Distributed Collaborative Filtering with Domain Specialization

Shlomo Berkvosky
Caesarea-Rothschild Institute
University of Haifa, Israel
slavax@cs.haifa.ac.il

Tsvi Kuflik
Management Information Systems Department
University of Haifa, Israel
tsvikak@is.haifa.ac.il

Francesco Ricci
Faculty of Computer Science
Free Univ. of Bozen-Bolzano, Italy
fricci@unibz.it

ABSTRACT
User data scarcity has always been indicated among the major problems of collaborative filtering recommender systems. That is, if two users do not share sufficiently large set of items for whom their ratings are known, then the user-to-user similarity computation is not reliable and a rating prediction for one user can not be based on the ratings of the other. This paper shows that this problem can be solved, and that the accuracy of collaborative recommendations can be improved by: a) partitioning the collaborative user data into specialized and distributed repositories, and b) aggregating information coming from these repositories. This paper explores a content-dependent partitioning of collaborative movie ratings, where the ratings are partitioned according to the genre of the movie and presents an evaluation of four aggregation approaches. The evaluation demonstrates that the aggregation improves the accuracy of a centralized system containing the same ratings and proves the feasibility and advantages of a distributed collaborative filtering scenario.

Categories and Subject Descriptors
H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *distributed systems, user profile and alert services.*

General Terms
Algorithms, Measurement, Performance, Experimentation

Keywords
Distributed Collaborative Filtering, Recommender Systems, Mediation of User Modeling Data.

1. INTRODUCTION
E-Commerce Web-sites offer today a large quantity of items with different characteristics and types. Hence, user's searches often bring a potentially overwhelming set of items and options which lead to the “information overload" problem. User can be overwhelmed by the quantity of the information displayed and, without some support, the process of filtering out irrelevant items and finally select the most appropriate one could be very difficult, if not impossible [20].

RecSys'07, October 19–20, 2007, Minneapolis, Minnesota, USA.

Recommender systems [1], [7] are aimed at addressing this problem, suggesting to the users those items which best suit their needs and preferences in a particular situation and context. Collaborative Filtering (CF) [9] is one of the most popular and widely-applied recommendation techniques, generating personalized recommendations for rating predictions. CF assumes that people with similar tastes, i.e., people who agreed in the past, will also agree in the future. Hence, CF predictions, i.e., ratings on items not yet evaluated by the user, are generated by averaging the opinions of people with similar tastes.

The input for the CF prediction generation algorithm is a matrix of users' ratings on items, referred to as the *ratings matrix*. CF algorithm is typically decomposed into three generic stages [9]:

(1) **similarity computation:** assessing the similarity of all the users to the active user, i.e., the user for whom a recommendation is searched,

(2) **neighborhood formation:** selecting the *K* most similar users to the active user,

(3) **prediction generation:** computing the active user rating prediction. This is done for a target item whose rating is unknown, and is obtained by weighting the ratings of the *K* most similar users, found at (2) on the target item according to the user-to-user similarity computed at (1).

In a CF system the items that are finally recommended to the active user are typically those having maximal rating prediction. In some cases all the items are suggested, but in this case they are ranked according to the predicted ratings.

CF recommender systems basically suffer from the scarcity of ratings. This problem is referred to as *data sparsity* since the ratings matrix is sparse and only a small fraction of all possible user-item entries is known [14]. In fact, on one hand ratings are required to compute accurate and reliable user-to-user similarity. This similarity is computed with various correlation functions, e.g., Pearson correlation [17], that estimates the correlation of the ratings vectors of two users, i.e., the rows of the ratings matrix. The ratings of the users can be considered their User Models (UMs), as they describe the preferences of the users. On the other hand, these ratings are used to predict the rating that another user would assign to the same item and therefore are important for the recommendation process. Two instances of the general rating scarcity problem are the *new item* and *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 could be an overlap of items rated by this user and the active user. Hence, user-to-user similarity cannot be reliably computed and accurate predictions for the active user cannot be generated.

To overcome the sparsity problem a number of solutions has been suggested [16],[22]. In a previous work we proposed to enrich the UMs, i.e., the ratings vectors of a target recommender system by *mediation* (i.e., import and aggregation) of user modeling data collected by other recommender systems [2]. The mediation enriches the UMs available to the target system and therefore improves the prediction accuracy. For example, consider a movie recommender system not having a sufficient user modeling data about a user. Let us assume that three additional recommenders on TV programs, books and CDs could provide additional user modeling data, i.e., ratings on TV programs, books and CDs. If this data is available to the movie recommender, it can more reliably compute the user-to-user similarity on a larger set of ratings, and derive more accurate recommendations.

This paper focuses on a radically different approach. Instead of increasing the size of the UMs, we propose here to partition the full ratings matrix into smaller matrices. Each new smaller matrix contains the ratings of all the users on the items belonging to a certain topic or domain, e.g., the movies having a particular genre. These matrices are supposed to mimic the user modeling data repositories of CF-based recommender systems having a focused knowledge on a specific domain. Moreover, we assume that these systems can communicate each other using a simple request-response protocol where the target system, i.e., the system that must provide a recommendation for an item in its own domain can ask the *remote* systems are competent in other domains, to provide various types of data that are then exploited for computing the rating prediction and the recommendations.

We stress that our model assumes that user identity is shared, such that if, for instance, the target system requests user modeling data for a user identified by ID_x, the remote systems can identify this user in their rating matrixes and provide relevant information. Moreover, we also assume that each item has a unique identifier and that one item can belong to multiple systems, e.g., the movie "Polar Express" can belong to the systems specialized in cartoons and in fantasy movies at the same time.

In this setting, a request issued by the target system, can specify the user and/or the item for which a prediction has to be made. And, given a request, this paper elaborates on four types of user modeling data that can be sent back by the remote systems:

(1) UMs stored by the remote system,
(2) lists of the neighborhood candidates computed by the remote system,
(3) degrees of similarity between the active user and the other users, computed over the data stored by the remote system,
(4) complete predictions generated by the remote system.

In this paper we present the distributed model of domain specialized and cooperating recommender systems and we describe the implementation and the experimental evaluation of that model using the EachMovie dataset [14]. This paper extends the results presented in [5], where we experimented only one kind of response, listed at point (4). We note here that this distributed scenario of local and remote systems is not a strict requirement. The techniques here described improve the accuracy of the prediction and can be implemented even when the data are stored in a single central repository. We stress the concept of a distributed scenario because we believe this situation will be more and more common in the future of the Web.

To prove the feasibility of this domain-specialization in CF, the experimental evaluation here described, compares its prediction accuracy with the accuracy of the traditional centralized approach. The results show that when the user modeling data are distributed among multiple domain-specialized repositories, mediating user modeling data and recommendations from these repositories can improve the accuracy of the generated predictions compared with the original centralized recommender system. Moreover, in this paper we shall draw some conclusions regarding the conditions under which the proposed mediation approaches improve the accuracy of CF predictions.

In conclusion the main contributions of this paper are:

- A distributed model of CF domain-specialized recommender systems that communicate in a cooperative way with a simple request-response protocol,
- A method for exploiting the additional knowledge provided by the classification of an item into a domain and the proof that it can increase the accuracy of CF,
- The demonstration that averaging predictions made by domain specialized systems can improve prediction accuracy,
- The validation that accuracy of CF can be improved by basing the user-to-user similarity assessment only on the items belonging to the same domain (or similar domains) of the item whose rating has to be predicted.

The paper is organized as follows: Section 2 presents prior works on decentralized CF, Section 3 discusses our domain-specialization mediation approaches, Section 4 presents the experimental results, and Section 5 concludes and presents future research directions.

2. DECENTRALIZED COLLABORATIVE FILTERING

Most CF recommender systems are centralized, i.e., the storage of the UMs in the ratings matrix and the generation of the predictions are performed by a single standalone component. Due to the commercial nature of nowadays online services, these systems typically neither cooperate nor share their UMs. However, since the accuracy of the predictions depends on the richness of the UMs, CF systems may benefit from importing and aggregating user modeling data from multiple sources [2],[16].

An initial evaluation of CF over a set of distributed UMs partitioned among several repositories was presented in [3]. However, only an import of potentially similar users was implemented and evaluated, whereas the partitioning of items among the repositories was done randomly, such that there was no way to identify correlations between groups of items.

In [22], the authors discuss a multi-agent architecture, where recommending agent could generate predictions by aggregating community-based information imported from multiple remote profiling agents, managing browsing logs of pages stored by a Web-server. However, the pages stored by the servers were treated as unrelated items, such that the correlations between the groups of pages stored by various Web-servers were neglected.

A similar approach for aggregating community-based information in community-based Web search was proposed in [8]. There, the users were grouped into communities of like-minded searchers, such that past search histories of the users were aggregated across various communities. Although the similarities of the communities were considered for the collaborative search results, the adaptation was done at the level of the communities and not of the individual users.

Another example of exploiting cross-domain dependencies is item-to-item CF [19]. That approach generated predictions basing on the similarity of items, rather than of users, but still used the ratings as an input for the similarity computation. While the accuracy of item-to-item CF outperformed the accuracy of traditional CF, it still could not generate accurate predictions for a sparse ratings matrix. Moreover, the similarity between items from different domains could be computed only if a non-empty set of users rated the items, i.e., if the domains had a non-empty set of common users.

Unlike the above approaches, this work aims at developing, evaluating and analyzing various ways for importing and aggregating multiple domain-related CF user modeling data while taking into account the relationships (e.g., correlation) between the domains.

3. DOMAIN-BASED RATINGS PARTITIONING

Traditional CF recommender systems store the ratings in a two-dimensional matrix $\boldsymbol{V}=(v_{ij})$, where $i=1, \ldots n$ and $j=1, \ldots m$. Here v_{ij} represents the rating assigned by user i to item j, and v_{ij} is either a numeric value, for instance an element of the set $\{1, 2, 3, 4, 5\}$ or a special value '?' denoting that the rating is unknown. Note that m, the number of items managed by the system, is typically significantly larger than the number of ratings provided by an average user, i.e., the majority of the ratings is unknown. This leads to a very sparse ratings matrix and to the sparsity problem of CF recommender systems [14].

Conversely, in a domain-distributed setting, the ratings matrix $\boldsymbol{V}=(v_{ij})$ is stored in a semi-centralized way. In this case, every domain d stores a subset of the ratings, here denoted with a matrix $\boldsymbol{V_d}$. The structure of $\boldsymbol{V_d}$ is similar to the structure of $\boldsymbol{V}$, i.e., it is a two-dimensional matrix representing the ratings given by a set of users on a set of items. However, the set of items in the matrix is restricted to the items that belong to a certain topic or domain d. Hence, $\boldsymbol{V_d}=(v_{ij})$, $i=1,\ldots,n$, and $j \in J_d$, where J_d is the subset of indexes of items belonging to domain or topic d. This setting can be considered as a vertical partitioning of the ratings matrix M (Figure 1).

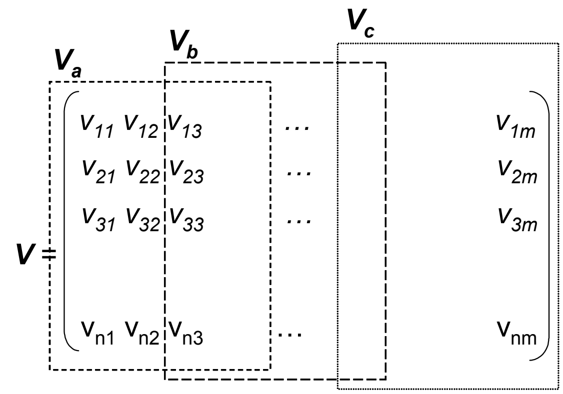

Figure 1. Domain-related partitioning of the ratings matrix

In the above figure we assume to have three domains: a, b and c, such that the ratings are partitioned into three subsets. Note that this is not exactly vertical partitioning of the ratings matrix as an item can belong to many domains. 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 items, their providers, or E-Commerce sites.

Let now assume that for every domain d, the recommendations are built using the ratings in the matrix $\boldsymbol{V_d}$. And let us further assume that when a rating prediction for an item belonging to d, the target system that manages $\boldsymbol{V_d}$ could benefit by contacting remote systems that manage other domains/topics, e.g., d' and d''. In this scenario various types of data can be exchanged. These types of data are those exploited in the three stages of CF predictions generation: similarity computation, neighbor formation, and rating prediction. For the similarity computation, the UMs, i.e., the ratings of a user on items in the remote domains can be imported from the remote systems. For the neighborhood formation, either the list of candidates for being the nearest-neighbors or users' similarities computed by the remote systems could be imported. Finally, for the prediction generation, complete predictions for items, generated by the remote systems, could be imported.

3.1 Importing User Modeling Data in CF

A typical recommendation scenario is initiated by a recommendation request issued by a user i to a CF recommender system R_t in the target application domain t, with ratings knowledge contained in the ratings from the matrix $\boldsymbol{V_t}$. As a result, the target system R_t selects a set of items that can be potentially recommended $Items_t$ and initiates a prediction generation process for every item $j \in Items_t$. To enhance the accuracy of the predictions, R_t queries a set of available remote CF recommender systems $\{R_d\}_{d \in D}$ from other **closely-related** domains d for related user modeling data. Note that the relations between target domain t and domains in D will be discussed later. The query is formulated as a triplet $q=\langle i, j, t\rangle$, where i is the identifier of the active user, i.e., a user for whom a recommendation is generated, j is the target item identifier, i.e., an item, possibly null, for which a rating prediction is computed, and t is the target domain. In the rest of this paper we shall consider situations and scenarios, where the remote systems return combinations of the following user modeling data:

- **Centralized Prediction:** All the ratings managed by $\{R_d\}_{d\in D}$ that are contained in V_d, $d\in D$. In this case we shall also assume that all the domains are related, i.e., D is the full set of domains.
- **Distributed Peer Identification:** The identifiers of some users that the remote systems $\{R_d\}_{d\in D}$ consider as "similar" to the target user i.
- **Distributed Neighborhood Formation:** The identifiers of some users that the remote systems $\{R_d\}_{d\in D}$ consider as "similar" to the target user i, together with their similarities to the target user i. Note that these similarities are computed by the remote system using the ratings in V_d, $d\in D$.
- **Local Prediction:** the remote systems $\{R_d\}_{d\in D}$ do not return any data.
- **Distributed Prediction:** The rating predictions for item j computed by the remote systems $\{R_d\}_{d\in D}$ using the ratings contained in V_d, $d\in D$.

We shall now describe these five scenarios in turn.

In the first scenario, the UMs themselves (i.e., the rating vectors) stored by a remote system R_d operating on another domain d, are sent to the target system, i.e., the matrix V_d itself. Upon receiving the set of responses V_d , $d\in D$, R_t constructs the global ratings matrix V by aggregating local and imported data. Over V the standard mechanism, with similarity computation, K nearest-neighbors selection and predictions generation is applied. Since the constructed matrix V can be considered as the standard centralized CF matrix, this approach is referred to as **Centralized Prediction** and it serves as a baseline for the experimental comparisons.

In the second and third scenarios (**Distributed Peer Identification** and **Distributed Neighborhood Formation**) the remote systems respond with nearest-neighbors data computed by the remote systems R_d, only using the ratings in V_d.

In **Distributed Peer Identification** we assume that the user-to-user similarity is somewhat uniform across multiple domains. Hence, one can conjecture that if two users are similar in a remote system focused on some domain d similar to the target domain t, these users may be also similar in t. Practically, this means that R_d responds to the query q by sending to R_t the set of K identities I_d of the users most similar to the active user i. Upon receiving the set of responses I_d, R_t aggregates these sets of nearest-neighbors into the overall set of nearest-neighbors candidates, computes their similarity values according to the local ratings matrix V_t, selects the set of K nearest-neighbors, and generates the predictions.

In **Distributed Neighborhood Formation** approach the user-to-user similarity, used in the target system R_t to compute the neighbors and estimate the rating prediction, is computed as an average of the similarity values computed by the remote systems $\{R_d\}_{d\in D}$. Upon receiving the request q, every remote system R_d computes the similarity between the active user i and the other users using the ratings in V_d. A set of K nearest-neighbors is selected, and their identifiers, I_d, together with their similarity values, $sim_d(i,l)$, $l\in I_d$, are sent to R_t. Upon receiving this response, R_t averages the domain-specific similarity values into the overall similarity metric using inter-domain correlation values (examples of these correlation measures will be shown later). The overall similarity is given by:

$$sim(i,l) = \frac{\sum_{d\in D} cor(d,t) sim_d(i,l)}{\sum_{d\in D} cor(d,t)}$$

where $sim_d(i, l)$ is the local similarity value in the application domain d, between the target user i and the user l, and $cor(d,t)$ is the correlation of the target domain t and remote domain d. When the overall similarity value is computed, K nearest-neighbors can be selected and the predictions are generated.

The fourth scenario deals with CF prediction generated locally by the target system, and for this reason it is referred to as **Local Prediction**. According to it, the predictions are generated using only the data stored in the ratings matrix V_t of the target system. This is done similarly to *Centralized Prediction*, 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 Prediction* disregards the fact that the items typically belong to multiple topics or domains and treats each domain independently. Hence, we have introduced a variant of this approach, which is called **Distributed Prediction.** Here, given an item j whose rating must be predicted, every remote system R_d, $d\in D$, specialized on another topic or domain d to which the target item j belongs, generates a separate local prediction using the ratings stored in its ratings matrix V_d. The computed predictions are sent to R_t. Upon receiving the set of predictions, R_t aggregates all the predictions into a single value by **averaging** the set of remote predictions. In the experimental evaluation, *Distributed Prediction* is compared with the *Local Prediction*, which exploits only the ratings in the target domain t, i.e., generates ratings predictions using the data stored in its target domain t ratings matrix V_t only.

3.2 Computing Inter-Domain Correlations

As we already mentioned, the *Distributed Neighborhood Formation* method computes the user-to-user similarity as a weighted average of the similarities computed by remote systems. Here the weights are aimed at taking into account the closeness of the target topic/domain t, with respect to the remote topics/domains d. Hence this weights represent inter-domains correlation metric $cor(d_1,d_2)$ between two application domains d_1 and d_2. This sub-section discusses two alternative correlation computation techniques: **content-based** and **ratings-based**. Assuming stable contents of the domains and stable ratings on the domain items, the inter-domains correlation computation can be considered as a one time pre-processing process that can be conducted offline.

The **content-based** correlation computation technique assumes that the textual descriptions of the items belonging to a domain can be considered as reliable source of knowledge about the characteristics of the domain [4]. Hence, the similarity of two application domains d_1 and d_2 is computed as a three-stage process: (1) mining the textual descriptions of the items in the domains from external data sources, such as the Web or other specialized databases, (2) representing the mined textual contents as feature vectors v_1 and v_2, where $v_i=(w_{i1}, \ldots, w_{in})$ and w_{ij} is the tf-idf weight [18] of the term j appearing in the domain i, and (3) computing inter-domain correlation as the cosine similarity of their respective feature vectors:

$$cor_{content}(d_1,d_2) = sim(v_1,v_2) = \frac{v_1 \cdot v_2}{\| v_1 \| * \| v_2 \|}$$

where $\cdot$ denotes the inner product between two feature vectors, and $||v_i||$ denotes the norm of a vector, i.e., the square root of the inner product of a vector with itself. The result of this computation is a scalar, reflecting the correlation of two domains on the base of their textual contents.

Alternatively, the **ratings-based** correlation is based on the correlations of ratings on the items in the domains [6], assuming that the domains share a non-empty set of common users. Given two items, j and k, their ratings-based correlation $sim(j,k)$ can be computed as the correlation, e.g., cosine similarity, of their respective ratings vectors [18]. Using item-based similarity metric, inter-domain correlation is computed as the average similarity of all the possible pairs of different items that belong to these application domains:

$$cor_{ratings}(d_1,d_2) = AVG\{sim(j,k) : j \neq k, j \in J_{d_1}, k \in J_{d_2}\}$$

where $sim(j,k)$ is the ratings-based similarity of two items, and J_d is the set of item indexes for items in domain d. Also the result of this computation is a scalar, reflecting the correlation of two domains, but in this case it is computed on the base of the similarity of ratings on items in the domains.

4. EXPERIMENTAL EVALUATION

One of the main difficulties in evaluating cross-domain mediation and collaborative recommendations using multiple user modeling data is the lack of publicly available data, representing the ratings of the same users on items classified in multiple domains or topics. Although there exist several datasets from various domains (e.g., movies, books, jokes, browsing logs), none of them are cross-linked, i.e., they do not allow identifying their users in other datasets. Moreover one of our proposed techniques (*Distributed Prediction*) requires that an item simultaneously belong to more than one topic/domain.

Hence, experimental evaluation of the proposed approaches involved EachMovie dataset of movie ratings [14], where the items were classified into domains. Although the items of EachMovie belong to a single domain of movies, domain classification was achieved by classifying the movies 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 such that each movie belongs, on average, to *1.366* genres. Hence, in these experiments the sets of movies in genre-related matrices were not disjoint, and the *Distributed Prediction* technique could be applied.

Table 1 summarizes the distribution of movies and ratings among genre-related ratings matrices and the sparsity of each matrix. The sign *K* in the number of ratings row denotes one thousand ratings.

Table 1. Data Distribution Statistics

	action	animat.	comedy	drama	family	horror	romance	Thriller
num. Movies	*198*	*43*	*400*	*536*	*145*	*87*	*137*	*177*
num. ratings	*1,166K*	*193K*	*2,209K*	*3,056K*	*800K*	*433K*	*681K*	*991K*
sparsity (%)	*91.923*	*93.852*	*92.425*	*92.180*	*92.432*	*93.181*	*93.179*	*92.321*

To compute inter-domain correlations, both content-based and ratings-based techniques were applied. Content-based technique exploited the lists of movie keywords mined from the IBDb [11] to generate genre-related tf-idf feature vectors and compute inter-genre correlations [18]. Ratings-based technique exploited the ratings on the movies in EachMovie. Tables 2 and 3 show the matrices of inter-genre correlation. Table 2 stands for the content-based and Table 3 for the ratings-based technique.

Both techniques produced symmetric matrices, i.e., $cor(d_1,d_2)=cor(d_2,d_1)$. The diagonal values in content-based matrix are *1*. This is explained by the fact that the correlation of a feature vector and itself is *1*. Also other inter-genre correlations are relatively high, above *0.73*. Conversely, in ratings-based matrix, the correlation values are lower, since they are computed using the ratings vectors of the movies, which are typically different even within the same genre. Nonetheless, for many genres the diagonal values, i.e., the inter-domain correlation is higher than the correlation with other genres (same row or column).

The CF approaches discussed in previous section were implemented and evaluated. Cosine similarity was selected as the users' similarity metric [18][1]. The minimal number of movies rated by users that we required in order to compute inter-users similarity was *6* (predictions could not be generated for users that rated less than *6* movies). The number of nearest-neighbors returned by remote domains to the target domain in *Distributed Peer Identification* and *Distributed Neighborhood Formation* was *20*. The number of nearest-neighbors used for the prediction generation was *20*.

Table 2. Inter-genre correlations – content-based

	action	animat.	comedy	drama	family	horror	romance	thriller
action	*1.000*	*0.860*	*0.935*	*0.932*	*0.820*	*0.902*	*0.913*	*0.943*
animat.	*0.860*	*1.0000*	*0.913*	*0.848*	*0.914*	*0.765*	*0.838*	*0.787*
comedy	*0.935*	*0.913*	*1.000*	*0.965*	*0.905*	*0.868*	*0.957*	*0.903*
drama	*0.932*	*0.848*	*0.965*	*1.000*	*0.841*	*0.873*	*0.987*	*0.938*
family	*0.820*	*0.914*	*0.905*	*0.841*	*1.000*	*0.739*	*0.832*	*0.772*
horror	*0.902*	*0.765*	*0.868*	*0.873*	*0.739*	*1.000*	*0.850*	*0.939*
romance	*0.913*	*0.838*	*0.957*	*0.987*	*0.832*	*0.850*	*1.000*	*0.913*
thriller	*0.943*	*0.787*	*0.903*	*0.938*	*0.772*	*0.939*	*0.913*	*1.000*

Table 3. Inter-genre correlations – ratings-based

	action	animat.	comedy	drama	family	Horror	romance	thriller
action	0.129	0.095	0.078	0.067	0.086	0.093	0.075	0.109
animat.	0.095	0.167	0.074	0.059	0.125	0.077	0.074	0.082
comedy	0.078	0.074	0.072	0.058	0.071	0.065	0.070	0.074
drama	0.067	0.059	0.058	0.063	0.056	0.060	0.065	0.069
family	0.086	0.125	0.071	0.056	0.119	0.067	0.072	0.076
horror	0.093	0.077	0.065	0.060	0.067	0.149	0.060	0.098
romance	0.075	0.074	0.070	0.065	0.072	0.060	0.091	0.074
thriller	0.109	0.082	0.074	0.069	0.076	0.098	0.074	0.109

The following experiments evaluated the effect of the sparsity of the target user ratings in the target domain ratings matrix, on the accuracy of the predictions. Hence, the users were partitioned into *12* categories, according to the percentage of 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 were generated using the following CF approaches: *Centralized Prediction*, *Distributed Peer Identification*, *Distributed Neighborhood Formation* (including three different

[1] Limited experimental evaluation using Pearson correlation [17] as the similarity metric yielded similar results.

variants of inter-domain correlation computation: content-based, ratings-based and uniform), *Local Prediction*, and *Distributed Prediction*. The predictions' accuracy was measured using the widely-used Mean Average Error (*MAE)* metric [10]:

$$MAE = \frac{\sum_{i=1}^{N} | p_i - r_i |}{N}$$

where N denotes the total number of the predictions, p_i is the predicted rating and r_i is the real rating on the movie in prediction number i. In the following figures, the horizontal axis shows the percentage of rated movies in the target genre and the vertical axis the *MAE*. The baseline for all the comparisons is *Centralized Prediction*, as its results are identical to the results that would have been obtained in traditional centralized CF.

The prediction accuracy of *Local Prediction*, *Distributed Prediction* and *Centralized Prediction* (Figure 2) shows that both *Local Prediction* and *Distributed Prediction* CF outperform *Centralized Prediction* 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 Prediction*, or over the ratings from other genres in *Distributed Prediction*, yields more accurate similarity values than the similarity computation over all the available ratings. This means that the computation of the user-to-user similarity is not necessarily less accurate when fewer ratings are used. This also shows that when a prediction for an item is generated, it is beneficial to base the user-to-user similarity only on relevant items. This is in line with other results that have tried to use item weighting in similarity computation [12][21][23]. In our approach we rather use a feature selection approach [13], where the selected features (items) are those that an independent knowledge source (the genre classification) evaluated as relevant. Hence we believe that the achieved improvement is originated by the fact that the ratings from these genres are more important for computing the similarity value in the target genre, whereas the other ratings do not introduce additional valuable information and may rather introduce noise into the computation.

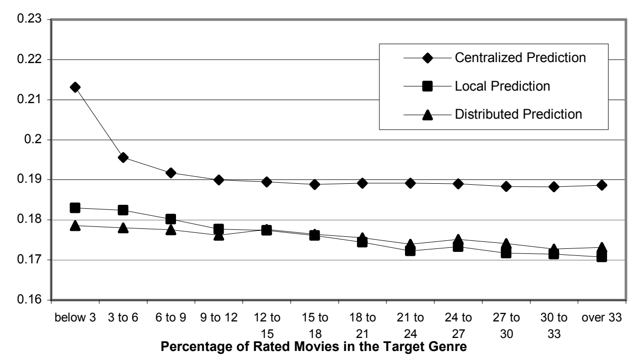

Figure 2. Centralized, Local and Distributed Predictions.

The comparison of *Local Prediction* and *Distributed Prediction* approaches shows that for a small percentage of rated movies, i.e., sparse ratings matrix, *Distributed Prediction* 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 under certain conditions *Local Prediction* and *Distributed Prediction* approaches are inapplicable. For example, for the group of users that rated less than *3%* of movies, predictions can be generated only for comedies and dramas, as only in these genres *3%* of the number of movies is above *6* movies, the minimal number of movies we used for the similarity computation. Hence, although the accuracy of *Local prediction* and *Distributed Prediction* is higher, they are not capable to generate predictions for certain movies, and that could negatively affect the ability of the system to recommend all the interesting movies.

The results of *Distributed Peer Identification* and *Centralized Prediction* approaches (Figure 3) show that for a small percentage of rated movies, *Centralized Prediction* is more accurate. This can be explained by the fact that when the user rated a small percentage of movies, the nearest-neighbors candidates computed by the *Distributed Peer Identification* are not accurate and differ from the real nearest-neighbors. However, the accuracy of the candidates set increases with the percentage of movies rated by the user and *Distributed Peer Identification* outperforms *Centralized Prediction* starting from the group of users that rated between *9%* to *12%* of movies (statistically significant, $p=0.03896$). It should be stressed that the accuracy of *Distributed Peer Identification* is bounded by the accuracy of the *Local Prediction*. Their prediction generation is based on ratings from the target genre only, but while the set of K nearest-neighbors in *Distributed Peer Identification* is found by an approximated heuristic search, in *Local Prediction* it is found by an exhaustive search of all the users. As such, their accuracy may be identical only if the approximated search in *Distributed Peer Identification* will find the real set of K nearest-neighbors.

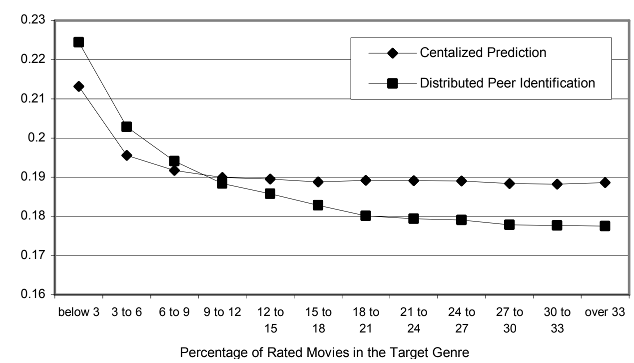

Figure 3. Centralized Prediction and Distributed Peer Identification.

Figure 4 shows the results of *Distributed Neighborhood Formation* (*Cross-Genre*) and *Centralized Prediction* approaches. Three particular instances of *Cross-Genre* correlations were evaluated: (1) *Cross-Genre* with content-based inter-genre correlations, (2) *Cross-Genre* with ratings-based inter-genre correlations, and (3) *Cross-Genre* with uniform inter-genre correlations, where all the correlations are set to *1*. The latter approach was aimed at evaluating the contribution of other *Cross-Genre* and served as a baseline for experimental comparisons of *Cross-Genre* approaches.

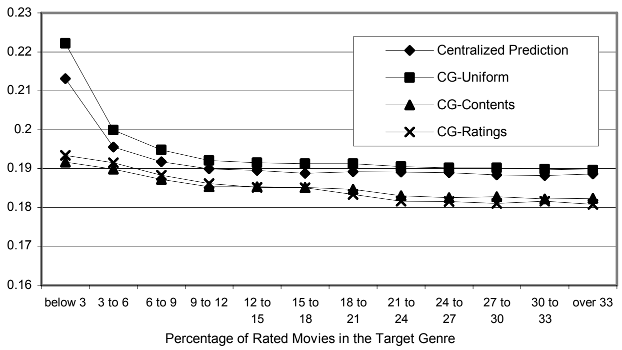

Figure 4. Cross-genre Uniform, Content-based, and Ratings-based Distributed Neighbourhood Formation and Centralized Prediction.

The results show that both content- and ratings-based *Cross-Genre* (Distributed Neighborhood Formation) outperform *Centralized Prediction* (statistically significant, $p=0.00058$ and $p=0.00024$, respectively). It can be explained by the observation that the weighted similarity metric, which aggregates domain-related correlations, i.e., inter-genre correlations, is more accurate than the *Standard* similarity metric assigning equal weights to all the ratings. It should be stressed that the accuracy of the *uniform Cross-Genre* correlation is lower than the accuracy of *content-based* and *ratings-based Cross-Genre* correlations, and it is also worse than the accuracy of *Centralized Prediction*. This shows that *Cross-Genre* correlation is beneficial and improves the accuracy of the predictions only when a meaningful correlation measure is used.

Although content-based and ratings-based inter-genre correlation matrices were different, the accuracies of both approaches are very similar. This reinforces the validity of ratings-based similarity computation [6]. We further observe that it is quite surprising here that the knowledge contained in the item descriptions is not improving the rating-based approach, as it is normally observed in hybrid recommender systems [7]. We conjecture that better content-based correlation measures could be defined and used.

The comparison of content- and ratings-based *Cross-Genre* (*Distributed Neighborhood Formation*) and *Local Prediction* approaches shows that the latter is more accurate for any percentage of rated movies (statistically significant, $p=4.13E-10$ and $p=5.66E-13$, respectively). However, as discussed earlier, *Local Prediction* may be inapplicable for a low percentage of rated movies due to the sparsity of ratings in the target domain ratings matrix. In this case, Distributed Neighborhood Formation should be applied, as its accuracy still outperforms the accuracy of *Centralized Prediction*.

In summary, the proposed approaches can be partitioned intro three groups according to their performance. The best accuracy of the predictions is shown by the first group of *Local Prediction* and *Distributed Prediction* approaches. However, it should be observed that these approaches are not applicable for very sparse datasets, where the traditional CF is more appropriate. The second group includes two weighted *Distributed Neighborhood Formation* approaches: content-based and ratings-based. Both of them improve the accuracy of the predictions, compared to the traditional CF and to a simplistic uniform approach, assigning equal weights to all the inter-genre correlations. Finally, *Distributed Peer Identification* approach belongs to the third group which shows the worst accuracy of the predictions. Its accuracy is inferior to the accuracy of traditional CF for sparse rating vectors, while for denser rating vectors it generates more accurate predictions than traditional CF.

5. CONCLUSIONS AND FUTURE WORK

This work focuses on the generation of CF recommendations using multiple distributed sources of user modeling data, i.e., multiple sets of users' ratings on items. In particular, it implements and evaluates the effect of exchanging four types of user modeling data: (1) complete UMs – *Centralized Prediction* approach, (2) lists of the nearest-neighbors candidates – *Distributed Peer Identification* approach, (3) degrees of users' similarity computed over the local data – *Distributed Neighborhood Formation* approach, and (4) complete predictions – *Distributed Prediction* approach. The fifth evaluated approach, *Local Prediction*, referred to a setting, where no user modeling data is exchanged between the systems.

Experimental evaluation, conducted in a movies dataset, showed that generating predictions exploiting a domain-related (genre) partitioning of user modeling data yields a higher accuracy than that of the original centralized repository with the traditional CF recommendation generation mechanism. All the proposed CF approaches (excluding *Distributed Peer Identification* for sparse rating vectors) improve the accuracy of the generated predictions in comparison to *Centralized Prediction* approach, whose accuracy is identical to the accuracy of traditional centralized CF. Hence, the experiments show that when the user modeling data are distributed among multiple repositories, importing and aggregating various types of user modeling data from these repositories can improve the accuracy of the generated predictions compared with the original centralized recommender system.

The main shortcoming of the presented work lays in the difficulty to apply it in recommendation applications, where it is impossible to classify the items into several topics or domains. In this case the *Distributed Prediction* approach is not feasible, since it requires that an item belong to at least two independent topics or domains. Hence, it is planned in the future to consider and generate real datasets of cross-domain and cross-topics user modeling data and to evaluate there the proposed approach. These additional data sets will help us to understand the rationale of the observed reduction of the prediction error. At this stage of the research we conjecture that the two important factors are: a) the usage of a more precise user-to-user similarity computation that depends on the target item (its genre in this example), b) the combination of predictions or similarity computations performed by multiple systems.

Also, we plan in the future to exploit various feature selection and machine learning mechanisms [13] for devising the exact user modeling data that should be imported in order to optimize the accuracy of the predictions. Particularly, the issue of a weighted combination of complete predictions from remote domains will be investigated in depth. In addition, it is planned to evaluate the proposed cross-domain mediation with other recommendation techniques and application domains.

6. REFERENCES

[1] Adomavicius, G. Tuzhilin, A., 2005, “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions”, in IEEE Transactions on Knowledge and Data Engineering, vol 17(6).

[2] Berkovsky, S., 2006, “Decentralized Mediation of User Models for a Better Personalization”, in proc. of the AH Conference, Dublin, Ireland.

[3] Berkovsky, S., Busetta, P., Eytani, Y., Kuflik, T., Ricci, F., 2005, “Collaborative Filtering over Distributed Environment”, in proc. of the DASUM Workshop, Edinburgh, UK.

[4] Berkovsky, S., Goldwasser, D., Kuflik, T., Ricci, F., 2006, “Identifying Inter-Domain Similarities through Content-Based Analysis of Hierarchical Web-Directories”, in proc. of the ECAI Conference, Riva del Garda, Italy.

[5] Berkovsky, S. Kuflik, T., Ricci, F., 2007, “Cross-Domain Mediation in Collaborative Filtering”, in proc. of the UM Conference, Corfu, Greece.

[6] Bridge, D., Kelly, J.P., 2006, “Ways of Computing Diverse Collaborative Recommendations”, in proc. of the AH Conference, Dublin, Ireland.

[7] Burke, R., 2002, “Hybrid Recommender Systems: Survey and Experiments”, in User Modeling and User-Adapted Interaction, vol. 12(4).

[8] Freyne, J., Smyth, B., 2005, “Communities, Collaboration and Cooperation in Personalized Web Search”, in proc. of the ITWP Workshop, Edinburgh, UK.

[9] Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J., 1999, “An Algorithmic Framework for Performing Collaborative Filtering”, in proc. of the SIGIR Conference, Berkeley, CA.

[10] Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J., 2004, “Evaluating Collaborative Filtering Recommender Systems”, in ACM Transactions on Information Systems, vol. 22(1).

[11] IMDb, The Internet Movie Database, *http://www.imdb.com*

[12] Jin, R., Chai, J. Y., Si, L , 2004, “An Automatic Weighting Scheme for Collaborative Filtering”, in proc. of the SIGIR Conference, New York, NY.

[13] Kohavi, R., John, G.H., 1997, “Wrappers for Feature Subset Selection”, in Artificial Intelligence Journal, vol.97(1-2).

[14] Linden, G., Smith, B., York, J., 2003, “Amazon.com Recommendations: Item-to-Item Collaborative Filtering”, in IEEE Internet Computing, vol. 7(1).

[15] McJones, P., 1997, “EachMovie Collaborative Filtering Data Set”, HP Research.

[16] Miller, B.N., Konstan, J.A., Riedl, J., 2004, “PocketLens: Toward a Personal Recommender System”, in ACM TOIS, vol.22 (3).

[17] Pennock, D.M., Horvitz, E., Giles, C.L., 2000, “Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering”, in proc. of the AAAI Conference, Austin, Texas.

[18] Salton, G., McGill, M., 1983, “Introduction to Modern Information Retrieval”, McGraw-Hill Publishing.

[19] Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001, “Item-Based Collaborative Filtering Recommendation Algorithms”, in proc. of the WWW Conference, Orlando, FL.

[20] Schafer, J. B., Konstan, J. A., Riedl, J., 2001, “E-Commerce Recommendation Applications”, in Data Mining and Knowledge Discovery, vol. 5(1-2).

[21] Yu, K., Xu, X., Ester, M., Kriegel, H.P. 2003, “Feature Weighting and Instance Selection for Collaborative Filtering: An Information-Theoretic Approach”, in Knowledge Information Systems, vol. 5(2).

[22] Weiss, M., 2002, “Exploiting Communities of Interest to Find Information on the Web”, Discussion Paper, Carleton University, Ottawa, Canada.

[23] Zeng, C., Xing, C.X., Zhou, L. Z., 2003, “Similarity Measure and Instance Selection for Collaborative Filtering”, in proc. of the WWW Conference, New York, NY.