Cross-Domain Mediation in Collaborative Filtering

Shlomo Berkovsky[1], Tsvi Kuflik[1], and Francesco Ricci[2]

[1] University of Haifa, Haifa
slavax@cs.haifa.ac.il, tsvikak@is.haifa.ac.il
[2] Free University of Bozen-Bolzano, Italy
fricci@unibz.it

Abstract. One of the main problems of collaborative filtering recommenders is the sparsity of the ratings in the users-items matrix, and its negative effect on the prediction accuracy. This paper addresses this issue applying cross-domain mediation of collaborative user models, i.e., importing and aggregating vectors of users' ratings stored by collaborative systems operating in different application domains. The paper presents several mediation approaches and initial experimental evaluation demonstrating that the mediation can improve the accuracy of the generated predictions.

1 Introduction

Nowadays, the overwhelming amounts of information raise a need for intelligent systems providing personalized services tailored to users' needs and interests, represented by their User Models (UMs). Collaborative Filtering (CF) 2 is one of the most popular and widely-used personalization techniques, generating personalized predictions in recommender systems. CF assumes that people with similar tastes, i.e., people who agreed in the past, will also agree in the future. Hence, CF predictions are generated by aggregating the opinions of people with similar tastes.

The input for the CF algorithm is a matrix of users' ratings on items, referred to as the ratings matrix. The CF algorithm is typically decomposed into three stages: (1) similarity computation: weighting all the users with respect to their similarity with the active user, (2) neighborhood formation: selecting *K* most similar users, i.e., nearest-neighbors for the prediction generation, and (3) prediction generation: computing the prediction by weighting the ratings of the neighbor users on the target item 2.

CF recommender systems suffer from the *new item* and the *new user* bootstrapping problems. The new item problem refers to the fact that if the number of users that rated an item is small, accurate predictions for this item cannot be generated. The new user problem refers to the fact that if the number of items rated by a user is small, it is unlikely that there is an overlap of products rated by this user and other users. Hence, users' similarity cannot be reliably computed and accurate predictions for the user cannot be generated. These problems are referred to as particular cases of a CF *sparsity* problem, where the contents of the ratings matrix are insufficient for generating accurate predictions. To overcome the sparsity, 1 proposed to enrich the UMs of the target recommender system by a mediation (i.e., import and aggregation) of user

C. Conati, K. McCoy, and G. Paliouras (Eds.): UM 2007, LNAI 4511, pp. 365–369, 2007.

modeling data from other recommender systems. Mediation enriches the UMs available to the target system and upgrades the accuracy of the generated predictions.

This paper focuses on cross-domain mediation of UMs in CF, which is one of the mediation modes discussed in 1. In cross-domain mediation, the user modeling data is imported from remote systems exploiting the same CF recommendation technique as the target system, in other application domains. Hence, both target and remote systems represent the UMs as a list of ratings provided by a user on the domain items. In this setting, four types of user modeling data can be imported: (1) UMs stored by the remote system, (2) lists of the neighborhood candidates, (3) degrees of similarity between the active user and the other users, computed over the data stored by the remote system, and (4) complete predictions generated by the remote system. This paper elaborates on the last type of cross-domain mediation in CF and presents its implementation and evaluation using the EachMovie dataset 3. Experimental results demonstrate that importing external user modeling data allows achieving higher accuracy of the predictions.

2 Cross-Domain Mediation in Collaborative Filtering

Traditional CF recommender systems store the ratings in a two-dimensional matrix (or map) *M*:*(user*$_{id}$, *item*$_{id}$*)*→*rating*, where *user*$_{id}$ and *item*$_{id}$ represent the unique identifiers of users and items and *rating* represents the explicit evaluation given by a user *user*$_{id}$ on an item *item*$_{id}$. Note that the number of items typically managed by the system is significantly larger than the number of ratings provided by an average user. This leads to a very sparse ratings matrix *M* and to the sparsity problem of CF.

Conversely, in a domain-distributed setting, the ratings matrix *M* is split, i.e., every domain *d* stores a local ratings matrix M_d. The structure of M_d is similar to the structure of *M*, i.e., it is a two-dimensional matrix of ratings given by a set of users on a set of items. However, this set of items in M_d is restricted to items that belong to a certain application domain *d*, i.e., M_d: *(user*$_{id}$, *item*$_{id}$*)*→*rating,* such that $item_{id} \in d$. Hence, this setting can be considered as a vertical partitioning of the ratings matrix *M* (figure 1).

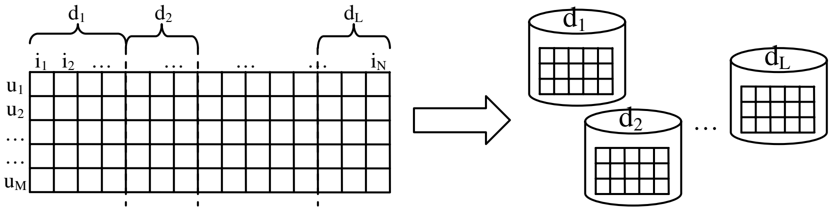

Fig. 1. Domain-related vertical partitioning of the ratings matrix

Note that this is not exactly vertical partitioning of the ratings matrix. In a real vertical partitioning, the partitioned sets of items are disjoint, i.e., every item belongs to a single group of items. In domain-related vertical partitioning, certain items may belong to multiple domains or categories. This setting is not uncommon if the above representation of domains is downscaled to the representation of E-Commerce services. In this case, ambiguous categorization of items may be explained by different classifications of products, their providers, or E-Commerce sites.

Similarly to a centralized CF recommender system, a typical scenario is initiated by a recommendation request issued by a user $user_{id}$ to a CF recommender system R_t in the target application domain t. The target system R_t selects a set of items $\{item_{id}\}$ that can be recommended and initiates a prediction generation process for every $item_{id}$. To enhance the accuracy of the predictions, R_t requests relevant user modeling data from a set of remote CF recommender systems $\{R_d\}$, operating on domains d. The query is formulated as a triple q=<$user_{id}$, $item_{id}$, t>. In the following discussion, let us assume that the identities of the users and items are unique in all the domains.

According to the first mediation approach, the UMs (i.e., the rating vectors), stored by a remote system R_d, operating in another domain d, are imported. For the sake of simplicity, let us assume that R_d responds to q by sending to R_t the content of the local repository of UMs, i.e., $resp_d=M_d$, where M_d is local ratings matrix containing only the items that belong to domain d. Upon receiving the set of responses $\{resp_d\}$, R_t constructs the unifying ratings matrix M by integrating local and imported data. Over M, traditional CF mechanism is applied. Since the reconstructed matrix M can be considered as the traditional centralized CF matrix, this approach is referred to as ***Standard*** CF and serves as a baseline for the experimental comparisons.

The second mediation approach is called ***Heuristic*** and it imports into the target system a list of nearest-neighbors computed by the remote systems R_d. It relies on a heuristic assumption that similarity of users spans across multiple application domains. Hence, if two users are similar in a certain remote application domain d, they may be also similar in the target domain t. Practically, this means that R_d responds to q by sending to R_t the set of K identities of the users most similar to the active user, i.e., $resp_d=\{user_{id}\}$. Upon receiving the set of responses $\{resp_d\}$, R_t aggregates these sets of nearest-neighbors into the overall set of (heuristic) candidates for being the nearest-neighbors, computes their true similarity values according to the local ratings matrix M_t, selects the set of K nearest-neighbors, and generates the predictions.

The third approach is called ***Cross-domain*** mediation. Here, to compute the overall similarity between users, the target system imports domain-dependent similarity values and aggregates them into an overall similarity value. Upon receiving the request q, every remote system R_d computes locally, i.e., according to the contents of the local ratings matrix M_d, the similarity between the active user and the other users in M_d. A set of K nearest-neighbors is selected, and their $user_{id}$ together with their similarity values are sent to R_t. In other words, $resp_d=\{(user_{id}, sim_d)\}$, where $sim_d=sim(user_{id}, user_{act})$ is the local similarity between a user $user_{id}$ and the active user $user_{act}$, computed using their ratings in the application domain d using a certain similarity metric sim. Upon receiving the set of responses $\{resp_d\}$, R_t aggregates the domain-related similarity values into the overall similarity metric using inter-domain correlation values. As the overall similarity is computed, the K nearest-neighbors are selected and the predictions are generated.

The fourth mediation approach deals with complete CF predictions generated locally by the system R_t from the target domain t and is referred to as ***Local***. According to it, the predictions are generated using only the data stored in the ratings matrix M_t of the target system. This is done similarly to the centralized CF, but using a restricted set of ratings on items from t: local similarity values are computed, the set of

K nearest-neighbors is selected and the predictions are generated. However, *Local* CF disregards the fact that the items may belong to several application domains and treats each domain independently. Hence, according to ***Remote-Average*** variant of *Local* CF, every remote system R_d from another application domain *d*, to which the predicted item belongs, generates a local prediction using the ratings stored in its ratings matrix M_d. The computed predictions are sent to R_t, i.e., $resp_d = pred_d$. Upon receiving the set of responses $\{resp_d\}$ and generating a local prediction using its matrix M_t, R_t aggregates the predictions into a single value by averaging the set of all local predictions.

3 Experimental Evaluation and Conclusions

Experimental evaluation of the proposed mediation approaches involved EachMovie dataset of movie ratings 3. To mimic domain-related vertical partitioning of the ratings matrix, the movies were partitioned according to their genres. Eight genre-related ratings matrices were created: *action*, *animation*, *comedy*, *drama*, *family*, *horror*, *romance*, and *thriller*. In EachMovie, the movies usually belong to multiple (up to *4*) genres. Each movie belongs, on average, to *2.376* genres. Hence the sets of movies in the genre-related matrices were not disjoint. Table 1 summarizes the distribution of movies and ratings among genre-related ratings matrices and sparsity of each matrix.

Table 1. Data Distribution among Genres-Related Matrices

	action	animation	comedy	drama	family	horror	romance	thriller
num. of movies	198	43	400	536	145	87	137	177
num. of ratings	1,166,032	192,769	2,209,218	3,056,203	800,118	432,568	681,409	991,083
sparsity (%)	91.923	93.852	92.425	92.180	92.432	93.181	93.179	92.321

Local and *Remote-Average* CF approaches discussed in previous section were implemented and evaluated. Cosine Similarity was selected as the users' similarity metric, and the minimal number of movies rated by users for the similarity computation was *6* (predictions could not be generated for users that rated below *6* movies). The number of nearest-neighbors used for the prediction generation was *20*.

The experiment evaluated the effect of sparsity of the target user ratings on the accuracy of the predictions. Hence, the users were partitioned to *12* categories, according to the percentage of the rated movies in the target genre: below *3%*, *3%* to *6%*, …, *30%* to *33%*, and over *33%*. For every group, *1,000* predictions were generated for various combinations of user, movie, and target genre. The predictions' accuracy was measured using the MAE metric 2. The baseline for the comparisons is *Standard* CF, as its results are similar to the results that would have been obtained in traditional centralized CF.

The results show that both *Local* and *Remote-Average* CF outperform *Standard* CF for any percentage of rated movies (statistically significant, *p=2.78E-07* and *p=1.63E-06*, respectively). It can be explained by arguing that the similarity computation over the ratings from the target genre only in *Local* CF (or over the ratings from other movie genres in *Remote-Average*) yields more accurate similarity values than the similarity computation over all the available ratings. This explained by the observation that the ratings from these genres are important for computing the similarity value in the relevant genre, whereas the other ratings may insert noise into the computation. As a result, the predictions are more accurate.

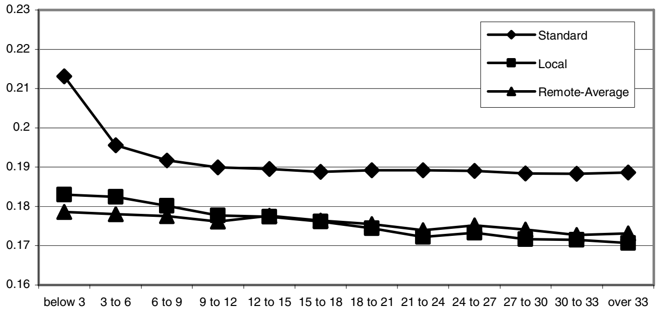

Fig. 2. Local, Remote-Average and Standard CF Approaches

Comparing *Local* and *Remote-Average* CF approaches shows that for a small percentage of rated movies, i.e., sparse ratings matrix, *Remote-Average* CF is slightly more accurate (statistically insignificant). It can be explained by the fact that the predictions are generated using additional knowledge acquired by importing data from other relevant genres and not using the data from the target genre only. For a higher percentage of rated movies, the local data is sufficient and the imported data hampers the accuracy of the predictions.

It should be stressed that in certain conditions *Local* and *Remote-Average* CF approaches are inapplicable. For example, for the group of users that rated less than *3%* of the movies, predictions can be generated only for comedies and dramas, as only in these cases *3%* of the movies is greater than *6*, a minimal number of movies for the similarity computation. Hence, although the accuracy of *Local* and *Remote-Average* CF is higher, they cannot generate predictions for certain movies that will negatively effect on the ability of the system to recommend all the interesting movies.

In summary, the evaluation showed that importing user profile data from other domains yields more accurate predictions. However, this is not applicable for sparse data and aggregating local degrees of similarity (i.e., *Cross-Domain* CF approach) is supposedly a more appropriate solution. In the future, it is planned to implement and evaluate the rest of the proposed cross-domain CF approaches.

References

1. S.Berkovsky, “*Decentralized Mediation of User Models for a Better Personalization*”, in proc. of the AH Conference, 2006.
2. J.L.Herlocker, J.A.Konstan, A.Borchers, J.Riedl, “*An Algorithmic Framework for Performing Collaborative Filtering*”, in proc. of the SIGIR Conference, 1999.
3. P. McJones, “*EachMovie Collaborative Filtering Data Set*”, HP Research, 1997.