u. We then collapse out all the $\phi_{(\cdot)}$ and update each user u's nth topic label as follows:

$$p(z_{u,n} = x \mid Z_{u,\neg n}, W, \theta_u, \beta) \propto \theta_{u,x} \cdot \frac{n^x_{w_{u,n}} + \beta}{n^x_{\cdot} + V\beta}, \tag{9}$$

where $n^x_{w_{u,n}}$ is the number of times topic x is assigned to word $w_{u,n}$, excluding the current word $w_{u,n}$'s topic assignment. V refers to vocabulary size, and β is the parameter of the Dirichlet prior on the $\phi_{(\cdot)}$.

2.4.2. M-Step. In this step, we perform gradient descent to learn the parameters by fixing the values of topic labels. We reformulate the objective function $\mathcal{L}$ as

$$\mathcal{L} = \sum_{\langle u,i,j\rangle \in \mathcal{P}} \ln\left(1 + e^{-(r_{ui} - r_{uj})}\right) - \mu \left(\sum_{u,n} \ln p(w_{u,n}|\theta, \phi, z_{u,n}) + \sum_{i,n} \ln p(w_{i,n}|\theta, \phi, z_{i,n}) \right) + \lambda R(\Theta). \tag{10}$$

By computing the first derivatives of $\mathcal{L}$ with respect to the variables in Θ, we can then update them using gradient descent. We leave the details to the appendix.

3. EXPERIMENTS

In this section, we empirically evaluate the various components of our proposed model for microtopic recommendation. We conduct experiments to answer the following research questions: (i) How much can collaborative filtering help for microtopic recommendation compared with a popularity-based baseline? (ii) Does content help on top of collaborative filtering for this task, and if so, what content is the most useful? (iii) Does our method perform better than other baseline methods that also use a hybrid of collaborative filtering and content-based recommendation? (iv) Can user and microtopic attributes help the recommendation task and if so, which attributes are the most useful? (v) Does our method work well for cold-start users?

3.1. Data Set

For the evaluation, we use two datasets from Sina Weibo and Twitter.

3.1.1. Sina Weibo Dataset. Our first dataset was crawled from Sina Weibo, a popular Chinese microblogging service. We started by selecting 100 seed microtopics published within three months before November 1, 2014. We then crawled the users who had participated in these microtopics together with their comments published on the microtopics' pages. With the usernames of these users, we were able to collect all the other microtopics on which they had commented. With these additional microtopics, we could repeat the same process. We iteratively ran the process 3 times. All together, we got 22,194 users and 164,462 microtopics they adopt within three months before November 1, 2014. We then removed those microtopics that had fewer than 5 participates and inactive users (with fewer than 30 followers or fewer than 50 posts). Finally, we obtained 11,347 users, 13,188 microtopics, and 783,118 posting records of these users on these microtopics. For the crawled users, we also obtained their latest 200 published content, profile information including gender, status (verified or unverified user), and location. For the microtopics, we crawled their earliest 200 comments and category information.

3.1.2. Twitter Dataset. The second dataset was constructed from a Twitter dataset that spans the second half of 2009 [Yang and Leskovec 2011]. We first selected popular hashtags that had more than 100 participates between September and December 2009. In this way, we got 19,886 hashtags as candidate microtopics. Then we obtained the users who had participated in these microtopics together with their posts mentioning the microtopics. After removing inactive users and meaningless microtopics (e.g., #2!),

Table II. Statistics of Our Datasets

Dataset	Users	Microtopics	Adoption behaviors
Sina Weibo	11,347	13,188	783,118
Twitter	20,625	15,947	1,152,409

we finally obtained 20,625 users, 15,947 microtopics, and 1,152,409 posting records of these users on these microtopics. For each user, we crawled their published content from July 1, 2009, as user content. To get the content of a microtopic, we extracted all tweets that contain the microtopic from the original dataset and ranked them in chronological order. We used the earliest 200 tweets to represent the comments of a microtopic. Note that there is no user profile or microtopic category information in this dataset.

The statistics of our datasets are shown in Table II.

3.2. Experimental Settings

3.2.1. Baseline Methods. For comparison, we consider the following baseline methods:

—**PR:** Popularity ranking. For each user, we recommend microtopics to her simply based on popularity. Here we use the number of participants to measure the popularity of a microtopic.
—**PMF:** Probabilistic matrix factorization [Mnih and Salakhutdinov 2007]. The original model is designed for numerical ratings. For our task, we use 0s and 1s as rating scores.
—**BPR:** Bayesian personalized ranking matrix factorization [Rendle et al. 2009]. BPR differs from PMF in that it offers an optimization criterion based on BPR for personalized ranking, which we adopted for our method.
—**OCCF:** This is a method proposed in Li et al. [2010]. In this method, the content similarities between users and items are applied to weight the negative examples in One-Class Collaborative Filtering. Specifically, the weight for negative example between user u and item i is set to $(1-\mathrm{sim}(u,i))$ where $\mathrm{sim}(u,i)$ is the cosine similarity between term vectors representing u and i with TF-IDF weighting.
—**MCF:** Matrix Co-Factorization model proposed in Fang and Si [2011], which incorporates rich side information and implicit feedback. This method bears similarity to our method in that it also assumes that user and item latent factor vectors are associated with the content associated with a user or an item. The difference is that it is not a probabilistic model but is based on matrix factorization.

We refer to our proposed model as the MTRM (see Figure 3). Since we would like to empirically test the effectiveness of different sources of content, we first compare three degenerate versions of our model as follows. In all these three degenerate versions, no user or item attribute is incorporated yet.

—**MTRM-UC:** Our model incorporating users' posts as user content (i.e., a pseudo document for each user).
—**MTRM-IC:** Our model incorporating posts on microtopic pages as item content (i.e., a pseudo document for each microtopic).
—**MTRM-UCIC:** Our model incorporating both user's posts and posts on microtopic pages as content (i.e., a pseudo document for each user and a pseudo document for each microtopic).

Finally, as we will show in Section 3.3, using user content is much more effective than item content for our problem. We then test the performance of our full model with user content and user/item attributes:

—**MTRM-UC-ATTR:** Our model incorporating both user content and user/item attributes.

3.2.2. Experimental Setup. For both of our datasets, all the experimental settings are the same, unless otherwise noted. Similarly to the settings of many other studies on recommendation [Pan et al. 2008; Wang and Blei 2011], we hold out a percentage of the entries of the microtopic adoption matrix and use the remaining entries as training data. In particular, we perform fivefold cross validation. We first randomly divide the positive entries of the adoption matrix into five subsets. In each run, we use four subsets as training data and half of the fifth subset for validation to tune the hyperparameters. Then we use the other half of the fifth subset for testing. This is repeated 5 times and we report the average performance.

Recall that in BPR, we need to sample negative feedback to construct user preference data. For every user's each adopted microtopic in our training data, we randomly sample 5 microtopics that the user has not adopted as negative feedback. For other baseline methods, we use the same sampled data as negative instances for fair comparison. For evaluation, for each user in the test data, we randomly sample 1,000 microtopics that the user has not adopted and have not been used as negative feedback in training and then mix them with those that the user has adopted in the test data. In other words, we make sure there is no overlap of user-microtopic pairs between the training and the test data.

For our models, we perform 200 runs of Monte Carlo EM. In each run, we run 10 iterations for Gibbs sampling and another 10 iterations of gradient descent. For the parameter μ that is balancing the likelihood of textual content and the adoption errors, we found that in MTRM-UC and MTRM-IC, when μ is set to between 0.01 and 0.1, we can achieve good performance in both datasets. In MTRM-UCIC, when we incorporate both the user textual content and the microtopic content, we set the same $\mu = 0.01$ for both types of content. For all the zero-mean Gaussian priors in our model, we set the variances to be 0.01, and the regularization term λ is set to be 0.01 empirically. For MCF, we set the weight of negative instances to be 0.01 according to Pan and Scholz [2009] and Fang and Si [2011]. In OCCF, the weight is computed based on the content dissimilarity (Section 3.2.1).

We tested with latent factor size K ranging from 10 to 100 with a gap of 10. Finally, we found on Sina Weibo dataset, for the baseline methods PMF, BPR, and OCCF, $K = 20$ is an optimal setting. For MCF and our models, $K = 30$ is an optimal setting. While on Twitter dataset, $K = 30$ is an optimal setting for PMF, BPR, and OCCF. For MCF and our models, $K = 50$ is an optimal setting. A larger K cannot improve the results.

3.2.3. Evaluation Metrics. As we have pointed out in Section 2, we treat our microtopic recommendation problem as a ranking problem where for each user we would like to rank the microtopics based on how likely the user is going to browse and comment on them. Therefore, instead of looking at binary predictions and measuring prediction errors, we care more about the quality of the top-ranked microtopics. To this end, we choose two recall-based evaluation metrics, namely Mean Percentage Ranking (`MPR`) [Hu et al. 2008] and `Recall@M` [Wang and Blei 2011].

1. MPR: Assume for each user u we have a set of microtopics to rank. We use I_u^{test} to denote this set. After using an algorithm to rank the microtopics, let p_{ui} denote the percentile-ranking of microtopic i within I. For example, $p_{ui} = 0\%$ means microtopic i is predicted to be the most desirable for user u, thus preceding all other microtopics in the list. On the other hand, $p_{ui} = 100\%$ indicates that microtopic i is predicted to be the least preferred, thus placed at the end of the list. Our basic measurement of recommendation quality is the expected percentile ranking of user u's adopted microtopics in the test set, which is

$$\overline{p_u} = \frac{\sum_{i \in I_u^{\text{test}}} \hat{r}_{u,i}\, p_{ui}}{\sum_{i \in I_u^{\text{test}}} \hat{r}_{u,i}}, \tag{11}$$

where $\hat{r}_{u,i}$ is a binary number indicating the ground truth of whether user u has actually adopted microtopic i. The lower the value of $\overline{p_u}$, the better the quality of the ranking. Note that without any prior knowledge, if we randomly rank the microtopics in I_u^{test}, then the expected value of $\overline{p_u}$ will be 0.5.

2. Recall@M: Another metric we choose is `Recall@M`, which was used in Wang and Blei [2011]. The definition of `Recall@M` is as follows:

$$\texttt{Recall@M} = \frac{\text{number of microtopics adopted by user in top } M}{\text{total number of microtopics adopted by user}}. \tag{12}$$

Although this is not a typically used metric for ranked lists, the justification given by the authors of Wang and Blei [2011] is that in the ground-truth item adoption matrix, the zeros do not necessarily mean true negatives, because it could be either the user does not like the item or the user is not aware of the item. Therefore, we cannot accurately compute precision. Another way to understand `Recall@M` is that this metric is similar to `Precision@M`, but it is further normalized by the total number of items adopted by a user. Therefore, when we take the average of this metric across multiple users, those users who more actively adopt items will be weighted lower in the average `Recall@M`.

3.3. Collaborative Filtering with Rich Content

Since our baseline methods do not make use of user/item attribute, we first compare the baselines with the versions of our model that do not use attribute information either. In other words, we compare the baselines with MTRM-UC, MTRM-IC, and MTRM-UCIC. The goal here is threefold. First, we would like to see how much collaborative filtering can help the popularity-based baseline. Second, we would like to find out what content is useful for improving the recommendation results. Third, we would like to verify the performance of our method by comparing the degenerate models with other hybrid methods.

3.3.1. Results. In Table III and Table IV, we compare the results of our method and the baseline methods on Sina Weibo dataset and Twitter dataset, respectively. We use `MPR`, `Recall@10`, `Recall@50`, and `Recall@100` as the evaluation metrics. Note that for `MPR`, the lower the value, the better the results.

Results in Table III and Table IV show the following:

(i) PMF, the basic collaborative filtering method, clearly outperforms PR, the popularity-based method. The differences are quite substantial, showing that personalized recommendation of microtopics is very important.

(ii) OCCF achieves a similar result with PMF. Among PMF, BPR, and OCCF, BPR is giving consistent results in both `MPR` and `Recall`, showing that for our microtopic recommendation task, a ranking-based objective function gives more promising results than a rating based one.

(iii) MCF and the three degenerate versions of MTRM are able to improve the recommendation performance over OCCF and BPR by deeply incorporating user-generated

Table III. Performance Comparison on Sina Weibo Dataset

Metric	PR	PMF	BPR	OCCF	MCF	Degenerate Variations of MTRM		
						MTRM-IC	MTRM-UC	MTRM-UCIC
MPR	0.3381	0.1252$*$	0.1178$*$	0.1169	0.0984$*$	0.0945	**0.0822**$*$	0.0829
Recall@10	0.0908	0.1677$*$	0.1725	0.1412	0.2351$*$	0.2473$*$	0.2729$*$	**0.2830**
Recall@50	0.2091	0.4552$*$	0.4699	0.4294	0.5084$*$	0.4954	0.5227$*$	**0.5267**
Recall@100	0.3014	0.5997$*$	0.6077	0.5794	0.6460$*$	0.6318	0.6529$*$	**0.6590**

Note: $*$ indicates that the result is better than the method in the previous column at 5% significance level by Wilcoxon signed-rank test.

Table IV. Performance Comparison on Twitter Dataset

Metric	PR	PMF	BPR	OCCF	MCF	Degenerate Variations of MTRM		
						MTRM-IC	MTRM-UC	MTRM-UCIC
MPR	0.2914	0.1173$*$	0.1065$*$	0.1145	0.0876$*$	0.0755$*$	0.0640$*$	**0.0619**
Recall@10	0.1522	0.2178$*$	0.2685$*$	0.2068	0.3298$*$	0.3669$*$	0.3917$*$	**0.4029**
Recall@50	0.2592	0.6092$*$	0.6303$*$	0.6041	0.6694$*$	0.6873	0.7137$*$	**0.7186**
Recall@100	0.3423	0.7280$*$	0.7428$*$	0.7286	0.7765$*$	0.7936	0.8198$*$	**0.8232**

Note: $*$ indicates that the result is better than the method in the previous column at 5% significance level by Wilcoxon signed-rank test.

content into collaborative filtering. The results are consistent with previous findings [Fang and Si 2011; Hong et al. 2013; McAuley and Leskovec 2013].

(iv) Comparing with MCF, our models always perform better in terms of `MPR` and `Recall`, although MCF has incorporated both user content and the microtopic content through matrix co-factorization.

(v) Finally, we find that interestingly using pseudo documents for users is more effective than using pseudo documents for microtopics. We hypothesize that this is because the posts published on a microtopic's page are very diverse. In contrast, normal posts published by the same user may be more coherent and focused. Generally, we also find that the topics learned by MTRM-UC are more meaningful. The topics learned by MTRM-IC are a bit harder to interpret. When microtopics' pseudo documents are used on top of users' pseudo documents, the performance is very close to that of not adding them, especially on Sina Weibo dataset. Therefore, for the next experiment of using user/item attributes (Section 2.3), we use user content only.

3.3.2. Parameter Sensitivity Analysis. We would like to analyze how sensitive the performance of our model is with regard to the parameters.

First, we vary the value of μ while fixing the other parameters on both datasets. We show the results in terms of `MPR` for the three methods MTRM-IC, MTRM-UC, and MTRM-UCIC in Figure 4 and Figure 5. Recall that μ controls the relative importance of collaborative filtering and content in the objective function. We try the following values of μ: 0.001, 0.01, 0.1, 1, and 10. Since there are two types of content in MTRM-UCIC, we set $\mu_{UC} = \mu_{IC}$ for simplicity. The two figures show that the best results are achieved when μ is set to be between 0.01 and 0.1. The values of `MPR` increase when μ is larger than 1.

Figure 6 shows the `MPR` results when we vary the number of topics K from 10 to 50 on Sina Weibo dataset. We find that for all these three methods, the performance improves when K increases. The result of `MPR` become flattened when K reaches 30. In Figure 7, we find these three methods perform best when $K = 50$. The result of `MPR` does not change much when K is between 30 and 60.

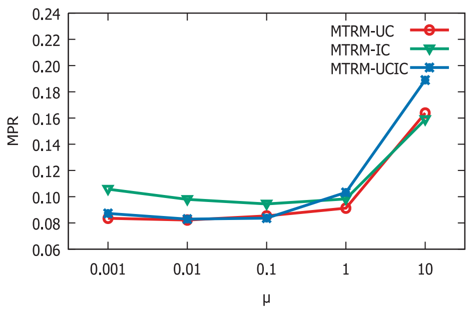

Fig. 4. MPR on Sina Weibo Dataset ($K = 30$).

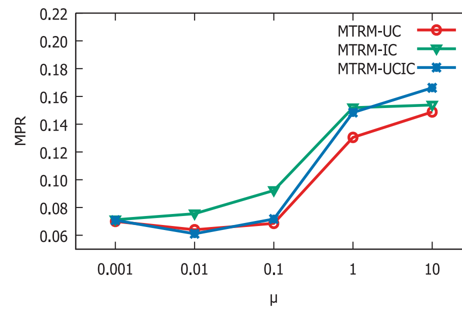

Fig. 5 MPR on Twitter Dataset ($K = 50$).

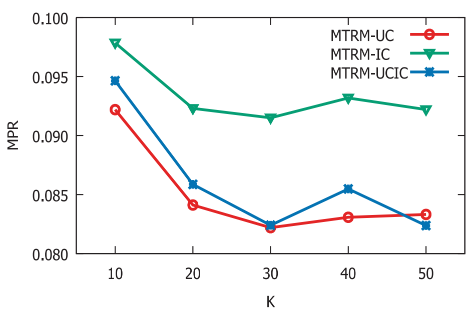

Fig. 6. MPR on Sina Weibo Dataset ($\mu = 0.01$).

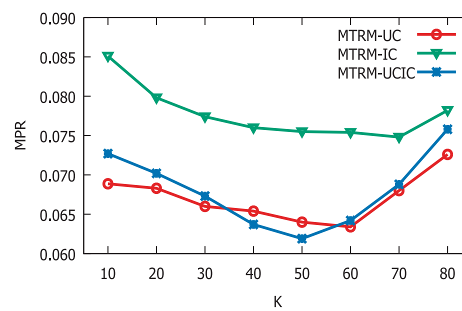

Fig. 7. MPR on Twitter Dataset ($\mu = 0.01$).

Through the overall results, we can see in most settings that MTRM-UC and MTRM-UCIC perform much better than MTRM-IC, and MTRM-UC is close to MTRM-UCIC, meaning that modeling user content is empirically better than modeling microtopic content in our task.

3.4. Integrating Attributes

In this section, we empirically study how much user and microtopic attributes may help improve the recommendation results. Recall that in our model, we assume the latent factor vector of a user or a microtopic is close to the linear combination of a set of coefficients corresponding to the attributes the user or microtopic has. In Table III, we found that compared with MTRM-UC, MTRM-UCIC improves recall slightly but gives a much lower MPR, which means integrating user posts as content can capture most of the textual information. Next, we will incorporate user or microtopic attributes on top of the MTRM-UC model. Note that there is no user or microtopic attribute information in our Twitter dataset. Therefore, we only use the Sina Weibo dataset in this part.

3.4.1. User Attributes. We have collected the following user attributes: (i) gender. Many users reveal their gender information in Sina Weibo. In total, of the 11,347 users, 10,822 have their gender information public. The number of males is 3,879 and the number of females is 6,943. (ii) status. In Sina Weibo, a user can be verified or unverified. Verified users are those who have been provided with a confirmation of identity by Sina Weibo. We obtained the status of 11,322 users, of that 1,380 are verified users. (iii) location. Many Sina Weibo users indicate the cities or provinces they come from. We were able to crawl the location information of 10,100 users.

Table V. Performance of Integrating Different Attributes on the Sina Weibo Dataset

Metric	MTRM-UC	+gender	+status	+location	+category	MTRM-UC-ATTR
MPR	0.0822	0.0809∗	0.0840	0.0817	0.0792∗	**0.0789**∗
Recall@10	0.2729	0.2774	0.2715	0.2773	0.2810∗	**0.2924**∗
Recall@50	0.5227	0.5419∗	0.5359	0.5376∗	0.5421∗	**0.5474**∗
Recall@100	0.6529	0.6723∗	0.6648	0.6679∗	0.6728∗	**0.6798**∗

Note: Comparison of performance before and after incorporating each type of user and microtipic attributes. MTRM-UC-ATTR refers to our model using user content and best setting of features (user gender, user location, and microtopic category). ∗ indicates that the result is better than the method in the first column (MTRM-UC) at 5% significance level by a Wilcoxon signed-rank test.

The results of incorporating users' attributes are shown in Table V, indicated by +gender, +status, and +location. We can see that adding gender information and location information turns out to be more useful in improving microtopic recommendation.

A close examination of the data gives some examples. For instance, #我和闺蜜的那些事# (*Stories between girlfriends and me*) is a microtopic for girls to share secrets they had with their girlfriends. Clearly, this microtopic is meant for female users mostly. Another microtopic, #养生美容知识# (*Knowledge of health and cosmetology*), talks about cosmetology, which is also a female-oriented micrtopic. We found that, indeed, very few male users would touch these microtopics.

The improvement after incorporating gender and location, however, is not very substantial. We found that this is because the majority of microtopics are not gender specific or location specific. Therefore, the benefit of incorporating gender or location is limited. As for the attribute of verification status, since only around 10% users are verified users, incorporating this attribute does not seem to be useful.

3.4.2. Microtopic Category. Sina Weibo organizes microtopics into 16 main categories, such as *Society*, *Celebrities*, and *Economics*. This helps users search for specific topics. We would like to test whether it also helps in the recommendation task. For example, if someone likes #我是歌手3# (*I Am a Singer*, a reality TV show in China), then she may also be interested in other microtopics under the same category "TV program" such as #中国梦之声# (*Chinese Idol*, another reality TV show). To verify this hypothesis, we try to incorporate the category information of microtopics into our model to see if people's participation behaviors have some fixed patterns on specific categories of microtopics.

In Table V, we find that, compared to MTRM-UC, the integration of category information (+category) gives a more than 3.5% decrease in `MPR` (lower `MPR` indicates better performance) and a 3% improvement in `Recall` relatively. Compared to all the user attributes, microtopic category information gives more of an improvement. Finally, if we combine the attributes from users and microtopics, then we find the best result we can achieve is to use user gender, user location, and microtopic category information (MTRM-UC-ATTR), as shown in Table V.

3.5. Cold Start Users

The previous experimental results show that collaborative filtering together with content modeling and user/microtopic attributes can largely improve the performance of popularity-based recommendations. The improvement mainly comes from collaborative filtering, as we can see when we compare PMF with PR. However, for cold-start users, that is, those who have not participated in many microtopics, collaborative filtering is less effective, because there is not much personal data of these users from which to learn. We hypothesize that for these cold-start users, the incorporation of content and attributes may be more important. To verify this hypothesis, we take those test users

Table VI. Statistics of Data in Cold Start Case

Dataset	Training Data		Test Data	
	Users	Adoptions	Users	Adoptions
Sina Weibo	11,347	626,494	36	107
Twitter	20,625	921,927	61	167

Table VII. Performance Comparison for Cold Start Users on Sina Weibo Dataset

Metric	PR	PMF	BPR	OCCF	MCF	MTRM-UC-ATTR
MPR	0.2879	0.2172$*$	0.1843$*$	0.2016	0.0957$*$	**0.0905**$*$
Recall@10	0.1717	0.0968	0.1830$*$	0.0909	**0.4358**$*$	0.4334
Recall@50	0.2880	0.3701$*$	0.4142$*$	0.3944	0.5873$*$	**0.6334**$*$
Recall@100	0.4067	0.5224$*$	0.5279	0.5570$*$	0.7010$*$	**0.7357**$*$

Note: $*$ means the result is better than the method in the previous column at 5% significance level by a Wilcoxon signed-rank test.

Table VIII. Performance Comparison for Cold Start Users on Twitter Dataset

Metric	PR	PMF	BPR	OCCF	MCF	MTRM-UCIC
MPR	0.2814	0.1547$*$	0.1188$*$	0.1293	0.0855$*$	**0.0599**$*$
Recall@10	0.1667	0.1492	0.2063$*$	0.1591	0.4238$*$	**0.4668**$*$
Recall@50	0.2745	0.5806$*$	0.6562$*$	0.5719	0.7042$*$	**0.7352**$*$
Recall@100	0.3723	0.6860$*$	0.7360$*$	0.7325	0.8027$*$	**0.8429**$*$

Note: $*$ means the result is better than the method in the previous column at 5% significance level by a Wilcoxon signed-rank test.

who have fewer than five adopted microtopics in the training data as cold-start users. The statistics of training and test data are shown in Table VI.

We compare the recommendation performance of these users between the various methods.

From Table VII and Table VIII, the following observations can be made:

(i) The absolute performance of the method based on popularity ranking (PR) performs better for cold-start users than that for all users (shown in Table III and Table IV). We believe that this is because for cold-start users, since they have not explored microtopics much, they are more likely to browse and participate in hot microtopics.

(ii) For cold-start users, although collaborative filtering methods (PMF and BPR) still helps in MPR, their relative improvement is small compared with for all users. The absolute performance of PMF and BPR is also lower than that for all users.

(iii) However, by deeply incorporating content and context information, MTRM-UC-ATTR achieves better performance comparing with all the baseline methods (Table VII). The relative improvement on the cold-start users is much higher than the relative improvement on all users.

(iv) Specifically, on the Twitter dataset (Table VIII), MTRM-UCIC still outperforms other methods although without any attribute information.

In general, both MCF and MTRM still perform well in cold-start cases, while PMF, BPR, and OCCF do not. We can understand this phenomenon in two ways. First, both MCF and MTRM have incorporated the contents of users and microtopics deeply into their objective functions, while PMF, BPR, and OCCF only consider the ratings. It indicates that the contents play an important role in cold-start cases where ratings are not available or scarce. It also verifies our basic motivation of using contents for microtopic recommendation in this work. Second, the results of MCF and MTRM on cold-start users appear to be slightly better than all users. However, considering that the number of cold-start users are very small, their averaged scores might not be directly

Table IX. Comparison of Topics Learned from User Content and Item Content

Model	Topical words
MTRM-UC	love, 男人*(men)*, 女人*(women)*, 生活*(livelihood)*, life, 人生*(human life)*, 爱情*(love)*, 幸福 *(happiness)*, 快乐*(happy)*, 生命(life).
	中国*(China)*, 文明*(civilization)*, 志愿者*(volunteer)*, 考试*(exam)*, 考研*(postgraduate exam)*, 招聘*(recruit)*, 面试*(interview)*, 公务员*(civil servants)*, 报名*(sign up)*, 事业单位*(institution)*.
MTRM-IC	鼓掌*(applaud)*, 爱你*(love you)*, 威武*(powerful)*, 加油*(come on)*, 萌*(cute)*, 开心*(happy)*, 期待 *(expect)*, 赞*(thumb-up)*, 愤怒*(angry)*, 气愤*(indignant)*.
	支持*(support)*, 投票*(vote)*, 分享*(share)*, 关注*(focus on)*, 粉丝*(followers)*, 活动*(activity)*, 互粉 *(follow each other)*, 青春*(youth)*, 签到*(sign in)*, 好运*(good luck)*.

comparable to that of all users. In fact, we have found that the standard deviations of the cold-start users are much larger than that of users overall. For example, in the Sina Weibo dataset, the standard deviation of `Recall@100` in cold-start users is 0.3807, while it is 0.2636 in all users.

The overall results show that the effect of incorporating user-generated content is more pronounced on cold-start users.

3.6. Discussions

In this section, we further analyze and discuss our results. First, we show some sampled topics learned by our model.

Table IX shows popular topics learned from user content (MTRM-UC) and item content (MTRM-IC) on Sina Weibo dataset. Due to the limited space, we only show the top-10 words within each topic. Generally, we find that the topics learned by MTRM-UC are more coherent and meaningful. While the topics learned by MTRM-IC are about attitudes or emotion. This is consistent with the finding that user content is better for microtopic recommendation than item content in Section 3.3.

Next, we conduct some error analysis on our results. For those users whose ranking results are very poor, we find that they have very sparse adoption records and it is indeed very hard to observe any pattern in their adoption behaviors. For example, one of the users has totally commented on four microtopics: #Halloween#, #iPhone6#, #豆瓣电影# (*Douban Movie*, a movie review site), and #奔跑吧兄弟# (*Running Man*, a live TV show). We can see that these four topics barely have any relations. We then checked this user's published posts and found that still his posts were not very relevant to the four microtopics he had commented on. Generally, if a user seldom participates in microtopics and does not publish many posts, or if a user publishes posts that are unrelated to his or her commented microtopics, then it is hard for our model to make good recommendations for this user.

4. RELATED WORK

In this section, we present two lines of work close to our task, namely, collaborative filtering and recommendation tasks on microblogs.

4.1. Collaborative Filtering

Recommendation methods can be classified as content-based recommendation [Balabanović and Shoham 1997; Hannon et al. 2010], collaborative filtering [Goldberg et al. 1992; Koren 2010; Koren et al. 2009], and hybrid approaches [Claypool et al. 1999; Schein et al. 2002]. Content-based approaches make predictions based on item similarity or the similarity between user and item profiles. These approaches need efforts to collect and extract knowledge from item or user content. Collaborative filtering (CF) methods, on the other hand, do not require user or item content. They analyze the relationship between users and items to identify new user-item associations. The latent factor model is one of the most successful CF models, in that users and items are profiled in a latent factor space of lower dimensionality. As the most representative latent factor model, matrix factorization (MF) has been successfully applied to various recommendation tasks [Koren et al. 2009; Lee and Seung 2001; Ma et al. 2008; Mnih and Salakhutdinov 2007; Tang et al. 2013]. Nevertheless, collaborative filtering suffers from the cold-start problem where few ratings (adoptions) can be obtained for a new item or user. Therefore, hybrid approaches combining content-based and CF-based methods are proposed to overcome their limitations, such as Matrix Factorization with Features [Li et al. 2010], Matrix Co-Factorization Models [Fang and Si 2011; Hong et al. 2013], and Regression-based Latent Factor Models [Agarwal and Chen 2009; Chen et al. 2012].

Recent studies seek to incorporate textual content to provide an interpretable latent structure for users and items [Diao et al. 2014; McAuley and Leskovec 2013; Wang and Blei 2011]. Wang and Blei [2011] first applied Latent Dirichlet Allocation method (LDA) [Blei et al. 2003] on item-specific textual content to recommend new scientific articles. The method profiles each item as a combination of its topic distribution and a latent vector. In sentiment analysis of product reviews, McAuley et al. [2013] and Diao [2014] assumed the topic distribution of each review is produced by the latent factors of the item. These methods could provide an interpretation to each latent factor, because factors and topics are in the same space. Our task can benefit from such approaches, but differently, we have textual content associated with both users and items (microtopics) in our task. Besides that, our model combines auxiliary information on microblogs as well.

Most of the above methods assume that users have explicit ratings for items, but, in reality, there are many cases where only implicit feedbacks are observed [Hu et al. 2008; Oard et al. 1998; Pan and Scholz 2009; Pan et al. 2008; Rendle et al. 2009]. In our task, since the users tend not to give explicit ratings to the microtopics, we make the first attempt to use a ranking optimization criterion together with topic modeling to infer users' preferences from user adoption behaviors.

4.2. Recommendation Tasks on Microblogs

With the popularity of microblogs, a growing number of studies have been proposed to profile users on microblogs to provide better recommendation services. There are three main recommendation tasks involved, namely followee recommendation, tweet recommendation, and hashtag recommendation. Twittomender recommends followees by exploiting a variety of recommendation strategies, including both content-based and collaborative filtering approaches [Hannon et al. 2010, 2011]. Abel et al. [2011a, 2011b] recommend external websites linked to Twitter by incorporating user profile modeling and temporal recency. Uysal and Croft [2011] present a learning-to-rank algorithm for tweet recommendation. Chen et al. [2012] proposed a regression-based tweet recommendation method by leveraging tweet topics, user social relations, and tweet features. Yan et al. [2012] presented a co-ranking framework for a tweet recommendation system that takes popularity, personalization, and diversity into account. Hashtag plays an important role in helping effectively organize and search tweets. Godin et al. [2013] applied topic models for Twitter hashtag recommendation. Ma et al. [2014] proposed two PLSA (Probabilistic Latent Semantic Analysis)-style topic models to incorporate user, time, hashtag, and tweet content for the task. Liu et al. [2012] assumed the content and hashtags of a document are talking about the same themes but written in different languages. Under the assumption, hashtag recommendation is modeled as a translation process from document content to hashtags. Similarly, Ding et al. [2012, 2013] proposed topical translation model for hashtag suggestion on Sina Weibo. Zhang et al. [2014] proposed a novel method that extends the translation based model and incorporates the temporal and personal factors. Most of the above work focuses on hashtag suggestion for a single tweet. Liang et al. [2012] proposed to recommend time-aware topics such as tag terms and keywords by incorporating implicit information network formed among users, topics, and microblogs.

We study the problem of microtopic recommendation at the user level. To better profile users and microtopics, we propose a joint probabilistic latent factor model to combine user adoption behaviors, user microtopic content, and contextual information.

5. CONCLUSIONS AND FUTURE WORK

In this article, we study the problem of personalized microtopic recommendation on microblogs. To utilize the rich information available, we proposed a joint probabilistic latent factor model to seamlessly integrate user adoption behaviors and user microtopic textual and contextual information. We design experiments to evaluate our model against several state-of-the-art models. By comparing with a popularity-based ranking method, we found that collaborative filtering significantly helped, indicating that it is important to personalize the ranked list of microtopics for individual users. Second, we found it beneficial to incorporate users' historical posts or microtopics' comments to help better learn users' latent factor vectors or microtopics' latent factor vectors. Last but not least, by incorporating both user and microtopic attributes, our model can further improve the recommendation performance. The overall experimental results show that our model outperforms the competitive baseline methods effectively, especially in the circumstance that users have few adoption behaviors.

There are a few directions we would like to explore in the future. First, social recommendation has been studied in recent years [Ma et al. 2008; Tang et al. 2013]. Presumably, a user often learns about a microtopic through other users whom he or she follows. We have not incorporated social relations between users such as following relations and mention relations, partly because such data are harder to crawl. Second, deep learning models recently have shown great potential for learning effective representations and deliver state-of-the-art performance in natural language processing applications [Salakhutdinov and Hinton 2009; Kalchbrenner et al. 2014]. Actually, some attempts have been made to develop deep learning models for collaborative filtering. For example, Wang et al. [2015] proposed a hierarchical Bayesian model called collaborative deep learning, which jointly performs deep representation learning for the content information and collaborative filtering for the ratings (feedback) matrix. By replacing the topic model with deep learning models such as Stacked Denoising Autoencoders (a feedforward neural network for learning representations of the input data by learning to predict the clean input itself in the output), we believe that our model can also learn an effective deep feature representation from content. All these issues will be left as our future works.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments.

REFERENCES

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011a. Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization*. Springer, 1–12.

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011b. Semantic enrichment of twitter posts for user profile construction on the social web. In *The Semanic Web: Research and Applications*. 375–389.

Deepak Agarwal and Bee-Chung Chen. 2009. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 19–28.

Marta Arias, Argimiro Arratia, and Ramon Xuriguera. 2013. Forecasting with twitter data. *ACM Trans. Intell. Syst. Technol.* 5, 1 (2013), 8.

Marko Balabanović and Yoav Shoham. 1997. Fab: Content-based, collaborative recommendation. *Commun. ACM* 40, 3 (1997), 66–72.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. 2012. Collaborative personalized tweet recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 661–670.

Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, Vol. 60. Citeseer.

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 *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 193–202.

Zhuoye Ding, Xipeng Qiu, Qi Zhang, and Xuanjing Huang. 2013. Learning topical translation model for microblog hashtag suggestion. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. AAAI Press, 2078–2084.

Zhuoye Ding, Qi Zhang, and Xuanjing Huang. 2012. Automatic hashtag recommendation for microblogs using topic-specific translation model. In *Proceedings of the 24th International Conference on Computational Linguistics*. Citeseer, 265.

Yi Fang and Luo Si. 2011. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. ACM, 65–69.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web Companion*. 593–596.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proc. Natl. Acad. Sci. U.S.A.* 101, suppl 1 (2004), 5228–5235.

John Hannon, Mike Bennett, and Barry Smyth. 2010. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the 4th ACM Conference on Recommender Systems*. ACM, 199–206.

John Hannon, Kevin McCarthy, and Barry Smyth. 2011. Finding useful users on twitter: Twittomender the followee recommender. In *Advances in Information Retrieval*. Springer, 784–787.

Liangjie Hong, Aziz S. Doumith, and Brian D. Davison. 2013. Co-factorization machines: Modeling user interests and predicting individual decisions in twitter. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. ACM, 557–566.

Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 263–272.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 655–665.

Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is twitter, A social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 591–600.

Daniel D. Lee and H. Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 556–562.

Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. 2010. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 959–968.

Huizhi Liang, Yue Xu, Dian Tjondronegoro, and Peter Christen. 2012. Time-aware topic recommendation based on micro-blogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, 1657–1661.

Zhiyuan Liu, Chen Liang, and Maosong Sun. 2012. Topical word trigger model for keyphrase extraction. In *COLING*. 1715–1730.

Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. ACM, 931–940.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2014. Tagging your tweets: A probabilistic modeling of hashtag annotation in twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 999–1008.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 165–172.

Andriy Mnih and Ruslan Salakhutdinov. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 1257–1264.

Douglas W. Oard, Jinmook Kim, and others. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems*. 81–83.

Rong Pan and Martin Scholz. 2009. Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 667–676.

Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Proceedings of the 8th IEEE International Conference on Data Mining, 2008 (ICDM'08)*. IEEE, 502–511.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *Int. J. Approx. Reason.* 50, 7 (2009), 969–978.

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 253–260.

Erez Shmueli, Amit Kagian, Yehuda Koren, and Ronny Lempel. 2012. Care to comment?: Recommendations for commenting on news stories. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 429–438.

Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: A review. *Soc. Netw. Anal. Min.* 3, 4 (2013), 1113–1133.

Ibrahim Uysal and W. Bruce Croft. 2011. User oriented tweet ranking: A filtering approach to microblogs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 2261–2264.

Hanna M. Wallach. 2006. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 977–984.

Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 448–456.

Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

Jinpeng Wang, Wayne Xin Zhao, Yulan He, and Xiaoming Li. 2014. Infer user interests via link structure regularization. *ACM Trans. Intell. Syst. Technol.* 5, 2 (2014), 23.

Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 516–525.

Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 177–186.

Qi Zhang, Yeyun Gong, Xuyang Sun, and Xuanjing Huang. 2014. Time-aware personalized hashtag recommendation on social media. In *COLING*. 203–212.