PROBABILITY-BASED EXTENDED PROFILE FILTERING
An Advanced Collaborative Filtering Algorithm for User-generated Content

Toon De Pessemier, Kris Vanhecke, Simon Dooms, Tom Deryckere and Luc Martens
Wireless and Cable group, Dept. of Information Technology, IBBT, Ghent University
Gaston Crommenlaan 8, Ghent, Belgium

Keywords: Recommender system, Personalization, Collaborative filtering, Profile, User-generated Content, Algorithm.

Abstract: The enormous offer of (user-generated) content on the internet and its continuous growth make the selection process increasingly difficult for end-users. This abundance of content can be handled by a recommendation system that observes user preferences and assists people by offering interesting suggestions. However, present-day recommendation systems are optimized for suggesting premium content and partially lose their effectiveness when recommending user-generated content. The transitoriness of the content and the sparsity of the data matrix are two major characteristics that influence the effectiveness of the recommendation algorithm and in which premium and user-generated content systems can be distinguished.
Therefore, we developed an advanced collaborative filtering algorithm which takes into account the specific characteristics of user-generated content systems. As a solution to the sparsity problem, inadequate profiles will be extended with the most likely future consumptions. These extended profiles will increase the profile overlap probability, which will increase the number of neighbours in a collaborative filtering system. In this way, the personal suggestions are based on an enlarged group of neighbours, which makes them more precise and diverse than traditional collaborative filtering recommendations. This paper explains in detail the proposed algorithm and demonstrates the improvements on standard collaborative filtering algorithms.

1 INTRODUCTION

Various Web 2.0 sites (e.g. YouTube, Flickr, Digg, Google Video…) have an overwhelming bulk of user-generated content available for online consumers. Although this exploding offer can be seen as a way to meet the specific demands and expectations of users, it has complicated the content selection process to the extent that users are overloaded with information and risk to 'get lost': although there is an abundance of content available, it is often difficult to obtain useful and relevant content.

Traditional filtering tools, e.g. keyword-based or filtered searches, are not capable to weed out irrelevant content. A second filtering based on the general popularity (expressed by user ratings or consumption patterns) can assist but requires a broad basis of user feedback before it can make reasonable suggestions. Moreover, this technique does not consider personal preference and individual consumption behaviour, since only the most popular content will be favoured by the majority of the community. This situation reinforces the role of (collaborative) filtering tools and stimulates the development of recommendation systems that assist users in finding the most relevant content.

2 RELATED WORK

The overabundance of content and the related difficulty to discover interesting content items have already been addressed in several contexts. Online shops, like Amazon, apply collaborative filtering (CF) to personalize the online store according to the needs of each customer (Linden et al., 2003). Purchasing and rating behaviour are valuable information channels for online retailers to investigate consumers' interests and generate personalized recommendations (Karypis, 2001).

Netflix is an online, mail-based, DVD rental service for customers in the United States. They have the possibility to express their appreciation for a rented movie by a star-rating mechanism on the Netflix website. This simple feedback method forms the basis of the Netflix recommender. Convinced by the potential of a good recommendation system, Netflix pub-

De Pessemier T., Vanhecke K., Dooms S., Deryckere T. and Martens L.
PROBABILITY-BASED EXTENDED PROFILE FILTERING - An Advanced Collaborative Filtering Algorithm for User-generated Content.
DOI: 10.5220/0002780102190226
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

lished a large dataset and started a competition to find the most suitable recommendation algorithm for their store in October 2006. In this context, many research groups have competed to find the best movie recommendation algorithm based on the Netflix dataset (Bell and Koren, 2007).

The introduction of digital television entailed an increase in the number of available TV channels and the information overload linked thereto. Consequently, new standards to describe this content, e.g. TV-Anytime (Evain and Martínez, 2007) and advanced electronic program guides, which simplify the navigation and selection of TV programs, became necessary (Lee et al., 2005). Several personalized TV guide systems which filter and recommend TV programs according to the user's preferences, have been developed for set-top boxes (Zhang et al., 2005) and personal digital recorders (Kurapati et al., 2001; Yu and Zhou, 2004).

Besides these traditional premium content sources (i.e. professionally-generated content), user-generated content (like personal photos, videos, or bookmarks) received a more prominent role on the web in recent years. These web 2.0 applications use more pragmatic approaches, like tagging, to annotate content than the traditional classification systems like TV-Anytime. Such a metadata description which is the contribution of the whole community, also known as a folksonomy, became very popular on the web around 2004. Since users might tag in a different manner and use other synonyms, the annotation and classification of this user-generated content is less strict, which causes an additional difficulty for content-based recommendation systems (Golder and Huberman, 2005).

3 COLLABORATIVE FILTERING

3.1 Traditional Collaborative Filtering Techniques

CF techniques are the most commonly used recommendation algorithms because they generally provide better results than content-based techniques (Herlocker et al., 1999). Most user-based CF algorithms start by finding a set of customers whose purchased or rated items overlap the user's purchased and rated items. Customers can be represented as an N-dimensional vector of items, where N is the number of distinct catalogue items. Purchased or rated items are recorded in the corresponding components of this vector. This profile vector is extremely sparse (i.e. contains a lot of missing values) for the majority of customers who purchased or rated only a very small fraction of the catalogue items.

The similarity of two customers, j and k, symbolized by their consumption vectors, U_j and U_k, can be measured in various ways. The most common method is to measure the cosine of the angle between the two vectors (Sarwar et al., 2000).

$$Sim(\vec{U}_j,\vec{U}_k) = cos(\vec{U}_j,\vec{U}_k) = \frac{\vec{U}_j \cdot \vec{U}_k}{||\vec{U}_j||\,||\vec{U}_k||} \quad (1)$$

Next, the algorithm aggregates the consumed items from these similar customers, eliminates items the user has already purchased or rated, and recommends the remaining items to the user (Linden et al., 2003).

An alternative for this user-based CF technique is item-based CF, a technique that matches each of the user's purchased and rated items to similar items and then combines those similar items into a recommendation list. For measuring the similarity of items, the same metrics can be employed as with the user-based CF. Because of scalability reasons, this technique is often used to calculate recommendations for big online shops, like Amazon, where the number of users is much higher than the number of items.

3.2 Collaborative Filtering in Sparse Data Sets

Despite the popularity of CF, its applicability is limited due to the sparsity problem, which refers to the situation that the consumption data in the profile vectors are lacking or are insufficient to calculate reliable recommendations. In an attempt to provide high-quality recommendations even when data profiles are sparse, some solutions are proposed in literature (Papagelis et al., 2005). Most of these techniques use trust inferences, transitive associations between users that are based on an underlying social network, to deal with the sparsity and the cold-start problems (Weng et al., 2006). Nevertheless, these underlying social networks are in many cases insufficiently developed or even nonexistent for (new) web-based applications that desire to offer personalized content recommendations.

Default voting is an extension to the traditional CF algorithms which tries to solve this sparsity problem without exploiting a social network. A default value is assumed as 'vote' for items without an explicit rating or purchase (Breese et al., 1998). Although this technique enlarges the profile overlap, it can not identify more significant neighbours than the traditional CF approach.

A direct consequence of this sparsity problem is that the number of similar customers, the neighbours of the users, can be very limited in a user-based CF technique. Furthermore, because of this sparsity, the majority of these neighbours will also have a small number of consumed items in their profile vectors. Because the prospective personal recommendations are limited to this set of consumptions of neighbours, the variety, quality and quantity of the final recommendation list might be inadequate.

A comparable reasoning is applicable to item-based CF techniques that work on sparse profile data. Users might have consumed a small number of items, which in turn also have a limited number of neighbouring items. Again, the CF algorithm is restricted to a narrow set of product to generate the personal suggestions which is disastrous for the efficiency of the recommender.

3.3 Collaborative Filtering for User-generated Content Systems

Because of the nature of user-generated content systems, the number of content items is in many cases significantly bigger compared to premium content systems. User-generated content requires less production efforts. Accordingly, the content production rate and the number of distinct publishers are massive. For example, YouTube enjoys 65,000 daily new uploads (Cha et al., 2007). Due to these varied content (production) characteristics, the sparsity problem will even become worse for user-generated content. Therefore, the recommender performance might be disappointing if the traditional CF techniques will be ported from premium content delivery systems to user-generated content delivery systems without any adaptation.

We developed an advance CF algorithm that extends profiles based on the probability that an item will be consumed in the future. These extended profiles will increase the profile overlap probability, which will increase the number of neighbours in a CF algorithm. These extended profiles, in reaction to the sparsity problem, lead to more precise and varied content recommendations.

4 PROBABILITY-BASED EXTENDED PROFILE

One of the consequences of a sparse data matrix is that the number of neighbours for a user/item might be very limited in a user-based/item-based CF system. Having no or limited neighbours leads to insufficient and imprecise recommendations. The majority of the similarity metrics that are used in CF systems rely on the profile overlap to determine the similarity of two users/items. So, to increase the amount of neighbours, the number of overlapping profiles has to be increased, which can be achieved by a greater amount of consumption behaviour. Because stimulating users to consume more content is not an option in most cases, we opted for an artificial profile extension based on the future consumption probability.

Our developed algorithm is an iterative process. In the first phase, a traditional CF algorithm will try to generate personal suggestions based on the existing profiles. In the second phase, all the initial profiles that do not contain a minimum number of consumptions will be extended. These first two stages can be repeated in order to reach a minimum threshold for the profile size. The extended profiles will be used to recalculate the similarities in a third phase; and at last, the final recommendations will be generated in the fourth phase.

4.1 Traditional CF Recommendations

In the first phase, a traditional CF algorithm will be used to generate standard CF recommendations. These recommendations will be applied to extend the sparse profiles in the next phase.

4.2 Extending Profiles

To make sparse profiles more dense, possible future consumptions are inserted in the profile vectors, in the second phase. These additional consumptions are based on two information sources: the general probability and the profile-based probability that the item will be consumed in the near future.

4.2.1 User-based

In user-based CF systems, existing user profiles will be supplemented with the items that have the biggest probability to be consumed by the user in the near future. The general probability that a specific item will be consumed by a specific user without a priori knowledge of the user is proportionally to the current popularity of the item. Especially for user-generated content systems, this popularity can vary rapidly in time. In addition, the probability that the item will be consumed by the user can also be calculated based on the user's profile as a priori knowledge. This probability will be inverse proportional to the index of the item in a personal top-N recommendation list, and can be estimated by the confidence value which is calculated by the traditional CF system in phase 1. After all, this top-N recommendation list is a prediction of the items which the user will like/consume in the near future.

4.2.2 Item-based

In the item-based case, item profiles, which contain the users who consumed the item in the past, will be supplemented with the most likely future consumers. The general probability that a specific user will consume a specific item, without any knowledge of the item, is proportional to the present intensity of the consumption behaviour of that user. With some a priori knowledge of that item, the calculations can be repeated. Then, the probability will be inverse proportional to the index of the user in a top-N list of users who are the most likely to consume the product in the future. This list and the associated confidence values can be generated by the results of the traditional item-based CF (Segaran, 2007).

4.2.3 Potential Consumption Behaviour

Based on this calculated general and profile based probability, the user or item profiles will be completed until the minimum profile threshold is reached. However, these predicted consumptions will be marked as uncertain in contrast to the initial assured consumptions. For example, for a web shop, the real purchases correspond to a 1, which refers to a 100% guaranteed consumption, while the potential future consumptions are represented by a decimal value between 0 and 1, according to the probability value, in the profile vector. This second phase can consist of several successive iterations to complete the profiles.

4.3 Recalculating the Similarities

Based on these extended profile vectors, the similarities will be recalculated with the chosen similarity metric, e.g. the cosine similarity (equation 1), in a third phase. Because of the added future consumptions, the profile overlap and accordingly the number of neighbours will be increased compared to phase 1. For the item-based case, these similarities can e.g. be used for a ‘related products’ section in online shops.

4.4 Generating Recommendations

To produce personal suggestions, a recommendation vector will be generated based on these extended profile vectors, in a fourth phase. For a user-based algorithm, the recommendation vector, R_j, for user j can be calculated as:

$$\vec{R}_j = \frac{\sum_{k=1,k\neq j}^{N} \vec{U}_k \cdot Sim(\vec{U}_j, \vec{U}_k)}{\sum_{k=1,k\neq j}^{N} Sim(\vec{U}_j, \vec{U}_k)} \qquad (2)$$

where U_j and U_k represent the consumption vectors of users j and k, which might contain real values. Subsequently, the top-N recommendations are obtained by taking the indices of the highest components of the recommendation vector, R_j, and eliminating the items which are already consumed by user j in the past.

5 EVALUATION DESIGN AND MEASUREMENT

5.1 Data Set

To estimate the effectiveness of personal recommendations, two different evaluation methods are possible. On the one hand, online evaluations measure the user interactivity (e.g. clicks, buying behaviour) with the personal suggestions on a running service. Offline evaluations, on the other hand, use a test set with consumption behaviour which has to be predicted based on a training set with consumption history. Although online evaluation methods are the most close to reality, we opted for an offline evaluation based on data sets because such an evaluation is fast, reproducible and commonly used in recommendation research.

Therefore, we compared the proposed recommendation algorithm with the traditional CF algorithm based on evaluation metrics which are generated by an offline analysis using a data set with consumption behaviour. Because the datasets that are commonly used to benchmark recommendation algorithms (e.g. Netflix, Movielens, Jester) contain to few ‘small’ profiles and handle only premium content, we evaluated our algorithm on a dataset of PianoFiles[1], a user-generated content site that offers users the opportunity to manage their collection of music sheets. The logged consumption behaviour for constructing profiles consists of individual additions of music sheets to personal collections. The full data set contains 401,593 items (music sheets), 80,683 active users and 1,553,586 distinct consumptions in chronological order.

5.2 Evaluation Method

For evaluation purposes, we use 50% of the consumptions that are the most recent ones as the test set and

[1] http://www.pianofiles.com/

use the remaining 50% of the consumption records as the input data. In order to study the performance of the algorithm under data of different sparsity levels, we form ten different training sets by selecting the first 10%, 20%, 30%, until 100% of the input data. The recommendation algorithm uses these different training sets in successive iterations to generate personal suggestions.

As commonly done for the evaluation of recommendations under sparse data (Huang et al., 2004), the test set is first filtered to only include consumptions that are possible to predict with the input data as a priori knowledge. A consumption of a sheet that is not contained in the input data or a consumption of a user without any consumption behaviour in the input data is not possible to predict with CF techniques. All users in this filtered test set are included into a set of target consumers. For each of these consumers, the algorithm generates five ordered lists of 10, 20, 30, 40 and 50 recommendations respectively, which will be compared with the test set. This offline evaluation methodology, in which a data set is chronologically splitted in training set and test set, is commonly used for evaluating recommendation algorithms (Hayes et al., 2002).

6 EVALUATION

6.1 Evaluation Metrics

One of the most used error metrics is the Root Mean Squared Error (RMSE) (Herlocker et al., 2004; Campochiaro et al., 2009), which is also adopted by the official Netflix contest. However, the Netflix contest is mainly focused on predicting accurate ratings for an entire set of items, while web-based (e-commerce) applications are most interested in providing the users with a short recommendation list of interesting items (Campochiaro et al., 2009). To evaluate this top-N recommendation task, i.e. a context where we are not interested in predicting user ratings with precision, but rather in giving an ordered list of N attractive items to the users, error metrics are not meaningful. Therefore, information-retrieval classification metrics, which evaluate the quality of a short list of recommendations, are the most suitable.

The most popular classification accuracy metrics are the precision and recall (Campochiaro et al., 2009). The precision is the ratio of the number of recommended items that match with future consumptions, and the total number of recommended items. In offline evaluations, the consumptions of the test set represent the future consumptions, and the recommendations that match with these consumptions are called the *relevant recommendations*.

$$Precision = \frac{\#\,Relevant\ recommendations}{\#\,Recommendations} \qquad (3)$$

The recall stands for the ratio of the number of relevant recommendations and the total number of future consumptions. Only these future consumptions are considered as *relevant items* for the end-users in offline evaluations.

$$Recall = \frac{\#\,Relevant\ recommendations}{\#\,Relevant\ items} \qquad (4)$$

It has been observed that precision and recall are inversely related and are dependent on the length of the result list returned to the user (Herlocker et al., 2004). When more items are returned, precision decreases and the recall increases. Therefore, in order to understand the global quality of a recommendation system, we may combine precision and recall by means of the F1-measure

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad (5)$$

6.2 Bench-marked Algorithms

Because item-based algorithms generally achieved a very low performance on the PianoFiles dataset, we did not include any item-based technique in the evaluation. This poor performance is mainly due to the nature of the dataset, which contains a much larger number of items than users. Therefore, forming item neighbourhoods is actually much more difficult than forming user neighbourhoods (Huang et al., 2004). Furthermore, with this proportion of users and items, there is a big risk that item-based algorithms will trap users in a ‘similarity hole’, only giving exceptionally similar recommendations; e.g. once a user added a sheet of Michael Jackson to her collection, she would only receive recommendations for more Michael Jackson sheets (McNee et al., 2006).

Compared to this item-based CF, a user-based strategy achieves much better results on the PianoFiles dataset. In a first evaluation, we benchmarked this standard user-based CF algorithm (UBCF), which operates on the initially existing profiles (Segaran, 2007), against the user-based version of our probability-based extended profile filtering (UBExtended), which extends the sparse profiles before generating the actual recommendations. For this performance evaluation, the UBExtended algorithm expands sparse profiles to a target size of 6 consumptions and the cosine similarity (equation 1) is used as a measure to compare profile vectors in both algorithms.

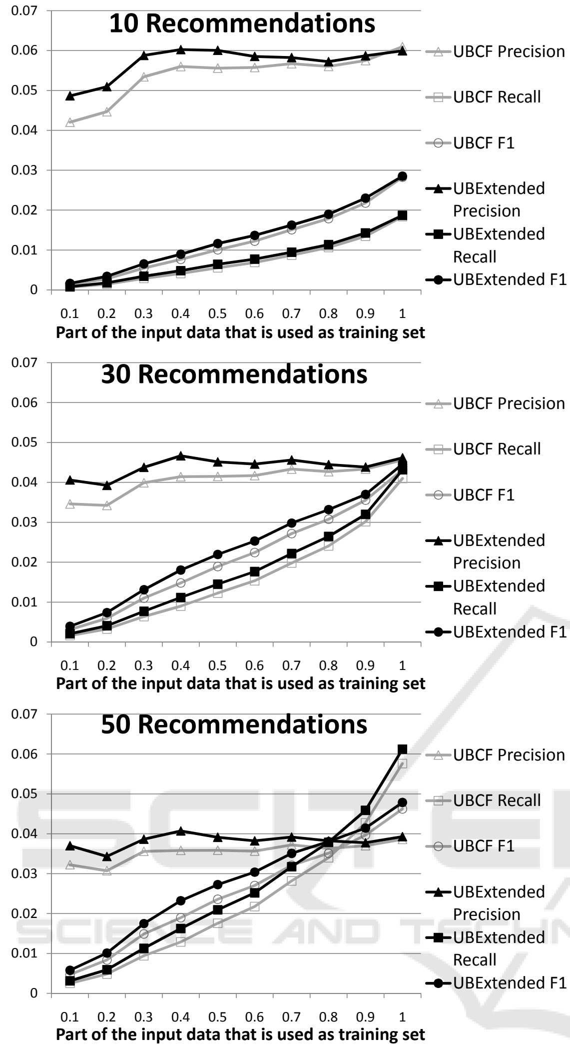

Figure 1: The evaluation of the UBCF and UBExtended algorithm based on the initial training set.

6.3 Complete Training Set

The graphs in Figure 1 illustrate that the evaluation metrics for these two algorithms increase, while more training data becomes available. As the size of the training set increases, more user behaviour becomes available, including the behaviour of new users for which no information was available in the first part(s) of the training set. This additional data enables the generation of recommendations for more users, which explains the increasing recall value.

Besides information of new users, supplementary training data will also contain consumption behaviour of users who have already an initial music collection. This extra information can refine the user profiles, which leads to a higher precision. However, after the profile size reached a critical point, supplementary training data has no more additional information value, which leads to a stagnating precision value. Moreover, the recommendations for new users, which will generally have a lower precision value because of a limited early profile, will enhance this stagnation effect. At last, the F1 metric will follow the progress of this precision and recall graph closely because of its definition. Besides these general trends, the graphs indicate that the UBExtended algorithm exceeds the standard UBCF in all three evaluation metrics (precision, recall and F1) and for different sizes of the recommendation list. This improvement is especially noticeable for small training sets, which mainly consist of sparse user profiles.

6.4 Filtered Training Set: Sparse Profiles Only

To illustrate this superiority of the UBExtended algorithm for sparse profiles, a second evaluation is performed. To scrutinize the recommender performance for the subset of users with a sparse profile, the training set is submitted to an extra filter. This filter removes all the users with more than x consumptions from the training set to simulate the situation of a very novel content delivery system without ‘well-developed’ user profiles. In accordance with the first evalution in which we extended sparse profiles to a target size of 6 consumptions, we chose in this second analysis for a filter that removes all users with a profile that is larger than this target size. In this way, a standard UBCF that operates on a dataset with only sparse profiles (profile size ≤ 6 consumptions), will be compared with the UBExtended algorithm which broadens these profiles to the target size (profile size $= 6$ consumptions) before generating recommendations. Given the long-tail distribution of the profile size in content delivery systems (Figure 2), this subset of sparse-profile consumers, constitutes a considerable segment of the system users.

Since, the filter modifies the training and test sets, the absolute values of this evaluation can not be compared to the absolute values of the first evaluation, which is based on the unfiltered data sets. However, the differences between the UBCF and UBExtended algorithm in this second evaluation, illustrated in Figure 3, compared with the differences between these algorithms in the first evaluation (Figure 1), confirm that the performance improvement of the UBExtended algorithm increases for more sparse datasets. Finally, the graphs of this second evaluation show that for small training sets the precision might slightly fluctuate due to unsufficient data and many new users.

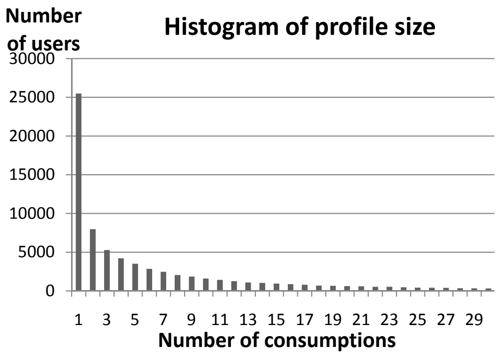

Figure 2: Histogram of the consumption profile size for the PianoFiles dataset.

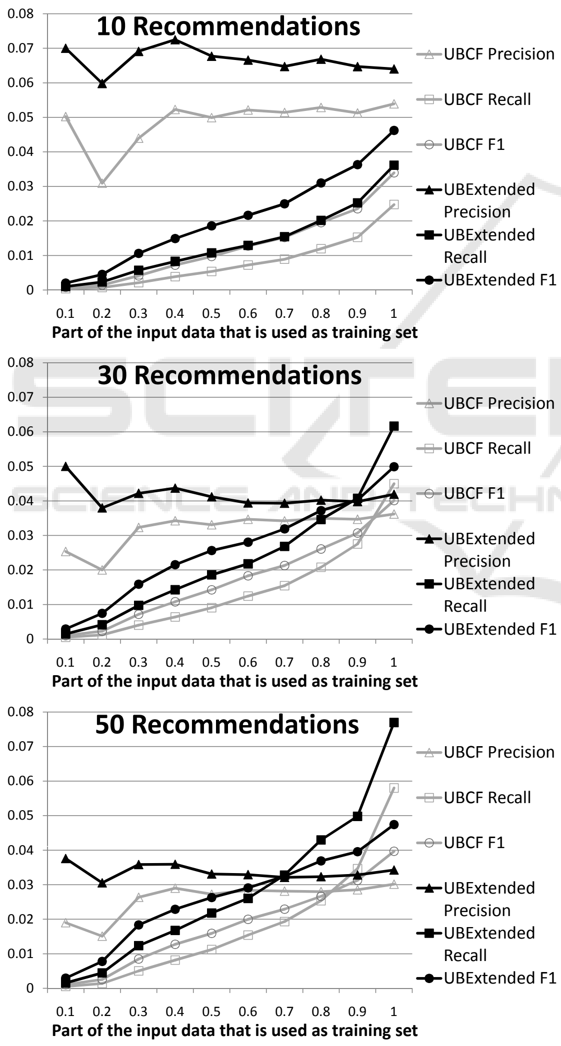

Figure 3: The evaluation of the UBCF and UBExtended algorithm based on the training set, which contains only sparse profiles.

7 OPTIMISATION & DRAWBACK

Since the parameters of the UBExtended algorithm are not yet optimized, the performance difference between the two bench-marked algorithms might even increase considerably. The UBExtended algorithm extends sparse profiles until each profile contains a predefined number of consumptions. This target profile size is an important parameter that has to be optimized in function of the performance metrics. Although we have chosen a fixed value of 6 consumptions for the extended profiles in our evaluation, we believe this parameter might be a function of general dataset statistics, namely the overall sparsity of the data matrix, the number of items and the number of users. Moreover, the procedure of extending the profiles, which is based on general and profile-based influences, can be fine-tuned. An optimal balance between general and profile-based information to extend the profiles might result in more precise recommendations. Finally, there are some typical CF parameters which have to be determined such as the similarity measure and the number of neighbours which are used to calculate the recommendations.

Unfortunately, the accuracy improvement acquired with the UBExtended algorithm is associated with an extra calculation cost. Compared to the standard UBCF, the UBExtended algorithm consists of 2 extra phases: extending the profiles and recalculating the similarities after this extension. Especially the similarity calculation can be time-consuming due to its quadratic nature and therefore may pose problems for systems that calculate the recommendations in real-time (i.e. generating recommendations when the web-page is requested). However, since most recommender systems schedule the calculations and update their recommendations periodically, the additional calculation time is no major problem.

8 CONCLUSIONS & FUTURE WORK

In this research, we developed an advanced collaborative filtering algorithm which takes into account the specific characteristics of user-generated content systems. The algorithm extends sparse profiles with the most likely future consumptions based on general and personal consumption behaviour. Our experimental study, using a dataset from the user-generated content site PianoFiles, showed that the user-based version of the proposed algorithm achieves better performances than the standard user-based collaborative filtering algorithm, especially on sparse data sets. These results proof that there is a need to adapt traditional collaborative filtering techniques to the specific characteristics, such as the sparsity, of user-generated content websites. In future research, we will optimize the algorithm parameters to further improve the performance results. Besides we will investigate if the principle of profile extension is applicable in other types of collaborative filtering algorithms.

ACKNOWLEDGEMENTS

We would like to thank the Research Foundation - Flanders (FWO), for the research position of Toon De Pessemier (Aspirant FWO). We would also like to express our appreciation to Thomas Bonte, the founder of PianoFiles, for putting the data set of his website at our disposal.

REFERENCES

Bell, R. M. and Koren, Y. (2007). Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann.

Campochiaro, E., Casatta, R., Cremonesi, P., and Turrin, R. (2009). Do metrics make recommender algorithms? *Advanced Information Networking and Applications Workshops, International Conference on*, 0:648–653.

Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. (2007). I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 1–14, New York, NY, USA. ACM.

Evain, J.-P. and Martínez, J. M. (2007). Tv-anytime phase 1 and mpeg-7. *J. Am. Soc. Inf. Sci. Technol.*, 58(9):1367–1373.

Golder, S. and Huberman, B. A. (2005). The structure of collaborative tagging systems. *ArXiv Computer Science e-prints*.

Hayes, C., Massa, P., Avesani, P., and Cunningham, P. (2002). An on-line evaluation framework for recommender systems. In *In Workshop on Personalization and Recommendation in E-Commerce (Malaga*. Springer Verlag.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA. ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53.

Huang, Z., Zeng, D., and H.Chen (2004). A link analysis approach to recommendation with sparse data. In *AMCIS 2004: Americas Conference on Information Systems*, New York, NY, USA.

Karypis, G. (2001). Evaluation of item-based top-n recommendation algorithms. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254, New York, NY, USA. ACM.

Kurapati, K., Gutta, S., Schaffer, D., Martino, J., and Zimmerman, J. (2001). A multi-agent tv recommender. In *Proceedings of the UM 2001 workshop Personalization in Future TV*, pages 13–14.

Lee, H., Yang, S.-J., Kim, J.-G., and Hong, J. (2005). Personalized tv services based on tv-anytime for pdr. *Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on*, pages 115–116.

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

McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1097–1101, New York, NY, USA. ACM.

Papagelis, M., Plexousakis, D., and Kutsuras, T. (2005). *Trust Management*. Springer Berlin / Heidelberg.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 158–167, New York, NY, USA. ACM.

Segaran, T. (2007). *Programming collective intelligence*. O'Reilly.

Weng, J., Miao, C., Goh, A., Shen, Z., and Gay, R. (2006). Trust-based agent community for collaborative recommendation. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1260–1262, New York, NY, USA. ACM.

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

Zhang, H., Zheng, S., and Yuan, J. (May 2005). A personalized tv guide system compliant with mhp. *Consumer Electronics, IEEE Transactions on*, 51(2):731–737.