Journal of Universal Computer Science, vol. 17, no. 14 (2011), 2009-2028
submitted: 7/12/10, accepted: 28/9/11, appeared: 1/10/11 © *J.UCS*

A Visited Item Frequency Based Recommender System: Experimental Evaluation and Scenario Description

Roberto Konow
(Informatics and Telecommunications Engineering School
Universidad Diego Portales, Santiago de Chile, Chile
roberto.konow@mail.udp.cl)

Wayman Tan, Luis Loyola
(SkillUpJapan Corporation
Tokyo, Japan
w.tan@skillupjapan.co.jp, loyola@skillupjapan.co.jp)

Javier Pereira
(Informatics and Telecommunications Engineering School
Universidad Diego Portales, Santiago de Chile, Chile
javier.pereira@udp.cl)

Nelson Baloian
(Department of Computer Science
Universidad de Chile, Santiago de Chile, Chile
nbaloian@dcc.uchile.cl)

Abstract: There has been a continuous development of new clustering and prediction techniques that help customers select products that meet their preferences and/or needs from an overwhelming amount of available choices. Because of the possible huge amount of available data, existing Recommender Systems showing good results might be difficult to implement and may require a lot of computational resources to perform in this scenario. In this paper, we present a more simple recommender system than the traditional ones, easy to implement, and requiring a reasonable amount of resources to perform. This system clusters users according to the frequency an item has been visited by users belonging to the same cluster, performing a collaborative filtering scheme. Experiments were conducted to evaluate the accuracy of this method using the Movielens dataset. Results obtained, as measured by the F-measure value, are comparable to other approaches found in the literature which are far more complex to implement. Following this, we explain the application of this system to an e-content site scenario for advertising. In this context, a filtering tool is shown which has been developed to filter and contextualize recommended items.

Key Words: Recommender System, Collaborative Filtering, Clustering, TF-IDF, F-Measure, Advertising, e-content

Category: H.3.1, H.1.m, H.4.m, J.0.m

1 Introduction

Nowadays, when e-commerce has penetrated almost all branches, it becomes important to present each one of the millions of potential customers with a personalized offer. In this scenario, Recommender Systems play an important role. For more than a decade, there has been continuous development of new clustering and prediction techniques that help customers select products that meet their preferences and/or needs from an overwhelming amount of available choices [Sarwar *et al.* 2002]. Examples of those applications include recommendation systems for buying books, CDs and other products at Amazon, recommendation of movies to be seen at Netflix and recommendations for listening to certain types of music at Last.fm.

During the last five years we have seen a gradual market shift on recommender systems from electronic commerce to content streaming and general media delivery, including music and movies. There are many media delivery companies that are making efforts to improve their Recommendation Systems. One particular example is that of Netflix, the online DVD rental pioneer in the US, which offers a 1 million dollar prize to anyone contributing to improve its movie recommender system Cinematch. Over the last 2 years we have seen a soar in video-on-demand and IP-based television (IPTV) services. According to the US IPTV Forecast and Outlook report from Strategy Analytics, it is expected that IPTV revenues will grow rapidly, reaching 14 billion US dollars in 2012 up from 694 million US dollars in 2007 [Piper 2010]. Only in 2008, the global IPTV market grew 63% while the US market saw a bad year due to the global economic downturn, according to the Broadband Forum [Broadband 2010], a worldwide consortium of around 200 companies from the telecommunication and information technology sector. In [Cotriss 2009] and [Van den Dam 2007] we also see pertinent background information supporting the thesis that advertising in IPTV will become an important business in the near future.

One of the main revenue sources in the IPTV industry is expected to be advertisement and more specifically, customer targeted advertisement. However, implementing effective advertising in a real scenario faces some technical challenges due to the huge amount of advertised items available for showing. This means, that a small portion of them might be shown while a user is viewing a movie or just visiting the site. In this scenario recommender systems have an important role to play selecting the right subset of advertisement items to watch a user will most likely react. However, traditional recommender systems will most probably face serious complications in a real scenario since the amount of data concerning users as well as advertisement items is huge and sometimes not very accurate. On the other hand, there are some basic data that might help to drastically filter the number of suitable advertisement items for a certain user with a very simple method, like defining a the target user for a certain advertising item to be delimited by age, gender or geographical area.

In this paper we present a solution implemented for a real case (an IPTV company in Japan). Here both were applied, an automatic hybrid recommendation algorithm based on clustering the user pool according to their preferences and a collaborative filtering process according to their reactions to previous recommendations. This is followed by a semi-automatic filtering process based on a profile definition of target user previously defined by the advertiser. The process is shown in Figure 1 and consists of 6 steps.

The rest of the paper is organized as follows: The next section describes the pertinent state of the art for recommender systems. The third section presents a recommender system tailored for a general scenario with large amounts of users and potential items to recommend. In this chapter the user clustering and the implicit collaborative filtering process are described in detail. An evaluation of this automatic recommender process is presented using the Movielens database. The aim of this evaluation was testing the suitability of the process for predicting items (in this case movies) a user belonging to a certain cluster will chose. After this, the fourth section shows the tools developed in order to obtain strategic information about potential users in order to help advertisers define the filters they would like to apply to the advertising items they provide.

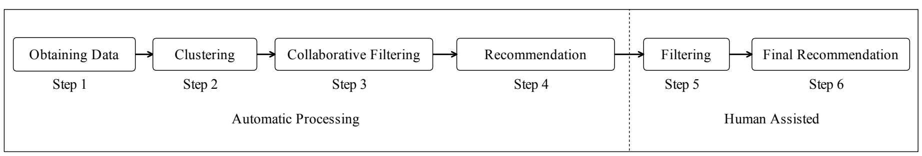

Figure 1: *Process diagram of the system*

2 Research Background

Personalization [Candillier *et al.* 2008] consists of gathering, storing and analyzing information [Schirru *et al.* 2010] about visitors of a web site or system in order to deliver the right information to each visitor at the right time [Arunachalam and Thambidurai 2010]. Personalization meaning the ability a system has to recommend items a user might find interesting. In this sense, a recommender system may offer particular types of personalization mechanisms [Manouselis and Costopoulou 2007]:

- managing the information overload [Maes 1994, Klein *et al.* 2006],

- aiding to detect the user's preferences,
- relating them to predicted preferred items,
- filtering them from non-interesting or non-relevant responses.

From a process point of view, a recommendation is a response to a user request where an inference task produces a list of items credibly correlated or associated with the user preferences. Formally, let U be a set of users, I a set of items and $v(u, i) : U \times I \to \Re$ a value function, or *rate*, measuring the explicit (or implicit) preference of a user $u \in U$ for an item $i \in I$. Hence, the *user-item* data structure defined as $V = [v(u, i)]_{u \in U, i \in I}$ corresponds to the matrix containing the rates of users for items.

Regularly, a recommender system computes the *aggregated rate* or predicted value of an active user for a given item. Based on that value, the list of most preferred items may be recommended. More precisely, the Top-N recommendation problem may be defined as follows [Deshpande and Karypis 2004]:

> *Given a user-item matrix V and a set of items I that have been rated (or viewed) by a user, identify an ordered set of items X such that $\mid X \mid \leq N$ and $X \cap I = \emptyset$.*

According to the process implemented to identify X, recommender systems may be classified as: *content-based* [Pazzani and Billsus 2007], presenting the user with items similar to those preferred in the past, mainly based on items features or descriptive tags [Memmel *et al.* 2009]; or *collaborative filtering*, where items preferred by similar users are presented to the active user; *hybrid* systems are also recognized, which combine content-based and collaborative filtering approaches [Adomavicious and Tuzhilin 2005]. Indeed, in collaborative filtering two main approaches are usually recognized:

- Memory-based:

  In memory-based algorithms, recommendations are computed based on previously rated items. The *user-based* algorithm class is frequently implemented, which unfolds in three main steps. In the first step, the most similar users, as compared to the active one, are identified. Regular techniques may be used to compute similarity between pairs of rating vectors in V [Choi *et al.* 2010]: Pearson correlation, Jaccard Pearson or the cosine similarity, among others. In the second step, an active user's neighborhood is discerned, based on the similarity measure. Classical methods of doing this are center-based neighborhood, K-Nearest Neighbor and clustering. In the third step, a list of recommendations, ordered by the predicted value, is presented. The value $v(u, i)$ may be calculated as the simple average or the weighted sum of ratings for items evaluated by nearest neighbors, not rated by the active user. Although the user-based approach is very popular, it has two documented drawbacks. First, the low performance in contexts of high number of items/users and sparsity of matrix V, and the "cold-start" problem (when no ratings are available for a user interacting for a first time with a recommender system. [Schein *et al.* 2002]).

– Model-based:

In model-based approaches, a model derived from the analysis of available data is used to predict the $v(u, i)$ values [Sarwar *et al.* 2002] . This is an "off-line" process, updating the model every time enough changes on V have occurred. One implementations of this approach is that users are clustered into classes such that an item rating is predicted from ratings in a class. Several techniques have been implemented for clustering purposes [Sandvig *et al.* 2008]: K-Nearest Neighbor, k-Means clustering, probabilistic Latent Semantic Analysis or Principal Component Analysis, among others [Adomavicious and Tuzhilin 2005]. In some cases, the *item-based* technique is usually implemented, where predicted ratings are based on items correlations instead of users' similarities. It has been argued that if the item-based method is less dynamic than the user-based method, then a model may be constructed [Deshpande and Karypis 2004]. However, in this approach model obsolescence should be considered since changes may affect the accuracy of the recommendations.

The recommender system proposed in this article , which is based on the work presented in [Konow *et al.* 2010], may be classified as an hybrid one as it has characteristics of both model types: It can be considered a model-based approach since it does cluster users according to demographics and user's preferences for movie categories. In addition, aggregated rates are calculated on-line, in a memory-based method, considering users with similar preferences. Practical reasons justify this model. First, assuming that the user's preferences for movie categories are relatively stable, there is no need for frequent user clustering, a process which takes time and resources. Second, instead of maintaining the preferences vector for each user we maintain one for the whole cluster. Clearly, usefulness of our model assumes that clusters are correctly defined and the *nearest* neighbors are detected.

When selecting a recommender system algorithm, properties affecting the user experience need to be identified [Shani and Gunawardana 2011]. Consequently, different techniques exist for evaluating recommender systems, depending on the recommendation purposes [Hernandez and Gaudioso 2008]. On one hand, a recommender system may be evaluated by metrics according to the Information Retrieval research area: recall, precision and ROC. Thus, *recall* measures the coverage of useful items proposed by the recommender system, while *precision* measures the capacity of the system to show only relevant items among recommendations presented to the active user. On the other hand, when items are rated or a measure is assigned to them (for instance, the number of times an item has been chosen), error is usually measured as the capacity of the system to correctly predict the value that a user would assign to an item not yet evaluated. Common measures are mean absolute error (MAE), mean square error (MSE) or root mean square error (RMSE), among others. In addition, different techniques and metrics exist for evaluation in terms of the inferred and real ratings (representing the real interests of a user): Spearman correlation, normalized distance-based performance measure or Half-life utility [Buriano *et al.* 2006]. As will be explained later, our recommender system's quality must be evaluated using metrics taking into account the type of rate used to evaluate items. In this case, the most promising results have been obtained when a "most viewed" rate is used.

3 The proposed Recommender System

Let us consider that the automatic recommendation process is triggered by a user request and obeys to the following steps (see Figure 1):

1. Identification of the active user's preferences (obtaining and representing data).
2. Searching for similar users matching these preferences (clustering).
3. Inference of credibly most preferred items (collaborative filtering).
4. Presentation of preferred items and/or preference values (final recommendation).

Step 1 usually assumes a data structure to represent both a user and his/her preferences. In the case of memory-based approaches the data is compiled into the V matrix. Differently, in the model-based side of our system, personal data such as genre, age and the user's preference for movie categories are used as the information base for clustering purposes. Knowing the cluster where the active user belongs, Step 2 consists of a systematic identification of similar users inside, using the K-Nearest Neighbor algorithm. In order to select the distance measure that best performs in our model, this algorithm has been tested with different metrics: Pearson, Jaccard Pearson, cosine and Euclidean (see Section 3.3). In Step 3, the most suitable items are searched among items rated by similar users and the aggregated rate is computed for each one of them. Step 4 corresponds to the recommendation stage, where the Top-N items are listed. Let us consider the scenario in where a user watching a movie may simultaneously receive a recommendation concerning other movies, coming from the service provider. The main hypothesis behind this work is that people with similar preferences for a certain movie genre and similar profile characteristics may have similar preferences (negative or positive) for movies. In fact, there are reasonable arguments supporting this hypothesis. For example, we might expect that people watching frequently musical movies will be attracted to follow links for a concert or pop music movies. Moreover, if two persons watching the same movie are also in the given age range, there is a high possibility that they like the same music and hence would follow the same items. In the same way, people frequently watching cooking programs on the TV may be also interested in programs about restaurants, or programs where special cooking recipes are shown.

3.1 Clustering users

Many recommender systems find relationships between clusters of users and clusters of items. A newcomer to the system is classified into a cluster in order to present her/him with the most relevant items for the cluster. Given the hypothesis that people watching similar movies might have common preferences and hence follow similar links, we group the users according to the genre of the movies they already watched combined with their age and gender.

In order to measure the preference one user has for a particular movie genre, we adapt the Term Frequency-Inverse Document Frequency, TF-IDF, which is commonly used in the Information Retrieval domain [Spark 1973]. Therefore, let $M_{u,i}$ be the movies of genre i watched by a user u, $M_{u,\bullet}$ the total number of movies watched by u, $M_{\bullet,i}$ the total number of movies of genre i and M the total number of movies. Then, the preference of a user u for a genre i, $P_{u,i}$, expressed as a TF-IDF score, may be defined as follows:

$$P_{u,i} = \frac{M_{u,i}}{M_{u,\bullet}} \times \log(\frac{M}{M_{\bullet,i}}). \tag{1}$$

The score effectively reflects the genre preference of a user. For example, a user who has watched several movies, each one from a different genre, will have a low score for all genres, meaning that the user does not have a special preference for any of them. On the other hand, a user who has watched a small number of movies but most of them are from the same genre, will have a high score indicating a strong preference towards that genre.

The following expression normalizes the previous value, when considering the total sets of genres:

$$\widetilde{P}_{u,i} = \frac{P_{u,i}}{\sqrt{\sum_j P_{u,j}^2}}. \tag{2}$$

Based on this measure a clustering process may be developed. The X-means clustering algorithm [Pelleg and Moore 2000] has been implemented in this case. In Figure 2 five clusters are shown, formed when age and preferred movie genre are used as parameters in (2). The Y axis shows the genres of movies, the X axis shows the age. Bullets inside a cluster indicate a similar score. These results are used by the recommender engine to differentiate one type of user from another based on the genre of movies they watch, thus implementing a preliminary user behavior categorization. Utilizing a collaborative filtering approach, these results can be used to establish an indirect relationship among users within the same cluster.

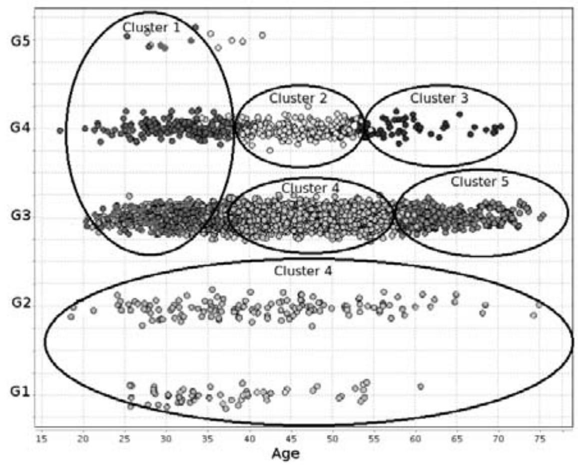

Figure 2: Data clustering from a Movielens sample

3.2 An Implicit Collaborative Filtering method

The goal of the recommender system is selecting a limited number of movies from a huge set, which are the most likely to be clicked by the user. The display time during the film (timing factor), the layout for recommendations and their number per unit of time, plays an important role in the decision of the user on whether to click on it or not. However, we are not going to tackle those aspects in this paper since these are issues for marketing experts, human-computer interface experts and graphic designers, and can be approached independently of (but complementary to) the recommender system itself. We hereby assume the problem the recommender system has to solve is to choose a certain number d of relevant movies for each user.

When selecting the movies, the system knows that a certain user u, aged n, of a certain gender s (male or female), who has been assigned to a cluster C_j ($1 \leq j \leq N_c$, being N_c the number of clusters) is selecting the movie i_g of a certain genre g. We describe this situation with a pair user-movie $(u_{n,s,c}, i_g)$. At the beginning, the system has little information to give preference to one item over others, so the subset of the d selected items which will be presented to the user can be selected randomly. The user will choose some of them and this information will be recorded.

As the system is being used, information is collected about the number of times a user $u \in C_j$ has chosen a certain recommended item A_i. This is registered in a counter variable c_{iju}. Thus, for each item A_i there are N_a associated counter variables, each one for a item. This information is used by the system in the following way: a portion d_{rec} of the d alternative movies the system will present to the user will be chosen from the ones having the highest aggregated value $c_{ij} = \sum_{r \in K_{ju}} c_{ijr}$, where K_{ju} is the K-Nearest Neighbor of the user u. In other words, these will be the items most frequently followed by users belonging to the same cluster.

Another group d_{rand} of items will be chosen randomly, maintaining the equation $d_{rec} + d_{rand} = d$. The reason for having a certain portion of the movies randomly chosen is to avoid the *self-breeding* phenomena, which will cause the system to select always items from the set having the highest counting, which will in turn be the only one with the possibility to increase its number of counts. The numbers d_{rec}, d_{rand} and d are parameters of the system which may be set according to marketing criteria (see Section 4).

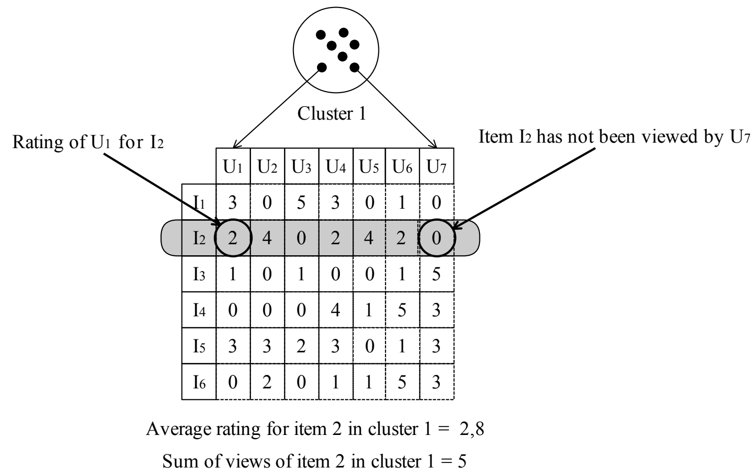

Figure 3: *Movie evaluation schema*

The first rule to choose *drec* reflects the hypothesis that users from the same cluster are likely to find interesting the same items. The second rule avoids that a set of items are repeatedly shown because they were the first ones to be selected by the users of the cluster. In Figure 3 a schematic view of the data structure is shown. The elements from I_1 to I_6 represent the movies and U_1 to U_7 the users in the active user neighborhood. We may assume that values represent the number of times that items have been visited by users. In such a case, a 0 value means that the user has not seen the respective item. In terms of most viewed items by users in the neighborhood, I_5 is the first item chosen by the recommender engine to be presented to the active user. Next, I_2, I_6 are to be included in the Top-N list, etc. Depending on weather the user clicks on any recommended item, the respective counter will be updated. Notice that a similar structure could be used for ratings of users to items, as suggested by the schema in Figure 3.

3.3 Evaluating the recommender system model

In order to evaluate the quality [Hernandez and Gaudioso 2008] of the proposed recommendation system, we conducted an experiment using the MovieLens (www.movielens.org) corpus, which is an open database of anonymous people rating movies. It is maintained by the GroupLens community for research purposes. This is an offline experiment since pre-collected data about users choosing and/or rating data is used [Shani and Gunawardana 2011]. At the time we conducted the test, it contained 3900 movies with 1 million ratings, provided by 6040 users having rated at least 20 movies each; ratings were integer numbers from 1 up to 5. Sparsity of this dataset was measured as $(1000000 - 1)/(6043 * 3900)) = 0.04$.

The experiment setting was defined as follows. First, users were clustered according to the watched movies genre, age and gender (see Figure 2). Second, in order to simulate items not viewed or rated by users, a testing set was defined by randomly deleting 20% of ratings for each user. Thus, the idea was testing if the predicted values match those items previously deleted. Third, four K-NN neighborhood clustering process were implemented for analysis, using the following distance measures: cosine, Euclidean, Pearson, Jaccard Pearson. These metrics were computed by a RapidMiner software implementation (www.rapidminer.com), which allows for an easy selection of clustering parameters and algorithms. The number of neighbors, K, was set at 150 and 500. Fourth, a Top-N strategy [Adomavicious and Tuzhilin 2005] was evaluated, defining $N =$ 20 fixed in this setting.

The recommender system presents the user with a *manageable* list of relevant movies that she/he should click in order to reach the streaming web site. Henceforth, in order to evaluate the recommender system the F-measure metrics was selected, which is defined as the harmonic mean between recall and precision, expressed as follows [Hernandez and Gaudioso 2008]:

$$F_{measure} = 2\frac{precision \times recall}{precision + recall}. \tag{3}$$

Accuracy of the recommender system was evaluated by its capacity to retrieve relevant items among the first 20 recommended movies. Given a specific distance measure and K, up to 33 simulations were run, each one selecting a random testing set (20% of ratings). Then, the average $F_{measure}$ value was computed. Two independent criteria were selected for computing the Top-N items: (i) Predicted-Rating, and (ii) Most-Viewed items. In Figure 4, the $F_{measure}$ metric is compared for the different distance measures, in the case of items recommended by predicted value or most viewed. Only the results for $K \in \{150, 500\}$ are depicted. Qualitative analysis shows that, given a distance measure, the Most-Viewed criterion outperforms Predicted-Rating. Furthermore, the Jaccard Pearson measure is clearly the best decision in this model. Notice that, given a criteria, the Euclidean based recommendations are outperformed in all cases.

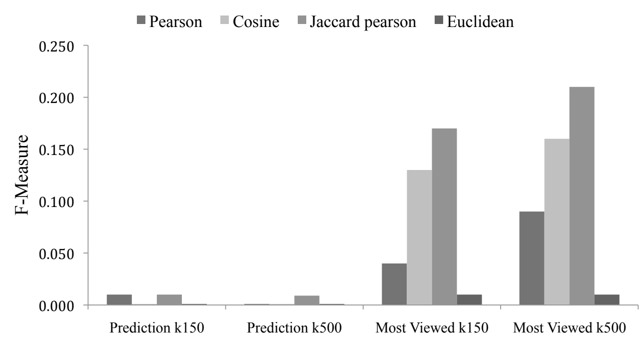

Figure 4: *$F_{measure}$ for "predicted value" and "most viewed" criteria*

Let F^p and F^v be the respective F-measures in the case of predicted ratings (p) and most viewed (v) criteria. The hypothesis to be tested is $H_0 : F^p = F^v$. Values for results in Figure 4 are presented in Table 1. One will notice that values for the Most-Viewed criterion are greater or equal (except for one case) to the Predicted-Rating results, in all cases. Interestingly, the mean $F_{measure}$ obtained in the case of Jaccard Pearson and most viewed (0.21) is comparable to results obtained by [Cremonesi *et al.* 2011] in the case of Non-Normalized Cosine Neighborhood, Asymmetric SVD or Top Popular algorithms.

Given that four methods (distance measures) do not guarantee that statistical significance is achieved, two fictitious metrics were included. The Max-Min row corresponds to a fictitious method having the best values on the prediction criterion, but the worst ones on the viewing criterion. It is assumed that such a method represents a conservative optimal situation in favor of Predicted-Rating. In addition, a Mean row has been included, which corresponds to a fictitious method reaching the mean value obtained for the real distance measures. This method takes into account a moderately acceptable situation in favor of Most-Viewed.

Table 1: $F_{measure}$ for different distance measures

Distance	Predicted-Rating		Most-viewed	
	$K = 150$	$K = 500$	$K = 150$	$K = 500$
Pearson	0.001	0.001	0.040	0.090
Cosine	0.001	0.001	0.130	0.160
Jaccard Pearson	0.010	0.009	0.170	0.210
Euclidean	0.001	0.001	0.010	0.010
Max-Min	0.010	0.009	0.010	0.010
Mean	0.000	0.003	0.090	0.120

The Wilcoxon rank-sum test was applied on F-measures data contained in Table 1. The $F^v - F^p$ negative mean rank was zero both on $K = 150$ and $K = 500$, with $p = 0.043$ (two-tailed) in the first case and $p = 0.028$ (two-tailed) in the latter option. This means that H_0 is rejected at 5% of confidence, but also that the F^v values are higher than the F^p values, which indicates that the Most-Viewed criterion prevails. That seems appropriate for our model since, first, it is not expected that users regularly rate movies and, second, follow-up users' activities well represent the advertising scenario presented in Section 4.

4 The advertising scenario

4.1 Introduction

In a real setting, a Japanese digital media distribution company is planning to add advertising to its delivered video contents (movies and TV programs). The advertising may consist of linked banners, pictures or even in-roll videos which are displayed in multiple ways according to the web page design, including overlays, side bars, etc. The goal of the company is to maximize the number of times that users click on those linked ads since the ad owner pays for each visit to its website generated from the system. The restriction is that during the movie, only a limited number of ads can be shown from a huge available set.

This is true not only because of the limited duration of a movie, but also because presenting too much advertising may upset the viewer or make him/her simply ignore it [Unni and Harmond 2007]. It is then necessary to select and show those ads which the user will most probably follow.

One of the main revenue sources in the IPTV industry is expected to be advertisement and more specifically customer targeted advertisement. In this scenario recommender systems have an important role to play. Recommender systems for customer targeted advertisement are essentially an extension of the existent algorithms for movie-based and user-based recommendation but there are some fundamental differences: a system for customer targeted advertisement must be able to select the best commercial considering at least four factors:

1. who is watching the movie;
2. what is the content (type of movie) being watched;
3. what is the popularity of the scene being played for both users in general and users with a similar preference profile;
4. what is the business model being applied (e.g. maximization of revenues for sponsors);

4.2 The Computer assisted filtering process

The more information we have about users' background, preferences and products characteristics, the better a recommender system may perform [Kiewra 2005]. However, in real systems the availability of this information, as well as the accuracy of it, depends on many uncontrollable factors: users might not want to provide or upgrade their private information; detailed information about advertisement itself is difficult to obtain from the advertising providers, since the number of advertisement items is huge. In consequence, a content-based strategy cannot be applied due to the little information available about the content of the items. The most suitable type of recommender system in this scenario is a collaborative filtering one.

However, it has been shown that the computation of the recommended item vs. user matrix takes $O(U \times A)$ [Linden *et al.* 2003], being U the number of users and A the number of advertising items. This means that a filtering process becomes necessary. Hence, let us consider that during the sign-up process users provide personal information such as age, gender and geographical location. Besides, the movie-related information stored by the system includes title, genre (refined in sub-genres), duration, actors, director, etc. In order to embed, a first filtering stage in the system we may ask the ad owners to include some target metadata about the type of people the advertisement is aimed at, which matches exactly with the information we have from the users: range of age, gender(s) and location (city, prefecture, whole country). Additionally, the ad owner can choose a number of film genders where the advertisement should never be included. This may be particularly interesting when the advertiser might want to avoid its product to be associated with a certain type of films or film content. In any case, we expect a relatively small number of genres avoided by most ad owners compared to the number of genres in which they would like to show their ads.

The digital company has implemented a system to acquire information directly from the users and the ad owners, but it also automatically builds up behavioral information for each user based on its history, for instance, by recording what kind of movies a certain user prefers or at what time he watches them. The system utilizes both the explicitly provided data and the generated behavioral data to establish similarity relations among users through their unique preferences (for instance, see Figure 2).

Data has been obtained from three sources. First, the user's profile, which the user may or may not accurately complete during the registration process. This data set includes gender, age, and address, among others. The second source is the information about the movies. This data set includes, among others, the title, cast, duration, genre and sub genre. The third source is the media access log files which are automatically generated on the media delivery servers. The media access log files include the time of play, stop and pause events, the user ID, the purchased movie ID and the IP address of the user's machine (although this might be that of a firewall or proxy) among other parameters. In particular, through the IP address it is possible to know the location from where each user is connecting using a Geo-IP database service like http://www.hostip.info/ or http://www.ipinfodb.com/. In this particular case, we have used the services of Maxmind GeoIP because its coverage of Japanese locations is quite complete. This information is very convenient especially when the user's address is not provided on the registration. The log files also include data about how long the user watched a particular movie, including the starting point and the end point. This data correspond in many cases to users seeking some particular scene within the movie. This data could be very useful to display advertising before or after the most popular scenes of a movie.

The media servers' log files are parsed using several scripts written in Python. Using the data mentioned in the previous section we implemented a complete log analyzer that is used to merge the information of the access logs and the information available in the database. The log analyzer works as the preliminary process in order to obtain relevant information from users' behavior.

4.3 Where to include recommendations

The display time during the film (timing factor), the layout for recommendations and their number per unit of time plays an important role in the decision of the user on whether to click on it or not. However, we are not going to tackle those aspects in this paper since these are issues for marketing experts, human-computer interface experts and graphic designers, and can be approached independently from (but complementary to) the recommender system itself. We hereby assume the problem the recommender system has to solve is that of choosing a certain number of relevant movies for each user.

Data available in log files is of paramount importance to target advertising since it allows for the answering of questions like:

- What are the most popular movies for female/male customers between X and Y years old living in the regions A, B and C?
- Which are the hottest scenes inside a certain movie for a given group of users?
- Where do the majority of the young customers live?
- How many elderly customers do we have, and what do they watch?

This information certainly helps advertisers in defining the genre of the films where the ad should not be displayed, the age of the target audience and the geographical location where the ad should appear (see section 3.1). It also helps the business planning division of the company to decide which type of advertising has more possibilities of being successful and when to show it during the play screening of a certain movie. Since the data contained in the log file is so large it is very important to display it in an aggregated and compact way. In order to do this, we developed a log analyzer tool, which graphically displays this information, allowing a technical user to set the relevant parameters to filter the information.

The main functionality of this tool displays an interface where a specialized user has to enter the age, gender and location parameters for filtering purposes. As an example, in Figure 5, the user has chosen to filter the data for male and female customers between 18 and 40 years old living in Tokyo, Kyoto, Osaka, Kanagawa, Aichi, Chiba, Saitama and Hokkaido. After this, various charts are displayed.

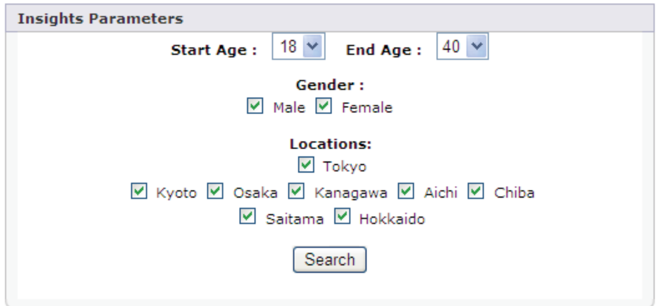

Figure 5: *Filtered data from Japanese database*

Following this, two pie charts showing the age distribution and gender distribution of the selected customers group are displayed (see Figure 6).

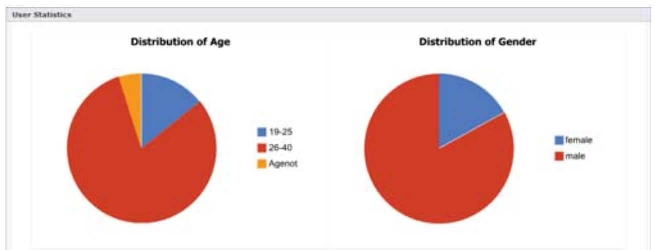

Figure 6: *Distribution of age and genre of the selected group*

In Figure 7 a pie chart showing the regional distribution of the customers group is displayed, while in Figure 8 a Google Map shows the amount of customers from each city.

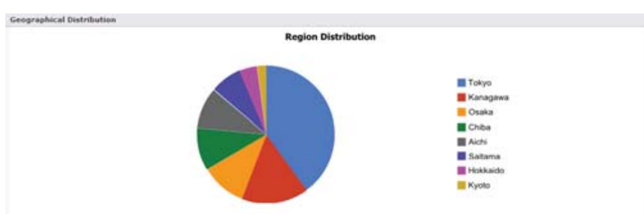

Figure 7: *Distribution of selected users by region*

Figure 8: *Google Maps usage for showing user visits*

Among other functionalities, the log analyzer presents statistics for a certain movie genre or a single movie. For example, in Figure 9 we see a chart showing the number of people who have seen a certain part of a movie. The movie is divided into several one-minute-long pieces and the log analyzer counts how many users have seen each piece. By looking at this report we can easily find the scenes in the movie attracting large number of people. This information can be used to decide when (or when not) to display advertisements.

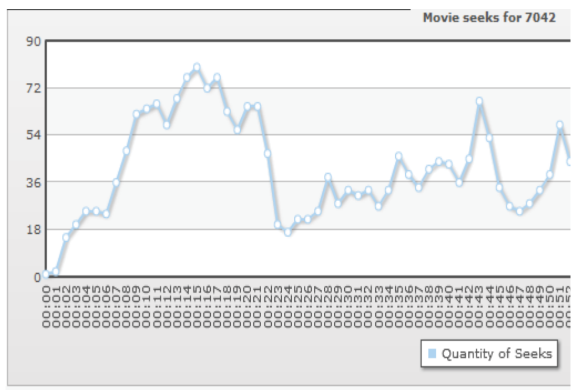

Figure 9: *Number of "seeks" for a certain movie*

5 Conclusions

In this paper we presented a recommender system which was developed to solve the problem of recommending items to people while watching movies offered by a IPTV Company in a real scenario. Existing recommender systems showing a good performance require a great amount of computing power and time and are difficult to implement because of their complexity. We developed a recommender system based on users clustering and counting of times that users in a cluster select advertising items. The recommendation process consists of 6 steps, which can be grouped into two categories: full automatic and human assisted. The full automatic set of steps comprises a users' clustering process taking in account their item preferences and adapting a TF-IDF score among other characteristics like gender and age. Once the clusters are calculated, a collaborative filtering algorithm is applied inside each cluster. It is used to select the TOP-20 recommended items for each cluster. Various K-Nearest Neighbor distance metrics were evaluated by running several experiments using the Movielens Database.

The selected criteria for evaluating the performance of the recommender system were two: Most-Viewed (ranking the top 20 most viewed movies from inside the cluster) and Predicted-Rating (ranking the top 20 movies that on average were the best rated). Results show that the most viewed criteria outperforms using the F-Measure evaluation value. Using the Jaccard Pearson distance measure for the K-Nearest Neighbors selection and using a Most-Viewed criteria, an average F-Measure value of 0.21 was obtained. These results can be compared to Non-Normalized Cosine Neighborhood, Asymmetric SVD or Top Popular algorithms. However, the method presented in this paper can be seen as a more simple to understand and implement.

A log analyzer tool was implemented which is used to help company marketing experts to decide whether to apply "filters" to advertisings in order to select the population it should or should not reach, like gender, age, and/or geographical location. This tool also helps deciding in which part of the movie the advertising item should be shown. This human assisted filtering process needs to be evaluated and is part of our future work.

The main contribution of this work is a recommender system which performs as good as others in the literature, but being much simpler to understand and implement. It uses only the strict necessary information to compute results, saving computational as well as human resources. It also tackles the cold start and over-fitting problems, very common in recommender systems. However, future tests need to be done in order to cope with performance aspects as computational efficiency, the effects of the choice of parameters in the model and the clustering approach. An evaluation process on the real scenario described above is also intended, refining our results by testing pre and post filtering provided by the log analyzer tool.

References

[Adomavicious and Tuzhilin 2005] 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.

[Broadband 2010] Broadband Forum: Strong DSL and IPTV growth, press article released on April 2010, Last visited on September 2010 http://mybroadband.co.za/news/broadband/11908-Strong-DSL-and-IPTV-growth.html.

[Buriano *et al.* 2006] Buriano, L., Marchetti, M., Carmagnola,F., Cena, F., Gena, C. and Torre, I. "The role of ontologies in context-aware recommender systems". MDM'06 Proceedings of the 7th International Conference on Mobile Data Management (2006), 80.

[Candillier *et al.* 2008] Candillier, L., Jack, K, Fressant, F. and Meyer, F. "State-of-the-art Recommender Systems". In Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling. Eds. Chevalier, M., Christine, J., Soulé-Dupuy, C., Information Science Reference Publisher (2008), 1-22.

[Choi *et al.* 2010] Choi, S. S., Cha, S. H. and Tappert, C. "A Survey of Binary Similarity and Distance Measures". Journal on Systemics, Cybernetics and Informatics, 8, 1(2010), 43-48.

[Cotriss 2009] Cotriss, D.: IPTV Advertising Comes To Life , http://www.dailyiptv.com/news/iptv-advertising-rises-031207/, October 2010.

[Cremonesi *et al.* 2011] Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A. and Turrin, R. "Comparative evaluation of recommender system quality". Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems, Vancouver, BC, Canada, May 07-12 (2011), 1927-1932.

[Deshpande and Karypis 2004] Deshpande, M. and Karypis, G. "Item-based top-N recommendation algorithms". ACM Transactions on Information Systems (TOIS), 22, 1(2004), 143-177.

[Hernandez and Gaudioso 2008] Hernández del Olmo, F. and Gaudioso, E. "Evaluation of recommender Systems: a new approach". Expert Systems with applications: An International Journal, 35 (2008), 790-804.

[Kiewra 2005] Kiewra, M. "RankFeed - Recommendation as Searching without Queries: New Hybrid Method of Recommendation". Journal of Universal Computer Science, 11, 2(2005), 229-249.

[Klein *et al.* 2006] Klein, G., Brian M and Hoffman, R. "Making Sense of Sensemaking 2: A Macrocognitive Model". Intelligent Systems, IEEE, 21, 5 (2006), 88-92.

[Konow *et al.* 2010] Konow, R., Wayman, T., Loyola,L. , Pereira, J. and Baloian,N. "Recommender System for contextual advertising in IPTV scenarios", Computer Supported Collaborative Work in Design CSCWD (2010), 617-622.

[Arunachalam and Thambidurai 2010] Arunachalam, K. and Thambidurai, P. "Collaborative Web Recommendation Systems - A Survey Approach". Global Journal of Computer Science and Technology, 9, 5 (2010), 30-35.

[Linden *et al.* 2003] Linden, G., Smith, B. and York, J. "Amazon.com recommendations: item-to-item collaborative filter", IEEE Internet Computing, 7, 1(2003), 76-80.

[Maes 1994] Maes, P. "Agents that reduce work and information overload", Communications of the ACM, 37, 7 (1994), 30-40.

[Manouselis and Costopoulou 2007] Manouselis , N. and Costopoulou, C. "Analysis and classification of multi-criteria recommender systems". World Wide Web: Internet and Web Information Systems, Special issue on Multi-channel Adaptive Information Systems on the WWW, 10, 4 (2007), 415-441.

[Memmel *et al.* 2009] Memmel, M., Kockler, M. and Schirru, R. "Providing Multi Source Tag Recommendations in a Social Resource Sharing Platform". Journal of Universal Computer Science, 15, 3 (2009), 678-69.

[Pazzani and Billsus 2007] Pazzani, M. and Billsus, D., "Contet-Based Recommendation Systems", The Adaptive web, LNCS 4321 (2007), 325-341.

[Pelleg and Moore 2000] Pelleg D. and Moore, A. "X-means: Extending K-means with efficient estimation of the number of clusters", Proc. 17th Int. Conf. Machine Learning (ICML"00) (2000) 727-734.

[Piper 2010] Piper, B., "Report on IPTV Forecast and Outlook: 13.7 Billion by 2012", Strategy and Analytics, Feb. 27th (2008).

[Sandvig *et al.* 2008] Sandvig, J.J., Mobasher, B. and Burke, R. "A Survey of Collaborative Recommendation and the Robustness of Model-Based Algorithms". Bulletin of the Technical Committee on Data Engineering, 31, 2 (2008), 3-13.

[Sarwar *et al.* 2002] Sarwar, B.M., Karypis, G., Konstan, J. and Riedl, J. "Scalable neighborhood formation using clustering". Proceedings of the Fifth International Conference on Computer and Information Technology (2002).

[Schein *et al.* 2002] Schein, A., Popescul, A., Ungar, L. and Pennock, D. "Methods and Metrics for Cold-Start Recommendations". Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002). New York City, New York: ACM., 253-260.

[Shani and Gunawardana 2011] Shani, G. and Gunawardana, A. "Evaluating Recommendation Systems". In Recommender Systems Handbook (2011), 257-297.

[Spark 1973] Spark Jones, K. "A statistical interpretation of term specificity and its application in retrieval". Journal of Documentation 28, 1(1973), 11-21.

[Schirru *et al.* 2010] Schirru, R., Baumann, S., Memmel, M. and Dengel, A. "Extraction of Contextualized User Interest Profiles in Social Sharing Platforms" Journal of Universal Computer Science, 16, 16 (2010), 2196-2213.

[Unni and Harmond 2007] Unni, R. and Harmond, R., "Perceived Effectiveness of Push vs. Pull Mobile Location-based Advertising". Journal of Interactive advertising, 7, 2(2007), 28-40.

[Van den Dam 2007] Van den Dam, R. "IPTV ADVERTISING: a gold mine for telcos?" last visited October 2010 http://broadcastengineering.com/RF/ broadcasting_iptv_advertising_gold/index.html.