

Diffusion-aware Personalized Social Update Recommendation

Ye Pan¹, Feng Cong², Kailong Chen¹, Yong Yu¹

¹Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai China
{yepan,chenkl,yyu}@apex.sjtu.edu.cn

²Sino-US Global Logistics Institute, Shanghai Jiao Tong University
cfier@sjtu.edu.cn

ABSTRACT

Many Internet users have encountered serious information overload problem on social networks such as Facebook and Twitter, where users can consume the streams of social updates from their social connections. Traditional methods solving this problem include collaborative filtering and information diffusion modeling. Both methods answer the “who will adopt what” question from different perspective, while either of them only captures single-faceted knowledge of evidences. In this paper, we solve the personalized social update recommendation problem by proposing a framework which integrates the advantages of collaborative filtering and the characteristics of diffusion processes. The main contributions of this paper are three folds. First, we propose a plenty of diffusion features which capture the characteristics of diffusion processes. Second, we build a joint model which takes the advantages of both collaborative filtering and the characteristics of diffusion processes for recommendation. Finally, experiments on two real-world datasets show that our joint model outperforms the methods capturing single-faceted knowledge and several other baselines.

Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval—*Information filtering*

Keywords

Collaborative Filtering, Information Diffusion, Social Updates, Personalized Recommendation

1. INTRODUCTION

Many Internet users have encountered serious information overload problem on information-driven social networks, such as Facebook [23], Twitter[8], LinkedIn[18]. The most distinguishing feature of information-driven social networks is that they allow users to share their updates with their

neighbors, and users can consume the streams of social updates from their connections via customized news feeds in their homepages. However, this characteristic is a double-edged sword in a way [23]. On one hand, it accelerates the propagation of information, because some popular information will reach more users via the sharing mechanism. On the other hand, this leads to hundreds or even thousands of updates posted to many users every day. It is very inconvenient for many users, especially those who are active and have many neighbors to find what they want, as personally useful social updates may be flooded by many others.

Some recent works have tried to tackle information overload problem via developing social update recommendation techniques in different social networks [8, 19]. Collaborative filtering, as a promising technology for recommender systems, has played an important role in solving information overload problem [11]. Collaborative filtering has attracted much attention since the Netflix Prize [5] in the academia and also been applied in some successful commercial systems, including Amazon [28], Netflix and Google News [11]. Collaborative filtering aims to build a recommender system by learning users’ preferences based on the community data. It infers the unobserved user preferences by discovering the correlation between users and items based on observed user preferences. The most popular collaborative filtering method nowadays is the latent factor model, as it has helped winning prizes in a lot of competitions.

On the other hand, some research works have accomplished the recommendation task on social networks in the view of information diffusion [36, 25], as information access behaviors of social network users can be modeled as an information diffusion issue. Research works on information diffusion mainly focus on examining the characteristics of diffusion processes [6] and modeling the diffusion by detecting the factors that influencing the diffusion processes [14, 38]. Social networks play important roles in determining whether a user will adopt an item [17, 3], as user behaviors are confounding results driven by homophily [31], social influence and some other external influence [32, 1].

Collaborative filtering and information diffusion modeling on social networks have many internal correlations, as they are targeting the similar problem to some extent in different views. As a technique for recommender systems, collaborative filtering tries to answer whether a user will adopt or give positive feedback for a specific item. Once whether a user will adopt an item is well predicted, the details of information diffusion process can be well described, such as

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RecSys’13, October 12–16, 2013, Hong Kong, China.

Copyright 2013 ACM 978-1-4503-2409-0/13/10 ...\$15.00.

<http://dx.doi.org/10.1145/2507157.2507177>.

whether an item will generate a large cascade [4, 37], what is the depth and the width of dissemination trees [27]. Both collaborative filtering and information diffusion modeling focus on answering the “who will adopt what” question, but they are based on different assumptions. As collaborative filtering assumes users who agreed in the past are likely to agree again in the future, it infers individual preferences from those people with similar tastes. Collaborative filtering actually used the similarity ties [9] as information sources for predicting users’ adoption. On the other hand, information diffusion modeling often use the social ties as information sources for dictating users’ behaviors [14, 3], as the actions of neighbors in a social network are highly correlated. The emphasis of diffusion model is usually the relationships between users in diffusion graph.

In this paper, we solve the personalized social update recommendation problem by proposing a framework which takes the advantages of collaborative filtering and the characteristics of diffusion processes. As we see, social ties and similarity ties play different roles for predicting future behaviors, and this suggests combining them in a hybrid recommender system will become a promising approach [9]. Our proposed model caters for this intuition by integrating collaborative filtering and the characteristics of diffusion processes, as the former captures the predicting power from similarity ties and the latter captures predicting power from social ties. Some previous works have used the social network structure to help improve the collaborative filtering recommender systems, but they only considered the social relations rather than the dynamic diffusion processes. Some other previous works have used information diffusion processes for recommendation, but the knowledge inside each node of the diffusion graph is ignored.

The main contributions of this paper are three folds. (1) We propose plenty of diffusion features which are helpful for prediction by capturing the characteristics of diffusion processes; (2) We propose a joint model which naturally embeds the the diffusion features in the traditional collaborative filtering model to simultaneously capture the predicting power of collaborative filtering and diffusion features. As far as we know, this is the first work to combine the two parts together; (3) We examine our method on two real-world datasets, Twitter and Sina Weibo. The evaluation results verify that our joint model utilizing all the diffusion features outperforms several baselines.

2. RELATED WORKS

2.1 Information Diffusion Analysis and Characteristics

Information diffusion, or information dissemination, information propagation in online social networks has attracted much attention in research field of online social network mining. Information diffusion in social network is mainly driven by word-of-mouth spread via social ties and social networks represent a fundamental medium. As more and more online social networks such as Facebook, Twitter, Flickr have provided much real world data, research works on information diffusion have become easier to study [24, 35].

Discovering the characteristics of diffusion processes is a central issue on information diffusion in online social networks and many experimental results of qualitative and quantitative analyses across different networks have been pro-

posed. Cha [6] analyzed large-scale traces of information diffusion in the Flickr social network and found that even popular photos spread narrowly and slowly. Romero [35] explored the diffusions of hash tags on Twitter and found that according to their types and topics they exhibited different mechanics, which was interpreted as “stickiness” and “persistence”. After examining the characteristics of the diffusion processes, diffusion models were proposed to mimic diffusion processes that had characteristics consistent with the real data [27, 13]. Though most online social networks are user-centric, they may have totally different diffusion characteristics, as the behaviors of users and the social relations shaped by different mechanisms of networks are semantically different.

Some other research works tried to infer and recover the complete and exact diffusion process when the underlying propagation networks were unobserved [15], or when some confounding factors such as homophily, external influence made it difficult to measure the spread of information sharing behaviors [15, 2, 32]. In our experiments, we use the retweeting actions of microblog users to construct the explicit diffusion networks and do not introduce many confounding factors.

2.2 Personalized Social Update Recommendation

Recommender systems have enjoyed a surge of interest in academia since Netflix Prize competition and been adopted in many commercial systems [5, 21]. Towards building a personalized recommender system, collaborative filtering is a promising technology by learning users’ preferences based on the community data [21, 34]. It works by discovering the correlation between users and items based on observed user preferences so that unobserved user preferences can be inferred. Collaborative filtering methods have been very popular methods since they help win many open competition including Netflix Prize and KDDCUP.

Social update recommendation has recently attracted much attention to solve the information overload problems in social networks [23, 8, 19, 10, 7]. Different from previous recommendation problems, social update recommendations focus more on incorporating some other special characteristics, such as the sharing behaviors of users, social influence and correlations. The most related category of research works are personalized social recommendation [20, 30, 16, 40, 41]. Most of these works were based on the intuition that user tastes were highly correlated to the ones of their friends or people they trusted then used this to make more accurate guess of user behaviors for improving recommendation. But their works only considered the user-neighbor interaction rather than the characteristics of information diffusion process, which revealed the dynamics of user adoption behaviors and help improve recommendation experimentally. Some other previous works used information diffusion for recommendation [36], but they did not take the similarity ties into consideration, which were the main information source of making collaborative filtering effective for recommendation. Our method takes advantage of collaborative filtering and information diffusion characteristics to deal with the recommendation problem to greatly improve the personalized social update recommendation performance.

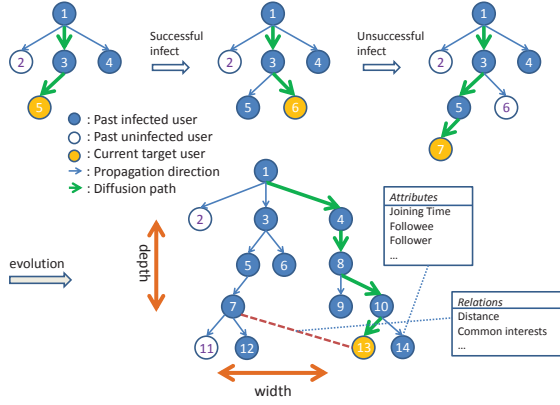


Figure 1: Evolution and features of cascades.

3. DIFFUSION-AWARE MODEL

In this section, we first introduce the diffusion features used in our work. We then propose the joint model which integrates collaborative filtering and diffusion processes in social update recommendation.

3.1 Diffusion Features in Diffusion Process

Previous works solving personalized social update recommendation ignore the effect of information cascades. Cascades are subgraph of the social networks which are consisted of the infected users. However, cascades contain lots of important information, which can greatly influence users' behavior. For example, users are more likely to adopt the information when the cascade contains lots of people they are interested in. Diffusion features are usually varied by the time and are different for each user. Figure 1 shows the evolution of the cascades. The user number is ordered by the time each user received this information. Each time there is only one user who is currently receiving the information. As the time passed, some users are infected while some are not and the cascade becomes bigger and more complex. Each user in the cascade has own attributes such as the time joined into the cascade, number of followees and number of followers. What's more, users in the cascade or path also have strong relations between each other such as distance between them and their common interests. These attributes and relations inside the cascade and path can be used to help predicting the users' behaviour more accurately. In the following, we propose all the features of cascades and propagations paths which are used in our work.

3.1.1 Popularity related features

Popularity related features refer to those features which are highly relevant to the popularity of cascades. General speaking, users in social networks tend to adopt more popular information.

- **Cascade size:** The size of a cascade equals to the number of infected users in the past and reveals the popularity of the information. Bigger cascade size usually leads to a more popular information, which will be accepted by more users.
- **Cascade depth and width:** The depth of a cascade is defined as the length of the longest path in the diffusion

tree. The width of a cascade is defined as the number of leaves in the diffusion tree.

- **Length of propagation path:** Length of the propagation path is defined as the length of the path from the root user to the target user in a diffusion tree. Length of a propagation path shows the penetrating power of the information. In common situations, users are more likely to be infected by the information which has longer propagation path.
- **Temporal popularity of topics :** Some hot topics spread very fast in a certain period of time. Users are more likely to adopt these topics during the beginning eruptible time, but have less interests when these topics are not popular. This feature estimates the temporal popularity of a topic after a period of time:

$$Temporal(i, u) = \frac{|C_{i,u}|}{t_{u,i} - t_p} \quad (1)$$

, where $C_{i,u}$ represents the cascade of information i when user u receives the information. t_u represents the time that user u receives information i and t_p represents the time that the publisher sends out information i .

- **Number of influenced users:** If at least one of a user's followee posted or reposted the information, the user is influenced by the information no matter whether this user will adopt the information. Popular information usually has a large sum of influenced users.
- **Celebrity Score:** If more celebrities have agreed on an opinion, the opinion will be adopted by more people. One reason is that celebrities usually have stronger influence. Another reason is that celebrities have more followees but less followers, which leads to the fact that the information flowed from them can be seen by more people. This feature estimates the power of celebrity effect:

$$Celebrity(i, u) = \sum_{v \in C_{i,u}} \frac{|FER(v)|}{|FEE(v)|} \quad (2)$$

, where $C_{i,u}$ represents the cascade of information i when user u receives the information. $FER(v)$ refers to the follower set of user v and $FEE(v)$ refers to the followee set of user v . Considering propagation path instead of the cascade, this feature can also be defined as following:

$$CelebrityPath(i, u) = \sum_{v \in P_{i,u}} \frac{|FER(v)|}{|FEE(v)|} \quad (3)$$

, where $P_{i,u}$ represents the propagation path of information i when user u receives the information.

3.1.2 Neighbor related features

Neighbor related features measure the relation between target user and infected neighbors of target user in cascades. Here neighbors are refer to users who are close to target user in distance or have similar interests. If cascades are highly related to neighbors of the target user, the target user will more easily adopt the information. Unlike popularity related features, neighbor related features are personalized.

- **Number of adopted followees:** Users get all the information from their followees. If there are many infected followees in the cascades or in the propagation path, the user will be surrounded by this information, which makes the user easier to adopt the information.
- **Number of adopted friends:** If two users follow each other, we say these two users are friends. Friends are special relationship between two users in social network. Unlike unidirectional following, relationship of friends shows the relation of two users are very close. If many friends have been infected in the cascade or the propagation path, the user will more likely be infected.
- **Brotherhood Score:** In social network, two users who have more common followers tend to be more similar. This feature estimates the power of this relationship:

$$Brother(i, u) = \log\left(\frac{\sum_{v \in C_{i,u}} |FER_{u,v}|}{|C_{i,u}|} + 1\right) \quad (4)$$

, where $FER_{u,v}$ represents the number of common followers of u and v . $C_{i,u}$ represents the cascade of information i when user u receives the information. By same strategy of celebrity score, this feature can be defined on propagation path.

- **Co-father relationship Score:** People follow another people in a social network as he/she appreciates one of their specialties or hobbies. If two people share a lot of common followers, they are more likely to be similar users. This feature estimates the power of this relationship:

$$CF(i, u) = \log\left(\sum_{v \in C_{i,u}} \sum_{w \in FER_{u,v}} \frac{1}{|FEE(w)| + 1} + 1\right) \quad (5)$$

, where $FEE(w)$ refers to the followee set of user w . $FER_{u,v}$ represents the number of common followers of u and v . $C_{i,u}$ represents the cascade of information i when user u receives the information. By same strategy of celebrity score, this feature can be defined on propagation path.

- **Grandfather-grandchild relationship Score:** If many followee of one user are another user's follower, they are more likely to be similar users. This feature estimates the power of this relationship:

$$G(i, u) = \log\left(\sum_{v \in C_{i,u}} \sum_{w \in S} \frac{1}{|FEE(w)| + 1} + 1\right) \quad (6)$$

$$S = FEE(u) \cap FER(v) \quad (7)$$

, where $FEE(u)$ and $FEE(w)$ refer to the followee set of user u and user w respectively. $FER(v)$ refers to the follower set of user v . $C_{i,u}$ represents the cascade of information i when user u receives the information. By same strategy of celebrity score, this feature can be defined on propagation path.

Combining all these cascades features showed above, regression models can be used to predict the extent of preference toward the information given its cascade.

3.2 Diffusion-aware Social Update Recommendation Model

In the previous part, we introduce the diffusion features in diffusion processes. In this part, we will show how we utilize these features in our joint model. The joint model is an extension of our previous work[8]. In the following, we will propose the principle of this diffusion-aware model and explain the parameters. We divide this joint model into several parts for introduction. In the first place, we propose the diffusion level decomposition strategy in the joint model. We then introduce the method incorporating all diffusion based explicit features. Finally, we introduce the collaborative ranking methods for model training.

3.2.1 Diffusion level decomposition

Our previous model[8] improves traditional collaborative filtering by decomposing the latent factor of items. It works well especially when the latent factor of words and publishers are very closed to the latent factor of items. However, our previous model loses a lot of information from the perspective of diffusion. When we need to recommend an item to a user, it is highly related to who have adopted the same or similar items. It leads to a great influence from the past users who have adopted this item. For example, the decision of a user to adopt an item will be influenced by his friends who have already adopted. From the view of diffusion graph and collaborative filtering, if a user adopts an item, the latent factor of this user should be very similar to the users who have already adopted the item in the past. If a user adopts an item, we decompose the latent factor of a user into a combination of latent factors of similar users, to get the following factorization model:

$$\hat{y}_{u,i} = bias + \left(p_u^T + \sum_{s \in C_i} \alpha_s p_s^T \right) q_i \quad (8)$$

Here bias denotes any form of possible bias, C_i is the cascade of tweet i in the past, p_s represents the latent factor of each infected user in C_i and α_s denotes the weight parameter of each user in the cascade. As the influence of people who have adopted the information differs from each other, the weight α_s is usually different for each C_i . For instance, the publisher of an information and the neighbours are usually the most important influencers to a user whom we want to recommend item to.

In many cases, if we want to select part of users in cascade as the influencers, we can make α_s of the other users equal 0. In the final joint model, we make $\alpha_s = 1$ if and only if user s is the direct sender of user i . That's to say we only use the user who delivers the information instead of all users who have adopted the information in the past. The most important reason is that we found users are usually influenced by their closest users in the cascade. This observation makes sense in many social networks such as Twitter and Sina Weibo. The direct sender of the information in a cascade is the followee even the friend of user i who is particular in the cascade. While the rest of users in a cascade usually have less relationship with the user i .

This decomposition strategy matches our intuition well. Firstly, it gets more opinions at diffusion level, which helps collaborative filtering based methods get better performance. What's more, it overcomes the problem of data scarcity for analyzing users. The latent factor of an un-

known user can be estimated from the infective in the past. Moreover, this model not only considers the details inside a node but also leverages the global dynamics of diffusion graph, which unifies the recommendation and diffusion process together.

3.2.2 Explicit features of diffusion

Besides the latent factors, all the features of the diffusion tree shown in 3.1 are considered in this part. These features directly measure whether an information will be accepted. We use a linear combination of these features to get the bias prediction:

$$\hat{y}_{u,i} = \sum_j b_j \gamma_j^d \quad (9)$$

, where b is a weight parameter vector and γ^d is a diffusion explicit feature we extracted.

Combining two strategies shown above, we get the final joint model as following:

$$\hat{y}_{u,i} = \sum_j b_j \gamma_j^d + \left(p_u^T + \sum_{s \in C_i} \alpha_s p_s^T \right) \left(\frac{1}{Z} \sum_{w_j \in T_i} q_{w_j} + \alpha d_{p(i)} \right) \quad (10)$$

, where T_i is the term set of tweet i , Z is the normalization term for features and $d_{p(i)}$ is a latent factor of the publisher of tweet i .

3.2.3 Model training

Ranking-oriented personalized recommendation has attracted much attention in recent years [39, 29, 22, 33]. To adapt to the scenario that rank results of items are presented to users, the idea of ranking-oriented methods is more natural and better than the traditional collaborative filtering methods which aims at predicting the ratings of users [21]. Most of these methods take advantages of the learning to rank techniques [26, 12] to optimize the final ranking criterion for personalized recommendation results.

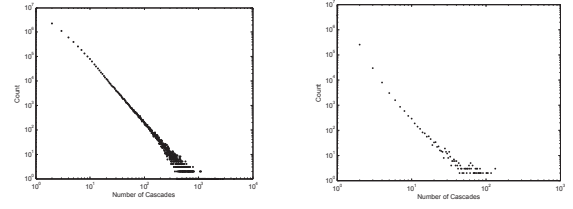
Under the scenario of social update recommendation, we also introduce optimizing the ranking criterion and collaborative ranking for personalized tweet recommendation. Collaborative ranking is a ranking version of collaborative filtering and provides item ranking results according to relative preference rather than user rating estimation. The collaborative ranking method we used here 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:

$$\hat{y}_{u,i} = \mu + b_u + b_i + p_u^T q_i \quad (11)$$

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 items k and h the pairwise ranking model for preference is defined as follows:

$$P(r(k) > r(h)|u) = \frac{1}{1 + e^{-(\hat{y}_{u,k} - \hat{y}_{u,h})}} \quad (12)$$



(a) Cascade count distribution over cascade size in Sina Weibo (b) Cascade count distribution over cascade size in Twitter

Figure 2: Overview of the dataset

, where $\hat{y}_{u,k}$ is the predicted preference of user u for tweet k , r is short notation for rank order. Equation 12 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:

$$\mathcal{D} = \{ \langle u, k, h \rangle \mid k \in Re(u), h \notin Re(u) \} \quad (13)$$

, 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:

$$\min \sum_{\langle u, k, h \rangle \in \mathcal{D}} \ln \left(1 + e^{-(\hat{y}_{u,k} - \hat{y}_{u,h})} \right) + \text{regularization} \quad (14)$$

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

4. EXPERIMENTS

4.1 Dataset

We evaluate our model in two datasets, Sina Weibo and Twitter. Sina Weibo is the most popular social network in China. Just like posting tweets, users of Sina Weibo share their updates by posting weibos. The data of Sina Weibo is downloaded from WISE 2012 Challenge. We randomly select users who posted more than 1000 weibos. After randomly selecting 11786 qualified users, we got 36311395 weibos in our database. We then randomly split these weibos into training set with possibility of 0.8 and test set with possibility of 0.2. Finally the dataset is split into a training set with 29054037 weibos and test set with 7257358 weibos. The forwarded weibos are regarded as positive samples and the others are negative samples in experiments. With one positive, we have another 20 negatives in both training set and test set. Figure 2(a) shows the real cascade size with count of cascades in Sina Weibo dataset. From the figure, we can see that most of the cascades are very small, which makes diffusion-based recommendation more difficult.

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 28925 users in our base and downloaded all the statuses they had posted. 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 nineteen 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. Then we randomly split the samples into test set and training set with possibility of 0.2 into test set and possibility of 0.8 into training set. Each of them had about 199 messages on average. Finally the dataset is split into a training set with 4601966 tweets and test set with 1149389 tweets. Figure 2(b) shows the real cascade size in our dataset. Every tweet in cascade can be found in our base. From the figure, we can see that most tweets are not retweeted or are only retweeted a few times, which shows the sparsity of the dataset. It also shows the fact that most of the cascades are very small as the dataset of Sina Weibo.

Each dataset has its own characteristic. Among two datasets, Sina Weibo is more suitable for cascades evaluations as this dataset records the diffusion path of each weibo. While on Twitter, we can only guess the diffusion path by looking for which followee posted the tweet with same source tweet ID, as each tweet only records the source tweet ID but not the diffusion path. If more than one followees have posted a tweet with same source tweet ID, we can't make sure which one among them exactly delivered this tweet to the user. In this case, we assume that the user is influenced by all these followees. Though having the data of diffusion path, the dataset of Sina Weibo is lack of content data. The dataset of Twitter offsets this drawback and makes it possible to compare our joint model with the best baseline.

4.2 Evaluation Metric

In the evaluations, tweet ratings are considered binary in our scenario. Retweeting a tweet corresponds to a 1 rating, while not retweeting correspond to a 0 rating. We use Mean Average Precision (MAP), a popular rank evaluation method to evaluate the proposed model. For a single user, average precision is defined as the average of the standard $P@n$ values for all retweeted tweets:

$$AP = \frac{\sum_{n=1}^N P@n \times retweet(n)}{|R|} \quad (15)$$

, 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 n th tweet in the result list. Finally, MAP can be obtained by averaging the AP values of all the users.

We also use $P@n$ to evaluate our model. As there are limited positives in our Twitter dataset, we modify the standard $P@n$ to show more obvious results. We define the $P@n$ as following:

$$P@n = \frac{TP_n}{R_n} \quad (16)$$

, where TP_n is known as the number of true positives in top n tweets and R_n is number of retweeted Tweets in top n tweets.

4.3 Evaluations on Sina Weibo

4.3.1 Method Comparison

In this part, we compare our models to several others. As lots of baseline methods such as RankSVM, LDA and

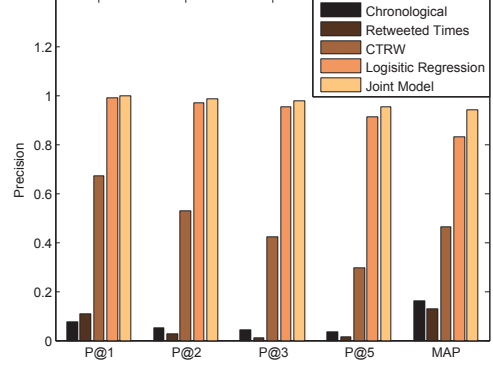


Figure 3: Methods Comparison in Sina Weibo

JointMF have been beaten by CTR model, we only list some important baselines here. 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 social networks.
- **RETWEETED TIMES:** Retweeted Times is an estimate of the popularity of a weibo in the history. Tweets with more retweeted times are more possibly to be liked by people. This ranking strategy ignores personalization and assumes the user's interests are the same as general public's.
- **CTRW:** Collaborative tweet ranking model integrating social relations and explicit features proposed in our previous work[8]. As the dataset of Sina Weibo doesn't have content information, here CTR model is degraded into a version without term factors. All the other parts are as same as CTR model. Stochastic gradient descent is used for parameter estimation. In experiments, the number of latent factors is set to 32, as the bigger number will bring little improvement on performance. The normalization term Z for term factors is set as $|T_i|^{\frac{1}{2}}$ according to the experimental results. As far as we known, this is one of the best models to solve this problem. We regard it as the most important baseline.
- **LOGISTIC REGRESSION:** Excluding the latent factors, we get the logistic regression model which utilizes all of cascades features listed in 3.1. We train logistic regression model on the training data and then evaluate the model by the test data. All the parameters use the default settings.
- **JOINT MODEL:** The joint model integrates collaborative filtering, information diffusion factor, social factor and all other explicit features together. The joint model here is also degraded into the version without term factors like CTRW. The parameter configurations are as same as CTRW model.

Figure 3 shows the performance comparison results of each model mentioned above. Although lack of term factor, the performance of CTRW model are still better than

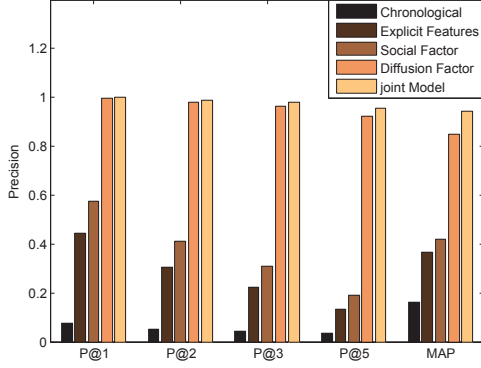


Figure 4: Components Comparison of the Joint Model in Sina Weibo

the chronological and retweeted times. The main reason is that the model of chronological and retweeted times don't take personalized knowledge into consideration. On the other hand, CTRW and the joint model gets the MAP value of 0.465351 and 0.945481 respectively. The performance of CTRW is much lower than the joint model. The most possible reason is that the joint model improves the personalized collaborative filtering by considering in the background of information diffusion. By leveraging collaborative filtering and information diffusion, the joint model captures users' interests both from the personal features inside a user node and the global dynamic in the diffusion graph. What's more, using all of diffusion features in this work, logistic regression model achieves the MAP value of 0.831853. The performance is much better than CTRW, but lower than the joint model. The joint model improves logistic regression model by combining collaborative filtering model. This result shows that the latent factor parts do bring improvement on the performance.

Also it is interesting to know which part of our joint model affords the most contribution. We evaluate each part of the joint model and show the effectiveness of each component in Figure 4. The latent factor number of all three components are set to 32. Here the chronological model and the joint model are used as references. From the figure, we can see that the MAP value of explicit features is 0.368272, which is better than the chronological model. It proves these explicit features are useful. The social factor achieves the MAP value of 0.420642, which is a little better than explicit features. The performance of diffusion factors is 0.851439, which stands out among all the components in the final model. This reflects the fact that the influence of the specific followees are quite enormous in social update recommendation. In addition, combining all three parts, the joint model achieves the best MAP value, which verifies our intuition that combining diffusion processes and collaborative filtering are meaningful.

4.4 Evaluations on Twitter

Although our model integrating diffusion process and collaborative filtering achieves good performance, the Sina Weibo dataset does not have contents. It is unknown whether our combined model can beat CTR model when considering content of the information. The dataset of Twitter overcomes this drawback. In this part, we evaluate our model

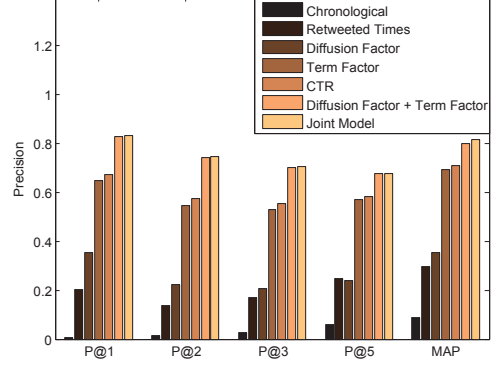


Figure 5: Evaluation results in Twitter

on Twitter and compare our model with the complete CTR model and some other baselines. The parameters setting of CTR model is as same as CTRW which is shown in 4.3.1.

Figure 5 shows the performance comparison results of CTR, the joint model and some other models mentioned above. As what have showed in our previous work, it is not surprising that the performance of CTR model are better than the chronological and retweeted times. Unlike Sina Weibo, the performance of retweeted time is much better in the dataset of Twitter. This is due to the API of Twitter can exactly return the retweeted time of each tweet, while this value in Sina Weibo can only be estimated by counting the number of weibos in incomplete cascades. On the other hand, CTR and the joint model gets the MAP value of 0.709184 and 0.809394 respectively. The performance of CTR is lower than the joint model. This shows the result that our joint model can still beat CTR when considering the influence of social content in the meantime.

We also compare the term factor and diffusion factor in this graph. The diffusion factor and term factor achieves the MAP value of 0.355297 and 0.691963 respectively. The performance of diffusion factor is lower than the term factor. The most probable reason is that the dataset of Twitter is not suitable for cascades evaluation. As explained in 4.1, we can't get the exact information sender and diffusion path in Twitter. Another reason is that there are 3080 weibos for each user in Sina Weibo dataset on average, while only 199 tweets in Twitter. The diffusion part of the joint model may not be sufficiently trained in Twitter. On the other hand, this may be also caused by the fact that the preference of the users to the tweets are mainly determined by the contents of the tweets instead of who delivers the information. However, the performance of the term factors can be greatly improved by combining diffusion factor, which still verifies our intuition that information diffusion modeling and collaborative filtering capture different kind of characteristics in personalized social update recommendation.

5. CONCLUSION AND FUTURE WORK

Information diffusion modeling and collaborative filtering on social networks are highly related. In this paper, we propose a plenty of meaningful diffusion features which capture the characteristics of diffusion processes. What's more, we introduce a joint model leveraging collaborative filtering and information diffusion. Our approach both takes the advantages of information diffusion characteristics and col-

laborative filtering. Experiments on two real-world datasets show that the diffusion part of our model achieves good performance and our joint model outperforms several baseline methods.

One of our future work is to take time into consideration. The time information can help recommendation system to recommend right thing at right time, which makes the recommendation more suitable. Also it can make diffusion prediction more accurate considering when and where will a specific information flow to. Since the framework of our model is generic, it will be convenient to incorporate other information.

Another interesting issue is to solve cold start problem. New users or inactive users usually have few information for our methods to detect their interests. One possible approach is to transfer the actions of users from other domains. For instance, users who are not active in Twitter may be active in Facebook. Thus the actions of users in Facebook may help improving the recommendation in Twitter.

6. REFERENCES

- [1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *SIGKDD*, 2008.
- [2] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *PNAS*, 2009.
- [3] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *EC*, 2012.
- [4] E. Bakshy, J. Hofman, W. Mason, and D. Watts. Everyone's an influencer: quantifying influence on twitter. In *WSDM*, 2011.
- [5] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD Cup and Workshop*, 2007.
- [6] M. Cha, A. Mislove, and K. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *WWW*, 2009.
- [7] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *CHI*, 2010.
- [8] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *SIGIR*, 2012.
- [9] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *SIGKDD*, 2008.
- [10] 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 *SIGIR*, 2011.
- [11] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*, 2007.
- [12] P. Donmez, K. Svore, and C. Burges. On the local optimality of lambdarank. In *SIGIR*, 2009.
- [13] M. Götz, J. Leskovec, M. McGlohon, and C. Faloutsos. Modeling blog dynamics. In *ICWSM*, 2009.
- [14] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.
- [15] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW*, 2004.
- [16] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. Social media recommendation based on people and tags. In *SIGIR*, 2010.
- [17] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 2006.
- [18] L. Hong, R. Bekkerman, J. Adler, and B. Davison. Learning to rank social update streams. In *SIGIR*, 2012.
- [19] L. Hong, R. Bekkerman, J. Adler, and B. D. Davison. Learning to rank social update streams. In *SIGIR*, 2012.
- [20] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *SIGKDD*, 2009.
- [21] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 2008.
- [22] Y. Koren and J. Sill. Ordrec: an ordinal model for predicting personalized item rating distributions. In *RecSys*.
- [23] K. Koroleva, H. Krasnova, and O. Günther. Stop spamming me!-exploring information overload on facebook. In *AMCIS*, 2010.
- [24] D. Lazer, A. Pentland, L. Adamic, S. Aral, A. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 2009.
- [25] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. *EC*, 2006.
- [26] P. Li, C. Burges, and Q. Wu. Learning to rank using classification and gradient boosting. In *NIPS*, 2007.
- [27] D. Liben-Nowell and J. Kleinberg. Tracing information flow on a global scale using internet chain-letter data. *PNAS*, 2008.
- [28] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 2003.
- [29] N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *SIGIR*, 2008.
- [30] H. Ma, H. Yang, M. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, 2008.
- [31] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.
- [32] S. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *SIGKDD*, 2012.
- [33] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *UAI*, 2009.
- [34] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *SIGIR*, 2011.
- [35] D. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *WWW*, 2011.
- [36] X. Song, Y. Chi, K. Hino, and B. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. In *WWW*, 2007.
- [37] E. Sun, I. Rosenn, C. Marlow, and T. Lento. Gesundheit! modeling contagion through facebook news feed. In *ICWSM*, 2009.
- [38] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *SIGKDD*, 2009.
- [39] M. Weimer, A. Karatzoglou, Q. Le, A. Smola, et al. Cofrank-maximum margin matrix factorization for collaborative ranking. In *NIPS*, 2007.
- [40] 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 *WWW*, 2011.
- [41] M. Ye, X. Liu, and W. Lee. Exploring social influence for recommendation: a generative model approach. In *SIGIR*, 2012.