A Recommender System Based on Multi-features

Maria Trujillo, Marta Millan, and Edward Ortiz

School of Systems Engineering and Computer Sciences,
Universidad del Valle, Ciudad Universitaria Melendez, Cali, Colombia
{mtrujillo,millan}@eisc.univalle.edu.co

Abstract. Recommender systems are tools to help users find items that they deem of interest to them. They can be seen as an application of data mining process. In this paper, a new recommender system based on multi-features is introduced. Demographic and psychographic features are used to asses similarities between users. The model is built on a collaborative filtering method and addresses three problems: sparsity, scalability and cold-star. The sparsity problem is tackled by integrating users-documents relevant information within meta-clusters. The scalability and the cold-start problems are considered by using a suitable probability model calculated on meta-cluster information. Moreover, a weight similarity measure is introduced in order to take into account dynamic human being preferences behaviour. A prediction score for generating recommendations is proposed based on the target user previous behaviour and his/her neighbourhood preferences on the target document.

Keywords: Recommender systems, collaborative filtering, hierarchical clustering, weight similarity measure.

1 Introduction

The World Wide Web development provides a way for accessing digital information in different contexts and domains. Huge amounts of available information make it difficult for users to get information according to their needs and preferences. Recommender systems are common and powerful solution tools. Their goal is to provide a user with personalised recommendations based on either his/her tastes and preferences or a group of people with similar tastes and preferences. In this context, personalisation could be seen as an application of a data mining process [20] in which data collection, pre-processing, building user profiles and evaluation phases are done in order to deliver personalised recommendations.

Depending on how recommendations are generated, recommender systems are classified into three categories [1]: content-based, collaborative and hybrid approaches.

Content-based filtering systems generate a user profile based on the content description of items in which that user has expressed some interest. In most content-based filtering systems, the content descriptors are textual features. These systems rely on well-known documented modelling techniques based on information retrieval [4] and information filtering [32]. Profiles are individual and personal by nature and

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4706, Part II, pp. 370–382, 2007.

are based on the user's previous rating of items [5] [6] [12] [15] [18] [19] [21] [27]. These techniques tend to be overspecialised on a limited set of items that are explicitly associated with previous user preferences. Thus, the user would never receive a recommendation for items to which he/she has not given a rating.

Collaborative recommender systems are based on an important feature of human behaviour that is the tendency to consume a limited set of items. Thus, a set of items becomes fashionable for a group of people with similar preferences. Collaborative filtering systems are based on similarity of users in which a neighbourhood to each user is generated. A neighbourhood is built taking into account user's similarity. An item is recommended to a target user depending on the ratings, explicit or implicit, his/her neighbourhood has assigned to it. Since collaborative filtering systems rely on user-to-user similarities, three major limitations have been identified: sparsity, scalability, and cold-star. Sparsity occurs when only a few of the total number of items available in a database are rated by the users. Scalability refers to a recommender system capable of providing recommendations even when the number of users and items in a database scales up to thousands and beyond. Finally, cold-start usually describes the case in which items cannot be recommended for new users. This problem also applies to new and obscure items and to users with eclectic tastes.

Hybrid systems combine the advantages of content-based and collaborative recommendation systems in order to overcome their limitations. These methods have also been used in [5] [8] [11] [16] [28].

In this paper, we introduce an approach that combines memory-based and model-based strategies of collaborative filtering. The recommendation calculations are divided into two phases: user preferences pattern building in an off-line phase and user preferences pattern prediction in an on-line phase. A user preferences pattern building is based on users' characteristics and preferences and is calculated by clustering users with similar characteristics and preferences. A weight similarity measure is proposed for clustering users in order to take into account dynamic human being preferences behaviour. Items information is integrated into user clusters in order to produce meta-clusters users-items. The user preferences pattern prediction uses a probability model for calculating the items closest to those which an active user may be interested in.

The paper is organized as follows. Related works including limitations and discussion of recommender methods based on ratings are outlined in Section 2. A recommender system based on multi-features is introduced in Section 3. Deployment and preliminary evaluation are presented in Sections 4 and 5, respectively. Final remarks are made in Section 6.

2 Related Works

At present, several recommender systems have been designed and developed based on different approaches and techniques [6] [9] [15] [19] [23] [26] [27] [29] [30]. In [8] recommendation techniques are deeply analysed taking into account the input data, the recommendations and the algorithms to apply on data. Five classes of recommendation techniques are proposed in terms of the background data, input data and the algorithm to generate recommendations: collaborative, content-based, demographic, utility-based and knowledge-based. Among them, the collaborative filtering technique is the one most frequently applied [13] [14] [18] [21] [23] [27].

A collaborative filtering process consists of three steps [24]: customer profile construction, neighbourhood formation and recommendation generation. Based on a preferences database for items by user, neighbours are calculated for each new user. A user's neighbourhood shares a new user's tastes and preferences. Other techniques, including clustering and Bayesian networks have also been applied. In particular, clustering techniques build clusters or groups of users who share similar preferences. Clustering techniques are useful for improving performance because the size of a users group is reduced. Though the scalability problem is solved when cluster algorithms are applied to large user data sets [14], the quality of recommendations may be reduced [7] [25].

Several methods have been proposed for solving major limitations of collaborative filtering [15] [22] [31]. ClustKNN [22] is a hybrid collaborative filtering algorithm which addresses the scalability problem using the k-means algorithm for building a user model and the k-nearest neighbours (KNN) clustering technique for calculating predictions.

Kim *et al.* [15] addressed two of the major limitations of collaborative filtering – cold start and data sparseness – using blurring values and a conditional probability model. This probabilistic model is generated under the assumption that items are not related to each other in any way, *i.e.* they are independent.

Yeong *et al.* [31] used binary information for representing user's behaviour over a period of time. A user's pattern model is built using a clustering self organizing map and items are related through association rules. Recommendations are calculated by mapping related items and user patterns. The main drawback of this approach is the high level of sparseness in the binary database which reduces recommendation accuracy.

An item-to-item collaborative filtering algorithm is proposed in [19]. According to the authors, it is scalable independently of the number of customers and items taken into consideration. A recommender system based on intelligent agents is proposed by Kim and Lee [17], in which a user preferences and updates profiles are learned by interacting user-system using a social filtering strategy.

Most collaborative techniques work based on ratings about items provided for users. This kind of systems is classified into either memory-based or model-based. The former uses a utility function, like correlation, in order to calculate similarity between users. The latter builds a model from historical data to recommend other items. In both cases, ratings can be obtained explicitly or implicitly.

Explicitly ratings are subjective user evaluations or voting. Thus, the utility function is evaluated using ratings in calculating mathematical operations regardless of the fact that these mathematical operations are not defined for subjective user evaluations. Indeed, the mean of preferences is meaningless. Although subjective evaluations are usually represented by numbers, they are really labels such as: "low", "medium", "high". Moreover, a distance is not defined between labels though labels could be ordered and their ranks can be used for calculations. Garden *et al.* [11] and Herlocker *et al.* [14] have considered the use of the Spearman correlation, which is just the Pearson correlation computed for the ranks of the ratings.

3 A Recommender System Based on Multi-features

The proposed approach is integrated into a computer sciences digital library research project. In this context, users are students and items are digital documents, such as papers, books, research reports and theses. This approach uses an unobtrusive method for recommendation calculations that takes into account not only the user's registration basic information but also information related to the documents he/she has previously downloaded.

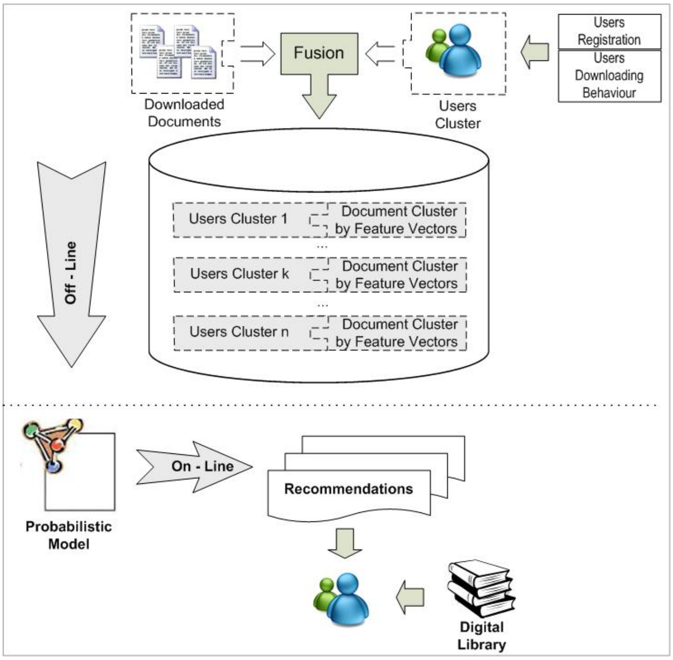

Fig. 1. Digital library recommender architecture

The system architecture is shown in Fig. 1. This system combines the advantages of memory-based and model-based collaborative recommendation systems. Memory-based – in an off-line phase – deals with meta-clusters calculations in order to reduce the computational complexity – scalability problem. Model-based – in an online phase – uses a conditional probability model for the preferences prediction based on meta-clusters information. Thus, a user preferences pattern is calculated in an off-line phase. His/her preferences prediction is calculated in an on-line phase.

3.1 Similarity Based on Multi-features

All users have to register in the digital library. This makes two types of features available: demographic – such as age, university relation, and higher academic degree – and psychographics – such as interest areas – which correspond to ACM computer science knowledge areas. When records about users downloaded documents are available, areas of downloaded documents are used as psychographic features.

Moreover, since human being preferences are dynamic over time, demographic and psychographics features are classified as static or dynamic. This classification is used in calculating similarity between users. On the other hand, a 2x2 contingency table is commonly used for calculating similarity measures. However, **d** can be seen as sparse information and has no meaning in user-based similarity. Neither the calculation of **d** nor similarity measures based on **d** are considered in this work. Fig. 2 shows the features classification and their contingency tables.

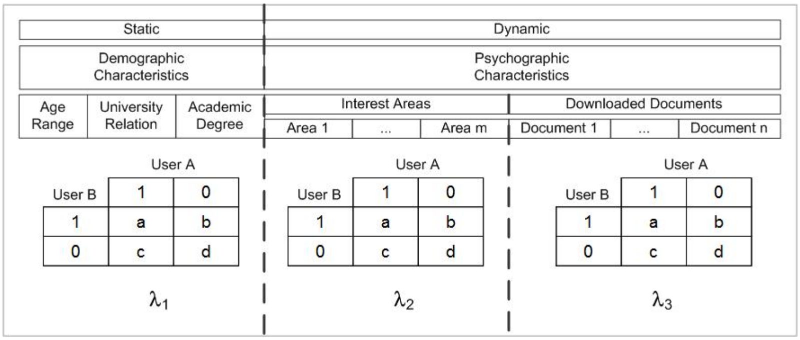

Fig. 2. Features classification, contingency tables and weight factors

The similarity calculation between users ***A*** and ***B*** is defined as follows. The **a**, **b** and **c** values are calculated into each feature space.

```
DemographicFeature(userA,userB)
Features = {ageRange,universityRelation,academicDegree}
for f in Features
  if (userA.getFeature(f)==userB.getFeature(f))
     a=a+1
  else
     b=b+1
     c=c+1
```

Let L_a and L_b be lists or sets of interest areas declared by **user*A*** and **user*B***, respectively.

```
InterestAreas(La,Lb)
a=|La∩Lb|
b=|Lb|-|La∩Lb|
c=|La|-|La∩Lb|
```

Let D_a and D_b be lists or sets of downloaded documents by **user*A*** and **user*B***, respectively.

```
DownloadedDocuments(Da,Db)
a=|Da∩Db|
```

b= | D_b | − | $D_a \cap D_b$ |
c= | D_a | − | $D_a \cap D_b$ |

where |.| is the cardinality of a set or a list of items.

In a standard way of calculating similarities between users, the three 2x2 contingency tables – in Fig. 2 – are reduced to one by summing them up as a matrix operation. However, we propose a weight similarity measure in order to take advantage of dynamic human being preferences behaviour. The weight similarity measure is defined by

$$S(A,B)=\sum_{i=1}^{3}\lambda_i S_i(A,B), \tag{1}$$

where λ_i is a weight that $\lambda_i \in [0,1]$, and $\sum_{i=1}^{3}\lambda_i = 1$.

The use of weights – in Fig. 2 – allows us to define which characteristics are more important at a specific moment. For instance, for a new user with no downloaded records the λ_3 value is set to zero. On the other hand, static information may be irrelevant when there are downloaded behaviour records available. In this case the λ_1 value could tend to zero. When the user downloaded behaviour is not consistent with declared interest areas, the λ_2 value could tend to zero. The similarity measure S_i can be chosen from a large list of available similarity measures in the literature [3] [10]. Table 1 shows commonly used similarity measures.

Table 1. Similarity measures

Name	Measure
Sokal and Sneath	a/(a+2(b+c))
Jaccard	a/(a+b+c)
Dice	2a/(2a+b+c)
Ochiai	a/sqrt((a+b)(a+c))
Kulczynski	(a/(a+b) + a/(a+c))/2

In practical applications, distances rather than similarities are of interest. The similarity S is transformed into a distance d [10]. Large similarities correspond to small distances and *vice versa*. The transformation is calculated as:

$$d(A,B = 1 - S(A,B), \quad if\ S \in [0,1], \tag{2}$$

$$d(A,B) = 1 - \frac{S(A,B)}{\max\{S\}}, \quad if\ S \notin [0,1]\cdot \tag{3}$$

Once a user-to-user distance matrix is available a hierarchical algorithm is used for clustering. Suppose every user is regarded as a cluster with only one element. Then the algorithm starts using the following steps:

1. Combine the two clusters with the smallest distance.
2. Recalculate the distances for the newly formed group to all remaining clusters.
3. Repeat steps 1 and 2 until all users lie in one big cluster.

The hierarchical algorithm result is usually presented in a graphical way, using a dendrogram, as it is shown in Fig. 3.

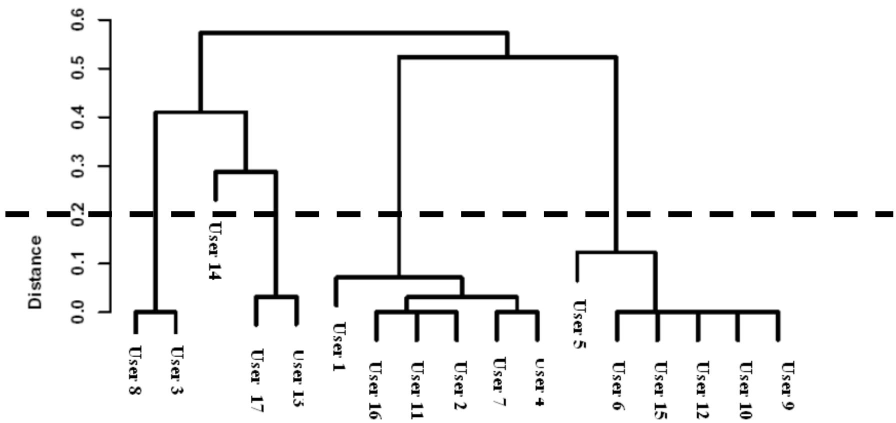

Fig. 3. Dendrogram: cluster formation process

The number of clusters in the partition is defined according to a predefined distance value, *d*. For instance, by setting d=0.2 five clusters are obtained in Fig. 3. The centroid of a user-cluster is the mode value of each feature (attribute). The centroid is used in classifying new users, those who have not been registered.

Users and documents information is linked by a fusion process. Fusing user-clusters and documents is the concatenating of downloaded documents within each user-cluster. This information is called meta-cluster.

Moreover, each document has a features vector that contains knowledge area, and language (*e.g.* mainly English, French or Spanish) of the document. Documents are clustered by these two features within each meta-cluster. Meta-clusters are used for calculating the predicted score.

3.2 Predicted Scores

Recommendations are based on the probability that a user has preference for a documents. If this probability is high it is more likely that a recommendation on a document will be useful for the user. For the sake of completeness, we outline the notation used in this section.

In our approach, a target user belongs to a meta-cluster $\boldsymbol{k}$ and a target document has a feature vector $\boldsymbol{j}$. Let $\boldsymbol{A}$ be the target user who belongs to the meta-cluster $\boldsymbol{k}$. Let D_a be the list of downloaded documents by $\boldsymbol{A}$. Let D_{ja} be the list of downloaded documents with feature vector $\boldsymbol{j}$ by $\boldsymbol{A}$. Let $\boldsymbol{d}$ be the target document, with feature vector $\boldsymbol{j}$. Let C_k be the list of user in the meta-cluster $\boldsymbol{k}$. Let C_{kd} the list of users in the meta-cluster $\boldsymbol{k}$ who downloaded $\boldsymbol{d}$.

The predicted score that $\boldsymbol{A}$ has preferences for $\boldsymbol{d}$ is proportional to the conditional probability that $\boldsymbol{A}$ has preferences for documents with features vector $\boldsymbol{j}$ plus the conditional probability that users in $\boldsymbol{C}_k$ have shown preferences for $\boldsymbol{d}$.

The predicted score for $\boldsymbol{A}$ having preferences for $\boldsymbol{d}$ can be computed as:

$$P_s(\boldsymbol{d}|\boldsymbol{A}) = \frac{|D_{ja}|}{|D_a|} + \frac{|C_{kd}|}{|C_k|}, \quad (4)$$

where |.| is the cardinality of a set or a list of items.

Thus, the predicted score is calculated based on the target user previous behaviour and his/her neighbourhood preferences on the target document.

If $\boldsymbol{A}$ has not shown preferences for documents with features vector $\boldsymbol{j}$, the predicted score is calculated based solely on his/her neighbourhood preferences. When user $\boldsymbol{A}$ and users in his/her neighbourhood have shown particularly interest in documents with features vector $\boldsymbol{j}$, it may occur that the predicted score takes values larger than one.

4 Deployment

Our approach for recommending documents is integrated into a digital library experimental framework. The architecture of the integrated system is shown in Fig. 4. The digital library is supported on a data base that contains information mainly on user registrations and documents. The recommender system is supported on a data mart. Information such as document metadata, documents downloaded by user, and characteristic vectors is stored in the data mart. Recommendations are generated based on historical downloaded data, demographic and psychographics user's data.

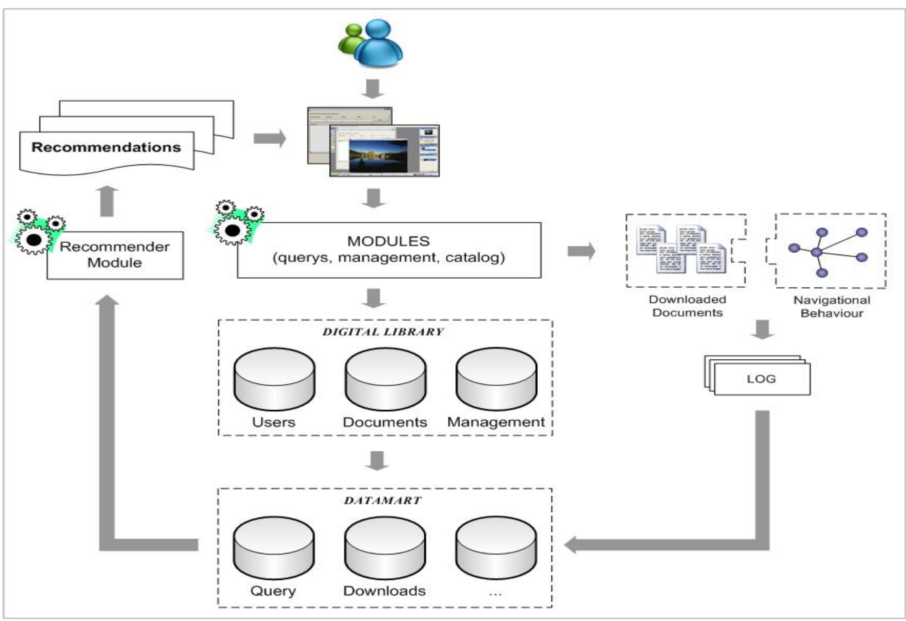

Fig. 4. The System Architecture

Recommendations are twofold: a list of recommendations based on collaborative filtering and a second list based on downloaded document frequencies. Documents in the former, as in other recommender systems, are ranked according to their predicted score – in Equation 4. A verification process on previous recommendations is carried on in order not to deliver the same recommendations. However, collaborative filtering techniques provide recommendations regardless of users current interests. As for the second list, a user current interest is shown as his/her navigational behaviour. After his/her first search using key words, the main area of knowledge associated to those key words is used in a documents query and documents are sorted out by downloaded frequency. A list of n-top documents is generated and a recommendation window is shown. Moreover, users have the choice of entering the window and check the recommendations included in the list or closing that window for the time being. Whatever courses of action a user takes are registered in the data mart.

5 Preliminary Evaluation

We have presented an approach to personalised information retrieval tasks in a digital library environment. The personalisation process [2] is defined by three stages: understanding customers, delivering personalised offerings, measuring personalisation impact. In this paper, we focused on understanding users and delivering personalised offering phases. Moreover, once the digital library data mart contains enough information – in this specific domain – we will be able to evaluate our approach.

Current available repositories and dataset are not adequate for the evaluation of the proposed recommender system. However, the performance of the proposed recommendation system was preliminary evaluated using the MovieLens data set, developed at the University of Minnesota [available at http://www.grouplens.org]. The dataset contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.

A prediction accuracy metric, the relative frequency with which the system makes correct decisions about whether a movie is of interest to a user, was used in the preliminary evaluation.

In a preprocessing step, 100 users were selected with the following available demographic features: Gender={ 75 male users, 25 female users }, Age Range={ 38 users in "18-24", 36 users in "25-34", 14 users in "35-44" , 12 users in "45-49"} and Occupation={ 17 users are "other", 8 users are "academic/educator", 33 users are "clerical/admin", 17 users are "lawyer", 5 users are "sales/marketing", 20 users are "self-employed"}

Records about a user's movie genres of interest are not available to be used as psychographic features. In this evaluation, the λ_2 value – in Fig. 2 – was set to zero, in Equation 1. As a psychographic feature, we took as implicit rating the action of rating a movie. In a pre-processing phase, the most frequent rated genre, which is drama, was selected for the experimental validation. Available information was divided into 90% training set and 10% cross-validation set. That is, when a user has 10 ratings, 9 ratings are used for building the model and 1 rating is used for validating the model.

Table 2. Total of available implicit ratings for building the model

Total of used rating	Total No. of users
[1, 10]	20
(10, 20]	19
(20, 30]	15
(30, 100]	28
(100, 200]	14
(200, 305]	4

Table 2 summarises the number of ratings in the training set used for building the model. For instance, 20 of the selected users had rated between 1 and 10 drama movies.

Clusters were calculated using λ_1=0.3, λ_2=0, and λ_3=07. Six clusters were formed. Table 3 summarises total of users by cluster.

Table 3. Number of users by cluster

Cluster	Total No. of users
Cluster 1	14
Cluster 2	14
Cluster 3	3
Cluster 4	32
Cluster 5	4
Cluster 6	33

Table 4 shows the values obtained for the prediction accuracy metric.

Table 4. Results obtained for the prediction accuracy metric

Recommendations list size	Value of the prediction accuracy metric
1	37%
2	47%
3	54%
4	65%
5	73%
6	77%
7	79%
8	80%

When one recommendation was generated for each user, 37% of the users had rated the recommended movie in his/her cross-validation set. When a list with 8 recommendations was generated for each user, 80% of the users had rated at least one of the recommended movies in his/her cross-validation set.

6 Final Remarks

Our proposal takes into account a user model that integrates user demographic features (*e.g.* user individual features such as educational, age and gender), user psychographics features (*e.g.* interest areas) and user behavioural downloaded documents. We proposed a weight similarity measure that combines those features. Knowledge about user's interest areas could be replaced by knowledge about actual interest represented by means of downloaded documents. This can be made by tuning the weights in the proposed similarity measure. For generating recommendations, a prediction score is calculated based on the target user previous behaviour and his/her neighbourhood preferences on the target document.

Three main drawbacks of the collaborative filtering technique were considered. The sparsity problem was tackled by integrating users-documents relevant information within meta-clusters. The scalability and the cold-start problems were considered by using a suitable probability model calculated on meta-cluster information.

Our next step, when the digital library data mart contains enough information, is to

- calibrate and characterise the weights in the proposed similarity measure,
- propose an asymmetric similarity measure for reducing the impact of large quantities of downloaded documents against small quantities of downloaded documents. We have detected that the difference in document downloaded quantities may underestimate the similarity between users,
- build a lifetime model for evaluating recommendations, and
- use a Bayesian approach for taking into account information no longer used in recommendations calculation, as *a priori*.

Acknowledgments. This work was supported in part by the Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología "Francisco José de Caldas" (Colciencias) under Grant No. 1406-14-17902. The authors would like to acknowledge Edward Gallego's contribution to the first version of the recommender system.

References

1. Adomavicius, G., Tuzhilin, A.: Toward a Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transaction on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Adomavicius, G., Tuzhilin, A.: Personalization Technologies: A Process-Oriented Perspective. Communications of the ACM 48(10), 83–90 (2005)
3. Anderberg, M.: Cluster Analysis for Applications. Academic Press, New York (1973)
4. Baeza, R., Ribeiro, B.: Modern Information Retrieval. Addison Wesley ACM Press Series (1999)
5. Balabanovic, M., Shoham, Y.: Content-based collaborative recommendation. Communications of the ACM 40(3), 66–72 (1997)
6. Bollacker, K., Lawrence, S., Giles, L.: A System for Automatic Personalized Tracking of Scientific Literature on the Web. In: International Conference on Digital Libraries. Proceedings of the fourth ACM Conference on Digital libraries, Berkeley, California, USA, pp. 105–113. ACM, New York (1999)

7. Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of predictive algorithms for collaborative filtering. In: Proceedings of 14th Annual Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Morgan Kaufman, San Francisco (1998)
8. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. In: User Modelling and User-Adapted Interaction 2007 (to appear)
9. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: Proceedings of ACM SIGIR'99 Workshop on Recommender Systems. ACM Press, New York (1999), citeseer.ist.psu.edu/claypool99combining.html
10. Cox, T., Cox, M.: Multidimensional Scaling, 2nd edn. Chapman and Hall, London (2001)
11. Garden, M., Dudek, G.: Mixed Collaborative and Content-Based Filtering with User-Contributed Semantic Features. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence. AAAI Press, Menlo Park, CA (2006)
12. Golberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A Constant Time Collaborative Filtering Algorithm. UCB Electronics Research Laboratory Technical Report M00/41 (2000), Available at www.ieor.berkeley.edu/golberg/pubs/
13. Golberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative Filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)
14. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An Algorithm Framework for Performing Collaborative Filtering. In: Proceedings of ACM SIGIR'99, Berkeley, California, pp. 230–237. ACM Press, New York (1999)
15. Kim, H., Kim, J., Herlocker, J.: Feature-Based Prediction of Unknown Preferences for Nearest-Neighbour Collaborative Filtering. In: Perner, P. (ed.) ICDM 2004. LNCS (LNAI), vol. 3275, pp. 435–438. Springer, Heidelberg (2004)
16. Kim, Y., Yum, B., Song, J., Kim, S.: Development of a recommender system based on navigational and behaviour patterns of customers in e-commerce sites. Expert Systems with Applications 28, 381–393 (2005)
17. Kim, J., Lee, E.: Intelligent Information Recommender System on the Internet. In: 1999 International Conference on Parallel Processing Workshops (ICPPW'99), pp. 376–380 (1999)
18. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. Communications of the ACM 40(3), 77–87 (1997)
19. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
20. Mobasher, B.: Data Mining for Personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. LNCS, vol. 4321. Springer, Heidelberg (2007), http://maya.cs.depaul.edu/ mobasher/
21. Pennock, D.M., Horvitz, E., Lawrence, S., Giles, C.L.: Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach. In: Proceedings of the 16th Conference on Uncertainty and Artificial Intelligence, Stanford California, pp. 473–480 (2000)
22. Rashid, A.M., Lam, H., Karypis, G., Riedl, J.: ClustKNN: A highly Scalable Hybrid Model-& Memory-CF Algorithm. In: Proceedings of WEBKDD'06, Philadelphia, Pennsylvania USA (2006)
23. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work, Chapel Hill, NC USA, pp. 175–186. ACM Press, New York (1994)

24. Schafer, J.B., Konstan, J.A., Reidl, J.: E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery 5(1-2), 115–153 (2001)
25. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for E-commerce. In: EC '00 Proceedings of the 2nd ACM conference on Electronic commerce, Minneapolis, Minnesota, United States, pp. 158–167. ACM, New York (2000)
26. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithm. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, pp. 285–295 (2001)
27. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating "Word of Mouth". In: Conference on Human Factors in Computing Systems. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver Colorado, USA, pp. 210–217 (1995)
28. Torres, R., McNee, S., Abel, M., Konstan, J., Riedl, J.: Enhancing Digital Libraries with Techlens. In: Proceedings of JCDL'04, Tucson, Arizona, USA, pp. 228–236 (2004)
29. Ungar, L., Foster, D.: A Formal Approach to collaborative filtering. In: Proceedings of the Conference on Automated Learning and Discovery (CONALD'98) (1998)
30. Ya Tang, T., McCalla, G.: Mining Implicit Rating for Focused collaborative Filtering for Paper Recommendations. Available at http://www.ia.uned.es/elena/um03-ws/papers/UM03workshop-Tiffany.pdf
31. Yeong, B.C., Yoon, H.C., Soung, H.K.: Mining Changes in Customer Buying Behaviour for Collaborative Recommendations. Expert Systems with Applications 28(2), 359–369 (2005)
32. Xuan, Z., Yuan-Yuan, S., Hyoung-Gook, K.: An Integrated Music Recommender System. IEEE Transaction on Consumer Electronic 52(3), 917–925 (2006)