User Message Model: A New Approach to Scalable User Modeling on Microblog*

Quan Wang[1], Jun Xu[2,**], and Hang Li[2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
quanwang1012@gmail.com
[2] Noah's Ark Lab, Huawei Technologies, Hong Kong
nkxujun@gmail.com, hangli.hl@huawei.com

Abstract. Modeling users' topical interests on microblog is an important but challenging task. In this paper, we propose User Message Model (UMM), a hierarchical topic model specially designed for user modeling on microblog. In UMM, users and their messages are modeled by a hierarchy of topics. Thus, it has the ability to 1) deal with both the data sparseness and the topic diversity problems which previous methods suffer from, and 2) jointly model users and messages in a unified framework. Furthermore, UMM can be easily distributed to handle large-scale datasets. Experimental results on both Sina Weibo and Twitter datasets show that UMM can effectively model users' interests on microblog. It can achieve better results than previous methods in topic discovery and message recommendation. Experimental results on a large-scale Twitter dataset, containing about 2 million users and 50 million messages, further demonstrate the scalability and efficiency of distributed UMM.

Keywords: microblog, user modeling, topic modeling

1 Introduction

Microblogging systems such as Twitter and Sina Weibo[3] have become important communication and social networking tools. Recently, mining individual users' topical interests from their messages (tweets) attracts much attention. It has been demonstrated to be useful in many applications such as user clustering [10], friend recommendation [18], influential user detection [24], and user behavior prediction [2]. Various statistical topic modeling approaches have been applied to modeling users' interests on microblog [2, 18, 24, 26, 29]. However, it remains a non-trivial task with the following challenges.

1) *Data sparseness and topic diversity.* Microblog messages are short (restricted to 140 characters) and may not provide sufficient information. Therefore,

* This work was done when the first author visited the Noah's Ark Lab of Huawei Technologies.

** Jun Xu is currently affiliated with Institute of Computing Technology, Chinese Academy of Sciences.

[3] Sina Weibo (http://weibo.com) is a popular microblogging system in China.

taking each individual message as a short document and directly applying topic modeling approaches may not work well [10, 29]. That is, the data sparseness problem occurs. To tackle the problem, previous studies proposed to aggregate messages posted by each user into a "long" document and employ topic modeling approaches on aggregated documents [10, 18, 24]. However, such an aggregation strategy ignores the fact that topics discussed in different messages are usually different. Aggregating these topic-diverse messages into a single document and characterizing it with a unified topic distribution may be inaccurate. That is, the topic diversity problem occurs. We need to effectively deal with both problems.

2) *Joint modeling of users and messages.* In some applications (e.g., personalized message recommendation), not only users' topical interests but also messages' topic distributions need to be identified (e.g., to judge how much a user likes a message at semantic level). Therefore, modeling users and messages simultaneously is always preferred.

3) *Scalability and efficiency.* With the rapid growth of microblogging systems, more and more data is created every day. User modeling techniques which can efficiently handle large-scale datasets are sorely needed.

To address these challenges, we propose a novel user modeling approach, referred to as User Message Model (UMM). UMM is a hierarchical topic model in which users and their messages are modeled by a hierarchy of topics. Each user corresponds to a topic distribution, representing his/her topical interests. Each message posted by the user also corresponds to a topic distribution, with the user's topic distribution as the prior. Topics are represented as distributions over words. We further propose a distributed version of UMM which can efficiently handle large-scale datasets containing millions of users.

The advantages of UMM are as follows. 1) UMM can effectively deal with both the data sparseness problem and the topic diversity problem which previous methods suffer from. 2) UMM can jointly model users and messages in a unified framework. 3) UMM is easy to be implemented through distributed computing, and can efficiently handle large-scale datasets. To our knowledge, UMM is the first user modeling approach that can address all the challenges discussed above.

Experimental results on both Sina Weibo and Twitter datasets show that UMM can effectively model users' interests on microblog. It can achieve better results than previous methods in topic discovery and message recommendation. Experimental results on a large-scale Twitter dataset, containing about 2 million users and 50 million messages, demonstrate the efficiency and scalability of the distributed version of UMM.

2 Related Work

Mining users' topical interests from their messages (tweets) is a key problem in microblog analysis. A straightforward approach is to directly apply the Latent Dirichlet Allocation (LDA) [3] model on individual messages and simply represent each user by aggregating the topic distributions of his/her messages [23]. However, as messages on microblog are short, the data sparseness problem oc-

curs. To tackle this problem, previous studies proposed to aggregate messages by user and then employ the LDA model on aggregated messages (user-level LDA) [8, 10]. Hong and Davison empirically demonstrated that user-level LDA can achieve better performance in user and message classification [10]. The effectiveness of user-level LDA in influential user detection and friend recommendation was further demonstrated in [24] and [18]. Ahmed et al. later proposed a time-varying user-level LDA model to capture the dynamics of users' topical interests [2]. Recently, Xu et al. employed a slightly modified Author-Topic Model (ATM) [21] to discover user interests on Twitter [26, 27]. In fact, ATM is equivalent to user-level LDA when applied to microblog data [29]. Since different messages posted by the same user may discuss different topics, user-level LDA is plagued by the topic diversity problem. The proposed UMM can address both the data sparseness problem and the topic diversity problem.

Besides automatically discovered topics, users' interests can be represented in other forms, e.g., user specified tags [23, 25], ontology-based categories [16], and automatically extracted entities [1]. However, these methods rely on either external knowledge or data labeling, which is beyond the scope of this paper. There are also other studies on microblog topic modeling [5, 6, 12, 19, 20, 28], but they do not focus on identifying users' interests.

3 User Message Model

3.1 Model

Suppose that we are given a set of microblog data consisting of U users, and each user u has M^u messages. Each message m (posted by user u) is represented as a sequence of N_m^u words, denoted by $\boldsymbol{w}_m^u = \{w_{mn}^u : n = 1, \cdots, N_m^u\}$. Each word w_{mn}^u comes from a vocabulary $\mathcal{V}$ with size W.

User Message Model (UMM) is a hierarchical topic model that characterizes users and messages in a unified framework, based on the following assumptions. 1) There exist K topics and each topic ϕ_k is a multinomial distribution over the vocabulary. 2) The first layer of the hierarchy consists of the users. Each user u is associated with a multinomial distribution π^u over the topics, representing his/her interests. 3) The second layer consists of the messages. Each message m is also associated with a multinomial distribution θ_m^u over the topics. The message's topic distribution θ_m^u is controlled by the user's topic distribution π^u. 4) The third layer consists of the words. Each word in message m is generated according to θ_m^u. The graphical representation and the associated generative process are given in Figure 1 and Figure 2 respectively. Note that ϕ_k and π^u are sampled from symmetric Dirichlet distributions, while θ_m^u are sampled from an asymmetric Dirichlet distribution with parameter $\lambda^u \pi^u$. Here, π^u is a K-dimensional vector, denoting the topic distribution of user u; λ^u is a scalar, controlling how a message's topic distribution might vary from the user's topic distribution; $\lambda^u \pi^u$ means multiplying each dimension of π^u by λ^u.

UMM has similarity with Hierarchical Dirichlet Process (HDP) [22], but it differs from HDP. First and foremost, UMM fully exploits the user-message-

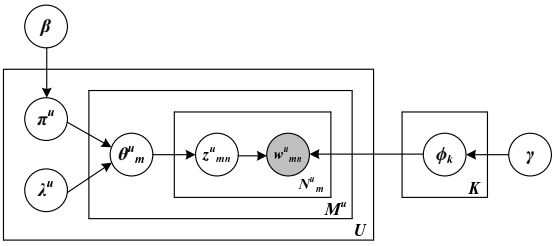

Fig. 1. Graphical representation of UMM.

1: for each topic $k = 1, \cdots, K$
2: draw word distribution $\phi_k|\gamma \sim \text{Dir}(\gamma)$
3: for each user $u = 1, \cdots, U$
4: draw topic distribution $\pi^u|\beta \sim \text{Dir}(\beta)$
5: for each message $m = 1, \cdots, M^u$ posted by the user
6: draw topic distribution $\theta^u_m|\pi^u, \lambda^u \sim \text{Dir}\left(\lambda^u \pi^u\right)$
7: for each word index $n = 1, \cdots, N^u_m$ in the message
8: draw a topic index $z^u_{mn}|\theta^u_m \sim \text{Mult}\left(\theta^u_m\right)$
9: draw a specific word $w^u_{mn}|z^u_{mn}, \phi_{1:K} \sim \text{Mult}\left(\phi_{z^u_{mn}}\right)$

Fig. 2. Generative process of UMM.

word hierarchy to perform better user modeling on microblog, particularly to address the data sparseness and topic diversity problems (as demonstrated in Section 3.3), while HDP is not specially designed for microblog data. In addition, UMM keeps a fixed number of topics, while the topic number in HDP is flexible.

3.2 Inference

We employ Gibbs sampling [9] to perform inference. Consider message m posted by user u. For the n-th word, the conditional posterior probability of its topic assignment z^u_{mn} can be calculated as:

$$\begin{aligned} & P\left(z^u_{mn} = z \middle| w^u_{mn} = w, \boldsymbol{w}^{-u}_{-mn}, \boldsymbol{z}^{-u}_{-mn}\right) \\ \propto & \left(\left(N^u_{z|m}\right)^{\neg z^u_{mn}} + \lambda^u \frac{\left(N_{z|u}\right)^{\neg z^u_{mn}} + \beta}{\left(N_{\cdot|u}\right)^{\neg z^u_{mn}} + K\beta}\right) \frac{\left(N_{w|z}\right)^{\neg z^u_{mn}} + \gamma}{\left(N_{\cdot|z}\right)^{\neg z^u_{mn}} + W\gamma}, \end{aligned} \tag{1}$$

where $\boldsymbol{w}^{-u}_{-mn}$ is the set of all observed words except w^u_{mn}; $\boldsymbol{z}^{-u}_{-mn}$ the set of all topic assignments except z^u_{mn}; $N^u_{z|m}$ the number of times a word in message m has been assigned to topic z; $N_{z|u}$ the number of times a word generated by user u (no matter which message it comes from) has been assigned to topic z, and $N_{\cdot|u} = \sum_z N_{z|u}$; $N_{w|z}$ the number of times word w has been assigned to topic z, and $N_{\cdot|z} = \sum_w N_{w|z}$; $(\cdot)^{\neg z^u_{mn}}$ the count that does not include the current assignment of z^u_{mn}. Figure 3 gives the pseudo code for a single Gibbs iteration.

1: for each user $u = 1, \cdots, U$
2: for each message $m = 1, \cdots, M^u$ posted by the user
3: for each word index $n = 1, \cdots, N^u_m$ in the message
4: $z_1 \leftarrow z^u_{mn}$; $w \leftarrow w^u_{mn}$
5: $N_{z_1|u} \leftarrow N_{z_1|u} - 1$; $N^u_{z_1|m} \leftarrow N^u_{z_1|m} - 1$; $N_{w|z_1} \leftarrow N_{w|z_1} - 1$
6: sampling $(z^u_{mn} \leftarrow z_2) \propto \left(N^u_{z_2|m} + \lambda^u \frac{N_{z_2|u}+\beta}{N_{\cdot|u}+K\beta}\right) \frac{N_{w|z_2}+\gamma}{N_{\cdot|z_2}+W\gamma}$
7: $N_{z_2|u} \leftarrow N_{z_2|u} + 1$; $N^u_{z_2|m} \leftarrow N^u_{z_2|m} + 1$; $N_{w|z_2} \leftarrow N_{w|z_2} + 1$

Fig. 3. One iteration of Gibbs sampling in UMM.

After obtaining the topic assignments and the counts, π^u, θ^u_m, and ϕ_z can be estimated as:

$$\pi^u_z = \frac{N_{z|u} + \beta}{N_{\cdot|u} + K\beta}, \quad \theta^u_{m,z} = \frac{N^u_{z|m} + \lambda^u \pi^u_z}{N^u_{\cdot|m} + \lambda^u}, \quad \phi_{z,w} = \frac{N_{w|z} + \gamma}{N_{\cdot|z} + W\gamma},$$

where π^u_z is the z-th dimension of π^u, $\theta^u_{m,z}$ the z-th dimension of θ^u_m, and $\phi_{z,w}$ the w-th dimension of ϕ_z.

3.3 Advantages

We compare UMM with Message Model (MM), User Model (UM), and Author-Topic Model (ATM), and demonstrate its advantages in mining users' interests on microblog. MM is a message-level LDA model, where each individual message is treated as a document [23]. As messages on microblog are short and may not provide sufficient information to train the LDA model, MM suffers from the data sparseness problem. UM is a user-level LDA model, where messages posted by the same user are aggregated into a single document [10, 18, 24]. As different messages posted by the same user may discuss different topics, UM suffers from the topic diversity problem. ATM was first proposed to model authors of scientific papers [21]. When applied to microblog, where each message (document) belongs to a single user (author), ATM is equivalent to UM [29] and also suffers from the topic diversity problem.

As opposed to the existing methods, UMM naturally models users and messages in a unified framework, and effectively deals with both the data sparseness and the topic diversity problems. Consider the Gibbs sampling procedure listed in Eq. (1). The first term expresses the probability of picking a specific topic in a message, and the second the probability of picking a specific word from the selected topic. To pick a topic, one can rely on information either from the current message or from the current user. In the former case, a topic is picked with probability proportional to the number of times the other words in the current message have been assigned to the topic, i.e., $\left(N^u_{z|m}\right)^{\neg z^u_{mn}}$. In the latter case, a topic is picked with probability proportional to the number of times the other words generated by the current user have been assigned to the topic, i.e., $\left(N_{z|u}\right)^{\neg z^u_{mn}} + \beta$. Parameter λ^u makes a tradeoff between the two cases. In this

Table 1. Complexities of MM, UM, UMM, and AD-UMM.

Method	Time Complexity	Space Complexity
MM	$\mathcal{N}KT$	$3\mathcal{N} + KW$
UM	$\mathcal{N}KT$	$2\mathcal{N} + KW + KU$
UMM	$\mathcal{N}KT$	$3\mathcal{N} + KW + KU$
AD-UMM	$(\frac{\mathcal{N}K}{P} + KW \log P)T$	$\frac{3\mathcal{N}+KU}{P} + KW$

way, UMM leverages the "specific but insufficient" message-level information and the "rich but diverse" user-level information. This is the key reason that UMM can effectively address both the topic diversity problem and the data sparseness problem which plague the existing methods.

Table 1 further compares the time and space complexities of MM, UM, and UMM. For time complexity, at each Gibbs iteration, all the three methods need to calculate the probability of picking each of the K topics for each word. Thus, they have the equal time complexity of $\mathcal{N}KT$, where $\mathcal{N} = \sum_{u=1}^{U} \sum_{m=1}^{M^u} N_m^u$ is the number of words in the whole collection, K the number of topics, and T the number of Gibbs iterations. For space complexity, all the three methods need to store the observed words $\{w_{mn}^u\}$, the corresponding topic assignments $\{z_{mn}^u\}$, and the word-topic counts $\left\{N_{w|z}\right\}$, for which the total space used is $2\mathcal{N} + KW$. In addition, the user-specific counts $\left\{N_{z|u}\right\}$ and/or the message-specific counts $\left\{N_{z|m}^u\right\}$ need to be stored. The space used for the former is KU, and for the latter can be reduced to $\mathcal{N}$ by caching the relatively small set of nonzero counts [9]. Thus, the space complexities of MM, UM, and UMM are $3\mathcal{N} + KW$, $2\mathcal{N} + KW + KU$, and $3\mathcal{N} + KW + KU$, respectively. We can see that UMM is comparable with MM and UM in terms of both time and space complexities.

3.4 Scaling up on Hadoop

To enhance the efficiency and scalability, we borrow the idea of AD-LDA [17] and design a distributed version of UMM, called Approximate Distributed UMM (AD-UMM). We implement AD-UMM on Hadoop[4], an open-source software framework that supports data-intensive distributed applications.

AD-UMM distributes the U users over P machines, with $U_p = \frac{U}{P}$ users and their messages on each machine. Specifically, let $\boldsymbol{w} = \{w_{mn}^u\}$ denote the set of words in the whole collection, and $\boldsymbol{z} = \{z_{mn}^u\}$ the set of corresponding topic assignments. We partition $\boldsymbol{w}$ into $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_P\}$ and $\boldsymbol{z}$ into $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_P\}$, and distribute them over the P machines, ensuring that messages posted by the same user are shuffled to the same machine. User-specific counts $\left\{N_{z|u}\right\}$ and message-specific counts $\left\{N_{z|m}^u\right\}$ are likewise partitioned and distributed. Topic-specific

[4] http://hadoop.apache.org/

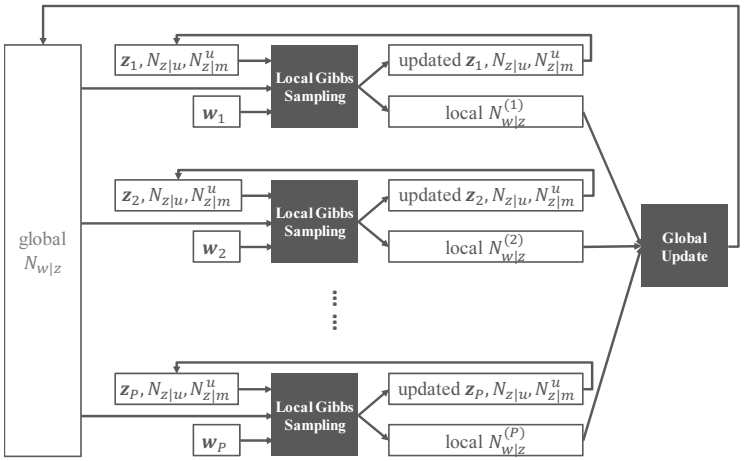

Fig. 4. One iteration of AD-UMM on Hadoop.

counts $\left\{N_{w|z}\right\}$ and $\left\{N_{\cdot|z}\right\}$ are broadcasted to all the machines. Each machine p maintains its own copy, denoted by $N^{(p)}_{w|z}$ and $N^{(p)}_{\cdot|z}$.

In each iteration, AD-UMM first conducts local Gibbs sampling on each machine independently, and then performs a global update across all the machines. During the local Gibbs sampling step on machine p, for each message m shuffled to the machine, the topic assignment of word w^u_{mn} is sampled according to:

$$P\left(z^u_{mn}=z\middle|w^u_{mn}=w,\boldsymbol{w}^{-u}_{-mn},\boldsymbol{z}_{p^{-u}_{-mn}}\right)$$
$$\propto\left(\left(N^u_{z|m}\right)^{\neg z^u_{mn}}+\lambda^u\frac{\left(N_{z|u}\right)^{\neg z^u_{mn}}+\beta}{\left(N_{\cdot|u}\right)^{\neg z^u_{mn}}+K\beta}\right)\frac{\left(N^{(p)}_{w|z}\right)^{\neg z^u_{mn}}+\gamma}{\left(N^{(p)}_{\cdot|z}\right)^{\neg z^u_{mn}}+W\gamma},$$

where $\boldsymbol{z}_{p^{-u}_{-mn}}=\boldsymbol{z}_p\setminus\{z^u_{mn}\}$. After machine p reassigns $\boldsymbol{z}_p$, $\left\{N_{z|u}\right\}$, $\left\{N^u_{z|m}\right\}$, and $\left\{N^{(p)}_{w|z}\right\}$ are updated. To merge back to a single set of word-topic counts $\left\{N_{w|z}\right\}$, a global update is performed across all the machines:

$$N_{w|z}\leftarrow N_{w|z}+\sum_{p=1}^{P}\left(N^{(p)}_{w|z}-N_{w|z}\right),\quad N^{(p)}_{w|z}\leftarrow N_{w|z}.$$

The count $N_{\cdot|z}$ is then updated by

$$N_{\cdot|z}\leftarrow\sum_w N_{w|z},\quad N^{(p)}_{\cdot|z}\leftarrow N_{\cdot|z}.$$

The whole procedure is shown in Figure 4.

Table 1 compares the time and space complexities of UMM and AD-UMM, where we have assumed that users and messages are almost evenly distributed.

As the total number of words in the collection (i.e., $\mathcal{N}$) is usually much larger than the vocabulary size (i.e., W), it is clear that AD-UMM outperforms UMM in terms of both time and space complexities.

4 Experiments

We have conducted three experiments. The first two tested the performance of UMM in topic discovery and message recommendation, and the third one tested the efficiency and scalability of AD-UMM.

4.1 Datasets

The first two experiments were conducted on two datasets: Weibo and Twitter. The Weibo dataset consists of 2,446 randomly sampled users and all the messages posted and re-posted by them in three months (Aug. 2012 – Oct. 2012). The messages are in Chinese. The Twitter dataset consists of 2,596 randomly sampled users and all the messages posted and re-posted by them in three months (Jul. 2009 – Sep. 2009). The messages are in English. For re-posted messages, only the original contents were retained. URLs, hash tags (#新浪微博#, #Twitter), and mentions (@用户, @User) were further removed. For the Weibo dataset, the messages were segmented with the Stanford Chinese Word Segmenter[5]. For both datasets, stop words and words whose frequencies in the whole dataset are less than 5 were removed. Messages which contain less than 5 words and users who have less than 10 messages were further removed.

We split each dataset into two parts according to the time stamps: messages in the first two months were used for topic discovery (denoted as "Weibo-I" and "Twitter-I") and messages in the third month were used for message recommendation (denoted as "Weibo-II" and "Twitter-II"). Since in the recommendation task messages were further filtered by a five-minute-window (as described in Section 4.3), Weibo-II and Twitter-II have much fewer users and messages.

The third experiment was conducted on a large-scale Twitter dataset (denoted as "Twitter-III"), consisting of about 2 million randomly sampled users and the messages posted and re-posted by them in three months (Jul. 2009 – Sep. 2009). Twitter-III was preprocessed in a similar way, and finally we got about 50 million messages. Table 2 gives some statistics of the datasets.

4.2 Topic Discovery

The first experiment tested the performance of UMM in topic discovery, and made comparison with UM and MM. In the methods, K was set to 100, γ (the Dirichlet prior on the topic-word distribution) was set to 0.01, and β (the Dirichlet prior on the user-topic/message-topic distribution) was set to $10/K$. In UMM, λ^u was set to 10 for all users.

[5] http://nlp.stanford.edu/software/segmenter.shtml

Table 2. Statistics of the datasets.

Dataset	# Users	# Messages	Vocabulary Size
Weibo-I	1,900	343,888	109,447
Twitter-I	1,929	1,055,613	75,990
Weibo-II	1,204	32,091	53,084
Twitter-II	721	9,324	15,049
Twitter-III	2,076,807	48,264,986	944,035

Table 3 shows the top-weighted topical interests of two randomly selected users on Weibo-I, generated by UMM, UM, and MM. The user biographies are also shown for evaluation. From the results, we can see that 1) The readability of the UMM topics is better than or equal to that of the UM and MM topics.[6] Almost all the UMM topics are readable, while some of the UM and MM topics are hard to understand. For example, in the first UM topic for the first user, the word " 人生(life)" is mixed with "经济(economy)" and "金融(commerce)". And in the first MM topic for the second user, the words "电影(movie)" and "导演(director)" are mixed with "音乐(music)" and "声音(voice)". 2) UMM characterizes users' interests better than UM and MM. The top interests of the users discovered by UMM are quite representative. However, for the first user, the top interests discovered by MM are "房地产(real estate)", "社会(society)", "信息安全(information security)", and "电子产品(electronic products)", where the last two seem less representative. And for the second user, the top interests discovered by UM are pretty vague and not so representative.

Table 4 further shows the top-weighted topics of two randomly selected messages generated by UMM on Weibo-I. The color of each word indicates the topic from which it is supposed to be generated. From the results, we can see that UMM can also effectively capture the topics discussed in microblog messages, and the topic assignments of the words are also reasonable. We have conducted the same experiments on Twitter-I and observed similar phenomena.

4.3 Message Recommendation

The second experiment tested the performance of UMM in message recommendation. We formalize the recommendation task as a Learning to Rank problem [15]. In training, a ranking model is constructed with the data consisting of users, messages, and labels (whether the users have re-posted, i.e., have shown interests in the messages). In ranking, given a user, a list of candidate messages are sorted by using the ranking model.

The data in Weibo-II and Twitter-II were transformed for the ranking task, consisting of user-message pairs and their labels. The label is positive if the message has been re-posted by the user, and is negative if the message might

[6] Topic readability refers to the coherence of top-weighted words in a topic.

Table 3. Top-weighted topics of users generated by UMM, UM, and MM on Weibo-I.

	User Bio	Top-weighted Topics			
UMM	知名地产商 (a real estate merchant)	房价(house price) 房地产(real estate) 土地(land) 北京(Beijing) 调控(control)	中国(China) 社会(society) 国家(country) 自由(freedom) 政治(politics)	人生(life) 学会(learn) 智慧(wisdom) 朋友(friends) 境界(realm)	经济(economy) 企业(enterprise) 市场(market) 增长(growth) 危机(crisis)
	中国最佳原创娱乐杂志 (best entertainment magazine in China)	电影(movie) 故事(story) 导演(director) 演员(actor) 明星(star)	设计(design) 时尚(fashion) 创意(creativity) 衣服(clothes) 颜色(color)	新闻(news) 媒体(media) 记者(journalist) 报道(news report) 网友(Internet users)	音乐(music) 声音(voice) 歌曲(song) 现场(live) 演唱会(concert)
UM	知名地产商 (a real estate merchant)	人生(life) 经济(economy) 金融(commerce) 企业家(entrepreneur) 企业(enterprise)	中国(China) 国家(country) 政府(government) 美国(America) 社会(society)	生活(life) 世界(world) 时间(time) 问题(problem) 社会(society)	房价(house price) 房地产(real estate) 北京(Beijing) 调控(control) 城市(city)
	中国最佳原创娱乐杂志 (best entertainment magazine in China)	电影(movie) 网友(Internet users) 媒体(media) 曝光(exposure) 日前(a few days ago)	生活(life) 世界(world) 时间(time) 问题(problem) 社会(society)	活动(activity) 中国(China) 北京(Beijing) 时间(time) 支持(support)	女人(woman) 喜欢(like) 男人(man) 人生(life) 幸福(happiness)
MM	知名地产商 (a real estate merchant)	房价(house price) 房地产(real estate) 房子(house) 北京(Beijing) 土地(land)	社会(society) 中国(China) 改革(reform) 自由(freedom) 政治(politics)	信息(information) 用户(users) 安全(security) 网站(website) 密码(password)	苹果(Apple) iPhone iPad 电脑(computer) 产品(product)
	中国最佳原创娱乐杂志 (best entertainment magazine in China)	电影(movie) 音乐(music) 声音(voice) 歌曲(song) 导演(director)	衣服(clothes) 时尚(fashion) 颜色(color) 头发(hair) 漂亮(pretty)	男人(man) 女人(woman) 结婚(marry) 女性(female) 爱情(love)	新闻(news) 媒体(media) 记者(journalist) 报道(news report) 杂志(magazine)

Table 4. Top-weighted topics of messages generated by UMM on Weibo-I.

Message	Top-weighted Topics			
北京市(Beijing)财政(financial)收入(income)增长 (growth)是GDP增长(growth)的三倍多,是城镇(cities and towns)居民(residents)收入(income)增长(growth)的近四倍.	美元 (dollar) 增长 (growth) 利润 (profit) 收入 (income) 营收 (revenue)	经济 (economy) 企业 (enterprise) 市场 (market) 增长 (growth) 危机 (crisis)	政府 (government) 国家 (country) 部门 (department) 政策 (policy) 管理 (management)	房价 (house price) 房地产 (real estate) 土地 (land) 北京 (Beijing) 调控 (control)
记者(journalist)与真相(truth)讲的不仅仅是故事(story), 还有良心(conscience)与责任(responsibility),也是对权力(authority)的监督(supervise).	中国 (China) 社会 (society) 国家 (country) 自由 (freedom) 政治 (politics)	政府 (government) 国家 (country) 部门 (department) 政策 (policy) 管理 (management)	新闻 (news) 媒体 (media) 记者 (journalist) 报道 (news report) 网友 (Internet users)	文学 (literature) 莫言 (Mo Yan) 诺贝尔 (Nobel) 作家 (writer) 小说 (novel)

have been seen by the user but has not been re-posted by him.[7] We randomly split each dataset into 5 parts by user and conducted 5-fold cross-validation.

Table 5 lists the features used in the ranking model. The seven basic features are suggested in [7, 4, 11]. To calculate the two term matching features, messages, user's historical posts, and their profile descriptions are represented as term frequency vectors. The topic matching features are calculated by UMM, UM, and MM models trained on Weibo-I and Twitter-I. Given user u and mes-

[7] Messages posted within 5 minutes after a re-posted message are assumed to be seen by the user.

Table 5. Features used for message recommendation.

Feature	Description
URL	Whether the message contains URLs
Hash tag	Whether the message contains hash tags
Length	Number of words in the message
Verified publisher	Whether the author of the message is a verified account
Follower/Followee ratio	Logarithm ratio of #follower and #followee of the author
Mention	Whether the message mentions the user
Historical forwarding	Times the user forwarded the author's posts in the past
Historical post relevance	Cosine similarity between the message and the user's posts
User profile relevance	Cosine similarity between the message and the user's profile
MM score	Topic matching score based on MM
UM score	Topic matching score based on UM
UMM score	Topic matching score based on UMM

sage m, the topic matching score is calculated as the dot product of their topic representations: $s(u, m) = \langle \pi^u, \theta^m \rangle$. We retain top 5 topics in π^u and θ^m, and truncate other topics. As UM/MM cannot directly output topic representations for messages/users, we calculate them using the learned topic assignments.

When training topic models, we set $K = 10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000$. The other parameters were set in the same way as in Section 4.2. We employed Ranking SVM [14] to train the ranking model. Parameter c was set in $[0, 2]$ with interval of 0.1, and the other parameters were set to default values. We tested the settings of using the basic features only (denoted as "Basic"), the basic features plus the term matching features (denoted as "Basic+Term"), and the basic features, the term matching features, and one of the topic matching features (denoted as "Basic+Term+UMM" for example). For evaluation, we employed a standard information retrieval metric of NDCG [13].

Table 6 reports the recommendation accuracies on Weibo-II and Twitter-II. The results indicate that 1) Topic matching features are useful in message recommendation. They can significantly (t-test, p-value < 0.05) improve the accuracies achieved by using only the basic and term matching features. 2) UMM performs the best among the three topic models. The improvements of UMM over MM and UM are statistically significant on Weibo-II (t-test, p-value < 0.05). 3) Content features (term matching and topic matching features) are more useful on Weibo-II than on Twitter-II, because more contents can be written in Chinese than in English with limited number of characters.

4.4 Scalability of AD-UMM

We first compared the efficiency of AD-UMM and UMM on Twitter-I. We built a 10-machine mini Hadoop cluster, each of which has a 2-core 2.5GHZ CPU and 2GB memory. In the cluster, 9 machines were used for distributed computing and 1 for scheduling and monitoring. AD-UMM was implemented on the Hadoop cluster, while UMM was implemented on a single machine. In UMM,

Table 6. Recommendation accuracies on Weibo-II and Twitter-II.

Method	Weibo-II			Twitter-II		
	NDCG@1	NDCG@3	NDCG@10	NDCG@1	NDCG@3	NDCG@10
Basic	0.5540	0.5668	0.6164	0.6962	0.7171	0.7661
Basic+Term	0.6412	0.6416	0.6828	0.7157	0.7377	0.7882
Basic+Term+MM	0.6860	0.6669	0.7037	0.7296	0.7439	0.7932
Basic+Term+UM	0.6736	0.6614	0.7010	0.7254	0.7405	0.7925
Basic+Term+UMM	**0.7143**	**0.6818**	**0.7164**	**0.7338**	**0.7440**	**0.7942**

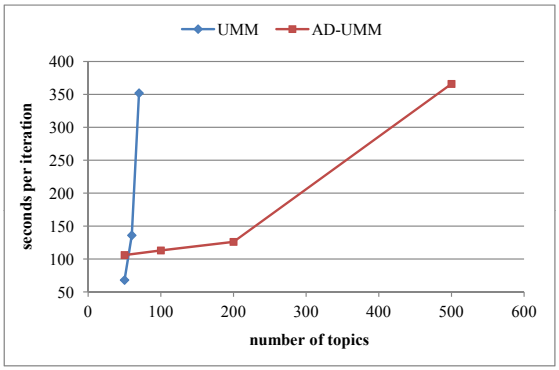

Fig. 5. Execution time of UMM and AD-UMM on Twitter-I.

with the limited 2GB memory, we set K in $\{50, 60, 70\}$. In AD-UMM, we set K in $\{50, 100, 200, 500\}$. The other parameters were set in the same way as in Section 4.2. Figure 5 reports the average execution time per iteration (sec.) of UMM and AD-UMM on Twitter-I. The results indicate that AD-UMM is much more efficient than UMM, particularly when the number of topics gets large.

We further tested the scalability of AD-UMM on Twitter-III. Figure 6 shows the average execution time per iteration (min.) of AD-UMM when K (number of topics) equals 500, with P (number of machines) varying from 4 to 9. Figure 7 shows the execution time when P equals 9, with K varying in $\{500, 1000, 2000, 5000\}$. Here, "Local Gibbs Sampling" and "Global Update" refer to the time costed in the local Gibbs sampling and global update steps respectively, and "Total" means the total time. The results indicate that 1) The execution time decreases linearly as the number of machines increases. 2) The execution time increases linearly as the number of topics increases. As a result, it is practical for AD-UMM to handle huge number of users, messages, and topics with an appropriate number of machines.

5 Conclusions

We have proposed a new approach to mining users' interests on microblog, called User Message Model (UMM). UMM works better than the existing methods,

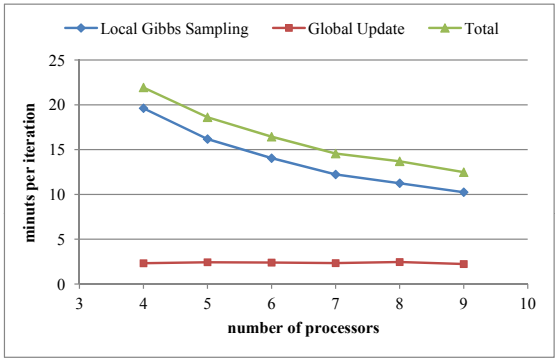

Fig. 6. Execution time of AD-UMM with various P values on Twitter-III.

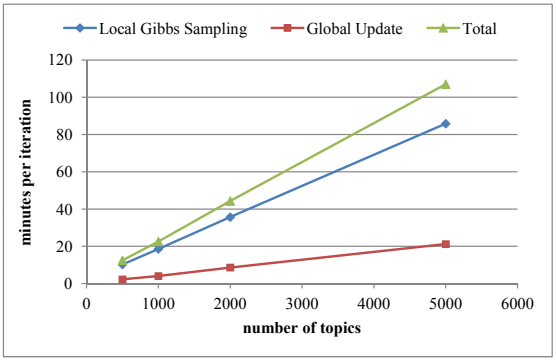

Fig. 7. Execution time of AD-UMM with various K values on Twitter-III.

because it can 1) deal with the data sparseness and topic diversity problems, 2) jointly model users and messages in a unified framework, and 3) efficiently handle large-scale datasets. Experimental results on Sina Weibo and Twitter datasets show that UMM indeed performs better in topic discovery and message recommendation. Experimental results on a large-scale Twitter dataset further demonstrates the scalability and efficiency of distributed UMM. As future work, we plan to apply UMM to various real-world applications and test its performances in the applications.

References

1. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing user modeling on twitter for personalized news recommendations. User Modeling, Adaption and Personalization, Lecture Notes in Computer Science (2011)
2. Ahmed, A., Low, Y., Aly, M., Josifovski, V., Smola, A.J.: Scalable distributed inference of dynamic user interests for behavioral targeting. In: Proceedings of the

17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
3. Blei, D., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research (2003)
4. Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E., Yu, Y.: Collaborative personalized tweet recommendation. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (2012)
5. Diao, Q., Jiang, J.: A unified model for topics, events and users on twitter. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013)
6. Diao, Q., Jiang, J., Zhu, F., Lim, E.P.: Finding bursty topics from microblogs. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (2012)
7. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.Y.: An empirical study on learning to rank of tweets. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (2010)
8. Grant, C., George, C.P., Jenneisch, C., Wilson, J.N.: Online topic modeling for real-time twitter search. In: Proceedings of the 20th Text REtrieval Conference (2011)
9. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences (2004)
10. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the 1st Workshop on Social Media Analytics (2010)
11. Hong, L., Doumith, A.S., Davison, B.D.: Co-factorization machines: Modeling user interests and predicting individual decisions in twitter. In: Proceedings of the 6th Annual ACM International Conference on Web Search and Data Mining (2013)
12. Hu, Y., John, A., Wang, F., Kambhampati, S.: Et-lda: Joint topic modeling for aligning events and their twitter feedback. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence (2012)
13. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (2002)
14. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002)
15. Li, H.: Learning to Rank for Information Retrieval and Natural Language Processing (2011)
16. Michelson, M., Macskassy, S.A.: Discovering users' topics of interest on twitter: A first look. In: Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data (2010)
17. Newman, D., Asuncion, A., Smyth, P., Welling, M.: Distributed inference for latent dirichlet allocation. In: Advances in Neural Information Processing Systems (2007)
18. Pennacchiotti, M., Gurumurthy, S.: Investigating topic models for social media user recommendation. In: Proceedings of the 20th International Conference Companion on World Wide Web (2011)
19. Ramage, D., Dumais, S.T., Liebling, D.J.: Characterizing microblogs with topic models. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (2010)
20. Ren, Z., Liang, S., Meij, E., de Rijke, M.: Personalized time-aware tweets summarization. In: Proceedings of the 36th annual international ACM SIGIR conference on Research and development in information retrieval (2013)

21. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2004)
22. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. Journal of the American Statistical Association (2006)
23. Wen, Z., Lin, C.Y.: On the quality of inferring interests from social neighbors. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)
24. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (2010)
25. Wu, W., Zhang, B., Ostendorf, M.: Automatic generation of personalized annotation tags for twitter users. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (2010)
26. Xu, Z., Lu, R., Xiang, L., Yang, Q.: Discovering user interest on twitter with a modified author-topic model. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (2011)
27. Xu, Z., Zhang, Y., Wu, Y., Yang, Q.: Modeling user posting behavior on social media. In: Proceedings of the 35th annual international ACM SIGIR conference on Research and development in information retrieval (2012)
28. Yuan, Q., Cong, G., Ma, Z., Sun, A., Magnenat-Thalmann, N.: Who, where, when and what: discover spatio-temporal topics for twitter users. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2013)
29. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: Proceedings of the 33rd European Conference on Advances in Information Retrieval (2011)