Attribute-Aware Collaborative Filtering

Karen Tso and Lars Schmidt-Thieme

Computer-based New Media Group (CGNM),
Institute for Computer Science,
University of Freiburg, 79110 Freiburg, Germany

Abstract. One of the key challenges in large information systems such as online shops and digital libraries is to discover the relevant knowledge from the enormous volume of information. Recommender systems can be viewed as a way of reducing large information spaces and to personalize information access by providing recommendations for information items based on prior usage.

Collaborative Filtering, the most commonly-used technique for this task, which applies the nearest-neighbor algorithm, does not make use of object attributes. Several so-called content-based and hybrid recommender systems have been proposed, that aim at improving the recommendation quality by incorporating attributes in a collaborative filtering model.

In this paper, we will present an adapted as well as two novel hybrid techniques for recommending items. To evaluate the performances of our approaches, we have conducted empirical evaluations using a movie dataset. These algorithms have been compared with several collaborative filtering and non-hybrid approaches that do not consider attributes. Our experimental evaluations show that our novel hybrid algorithms outperform state-of-the-art algorithms.

1 Introduction

Recommender systems use collaborative filtering to generate recommendations by predicting what users might be interested in, given some user's profile. It is commonly used as a customization tool in e-commerce and is seen as a personalization technology. Unlike the conventional approach where all users view the identical recommendations, a recommender system further personalizes these recommendations such that each user will receive customized recommendations that suit his/her tastes. A few prominent online commercial sites (eg. amazon.com and ebay.com) offer this kind of recommendation services.

Two prevailing approaches to developing these systems are Collaborative Filtering (CF; Goldberg et al. 1992) and Content-Based Filtering (CBF). There are two different recommendation tasks typically considered: (i) predicting the ratings, i.e., how much a given user will like a particular item, and (ii) predicting the items, i.e., which N items a user will rate, buy or visit next (topN). As most e-commerce applications deals with implicit ratings, the latter seems to be the more important task and we will focus on it for the rest of the paper.

In CF, recommendations are generated first by computing the similarities between others' profiles to identify a set of users, called "*neighborhood*" pertaining to a particular user's profile. Usually, the similarities between the profiles are measured using Pearson's Correlation or Vector Similarity. Finally, the recommendations are derived from this neighborhood. One technique in generating the topN recommendations is the Most-Frequent Recommendation (Sarwar *et al.* 2000) where the frequency of all items of the neighborhood is considered and the N number of items with the highest frequency is returned.

There are two general classes of CF algorithms — Memory-based (User-Based) and Model-based (Resnick *et al.* 1994; Breese *et al.* 1998; Sarwar*et al.* 2000). User-Based CF is one of the most successful and prevalent techniques used in recommender systems. The entire database is employed to compute the similarities between users. Using this similarity, a dualistic form of the User-Based CF called the *Item-Based topN* algorithm emerged (Deshpande and Karypis 2004). It uses the items instead of the users to determine the similarities. The Item-Based CF has claimed to significantly outperform the User-Based CF. On the other hand, the model-based CF builds a model by learning from the database (Breese *et al.* 1998; Aggarwal *et al.* 1999).

In CBF methods, the users are defined by the associated features of his/her rated items. These features are usually the attributes or description of the object. In contrast to CF techniques, CBF recommends items to users based solely on the historical data from the users (Balabanovic and Shoham 1997; Burke 2002; Ziegler *et al.* 2004).

Since attributes usually contain meaningful and descriptive information of objects, there have been attempts in combining these two approaches, so-called hybrid approaches, to gain better performance. In this article, we will introduce three methods which incorporate item attributes and focus on the topN recommendation algorithm. Our first two techniques use the standard hybrid model by combining content-based and collaborative filtering. Our third technique integrates attributes directly into collaborative filtering, instead of incorporating attributes via a content-based submodel.

2 Related Work

There are many ways in incorporating attributes into collaborative filtering. One of the first hybrid recommender systems is Fab (Balabanovic and Shoham 1997). Its recommender engine first identifies items (pages) on a current attribute (topic). It then receives highly rated items from the user's similar neighbors and discards items that have already been seen by the user. When the user rates a new item, his/her profile will be updated and this information will be passed on to his/her neighbors. One of the simplest hybrid approaches is the linear combination of recommendation weighted average of CBF and CF predictions (Claypool *et al.* 1999). Few others attempted to use the inductive learning approach. For instance, Basu *et al.* (1998) considered recommendation as a classification problem and used using hybrid features to predict whether a user will like or dislike an item. The learning task can also be treated by developing a kernel that learns a mapping from user-item pairs to a set of ratings (Basilico and Hofmann 2004). Another method is to learn a vector of weighted attributes using the Winnow algorithm (Pazzani 1999). CF is then applied using the matrix containing the weight of each user's content-based profile, instead of using the rating matrix. Melville *et al.* (2002) followed a two-stage approach: first they applied a naïve Bayesian classifier as content-based predictor to complete the rating matrix, then they re-estimated ratings from this full rating matrix by CF. In our paper, we have selected Melville's model in its adapted form as our hybrid baseline model.

3 Hybrid Attribute-Aware CF Methods

We propose three effective Attribute-Aware collaborative filtering methods.

- Sequential CBF and CF (adapted content-boosted CF),
- Joint Weighting of CF and CBF, and
- Attribute-Aware Item-Based CF.

All three approaches recommend topN items that contain the highest frequency of their neighboring items. Similarity between two users is computed using Vector Similarity. The first two algorithms apply CBF and CF paradigms in two separate processes before combining them together at the point of prediction. Our third approach, however, does not employ CBF algorithm; instead item attributes are directly incorporated at the model-building stage.

Sequential CBF and CF Our first approach termed, "Sequential CBF and CF" is an adapted form of Melville's original hybrid model — Content-Boosted Collaborative Filtering (CBCF) (Melville *et al.* 2002). The reason why we do not use the CBCF directly is because the original model is intended for predicting ratings, whereas this paper focuses on the topN problem. Hence, the CBCF is adapted such that it will recommend N number of items to the user instead of inferring the rating of an item. This model is used as our hybrid baseline for evaluating the other two approaches. Recommendations are generated using CF.

CBCF first uses a naïve Bayesian classifier to build a content-based model for each user. Next a full matrix is formed by combining the actual ratings and the predicted ratings learned from the CBF predictor. The adaptation takes place when applying CF. Instead of finding the weighted sum of ratings of other users to compute the prediction ratings for the current user, the full matrix is sparsified by considering solely items with high ratings.

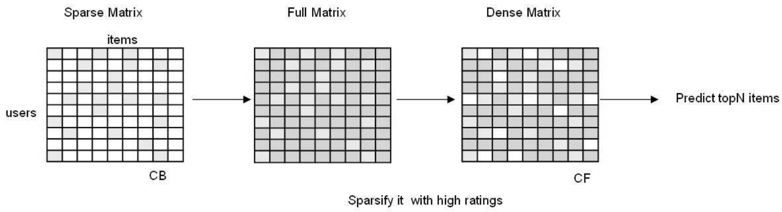

Fig. 1. CF and CBF processes done in sequence

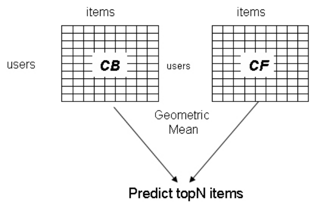

Fig. 2. CF and CBF processes done in parallel

Joint Weighting of CF and CBF Similarly, our second approach also applies both CBF and CF. Again, naïve Bayesian classifier is utilized here. However, instead of inferring the class or rating of an item based on attributes, it predicts how much a user will like the attributes. Let

- U be a set of **users**,
- I be a set of **items**,
- B be a set of (binary) **item attributes**,
- $D_{i,b} \in \{0, 1\}$ specify whether item $i \in I$ has attribute $b \in B$,
- $O_{u,i} \in \{0, 1\}$ specify whether item $i \in I$ occurred with user $u \in U$ (i.e., u has rated/bought/visited item i).

$$\hat{p}^{\mathrm{cb}}(O_{u,.} = 1 \mid D_{.,b}, b \in B) := \frac{1}{k} P(O_{u,.}) \cdot \prod_{b \in B} P(D_{.,b} \mid O_{u,.}) \quad \text{where } k := P(D_{.,b}, b \in B) \tag{1}$$

Unlike the first approach where the two processes are done sequentially – content-based first then CF, the order of these processes is unimportant for the latter and serves as the complementary view for each other. Equation 1 generates predictions using attributes (CBF) and this is joined with the outputs of CF by computing the geometric mean of the outputs. This mean combination is then used for performing the topN prediction.

$$\hat{p}(O_{u,i} = 1) \sim \hat{p}^{\mathrm{cb}}(O_{u,i} = 1)^{\lambda} \cdot p^{\mathrm{cf}}(O_{u,i} = 1)^{1-\lambda} \quad \text{with } \lambda \in [0, 1] \tag{2}$$

where λ is used to weight the content-based and collaborative methods, e.g., for $\lambda = 0$, we get pure collaborative filtering and for $\lambda = 1$, pure content-based filtering.

Attribute-Aware Item-Based CF Our third approach extends the Item-Based topN CF (Deshpande and Karypis 2004). Rather than using CBF algorithms, it exploits the content/attribute information by computing the similarities between items using attributes thereupon combining it with the similarities between items using user ratings. This is shown in Equation 3, where $\text{isim}^{\text{ratings}}$ corresponds to the item similarities computed using Vector Similarity with the ratings and $\text{isim}^{\text{attributes}}$, computed with the attributes.

$$\text{isim}^{\text{attributes}}(i,j) := \frac{\langle D_{i,.}, D_{j,.}\rangle}{||D_{i,.}||_2||D_{j,.}||_2} \tag{3}$$

$$\text{isim}^{\text{combined}} := (1-\lambda)\ \text{isim}^{\text{ratings}} + \lambda\ \text{isim}^{\text{attributes}} \quad \text{with } \lambda \in [0,1] \tag{4}$$

Again, λ is used to adjust the corresponding weight on CBF and CF. In this case, setting λ to 0 is the same as computing Pure Item-Based.

4 Evaluation and Experimental Results

In this section, we present the evaluation of our three attributes-aware recommendation algorithms and compare their performances with various non-hybrid baseline models, as well as the Sequential CBF-CF as a comparison against an existing hybrid model. The non-hybrid models we have selected are: Most Popular, Pure CF and Pure CBF. Most Popular is the most basic model that simply returns the N most-frequently rated items over all users, i.e., it is not personalized. Pure CF corresponds to the classical User-Based CF. Pure CBF uses the naïve Bayesian as predictor as shown in Equation 1 and applies it as a topN problem by returning the N items which contain the attributes the user likes most.

We evaluated the performance of our algorithms with the data obtained from MovieLens (*ml*; MovieLens 2003), which corresponds to movie ratings. The ratings are expressed on a 5-point rating scale and indicate how much a user likes a movie. Since our algorithms do not take the actual ratings into account, the ratings are treated as a binary value of whether the user has seen or not seen a movie. We have chosen the *ml* dataset containing approximately one million ratings of 3592 movies made by 6,040 users. In addition, the genres of each movie are provided. There are in total 18 different genres for the *ml* dataset. The genres of each movie, which are identical to the ones provided by the Internet Movie Database (IMDB), are selected as the content information/attributes for each item.

The datasets are split into 80% training set and 20% testing set by randomly assigning the non-zero entries of rows from the rating matrix to the testing set. The quality of these predictive models are measured by comparing the recommendations (topN set) predicted using the training data against the actual items from the testing set.

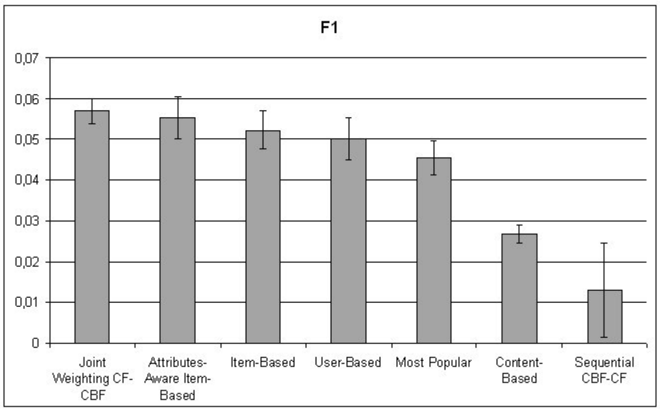

Fig. 3. F1 of different recommendation algorithms

The experiments are tested on ten random subsets of the *ml* dataset with 1000 users and 1500 items each. The results we present here are the average of the ten random trials.

Metrics

Our paper focuses on the topN problem, which is to predict a fixed number of top recommendations and not the ratings. Suitable evaluation metrics are Precision and Recall. Similar to Sarwar *et al.* (2000), our evaluations consider any item in the topN set that matches any item in the testing set as a “hit”. F1 measure is also used to combine Precision and Recall into a single metric.

$$\text{Precision} = \frac{\text{Number of hits}}{\text{Number of recommendations}}$$
$$\text{Recall} = \frac{\text{Number of hits}}{\text{Number of items in test set}}$$
$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Experiment Results

The results of the average of ten random trials are presented in Figure 3. The parameters selected for each algorithms are shown in Table I. They are selected to be optimal for our algorithms by means of grid search. Additional parameters, threshold and max, for the Sequential CBF-CF are set to 50 and 2 accordingly as chosen in the original model.

Comparing the performance achieved by our two novel hybrid algorithms, we can see that Attribute-Aware Item-Based CF and Joint Weighting CF-CBF outperform the other classical models. The results of CBF and Sequential CBF-CF models are far below the baseline Most Popular model.

Name:	Neighborhood Size	λ
joint weight CF-CBF	90	0.15
attr-item CF	400	0.05
item based	400	-
sequential cb-cf	90	-
user-based	90	-

Table 1. The parameters chosen for the respective algorithms.

Although Melville *et al.* (2002) reported that CBCF performed better than User-Based and Pure CBF for ratings, it fails to provide quality topN recommendations for items in our experiments. Thus, we focus our discussion mainly on our other two algorithms. To evaluate the immediate effect on the quality of recommendations after the incorporation of attributes, we compare the Attribute-Aware Item-Based CF and Joint Weighting CF-CBF methods with their base algorithms that do not consider attributes.

Although only 18 attributes are used, our Attribute-Aware Item-Based and Joint Weighting CF-CBF show already significantly good results. As we can see from Figure 3, the performance increases about 5.7% after introducing attributes to its based algorithm — Item-Based topN. Integrating attributes using Joint Weighting CF-CBF methods gives even better performance. As this model is derived from the CF and CBF models, it does approximately 14% better than the CF and more than 100% increases in comparison with the CBF method. Furthermore, Joint Weighting CF-CBF algorithm holds the smallest standard deviation (5.26%). This proves the results from this model to be reasonably reliable.

5 Conclusions and Future Works

The aim of this paper is to improve the quality of topN recommendations by enhancing CF techniques with content information of items. We have proposed three different hybrid algorithms. One of them, to be an adapted formed of an exiting hybrid model (Sequential CBF-CF) and two other novel hybrid models: Attribute-Aware Item-Based and Joint Weighting CF-CBF. We have shown that our two novel hybrid models give the best performance in comparison with the Most Popular, User-Based, Item-Based, Content-Based and the Sequential CBF-CF models.

Incorporating a small amount of attributes already gives reasonably significant results; we can anticipate that by providing more valuable/positive attributes, the quality of recommendations should gradually increase respectively. Experiments with more attributes as well as to test the algorithms on various larger datasets are also planned for future works.

References

AGGARWAL, C. C., WOLF, J. L., WU, K.-L. and YU, P. S. (1999): Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of ACMSIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York.

BALABANOVIC, M. and SHOHAM, Y. (1997): Fab: Content-based, collaborative recommendation. *Commun. ACM 40*, 66-72.

BASILICO, J. and HOFMANN, T. (2004): Unifying collaborative and content-based filtering. In *Proceedings of the 21 st International Conference on Machine Learning*, Banff, Canada, 2004.

BASU, C., HIRSH, H., and COHEN, W. (1998): Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the 1998 Workshop on Recommender Systems*. AAAI Press, Reston, Va. 11-15.

BILLSUS, D. and PAZZANI, M. J. (1998): Learning collaborative information filters. In *Proceedings of ICML. 46-53.*

BREESE, J. S., HECKERMAN, D. and KADIE, C. (1998): Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98).* G. F. Cooper, and S. Moral, Eds. Morgan-Kaufmann, San Francisco, Calif., 43-52.

BURKE, R. (2002): Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User Adapted Interaction*, 12/4, 331-370.

CLAYPOOL, M., GOKHALE, A. and MIRANDA T. (1999): Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation.*

DESHPANDE, M. and KARYPIS, G. (2004): Item-based top-N recommendation algorithms, *ACM Transactions on Information Systems 22/1*, 143-177.

GOLDBERG, D., NICHOLS, D., OKI, B. M. and TERRY, D. (1992): Using collaborative filtering to weave an information tapestry. *Commun. ACM 35*, 61-70.

MELVILLE, P., MOONEY, R. J. and NAGARAJAN, R. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence* (AAAI-2002), pp. 187-192, Edmonton, Canada, July 2002.

MITCHELL, T. (1997): Machine Learning. New York, NY: McGraw-Hill.

MOVIELENS (2003): Available at http://www.grouplens.org/data.

PAZZANI, M. J.(1999): A framework for collaborative, content-based and demographic filtering.*Artificial Intelligence Review 13(5-6):393408.*

RESNICK, P., IACOVOU,N., SUCHAK, M., BERGSTROM, P. and RIEDL, J. (1994): GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 Conference on Computer Supported Collaborative Work.* R. Furuta and C. Neuwirth, Eds. ACM, New York. 175-186.

SARWAR, B. M., KARYPIS, G., KONSTAN, J. A. and RIEDL, J. (2000): Analysis of recommendation algorithms for E-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC'00).* ACM, New York. 285-295.

ZIEGLER, C., SCHMIDT-THIEME, L., LAUSEN, G. (2004): Exploiting Semantic Product Descriptions for Recommender Systems, *Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop (SWIR '04)*, July 25-29, 2004, Sheffield, UK.