User Model User-Adap Inter (2009) 19:35–63
DOI 10.1007/s11257-008-9055-z

ORIGINAL PAPER

Cross-representation mediation of user models

Shlomo Berkovsky · Tsvi Kuflik · Francesco Ricci

Received: 13 February 2007 / Accepted in revised form: 21 May 2008 /
Published online: 10 September 2008

Abstract Personalization is considered a powerful methodology for improving the effectiveness of information search and decision making. It has led to the dissemination of systems capable of suggesting relevant and personalized information (or items) to the users, according to their characteristics and preferences, as represented by a User Model (UM). Since the quality of the personalization largely depends on the size and accuracy of the managed UMs, it would be beneficial to enrich the UMs by mediating, i.e., importing and integrating, UMs built by other personalization systems. This work discusses and evaluates a cross-representation mediation of UMs from collaborative filtering to content-based recommender systems. According to this approach, a content-based recommender system, having partial or no UM data, can generate recommendations for users by mediating UM data of the same users, collected by a collaborative filtering system. The mediation process transforms the UMs from the collaborative filtering ratings to the content-based weighted item features. The mediation process exploits the item descriptions that are typically not used by the collaborative filtering recommender systems. An experimental evaluation conducted in the domain of movies shows that for users with small collaborative filtering UMs, i.e., users with few item ratings, the accuracy of the recommendations provided using the mediated content-based UMs is superior to that using the original collaborative filtering UMs. Moreover, it shows that the mediation can be used to improve a

S. Berkovsky (✉)
CSIRO ICT Centre, Hobart, Tasmania
e-mail: shlomo.berkovsky@csiro.au

T. Kuflik
University of Haifa, Haifa, Israel
e-mail: tsvikak@is.haifa.ac.il

F. Ricci
Free University of Bozen-Bolzano, Bolzano, Italy
e-mail: fricci@unibz.it

content-based recommender system by incrementally mediating collaborative filtering UM data (item ratings) and enriching the available content-based UMs.

Keywords Recommender systems · User modeling · Mediation of user models · Collaborative filtering · Content-based filtering

1 Introduction

The quantity of information available on the Web is growing rapidly and exceeding limited human cognitive capabilities. Moreover, in many search scenarios users would like to choose among a set of alternative items or services, but do not have sufficient knowledge or time for such decision-making. There is therefore a pressing demand for intelligent systems that provide personalized information services, i.e., deliver information tailored to specific user preferences and needs. This type of system is referred to in the literature as a *personalization* system (Mulvenna et al. 2000), and this work focuses on *recommender* systems (Resnick and Varian 1997), a particular type of personalization system.

Providing personalized recommendations to users requires modeling of their characteristics, preferences, and needs. This information is referred to in the literature as a User Model (UM) (Kobsa 2001). Typically, recommender systems build and maintain system-specific UMs, tailored to the requirements and technologies used by the system, e.g., the user's preferred item features or the user's ratings for some items. Because of this limitation, i.e., the fact that the UMs are system-specific, such UMs are *partial*, since they provide only a local view of the user that cannot be easily shared among multiple systems. Another major challenge of recommender systems is the support of new users, as in this case the system does not yet have any UM data for such users and it needs to bootstrap the UMs before providing any recommendations. Then, the accuracy of the provided recommendations largely depends on the quality and accuracy of the user representation in the UM. Hence, recommender systems may benefit from enriching their UMs by importing partial UMs built by other recommender systems and integrating them with the locally available UMs. This process is referred to as the *mediation* of the UMs (Berkovsky et al. 2008).

The main goal of the UM mediation process is to acquire partial UMs built by various recommender systems, and to convert and integrate them into a single UM, as needed by a target system for providing personalized recommendations to its users. An analysis of the state-of-the-art in recommendation techniques has yielded the definition of *experience* as the core UM data representation unit, referring to the connection of three components: *user*, *item*, and *context* (Berkovsky et al. 2008). An experience represents either an explicit or an implicit user evaluation for an item in a particular contextual condition. Based on this representation, there are four groups of experiences that may be valuable for the mediation process:[1]

[1] Although other experiences may be valuable for the UM mediation, our research focuses on the mediation of the first two groups, since the mediation of the other two groups requires several inference mechanisms to be applied, which may 'deteriorate' the original UM data.

- Experiences of the same user for the same item in the same context, where certain experience components may be *represented* in different ways.
- Experiences *differing* only in *one component* These include experiences of another user for the same item in the same context, experiences of the same user for another item in the same context, and experiences of the same user for the same item in another context.
- Experiences *differing in two components* These include experiences of the same user for another item in another context, experiences of another user for the same item in another context, and experiences of another user for another item in the same context.
- Experiences where the values of *all three components are different*, i.e., experiences of another user for another item in another context.

This work deals with a combination of the first and second groups of experiences, i.e., the cases where the mediation involves experiences of the same user for the same or different items in the same contextual conditions, whereas some of the experience components may be represented in different ways. Although such experiences may be valuable for the UM mediation, their mediation requires possible heterogeneities in the representations of the experience components to be resolved (Bernstein and Melnik 2004), e.g., structural heterogeneity, synonymy, or multilingualism. To overcome the heterogeneities, mechanisms capable of identifying the relationships between the heterogeneous representations of the experiences must be applied. This mediation is referred to in the rest of the paper as *cross-representation mediation*. For example, consider cross-representation mediation between two movie recommender systems, storing the evaluations of possibly overlapping sets of movies, provided by the same user using the two systems independently. The first stores a user's preferences as a list of ratings for movies, given on a discrete scale from 1 (horrible) to 5 (excellent), while the second stores the preferences of the same user as aggregated ratings for movie genres.

In particular, this work focuses on a specific type of cross-representation UM mediation: from a collaborative filtering (CF) to a content-based (CB) recommender system. In collaborative filtering systems, the UMs are represented by explicit ratings provided by the users for a set of items managed by the system (Herlocker et al. 1999). Note that content features of neither the users nor the items, besides their unique identities, are typically stored by collaborative filtering systems. Conversely, in content-based systems, the items are represented by features that characterize the items, and the UMs are typically represented by weights assigned to these features and representing the user's degree of preference for these features (Morita and Shinoda 1994; Pazzani 1999; Billsus and Pazzani 2000).

Mediation between these two types of UMs requires identification of regularities among the features of positively or negatively rated items, where the ratings for the items are derived from the collaborative filtering UM. As no item features are originally stored by the collaborative filtering recommender systems, the features describing the rated items should be extracted from an external domain knowledge base. The user features are then assigned numeric values reflecting the ratings given by the user. As a result, a set of ratings stored in the collaborative filtering UM is mediated to the weighted list of features liked/disliked by the user and integrated with the existing

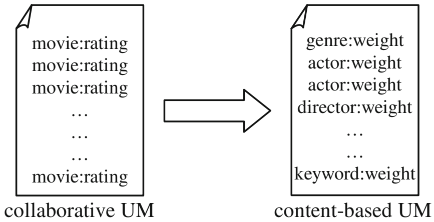

Fig. 1 Collaborative filtering to content-based mediation of user models

weights of the features stored in the content-based UM. Figure 1 schematically shows the mediation of collaborative filtering to content-based UMs for the domain of movies.

The proposed mediation mechanism was implemented and its accuracy was evaluated using the EachMovie (McJones 1997) movie ratings dataset. The IMDb database (The Internet Movie Database, http://www.imdb.com) was exploited for extracting the features of the rated movies, such as genres, actors, directors, etc. The UM mediation was accomplished by converting the collaborative filtering ratings from the source system to the degrees of the user's preference for the item features. The conversion was based on the assumption that users' ratings for the movies reflect their preferences regarding certain content features of the movies, e.g., the preferred movie genre, or disliked actors or directors. As the content-based UMs were generated, they were exploited in order to generate content-based recommendations, which allowed experimental evaluation of the mediation.

The evaluation compared the accuracy of the generated recommendations, using the commonly used Mean Absolute Error (MAE) (Herlocker et al. 2004). The results demonstrate that content-based recommendations can achieve a high level of accuracy. Moreover, in some cases, for users with sparse collaborative filtering UMs, the content-based recommendations generated exploiting the mediated content-based UMs outperform the recommendations generated exploiting the original collaborative filtering UMs. The evaluation also showed that incrementally mediating larger and larger parts of the UMs from the source collaborative filtering system and integrating them with the UMs already available in the target content-based system increases the accuracy of the predictions. This proves the usefulness of the proposed cross-representation mediation approach and demonstrates its applicability in a setting where independent recommender systems exchange their UM data, while continuing to use their original recommendation techniques.

The rest of the paper is organized as follows. Section 2 briefly presents prior research efforts on the mediation and integration of UMs. Section 3 describes the proposed approach for cross-representation UM mediation and elaborates on the mediation of collaborative filtering UMs to content-based UMs. Section 4 presents the experimental evaluations conducted and analyzes their results. Finally, Sect. 5 concludes and presents several directions for future research.

2 Mediation and integration of user models

Centralized construction of the UMs as an aggregation of partial UMs stored by individual personalization systems was proposed in Kay et al. (2003). To use such centralized UMs, each personalization system maintains a dedicated mechanism capable of extracting and updating the relevant parts of the centralized UMs. A similar approach was discussed in Niederee et al. (2004), proposing the use of the Unified User Context Model (UUCM) for aggregating the partial UMs collected by individual personalization systems. To provide personalized services, each system extracts the required UM data from the UUCM, delivers the service, and updates the UUCM. However, the centralized aggregation of the UMs in both systems poses several security and privacy challenges, since a malicious user could, just with a single successful attack on the centralized system, access all the centralized UM data.

A different approach was proposed in Berkovsky et al. (2008). There, the UM data aggregation task was accomplished by a UM *mediator*, capable of importing and integrating partial UMs collected by a decentralized group of personalization systems. The mediator provides a scalable platform for privacy-enhanced UM data exchange and allows an ad hoc generation of the UMs for a target system through importing and integrating partial UMs built by other systems. Hence, the mediator facilitates bootstrapping UMs for personalization systems where no UMs exist, or enriching the existing UMs, thus leveraging the quality of the personalized services provided to the users. The approach described in this work provides a concrete example of such a mediator-based process in the specific case of mediation between collaborative-filtering and content-based recommender systems.

GUMO is a comprehensive set of UM ontologies that allows uniform interpretation of distributed UMs in intelligent environments (Heckmann et al. 2005). GUMO simplifies UM data exchange between various personalization systems and paves the way for overcoming the problem of syntactical and structural heterogeneities of specific UMs. Following GUMO, (Lorenz 2005) proposed an open architecture for agent-based sharing of partial UMs, where the agents manage locally collected UMs that are centrally integrated into the global UM. However, neither the sharing policy nor the conversion mechanisms between various UM representations were defined, such that the practical UM sharing between any two systems is still an open problem.

In the domain of recommender systems, prior research tried to integrate multiple recommendation techniques in a single recommendation generation process. These systems are referred to in the literature as *hybrid* recommender systems (Burke 2002). Hybrid recommender systems typically combine the outputs or the techniques of several individual recommender systems to build a new recommender and in this way improve the accuracy of the generated recommendations. A classification of possible hybridization techniques was presented in Burke (2002), where seven types of hybrid recommenders were introduced and compared: weighted, mixed, switching, cascade, feature combination, feature augmentation, and meta-level. Recently (Zanker and Jessenitschnig 2008), showed many notable examples of hybrid recommender systems exploiting a wide range of personalization techniques. The results showed the benefits of exploiting knowledge-based, utility-based, association rules-based, and collaborative-based recommendation techniques in the construction of a UM, based solely on explicit requirements collected from the user forms.

It is important to stress that the ultimate goal of hybrid recommenders is, given two existing recommender systems, e.g., a content-based and a collaborative system, to generate a new third recommender, e.g., a hybrid content-based/collaborative system. From our perspective, a more abstract classification of hybrid approaches should comprise two main groups, according to the *stage* of the recommendation generation process where the hybridization occurs:

- If the hybridization occurs at the end of the recommendation process, i.e., at the *output* level, complete recommendations computed by the source systems are integrated and/or presented to the user. This group comprises the mixed, weighted, switching, and cascade approaches. For example in the mixed approach (Balabanovic and Shoham 1997), recommendations computed by collaborative and content-based systems are combined together and shown to the user. In order to perform hybridizations in this group, additional information related to the source systems may be necessary. For instance, to apply a switching approach, a measure of the confidence in the recommendations produced by the source systems may be necessary. However, the hybridized systems in this group use the original recommendation technologies of the source systems without modifying them. This hybridization requires a loose coupling of the source systems and is typically relatively simple to implement.
- The hybrid approaches in the second group operate at the level of the *recommendation techniques* or *models* used by the source systems. The feature augmentation, feature combination and meta-level approaches fall into this group. For example (Basu et al. 1998) achieves the content-based/collaborative merger by treating collaborative information (i.e., ratings of users) as additional feature data associated with each example and using content-based techniques over this augmented data. The authors apply the inductive rule learner to the task of recommending movies using both the user ratings and content features. In this case, the hybrid system integrates and modifies a content-based prediction technique with the data coming from a collaborative system. Another example of a hybrid system in this group was proposed in Pazzani (1999), where a collaborative technique is modified, such that the content-based UMs generated by a content-based technique are used for computing user-to-user similarities.

In this work we address a different form of system integration, where neither the output of the system nor the internal techniques or models are exchanged, integrated, or modified. We are interested in exchanging the background data of the recommendation generation process, i.e., the UMs, between the systems *without modifying any of the original systems*. In particular, this work deals with cross-representation mediation of UM data from a collaborative to a content-based recommender system. That is, the recommendation generation mechanism used by the target system is pure content-based. It can be applied with or without the UM mediation, and it exploits standard content-based UM data. The case that we are studying occurs when UM data coming from a collaborative filtering system are sent to a content-based system to enrich its UMs, i.e., its background data. Hence, the proposed approach cannot be rigorously defined as purely hybrid; it can be considered as a particular case of *background data mediation*, where the mediated data are the UM data. Moreover, it should be stressed that mediation of the UM data in the context of the mediation framework presented in Berkovsky et al. (2008) is more flexible than the data hybridization methods presented in Burke (2002). In a mediation scenario, the UM data arriving to a target system may be coming from various systems using different recommendation techniques and the mediation implies the application of ad hoc dynamically selected mediation modules converting the UM data from the sources to the target system, whereas classical hybridizations integrate specific techniques and approaches.

In summary, this work demonstrates the feasibility and usefulness of the generation of recommendations using a single content-based filtering recommendation technique, but with (1) *multiple sources of UM data*, i.e., both collaborative filtering and content-based UMs, and (2) application of a *specific cross-representation mediation* scenario, i.e., mediation that uses parts of the collaborative UMs to bootstrap and/or enrich the content-based UMs.

3 Collaborative filtering to content-based mediation

Collaborative filtering is one of the most widely-used recommendation techniques. It recognizes cross-user correlations and generates recommendations for items by weighting the opinions of similar users (Herlocker et al. 1999). A collaborative filtering algorithm typically operates according to the following three stages: (1) in the *similarity computation* the similarity of the active user, i.e., the user who requested the recommendation, to all the other users is computed; (2) in the *neighborhood formation* the set of K users, most similar to the active user, is identified; and (3) in the *recommendation generation* a prediction of the ratings of the active user for items is computed by combining the ratings of the selected users for these items. In short, collaborative filtering systems recommend items that were liked in the past by other users similar to the active user.

The input for the collaborative filtering is a matrix of users' ratings for a set of items. In this matrix, each row represents the ratings of a single user and each column represents the ratings for a single item. Thus, collaborative filtering UMs are represented by the user's ratings, i.e., fixed-size lists of pairs

$$UM_{CF} = \{i_1 : r_1, i_2 : r_2, \ldots, i_n : r_n\} \tag{1}$$

where every entry $i_k : r_k$, corresponds to the rating r_k provided explicitly or implicitly by the user for the item i_k. If the user's rating for an item is not available, then a special *null* value is used, such that the UMs in collaborative filtering systems typically store ratings for a small subset of the full set of items managed by the system. Moreover, collaborative filtering systems typically do not store any item- or user-related content features, besides their unique identities.

Content-based filtering (Morita and Shinoda 1994) builds personalized recommendations by taking as input: (1) a list of features (usually terms extracted from a textual description) describing the items in a given domain, possibly weighted according to a predefined scale; (2) a set of weights explicitly or implicitly assigned by the user to the above list of features; and (3) the set of available items C, which have not yet been rated by the user, and are candidates for the recommendations. The output recommendation is a subset of C, containing the items whose features match the features preferred by the user. In short, content-based systems recommend items similar to the items that were positively rated in the past by the user.

Thus, in content-based recommender systems the UMs are represented as a weighted list of domain features

$$UM_{CB} = \{f_1 : w_{f(1)}, f_2 : w_{f(2)}, \ldots, f_n : w_{f(n)}\} \tag{2}$$

where f_k denotes a certain domain feature and $w_{f(k)}$ denotes the corresponding weight of this feature. It should be noted that for the generation of content-based UMs the information about the user's preferences represented by the user's ratings is typically transferred to the feature weights. This can be done either by applying various machine learning techniques or by computing the centroid of the feature-based representation of the items rated by the user (Billsus and Pazzani 2000). Note the heterogeneity of the content-based UM representation, as the features of items are largely dependent upon the domain of the system. For example, features useful for a music recommender system will not be useful for a travel recommender system, and vice versa. Even within the same application domain the features may vary across different systems, and their representations may be heterogeneous. For example, the feature weights may express a binary positive or negative evaluation or a fine-grain evaluation between 0 and 1.

This work aims at developing a mediation mechanism capable of converting the collaborative filtering UMs (or their parts), represented by a set of ratings explicitly given by a user, to the content-based UMs, represented by a set of features and their corresponding weights. We stress that the mediation mechanism works on an individual user-by-user basis, i.e., only the collaborative-based UM of the active user is required to generate or modify his/her content-based UM. The rest of this section elaborates on the cross-representation UM mediation applied in the domain of movies. Initially it presents the UM mediation mechanism, and then it discusses the details of fine-tuning the content-based prediction mechanism. Although the discussion below focuses on the domain of movies, the proposed mediation approach can be applied in a similar manner also to other domains.

3.1 User models mediation and content-based recommendations

In the movies domain, a collaborative filtering UM comprises a set of movies and their respective ratings explicitly provided by the user, as shown in (1). For example, consider the following $UM_{CF} = \{$"*The Lord of The Rings*":1, "*The Matrix*":0.8, "*Psycho*":0.2, "*Friday the 13th*":0, "*Star Wars*":0.9, "*The Nightmare on the Elm Street*":0.1", "*Alien*":0.9$\}$, where the movie ratings are given on a continuous scale ranging between 0 and 1. Although this collaborative filtering UM represents the user with a set of ratings only, it can be easily recognized that the user likes science-fiction movies and dislikes horror movies. Hence, the content-based UM of this user may be $UM_{CB} = \{$*science-fiction*:0.9, *horror*:0.1$\}$, where the genre weights are computed as an average of the ratings for the movies from this genre. Similarly to the genre weights, the weights of other movie features, such as directors, producers, and actors, can be computed using the ratings.

To generate the content-based UMs and to handle the mediation of the collaborative to content-based UMs, a movie knowledge base is needed for extracting some features of the movies, such as the lists of genres, actors, directors, and so forth. In this work, an offline version of the IMDb movie database (http://www.imdb.com) served as the mediation knowledge base. The IMDb provides movie information from 49 feature categories, such as genres, actors, directors, writers, cinematographers, composers, keywords, languages, and many others. For the sake of simplicity, only seven feature categories were exploited in this work: *genres*, *keywords*, *actors*, *actresses*, *directors*, *production countries*, and *languages*, as these categories are expected to have the strongest impact on the user's decision in selecting, seeing, and rating a movie (Tintarev 2007).

The generation of a content-based UM and the mediation of a collaborative filtering UM exploits as input the user's ratings for a set of movies and the content features of the rated movies extracted from the IMDb. In the collaborative filtering to content-based mediation, a positive rating given by the user for a movie increases the weights of the content features of that movie in the mediated content-based UM, whereas a negative rating decreases them. The update of the content feature weights is repeated for all the rated movies in the collaborative filtering UM. This yields a content-based UM that can be used to generate pure content-based recommendations. The details of the mediation process are fully illustrated in the rest of this section.

It should be stressed that the numeric rating assigned by a user to a movie is a subjective evaluation, i.e., depending on the user's personal preferences and evaluation scale. With respect to the evaluation scale, consider the case where two users provide the same rating, e.g., three, for a movie. If the first user has an average rating of 2, and the second 4, then the first is an expression of a more positive evaluation than the second. Hence, the values of all the ratings should be normalized in order to eliminate the impact of users' individual evaluation scales. This is done by subtracting the average rating of the user from each provided rating:

$$r'_{ui} = r_{ui} - r_u \tag{3}$$

where r'_{ui} denotes the normalized rating value given by user u for movie i, r_{ui} denotes the original rating, and r_u denotes the average rating of the user computed over all the available ratings provided earlier by the user. Hence, when a movie is assigned a rating above the user's average rating, it is treated as a positive rating. Conversely, a rating below the user's average is treated as a negative rating.

The main assumption behind the collaborative filtering to content-based UM mediation is that the ratings in a collaborative UM implicitly reflect user's preferences regarding certain features of the movies, such as genre, director, or actors. However, a single rating cannot reliably determine the features preferred by the user, and all the ratings given by the user must contribute to the computation of the features weights in the UM. This is achieved by considering all the available movie ratings in the collaborative filtering UM, extracting the movie features belonging to one of the above seven categories from the IMDb, and increasing[2] the weights of these features in the content-based UM by the normalized movie rating r'_{ui}.

[2] The weights may decrease if the rating is lower than the average rating of the user.

For example, consider a rating $r_{ui} = 0.9$ on a scale between 0 and 1, given by a user whose average rating is $r_u = 0.6$ to the movie "*Star Wars*". The normalized value of such a rating is $r'_{ui} = 0.3$. According to the IMDb, the *genres* of "*Star Wars*" are *action*, *adventure*, *fantasy,* and *science-fiction*. Hence, the weights of these genres are increased by 0.3. Also, the weights of the movie *director*, all the *actors*, and *actresses* involved in the movie, and the other movie features are increased by 0.3. Note that the feature weights can increase or decrease depending on the positive or negative normalized rating. In general, features appearing in the movies having positive normalized ratings will normally have positive weights in the UM, whereas features appearing in the movies having negative normalized ratings will have negative weights.

More formally, the weight $w^u_{f(i)}$ of a feature f_i occurring in the movies rated by a user is computed by the following formula:

$$w^u_{f(i)} = \frac{\sum_{j=1}^{M} r'_{uj}\delta_{ij}}{\sum_{j=1}^{M} \delta_{ij}} \tag{4}$$

where M is the total number of movies rated by the user u, r'_{uj} is the normalized rating of the user u on movie j, and δ_{ij} is a 0/1 indicator function, which is 1 if the feature f_i is present in the movie j, and 0 otherwise (i.e., the weight is normalized by the frequency of the feature).

As we mentioned, the weights $w^u_{f(i)}$ of the features occurring in movies with positive (negative) ratings tend to increase (decrease). Conversely, $w^u_{f(i)}$ of the features occurring with similar probability both in positively and negatively rated movies, and of the features appearing infrequently in the movies rated by the user u, will remain close to their initial value. In order to take this into account, the frequency $c^u_{f(i)}$ of each feature f_i, i.e., the number of movies that have been rated by the user u and contain the feature f_i, is recorded.

In the mediation process, collaborative to content-based conversion is repeated for all the ratings available in the collaborative filtering model UM_{CF}. When the mediation is completed, the content-based UM of a user u shown by formula (2) contains a set of features belonging to the seven categories (*genres*, *keywords*, *actors*, *actresses*, *directors*, *production countries,* and *languages*) and their computed feature weights. The number of features in the UM depends on the number of movies rated by the user and the number of features present in the descriptions of these movies. For example, the number of features in these 7 categories is 213 for "*Star Wars*" and 116 for "*Psycho*". In general, the number of features in the content-based UM increases with the number of movies rated by a user. Hereafter, the notation $F(u)$ denotes the list of the features, whose weights were computed for user u and this is a subset of the full set of F available features.

Note that the proposed approach for feature weight computation is applicable both for bootstrapping empty content-based UMs and for enriching the existing ones. In both cases, if a feature f_i is not stored in the content-based UM, its weight $w^u_{f(i)}$ is initialized with the normalized movie rating r'_{ui} and the frequency $c^u_{f(i)}$ is set to 1. Otherwise, the content-based UM is incrementally updated: the weight $w^u_{f(i)}$ of f_i is increased by the normalized rating r'_{ui} and the frequency $c^u_{f(i)}$ is increased by 1. Such an update of the existing content-based UM assigns uniform weights to the UM data stored by the content-based system and to the data coming from the collaborative system. This may be an over simplification, since the UMs in the two systems may differ in accuracy, confidence, or freshness. More accurate ways for updating the existing content-based UM will be investigated in the future.

Now we discuss the rating prediction method of the content-based recommender system. Given a movie m, which has not yet been rated by the user u, a predicted rating for m is generated by (1) extracting from the IMDb the set $F(m)$ of all the features of m appearing in the movie content description, and (2) computing the rating prediction r_{um} of user u for movie m as a weighted average of the weights $w^u_{f(i)}$ of the features that appear both in the content-based UM of a user u, i.e., in the user's set of features $F(u)$, and in the content description $F(m)$ of the movie m:

$$r_{um} = \frac{\sum_{j \in F(u) \cap F(m)} w^u_{f(j)} c^u_{f(j)}}{\sum_{j \in F(u) \cap F(m)} c^u_{f(j)}} \tag{5}$$

As can be seen, the rating prediction r_{um} is computed by a weighted average of the feature weights, where the relative importance of each feature is determined by the frequency of that feature in the movies rated by the user.

Depending on the personalization task, the system will use the predicted ratings for movie recommendations. For example, if the system should recommend one movie, then the rating prediction r_{um} is generated for every unrated movie and the movie with the highest predicted rating r_{um} is recommended to the user. If a set of movies should be recommended, then the movies are sorted according to their predicted ratings and top-N movies are recommended to the user.

It should be stressed that the recommendations are generated solely on the base of content-based UM, possibly derived from collaborative filtering UM data, and the prediction method follows a simple content-based approach. Being a pure content-based recommendation approach, the proposed mediation resolves the problems of collaborative filtering recommender systems. For instance, content-based systems do not suffer from the *new item problem* (McNee et al. 2003) of the collaborative filtering systems, where accurate recommendations for a new item cannot be generated unless the system obtains a sufficient number of ratings for that item. Nevertheless, being a pure content-based recommendation approach, recommendations may lack *serendipity* (McNee et al. 2006). In fact, a content-based system can recommend only movies that are similar to the movies already rated by the user and cannot provide 'surprising' recommendations for new types of movies.

3.2 Fine-tuning of the prediction mechanism

Since the IMDb knowledge base contains extensive content data for each movie, content-based UMs built from collaborative UMs consisting of only a few ratings may comprise thousands of features. Some of these features may be important and clearly reflect user's preferences, while many may be irrelevant or even *noisy* features and may hamper the accuracy of the generated recommendations. Two issues should be resolved in order to improve the accuracy of the recommendations: (1) identification of the noisy features that hamper the accuracy of the prediction mechanism and should be ignored by the prediction mechanism, and (2) identification of the categories of features that are more important for the recommendation generation.

In this work we refer to two categories of irrelevant features:

- *Low-frequency features* These are features f_i that have a low frequency $c^u_{f(i)}$ among the movies rated by the user u. In the domain of movies, these features may represent, for instance, actors playing marginal roles in the movies. Although their relative importance $c^u_{f(i)}$ in the predicted rating computation shown by formula (5) is low, the number of such features increases quickly with the number of rated movies in the collaborative filtering UM. Hence, a large number of such features may outweigh the more important features and decrease the accuracy of the generated recommendations.
- *Neutral features* These are the features f_i, to which the user is indifferent, i.e., features of no special importance to the user. As such, these features are sometimes assigned positive and sometimes negative values and, when the content-based UM is generated, the weight $w^u_{f(i)}$ of these features is close to 0. However, if their frequency $c^u_{f(i)}$ is high, the impact of their weights in the predicted rating computation shown by formula (5) may increase. Moreover, similarly to the low-frequency features, a large number of such features may outweigh important features and deteriorate the accuracy of the recommendations.

To minimize the impact of these two types of irrelevant features, two thresholds were defined. The *min-occur* threshold denotes the minimal frequency $c^u_{f(i)}$, for which a feature f_i is taken into account by the prediction mechanism. It is designed to eliminate the impact of the low-frequency features by considering only the features whose frequency $c^u_{f(i)}$ is above the *min-occur* threshold. The *conf* feature denotes the minimal feature weight $w^u_{f(i)}$, for which a feature is taken into account by the prediction mechanism. It is designed to eliminate the impact of the neutral features by considering only the features whose weight $w^u_{f(i)}$ is above the *conf* threshold. Note that these two thresholds are independent, and in order to be taken into account by the prediction mechanism, a feature weight and frequency should be greater than both of them.

The second issue deals with determining the important feature categories, and it can be resolved using a feature selection approach (Kohavi and John 1997). Feature selection is defined as follows: "given an inducer I and a dataset D with a set of features $\{X_1, X_2, \ldots, X_n\}$, an optimal feature subset $\{X_{i1}, X_{i2}, \ldots, X_{im}\}$, where $m \leq n$ is a subset of the features such that the accuracy of the induced classifier I using this set of features is maximal." In this case, the inducer I is the content-based prediction mechanism and D is the IMDb with the initial $n = 7$ feature categories: *genres*, *keywords*, *actors*, *actresses*, *directors*, *production countries*, and *languages*.

Most of the feature categories contain a large number of features. Hence, instead of addressing the problem of selecting the features from each category, we focus on the selection of categories of features. In other words, the search process is limited to the subset of all the possible subsets of features, where the features from a certain feature category are either all present or all absent. The motivation for such a simplification is the large number of features (more than 50,000), which makes a full search

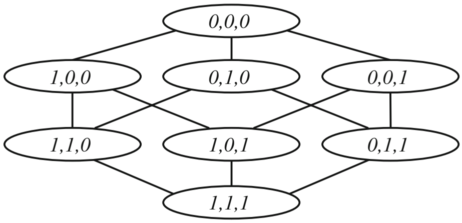

Fig. 2 Feature selection search space

```
Hill-climbing (Initial-state s, Evaluation-function eval)
 (1) let v=s
 (2) expand v: generate all v's children states
 (3) compute eval(w) for each child w of v
 (4) let v'= the child w with the highest eval(w)
 (5) if eval(v')>eval(v)
 (6)   v=v'
 (7)   goto step(2)
 (8) return v
```

Fig. 3 Hill-climbing heuristic search for the feature selection

of the best features subset impractical. Hence, the goal of the feature selection is to select the feature categories that should be taken into account by the content-based prediction mechanism for the recommendation generation, while ignoring the other categories.

The wrapper feature selection approach (Kohavi and John 1997) conducts the selection as an automated search in the space of states, where each state consists of $n = 7$ bits, representing a certain combination of categories that are taken into account for the recommendation generations. For example, consider a state S represented by *genres* = 1, *keywords* = 0, *actors* = 1, *actresses* = 1, *directors* = 0, *production countries* = 0 and *languages* = 0. This means that the categories of *genres*, *actors*, and *actresses* are taken into account for the rating prediction computation, whereas *keywords*, *directors*, *production countries*, and *languages* are ignored. For the sake of clarity, the order of feature categories is considered fixed and the states are denoted by their respective binary vectors, e.g., $S = (1, 0, 1, 1, 0, 0, 0)$. Figure 2 shows the search space for $n = 3$ as a graph of states, where the edges indicate addition or deletion of a certain feature category.

The size of the search space is $O(2^n)$ states. The goal of the search is to find the state having the greatest inducer evaluation accuracy, i.e., the highest accuracy of the generated recommendations. Since the size of the search space is exponential and the task of evaluating the accuracy of the recommendations in each state is relatively expensive, applying an exhaustive search is impractical. Hence, a heuristic search over the space of states is applied. In this work, a hill-climbing heuristic search algorithm was applied (Russell and Norvig 1995). Pseudo-code shown in Fig. 3 outlines the search.

The algorithm starts with the initial search state representing the initial combination of features categories as the current node (step 1). Then, it expands the current node by generating its children states by either adding or removing feature categories (step 2), and evaluates each one of the children states by computing the accuracy of the recommendations for the respective combination of features categories (step 3). Then, the state with the highest evaluation is chosen as the current state (step 4), and the

```
Recommend (Content-based-UM u, set-of-unseen-movies M)
(1)  foreach m∈M
(2)    retrieve F(m)=set of m features from the categories
                      chosen by the feature selection
(3)    for each f(i)∈F(m)
(4)      if f(i)∈ F(u) AND |w^u_f(i)|>conf AND c^u_f(i)>min-occur
(5)        take f(i) into account for the prediction of r_um
(6)    compute r_um as shown by formula (5)
(7)  return m with maximal predicted r_um
```

Fig. 4 Fine-tuned content-based recommendation generation

algorithm iteratively repeats the steps of the current state expansion and evaluation until the evaluation values of the current state improves (step 5–7). Finally, the algorithm returns the state with the highest evaluation, i.e., the combination of feature categories where the accuracy of the generated recommendations was maximal (step 8). The returned state denotes the subset of feature categories that should be taken into account for the predicted rating computation.

In conclusion, the revised prediction mechanism is modified to take into account only the features from the categories selected by the feature selection algorithm, whose frequency $c^u_{f(i)}$ in the content-based UM is above the *min-occur* threshold, and whose weight $w^u_{f(i)}$ is above $+$*conf* or below $-$*conf*. Pseudo-code shown in Fig. 4 describes the details of the fine-tuned recommendation generation process.

The goal of the above algorithm is to recommend a movie among a set of potentially recommendable movies. Hence, it generates a separate rating recommendation for each one of the movies (step 1). For this purpose, the set of movie features from the categories chosen by the feature selection is extracted from the IMDb (step 2). A feature f_i is taken into account by the prediction mechanism only if its frequency $c^u_{f(i)}$ is above the *min-occur* threshold and the absolute value of its weight $|w^u_{f(j)}|$ is above the *conf* threshold (step 3–5). Finally, the predicted rating of each recommendable movie is computed (step 6) and the movie with the highest predicted ratings is recommended to the user (step 7).

It should be noted that the proposed prediction mechanism assigns uniform weights to the features from all the categories and incorporates no additional weighting factor that reflects the importance of a particular category for the user. In fact, for some users certain categories could be more important and therefore should affect the rating prediction more strongly. For example, the movie *director* may be very important for a user, while the *actors*, *actresses*, and the other categories may be less important. Hence, the *directors* category should be assigned a greater weight than the other categories. Moreover, in real-life situations, the user's ratings may depend on a combination of features from several categories, e.g., the user may like only the movies in a certain *language*, from a certain *genre*, and directed by a certain *director*. While highlighting the importance of the category weighting and discovering the dependencies between the features, we focus on the feature categories selection task. This restriction is reasonable, since after the categories are selected, the fine-grained weights of the specific features within the categories can be computed. The other weighting issues that we mentioned above remain beyond the scope of this work and are referred to as future research directions.

Table 1 Distribution of ratings among the users in the dataset

Rated movies	Under 25	26–50	51–75	76–100	101–125	126–150	151–175	176–200	201–225	226–250	251–300	301–500	Over 500
Number of users	17,321	13,788	6,514	3,609	2,302	1,349	887	609	441	327	358	436	47
Percentage of users	36.09	28.73	13.57	7.52	4.80	2.81	1.85	1.27	0.92	0.68	0.75	0.91	0.098

4 Experimental results

The above collaborative to content-based mediation of UMs was evaluated using the EachMovie dataset (McJones 1997). EachMovie is a collaborative filtering dataset, storing 2,811,983 ratings of 72,916 users on 1,628 movies. The ratings are given on a discrete scale between 0 and 1, and the possible ratings are 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0. EachMovie provides for every movie a URL to the movie description in the IMDb. As some URLs were outdated or invalid, in our evaluation we could use only a set of 1,529 movies with valid URLs. For these movies, we identified a set of 47,988 users who rated more than 10 movies with rating variance diverse from 0. In total, we obtained 2,667,605 ratings, yielding a rather sparse dataset with ratings density of 3.64%.

We analyzed the distribution of the users in the dataset according to the number of movies rated by them. For this, we partitioned all the available users into 13 bins, according to the number of rated movies: under 25, 26 to 50, 51 to 75, 76 to 100,…, 201 to 225, 226 to 250, 251 to 300, 301 to 500 and over 500 movies. Table 1 shows the distribution of users among the bins. As can be seen from the table, most of the users in the dataset rated a relatively small number of movies. For example, 64.83% of the users rated fewer than 50 movies, 78.4% of the users rated fewer than 75 movies, and 85.92% of the users rated fewer than 100 movies.

For the collaborative filtering to content-based mediation of UMs, we used an off-line version of the IMDb dataset. Although the IMDb provides movie information in 49 feature categories, as mentioned earlier, we restricted the mediation mechanism to the following seven feature categories: *genres*, *keywords*, *actors*, *actresses*, *directors*, *production countries,* and *languages*. To analyze the statistical properties of various feature categories, we computed for each category the overall number of features that occur in the descriptions of the above 1,529 movies. Table 2 shows the number of features in every feature category. Normally, a movie belongs to only a few *genres*, *production countries*, and *languages*. Hence, the overall number of features in these categories is relatively small. Conversely, as the number of *keywords*, *actors*, *actresses*, and *directors* for every movie may be high, the overall number of features in these categories is significantly higher.

The first part of the experiments (i.e., fine-tuning of the thresholds and the feature selection) was conducted on a smaller dataset, hereafter referred to as the *FT* set. The *FT* set contains the ratings of 1,000 users who rated at least 100 movies. It comes out that in *FT* every user rated 168.94 movies on average. For each user in the *FT* set, 90%

Table 2 Number of features in various features categories

Category	Genres	Keywords	Actors	Actresses	Directors	Countries	Languages
Features	24	8,993	29,543	14,139	1,111	60	73

of the ratings were defined as the training set and the remaining 10% were used as the testing set. The ratings in the training set served as input for the collaborative filtering to content-based UM mediation. Then, the generated content-based UM was used for generating content-based recommendations for the movies in the testing set. Hence, the overall number of the generated recommendations in the experiments conducted over the *FT* set was 16,894. Predictive accuracy of the generated content-based recommendations was evaluated using the Mean Absolute Error (MAE) (Herlocker et al. 2004):

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{6}$$

where N denotes the overall number of the generated recommendations, p_i denotes the predicted rating computed by formula (5), and r_i denotes the real rating provided by the user for movie i. Hence, the MAE compares the predicted rating values with the real ratings given by the users for the movies.[3]

4.1 Fine-tuning the thresholds and feature selection

The goal of the first set of experiments was to fine-tune the prediction mechanism. We considered two tasks: (1) selecting the most appropriate values for the *conf* and *min-occur* thresholds, and (2) applying the feature selection wrapper approach for selecting the feature categories that should be taken into account by the prediction mechanism. To simplify the fine-tuning of these parameters, the two tasks were performed sequentially, even if an optimal parameter selection should identify the best combination of these parameters.

The first fine-tuning task focused on determining the most appropriate values for the *conf* and *min-occur* thresholds. Since the two thresholds are independent, when determining their most appropriate values, one threshold was set to a constant value and the other was gradually modified. For each value of the modified threshold, the training subset of *FT* set served as input for the collaborative filtering to content-based UM mediation. The generated content-based UMs were used for generating content-based recommendations for the movies in the testing set (10% of the *FT* set). For each value of the *conf* and *min-occur* thresholds, the predicted accuracy of the recommendations using the given threshold values was evaluated using the MAE.

[3] As shown by formulae (3) and (4), all the available ratings are normalized to the range $[-1, 1]$. Hence, both the predicted values computed by formula (5) and the normalized real ratings used for the MAE computation are in the range $[-1, 1]$.

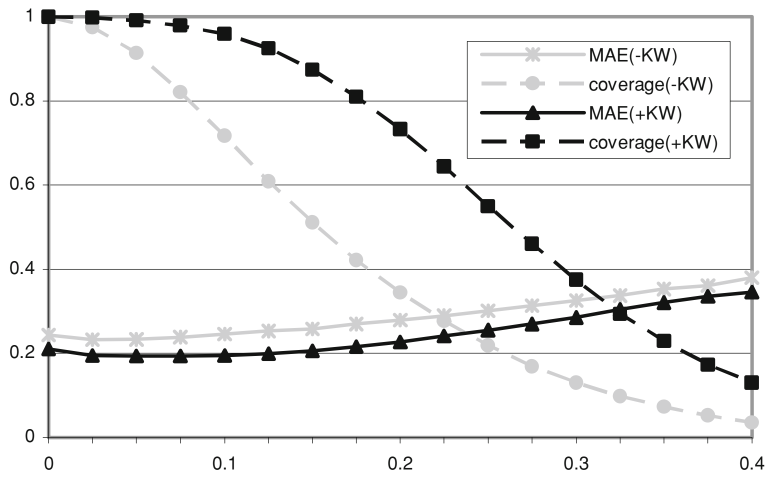

Fig. 5 MAE and coverage versus *conf* threshold

Table 3 Percentage of filtered features for various values of the *conf* threshold

conf	0.0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
Filtered features (%)	0.0	28.94	49.37	64.88	74.97	83.68	88.96	92.73	95.13

To find the most appropriate value of the *conf* threshold, the *min-occur* threshold was set to one for all the categories (i.e., a feature should occur at least in one movie rated by the user), and the values of *conf* threshold (i.e., the minimal weight of a feature) were gradually modified from 0 to 0.5. To provide an initial assessment of the relative importance of different categories, the recommendations were generated in two ways: (1) using features from all the available seven feature categories, and (2) using features in only six feature categories, excluding the *keywords* category. We note that high values of the *conf* threshold reduce the number of features taken into account by the content-based prediction mechanism. In fact, the predicted ratings for certain movies cannot be computed when there is no feature belonging both to the UM and to the movie model. To check this, for each value of *conf* we computed the *coverage*, i.e., the percentage of movies whose ratings were successfully computed (Herlocker et al. 2004). Figure 5 shows the results of the first experiment. The horizontal axis represents the values of the *conf* threshold, and the vertical shows the MAE and the coverage. The dotted curves show the coverage and the continuous curves the MAE. The dark curves show the results based on all the available seven feature categories, while the light curves are based on six categories, excluding the *keywords* features.

To analyze the experimental results, we present in Table 3 the percentage of features that were excluded when varying the *conf* threshold. This percentage was computed as a ratio between the number of features that were not considered by the prediction mechanism and the overall number of features in the content-based UM.

The results in Fig. 5 show that the MAE initially slightly decreases with the *conf* threshold, but increases after a certain value. These results can be explained by considering the data presented in Table 3. The initial decrease in the MAE can be explained by the removal of the neutral features for low values of *conf*. In fact, when *conf* is 0, the neutral features are not filtered, they produce noise in the prediction mechanism, and the MAE is higher. When *conf* is increased, a growing number of neutral features is filtered. For instance, 28.94% of the features are discarded for *conf* = 0.05, and the MAE decreases. However, even for small values of the *conf* threshold, a very high number of neutral features are filtered. For instance, 74.97% of the features are filtered for *conf* = 0.2, and important features may be discarded. As a result, the MAE increases when the *conf* threshold pass a certain value.

The coverage monotonically decreases with growing values of the *conf* threshold. This is due to the fact that the number of features filtered from the content-based UM monotonically increases with the *conf* threshold, as shown in Table 3. Hence, the number of features that remain in the content-based UM decreases with growing values of the *conf* threshold. This, in turn, decreases the number of overlapping features between the UM and the movie model and aggravates the computation of the predicted movie rating. To optimize both the MAE and the coverage, 0.025 was determined as the most appropriate *conf* value. For this value the MAE is minimal and the coverage is still high, over 0.99.

Considering the role played by the *keywords* features, we note that both the MAE and coverage improve when the *keywords* features are taken into account. In both cases the improvements are statistically significant. The results of the t-test are $p = 0.0239$ for the MAE and $p = 0.0499$ for the coverage. This example motivates feature category selection showing that the *keywords* feature category is important and should be taken into account by the prediction mechanism.

After the most appropriate value of the *conf* threshold was determined, the same experimental methodology was used when choosing the most appropriate value of the *min-occur* threshold, i.e., the minimal number of times a feature must occur in the movies rated by a user in order to be taken into account by the prediction mechanism.[4] The value of the *conf* threshold was fixed, and the value of the *min-occur* threshold was gradually modified to determine the most appropriate value. We used again the *FT* set and mediated the training ratings (90% of the *FT* set) from the collaborative filtering to the content-based UMs. The generated content-based UMs were used for generating content-based recommendations for the testing set (10% of the *FT* set).

We note that the frequencies of the features in different categories have considerable variations. For example, the frequencies of the features in the *genres* category are significantly higher than the frequencies in the *directors* category. This is due to the fact that the number of features in the *genres* category is 24 (see Table 2), and this is significantly smaller than the number of features in the *directors* category, which is 1,111. Since every movie typically belongs to only a few *genres* and is directed by a few *directors*, the frequencies of the *genres* feature are significantly higher than the

[4] In this experiment, the value of the *conf* threshold was first determined, and then applied for the *min-occur* thresholds. Repeating the experiment in the opposite order (determining first the *min-occur* thresholds and applying them for the *conf* threshold) produced similar results.

frequencies of the *directors* features. As no uniform scale for the possible frequencies of the features could be derived, a separate *min-occur* threshold experiment was conducted for each one of the seven feature categories.

To determine the *min-occur* thresholds for the categories, we assumed that the thresholds are independent. Hence, we isolated the impact of the *min-occur* threshold in every feature category by changing only one threshold in every experiment. That is, the values of the *min-occur* thresholds of 6 feature categories were set to 0 and the value of the seventh threshold was gradually modified. The recommendations were generated based on the features from all the categories and the MAE was computed as a function of the *min-occur* threshold. Figure 6 shows the separate results of the experiment for the seven feature categories. Note that, because of the differences in the ranges of the feature frequencies in the categories, the *min-occur* threshold values are represented by relative (ratio between the number of rated movies containing a given feature and the overall number of rated movies) and not absolute (the number of rated movies containing a given feature) values. Hence, the horizontal axis represents the applied relative threshold (in percents) of the *min-occur* parameter, while the vertical axis shows the corresponding values of the MAE.

As can be seen, for most of the feature categories the impact of the *min-occur* threshold is not as strong as that of the *conf* threshold. However, two groups of feature categories with different behaviors can be identified. In the first group, consisting of the *genres*, *production countries*, and *languages* categories, the numbers of possible features are relatively low (24 genres, 60 countries and 73 languages). For this group, the MAE monotonically increases with the *min-occur* threshold. Thus, any feature belonging to these categories is valuable and discarding features from these categories hampers the accuracy of the generated recommendations. Hence, the most appropriate value of the *min-occur* threshold for these categories is 0.

The second group, consisting of the *keywords*, *actors*, *actresses*, and *directors* categories, contains a significantly larger number of features (8,993 keywords, 29,543 actors, 14,139 actresses, and 1,111 directors). For these categories, the MAE initially decreases (slightly). Then, when a certain value of the *min-occur* threshold is reached, the MAE monotonically increases because, similarly to the impact of the *conf* threshold, for the higher *min-occur* threshold some important features are discarded.

Table 4 summarizes the most appropriate values of the *min-occur* threshold for various feature categories. The table shows the minimal percentage of rated movies, in which a certain feature should occur in order to be taken into account by the content-based prediction mechanism. For example, consider the *actors* feature category and a user who rated 500 movies. The *min-occur* threshold of 1.6% means that if a certain actor participated in at least 1.6% of movies rated by a user, i.e., in eight movies (or more), this actor will be taken into account by the prediction mechanism, and will be ignored otherwise. On the contrary, for the *genres* category, the *min-occur* threshold is 0. This means that all the available features from this category will be taken into account by the prediction mechanism.

After the optimal values of the *conf* and *min-occur* thresholds were determined, we proceeded to the second task, i.e., selecting the feature categories that should be taken into account by the prediction mechanism. For this task we implemented the above mentioned wrapper feature selection approach (Kohavi and John 1997). The

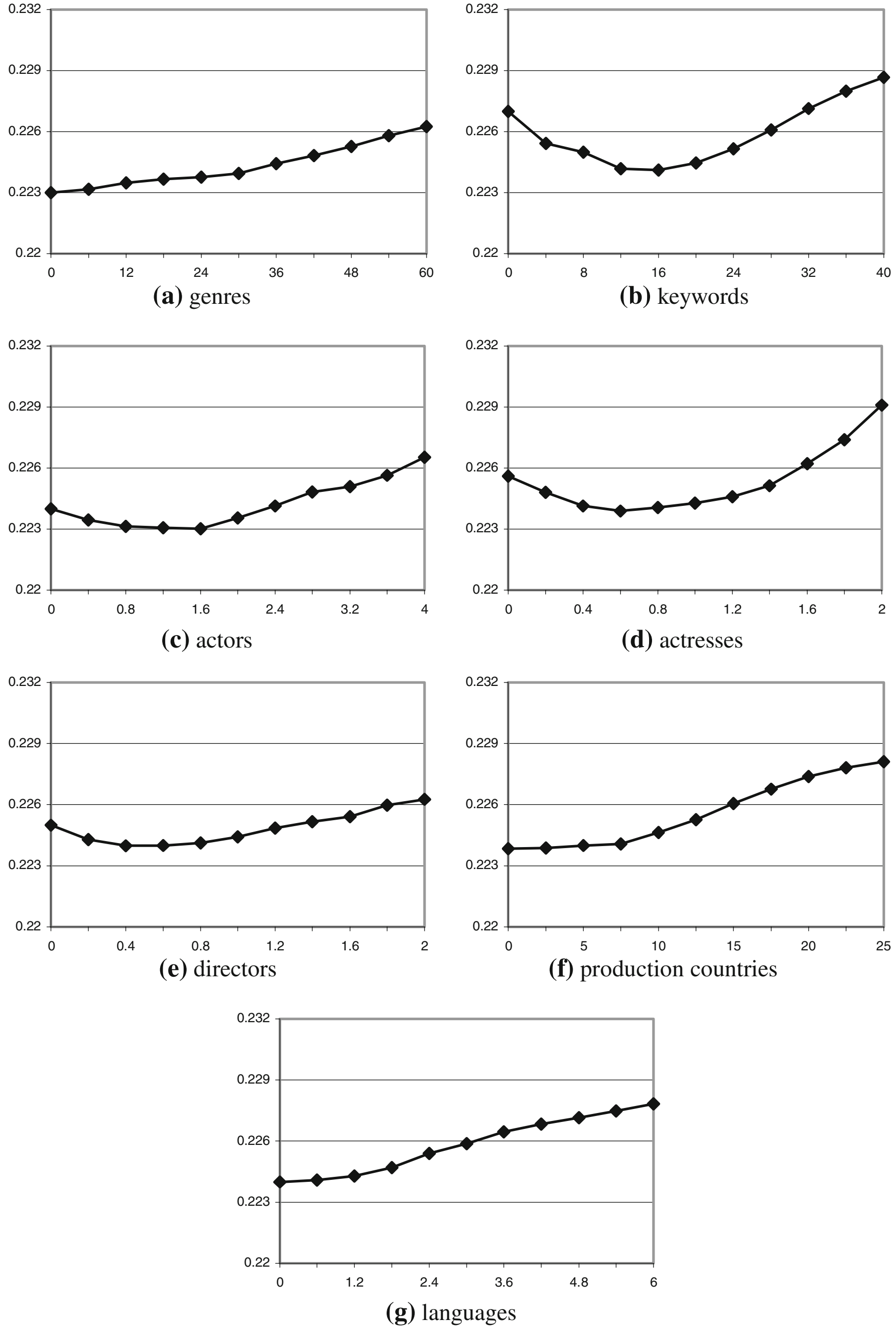

Fig. 6 MAE values versus *min-occur* threshold for: (**a**) *genres*, (**b**) *keywords*, (**c**) *actors*, (**d**) *actresses*, (**e**) *directors*, (**f**) *production countries*, and (**g**) *languages* categories

Table 4 Values of the *min-occur* threshold for various features categories

Category	Genres	Keywords	Actors	Actresses	Directors	Countries	Languages
min-occur (%)	0	16	1.6	0.6	0.4	0	0

initial search state for the feature selection was $S_0 = (0, 0, 0, 0, 0, 0, 0)$, where none of the available seven features categories was taken into account by the prediction mechanism. Hence, the implemented feature selection used forward selection of feature categories, i.e., the state expansions could bring only new feature categories that should be taken into account.

The accuracy of the recommendations for every state was computed using the previously described *FT* set. The training set (90% of the data) served as input for the collaborative filtering to content-based UM mediation. Then, the generated content-based UM was used for generating content-based recommendations for the movies in the testing set (10% of the data). Note that for each state of the search space, the content-based prediction mechanism took into account only those feature categories that were assigned 1 in the numeric vector representing the state, and ignored the categories that were assigned 0 in the vector.[5] The MAE was used to evaluate the predictive accuracy of the generated recommendations, i.e., to evaluate every state in the search space.

The execution of the wrapper feature selection yielded the following five feature categories: *genres*, *keywords*, *actors*, *actresses*, and *directors*. This means that only the *production countries* and *languages* categories were excluded by the feature selection.[6] Practically, this means that no correlation was discovered between the values of the *production countries* and *languages* features and the ratings assigned by a user. To validate these conclusions, we performed the accuracy experiment in two configurations: (1) taking into account the features from all seven feature categories, and (2) taking into account the features from five feature categories, excluding the *production countries* and *languages* categories. The results showed that the accuracy of the recommendations improved as a result of the feature categories selection. The results of this experiment will be presented and analyzed in the following section.

4.2 Accuracy of the recommendations

In this set of experiments, the features from the selected *genres*, *keywords*, *actors*, *actresses* and *directors* categories were taken into account by the prediction mechanism and the determined *conf* and *min-occur* thresholds were applied. These experiments were aimed at comparing the accuracy of the recommendations generated using

[5] Since the feature selection algorithm focused on selecting the most important feature categories, the values of both *min-occur* and *conf* thresholds in this experiment were set to 0. In other words, we searched for the best feature categories independently of these thresholds.

[6] Similar results were obtained when $S_0 = (1, 1, 1, 1, 1, 1, 1)$ was the initial search state and feature selection used backward selection of categories, i.e., the categories were removed.

the original collaborative filtering UMs and the mediated content-based UMs. For this experiment, the users in the full dataset were partitioned into 12 bins, according to the number of rated movies: below 25, 26 to 50, 51 to 75, 76 to 100,…,201 to 225, 226 to 250, 251 to 300, and 301 to 500 movies. Then, 325 users from each bin were selected, and their collaborative filtering UMs were partitioned into the training set (90% of the ratings) and the testing set[7] (10% of the ratings). The ratings in the training sets served as input for the collaborative filtering to content-based UMs mediation mechanism.

Then, two types of recommendations for the movies in the testing set were generated: (1) collaborative filtering recommendations based on the original collaborative filtering UMs, and (2) content-based recommendations based on the mediated content-based UMs. Predictive accuracy of the generated recommendations was compared (Herlocker et al. 2004). Hence, for each bin, the MAEs of the collaborative filtering and of the content-based recommendations for all the users in the bin were computed. We note that in this experiment, as well as in the previous ones, the training and the testing sets are disjoint.

To demonstrate the impact of the feature selection and validate our assumption regarding its importance for generating accurate recommendations, we computed the MAE of the content-based recommendations in two ways:

- Using all the features in the original seven feature categories: *genres*, *keywords*, *actors*, *actresses*, *directors*, *production countries* and *languages*. The results of this experiment are denoted in the chart by CB.
- Using only the features in the five categories selected by the feature category selection: *genres*, *keywords*, *actors*, *actresses*, and *directors*. The results of this experiment are denoted in the chart by CBFS.

Figure 7 shows the MAE values. The horizontal axis shows the number of ratings in the original collaborative filtering UMs, and the vertical axis shows the MAE of the generated predictions.

The comparison of the MAE of the content-based recommender system using all the seven feature categories and using the selected five categories shows that using feature categories selection reduces MAE for any size of UM. This result is statistically significant, with a t-test probability of $p = 0.0154$. Hence, the benefit of feature categories selection in the collaborative filtering to content-based UM mediation is clear. Because of this, the following analysis and comparison of collaborative and content-based recommendations mainly refers to the content-based recommendations using the selected five feature categories.

The chart shows that the MAE of the content-based recommendations for the UMs containing less than 50 movies is relatively low: around 0.16 when using the selected 5 feature categories. This can be explained by the observation that for a low number of rated movies in the collaborative filtering UMs, it is easy to determine the important content-based features and to compute their weights accurately. In this situation the

[7] In the fine-tuning experiment, we selected the *FT* set of 1,000 users who rated over 100 movies. For the accuracy experiment, we defined 12 other bins of 325 users, i.e., overall 3,900 users. Although there is some overlapping between the new set and the *FT* set, it is partial and only for the users who rated over 100 movies.

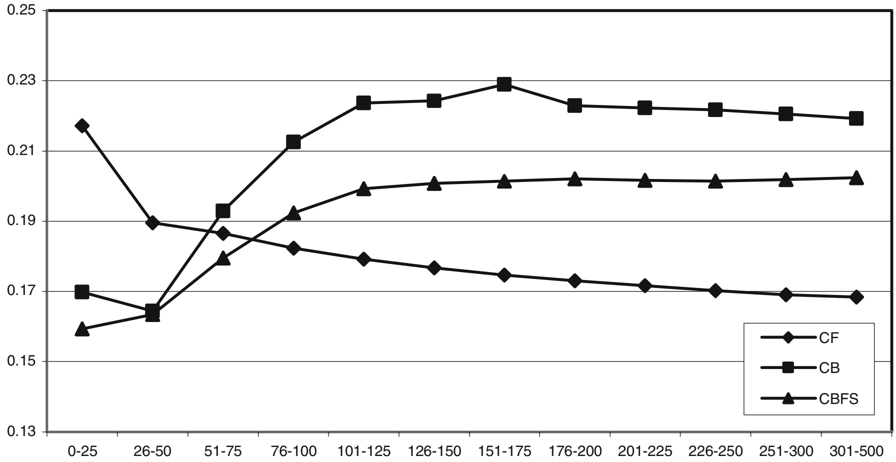

Fig. 7 MAE of content-based (with and without feature selection) and collaborative filtering recommendations versus the number of rated movies in the UM

number of neutral features is still low, and they do not affect the recommendation accuracy. For larger collaborative UMs, containing less than 100 rated movies, the MAE of the content-based system increases with the number of rated movies. We hypothesize that this is due to a larger number of neutral features, affecting the ratings prediction and hampering the accuracy of the generated recommendations.[8] For collaborative UMs with more than 100 rated movies, the MAE value stabilizes at approximately 0.20.

The comparison of the MAE of the content-based and collaborative recommendations shows that for the users who rated fewer than 75 movies, the content-based recommendation based on the mediated UMs are superior to the collaborative recommendations based on the same UMs. This result is statistically significant: the result of the t-test is $p = 0.0292$. Conversely, for the users who rated a larger number of movies, the accuracy of the collaborative filtering recommendations is superior to the accuracy of the content-based recommendations. Also this result is statistically significant, with a t-test probability of $p = 1.84\text{E-}07$. We hypothesize that applying more accurate fine-tuning and weighting mechanisms and discovering dependencies between specific features may further improve the accuracy of the content-based recommender system also for larger UMs. In the future, we plan to investigate these issues.

Furthermore, we note that according to Table 1, 78.4% of the users in the dataset rated fewer than 75 movies, i.e., the content-based recommender is better than the collaborative filtering system for the large majority of the users. In fact, the accuracy of the collaborative recommendations for these users is relatively low. Hence, the mediation of collaborative to content-based UMs and the generation of content-based recommendations provides a solid alternative to the collaborative filtering technique.

[8] For most of the bins the coverage is over 0.99, except the bin of less than 25 movies, where it is 0.97. That is, the recommendations can be computed for almost every movie.

The advantage of content-based approaches over collaborative based systems when there is a lack of UM data has already been extensively discussed in prior works (Burke 2002; Pazzani 1999). Content-based systems work better for users having *focused* preferences, i.e., the items that they like/dislike are highly similar. In fact, content-based systems typically approximate the UM with a set of features that aggregates the items liked by the user. If the user liked many different items, the content-based UM becomes closer to an average item, which can barely represent the focused preferences. The difficulty of content-based approaches in dealing with large UMs is also explained by the fact that with an increasing number of features the items are equally close to each other. And, in the proposed mediation approach, the number of features in content-based UMs increases with the number of rated items.

Finally, we would like to compare our results with those obtained from a similar experiment and reported in Basu et al. (1998). The latter work aimed at combining the collaborative filtering and content-based UMs and recommendation approaches. For this, (1) content features of the items rated in the collaborative filtering UM were extracted, (2) these features, together with the original collaborative ratings, were treated as the item features, and (3) content-based and hybrid recommendations were generated using these enriched UMs. The work also focused on the domain of movies, and information from 26 feature categories was extracted from the IMDb. Their experimental evaluation of the classification accuracy showed that the content-based recommender system is inferior to the original collaborative filtering system. Conversely, in this work we showed that for users with relatively small UMs, the predictive accuracy of the recommendations generated using the mediated content-based UMs is superior to that of the recommendations generated using the original collaborative UMs. We hypothesize that this improvement is achieved due to the particular rating prediction method we used and to its fine-tuning, i.e., the optimization of the *conf* and *min-occur* parameters and feature selection that restricted the prediction generation to 5 feature categories only instead of the 26 feature categories used in Basu et al. (1998).

4.3 Enriching a content-based system with collaborative-based user models

In previous experiments we showed how a content-based recommender system can be bootstrapped when no content-based UMs are available, using the UMs stored by a collaborative filtering system and external knowledge extracted from movie descriptions. In those experiments, all the training UM data coming from the collaborative filtering system were used to mediate content-based UMs. In a more realistic scenario, the two systems may be decoupled and may independently acquire UM data for a possibly overlapping population of users. For example, the collaborative system may collect the ratings of a user for a certain set of movies, whereas the content-based system may collect the opinions of the same user on a certain set of movie features. This setting raises the question of integrating the UM collected by the two systems and incrementally updating the content-based UMs.

This section presents an experiment, the results of which illustrate the outcome of UM mediation in a scenario similar to that described above. Here, we used the

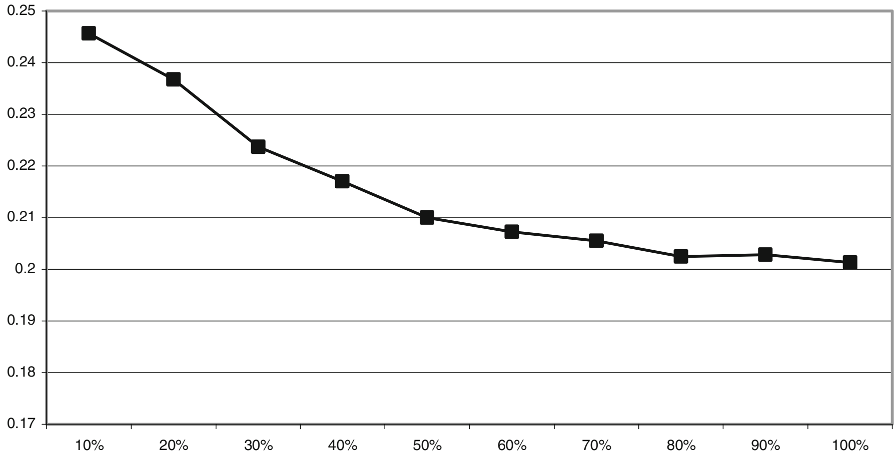

Fig. 8 MAE of the predictions versus the percentage of the mediated UM data

same EachMovie dataset (McJones 1997), but we simulated the existence of some content-based UM data and enriched these data with the UM data coming from the collaborative filtering recommender system. We also assumed that the two systems can cross-reference their users, i.e., the users are denoted in both systems with unique identification codes that are shared by the two systems. Then, we gradually transferred increasing parts of the collaborative filtering UM data of the shared users to the content-based system and we measured the accuracy of the generated content-based predictions.

Specifically, we considered again the *FT* set of ratings, which was used in previous experiments. We partitioned the *FT* set into the training set (90% of ratings) and the testing set (the remaining 10%). Then, a certain percentage of the training set collaborative filtering UM data of each user was mediated to the content-based UM and was used to generate content-based predictions. Hence, for example, when 20% of the collaborative filtering UM data was mediated to the content-based UMs, the content-based recommender system could already use these UMs to generate the predictions.

In this experiment, increasing parts of the collaborative filtering UMs were mediated to the content-based UMs, simulating a setting where certain content-based UM data are available to the content-based recommender system. For the additional parts of collaborative UMs transferred and mediated to the content-based system, the contribution of importing more UM data was evaluated. For example, the predictive accuracy of the content-based recommender system using the 20% of the mediated UMs was compared with the accuracy of the same system using an additional 10% of the collaborative filtering UM data, i.e., using in total 30% of the mediated UM data. The results of this experiment are shown in Fig. 8. The horizontal axis shows the percentage of the mediated collaborative filtering UM data, whereas the vertical shows the MAE of the predictions.

Figure 8 shows that the MAE decreases until 50–60% of the ratings are mediated from the collaborative filtering to the content-based recommender systems.

Afterwards, the MAE keeps decreasing, but more slowly. These results follow a rather general pattern of behavior. The initial ratings mediated from the collaborative filtering UM data contribute important information regarding user preferences, whereas the following mediated ratings contribute less and less information. However, the MAE continues decreasing with the mediation of UM data, as new UM data are still being imported and the accuracy of the UMs improves. These experiments demonstrate that our previous results are not limited only to a situation where the UMs of the content-based recommender systems are empty and the mediated UM data initializes the UMs. They demonstrate that the proposed mediation technique can be applied also to existing content-based recommender systems, as the mediation improves the accuracy of the predictions generated by the content-based recommender system by mediating additional UM data and enriching the available content-based UMs.

5 Conclusions and future research

This work focuses on cross-representation mediation of UMs, demonstrates its practical implementation, and evaluates the outcome of the collaborative to content-based filtering UM mediation. The mediation procedure allows bootstrapping the empty UMs and enriching the existing UMs in a content-based recommender system, and, as a result, more accurate recommendations are generated.

The experimental evaluation initially focused on the fine-tuning of the prediction mechanism. This included: (1) determining specific threshold values that allow accurate filtering of irrelevant features, and (2) applying the wrapper feature selection approach to determining the feature categories that should be taken into account by the prediction mechanism. The computed thresholds were applied to evaluate the accuracy of the generated content-based recommendations. The experimental results showed that for users with a small number of rated movies, who constitute the majority of the users, the content-based recommendations exploiting their mediated UMs are more accurate than collaborative filtering recommendations exploiting their original UMs. This allows us to draw the conclusion that cross-representation mediation of the UMs is feasible and beneficial, as it practically improves the accuracy of the recommendations provided to the users. Also, the experiments demonstrated the importance of feature selection, since the selection of the most relevant *categories* of features improved the accuracy of the recommendations.

The mediation and prediction mechanisms that we discussed are quite simple, as they assign uniform weights to different feature categories in the UMs. However, this may hamper the accuracy of the generated UMs, as not all the feature categories are of the same importance, and weighting may be applied to both the feature categories and the specific features within these categories. Moreover, the categories and features are treated in this work as mutually independent, whereas in some applications such dependencies can be identified. In the future, we plan to enhance the mediation process by applying machine learning and data mining techniques for the purpose of inferring the weights of the categories and specific features within the categories and identifying the dependencies between various features. We hypothesize that this will improve the accuracy of the recommendations further and strengthen the proposed mediation approach.

Although this work demonstrated practical implementation and evaluation of cross-representation mediation of UMs, it exploited a static offline dataset of ratings and did not involve studies with real users. In the future, we plan to evaluate the proposed approach in practical studies with real-life users and Web-based recommender systems. We also plan to evaluate extensively the proposed approach in different conditions: other application domains, e.g., cultural heritage, other types of mediations, e.g., the reverse mediation from content-based to collaborative UMs, and other types of recommender systems involved in the mediation, e.g., case-based recommender systems. In particular, we plan to test the UM mediation techniques on different types of user models. For instance, content-based UMs can rely on a more abstract representation of the content using a domain ontology, while collaborative filtering UMs can be exploited to learn a ranking for topics that may interest the user. These ranked lists can then be exploited for building new recommendations for topics and items belonging to those topics, similarly to the techniques proposed in Stamou and Ntoulas (2008).

In this work, the accuracy of the generated content-based recommendations was compared to the accuracy of the original collaborative filtering recommendations. However, prior studies showed that the highest accuracy of the recommendations may be achieved using various hybrid approaches (Burke 2002). In the future, we plan to compare the proposed approach with hybrid approaches. It is important to understand whether the information extracted from the IMDb and provided by the content-based technique can also benefit the collaborative filtering technique. For instance, content-based UMs can be used in a collaborative-by-content approach for estimating the user's similarity (Pazzani 1999). This would represent an example of two-stage mediation, where the collaborative UMs are mediated first from collaborative-filtering to content-based filtering and then back to collaborative filtering. This shows that the proposed mediation should be considered not only as a conversion process, but also as a process where each mediation stage introduces new knowledge that can be added to the generated UMs.

It is worth noting that collaborative filtering UMs can be manipulated by malicious attacks (Mehta and Nejdl 2008). As such, the users may refrain from providing ratings for certain items to minimize the possible negative impact on the privacy of their personal data. We are studying new techniques for privacy-preserving collaborative filtering (Berkovsky et al. 2007) and plan to apply these techniques to enhance the privacy aspects of the mediation and facilitate a wider sharing, distribution, and reuse of UM data.

Acknowledgements The authors gratefully acknowledge the support of the Caesarea Edmond Benjamin de Rothschild Foundation Institute for Interdisciplinary Applications of Computer Science (CRI) in Haifa and of the Istituto Trentino di Cultura—the Center for Scientific and Technological Research (ITC-irst) in Trento. The authors also thank Boris Bolshem and Sveta Ogiyenko for their assistance in the implementation of the system.

References

Balabanovic, M., Shoham, Y.: Content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)

Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: using social and content-based information in recommendation. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 26–30, pp. 714–720 (1998)

Berkovsky, S., Eytani, Y., Kuflik, T., Ricci, F.: Enhancing privacy and preserving accuracy of a distributed collaborative filtering. In: Proceedings of the Recommender Systems Conference, RecSys 2007, October 19–20, pp. 9–16 (2007)

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

Bernstein, P.A., Melnik, S.: Meta data management. In: Proceedings of the 20th International Conference on Data Engineering, ICDE 2004, 30 March–2 April, 875 pp (2004)

Billsus, D., Pazzani, M.J.: User modeling for adaptive news access. User Model. User-Adapt. Interact. **10**(2–3), 147–180 (2000)

Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adapt. Interact. **12**(4), 331–370 (2002)

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: GUMO—the general user model ontology. In: Proceedings of User Modeling 2005, 10th International Conference, UM 2005, Edinburgh, Scotland, UK, July 24–29, pp. 428–432 (2005)

Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99), August 15–19, pp. 230–237 (1999)

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

Kay, J., Kummerfeld, B., Lauder, P.: Managing private user models and shared personas. In: Proceedings of the UbiUM Workshop (2003)

Kobsa, A.: Generic user modeling systems. User Model. User-Adapt. Interact. **11**(1–2), 49–63 (2001)

Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. J. **97**(1–2), 273–324 (1997)

Lorenz, A.: A specification for agent-based distributed user modeling in ubiquitous computing. In: Proceedings of the DASUM Workshop (2005)

McJones, P.: EachMovie Collaborative Filtering Data Set. HP Research (1997)

McNee, S.M., Lam, S., Konstan, J., Riedl, J.: Interfaces for eliciting new user preferences in recommender systems. In: Proceedings of User Modeling 2003, 9th International Conference, UM 2003, Johnstown, PA, USA, June 22–26, pp. 178–187 (2003)

McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montreal, Quebec, Canada, April 22–27, pp. 1097–1101 (2006)

Mehta, B., Nejdl, W.: Unsupervised strategies for shilling detection and robust collaborative filtering. User Model. User Adapt. Interact. (2008) doi:10.1007/s11257-008-9050-4.

Morita, M., Shinoda, Y.: Information filtering based on user behavior analysis and best match retrieval. In: Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 3–6 July, pp. 272–281 (1994)

Mulvenna, M.D., Anand, S.S., Buchner, A.G.: Personalization on the net using web mining. Commun. ACM **43**(8), 123–125 (2000)

Niederee, C., Stewart, A., Mehta, B., Hemmje, M.: A multi-dimensional, unified user model for cross-system personalization. In: Proceedings of the E4PIA Workshop (2004)

Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artif. Intell. Rev. **13**(5–6), 393–408 (1999)

Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3), 56–58 (1997)

Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall publishers (1995)

Stamou, S., Ntoulas, A.: Search personalization through query and page topical analysis. User Model. User Adapt. Interact. (2008) doi:10.1007/s11257-008-9056-y.

Tintarev, N.: Explaining recommendations. In: Proceedings of User Modeling 2007, 11th International Conference, UM 2007, Corfu, Greece, June 25–29, pp. 470–474 (2007)

Zanker, M., Jessenitschnig, M.: Case-studies on exploiting explicit customer requirements in recommender systems. User Model. User Adapt. Interact. (2008) doi:10.1007/s11257-008-9048-y.

Authors' vitae

Shlomo Berkovsky, CSIRO–Tasmanian ICT Centre, GPO Box 1538, Hobart, Tas 7001, Australia. Shlomo Berkovsky received his PhD degree from the University of Haifa, Israel, where his research focused on mediation of user models in recommender systems. Currently, he is a research scientist at CSIRO (Commonwealth Scientific and Industrial Research Organization)–Tasmanian ICT Centre. His research project focuses on personalized delivery of health information. His research interests also include ubiquitous and distributed user modeling, recommender systems, context-aware personalization, and personalized text summarization.

Tsvi Kuflik, Department of Management Information Systems, University of Haifa, Mount Carmel, Haifa, 31905, Israel. Tsvi Kuflik received his BSc and MSc in Computer Science and PhD in Information Systems from the Ben-Gurion University of the Negev. Currently, he is a lecturer in the Management Information Systems Department at the University of Haifa. Before joining the University of Haifa he worked as a researcher at IRST research institute in Trento. At IRST, he worked on user modeling in active museums, work he continues at the University of Haifa in the framework of the collaboration projects of the University of Haifa and IRST, where he leads a research group focusing on supporting museum visitors. His research interests focus mainly on user modeling, ubiquitous user modeling and information filtering.

Francesco Ricci, Free University of Bozen-Bolzano, Faculty of Computer Science, Piazza Domenicani 3, I-39100 Bolzano, Italy. Francesco Ricci received his degree in Mathematics from the University of Padova, and he is now associate professor at the Faculty of Computer Science, Free University of Bozen-Bolzano. His current research interests include recommender systems, intelligent interfaces, constraint satisfaction problems, machine learning and case-based reasoning. He is the author of many scientific publications that have appeared in international journals and conferences proceedings, and he is on the editorial board of the Information Technology and Tourism Journal and he is a member of ACM.