2012 Second International Conference on Cloud and Green Computing

Movie Recommendation System Based on Movie Swarm

Sajal Halder
Dept. of Computer Engineering
Kyung Hee University
Yongin-si, Gyeonggi-do, Korea
sajal@khu.ac.kr

A. M. Jehad Sarkar
Dept. of Digital Information Engineering
Hankuk University of Foreign Studies,
Yongin-si, Gyeonggi-do, Korea
jehad@hufs.ac.kr

Young-Koo Lee
Dept. of Computer Engineering
Kyung Hee University
Yongin-si, Gyeonggi-do, Korea
yklee@khu.ac.kr

***Abstract*—A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Such a system can suggest a set of movies to users based on their interest, or the popularities of the movies. Although, a set of movie recommendation systems have been proposed, most of these either cannot recommend a movie to the existing users efficiently or to a new user by any means. In this paper we propose a movie recommendation system that has the ability to recommend movies to a new user as well as the others. It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. It generates movie swarms not only convenient for movie producer to plan a new movie but also useful for movie recommendation. Experimental studies on the real data reveal the efficiency and effectiveness of the proposed system.**

***Keywords*-Interesting movie, popular movie, movie swarm, recommendation system**

I. Introduction

Given the huge amount of movies are available all over the world, it is challenging for a user to find the appropriate movies suitable for his/her tastes. Different users like different movies or actors. It is important to find a method of filtering irrelevant movies and/or find a set of relevant movies.

Movie recommendation system is a process of exactly doing above tasks. Such a system has lot of implications and is inspired by the success of recommendation systems in different domains such as books [8], TV program [9], [15], jokes [4], news articles [11]. It is one of the most important research in the digital television domain [14].

The most well known recommendation systems are mainly based on Collaborative Filtering (CF) [3] and Content-based Filtering [2]. CF first tries to find out the groups of similar users automatically from a set of active users. The similarities between users are computed using correlation measure. It then recommends items to a user based on the opinions of the users groups. Although CF is successful in many domains, however, it has shortcomings such as, sparsity and scalability [12]. CF uses user ratings to find similar users. However, it is very difficult to find such since very few movies have ratings.

In this paper, we propose two methods important for movie recommendation: movie swarm mining that mines a set of movies suitable for producer for planning new movie and for new item recommendation, popular and interesting movie mining which can be used to solve new users problem. The effectiveness of our proposed methods demonstrated using MovieLens Data Sets.

The rest of this paper is organized as follows. In Section II we discuss the related work. In Section III, we define the problems. In Section IV, we discuss the proposed methods. In Section V, we show the experimental results and discuss various issued related to the system. In Section VI, we conclude the paper with a direction of future work.

II. Related Work

To the best of our knowledge, there are a number of methods have been proposed in recommendation system. The well-known recommendation system is Collaborative Filtering [3], which uses users assessment on observed items to measure users similarity. Such assessment is determined either explicitly or implicitly. In an explicit determination, users are asked to provide their ratings in a one-to-five scale, which are then used for measuring the similarity. In an implicit determination, users rating are determined based on the browsing behaviors. However, if the item set is large and users rate a small fraction these, it is often difficult to find similarities between users. This leads to low accuracy predictions or even to failure to make predictions.

Balabanovic et al. [2] had proposed a content-based recommendation system which can be applied in different domains, such as, books, movies, videos, or music. It uses different features, such as, author, genre, and most frequently used words. TF-IDF and Information Gain (IG) are used commonly to extract these [1], [10].

George et al. [6] had proposed a hybrid approach for movie recommendation system. This is a Web-based recommendation system, collects user ratings of movies in one-to-five scales by a graphical user interface. This process implemented in two variations; substitute and switching. The aim of substitute is to utilize collaborative filtering. The system uses a collaborative filtering technique as the main

978-0-7695-4864-7/12 $26.00 © 2012 IEEE
DOI 10.1109/CGC.2012.121

recommendation method. However, it uses a content-based technique for prediction if the number of available ratings falls below a given threshold.

In collaborative filtering [3], when a new user or a new item is introduced, the system had no predictions that can make recommendations. Content-based [2] methods can handle new items, however, fails to handle new users. Although a hybrid system [6] tried to incorporate both collaborative and content-based filtering, however, it also has the difficulties in dealing new users.

In this paper, we propose a recommendation system that has the ability to handle both new users and items. Firstly, movie swarms create swarm based on movie genres those are features based that cover content-based recommendation system. This process solves new item and new user recommendation issue. However, this process might be overloaded when a large number of same genre of movies are released. To solve this issue, we propose a method that uses popular and interesting movies.

III. Problem Definitions

Let $U_{db} = \{u_1, u_2, ..., u_n\}$ be the set of all users and $T_{db} = \{t_1, t_2, ..., t_m\}$ be the set of all timestamps in the database, $G_{db} = \{g_1, g_2, ..., g_p\}$ be the set of all movie genres and $I_{db} = \{i_1, i_2, ..., i_q\}$ be the set of all movie items. A subset of U_{db} is called an user set U. A subset of T_{db} is called time set T, a subset of G_{db} is called movie genres set G and a subset of I_{db} is called movie Items set I. The number of user set, time set, movie genres set and movie item set indicate $|U|$, $|T|$, $|G|$ and $|I|$ respectively.

Before describing more details, we have defined a set of terms.

Definition 1. *Short Time Movie Swarm (STMS): A swarm set (U, t, g) is said to be a short time movie swarm if all users in $U \subseteq U_{db}$ enjoy movie genres $g \in G_{db}$ at timestamps $t \in T_{db}$.*

Definition 2. *Long Time Movie Swarm (LTMS): A swarm set (U, T, G) is said to be a long time movie swarm if all users in $U \subseteq U_{db}$ enjoy movie genres $G \in G_{db}$ at timestamps $T \in T_{db}$, where $|T| \geq min_t$(minimum threshold).*

To avoid mining redundant long time movie swarms, we further give the definition of long time movie close swarm as follows:

Definition 3. *Long Time Movie Closed Swarm (LTMCS): A long time movie swarm (U, T, G) is called long time movie closed swarm if there are no T' , U' where $T \subset T'$ and $U \subset U'$ for particular genres $G_p \subseteq G_{db}$ and (U', T', G_p) is long time movie swarm.*

Definition 4. *Interesting Movie (IM) : A movie item $i \in I_{db}$ is said to be interesting movie if a short time swarm (U, t, i) enjoy the item i at time stamps t where $|U| \geq min_u^{im}$ and the average rating of this movie $avg_r(i) > r_{im}$.*

Definition 5. *Popular Movie (PM) : A movie item $i \in I_{db}$ is said to be popular movie if a short time swarm (U, t, i) enjoy the item i at time stamps t where $|U| \geq min_u^{pm}$ and the average rating of this movie $avg_r(i) > r_{pm}$.*

Definition 6. *Interesting Popular Movie (IPM) : A movie item $i \in I_{db}$ is said to be interesting popular movie if a short time swarm (U, t, i) enjoy the item i at time stamps t where $|U| \geq min_u^{pm}$ and the average rating of this movie $avg_r(i) > r_{im}$.*

In this paper it has been found out short time movie swarm, long time movie swarm, popular, interesting and interesting and popular movie genres set based on users who enjoy the movie.

IV. General System Architecture

Figure 1 depicts the system architecture of our proposed method that consists of two techniques that are movie swarm mining and interesting and popular movie mining. The preprocessing step is responsible for data collection and cleaning because data could be inaccurate, inconsistent and noisy. Mining techniques are then performed on the preprocessed data sets and find movie swarm sets, interesting and popular movie sets and similar users groups, these are very useful for movie recommendation system.

The following subsections, it has been described each techniques in details.

A. Movie Swarm Mining

Specifically, movie swarms (U, T, G) have to maintain two minimum thresholds min_u and min_t for (U, T) at movie genres set $G \neq \phi$. Where $U = \{u_{i1}, u_{i2}, ..., u_{ip}\} \subseteq U_{db}$ and $T \subseteq T_{db}$, it needs to satisfy three requirements: $|U| >= min_u$, $|T| >= min_t$ and $g_{t_i}(u_{i1}) \bigcap g_{t_i}(u_{i2}) \bigcap ... \bigcap g_{t_i}(u_{ip}) \neq \phi$ for any $t_i \in G$. There is at least one movie genre containing all the users in U at each timestamps in T. If $|T| = 1$ it is called short time movie swarm $(STMS)$ and otherwise it is long time movie swarm $(LTMS)$. Movie swarm mining find out interesting movie genres and groups of similar users $|S(U)| \geq min_u$ at timestamps $|S(T)| \geq min_t$ [5]. To find out movie swarm we have used two basic steps. Those steps are described below.

1) Pruning Step: In our proposed method, we have used two kinds of pruning for reduce time and space complexity. It also increase efficiency of our algorithm. Both of pruning steps are like this.

Definition 7. *Column Pruning (CP) : A movie genre set $g_i \in G_{db}$ that store set of users at timestamps T_{db} which number is less then min_u then we discard the movie genre g_i, it is called column pruning.*

Definition 8. *Row Pruning (RP) : At timestamps $T \subset T_{db}$ that store set of users information at movie genres G_{db} which*

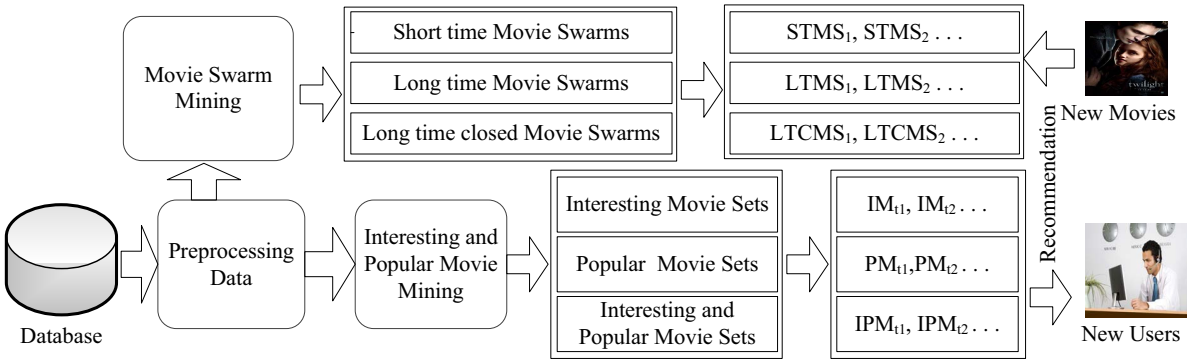


Figure 1. System Architecture

number is less then min_u then we discard the timestamps T, it is called row pruning.

We can define column and row pruning line as equation 1 and 2 respectively.

$$CP_{g_i} = g_i \text{ if } Max_{t=1}^{|T_{db}|} \left\{|S(U_{g_i}^t)|\right\} < min_u \qquad (1)$$

$$RP_T = T \text{ if } Max_{l=1}^{|G_{db}|} \left\{|S(U_l^T)|\right\} < min_u \qquad (2)$$

2) Frequent Patten Mining: Li et al. [7] method spatiotemporal swarm algorithm used Apriori algorithm and it was trajectory clustering technique. However, we have used vertical format because it needs scan data set only one time because movie data set are consists massive data. It is faster than Apriori algorithm. The support count of a timestamps is simply the number of the users who enjoy movie. The frequent k-timestamps can be used to construct the candidate (k+1)-timestamps based on the Apriori property.This process repeats, with k incremented by 1 each time, until no frequent timestamps or no candidate timestamps can be found.

Algorithm 1 shows the movie swarm mining process. From line 9 - 12 find all short time movie swarm (STMS) for each timestamps $t \in T_{db}$. At line 9 calculate users set U_g^t at time t who enjoy g movie genres, if $U_g^t \geq min_u$ then store as STMS. From line 13 - 23 we get all LTMS and LTCMS. At line 13 calculate users set at $k - timestamps$ and if users number and time stamps is grater then or equal parameter value then add as a LTMS. The line 18 check all subset at current swarm and discard from LTCMS. Finally add new swarm as a LTCMS.

Movie swarms carry very effective information for movie producer. From the movie swarm, producer gets an idea about current popular movies and concern about users interest. Which kinds of movie genres are exist in movie swarm that are interesting and popular. Users who exist within a group have similarity among them. When new movie release, its genres related interested users who create swarm can be recommended.

Algorithm 1: Movie Swarm Mining (U_{db}, T_{db}, G_{db}, min_u,min_t)

Data: U_{db}: users; G_{db}: movie genres; T_{db}: movie enjoy times; min_u, min_t: minimum threshold parameters.

Result: A set of movie swarm STMS, LTMS and LTCMS

1 $Preprocessing()$;
2 $CreateMatrix()$;
3 $STSS = LTMS = LTCMS = \Phi$;
4 S_k: Swarm of size at time set k;
5 **for** $k = 1$; $S_k \neq \phi$; $k++$ **do**
6 Rowpruning();
7 Columnpruning();
8 **for** $\forall g \in G_{db}$ **do**
9 calculate Users set U_g^t;
10 **if** $U_g^t \geq min_u$ **then**
11 $STMS = STMS \bigcup (U_l^t, t, g)$
12 **end**
13 calculate Users set U_g^{kt};
14 **if** $U_g^{kt} \geq min_u$ & $|kt| \geq min_t$ **then**
15 S_k = (U_g^{kt}, kt, g)
16 $LTMS = LTMS \bigcup S_k$
17 **end**
18 **if** $(\forall MS \in LTCMS) \subset S_k$ **then**
19 $LTCMS$ = $LTCMS$ - MS;
20 **end**
21 $LTCMS = LTCMS \bigcup S_k$;
22 **end**
23 **end**
24 **return** $STSS$, $LTMS$, $LTCMS$/* Return all sets of swarms */;

B. Interesting and Popular Movie Mining

Movie closed swarms have been proposed for finding more interesting movie genres and measurement of users similarity at the same times where users enjoy same genres of movie are frequent and periodic. However, using above process the interesting genres of movies are found but interesting movies are not discovered. In this circumstance, we propose another technique that finds interesting, popular movies called interesting and popular movie mining. We can represent interesting, popular and interesting and popular movie using equation 3, 4 and 5 respectively.

$$IM^t_{I_i} = I_i \text{ if } |S(u^t_{I_i})| \geq min^{im}_u \ \& \ avg_r(I_i) \geq r_{im} \quad (3)$$

$$PM^t_{I_i} = I_i \text{ if } |S(u^t_{I_i})| \geq min^{pm}_u \ \& \ avg_r(I_i) \geq r_{pm} \quad (4)$$

$$IPM^t_{I_i} = I_i \text{ if } IM(I_i) \ \& \ PM(I_i) \quad (5)$$

where $S(u^t_{I_i})$ is the set of users who enjoy item I_i at time t. Here $avg_r(I_i)$ is the average ranking of items I_i and r_{im}, r_{pm} are interesting and popular rate parameter value. Algorithm 2 shows our proposed methods that have been used to find out popular movie (PM), interesting movie (IM) and interesting popular movie (IPM). In this algorithm, we consider $min^{pm}_u > min^{im}_u$ and $r_{im} > r_{pm}$. Line number 6, 9 and 12 find out IM, PM and IPM respectively.

Algorithm 2: IPM(U_{db}, T_{db}, I_{db}, min^{im}_u, min^{pm}_u, r_{im}, r_{pm})

Data: U_{db}: movie users; I_{db}: movie items; T_{db}: movie enjoying timestamps; min^{im}_u, min^{pm}_u, r_{im}, r_{pm}: minimum threshold

Result: A set of IM, PM and IPM movie

1 $Preprocessing()$;
2 $IM = PM = IMP = \Phi$;
3 **for** $\forall I_i \in I_{db}$ **do**
4 calculate $S(U^t_{I_i})$; calculate $avg_r(I_i)$;
5 **if** $S(U^t_{I_i}) \geq min^{im}_u$ & $avg_r(I_i) \geq r_{im}$ **then**
6 $IM = IM \bigcup I_i$;
7 **end**
8 **if** $S(U^t_{I_i}) \geq min^{pm}_u$ & $avg_r(I_i) \geq r_{pm}$ **then**
9 $PM = PM \bigcup I_i$;
10 **end**
11 **if** $S(U^t_{I_i}) \geq min^{pm}_u$ & $avg_r(I_i) \geq r_{im}$ **then**
12 $IPM = IPM \bigcup I_i$;
13 **end**
14 **end**
15 **return** IM, PM, IPM/* Return movie sets */;

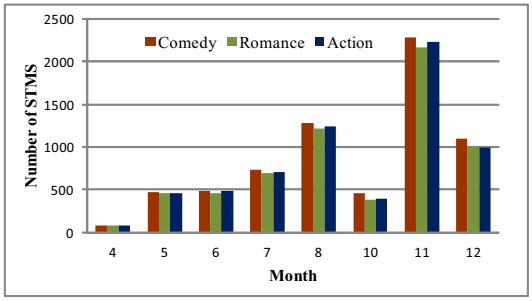
This movie mining is needed for recommend new users. When a user registered into the system, he/she has no previous knowledge so recommendation is very challenging in this issue. In this system has been proposed previous and current interesting or popular movie for new users. Most of the previous recommendation systems did not consider this case means new users problem.

V. Experiment

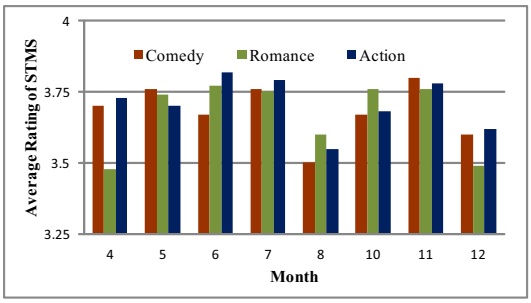
All the algorithms were implemented in Java, and all the experiments are carried out on 3.30 GHz Intel Core i5 system with 4GB memory.

A. Datasets

In our performance study, we have experimented on MovieLens datasets [13] that were collected by the GroupLens Research Project at the University of Minnesota. This data set consists of 1,000,209 anonymous ratings of 18 genres of 3,952 movies made by 6,040 MovieLens users who joined MovieLens in 2000.



(a) Number of STMS at different time

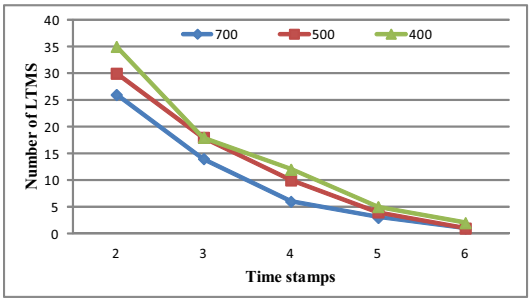


(b) Average rating of STMS at different time

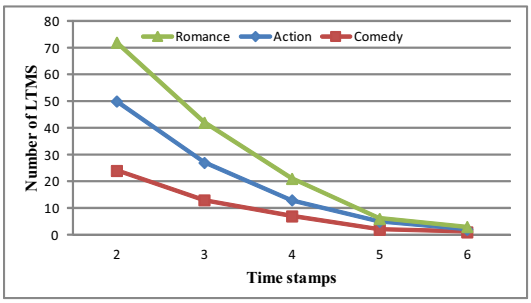
Figure 2. Short time movie swarm at different time stamps

B. Results for Movie Swarm Mining

In movie swarm, we need frequent genres of movie that users enjoy frequently. The movie producer gets good feedback from this mining technique and will encourage to produce movies, whose are more popular, and having a good chance to popular. Users within a swarm consider similar group of users and one user can recommend movies to other users who did not enjoy the particular movie.

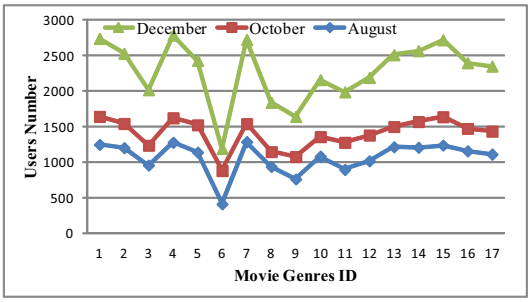


(a) Number of swarm at different time stamps and different threshold values for Action genre

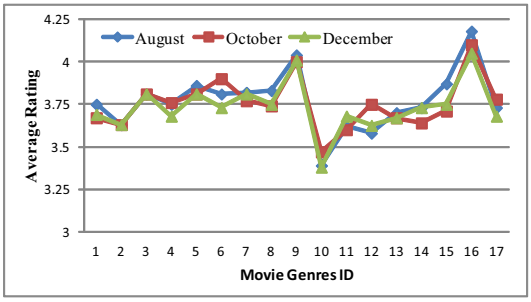


(b) Number of swarm at different time stamps and different genres for threshold value 700.

Figure 3. Long time movie swarm at different time stamps and different user threshold values



(a) Number of enjoying movies at different genres depends on time



(b) Rating of enjoying movies at different genres depends on time

Figure 4. Popular and interesting movie genres

Figure 2 and 3 show the result of our proposed method, which are very encouraging. Number of STMS and average rating of STMS are shows in figure 2. Figure 3(a) shows the number of LTMS swarm at different threshold values for action movie. The numbers of LTMS swarms are increased if threshold values decreased. At lowest threshold value, we get the large number of swarms and with the increases of timestamps, the number of swarm decreases. Figure 3(b) shows different genres movie swarm number at different time stamps. When timestamps increase, the number of swarm decreases. From figure, we find that action movie is more popular because they create maximum number of swarms. In the timestamps, if $k - timestamps$ value is smaller then it finds large number of swarm but if its value is large then it finds small number of swarms. Therefore, the parameter value is more effective to produce movie swarm.

C. Results for Interesting and Popular Movie Mining

Using movie swarm mining technique we find the popular genres of movies at timestamps t but we are not clear about interesting or popular movie at each timestamps t.

Figure 4(a) and 4(b) shows the popular movie genres and interesting movie based on user number and user rating. Visually, we obtain the large number of popular movie at $min_u^{pm} = 600$ and obtain less number of popular movie genres at $min_u^{pm} = 1200$. We also observed that the large number of interesting movie at $r_{im} = 3.5$ and obtain less number of interesting movie genres at $r_{im} = 3.8$. Therefore, we said that the number of popular and interesting movie decreases with the increases of users threshold value. We also show that movie enjoying number depends on time such as figure 4(a) at December all genres of movie enjoying number is very high compare than other time. So release any new movie at December have a high change to popular.

Finally, we said that our proposed method is efficient to recommend new users and new items than collaborating filtering and content-based recommendation system.

VI. Conclusion

In this paper, we first address the importance of movie recommendation system and movie data mining. We have proposed a new concept called movie swarm mining, uses two pruning rule and vertical data format frequent item mining. It solves the new item recommendation problem and provides an idea about the current trends of the popular movies and user interests. This is very helpful for movie producer to plan new movies. We also proposed algorithm for mining interesting and popular movie genres to recommend movies to a new user. With the help of two experiments in real datasets, MovieLens, we have shown the effectiveness of our proposed method.

The proposed method however has a shortcoming of finding the group of a user depending on the movie genre, if he/she enjoys diverse set of movies. In the future version of the system, we will be working to overcome this deficiency.

ACKNOWLEDGMENT

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2012-(H0301-12-2001)). We also thank anonymous reviewers for their valuable comments.

REFERENCES

[1] M. Balabanović. An adaptive web page recommendation service. In *Proceedings of the first international conference on Autonomous agents*, pages 378–385. ACM, 1997.

[2] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[3] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[5] Sajal. Halder, Md. Samiullah, A. M. Jehad Sarkar, and Young-Koo Lee. Movieswarm: Information mining technique for movie recommendation system. *In Submission*.

[6] G. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. *Multimedia tools and applications*, 36(1):55–70, 2008.

[7] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *Proceedings of the VLDB Endowment*, 3(1-2):723–734, 2010.

[8] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[9] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 263–266. ACM, 2003.

[10] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.

[11] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[12] Badrul M. Sarwar, George Karypis, Joseph Konstan, and John Reidl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT)*, 2002.

[13] MovieLens Data Sets. http://www.grouplens.org/node/73/.

[14] B. Smyth and P. Cotter. A personalized television listings service. *Communications of the ACM*, 43(8):107–111, 2000.

[15] Z. Yu and X. Zhou. Tv3p: an adaptive assistant for personalized tv. *Consumer Electronics, IEEE Transactions on*, 50(1):393–399, 2004.