Local Implicit Feedback Mining for Music Recommendation

Diyi Yang, Tianqi Chen, Weinan Zhang, Qiuxia Lu, Yong Yu
Dept. of Computer Science and Engineering
Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai China, 200240
{yangdiyi, tqchen, wnzhang, luqiuxia, yyu}@apex.sjtu.edu.cn

ABSTRACT

Digital music has experienced a quite fascinating transformation during the past decades. Thousands of people share or distribute their music collections on the Internet, resulting in an explosive increase of information and more user dependence on automatic recommender systems. Though there are many techniques such as collaborative filtering, most approaches focus mainly on users' global behaviors, neglecting local actions and the specific properties of music. In this paper, we propose a simple and effective local implicit feedback model mining users' local preferences to get better recommendation performance in both rating and ranking prediction. Moreover, we design an efficient training algorithm to speed up the updating procedure, and give a method to find the most appropriate time granularity to assist the performance. We conduct various experiments to evaluate the performance of this model, which show that it outperforms baseline model significantly. Integration with existing temporal models achieves a great improvement compared to the reported best single model for Yahoo! Music.

Categories and Subject Descriptors

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

Keywords

Collaborative Filtering, Recommender System, Local Implicit Feedback, Efficient Training

1. INTRODUCTION

During the past decades, music has experienced a quite fascinating transformation since expensive records and CDs are replaced by enjoying a lot of music free online. People could ask music store staff or just select from existing records to get what they wanted in the past, while nowadays they can surf on the Internet for music service, and turn to an automatic recommender system for a recommendation as the amount of music available expands explosively.

There are many approaches to music recommendation, such as Collaborative Filtering (CF)[12], which has been widely used and proved to be quite effective in handling users' preferences. Latent factor models, like matrix factorization (MF), and neighborhood models are two canonical approaches in CF to capture users' interests. MF, or singular value decomposition, maps users and items into the same low-dimension space, maintaining vectors of items and users. MF predicts a user's rating on an item through the inner product of two vectors. On the other hand, neighborhood models, put more emphasis on detecting the similarities and correlations between users or items, and make prediction based on observed information. Besides, there are other techniques for recommendation, such as graph-based models [26], and content-based approaches[8].

Though effective in maintaining users' overall preferences, many CF methods concentrate more on users' global interests. Such information might be enough to capture users' preferences in some cases, such as recommending academic papers or movies. However, music recommendation differs from those in several ways[1]. Firstly, music has a large item space and a low consumption time, where each song receives attention for a relatively short time. Secondly, a person can listen to songs in many situations, such as working, jogging or resting. However, he or she cannot read papers or watch movies all the time. Finally, people often consume music continually. That is, a user's behaviors in the next time interval are likely to be consistent with his/her recent behaviors.

Many traditional CF approaches focus on users' *global interests* that stay stable for a long time. They do not take the local consistent behaviors into full consideration[18]. That is, things which have happened just now or very recently are likely to influence users' decisions in the near future. We call these *local preferences*. For example, John might suddenly change his favorite Jazz to light music just because he was going to sleep. Or, a person might be blue and turn to some brooding tracks due to a week of rainy days. Such local information is often neglected and not captured. However, better performance may be achieved if local behaviors are considered. In this paper, we propose a simple and effective local implicit feedback model. We also design an efficient training algorithm to speed up the training. Our main contributions are as follows:

[1]http://recsys.acm.org/2011/tutorials.shtml, by Oscar Celma and Paul Lamere

RecSys'12, September 9–13, 2012, Dublin, Ireland, UK.

- **Local Implicit Feedback Model** We are the first to propose a local implicit feedback model. A combination of local and global information, represented by implicit feedback, can capture users' stable and local changing preferences well.

- **Efficient Training Algorithm** Due to the large amount of data to be processed and the high cost, we design an efficient training algorithm. It greatly decreases the complexity when using the classical stochastic gradient descent method.

- **Time Granularity** We verify our local model by testing with different time granularties, in order to get the most appropriate granularity to maximize performance. We also propose a method to find the optimal time granularity.

Experiments conducted on three different datasets using two evaluation metrics show that our local model outperforms baselines in capturing users' local preferences in both rating and ranking prediction. We also show that our local model is to some extent complementary to other time-aware models. We combine our model with the best single model reported on Yahoo! Music, which achieves an RMSE decrease from 22.346 to 21.879.

The remainder of this paper is organized as follows. In Section 2, we present our local implicit feedback model. Our efficient training algorithm is described in Section 3, and experiments are discussed in Section 4. Related work is in Section 5. We conclude our work and point out some future directions in Section 6.

2. LOCAL IMPLICIT FEEDBACK MODEL

In this section, we present our local implicit feedback model that extracts users' local preferences, and also present the ranking approach we used. To demonstrate that this model is complementary to time-aware models, we integrate our model with some temporal approaches.

2.1 Local Preferences

Local user preferences, as stated above, are mainly used to capture user's varying behaviors in a very local time period. That is, his/her actions during current time interval might have effects on his/her decisions in the next few minutes. His/her actions in the next time period tend to be consistent with his/her recent behaviors, but might be different from his/her global behaviors. For example, if Alice is a music lover keen on Hip Hop and Rock, but something unfortunate has happened so she is depressed at present, then it is likely for her to turn to some sad songs. It is possible that people might change their accustomed habits temporarily and return to them after a short time. Habits can also change quite gradually, and those changes might persist for a long time. Transient events affect users' preferences during a short time period, like a week, a day, an hour or even a minute. On one hand, a user's interests might be affected by his/her established preferences; on the other hand, they could be also influenced by a very local event, like a new record release or a song that rises to fame overnight. Moreover, such very local and transient incidents, tend to be more determined by the characteristic of music that people's consumption is always continuous. However, such contextual information might be unavailable to recommender systems. To solve this problem, we propose to use users' local behaviors to model their local preferences.

Representations	Descriptions
U,I	user set, item set
u,i	user u, item i
p_u,q_i	K dimension vectors of user u and item i
$N(u)$	the set of items rated by user u
$N(u,t)$	the set of items rated by user u during time period t
b_u, b_i	users/item bias
μ	overall average ratings
$bias$	b_u, b_i and μ
ϕ_i	global predictive influence of item i a K dimension vector
φ_j	local predictive influence of item j a K dimension vector

Table 1: Commonly Used Notations

2.2 Local Implicit Feedback

User local preferences are characterized by using users' implicit feedback in a short time period. The implicit feedback is represented by users' behaviors in history, i.e items he/she rated. Through rated/unrated binary information, the implicit feedback model is provided with a non-explicit ability to capture users' potential and global interests. This model was originally proposed by Koren[12], whose formulation was:

$$\widehat{r}_{ui} = \mu + b_u + b_i + {p_u}^T q_i + (|N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} \phi_j)^T q_i \quad (1)$$

Table 1 gives some commonly used notation for our work. Here, given two items i and j, ϕ_j is an indication of user u's preference, and will be high if j is predictive on item i. Based on MF and implicit feedback, this model focuses more on users' global behaviors. Since implicit feedback works well in characterizing users' global and potential interests, we implement our local preferences idea based on it. Our local implicit feedback model is formulated as this:

$$\widehat{r}_{ui}(t) = b_u + b_i + \left(p_u + \frac{\sum_{j \in N(u)} \phi_j}{\sqrt{|N(u)|}} + \frac{\sum_{j \in N(u,t)} \varphi_j}{\sqrt{|N(u,t)|}} \right)^T q_i \quad (2)$$

Here, user' local preferences are characterized by using their very localized rating history represented by $N(u,t)$, which we call **Local Implicit Feedback**. By using users' implicit information during a localized time interval, we assume that user behaviors during that period correlate to their current decisions. The **Time Granularity** is defined as the length of the local time interval, which can be a minute, an hour, a day and even a week. A period larger than a week is beyond what we define as a local time period. For example, if we set the time granularity as day, it means a user's ratings during the current day will be picked as his/her local implicit feedback. The most appropriate time granularity that directly and accurately reflects users' local interests, can be discovered by testing different granularity settings. However, it could be observed that due to differences in various music datasets, the optimal time granularity might be in different datasets. If there are few items rated during a time period, then that granularity might have little ability to capture users' local behaviors. Having implemented our local mod-

el with classical SVD++[12] , we could characterize users' global interests with the global implicit feedback, and local preferences by using the local implicit feedback model. We present detailed statistics in the experimental part.

2.3 Temporal Dynamics Integration

To investigate whether our implicit feedback model can further improve the performance of the existing temporal models, we integrate our model with those existing approaches. Some classical temporal models have already been proposed, such as [13, 5]. Incorporating those time-aware models with our local implicit feedback into an integrated model, we get the following formulation.

$$\widehat{r}_{ui} = \mu + b_i + b_i(t) + b_u + b_u(t) + p_u(t)^T q_i(t) + \left(\frac{\sum_{j \in N(u)} \phi_j}{\sqrt{|N(u)|}} + \frac{\sum_{j \in N(u,t)} \varphi_j}{\sqrt{|N(u,t)|}} \right)^T q_i(t) \quad (3)$$

$p_u(t)$, $q_i(t)$ are used to denote the corresponding p_u and q_i that change over time. This integration model is used to test that our implicit feedback model has an unique ability to discover some unheeded information.

2.4 Ranking Optimization

There are two kinds of prediction tasks, rate prediction and top K ranking recommendation. Rate prediction is a general recommendation orientation [13], and ranking based approaches [17] are popularly used on datasets like Pandora[2] and Last.fm. Both rate prediction and ranking task will be studied in our paper.

To recommend the top K items, we have to rank items over the $\widehat{r}_{ui}$ that directly reflects users' interests. Optimization of ranking order is needed to get the updated parameters. Most traditional ranking approaches maximize the area under the ROC curve as follows:

$$AUC(u) := \frac{1}{|N(u)^+||I \setminus N(u)^+|} \sum_{i \in N(u)^+} \sum_{j \in I \setminus N(u)^+} \delta(\hat{r}_{ui} - \hat{r}_{uj})$$

where $N(u)^+$ is the set of users' liked items in $N(u)$. It means user u prefers items in $N(u)^+$ over $I \setminus N(u)^+$. To optimize the rank order, we first define a hard 0-1 function as this:

$$\delta(x) = \begin{cases} 1 & x > 0 \\ 0 & otherwise \end{cases} \quad (4)$$

$\delta(x)$ is non-differentiable, thus we can replace it with surrogate functions $-l(x)$. One widely-used surrogate function is logistic loss, which is adopted later in our ranking approaches.

$$l(x) = \ln(1 + e^{-x}) \quad (5)$$

3. EFFICIENT TRAINING ALGORITHM

In this section, we present our efficient training algorithm that is used to speed up training in our experiments, after which we provide a complexity analysis.

3.1 Traditional Updating

We can characterize implicit feedback models described in Equation 1 and 2 with the following general implicit feedback model:

$$\widehat{r}_{u,i} = bias + \left(p_u + \sum_{j \in \Gamma(u)} \alpha_j \psi_j \right)^T q_i \quad (6)$$

Here, $\Gamma(u)$ stands for implicit feedback information that could include global implicit feedback described in Equation 1 as well as local feedback in Equation 2. ψ_j is the implicit feedback term, and could be ϕ_j in global implicit feedback or φ_j in local model. Meanwhile, α_j could be given the value $\beta_j = \frac{1}{\sqrt{|N(u)|}}$ for global implicit feedback terms, and $\gamma_j = \frac{1}{\sqrt{|N(u,t)|}}$ for local terms.

The traditional stochastic gradient descent algorithm [14] updates the above model as follows:

$$p_u = p_u + \eta \left(\hat{e} q_i - \lambda_1 p_u \right) \quad (7)$$

$$q_i = q_i + \eta \left(\hat{e} p_u - \lambda_2 q_i \right) \quad (8)$$

$$\psi_j = \psi_j + \eta \left(\hat{e} \alpha_j q_i - \lambda_3 \psi_j \right), \forall \alpha_j \neq 0, \alpha_j \in \Gamma(u) \quad (9)$$

We focus on p_u, q_i and ψ_j; the rules for updating *bias* are omitted. The λs are regularization parameters. η is the learning rate, and $\hat{e}$ is the difference between actual and predicted ratings. Obviously, the cost of updating ψ_j is linearly related to the number of non-zero entries in $\Gamma(u)$, i.e. proportional to the number of items the users have rated.

3.2 Efficient Training

The traditional updating procedure becomes expensive when the average number of items rated by users is large. For example, last.fm customers have listened to 3996 songs on average, so their updates would be at a great cost. When we add more implicit feedback information, that problem becomes more prominent. Thus, some optimization methods are needed.

Before turning to our efficient algorithm, let us do some observations on the general model first. Define a derived user implicit feedback factor p^{im} as follows:

$$p^{im} = \sum_j \alpha_j \psi_j \quad (10)$$

The updating rule of ψ_j after one step is like this (omitting regularization terms):

$$\Delta \psi_j = \eta \hat{e} \alpha_j q_i \quad (11)$$

Difference in p^{im} before and after updating is as follows:

$$\Delta p^{im} = \eta \hat{e} (\sum_j {\alpha_j}^2) q_i \quad (12)$$

With more care, we find that there exists a relation between Δp^{im} and $\Delta \psi_j$, which is formulated as follows:

$$\Delta \psi_j = \frac{\alpha_j}{\sum_k {\alpha_k}^2} \Delta p^{im} \quad (13)$$

Therefore, to get new Δp^{im}, we do not need to update each ψ_j. For logistic loss or square loss, those update rules and relations also remain valid. However, if the L_2 regularization term is added, update rules changes to this:

$$\Delta \psi_j = \eta (\hat{e} \alpha_j q_i - \lambda \psi_j) \quad (14)$$

Corresponding differences in p^{im} change similarly. Those changes do not affect the use of this relation, since the regularization term is usually small.

[2]http://www.pandora.com

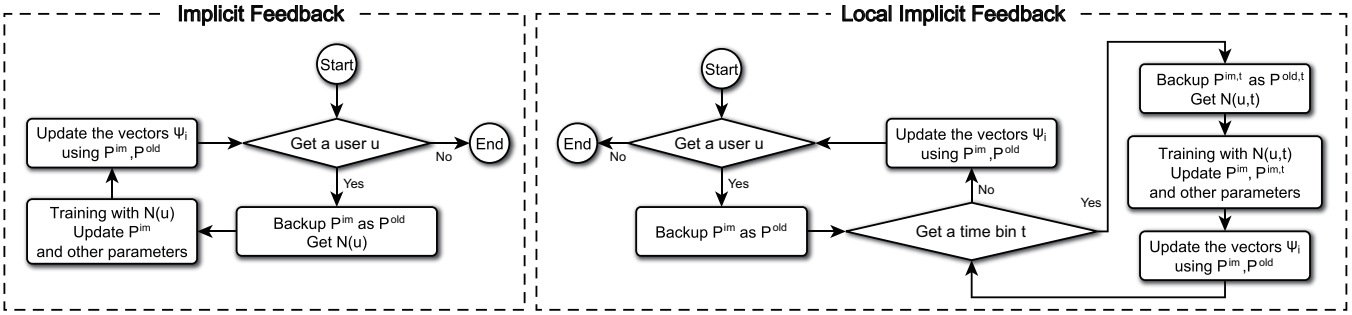


Figure 1: Comparison of Efficient Training Algorithm of Global and Local Implicit Feedback Models

This general relation is used to speed up the stochastic gradient descent training, and can also handle all variants of implicit feedback models, such as SVD++ in Equation 1 and local implicit feedback in Equation 2. We develop an efficient algorithm for our local implicit feedback model described in Equation 2 by using the general relation in Equation 13. The detailed algorithm is presented as Algorithm 1. The general idea is that we do not need to update each ϕ_j or φ_j. Instead, we directly update p^{im} and $p^{im,t}$, and return changes to ϕ_j and φ_j using the relation in Equation 13. Figure 1 presents an intuitive comparison of local and global implicit feedback models using the efficient training algorithm. It is obvious that efficient training of SVD++ is just a special case of Algorithm 1.

Algorithm 1 Efficient Training Algorithm for Training with Local Implicit Feedback

for all user u **do**
  $p^{im} \leftarrow \sum_{j \in N(u)} \beta_j \phi_j$
  $p^{old} \leftarrow p^{im}$
  for all time intervals of user u **do**
    $p^{im,t} \leftarrow \sum_{j \in N(u,t)} \gamma_j \varphi_j$
    $p^{old,t} \leftarrow p^{im,t}$
    for all training examples in $N(u,t)$ **do**
      update relevant parameters
      replace $\sum_j \gamma_j \varphi_j$ with $p^{im,t}$,$\sum_j \beta_j \phi_j$ with p^{im}
      update $p^{im,t}$ and p^{im} directly
    end for
    for all i, $\gamma_j \neq 0$ **do**
      $\varphi_j \leftarrow \varphi_j + \frac{\gamma_j}{\sum_k {\gamma_j}^2}(p^{im,t} - p^{old,t})$
    end for
  end for
  for all i, $\beta_j \neq 0$ **do**
    $\phi_j \leftarrow \phi_j + \frac{\beta_j}{\sum_k {\beta_j}^2}(p^{im} - p^{old})$
  end for
end for

3.3 Complexity Analysis

Compared with classical stochastic gradient descent algorithm, our efficient training algorithm achieves a significantly reduction in time complexity. Algorithm 1 has a time complexity of $O(N_e K + U\overline{R})$ for general implicit feedback form, i.e. $O(N_e K)$, usually $K > 1$. N_e stands for all non-zero entries, and K refers to the latent dimension. U is the number of users, while $\overline{R}$ represents the average number of items rated by a user. The cost of updating p^{im} for all users amounts to the product of dimension and all non-zero entries of user-item rating samples. While the complexity of updating all ψ_j for all users is approximately the number of non-zero entries for all users. Thus, the overall complexity is the product of all non-zero entries and dimension. Compared to the cost of updating p^{im}, this can be neglected. However, since original stochastic gradient descent needs to update ψ_j whenever you get a updated p^{im}, its complexity is $O(N_e K\overline{R})$. Obviously, $\xi = \frac{O(N_e K\overline{R})}{O(N_e K)} = O(\overline{\mathbf{R}})$. That is, our training algorithm performs $\overline{\mathbf{R}}$ times faster than original one, where $\overline{\mathbf{R}}$ is larger than 100 in general.

4. EXPERIMENT

In this section, we present our experimental results to evaluate the proposed local implicit feedback models. There are several questions we want to answer.

- Can this local implicit feedback model perform better than a model with only global implicit feedback?
- Which time granularity is the most powerful in reflecting users' local changing preferences? Could we give some methods to find a relatively optimal time granularity?
- Could our local implicit feedback model extract users' potential preferences that complement the existing time-aware models?

4.1 Experimental Setup

Our experiments are conducted on three datasets, Yahoo! Music[3], Last.fm[4], and Douban Music[5]. Yahoo! Music dataset is used by 2011 KDD Cup Workshop track 1. Last.fm is a famous personalized music website. We use the dataset provided by Celma et al.[4]. Douban Music is the largest Chinese music recommendation website. We crawl the dataset from users' recent listening history along with time stamps. Table 2 is a statistical comparison for each dataset. Here, STG stands for the Smallest Time Granularity provided by the datasets or available, and AIR is the Average Items Rated per user. Traditional rating prediction is performed on Yahoo! Music, and is evaluated with RMSE(root mean square error). We train local models on the provided training set and test them on the validation set. Personalized ranking is implemented on the remaining Last.fm and Douban Music datasets. We follow the top-N recommendation evaluation metric proposed by Cremonsei

[3]http://kddcup.yahoo.com
[4]http://last.fm
[5]http://music.douban.com

Dataset	User	Item	Ratings	STG	AIR
Yahoo! Music	1000990	624961	253M	Min	252
Last.fm	1001	471997	3.95M	Min	3996
Douban Music	22454	497744	2.94M	Day	88

Table 2: Comparison of DataSets

et al.[6]. First, we randomly 4 : 1 split the original training sets into training and validation sets. Then we get our final testing sets by sampling 1000 not rated items for each high rating example in the validation sets. Recall@K is used as evaluation metric like this:

$$Recall@K = \frac{H}{T} \tag{15}$$

For each positive example with 1000 negative samples, it is called a hit if the positive example is ranked in the top K among the 1001 items. H is the overall hit number. T is the total number of items that users like in the test set.

4.2 Performance Comparison

We conduct several kinds of experiments on different datasets to evaluate the performances of purely global implicit feedback model and our local implicit feedback models.

4.2.1 Model Names

To fully illustrate the local models later, detailed descriptions of relevant models referred to Equation 2 are given as follows:

- IMFB: is for an implicit feedback model based on basic MF and global information, i.e the classical SVD++.
- MLIF: is minute local implicit feedback, regarding a minute's ratings as local implicit information.
- HLIF: is hour local implicit feedback model, using an hour' ratings as local implicit information.
- DLIF: is day local implicit feedback, treating a day's ratings as local information.
- WLIF: is week local implicit feedback model, taking a week's ratings as local information.

The MLIF, HLIF, DLIF and WLIF models are all implemented with IMFB, a combination of global and local implicit feedback information. IMFB is regarded as a baseline. The five models are all trained with our efficient training algorithm, and are built based on the time granularity available in the corresponding dataset. Through comparisons we set the latent factor as 100 in our experiments.

4.2.2 RMSE Performance on Yahoo! Music

Since Yahoo! Music is a rate prediction task, RMSE is adopted as our evaluation metric. It is appropriate to this specific dataset. The detailed RMSE results of the five models are in Figure 2. From the RMSE results of each of the models with different time granularity, we can get some basic observations. Firstly, we find that our proposed approaches do give significant improvements in this rate prediction task. MLIF, which achieves the best performance in terms of accuracy, attains an RMSE of 22.673. While our baseline model IMFB gives 23.233. It can be also observed that the performances of the day and week models are similar, with RMSE of 22.922 and 22.972 respectively. No matter what

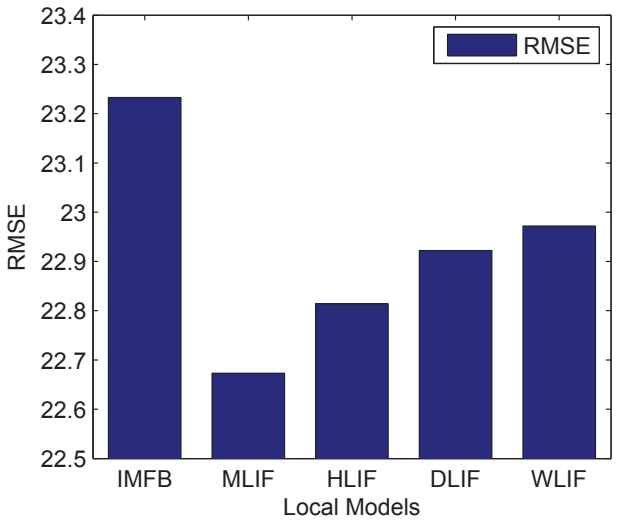


Figure 2: RMSE of Local Models on Yahoo! Music

Models,Recall@K	5	10	15	20
IMFB	0.173	0.240	0.304	0.335
MLIF	0.175	0.250	0.305	0.339
HLIF	0.220	0.288	0.338	0.373
DLIF	**0.224**	**0.301**	**0.345**	**0.383**
WLIF	0.224	0.297	0.341	0.380

Table 3: Recall of Localized Models on Last.fm

time interval is used, a local model al ways outperforms the baseline, though degrees of improvement are different. Thus, we can conclude that a local implicit feedback model works better than a purely global model. Secondly, it seems possible that a smaller time granularity gives better performance. That is, MLIF works better than DLIF, while DLIF is superior to others except the minute model. However, it might not be true on all datasets. The different performance among various time granularity reflect different descriptions of users' local behaviors and we will discuss this in detail in Section 4.3. We do not present the basic MF method here, since it is generally inferior to SVD++[12].

4.2.3 Recall Performance on Last.fm

In this section, we report the performance of our local implicit feedback models. From Table 2, the STG of Last.fm is minute. Based on time granularity available, minute, hour, day and week local implicit feedback models are constructed. We give the comparison of the best local implicit feedback against the global one in Figure 3. The detailed recall results of all models are shown in Table 3. It is evident that our implicit feedback model achieves a significant improvement over the baseline. Our localized model DLIF achieves a recall of 0.383 when $K = 20$, which means an item that a user likes has a probability of 38.3% to be ranked in the top 20 in DLIF during a large sample collection (1001 items). While the baseline gives a Recall@20 of 0.335. That is, day local implicit feedback has a 14% improvement over the baseline. Such big progress indicates that our model does outperform a purely global model significantly. Table 3 shows detailed recall results for the local implicit feedback model. From Table 3, we know that compared to the

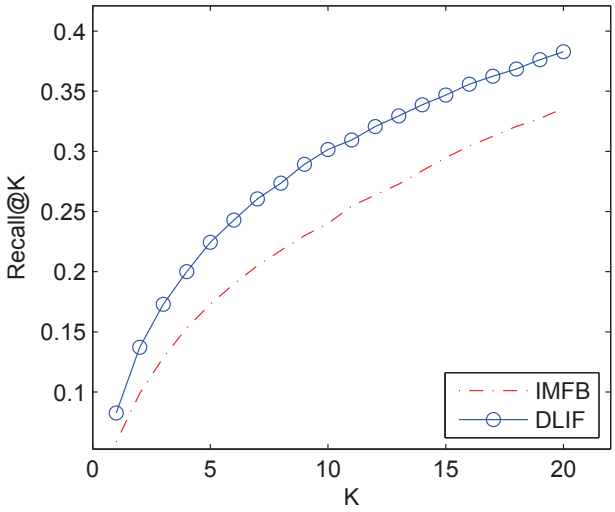


Figure 3: Recall@K of Localized and Baseline Models on Last.fm

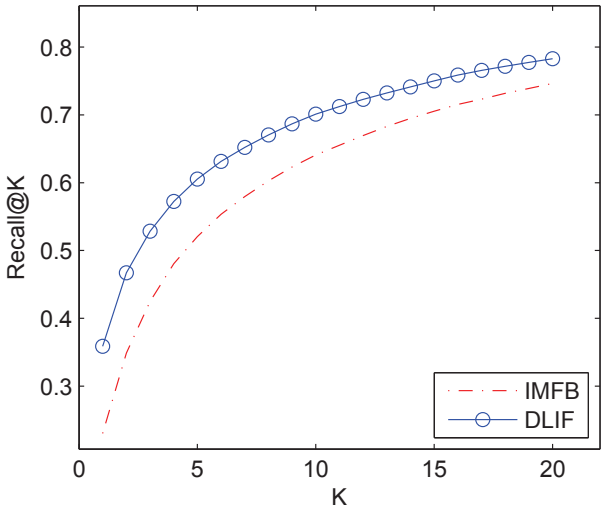


Figure 4: Recall@K of Localized and Baseline Models on Douban

Models,Recall@K	5	10	15	20
IMFB	0.520	0.640	0.705	0.746
DLIF	**0.605**	**0.701**	**0.750**	**0.783**
WLIF	0.602	0.697	0.739	0.780

Table 4: Recall of Localized Models on Douban

baseline, local implicit feedback models give improvement to different extents. Obviously, DLIF, HLIF and WLIF give similar performance. HLIF and WLIF give Recall@20 of 0.373 and 0.380 respectively, and DLIF is a little higher than either of them. Moreover, MLIF gives approximately the performance of IMFB. With a recall@20 of 0.339, it is a bit better than IMFB. The experiments show that our local implicit feedback model can achieve significant improvement over baseline on ranking task.

4.2.4 *Recall Performance on Douban*

This dataset is handled similarly to Last.fm, and shares the same evaluation methodology with it. Experiments are conducted on three models, IMFB, DLIF and WLIF. We present our results on Douban Music as Table 4, and comparison curves as Figure 4. Figure 4 gives us an abstract impression that a local implicit feedback model outperforms IMFB. As anticipated, a local implicit feedback model gives a consistent improvement in terms of top K precision from Table 4. With a Recall@20 of 0.746 for IMFB, DLIF and WLIF outperform it with 0.783 and 0.780 respectively. Such improvement also can be observed from Recall@5, Recall@10 and Recall@15 in Table 4. Therefore, overall performances show again that local implicit feedback models do work in capturing users' preferences that are neglected by a global model.

4.3 Resolution of Time Granularity

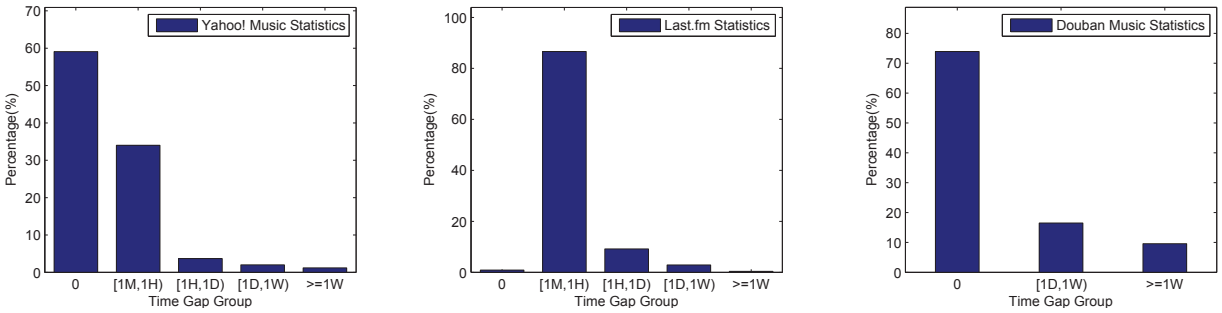
In this part, we evaluate the performance of our local implicit feedback models under different time granularity, and propose an empirical standard to find the most appropriate time granularity for each dataset. For convenience, define ARP(u) as the set of time-adjacent rating pairs of user u, and ARP as the union of all users' ARP(u). Statistics are done on the percentage of all ARP pairs which are categorized into five groups by their time differences. Time gap groups include: 0(No time difference[6]), $[1M, 1H)$, $[1H, 1D)$, $[1D, 1W)$ and larger than 1 week. M, H, D, W refer to minute, hour, day and week respectively. The results are presented in Figure 5. The statistics is a direct reflection of users' behavior frequency distributions.

From statistics of Yahoo! Music in Figure 5(a), we find that nearly 60% ARP pairs have no timestamp difference. Recalling the results in Figure 2, we find the minute model gives the best performance among all time granularity. This can be explained by our statistics since minute is the *smallest granularity that contains sufficient information*: 60% of ARP pairs have no time difference, indicating that sufficient feedback information can be obtained within a minute. Under the condition of sufficient information, a smaller granularity can lead to a better modeling of users' behaviors.

In Last.fm, it appears that most ARP pairs lie in group $[1M, 1H)$. Using the day as time granularity allows us to cover sufficient information. The results in Table 3 suggest that day is the most proper time granularity, which achieves the best Recall@20 of 0.383. This is consistent with our previous observation. In Douban, we can see that almost 75% of ARP pairs have no difference with each other. The results in Table 4 show that day is the most appropriate. This is also consistent with our previous observation since day is the smallest time granularity with sufficient information.

Through comparison, we discover that the appropriate time granularity depends more on the property of specific dataset, and generally varies in different datasets. From the experiment results, we observe that the most proper time granularity is given by the smallest granularity that contains sufficient information. The most appropriate time granularity are minute, day and day on Yahoo! Music, Last.fm and Douban respectively. Moreover, by analyzing statistics on

[6]This means there is no difference in timestamp, due to the limitation of minimal timestamp available on dataset



(a) Yahoo! Music (b) Last.fm (c) Douban Music

Figure 5: Statistics on Yahoo! Music, Last.fm and Douban Music

Models	RMSE
Integrated.MF	22.346
Integrated.Minute.LIF	**21.879**

Table 5: RMSE Performance on Yahoo! Music

each dataset, we can choose the appropriate time granularity using the empirical standard.

4.4 Integration with Temporal Dynamics

The experimental results presented in Section 4.2 show that local implicit feedback performs well on music datasets. Using a combination of global and local implicit feedback information, our local implicit feedback models capture not only their global interests, but also users' local behaviors that could be indicated through the local rating entries. However, it can be argued that, even if our models captures users' changing interests, it might be largely due to some general influences of changes that existing time-aware models could handle. To answer this question, we conduct a comparison experiment to show that local model captures an unique property of music that differs from properties mined by existing time-aware models on the Yahoo! Music dataset. Here, Integrated.MF stands for the model adopted in [5], the reported best single model on Yahoo! Music, which contains state-of-the-art temporal dynamics modeling approaches. Integrated.Minute.LIF, represents the combination of a local model and Integrated.MF referred to Equation 3. We take the best local implicit feedback model on Yahoo! Music here to integrate with Integrated.MF. That is, Integrated.Minute.LIF is a combination of local implicit feedback (MLIF) and a state-of-the-art temporal approach. Results are shown in Table 5. The combination gives a significant improvement from 22.346 to 21.879, better than all local models on Yahoo! Music dataset. Therefore, it can be concluded that a local implicit feedback model is complementary to time-aware models, and combining it with the best single model reported can provide better performance. In conclusion, our proposed local model does have an unique capability to discover users' potential preferences from a different perspective.

5. RELATED WORK

Collaborative filtering has been an effective technique for recommender systems during the past years. Compared with content-based models [25], CF approaches do not necessarily need any attributes of users/items. In general, CF models are often adaptive and flexible since they learn from the dynamically changing user feedback [16]. Neighborhood based models [23, 21] and latent factor models [14] are two canonical approaches to CF. Recently, context-aware models [1] are proposed to capture users' behaviors on specific context. Context-aware models always leverage users' context information such as mood [24], location [19], social relationship [3] etc.. However, these contextual information are often unavailable.

User implicit feedback plays an important role in information retrieval and data mining applications[2, 11]. Due to the limitations on obtaining explicit feedback (e.g. detailed ratings), implicit feedback, has been paid much attention [22] due to availablility. For some recommender system applications, users do not always return their explicit feedback due to application limitations. Thus, implicit feedback can be leveraged to improve the recommendation performance. Hu et al. [10] studied CF implicit feedback datasets and proposed to transform users' implicit feedback into training data in a preference-conference format. Koren [12] discovered that incorporating implicit feedback into a neighborhood integrated latent factor model (SVD++) could give significant improvement. However, training with implicit feedback becomes quite expensive in general recommender systems, and is not well studied. G. Takacs proposed a specific unified approach of factor and neighbor based models for large recommender systems[9], differing from our general efficient training one.

Since users' preferences generally change over time, recommender systems should capture both global and local changing interests in order to provide more accurate recommendations. Temporal dynamic models distinguish users/items from different time slots so as to maintain users' changing preferences. Koren [13] proposed a time day bin approach in capturing users' fixed preferences during a time period. Xiang et al.[26] designed a session-based temporal graph and applied a personalized random walk on it. The graph has three types of nodes for users, items, and user sessions, and is used to capture users' long term and short term interests. Chen et al. [5] proposed a multi-resolution temporal CF model that consists of time-dependent user/item bias, latent factors, time-dependent neighborhood, and rating session to cope with Yahoo! Music recommendation. Rendle et al. [20] proposed personalized Markov chains together with MF in capturing both sequential effects and long

term user-state, which focuses on influences of sequential actions on the next action. There are also other recommendation approaches involving temporal information in various ways, such as decreasing weights for old data [7], and time-dependent iterative prediction in a growing dataset[15].

6. CONCLUSION

In this paper, we propose a simple and effective local implicit feedback model to mine users' local interests. We also design an efficient training algorithm to speed up the training procedure. Experiments conducted on three datasets show that our local implicit feedback models significantly outperform the global implicit feedback model, and have a capability different from existing time-aware models in capturing users' changing preferences. Meanwhile, we observe that the appropriate resolution changes on different datasets, and also give an empirical standard to determine the most appropriate time granularity. In the future, we plan to take this further to extract personalized local implicit feedback information and other localized properties.

7. ACKNOWLEDGEMENT

Yong Yu is supported by grants from NSFC-RGC joint research project 60931160445.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. *Recommender Systems Handbook*, pages 217–253, 2011.

[2] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2006.

[3] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proceeding of the 17th international conference on World Wide Web*, pages 199–208. ACM, 2008.

[4] O. Celma. *Music Recommendation and Discovery in the Long Tail*. Springer, 2010.

[5] T. Chen, Z. Zheng, Q. Lu, X. Jiang, Y. Chen, W. Zhang, K. Chen, Y. Yu, N. Liu, B. Cao, L. He, and Q. Yang. Informative ensemble of multi-resolution dynamic factorization models. In *KDD-Cup Workshop*, 2011.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[7] Y. Ding and X. Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 485–492. ACM, 2005.

[8] A. Ferman, J. Errico, P. Beek, and M. Sezan. Content-based filtering and personalization using structured metadata. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 393–393. ACM, 2002.

[9] B. D. G.Takacs, I.Pilaszy. A unified approach of factor models and neighbor based methods for large recommender systems. In *In Proc. of ICADIWT-08, 1st IEEE Workshop on Recommender Systems and Personalized Retrieval*, pages 186–191, 2008.

[10] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. IEEE, 2008.

[11] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2005.

[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*, KDD '08, pages 426–434, New York, NY, USA, 2008. ACM.

[13] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 447–456, 2009.

[14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42, August 2009.

[15] N. Lathia, S. Hailes, and L. Capra. Temporal collaborative filtering with adaptive neighbourhoods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 796–797. ACM, 2009.

[16] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data (Second Edition)*. Springer Verlag, 2011.

[17] 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.

[18] J. R. Lorraine McGinty. *On the Evolution of Critiquing Recommenders*. Springer, 2011.

[19] M. Park, J. Hong, and S. Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. *Ubiquitous Intelligence and Computing*, pages 1130–1139, 2007.

[20] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 811–820, New York, NY, USA, 2010. ACM.

[21] 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.

[22] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2005.

[23] 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.

[24] Y. Shi, M. Larson, and A. Hanjalic. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 34–40. ACM, 2010.

[25] R. Van Meteren and M. Van Someren. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, 2000.

[26] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 723–732. ACM, 2010.