An Experimental Study in Cross-Representation Mediation of User Models

Federica Cena
Università degli Studi di Torino
Dipartimento di Informatica
Torino - Italy
federica.cena@unito.it

Cristina Gena
Università degli Studi di Torino
Dipartimento di Informatica
Torino - Italy
cristina.gena@unito.it

Claudia Picardi
Università degli Studi di Torino
Dipartimento di Informatica
Torino - Italy
claudia.picardi@unito.it

ABSTRACT

The paper presents the result on cross-representation mediation of user models in the context of movie recommendation. We analyze the possibility of initializing the user models for a content-based recommender starting from movie ratings provided by users in other social applications. We focus in particular on (i) an approach for inferring user model preferences from rating and (ii) the experimentation of several methods to determine default user-model preferences from community-based ratings.

We tested different variations of the proposed approach exploiting a subset of the MovieLens 10M Dataset, computing rating predictions, and analyzing the mean absolute error between such predictions and the actual ratings provided by users.

The results show that the approach is in general feasible, with MAEs decreasing for users who provided more ratings and stabilizing around 0.15. They also show that community-based ratings are best used by inferring from them an "average user model", whose preferences can fill in missing values from individual user models.

CCS Concepts

•**Information systems** → **Recommender systems;** *Social recommendation;* Social networks;

Keywords

Cross-representation mediation of user models, Movie recommendation, Content-based recommender systems

1. INTRODUCTION

Nowadays not only users leave a large amount of ratings spread among several web sites, but also the same items (e.g., movies, books, songs, hotels, restaurants, etc.) are rated several times in different systems, from e-commerce web sites to social network systems. On the one side, if a user is new in a target system (i.e., has few to none rated items), but has a rating history in another source system in the same domain, the already rated items can be imported and used to recommend relevant items in the target domain [19]. Moreover, knowledge acquired in a source domain could be transferred and exploited in another target domain. Cantador et al. [8] propose to leverage all the available user data provided in various systems and domains in order to generate a more complete user model and better recommendations, resulting in the so-called *cross-domain recommendation.*

On the other side, as a practical application of knowledge transfer in the context of a single domain (e.g. movies), a target system may import user ratings of overlapping items from a source system as auxiliary data, in order to address the data-sparsity problem, which is a common challenging problem in many newly launched collaborative filtering (CF) recommenders, see for example [15], [18].

For instance, if a company decides to launch a new movie CF recommender, it might import user ratings from another CF recommender in order to have an initial and reliable rating matrix. Such data is made available by several sources: MovieLens[1] makes its ratings and its rating matrix publicly available, while theMovieDB [2] provides a large movies database rich of information regarding movies such as title, cast, images, keywords, trailers, similar movies, ratings and so on, available through a public API. In this way a source system may be used to overcome the cold start problem related to users and items,

Cantador et al. [8] define this situation as the *system cold-start problem* (system bootstrapping), related to situations in which a recommender is unable to generate recommendations due to an initial lack of user preferences. They propose as possible solution to bootstrap the system with preferences from another source outside the target domain by *transferring rating patterns.*

Information imported from a source system, for instance a CF movie recommender, could be also used to initialize a content-based (CB) recommender. This approach has been defined as *(cross-representation) mediation of user models*, which is typical from CF to CB recommender systems [5]. According to this approach, a CB recommender system, having partial or no user model (UM) data, can generate recommendations for users by mediating UM data of the same users, collected by a CF system. The mediation process transforms the UMs from the representation used by the CF recommender (i.e. ratings) to the content-based repre-

[1] https://movielens.org/
[2] https://www.themoviedb.org/

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

sentation (i.e. expressions of interest toward item features). The mediation process exploits the item descriptions that are typically not used by CF recommender systems. Such user model mediation may offer a potential solution to mitigate the cold-start and sparsity problems in recommender systems, and it may also help avoiding the paradox of the active user [10], which recommender systems may suffer from: Users often refuse to visit the sites that ask them to reply to an interview first (or fill in forms and questionnaires, rating long list of items, etc.) because they would save time getting their immediate task done. However, user model mediation does not solve the problem of missing values in the user model at the beginning of the interaction. While it is true that missing values in a user model can be predicted by a content-based approach, this can be done after the user has made a certain amount of interaction of the system. In a CF scenario, instead, a common solution for these problems is to fill the missing ratings with default values such as the middle value of the rating range, and the average user or item rating, see [17].

In this paper we experiment with an approach that combines some of the solutions described above. We in fact analyze a cross-representation mediation method which, in the context of movie recommendations. imports knowledge on items and users from external sources, on order to avoid asking users for information on their tastes. The knowledge on users is in the form of user ratings, as it comes from a CF movie recommender (in our case, MovieLens). The user model is inferred from such ratings. In this respect, the task is analogous to the one described in [5], although our algorithm for inferring the user model, and for generating rating predictions, is different. Also, we enrich the inferred user models with default values, computed from other users' data imported again from external sources. We experimented with several approaches to compute such default values. Default values are also useful in the extreme case where no information is present in the user model. When a new user registers into the system she may not wish to (or be able to) provide access to her data (explicitly given, or from other sources); in this case we have a *reticent user*. In this situation, the system will fill the missing user model values with the default values coming from the community of users. As soon as the user's interest value become available, this will override community's interest values.

To summarize, this paper provides experimental results on:

- the exploitation of *user model mediation* to enable a warm start in a cross-system recommendation scenario, thus avoiding the well known cold start problem in recommender systems and the paradox of active users.

- the exploitation of different strategies for calculating default values, which are generated starting from values of the user community of the system, in order to solve the problem of *missing values* in the user model, which is especially problematic in the early interactions.

This paper is organized as follows: Section 2 describes our general approach and motivations; Section 3 present related work in the field; Section 4 describes the user modeling approach, and the methods we experimented with for computing default values; Section 5 describes the experiments and their results; and Section 6 presents some discussion on future work and concludes the paper.

2. BACKGROUND AND APPROACH

The approach presented in this paper has been inspired by the development of ReAL CODE[3] (Recommendation Agent for Local Contents in an Open Data Environment), a movie social network and a movie recommender system. ReAL CODE retrieves information from external sources, such as Facebook and TheMovieDatabase, and automatically maps it into its own knowledge base.

In particular ReAL CODE is designed to import user data from Facebook as bootstrapping information to initialize the user model at the beginning of the interaction, to solve the cold start problem and to avoid the paradox of the active user. We consider actions that users can make on movies within Facebook: in particular we consider ratings, expressed on a 1-5 rating scale[4]. Information imported from Facebook is used to initialize the CB user model, which however could be in principle enriched with ratings coming from other sources as well. In ReAL CODE, user model interest values are then updated by considering all the possible user actions as an implicit source of interest following an approach similar to the one described in [9]. A detailed description of this system is out of the scope of this paper, but we presented the system as it was the context of the experiments described in the paper.

The content-based user model of ReAL CODE (see Sec. 4) contains the user interest towards all the features of the "movie" object. The "movie" object is represented by means of the following five main features:

- *genres*, whose values are taken from the genres taxonomy of TheMovieDB containing 17 values;

- *tags*, that are keywords assigned by users to the movie

- *actors*, who play roles in the movie;

- *directors*, that direct the movie;

- *production countries*, countries in which the movie has been produced and those in which has been distributed.

Notice that both the above features and the related data set are imported from TheMovieDB, which is in fact the main source for the movie knowledge base of the system.

For each imported rating (*user*, *movie*, *rating*), our goal is to propagate the *rating* values to the features characterizing *movie* in the user model of *user*. Sec. 4 details the propagation method. This mediation is inspired by the work of Berkovsky et al. [5], with a different approach concerning user model extraction. As is it suggested in [5], a collaborative filtering UM in the movie domain comprises a set of movie ratings explicitly provided by the user. The CF user model may contains information as {Star wars:1, Blade runner:0.8, Serendipity:0.2, Sleepless in Seattle: 0.2}, where the movie ratings are translated on a continuous scale ranging between 0 and 1. Although this collaborative filtering UM represents the user with only a few rating, it can be

[3]http://www.realcode.it/. ReAL CODE is a research project funded in the context of POR FESR 2007/2013 of the Piedmont Region, Italy.

[4]We do not consider other input from Facebook.

hypothesized that the user likes science-fiction movies and dislikes romantic comedies. Hence, the content-based UM of this user may be {science-fiction:0.9, romantic comedy: 0.2}, where the genre weights are computed as an average of the ratings for the movies from this genre. Similarly to the genre weights, Berkovsky et al. [5] propose that the weights of other movie features, such as directors, producers, and actors, can be inferred from ratings.

Regarding recommendations, suggestions are given using a content-based approach, as will be described in section 5.

According to the classification by Cantador et al. [8] our system can be classified as i) a *cross-system recommender* since items may correspond across different systems, and users may overlap in a *single domain* (i.e., movie); ii) a cross (representation) *mediator* (from CF to a CB approach).

To summarize, our system:

- is designed to import domain data (e.g., movie description) from external sources;
- is designed to import user ratings from another (source) system in order to overcome the cold start problem of the user model, and the paradox of active user;
- mediate the imported user data from a rating representation to a content-based representation;
- fill the missing values of the user model with default values generated starting from the user community, as will be described in Section 4. Since at the beginning, the system, as usual, will miss community data, we propose to bootstrap the system with rating values coming from a source system (in particular we exploited the MovieLens data set). This solution may also help in solving the data sparsity problem. As soon as real user data become available, user data will override community and user data.

Notice that, we could also use the Facebook data for bootstrapping the system community, as example, the ratings given by the user's friends. However, from our initial testing, this approach does not completely solve the problem of missing values, since rating coming from user's friends may be extremely sparse.

3. RELATED WORK

In this article, we investigate the idea of user model transfer as a way to infer missing user models values for new users or reticent ones.

First, we present a study on cross-representation mediation of user models from collaborative filtering to content-based between different systems in the same domain, in order to address the *new user problem.* This issue takes place when a user starts to use a new recommender, which has no knowledge of the user's interests and is not therefore able to provide recommendations. This can be addressed by transferring user models created in another (source) system, "translating" and importing them to the target system. Many works exploit cross-system personalization for this goal, such as [23, 1, 20, 19, 4, 2, 3, 16, 11, 22].

User model transfer can happen in different ways, depending on representation modalities: the simpler case is when the representation modalities are the same, while the case when representation modalities are different (e.g., from CF to CB) is more complex and needs some form of "translation" or *mediation.* The most favorable scenario implies that different systems share user preferences of the same type and the same type of representation. This scenario was addressed, for example, by Berkovsky et al. with a mediation strategy for cross-domain collaborative filtering [2, 3].

An example of user model transfer from CB to CB is provided by Wongchokprasitti et al. [23], who compare different transfer strategies in an academic information setting where the source system is a search system for scientific papers and the target system is a system for sharing academic talks. Both systems represent the user with keyword-based user models. Another example is provided by Abel et al. [1] where users are modelled exploiting their Social Web activities in different systems (tags-based and form-based user models).

When data coming from the source system have a different representation, a *cross-representation mediation* is necessary [4]. Mediation aims at resolving the heterogeneity in the representations of the same user for the same item in the same context. In this paper we propose an approach to this issue: inferring the CB user model from user ratings coming from a CF system. A similar approach is the one by Berkovsky et al. [5], which proposes a mediation framework by importing and integrating in a CB system the data collected by CF systems. Here the authors discuss an approach that seems particularly effective for users with a small number of ratings. This suits their goals, as they aim at replacing the CB recommendation with a CF recommendation once the user has provided a larger number of rating. Our approach differs from theirs in (i) how we infer the user model, (ii) how we apply it in order to compute rating predictions and (iii) how we treat the case of missing values in the user model. Also our goal is different, as we are interested in CB recommendations that are effective also for users with larger numbers of ratings,.

The problem of filling in *missing ratings/user's interest values* in recommender systems, both for new and reticent users, is discussed by several authors. One possible solution is to fill the value with community-based preferences from another source outside the target system [8]. In CF recommenders, in particular, the missing ratings of items are filled with default values [6, 12], such as the middle value of the rating range, or the average user or item rating.

We adopt a similar approach (i.e. using community-based information), but we combine it with cross-representation mediation (i.e. the community-based information is a set of ratings from which we infer user model default preferences).

This is in line with the idea of social information access [7], i.e., methods for organizing users past interaction within an information system, in order to provide better access to information to the future users of the system. Usually, this approach has been used for improving search results, for example re-ordering link using social wisdom, suggest additional results found by earlier searches or providing social annotations of search results based on link popularity or past link selection by other users [21]. Another application is adaptive community-based course planning, to provide personalized access to course information and provide social recommendation about courses [13].

4. USER MODEL

As already discussed, our goal is to infer the user model for content-base recommendation from a set of movie ratings the user may have provided either within the system itself, or in other systems — possibly collaborative-filtering recommenders, collecting movie ratings as part of the user profile.

In order to do this, we first need to describe how each item - in our case, movie - is described.

4.1 Movie Description

As it is common in content-based systems, each movie m, besides being characterized by an *id* and a *title*, is further described by a set of features $desc(m) = \{F_1, \ldots, F_n\}$ where each feature F_i is a pair $(category, value)$.

In our experiments, we extracted movie descriptions from TheMovieDB[5] and focused on the following set of categories:

$$FCat = \{genre, directors, actors, production\_country, tags\}.$$

In extracting the *actor* features, we limited ourselves to the five actors identified as *main cast*.

4.2 User Model Extraction

For each user u, her user model $\mathrm{UM}(u)$ is constituted by two functions:

- the **interest** function $int_u : Features \longrightarrow [0,1]$ expresses the interest of user u in a certain feature F as a real number in the $[0,1]$ interval;
- the **action count** function $act_u : Features \longrightarrow \mathbb{N}$ counts the number of action user u performed within the system that are pertinent to a certain feature F.

Given a set of movie ratings provided by user u on a scale $[l,u]$, we denote by $mov(u)$ the set of movies rated by u, and by $rat(u,m)$ the normalization on a $[0,1]$ scale of the true rating originally given by u. Moreover, for a given feature F, we denote by $mov(u|F)$ the set of movies rated by u having F in their description; formally $mov(u|F) = \{m \in mov(u) \mid F \in desc(m)\}$.

We then compute interest and action count as follows:

$$act_u(F) = |mov(u|F)|, \tag{1}$$

$$int_u(F) = \frac{\sum_{m_i \in movies(u|F)} r_i}{act_u(F)}. \tag{2}$$

As an example, let us suppose $F = (actor, Ewan\ McGregor)$ and that a user u has rated three movies with this F in their description:

Trainspotting	0.7
Moulin Rouge!	0.8
Star Wars - Episode I	0.5

Then we have:

$$act_u(F) = 3$$

$$int_u(F) = \frac{0.7 + 0.8 + 0.5}{3} = 0.\overline{6}$$

4.3 Default user values

As discussed in 1, the interest function we infer from user ratings tends to be under-defined for "sparse" feature categories (such as the *actor* and *director* categories), where there are a lot of possible different feature values. As the number of ratings provided by the user grows larger, the number of missing values decreases; however there are bound to be missing values even for user models inferred from a large number of ratings.

In order to tackle this problem, we chose to test a community-based approach where a default interest function int_{def} is computed from other users' ratings. Such function is used in place of int_u whenever we need to evaluate a feature F with $act_u(F) = 0$.

We experimented with three methods, resulting in three different default interest functions:

- **Middle value**: the external ratings are used to compute a global average rating. If ratings were given randomly in the $[0,1]$ interval, this value would be 0.5. However it is well known that users tend to use more frequently higher ratings than lower ones, as either they do not rate things they do not like, or - being good recommenders to themselves - they do not watch movies that are evidently not to their tastes. The default interest function int_{mid} is in this case a constant, and it is equal to the global average rating.

- **Average user**: the external ratings are used as if they all belonged to the same user, to infer - in the same way described above - an average user model, with its own interest function int_{ave}. This "average" interest function is then used as default.

- **Notoriety-based**: it may be argued that, if a feature does not appear among a user ratings, it is because that user never watches - and therefore never rates - movies with that feature. So, absence from the user model may suggest a dislike. On the other hand, a feature may be absent from the user model because it is not very common and therefore the user has never encountered it, so she does not know whether she likes it or not. In order to take into account these different situations, we use external ratings to define a notoriety measure for each feature. The notoriety $ntr(F)$ of a feature F with category C is therefore computed as the number of ratings given to movies with F in it, normalized by the maximum notoriety achieved within C in order to obtain a value between 0 and 1. The default interest function then is computed as: $int_{ntr}(F) = int_{mid}(F) * (1 - ntr(F))$.

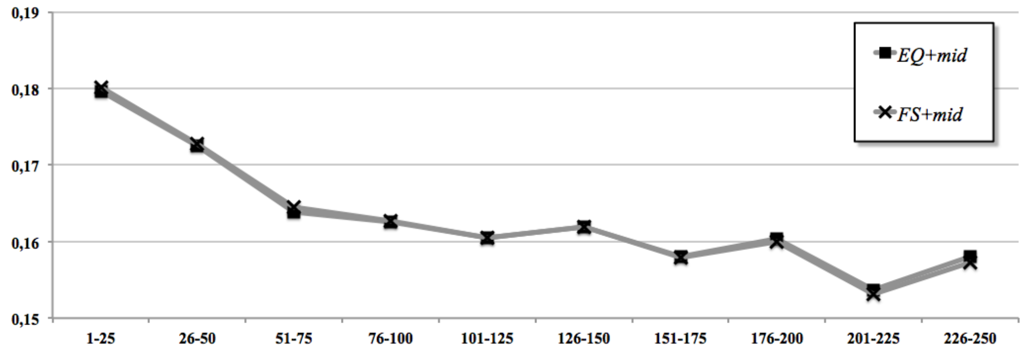
We denote by int_{u+def} the interest function obtained combining a given user's interest function int_u with a default interest function int_{def}:

$$int_{u+def}(F) = \begin{cases} int_u(F) & \text{if } act_u(F) > 0 \\ int_{def} & \text{otherwise.} \end{cases}$$

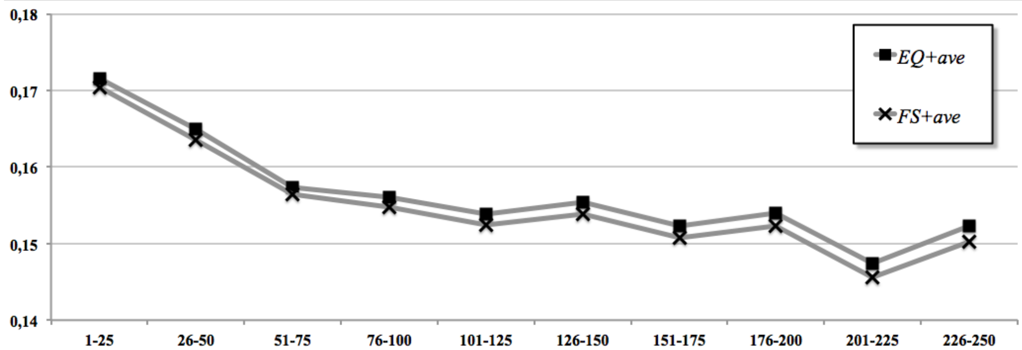
5. EVALUATION

We evaluated our approach using the MovieLens 10M Dataset [14] and, in order to compare our results with existing work, we replicated the experimental settings by Berkovsky et al. [5]. The main difference with their settings relies on the dataset. In fact Berkovsky et al. exploited the EachMovie dataset, storing 2,811,983 ratings of 72,916 users on 1,628

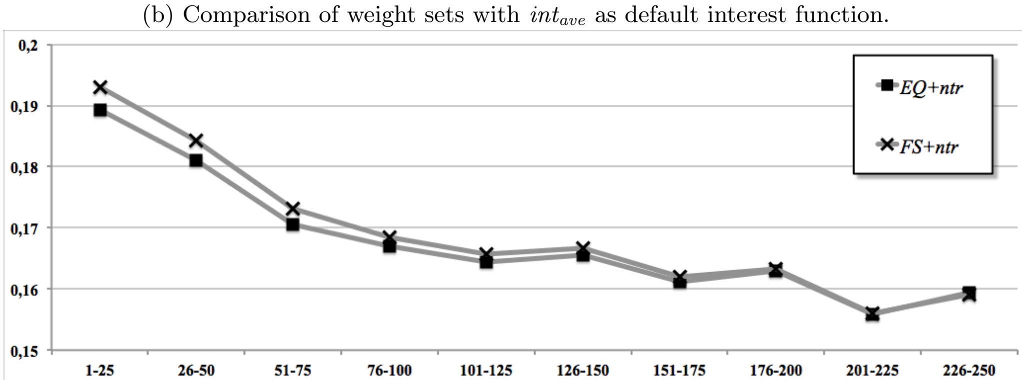
[5]http://www.themoviedb.org/



(a) Comparison of weight sets with int_{mid} as default interest function.



(b) Comparison of weight sets with int_{ave} as default interest function.



(c) Comparison of weight sets with int_{ntr} as default interest function.

Figure 1: Comparison of weight sets FS and EQ with each default interest method.

movies, which is no longer available.; nonetheless, MovieLens is an evolution of this dataset.

Therefore in our evaluation all users are regarded as "new" to our system (which has no information on them prior to user model extraction), while movies are assumed to be already described within our system (i.e. user model extraction does not have to deal with"new" movies).

We first randomly selected 5000 users from the MovieLens DB, whose 701017 samples were used as community ratings to compute the three default interest functions int_{mid}, int_{ave}, and int_{ntr}, according to the three different methods (middle value, average user, notoriety-based) described in the previous section.

We then created 10 groups of 325 users according to the number of ratings available for each of them. The first group contained users with 1 to 25 samples, the second group contained users with 26 to 50 samples, and so on, up to the tenth group, containing users with 226 to 250 samples. The 325 users in each group were randomly selected among those with the proper number of samples.

For each group $i = 1, \dots, 10$ we randomly split users samples into ten subsets $j = 1, \dots, 10$ to be analyzed for ten-fold cross-evaluation: in fold j, the evaluation set E_i^j coincided with the j-th subset, while the training set T_i^j contained all the remaining samples. Thus for each user, 9/10 of her samples were used for the training set, while the remaining 1/10 was used for the evaluation set.

For each user in each group, we extracted from her training set a user model, as described in section 4.2. We then run a basic content-based recommender on the movies in the users' evaluation set, and obtained *rating predictions*. Finally, we computed the mean absolute error (MAE) between such predictions and the *actual* ratings provided by the user herself.

In order to ensure the statistical significance of the observed differences between approaches, we performed a paired t-test for each hypothesis of the form “approach X has a lower MAE than approach Y on group i”.

5.1 Recommendation

We computed rating predictions (for each user u and movie m in her evaluation set) according to the following formula:

$$pr_{u,m} = \frac{\sum_{c \in FCat} w_c \cdot score_c(u,m)}{\sum_{c \in FCat} w_c}, \qquad (3)$$

that is, as the normalized weighed sum of feature category scores $score_c(u, m)$. Each feature category score is in turn computed as the average interest the user has for the movie features associated with category c:

$$score_c(u,m) = \frac{\sum_{F \in desc(m), cat(F)=c} int_{u+def}(F)}{|\{F \in desc(m) | cat(F) = c\}|}.$$

As to the weights used in formula (3), we experimented with two weight sets:

- **Equal weights, all categories (EQ)**: $w_c = 1$ for every feature category c;
- **Equal weights, excluding *production_country* category (FS)**: $w_c = 0$ for $c = production\_country$; $w_c = 1$ for all other categories.

The second weight set exploits the *feature selection* analysis presented in [5], whose results lead the authors to exclude from their content-based recommender two categories: *production_country* and *language*. The latter, however, does not belong to our category set *FCat*.

As an example, let us suppose we want to recommend the movie *The Men Who Stare at Goats* to a user u with the following pertinent interest values in her user model:

(*genre*, *comedy*)	0.9
(*actor*, *Ewan McGregor*)	0.6
(*actor*, *George Clooney*)	0.3
(*production_country*, *us*)	0.9
(*tag*, *new age*)	0.45

While the movie descriptors are:

genre	*comedy*
genre	*war*
actor	*Ewan McGregor*
actor	*George Clooney*
director	*Grant Heslov*
production_country	*us*
tag	*new age*
tag	*paranormal*

Some features are missing from the user model; therefore we need to resort to the default interest function. Let us suppose we are using the int_{mid} function, which assigns to all features the constant value 0.706.

With the combined interest function int_{u+mid} we can now score the five categories:

$$\begin{aligned}
score_{genre} &= \frac{0.9 + 0.706}{2} = 0.803 \\
score_{actor} &= \frac{0.6 + 0.3}{2} = 0.45 \\
score_{director} &= 0.706 \\
score_{production_country} &= 0.9 \\
score_{tag} &= \frac{0.45 + 0.706}{2} = 0.578
\end{aligned}$$

Last, we can compute the expected rating. With equal weights for the five categories we obtain:

$$pr = \frac{0.803 + 0.45 + 0.706 + 0.9 + 0.578}{5} = 0.687$$

while if we apply feature selection we obtain:

$$pr = \frac{0.803 + 0.45 + 0.706 + 0.578}{4} = 0.634$$

5.2 Results

We tested each combination of a default interest method (*mid*, *ave*, *ntr*) and a weight set (EQ, FS). For each combination we repeated the experiment ten times, each with a different partition of the initial samples between training and evaluation sets (ten-fold cross evaluation). We then checked the resulting MAEs for each pair of combinations in order to test the statistical significance of their differences.

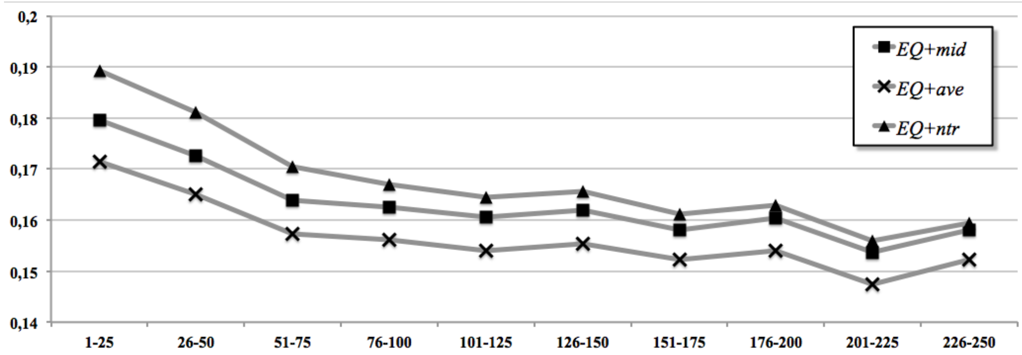
For each combination, we plotted the average MAE of each user group. In all the following diagrams, MAE values are represented on the vertical axis, while the number of ratings provided by users (i.e., user groups) are represented on the horizontal axis.

Figure 1 shows the comparison, for each default interest method, of the two weight sets, while figure 2 shows the comparison, for each weight set, of the three default interest methods.

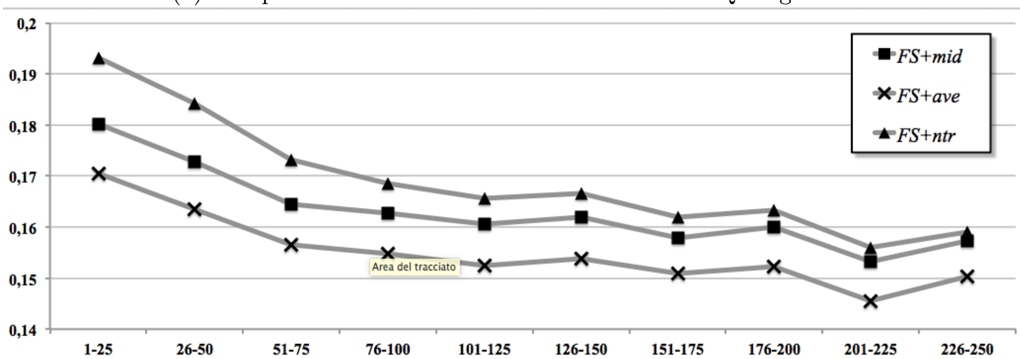
We see that with both weight sets the int_{ave} default function provides the best results (figure 2). Also, with both weight sets the middle value method (*mid*) outperforms the notoriety-based approach (*ntr*). The observed differences are statistically significant: for the FS diagram (a), $t \geq 3.828$ corresponding to a confidence $\geq 99.9\%$, while for the EQ diagram (b) $t \geq 2,908$, corresponding to a confidence $\geq 99\%$.

On the other hand, excluding the production country category (figure 1) does not result in significant changes in MAEs. While we can observe, in the case of default function int_{ave}, a consistently lower average MAE for the FS weights, the observed difference is statistically significant with confidence $\geq 95\%$ ($t \geq 2.16$) only for user groups with more than 125 samples. This should not be taken as a negative result with respect to feature selection, since it shows that excluding the *production_country* category does not generally affect the prediction accuracy, even improving it under some circumstances.

Figure 3 compares our best-performing combination (FS+*ave*) with the results in [5], which in turn compared their cross-representation mediation approach (CBFS) with a standard collaborative filtering technique (CF) applied to the same data. The cross-representation mediation approach proposed by Berkovsky and colleagues obtains a lower MAE for users with less than 70 samples; for more than 70 samples, predictions tend to worsen as the number of samples increases, stabilizing approximately at 0.20.



(a) Comparison of default interest methods with EQ weight set.



(b) Comparison of default interest methods with FS weight set.

Figure 2: Comparison of default interest methods with each weight set.

We can see in figure 3 that our approach effectively contrasts the tendency to obtain worse predictions for users with larger training sets. Although it could be expected that using a default interest function would significantly improve rating predictions (in [5] features missing from the user model are simply ignored), it is interesting to notice that such improvements appears to be more evident for larger training sets, where the user model is bound to be richer and the impact of the default interest function should consequently be lower. This suggests that the improvements we see are at least partly due to our different approach in extracting the user model.

However, for users with 1 to 25 samples, i.e., for the smaller training sets, the CBFS approach in [5] obtains a lower MAE (approximately 0.16 vs. approximately 0.17). This suggests that a combination of the two may be more effective than each separately.

6. DISCUSSION AND CONCLUSION

In this paper we discussed the results of an experimental study on cross-representation mediation of user models in the context of movie recommendation. The goal was to analyze the possibility of initializing the user models for a content-based recommender starting from movie ratings provided by users in other social applications. While previous ratings are commonly used as user model in collaborative-filtering approaches, in order to determine user similarity, content-based recommenders need the user model to express an "interest function", associating with each possible feature in the domain items characterization a value expressing the interest of the user for the feature itself.

In our study we focused in particular on two points concerning content-based user models:

- An algorithm to infer a content-based user model from the user's previous ratings. Such an algorithm can be exploited both to initialize the user model from the ratings she provided to other (not necessarily recommender) applications, and to maintain the user model without explicitly asking for her preferences in terms of movie features.
- A comparison of different methods for determining a default interest function from community ratings, to be used when the user's previous ratings do not allow to infer an interest value for a certain feature. In particular we experimented with a fixed default value established as the average of all community ratings, with a default user model inferred by considering the whole community as a single user, and with a mixed approach based on the notoriety of a feature.

Both the user model inference algorithm and the default interest function are particularly useful in case of system bootstrapping. The user model can be initialized importing data the same user provided to other applications. Whenever these are insufficient, default values from community ratings can be used.

We benchmarked our experimental results against the approach discussed in [5], recreating a similar experimental settings and comparing the mean average error (MAE). The analysis shows that:

- As it could be expected, the MAE decreases when the user model is inferred from a larger number of ratings

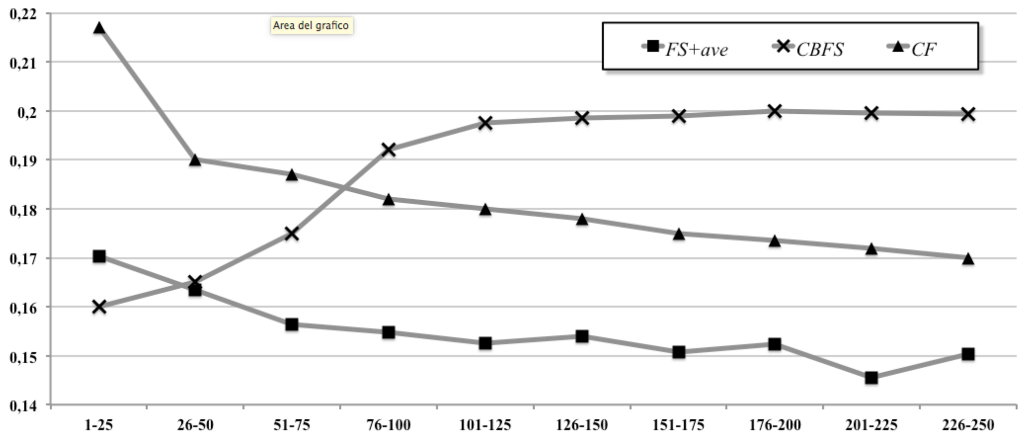


Figure 3: Comparison with approach in [5].

(the approach presented in [5] has the opposite behavior).

- The best results (lowest MAE) are obtained when the default interest function is computed as a default user model, inferred from community ratings by considering the overall community as a single user.

- The feature selection results presented in [5] prove useful in our case too, as removing the unselected features provides similar, when not better, MAEs.

- For users with a number of previous ratings higher than 25 (up to 250 in our analysis), our system improves over [5] and the improvement grows with the number of ratings. For users with less than 25 ratings, the approach presented in [5] has a lower MAE.

These results open the door to several further investigations. A first line of inquiry goes in the direction of combining our approach with [5], in order to benefit from the advantages of both. In fact, the two algorithms for user model inference take into account quite different aspects, and this is reflected in the different behavior they exhibit when used in recommendation. As future work, we will implement their approach using the MovieLens dataset, in order two have the two studies completely comparable.

A second study may investigate the possibility of using community ratings for fine tuning the weight of each feature category, rather than adopting the coarser approach of feature selection, where each category basically weights either 1 (selected) or 0 (discarded).

Finally, is better to remind that the system here described has been presented as it was in the context of the experiment detailed in the paper. As described, in the running system the user model is initialized getting user data from Facebook as bootstrapping information at the beginning of the interaction, while the community values are bootstrapped in the way we described in the paper. In the future, we can further investigate the benefit of using the data coming from the preferences of the user's friends, which could be mixed with the ratings coming from the bootstrapping community, since Facebook data are in general too sparse for covering all the user's model missing values. However, our system is designed to acquire data from different sources, both for the user model, and regarding the community values used for the system bootstrapping. Thus, in the future, we may also consider other sources of information to replace or complete the ones here described.

7. REFERENCES

[1] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2):169–209, 2012.

[2] S. Berkovsky, T. Kuflik, and F. Ricci. Cross-domain mediation in collaborative filtering. In C. Conati, K. McCoy, and G. Paliouras, editors, *User Modeling 2007: 11th International Conference, UM 2007, Corfu, Greece, July 25-29, 2007. Proceedings*, pages 355–359, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[3] S. Berkovsky, T. Kuflik, and F. Ricci. Distributed collaborative filtering with domain specialization. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 33–40, New York, NY, USA, 2007. ACM.

[4] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Model. User-Adapt. Interact.*, 18(3):245–286, 2008.

[5] S. Berkovsky, T. Kuflik, and F. Ricci. Cross-representation mediation of user models. *User Model. User-Adapt. Interact.*, 19(1-2):35–63, 2009.

[6] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[7] P. Brusilovsky. Social information access: The other side of the social web. In V. Geffert, J. Karhumäki, A. Bertoni, B. Preneel, P. Návrat, and M. Bieliková, editors, *SOFSEM 2008: Theory and Practice of Computer Science: 34th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, January 19-25, 2008.*

Proceedings, pages 5–22, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[8] I. Cantador, I. Fernández-Tobías, S. Berkovsky, and P. Cremonesi. *Recommender Systems Handbook*, chapter Cross-Domain Recommender Systems, pages 919–959. Springer US, Boston, MA, 2015.

[9] F. Carmagnola, F. Cena, L. Console, O. Cortassa, C. Gena, A. Goy, I. Torre, A. Toso, and F. Vernero. Tag-based user modeling for social multi-device adaptive guides. *User Model. User-Adapt. Interact.*, 18(5):497–538, 2008.

[10] J. M. Carroll and M. B. Rosson. Paradox of the active user. In *Interfacing thought: cognitive aspects of human-computer interaction*, pages 80–111. The MIT Press, Cambridge, MA, 1987.

[11] P. Cremonesi and M. Quadrana. Cross-domain recommendations without overlapping data: Myth or reality? In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 297–300, New York, NY, USA, 2014. ACM.

[12] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, Jan. 2004.

[13] R. Farzan and P. Brusilovsky. Social navigation support in a course recommendation system. In V. P. Wade, H. Ashman, and B. Smyth, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems: 4th International Conference, AH 2006, Dublin, Ireland, June 21-23, 2006. Proceedings*, pages 91–100, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[14] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19, 2016.

[15] B. Li, Q. Yang, and X. Xue. Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 2052–2057, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[16] M. Nakatsuji, Y. Fujiwara, A. Tanaka, T. Uchiyama, and T. Ishida. Recommendations over domain specific user graphs. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 607–612, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[17] X. Ning, C. Desrosiers, and G. Karypis. *Recommender Systems Handbook*, chapter A Comprehensive Survey of Neighborhood-Based Recommendation Methods, pages 37–76. Springer US, Boston, MA, 2015.

[18] W. Pan, E. W. Xiang, L. N. N., and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, AAAI'10, pages 210–235. AAAI Press, 2010.

[19] S. Sahebi and P. Brusilovsky. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *Proceedings of the 21st Conference on User Modeling, Adaptation and Personalization*, Umap13, pages 289–295, 2013.

[20] B. Shapira, L. Rokach, and S. Freilikhman. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23, 2013.

[21] B. Smyth, J. Freyne, M. Coyle, P. Briggs, and E. Balfe. I-spy — anonymous, community-based personalization by collaborative meta-search. In F. Coenen, A. Preece, and A. Macintosh, editors, *Research and Development in Intelligent Systems XX: Proceedings of AI2003, the Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 367–380, London, 2004. Springer London.

[22] A. Tiroshi and T. Kuflik. Domain ranking for cross domain collaborative filtering. In J. Masthoff, B. Mobasher, M. C. Desmarais, and R. Nkambou, editors, *User Modeling, Adaptation, and Personalization: 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012. Proceedings*, pages 328–333, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[23] C. Wongchokprasitti, J. Peltonen, T. Ruotsalo, P. Bandyopadhyay, G. Jacucci, and P. Brusilovsky. User model in a box: Cross-system user model transfer for resolving cold start problems. In F. Ricci, K. Bontcheva, O. Conlan, and S. Lawless, editors, *User Modeling, Adaptation and Personalization: 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 – July 3, 2015. Proceedings*, pages 289–301, Cham, 2015. Springer International Publishing.