We Can Learn Your #Hashtags: Connecting Tweets to Explicit Topics

Wei Feng [1], Jianyong Wang [2]

Department of Computer Science and Technology, Tsinghua University
Beijing, China
[1] feng-w10@mails.tsinghua.edu.cn
[2] jianyong@tsinghua.edu.cn

***Abstract*—In Twitter, users can annotate tweets with hashtags to indicate the ongoing topics. Hashtags provide users a convenient way to categorize tweets. From the system's perspective, hashtags play an important role in tweet retrieval, event detection, topic tracking, and advertising, etc. Annotating tweets with the right hashtags can lead to a better user experience. However, two problems remain unsolved during an annotation: (1) Before the user decides to create a new hashtag, is there any way to help her/him find out whether some related hashtags have already been created and widely used? (2) Different users may have different preferences for categorizing tweets. However, few work has been done to study the personalization issue in hashtag recommendation. To address the above problems, we propose a statistical model for personalized hashtag recommendation in this paper. With millions of <tweet, hashtag> pairs being published everyday, we are able to learn the complex mappings from tweets to hashtags with the wisdom of the crowd. Two questions are answered in the model: (1) Different from traditional item recommendation data, users and tweets in Twitter have rich auxiliary information like URLs, mentions, locations, social relations, etc. How can we incorporate these features for hashtag recommendation? (2) Different hashtags have different temporal characteristics. Hashtags related to breaking events in the physical world have strong rise-and-fall temporal pattern while some other hashtags remain stable in the system. How can we incorporate hashtag related features to serve for hashtag recommendation? With all the above factors considered, we show that our model successfully outperforms existing methods on real datasets crawled from Twitter.**

I. Introduction

Twitter is one of the most popular microblogging platforms around the world[1]. Users can post text messages of up to 140 characters to tell others "what they are doing" or "what is happening". User-generated short messages are called tweets. By following others, users can keep up with their latest posts. Tweets can contain URLs, embedded images/videos, user mentions, locations, and hashtags. We will focus on the hashtag adoption in this paper.

Hashtags are words prefixed with "#" and are used to indicate the topics of tweets. For example, "#Election2012" can be used in tweets related to United States presidential election of 2012. An example tweet is "Why did Mitt Romney lose the presidential election? Here's a roundup of conservative commentary: http://bit.ly/YZFQod #Election2012".

Hashtags play an important role in Twitter. Popular hashtags can become trending topics in the home page of Twitter. The functions of hashtags are briefly summarized as follows: (1) Users can categorize and search tweets by hashtags. (2) Hashtags can lead to temporary discussion groups driven by special events or interests. (3) Hashtags are the core elements in event detection and tracking [2], [3], [4], tweets retrieval [5], [6], analysis of information diffusion [7], [8], and advertising [9], [10], [11]. Thus annotating tweets with the right hashtags is the foundation for many high-level applications.

Despite the great importance of hashtags, a few problems remain unsolved when a user wants to annotate a tweet:

- Before creating a new hashtag, is there any way for the user to find out whether some related hashtags have already been created and widely used? Without hashtag recommendation, hashtags can easily explode since different users may choose different words as hashtags to describe the ongoing topic, although some of them may represent similar meanings. On the other hand, users can only handle a portion of information they receive[12]. Hashtag recommendations can help users reach consensus on the adoption of hashtags, which not only controls hashtag explosion but also facilitates topic detection and tracking, hashtag-based retrieval tasks, and other hashtag related tasks.
- Different users may have different preferences for categorizing tweets. Without personalized hashtag recommendation, users will spend a lot of time on categorizing tweets and maintaining their existing classification systems.
- According to our dataset, only 20% tweets are annotated with hashtags. This means 80% of the tweets are not associated with explicit topics and they cannot be retrieved according to hashtags. Hashtag recommendation can help reduce the number of un-annotated tweets.

To address the above problems, we face three challenges:

- Hashtags have strong rise-and-fall patterns. Some event-specific hashtags may burst in a short time and disappear when people are attracted by other hot events. Meanwhile, we also have some general hashtags (e.g., "#travel") remain stable in the system. Thus we have to consider temporal characteristics of different hashtags.
- Tweets are extremely short but they are not fully unstructured. It calls for effective methods to incorporate auxiliary information (e.g., user mentions, and Web page

links) to measure relevance between tweet content and hashtags.
- Users' hashtag adoption history, as well as their locations and social network, provide valuable clues for predicting hashtag adoptions. All the information must be incorporated into a unified model.

We propose a statistical model for Personalized Hashtag Recommendation (PHR) in this paper. With millions of user-generated <tweet, hashtag> pairs being published everyday, we are able to learn the complex mapping from a tweet to a hashtag. Our model tries to consider all sources of information available in Twitter: (1) Content-related features like terms, URLs and user mentions. (2) User-related features like tweeting histories, social influence, and locations. (3) Hashtag-related features which describes the temporal patterns of hashtags adoptions.

To the best of our knowledge, this is the first paper to study Personalized Hashtag Recommendation at the tweet level. The differences between existing work and ours will be dicussed in detail in Section II.

Our contributions are summarized as follows,

- We are among the first to study personalized hashtag recommendation at the tweet level, which helps users find the right hashtags according to their annotation preferences.
- We propose a general hybrid recommendation model. Explicit features and latent factor models are combined together. The model is fully extensible to new features or latent factors.
- We have conducted extensive experiments on real datasets crawled from Twitter. Our model is empirically evaluated to be more effective than existing models.

The rest of the paper is organized as follows: Related work is introduced in Section II. In Section III, we introduce the dataset for principle analysis and give a formal definition of our problem. In Section IV, we discuss basic recommendation strategies and introduce our framework. In Section V, we discuss our model in detail, including various features we adopt and the optimization process. Section VI describes our experimental study on datasets crawled from Twitter, where we show that our model is more effective than the existing models. Finally, we conclude the paper in Section VII.

II. Related Work

In this section, we will introduce some related work and discuss the differences from ours.

First, we introduce two work that are considered most relevant to this paper: content-based hashtag recommendation[13], and user-level hashtag recommendation[14]. The comparison between their work and ours has been summarized in Table I, which will be explained in the following.

TABLE I
COMPARISON WITH GENERAL HASHTAG RECOMMENDATION. AUXILIARY INFORMATION INCLUDES TEMPORAL AND SPATIAL CHARACTERISTICS OF HASHTAG ADOPTIONS AND ANY OTHER KIND OF EXPLICIT FEATURES.

Method	Yang[14]	Khabiri[13]	Ours
Content based	√	√	√
Collaborative Filtering			√
Auxiliary Information			√
Social Influence	√		√
User Level	√		
Tweet Level		√	√

Content-based Hashtag Recommendation. Khabiri[13] has recently proposed a content-based hashtag recommendation method: recommending hashtags given the content of a tweet, where a tweet is represented by a bag of words. The relevance between a word and a hashtag is measured on a hashtag-word co-occurrence graph. The final relevance score between a tweet and a hashtag is computed as an aggregation of all the hashtag-word relevance scores. However, this method cannot provide personalized results since user information is ignored. Moreover, it treats tweets as fully un-structured documents. In contrast, our model makes use of auxiliary information such as Web page links, mentions, and time-stamps to identify the ongoing topics.

User-level Hashtag Recommendation. Yang[14] has proposed a user-level hashtag recommendation method recently, which predicts whether or not a hashtag may be adopted by the target user in the future. Two types of features are studied: (1) role-unspecific features which describes basic characteristics of users and hashtags (e.g., the number of unique hashtags used by user u, and the number of tweets containing hashtag h); and (2) role-specific features which describe the relevance between the target user u and a candidate hashtag h (e.g., cosine similarity between u's profile and h's profile, and sum prestige of users who have used h). A SVM classifier with the RBF kernel is used for prediction. However, since tweet-specific information is ignored, it recommends the same set of hashtags regardless of which tweet is being considered. Moreover, neither user locations nor social network information are considered.

Personalized Tag Recommendation. In social tagging systems like Delicious[1] and Flickr[2], users can annotate items with their own tags, in which case items are organized in their own way. When a user wants to annotate an item, personalized tag recommendation suggests hashtags by considering both the users's annotation preference and tags' relevance to the current item.

The state of the art methods are either based on graph models[15], [16], [17] or tensor factorization[18], [19], [20], where annotation behavior is represented by $<user, item, tag>$ triples. It seems that these methods can be adapted to solve our problem if we treat tweets as general items in social tagging systems. However, this cannot work for the following reason. In personalized tag recommendation, item IDs are used in graph construction or tensor factorization, which requires that items should exist in both the training set and the test set. But what we do is to recommend hashtags for new tweets

[1] delicious.com
[2] www.flickr.com

instead of existing tweets.

Besides the above methods, Lu[21] has proposed a content-based method to recommend hashtags for Web pages, where new Web pages are mapped to existing ones according to their content similarity. However, this method does not take personalization into account. Moreover, a tweet is usually much shorter than a Web page, which makes it extremely important to make good use of their rich auxiliary information (e.g., mentions, links, and time stamps) for recommendation. All the auxiliary information is not considered in their work since it is proposed for annotating Web pages.

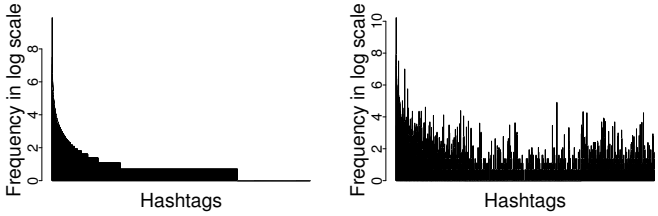
Text Classification. By treating hashtags as class labels, hashtag recommendation can also be considered as a traditional text classification problem[22]. There are mainly two differences between text classification and our problem: (1) Class labels in text classification are considered to be constants in both the training set and the test set, while hashtags often have strong rise-and-fall patterns[23]. Some hashtags burst in the training set may disappear in the test set. Similarly, some rarely used hashtags in the training set may grow rapidly in the test set. Meanwhile, hashtags like '#travel' and '#sports' may remain stable in the system. In other words, the temporal characteristics of hashtag adoption is not considered in traditional text classification. (2) Tweets are not fully un-structured documents. They contain rich auxiliary information. Traditional text classification tasks are not designed for categorizing tweets.

Besides traditional text classification techniques, some recent works have focused on short text classification [24], [25]. [25] has extracted eight features for 5-class classification (i.e., news, events, opinions, deals, and private messages). Limited by its small parameter space, this method cannot handle classification problem with millions of hashtags as class labels. [24] has further considered changes in word probability to classify Tweet stream, which is only based on texts. It does not consider Twitter-specific features nor user's preferences.

Other Hashtag Related Research. Besides general hashtag recommendation problem, many work have been done in studying the general patterns for hashtag propagation [26], [27], [23], [7], [12]. [7] predicts the number of times that a hashtag is used in a specific community in a time frame. It studies various feature types in information diffusion. A linear regression model is used to combine content features, temporal features, and topological features. [26] studies the peaks in the popularity of hashtags. By linking hashtags to events in the physical world, [26] finds that hashtags are mostly driven by external events instead of internal information spreads. [27] predicts whether a hashtag will be popular in the next day. [12] finds that the entropy of hashtags keeps growing while users' attention is limited to a small range of topics. It proposes an agent-based model to study how limited attention of individual users affects the popularity of hashtags. Recently, [23] proposes a unified model to explain all the rise-and-fall patterns during hashtag propagation.

TABLE II
DATASET STATISTICS: #TWEET$^+$ REPRESENTS THE NUMBER OF TWEETS THAT HAVE HASHTAGS. #MEANFREQ REPRESENTS THE MEAN FREQUENCY THAT A HASHTAG IS ADOPTED.

#User	#Tweet	#Tweet$^+$	#Hashtag	#MeanFreq.
0.12M	8.1M	1.6M	0.11M	15



(a) February, 2012 (b) March, 2012

Fig. 1. Hashtag Distributions in February and March, 2012

III. PRELIMINARIES

A. Dataset for Principle Analysis

We crawled 8.1 million tweets from 120,000 users with breath-first strategy using Twitter's REST API. For each user, we crawled his/her profile, following list, and their most recent 3200 tweets[3] before April 1st, 2012. Tweets were crawled in reverse chronological order and recent tweets are preferred over older tweets. In other words, we first crawled tweets generated during March, then February and so on. According to this strategy, tweets will become sparse if they were created much earlier than March (we started to crawl at the first day of April). To avoid this sampling bias, we only focused on the dense part: tweets posted during February and March. The basic statistics about the dataset are shown in Table II. We can see that about 20% (1.6M/8.1M) tweets contain hashtags[4]. These annotated tweets are the foundation for connecting a tweet to its relevant hashtags.

Before delving into the prediction algorithm, a question needs to be answered first: Is it possible to predict hashtags based on the past data? If hashtags change greatly from month to month, then a statistical model based on the last month would absolutely fail. To have a deeper insight into how hashtags evolve over time, we compared hashtag distributions in February and March, 2012. The results are shown in Figure 1. We have two observations: (1) The overall distribution is similar. (2) The tail part of hashtag distribution in March is quite different from that of February. This is mainly because popular events differ from month to month. New hashtags and some rarely used hashtags may become popular in March.

To quantify how quickly hashtags evolve over the two months, we analyze the results in different granularity. The results are shown in Table III. Now we discuss the results in detail:

- **Comparison of February and March**. This is used to test whether we can predict hashtags for the next month given the knowledge of the current month. We can draw several conclusions from the result: (1) 99%

[3]The number is limited by Twitter.
[4]Retweets are also considered.

TABLE III
HASHTAG'S EVOLUTION OVER TIME. "TOP-100", "TOP-500" AND "TOTAL" REPRESENT THE NUMBER OF HASHTAGS SHARED IN COMMON BETWEEN THE TRAINING SET AND THE TEST SET WHEN TAGS ARE SORTED IN DESCENDING ORDER OF THEIR FREQUENCY, RESPECTIVELY. "TRAINING%" AND "TEST%" REPRESENT THE PROPORTION OF HASHTAGS IN COMMON IN THE TRAINING SET AND THE TEST SET, RESPECTIVELY.

Training Set	#Hashtag	Test Set	#Hashtag	Top-100	Top-500	Total	Training%	Test%
February	91,821	March	114,453	62(62%)	416(83%)	91,020	99%	80%
Last Week of February	35,578	1st Week of March	36,955	69(69%)	419(84%)	34,875	98%	94%
February	91,821	1st Week of March	36,955	71(71%)	389(78%)	36,835	40%	99%

hashtags in February also appear in March. This means that the global-level hashtag dictionary remains stable from month to month. (2) With the fast development of Twitter, the number of hashtags in March is 1.2 times bigger than that in February. This is in consistent with a recent research[12] which states that the entropy of hashtags keeps increasing. (3) The overlap of the top 100 hashtags (denoted by top-100) is much smaller than that of the top 500 hashtags (denoted by top-500). This means there are some new burst events in each month. These burst events are mainly influenced by our physical world and are very hard to predict, since future events are hardly predictable.

- **Comparison of Last Week of February and the First Week of March**. This comparison focuses on whether hashtags are predictable from week to week, in which case the model is trained and updated by week instead of by month. Compared with the former situation, we can draw the following conclusions: (1) The overlap of top-100 increased from 62% to 69%. This indicates updating model with a higher frequency can benefit predictions for new events. (2) The overlap of hashtags with the test data has been improved from 80% to 94%. In addition, hashtags remain more stable from week to week than month to month.
- **Comparison of February and the First Week of March**. This experiment tries to answer an important question: Given more data about the past (expanded from one week to one month), can we improve the prediction performance? As we can see from Table III, the overlaps with top-100 and the test data are both improved while the overlap with training data drops sharply from 98% to 40%. The benefit brought by more data is not so obvious. More evaluation will be covered in our experiments.

B. Problem Statement

Given a tweet $d \in D$ and its publisher $u \in U$, Personalized Hashtag Recommendation ranks hashtags $h \in H$ according to both (1) the relevance to the content of the tweet, and (2) the publisher's preference of annotation.

Personalized Hashtag Recommendation facilitates users in two ways: (1) find out whether some suitable hashtags have already been created to describe the current tweets; (2) categorize tweets with the user's preference. From the system's perspective, our work can also help control the explosion of hashtags.

Note that hashtag set H is constructed from the past data. Thus we do not aim to extract new hashtags from the content, and this work can be considered as a complement for some NLP techniques like keyphrase extraction and named entity recognition, etc. As discussed in the previous section, although the overlap of the top 100 hashtags ranges from 62% to 71%, the overall overlaps for top-500 and the test dataset still remain high. This means that it is still possible to recommend hashtags for partially new events as long as the model is updated at a reasonable frequency (by week or even by day).

TABLE IV
THE TOP THREE PRIVATE HASHTAGS, BURST HASHTAGS, AND GENERAL HASHTAGS

Private	Burst	General
#myweekendride	#wikipediaBlackout	#ff
#WorstBuy	#2012OlympicCeremony	#jobs
#ClusterofThoughts	#letstalkiphone	#travel

IV. FRAMEWORK

A. Recommendation Strategies

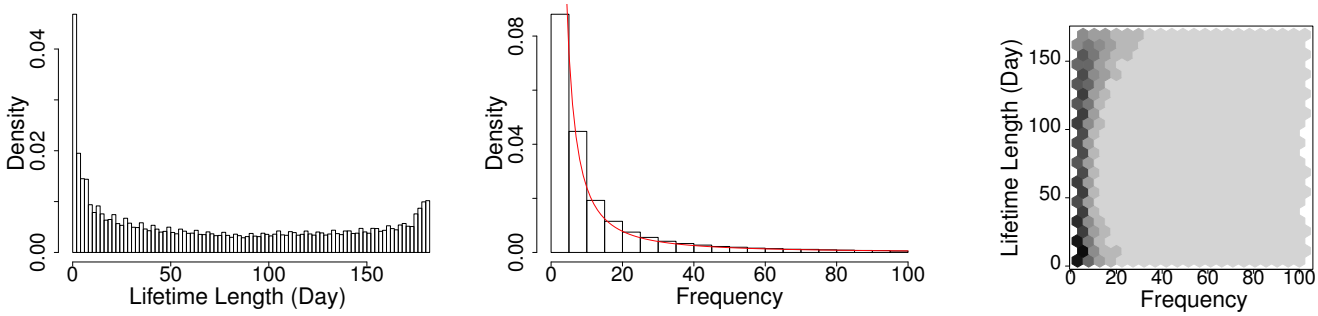
We first introduce several recommendation strategies for personalized hashtag recommendation:

Content-Relevant Strategy. This strategy ranks hashtags according to their relevance to tweet content regardless of publishers' personal annotation preferences. [13] adopts this strategy for content-based tweet-level hashtag recommendation.

User-Relevant Strategy. This strategy ranks hashtags according to the publisher's annotation preference regardless of which tweet is being annotated. [14] uses this strategy for user-level hashtag recommendation. Although this strategy cannot be directly used for tweet-level hashtag recommendation, it is an important component in our model.

According to the relation between hashtags' lifetime length and frequency, we classify hashtags into the following three categories and discuss the corresponding recommendation strategies. Some typical example hashtags of each category are shown in Table IV.

Private Hashtags with User-Relevant Strategy. A private hashtag has a long lifetime but a low frequency. As Table IV shows, these hashtags are only limited by personal usage. To find hashtags belonging to this category, we define personalness as the degree of being private as the ratio between lifetime length and frequency. If we rely heavily on content-relevant strategy, hashtags from popular tweets will dominate the results, making it almost impossible to recommend private hashtags. In contrast, user-relevant strategy



(a) Distribution of lifetime length

(b) Distribution of frequency

(c) Hashtag densities by varying lifetime length and frequency.

Fig. 2. The relation between lifetime length and frequency.

puts more emphasis on the user's annotation history, which can promote these private hashtags.

Burst Hashtags with Content-Relevant Strategy. Burst hashtags describe events happening in the physical world. They usually burst in a short time with a global discussion in Twitter. When a hot event fades out, its related hashtags will also disappear. We define burstiness as the ratio of frequency and lifetime length, which is the reciprocal of personalness. As shown in Table IV, all the hashtags of this category are event-specific. To recommend such hashtags, we focus more on finding content-relevant hashtags instead of user-relevant hashtags, since burst hashtags only represent users' temporary interests.

General Hashtags with Two Strategies Combined. General hashtags lie between private hashtags and burst hashtags. They have long lifetimes as well as high frequencies. As shown in Table IV, these hashtags describe high-level topics that can last long in the system. We define generality as the product of lifetime length and frequency. Unlike burst hashtags, general hashtags represent users' general interests. To recommend such hashtags, we should use a hybrid method that combines content-relevant strategy and user-relevant strategy.

The overall relation between lifetime length and frequency is shown in Figure 2. In Figure 2a, the first peek is caused by extremely short-lived hashtags, which corresponds to burst hashtags. The second peek is caused by extremely long-lived hashtags, which represents general hashtags. As shown in Figure 2b, frequency follows a power-law distribution, which tells that most hashtags have low frequencies. Finally, we show how hashtag density changes by varying lifetime length and frequency in Figure 2c. The density pattern can be easily explained by mixing the distribution of frequency along the x-axis and the distribution of lifetime length along the y-axis.

In this section, we have discussed content-relevant strategy and user-relevant strategy on three types of hashtags. In this paper, we combine both strategies for personalized hashtag recommendation.

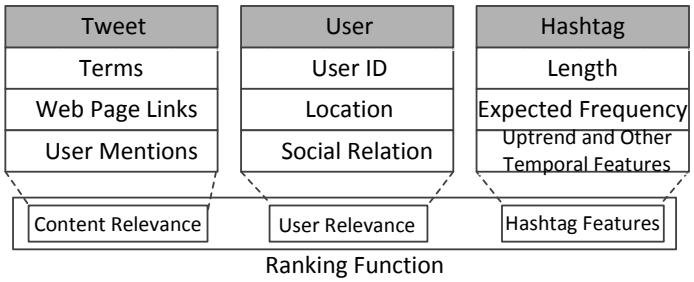


Fig. 3. Framework

B. Framework

Different from traditional recommender systems which only deal with $<user, item>$ pairs, personalized hashtag recommendation handles $<user, tweet, hashtag>$ triples with rich auxiliary information:

- **Tweet-Related Features**. They are terms, Web links, and user mentions, and we will discuss them in Section V-A.
- **User-Related Features**. They are user IDs, locations, and social relations, which will be covered in Section V-B.
- **Hashtag-Related Features**. They are hashtag IDs, length, popularity, recency, stability, uptrend and time-decay, which will be discussed in Section V-C.

The overall framework is shown in Figure 3. These features can be grouped into two categories and are handled with different approaches:

Numeric Features with Linear Discriminative Models. Numeric features include hashtag length, popularity, and uptrend, etc. The final ranking score is a linear combination of these numeric features. Suppose we have a user u, a tweet d and a candidate hashtag h. Let $\mathbf{x}$ denote the feature vector composed of numeric features and $\boldsymbol{\theta}$ denote the feature weights, the ranking score r_{udh} is defined as

$$r_{udh} = \boldsymbol{\theta}^T \mathbf{x} \tag{1}$$

Categorical Features with Latent Factor Models. These features include user IDs, term IDs, and hashtag IDs, etc. All of them have high dimensions and are very sparse. For example, the i-th term is represented by a long sparse vector $(0,...,1,...,0)^T$ with '1' at the i-th entry. Some of the recent work[28], [29] have shown that latent factor models are very

TABLE V
TOP-3 HASHTAGS FOR POPULAR LINKS AND MENTIONS

Amazon	BBC Tech.
#Win	#Apple
#VideoGame	#Google
#GiftCard	#Twitter

(a) Links

@BarackObama	@JLin7
#Obama2012	#Knicks
#Iran	#Linsanity
#Israel	#JeremyLin

(b) Mentions

TABLE VI
TOP-3 HASHTAGS FOR POPULAR USERS AND LOCATIONS.

SAP	IBMBigData
#SAP	#bigdata
#HANA	#hadoop
#BI	#analytics

(a) Users

New York	Toronto
#Knicks	#Toronto
#nyfw	#cdnpoli
#Oscars	#TTC

(b) Locations

effective in handling sparse categorical features, particularly in computing the relevance score between two objects (such as a user and a hashtag) represented by their IDs. Let $\mathbf{u}$ and $\mathbf{h}$ denote the latent factors for user u and hashtag h (which are both low-dimension vectors), respectively. Let $Rel(u, h)$ denote the relevance score between u and h. According to latent factor models, $Rel(u, h)$ is computed as follows:

$$Rel(u, h) = \mathbf{u}^T \mathbf{h} \tag{2}$$

where both $\mathbf{u}$ and $\mathbf{h}$ are learned to fit the data (whether user u has adopted hashtag h). Similarly, let $\mathbf{w}$ denote the latent factor for term w, the relevance score between term w and hashtag h is computed as $\mathbf{w}^T\mathbf{h}$.

Our model combines linear discriminative models with latent factor models since we both have numeric features and categorical features. Formally, given a tweet $d \in D$ composed by user $u \in U$ and a hashtag candidate $h \in H$, the ranking score r_{udh} for hashtag h is

$$r_{udh} = \boldsymbol{\theta}^T \mathbf{x} + Rel(u, h) + Rel(d, h) \tag{3}$$

where $\boldsymbol{\theta}^T\mathbf{x}$ measures the contribution from explicit numeric features, $Rel(d,\ h)$ and $Rel(u,\ h)$ measure the content-relevance and the user-relevance, respectively. They both use latent factor models. This equation is a natural extension of Equation 1. In the next section, we will discuss each component in detail.

V. PERSONALIZED HASHTAG RECOMMENDATION

A. Measuring Content Relevance

The first and most intuitive principle is to recommend content-relevant hashtags. Besides basic terms, tweets can also contain Web links and user mentions. All of them can be used to measure the content-relevance.

Measuring Relevance by Terms. The basic idea is to find highly co-occurred <term, hashtag> pairs. Suppose an un-annotated tweet is talking about the United States presidential election of 2012. It has terms like "Barack Obama" and "Mitt Romney"[5]. Meanwhile, thousands of tweets with "Barack Obama" and "Mitt Romney" are annotated with hashtag "#Election2012". Then there is a high probability that tweet d should also be annotated with "#Election2012".

Measure Relevance by Links. Besides pure textual descriptions, we can also insert Web links to tweets to provide more information. We find that different webpages or websites are described with different hashtags. We show the top-3 frequent hashtags for two popular websites in Table Va.

[5]Mitt Romney is the major challenger in this election.

Measure Relevance by Mentions. To explicitly inform other users of a new post, we can add user mentions to tweets with the format of "@username". There are three scenarios for using user mentions: (1) The tweet is talking about the mentioned user. (2) The publisher wants to notify her/him since she/he might be interested. (3) Starting a conversation or replying to someone. We are only interested in the former two scenarios. We list the top 3 frequent hashtags for the two well-known users in Table Vb. We can see that the US president Obama and the NBA player Jeremy Lin are related with different topics. Thus user mentions can also indicate the ongoing topics.

Now we give a formal description on how to compute content relevance. Suppose the target tweet $d \in D$ contains $k^{(w)}$ words $\{w_1,\ w_2,\ ...,\ w_{k^{(w)}}\}$, $k^{(l)}$ links $\{l_1, l_2, ..., l_{k^{(l)}}\}$, and $k^{(m)}$ mentions $\{m_1, m_2, ..., m_{k^{(m)}}\}$, the content-relevance score between tweet d and hashtag h is computed by

$$Rel(h, d) = [\sum_{i=1}^{k^{(w)}} \alpha_i^{(w)} \mathbf{w}_i^T + \sum_{i=1}^{k^{(l)}} \alpha_i^{(l)} \mathbf{l}_i^T + \sum_{i=1}^{k^{(m)}} \alpha_i^{(m)} \mathbf{m}_i^T]\mathbf{h} \tag{4}$$

where

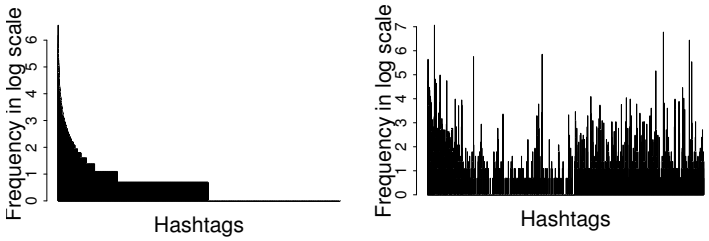
- $\mathbf{w}_i$, $\mathbf{l}_i$, $\mathbf{m}_i$, $\mathbf{h}$ represent the latent factors for term w_i, link l_i, mention m_i, and the candidate hashtag h, respectively.
- $\alpha_i^{(w)}$, $\alpha_i^{(l)}$, and $\alpha_i^{(m)}$ are weights of each latent vectors. We require $\sum_{k=1}^{k=k^{(*)}} \alpha_i^{(*)} = 1$, otherwise a tweet with many terms will dominate the other latent factors.
- $\sum_{i=1}^{k^{(w)}} \alpha_i^{(w)} \mathbf{w}_i^T$, $\sum_{i=1}^{k^{(l)}} \alpha_i^{(l)} \mathbf{l}_i^T$, and $\sum_{i=1}^{k^{(m)}} \alpha_i^{(m)} \mathbf{m}_i^T$ represent weighted average of terms, links, and mentions, respectively.

Choices of $\boldsymbol{\alpha}^{(*)}$. For terms, $\alpha_i^{(w)}$ is defined to be TF-IDF(w_i) / ($\sum_{k=1}^{k^{(w)}}$TF-IDF(w_k)). In this way, we can punish common words and promote informative words. Since most tweets contain one or two links or mentions, $\alpha_i^{(l)}$ and $\alpha_i^{(m)}$ are defined to be the reciprocal of $k^{(l)}$ and $k^{(m)}$, respectively. In other words, links and mentions are both equally weighted.

Term-Hashtag Affinity. Besides latent factors, we also use probability $p(h|w)$ to model the relevance between hashtag h and term w, which is estimated as the ratio of the number of times that h and t co-occurred and the number of times that t co-occurred with each term. Suppose a tweet contains k terms, i.e., $d = \{w_1, ..., w_k\}$, term-hashtag affinity is defined to be the average of $p(h|w_i)$ $(i = 1, .., k)$. This explicit feature is in $\boldsymbol{\theta}^T\mathbf{x}$ term in Equation 3.

B. Measuring User Relevance

We measure user relevance from three aspects: (1) the preference of the user herself/himself; (2) the preferences of



(a) New York (b) Toronto

Fig. 4. Comparison of Hashtag Distributions of New York and Toronto

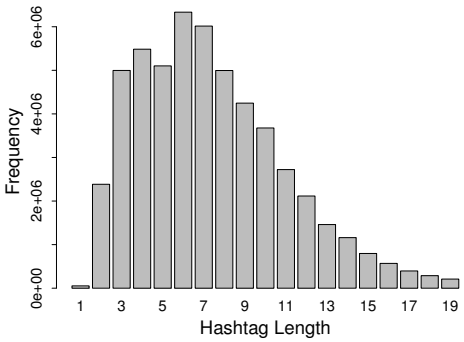


Fig. 5. Distribution of Hashtag Length

her/his friends; (3) the location of the user.

Measuring User Preference by User IDs. Since different users are interested in different topics, annotation behaviors differ from user to user. The top 3 most popular hashtags for "SAP" and "IBMBigdata" are shown in Table VIa. The result reflects the fact that SAP uses HANA[6] as its solution for big data analysis while IBM is focused on Hadoop. We can also see that each user has his/her own way to categorize tweets, even for tweets of a particular topic (big data analysis). We also find that users prefer to use hashtags which have been used before. For a single user, he/she uses about 2.6 hashtags he/she used last month. In contrast, this overlap drops sharply to 0.03 for different users.

Incorporating Social Influence. Since users mostly receive tweets from their followees, their adoptions of hashtags are influenced by their neighbors. We compute the similarities of hashtags for neighbors and non-neighbors, respectively. We find that a user has 1.5 hashtags in common with their followees while only has 0.3 hashtags in common with non-neighbor users. Moreover, for users who have retweeted each other, the number of hashtags in common increases to 7.5.

Measuring Relevance by Locations. Users provide their location information in their profiles, including countries, states, and cities. Since local events differ from city to city, different places may have different hashtag distributions. We analyzed the hashtag distributions of New York and Toronto, the results are shown in Figure 4. We can see that the two distributions are very different. The top 3 hashtags are shown in Table VIb, where #nyfw represents New York Fashion Week. #cdnpoli represents generic Canadian political issues. #TTC represents Toronto Transit Commission. We find that people in New York are talking about "New York fashion week" (with hashtag #nyfw) while users in Toronto are talking about "Generic Canadian political issues" (with hashtag #cdnpoli).

Now we give a formal description on how to compute user relevance. Suppose user u has $k^{(f)}$ friends $\{u_1, u_2, ..., u_{k^{(f)}}\}$, and $k^{(p)}$ locations $\{p_1, p_2, ..., p_{k^{(p)}}\}$. The user-relevance score between user u and hashtag h is

$$Rel(h,u) = [\beta \mathbf{u}^T + (1-\beta)\sum_{i=1}^{k^{(f)}} \alpha_i^{(f)} \mathbf{u}_{f_i}^T + \sum_{i=1}^{k^{(p)}} \alpha_i^{(p)} \mathbf{p}_i^T]\mathbf{h} \quad (5)$$

[6]http://www.saphana.com/welcome

where

- $\mathbf{u}$, $\mathbf{u}_{f_i}$, $\mathbf{p}_i$, $\mathbf{h}$ represent the latent factors for user u, her/his i-th friend u_{f_i}, location p_i, and the candidate hashtag h, respectively. $\mathbf{u}$ and $\mathbf{u}_i$ use the same latent vector space.
- $\alpha_i^{(f)}$ and $\alpha_i^{(p)}$ are weights of the corresponding latent vectors. We require $\sum_{k=1}^{k=k^{(*)}} \alpha_i^{(*)} = 1$, otherwise a user with many friends or with many locations will dominate the other latent factors.
- $\beta \mathbf{u}^T + (1-\beta)\sum_{i=1}^{k^{(f)}} \mathbf{u}_{f_i}^T$ combines u's personal preference with her/his friends. $\beta \in [0,1]$ controls the biases.

Choices of $\boldsymbol{\alpha}^{(*)}$. For friends, $\alpha_i^{(f)}$ is defined as RT_COUNT(u, u_{f_i}) / ($\sum_{k=1}^{k^{(f)}}$RT_COUNT(u, u_{f_k})), where RT_COUNT represents the times of u retweeting u_{f_i}. u is considered to trust u_{f_i} more if she/he retweets u_{f_i} more. In this way, the strength of a relation is considered. For locations, we set them to be equally weighted since most users have only one or two locations.

C. Incorporating Hashtag Features

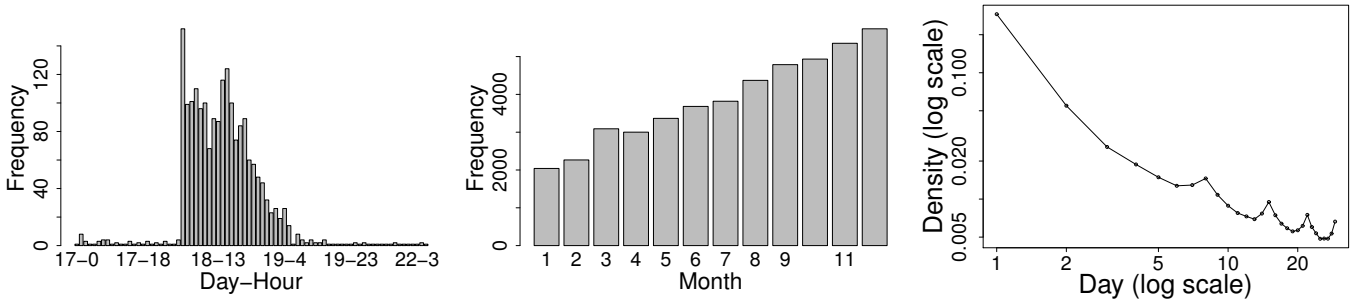
In this section, we will introduce some numeric features that describe the basic property of a hashtag.

Character Length. We analyzed how character length of a hashtag affects its adoption. The result is shown in Figure 5. We can see that hashtags of length 3 to 10 are more preferred.

Now we focus on how to incorporate temporal aspects of hashtags. Remind that we classify hashtags into three categories in Section IV-A, i.e., private hashtags, burst hashtags, and general hashtags. We find that burst hashtags and general hashtags have more stronger temporal patterns than private hashtags. The temporal patterns of #wikipediablackout and #travel are shown in Figure 6a and 6b. We can see that #wikipediablackout has a sharp rise pattern and a power-law fall pattern. This is consistent with the recent work[23]. In contrast, the frequency of #travel increases slowly from month to month, which can be explained by the development in Twitter. The overall temporal pattern averaged by all hashtags is shown in Figure 6c, which can be well fitted by power-law distribution.

According to the characteristics of these temporal patterns, we introduce the following numeric features:

Expected Frequency by Time Decay. Suppose a hashtag is used for N times in total. Let N_t denote the expected



(a) Temporal Pattern of #wikipediablackout (b) Temporal Pattern of #travel in 2011 (c) Overall Temporal Patterns

Fig. 6. The relation between Overall Temporal Patterns and frequency.

frequency at day t. According to the power-law distribution, the expected frequency at day t is $N_t = Nt^{-\lambda}$, where λ controls the speed of decay and is fitted to 1.65 according to our data. Since we cannot know the real N, we replace N with the highest frequency N_0 of the target hashtag.

Time Span since Last Occurrence. This feature is used to filter out the out-dated hashtags and promote the currently used hashtags.

Uptrend. Uptrend measures whether a hashtag will grow or decend in the future. It is defined as $N(t_n)/N(t_{n-1})$, where t_n and t_{n-1} are two consecutive sampling time stamps. The interval is a day in this paper.

Frequency of Last Day of Occurrence. This feature represents whether the hashtag is popular according to the newest data.

All the introduced numeric features are combined together as the feature vector $\mathbf{x}$ in Equation 3. Before we annotate a tweet, these features will help us filter out the out-dated hashtags and promote the hashtags created recently.

D. Learning Parameters

We have introduced all the features needed for the ranking function defined in Equation 3. In this section, we employ an optimization framework to learn the best feature weights from the data.

The basic idea of our optimization framework is to account for a loss whenever our recommended hashtag is not adopted in the training data. Minimizing the total loss through the training data will lead us to the best feature weights.

Loss Function First we model our task as a binary classification problem. Given a tweet d composed by user u and a candidate hashtag h, if u adopts h in tweet d, then the triple $<u, d, h>$ is treated as a positive example. Otherwise this triple is a negative example. Let $\overline{r}_{udh} \in \{0, 1\}$ denote the class label of triple $<u, d, h>$, where 0 represents a negative example and 1 represents a positive example. Suppose the ranking score is $\widehat{r}_{udp}$, the loss function is defined as follows:

$$loss = (\overline{r}_{udh} - 1)\log(1 - \widehat{r}_{udh}) - \overline{r}_{udh}\log(\widehat{r}_{udh}) + regularzation \quad (6)$$

where

- $\widehat{r}_{udh}$=sigmoid(r_{udh}), where r_{udh} is the ranking score defined by Equation 3, and sigmoid(x) = 1 / (1 + e^{-x}) maps r_{udh} to the range of (0, 1).
- the $regularzation$ term is used to prevent over-fitting, which will be discussed later.

Now we describe the intuition behind this loss function. If $\widehat{r}_{udh}$ is close to the real label $\overline{r}_{udh}$, the loss is close to 0. However, if $\widehat{r}_{udh}$ is close to 0 while $\overline{r}_{udh}$ is 1, the loss will go to positive infinity. By minimizing the loss function, we can push the predicted label $\widehat{r}_{udh}$ to be close to the real label $\overline{r}_{udp}$.

Regularization. We use L2 regularization and it is defined as follows

$$\begin{aligned} regularzation = {} & \lambda_{\boldsymbol{\theta}}||\boldsymbol{\theta}||^2 + \lambda_{\mathbf{u}}||\mathbf{u}_{(*)}||^2 + \lambda_{\mathbf{p}}||\mathbf{p}_{(*)}||^2 \\ & + \lambda_{\mathbf{w}}||\mathbf{w}_{(*)}||^2 + \lambda_{\mathbf{l}}||\mathbf{l}_{(*)}||^2 + \lambda_{\mathbf{m}}||\mathbf{m}_{(*)}||^2 + \lambda_{\mathbf{h}}||\mathbf{h}_{(*)}||^2 \end{aligned} \quad (7)$$

where $\lambda_{(*)}$ are constants that control the sensitiveness to big parameters. We put different λ on different group of feature weights. A good practice is to set $\lambda_{(*)}$ proportional to the square of number of parameters they punish.

Sampling Negative Examples. Both positive and negative examples are needed since we use a discriminative model. For each positive example $<u, d, h>$ ($u \in U$, $d \in D$, $h \in H$), we randomly choose an unused hashtag u' to replace the original hashtag u. Each time we meet the positive example, the corresponding negative example is constructed. We find this simple strategy works well for our problem in practice.

Minimize the Loss Function. We adopt stochastic gradient descent to minimize the loss function. Training instances are loaded one by one into the main memory, in which case loading all the instances is nearly impossible for large scale data. Stochastic gradient descent consists of three steps: (1) load a training instance $<u, d, h>$ and its class label r_{udh}. (2) compute the gradients with respect to each parameter related to $<u, d, h>$. The gradients will tell us how much the loss function changes when the parameters change. (3) Update parameters with a tiny step towards the descending gradient to minimize the loss function.

We take hashtag latent factor h as an example to illustrate the idea of stochastic gradient descent. Given a training instance $<u, d, h>$ and its label r_{udh}, we first need to compute

TABLE VII
DATASET STATISTICS

Dataset	#User	#Social Relation	#Tweet	#Hashtag	#Links	#Mention	#Location
Week-Day	56,968	584,018	465,373	20,137	23,931	15,108	10,647
Week-Week	69,537	670,966	910,790	35,047	39,478	31,529	12,053
Month-Week	91,896	1,092,634	1,889,186	43,678	76,559	105,246	15,454

the gradient with respect to h according to the loss function defined in Equation 6. We have

$$\frac{\partial l}{\partial \mathbf{h}} = (\overline{r}_{udh} - \widehat{r}_{udh})\frac{\partial r_{udh}}{\partial \mathbf{h}} + 2\lambda_{\mathbf{h}}\mathbf{h} \tag{8}$$

According to the chain rule, we need to further compute $\frac{\partial r_{udh}}{\partial \mathbf{h}}$. According to Equation 4 and Equation 5, we have

$$\begin{aligned}\frac{\partial r_{udh}}{\partial \mathbf{h}} =&(\sum_{i=1}^{k^{(w)}} \alpha_i^{(w)}\mathbf{w}_i^T + \sum_{i=1}^{k^{(l)}} \alpha_i^{(l)}\mathbf{l}_i^T + \sum_{i=1}^{k^{(m)}} \alpha_i^{(m)}\mathbf{m}_i^T) \\ &+ [\beta\mathbf{u}^T + (1-\beta)\sum_{i=1}^{k^{(f)}} \alpha_i^{(f)}\mathbf{u}_{f_i}^T + \sum_{i=1}^{k^{(p)}} \alpha_i^{(p)}\mathbf{p}_i^T]\end{aligned} \tag{9}$$

Plug the above equation into Equation 8, we get the gradient with respect to h.

The final step is to update **h** according to the gradient:

$$\mathbf{h}^{(t+1)} = \mathbf{h}^{(t)} - lr\frac{\partial l}{\partial \mathbf{h}^{(t)}} \tag{10}$$

where lr represents the learning rate and is empirically set to 0.1 for our task. We repeat the above process for each training instance from training data until the parameters converge.

VI. EXPERIMENTAL STUDY

A. Datasets

We use three subsets of the raw dataset for evaluation:

- **The Last Week of February vs The First Day of March, 2012**. This is used to test whether we can predict for the next day using data from last week. Models need to be updated every day.
- **The Last Week of February vs The First Week of March, 2012**. Compared with the first dataset, the test set is expanded from a day to a week while the training set remains unchanged. This is used to test how the amount of the test data affects the predicting performance. Models are updated every week.
- **February vs The First Week of March, 2012**. This is used to test whether we can predict for the next week given the data from the last month. Models need to be updated every week. Compared with the second dataset, the test data remains unchanged while the training data is expanded from a week to a month. We can find how the amount of training data affects the final performance.

The basic statistics of the datasets are shown in Table VII. We preprocessed the data as follows: (1) All the words and hashtags in tweets are stemmed and transformed to lowercase. Stopwords are removed. Retweets and replies are removed since we only focus on annotation on originally composed tweets. (2) Due to the limited space of tweets, most URLs are shortened using short URL services like TinyURL, bitly. We followed the redirects of each shorten URL and crawled the original long URL. Since URLs change frequently from day to day, we only used the truncated address at the last level. For example, "www.bbc.co.uk/food/ pageName" is truncated to "www.bbc.co.uk/food". (3) Like [14], we used retweet network as the social network. The relation is directed and the weight of a link is the retweet count. (4) Since we only recommend hashtags that are learned from the training data, new hashtags are removed from the test data. Among all the tweets in the test set, only 12% tweets are annotated by brand new hashtags.

B. Evaluation Metric

We use Mean Average Precision (MAP) to measure the performance, which is a widely used metric in ranking problems. First we introduce the definition of Average Precision (AP). Given a ranked list of n hashtags (h_1, h_2, ..., h_n), AP is defined as

$$AP = \frac{\sum_{k=1}^{n} Precision@k \times isAdopted(h_k)}{number\ of\ hashtags\ adopted} \tag{11}$$

where the indicator function $isAdopted(h_k)$ is 1 only if hashtag h_k is adopted, otherwise $isAdopted(h_k)$ is 0. Suppose we have N recommendation lists and the AP score of the i-th list is denoted as AP_i, MAP is defined as the mean of all APs:

$$MAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{12}$$

All the experiments were conducted on a server with Intel Xeon E5310 1.60GHz CPU (8 cores) and 20G memory. We implemented the algorithms in C++ with the support of Eigen libary[7] for fast vector/matrix manipulations.

C. Baseline Methods

TensorFac. Rendle[19] has proposed a tensor factorization method that considers pairwise interactions among users, items (tweets) and tags, which can be considered as the state of the art method for personalized tag recommendation. Since the original version cannot handle new tweets. we make the following adaption: (1) For each new tweet in the test set, find the top-5 similar tweets according to the cosine similarity of their term vectors. (2) The latent vector for the new tweet is computed as a weighted average of the latent vectors of the top-5 similar tweets. The averaged latent vector is used for prediction. Since latent vectors are learned for both users and tweets, this baseline considers both content and user preference.

GraphRec. Khabiri[13] proposed a general hashtag recommendation method based on the content of the tweet. To

[7]http://eigen.tuxfamily.org

the best of our knowledge, this is the most relevant work. GraphRec is summarized as follows: (1) Build a directed graph based on the co-occurrences of terms and hashtags. The weight of a link is initialized as the co-occurrence of the two corresponding nodes. (2) Normalize link weights so that the total out-link weights are summed to one, which is very similar to PageRank. (3) Compute the relevance between terms and hashtags according to the links. Readers can find more details about this step in [13]. (4) Given a tweet of k terms, the ranking score for hashtag h is the summation of relevance scores between h and each term. Since this is a graph-based method, we call this model GraphRec. This baseline only considers tweets' content.

UserLevelRec. Yang[14] proposed a user-level hashtag recommendation model that predicts which hashtags may be adopted by the target user in the future regardless of which tweet is being considered. Two types of features are studied: (1) role-unspecific features which describes the basic characteristics of users and hashtags (e.g., the number of unique hashtags used by user u, and the number of tweets containing hashtag h), and (2) role-specific features which describe the relevance between the target user u and a candidate hashtag h (e.g., cosine similarity between u's profile and h's profile, and sum prestige of users who have used h). A SVM classifier with the RBF kernel is used for prediction. This baseline only considers users' preference.

Baseline+. This model combines GraphRec and UserLevelRec, which uses node weights computed in GraphRec as additional features in UserLevelRec. This baseline considers both tweets' content and users' preference.

To have a deeper insight into our model, we decompose our model into several parts:

Content-based. This model considers all content related features, i.e., terms, links, and mentions as discussed in Section V-A.

User-based. This model considers all user related features, which include personal preferences, social relations, and locations as discussed in Section V-B.

Hybrid. This model is a combination of Content-based and User-based without hashtag features.

Hybrid+. Based on Hybrid, this model further incorporates hashtag specific features (e.g., temporal characteristics) discussed in Section V-C. This is our final model.

D. Overall Results

The overall results are shown in Figure 7 and Table VIII.

Analysis of Results on "Week-Day" Dataset. First we analyze the performance of the baselines. We have the following observations: (1) Since TensorFac considers both content and user preference, it is better than both Content-based and User-based. However, it is slightly worse than Baseline+. (2) UserLevelRec is surprisingly better than GraphRec. By intuition, UserLevelRec should not be so effective since it recommends the same hashtags for all tweets. However, the

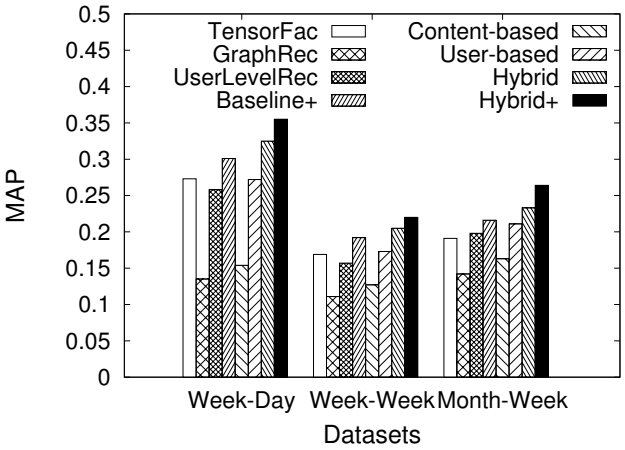


Fig. 7. Comparison of Different Models

test set only contains tweets for one day and users only use 1.2 hashtags on average on this day. Since interest drifting is unlikely to happen in such a short time, recommending hashtags most preferred by the user or her/his neighbors is still a good strategy. On the other hand, recommending hashtags based on the content of a single tweet is still difficult since a tweet can only have 140 characters at most. (3) Since Baseline+ considers both content and user preferences, it achieves the best performance among all the baselines.

Now we discuss the results of Content-based and User-based. We have the following observations: (1) Although GraphRec and Content-based are both purely content-based models, Content-based is slightly better than GraphRec. This is mainly because Content-based makes use of two more indicators, i.e., web links and mentions. This indicates that auxiliary information can help improve the performance. (2) User-based is slightly better than UserLevelRec. This is mainly because latent factor model is very effective in measuring the relevance between two objects. Compared with UserLevelRec, User-based further makes use of Twitter-specific features like social relation and user locations. (3) Like results of GraphRec and UserLevelRec, User-based is surprisingly better than Content-based, which again proves that user relevance is an effective factor in predicting hashtag adoptions.

Finally, we analyze the results of Hybrid and Hybrid+. Since Hybrid is a combination of Content-based and User-based, it is not surprised to see that Hybrid has a better performance than both its individual components. This proves our assumption that recommended hashtags should be both user-relevant and content-relevant. By further considering temporal patterns of hashtag adoption, the performance of hybrid+ is again improved and has successfully outperformed all baselines. This proves that it is very effective to combine the numeric feature-based linear discriminative model with the ID feature-based latent factor models.

Analysis of Results on "Week-Week" Dataset. Instead of predicting for the next day on the "Week-Day" dataset, we are predicting for the next week with the same training data.

TABLE VIII
COMPARISON OF DIFFERENT MODELS IN MAP. THE FIRST FOUR METHODS ARE BASELINES.

Method	Week-Day	Week-Week	Month-Week
TensorFac	0.273	0.169	0.191
GraphRec	0.135	0.111	0.142
UserLevelRec	0.258	0.157	0.178
Baseline+	0.301	0.182	0.216
Content	0.154	0.127	0.163
User	0.272	0.173	0.211
Hybrid	0.325	0.205	0.233
Hybrid+	**0.355**	**0.220**	**0.264**

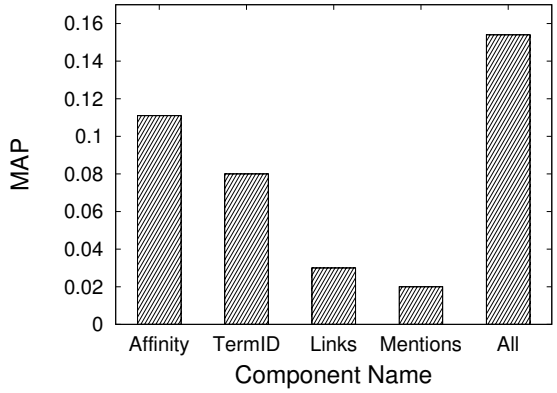


Fig. 8. Performance of Each Component in Content-based Methods

Comparing the results between two datasets, we have the following observations: (1) The performance on "Week-Week" is generally worse than that on the "Week-Day" dataset. This indicates users' interest may get drifted during a week (e.g., users may pay attention to some new events happened in that week instead of old events). Thus predicting for the next week is more difficult then predicting for the next day. (2) The gap between content-based and User-based is smaller in the "Week-Week" dataset. As time passes by, User-based becomes less reliable since the test set now contains seven days instead of only one day. In this case, recommending content-relevant hashtags becomes more and more important.

Analysis of Results on "Month-Week" Dataset. Compared with the "Week-Week" dataset, the training set is expanded from one week to one month. We find that the performance on this dataset is generally better than that on the "Week-Week" dataset. This indicates more training data can lead to more reliable recommendations. Content-based has more data to measure the relevance between hashtags and tweets and User-based is supported by richer user interaction histories.

To summarize, our method successfully outperforms existing methods on three datasets by combining explicit numeric features and latent factors into a single model.

E. Analysis of Each Component

In this section, we identify the most valuable features in Content-based and User-based by building models on each feature alone on the "Week-Day" dataset.

Analysis of Content-Based. Content-based consists of four components: affinity score, factorization on term-hashtag relation, factorization on link-hashtag relation, and factorization on mention-hashtag relation. The results are shown in Figure 8. As we can see, affinity is the most effective feature. Remind that affinity is calculated based on the co-occurrences of hashtags and terms, which is considered to be more reliable. By introducing latent factors, factorization on term-hashtag makes it possible to measure the relevance between any pair of <term, hashtag>. However, when all the hashtags have relevance scores, the results become less reliable than the results given by affinity. Links and mentions have poor performance when used alone. This is mainly because only 50% tweets contain links and 30% contain mentions. And hashtags for these links change over time. So they can only be used as weak indicators. Finally, when all the features are combined together, the performance is further improved.

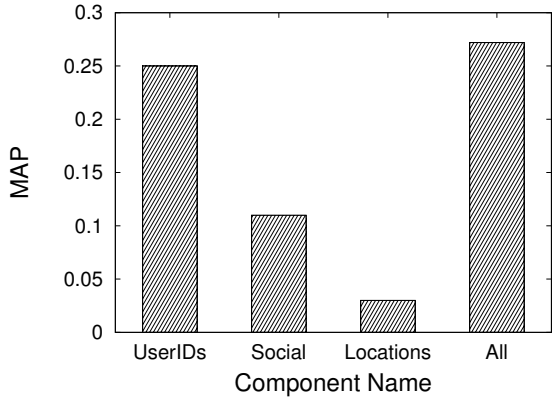


Fig. 9. Performance of Each Component in User-based Methods

Analysis of User-Based. User-based contains three components: factorization on user-hashtag relation, factorization on neighbor-hashtag relation, and factorization on location-hashtag relation. The results are shown in Figure 9. As we can see, factorization on user-hashtag relation contributes most to the model. It represents the preference of the owner. Representing users by her/his neighbors, factorization on neighbor-hashtag relation is the second effective component. This indicates that users' preferences can be partially inferred from their neighbors. By factorization on location-hashtag relation, we can recommend hashtags that are preferred by a specific location. However, it does not consider user information nor tweet information. Thus this is a weak indicator. It is not surprised to see factorization on location-hashtag gives the worst performance among all the components. Finally, when all the components are combined, we can get a better performance.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have studied the problem of personalized hashtag recommendation, which suggests both content-relevant and user-relevant hashtags when users are composing tweets. Our work can help users reach consensus on hashtag adoptions, which can further control the hashtag explosion. In the meanwhile, our personalized recommendations also help users maintain their own classification system. To achieve this, we explored (1) content-related features including terms,

links, and mentions; (2) user-related features including user IDs, locations, and social relation; and (3) hashtag-related features which describe temporal characteristics of hashtag adoption. Finally, we proposed a unified model to incorporate both latent factors and explicit features, which is proved to be effective on the real dataset crawled from Twitter. We find that recommending hashtags that are both content-relevant and user-relevant achieves the best performance. User-related features are found to be particularly effective when we are predicting for the next day. On the other hand, limited by length of tweets, content-related features are found to be less effective than user-related features. Hashtag-related features are found to be effective in all three datasets, which indicates that modeling temporal patterns of hashtag adoptions plays an important role in the final model.

Our work is an initial study of personalized hashtag recommendation. It can be extended in many directions. Firstly, a new hashtag that does not exist in the training set cannot be recommended in the test set. Our work addresses the problem by updating the model at a higher frequency (i.e., by day). However, a more elegant solution is to develop an online learning algorithm to extract new hashtags based on the tweet content, in which case novel hashtag extraction techniques are further needed. Secondly, many recommendation tasks in Twitter such as tweet recommendation, location recommendation, and user attribute inference, share the same framework with our work (that is, recommendation based on numeric features and ID features). Since all these tasks can be considered as different aspects of user modeling, they may reinforce each other when they are trained jointly with a shared framework. (3) Our work can also be extended to topic recommendation, which aims at finding interesting topics from the Twitter stream for users.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grant No. 61272088 and National Basic Research Program of China (973 Program) under Grant No. 2011CB302206.

REFERENCES

[1] H. Kwak, C. Lee, H. Park, and S. B. Moon, "What is twitter, a social network or a news media?" in *WWW*, 2010, pp. 591–600.

[2] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang, "Discover breaking events with popular hashtags in twitter," in *CIKM*, 2012, pp. 1794–1798.

[3] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "Pet: a statistical model for popular events tracking in social communities," in *KDD*, 2010, pp. 929–938.

[4] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *SIGMOD Conference*, 2010, pp. 1155–1158.

[5] M. Efron, "Hashtag retrieval in a microblogging environment," in *SIGIR*, 2010, pp. 787–788.

[6] L. B. Jabeur, L. Tamine, and M. Boughanem, "Uprising microblogs: a bayesian network retrieval model for tweet search," in *SAC*, 2012, pp. 943–948.

[7] O. Tsur and A. Rappoport, "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities," in *WSDM*, 2012, pp. 643–652.

[8] J. Yang and J. Leskovec, "Modeling information diffusion in implicit networks," in *ICDM*, 2010, pp. 599–608.

[9] A. L. Brooks and C. Cheshire, "Ad-itudes: twitter users & advertising," in *CSCW (Companion)*, 2012, pp. 63–66.

[10] Y.-M. Li and Y.-L. Shiu, "A diffusion mechanism for social advertising over microblogs," *Decision Support Systems*, vol. 54, no. 1, pp. 9 – 22, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923612000826

[11] E. Wright, N. M. Khanfar, C. Harrington, and L. E. Kizer, "The lasting effects of social media trends on advertising," *Journal of Business & Economics Research (JBER)*, vol. 8, no. 11, 2010. [Online]. Available: http://cluteonline.com/journals/index.php/JBER/article/view/50

[12] L. Weng, A. Flammini, A. Vespignani, and F. Menczer, "Competition among memes in a world with limited attention," *Scientific Reports*, vol. 2, no. 335, 03/2012 2012. [Online]. Available: http://www.nature.com/srep/2012/120329/srep00335/full/srep00335.html?WT.mc_id=FBK_SciReports

[13] E. Khabiri, J. Caverlee, and K. Y. Kamath, "Predicting semantic annotations on the real-time web," in *HT*, 2012, pp. 219–228.

[14] L. Yang, T. Sun, M. Zhang, and Q. Mei, "We know what @you #tag: does the dual role affect hashtag adoption?" in *WWW*, 2012, pp. 261–270.

[15] W. Feng and J. Wang, "Incorporating heterogeneous information for personalized tag recommendation in social tagging systems," in *KDD*, 2012, pp. 1276–1284.

[16] R. Jäschke, L. B. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag recommendations in folksonomies," in *PKDD*, 2007, pp. 506–514.

[17] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang, "Personalized tag recommendation using graph-based ranking on multi-type interrelated objects," in *SIGIR*, 2009, pp. 540–547.

[18] S. Rendle, L. B. Marinho, A. Nanopoulos, and L. Schmidt-Thieme, "Learning optimal ranking with tensor factorization for tag recommendation," in *KDD*, 2009, pp. 727–736.

[19] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *WSDM*, 2010, pp. 81–90.

[20] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 2, pp. 179–192, 2010.

[21] Y.-T. Lu, S.-I. Yu, T.-C. Chang, and J. Y. jen Hsu, "A content-based method to enhance tag recommendation," in *IJCAI*, 2009, pp. 2064–2069.

[22] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[23] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: model and implications," in *KDD*, 2012, pp. 6–14.

[24] K. Nishida, T. Hoshide, and K. Fujimura, "Improving tweet stream classification by detecting changes in word probability," in *SIGIR*, 2012, pp. 971–980.

[25] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," in *SIGIR*, 2010, pp. 841–842.

[26] J. Lehmann, B. Gonçalves, J. J. Ramasco, and C. Cattuto, "Dynamical classes of collective attention in twitter," in *WWW*, 2012, pp. 251–260.

[27] Z. Ma, A. Sun, and G. Cong, "Will this #hashtag be popular tomorrow?" in *SIGIR*, 2012, pp. 1173–1174.

[28] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[29] S. Rendle, "Factorization machines with libfm," *ACM TIST*, vol. 3, no. 3, p. 57, 2012.