

A Time-based Recommender System using Implicit Feedback

T. Q. Lee

Department of Mobile Internet
Dongyang Technical College
Seoul, Korea

Y. Park

Dept. of Computer Science & Information Systems
Bradley University
Peoria IL, USA

Abstract - Recommender systems provide personalized recommendations on products or services to customers. Collaborative filtering is a widely used method of providing recommendations based on explicit ratings on items from other users. However, in some e-commerce environments such as a mobile environment, it is difficult to collect explicit feedback data; only implicit feedback is available.

In this paper, we present a method of building an effective collaborative filtering-based recommender system for an e-commerce environment without explicit feedback data. Our method constructs pseudo rating data from the implicit feedback data. When building the pseudo rating matrix, we incorporate temporal information such as the user's purchase time and the item's launch time in order to increase recommendation accuracy.

Based on this method, we built a recommender system for a mobile e-commerce environment and conducted several experiments. Empirical results show our collaborative filtering-based recommender system to be significantly more accurate than a pure collaborative filtering system.

Our approach can be applied to building collaborative filtering-based recommender systems for any e-commerce environment without explicit feedback.

Keywords: E-commerce, recommender system, implicit feedback, temporal information, mobile environment

1 Introduction

A variety of recommender systems have been developed with the growth of e-commerce [1,2]. These systems provide personalized recommendations on products or services to customers, who then spend less time searching for the right product or service.

Collaborative filtering is a widely used and proven method of providing recommendations. Most collaborative filtering-based recommender systems rely on explicit feedback that is collected directly from users. Ratings and reviews are typical examples of explicit feedback. Because it is easier to quantify ratings than reviews, in practice most collaborative filtering methods use rating data. Collaborative filtering algorithms focus on similarity among users or similarity among items using users' ratings.

For collaborative filtering-based recommender systems, the recommendation quality depends on the quality of feedback. When users rate honestly, using rating information is one of the best ways to quantify user preferences. However, many users assign arbitrary ratings that do not reflect their true opinions.

In some environments, it is either impossible or very difficult to obtain explicit feedback. For example, in a mobile environment the service fee is charged based on the duration of the connection. To minimize this fee, users want to reduce the connection time as much as possible. Thus, it is not practical to expect users' active participation in ratings. In this case, we have to rely solely on implicit feedback. The amount of implicit feedback data is usually enormous. However, the amount of relevant data that can be used for recommendation is relatively small and can only be extracted through preprocessing. This log-based recommender system may not be suitable for real time recommendations in a large-scale e-commerce environment.

In some e-commerce environments, recommendation accuracy is extremely important. For example, the display size of mobile devices is still limited. The recommender system should ideally provide a small set of highly relevant items that fit in the display.

In this paper, we present a novel method of building a collaborative filtering-based recommender system with

high accuracy for an e-commerce environment even in the absence of explicit feedback:

- *In our method, we construct pseudo rating data from the implicit feedback data for collaborative filtering. When building the pseudo rating matrix, we incorporate temporal information – user purchase time and item launch time – in order to increase recommendation accuracy. (Section 2) We can tune the rating function so that it provides the highest recommendation accuracy.*
- *Based on the method, we built a recommender system for a mobile e-commerce environment and conducted several experiments. (Section 3) Empirical results from several experiments have shown our method to be effective.*
- *Our method can be applied to building effective collaborative filtering-based recommender systems for any e-commerce environment without explicit feedback.*

Our approach will be especially useful because we can build an effective recommender system based on collaborative filtering (which is recognized as an effective method) only using implicit feedback such as purchase information and temporal information.

The product launch time was used in [3] in order to improve the performance of the collaborative filtering using explicit ratings. The user rating time was considered to improve the precision of item-based collaborative filtering in [4]. However, our collaborative filtering constructs a rating matrix from implicit feedback by using both item launch time and user purchase time.

2 Time-based Collaborative Filtering with Implicit Feedback

2.1 The Problem and Our Approach

Most collaborative filtering-based recommender systems use explicit feedback (ratings) that are collected directly from users. When users rate truthfully, using rating information is one of the best ways to quantify user preferences. However, many users assign arbitrary ratings that do not reflect their honest opinions. In some e-commerce environments, it is difficult to ask users to give ratings. For instance, in a mobile e-commerce environment the service fee is dependent on the connection time.

This paper deals with the problem of building an effective collaborative filtering-based recommender system for an e-commerce environment without using explicit feedback data. The main idea of our approach is constructing pseudo rating data from the implicit feedback data. When building the pseudo rating matrix, we incorporate temporal information such as user purchase time and item launch time in order to increase recommendation accuracy.

2.2 Pseudo-Rating Matrix Using Temporal Information

Because there is no explicit user feedback, we construct a pseudo rating matrix from implicit feedback such as purchase information. This pseudo rating matrix will be used for collaborative filtering.

A Simple Pseudo Rating Matrix: In a simple pseudo rating matrix, we can simply assign 1 as a rating value when a user u purchased an item i .

For example, consider an environment with three users and four items. The purchase information is as follows: User A purchased Item 1, 3 and 4. User B purchased Item 2 and 3. User C purchased Item 1 and 4.

Table 1. A Simple Pseudo Rating Matrix from Purchase Information

	Item 1	Item 2	Item 3	Item 4
User A	1		1	1
User B		1	1	
User C	1			1

The simple pseudo rating matrix is given in Table 1. In general, the time when each item was launched and the time when each user purchased it are different.

Table 2. Item Launch Time and User Purchase Time

	Item 1 :LTime 1	Item 2 :LTime 2	Item 3 :LTime 3	Item 4 :LTime 4
User A	PTime 1		PTime 2	PTime 3
User B		PTime 4	PTime 5	
User C	PTime 6			PTime 7

Suppose the above example has the temporal information summarized in Table 2. Here, LTime means the item's launch time and PTime means the user's purchase time.

A Time-based Pseudo Rating Matrix: We incorporate two kinds of temporal information - the time when the item was launched and the time when the user purchased an item - into the simple pseudo rating matrix. We make two observations:

- *More recent purchases better reflect a user's current preference.*
- *Recently launched items appeal more to users.*

Based on these observations, we define a rating function w that computes rating values (rather than simply assigning 1) as follows:

$w(pi, lj) =$ The rating value when an item with launch time lj was purchased at time pi .

Table 3. The Pseudo Rating Matrix using Temporal Information

	Item 1	Item 2	Item 3	Item 4
User A	$w(p1,l1)$		$w(p2,l3)$	$w(p3,l4)$
User B		$w(p4,l2)$	$w(p5,l3)$	
User C	$w(p6,l1)$			$w(p7,l4)$

The pseudo rating matrix for the above example after considering both launch time and purchase time is given in Table 3.

Although the pseudo rating matrix is not constructed directly by users, to some extent it reflects their preferences. We may design different rating functions and compare their accuracy. In this way, we choose the one that fits best with the given e-commerce environment.

2.3 Our Time-Based Recommender System

Our time-based recommender system consists of the following phases:

- *Collect implicit feedback data*
- *Construct a pseudo rating matrix*
- *Compute neighbors*

- *Recommend items*

Collect implicit feedback data: We collect two kinds of data – the user purchase and purchase time data and the item launch time data. This data is usually available in a typical e-commerce environment and will be used to construct a pseudo rating matrix.

Construct a pseudo rating matrix: We define a rating function and construct a pseudo rating matrix using the computed rating values. The rating function will depend on the type of product or service to be recommended.

If users are sensitive to the item's launch time, then the rating function should give more weight to new products or services. If the user's purchase time is important in estimating his or her current preference, then the rating function should give more weight to recent purchases.

We can find the right rating function by first designing several candidate rating functions. Using training data we compare the effectiveness of each rating function to the given product or service domain and choose the best one.

Compute similar neighbors: Given an active user, we compute neighbors whose preferences are similar to those of the active user's from the pseudo rating matrix. We then use two similarity measures – the Pearson correlation coefficient and the Cosine similarity. Let a and b be two users, P_{aj} be the user a 's current preference to the item j , and $\overline{P_a}$ be the user a 's current average preference. The Pearson correlation coefficient is defined as follows:

$$P_sim(a,b) = \frac{\sum_j (P_{aj} - \overline{P_a})(P_{bj} - \overline{P_b})}{\sqrt{\sum_j (P_{aj} - \overline{P_a})^2} \sqrt{\sum_j (P_{bj} - \overline{P_b})^2}}$$

The Cosine similarity is defined as follows:

$$C_sim(a,b) = \frac{\sum_j (P_{aj})(P_{bj})}{\sqrt{\sum_j (P_{aj})^2} \sqrt{\sum_j (P_{bj})^2}}$$

The top k neighbors are chosen based on their similarity values to the active user.

Recommend items: We recommend the top m items among all items that are recommended by the k neighbors. Let a and t be two users, j be an item, $sim(a, t)$ be the similarity value between user a and user t . The predicted score function is defined as follows:

$$PS(a, j) = \frac{\sum_t sim(a, t)(P_{jt} - \bar{P}_t)}{\sum_t sim(a, t)}$$

3 Experiments and Results

In order to observe the effectiveness of system, we have conducted several experiments on real-world data in a mobile e-commerce environment. We compared the accuracy of our time-based recommender system with the recommender system using pure collaborative filtering.

All simulations were implemented by VBA (Visual Basic for Applications) on Excel worksheets.

3.1 Implicit Feedback Data Set

Korea is considered one of the most advanced countries in the area of Information Technology. SKTelecom is one of the major companies that provide mobile internet services in Korea.

For the implicit feedback data set, we used transaction data from June 2004 to August 2004. In this period, users purchased and downloaded character images (wallpaper) to their mobile devices via the Internet. The total number of users who purchased at least one image was 1,922. The total number of character images is 9,131. The total number of transactions is 65,101.

We used 80% of the total transaction data as training data. We built a time-based recommender system using these training transactions.

The remaining 20% of the total transaction data was used as test data to demonstrate the effectiveness of our time-based recommender system. This was done by comparing the accuracy of our recommendations with those of a pure collaborative filtering-based recommender system for the users in the test transactions.

3.2 Experiment Design

Pseudo Rating Matrix: In our experiments, we used the following rating function. We divided the item launch times into three groups - *old launch*, *middle launch*, and *recent launch* groups. Similarly, we

divided the users' purchase times into three groups - *old purchase*, *middle purchase*, and *recent purchase* groups

Table 4. The Rating Function W

	<i>old purchase group</i>	<i>middle purchase group</i>	<i>recent purchase group</i>
<i>old launch group</i>	0.7	1.7	2.7
<i>middle launch group</i>	1	2	3
<i>recent launch group</i>	1.3	2.3	3.3

Our rating function values are given in Table 4. Note that our rating function assigns higher rating values to recent purchases and recent launches than old purchases and old launches. We then constructed the pseudo rating matrix. Of course, we can design different rating functions by using a different number of groups and assigning them different weights. We can then choose the best rating function among them.

Similar Neighbors and Recommended Items: We used both the Pearson correlation coefficient and the Cosine similarity in order to find neighbors.

In total, we computed 10 similar neighbors to the active user. We tried different numbers of neighbors, but found that the number of neighbors did not affect the accuracy of our recommender system.

We recommended a total of 10 items by considering the size of cell phone displays.

3.3 Empirical Results

In our experiment, we defined accuracy as the ratio of the number of items recommended and purchased to the number of items recommended by the system. We compared our time-based recommender system with the pure collaborative filtering-based recommender system.

3.3.1 Results with Pearson correlation coefficient similarity

In the case of the Pearson correlation coefficient, we need to compute the standard deviation. Thus, we have

to include the users who did not purchase the items by using the rating value 0.

The number of items recommended by the pure CF system and actually purchased by the users is 123. The number of items recommended by our CF system and actually purchased by the users is 180.

Table 5. Results with Pure CF using Pearson Correlation Coefficient

	pure CF	time-based CF
<i># items recommended & purchased</i>	123	180
<i># items recommended & purchased per user</i>	0.11	0.16

The accuracy comparison of both the pure CF and our proposed CF is presented in Table 5. The number of items purchased per each user is 0.11 items in the case of the pure CF. Our CF improves the number of item purchased per each user to 0.16 items.

3.3.2 Results with Results with Cosine similarity

In the case of using Cosine similarity, the number of items recommended by the pure CF and actually purchased by the users is 127. The number of items recommended by our CF and actually purchased by the users is 170.

Table 6. Results with Pure CF using Cosine Similarity

	pure CF	time-based CF
<i># items recommended & purchased</i>	127	170
<i># items recommended & purchased per user</i>	0.11	0.15

The accuracy comparison of both the pure CF and our proposed CF is presented in Table 6. The number of items purchased per each user is 0.11 items in the case

of the pure CF. Our CF improves the number of item purchased per each user to 0.15 items.

Accuracy improvement of our time-based CF is depicted in Figure 1 and Figure 2.

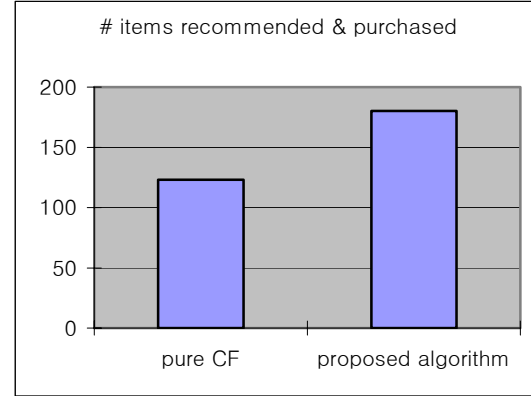


Figure 1. Accuracy Improvement under Pearson Correlation Coefficient

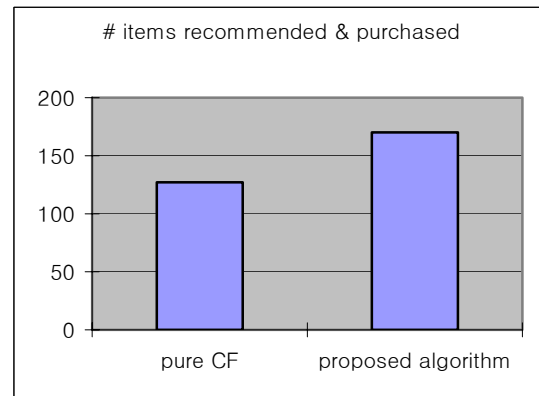


Figure 2. Accuracy Improvement under Cosine Similarity

Table 7. Accuracy Improvements over pure CF

Accuracy Improvement	Pearson Correlation Coefficient	Cosine Similarity
<i># items recommended & purchased</i>	46 %	34 %
<i># items recommended & purchased per user</i>	46 %	34 %

3.4 Summary & Discussions

As shown in Table 7 and Figure 1 and 2, our CF showed accuracy improvement by 46 % over the pure CF under the Pearson correlation coefficient similarity measure. Under the Cosine similarity measure, our CF improved accuracy by 34 % over the pure CF. This result might be dependent on the type of items. In our experiment, the items are character images targeted mainly to younger users. This might have somewhat affected the degree of accuracy improvement. However, we do not believe that this factor is significant.

In both cases, we can see significant accuracy improvement by incorporating temporal information. This confirms our observation that more recent purchases better reflect a user's current preference and more recently launched items better appeal to users.

4 Comparison with Related Work

Content-based and Collaborative Filtering:

There are three main approaches to recommendation – content-based, collaborative filtering and hybrid. Content-based recommendation (or information filtering) is the method of suggesting recommendations to an active user based on descriptive features associated with items and the user's ratings of items [5,6].

Collaborative filtering is the widely used method of providing item recommendations to an active user based on ratings from other users. There are two basic methods of automatic collaborative filtering – user-based and item-based. GroupLens is an example of user-based collaborative filtering [7,8].

User-based collaborative filtering algorithms focus on the similarity among users and thus have problems with scalability as the number of users increases. Item-based collaborative filtering algorithms improve scalability by focusing on the similarity among items using user ratings rather than on the similarity among users themselves [9].

The recommender system proposed in this paper can be viewed as collaborative filtering that incorporates content feature, such as item launch time.

Using Time Information: Two kinds of time information - product launching time or user rating time - have been used in collaborative filtering using explicit

ratings to improve either recommendation accuracy or performance.

The product launch time was used in [3]. By using temporal features of items, they presented a technique to scale down candidate sets in the context of movie recommendations. They used movie production years to improve the performance of the collaborative filtering-based recommender system. It was also reported that the inclusion of production year improved the accuracy of the recommender system.

The user rating time was considered in [4] in order to improve the precision of item-based collaborative filtering. An algorithm to compute the weights based on rating times for different items was proposed.

In our method, we construct a rating matrix from implicit feedback by using both item launch time and user purchase time.

Implicit Feedback: Examples of implicit feedback include user purchasing patterns, Web page visits, and Web surfing paths. A recommender system based on information built by analyzing a Web log was introduced in [10,11]. The amount of data in the log is usually enormous. But, the amount of relevant data that is useful for recommendation is relatively small and can only be extracted through preprocessing. A Web log-based recommender system may not be suitable for real time recommendation in a large-scale e-commerce environment.

Our method focuses on building a recommender system based on the collaborative filtering technique with implicit feedback.

5 Conclusion and Future Work

We have presented a novel approach to building an effective collaborative filtering-based recommender system based on implicit feedback. The main idea of our approach is the construction of pseudo rating data from implicit feedback data by incorporating temporal information – user purchase time and item launch time – to achieve better recommendation accuracy. We can tune the rating function such that it provides the highest level of accuracy.

Empirical results from several experiments in a mobile e-commerce environment support this idea. Our approach can be applied to any e-commerce environment where it is impossible or difficult to collect explicit feedback data. Future work will include investigations of

- *other types of information (other than temporal) that can be incorporated in the pseudo rating matrix for accuracy improvement,*
- *sensitivity of various temporal information to accuracy, and*
- *new similarity measures that are more suitable to the pseudo rating matrix.*

6 Acknowledgments

We would like to thank Dr. Y. H. Cho for permitting us to share the Data Set. This work is supported in part by the Caterpillar Research Fellowship.

4 References

- [1] Recommended systems Special Issue. *Communication of ACM*, 40(3), March 1997.
- [2] Adomavicius, G. and Tuzhilin, A. Toward 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(6), 2005, 734 – 749.
- [3] Tang, T. Y., Winoto, P., and Chan, K. C. C. Scaling Down Candidate Sets Based on the Temporal Feature of Items for Improved Hybrid Recommendations. *Intelligent Techniques in Web Personalization 2003*, Springer, LNAI 3169, 169-185.
- [4] Ding, Y. and Li, X. Orłowska M. Recency-Based Collaborative Filtering. *Australian Computer Science Communications*, Vol., 28 No 2, Australasian Database Conference 2006 (ADC06), Tasmania, Jan 2006, ACM Digital Library, 99-107.
- [5] Balabanovic, M. and Shoham, Y. Fab: Content-based, collaborative recommendation. *Communications of the Association for Computing Machinery*, 1997, 40(3), 66-72.
- [6] Mooney, Raymond J. and Roy, Loriene. Content-Based Book Recommending Using Learning for Text Categorization. In *Proceedings of DL-00*, 5th ACM Conference on Digital Libraries, June 2000, 195 - 204.
- [7] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of CSCW '94*, Chapel Hill, NC.
- [8] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3), 1997, 77-87.
- [9] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based collaborative filtering recommendation algorithms, In *Proceedings of ACM International World Wide Web Conference (WWW10)*, Hong Kong, 2001, 285 - 295
- [10] Mobasher, B., Dai, H., Luo, T., Sun, Y., and Zhu, J. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM*, Vol. 43(8), 2000, 142-151.
- [11] Anderson, C. R., Domingos, P., and Weld, D. S. Personalizing Web Sites for Mobile Users. In *Proceedings of the 10th Conference on the World Wide Web (WWW10)*, 2001.