Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 3055–3062
Expert Systems with Applications
www.elsevier.com/locate/eswa

A time-based approach to effective recommender systems using implicit feedback

Tong Queue Lee [a], Young Park [b,*], Yong-Tae Park [c]

[a] *Department of Mobile Internet, Dongyang Technical College, Seoul, Republic of Korea*
[b] *Department of Computer Science and Information Systems, Bradley University, Peoria, IL 61625, USA*
[c] *Department of Industrial Engineering, Seoul National University, Seoul, Republic of Korea*

Abstract

Recommender systems provide personalized recommendations on products or services to customers. Collaborative filtering is a widely used method of providing recommendations using explicit ratings on items from users. In some e-commerce environments, however, 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 both user-based and item-based collaborative filtering-based recommender systems for character images (wallpaper) in a mobile e-commerce environment and conducted a variety of experiments. Empirical results show our time-incorporated recommender system is significantly more accurate than a pure collaborative filtering system.

Keywords: E-commerce; Recommender system; Collaborative filtering; Implicit feedback; Temporal information; Mobile environment

1. Introduction

A variety of recommender systems have been developed with the growth of e-commerce (Adomavicius & Tuzhilin, 2005; Resnick & Varian, 1997). 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

* Corresponding author. Tel.: +1 309 677 2457; fax: +1 309 677 4504. *E-mail address:* young@bradley.edu (Y. Park).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.06.031

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 higher 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 improve recommendation accuracy (Section 3). We can tune the rating function so that it provides the highest recommendation accuracy.
- Based on the method, we built recommender systems for a mobile e-commerce environment and conducted a variety of experiments on both user-based and item-based collaborative filtering with different rating functions (Section 4). Empirical results from these experiments have shown our method to be effective.
- Our method could be applied to building effective collaborative filtering-based recommender systems for an 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 Tang, Winoto, and Chan (2003) 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 Ding, Li, and Orlowska (2006). However, our collaborative filtering constructs a rating matrix from implicit feedback by using both item launch time and user purchase time.

2. 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 (Balabanovic & Shoham, 1997; Mooney, Raymond, & Roy, 2000).

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 (Konstan et al., 1997; Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994).

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 (Sarwar, Karypis, Konstan, & Riedl, 2001).

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 launch 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 Tang et al. (2003). 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 Ding et al. (2006) 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 Mobasher, Dai, Luo, Sun, and Zhu (2000). 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.

3. Time-based collaborative filtering (CF) with implicit feedback

3.1. Implicit feedback

Most collaborative filtering-based recommender systems use explicit feedback (ratings) that are collected directly

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

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.

3.2. 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 items 1, 3 and 4. User B purchased items 2 and 3. User C purchased items 1 and 4.

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.

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.

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.

3.3. Our time-based approach to CF

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. These data are 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.

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

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	$w(p6,l1)$			$w(p7,l4)$

Compute similar neighbors: In the case of user-based CF, 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_{ai} be the user a's current preference to the item i, and $\overline{P_a}$ be the user a's current average preference. The Pearson correlation coefficient for user-based CF is defined as follows:

$$P\_\mathrm{sim}^{(U)}(a,b)=\frac{\sum_i(P_{ai}-\overline{P_a})(P_{bi}-\overline{P_b})}{\sqrt{\sum_i(P_{ai}-\overline{P_a})^2}\sqrt{\sum_i(P_{bi}-\overline{P_b})^2}}$$

The Cosine similarity for user-based CF is defined as follows:

$$C\_\mathrm{sim}^{(U)}(a,b)=\frac{\sum_i(P_{ai})(P_{bi})}{\sqrt{\sum_i(P_{ai})^2}\sqrt{\sum_i(P_{bi})^2}}$$

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

For item-based CF, we compute correlation between items from the pseudo rating matrix. We then use two similarity measures – the Pearson correlation coefficient and the Cosine similarity. Let i and j be two items, P_{ai} be the user a's current preference to the item i, and $\overline{P_i}$ be the item i's current average preference. The Pearson correlation coefficient for item-based CF is defined as follows:

$$P\_\mathrm{sim}^{(I)}(i,j)=\frac{\sum_a(P_{ai}-\overline{P_i})(P_{aj}-\overline{P_j})}{\sqrt{\sum_a(P_{ai}-\overline{P_i})^2}\sqrt{\sum_a(P_{aj}-\overline{P_j})^2}}$$

The Cosine similarity for item-based CF is defined as follows:

$$C\_\mathrm{sim}^{(I)}(i,j)=\frac{\sum_a(P_{ai})(P_{aj})}{\sqrt{\sum_a(P_{ai})^2}\sqrt{\sum_a(P_{aj})^2}}$$

The top l neighbors are chosen based on their similarity values to every item.

Recommending items: In the case of user-based CF, we recommend the top m items among all items that are recommended by the k neighbors. Let a and c be two users, i be an item, $\mathrm{sim}(a,c)$ be the similarity value between user a and user c. The predicted score function is defined as follows:

$$\mathrm{PS}^{(U)}(a,i)=\frac{\sum_c\mathrm{sim}(a,c)(P_{ci}-\overline{P_c})}{\sum_c\mathrm{sim}(a,c)}+\overline{P_a}$$

For item-based CF, we recommend the top m items among all items that are recommended by the k neighbors. Let a be a user, i and k be items, $\mathrm{sim}(i,k)$ be the similarity value between item i and item k. The predicted score function is defined as follows:

$$\mathrm{PS}^{(I)}(a,i)=\frac{\sum_k\mathrm{sim}(i,k)(P_{ak})}{\sum_k\mathrm{sim}(i,k)}$$

4. 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 Visual Basic for Applications (VBA) on Excel worksheets.

4.1. Implicit feedback dataset

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 dataset, 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 1922. The total number of character images is 9131. 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.

4.2. Experiments design

Pseudo rating matrix: In our experiments, we used two kinds of rating functions – one coarse rating function called W_3 and the other fine rating function called W_5.

The rating function W_3 is designed as follows: 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.

The rating function W_3 assigns to total nine combinations of groups rating values ranging from 0.7 to 3.3 that are shown in Table 4.

Table 4
The rating function W_3

	Old purchase group	Middle purchase group	Recent purchase group
Old launch group	0.7	1.7	2.7
Middle launch group	1	2	3
Recent launch group	1.3	2.3	3.3

Similarly, we design the rating function W_5 as follows: We divided the item launch times into five groups – oldest launch, older launch, middle launch, recent launch, and most recent launch groups. We divided the users' purchase times into five groups – oldest purchase, older purchase, middle purchase, recent purchase and most recent purchase groups.

The rating function W_5 assigns to 25 combinations of groups rating values ranging from 0.2 to 5.0 that are shown in Table 5.

Note that our rating functions W_3 and W_5 both assign 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 items recommendation: We used both the Pearson correlation coefficient and the Cosine similarity in order to find neighbors. In total, we computed 10 similar neighbors each to the active user and item. 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.

Table 5
The rating function W_5

	Oldest purchase group	Older purchase group	Middle purchase group	Recent purchase group	Most recent purchase group
Oldest launch group	0.2	1.2	2.2	3.2	4.2
Older launch group	0.4	1.4	2.4	3.4	4.4
Middle launch group	0.6	1.6	2.6	3.6	4.6
Recent launch group	0.8	1.8	2.8	3.8	4.8
Most recent launch group	1.0	2.0	3.0	4.0	5.0

Accuracy of recommended items: 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 collaborative filtering-based recommender system with the pure collaborative filtering-based recommender system.

Types of CF: We constructed recommender systems using both user-based and item-based collaborative filtering.

4.3. Empirical results – user-based CF

Empirical 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 user-based CF system and actually purchased by the users is 123. The numbers of items recommended by our time-based user-based CF systems and actually purchased by the users are 180 for W_3 and 181 for W_5, respectively.

The accuracy comparison of both the pure user-based CF and our proposed time-based user-based CF is presented in Table 6. The number of items purchased per each user is 0.11 items in the case of the pure user-based CF. Our time-based user-based CF improves the number of items purchased per each user to 0.16 items for both W_3 and W_5.

Accuracy improvement of our time-based user-based CF is depicted in Fig. 1.

Empirical results with Cosine similarity: In the case of using Cosine similarity, the number of items recommended by the pure user-based CF and actually purchased by the users is 127. The numbers of items recommended by our time-based user-based CF systems and actually purchased by the users are 170 for W_3 and 174 for W_5, respectively.

The accuracy comparison of both the pure user-based CF and our proposed time-based user-based CF is presented in Table 7. The number of items purchased per each user is 0.11 items in the case of the pure user-based CF. Our time-based user-based CF improves the number of item purchased per each user to 0.15 items for W_3 and 0.16 items for W_5.

Table 6
User-based CF results with Pearson correlation coefficient

	Pure CF	Time-based CF	
		Rate function W_3	Rate function W_5
# Items recommended and purchased	123	180	181
# Items recommended and purchased per user	0.11	0.16	0.16

Fig. 1. Accuracy improvement of user-based CF under Pearson correlation coefficient.

Table 7
User-based CF results with Cosine similarity

	Pure CF	Time-based CF	
		Rate function W_3	Rate function W_5
# Items recommended and purchased	127	170	174
# Items recommended and purchased per user	0.11	0.15	0.16

Accuracy improvement of our time-based user-based CF is depicted in Fig. 2.

Discussions on time-based user-based CF experiments: As shown in Table 8 and Figs. 1 and 2, our time-based user-based CF showed accuracy improvement by 46% for W_3 and 47% for W_5 over the pure user-based CF under the Pearson correlation coefficient similarity measure. Under the Cosine similarity measure, our time-based user-based CF improved accuracy by 34% for W_3 and 37% for W_5 over the pure user-based 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.

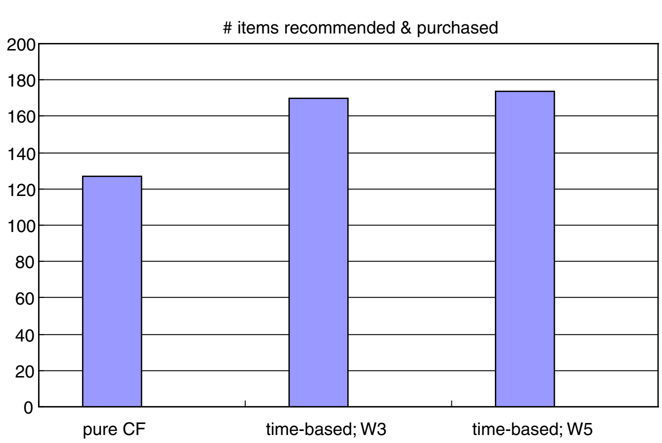

Fig. 2. Accuracy improvement of user-based CF under Cosine similarity.

Table 8
Accuracy improvements of time-based user-based CF over pure user-based CF

Accuracy improvement		Pearson correlation coefficient (%)	Cosine similarity (%)
# Items recommended and purchased	W_3	46	34
	W_5	47	37

4.4. Empirical results – item-based CF

Empirical results with Pearson correlation coefficient similarity: The number of items recommended by the pure item-based CF system and actually purchased by the users is 68. The numbers of items recommended by our time-based item-based CF systems and actually purchased by the users are 84 for W_3 and 85 for W_5, respectively. The number of items purchased per each user is 0.06 items in the case of the pure item-based CF. Our time-based item-based CF improves the number of items purchased per each user to 0.08 items for both W_3 and W_5.

The accuracy comparison of both the pure item-based CF and our proposed time-based item-based CF is presented in Table 9.

Accuracy improvement of our time-based item-based CF is depicted in Fig. 3.

Empirical results with Cosine similarity: In the case of using Cosine similarity, the number of items recommended by the pure item-based CF and actually purchased by the users is 72. The numbers of items recommended by our time-based item-based CF systems and actually purchased by the users are 79 for W_3 and 85 for W_5, respectively. The number of items purchased per each user is 0.06 items in the case of the pure item-based CF. Our time-based item-based CF improves the number of item purchased per each user to 0.07 items for W_3 and 0.08 items for W_5.

The accuracy comparison of both the pure item-based CF and our proposed time-based item-based CF is presented in Table 10.

Accuracy improvement of our time-based item-based CF is depicted in Fig. 4.

Discussions on time-based item-based CF experiments: As shown in Table 11 and Figs. 3 and 4, our time-based user-based CF systems showed accuracy improvement by 24% for W_3 and 25% for W_5 over the pure item-based

Table 9
Item-based CF results with Pearson correlation coefficient

	Pure CF	Time-based CF	
		Rate function W_3	Rate function W_5
# Items recommended and purchased	68	84	85
# Items recommended and purchased per user	0.06	0.08	0.08

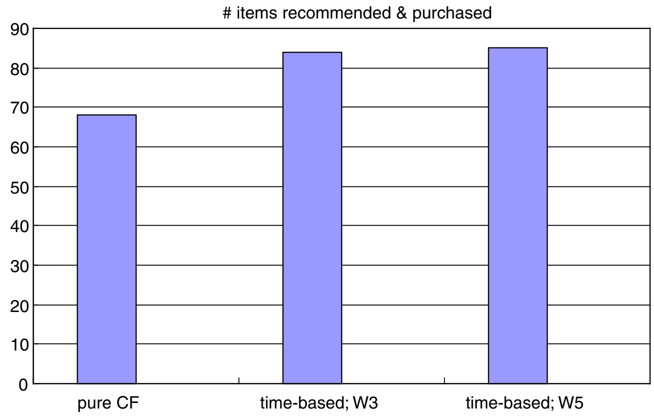

Fig. 3. Accuracy improvement of item-based CF under Pearson correlation coefficient.

Table 10
Item-based CF results with Cosine similarity

	Pure CF	Time-based CF	
		Rate function W_3	Rate function W_5
# Items recommended and purchased	72	79	85
# Items recommended and purchased per user	0.06	0.07	0.08

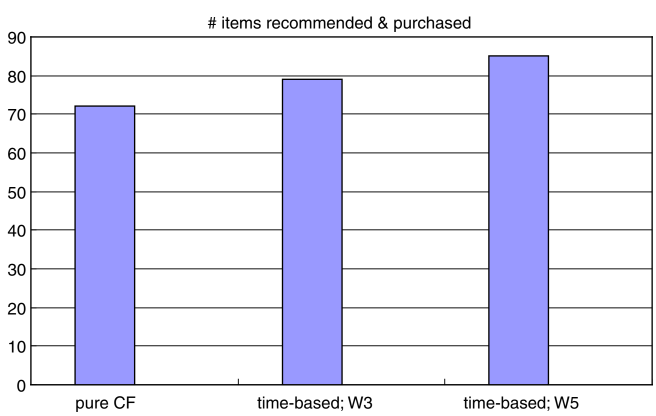

Fig. 4. Accuracy improvement of item-based CF under Cosine similarity.

Table 11
Accuracy improvements of time-based item based CF over pure item-based CF

Accuracy improvement		Pearson correlation coefficient (%)	Cosine similarity (%)
# Items recommended and purchased	W_3	24	10
	W_5	25	18

CF under the Pearson correlation coefficient similarity measure. Under the Cosine similarity measure, our time-based item-based CF systems improved accuracy by 10% for W_3 and 18% for W_5 over the pure item-based CF.

4.5. Experiments summary

From the empirical results we can make the following observations:

- In the case of user-based CF, we can see quite significant accuracy improvement (34–47%) by incorporating temporal information. This confirms our approach that more recent purchases better reflect a user's current preference and more recently launched items better appeal to users.
- In the case of item-based CF, the accuracy improvement (10–25%) is less than that of the user-based CF case. However, we can still achieve significant accuracy improvement by incorporating temporal information. This also confirms our approach.
- The accuracy improvement of the finer rating function W_5 over the coarser rating function $W3$ is not that significant (1–3% for user-based CF and 1–8% for item-based CF). However, we can see an improvement and this confirms our idea of tuning the rating function for the given product or service domain and choosing the best one.

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.

Empirical results from several experiments on collaborative filtering based recommender systems using implicit feedback in a mobile e-commerce environment support this idea. We can tune the rating function such that it provides the highest level of accuracy. Our approach could be applied to an e-commerce environment where it is impossible or difficult to collect explicit feedback data.

Future work will include investigations of

- applying the proposed method to more real-world datasets,
- 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.

Acknowledgements

We thank Dr. Y. H. Cho for permitting us to share the Data Set. This work is supported in part by Dongyang

Technical College Academy Research Expenses and the Caterpillar Research Fellowship.

References

Adomavicius, G., & Tuzhilin, A. (2005). 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), 734–749.

Balabanovic, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the Association for Computing Machinery, 40*(3), 66–72.

Ding, Y., Li, X., & Orlowska, M. (2006). Recency-based collaborative filtering. Australasian database conference 2006 (ADC06), Tasmania, January 2006. *Australian Computer Science Communications, 28*(2), 99–107.

Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., & Riedl, J. (1997). GroupLens: Applying collaborative filtering to usenet news. *Communications of the ACM, 40*(3), 77–87.

Mobasher, B., Dai, H., Luo, T., Sun, Y., & Zhu, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM, 43*(8), 142–151.

Mooney Raymond, J., & Roy, Loriene (2000). Content-based book recommending using learning for text categorization. *ACM Conference on Digital Libraries*, 195–204.

Resnick, P., & Varian, H. R. (1997). Recommender Systems. *Communications of the ACM, 40*(3), 56–58.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of CSCW '94*, Chapel Hill, NC.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of ACM international world wide web conference (WWW10), Hong Kong* (pp. 285–295).

Tang, T. Y., Winoto, P., & Chan, K. C. C. (2003). Scaling down candidate sets based on the temporal feature of items for improved hybrid recommendations. In *Intelligent techniques in web personalization 2003*. Springer, LNAI 3169 (pp. 169–185).