

Alleviating the New User Problem in Collaborative Filtering by Exploiting Personality Information*

Ignacio Fernández-Tobías · Matthias Braunhofer · Mehdi Elahi · Francesco Ricci · Iván Cantador

Received: date / Accepted: date

Abstract The new user problem in recommender systems is still challenging, and there is not yet a unique solution that can be applied in any domain or situation. In this paper we analyse viable solutions to the new user problem in collaborative filtering that are based on the exploitation of user personality information: (a) *personality-based collaborative filtering*, which directly improves the recommendation prediction model by incorporating user personality information; (b) *personality-based active learning*, which utilizes personality information for identifying additional useful preference data in the target recommendation domain to be elicited from the user; and (c) *personality-based cross-domain recommendation*, which exploits personality information to better use user preference data from auxiliary domains which can be used to compensate the lack of user preference data in the target domain. We benchmark the effectiveness of these methods on large datasets that span several domains, namely movies, music and books. Our results show that personality-aware methods achieve performance improvements that range from 6% to 94% for users completely new to the system, while increasing the novelty of the recommended items by 3% to 40% with respect to the non-personalized popularity baseline. We also discuss the limitations of our approach and the situations in which the proposed methods can be better

* This paper or a similar version is not currently under review by a journal or conference, nor will it be submitted to such within the next three months. This paper is void of plagiarism or self-plagiarism as defined in Section 1 of ACM's Policy and Procedures on Plagiarism.

Ignacio Fernández-Tobías · Iván Cantador
Universidad Autónoma de Madrid, Madrid, Spain
E-mail: {ignacio.fernandezt, ivan.cantador}@uam.es

Matthias Braunhofer · Mehdi Elahi · Francesco Ricci
Free University of Bozen-Bolzano, Bozen-Bolzano, Italy
E-mail: {mbraunhofer, mehdi.elahi, francesco.ricci}@unibz.it

applied, hence providing guidelines for researchers and practitioners in the field.

Keywords recommender systems · collaborative filtering · user personality · cold-start · cross-domain · active learning

1 Introduction

Recommender systems are information search and decision support tools that address the problem of information overload by generating personalized suggestions for items, e.g., products and services, that suit the specific user’s needs and constraints [59, 60, 33]. Collaborative Filtering (CF) is a well known recommendation technique that exploits ratings for items given by a network of users. A CF system generates recommendations by analysing the similarities and relationships between the users, which can be extracted by observing the users’ interactions with the items managed by the system [38, 16]. Some popular examples of successful CF recommender systems are *Amazon*¹, *Netflix*², *iTunes*³ and *Last.fm*⁴.

Collaborative filtering can be implemented in several variants, such as user- [29] and item-based [45] heuristics, and Matrix Factorization (MF) models [38]. However, regardless of the specific variant that is used, CF methods have a common limitation: the so called *new user* cold-start problem, which occurs when a system cannot generate personalized and relevant recommendations for a user who has just registered into the system. Although many solutions have been proposed [22, 23, 31, 53, 55, 69, 44, 66], this problem is still challenging, and there is not a unique solution for it that can be applied to any domain or situation. Indeed, as we shall show later, different approaches better suit specific situations, e.g., when the new user has entered either zero or only a few ratings.

In this paper we propose to address the new user problem by assuming that the system has information about the users’ personality. Such information is used to enhance the effectiveness of collaborative filtering. In fact, research has already shown that, in certain domains, people with similar personality traits are likely to have similar preferences [10, 13, 56, 58], and that correlations between user preferences and personality traits allow improving personalized recommendations [31, 69]. Hence, the common gist of the methods proposed in this paper is to exploit user personality information in order to identify the most useful user preference information (ratings or “likes”) for the system to generate accurate recommendations for a new user.

¹ <http://www.amazon.com>

² <http://netflix.com>

³ <http://www.apple.com/itunes>

⁴ <http://www.last.fm>

More specifically, we present three novel methods to alleviate the new user problem in collaborative filtering: (a) *personality-based collaborative filtering*, which directly improves the recommendation prediction model by incorporating user personality information; (b) *personality-based active learning*, which utilizes personality information for identifying additional and useful user preference data to be elicited in a target domain; and (c) *personality-based cross-domain recommendation*, which exploits personality information to better exploit user preference data from auxiliary domains in order to compensate the lack of user preference data in the target domain.

More specifically, the first method exploits the users' personality information in a Matrix Factorization (MF) collaborative filtering model. We focus on MF since it is an accurate CF technique [38]. While the classical MF is trained exclusively on a set of ratings, our personality-based matrix factorization method allows to partially compensate the lack of ratings with personality information. In CF it has already been shown that personality can help overcoming the new user problem [31, 69]. Nonetheless, differently to our work, previous studies have investigated the incorporation of personality into CF heuristics instead of into MF models. In this paper, we extend the MF model [32] by incorporating additional latent feature vectors, which are related to user personality, and by performing a new training procedure based on the Alternating Least Squares technique. As we shall show, our method is beneficial in cold-start situations where there is no rating for the target user.

The second method is an Active Learning (AL) technique that, by knowing the users' personality, aims at finding and eliciting the most informative user ratings with minimal number of requests. In general, in AL it has been shown that asking a user to provide ratings for a set of selected items can improve the accuracy of CF [62, 22, 20, 21, 54, 55]. Traditional active learning methods need some pre-existent ratings in order to select even more ratings to collect from the user. Our method identifies the items to request the user to rate by considering the user personality, which may be easier to obtain than a bootstrapping set of ratings [5], and, since personality is not domain dependent, can be obtained once and then reused in several recommendation domains. Some existing AL approaches have already exploited user personality information [18, 5, 22]. In contrast to previous work, our personality-based active learning method is optimized for positive-only feedback (e.g. *likes*, click-through data, item consumption counts) instead of ratings, which is more easily acquired by the system. We also provide experimental results on a relatively larger dataset in several domains, and show that our personality-aware AL method is able to acquire more *likes* than a number of baseline methods for users completely new to the system.

Finally, the third method addresses the new user problem by enhancing cross-domain recommendation techniques with user personality information. Cross-domain recommender systems aim at improving their performance in a target domain by relying on information about the user's preferences in a related source domain [9]. The goal of these systems is to transfer useful

knowledge from auxiliary domains to the target domain in order to build in the target domain better rating predictions and item recommendations. Recently, cross-domain recommendation techniques have been applied to address the new user problem, by leveraging the knowledge of common rating patterns [25, 42, 52], shared social tags [65, 23], and linked semantic concepts [35] in the source and target domains. To the best of our knowledge, our cross-domain recommendation method represents the first attempt to exploit user personality as a “bridge” to transfer user preference knowledge between domains, and address cold-start situations in a target domain.

Conducting experiments on a large dataset in various application domains, namely movie, music and book recommendations, our empirical results reveal that the proposed methods, based on the exploitation of user personality, allow CF to better tackle the new user problem. This is especially true in the case of completely new users with no rating history at all (i.e. users having no training ratings), who are typically provided with non-personalized suggestions based only on the popularity of the items. We show that these users can significantly benefit from the application of the proposed methods, which are able to generate personalized recommendations that boost the precision of the system in ranges from 6% to 94%, depending on the domain. We also show, however, that this benefit vanishes once a sufficient number of training ratings for a user becomes available.

The remainder of the paper is structured as follows. In Section 2 we review the relevant state of the art, and in Section 3 we detail our methods. In Section 4 we describe the dataset and methodology used to evaluate the methods. Next, in Section 5 we report the results of the conducted evaluation and provide a comprehensive discussion comparing the methods and describing possible application scenarios for them. Finally, in Section 6 we provide some conclusions and future work research lines.

2 Related Work

2.1 On the Relationships Between User Personality and Preferences

Personality is a predictable and quite stable factor that forms human behaviors. In psychology literature, personality is described as a “consistent behavior pattern and interpersonal processes originating within the individual” [8], accounting for individual differences in people’s emotional, interpersonal, experiential, attitudinal and motivational styles [34]. Different models have been proposed to characterize and represent human personality. Among them, the Five Factor model (FFM) [14] is considered one of the most comprehensive, and has been mostly used to build user profiles [31]. The FFM introduces five broad domains or dimensions – called factors or traits, and commonly known as the *Big Five* – to describe an individual’s personality: *openness*, *conscientiousness*, *extraversion*, *agreeableness* and *neuroticism*. The openness

(OPE) factor reflects a person’s tendency to intellectual curiosity, creativity and preference for novelty and variety of experiences. The conscientiousness (COS) factor reflects a person’s tendency to show self-discipline and aim for personal achievements, and have an organized (not spontaneous) and dependable behavior. The extraversion (EXT) factor reflects a person’s tendency to seek stimulation in the company of others, and put energy in finding positive emotions. The agreeableness (AGR) factor reflects a person’s tendency to be kind, concerned, truthful and cooperative towards others. Finally, the neuroticism (NEU) factor reflects a person’s tendency to experience unpleasant emotions, and refers to the degree of emotional stability and impulse control.

The measurement of the five factors is performed by assessing “items” that are self-descriptive sentences or adjectives, and are commonly presented to the subjects in the form of short questions. In this context, the International Personality Item Pool⁵ (IPIP) is a publicly available collection of items for use in psychometric tests, and the 20-100 item IPIP proxy for Costa and McCrae’s commercial NEO PI-R test (IPIP-NEO, see [28]) is one of the most popular and widely accepted questionnaires to measure the Big Five in adult men and women without overt psychopathology.

Personality influences how people make decisions [49]. Moreover, it has been shown that people with similar personality traits are likely to have similar tastes. For example, in [58] Rentfrow et al. investigated how music preferences are related with personality in terms of the FFM. They show that “reflective” people with high openness usually have preferences for jazz, blues and classical music, and “energetic” people with high degree of extraversion and agreeableness usually appreciate rap, hip-hop, funk and electronic music. In [56] Rawlings et al. observed that openness and extraversion are the personality factors that best explain the variance in personal preferences. They showed that people with high openness tend to like diverse music styles, and people with high extraversion are likely to have preferences for popular music. In the movie domain, Chausson [13] presented a study showing that people open to experiences are likely to prefer comedy and fantasy movies, conscientious individuals are more inclined to enjoy action movies, and neurotic people tend to like romantic movies. Odic et al. [50] explored the relations between personality factors and induced emotions while watching movies in different social contexts (e.g. watching a movie alone and with someone else), and observed different patterns in experienced emotions as functions of the extraversion, agreeableness and neuroticism factors. More recently, Braunhofer et al. [6] have shown that exploiting personality information in collaborative filtering is more effective than exploiting demographic information of users, which is a more typical approach for dealing with the new user problem in recommender systems. In particular, it has been shown that exploiting even a single personality factor (out of the five factor) may lead to a considerable improvement in recommendation accuracy.

⁵ International Personality Item Pool, <http://ipip.ori.org>

Extending the spectrum of analyzed domains, in [57] Rentfrow et al. studied the relations between personality factors and user preferences in several entertainment domains, namely movies, TV shows, books, magazines and music. They focused their study on five content categories: aesthetic, cerebral, communal, dark and thrilling. The authors observed positive and negative relations between such categories and some of the personality factors, e.g. they showed that aesthetic content relate positively with agreeableness and negatively with neuroticism, and that cerebral content correlate with extraversion. Cantador et al. considered also several domains (movies, TV shows, books and music) [10], and presented a preliminary study on the relations between personality types and entertainment preferences. Analyzing a large dataset of personality factor and genre preference user profiles, the authors extracted personality-based user stereotypes for each genre, and inferred association rules and similarities between types of personality of people with preferences for particular genres. Finally, in the multi-domain scenario of the Web, Kosinski et al. [40] presented a study revealing meaningful psychologically relations between user preferences and personality for certain websites and website categories.

We notice that, as showed in [10], additional user characteristics, such as the user’s age and gender, and more fine-grained personality representations such as those based on personality facets, e.g. the *imagination*, *artistic interests*, and *emotionality* facets for the *openness* factor, are likely to be of importance when discovering relationships between user preferences and personality. In the reviewed works and in this paper, such type of characteristics are not taken into consideration, and are left for future investigation. We develop and evaluate our methods upon the fact that there exist certain relationships between user preferences and personality, which can benefit CF in the cold-start, as done in previous work and show in the next subsection.

2.2 Personality-based Collaborative Filtering

The existence of certain relationships between personality characteristics and user preferences, has motivated earlier studies supporting the hypothesis that exploiting personality information in CF is beneficial. In [69] Tkalčić et al. applied and evaluated three user similarity metrics for heuristic-based CF: a typical rating-based similarity based on Euclidean distance with personality data (five factors), and a similarity based on a weighted Euclidean distance with personality data. Their results show that approaches using personality data may perform statistically equivalent or better than rating-only based approaches, especially in cold-start situations. In her PhD dissertation [48], Nunes explored the use of a personality user profile composed of IPIP-NEO items and facets in addition to the Big Five factors, showing that fine-grained personality user profiles can achieve better recommendations. Following the findings of Rentfrow and Gosling [58], in [30,31] Hu and Pu presented a CF

approach that leverages the correlations between personality types and music preferences: the similarity between two users is estimated by means of the Pearson’s correlation coefficient on the users’ five factors scores. Combining this approach with a rating-based CF technique, the authors showed significant improvements over the baseline of considering only ratings data. Finally, in [61] Roshchina presented an approach that extracts five factors profiles by analyzing hotel reviews written by users, and incorporates these profiles into a nearest neighbor algorithm to enhance personalized recommendations.

It is worth noting that the above mentioned works on personality-aware CF make use of heuristic-based methods to compute user similarities and item rating estimations. Differently from them, in this paper we propose a Matrix Factorization CF model – which has been shown to be more effective than heuristic approaches – that integrates the users’ rating data and personality information. Moreover, with respect to previous work, the experimental study presented here has been conducted on much larger datasets composed of “likes”, which are positive only evaluations, rather than ratings, in several domains. Specifically, as described in Section 4, our dataset consisted of 159,551 users and 16,303 items in the movie, music and book domains, while in [30,31] the considered data set contains only 111 users, in [48] and [61] it is around 100, and in [69] 52; and all of these data sets contain only a very limited number of items.

Moreover, we shall illustrate a more diverse set of results. We will observe that the users’ personality is not equally useful in all the considered domains. For instance, the usage of personality in the movie and music domains yields to higher precision compared to the book domain. This can be due to the strength of the correlation between people personality and the characteristics of the domain. Hence, the personality may affect much more their decision on choosing which movie to watch or which song to listen rather than which book to read. We will also show that, while the personality information can improve significantly the recommendation precision when the user has not yet entered a single *like*, it is not effective anymore when the user starts entering certain number of *likes*.

2.3 Active Learning in Collaborative Filtering

Active Learning in CF aims to actively acquire user preference data in order to improve the output of the recommendation process [62,22,20,21]. In two separate works [54,55], Rashid et al. proposed eight techniques that CF systems can use to acquire user preferences in the sign-up stage: *entropy*, where items with the largest rating entropy are preferred; *random*; *popularity*; $\log(\text{popularity}) * \text{entropy}$, where items that are both popular and have diverse ratings are preferred; and finally *item-item personalized*, where the items are proposed randomly until one rating is acquired, and then a recommender system is used to predict the ratings for items that the user is likely

to have seen; *IGCN*, which builds a tree where each node is labeled by a particular item to be asked to the user to rate; and *Entropy0*, which extends the *entropy* method by considering the missing value as a possible rating (category 0). They conducted offline and online simulations, and concluded that *IGCN* and *Entropy0* perform the best in terms of accuracy.

In [26] and [27], Golbandi et al. proposed four strategies for rating elicitation in CF. The first method, *GreedyExtend*, selects the items that minimize the Root Mean Square Error (RMSE) of the rating prediction (on the training set). The second one, named *Var*, selects the items with the largest $\sqrt{\text{popularity} \times \text{variance}}$, i.e., those with many and diverse ratings in the training set. The third one, called *Coverage*, selects the items with the largest coverage, which is defined as the total number of users who co-rated both the selected item and any other item. The fourth method, called *Adaptive*, is based on decision trees where each node is labeled with an item (movie). The node divides the users into three groups based on their ratings for that item: *lovers*, who rated the item high; *haters*, who rated the item low; and *unknowns*, who did not rate the item. Their results showed excellent performance of the *Adaptive* method in terms of reduction of RMSE compared with other strategies.

Among the previous strategies, we have chosen *Entropy0* as baseline method, since it (or its similar variant) has shown excellent results in several works [54, 55, 46, 12, 63, 47]. This method aims at balancing the quantity and quality of the acquired ratings, in the sense that it attempts to collect as many ratings as possible, but also take their relative informativeness into account. It not only scores higher the items that are likely to be known by the users, but also brings valuable information about their preferences. We note that we could not use a decision tree based strategy as baseline, since that type of solutions exploit an RMSE reduction criterion, in training datasets that contain ratings in a Likert 1-5 scale. In contrast, in this work we use unary rated data sets (a user expresses only a set of “likes”) and hence, RMSE is not an appropriate error measure. That is also why we cannot use traditional recommendation evaluation metrics, such as RMSE or MAE.

2.4 Cross-domain Recommender Systems

The proliferation of e-commerce sites and online social networks has led users to provide feedback and maintain profiles in multiple systems, reflecting a variety of their tastes and interests. Leveraging all the user preferences available in several systems or domains may be beneficial for generating more encompassing user models and better recommendations, e.g. through mitigating the cold-start and sparsity problems in a target domain, or enabling personalized cross-selling recommendations for items from multiple domains.

In this context, cross-domain recommender systems aim to generate or enhance personalized recommendations in a target domain by exploiting knowl-

edge (mainly user preferences) from auxiliary source domains [24, 70]. This problem has been addressed from various perspectives in different research areas. It has been tackled by means of user preference aggregation and mediation strategies for the cross-system personalization problem in user modeling [1, 3, 64], as a potential solution to mitigate the cold-start and sparsity problems in recommender systems [15, 65, 68, 23] and as a practical application of knowledge transfer in machine learning [25, 42, 52].

We distinguish between two main types of cross-domain recommendation approaches: those that aggregate knowledge from various source domains into the target domain where recommendations are performed, and those that link or transfer knowledge between domains to support recommendations in a target domain. The knowledge aggregation methods merge user preferences (e.g. ratings, social tags, and semantic concepts) [1], mediate user modeling data exploited by various recommender systems (e.g. user similarities and user neighborhoods) [64], and combine single-domain recommendations (e.g. rating estimations and rating probability distributions) [3]. The knowledge linkage and transfer methods may relate domains by a common knowledge (e.g. item attributes, association rules, semantic networks, and inter-domain correlations) [15, 68], share implicit latent features that relate source and target domains [52, 65], and exploit explicit or implicit rating patterns from source domains in the target domain [25, 42].

With respect to previous work, to the best of our knowledge, this paper presents the first attempts to exploit user personality for cross-domain recommendation. Our methods can be considered as aggregation cross-domain recommendation approaches that merge both user personality traits and preferences from various domains.

3 Personality-Based Recommendation Methods for New Users

As already mentioned, in this paper we address the *new user* cold-start problem in CF, which occurs when a recommender system is unable to accurately suggest items in a target domain to a user for which no or few preference data are available in that domain. In order to illustrate this problem, we consider the typical input dataset used by CF, i.e., a sparse matrix that contains some users' ratings (*likes*) for a certain number of items. The rows of this matrix contain the users' ratings and the columns contain the items' ratings. When a new user registers into the system, a new empty row is added to the rating matrix and the system is unable to generate any rating predictions for this user.

The way in which we tackle the problem is depicted in Figure 1. We distinguish between two main types of approaches: (i) directly asking the user to rate some items in order to gather some information about her preferences before computing any recommendation, and (ii) exploiting auxiliary informa-

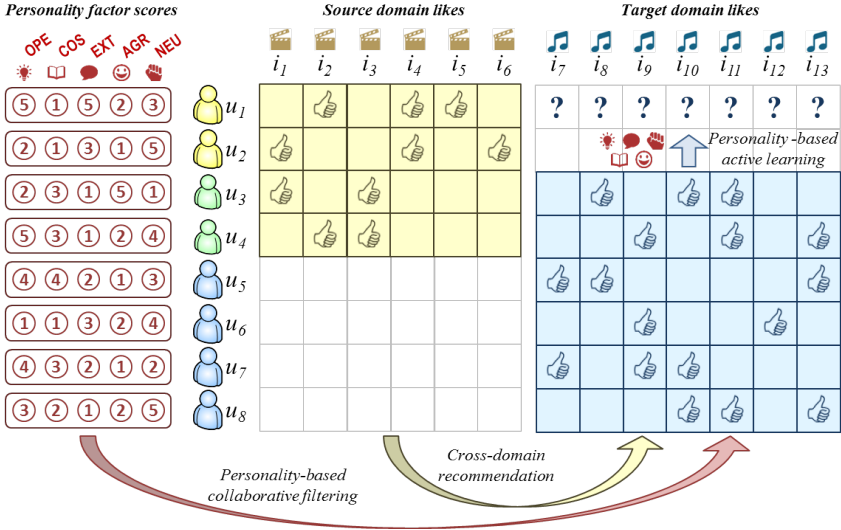


Fig. 1 Exploiting user personality and cross-domain preferences to address the new user cold-start problem. The users’ personality factors and preferences (*likes*) for items in two domains –a target domain (music) and an auxiliary source domain (movies)– are considered. User u_1 has no ratings in the target domain.

tion about the user’s preferences or other personal information that might be useful for the system to establish similarities between her and other users.

Both approaches have specific advantages and disadvantages. Preference elicitation approaches are more robust in the sense that they tackle the problem by directly acquiring the data the system ultimately needs. On the negative side, they require an initial effort from the user to rate some items when she registers, which may discourage her from using the system. Also, the system has to be careful when selecting the items to request the user to rate: the user should be familiar with them; otherwise she will not be able to rate them and her perception of the usefulness of the system could also be negatively affected. In contrast, approaches that exploit auxiliary information do not require the user to rate items. Nonetheless, a method to inject the additional user data into the CF framework has to be devised, and it may be hard to know in advance whether the auxiliary information will actually be helpful in the cold-start. Furthermore, it is typically the case that the auxiliary information is explicitly requested to the user, thus requiring an initial effort from her as in preference elicitation strategies.

Our work is based on the hypothesis that information about the user’s personality is available and can be used to enhance both types of approaches for tackling the cold-start problem. First, we propose a novel extension of a Matrix Factorization technique for positive-only feedback (click-through data, browsing history, item consuming counts) that is capable of exploiting

auxiliary personality information for recommendation. Second, we propose a personality-based MF algorithm for preference elicitation. Before recommending items, personality is exploited in the “likes” acquisition phase to more effectively select items that the user is likely to be able to like. Finally, we further extend the proposed personality-based MF algorithm integrating cross-domain user preferences. When little or no information about the user is available in a target domain (e.g. music), but her preferences in a different source domain are known (e.g. movie), we conjecture that personality information can be used to better leverage the auxiliary cross-domain data in order to provide recommendations in the target domain. We detail these three methods in the following subsections.

3.1 Personality-based Matrix Factorization for Positive-only User Feedback

In this section we describe the proposed matrix factorization method extended with personality information. First, let U, I be the sets of users and items registered in a system, respectively, and let $\mathbf{p}_u \in \mathbb{R}^k, \mathbf{q}_i \in \mathbb{R}^k$ be latent feature vectors for user $u \in U$ and item $i \in I$. In a simple matrix factorization model, the user u ’s preference towards item i is estimated with the dot product of the user and item latent feature vectors:

$$\hat{r}_{ui} = \mathbf{p}_u \cdot \mathbf{q}_i \quad (1)$$

A list of recommended items for user u is generated by sorting the items in I by decreasing order of estimated preference, eventually ignoring those that the user has already rated.

This method has been extensively studied in the literature, and it is known to yield inaccurate predictions in cold-start situations. When little information about the user is known, the learned parameter \mathbf{p}_u is unlikely to model properly the user’s latent preferences, and for users completely new to the system this method is simply unable to compute any rating prediction. In our adaptation, we overcome this limitations by introducing additional parameters to model the user’s personality.

Among the existing models for representing personality, in this work we focus on the Five Factor one. As explained in Section 2, in the FFM the personality of each user is described using five independent dimensions or factors, namely *openness*, *conscientiousness*, *extraversion*, *agreeableness* and *neuroticism*. A user’s personality profile is thus represented with a score for each factor, typically a real number in a range such as $[1, 5]$.

In order to use this information we follow the same strategy as in [18] and map the five factors to a fixed set of Boolean attributes \mathcal{A} . Specifically, let $\mathbf{u} = (ope_u, con_u, ext_u, agr_u, neu_u)$ be the vector representation of user u ’s FF scores. We first discretize each score by rounding it to the nearest integer, and then we map each score to a different attribute depending on its value

and factor. Therefore, we consider 25 possible attributes, five for each personality factor: $ope_1, ope_2, \dots, ope_5, con_1, \dots, neu_5$. For instance, a user with personality profile $\mathbf{u} = (2.3, 4.0, 3.6, 5.0, 1.2)$ will be assigned the set of attributes $\mathcal{A}(u) = \{ope_2, con_4, ext_4, agr_5, neu_1\}$. We considered and evaluated other personality factor discretization schemes and personality-based user profile representations, but they performed worse when incorporated into the MF model.

Once the user’s personality factor scores are transformed, we modify Equation 1 as in [18] to take personality information into account when computing predictions. Specifically, they define a new additional latent feature vector $\mathbf{y}_a \in \mathbb{R}^k$ for each attribute $a \in \mathcal{A}$. Now, the users are not only modeled in terms of their preferences, but also considering their personality attributes:

$$\hat{r}_{ui} = \mathbf{q}_i \cdot \left(\mathbf{p}_u + \sum_{a \in \mathcal{A}(u)} \mathbf{y}_a \right) \quad (2)$$

One important feature of this model is that it is capable of computing rating/like predictions even if the user is completely new to the system, making it ideal for cold-start situations. In such cases, the vector \mathbf{p}_u is ignored and ratings/likes are estimated only on the basis of attribute information.

The prediction model defined in Equation 2 is inspired by the well known and widely used SVD++ model [37]. SVD++ incorporates implicit feedback by introducing latent feature vectors for items rated by the user, whereas in Equation 2 the user’s profile is augmented with latent feature vectors that model her personality. Unlike [18] and [37], the method we propose in this paper is intended for the top N recommendation task in the presence of positive-only user feedback (*likes*), rather than rating prediction. We argue that positive-only feedback is more common in real applications, where users are usually not inclined to explicitly evaluate the items. Click-through data, browsing history, or item consuming counts are instead more easily acquired by the system, without requiring any effort of the user. However, it must be taken into account that in this setting, information about the users’ *dislikes* is not available, and the fact that a user did not select a particular item might either indicate that the item is unknown to her or that she actually dislikes it.

In order to deal with the different nature of this kind of feedback, we follow Hu et al.’s work [32], where the matrix factorization method was adapted for positive-only feedback. In their model, predictions are still computed using Equation 1, but unlike standard MF that learns the model parameters by only considering the observed ratings, Hu et al.’s method also takes into account the not observed ones. They argue that in this case the commonly used stochastic gradient descent algorithm is no longer efficient, and propose an alternative optimization procedure based on *Alternating Least Squares* (ALS). In our personality-based model we incorporate the same learning approach but for a different prediction model, namely we used that one shown in

Equation 2. Moreover, the resulting loss function penalizes prediction errors over all possible user-item pairs, not only those for which an interaction was observed, and includes the additional model parameters for the personality factors:

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}, \mathbf{Y}) = \sum_u \sum_i c_{ui} (x_{ui} - \hat{x}_{ui})^2 + \lambda (\|\mathbf{P}\|^2 + \|\mathbf{Q}\|^2 + \|\mathbf{Y}\|^2) \quad (3)$$

Here $x_{ui} = 1$ if user u consumed (clicked, liked, purchased) item i , and $x_{ui} = 0$ otherwise. \hat{x}_{ui} is the model's prediction computed using Equation 2. Each row of the matrices $\mathbf{P} \in \mathbb{R}^{|U| \times k}$, $\mathbf{Q} \in \mathbb{R}^{|I| \times k}$, $\mathbf{Y} \in \mathbb{R}^{|A| \times k}$ contains the latent feature vector of a user, an item and an attribute, respectively. The confidence parameter c_{ui} controls how much the model penalizes mistakes in the prediction of x_{ui} , and is set to $c_{ui} = 1 + \alpha k_{ui}$ as proposed in [32]. k_{ui} represents user u 's feedback for item i , which is binary in the case of clicks and *likes*, or a positive number e.g. for item consuming counts, and is set to $k_{ui} = 0$ in the case that no interaction was observed. The constant α models the increase in confidence for observed feedback. Finally, the regularization parameter $\lambda \in \mathbb{R}^+$ is used to prevent overfitting.

The model parameters \mathbf{P} , \mathbf{Q} and \mathbf{Y} are automatically learned by minimizing the loss function over all the user-item training pairs. We extend the method in [32] and derive an ALS-based algorithm with an extra step for the additional \mathbf{Y} parameters of the minimization problem defined in Equation 3. ALS is based on the observation that when all the parameters but one are fixed, (3) becomes a standard least-squares problem with a solution that can be explicitly computed. First, we fix \mathbf{Q} and \mathbf{Y} , and solve the optimization problem analytically for each \mathbf{p}_u by setting the gradient to zero:

$$\mathbf{p}_u = (\mathbf{Q}^T \mathbf{C}^u \mathbf{Q} + \lambda \mathbf{I})^{-1} \mathbf{Q}^T \mathbf{C}^u (\mathbf{x}(u) - \mathbf{Q} \sum_{a \in A(u)} \mathbf{y}_a) \quad (4)$$

where \mathbf{C}^u is a $|I| \times |I|$ diagonal matrix such that $\mathbf{C}_{ii}^u = c_{ui}$, $\mathbf{x}(u)$ is a column vector with all the x_{ui} values for user u . Let for simplicity $\mathbf{z}_u = \mathbf{p}_u + \sum_{a \in A(u)} \mathbf{y}_a$. We then fix \mathbf{P} and \mathbf{Y} and optimize each \mathbf{q}_i in a similar fashion:

$$\mathbf{q}_i = (\mathbf{Z}^T \mathbf{C}^i \mathbf{Z} + \lambda \mathbf{I})^{-1} \mathbf{Z}^T \mathbf{C}^i \mathbf{x}(i) \quad (5)$$

Analogously, \mathbf{C}^i is a $|U| \times |U|$ diagonal matrix where $\mathbf{C}_{uu}^i = c_{ui}$, $\mathbf{x}(i)$ is a column vector with all the x_{ui} values, and the matrix \mathbf{Z} contains the \mathbf{z}_u vectors as rows. Finally, we fix \mathbf{P} and \mathbf{Q} , and optimize for each \mathbf{y}_a :

$$\mathbf{y}_a = \left[\mathbf{Q}^T \left(\sum_{u \in \mathcal{U}(a)} \mathbf{C}^u \right) \mathbf{Q} + \lambda \mathbf{I} \right]^{-1} \sum_{u \in \mathcal{U}(a)} \mathbf{Q}^T \mathbf{C}^u (\mathbf{x}(u) - \mathbf{Q} \mathbf{z}_{u \setminus a}) \quad (6)$$

where $\mathcal{U}(a) = \{u \in U \mid a \in \mathcal{A}(u)\}$ is the set of users that have attribute a , and $\mathbf{z}_{u \setminus a} = \mathbf{p}_u + \sum_{b \in A(u), b \neq a} \mathbf{y}_b$ is defined as before but without including attribute a . Note that, unlike the case of user and item attributes, re-computing

an attribute vector depends on the current state of all the other attribute features through the $\mathbf{z}_{u \setminus a}$ vector.

During training, we alternate between three steps fixing a different set of latent feature parameters each time. This process is repeated for a fixed number of iterations T , as depicted in Algorithm 1. Finally, once the training

Algorithm 1 ALS for user personality-based Matrix Factorization

```

procedure TRAIN
  for  $iter \leftarrow 1, T$  do
    P step: Fix  $\mathbf{Q}, \mathbf{Y}$  and optimize all  $\mathbf{p}_u$  in parallel using Eq. 4
    Q step: Fix  $\mathbf{P}, \mathbf{Y}$  and optimize all  $\mathbf{q}_i$  in parallel using Eq. 5
     $\triangleright$  Computation of attribute vectors cannot be parallelized
    Y step:
      for all  $a \in A$  do
        Fix  $\mathbf{P}, \mathbf{Q}, \mathbf{y}_{b \neq a}$  and optimize  $\mathbf{y}_a$  using Eq. 6
      end for
    end for
  end procedure

```

stage is complete, we use the learned parameters to compute predictions for the test users using Equation 2. For each user, we estimate all the scores for unknown items and sort them in descending order. The top ten items of the list are recommended to the user as the more likely to be relevant.

Computational complexity. Similarly to Hu et al.’s model, the complexity of our personality-aware MF method is $\mathcal{O}(k^3|U| + k^2|R_+|)$ for the P-step and $\mathcal{O}(k^3|I| + k^2|R_+|)$ for the Q-step, where $|R_+|$ is total number of observed ratings. Here we have used an optimization described in [32] to reduce the complexity from $|U| \cdot |I|$ to $|R_+|$ terms. We refer the reader to that paper for more details. In these steps, the latent feature vectors can be easily computed in parallel within each step. The main computational cost relies on the Y-step, in which we have to iterate over the whole $\mathcal{U}(a)$ set for each attribute. Updating all the attribute vectors has complexity $\mathcal{O}(k^3|A| + k^2|A|(|I| + |R_+|))$, with the drawback that it cannot be parallelized since the re-computation of each attribute vector depends on the current state of the others. We note, however, that the number of attributes $|A| = 25$ is small, and the overhead required by the additional latent features is acceptable, making the complexity of our algorithm comparable to that of standard ALS-based MF. Also, we are considering exactly five attributes for each user, one for each dimension of the FFM, so $|\mathcal{A}(u)| = 5$, and recommendations are computed faster.

3.2 Personality-based Active Learning

In this section we describe the three AL methods that we have considered and compared in the experiments. The first one, *Personality-Based Binary-*

Predicted was originally proposed in [18, 4, 5]. It first transforms a given rating matrix to a binary matrix, by mapping null entries to 0, and not null entries to 1. Hence, this new matrix models if a user rated or not an item, regardless of its value [37]. Afterward, from this rating matrix it derives an extended matrix factorization model that profiles users not only in terms of their binary ratings, but also using their Big Five personality traits. Hence, by selecting the items with the highest score this method attempts to identify what the user has most likely experienced, in order to maximize the likelihood that the user can provide the requested rating.

In this paper, we applied this strategy to our scenario where positive-only user feedback are given. Thus, the transformation step becomes unnecessary and we can directly learn the personality-based matrix factorization model as in Equation 2. This is used to predict and assign a ratable score to each candidate item i (for each user u). Higher predicted scores indicate a higher probability that the target user has consumed (liked) the item i , and hence may be able to rate it. This will maximize the chance that the selected items are actually familiar to the user, and hence they are ratable. Selecting items that are more familiar to the users, will result in collecting more number of ratings.

The second method is *Binary-Predicted* [22, 19]. It is identical to the personality-based binary-predicted method except that users are modeled only in terms of their binary ratings, ignoring their Big Five personality traits, i.e., instead of using the extended personality-based matrix factorization model shown in Equation 2, it adopts the standard one, as in Equation 1.

Finally, we have considered *Entropy0* [55, 26, 27]. This method measures the dispersion of the ratings for an item, and attempts to combine the effect of the popularity with the diversity of the ratings, which may provide more useful (discriminative) information about the user's preferences. *Entropy0* score is computed by using the relative frequency of each of the two possible rating values, i.e., *like* (mapped to 1) and *unknown* (mapped to 0):

$$Entropy0(i) = - \sum_{r=0}^1 p(x_{ui} = r) \log p(x_{ui} = r) \quad (7)$$

where $p(x_{ui} = r)$ is the probability that the item rating x_{ui} is equal to r .

This method returns 0 for an item i if and only if its rating value is certain, i.e., if $p(x_{ui} = 0) = 1$ or $p(x_{ui} = 1) = 1$. In contrast, it returns the maximum score for an item i if its two rating values are equally distributed, i.e., $p(x_{ui} = 0) = \frac{1}{2}$ and $p(x_{ui} = 1) = \frac{1}{2}$; in this case, $Entropy0(i) = \log 2$. Since an item is liked by only a small fractions of the users, the score computed by this method essentially measures the popularity of the item: entropy grows with the number of likes when the probability to be liked is small.

3.3 Personality-based Cross-domain Collaborative Filtering

In this section we present the third technical contribution of this paper, a cross-domain rating prediction method. We hypothesize that personality information can be exploited to enhance cross-domain recommendations by enriching user profiles not only with preferences from auxiliary domains but also by leveraging the Big Five scores of the user.

In order to understand the effect of personality on cross-domain recommendation, we first adapt the personality-based matrix factorization method proposed in Section 3.1 by replacing personality attributes with cross-domain ratings. Let \mathcal{S} be a source domain, \mathcal{T} the target domain, and $I_{\mathcal{S}}, I_{\mathcal{T}}$ their respective sets of items. We estimate the user u 's preference for item $i \in I_{\mathcal{T}}$ as

$$\hat{x}_{ui} = \mathbf{q}_i \cdot \left(\mathbf{p}_u + \sum_{j \in I_{\mathcal{S}}(u)} \mathbf{y}_j \right) \quad (8)$$

where $I_{\mathcal{S}}(u)$ is the set of items in the source domain for which user u expressed a preference. This method is a simple extension of SVD++ [37] that expands the user's latent representation in the target domain \mathbf{p}_u with latent feature vectors modelling the effect of user feedback in a source domain. Another difference relies on the training algorithm, which is here based on ALS instead of stochastic gradient descent, as described in Section 3.1. It is worth noticing that in order for this model to be successful the sets of users from the source and target domains must overlap. Even when there are users with data in both domains, the preferences from the source domain may not be relevant for recommendation in the target domain, which is another limitation of the approach. Intuitively, user *likes* from an unrelated source domain such as restaurants are not indicative of her tastes in e.g. music target domain.

Then, we combine both user personality and source domain preferences into a common set of user attributes. We aim to understand if personality information can be used to enhance cross-domain recommendations in the cold-start stage, or if, on the other hand, only cross-domain preferences are useful to improve the system performance. Specifically, we predict user preferences as follows:

$$\hat{x}_{ui} = \mathbf{q}_i \cdot \left(\mathbf{p}_u + \sum_{a \in \mathcal{A}(u)} \mathbf{y}_a + \sum_{j \in I_{\mathcal{S}}(u)} \mathbf{y}_j \right) \quad (9)$$

The above model is also trained using the ALS technique described in Section 3.1, and despite its simplicity we believe it is, to the best of our knowledge, the first attempt to enhance cross-domain matrix factorization with personality information. We note that, differently from the personality-based method that we have previously described, the number of parameters is here much larger, which has a direct impact on the complexity of the

learning process. We are nevertheless interested in comparing the benefits of personality information and cross-domain preferences for new users, and thus use the same recommendation model for both.

4 Experimental Evaluation

4.1 Dataset

The dataset used in our experiments is part of the database made publicly available in the myPersonality project⁶ [2]. myPersonality is a Facebook application with which users take psychometric tests and receive feedback on their personality factor scores. The users allow the application to record personal information from their Facebook profiles, such as demographic and geo-location data, *likes*, status updates, and friendship relations, among others. In particular, as of March 2015, the tool instantiated a database with 46 million Facebook *likes* of 220,000 users for 5.5 million items of diverse nature – people (actors, musicians, politicians, sportsmen, writers, etc.), objects (movies, TV shows, songs, books, video games, etc.), organizations, events, etc. – and the Big Five scores of 4 million users, collected using 20 to 336 item IPIP questionnaires.

Due to the size and complexity of the database, in this paper we restrict our study to a subset of it. Specifically, we selected the *likes* assigned to the items belonging to one of the following three domains: books, movies and music. To determine which items in the original database belong to each of such domains, we used Facebook item categorization data. Specifically, we manually identified certain categories for each domain, e.g. “Music genre”, “Musician/Band”, “Album” and “Song” for the music domain.

Such categories were not always assigned correctly. For instance, there were many music “Albums” annotated with the “Musician/Band” category. Moreover, the names of the items were not always correct, e.g., some of them contained misspellings, and often were not used in a single, concise way, e.g., they were given in terms of morphological deviations, such as **science fiction**, **science-fiction**, **sci-fi** and **sf**.

In order to address the above issues – checking misspellings, unifying morphological deviations, and rectifying categorizations – we performed a number of transformations that consolidated incorrect and duplicate items with correct ones, while exploiting external knowledge to set the items categories. Since it is outside of the focus of this paper, we do not enter into details about the mentioned data transformations. We just mention that such operations were proposed in previous work [67,11], and have been validated by automatically mapping the processed names of the items with the URIs of entities in

⁶ myPersonality project, <http://mypersonality.org>

Table 1 Statistics of the used dataset

Domain	Books	Movies	Music
Number of users	91,854	141,123	145,476
Number of items	4,543	5,389	6,371
Number of likes	355,112	1,837,152	2,835,329
Avg. number of likes/user	3.87	13.02	19.49
Number of users with ≥ 20 likes	1,208	26,951	43,702

DBpedia⁷ [41] (the Wikipedia ontology) via SPARQL⁸ queries; we discarded those items that could not be mapped to DBpedia entities. For instance, in the music domain, those items whose names were consolidated as `mozart`, were mapped to `http://dbpedia.org/page/Wolfgang_Amadeus_Mozart`, and maintained as a single item in the final dataset.

The whole process was conducted on the 6,500 most popular items in the dataset, i.e., the items with highest numbers of *likes*. Note that this may favor the good performance of popularity-based recommendation methods, as we shall observe in the next section. The final dataset is described in Table 1. It consists of 5,027,593 *likes* from 159,551 users on 16,303 items. Its minimum, maximum and average (standard deviation) numbers of *likes* per user are 1, 164 and 3.87 (4.46) for books, 1, 741 and 13.02 (18.78) for movies and 1, 648 and 19.49 (28.80) for music. We note that in order to be able to evaluate the effectiveness of using personality on user with various degrees of coldness (i.e., containing different numbers of *likes*), only users that entered a minimum of 20 *likes* where considered. After that, there were 1,208 users in the book domain, out of which 1,200 (99.34%) and 1,190 (98.51%) had at least one preference in the movie and music domains, respectively; 26,951 users in the movie domain, out of which 23,826 (88.40%) and 26,810 (99.48%) had also preferences in the book and music domains, respectively; and finally, 43,702 users in the music domain, out of which 34,215 (78.29%) and 43,134 (98.70%) with also book and movie preferences, respectively.

4.2 Evaluation Setting

The evaluation of the proposed methods (i.e., personality-based matrix factorization CF, personality-based active learning as well as personality-based cross-domain recommendation) was conducted utilizing a modified user-based 5-fold cross-validation strategy, based on Kluver et al.’s methodology for cold-start evaluation [36].

Our goal is to understand how the different methods perform as the number of observed *likes* in the target domain increases. First, we divide the set

⁷ <http://http://dbpedia.org>

⁸ <http://www.w3.org/TR/rdf-sparql-query>

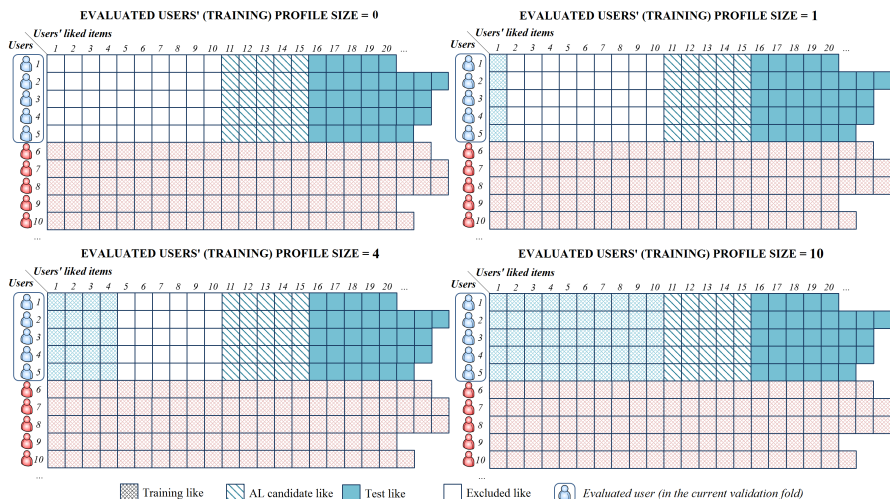


Fig. 2 Overview of evaluation setting in a given cross-validation fold. Each test user u 's data is split into training, candidate, and testing sets. Different cold-start profile sizes are simulated by sequentially adding *likes* to each u 's training set.

of users into five subsets of roughly equal size. In each cross-validation stage, we keep all the data from four of the groups in the training set. Then, for each user u in the fifth group –the test users– we randomly split her *likes* into three subsets, as depicted in Figure 2: (i) a *training set*, initially empty and incrementally filled with u 's *likes* one by one to simulate different cold-start profile sizes, (ii) a *candidate set* containing the set of *likes* to be elicited by the active learning strategies, and (iii) a *testing set* used to compute the performance metrics.

The above procedure was modified for the cross-domain scenario by extending the training set with the full set of *likes* from the auxiliary domain, in order to obtain the actual training data for the predictive models. Similarly, this evaluation strategy was further modified to test the performance of the active learning strategies. In particular, the evaluation of an active learning method for a specific user profile size closely follows the evaluation approach proposed by Elahi et al. [22], and proceeds in the following way:

1. The performance metrics are measured on the testing set, after training the rating prediction model (in our case, Implicit Matrix Factorization) on the training set.
2. For each user in the testing set:
 - (a) Using the active learning method, the top $N = 5$ candidate items that are not yet in the training set are computed for rating elicitation.
 - (b) Assign to the training set the user's likes for these candidate items as found in the candidate set, if any.

3. The performance metrics are measured again on the testing set, after having re-trained the rating prediction model on the new, extended training set.

We adopted three widely used accuracy and ranking metrics for positive-only feedback (i.e., one-class collaborative filtering) [71]: *Mean Average Precision (MAP)*, *Half-Life Utility (HLU)* and *Mean Percentage Ranking (MPR)*. We also measured two metrics for assessing novelty and coverage: *Average-Popularity* and *Spread*.

- *MAP* measures the overall precision performance based on precision at different recall levels [43]. It is calculated as the mean of the average precision (AP) over all users in the test set. A larger *MAP* corresponds to a better recommendation performance.
- *HLU* measures the utility of a recommendation list for a user, with the assumption that the likelihood that the user will view/choose a recommended item decays exponentially with the item’s ranking [7, 51]. A larger *HLU* corresponds to a better recommendation performance.
- *MPR* estimates the user satisfaction of items in a ranked recommendation list, and is calculated as the mean of the percentile ranking of each test item within the ranked list of recommended items for each test user [32]. It is expected that a randomly generated recommendation list would have a *MPR* of around 50%. A smaller *MPR* corresponds to a better recommendation performance.
- *AveragePopularity* measures how novel the recommendations (or items requested to be liked, as for active learning) are to the user [72]. It is expected that users prefer lists containing more novel (less popular) items. However, if the presented items are too novel, then the user is unlikely to have any knowledge of them, and will not be able to understand or rate them. Therefore, moderate values indicate a better performance [36].
- Finally, *Spread* is a metric of how well the recommender or active learning method spreads its attention across many items [36]. It is assumed that algorithms with a good understanding of its users are able to provide different users with different items. However, like novelty, one could not expect to achieve a perfect spread (presenting each item an equal number of times) without making avoidably bad recommendations or unfulfillable rating requests. Hence, moderate values are preferable.

In our experiments we observed equivalent behaviour of the methods in terms of MAP, HLU, and MPR. Hence, for brevity, we only report MAP values in the analysis presented in Section 5.

5 Experiment Results

5.1 Evaluating Personality-based Matrix Factorization

The goal of a first set of experiments was to understand if personality information can be used to improve the performance of matrix factorization for positive-only feedback in cold-start situations. Using the methodology described in Section 4.2, we computed HLU, MAP and MPR for different amounts of observed *likes* for items in the training set of the target domain. Specifically, we distinguish between two different scenarios:

- *Extreme cold-start*, in which there are no *likes* at all from the active user, and recommendations are computed only on the basis of personality information.
- *Moderate cold-start*, in which we assume that only one *like* is given, and incrementally evaluate the system performance with larger and larger profile size of the active user. We aim at understanding how the different methods behave as the amount of available ratings/likes increases.

We compare our proposed personality-based matrix factorization method (**Personality MF**), which computes predictions using Equation (2), against Hu et al.’s method (**iMF**), which uses Equation (1) and does not use any auxiliary information. We also include a non-personalized baseline that always recommends the most popular items (**Most popular**). Results in terms of MAP@10 for the *extreme cold-start* scenario are shown in Figure 3, for the three domains available in our dataset. The results for HLU and MPR were very similar and we therefore do not report them here. We note that the small values obtained are due to the large item catalogues in our dataset. The set of possible candidate items to recommend for each test user is therefore also large, which leads to a low probability of matching a test item in the user’s recommendation list.

We see that in all cases Personality MF significantly outperforms iMF and the Popularity-based baseline (Wilcoxon signed-rank test, $p < 0.05$). Our personality-based method is specially beneficial in the books and music domains, where it achieves relative improvements of 64% and 94%, respectively. The relationships between user personality and preferences seem to be stronger in these domains, although a more exhaustive analysis is required to confirm this. Nonetheless, we could conclude that personality information is highly beneficial in the extreme cold-start situation, and that it is able to mitigate the total absence of user ratings/likes and recommend relevant items.

In Figure 4 we show the performance of the different methods as the profile size of the new user increases, again in terms of MAP@10. The Most popular baseline is clearly not a competitive approach now, and the personalized methods perform better and better as more ratings are available. We do not appreciate a difference in performance between iMF and Personality

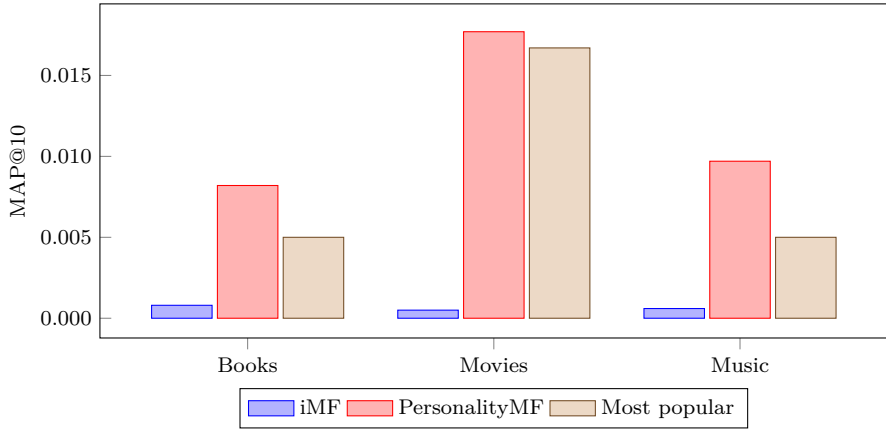


Fig. 3 MAP@10 in the extreme cold-start scenario.

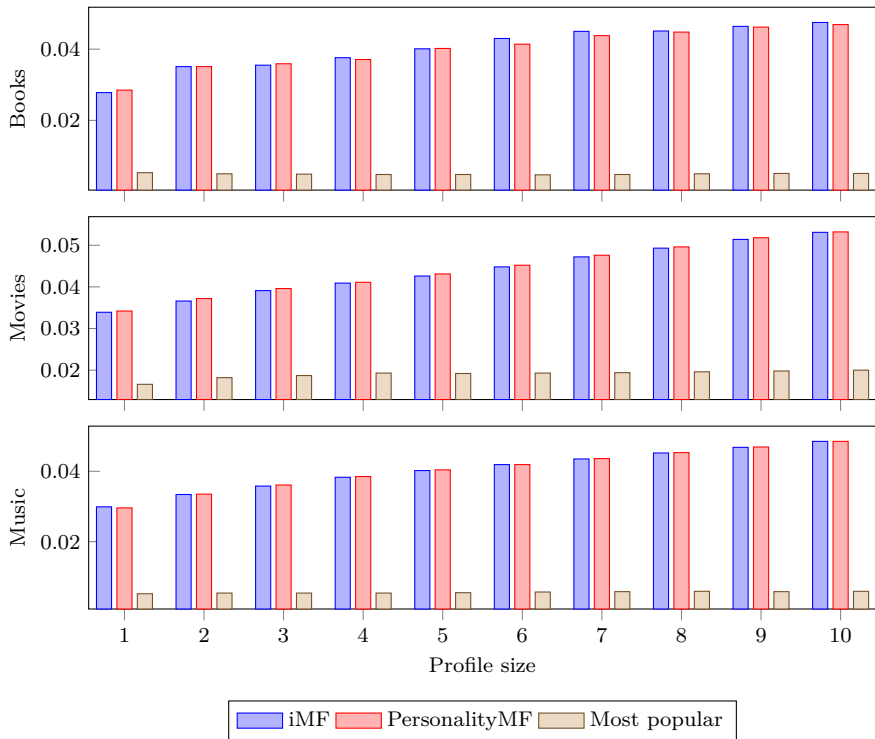


Fig. 4 MAP@10 for different cold-start user profile sizes.

Table 2 Novelty and coverage of collaborative filtering methods in the cold-start. Results for the *moderate* scenario with profile sizes 1–10 are stable, hence we report the average.

Domain	Method	Extreme cold-start		Moderate cold-start	
		Avg. Popularity	Spread	Avg. Popularity	Spread
Books	iMF	7.12	3.32	142.80	6.29
	Personality MF	185.19	5.26	144.59	6.27
	Most popular	231.26	3.32	237.04	3.47
Movies	iMF	186.80	3.32	4056.75	6.43
	Personality MF	5717.94	4.75	4080.65	6.43
	Most popular	6447.28	3.32	6637.56	3.47
Music	iMF	311.36	3.32	6565.61	6.87
	Personality MF	9846.59	4.73	6592.58	6.86
	Most popular	10 877.38	3.32	11 113.77	3.45

MF in any of the domains, indicating that personality information is not determinant once *likes* data can be exploited. Our results differ from those reported in [31], where it was shown that the user-based nearest neighbors method enhanced with personality clearly achieves better performance than using only the ratings, for users in a music recommender system with 2, 5, and 10 ratings. It is worth noting that in this work we report results in a different, bigger dataset (43,702 vs. 111 users, see Table 1) composed of *likes* (positive-only feedback) instead of ratings. Also, we analyze the effects of integrating personality in the matrix factorization method, as opposed to user-based nearest neighbors, and evaluate the performance for users completely new to the system.

We conclude that, in terms of accuracy, personality proves useful for completely new users in the three analysed domains. In the remaining cases, iMF is competitive enough, and does not require any additional information. We argue, nonetheless, that the extreme cold-start is a critical stage of a recommender system; the system must keep the user engaged, and exploiting personality is a good option to find relevant items for the user. Also, once some *likes* are observed, more subtle relations between user preferences and personality could be unveiled by taking into account additional variables using more fine-grained representations of personality as suggested in [48].

In addition to accuracy, we also analyzed the coverage and the ability of the different methods to provide novel recommendations. In Table 2 we show the average popularity and the spread of the recommended items by each method. We observe similar behavior in all the considered domains: Personality MF and iMF on average recommend items with the same moderate popularity, except for completely new users. In that case, Personality MF recommends less novel items but still not simply the most popular ones — between 9.5% and 20% less popular on average, compared to the baseline. In terms of coverage, the personalized methods recommend more varied items than the Most popular method, which always suggests the same set of

items. We again see that without any available *likes*, personality-based MF approaches the behavior of the Most popular baseline. It is worth noting that in the extreme cold-start situation the coverage of iMF is similar to the Most popular method, while Personality MF is much better in that respect.

5.2 Evaluating Personality-based Active Learning

In this section we present the experiment results for the proposed active learning methods. We first illustrate the number of *likes* elicited by the methods. Then, we present their performance in term of accuracy and ranking quality. Finally, we show their performance with respect to the novelty and coverage metrics.

5.2.1 Number of Acquired Likes

The number of acquired ratings is an important measure of the performance of an active learning method. In fact, certain methods can elicit more ratings by better estimating what items the user has actually experienced, and is therefore able to rate.

Figure 5 shows the average number of acquired *likes* for users having a profile size of 0 (i.e., completely cold users). We can observe that the best active learning method for all the considered domains is the *personality-based binary-predicted* method, which is able to acquire, on average per user, 0.142, 0.165 and 0.153 *likes* in the books, movies and music domain, respectively (out of a maximum of 5). It outperforms the second-best method, *Entropy0*, that is able to elicit 0.125 (for books), 0.155 (for movies) and 0.147 (for music) *likes*. The corresponding p-values were 0.10, 0.03 and 0.03 for books, movies and music, respectively, according to Wilcoxon signed-rank tests, and thus marginally statistically significant to statistically significant. This shows that by exploiting the user’s personality our proposed personality-based binary-predicted method can better estimate which items may have been experienced and liked by completely new users whose preferences are unknown.

Figure 6 illustrates the average number of acquired *likes* for new users having from 1 up to 10 *likes* in their profiles. The first observation is that the difference between the *personality-based binary-predicted* method and the standard *binary-predicted* method vanishes when at least one training like is available. As already observed in the comparison between *Personality MF* and *iMF* in Section 5.1, this suggests that at this stage the effects of personality exploitation in the underlying rating prediction model of the *binary-predicted* method are minor to non-existent. In any case, it is clear from the figure that by personalizing the rating elicitation process as done by both the *personality-based* and the *binary-predicted* methods we are able to obtain a considerably larger number of *likes* from the user; in all domains, the

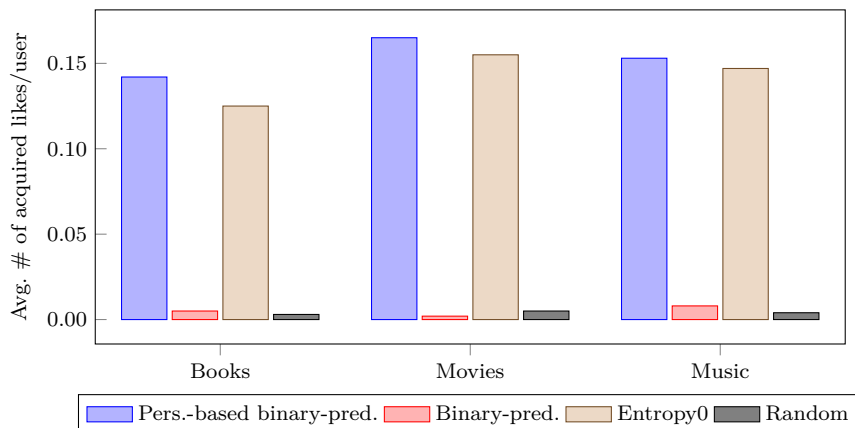


Fig. 5 Average number of user *likes* acquired in the extreme cold-start scenario.

personalized active learning methods systematically perform better than the *Entropy0* and *Random* methods.

5.2.2 Accuracy and Ranking Performance

An active learning method may not be able to elicit a large number of ratings, but those actually elicited can be very useful for improving future recommendations, either for the target user or for other users in the system. Therefore, it is essential to inspect how the accuracy and ranking quality of the generated recommendations changes based on the acquired ratings/likes.

Figure 7 shows the MAP results for completely new users (i.e., profile size = 0) when the cut-off level was set to 10 (i.e., MAP@10). Again, the results for HLU and MPR were equivalent, so we do not report them for brevity. It can be seen from the figure that the proposed *personality-based binary-predicted* method does not only elicit the largest number of *likes* from users without any *likes* history, but also leads to the highest improvement in MAP compared to all other tested methods. The MAP of the recommender increased to 0.012, 0.015 and 0.009 for books, movies and music, respectively, after extending the set of training *likes* with the *likes* acquired via the *personality-based binary-predicted* method. The second-best active learning method is *Entropy0*, which is able to achieve a MAP of 0.011 (for books), 0.015 (for movies) and 0.008 (for music). These differences in MAP between personality-based binary-predicted and *Entropy0* are, however, not statistically significant (Wilcoxon signed-rank test), except for the music domain ($p = 0.03$). As expected, both the *binary-predicted* and *Random* active learning methods fail to achieve any noteworthy improvements in terms of MAP.

Finally, when considering users having between 1 and 10 *likes* in their profile, only minor improvements in terms of the system MAP were achieved

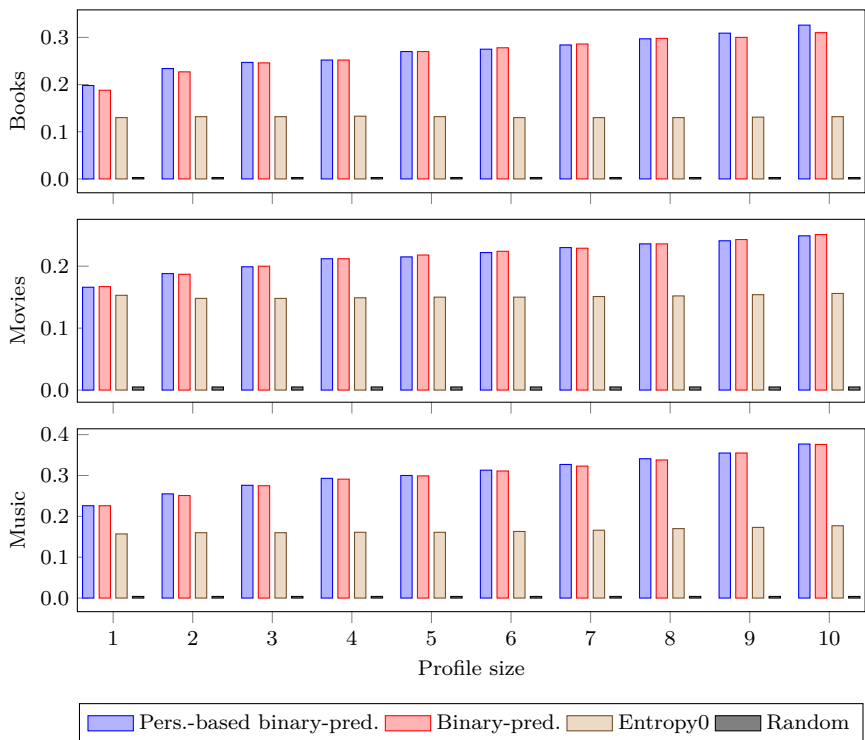


Fig. 6 Average number of user *likes* acquired for different cold-start user profile sizes.

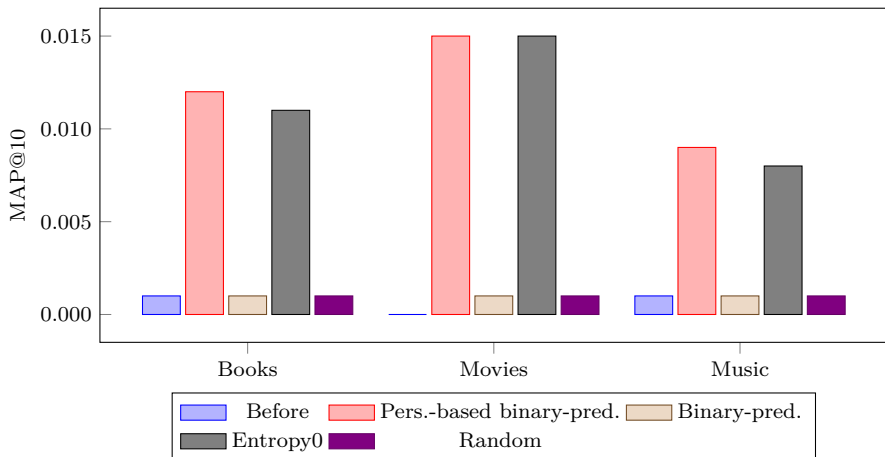


Fig. 7 MAP@10 before and after applying AL in the extreme cold-start scenario.

Table 3 Novelty and coverage of AL strategies in the cold-start. Results for the *moderate* scenario with profile sizes 1–10 are stable, thus we report the average.

Domain	AL method	Extreme cold-start		Moderate cold-start	
		Avg. Popularity	Spread	Avg. Popularity	Spread
Books	Pers.-based bin.-pred.	234.07	3.11	175.05	3.91
	Binary-pred.	4.80	1.61	172.22	3.94
	<i>Entropy0</i>	293.60	1.61	298.22	1.74
	Random	6.89	6.89	7.14	6.89
Movies	Pers.-based bin.-pred.	5765.34	2.54	3401.59	4.13
	Binary-pred.	37.92	1.61	3413.81	4.15
	<i>Entropy0</i>	5942.56	1.61	6492.61	1.76
	Random	133.65	8.01	138.19	8.01
Music	Pers.-based bin.-pred.	10 701.71	2.80	7230.68	4.41
	Binary-pred.	517.60	1.61	7207.72	4.42
	<i>Entropy0</i>	11 840.96	1.61	12 065.60	1.70
	Random	252.38	8.59	258.12	8.59

by applying active learning. Therefore, the active learning methods seem to be little effective in improving the system accuracy and ranking quality in case the users are already known by the system (i.e., they have at least one *like*).

5.2.3 Novelty and Coverage

When evaluating an active learning method, it is not only important to know how it affects the recommender system performance, but it is also important to understand how users would react to the system requests. For that reason, as we mentioned in Section 4.2, we propose to measure the *AveragePopularity* and the *Spread* of the items requested by the system to rate.

Table 3 shows the average popularity and the spread of items requested to be liked by each active learning method. As can be seen, in terms of average popularity, the baseline *Entropy0* method requests users to provide ratings for the most popular items, followed, with a significant gap, by the *personality-based binary-predicted* and *binary-predicted* methods, and then *Random*, which obviously has the lowest average popularity and the largest spread. As stated earlier, even though users are able to rate many of the items presented by *Entropy0*, we expect that they will feel these items as too popular and poorly adapted to their preferences and interests. On the other hand, the popularity of the items provided by random is too low, which turns out to be the reason for its low number of elicited *likes*. The *personality-based binary-predicted* and *binary-predicted* methods perform well, with the former also working for users with 0 training *likes*.

Table 3 also shows that, as expected, *Entropy0* has the lowest spread, as it essentially requests to rate a small set of (popular) items. Also, as expected,

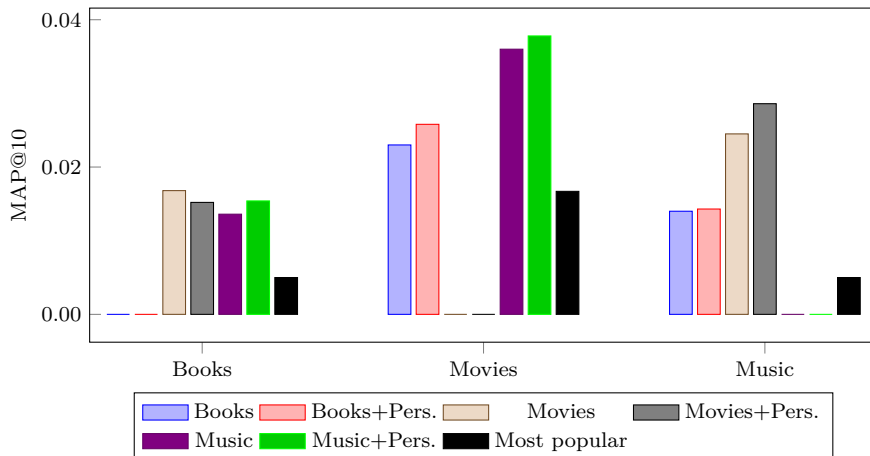


Fig. 8 MAP@10 of cross-domain methods in the extreme cold-start scenario. The x axis represents the target domain, and bars correspond to recommendation methods with different combinations of source domain and personality.

the best spread result is obtained by the *Random* method, in which every item in the catalog, regardless of whether it is very popular (ratable) or not at all popular (not ratable), has the same chance to be presented to the user. Again, personality-based binary-predicted (for all profile sizes) and binary-predicted (for profile sizes ≥ 1) yield the best trade-off between high spread and high ratability.

5.3 Evaluating Personality-based Cross-domain Collaborative Filtering

The goal of the last set of experiments is to test whether personality information can be exploited to further enhance cross-domain techniques in cold-start situations. We aim to validate our hypothesis that personality can be effectively combined with cross-domain user preferences in the matrix factorization framework to address the cold-start problem.

In the following we compare the matrix factorization method that uses only cross-domain ratings as attributes, and computes predictions using Equation (8) (methods *books*, *movies*, and *music*, depending on the source domain), and cross-domain ratings combined with personality as in Equation (9), which we refer to by adding the *+personality* suffix. Note that these methods differ from those reported in Section 5.1 as they exploit information from a source domain. We show in Figure 8 the performance of the different methods in the *extreme cold-start* scenario.

In two cases out of three, combining personality information with cross-domain ratings further improves the performance when no preferences about the user are available in the target domain. Only in the **books** domain, the

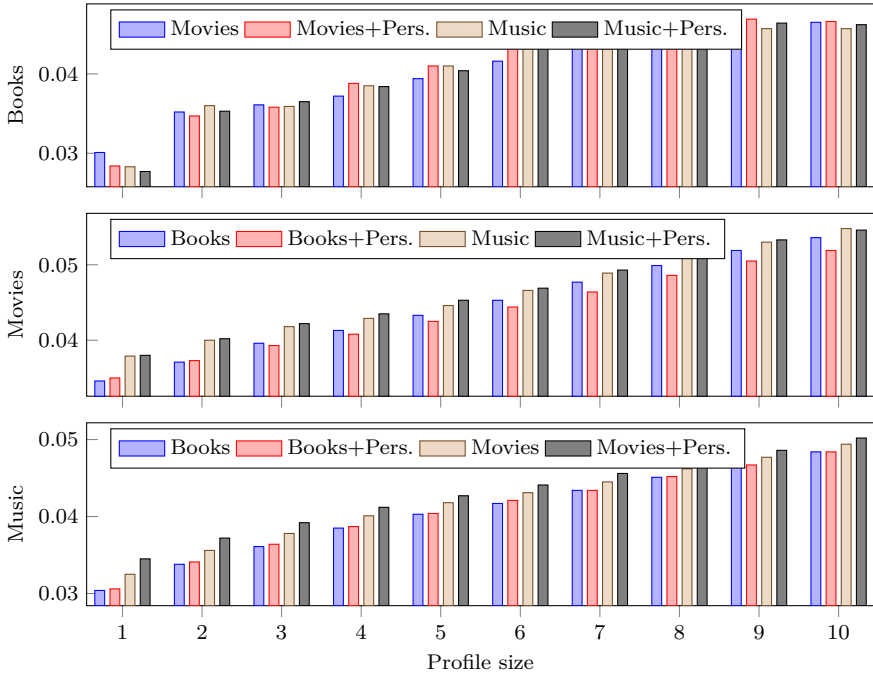


Fig. 9 MAP@10 of cross-domain methods for different profile sizes.

best results are obtained using movie data only. In this case, adding personality information does not improve the performance, but it is beneficial if the available auxiliary information consists of music ratings (13.2% relative improvement over the cross-domain method without personality). When predicting **movie** preferences, we observe that cross-domain methods enhanced with personality information always achieve better performance. In fact, the overall best results are obtained by combining music preferences and personality (5% improvement of Music+Pers. over Music), and if only book *likes* are available as auxiliary information, the accuracy can be further improved by considering personality (by 12.2%). In the case of **music** recommendations, we observe a symmetrical trend, where the best results are achieved combining personality with movie *likes* (16.7% improvement of Music+Pers. over Music). On the other hand, adding book *likes* is clearly beneficial to the system but in this case exploiting personality information yields only a minimal improvement.

The results for the *moderate* cold-start are shown in Figure 9. Differently from the *extreme* cold-start scenario, we cannot conclude that personality is beneficial for larger user profile sizes in the **books** domain. In the case of **movies**, we obtain small improvements combining personality with music ratings, but the effect is the opposite when dealing with book ratings.

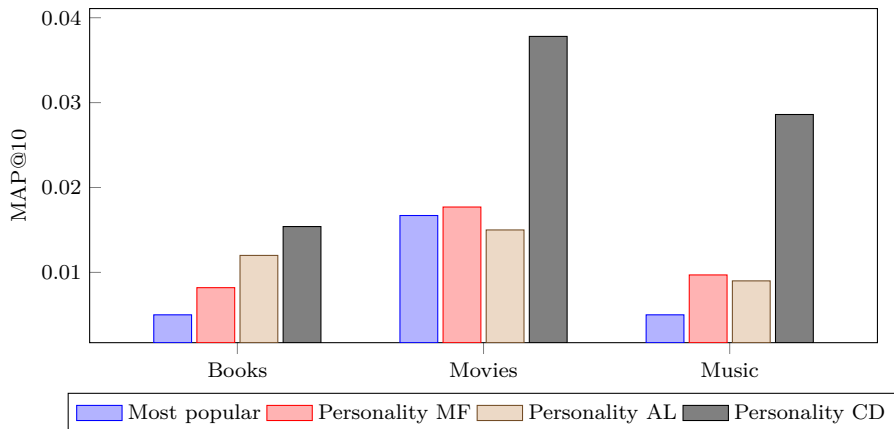


Fig. 10 Comparison of the best performing methods in the extreme cold-start situation.

Finally, when recommending **music**, we clearly see the advantages of combining personality with auxiliary movie ratings, which consistently gives the best overall results.

Regarding *novelty*, the average popularity of the recommended items by the cross-domain methods is very similar, and much lower than the Most popular baseline as expected (on average, 145–148 vs. 236 in the books domain, 3,980–4089 vs. 6,620 for movies, and 6,622–6,753 vs. 11,092 for music). In terms of *coverage*, the spread of the item distribution is again similar among cross-domain methods, whereas it is much lower for the baseline (on average, 6.23–6.25 vs. 3.45 in books, 6.40–6.48 vs. 3.45 in movies, and 6.78–6.88 vs. 3.44 in music).

5.4 Discussion

We now discuss the results achieved by all the tested methods to address the new user problem, focusing on the benefits of the personality-based methods that we propose in this work. In Figure 10 we compare the MAP@10 values of the best-performing methods for the new user problem in the *extreme* cold-start situation, i.e., for users completely new to the system.

The *Personality-based cross-domain* (Personality CD) method is the best performing method in all the considered domains, as it uses more information than MF and AL approaches. The boost in precision, specially in the movies and music domains, comes at the cost of collecting the auxiliary information and the time required to train the models. It can be a compelling approach if cross-domain ratings are available at the time of designing the target system – e.g. if the catalog of items is expanded with a new domain – and the goal is to optimize for precision regardless of the training complexity.

The *Personality-based active learning* (Personality AL) method is a good alternative when the new user situation is extreme, as the proposed preference elicitation process minimizes the effort required from the user. We argue that additional aspects such as the design of the user interface are of great importance here. Although the recommendation model has to be trained again after the rating acquisition, the complexity in this case is much lower than with cross-domain methods. Also, the improvements in terms of precision are notable in the books and music domains. In the case of movies, we see that additional elicited *likes* are needed for the baseline iMF recommender to achieve better performance, as users seem to favor popular movies. However, there is a clear trade-off between the effort required from the user and the gain in system performance.

When no auxiliary *likes* are available and there is no chance to ask the user to rate some items, the *Personality-based Matrix Factorization* (Personality MF) method effectively exploits personality information when the new user problem is extreme. The proposed Personality MF method is fast to train and provides better precision than the Most popular baseline and standard iMF, which is unable to compute meaningful recommendations. In the moderate new user situation, as more *likes* are available, we did not find significant improvements using personality with respect to iMF. We argue, nonetheless, that being able to provide recommendations for completely new users is a very desirable quality of RS that is worth the acquisition of personality information.

Our conclusions are summarized in Table 4. In the **books** domain, we see that personality-based AL is likely the best approach, as it offers good precision in the extreme new user situation and overall good novelty and coverage. The boost in performance achieved by cross-domain methods may not be worth the extra time required to train the model. For **movies**, personality-based cross-domain is a good candidate approach. It offers roughly twice the precision maintaining good novelty and coverage, and is also able to provide better performance as the number of available *likes* grows (using auxiliary music *likes*, the best performing method). Finally, in the case of **music** recommendations, we find again that personality-based cross-domain CF is a compelling approach. However, due to the large number of *likes* in this domain (see Table 1), the training time is considerably larger than for the rest of the methods. Unless the extra precision is required, a good alternative is Personality MF, which is the second best in terms of precision and offers better novelty and coverage than AL in the extreme cold-start.

6 Conclusions and Future Work

In this paper we have presented three approaches to address the new user cold-start problem in collaborative filtering, namely, personality-based matrix factorization, personality-based active learning, and personality-based

Table 4 Summary of the performance of the different methods.

Domain	Method	Time	Extreme cold-start			Moderate cold-start		
			Prec.	Nov.	Cov.	Prec.	Nov.	Cov.
Books	Most popular	✓						
	Personality MF	✓	✓	✓	✓		✓	✓
	Personality AL	✓	✓	✓	✓		✓	✓
	Personality CD		✓	✓	✓		✓	✓
Movies	Most popular	✓	✓					
	Personality MF	✓	✓	✓	✓	✓	✓	✓
	Personality AL	✓			✓		✓	✓
	Personality CD		✓	✓	✓	✓	✓	✓
Music	Most popular	✓						
	Personality MF	✓	✓	✓	✓		✓	✓
	Personality AL	✓	✓		✓		✓	✓
	Personality CD		✓	✓	✓	✓	✓	✓

cross-domain recommendation. They can be used in different scenarios. For example, if in addition to the target domain, an auxiliary domain knowledge is available, the latter solution could be the best. However, if such knowledge is not available, but the system can request the users to give more ratings, the active learning solution may be preferable. In neither of these scenarios, we can simply incorporate the personality information in the model and improve the classical MF model.

These conclusions have been derived from a number of extensive offline experiments, which allowed us to compare our methods with state of the art techniques. This has been done by designing and implementing an evaluation procedure that simulates user profile evolution, hence, considering different new user cold-start situations: (a) extreme situation, i.e., when there is no single rating/like from the new users, and (b) moderate situation, i.e. when there is at least one like provided by the new users. Moreover, we have conducted the evaluation in terms of several metrics, such MAP (ranking quality), Average Popularity (novelty), and Spread (coverage).

In this work we were not concerned with the actual acquisition of user personality information, and always assumed that it was already available. However, before being able to use such information in the recommendation process, a system has to obtain it from the user. This may be done either explicitly, i.e., by requesting the users to fill up a personality questionnaire, or implicitly, i.e., by analyzing the users' behavioral patterns during the interaction with the system [39]. It has been observed that explicit personality acquisition is more accurate [17]. However, this comes with a cost; the users must provide explicitly additional information (personality information) while they may not be keen on doing that. This is why one must consider accuracy improvement as well as its cost. Indeed, a topic for future work is the automatic inference of a user's personality traits from auxiliary preferences

or social network profiles. This is a challenging subject that is getting a lot of attention from the research community. An effective solution of this problem combined with the exploitation of cross-domain ratings can potentially reduce the effort required from the user, specially in active learning approaches in which also ratings have to be acquired.

The results presented in this paper clearly depend, as in any experimental study, on the chosen simulation setup, which can only partially reflect the real evolution of a recommender system. For example, in this work we assumed that a randomly chosen set of *likes* among those that the user really gave to the system, represents all her known *likes*. However, this set could not obviously represent all the true user likes; it contains only the *likes* expressed by the user while interacting with the system. In fact, many more items are likely to be of interest to the user, but they are not included in the dataset. This is a common problem of any offline evaluation of a recommender system, where the performance of the recommendation method is estimated on a test set that is never a good sample of the true user preferences. Therefore, it is necessary to further evaluate the proposed solutions in alternative evaluation methodologies, and in particular in a live user study.

In recommender systems, the users are interested in recognizing that the entered ratings are immediately used in the recommendations generated by the system. However, an active learning solution, as the one we developed in this paper, chooses a set of items (rather than a single item), and presents them to the user to rate. The system is thus retrained only after the user submits the whole *batch* of ratings. In contrast, in *sequential* active learning the items to be rated are selected one by one, by choosing each successive item to be rated on the base of the user's ratings provided to the previously requested items. For this reason, as an extension of our current batch active learning method, we will implement a sequential (conversational) selection of items [62].

Finally, we stress again that in this paper we used a dataset collected in a popular Facebook social network. This dataset, similarly to other social networks datasets, contains user preference data expressed as *likes* selections. All the not selected items should not be automatically labeled as *dislikes*, as this set will surely contains may items that the user likes. That makes it difficult for the system to infer the users' true preferences. One possibility to solve this problem is to train the system using only the explicit *likes*. However, further studies will be done in order to develop more effective methods that can effectively make use of this type of data.

Acknowledgements This work was supported by the Spanish Ministry of Economy and Competitiveness (TIN2013-47090-C3). We thank Michal Kosinski and David Stillwell for their attention regarding the dataset.

References

1. Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013.
2. Yoram Bachrach, Michal Kosinski, Thore Graepel, Pushmeet Kohli, and David Stillwell. Personality and patterns of facebook usage. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 24–32. ACM, 2012.
3. Shlomo Berkovsky, Tsvi Kufflik, and Francesco Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, 2008.
4. Matthias Braunhofer, Mehdi Elahi, Mouzhi Ge, and Francesco Ricci. Context dependent preference acquisition with personality-based active learning in mobile recommender systems. In *Learning and Collaboration Technologies. Technology-Rich Environments for Learning and Collaboration*, pages 105–116. Springer, 2014.
5. Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci. Techniques for cold-starting context-aware mobile recommender systems for tourism. *Intelligenza Artificiale*, 8(2):129–143, 2014.
6. Matthias Braunhofer, Mehdi Elahi, and Francesco Ricci. User personality and the new user problem in a context-aware point of interest recommender system. In *Information and Communication Technologies in Tourism 2015*, pages 537–549. Springer International Publishing, 2015.
7. John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc, 1998.
8. Jerry M. Burger. *Personality*. Wadsworth Publishing, 2010.
9. Iván Cantador and Paolo Cremonesi. Tutorial on cross-domain recommender systems. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 401–402. ACM, 2014.
10. Iván Cantador, Ignacio Fernández-Tobías, Alejandro Bellogín, Michal Kosinski, and David Stillwell. Relating personality types with user preferences in multiple entertainment domains. In *UMAP’13 Workshops*. Springer, 2013.
11. Iván Cantador, Ioannis Konstas, and Joemon M. Jose. Categorising social tags to improve folksonomy-based recommendations. *Journal of Web Semantics*, 9(1):1–15, 2010.
12. Giuseppe Carenini, Jocelyn Smith, and David Poole. Towards more conversational and collaborative recommender systems. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 12–18. ACM, 2003.
13. Olivia Chausson. Who watches what? assessing the impact of gender and personality on film preferences, 2010.
14. Paul T. Costa and Robert R. McCrae. *Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO FFI): Professional Manual*. Psychological Assessment Resources, 1992.
15. Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. Cross-domain recommender systems. In *Proceedings of the 11th International Conference on Data Mining Workshops*, pages 496–503, 2011.
16. Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 2011.
17. Greg Dunn, Jurgen Wiersema, Jaap Ham, and Lora Aroyo. Evaluating interface variants on personality acquisition for recommender systems. In *User Modeling, Adaptation, and Personalization*, pages 259–270. Springer, 2009.
18. Mehdi Elahi, Matthias Braunhofer, Francesco Ricci, and Marko Tkalcić. Personality-based active learning for collaborative filtering recommender systems. In *The 12th*

- Symposium of the Italian Association for Artificial Intelligence*, volume 8249, pages 360–371. Springer, 2013.
19. Mehdi Elahi, Valdemaras Repsys, and Francesco Ricci. Rating elicitation strategies for collaborative filtering. In Christian Huemer and Thomas Setzer, editors, *Proceedings of the 12th International Conference on E-Commerce and Web Technologies*, volume 85, pages 160–171. Springer, 2011.
 20. Mehdi Elahi, Francesco Ricci, and Neil Rubens. Adapting to natural rating acquisition with combined active learning strategies. In *Proceedings of the 20th International Conference on Foundations of Intelligent Systems*, pages 254–263. Springer, 2012.
 21. Mehdi Elahi, Francesco Ricci, and Neil Rubens. Active learning in collaborative filtering recommender systems. In *E-Commerce and Web Technologies*, pages 113–124. Springer, 2014.
 22. Mehdi Elahi, Francesco Ricci, and Neil Rubens. Active learning strategies for rating elicitation in collaborative filtering: A system-wide perspective. *ACM Transactions on Interactive Intelligent Systems*, 5(1):13:1–13:33, January 2014.
 23. Manuel Enrich, Matthias Braunhofer, and Francesco Ricci. Cold-start management with cross-domain collaborative filtering and tags. In *Proceedings of the 14th International Conference on E-Commerce and Web Technologies*, pages 101–112, 2013.
 24. Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems: A survey of the state of the art. In *Proceedings of the 2nd Spanish Conference on Information Retrieval*, pages 187–198, 2012.
 25. Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. Cross-domain recommendation via cluster-level latent factor model. In *Proceedings of the 2013 European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 161–176, 2013.
 26. Nadav Golbandi, Yehuda Koren, and Ronny Lempel. On bootstrapping recommender systems. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1805–1808. ACM, 2010.
 27. Nadav Golbandi, Yehuda Koren, and Ronny Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 595–604. ACM, 2011.
 28. Lewis R. Goldberg, John A. Johnson, Herbert W. Eber, Robert Hogan, Michael C. Ashton, C. Robert Cloninger, and Harrison G. Gough. The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality*, 40(1):84–96, 2006.
 29. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237. ACM, 1999.
 30. Rong Hu and Pearl Pu. A comparative user study on rating vs. personality quiz based preference elicitation methods. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 367–372. ACM, 2009.
 31. Rong Hu and Pearl Pu. Enhancing collaborative filtering systems with personality information. In *Proceedings of the 5th ACM Conference on Recommender Systems*, pages 197–204. ACM, 2011.
 32. Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 263–272. IEEE, 2008.
 33. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: An introduction*. Cambridge University Press, 2010.
 34. Oliver P. John and Sanjay Srivastava. The big five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 1999. vol. 2, pp. 102-138.
 35. Marius Kaminskas, Ignacio Fernández-Tobías, Francesco Ricci, and Iván Cantador. Knowledge-based identification of music suited for places of interest. *Information Technology & Tourism*, 14(1):73–95, 2014.

36. Daniel Kluver and Joseph A Konstan. Evaluating recommender behavior for new users. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 121–128. ACM, 2014.
37. Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434. ACM, 2008.
38. Yehuda Koren and Robert Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer Verlag, 2011.
39. M. Kosinski, D. Stillwell, and T Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, pages 2–5, March 2013.
40. Michal Kosinski, David Stillwell, Pushmeet Kohli, Yoram Bachrach, and Thore Graepel. Personality and website choice. In *Proceedings of the 3rd Annual ACM Web Science Conference*. ACM, 2012.
41. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.
42. Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 2052–2057, 2009.
43. Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 959–968. ACM, 2010.
44. Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4, Part 2):2065 – 2073, 2014.
45. Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.
46. Sean M. McNee, Shyong K. Lam, Joseph A. Konstan, and John Riedl. Interfaces for eliciting new user preferences in recommender systems. In *Proceedings of the 9th International Conference on User modeling*. Springer, 2003.
47. Carlos Eduardo Mello, Marie-Aude Aufaure, and Geraldo Zimbrão. Active learning driven by rating impact analysis. In *Proceedings of the 4th ACM Conference on Recommender systems*, pages 341–344. ACM, 2010.
48. Maria Augusta S N Nunes. *Recommender Systems based on Personality Traits: Could Human Psychological Aspects Influence the Computer Decision-making Process?* VDM Verlag, 2009.
49. Maria Augusta S. N. Nunes and Rong Hu. Personality-based recommender systems: An overview. In *Proceedings of the 6th ACM Conference on Recommender Systems*, pages 5–6, 2012.
50. Ante Odic, Marko Tkalcić, Jurij F Tasic, and Andrej Kosir. Personality and social context: Impact on emotion induction from movies. In *UMAP'13 Workshops*, 2013.
51. Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 502–511. IEEE, 2008.
52. Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.
53. Seung-Taek Park and Wei Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems*, pages 21–28, 2009.
54. Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. Mcnee, Joseph A. Konstan, and John Riedl. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 2002 International Conference on Intelligent User Interfaces*, pages 127–134. ACM Press, 2002.

55. Al Mamunur Rashid, George Karypis, and John Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. *SIGKDD Explorations*, 10(2):90–100, 2008.
56. David Rawlings and Vera Ciancarelli. Music preference and the five-factor model of the neo personality inventory. *Psychology of Music*, 25(2):120–132, 1997.
57. Peter J. Rentfrow, Lewis R. Goldberg, and Ran Zilca. Listening, watching, and reading: The structure and correlates of entertainment preferences. *Journal of Personality*, 79(2):223–258, 2011.
58. Peter J. Rentfrow, Samuel D. Gosling, et al. The do re mi’s of everyday life: The structure and personality correlates of music preferences. *Journal of Personality and Social Psychology*, 84(6):1236–1256, 2003.
59. Paul Resnick and Hal R Varian. Recommender systems. *Communications of ACM*, 40(3):56–58, 1997.
60. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer, 2011.
61. Alexandra Roshchina. *TWIN Personality-based Recommender System*. Institute of Technology Tallaght, Dublin, 2012.
62. Neil Rubens, Dain Kaplan, and Masashi Sugiyama. Active learning in recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul Kantor, editors, *Recommender Systems Handbook*, pages 735–767. Springer Verlag, 2011.
63. Neil Rubens and Masashi Sugiyama. Influence-based collaborative active learning. In *Proceedings of the 2007 ACM Conference on Recommender systems*, pages 145–148. ACM, 2007.
64. Bracha Shapira, Lior Rokach, and Shirley Freilikhman. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247, 2013.
65. Yue Shi, Martha Larson, and Alan Hanjalic. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. In *User Modeling, Adaption and Personalization*, pages 305–316. Springer, 2011.
66. Le Hoang Son. Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*, pages –, 2014.
67. Martin Szomszor, Harith Alani, Iván Cantador, Kieron O’Hara, and Nigel Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. In *Proceedings of the 7th International Semantic Web Conference*, pages 632–648, 2008.
68. Amit Tiroshi, Shlomo Berkovsky, Mohamed Ali Kâafar, Terence Chen, and Tsvi Kuflik. Cross social networks interests predictions based on graph features. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 319–322, 2013.
69. Marko Tkalčić, Matevz Kunaver, Andrej Košir, and Jurij Tasic. Addressing the new user problem with a personality based user similarity measure. In *Proceedings of the 1st International Workshop on Decision Making and Recommendation Acceptance Issues in Recommender Systems*, page 106, 2011.
70. Pinata Winoto and Tiffany Ya Tang. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? A study of cross-domain recommendations. *New Generation Computing*, 26(3):209–225, 2008.
71. Yuan Yao, Hanghang Tong, Guo Yan, Feng Xu, Xiang Zhang, Boleslaw K Szymanski, and Jian Lu. Dual-regularized one-class collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 759–768. ACM, 2014.
72. Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on the World Wide Web*, pages 22–32. ACM, 2005.