Collaborative Personalized Tweet Recommendation

Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao and Yong Yu
Dept. of Computer Science and Engineering, Shanghai Jiao Tong University
No. 800, Dongchuan Road, Shanghai, China 20024
chenkl,tqchen,gqzheng,kingohm,yaoenpeng,yyu @apex.sjtu.edu.cn

ABSTRACT

Twitter has rapidly grown to a popular social network in recent years and provides a large number of real-time messages for users. Tweets are presented in chronological order and users scan the followees' timelines to find what they are interested in. However, an information overload problem has troubled many users, especially those with many followees and thousands of tweets arriving every day. In this paper, we focus on recommending useful tweets that users are really interested in personally to reduce the users' effort to find useful information. Many kinds of information on Twitter are available for helping recommendation, including the user's own tweet history, retweet history and social relations between users. We propose a method of making tweet recommendations based on collaborative ranking to capture personal interests. It can also conveniently integrate the other useful contextual information. Our final method considers three major elements on Twitter: tweet topic level factors, user social relation factors and explicit features such as authority of the publisher and quality of the tweet. The experiments show that all the proposed elements are important and our method greatly outperforms several baseline methods.

Categories and Subject Descriptors

H.4.0 [**Information Systems**]: Information Systems Applications—*General*

Keywords

Twitter, Tweet Recommendation, Collaborative Ranking, Personalization

1. INTRODUCTION

Twitter is the most popular microblogging service and an important social network with over 200 million users as of 2011. It allows users to share information with their friends or the public by posting text messages of up to 140 characters, which are called tweets or statuses. Most users update their Twitter messages frequently and over 200 million tweets are generated per day. Users can access a filtered timeline of tweets from specific persons by explicitly following them. The retweet mechanism allows users to share information with their followers and accelerate the spread of information in the social network. On Twitter, users follow celebrities, friends or anyone else they care about and benefit greatly from the fresh information from these followees.

However, as a result of the rapidly increasing number of tweets, most Twitter users encounter a serious problem of information overload. It has been reported that Twitter users follow 80 people on average [22], which leads to hundreds or even thousands of tweets posted to each user every day. This bothers the active users more seriously, because they may have many more followees than regular users. The freshness of tweets is considered most important and all followees' tweets are shown to users in chronological order. Informative and useful tweets for the user may be flooded by other tweets that the user does not care about at all. For example, some followees update travel information so frequently that their tweets may prevent users from seeing important tweets such as international news. It is rather inconvenient for users to scan all the tweets posted in the last several hours or days to find useful information. A basic assumption on Twitter is that all the followees' statuses are considered equally important and they are posted to all followers sequentially. However, this assumption fails under many circumstances because users may consider only some specific aspects of the followees worth their attention.

Recommending useful tweets to a user is an important challenge and the focus of this paper. Intuitively, a tweet is useful to a user, if the user is interested in or willing to read the tweet. Whether a user is interested in a tweet is determined by many factors, such as the quality of the tweet and the authority of the publisher . Personal interest is also an important factor to decide whether a tweet is personally useful. Traditional methods analyze the content of users' posted or retweeted statuses to discover the topics of interest for Twitter users. However, profiling users' personal interests in this way is very difficult. Most users' personal preferences are related to tweets, which are short, informal, ungrammatical and noisy. Directly applying text mining and analyzing techniques designed for and tested on a traditional corpus of long text often leads to poor results [17].

Besides the user's own tweet history, there are many other kinds of important information available on Twitter. For

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
Copyright 2012 ACM 978-1-4503-1472-5/12/08 ...$15.00.

example, the retweet history of other users can be used with the current user's tweet history to improve recommendation. In another direction, social relations between users can also greatly influence behavior. In order to fully utilize such information, we propose a framework based on collaborative ranking to capture the personal interests. Collaborative filtering and collaborative ranking are promising technologies for recommender systems. Collaborative ranking is a ranking version of collaborative filtering and provides item ranking results according to relative preference rather than user rating estimation. It works by discovering the correlation between users and items based on observed user preferences so that unobserved user preferences can be inferred from the observed ones. In personalized tweet recommendation, tweets are regarded as items, and the preferences of users on the tweets are the correlation between users and items. Our approach generalizes the traditional collaborative ranking approach by incorporating Twitter features such as content information, and social relation information, so our model fully utilizes the information mentioned on Twitter and can do better personalized recommendations.

The major contributions of the paper are as follows: (1) We use topic level latent factors of tweets to capture users' common interests over tweet content, which helps us to solve the problem of information sparsity in users' retweet actions. This allows us to adjust the collaborative filtering technique to solve the recommendation problem. (2) We introduce latent factors to model users' social relations, which greatly affect users' decisions in a social network. (3) Our model incorporates explicit features such as authority of the publisher and quality of the tweet, which can help further improve the recommendation results. The experimental results show the listed contributions can help improve tweet recommendation performance, and the combination of all the elements makes up a final model that greatly outperforms several baseline methods.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. Some preliminaries to our work are introduced in Section 3. In Section 4 we describe the collaborative ranking method for tweet recommendation. Section 4.2 introduces the features of Twitter incorporated in our model. Experimental results are presented in Section 5. Finally, we conclude the paper in Section 6.

2. RELATED WORK

2.1 Recommender Systems on Twitter

As Twitter has become a popular social medium and had great impact, plenty of research focuses on analyzing the personal interest of users and building recommender systems according to that. Wu introduces a system to generate personalized tags for Twitter users to label their interest by extracting keywords from tweets [33]. Twopics [19], an entity-based profiling approach, aims to discover the topics of interest for Twitter users by examining the entities they mention in their Tweets. Recommending influential Twitter users is also a major research direction of Twitter recommender systems. Kwak [13] proposes three methods to estimate the influence of users on Twitter: the number of followers, PageRank and the number of retweets. Twittomender [9, 8] builds a followee recommender system for Twitter by profiling Twitter users by tweets and the relationships of their Twitter social graphs. TwitterRank [32], an extension of Pagerank, measures the influence of users in Twitter taking both the topical similarity between users and the link structure into account.

Some recommendation systems based on tweet recommendations have been proposed. Chen [4] studied URL recommendation on Twitter as a means to better direct user attention in information streams. Sun [31] constructs a diffusion graph to select a small subset of micro-blogs as recommended emergency news feeds for regular users. Duan [7] studies three types of features and proposes a tweet ranking strategy by applying a learning to rank algorithm. Naveed [20] used a learning approach based on pure content features to predict the probability of a message being retweeted. However, their work does not consider personalization and suggests tweets by considering only tweet quality and authority of the publisher. Also, their framework calculates the relevance between tweets and users' queries for retrieval, which requires a query to be provided in order to do recommendation. Kapanipathi [11] presents a Semantic Web approach to filter public tweets matching interests from personalized user profiles based on the generated profiles. Ramage [23] presents a scalable implementation of Labeled LDA that maps the content of the Twitter feed into dimensions. This helps improve methods for finding new users and for filtering tweets in the meanwhile. However, all their methods use only content based recommendation methods for suggesting tweets without considering the collaborative view. Moreover, they do not consider social relations between users in their approach. Cui [5] proposed a matrix factorization approach for item-level social influence modeling and devised an efficient projected gradient method to solve it. Yang [34] established a joint friendship-interest propagation model, FIP, to address recommendation and link prediction in a unified framework. But their work does not integrate specific characteristics of the Twitter platform like asymmetric social relations and is not trivial to adapt to tweet recommendation. Zaman [35] uses Matchbox, a large scale online Bayesian recommendation method to predict retweets. However, their method does not incorporate global features which are independent of users and items and more flexible for prediction. Also they do not take advantage of relative preferences to adapt to the ranking scenario of recommending tweets naturally. Our method takes advantage of relative preferences and collaborative ranking to learn user preferences based on the Twitter community data, and integrates tweet content, user social relations and other explicit features like publisher authority to greatly improve the personalized recommendation performance.

2.2 Collaborative Filtering and Collaborative Ranking

Collaborative filtering aims to build a recommender system by learning users' preferences based on the community data. In contrast to content-based methods, it does not require creation of explicit user profiles and specific domain knowledge. In domains such as movies, news, and advertisements, the content of documents is so short and limited that users' preferences cannot always be explicitly described using the terms or topics extracted by analyzing the content. So much research uses collaborative filtering for recommender systems. In recent years, collaborative filtering has attracted great attention and has been implemented in many large commercial systems, such as Google News [6] and Amazon [15]. Collaborative Filtering can be broadly divided into two categories, neighborhood-based approaches [28, 29] and model-based approaches [12, 24]. The former recommends items from the perspective of similarity of its neighbors, while the latter emphasizes latent factors using a probabilistic latent factor model or matrix factorization methods to solve problems.

Based on these widely used approaches, collaborative ranking, a ranking-oriented collaborative filtering method, is used to adapt the learning to rank scenarios. Collaborative ranking predicts a preference ordering over the yet unrated items by using a collaborative filtering method. Just like the pairwise models of learning to rank [3, 10], Pessiot [21] proposed a pairwise preference error minimization framework to optimize the item ranking. Rendle [25] proposed a Bayesian probabilistic ranking model. In the work of EigenRank[16], first user-user similarity is measured by the correlation between their rankings.

In another direction, some external information besides user-item interaction such as contextual information, user profiles and item descriptions is involved to improve recommendation. Stern [30] presents Matchbox, a large scale recommender system integrating meta data features to deal with cold start problem. Agarwal [1] proposes a regression-based latent factor model simultaneously incorporating features and past interactions. Rendle [26] proposes a context-aware factorization machine to take contextual information into account to enhance predictions. Li [14] proposes to leverage information extraction techniques to help recommender systems to train better context-aware recommendation models. In our approach, we explore the specific features of Twitter and integrate the significant ones into our collaborative ranking model to improve recommendation performance.

3. PRELIMINARY

3.1 Personal Interest Indicator

The goal of personalized tweet recommendation is to estimate the value of a tweet for each user. On Twitter, users follow many celebrities, friends, media organizations and other information sources. Tweets posted by all the publishers are pushed to users without any filtering. However, not all the information in these tweets caters for the users' personal interests, and not every aspect of his followees is worth paying attention to. For example, many people follow Obama in order to learn policies released by the White House. However, sometimes Obama will publish tweets about his interviews in other countries, which can not catch the interests of these followers. For them, a tweet recommender system should suggest the political tweets of Obama with higher priority than others.

How to detect personal interest is the key problem to be solved in this paper. There are three kinds of important user behaviors on Twitter: following another user, publishing a tweet, and retweeting a status posted by a followee. By analyzing these user behaviors, we are able to find users' personal interests. In the example mentioned above, if a person is interested in economic policy, it is more likely that he will follow some economic commentators or that tweets discussing economics will be published or retweeted by him. In our method, this information is represented as features and there is substantial evidence that the features predict a tweet's utility for a particular user.

However, the informativeness and usefulness of a tweet for a user cannot be directly measured by analyzing the tweet or the user's public Twitter profile. The main obstacle which raises the difficulty of evaluation for personalized tweet recommendation is that we must have enough user-specific relevant judgement data. These data for a particular user can only be decided by the user himself in the user experience study. In some work, Twitter users were asked to rate the quality of posts from users they follow [23]. However, this requires a great deal of time and effort, and becomes infeasible when the data set is large. Fortunately, some user actions indirectly reflect users' judgements of the usefulness. These actions include replying to a tweet, retweeting followees' statuses and tagging tweets as favorites. Because getting details of replying and tagging actions need the users' authorization, we only consider the retweeting action. For evaluation, we make an assumption below:

Assumption *Users' retweeting actions reflect their personal judgement of informativeness and usefulness.*

Intuitively, if a tweet is retweeted by a user, it means that the user has carefully read the tweet rather than just taking a short glance at it. Retweeting is a rough measure of personal usefulness. Of course, it's also true that a lack of retweeting does not necessarily mean the user is not interested in the tweet. However, to the best of our knowledge, this is a common problem for evaluation approaches for personalized tweet recommendation without user-specific judgement data.

We consider tweet recommendation in relation to the personal interests of a user. Given a collection of posted tweets by followees, we try to rank these tweets in order to put tweets which the user is likely to be interested in on top.

3.2 Why Collaborative Ranking

One important assumption underlying collaborative ranking and collaborative filtering is that users who agreed in the past are likely to agree again in the future [27]. This assumption allows us to take user preference histories and predict items which they might enjoy by analyzing similar users. In personalized tweet ranking, we can also make a similar assumption:

Assumption *Users who have retweeted similar statuses in the past are likely to retweet similar statuses in the future.*

This assumption makes collaborative ranking applicable to personalized tweet ranking. Collaborative ranking has advantages over content-based methods when it is difficult to analyze the content of tweets. Semantic relatedness of tweets which is not detected by content-based methods can be inferred using collaborative ranking. For example, a user interested in economic policies may have retweeted many statuses of president Obama. With a content-based method, it is difficult to be aware of this user's interest in other economic commentators' tweets when we lack sufficient word co-occurrence information for calculating tweet similarities. However, by analyzing other users' tweets, it can be found that people who have retweeted Obama's statuses are more likely to retweet statuses of other economic commentators.

And this implicitly detects users' probable interest in economics without analyzing the content of tweets. To adapt to the scenario that rank results of tweets are presented to users, a collaborative ranking method is better than collaborative filtering for different optimization criterion.

4. COLLABORATIVE RANKING METHOD FOR TWEET RECOMMENDATION

4.1 Optimizing Ranking Criterion for Tweet Recommendation

Here we introduce optimizing the ranking criterion for personalized tweet recommendation. Collaborative ranking is an extension of the latent factor model with ranking optimization criterion . In the latent factor model, each user u and item i have a low dimensional representation $p_u \in R^d$ and $q_i \in R^d$ in the latent feature space. The rating score is predicted by measuring the affinity between user and item:

$$\hat{y}_{u,i} = \mu + b_u + b_i + p_u^T q_i \tag{1}$$

Where y is the predicted preference of user u for item i. μ is the overall average rating, and b_u and b_i are user bias and item bias on rating score.

To adapt to the scenario of tweet ranking, we modify the model for the collaborative ranking setting according to ranking optimization criterion. Give a user u and two tweets k and h the pairwise ranking model for tweet preference is defined as follows:

$$P\left(r(k) > r(h)|u\right) = \frac{1}{1 + e^{-(\hat{y}_{u,k} - \hat{y}_{u,h})}} \tag{2}$$

Where $\hat{y}_{u,k}$ is the predicted preference of user u for tweet k, r is short notation for rank order. Equation 2 models the probability of item pairs' rank orders for a given user. We can get preference pair of items for a given user by assuming a user prefer the tweets he retweeted to the rest of tweets. Formally, we define rank preference set $\mathcal{D}$ as follows:

$$\mathcal{D} = \{< u, k, h > | k \in Re(u), h \notin Re(u)\} \tag{3}$$

Where $Re(u)$ is the set of tweets user u retweeted. Because the number of possible choices of negative sample h is large, we use sampling techniques to get negative samples in the training procedure. To learn the model, we maximize the log-likelihood over the $\mathcal{D}$ to estimate parameters. The maximizing procedure is converted to solve the objective below:

$$\min \sum_{<u,k,h>\in\mathcal{D}} \ln\left(1 + e^{-(\hat{y}_{u,k} - \hat{y}_{u,h})}\right) + \text{regularization} \tag{4}$$

L2 regularization is used as the default choice of regularization term. It can be consider as involving the prior probability for model parameters in a Bayesian view.

4.2 Incorporating Twitter Features for Recommendation

In tweet recommendation, tweets are regarded as items and rated by users according to usefulness. When users and tweets are characterized by a single latent factor, it is difficult to capture the details such as content of tweets, or followees of users. This leads to extreme sparsity of information for analyzing users and tweets, which is the main obstacle for recommendation. In our model, user and item factors can be decomposed into combinations of latent factors of related components, such as words, publisher, followee, etc. This helps involve all the related information in a latent factor model. On the other hand, we also try to incorporate contextual information into the system. Some user-independent variables, such as length of tweets and the number of times something is retweeted, can affect the preferences independent of the personal interests of the user. User-dependent contextual variables contain the number of times the publisher is mentioned and the similarity between tweet content and user profile. These contextual variables are converted into explicit features in our model. We can enrich user features to better describe user preferences, and add more item features to describe tweets' properties. Explicit features are directly related to users' preference for the tweets because the describe the contextual information.

In the following sections we describe the latent factor features and explicit features for tweet recommendation in our model. They will influence the recommendation through the factorization part. Explicit features are elements that contribute to the bias effect of users' decisions. Contextual information can be encoded as explicit features to improve recommendation.

4.3 Topic Level Decomposition of Tweets

As discussed in the previous section, directly applying the basic latent factor model to tweet recommendation encounters a serious problem of data sparsity due to the lack of retweet data. We use information on the content of tweets to solve this problem. A tweet is mainly composed of several words at the content level. We decompose the latent factors of a tweet into a combination of latent factors of words, to get the following factorization model:

$$\hat{y}_{u,i} = bias + p_u^T \left(\frac{1}{Z}\sum_{w\in T_i} q_w\right) \tag{5}$$

Here $bias$ is used to indicate any form of possible bias to simplify the equation, T_i is the term set of tweet i, Z is the normalization term for features. This strategy actually converts the problem to asking whether the user is interested in the words or topics, rather than directly asking whether he is interested in the tweet. This decomposition gets more opinions from users at topic level, which helps collaborative filtering based methods get better performance. Moreover, at topic level we do not have to collect a large set of user data to have enough opinions on tweets. This helps reduce the number of users involved in the collaborative ranking algorithm and makes the process more scalable.

4.4 Incorporating Social Relations

Besides considering the content of tweets, we also try to incorporate the social relation between the user and the publisher in our model. Intuitively, if a user's interests are similar to a publisher's tweet topics, the user is more likely to retweet the publisher's tweets. So we measure the possibility of retweeting a status by considering the affinity of the user and the publisher's latent factors. Equation 6 gives a model that takes social relation into account.

$$\hat{y}_{u,i} = bias + p_u^T d_{p(i)} \tag{6}$$

$d_{p(i)}$ is a latent factor of the publisher of tweet i. This strategy incorporates the prior probability of retweeting a specific followee's statuses without considering the content of the tweet. This is reasonable under certain circumstances. For instance, a fanatical basketball fan may retweet any statuses of LeBron James even when he does not read the content. This also caters for the simple assumption that a user is more likely to retweet the statuses of a publisher who he has retweeted many times before. Furthermore, we can incorporate social relation information together with tweet content information. The modified model can be shown by the following equation.

$$\hat{y}_{u,i} = bias + p_u^T \left(\frac{1}{Z} \sum_{w_j \in T_i} q_{w_j} + \alpha d_{p(i)} \right) \tag{7}$$

Where λ indicates the importance of the publisher's latent factor relative to the words'.

4.5 Explicit Features

Besides the latent factors, information such as tweet quality, and publisher authority can be indicated as features. These features explicitly reflect the possibility of retweeting a status. We use a linear combination of these features to get retweet prediction:

$$\hat{y}_{u,i} = \sum_j b_j \gamma_j \tag{8}$$

Where b is a weight parameter vector and γ is an explicit feature we extracted. We further incorporate these explicit feature into the latent factor model described in Equation 7, to get our final predictor:

$$\hat{y}_{u,i} = \sum_j b_j \gamma_j + p_u^T \left(\frac{1}{Z} \sum_{w_j \in T_i} q_{w_j} + \alpha d_{p(i)} \right) \tag{9}$$

We can incorporate explicit features into our framework by redefining γ. In the following subsections, we will describe in detail the explicit features we use. The features are divided into four categories according to their sources.

4.5.1 Relation Features

Relation features refer to those features which represent the relation between the user and the publisher. A user is likely to prefer tweets posted by specific users, such as their intimate friends, celebrities, or people who share common interests. Relation features describe the relationship between the user and the publisher by analyzing their communication and social networks.

- Co-follow Score: This feature estimates the similarity of followee sets of the user and the publisher. If two users follow many identical publishers, it is likely that they share many common interests. Given a user U and a publisher P we use Jaccard similarity to measure this score:

$$Jaccard(U, P) = \frac{|Followee(U) \bigcap Followee(P)|}{|Followee(U) \bigcup Followee(P)|} \tag{10}$$

  where Followee(U) refers to the followee set of user U.

- Mention Score: This feature measures the number of times user U has mentioned publisher P in his previous tweets. If U has mentioned P many times before, it means that P gets much attention from U and U is likely to consider P's tweets useful.

- Friend: On Twitter, it is often observed that two users follow each other [32]. They are considered to be friends because they may know each other well. This type of relationship is like the ones in social networks like Facebook. This feature is binary. It is 1 when the publisher of a tweet is a friend of the follower and 0 otherwise.

4.5.2 Content-relevance Features

Content-relevance features measure the relevance between a tweet and a user by analyzing the content of tweets. The statuses posted or retweeted by a user previously reveal his personal interests to a degree. If a tweet is similar to a user's posts, it is likely that the user will consider the tweet to cater for his concerns.

- Relevance to Status History: This feature estimates the relevance between a tweet T and the posting history of a user U. We calculate similarity scores between T and every single post in the user's history and sum the scores to get the relevance score:

$$Relevance(T, U) = \sum_{T_i \in Tweets(U)} TweetRel(T, T_i) \tag{11}$$

  Tweets(U) represents the set of tweets posted by U. $TweetRel(T, T_i)$ measures the relevance between two tweets. Here we represent the tweets as vectors by term frequency and calculate the inner product as the result.

- Relevance to Retweet History: This feature estimates the relevance between a tweet T and the retweeted history of a user U. Similarity scores between T and every post retweeted by U are calculated and summed to get the relevance score:

$$Relevance(T, U) = \sum_{T_i \in Retweets(U)} TweetRel(T, T_i) \tag{12}$$

  ReTweets(U) refers to the set of tweets retweeted by U.

- Relevance to Hash Tags: Tweet Publisher is able to insert hash tags to emphasize the key words of a tweet. Intuitively, these hash tags are a summary of a user's personal interests. This feature estimates the count of words ever appeared as hash tags through the user's posting history for a tweet.

4.5.3 Tweets' Content-based Feature

Some content-based features can be used to measure the quality and popularity of a tweet. Based on the assumption that a user will prefer a high-quality tweet without considering personal interests, we introduce these features to predict whether a user considers a tweet useful.

- Length of Tweet: This feature estimates the number of terms in a tweet. A long tweet is more likely to be an informative one because it contains more information than a short one. Intuitively speaking, the user has put more effort into and spent more time editing longer tweets. This may help the tweet become popular and get more attention.

- Hash Tag Count: if a tweet has inserted hash tags, it is considered to be more informative and useful. Intuitively, the publisher spends time on tagging the tweet because he thinks the tweet may be useful. The number of hash tags in a tweet is measured by this feature.

- URL Count: On Twitter, publishers often include a URL as a supplement in their tweets. This is because the publisher cannot summarize their information in 140 characters and they use a URL to point to fuller information on another web site. The number of URLs in a tweet is estimated by this feature.

- Retweet Count: Twitter records the number of times a tweet has been retweeted. A tweet retweeted more times is more likely to be a useful one, because many other people have suggested the tweet and recommended it to their followers. This feature is an objective way to estimate the popularity of a tweet because the evaluation is from the general public. It is often used in other recommender systems as substantial evidence of the utility of a tweet and the authority of a user.

4.5.4 Publishers' Authority Feature

Intuitively speaking, a user is likely to prefer a tweet published by an authoritative user over others. Moreover, if a tweet is published by an authoritative user, it is more likely to be a high-quality tweet. Some features are substantial evidence of a user's authority. They can also be used to predict the popularity of a tweet.

- Mention Count: This feature estimates the times the publisher is mentioned in tweets. If a publisher has been mentioned frequently, it means he is influential and popular on Twitter.

- Followee Count: This feature records the number of people who follow the publisher. It is an objective measure of the popularity of a user based on public opinion.

- Follower Count: Users who have more followers are likely to be active on Twitter. Intuitively, active users will get more attention than inactive ones.

- Status Count: This feature measure the number of tweets ever posted by a user. Similar to follower count , this feature is also a substantial measure of activeness.

4.6 Parameter Estimation

A local minimum of the objective function given by Eq.4 can be found by performing stochastic gradient descent. To learn the collaborative ranking model, once we get a tuple $< u, k, h >$ in preference set $\mathcal{D}$, we calculate the descent of the related parameters as below and update the parameters by moving in the direction of negative gradient:

$$\frac{\partial \mathcal{L}}{\partial p_u} = \hat{e}\left(\frac{1}{Z_s^+}\sum_{s\in k} q_s - \frac{1}{Z_s^-}\sum_{s\in h} q_s + \alpha(d_{p(k)} - d_{p(h)})\right) - \lambda_1 p_u \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial q_w^+} = \frac{1}{Z_j^+}\hat{e}p_u - \lambda_2 q_w^+, \frac{\partial \mathcal{L}}{\partial q_w^-} = -\frac{1}{Z_j^-}\hat{e}p_u - \lambda_2 q_w^- \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial d_{p(k)}} = \alpha\hat{e}p_u - \lambda_3 d_{p(k)}, \frac{\partial \ell}{\partial d_{p(h)}} = \alpha\hat{e}p_u - \lambda_3 d_{p(h)} \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial b_j} = \hat{e}(\gamma_j^+ - \gamma_j^-) - \lambda_4 b_j \tag{16}$$

$$\tag{17}$$

Here $\hat{e} = 1 - P\left(r(k) > r(h)|u\right)$ is the difference between the truth and predicted possibility of rank order. o^+ denotes the parameters for user u and tweet k while o^- denotes the parameters for user u and tweet h, where o is a placeholder. Then the algorithm loops over all the data in rank preference set $\mathcal{D}$ and updates the parameters by moving in the direction of negative gradient. This algorithm is computationally efficient and easy to implement. The complexity of the algorithm is discussed in section 5.6.

5. EXPERIMENTS

5.1 Dataset

To create the Twitter data set, we began with a randomly selected user and expanded the user-base by following their followers and followees' links. After following several steps of links, we got 8059 users in our base and downloaded all the statuses they had posted. As there is no API to directly get followees' posts for each user without authorization, we collected 1048 users who have over fifteen followees in our base and regarded the tweets posted by followees in the base as the scanned tweets of the users. The retweeted tweets are regarded as positive samples and the others are negative samples in experiments. To simulate the timeline of a user, we get one positive sample with another four negative samples from the scanned tweets and sort them in chronological order. We performed standard data preprocessing including stop word removal and stemming on the raw text. Finally we obtained a simulated scanned timeline of Twitter users in our sample; each of them had about 490 messages on average. Then the first four fifths of scanned tweets of each user is put in the training set and the others in the test set. Finally the dataset is split into a training set with 409680 tweets and test set with 102457 tweets. Figure 1 shows the number of tweets with specific retweet count in the dataset. The data is plotted on a log-log scale, the number of tweets that have not been retweeted is 102650 and is not shown in the figure due to the log-log plot style. From the figure, we see that most tweets are not retweeted or are only retweeted a few times, which shows the sparsity of the dataset. This fact coincides with the discussions in the previous sections, and motivates us to use as much information as possible to solve this problem. More discussion about sparsity is in section 5.5.

5.2 Evaluation Metrics

In the context of tweet recommendation, tweet ratings are considered binary in our scenario. Retweeting a tweet corresponds to a 1 rating, not -retweeting to a 0 rating. We use

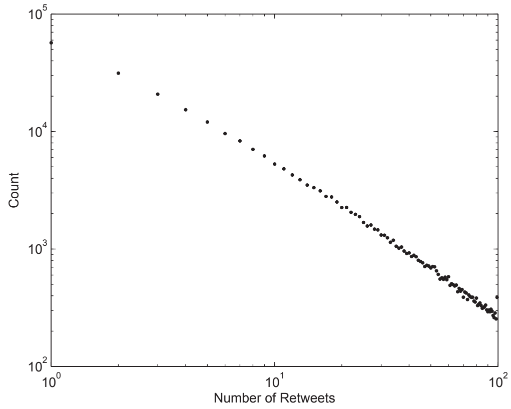

Figure 1: Sparsity of Retweets Data

Mean Average Precision (MAP), a popular rank evaluation method to evaluate the proposed approach. For a single user, average precision is defined as the average of the $P@n$ values for all retweeted tweets:

$$AP = \frac{\sum_{n=1}^{N} P@n \times retweet(n)}{|R|} \tag{18}$$

where n is the number of tweets; $|R|$ is the total number of retweeted tweets for the given user; $retweet(n)$ is a binary function to describe whether the user has retweeted the nth tweet in the result list. Finally, MAP can be obtained by averaging the AP values of all the users. $P@n$ is also used as our affiliated evaluation measure.

5.3 Method Comparison

We have compared our models to several others. The detailed implementations are listed below:

- Chronological: The tweets are ranked in chronological order. Without any interventions by algorithms, this strategy indicates the default user experience on Twitter.

- Retweeted Times: Retweeted Times is an objective estimate of the popularity of a tweet. This ranking strategy ignores personalization and assumes the user's interests are the same as general public's.

- Profiling: This ranking strategy calculates the similarity between a tweet and the user's profile and shows the tweets sorted by similarity score. Profiles are simply treated as collections of words from users' posted tweets and retweeted tweets as follows:

$$\begin{aligned} Profile(U) =& Frequency(Tweets(U)) \\ &+ w * Frequency(Retweets(U)) \end{aligned} \tag{19}$$

w measures the importance of the retweeted tweets for profiling a user, we tune the parameter to report the best result. We use Frequency(Tweets), a simple frequency count as the term weighting function, so that the profile vector can be represented as the frequency counts of the various words used in the posted and retweeted tweets. Then we calculate the inner product of the profile vector and tweet vector as the similarity measure.

- LDA: LDA[2] is a generative topic model and each document is viewed as a mixture of topics. We regard tweets as general documents here and learn an LDA model from tweets. After getting the topic distribution of each tweet, given a user U and a tweet T, the relevance score is calculated as below:

$$y_{U,T} = \sum_{T_0 \in Tweets(Followee(U))} \alpha^{\mathrm{I}(U,T_0)} D_{KL}(T_0 \| T) \tag{20}$$

Here $D_{KL}(T_0\|T)$ calculates the symmetric KL-divergence between the topic distribution of two tweets, the indicator function $\mathrm{I}(\cdot)$ is equal to 1 if the user has retweeted T_0 and equal to 0 otherwise.

- RankSVM: Using a Support Vector Machine to do ranking, the RankSVM algorithm[10] effectively integrates a bag of features in a model for learning retrieval functions. We use explicit features described in Section 4.5 and train RankSVM on the training data. Finally, the model is evaluated by the test data. The regularization parameter is tuned to 0.1 and the other parameters use the default settings.

- JOINTMF: The joint matrix factorization method jointly minimizes the loss functions of collaborative filtering and link prediction to get a better representation for users and items for prediction. It is similar to some other social recommendation methods such as FIP [34], and Sorec [18]. This method is to solve the following objective:

$$\begin{aligned} \min \lambda_y \sum_{<u,i>\in \mathrm{Y}} \ell(y_{u,i}, \hat{y}_{u,i}) + \lambda_s \sum_{<u,v>\in \mathrm{S}} \ell(s_{u,v}, p_u p_v) \\ + regularization \end{aligned} \tag{21}$$

Here Y and S are records of retweet interactions and following interactions, $y_{u,i}$ and $s_{u,v}$ are the observed ratings and follower-followee relationships in the training data. $\hat{y}_{u,i}$ is the factorization model of topic level decomposition.

- CTR: Collaborative tweet ranking model integrating topic level features, social relations and explicit features proposed in this paper. Stochastic gradient descent is used for parameter estimation. In experiments, the number of latent factors is set to 200, as the bigger number will bring little improvement on performance shown in our experiments. The normalization term Z for term factors is set as $|T_i|^{\frac{1}{2}}$ according to the experimental results.

Figure 2 shows the results of P@n and MAP on the test set. Chronological strategy gets 0.3082 MAP. Not surprisingly, as whether a user would like to retweet a status depends on his personal interests rather than on time, the performance of the chronological strategy is close to a random strategy, which can be estimated by the proportion of positive samples. Also ranking by the number of times something is retweeted performs poorly with 0.3365 MAP. This means that there is still a wide gap between personal interests and the focus of public attention, which indicates that personalization is very important on Twitter. The profiling method is a classic content-based method and gives much

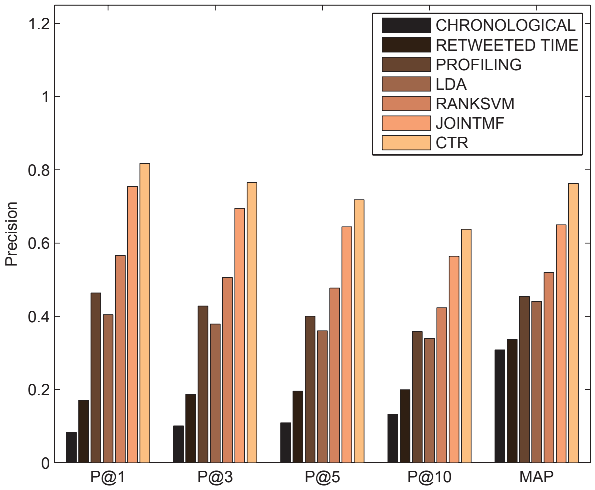

Figure 2: Evaluation Result of Compared Methods

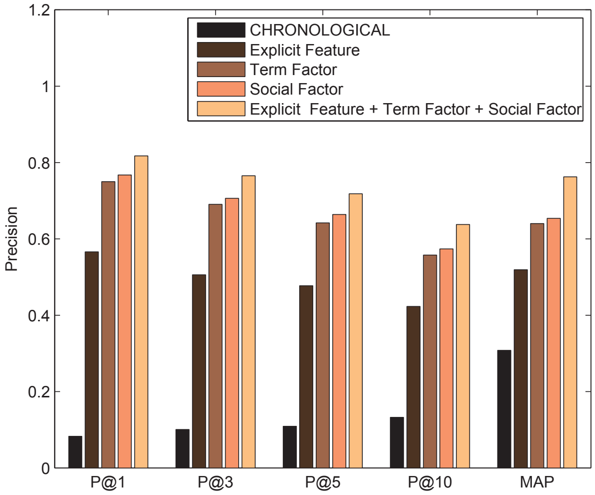

Figure 3: Comparison of CTR Components

better performance with 0.4538 MAP. The method based on an LDA topic model performs a little worse than the profiling method with 0.4408 MAP. This means that it is not a good choice to build a topic model on tweets for some specific reasons like the length of tweets. The RankSVM only considers the explicit features, and gets 0.5194 MAP. Finally, the joint matrix factorization method gets 0.6496 MAP and performs the best of the baseline methods. In comparison, the CTR model assimilates content based models by describing the information as features, and gives a 0.7627 MAP. Also, it takes advantage of collaborative filtering based methods by considering other users' opinions shared on Tweets in a global view. It achieved 46.84% and 17.41% improvements compared with RankSVM method and joint matrix factorization method in terms of MAP.

From the above results, we conclude that our proposed method gives a great improvement in recommendation performance. The result can be explained by the fact that the model includes more parameters to describe the personal interests, the attributes of tweets and user social relations,and this helps detect the detailed preferences of users.

5.4 Effectiveness of Feature Components

In the previous subsection, we have shown that our proposed CTR method greatly improves the recommendation performance. Because our CTR model consists of three major components — tweet topic latent factors, user social relation factors and explicit features — we would like to know the effectiveness of each component. We therefore conducted several experiments on CTR with only one of the components. The results are shown in Figure 3.

Figure 3 shows the P@n and MAP results of different component models. Performance of the chronological method is shown for reference. RankSVM is used to generate predictions using only the explicit features, and CTR is used to generate the predictions of the other three models. RankSVM givess comparatively low performance with 0.5194 MAP, since the model becomes a simple linear model for prediction and does not take advantage of collaborative filtering. However, it still performs better than the chronological strategy, which shows that explicit features are indeed useful. The models considering topic level factors or social factors get 0.6403 and 0.6540 MAP respectively. The two models outperform the explicit feature model by 16%. This result shows that introducing latent factors can help improve the recommendation, which supports the idea of applying collaborative filtering methods for tweet recommendation. Using topic level factors or social factors gives similar performance, which shows both parts are important to our final results. Finally, all the factors and explicit features are combined to get the highest performance with 0.7627 MAP. We conclude that all three components are effective and combining them will greatly improve tweet recommendation performance.

Matrix	User-Tweet	User-Term	User-Followee
SPARSITY	0.1059%	0.7561%	0.7829%
MAP	0.3138	0.6403	0.6540

Table 1: Observed Sparsity of Data

5.5 Data Sparsity

We also study the sparsity of our data. Traditional collaborative filtering methodsl regard the tweets as items and the retweet actions as interactions between users and items. From this point of view, the observed sparsity of the user-tweet interaction matrix is only 0.1059% and the collaborative ranking method only gets 0.3138 MAP, shown in table1. However, when the user-term interactions and user-followee interactions are taken into consideration, the data sparsity is greatly reduced. The observed sparsities of the user-term interaction matrix and user-followee interaction matrix are 0.7561% and 0.7829% respectively, which helps get better MAP performance. Our method is to alleviate data sparsity, as shown in Figure 1, by integrating all these elements and some other explicit features together to get better recommendations.

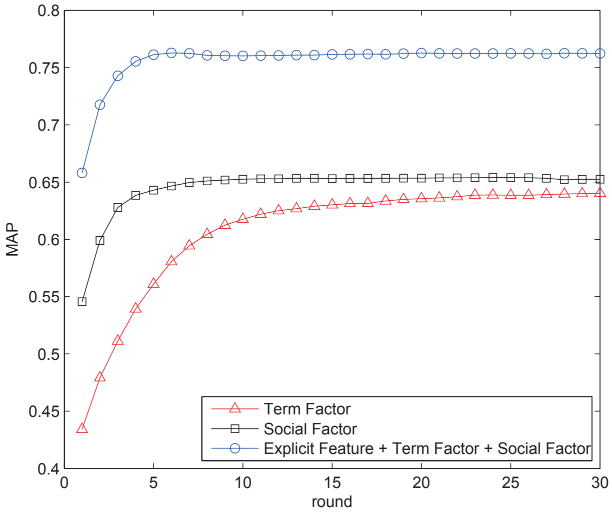

Figure 4: Runtime Convergence of CTR Models

5.6 Complexity and Runtime Convergence

In this subsection, we discuss the computation cost of the CTR model. The computational complexity of training our proposed CTR model is $O(kLNS)$. k is the number of latent factors. L is the average number of non-zero features in each training instance. N is the number of stochastic gradient descent steps. S is the amount of training data , which means the size of the rank preference set. The parameters k and L are determined by model settings, while the number of iterations N is not yet determined during model design. We can get N through runtime performance analysis. Figure 4 shows the runtime MAP of different latent factor CTR models mentioned in the previous subsection. We take ten iterations over the training set as one round, and calculate test mean average precision in each round. We find that all the models converge to a maximum value after 30 rounds. The result shows that our algorithm is stable and we only need to run the training algorithm for sufficient rounds to get the desired prediction.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a collaborative ranking model, CTR, for recommending useful tweets to Twitter users. Our approach takes advantage of collaborative filtering based recommendation by collecting preference information from many users. Moreover, extra contextual information helpful for detecting personal interests is incorporated in our model by careful design of features. Our final method makes use of tweet content, user social relations and other explicitly defined features. Experiments on real-world data show all the information used can help improve the recommendation performance, and our final method outperforms several baseline methods.

One future direction is to take more information into account such as the user's viewing history and tags of the tweet. We can also consider change of users' interests over time. Since our CTR method is generic, it is easy to incorporate more information by adding extra features.

The cold start problem is another tricky issue to deal with. In our experiments, we have assumed that each user has retweeted a sufficient number of statuses to reveal his personal interests. However, new users or inactive users have few explicit actions for our method to detect their interests. To address this, we can build a system involving an interview process, where users are asked for their opinions on certain chosen tweets to express their interests explicitly. This will be another direction of our future research.

7. ACKNOWLEDGEMENT

We thank the anonymous reviewers for their valuable and constructive comments and NSFC-RGC joint research project 60931160445 for generous support. We also thank Shenghua Bao from IBM Research China for his comments and discussions. Finally, we thank Sandy Harris from Shanghai Jiao Tong University for his carefully reading and revising our paper to correct grammatical errors.

8. REFERENCES

[1] D. Agarwal and B. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2009.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] Y. Cao, J. Xu, T. Liu, H. Li, Y. Huang, and H. Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM, 2006.

[4] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194. ACM, 2010.

[5] P. Cui, F. Wang, S. Liu, M. Ou, S. Yang, and L. Sun. Who should share what? item-level social influence prediction for users and posts ranking. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pages 185–194. ACM, 2011.

[6] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.

[7] Y. Duan, L. Jiang, T. Qin, M. Zhou, and H. Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 295–303. Association for Computational Linguistics, 2010.

[8] J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 199–206. ACM, 2010.

[9] J. Hannon, K. McCarthy, and B. Smyth. Finding useful users on twitter: twittomender the followee

recommender. *Advances in Information Retrieval*, pages 784–787, 2011.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[11] P. Kapanipathi, F. Orlandi, A. Sheth, and A. Passant. Personalized filtering of the twitter stream.

[12] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[13] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[14] Y. Li, J. Nie, Y. Zhang, B. Wang, B. Yan, and F. Weng. Contextual recommendation based on text mining. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 692–700. Association for Computational Linguistics, 2010.

[15] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[16] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 83–90, New York, NY, USA, 2008. ACM.

[17] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*.

[18] H. Ma, H. Yang, M. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[19] M. Michelson and S. Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 73–80. ACM, 2010.

[20] N. Naveed, T. Gottron, J. Kunegis, and A. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. 2011.

[21] J. Pessiot, T. Truong, N. Usunier, M. Amini, and P. Gallinari. Learning to rank for collaborative filtering. In *9th International Conference on Enterprise Information Systems*. Citeseer, 2007.

[22] Z. Qu and Y. Liu. Interactive group suggesting for twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 519–523. Association for Computational Linguistics, 2011.

[23] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*. The AAAI Press, 2010.

[24] S. Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, 2010.

[25] S. Rendle, C. Freudenthaler, Z. Gantner, and S.-T. Lars. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, 2009.

[26] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pages 635–644. ACM, 2011.

[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[29] Y. Shi, M. Larson, and A. Hanjalic. Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 125–132, New York, NY, USA, 2009. ACM.

[30] D. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, pages 111–120. ACM, 2009.

[31] A. Sun, J. Cheng, and D. Zeng. A novel recommendation framework for micro-blogging based on information diffusion. In *Proceedings of the 19th Workshop on Information Technologies and Systems*, 2009.

[32] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[33] W. Wu, B. Zhang, and M. Ostendorf. 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*, pages 689–692. Association for Computational Linguistics, 2010.

[34] S. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.

[35] T. Zaman, R. Herbrich, J. Van Gael, and D. Stern. Predicting information spreading in twitter. In *NIPS Workshop on Computational Social Science and the Wisdom of Crowds*, 2010.