J Intell Inf Syst (2017) 48:365–397
DOI 10.1007/s10844-016-0418-3

Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem

Marharyta Aleksandrova[1,2] · **Armelle Brun**[1] · **Anne Boyer**[1] · **Oleg Chertov**[2]

Received: 28 January 2016 / Revised: 1 June 2016 / Accepted: 7 June 2016 / Published online: 23 June 2016

Abstract Matrix factorization has proven to be one of the most accurate recommendation approaches. However, it faces one major shortcoming: the latent features that result from the factorization are not directly interpretable. Providing interpretation for these features is important not only to help explain the recommendations presented to users, but also to understand the underlying relations between the users and the items. This paper consists of 2 contributions. **First**, we propose to automatically interpret features as users, referred to as *representative users*. This interpretation relies on the study of the matrices that result from the factorization and on their link with the original rating matrix. Such an interpretation is not only performed automatically, as it does not require any human expertise, but it also helps to explain the recommendations. The **second** proposition of this paper is to exploit this interpretation to alleviate the content-less new item cold-start problem. The experiments conducted on several benchmark datasets confirm that the features discovered by a Non-Negative Matrix Factorization can be interpreted as users and that *representative users* are a reliable source of information that allows to accurately estimate ratings on new items. They are thus a promising way to solve the new item cold-start problem.

✉ Marharyta Aleksandrova
marharyta.aleksandrova@loria.fr; rita.v.aleksandrova@gmail.com

Armelle Brun
armelle.brun@loria.fr

Anne Boyer
anne.boyer@loria.fr

Oleg Chertov
chertov@i.ua

1 Université de Lorraine - LORIA, Campus Scientifique, 54506 Vandøeuvre lès Nancy, France

2 National Technical University of Ukraine “Kyiv Polytechnic Institute”, Kyiv, Ukraine

Keywords Recommender systems · Matrix factorization · Features interpretation · Cold-start problem

1 Introduction

Recommender systems (RS) aim at assisting users in their selection or purchase of items, by suggesting them items that fit their needs. RS have increasingly democratized these last ten years, and are now a classical tool proposed by many online services in diverse domains such as e-commerce (Huang 2011), tourism (Zanker et al. 2008), e-learning (Verbert et al. 2012; Klašnja-Milićević et al. 2015), etc.

Collaborative Filtering (CF) (Schafer et al. 2007) is a recommendation technique, which relies on users' preferences: generally the ratings they assign to items (represented as a rating matrix, see left part of the Fig. 1). CF estimates the ratings that a user, called an *active user*, would assign to items he/she has not rated yet. It assumes that users who agreed in the past will agree in the future too. CF ignores user and item attributes (demographics, interests on domains, items descriptions, etc.), contrary to content-based techniques (Lops et al. 2011). There are two major approaches in CF: neighborhood-based (NB) and matrix factorization (MF) (Koren 2008) (see right part of the Fig. 1).

The NB approach (Desrosiers and Karypis 2011) identifies for each active user, his/her similar-minded users (neighbors), using the rating matrix. It estimates the missing ratings of this active user by exploiting the ratings of his/her neighbors. NB is quite popular due to its simplicity, efficiency, accuracy and its ability to explain the recommendations provided (through the use of users' neighbors) (Koren 2008). However, NB has limitations on large and/or sparse datasets and it is time consuming.

The MF approach (Koren et al. 2009) relies on the idea that the ratings in the rating matrix can be explained by a small number of latent features (also referred to as factors). It factorizes the rating matrix into two low-rank matrices, which represent the relation between the users and items with the latent features. The MF approach has recently attracted more attention than the traditional neighborhood-based approaches (Adomavicius and Tuzhilin 2005), as it is adequate for large-scale and sparse datasets (Takacs et al. 2009) and it has proven to form highly accurate models of low-complexity, see Netflix Prize competition (Koren et al. 2009).

The features of MF are formed in such a way, that the resulting model fits the best the known ratings. Resulting features have no underlying physical meaning and the interpretation of these features is not an easy and obvious task. Thus, one shortcoming of MF is the

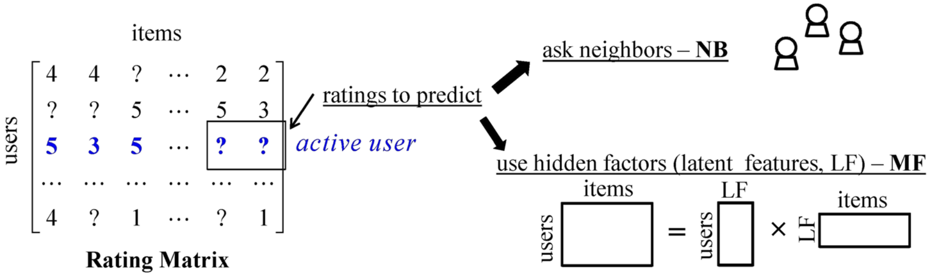

Fig. 1 Illustration of collaborative filtering

difficulty to explain the recommendations provided (as elements of the model have no real interpretation), contrary to NB approaches (Hu et al. 2008). However, providing an explanation or justification of the recommendations made to users is an important characteristics of recommender systems (Ortega et al. 2014). Several studies (Herlocker et al. 2000; Sinha and Swearingen 2002) show that explanations enhance the user satisfaction and increase user trust (fidelity) in the system. Users feel more comfortable when they understand why a certain item is recommended to them. Therefore, providing an interpretation of latent features in MF and thus making the resulting recommendations explainable can add this important characteristic to MF-based models.

In this work we present two contributions. **First**, we propose a new way to interpret features in MF. Its originality lies in the nature of the interpretation. We propose to interpret each feature as a user from the population of recommender system users, without modifying the original MF model. In addition, the proposed interpretation is performed automatically and does not require any human expertise. As the latent features in MF represent the relations between users and items (Koren et al. 2009), in this work we consider that feature-related users will represent the preferences of other users of the system. Thus, these feature-related users will be referred to as *representative users* (*RUs*, abbreviation *RU* will be used to refer to one representative user). From our point of view, this interpretation has important positive impacts on the way the model can be exploited. First, if such interpretation can be made, the recommendations presented to users can be easily explained. Indeed, the missing ratings will be estimated not through the set of features that have no physical interpretation, but through a set of users, who correspond to these features (representative users). This makes the MF model similar to neighborhood-based approaches, where the ratings are also computed through users, more precisely, through neighbors. Second, if the assumption that representative users are capable to represent the preferences of other users in the system holds, then they can be used to solve the content-less new item cold-start problem, by being asked to initially provide ratings on new items. The approach for solving the new item cold-start problem forms the **second** contribution of this paper.

This paper extends our previous research on feature interpretation (Aleksandrova et al. 2014) and the related solution to the cold-start problem (Brun et al. 2014).

The rest of the paper is organized as follows. Section 2 presents related works on relevant topics. Section 3 is dedicated to the description of our approaches to interpret latent features as users and to solve the new item cold-start problem. Section 4 describes the datasets and the evaluation procedure, while in Section 5 experimental results are presented. Finally, Section 6 is dedicated to the conclusions.

2 Related works

2.1 Recommendations via matrix factorization

In recommender systems, the matrix factorization approach assumes that a small number of latent features influence the ratings of users on items (Sarwar et al. 2000; Koren et al. 2009). MF aims at forming two low rank matrices, which represent the extent to which users and items are related to these latent features. When multiplying these two matrices the complete original rating matrix is reconstructed, thus allowing to estimate the missing ratings.

We start with the introduction of the notations used in this paper. Let U be the set of users and I be the set of items, of size N and M respectively. Let R, $dim(R) = N \times M$, be the rating matrix and, r_{lj} be the rating value of user $u_l \in U$, with $1 \leq l \leq N$, on item $i_j \in I$, with $1 \leq j \leq M$. Let G be a matrix. We denote by $\mathbf{g_{(l,*)}}$ the l^{th} row-vector of a matrix G and by $\mathbf{g_{(*,l)}}$ – the l^{th} column-vector of G. By $\vec{e}$ we denote some general vector, that is not necessarily associated with a matrix.

Let W and V be two matrices resulting from MF of dimensionality $K \times N$ and $K \times M$ respectively, where K is the number of latent features. Let us denote by F the corresponding set of features. The product of both matrices approximates the original rating matrix: $R \approx W^T V$. Therefore, the task of forming factor matrices W and V is usually reduced to solving an optimization problem in (1).

$$\min \left|\left|R - W^T V\right|\right|_F^2, \tag{1}$$

where operation $||*||_F^2$ denotes the Frobenius norm. In order to avoid overfitting, a regularization parameter λ can be added (Ma et al. 2011; Zhou et al. 2008). In this case the optimization problem will look as follows:

$$\min \left(\left|\left|R - W^T V\right|\right|_F^2 + \lambda \left(||W||_F^2 + ||V||_F^2\right)\right). \tag{2}$$

Different optimization procedures can be used for solving optimization problems in (1) or (2), among them Singular Value Decomposition (SVD) (Sarwar et al. 2000), Alternating Least Squares (ALS) (Zhou et al. 2008), Stochastic Gradient Descend (SGD) (Koren et al. 2009), Multiplicative Update Rules (Lee and Seung 2001). The usage of SVD can be restricted due to the fact that it requires all entries of the factorized matrix to be known, though it is not the case in reality. Contrary to SVD, ALS and SGD ignore missing ratings. Optimization procedures based on Multiplicative Update Rules are used to obtain non-negative factor matrices (Non-Negative Matrix Factorization, NMF), that is factor matrices with all entries being non-negative. In this paper we use Non-Negative Matrix Factorization with Multiplicative Update Rules as the optimization procedure. It was chosen due to its ability to represent an entity as an additive linear combination of canonical coordinates and following works (Zhang et al. 2006; Pessiot et al. 2006), also focused on the task of latent features interpretation.

The values in the matrix V represent the relation between the items and the latent features. The vector $\mathbf{v_{(k,*)}}$ $(1 \leq k \leq K)$ is the M-dimensional representation of feature f_k and $\mathbf{v_{(*,j)}}$ $(1 \leq j \leq M)$ the K-dimensional representation of the item i_j. The values in W represent the relation between the users and the latent features. Similarly, the vector $\mathbf{w_{(*,l)}}$ $(1 \leq l \leq N)$ is the K-dimensional representation of user u_l. Figure 2 shows these notations on the corresponding matrices. To calculate the unknown rating $r_{(l,i)}$, (3) is used, with vector inner product operation denoted by $(*, *)$.

$$r_{l,i} = (\mathbf{w_{(*,l)}}, \mathbf{v_{(*,j)}}) \tag{3}$$

Several MF-based models have been studied in recommender systems. Adding the aspect of uncertainty, the Probabilistic Matrix Factorization (PMF) models the predictive error of traditional MF as a Gaussian distribution (Mnih and Salakhutdinov 2007; Salakhutdinov and Mnih 2008). PMF was proven to be efficient and accurate on Netflix dataset (Mnih and Salakhutdinov 2007). When the original rating matrix is more than two-dimensional (for example, the third dimension can correspond to the content information) the generalization of matrix factorization, called Tensor Factorization, is used (Karatzoglou et al. 2010). In some situations it can be useful to perform simultaneous factorization of multiple matrices, for example for transferring knowledge from different domains (Huang et al. 2012). For such cases Collective Matrix Factorization (Singh and Gordon 2008) was proposed. If

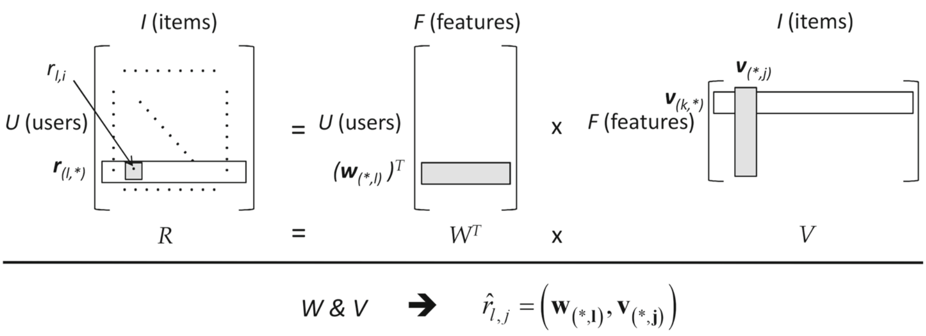

Fig. 2 Matrices and notations for MF

we want to obtain different latent spaces for users and items (note that in traditional MF the same set of features is shared by both users and items), Matrix Tri-Factorization should be used. This model decomposes the original rating matrix on the product of three matrices (Yoo and Choi 2009). Two of them correspond to user- and item-related features, and the additional third matrix represents the relationship between these two sets of features. Finally, both MF and NB can be united in the same framework. For example, in (Koren 2008) both models are merged, resulting in the increase of the recommendation accuracy.

2.2 Features interpretation

The values in both matrices that result from factorization are those that optimally describe the ratings in the original rating matrix: they are designed to be those that minimize the loss function in (1) or (2). As a consequence, the features are not directly interpretable. So they are generally only used to predict ratings. However, interpreting these features could be an important added value as it could not only help to explain the recommendations presented to users, but it could also help to understand the underlying relation between users and items.

Some authors claim that features of MF have a hidden meaning, however they do not provide a direct procedure to discover it. For example, in (Koren et al. 2009), where a movie dataset is used, the authors claim that features can represent obvious dimensions such as comedy/drama, amount of action, orientation to children, etc. They also mention that features may represent less well defined dimensions such as quirkiness, or represent completely uninterpretable dimensions. The same idea was used by Willemsen et al. (2011) to form diverse recommendation lists, which is an important task in recommender systems (as it was shown by Smyth and McClave (2001) and Ziegler et al. (2005), the quality of recommendations depends not only on the accuracy of the predicted ratings, but also on the ability of the system to recommend diverse items).

An alternative way to interpret MF is to align it with other interpretable models. Graus (2011) investigates the relationship between the latent features of MF and the attribute dimensions in Multidimensional Scaling. Though it was shown that the translation from latent features to attribute dimensions can be found, it turned out that these two entities are not very similar. McAuley and Leskovec (2013) propose to align features of the matrix factorization models with review topics learned through Latent Dirichlet Allocation (LDA), by introducing functional dependence between features and topics and simultaneously learning both models. This idea was further extended by a number of authors. For example, contrary to the baseline (McAuley and Leskovec 2013), Rossetti (2014) merges all the reviews into a single document; Hu et al. (2015) and Zhao et al. (2015) add social-based information; Xin et al. (2015) uses heterogeneous topics instead of homogenous. Using close ideas Zhang et al. (2014) extracts features from the reviews through the Phrase-level Sentiment Analysis, and then incorporates them into an MF-based framework.

Another group of researches attempted to directly interpret MF features (without using additional techniques). In most cases it was done through the Non-Negative Matrix Factorization, due to its ability to represent an entity as an additive linear combination of canonical coordinates. NMF was introduced as a method that allows learning parts of objects, for example discovering parts of faces on an image (Lee and Seung 1999). Later this property of NMF was used in many domains: speech processing (Behnke 2003), text mining (Chagoyen et al. 2006), computational biology (Devarajan 2008), etc. In the field of recommender systems NMF was used to obtain interpretable features of the factorization model. For example, Zhang et al. (2006) considered features as groups of interests, and Pessiot et al. (2006) interpreted them as imaginary users, who represent a certain behavioral type. However, in both approaches the interpretation of each feature is not so easy to perform as it has to be discovered manually by exploiting human expertise and by analyzing the content of the matrices: what attribute can be related to a given feature, who belongs to the group a given feature represents, etc.? Finally, the Representative-Based Matrix Factorization approach (RBMF) Liu et al. (2011) propose to associate latent features with real users or items by filling one of the factor matrices with ratings of chosen users/items and obtaining the other one through an optimization procedure.

2.3 Cold-start problem in collaborative filtering

As collaborative filtering (whether NB or MF) relies completely on the ratings, a problem occurs when there are either no known ratings for a specific item/user, or the number of known ratings for a specific item/user is very small. In this case it is impossible to make reliable recommendations (Bobadilla et al. 2013). The first problem (absence of ratings) is known as a cold-start or out-of-matrix prediction, while the second (very small number of known ratings) – as warm-start (in-matrix prediction) (Agarwal and Chen 2010; Lam et al. 2008). State of the art (Bobadilla et al. 2013; Park and Chu 2009) distinguishes three kinds of cold-start problems: new community, new item and new user. The new community problem refers to the start up of a new recommender system. New item and new user problems correspond to the cases when a new item/user enters an already existing system. In this paper we focus on the new item cold-start problem, however, we show that our method can be used for the new user cold-start as well.

A very popular solution to the new item cold-start problem relies on the content of the items. The recommender either switches to a content-based techniques or mixes content with CF. Recommendations are provided by comparing the properties or content of the items to the content of those items that are known to be of interest for an active user (Melville et al. 2002; Lam et al. 2008). The main limit of such a solution is the availability of the content, which depends on the type of items. Indeed, in some domains it is hard to automatically analyze the underlying content. In addition, users' interests cannot always be characterized by content properties contained in an item, for example when perfumes (Das et al. 2007) or Facebook pages (Xie et al. 2013) are considered.

In the absence of content a content-less new item cold-start problem is faced, which is the focus of this paper. A solution adopted to solve the content-less new item cold-start problem is to form a set of users, who will be asked to rate each new item in the system. The obtained ratings are used to estimate the preferences of other users on these new items. These users should represent the interests of the whole population as fully as possible and/or be capable to influence the preferences of others. Such set of users is referred to as seed users or seeds (Liu et al. 2011), representative users (Liu et al. 2011), influential users (Rashid 2007), power users (Seminario and Wilson 2014) or leaders (Esslimani et al. 2013).

Seed users can be chosen randomly, but there is no guarantee that their ratings will represent correctly or influence preferences of other users. They can also be chosen within a set of experts (Amatriain et al. 2009), which guarantees the quality of their ratings, but this solution is expensive and in some cases experts may not be available.

The problem of the automatic identification of seed users can be considered as a task of active learning, where the system automatically identifies those input elements, which will result in a better model construction (Houlsby et al. 2014). Some approaches in this direction have been proposed. For example, in the frame of neighborhood-based models, Esslimani et al. (2013) proposes to discover seed users based on their connectivity and average similarity. Rashid (2007) proposes to evaluate the importance of users as the negative influence rendered on the quality of recommendations, when these users are removed from the system. In (Houlsby et al. 2014) Bayesian Active Learning approach is used. The main idea here is to select those elements, that minimize uncertainty over the parameters of the model. Liu et al. (2011) identify seed users through the matrix maximal-volume concept. This concept identifies those columns in the matrix that are large in magnitude and are linearly independent. The set of users associated with the chosen columns form the set of seeds.

2.4 Cold-start problem in matrix factorization

Some approaches for the cold-start problem solution have been proposed in the frame of MF. Following the general tendency in CF, they can be divided into two groups: 1) those that require additional external information (like content, social structure or both), and 2) those that use seed users/items (for content-less problem).

Among the first group of approaches we can mention the following. Gantner et al. (2010) propose to solve the new item cold-start problem by learning a mapping function between latent features and item attributes (from the learning set). Given a new item and its corresponding attributes, the latent feature vector for the new item is computed. Then the rating matrix is filled for this new item. Saveski (2013) proposes a joint factorization of the rating matrix and the content matrix. Enrich et al. (2013) introduce the MF-based model, which transfers the knowledge from one domain to another through the shared tags. The structure of the social network is used as a source of additional information by Jamali and Ester (2011), where the feature vector of a user is constructed as being dependent on the feature vectors of his/her neighbors. Salakhutdinov and Mnih (2008) incorporates both social and content information into Probabilistic Matrix Factorization model (through the parameters of the model).

Moving to the second group of approaches, Zhou et al. (2011) and Sun et al. (2013) propose a solution to the new user problem (which is symmetric to the new item problem), where the feature vectors of MF are learned through the new user answers in the initial interview process. The seed items (those used in the interview) in this case are chosen through a decision tree construction. In (Liu et al. 2011) seed users/items are chosen as those who can be associated with the features through the matrix maximal-volume concept (see previous subsection).

2.5 Baseline approach for identification of representative users (Representative-based Matrix Factorization, RBMF)

Among all state of the art approaches, the Representative-based Matrix Factorization (RBMF), which was presented at the RecSys 2011 conference (Liu et al. 2011), is the closest to ours. It is the only approach proposed within the MF framework, that does not require any content information for alleviating the new item cold-start problem. It is based on the idea of searching a set of users, which can represent the interests of the rest of the users and asking their opinion on new items (seed users). That is why we choose RBMF approach as a baseline for comparison with the approach presented in this paper.

Within RBMF 2 models are proposed (Liu et al. 2011): user-RBMF and item-RBMF for solving the new item and the new user cold-start problems respectively. In our work we consider the first one, as this paper is dedicated to a solution of the new item cold-start problem (the item-RBMF is symmetrical).

User-RBMF represents the rating matrix R in a form $R \approx XA$, where the matrix A is formed of ratings of K users that were chosen as seed users and the matrix X is formed as the one that solves the optimization problem (4). We can see that the optimization problem of RBMF (4) is almost the same as the optimization problem for the standard regularized MF (see (2)) with the difference that one of the factor matrices is fixed and the entries of the other one are calculated through an optimization prodecure.

$$\min \left(||R - XA||_F^2 + \lambda\, ||X||_F^2 \right). \tag{4}$$

The seed users are found through the matrix maximum volume concept (see (Liu et al. 2011) and (Goreinov et al. 2010)). First, the rank-k SVD decomposition of the rating matrix R is performed, and after that the $maxvol$ algorithm (Goreinov et al. 2010) is used to find a $k \times k$ maximal-volume submatrix of the first matrix in the SVD decomposition. The users, that correspond to the rows of the chosen submatrix are considered as seed users.

For solving the new item cold-start problem, the matrix A is filled with ratings of seed users on new items and the missing values are predicted through the multiplication of matrices X and A.

3 The proposed approach

This section is dedicated to the description of our approach. First, we explain how the association between latent features resulting from matrix factorization and real users is done, that is, how we identify representative users (RUs). Second, we show how the notion of RUs can be used to explain the recommendations made with MF. Third, we present our approach to solve the new-item cold-start problem in the frame of MF through the notion of seed users (that can be instantiated with a set of RUs). Finally we conclude with the discussion of our approach and the state of the art methods.

3.1 Identification of RUs

We focus here on the way we propose to identify representative users (RUs) in the frame of MF. First, let us consider an example. Let L_1 and L_2 be 2 linear spaces of dimensionality respectively 6 and 3, with basic vectors in canonical form $\{\vec{w}_n\}$, $n \in \overline{1,6}$ and $\{\vec{f}_k\}$, $k \in \overline{1,3}$. Let the transfer matrix from L_1 to L_2 be specified by matrix P (5).

$$P = \begin{pmatrix} 0 & 0 & p_{13} & p_{14} & 1 & p_{16} \\ 1 & 0 & p_{23} & p_{24} & 0 & p_{26} \\ 0 & 1 & p_{33} & p_{34} & 0 & p_{36} \end{pmatrix} \tag{5}$$

We can say that $\boldsymbol{w}_5$, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are direct pre-images of $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$ respectively. Indeed, $P\boldsymbol{w}_5 = P\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T = \boldsymbol{f}_1$. By analogy, $P\boldsymbol{w}_1 = \boldsymbol{f}_2$, $P\boldsymbol{w}_2 = \boldsymbol{f}_3$. At the same time vectors $\boldsymbol{w}_3$, $\boldsymbol{w}_4$ and $\boldsymbol{w}_6$ will be mapped into linear combinations of basic vectors $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$. For example, $P\boldsymbol{w}_3 = p_{13}\boldsymbol{f}_1 + p_{23}\boldsymbol{f}_2 + p_{33}\boldsymbol{f}_3$ corresponds to the linear combination for $\boldsymbol{w}_3$.

The matrix W, resulting from the factorization of R, can be considered as a transfer matrix from the space of users to the space of features (Koren et al. 2009). So, analyzing the example considered above, we can say that if matrix W has a form similar to P (in equation (1)), *i.e.* W has exactly K unitary columns with one non-0 and equal to 1 element on different positions, then the users corresponding to these columns are direct pre-images of the K features. Following Guermeur et al. (2004), we say they represent the canonical coding of the features. As a consequence, we consider that the features can thus be directly interpreted as users (representative users). Though this idea is simple and we find it promising, according to our knowledge it was not exploited previously.

Obviously, in the general case, one cannot guarantee that the matrix W will be in a form similar to matrix P. Worse, none of the column-vectors of matrix W may directly represent the canonical form of a feature. However, we can consider as RUs those users, whose vectors in W are the closest to the required canonical form. The procedure, which we propose for the identification of representative users is described in details below.

Step 1: normalize matrix *W* Once the matrices W and V are obtained, the normalization of each of the N column vectors of the matrix W is performed, that results in unitary columns. The resulting normalized matrix will be denoted by W^{norm}. The normalization is performed in order to obtain the matrix W in the form closest to P. After such a transformation, the new matrix W^{norm} still represents the same relations between users and features, but with certain scaling coefficients. Next, the column-vectors of the W^{norm} matrix are analyzed with the aim of identification of the best candidates for representing latent features (those that are close to the canonical form).

Step 2: form groups of pre-image candidates In this step the groups of pre-image candidates are formed.

A user u_l is considered to be the best pre-image candidate for a feature f_k if the vector in matrix W^{norm} that corresponds to u_l (column-vector $\mathbf{w}^{norm}_{(*,l)}$) is the closest to the corresponding canonical vector (a vector with the only one non-0 and equal to 1 value on the position k, denoted by $\boldsymbol{c}_k$). The notion of closeness between vectors is expressed through the Euclidean distance. That is the task of identification of the representative user u_l is reduced to solving the optimization problem and his/her position (l) is defined by (6).

$$l = \underset{l' \in U}{\arg\min} \left[dist(\boldsymbol{c}_k, \mathbf{w}^{norm}_{(*,l')}) \right] \tag{6}$$

Let us consider the following example. Assume that we have a vector $\boldsymbol{\alpha}$ of the form $\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_K \end{pmatrix}^T$ with unique norm $\left(\sqrt{\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_K^2} = 1\right)$. Then the distance between $\boldsymbol{\alpha}$ and the first canonical vector $\boldsymbol{c}_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}^T$ is expressed by $dist^2(\boldsymbol{c}_1, \boldsymbol{\alpha}) = (1-\alpha_1)^2 + \alpha_2^2 + \ldots + \alpha_K^2$. Performing simple mathematical transformations results in (7).

$$dist^2(\boldsymbol{c}_1, \boldsymbol{\alpha}) = 2(1 - \alpha_1) \tag{7}$$

This means that the minimum of the distance is obtained under the condition $\alpha_1 \rightarrow$ max. Taking into account this reasoning, we consider a user u_l as a pre-image candidate for the feature f_k if the maximum value of the appropriate vector $\mathbf{w}_{(*,l)}^{norm}$ is situated on the position k.

Therefore, all users are divided into groups of pre-image candidates according to the position of the maximal value in the associated column vectors from the matrix W^{norm}. The corresponding formal procedure is presented in Algorithm 1. The algorithm has one input parameter – the matrix W^{norm}. Based on the dimensions of the input matrix W^{norm}, the number of groups K and the total number of users N is defined on the first step of the algorithm. Next we initialize K groups of pre-image candidates $GC\_1, \ldots, GC\_K$ as empty sets. Finally, making a loop through all N users of the system a user u_l is put in the group which is associated with the position of the maximal value in the corresponding vector $\mathbf{w}_{(*,l)}^{norm}$.

Algorithm 1 Form Groups Of Pre-image Candidates

procedure FormGroupsOfPrCand(W^{norm})
    $[K, N] = size\,(W^{norm})$
    for $k = 1 : K$ **do**
        $GC\_k = \{\}$
    end for
    for $l = 1 : N$ **do**
        $k = pos(max(\mathbf{w}_{(*,l)}^{norm}))$
        $GC\_k = GC\_k \bigcup \{u_l\}$
    end for
    return $GC\_1, \ldots, GC\_K$
end procedure

Step 3: identify RUs Once all users are divided into subgroups of pre-image candidates for each feature, we can identify RUs using Algorithm 2. In each group of pre-image candidates GC_k, the representative user is defined as a user $u_{l''}$ whose vector $\mathbf{w}_{(*,l'')}$ is the closest to the canonical vector $\boldsymbol{c}_k$. If for some reasons the chosen representative user can not be used for solving the assigned task, the next best candidate within the current pre-image candidates group can be considered as RU. Additionally, if a certain group of pre-image candidates, say GC_k is empty, that is on the step 2 there were no columns in the matrix W^{norm} with the maximum value being situated on the position k, the user from another group with the smallest value of distance to the canonical vector $\boldsymbol{c}_k$ can be chosen as RU for the feature k. In this way we ensure that all features will be associated with RUs.

Algorithm 2 Find RUs

procedure FindRUs($GC\_1, \ldots, GC\_K$)
    for $k = 1 : K$ **do**
        $l'' = \underset{u_{l'} \in GC\_k}{\arg\min} \left[dist(\mathbf{c}_k, \mathbf{w}_{(*,l')}^{norm}) \right]$
        $RU\_k = u_{l''}$
    end for
    return $RU\_1, \ldots, RU\_K$
end procedure

Note that the original MF model remains unchanged. The normalization of the matrix W in our approach is performed only for the identification of representative users. However, when computing recommendations, the original W and V matrices are used.

The presented procedure of RUs Identification (see Algorithm 3) results in a set of users (RUs) that are associated by bijective mapping with the latent features of MF. As latent features are considered to represent the relations among users and items, the obtained feature-related users should be capable to represent the same interconnections. It means that the set of these users should correctly reflect the interests of the whole population of users. Therefore, representative users can be used as a set of seed users to solve the new item cold-start problem. Indeed, asking their opinion on new items we can infer the opinion of the entire population.

Algorithm 3 RUs Identification

```
procedure RUsIdentification(W)
    W → W^norm
    [GC_1, ..., GC_K] = FormGroupsOfPrCand(W^norm)
    [RU_1, ..., RU_K] = FindRUs(GC_1, ..., GC_K)
    return RU_1, ..., RU_K
end procedure
```

3.2 Interpretation of recommendations

3.2.1 Theoretical statements

The way we propose to interpret features also has the advantage to explain the recommendations provided by MF. This explanation is done completely automatically and, contrary to many state of the art approaches, does not require neither human experience or interaction, nor external sources of information (like item reviews). Let us rewrite (3) in a form of (8).

$$r_{l,i} = \sum_{k=1}^{k=K} w_{k,l} v_{k,j} \tag{8}$$

As it was discussed in Section 2.1 the vector $\mathbf{w}_{(*,\mathbf{l})}$ is a K-dimensional representation of a user u_l (that is representation of a user u_l in the space of latent features), and the vector $\mathbf{v}_{(*,\mathbf{j})}$ is a K-dimensional representation of the item i_j (that is representation of an item i_j in the same space of latent features), see Fig. 2. Thus, if the set of features is interpreted as a set of representative users, then both vectors $\mathbf{w}_{(*,\mathbf{l})}$ and $\mathbf{v}_{(*,\mathbf{j})}$ represent user u_l and item i_j in a space of representative users. Therefore, value $v_{k,j}$ expresses preferences of a representative user k on the item j and $w_{k,l}$ – closeness of the user u_l to the representative user k. This makes the rating estimation process of MF being close to the one of NB. Indeed, the rating estimation process in NB is done using (9).

$$\hat{r}_{l,j} = \sum_{k'=1}^{k'=K'} sim(u_l, u_{n_{k'}}) r_{n_{k'},j}, \tag{9}$$

where K' – is the number of neighbors, $sim(u_l, u_{n_{k'}})$ – the similarity between an active user u_l and his/her k'-th neighbor $u_{n_{k'}}$, and $r_{n_{k'},j}$ – is the rating assigned by the neighbor $u_{n_{k'}}$ to the item j. The notion of similarity in NB is usually expressed through the correlation (Desrosiers and Karypis 2011).

Let us compare (8) and (9) with $w_{k,l}$ corresponding to $sim(u_l, u_{n_{k'}})$ (closeness of the user u_l to the k-th representative user or similarity between user u_l and his neighbor $u_{n_{k'}}$) and $v_{k,j}$ corresponding to $r_{n_{k'},j}$ (preferences of the k-th representative user on the item j or rating value of the neighbor $u_{n_{k'}}$ on this item). Note, however, that though the relation between latent features and representative users is bijective, it is not identical (recall from the previous section that representative users correspond to latent features with certain scaling coefficients). Thus vectors $v_{k,j}$ and $w_{k,l}$ may not directly correspond to ratings and similarity values, but reflect these dependencies in an indirect way. We will show the correctness of this statement through a toy example in the next subsubsection.

3.2.2 Toy example

In order to show that values $v_{k,j}$ and $w_{k,l}$ resulting from MF can be interpreted as ratings of representative users and as similarity values between representative users and the rest of the users from the data set, we provide here an analysis on a small toy example. In this way we show how recommendations can be interpreted. Let us consider a rating matrix given in Table 1, which represents the ratings of 12 users on 12 movies (items). Each movie is annotated with genre (with possible values *comedy* or *drama*) and the release decade (with possible values *70*, *80* or *90*). Analyzing the ratings provided in the table we can make some conclusions concerning the preferences of each user. For example, the first user likes comedy films and dislikes dramas, the second one has inverse preferences. The fifth user prefers the films released in 80s regardless of the genre and the sixth user likes comedies released in 70s, has middle preferences towards comedies released in 80s and dislikes comedies from 90s as well as drama films. The short description of the preferences for each user is provided in the last column of the Table 1.

After performing a non-negative matrix factorization of the rating matrix in Table 1 with number of features $K = 5$, we can identify representative users, following the approach presented in Section 3.1. The following users were identified as representative ones: u_3, u_8, u_1, u_7 and u_9 corresponding to features f_1, f_2, f_3, f_4 and f_5 respectively (see the first column of Table 1). Using the values in the matrix W and the provided association of latent features with the representative users, the preferences of the rest of the users can be decomposed into the linear combination of the interests of the representative users. For

Table 1 Model example of a rating matrix for interpretation

item characteristics													description
genre	comedy			comedy			drama			drama			
decade	70	80	90	70	80	90	70	80	90	70	80	90	
IDs/F	1	2	3	4	5	6	7	8	9	10	11	12	
$1/f_3$	*5*	*5*	*5*	*5*	*5*	*5*	*1*	*1*	*1*	*1*	*1*	*1*	comedy
2	*1*	*1*	*1*	*1*	*1*	*1*	*5*	*5*	*5*	*5*	*5*	*5*	drama
$3/f_1$	*5*	*3*	*1*	*5*	*3*	*1*	*5*	*3*	*1*	*5*	*3*	*1*	70_1 or $80_{0.5}$
4	*1*	*3*	*5*	*1*	*3*	*5*	*1*	*3*	*5*	*1*	*3*	*5*	$80_{0.5}$ or 90_1
5	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	80
6	*5*	*3*	*1*	*5*	*2*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	comedy & $(70_1 \text{or} 80_{0.5})$
$7/f_4$	*2*	*5*	*2*	*1*	*5*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	comedy&80
$8/f_2$	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*5*	*1*	*1*	*5*	drama & 90
$9/f_5$	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	drama & 80
10	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*4*	*1*	*1*	*5*	*1*	drama & 80
11	*5*	*5*	*1*	*5*	*5*	*1*	*5*	*5*	*1*	*5*	*5*	*1*	70 or 80
12	*1*	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	*1*	*1*	*5*	90

example, the linear combination for the second user is given in (10). The coefficients of the second part of the equation correspond to those in the matrix W^{norm}, that is the second linear combination corresponds to a vector with a unique norm.

$$\begin{aligned} u_2 &= \mathbf{1.4411}u_3 + \mathbf{1.5208}u_8 + 0u_1 + 0.0273u_7 + \mathbf{1.5027}u_9 \\ &= 2.5785\,(\mathbf{0.5589}u_3 + \mathbf{0.5898}u_8 + 0u_1 + 0.0106u_7 + \mathbf{0.5828}u_9) \end{aligned} \tag{10}$$

Recall that the second user likes drama regardless of the release decade. From the equality provided above we can see that representative users u_8, u_9 and u_3 have the main impact on the preferences of the considered user u_2. The user u_8 likes drama films released in 90s, user u_9 - dramas from 80s and user u_3 adds to this linear combination preferences of the films released in 70s, as it is his/her major interest. User u_7, who likes the comedies released in 80s has a small impact in the linear combination. At the same time, the linear combination coefficient for the user with opposite interests (user u_1, who prefers the comedy films) is equal to 0.

Let us consider one more example: the linear combination for user u_{10}, provided in (11). The major coefficient in the linear combination corresponds to user u_9, who has exactly the same preferences as the analyzed user u_{10} (both of them prefer drama films released in 80s). The rest of the coefficients in the linear combination (11) are relatively minor.

$$\begin{aligned} u_{10} &= 0u_3 + 0.3594u_8 + 0.0284u_1 + 0.4621u_7 + \mathbf{1.6522}u_9 \\ &= 1.7530\,(0u_3 + 0.2050u_8 + 0.0162u_1 + 0.2636u_7 + \mathbf{0.9425}u_9) \end{aligned} \tag{11}$$

In this way, we have shown that the values $w_{k,l}$ can be interpreted as similarity values between representative users and the rest of the users from the data set. Now we proceed to show that the values $v_{k,j}$ represent preferences of representative users on items.

In order to show that vectors $\boldsymbol{v_{(k,*)}}$, resulting from MF, can be interpreted as ratings of the representative users, we calculated the value of the Pearson correlation between the ratings of representative users and the lines of matrix V corresponding to the associated features (see Table 2). In the provided table the first row lists the representative users, the second row lists the rows of matrix V corresponding to the features associated with representative users, and the third row provides correlation values between ratings of representative users and corresponding rows in V. In the fourth row the mean correlation between a certain row in matrix V and all 12 users is given and the last row presents the mean of the absolute correlation between a certain row in matrix V and all 12 users.

From Table 2 we can see that ratings of representative users are highly correlated with corresponding rows from the matrix V (compared with the mean correlation). This shows that the vectors $\boldsymbol{v_{(k,*)}}$ can be interpreted are ratings of representative users on items. Note that this example has shown that features of the MF approach can be associated with representative users when NMF optimization algorithm is used, that is the algorithm that results

Table 2 Correlation of rows from matrix V with ratings of users

RUs	u_3	u_8	u_1	u_7	u_9
$\mathbf{v}_{(\mathbf{k},*)}$	$\mathbf{v}_{(\mathbf{1},*)}$	$\mathbf{v}_{(\mathbf{2},*)}$	$\mathbf{v}_{(\mathbf{3},*)}$	$\mathbf{v}_{(\mathbf{4},*)}$	$\mathbf{v}_{(\mathbf{5},*)}$
$corr\left(\mathbf{v}_{(*,\mathbf{k})}, \mathbf{r}_{(*,\mathbf{k})}\right)$	0.9187	0.8620	0.9894	0.9325	0.9790
$\underset{1\le l\le N}{\text{mean}}\left(corr\left(\mathbf{v}_{(*,\mathbf{k})}, \mathbf{r}_{(*,\mathbf{l})}\right)\right)$	-0.0521	-0.0657	-0.0023	0.1476	0.1411
$\underset{1\le l\le N}{\text{mean}}\left(abs\left(corr\left(\mathbf{v}_{(*,\mathbf{k})}, \mathbf{r}_{(*,\mathbf{l})}\right)\right)\right)$	0.4192	0.5554	0.4247	0.3825	0.4526

in non-negative factor matrices V and W. The validity of this statement should be tested for other optimization procedures, like ALS or SGD.

3.3 Seed users for alleviating cold-start problem in the frame of MF-based models

Here we present our approach to solve the cold-start problem by exploiting the ratings provided by the seed users (either the set of RUs or any other suitable set of users according to a certain criteria). As it was shown in the literature review section, the idea of asking some predefined users (seed users) to provide their rating on new items and then using these ratings to solve the cold-start problem is widely exploited. The novelty of our approach relies in the way these ratings are used in the frame of MF models and the way the seed users are chosen (as representative users).

Let I_{new} be the set of new items and S be the set of indexes of users, who are considered as seeds. The way the new item cold-start problem can be solved is explained below and is schematically presented in Fig. 3. The information in red represents additional data ratings provided by seed users and the information in grey areas represents values computed automatically.

By asking seed users u_{s_1}, u_{s_2}, ... ($s_k \in S$) to rate new items $i_{j_{new}} \in I_{new}$, the matrix R can be filled with these new ratings (represented as red points in matrix R in Fig. 3). For simplicity sakes, seed users are presented in the upper part of the matrix R, and the new items – in its right part. If the values in V that correspond to the new items (grey part of matrix V, Fig. 3) can be automatically computed from these new ratings, the estimated ratings of other users (not seed users, the grey part of the matrix R in Fig. 3) on the new items can also be computed by multiplying matrices W and V (arrows in the lower part of Fig. 3).

The challenge here is thus to define a way to compute the new values $v_{kj_{new}}$ in matrix V, for each $k \in 1..K$ and $i_{j_{new}} \in I_{new}$. The matrix V can be completed by exploiting both new rating values from matrix R and the vectors $\mathbf{w}_{(*,s_k)}$ in W (see arrows on the upper part of Fig. 3). For each new item $i_{j_{new}} \in I_{new}$ this task simply comes down to solve a linear equations system (12). Note that, in order to obtain the unique solution of the system (12), the number of seed users should be equal to the number of latent features. That is we should have exactly K seed users.

$$\begin{cases} r_{s_1,j_{new}} = \sum_{p=1}^{K} w_{p,s_1} \cdot v_{p,j_{new}} \\ \vdots \qquad\qquad \vdots \\ r_{s_k,j_{new}} = \sum_{p=1}^{K} w_{p,s_k} \cdot v_{p,j_{new}} \end{cases} \tag{12}$$

For solving the linear equations system (12) all the seed users have to provide their ratings on the new item j_{new}. However, it is not always the case in reality. Users, who have been chosen as seeds, may not be familiar with the item that they are asked to provide the rating on, and/or may have no desire to rate some items. In this situation filling procedures can be used, that is missing ratings from seed users can be replaced by either some mean values (global mean rating or the mean values by certain item/user) or the ratings of the other closest candidates (for example, the next best RU candidates, see Section 3.1). In the latter case, not only the rating of the next best seed user should be taken, but also vector from the matrix W, that corresponds to this new seed user, should be used while solving the system (12).

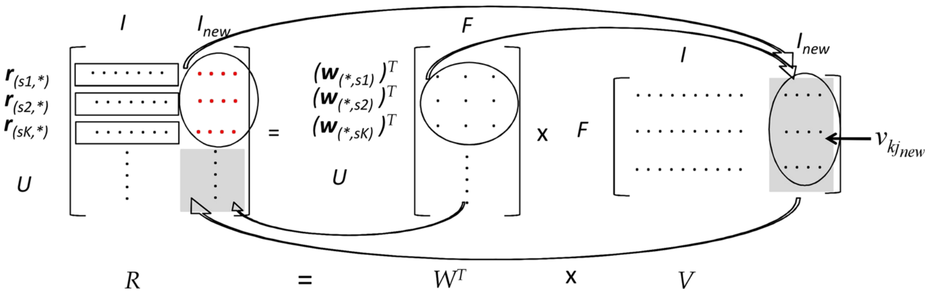

Fig. 3 MF: solving cold-start problem with seed users

3.4 Discussion

In this section we present how we propose to interpret latent features of MF as real users and how to use the obtained set of users as seeds for solving the new item cold-start problem. Our interpretation is based on a simple idea: finding canonical column-vectors (or those whose form is the closest to the canonical) in matrix W and then associating the users corresponding to these vectors with the features depending on the position of the maximum values in the vectors. The proposed interpretation procedure is simple and neither requires any human interaction (compared to the state of the art approaches that associate features with groups of interest (Zhang et al. 2006) or certain behavioral patterns (Pessiot et al. 2006)) nor any additional content information like the review topics (McAuley and Leskovec 2013).

Among works from the state of the art, (Liu et al. 2011) is the closest to ours in both ways: the way features of MF are interpreted (associated with users) and the resulting solution to the cold-start problem (the ratings on new items are predicted using the ratings of feature-associated users on these new items). However, to start with, authors of (Liu et al. 2011) form a rigid dependence of the features on the chosen users. This does not allow to solve the cold-start problem when one of the chosen users does not provide his/her ratings for some reasons. Contrarily, our approach allows to use ratings of the next best candidates in this case. Second, our method allows simultaneous identification of both representative users and representative items. In this case the matrix V should be analyzed in the same way as W was analyzed for the identification of representative users. At the same time the provided method is either user- or item-oriented. Third, to obtain an interpretable model, authors first fill one of the factor matrices with ratings of the representative elements (users or items), then the second matrix is formed through an optimization procedure, while our approach proposes interpretation within the original MF model (without changing it).

4 Data description and experimental protocol

4.1 Data description

To perform experimental evaluations of the proposed ideas we use 2 benchmark datasets: 100K MovieLens[1] and Jester.[2]

[1] https://movielens.org/

[2] http://www.ieor.berkeley.edu/goldberg/jester-data/

Table 3 Information about used data sets (MovieLens and Jester); θ – % of known ratings

charateristic	MovieLens	Jester
recommendation domain	movies	jokes
# users	943	24,983
# items	1,682	100
ratings characteristic	dicrete	real
ratings range	1 – 5	-10.00 – +10.00
ratings range after offset	1 – 5	+1.00 – +21.00
θ,%	6.3	72.5

MovieLens provides 100K discrete ratings on films ranging from 1 to 5 for 943 users on 1682 items. 6.3 % of user/item pairs have a rating value, the rest of ratings are unknown. Jester dataset (more precisely, its first most dense part) has 72.5 % of known ratings, that are real values ranging from -10.00 to +10.00 and are given by 24,983 users on 100 jokes. As in our experiments Non-Negative Matrix Factorization is used, which requires non-negativity of the input rating matrix, the ratings of the Jester dataset were offset by 11, thus resulting in the [+1; +21] values range. Table 3 summarizes information about both datasets used with percent of available ratings in the dataset, denoted by θ.

As we can see, MovieLens dataset is rather sparse. While simulating the procedure of asking seed users to provide their ratings on new items (extracting appropriate ratings from the test set), usually we can get no more than 40 % of answers. Thus we will be forced to use a filling procedure (see Section 3.3) that makes impossible to study some aspects of the proposed approach using a dataset with so small number of given ratings. Contrary to MovieLens, using Jester dataset we can obtain considerable number of new items, for which ratings of all seed users are known, and, consequently, we can study in detail the models proposed in this paper. Therefore, Jester dataset is used as a basic dataset in our experiments. The MovieLens dataset is used to confirm some results (those, which do not require the presence of the ratings of all seed users) and to study their data-independence. It may seem that using Jester dataset reduces our approach to the case of non-sparse rating matrices, which is not the case in many real applications. However, we study the performance of our models depending on the sparseness of the input rating matrix (learning set) as well. For this, some ratings are discarded from the learning subset to obtain, for example, a learning matrix with 10 % or 5 % of known ratings. At the same time, all ratings in the test set are preserved in order to provide high rate of answers of seed users concerning new items.

4.2 Alternative methods for seeds identification

In this subsection we describe several strategies for finding sets of seed users (ensembles of seeds users, or seeds), which will be further used as an alternatives for comparison with the approach proposed in this paper (representative users).

Inspired by works of Rashid et al. (2002) and Liu et al. (2011), we have chosen the following strategies for seeds identification as an alternative:

1. Top raters (*topK*) – the set of users, who provide the largest number of ratings in the system.

2. Most diverse users (*mDiv*) – the set of users, who have different rating behavior; the diversity is measured in terms of pairwise correlation, that is this set is formed of those users, who have the lowest pairwise correlation within their set.
3. Most dispersive (*mDisp*) – users of this set are chosen in such a way, that every selected user rates items differently; that is, he/she has dispersion in the values of provided ratings (contrary to the case, when a user rates all items equally, for example).
4. Most neighbors (*mNeigh*) – the set of users, who occur in the largest number for neighborhoods in the NB approach.

In all following experiments the number of seed users in the set is equal to the number of features of the MF model. The performance of the proposed model is evaluated similarly to (Liu et al. 2011), where different strategies of seed users identification were compared in the frame of the same method for the cold-start problem solution. Additionally we compare our approach with the baseline one, presented in Section 2.5.

4.3 Experimental protocol

Following the general tendency in the recommender systems community, in order to obtain more reliable experimental results, a 5-fold cross validation is performed. In every case, both for cold-start and non-cold-start evaluations, the original rating matrix is divided into test and learning sets, containing 20 % and 80 % of the original information respectively. This procedure is done 5 times in a way to obtain 5 non-intersecting test sets. This ensures some sort of independence of resulting pairs of test and learning sets (folds). After that values of required characteristics (evaluation measures) are calculated on each fold. The final result is a mean value of 5 values obtained for each fold.

For the non-cold-start case we use a classical evaluation protocol, that is 20 % of randomly chosen ratings form the test set and the rest 80 % are used as learning set for the model training. In the case of cold-start experiments, 20 % of items are randomly chosen as the new items (I_{new}) with their ratings forming the test set. The ratings of the remaining 80 % of items are used to train the model. Note that in this case the learning and the test sets do not necessarily contain exactly 80 % and 20 % of ratings respectively. Once the model is trained and the set of appropriate seed users is formed, their ratings are extracted from the test set (this procedure simulates the process of asking seed users to provide their ratings on new items). After this, the remaining ratings in the test set are used for the model evaluation. The procedure of forming learning and test sets for both cold-start and non-cold-start cases is schematically presented in Fig. 4. For the simplicity of visualization the test ratings and test items are grouped in the left part of the rating matrix R. In the experiments, when all seed users are required to provide a certain percent of ratings on new items, the test set is formed in another way. Only those new items, on which seed users of the chosen ensemble have the required percent of ratings (*actual test set*, see the set is in red bold rectangle in the left part of the Fig. 4) are used. In such cases the test set contains less than 20 % of items and the test set may vary for different ensembles of seed users.

4.4 Evaluation metrics

For the experimental evaluation, we use four metrics: Normalized RMSE (NRMSE), Normalized Distance-based Performance Measure (NDPM), Relative Deterioration (DET) and Test Coverage (COV). NRMSE is the normalized version of the widely used evaluation

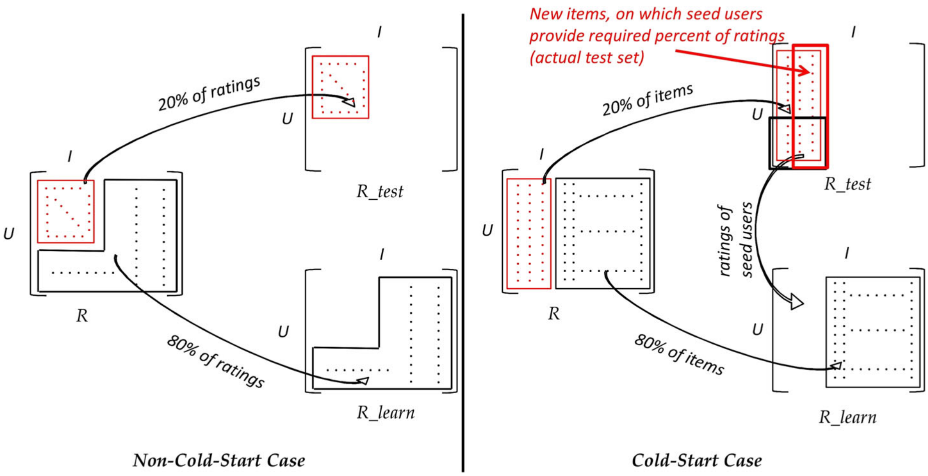

Fig. 4 Forming learning and test sets for non-cold-start and cold-start experiments

measure RMSE (Root Mean Square Error), which is computed through the difference between real and estimated ratings (Shani and Gunawardana 2011). NRMSE is a fraction of the RMSE value and the difference between maximum ($maxR$) and minimum ($minR$) possible rating values in the dataset (see (13)). Possessing the normalization property, NRMSE does not depend on the ranges of ratings in the input data.

$$NRMSE = \frac{RMSE}{maxR - minR} = \frac{\sqrt{\sum_{l=1}^{L}(r_l - \hat{r}_l)^2/L}}{maxR - minR} \tag{13}$$

where L corresponds to the number of ratings in the test set, r_l represents a rating value from the test set, $\hat{r}_l$ – the corresponding estimated value. The $NRMSE$ measure evaluates how close is the predicted rating to the real one.

However, in practice it is more important to predict the utility order of the items for a certain user rather than to estimate the exact value of the rating. In reality, the recommendation engine is usually requested to provide an ordered list of items, that the current user will like the most. Therefore, in the current work we will use a ranking measure as a primary evaluation measure. Still we present the results in terms of NRMSE as well for the comparison purposes, as this metric is often used for RS evaluation.

Ranking-based evaluations can be done through the NDPM measure (Shani and Gunawardana 2011). Assume that we have real and predicted ratings of items for a user u. Define C_u as the number of pairs of items for which the real ranking asserts an ordering (*i.e.* not tied), that is the number of pairs with different values of the ratings. Define as C^+ and C^- the number of pairs for which the model ranking asserts the correct order and incorrect order respectively. Finally, denote by C_u^0 the number of pairs where real ranking does not tie the elements (they have different ranking positions) but the model ranking ties. Thus the following equality holds: $C_u^0 = C_u - (C_u^+ + C_u^-)$. The NDPM is given by (14) according to (Shani and Gunawardana 2011).

$$NDPM = \frac{C_u^- + 0.5\, C_u^0}{C_u} \tag{14}$$

The maximum NDPM value, namely 1, is reached when the model places all non-tied by real ranking pairs of elements in inverse order, that is $C_u = C^-$, $C^+ = 0$ and $C_u^0 = 0$. Thus $NDPM = 1$ is the worst obtained value. The mean value (0.5) can be obtained for the case when the model ties all the elements, that is all the elements are predicted to have the same value of ratings. Note, however, that other configurations (not only tying all elements) can also result in $NDPM = 0.5$. Finally, the lowest NDPM (0) indicates that all non-tied by real ranking pairs of elements (those, having different ranking positions in reality) were correctly ordered by the model (the best ranking), that is $C_u = C^+$, $C^- = 0$ and $C_u^0 = 0$.

For comparison purposes, in some cases we will use not the exact values of the error metrics, but the relative deterioration of the error of analyzed model (AM) compared to the error of the base model (BM) (see (15)).

$$DET\,(AM, BM) = \frac{err\,(AM) - err\,(BM)}{err\,(BM)} 100\,\% \tag{15}$$

As Test Coverage (COV), we understand the percentage of ratings from the test set, for which the model can estimate rating values (see (16)).

$$COV = \frac{|T_{predicted}|}{|T|} 100\,\%, \tag{16}$$

where $|T_{predicted}|$ is the number of predicted ratings in the test set (the *actual test set* in Fig. 4), and $|T|$ is total number of ratings in test set.

5 Experimental results

In this section we present the experimental evaluation of the proposed approach. First, we focus on the identification of the optimal number of features for MF. Second, we analyze the characteristics of different sets of seed users. Finally, we analyze the performance of the MF-based solution for the new item cold-start problem.

5.1 MF: performance analysis

In this subsection we search for the optimal value of the number of features K for the MF-based models.

We conduct a series of experiments with different number of features and different values of the regularization parameter λ, for both Jester and Movielens datasets. For Jester, λ changes from 0 to 300 with an increment of 5, for Movielens – from 0 to 30 with an increment of 1. Note that the number of features K and the value of the regularization parameter λ are not the parameters of our model (as the proposed interpretation is made for the existing MF model), but the parameters of MF itself.

Tables 4 and 5 present the values of optimal configurations, with respect to different error measures for Jester and Movielens datasets, as well as errors (NRMSE and NDPM) for the boundary λ values (0 and 300/30). Minimum and maximum values of NRMSE and NDPM through different numbers of features K are presented in the tables as shadowed. The last row of the tables contains the difference between these maximum and minimum error values. As it is seen form the tables, when the optimal value of lambda is used for each number of features, the difference between error values for different number of features is insignificant and does not exceed for NRMSE and NDPM respectively 0.0007 and 0.0064 (Jester), 0.0109 and 0.0135 (MovieLens).

Table 4 Jester: optimal parameter values; maxDif – difference between maximum and minimum error values (presented as shadowed) through different number of features K

K	config	value/λ	
		NRMSE	NDPM
5	min	0.2120 / 0	0.3645 / 0
	opt	**0.2061 / 60**	**0.3533 / 55**
	max	0.2297 / 300	0.3837 / 300
10	min	0.2147 / 0	0.3678 / 0
	opt	**0.2054 / 75**	**0.3482 / 70**
	max	0.2296 / 300	0.3836 / 300
15	min	0.2190 / 0	0.3700 / 0
	opt	**0.2057 / 90**	**0.3493 / 110**
	max	0.2295 / 300	0.3836 / 300
20	min	0.2238 / 0	0.3721 / 0
	opt	**0.2056 / 90**	**0.3482 / 100**
	max	0.2296 / 300	0.3836 / 300
25	min	0.2290 / 0	0.3834 / 0
	opt	**0.2059 / 95**	**0.3492 / 100**
	max	0.2294 / 300	0.3835 / 300
50	min	0.2539 / 0	0.4073 / 0
	opt	**0.2055 / 100**	**0.3469 / 100**
	max	0.2294 / 300	0.3835 / 300
75	min	0.2675 / 0	0.4195 / 0
	opt	**0.2057 / 100**	**0.3484 / 100**
	max	0.2295 / 300	0.3836 / 300
maxDif		0.0007	0.0064

Analyzing the values given in the Tables 4 and 5, we can note that when the number of features K increases, the value of optimal λ also has a tendency to increase. This fact supports the existence of the overfitting problem, that is with the growth of its size the MF-model becomes more precise on the learning set, but less accurate on the test set. It is obvious that the quality of prediction on the learning set increases with the number of features K, thus the higher penalty (value of the regularization parameter λ) should be used to smooth this effect. As it was discussed previously, using the optimal value of the regularization parameter λ lets us obtain a precise model when the number of features K is not very large. Thus, in order to have an optimal model in terms of its size and representativeness, we used $K = 10$ as the number of features in all further experiments on both datasets.

5.2 Analysis of different sets of seed users

This subsection is dedicated to the analysis of the main characteristics of different sets of seed users. This is done in order to understand the ability of seeds to represent the interests of the entire population of users. We focus on the mean number of ratings per seed user, the average correlation within the set of seed users ($innerC$) and the average correlation of seed users with other users ($outerC$), as well as the ratio of these two correlation values ($outerC/innerC$).

The characteristics of different studied sets of seed users: representative users (RUs), top raters (topK), most diverse users (mDiv), most dispersive users (mDisp) and most neighbors (mNeigh), are depicted in Tables 6 and 7 (Jester and MovieLens datasets respectively). By χ (the first column of the tables) we denote the ratio of the mean number of ratings provided by seed users to the mean number of ratings per user in the whole dataset (not seed users). As χ is a relative characteristic, it is data-independent and lets us compare results for different

Table 5 MovieLens: optimal parameter values; maxDif – difference between maximum and minimum error values (presented as shadowed) through different number of features K

K	config	value/λ	
		NRMSE	NDPM
5	min	0.2584 / 0	0.3423 / 0
	opt	**0.2364 / 3**	**0.3050/ 4**
	max	0.2769 / 30	0.3237 / 30
10	min	0.2457 / 0	0.3330 / 0
	opt	**0.2421 / 6**	**0.3058 / 8**
	max	0.2755 / 30	0.3262 / 30
15	min	0.2711 / 0	0.3622 / 0
	opt	**0.2451 / 8**	**0.3100 / 10**
	max	0.2759 / 30	0.3247 / 30
20	min	0.2759 / 0	0.3734 / 0
	opt	**0.2452 / 9**	**0.3075 / 11**
	max	0.2763 / 30	0.3258 / 30
25	min	0.2852 / 0	0.3731 / 0
	opt	**0.2455 / 10**	**0.3051 / 12**
	max	0.2763 / 30	0.3261 / 30
50	min	0.2969 / 0	0.3868 / 0
	opt	**0.2473 / 10**	**0.3056 / 13**
	max	0.2763 / 30	0.3248 / 30
100	min	0.2979 / 0	0.3883 / 0
	opt	**0.2461 / 10**	**0.3047 / 14**
	max	0.2765 / 30	0.3246 / 30
500	min	0.2692 / 0	0.3586 / 0
	opt	**0.2435 / 8**	**0.2982 / 9**
	max	0.2754 / 30	0.3234 / 30
1000	min	0.2566 / 0	0.3374 / 0
	opt	**0.2424 / 7**	**0.2965 / 5**
	max	0.2751 / 30	0.3231 / 30
maxDif		0.0109	0.0135

datasets. We can see from the Tables 6 and 7 that for both datasets, RUs tend to rate less than in the other sets of seed users. However, RUs of the Jester dataset has a higher ratio χ then in MovieLens. As expected, top-raters have the highest number of ratings.

Now we proceed to the analysis of correlation. We assume that a set of users is more suitable for representing the interests of the entire population of users if it is composed of users with different behavioral patterns. That is the users of this set should be less correlated between them than with the users outside the set (the value of the ratio $outerC/innerC$ should be higher).

Table 6 Jester: characteristics of seed users; χ – ratio of the mean number of ratings provided by seeds to the mean number of ratings per user in the whole dataset

seeds	χ	innerC	outerC	outerC/innerC
RUs	0.7111	0.0107	0.0617	5.77
topK	1.1306	0.0650	0.0981	1.51
mDiv	0.8598	0.0137	0.0891	6.50
mDisp	0.9446	0.1091	0.1224	1.12
mNeigh	0.4367	0.7136	0.3290	0.46

Table 7 MovieLens: characteristics of seed users; χ – ratio of the mean number of ratings provided by seeds to the mean number of ratings per user in the whole dataset

seeds	χ	innerC	outerC	outerC/innerC
RUs	0.4838	0.0134	0.0766	5.72
topK	4.1616	0.2344	0.1840	0.79
mDiv	1.4892	0.0381	0.1143	2.70
mDisp	1.3750	0.1072	0.1520	1.42
mNeigh	1.0849	0.4468	0.2757	0.62

For both datasets, RUs are almost 6 times less correlated withing their set than with other users of the dataset (not seed users). The set of most diverse users (mDiv) has considerably lower inner correlation, compared to the value of the correlation with not seed users ($outerC/innerC > 1$). This is logical, as this set was formed as a set of those users, that are not correlated among each other. But as a random factor was used when forming this set (the first user of the most diverse set is chosen randomly), it can be not optimal. For example, for the Jester the value of inner correlation for the RUs and mDiv sets is very close (both sets are composed of highly diverse users), however for the MovieLens dataset we can see that the set of representative users has lower inner correlation. It means that both sets RUs and mDiv represent different behavioral patterns, but the set of representative users is more optimal.

The ratio of outer and inner correlation ($outerC/innerC$) of the top raters (topK) and most dispersive (mDisp) sets is close to 1 for both datasets. The value of the inner correlation of users from these sets is close to the mean pairwise correlation of the users from the whole datasets. The ensemble of most neighbors is composed of users that have higher correlation within the set than outside it. Therefore, these three sets are less suitable for representing the interests of the whole population of users.

Considering statements above, we can conclude that the set of representative users tend to be composed of users with different behavioral patterns (as it has the lowest inner correlation and high value of the ratio $outerC/innerC$) and thus it can be used for representing interests of the entire population of the users.

5.3 Cold-start for Jester

In the following experiments we analyze the performance of the proposed solution of the new item cold-start problem, i.e. exploiting seed users. We start with the detailed analysis of our approach. For this we first need that all seed users provide ratings on the new items. Due to the high sparsity of the MovieLens dataset, none of the new items in our simulation can get ratings from all seeds, but this is not the case for the Jester dataset. Therefore, in this subsection we focus on different aspects of the cold-start problem analysis performed on the Jester dataset. We will consider the case when all seed users provide ratings and will analyze different filling procedures when not all seed users can give their opinion on a specific new item. Also, we compare performance for different levels of learning dataset sparseness. Some results for the MovieLens dataset will be presented in the next subsection.

Let us introduce some notations. By *MF-RUs* model we denote an MF-based algorithm for solving cold-start problem (Section 3.3) with representative users (RUs) used as seed users. *MF-topK* will correspond to the same algorithm with the set of top raters used as seed users. Therefore, in such abbreviations the second part will correspond to the set of seed users used (*RUs*, *topK*, *mDiv*, *mDisp*, *mNeigh*).

5.3.1 Comparison of different MF-based models and RBMF

We start with the analysis of the performance of different MF-based models on Jester dataset. In this set of experiments, seed users are required to provide all ratings on new items. Thus, the test set is formed only of those new items, on which all seed users can give their ratings. We also compare the performance of the proposed approach with the baseline approach from the Section 2.5 (RBMF). Additionally, we study the effect of the sparsity of the learning set, that is we randomly discard some portion of ratings from the learning set in order to obtain the learning matrix with the required percentage of known ratings. At the same time all ratings in the test set are preserved, what ensures the presence of ratings provided by seed users on new items.

Figures 5 and 6 present the evolution of NRMSE and NDPM respectively for different models (MF-RUs, MF-topK, MF-mDiv, MF-mDisp, MF-mNeigh and RBMF) with respect to different percents of known ratings in the learning set, denoted by θ. The figures show that among all MF-* models, MF-RUs performs the best regadless the value of θ. The results of the MF-topK, MF-mDiv, MF-mDisp, MF-mNeigh models tend to be close between themselves. The RBMF model performs better than MF-RUs in terms of RMSE, however it provides consistently worse results in terms of NDPM. Still, the results provided by RBMF are the closest to those of MF-RUs model. Therefore we can conclude that the set of RUs can represent the interests of the entire set of users better, than other alternative sets of seed users and can better predict elements ranking than the benchmark model (RBMF). The MF-RUs model performance for small values of θ ($\theta = 10$ % or $\theta = 5$ %) prove that the proposed method can be used for sparse datasets as well. Now we move to the more detailed analysis of MF-RUs model.

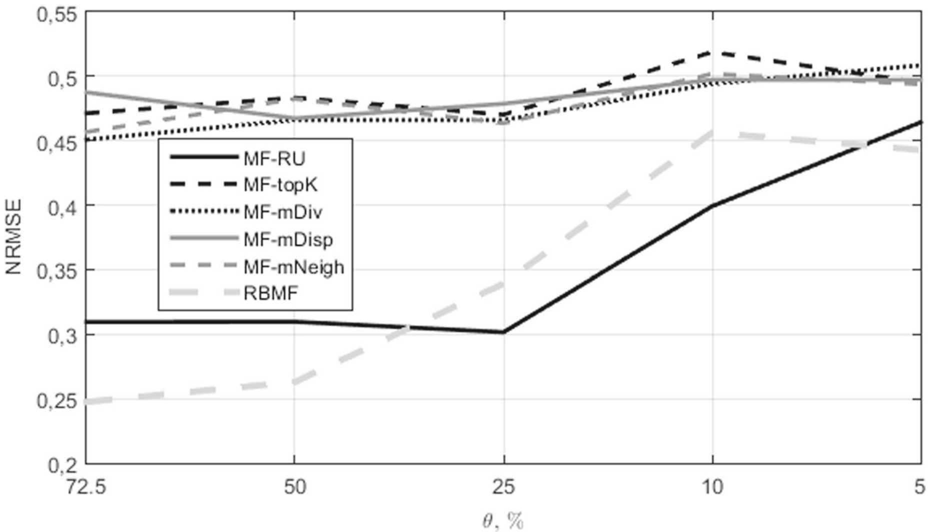

Fig. 5 Jester: dependence of NRMSE on percent of known ratings in the input rating matrix (θ)

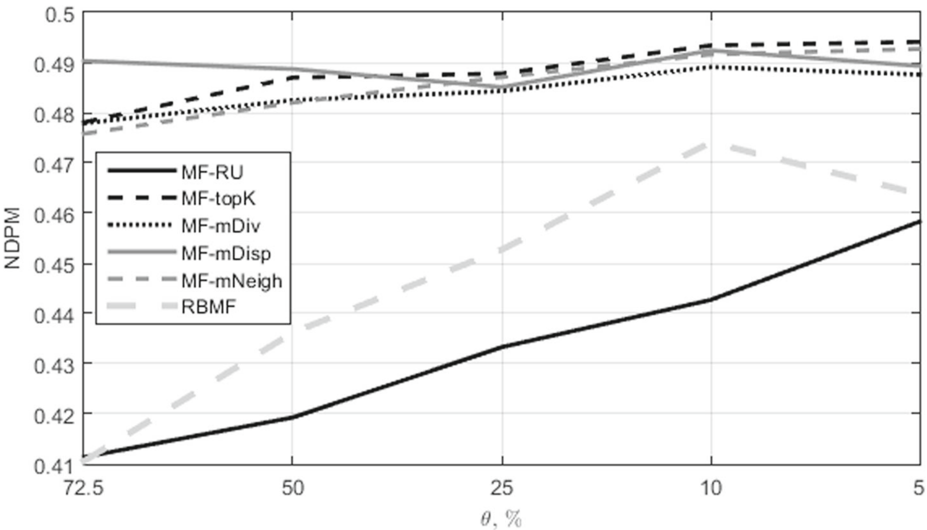

Fig. 6 Jester: dependence of NDPM on percent of known ratings in the input rating matrix (θ)

5.3.2 MF-RUs: not all seed users provide ratings on new items (No Coverage Growth)

Till now, we were studying the case when seed users in MF-based models provide all the ratings on new items. However, in reality not all users from the set of seed users may be able to provide ratings on new items for different reasons, like absence of knowledge about the item or simply non-willingness to answer. As it was proposed in the Section 3.3, unknown ratings can be filled with either mean values (we will use global mean *Gmean*, user mean *Umean* and user-item mean *UImean*) or ratings of other users *User* (in the case when RUs are chosen as seed users – the ratings of the next closest representative user, see Section 3.1).

We focus on studying different filling procedures. We compute error values of MF-RUs model depending on the percent of ratings provided by representative users for different mean-fillings and for filling with the ratings of the next closest representative user.

In order to analyze the influence of the missing ratings, as in the previous experiments, we evaluate error measures only on those new items, that have ratings from all representative users. Thus, the coverage (COV) on the test set does not change when the threshold of required number of ratings from the representative users (γ) decreases. Indeed, when the threshold of required number of ratings from representative users γ decreases, usually more new items can be used in the actual test set (see Fig. 4, right part). However, in this case the actual test set is fixed and is composed of those new items, on which representative users provide 100 % of ratings. The coverage growth when γ decreases will be studied in the next series of experiments (Section 5.3.3). In order to simulate the absence of some ratings of representative users, we randomly delete the required number of votes for each considered new item.

The resulting evolutions of NRMSE and NDPM on different values of γ and for different filling procedures are presented in Figs. 7 and 8. The lowest value of NRMSE is obtained when the Umean-filling (filling with the per-user mean rating) is used. Also, contrary to what has been expected, NRMSE has a tendency to decrease for Umean and UImean. When

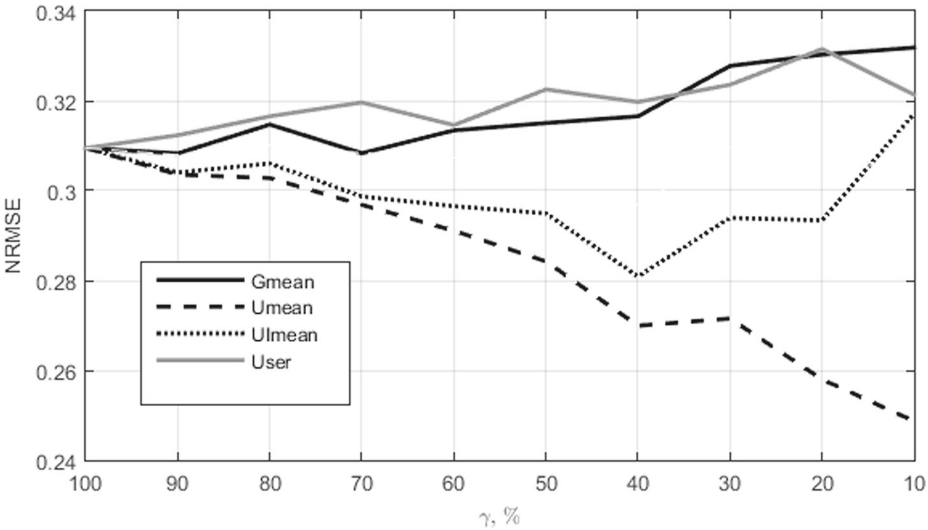

Fig. 7 Jester: dependence of NRMSE on the required number of ratings from RUs (γ) for different filling procedures

Gmean-filling is used, the value of NRMSE increases following the case for the User-filling, but usually stays lower than for the latter one. The decrease of the NRMSE when γ decreases shows that the rating value of the new item can be predicted as a per-user mean rating. Indeed, performing non-cold-start test for predicting new ratings with the per-user mean model (unknown ratings are estimated as the mean rating of an active user) we obtain RMSE equal to 0.2283 that is 11 % higher than RMSE of the MF model for $K = 10$

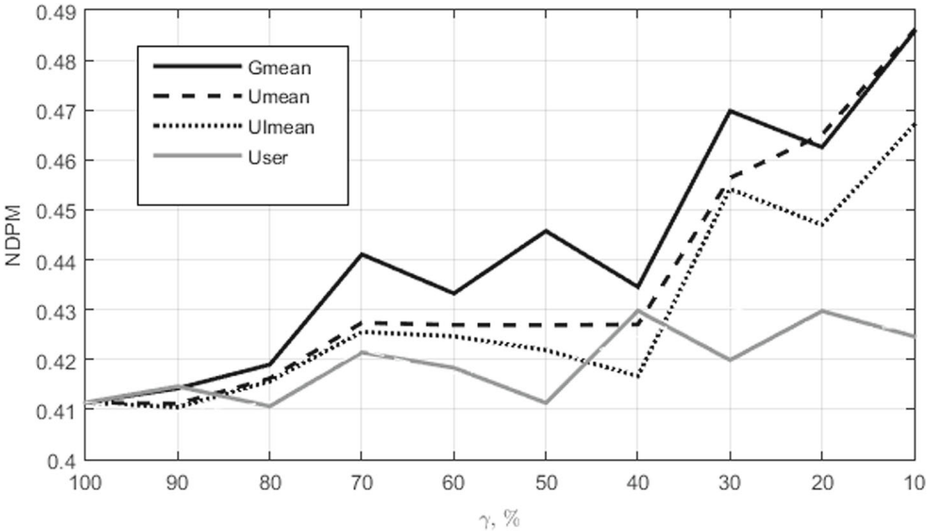

Fig. 8 Jester: dependence of NDPM on the required number of ratings from RUs (γ) for different filling procedures

(0.2054, see the Table 4) and even less than RMSE of the NB model with 10 neighbors equal to 0.2382. However, despite the fact that the values of the ratings can be predicted with the mean model, this model cannot estimate the ranking of the items, as all of them are predicted to have the same rank. This shows that the NRMSE measure has a bias when mean-fillings are used. Also, as it was discussed earlier (see Section 4.4), NDPM error measure has more practical meaning, so we will consider it as the main evaluation criteria.

Analyzing the dependence of NDPM (see Fig. 8) we can see that, as expected, the error value grows with the decrease of γ. Also, using the ratings of real users (filling with the ratings of the next closest candidate, User-filling procedure) can significantly improve the performance, specially for the case when the value of γ is small (30 %–10 %). It shows that the ratings of real users have practical value and are more suitable for ranking estimation than mean-fillings.

5.3.3 MF-RUs: not all seed users provide ratings on new items (Coverage Growth)

Finally, as it was mentioned in Section 5.3.2, we now analyze the performance of the models in the case of coverage growth. In this series of experiments the actual test set is not fixed for different values of γ. For example, if γ is set to 40 %, this means that all new items that have been rated by at least 40 % of the representative users are analyzed in the actual test set (in the previous case, the actual test set was composed of only those new items that have 100 % of ratings from representative users). This naturally results in a coverage growth while γ decreases.

We compute NRMSE and NDPM, as well as the test coverage COV, for different values of γ and different filling procedures. The results are presented in Figs. 9 and 10.

As in the previous case, where no coverage growth was considered, we can observe an unexpected behavior of NRMSE: the error tends to decrease with γ, either on a certain interval (for Gmean, UImean and User-filling) or for all values of γ (Umean-filling). Also mean-filling procedures result in a lower NRMSE than User-filling (due to the bias of the NRMSE measure discussed in Section 5.3.2). On the other hand, the analysis of the

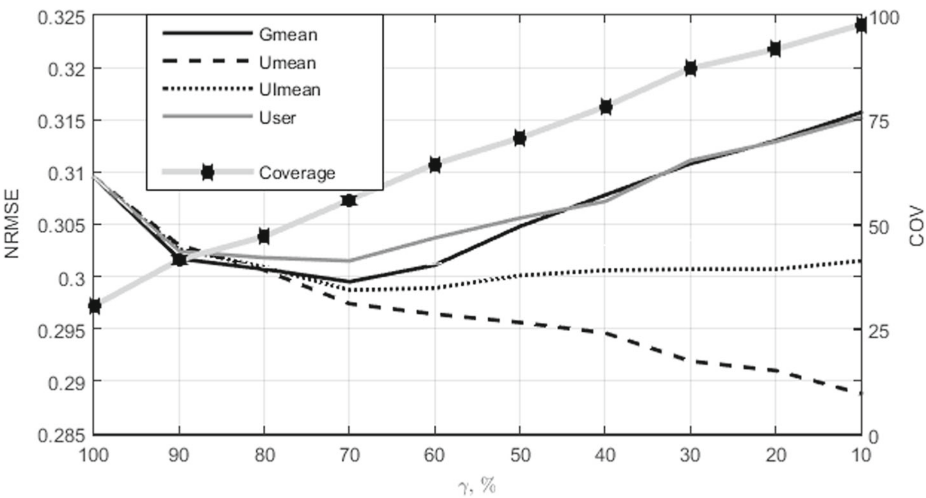

Fig. 9 Jester: dependence of NRMSE for different filling procedures and test coverage (COV) on the required number of ratings from RUs (γ)

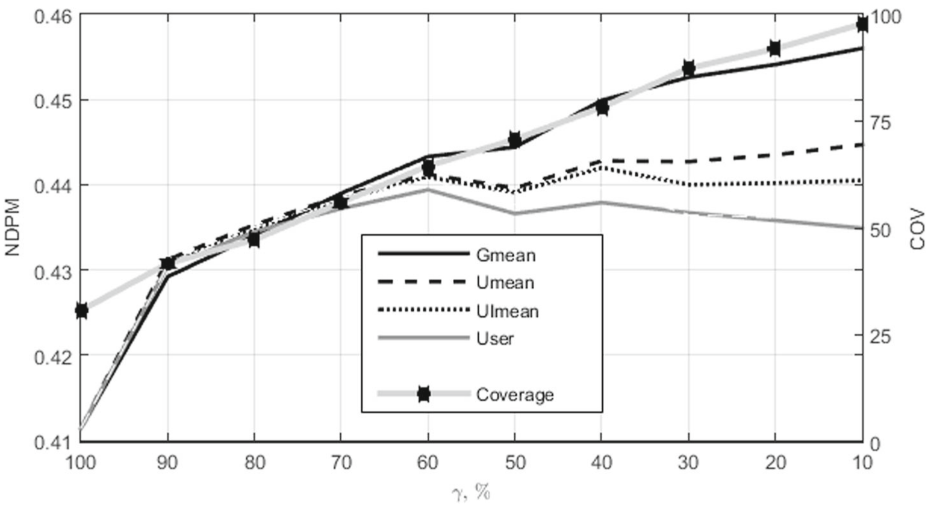

Fig. 10 Jester: dependence of NDPM for different filling procedures and test coverage (COV) on the required number of ratings from RUs (γ)

NDPM shows that the usage of ratings of real users instead of mean-fillings results in better elements ranking.

Moreover, in the case where ratings of other RU candidates are used instead of unknown values, we obtain noticeable improvements in terms of NDPM (compare $NDPM(\gamma = 10\ \%) = 0.4357$ for User-filling and $NDPM(\gamma = 10\ \%) = 0.4560$ for Gmean-filling) and in terms of coverage (compare $COV(\gamma = 100\ \%) = 31\ \%$ and $COV(\gamma = 10\ \%) = 98\ \%$). Thus we have shown that using ratings of representative users or other closest candidates can not only improve the quality of the ranking, but also allows accurately predict ratings for more new items.

To conclude we can say that our approach (MF-RUs) outperforms in terms of NDPM alternative methods of seed users identification as well as the benchmark RBMF approach. Also it is not computationally complex and does not require performing additional calculations, like calculation of the correlation matrix for the identification of mDiv and mDisp sets or prior SVD decomposition, as RBMF. Additionally, when using representative users as seeds the unknown ratings from seed users can be replaced with the ratings of other closest candidates for being representative users (User-filling procedure), that allows make predictions for more new items.

5.4 Cold-start for MovieLens dataset

In the previous subsection we studied different aspects of the proposed approach on the Jester dataset. To confirm some of the results, we will analyze our approach on the MovieLens dataset. As this dataset contains only 6 % of known ratings, none of the new items has more than 40 % of ratings provided by seed users ($\gamma \leq 40\ \%$). Note that the previous phrase does not mean that the proposed approach can not identify the required number of representative users, but that due to the sparsity of the dataset there is not enough of ratings in the test provided by RUs. In real situation, as it was mentioned in Liu et al. (2011) the chosen representative users will be encouraged to provide all ratings on the new items through, for example, the proposition of additional services from the side of RS. So, most of their ratings will be known.

Thereby, in every experiment we have to use one of the filling procedures. So it is impossible to perform the same series of experiments for MovieLens, as it was done for Jester. Therefore, in this subsection we search an answer for only two questions: 1) does MF-RUs model remain the best among other MF-based models as well as RBMF? and 2) which filling procedure will result in better performance of MF-RUs model? As both questions concern the comparison of the models, we will use relative deterioration (DET, see (15)) as an evaluation metric. Experiments performed on the Jester Dataset support the statement from the Section 4.4, that NDPM measure has more practical meaning than NRMSE, thus in this subsection we will focus on the evaluation of NDPM only.

Searching for the answer on the first question, we calculate the relative deterioration in terms of NDPM for different models (analyzed models), compared to the MF-RUs model (base model), for different values of γ. UImean-filling procedure (filling with the mean of the user and item mean values) was used, as it resulted in the best NDPM values for the Jester Dataset among other mean-filling procedures (see Figs. 8 and 10). Results are presented in Table 8.

A positive value of DET means that the MF-RUs model gives better results (lower value of NDPM). An absolute value in Table 8 indicates the relative deterioration in percents of the analyzed model comparing to the base one (MF-RUs). For example, the value 11.0 in the line $\gamma = 30$ % and the column $mDisp$ shows that for the specified γ MF-RUs model performs 11.0 % better than MF-mDisp (that is MF-RUs results in NDPM that is 11.0 % less than NDPM for MF-mDisp). As all the values in the Table 8 are positive, we can conclude that for all values of γ the MF-RUs model results in better ranking, compared to other models. Therefore we obtained the answer on the first question. Also we can see that when γ decreases from 30 % to 10 % the value of DET also decreases. It means that when seed users provide less ratings, the difference between the models performance diminishes. Indeed, for $\gamma = 10$ %, 90 % of ratings are filled with UImean values, but not with the ratings of seed users, thus the models with different ensembles of seed users become more similar. Also, supporting the same conclusion for the Jester dataset, results of the RBMF are the closest to MF-RUs model results.

The next question that we rise is: which of four filling procedures (Gmean, Umean, UImean, and User) will be the best for the *MF-RUs* model. Similar to the previous case, we calculate the relative deterioration DET in terms of NDPM of the analyzed models (MF-RUs with the mean-filling procedures) compared to the base model (MF-RUs with the User-filling). The positive value of DET indicates that MF-RUs with User-filling provides a lower NDPM value. Results are presented in the Table 9.

Table 8 MovieLens: relative deterioration (DET) of NDPM in % of different models compared to MF-RUs model through different values of γ (UImean-filling is used)

γ,%	SEEDS for MF-models				RBMF
	topK	mDiv	mDisp	mNeigh	
40	8.5	6.4	8.2	8.6	4.9
30	11.0	9.6	11.0	11.3	7.3
20	7.1	6.0	7.1	8.2	3.7
10	5.5	5.7	5.0	6.6	2.3

Table 9 MovieLens, MF-RUs model: relative deterioration of NDPM in % of mean-filling procedures compared to User-filling for different values of γ

γ,%	DET		
	Filling		
	Gmean	Umean	UImean
40	7.8	9.1	3.4
30	11.7	12.7	6.5
20	19.6	19.2	17.4
10	25.2	24.4	21.6

From the Table 9 we can say that using User-filling procedure we can obtain better ranking (as all the values in the table are positive). Also we can see that with the decrease of γ (percentage of known ratings provided by RUs) the gain of using User-filling increases. Indeed, the lower the value of γ, the more real information is obtained through the User-filling compared to the mean-filling procedures. This was also the case for the Jester dataset (see the Figs. 8 and 10: when γ decreases the gain of using User-filling increases).

To conclude, we can say that the results obtained for MovieLens dataset support the conclusions drawn from Jester: 1) among all models, MF-RUs gives the best values of NDPM and 2) using ratings of other closest users instead of mean-filling procedures improves the quality of items ranking.

6 Conclusions and future work

In this paper we proposed to interpret features of an already existing matrix factorization model as real users of the system – *representative users*. The proposed interpretation is done completely automatically and, contrary to many state of the art approaches, does not require neither human experience or interaction, like in (Zhang et al. 2006; Pessiot et al. 2006), nor external sources of information, like item reviews (McAuley and Leskovec 2013). It also allows to explain recommendations provided by MF models (see Section 3.2). Note that the proposed interpretation is done for the existing MF model, thereby we do not propose a new factorization technique.

As latent features represent the relations between users and items, we assume that the resulting feature-related representative users should be capable to correctly represent the interests of the whole population of users. This assumption forms the basis of the new-item cold-start problem solution proposed in this paper, where representative users are used as seed users to provide their opinion on new items. After receiving the opinion of representative users on new items and using their relation to other users of the system (through the latent features of the model), the ratings of other users on these new items can be estimated. This new item cold-start problem solution does not require any content information, and thus can be used in cases when this type of information is unavailable.

Using two datasets (MovieLens and Jester), the evaluation of the proposed approaches was performed. First of all, analyzing characteristics of the set of representative users it was shown that they tend to be composed of users with different behavioral patterns and can thus be used for representing the interests of the entire population of the users.

Considering the performance on the cold-start, it was shown that using representative users as seeds results in better ratings predictions than alternative sets of seed users (such as top raters or the set of most diverse users) and that the MF-RUs model provides better ranking than the baseline approach (RBMF). Also, if for some reasons the chosen representative users do not provide their ratings on new items, the next best candidates for being representative users can be successfully used. This allows not only to increase the accuracy of prediction (compared to the filling unknown ratings with some mean values, like global mean rating), but also to predict ratings for more new items. In our opinion the ability of representative users to solve the new item cold-start problem can be considered as a proof of the validity of the proposed MF features interpretation.

In our future work we would like to study the notion of users representativeness in the frame of matrix factorization-based models with the respect of time. That is to propose approaches for identification of those users, who can correctly present the changes of interests of the whole population with the time course. Further we would like to study if the usage of content information can allow to identify representative elements (either users or items) more accurately. Also it would be interesting to extend the proposed approaches for other MF models like Probabilistic MF, Tensor Factorization, Collective MF.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749.

Agarwal, D., & Chen, B.C. (2010). flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the 3d ACM international conference on web search and data mining* (pp. 91–100): ACM.

Aleksandrova, M., Brun, A., Boyer, A., & Chertov, O. (2014). Search for user-related features in matrix factorization-based recommender systems. In *Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases - Doctoral session* (p. 10).

Amatriain, X., Lathia, N., Pujol, J.M., Kwak, H., & Oliver, N. (2009). The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval* (pp. 532–539): ACM.

Behnke, S. (2003). Discovering hierarchical speech features using convolutional non-negative matrix factorization. In *Proceedings of the international joint conference on neural networks*, (Vol. 4 pp. 2758–2763): IEEE.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132.

Brun, A., Aleksandrova, M., & Boyer, A. (2014). Can latent features be interpreted as users in matrix factorization-based recommender systems? In *Proceedings of 2014 IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technologies*, (Vol. 2 pp. 226–233): IEEE.

Chagoyen, M., Carmona-Saez, P., Shatkay, H., Carazo, J.M., & Pascual-Montano, A. (2006). Discovering semantic features in the literature: a foundation for building functional associations. *BMC Bioinformatics*, *7*(1), 41.

Das, A.S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on world wide web* (pp. 271–280): ACM.

Desrosiers, C., & Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook* (pp. 107–144): Springer.

Devarajan, K. (2008). Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS Computational Biology*, *4*(7), 12.

Enrich, M., Braunhofer, M., & Ricci, F. (2013). Cold-start management with cross-domain collaborative filtering and tags. In *E-commerce and web technologies*, (Vol. 152 pp. 101–112): Springer.

Esslimani, I., Brun, A., & Boyer, A. (2013). Towards leader based recommendations. In *The influence of technology on social network analysis and mining* (pp. 455–470): Springer.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., & Schmidt-thieme, L. (2010). Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of 2010 IEEE 10th international conference on data mining (ICDM)* (pp. 176–185): IEEE.

Goreinov, S., Oseledets, I., Savostyanov, D., Tyrtyshnikov, E., & Zamarashkin, N. (2010). How to find a good submatrix. Matrix methods: theory, algorithms and applications, 247–256.

Graus, M. (2011). *Understanding the latent features of matrix factorization algorithms in movie recommender systems*: Master's thesis, Technische Universiteit Eindhoven.

Guermeur, Y., Lifchitz, A., & Vert, R. (2004). A kernel for protein secondary structure prediction. In *Kernel methods in computational biology* (pp. 193–206): MIT Press.

Herlocker, J.L., Konstan, J.A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on computer supported cooperative work* (pp. 241–250): ACM.

Houlsby, N., Hernandez-lobato, J.M., & Ghahramani, Z. (2014). Cold-start active learning with robust ordinal matrix factorization. In *Proceedings of the 31st international conference on machine learning (ICML-14)* (pp. 766–774).

Hu, G.N., Dai, X.Y., Song, Y., Huang, S.J., & Chen, J.J. (2015). A synthetic approach for recommendation: combining ratings, social relations, and reviews. In *Proceedings of the 24th international joint conference on artificial intelligence* (pp. 1756–1762): AAAI Press.

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of 8th IEEE international conference on data mining* (pp. 263–272): IEEE.

Huang, S. (2011). Designing utility-based recommender systems for e-commerce: evaluation of preference-elicitation methods. *Electronic Commerce Research and Applications*, *10*(4), 398–407.

Huang, Y.J., Xiang, E.W., & Pan, R. (2012). Constrained collective matrix factorization. In *Proceedings of the 6th ACM conference on recommender systems* (pp. 237–240): ACM.

Jamali, M., & Ester, M. (2011). A transitivity aware matrix factorization model for recommendation in social networks. In *Proceedings of IJCAI, the 22nd international joint conference on artificial intelligence*, (Vol. 11 pp. 2644–2649): Citeseer.

Karatzoglou, A., Amatriain, X., Baltrunas, L., & Oliver, N. (2010). Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM conference on recommender systems* (pp. 79–86): ACM.

Klašnja-Milićević, A., Ivanović, M., & Nanopoulos, A. (2015). Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, *44*(4), 571–604.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 426–434): ACM.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Lam, X.N., Vu, T., Le, T.D., & Duong, A.D. (2008). Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on ubiquitous information management and communication* (pp. 208–211): ACM.

Lee, D.D., & Seung, H.S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791.

Lee, D.D., & Seung, H.S. (2001). Algorithms for non-negative matrix factorization. In *Proceedings of advances in neural information processing systems* (pp. 556–562).

Liu, N.N., Meng, X., Liu, C., & Yang, Q. (2011). Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *Proceedings of the 5th ACM conference on recommender systems* (pp. 37–44): ACM.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook* (pp. 73–105): Springer.

Ma, H., Zhou, D., Liu, C., Lyu, M.R., & King, I. (2011). Recommender systems with social regularization. In *Proceedings of the 4th ACM international conference on web search and data mining* (pp. 287–296): ACM.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 165–172): ACM.

Melville, P., Mooney, R.J., & Nagarajan, R. (2002). *Content-boosted collaborative filtering for improved recommendations*: American Association for Artificial Intelligence.

Mnih, A., & Salakhutdinov, R. (2007). Probabilistic matrix factorization. In *Proceedings of advances in neural information processing systems*, (Vol. 2007 pp. 1257–1264).

Ortega, F., Bobadilla, J., & Hernando, A. (2014). Using hierarchical graph maps to explain collaborative filtering recommendations. *International Journal of Intelligent Systems*, *29*(5), 462–477.

Park, S.T., & Chu, W. (2009). Pairwise preference regression for cold-start recommendation. In *Proceedings of the ACM conference on recommender systems (RecSys)* (pp. 21–28): ACM.

Pessiot, J.F., Truong, T.V., Usunier, N., Amini, M.R., & Gallinari, P. (2006). Factorisation en matrices non-négatives pour le filtrage collaboratif. In *Proceedings of 3rd conference en recherche d'Information et applications* (pp. 315–326).

Rashid, A.M. (2007). *Mining influence in recommender systems*: PhD thesis, University of Minnesota.

Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., & Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on intelligent user interfaces* (pp. 127–134): ACM.

Rossetti, M. (2014). *Advancing recommender systems from the algorithm interface and methodological perspective*: PhD thesis, Università Degli Studi di Milano - BICOCCA.

Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on machine learning* (pp. 880–887).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J.T. (2000). Application of dimensionality reduction in recommender system, a case study. In *Proceedings of ACM WebKDD 2000 web mining for E-commerce workshop* (p. 12).

Saveski, M. (2013). *Cold start recommendations: a non-negative matrix factorization approach*: Master's thesis, Universidad Politecnica de Cataluna.

Schafer, J.B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291–324): Springer.

Seminario, C.E., & Wilson, D.C. (2014). Assessing impacts of a power user attack on a matrix factorization collaborative recommender system. In *Proceedings of the 27th international florida artificial intelligence research society conference* (pp. 81–86).

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook* (pp. 257–297): Springer.

Singh, A.P., & Gordon, G.J. (2008). Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 650–658): ACM.

Sinha, R., & Swearingen, K. (2002). The role of transparency in recommender systems. In *Proceedings of CHI'02 extended abstracts on human factors in computing systems* (pp. 830–831): ACM.

Smyth, B., & McClave, P. (2001). Similarity vs. diversity. In *Case-based reasoning research and development* (pp. 347–361): Springer.

Sun, M., Li, F., Lee, J., Zhou, K., Lebanon, G., & Zha, H. (2013). Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the 6th ACM international conference on web search and data mining* (pp. 445–454): ACM.

Takacs, G., Pilaszy, I., Nemeth, B., & Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, *10*, 623–656.

Verbert, K., Manouselis, N., Ochoa, X., & Wolpers, M. (2012). Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies*, *5*(4), 318–335.

Willemsen, M.C., Knijnenburg, B.P., Graus, M.P., Velter-Bremmers, L.C., & Fu, K. (2011). Using latent features diversification to reduce choice difficulty in recommendation lists. In *Proceedings of RecSys11 workshop on human decision making in recommender systems* (pp. 14–20).

Xie, Y., Chen, Z., Zhang, K., Jin, C., Cheng, Y., Agrawal, A., & Choudhary, A. (2013). Elver: recommending facebook pages in cold start situation without content features. In *Proceedings of 2013 IEEE international conference on big data* (pp. 475–479).

Xin, X., Lin, C.Y., Wei, X.C., & Huang, H.Y. (2015). When factorization meets heterogeneous latent topics: an interpretable cross-site recommendation framework. *Journal of Computer Science and Technology*, *30*(4), 917–932.

Yoo, J., & Choi, S. (2009). Weighted nonnegative matrix co-tri-factorization for collaborative prediction. *Advances in Machine Learning*, *5828*, 396–411.

Zanker, M., Fuchs, M., Hpken, W., Tuta, M., & Muller, N. (2008). *Evaluating recommender systems in tourism a case study from austria*: Springer.

Zhang, S., Wang, W., Ford, J., & Makedon, F. (2006). Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 6th SIAM conference on data mining*, (Vol. 6 pp. 548–552).

Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 83–92): ACM.

Zhao, D., Wang, J., Gao, A., & Yue, P. (2015). Learning to recommend with hidden factor models and social trust ensemble. In *Proceedings of international conference on computer science and intelligent communication* (pp. 87–91).

Zhou, K., Yang, S.H., & Zha, H. (2011). Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 315–324): ACM.

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic aspects in information and management* (pp. 337–348): Springer.

Ziegler, C.N., McNee, S.M., Konstan, J.A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on world wide web* (pp. 22–32): ACM.