

Context-aware Ensemble of Multifaceted Factorization Models for Recommendation Prediction in Social Networks

Yunwen Chen^{*}
Shanda Innovations
kddchen@gmail.com

Zuotao Liu
Shanda Innovations
zuotao.liu@gmail.com

Daqi Ji
Shanda Innovations
duky2001@gmail.com

Yingwei Xin
Shanda Innovations
xinyingwei@gmail.com

Wenguang Wang
Shanda Innovations
svd.wang@gmail.com

Lu Yao
Shanda Innovations
luyao.2013@gmail.com

Yi Zou
Shanda Innovations
vodkajoe@gmail.com

ABSTRACT

This paper describes the solution of Shanda Innovations team to Task 1 of KDD-Cup 2012. A novel approach called Multifaceted Factorization Models is proposed to incorporate a great variety of features in social networks. Social relationships and actions between users are integrated as implicit feedbacks to improve the recommendation accuracy. Keywords, tags, profiles, time and some other features are also utilized for modeling user interests. In addition, user behaviors are modeled from the durations of recommendation records. A context-aware ensemble framework is then applied to combine multiple predictors and produce final recommendation results. The proposed approach obtained 0.43959 (public score)/0.41874(private score) on the testing dataset, which achieved the 2nd place on leaderboard of the KDD-Cup competition.

Categories and Subject Descriptors

H3.3 [Information Systems]: Information Search and Retrieval-information Filtering; D.2.8 [Database Management]: Database Applications-Data Mining

General Terms

Data Mining, Algorithms, Machine Learning, Social Network, Recommendation

Keywords

Recommender Systems, Matrix Factorization, KDD Cup, Collaborative Filtering

^{*}Team leader and corresponding author.

1. INTRODUCTION

Social Networking Services (SNS) have gained tremendous popularity in recent years, and voluminous information is generated from social networks every day. It is desirable to build an intelligent recommender system to identify what interests users efficiently[1]. The task¹ of KDD-Cup 2012 Track 1 is to develop such a system which aims to capture users' interests, find out the items that fit to users' taste and most likely to be followed[12]. The datasets are provided by Tencent Weibo, one of the largest social networking websites in China, is made up of 2,320,895 users, 6,095 items, 73,209,277 training records, and 34,910,937 testing records, which is relatively larger than other publicly released datasets. Besides, it provides richer information in multiple domains, including user profiles, item categories, keywords, and social graph. Timestamps for recommendations are also given for performing session analysis. For each user in the testing dataset, an ordered list of the recommender results is demanded. Mean Average Precision (MAP)[15] is used to evaluate the results provided by 658 teams around the world.

Compared to traditional recommender problems, e.g., the Netflix Prize[3], where the scores users rate movies are predicted, the settings of KDD-Cup 2012 appear more complex. Firstly, there are much richer features between users on social networking websites. In the social graph, users can follow each other. Besides, three kinds of actions, including "comment" (add comments to someone's tweet), "retweet" (repost a tweet and append some comments) and "at" (notify another user), can be taken between users. User profiles contain rich information, such as gender, age, category, keywords and tags. So models that are capable to integrate various features are required. Secondly, items to be recommended are specific users, which can be a person, a group, or an organization. Compared to the items of traditional recommender systems, e.g. books on Amazon or movies on Netflix, items on social network sites not only have profiles, but also have their behaviors and social relations. As a result, item modeling turns out more complicated. Thirdly, the training data in the social networks is quite noisy, and the cold-start problem also poses severe challenge due to the very limited information for a large number of users in

¹<http://www.kddcup2012.org>

testing dataset. It is demanding to have an effective preprocessing to cope with this challenge.

In this paper we present a novel approach called Context-aware Ensemble of Multifaceted Factorization Models. Various features are extracted from the training data and integrated into the proposed models. A two stage training framework and a context-aware ensemble method are introduced, which helped us to gain a higher accuracy. We also give a brief introduction to the session analysis method and the supplement strategy that we used in the competition to improve the quality of training data.

The rest of the paper is organized as follows. Section 2 introduces preliminary of our methods. Section 3 presents the preprocessing method we used. In Section 4, we will propose Multifaceted Factorization Models, which is adopted in the final solution. A context-aware ensemble and user behavior modeling methods are proposed in Section 5. Experimental results are given in Section 6 and conclusions and future work are given in Section 7.

2. PRELIMINARY

In this section, we will first give a concise overview of the latent factor model, which forms the foundation of our solution. We then present some of its recent extensions specifically designed for collaborative filtering problems. Some preprocessing methods that we used are also introduced.

Before formally introducing the models, we first define our notational conventions. We denote U as the user set and I as the item set. Then we have the number of $|U|$ users and $|I|$ items, where function $|\cdot|$ indicates the quantity of members in a set. Special indexing letters are reserved for distinguishing users from items: for user u and for item i . A binary rating r_{ui} indicates the preference by user u for item i , where value $r_{ui} = 1$ means accepting the item and 0 means rejecting. The ratings are arranged in a $|U| \times |I|$ matrix $R = r_{ui}$. In the matrix, some cells have values from the training data, we use Ω to denote the set of (u, i, t) records for which ratings are observed at time t . We denote Ω^+ as all positive samples ($r_{ui} = 1$) and Ω^- as negative sample set ($r_{ui} = 0$).

2.1 Latent Factor Model

Recently, latent factor models comprise an alternative approach to collaborative filtering with the more holistic goal to uncover latent features that explain the relation between users and items[14]. Because of its attractive performance, availability and scalability[10][13], Singular Value Decomposition(SVD) is one of the most popular models. In this paper, we denote \hat{r}_{ui} as the prediction of user u 's probability to follow item i . The baseline predictor of SVD with user bias and item bias is defined as:

$$\hat{r}_{ui} = f(b_{ui} + \mathbf{q}_i^T \mathbf{p}_u) \quad (1)$$

where $f(\cdot)$ is a warping function used to map real values to certain ranges. \mathbf{p}_u is the d -dimensional user feature vector and \mathbf{q}_i is the item feature vector. d is a parameter that set beforehand. Bias factor b_{ui} is defined as:

$$b_{ui} = \mu + b_u + b_i \quad (2)$$

In this equation, parameter μ denotes the overall average rating. b_u and b_i indicate the deviations of user u and item i . These parameters can be learnt by solving the following regularized least squares problem:

$$\min_{\mathbf{p}^*, \mathbf{q}^*, b^*} \sum_{(u,i) \in K} \left(r_{ui} - f(b_{ui} + \mathbf{q}_i^T \mathbf{p}_u) \right)^2 + \lambda_1 \|\mathbf{p}_u\|^2 + \lambda_2 \|\mathbf{q}_i\|^2 + \lambda_3 b_u^2 + \lambda_4 b_i^2 \quad (3)$$

The first term in the above equation strives to find b_u , b_i , \mathbf{p}_u , and \mathbf{q}_i that fit the given ratings. The others are regularization terms with parameters λ_1 , λ_2 , λ_3 and λ_4 to avoid over-fitting.

2.2 Model Learning Approach

Stochastic Gradient Descent algorithm (SGD)[10] is a very effective approach for solving the optimization problems(Eq.3). It randomly loops through known ratings randomly in set Ω and takes a small gradient descent step on the relevant parameters along the direction that minimizes the error on each rating. Let us denote the prediction error as e_{ui} . The SGD updates the parameters based on the following equations:

$$\begin{aligned} b_u &\leftarrow b_u + \eta(e_{ui} - \lambda_1 b_u) \\ b_i &\leftarrow b_i + \eta(e_{ui} - \lambda_2 b_i) \\ \mathbf{q}_i &\leftarrow \mathbf{q}_i + \eta(e_{ui} \mathbf{p}_u - \lambda_3 \mathbf{q}_i) \\ \mathbf{p}_u &\leftarrow \mathbf{p}_u + \eta(e_{ui} \mathbf{q}_i - \lambda_4 \mathbf{p}_u) \end{aligned}$$

where parameter η indicates the learning rate, λ_1 , λ_2 , λ_3 , λ_4 are parameters that define the strength of regularization. The complexity of each iteration is linear in the number of ratings.

2.3 Implicit Feedback

Koren[8] indicated that the performance of recommendation systems can be improved significantly by accounting for implicit feedbacks. To solve the recommendation problem in social networks, high quality explicit feedbacks from users are usually limited, meanwhile, implicit feedbacks (e.g. the follow or be followed relations between users), which indirectly reflect opinion through observing user behavior, are more abundant and easier to obtain[7]. A conventional form of utilizing implicit information as user and item factors can be described in the following equation:

$$\hat{r}_{ui} = f \left(b_{ui} + \mathbf{q}_i^T (\mathbf{p}_u + \sum_{k \in F(u)} \beta_k \mathbf{y}_k) \right) \quad (4)$$

Here $F(u)$ indicates the set of user feedback records. \mathbf{y}_k is a feature vector associated with the user feedback information. β_k is a parameter and is usually set to $|F(u)|^{-1/2}$ in SVD++[8]. In the competition, many kinds of features, such as: sns relations, user actions, keywords, tags, etc. were used in our solution as latent factors and experiment results proved their effectiveness.

3. PREPROCESS

This section introduces the procedure of data preprocessing and pairwise training in our solution.

3.1 Session Analysis for Data Filtering

The training dataset consists of 73,209,277 binary ratings from 1,392,873 users on 4,710 items[12]. Among these ratings, there are 67,955,449 negative records and only 5,253,828 positive ones, which means 92.82% of the training ratings are negative samples. After carefully analyzing the negative records, we found that not all the negative ratings imply that the users rejected to follow the recommended items. For example, users may be attracted by something interesting on the Weibo website, and during these sessions, the recommended items were omitted by users. So eliminating these "omitted" records from the training dataset Ω is a very important preprocessing.

To remove those noise data, we adopt the following session analysis method. For each user u , we denote $t_0(u)$, $t_1(u)$, $t_2(u)$, ..., $t_m(u)$ as the timestamps of the recommended records in ascending order. Then the time interval between two adjacent timestamp can be easily calculated:

$$\Delta t_s(u) = t_{s+1}(u) - t_s(u) \quad (5)$$

where $0 < \Delta t_s(u)$, $0 \leq s < m$. When $\Delta t_s(u) < 3600(s)$, we denote it as $\Delta \ddot{t}_s(u)$. The threshold for session slicing is then obtained:

$$\tau(u) = \frac{1}{2} * \left(\tau_0 + \frac{\sum_{s=0}^m \Delta \ddot{t}_s(u)}{|\Delta \ddot{t}(u)|} \right) \quad (6)$$

Parameter τ_0 is set to 90 in our experiment. If $\Delta t_s(u) > \tau(u)$, the training records on time $t_{s+1}(u)$ and $t_s(u)$ will be separated to different subsets (sessions) $\psi_k(u)$ and $\psi_{k+1}(u)$. We denote $\psi_k^+(u)$ as all positive samples in $\psi_k(u)$.

For each session $\psi_k(u)$ and records $r_{ui} \in \psi_k(u)$, their validity are carefully evaluated. Only those samples which can simultaneously satisfy the following three formulas are reserved as valid training data.

$$\sigma_s - \sigma_- \leq \pi_- \quad (7)$$

$$\sigma_+ - \sigma_s \leq \pi_+ \quad (8)$$

$$0 < \frac{|\psi_k^+(u)|}{|\psi_k(u)|} \leq \epsilon \quad (9)$$

σ_s denotes the timestamp index of a record within the session. σ_- is the smallest index value of all the records $\sigma_k \in \psi_k^+(u)$, and σ_+ is the largest one as well. π_- , π_+ and ϵ are parameters on the effect of filtering. In our approach they are set to be 0, 3, 0.86, respectively. We picked out 7,594,443 negative records and 4,999,118 positive ones from the training dataset, and experimental results show that the proposed preprocessing effectively improves the prediction accuracy.

3.2 Pairwise Training

Different from conventional recommender applications, where Mean Square Errors (MSE) is often used to evaluate the predicted results, Mean Average Precisions (MAP) is employed in the competition as the evaluation metric[15]. For each candidate user in the testing dataset, the top-N items (N=3 in the competition) this user is most likely to follow are required, and average precisions of the top-N results are then calculated. MAP is the mean value of all these average precisions. Since MAP is commonly used to judge the

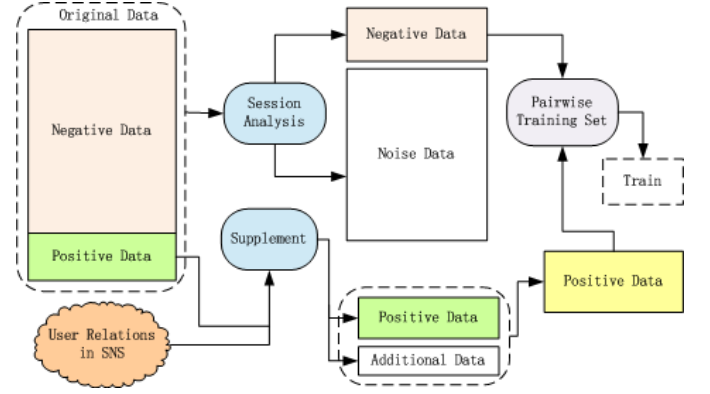


Figure 1: The procedure of data preprocessing and pairwise training

performance of different ranking algorithms, the KDD-Cup Track 1's task is more like a learning-to-rank problem[11]. In this case, we reformulate the task as a pairwise ranking job. For a given negative training sample (u, i) and a positive one (u, j) , the ranking objective is to give higher prediction value \hat{r}_{uj} over \hat{r}_{ui} . The objective function for this pairwise training is:

$$g(\hat{r}_{ui}, \hat{r}_{uj}) = \left(1 + \exp \left(- \left(b_j - b_i + (\mathbf{q}_j - \mathbf{q}_i)^T \mathbf{p}_u \right) \right) \right)^{-1} \quad (10)$$

Here we use sigmoid function for value wrapping. For each negative sample $r_{ui} \in \Omega^-$, we randomly select a positive sample r_{uj} in Ω^+ to build a training pair (r_{ui}, r_{uj}) . Similarly to Eq.(1), SGD can be used here for optimization. Note that b_u is omitted because user biases to item i and j are always the same.

3.3 Supplement Training Data

An ideal pairwise training requires a good balance between the num of negative and positive samples. However, in training dataset 94.9% of the users have less positive samples than negative ones. To deal with this problem, a preprocessing of sample supplement is added in our approach. Firstly, the users who have a far smaller number of positive samples than the negative samples are chosen. Then for each user u , we denote $S(u)$ as the items that the user followed and $A(u)$ as those user have interactions ('at', 'retweet' or 'comment') with. The best item was selected by the following function, and added to positive training set Ω^+ .

$$\operatorname{argmax}_{i \in S(u) \cap A(u)} (\xi_0 N_{at}(u, i) + \xi_1 N_{retweet}(u, i) + \xi_2 N_{comment}(u, i))$$

here $N_{at}(u, i)$, $N_{retweet}(u, i)$, $N_{comment}(u, i)$ indicate the numbers of each kind of actions user u gave to item i . Parameters ξ_0, ξ_1, ξ_2 are set to 2, 0.2, 1 respectively.

Figure 1 shows the procedure of data processing mentioned above. We filtered noise data through session analysis, and added positive samples from social graph. Then built set $(\hat{r}_{ui}, \hat{r}_{uj})$ for pairwise training.

4. MULTIFACETED FACTORIZATION MODELS

This section describes our approach named Multifaceted Factorization Models (MFM) used for user interest prediction. The motivation of this approach is to integrate all kinds of valuable features in social networks to improve the recommendation effects. MFM extends the SVD function to:

$$\hat{r}_{ui} = \ddot{b}_{ui} + d_{ui} + \ddot{\mathbf{q}}_i^T \ddot{\mathbf{p}}_u \quad (11)$$

where $\ddot{b}_{ui}, d_{ui}, \ddot{\mathbf{q}}_i, \ddot{\mathbf{p}}_u$ are defined in the following equations:

$$\begin{aligned} \ddot{b}_{ui} = & \mu + b_u + b_i + b_{hour} + b_{day(i)} + b_{sec(i)} \\ & + b_{u,gender(i)} + b_{u,age(i)} + b_{gender(u),i} + b_{age(u),i} \\ & + b_{kw(u,i)} + b_{tag(u,i)} + b_{tweetnum(u)} \end{aligned} \quad (12)$$

$$\begin{aligned} d_{ui} = & \left| R^k(u, i) \right|^{-\frac{1}{2}} \sum_{j \in R^k(u, i)} (r_{ui} - \bar{r}_{uj}) w_{ij} \\ & + \left| N^k(u, i) \right|^{-\frac{1}{2}} \sum_{j \in N^k(u, i)} c_{ij} \end{aligned} \quad (13)$$

$$\ddot{\mathbf{q}}_i = \mathbf{q}_i + \mathbf{z}_{day(i)} + \mathbf{z}_{sec(i)} \quad (14)$$

$$\begin{aligned} \ddot{\mathbf{p}}_u = & \mathbf{p}_u + \mathbf{y}_{age(u)} + \mathbf{y}_{age(u),gender(u)} + \mathbf{y}_{tweetnum(u)} \\ & + |S(u)|^{\alpha_1} \sum_{k \in S(u)} \mathbf{y}_k + |A(u)|^{\alpha_2} \sum_{l \in A(u)} \mathbf{y}_l \\ & + W(u, m) \sum_{m \in K(u)} \mathbf{y}_{kw(m)} + |T(u)|^{-\frac{1}{2}} \sum_{n \in T(u)} \mathbf{y}_{tag(n)} \end{aligned} \quad (15)$$

Detailed descriptions of these functions are given below.

4.1 Date-Time Dependent Biases

Temporal information is proved to be very effective for recommendation[9][4]. Firstly, users' action differs when time changes. For instance, in the day time a user who is busy with work may be less willing to follow and tweet others than in the evening, or a user's behavior may vary between weekdays and weekends. Secondly, the popularities of items change over time. As a result, we incorporate temporal information to \ddot{b}_{ui} and $\ddot{\mathbf{q}}_i$ in our solution.

Firstly, we designated three time factors as global biases and added them to b_{ui} in Eq.(12).

$$b_{hour} = b_{Bin(t)} \quad (16)$$

$$b_{day(i)} = \frac{t - d^-}{d^+ - d^-} b_{day(i)}^- + \frac{d^+ - t}{d^+ - d^-} b_{day(i)}^+ \quad (17)$$

$$b_{sec(i)} = \frac{t - s^-}{s^+ - s^-} b_{sec(i)}^- + \frac{s^+ - t}{s^+ - s^-} b_{sec(i)}^+ \quad (18)$$

where t is the timestamp of a recommendation record. $Bin(t)$ denote the bin index associated with the hour (a number between 1 and 24). d^- and d^+ indicate the begin time and end time of the entire dataset. s^- and s^+ indicate the begin time and end time of a day. In our experiment they are set to 1318348785, 1322668798, 0, and 86400 respectively.

Apart from the global biased, date and time factors can also be used as latent factors of items (shown in Eq.14). Regardless of whether the items were followed or not, the

date and time factors of the recommender records can tell us about item's preferences. So we extended the item-factors vector \mathbf{q}_i by adding $\mathbf{z}_{day(i)}$ and $\mathbf{z}_{sec(i)}$, which are defined by functions similar to linear Eq.(17) and (18). Note that we did not add time factors to \mathbf{p}_u , because we found that temporal information for users is very sparse and has little positive effect to the performance.

4.2 Integrate Social Relationships

A phenomenon observed from testing data is that 77.1% users do not have any rating records in the training set. Therefore it is very difficult to predict these users' interests merely through explicit feedbacks. Fortunately, due to social relationships, people are often influenced in attitude, taste or actions by their peer groups. M. Jamali[7] and Ma[5] proved that exploiting social network could alleviate the data sparse problem and improve the recommendation accuracy, especially for those cold start users.

In this section, two kinds of social relationships are introduced and used as the implicit feedbacks. The first and foremost implicit feedback is the user's SNS followships. Given a user u , we denote $S(u)$ as the set of users followed by u . Similar to SVD++ approach [2], we incorporated S_u factors to \mathbf{p}_u , which means that one user's interests can be influenced by the users he followed. The user latent vector \mathbf{p}_u is extended to:

$$\mathbf{p}_u + |S(u)|^{\alpha_1} \sum_{k \in S(u)} \mathbf{y}_k \quad (19)$$

Besides, we also explored the user actions, including "comment", "retweet", and "at". If user u has an action to user v , there may exist some latent common interests between them. The factorization model is then updated to:

$$\mathbf{p}_u + |S(u)|^{\alpha_1} \sum_{k \in S(u)} \mathbf{y}_k + |A(u)|^{\alpha_2} \sum_{l \in A(u)} \mathbf{y}_l \quad (20)$$

$A(u)$ is the set of users u had given actions to. α_1 and α_2 are parameters for normalization. We set them to -0.4 and -0.5 in the experiments.

It should be mentioned that a effective preprocess we did is to add each u to $S(u)$, which means that user u is a default follower of himself. Experimental result shows that it can improve the accuracy of recommendation.

4.3 Utilizing Keywords and Tags

Keywords are extracted from the tweet/retweet/comment of users, or from their self-introductions. Each keyword m has a weight $W(u, m)$ represents the relevance to user u . These weights are pre-calculated and provided by competition organizers. While tags are selected by users to give a summary of the topics they are interested in. Keywords and tags use different vocabularies. Both of them were used to explore users' latent interests.

Firstly, if user u share the same keyword or tag with item i , this keyword or tag can be seen as a feature of the common interests between u and i . The deviation of this keyword can be trained through the explicit feedbacks. So we added

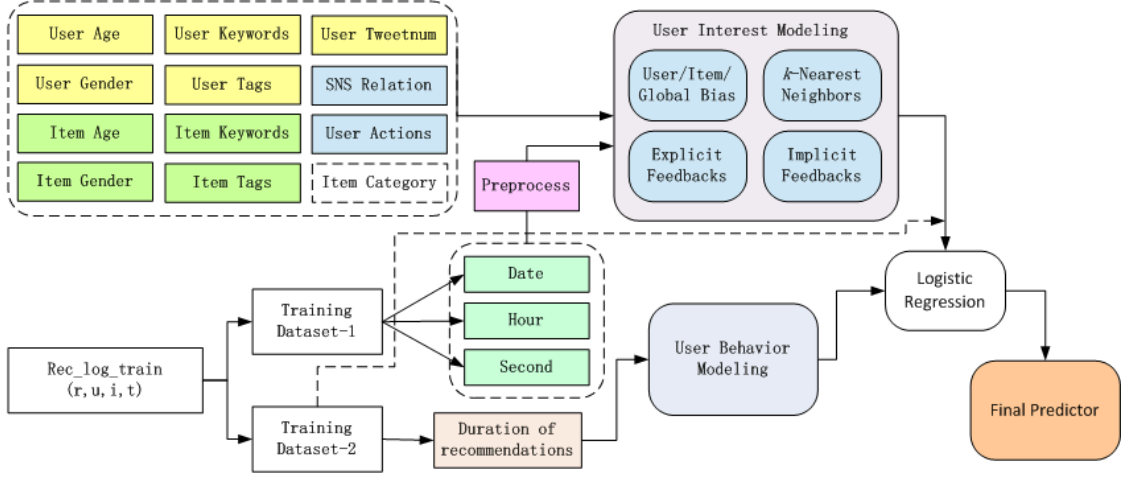


Figure 2: Context-aware ensemble of multifaceted factorization models

global bias $b_{kw(u,i)}$ and $b_{tag(u,i)}$ to \tilde{b}_{ui} in Eq.(12), where:

$$b_{kw(u,i)} = \sum_{m \in K(u) \cap K(i)} b_m \quad (21)$$

$$b_{tag(u,i)} = \sum_{n \in T(u) \cap T(i)} b_n \quad (22)$$

Here $K(\cdot)$ denotes the keyword set of a given user and $T(\cdot)$ denotes the corresponding tag set.

The second method to make use of keywords and tags is to consider them as latent factors. We believe that keywords and tags can be presented as parts of user-factors vector, so the following two terms were added:

$$W(u, m) \sum_{m \in K(u)} y_{kw(m)} + |T(u)|^{-\frac{1}{2}} \sum_{n \in T(u)} y_{tag(n)} \quad (23)$$

We found that weights $W(u, m)$ of the keywords from tweets have already been well normalized. As for every user, $W(u, m)$ satisfy

$$\sum_{m \in K(u)} W(u, m)^2 \approx 1 \quad (24)$$

On the other hand, keywords from user descriptions (i.e. *weight* = 2.0) have a lower quality, since they are extracted directly from the descriptions and without any postprocess. So in our models we only utilized the former.

4.4 k-Nearest Neighbors

The k -Nearest Neighbor approach is also used to improve the results. As shown in Eq.(13), $R^k(u, i)$ is the intersection between user u 's explicit feedbacks and its k -nearest neighbors $D^k(u)$, and $N^k(u, i)$ is the intersection between implicit feedbacks and $D^k(u)$. The implicit feedback set used here is sns relationship $S(u)$. \bar{r}_{uj} is an average rating of u pre-calculated. We used it as the approximation of r_{uj} . The item-item weights w_{ij} and c_{ij} represent the adjustments that need to make to the predicted rating of item i , given a known rating \bar{r}_{uj} of neighbor j .

Tags and keywords were used to calculate the distance between items and find out the neighbors. We denote $dis(i, j)$

as the distance between items i, j , and calculate it through the following linear function:

$$dis(i, j) = \rho * dis_{kw}(i, j) + (1 - \rho) * dis_{tag}(i, j) \quad (25)$$

where $dis_{tag}(i, j)$ and $dis_{kw}(i, j)$ are distances between two items in the vector space of tags and keywords respectively. ρ is a paramter set to 0.6. We mapped item i to a vector $kw(i)$ with the keywords $m \in K(i)$ and weight $W(i, m)$. Cosine distance was then used to calculate the distance $dis_{kw}(i, j)$ between two vectors $kw(i)$ and $kw(j)$. $dis_{tag}(i, j)$ was obtained in the same way.

4.5 User and Item Profiles

The profiles of users and items, such as gender and age, were proved to be very effective in modeling user interests. Users of comparable ages may share some common tastes. For example, users born in 1960s may be more familiar with the actors popular a few decades ago. Gender is another valuable feature for user interest modeling. An example is that males generally are more interested in military topics than females. So we added four bias term $b_{u,gender(i)}$ and $b_{u,age(i)}$ to the prediction formula. They represent the users' preferences to particular groups of items, who have the same gender or age. While $b_{gender(u),i}$ and $b_{age(u),i}$ represent the derivation of how attractive the item to particular type of users. We denote user u 's age as x , and

$$age(u) = \begin{cases} 0, & \text{if } x < 1950 \\ \text{ceil}((x - 1950)/2) + 1, & \text{if } 1950 \leq x < 2004 \\ 28, & \text{if } x \geq 2004 \\ 29, & \text{if } x \text{ is illegal.} \end{cases} \quad (26)$$

We also integrate $y_{age(u)}$ and $y_{age(u),gender(u)}$ in Eq.(15) to influence user vector p_u , because age and gender can also be handled as implicit feedbacks.

Tweet number of users is a very minor profile. However, we still use it through adding $b_{tweetnum(u)}$ in Eq.(12) and $y_{tweetnum(u)}$ in Eq.(15). We wrap it to the range of $[0, 15]$ by a log function beforehand.

Some other features, including categories of items, are abandoned, because they were considered to be low quality and did not bring positive effect in our models.

5. CONTEXT-AWARE ENSEMBLE

In this section, we first introduce the framework of our 2-level ensemble method. Then detailed introductions to user behavior modeling and training process are given. Finally, a supervised regression used to combine different predictor results is also introduced. The overall view of the workflow is presented in Figure 2.

5.1 Framework

Shown in Figure 2, our solution can be divided into three core stages. The first stage is the user interest modeling. We extend the SVD model to the proposed multifaceted factorization models in order to incorporate various features in social networks. The second stage is user behavior modeling using durations of recommendations. The user interest model and user behavior model is ensembled in the final stage. Logistic regression is used for a linear combination. During the ensemble, not only the predictor results of current timestamp is considered, but also the user's behaviors before and after are involved. In other words, the ensemble incorporates the "context" of recommendations.

The data used in the framework is set as follows. We split the original training set into two parts: Φ_1 is the records from first 23 days, and Φ_2 is those from the next 7 days. We firstly trained the multifaceted factorization models on Φ_1 , and made predictions on Φ_2 . Then we modeled user behaviors and trained our context-aware ensemble based on data Φ_2 . Lastly the whole training set was used to train the final predictor. After that, the final predictions were obtained on testing data.

5.2 User Behavior Modeling

The goal of the competition is to recommend up to three items that the user was most likely to follow. Yet there are two necessary conditions for the user to follow the item: 1) the user noticed the items recommended to him, 2) the item aroused his interests.

As stated earlier, the factorization models were used to locate users' interests. So in this section we introduce the solution about predicting whether the user noticed the recommendation area. We found that the durations of users on each recommendation are very valuable clues to model users' behaviors on social network website. For example, a user who was tweeting with friends and refreshing the page may have a shorter duration on the website than those who were reading the webpage carefully.

In Eq.(5) we already defined the time interval $\Delta t_s(u)$. Before using temporal information, we first spread $\Delta t_s(u)$ into discrete bins $\delta_s(u)$ to alleviate data sparsity.

$$\delta_s(u) = \begin{cases} -1, & \text{if } s < 0 \\ k, & \text{if } 0 \leq s < m, \text{ and } \theta_k \leq \Delta t_s(u) < \theta_{k+1} \\ -1, & \text{if } s \geq m \end{cases}$$

We set bin number to 16 in our solution. Parameters θ_k were assigned to have the bin well-distributed. $\delta_s(u)$ was

Table 1: Performances of Different Multifaceted Factorization Models on level 1

ID	Description	MAP
1	basic matrix factorization($d=40, \eta=0.001$)	0.3442
2	1+pairwise training	0.3458
3	2+preprocess of the training data	0.3495
4	3+sns as implicit feedback	0.3688
5	4+action as implicit feedback	0.3701
6	5+day as bias and item factor	0.3721
7	6+second as bias and item factor	0.3728
8	7+age&gender as bias and user factor	0.3854
9	8+user tags as bias and user factor	0.3863
10	9+user keywords as bias and user factor	0.3887
11	10+tweetnum as bias and user factor	0.3892
12	optimize parameters in 11($d=100, \eta=0.00025$)	0.3901
13	12+nearest neighbors	0.3907
14	13+supplimental training data	0.3911

used for modeling users' browsing habit in social networks. Let us denote $P(\delta_s)$ as the probability that a user noticed the recommendation area when the duration was δ_s . Then the training dataset Ω^- and Ω^+ were used to calculate the probabilities.

In addition, users' behavior not only can be predicted through the duration of current webpage, but also can be analyzed according to the durations of the webpages before and after. For instance a user who held a very short duration before and after, and took a long time on the current page, may have a higher probability to pay attention to the recommended items. Thus, we have the duration factor $\Gamma(r)$, where r denotes the range of the context considered. The formulas of $\Gamma(r)$ used in our solution is listed below.

$$\Gamma(1) = P(\delta_s) \quad (27)$$

$$\Gamma(2) = P(\delta_{s-1}, \delta_s) \quad (28)$$

$$\Gamma(3) = P(\delta_{s-1}, \delta_s, \delta_{s+1}) \quad (29)$$

$$\Gamma(4) = P(\delta_{s-2}, \delta_{s-1}, \delta_s) + P(\delta_{s-1}, \delta_s, \delta_{s+1}) \quad (30)$$

$$\Gamma(5) = P(\delta_{s-2}, \delta_{s-1}, \delta_s) + P(\delta_{s-1}, \delta_s, \delta_{s+1}) + P(\delta_s, \delta_{s+1}, \delta_{s+2}) \quad (31)$$

$\Gamma(r)$ will be blended with multifaceted factorization models through supervised regression method.

5.3 Supervised Regression

It has been proved that combination of different predictors can lead to significant performance improvement over individual algorithms[6]. Ensemble a set of predictors' results is a standard regression problem. In our solution, we choose logistic regression as the tool to deal with the problem. The big advantage of logistic regression is clearly the speed. We can generate our final predictor efficiently.

Firstly, we trained the multifaceted factorization models independently based on training dataset Φ_1 , and used it to predict the ratings in set Φ_2 . The predictions from user behavior models were also obtained. Then we got a total of M result vector F_i , where M is the number of individual predictors, and F_i is a N -dimensional vector. N equals the

Table 2: Performance of Context-aware Ensemble on level 2

ID	Description	MAP
1	level-1 models + $\Gamma(1)$ ensemble	0.4180
2	level-1 models + $\Gamma(2)$ ensemble	0.4252
3	level-1 models + $\Gamma(3)$ ensemble	0.4329
4	level-1 models + $\Gamma(4)$ ensemble	0.4365
5	level-1 models + $\Gamma(5)$ ensemble	0.4395

size of validation set $|\Phi_2|$.

$$Z = W_0 + \sum_{i=1}^m W_i * F_i \quad (32)$$

where W_i is the coefficient for different individual models. Z is the prediction vector $Z = (z_0, z_1, \dots, z_{N-1})$. Then prediction value of the i -th record is

$$\hat{r}_i = \frac{1}{1 + \exp(-1 * z_i)} \quad (33)$$

Parameters W_i can be easily obtained through the logistic regression approach. Finally the proposed multifaceted factorization models were trained again using the same initialization parameters on the entire training set, and then the final prediction was obtained on the testing dataset.

6. EXPERIMENTS

This section presents our experimental results. Detail information of the multifaceted factorization models is given in Table 1. We can conclude that user sns relations bring significant benefits to the model. User profiles, such as age and gender, are also very helpful. Since the cold start challenge is grave in the task, these features are very promising supplement to locate interests of inactive users.

Table 2 shows the performance of ensembles, where level-1 models with $\Gamma(5)$ ensemble gives the best performance. Durations of the recommendations show great effectiveness to predict users' response. The utilization of recommendation context also gives improvement. Although more context information brings better performance, the risk of overfitting also grows, so we limit the range of context to 5.

7. CONCLUSIONS

In this paper, we described the context-aware ensemble of multifaceted factorization models for recommendation prediction in social networks. SNS relationships and actions were integrated as implicit feedbacks. Various features, including age, gender, time, tweetnum, keywords, tags, etc., were also used for user interest modeling. A context-aware ensemble and user behavior modeling methods were proposed. Experimental results on KDD-Cup data set demonstrated the effectiveness of the techniques proposed.

8. ACKNOWLEDGMENTS

We would like to acknowledge the SIGKDD organizers for holding this exciting and inspiring competition. We also thank the colleagues at Shanda Innovation Institute for fruitful discussions.

9. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:634–749, 2005.
- [2] R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *SIGKDD Explorations*, 9(2):75–79, 2007.
- [3] J. Bennet and S. Lanning. The netflix prize. In *KDD Cup and Workshop*, 2007.
- [4] T. Chen, Z. Zheng, Q. Lu, X. Jiang, Y. Chen, W. Zhang, K. Chen, Y. Yu, N. N. Liu, B. Cao, L. He, and Q. Yang. Informative ensemble of multi-resolution dynamic factorization models. In *In the KDD-CUP' 2011 Workshop of 17th ACM SigKDD conference.*, San Diego, CA, USA., 2011.
- [5] M. Hao, Z. Dengyong, L. Chao, L. M. R., and K. Irwin. Recommender systems with social regularization. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining, WSDM '11*, pages 287–296, New York, NY, USA, 2011. ACM.
- [6] M. Jahrer and A. Toscher. Combining predictions for accurate recommender systems. *KDD'10 Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 693–702, 2010.
- [7] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.
- [8] Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.
- [9] Y. Koren. Collaborative filtering with temporal dynamics. pages 447–456, 2009.
- [10] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [11] T. Y. Liu. Learning to Rank for Information Retrieval. pages 904–904, 2010.
- [12] Y. Niu, Y. Wang, G. Sun, A. Yue, B. Dalessandro, C. Perlich, and B. Hamner. The tencent dataset and kdd-cup'12. In *KDD-Cup Workshop, 2012.*, 2012.
- [13] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup and Workshop*, pages 39–42, 2007.
- [14] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20 (NIPS'07)*, pages 1257–1264, 2008.
- [15] M. Zhu. Recall, Precision and Average Precision. 2004.