International Journal of Management Science
Vol 11, No 3, Special Issue, December 2005

Strategies for Selecting Initial Item Lists in Collaborative Filtering Recommender Systems

Hong Joo Lee*
Graduate School of Management,
Korea Advanced Institute of Science and Technology

Jong Woo Kim**
Associate Professor, School of Business, Hanyang University
17 Hanendang-dong, Seongdong-gu, Seoul 133-791, Korea

Sung Joo Park***
Graduate School of Management,
Korea Advanced Institute of Science and Technology

(Received May 2005 ; Revised Nov. 2005 ; Accepted Nov. 2005)

ABSTRACT

Collaborative filtering-based recommendation systems make personalized recommendations based on users' ratings on products. Recommender systems must collect sufficient rating information from users to provide relevant recommendations because less user rating information results in poorer performance of recommender systems. To learn about new users, recommendation systems must first present users with an initial item list. In this study, we designed and analyzed seven selection strategies including the popularity, favorite, clustering, genre, and entropy methods. We investigated how these strategies performed using MovieLens, a public dataset. While the favorite and popularity methods tended to produce the highest average score and greatest average number of ratings, respectively, a hybrid of both favorite and popularity methods or a hybrid of demographic, favorite, and popularity methods also performed within acceptable ranges for both rating scores and numbers of ratings.

Keywords: Recommender System, Collaborative Filtering, Initial Item List

* Email: hjlee@kgsm.kaist.ac.kr
** Corresponding Author, Email: kjw@hanyang.ac.kr
*** Email: sjpark@kgsm.kaist.ac.kr

1. INTRODUCTION

Recommender systems have become valuable resources for users seeking intelligent ways to filter the enormous volume of available information [1, 3, 5, 9, 10, 13]. Recommender systems apply data analysis techniques to help customers find desired items at e-Commerce sites [16]. Collaborative filtering is one of the most successful recommendation techniques, and many different applications have used it to recommend news, movies, music, books, etc. [2, 6, 8, 14, 16]. Users can express preferences by rating items they have already tried. Collaborative filtering recommender systems then compare each user's ratings to other users' ratings find the "most similar" users based on some criterion of similarity, and recommend items that similar users have liked in the past [13].

When users first register with a site, a collaborative filtering recommender system has no information about them. Collaborative filtering thus has a "new user" problem because recommendation performance is poor for users who have rated few or no items [8, 13, 14, 17, 18]. Although recommender systems must acquire some information about new users in order to make personalized recommendations, it is helpful for e-Commerce sites to provide an initial item list to learn about and gather preferences of new users. Because it is profitable for e-Commerce site operators to gather new users' ratings as quickly as possible in order to provide effective recommendations, they must find an appropriate strategy for selecting an initial item list [13]. Many e-Commerce sites exploit selection strategies such as user clustering based on demographic information- or popularity-based recommendation [13, 19].

In this study, we designed and analyzed seven selection strategies for providing an initial item list, including the popularity, favorite, clustering, genre, and entropy methods. Each of these strategies is devised without any explicit or implicit user preferences; they select a list of items to be presented to new users when they first register with a site. We used the MovieLens dataset to compare strategy performances based on average rating scores and average rating numbers.

The next section of this paper describes related work on selecting initial item lists; Section 3 suggests various strategies for providing an initial item list to new users; Section 4 compares the performances of the strategies discussed in Section 3, and Section 5 discusses the implications of the analyses, conclusions, and topics for further research.

2. RELATED WORK

According to Rashid et al. (2002), little work has been done to solve the new user problem by analyzing rating data to make informed decisions. Some e-Commerce sites do not make recommendations to new users until users have rated a certain number of products. For example, Nkino.com, a Korean Internet movie portal, makes film recommendations only after a user has rated more than ten movies.

One approach to solving the new user problem involves creating premade user categories or clusters, and quickly assigning new users to one. Partitioning can be accomplished by asking a user predetermined questions such as what movies he or she particularly likes or dislikes, or by clustering users based on demographic information [11, 13, 19]. This approach assigns new users to user categories that exhibit a similar user preference structure or contain similar demographic information, and then recommends the favorite movies of users in the same cluster. Some e-Commerce sites might also ask users to supply the names of their favorite films, actors, singers, athletes, and albums to provide personalized recommendations.

Rashid et al. (2002) applied movie popularity and entropy of ratings to select initial item lists for new users. An example of the population strategy may help clarify this process: when a new customer registers with Amazon.com, the site provides 15 popular items sold in each category. After the user has rated at least one item in the list presented, Amazon.com begins to provide personalized recommendations. The basic concept of entropy in information theory is based on how much information is carried by the signal[1]. A movie that some people hate and others like should supply more information than a movie liked by almost everyone [13]. The entropy of each movie can be calculated using the relative frequency of each rating, and movies with greater entropy are presented to users.

Pennock et al. (2000) explored the use of the expected value of information (VOI) in conjunction with collaborative filtering. To maximize the quality of recommendations, VOI computation can identify at each step the most valuable ratings information to seek from a user [12].

The new user problem is closely related to the sparsity problem of collaborative filtering recommender systems, which results when systems must make recommendations based on very sparse data [5, 7, 14, 15]. To overcome the sparsity

[1] From Wikipedia, http://en.wikipedia.org/wiki/Main_Page

problem, Huang et al. (2004) applied an associative retrieval framework and related spreading activation algorithms to explore transitive associations among consumers through their past transactions and feedback [4]. These approaches exploited item content information or implicit user preferences such as navigation and access patterns. Schein et al. (2001, 2002) used content information—casts of movie actors—along with pure collaborative filtering to alleviate the sparsity problem.

3. STRATEGIES FOR SELECTING ITEMS TO PRESENT TO NEW USERS

We can consider item attributes that can be used to devise selection strategies, including genre information, numbers of product ratings and rating scores by existing users, rating distribution, and users' demographic information. Recommender systems tend to perform reliably when users have rated over 20 items [5]. For this reason, we selected 20 items from each strategy to present to users during testing. The following strategies were used to select an initial item list:

(1) **Random**: We selected 20 products randomly with a uniform probability over the universe of items to compare the performance of other techniques.

(2) **Favorite**: The favorite method selects products with higher average preference scores. Average preference scores of products were calculated from ratings of the learning set as in following equation:

$$P_i = \frac{\sum_{j \in U_i} R_{ji}}{N_{U_i}} \tag{1}$$

P_i is the average preference score of product i and R_{ji} is the rating value user j gave to product i. U_i is the set of users who rated product i and N_{U_i} is the total number of users belonging to set U_i. We selected the top 20 products in descending order of average preference. Table 1 shows an example list of the top 20 favorite movies from the MovieLens dataset.

Table 1. An example list of the top 20 favorite movies

Movie Title	Average Preference Score
Schindler's List (1993)	4.473171
Casablanca (1942)	4.454023
Wrong Trousers, The (1993)	4.453488
Close Shave, A (1995)	4.418919
Shawshank Redemption, The (1994)	4.38172
Star Wars (1977)	4.371921
Usual Suspects, The (1995)	4.351064
Wallace & Gromit: The Best of Aardman Animation (1996)	4.333333
Citizen Kane (1941)	4.323308
Rear Window (1954)	4.321678
Silence of the Lambs, The (1991)	4.283582
12 Angry Men (1957)	4.252747
One Flew Over the Cuckoo's Nest (1975)	4.251397
Vertigo (1958)	4.241667
Raiders of the Lost Ark (1981)	4.235294
Empire Strikes Back, The (1980)	4.234375
Godfather, The (1972)	4.226481
Good Will Hunting (1997)	4.225352
Titanic (1997)	4.222222
Secrets & Lies (1996)	4.214953

(3) **Popularity**: Product rating numbers were calculated from the learning set as in the following equation:

$$S_i = \sum_{j \in U} r_{ji} \tag{2}$$

S_i is the rating number of product i and U is the set of all users in the learning set. If user j rated (or did not rate) product i, r_{ji} is 1 (or 0). We selected the top 20 products in order of descending popularity. Table 2 shows an example list of the 20 most popular movies from the MovieLens dataset.

Table 2. An example list of the top 20 popular movies

Movie Title	Rating Number
Star Wars (1977)	410
Fargo (1996)	373
Contact (1997)	352
Return of the Jedi (1983)	352
English Patient, The (1996)	342
Liar Liar (1997)	341
Scream (1996)	335
Toy Story (1995)	321
Independence Day (ID4) (1996)	298
Raiders of the Lost Ark (1981)	295
Godfather, The (1972)	293
Air Force One (1997)	290
Twelve Monkeys (1995)	286
Jerry Maguire (1996)	283
Pulp Fiction (1994)	279
Rock, The (1996)	273
Silence of the Lambs, The (1991)	269
Empire Strikes Back, The (1980)	260
Star Trek: First Contact (1996)	260
Mission: Impossible (1996)	258

(4) **Favorite*popularity**: The favorite*popularity method simultaneously uses the favorite and popularity methods. The average preference of product i, P_i and the rating number of product i, S_i, were normalized using the min–max algorithm and normalized values were multiplied as in the following equation:

$$PS_i = \frac{P_i - P_{Min}}{P_{Max} - P_{Min}} \times \frac{S_i - S_{Min}}{S_{Max} - S_{Min}} \tag{3}$$

PS_i is the final score of product i. P_{Max} is the maximum value of average preference scores and P_{Min} is the minimum value of average preference scores. S_{Max} is the maximum number of ratings and S_{Min} is the minimum number of ratings. We selected the top 20 products in order of descending PS_i. Table 3 shows an example list of the top 20 movies from the MovieLens dataset using the favorite*popularity method.

Table 3. An example list of the top 20 movies using the favorite*popularity method

Movie Title	Score
Star Wars (1977)	0.866213
Fargo (1996)	0.610306
Godfather, The (1972)	0.557757
Raiders of the Lost Ark (1981)	0.530596
Return of the Jedi (1983)	0.522957
Silence of the Lambs, The (1991)	0.506885
Empire Strikes Back, The (1980)	0.439436
Titanic (1997)	0.429278
Schindler's List (1993)	0.426967
Pulp Fiction (1994)	0.418995
Toy Story (1995)	0.396935
Princess Bride, The (1987)	0.390649
Shawshank Redemption, The (1994)	0.383898
Contact (1997)	0.383386
Usual Suspects, The (1995)	0.366097
Monty Python and the Holy Grail (1974)	0.344273
One Flew Over the Cuckoo's Nest (1975)	0.34345
Fugitive, The (1993)	0.339607
L.A. Confidential (1997)	0.332148
Casablanca (1942)	0.327958

(5) **Log popularity*entropy**: Rashid et al. (2002) proposed a log popularity*entropy method, by applying the entropy algorithm to select an initial item list. In contrast to popularity-based and pure entropy-based strategies, balanced strategies such as popularity*entropy and log popularity*entropy techniques exhibited improved performances in user effort and accuracy [13]. Rashid et al (2002) observed that popularity almost completely dominated popularity*entropy, and found that taking the logarithm of the ratings nearly linearized popularity, making it a better match for entropy [13].

Movie entropy can be calculated using Shannon's formula [13]:

$$H(p) = -\sum_{i=1}^{k} p_i * \log_2 p_i \qquad (4)$$

$H(p)$ is the entropy value of product p, and p_i is the ratio of i ratings. For example, the total number of ratings for product p is 20; five ratings have a score of 1, and 15 ratings have a score of 2. In this case, p_1 is 5/20 (0.25), and p_2 is 15/20 (0.75). The log popularity*entropy method multiplies log popularity (log S_i from (Equation 6)) with entropy $H(i)$ as in the following equation:

$$\log popularity * entropy_i = \log S_i * H(i) \tag{5}$$

Table 4 shows an example list of the top 20 movies from the MovieLens dataset using the log popularity*entropy method.

Table 4. An example movie list selected using the log popularity*entropy method

Movie Title	Score
Liar Liar (1997)	12.3101
Scream (1996)	12.2316
Independence Day (ID4) (1996)	12.2312
English Patient, The (1996)	12.0151
Saint, The (1997)	11.8261
Twister (1996)	11.4766
Evita (1996)	11.3719
Air Force One (1997)	11.3681
Contact (1997)	11.2762
Starship Troopers (1997)	11.2614
Leaving Las Vegas (1995)	10.9181
Mission: Impossible (1996)	10.9094
Birdcage, The (1996)	10.7579
Willy Wonka and the Chocolate Factory (1971)	10.7570
Twelve Monkeys (1995)	10.7556
Rock, The (1996)	10.7544
Conspiracy Theory (1997)	10.7293
In & Out (1997)	10.6768
Mars Attacks! (1996)	10.6635
Dante's Peak (1997)	10.6496

(6) **Genre**: Movies in the MovieLens dataset are classified into 19 genres including comedy, drama, unknown, etc. We excluded the unknown genre, which

contained only two movies. We selected movies from all other 18 genres using the favorite, popularity, favorite*popularity, and log popularity*entropy methods as follows:

- ***Genre—favorite***: This method involves selecting one movie with the best average preference in a genre. We chose 18 movies, one from each genre, and selected one additional movie from both the comedy and drama genres (which included more movies) to select a total of 20 movies.
- ***Genre—popularity***: This method sorts movies in each genre in order of descending popularity. We chose the top movie from each genre, which provided 18 movies. Then, we added the second-ranked movies in both the comedy and drama genres as in the Genre–favorite method.
- ***Genre—favorite*popularity***: This method involves selecting one movie with the highest favorite*popularity score in each genre. We chose 18 movies, one from each genre, then selected one more movie from both the comedy and drama genres for a total of 20 movies.
- ***Genre—log popularity*entropy***: In this method, movies in each genre are sorted in order of descending log popularity*entropy score. We chose the top movie from each genre, then selected the second-ranked movie from both the comedy and drama genres for a total of 20 movies.

(7) **Demographical clustering**: The MovieLens dataset contains demographic information about users, such as age, gender, and occupation. We classified users by age and gender. Demographic clustering (age) classifies users into five clusters: under 21, 21–30, 31–40, 41–50, and over 50. Demographic clustering (gender) simply classifies users into the male or female cluster. We selected 20 movies from each demographic cluster using the favorite, popularity, favorite*popularity, and log popularity*entropy methods as follows:

- ***Demographic—favorite***: In this method, movies are sorted in descending order of average preference score for each demographic cluster. We chose the top 20 movies for each cluster.
- ***Demographic—popularity***: In this method, movies are sorted in descending order of popularity for each demographic cluster. We chose the top 20 movies for each cluster.
- ***Demographic—favorite*popularity***: In this method, movies are sorted in descending order of favorite*popularity for each demographic cluster. We chose the top 20 movies for each cluster.

- ***Demographic—log popularity*entropy***: This method ranks movies in descending order of log popularity*entropy score for each demographic cluster. We chose the top 20 movies for each cluster, so each demographic cluster had a list of 20 movies.

After selecting movies for each demographic cluster, we provided a movie list based on a new user's age or gender. If a new user was 35 years old, we provided 20 movies selected from the 31–40 age cluster; if the new user was male, we provided 20 movies selected from the male cluster.

(8) **K-mean clustering**: The MovieLens dataset is basically composed of a user–product matrix, S. If the number of users is n and the number of products is m, matrix S is a $n \times m$ matrix. S_{ij} is the rating of user i on product j. We applied k-mean clustering to matrix S to classify movies using users' ratings. The resulting clusters are sets of movies that have similar user ratings; we chose the five clusters containing the most movies.

- ***Cluster—center***: From each cluster, we chose four movies that were located near the cluster's center. We selected movies nearest the cluster's center because we considered them most likely to represent cluster characteristics.
- ***Cluster—favorite***: In this method, movies are sorted in descending order of average preference score for each cluster. Because we had five clusters, we chose the top four movies from each cluster to create a list of 20 items.
- ***Cluster—popularity***: In this method, movies are sorted in descending order of popularity for each cluster. We chose the top four movies from each cluster.
- ***Cluster—favorite*popularity***: In this method, movies are sorted in descending order of favorite*popularity for each cluster. We chose the top four movies from each cluster.
- ***Cluster—log popularity*entropy***: This method ranks movies in descending order of log popularity*entropy score for each cluster. We chose the top four movies from each cluster.

4. PERFORMANCE COMPARISON EXPERIMENT

We used the MovieLens dataset to compare the performance of strategies suggested in Section 3. The dataset contains information about 943 users and

100,000 user ratings of 1,682 movies. In this experiment, we randomly classified 70% of users as a learning set (existing users) and assigned the other 30% as a test set (new users).

Movie lists selected by the techniques suggested in Section 3 were provided to the test set. Then, we calculated the test set's average number of ratings and average preference scores. If a tester rated a movie from the movie list provided, we added it to the number of ratings, and also used its rating score when calculating average preference scores. Table 5 shows the results of each method in terms of the test set's average number of ratings and average preference scores. The average number of ratings and average preference scores resulting from the 22 methods (17 methods using genre, demographic techniques, and k-mean clustering; five methods using the other five techniques) were statistically tested using ANOVA and their average values were ranked using Duncan's test. The null hypotheses were that the average numbers of ratings and average preferences scores resulting from the 22 methods would be the same. Both null hypotheses were rejected because F-values for average rating numbers and average preference scores were 1126.818 and 487.923, respectively.

Results of Duncan's test are summarized in the right-hand side of Table 5. The random method and k-mean clustering–center show low average numbers of ratings and user preference scores. When we compared average numbers of ratings, we found that the popularity method, demographic (age)–popularity, demographic (gender)–popularity, and demographic (gender)–favorite*popularity methods had the highest average number of ratings. The demographic (age)–favorite *popularity, demographic (age)–log popularity*entropy, demographic (gender)–favorite*popularity resulted in the fifth-highest average number of ratings. These results allow us to conclude that popularity methods provide adequately high numbers of user ratings. When we compared average user preference scores, we found that the favorite method and demographic (gender)–favorite had the highest average preference scores. The favorite*popularity, genre–favorite, demographic (age)–favorite, and k-mean clustering–favorite methods also resulted in higher average preferences. These results allow us to conclude that favorite methods provide adequately high average user preference scores.

While the favorite and popularity methods resulted in the highest average scores and greatest average number of ratings respectively, the favorite*popularity, demographic (age)–favorite*popularity, and demographic (gender)–favorite* popularity methods resulted in intermediate performances for both measures. The demographic (gender)–favorite*popularity method ranked eighth in average preference scores and ranked first in average number of ratings. The demographic (age)–favorite*popularity method also ranked eighth in average preference scores and ranked fifth in average number of ratings. These three methods fell short of the favorite method in average preference and the popularity method in average number of ratings, but outperformed the favorite method in average number of ratings and the popularity method in average preference.

Table 5. Summary of results

Methods		Average Number of Ratings of Users	Average Preference of Users	Compare Avg. Number of Ratings	Compare Avg. Preference
Random		1.42179	3.51005	22	20
Favorite		5.05559	4.31659	13	1
Popularity		9.20600	3.87231	1	10
*Favorite*Popularity*		7.95535	4.15812	10	5
*Log Popularity*Entropy*		7.45041	3.48823	11	20
Genre	*Favorite*	4.83074	4.25242	14	3
	Popularity	8.17314	3.80339	8	14
	*Favorite*Popularity*	7.01790	4.11209	12	7
	*Log Popularity*Entropy*	8.21177	3.74189	8	17
Demographic Clustering - Age	*Favorite*	1.85294	4.21041	20	4
	Popularity	9.11407	3.87081	1	10
	*Favorite*Popularity*	8.96808	3.93139	5	8
	*Log Popularity*Entropy*	8.83133	3.78793	5	14
Demographic Clustering - Gender	*Favorite*	4.00400	4.30544	18	1
	Popularity	9.16902	3.86249	1	12
	*Favorite*Popularity*	9.09776	3.90502	1	8
	*Log Popularity*Entropy*	9.03309	3.78786	5	14
K-mean Clustering	*Center*	1.76336	3.38419	20	22
	Favorite	3.13828	4.17319	19	5
	Popularity	4.78916	3.67816	14	18
	*Favorite*Popularity*	4.62249	3.84956	14	12
	*Log Popularity*Entropy*	4.69976	3.58228	14	19
F – Value				1126.818 (Sig. = 0.000)	487.923 (Sig. = 0.000)

* Compare Avg. Number of Ratings and Avg. Preference columns are ranks of Duncan's test, Alpha = 0.05 (higher rank means higher average)

Table 6. Summary of results: Demographic clustering–Age

Methods		Average Rating Number	Average Preference	Compare Avg. Number of Ratings	Compare Avg. Preference
Favorite	Under 21	0.608799929	3.469049183	2	2
	21 - 30	5.259388028	4.3099981	1	1
	31 – 40	0.528089999	3.736378859	2	2
	41 – 50	0.745724736	3.702380276	2	2
	Over 50	0.804966195	3.621216162	2	2
	F-Value	-	-	303.887 (Sig. = 0.000)	7.539 (Sig. = 0.000)
Popularity	Under 21	9.124672745	4.014674106	2	1
	21 – 30	10.17279741	3.851884914	1	2
	31 – 40	9.073349934	3.876800875	2	2
	41 – 50	8.203158757	3.746788666	4	5
	Over 50	7.148335348	3.904306817	5	2
	F-Value	-	-	108.175 (Sig. = 0.000)	55.426 (Sig. = 0.000)
Favorite Popularity*	Under 21	9.076694954	4.099729834	2	1
	21 – 30	9.88137728	3.928767354	1	1
	31 – 40	8.837029859	3.92548465	2	1
	41 – 50	8.384708172	3.692406877	4	5
	Over 50	7.039672224	3.930319651	5	1
	F-Value	-	-	71.711 (Sig. = 0.000)	6.219 (Sig. = 0.000)
*Log Popularity*Entropy*	Under 21	9.408793	3.931288	2	1
	21 – 30	9.909061	3.782627	1	1
	31 – 40	8.515334	3.801116	3	1
	41 – 50	7.87935	3.549627	4	5
	Over 50	6.997132	3.78925	5	1
	F-Value	-	-	127.392 (Sig. = 0.000)	5.937 (Sig. = 0.000)

* Compare Avg. Number of Ratings and Avg. Preference columns are ranks of Duncan's test, Alpha = 0.05 (higher rank means higher average)

Table 6 illustrates how the demographic clustering methods performed among age clusters. Each of the demographic clustering (age)–favorite, popularity favorite*popularity, log popularity*entropy methods exhibited different average values among age clusters, as shown in Table 6. The null hypothesis for each method was that results of average numbers of ratings and average preferences would be the same from all five clusters. The null hypotheses for all four methods were rejected.

In all methods, the 21–30 age cluster produced the highest average number of ratings. It is likely that users in the 21–30 age range are the main consumers of movie products. Older age ranges resulted in lower average numbers of ratings. The favorite method produced the highest average numbers of ratings and average preference scores in the 21–30 age cluster, meaning that users in the 21–30 age range were more likely to watch movies with higher average preference scores and to rate them with higher preference scores. The popularity method results showed that users under 21 rated popular movies with higher preference scores. Table 5 shows that the demographic clustering (age)–popularity, favorite*popularity, and log popularity*entropy methods produced higher average rating numbers than the popularity method.

Table 7. Summary of results: Demographic clustering–Gender

Methods		Average Number of People	Average Rating Number	Average Preference
Favorite	Male	199.7	5.40556	4.32309
	Female	83.3	0.63810	3.68820
	t-value (Sig.)		32.434 (0.000)	5.806 (0.000)
Popularity	Male	201.63333	9.58772	3.87512
	Female	81.36666	8.13932	3.82613
	t-value (Sig.)		13.724 (0.000)	3.832 (0.000)
Favorite Popularity*	Male	199.8	9.52580	3.91933
	Female	83.2	8.07002	3.86468
	t-value (Sig.)		14.578 (0.000)	4.1000 (0.000)
Log Popularity Entropy*	Male	199.76666	9.42686	3.79491
	Female	83.23333	8.08447	3.76920
	t-value (Sig.)		13.927 (0.000)	1.982 (0.052)

* Alpha = 0.05 (2-tailed)

Table 7 illustrates the performances of demographic clustering methods between gender clusters; we used a t-test to compare average rating numbers and average preferences of the demographic clustering (gender)–favorite, popularity, favorite*popularity, and log popularity*entropy methods between gender clusters.

Except for the average preferences resulting from demographic clustering (gender)–log popularity*entropy, average rating numbers and average preferences were significantly higher for male clusters than for female clusters. Also, the average rating numbers resulting from the popularity, favorite*popularity, and log popularity*entropy methods were higher than the popularity methods shown in Table 5 for the male clusters; the favorite methods also resulted in higher average preferences than the favorite methods shown in Table 5 for the male clusters.

5. DISCUSSION AND CONCLUSION

In this paper, we examined various techniques that provide initial item lists to allow systems to learn about new users, and analyzed their performances. The favorite method provided the highest user rating scores, but was unable to provide high quantities of ratings. This may have been because independent films are not widely popular but acquire higher rating scores from fans. The popularity method provided the highest average numbers of ratings, but was not able to provide higher average preference scores. Popular movies are easier for systems to recommend because similar users are more likely to have seen them [13]. According to Rashid et al. (2002), however, ratings of popular movies supply little information because most individuals like these movies. While favorite and popularity methods provided the highest average scores and highest average number of ratings respectively, both the favorite*popularity, demographic (age)–favorite* popularity, demographic (gender)–favorite*popularity methods performed within acceptable ranges. The results of demographic comparisons shown in Table 6 and 7 are related to consumption patterns of movie products by age and gender. Younger individuals tend to see more movies, see more popular movies, and give popular movies higher scores than older individuals. Male users tend to see more movies and give higher scores to popular movies than female users. Tables 6 and 7 show that use of demographic information when selecting initial item lists can be helpful to improve the numbers of ratings and average preference scores.

Performance of selection strategies and choice of appropriate strategies for e-

Commerce sites or recommender systems vary according to data sets and users. In this paper, we only used the MovieLens dataset; results of the experiment performed in Section 4 should not be generalized to other data sets. Operators of e-Commerce sites must select appropriate strategies, but before applying one of the strategies discussed above, experiments similar to the ones executed in Section 4 should be conducted in order to select the appropriate one(s).

In this study, we provided initial item lists that were massed together without any effort from users. However, user feedback can be helpful when selecting movies to present to new customers [11]. Finding solutions to the new user problem based on user feedback or product attributes would be an interesting topic for further research.

REFERENCES

[1] Ansari, A., S. Essegaier, and R. Kohli, "Internet recommendation systems," *Journal of Marketing Research* 37, 3 (2004), 363-375.

[2] Breese, J. S., D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998, 43-52.

[3] Herlocker, J. L., J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems* 22, 1 (2004), 5-53.

[4] Huang, Z., H. Chen, and D. Zeng, "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering," *ACM Transactions on Information Systems* 22, 1 (2004), 116-142.

[5] Kim, J. W., H. J. Lee, and S. J. Park, "Parameter Selection of Collaborative Filtering for e-Commerce Personalized Recommendation," in *The 7th International Conference on Electronic Commerce Research (ICECR-7)*, Dallas, U.S.A, 2004.

[6] Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Communication of the ACM* 40, 3 (1997), 77-87.

[7] Linden, G., B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing* 7, 3 (2003), 76-80.

[8] Mild, A. and M. Natter, "Collaborative filtering or regression models for Internet recommendation systems?" *Journal of Targeting, Measurement*

and Analysis of Marketing 10, 4 (2002), 304-313.
[9] Mobasher, B., R. Cooley, and J. Srivasta, "Automatic personalization based on web usage mining," *Communication of the ACM* 43, 8 (2000), 142-151.
[10] Mulvenna, M. D., S. S. Anand, and A. G. Buchner, "Personalization on the net using web mining," *Communication of the ACM* 43, 8 (2000), 122-125.
[11] Nquyen, H. and P. Haddawy. "The Decision-Theoretic Video Advisor," in *AAAI Workshop on Recommender Systems*, 1998, 76-80.
[12] Pennock, D. M., E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach," in *the Sixteenth Conference on Unceratainty in Aritificial Intelligence(UAI-2000)*, San Francisco, 2000, 473-480.
[13] Rashid, A. M., I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. "Getting to Know You: Learning New User Preferences in Recommender Systems," in *the ACM IUI'02*, San Francisco, 2002, 127-134.
[14] Sarwar, B., G. Karypis, J. Konstan, and J. Riedl, "Analysis of Recommendation Algorithms for e-Commerce," in *EC'00*, Minneapolis, 2000, 158-167.
[15] Sarwar, B., G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," in *WWW10*, Hong Kong, 2001, 285-295.
[16] Sarwar, B. M., G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study," in *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*, 2000.
[17] Schein, A. I., A. Popescul, L. H. Ungar, and D. M. Pennock. "Generative Model for Cold-Start Recommendations," in *SIGIR Workshop on Recommender Systems*, New Orleans, LA, 2001.
[18] Schein, A. I., A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and Metrics for Cold Start Recommendations," in *the ACM SIGIR'02*, Tampere, Finland, 2002, 253-260.
[19] Ungar, L. H. and D. P. Foster, "Clustering Methods for Collaborative Filtering," in *Workshop on Recommendation Systems at the Fifteenth National Conference on Artificial Intelligence*, 1998.