Expert Systems with Applications 42 (2015) 6784–6794

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Improving matrix factorization recommendations for examples in cold start

Uroš Ocepek [a,*], Jože Rugelj [b], Zoran Bosnić [a]

[a] University of Ljubljana, Faculty of Computer and Information Science, Slovenia
[b] University of Ljubljana, Faculty of Education, Slovenia

ARTICLE INFO

Article history:
Available online 4 May 2015

Keywords:
Recommender systems
Cold start
Matrix factorization
Imputation
Missing values

ABSTRACT

Recommender systems suggest items of interest to users based on their preferences (i.e. previous ratings). If there are no ratings for a certain user or item, it is said that there is a problem of a cold start, which leads to unreliable recommendations. We propose a novel approach for alleviating the cold start problem by imputing missing values into the input matrix. Our approach combines local learning, attribute selection, and value aggregation into a single approach; it was evaluated on three datasets and using four matrix factorization algorithms. The results showed that the imputation of missing values significantly reduces the recommendation error. Two tested methods, denoted with 25-FR–ME-* and 10-FR–ME-*, significantly improved performance of all tested matrix factorization algorithms, without the requirement to use a different recommendation algorithm for the users in the cold start state.

1. Introduction

Recommender systems suggest items of interest (e.g. movies, jokes, learning materials, music, etc.) to users based on their preferences (Schein & Popescul, 2002). Nowadays, collaborative filtering is the most used technology in this field. Its aim is to find similar users that had already rated similar items and deduce recommendations for other users using that data (Eckhardt, 2012). There are two major strategies that are commonly applied in collaborative filtering: (1) *A memory-based approach* that operates on the entire database of ratings collected by the system; and (2) *a model-based approach*, which uses the database to build a model and then uses this model to make recommendations (Takács, Pilászy, Németh, & Tikk, 2009).

Methods for collaborative filtering make more accurate recommendations when they have enough ratings for each user and each item. Otherwise, it is said that the *cold start* (CS) problem occurs, which leads to making unreliable recommendations due to an initial lack of ratings (Bobadilla, Ortega, Hernando, & Bernal, 2012b). There are three types of cold start problems: (1) *new system* – when the system has just been used/created; (2) *new user* – when a new user has just joined the system; and (3) *new item* – when a new item has been recently introduced into the system (Bobadilla & Ortega et al., 2012b; Formoso, Fernãndez, Cacheda, & Carneiro, 2013; Park & Chu, 2009; Schein & Popescul, 2002). In this paper, we focus on the *new user* problem, which appears when a new user, for whom no preferences are known, enters the system. Since the user had not previously rated any items, the system is unable to make personalized recommendations; thus, the performance of the collaborative filtering suffers. This situation can have two forms, either when the user has no ratings (we will refer to this situation as the *absolute cold start*), or when the user has too few ratings (*partial cold start*) (Formoso et al., 2013).

Researchers mostly avoid tackling the absolute cold start in recommender systems (Ahn, 2008; Bobadilla & Ortega et al., 2012b; Formoso et al., 2013; Lika, Kolomvatsos, & Hadjiefthymiades, 2014; Rashid, Karypis, & Riedl, 2008; Said, Jain, & Albayrak, 2012). In our paper, we propose a novel approach for alleviating both, absolute and partial cold starts by imputing missing values into the input matrix, thereby improving the recommendation performance. Our approach combines attribute selection and local learning to optimize the recommendation process. The idea of our approach stems from the field of semi-supervised machine learning, where unlabeled examples are used to optimize the classification performance (Wang, Li, & Zhang, 2008).

The paper is structured as follows. In Section 2 we present related work. In Section 3 we introduce our approach for solving CS-problem by imputing missing values for CS-users. Section 4 presents the methodology of our research, the procedure of designing an artificial data, and shows characteristics of artificial and real

* Corresponding author.
E-mail addresses: uros.ocepek@gmail.com (U. Ocepek), joze.rugelj@pef.uni-lj.si (J. Rugelj), zoran.bosnic@fri.uni-lj.si (Z. Bosnić).

http://dx.doi.org/10.1016/j.eswa.2015.04.071
0957-4174/

dataset. In Section 4.2 we show and explain results of this research. Finally, in Section 5 we summarize the results and give directions for future work.

2. Related work

Matrix factorization (MF) is a family of frequently used techniques in collaborative filtering. The goal of the MF techniques is to approximate a recommendation matrix R (of size $m \times n$) as a product of two much smaller matrices: $R \approx U \cdot I$, where U is a $m \times k$ and I is a $k \times n$ matrix (Takács & Pilászy, 2008), as shown in Fig. 1.

The most frequently used variations of MF in recommender systems are: non-negative matrix factorization with stochastic gradient descent (NG) (Funk, 2011; Koren, Bell, & Volinsky, 2009), non-negative matrix factorization with alternating least squares (NS) (Bell & Koren, 2007; Koren et al., 2009; Takács & Pilászy, 2008; Zhou, Wilkinson, Schreiber, & Pan, 2008), and semi-non-negative matrix factorization with missing data (SN) (Ding, Li, & Jordan, 2010; Mo & Draper, 2012). These methods are usually used for recommending to users that are not in the cold start state – there is not sparse input matrix.

Cold start problem is similar to the problem of sparse data. Hastie et al. (1999) describe three different algorithms for imputing missing data: imputation using the SVD (SVDimpute), nearest-neighbor imputation (KNNimpute), and imputation using regression. Troyanskaya et al. (2001) use the same approaches (SVDimpute and KNNimpute) and add row average as a method for the estimation of missing values in gene microarray data. Experimental results show that KNNimpute performs better than SVDimpute and row average method and provides a fast, accurate and robust approach for estimating missing data.

Li (2014) presents three strategies to deal with the missing values (sparse input matrices): (1) removing samples or features with the missing values; (2) estimating missing values by some machine learning algorithm; and (3) using weighting method combining with machine learning models. The first strategy, removing samples or features with missing values, is not suitable for recommendation systems as the system cannot make recommendations to the excluded users that were in the cold start state. The second approach, estimating missing values by machine learning methods, was identified as promising and is related to our approach described in the next section. The third strategy is the 0–1 weighting method that is combined with different matrix factorization algorithms. In our study we used two matrix factorization algorithms that are based on this approach: non-negative matrix factorization with alternating least squares (Bell & Koren, 2007; Koren et al., 2009) and semi-non-negative matrix factorization with missing data (Ding et al., 2010; Mo & Draper, 2012).

Cold start problem is more specific problem than sparse data problem, because when new user has no or too fee ratings, it is impossible to provide accurate recommendations. Approaches for solving the new user cold start problem can be divided into two main groups: the first group uses dedicated algorithms for users in the cold start state; and the second group performs additional inquiries to gather more information about the users. The first group separates new users (i.e. users in the cold start state) from the remaining users and uses two different recommendation approaches for each of the groups. Although this method is useful, it is hard to decide if enough data about a particular user is given to use the first or the second recommendation algorithm. To address this, Schein and Popescul (2002) suggested general heuristics and metrics for deciding when to make cold start recommendations. Alternative approaches with the same aim include: cross-level association rule mining (Leung, Chan, & Chung, 2008), tied Boltzmann machines (Gunawardana & Meek, 2008), or constructing the probabilistic relationship between user and item attributes (Wang & Wang, 2014). The new user problem can also be solved by using different similarity measures and metrics of trust in order to optimize recommendations (Ahn, 2008; Bobadilla, Ortega, & Hernando, 2012; Eckhardt, 2012).

Rosli, You, Ha, Chung, and Jo (2014) combine collaborative filtering with similarity values from the movie's "Facebook pages", which is similar to performing an additional inquiry about the user. The approach combines values of users' similarities according to their ratings and similarity values obtained from their genre interests. Guo, Zhang, and Yorke-Smith (2015) perform additional inquiries by proposing a new information source called prior ratings, and design a user study to validate the conceptual model of prior ratings. One of major concerns is how the system should provide recommendations when there are no prior ratings. Patra, Launonen, Ollikainen, and Nandi (2015) propose a new measure denoted BCF (Bhattacharyya coefficient in collaborative filtering) for the neighborhood-based collaborative filtering, which takes into account all ratings made by pairs of users. Experiments showed that BCF-based collaborative filtering can provide highly reliable recommendations with only a few user/item ratings. Disadvantage of this approach is that the system needs some starting ratings in order to make calculations with the similarity measures – this approach is therefore not useful for the absolute cold start problem.

The second group of approaches aims at performing additional inquiries about the user. Rashid et al. (2008) use an approach to ask users to provide some initial ratings before performing the recommendation. There is a concern, however, which items to inquire about from the new user; if we inquire about an item that is popular in a community, that user will probably rate that item highly, too. Consequently, the system will be unable to recommend items that are of a personalized taste to that new user, but will recommend items that are preferred by the majority. As an alternative, the content-based filtering asks new users to participate in a questionnaire (e.g. during registration into the system) in order to build their initial profiles (Park & Chu, 2009). Afterwards, Lika et al. (2014) use general classification methods with demographic data to identify other users with similar behaviors. Note, however, that both, rating numerous initial items and building a psychological/demographic profiles are invasive and can be time-consuming and unwelcome tasks.

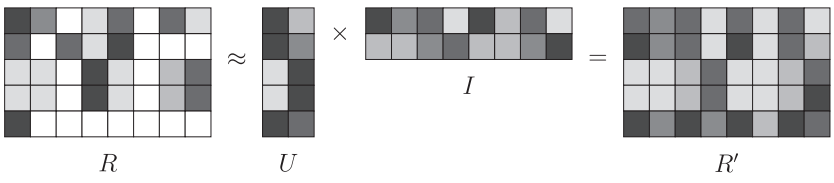

Fig. 1. Illustration of matrix factorization – the result of MF are matrices U and I, where their dot product R' is an approximation of the original matrix R. Different cell shadings denote different values of user preferences. The figure illustrates how matrix R' can be used to predict user preferences (i.e. recommendations) for items that were not given in the input matrix R (white fields).

Table 1
Comparison of the state-of-the-art approaches for solving the cold start problem with our proposed approach.

Reference	Absolute CS	Partial CS	No additional inquiring	Within the recommender system
Chen et al. (2013)	+	+		+
Formoso et al. (2013)	+	+		+
Lika et al. (2014)	+	+		+
Rosli et al. (2014)	+	+		+
Son (2014)	+	+		
Wang and Wang (2014)		+	+	
Guo et al. (2015)	+	+		+
Patra et al. (2015)		+	+	+
Pereira and Hruschka (2015)	+	+		
Zhang et al. (2015)		+	+	
Our approach	+	+	+	+

Pereira and Hruschka (2015) propose a hybrid recommendation system based on simultaneous co-clustering and learning (SCOAL). The system combines the SCOAL module, which performs clustering and assignment of users to appropriate clusters, with a special module for cold start recommendations. The advantage of the system is that it can provide recommendations when no ratings are available for the new user. Limitations of the proposed approach, however, are that additional inquiring is needed to assign a new user to the appropriate cluster.

Zhang, Cheng, Qiu, Zhu, and Lu (2015) present a DualDS framework that integrates discriminative selection process with dual regularization. The approach is based on exploring correlations between users and items. With the designed dual regularizer, the two selection tasks were integrated into one unified model. This approach was characterized as promising for resolving only partial cold start situations, but not also the absolute cold start situations.

Table 1 displays a comparison of the state-of-the-art approaches for resolving cold start problem with our proposed novel approach. Although there are many suitable approaches for resolving the absolute cold start problem, we tried to avoid performing additional inquiring about the user and thus overcome the limitations of methods of Wang and Wang (2014), Patra et al. (2015) and Zhang et al. (2015). At the same time, our goal was to design an approach that can use the same recommendation algorithm for users in the cold start state and for the other users.

3. Imputing missing values for CS users

We propose an alternative to the previous approaches: by selectively imputing missing ratings to items we try to improve the matrix factorization performance without separately processing cold start and complete examples. By using this approach, we try to "repair" the input matrix in order to reduce the predictive recommendation error. The idea for our strategy is also related to the semi-supervised learning that employs unlabeled data together with the labeled data to build more accurate classifiers (Zhu, 2005). In a similar way, we prepare unlabeled data (i.e. examples in the cold start state) to be used with other labeled examples in order to improve the learning process (Chapelle, Schölkopf, & Zien, 2006).

Our approach deduces missing ratings from other users and aggregates them. This concept is related to the work of Chen, Wan, Chung, and Sun (2013), who showed that the integration of a user model with trust and distrust networks to identify trustworthy users is efficient in the cold start problem. Expanding user profiles with items that are similar to items that have already been rated has also been shown to be beneficial (Formoso et al., 2013).

In this work we assume that the given input matrix contains user ratings, which are discrete values (integers). Each input matrix is structured as follows:

- rows (examples) represent *users*, each row contains all ratings from a particular user,
- columns (attributes) represent *items*, each column contains all ratings for a particular item.

We propose a procedure that takes a single user in the cold start state (we denote this user with *CS user*; note that the cold start can be either absolute or partial) and proceeds with the following four steps (illustrated in Fig. 2):

1. find certain % of users that are similar to a given CS user,
2. select relevant attributes (items) for the imputation process,
3. aggregate ratings from similar users and impute the result into ratings for the CS user, and
4. use a particular matrix factorization method on obtained modified matrix.

Each individual step is explained in the following subsections in greater detail.

3.1. Finding other similar users

In the first step, we find the most similar users to a given CS user with missing ratings. The selected ratings of these users are to be later aggregated and imputed in place of the missing ratings for the current CS user. To select the most similar users, we use the Euclidean distance to calculate similarity between the given CS user and other training examples (non-CS users). Afterwards, we rank the examples by their descending similarity and keep only a certain ratio of the most similar users for further processing. In our experiments, we experimented with five different settings for this parameter:

1%, 5%, 10%, 25% and 50%. Note that the above procedure is possible only when the user is in the partial cold start (few ratings are given), which makes it possible to compute Euclidean distances; the absolute cold start situations (no starting ratings are given) require additional attention. In these cases we translate an absolute cold start into a partial cold start by imputing a single attribute value, and afterwards proceed as initially described above. The details are given in Section 4.1.

3.2. Choosing relevant attributes

In the second step, we choose the most relevant and reliable attributes (recommendation items) that will be used to compute values of the missing attributes for the CS user. We experiment with four different approaches, the first two are based on simple descriptive statistics, while the second two are based on attribute evaluation metrics:

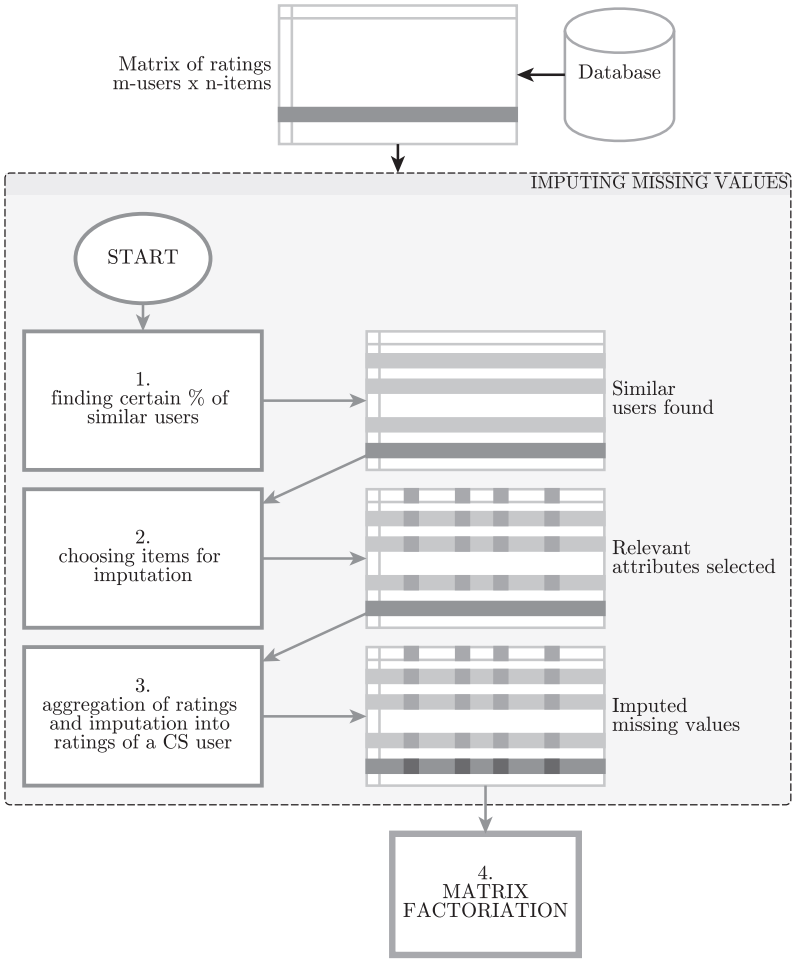

Fig. 2. Steps of the proposed imputation methodology approach.

- **SD** (standard deviation): we select only those attributes that have a standard deviation lower than a given threshold. The rationale behind this approach is that the missing values of an attribute can be more reliably (i.e. with lower error) replaced for attributes with a lower deviation. Among all attributes, we choose 10% of the best attributes that have the lowest SD.
- **FR** (frequent values): we select attributes that have the frequency of most frequent value higher than a given threshold. In our experiments, we select 10% of the attributes that have the highest frequency of the most frequent value.

The remaining two approaches select attributes that can be most reliably used as dependent (i.e. target) variables, when they are modeled as a function of the remaining attributes. Both iterate over all available attributes; in each iteration they treat the current attribute as the target attribute and all others as input attributes and evaluate attributes using:

- **IG** (information gain (Kent, 1983)), and
- **RR** (RReliefF (Robnik-Šikonja & Kononenko, 2003)).

For a current attribute in the iteration, we store the mean of the calculated attribute quality measures. When the last iteration finishes, we choose 10% of attributes that have the highest mean estimates, and are therefore expected to be the most accurately modeled as functions of other attributes.

The output of the current step is a set of attributes, which we will impute in the next step for each user in a cold start.

3.3. Imputing missing values

In the third step we propose four different strategies for imputing missing values for the user in a cold start state. The input for this step is the set of selected relevant attributes from the previous step and a single example with missing values. The proposed imputation strategies are:

- **ME** (mean value): imputation of attribute's mean value.
- **MF** (most frequent): imputation of the attribute's most frequent value.
- **LR** (linear regression) and **RT** (regression tree): two supervised learning models (linear regression and regression tree) that predict each of the attributes, which were selected for imputation, using all available input attributes. The models are built on known values of the chosen nearest examples (not in the cold start state) and used to impute missing attribute values for examples in a cold start state.

3.4. Recommendation using matrix factorization

In our study we used different matrix factorization algorithms and compared their recommendation accuracy between the

original matrices (that include users in the cold start state) and the matrices with imputed values. The used matrix factorization algorithms were:

- **NG** (Non-negative matrix factorization with stochastic gradient descent): the algorithm iterates through all existing ratings in the input matrix. For each existing rating, the system calculates the prediction and the associated prediction error, which is an input for modifying parameters by a magnitude that is proportional to the learning rate and the opposite direction of the gradient. The advantage of this algorithm is its appropriateness for use with sparse matrices. On the other hand, this approach is time-consuming when used with large matrices (Funk, 2011; Koren et al., 2009; Takács & Pilászy, 2008).
- **NS** (Non-negative matrix factorization with alternating least squares): the algorithm iterates between phases of fixing one of the two factor matrices, computing the other, and vice versa until convergence (Bell & Koren, 2007; Koren et al., 2009). This approach is not efficient on sparse matrices, as the algorithm factorizes entire matrices as the whole and does not iterate only through a given rating, such as the gradient descent does (Takács & Pilászy, 2008; Zhou et al., 2008). For this reason, an extension of NS uses a weighted matrix ('0' for missing data and '1' for valid data), what is proposed by Li (2014).
- **SN** (semi-non-negative matrix factorization with missing data): to handle missing data, we used an extension of semi-non-negative matrix factorization (SNMF) proposed by Mo and Draper (2012). SNMF iteratively updates factor matrices for users and items using a separation of positive and negative parts of the items matrix (Ding et al., 2010). According to regular SNMF, the SN algorithm takes into account a matrix with weights ('0' for missing data and '1' for valid data), what is also suggested by Li (2014). This algorithm is a good alternative to NG because it is fast and can handle large matrices with large numbers of ratings (Ding et al., 2010; Mo & Draper, 2012).
- **DF** (matrix factorization by data fusion (Žitnik & Zupan, 2013)): this approach can fuse heterogeneous data sources that are presented as matrices. First, the algorithm organizes all source matrices into a block representation. Then, it simultaneously factorizes all input matrices in the way that low-rank matrix factors are shared between the tri-factor decomposition of input matrices. The resulting system contains factors $S_{i,j}$ that are specific to each input matrix and factors G_i that are specific to each object type, such that $R_{i,j}$ is approximated as $R_{i,j} \approx G_i \cdot S_{i,j} \cdot G_j^T$.
  In this work, we adapted the former (**DF**) approach for matrices that include users in a cold start state. First, we rearrange the order of the examples (rows) in the matrix R to form four sub-matrices ($R_{1,2}, R_{1,3}, R_{4,2}$ and $R_{4,3}$) that correspond to four (different) data sources in the original paper of Žitnik and Zupan (2013). As shown in Fig. 3, sub-matrices $R_{1,2}$ and $R_{1,3}$ contain ratings of users that are not in a cold start state and matrices $R_{4,2}$ and $R_{4,3}$ contain ratings of users that are in a cold start state. The items (columns) of the latter are rearranged in such a way that the matrix $R_{4,3}$ is empty, while $R_{4,2}$ contains a few ratings. In order to get matrix $R_{4,3}$, we use factors (of input sub-matrices) (Fig. 3) of matrix tri-factorization and then calculate matrix $R_{4,3}$ (its approximation) by using the formula $R'_{4,3} = G_4 \cdot S_{4,2} \cdot S_{1,2}^T \cdot S_{1,3} \cdot G_3^T$.

4. Experimental evaluation

The goal of our experiments was to evaluate if the proposed imputation framework decreases the recommendation error, and to compare different choices of possible parameters. We performed a series of experiments by exhaustively selecting all possible parameters for steps that are depicted in Fig. 2.

To summarize, there were altogether 320 combinations of all parameters for the experiments:

- 5 percentages of considered nearest neighbors: 1%, 10%, 25%, 50% and 100%,
- 4 methods for selecting relevant attributes: SD (standard deviation), FR (frequent values), IG (information gain) and RR (RReliefF), and
- 4 imputation methods: ME (mean), MF (most frequent), LR (linear regression) and RT (regression tree),
- 4 matrix factorization algorithms: NG, NS, SN, and DF.

To briefly denote a particular combination of the former parameters, in the following we use the notation XX–YY–ZZ–WW, where XX denotes the percentage of considered similar users (1%, 10%, 25%, 50%, 100%), YY denotes attribute selection approach (SR, FR, IG, RR), ZZ denotes aggregation strategy (ME, MF, LR, RT), and WW denotes the matrix factorization algorithm (NG, NS, SN or DF).

Experimental framework. We tested the methodology on one synthetic and two real datasets. We evaluated the matrix factorization performance by comparing root relative squared errors (RRSE) of the raw matrix factorization method (no imputation) and matrix factorization method that uses our imputation methodology:

$$RRSE(R) = \sqrt{\frac{\sum_{(i,j)\in\tau}(r_{i,j} - \hat{r}_{i,j})^2}{\sum_{(i,j)\in\tau}(r_{i,j} - \bar{r})^2}},$$

where τ represents the set of test ratings, $r_{i,j}$ is a real rating value for an item j given by user i, and $\hat{r}_{i,j}$ is the predicted value by a matrix

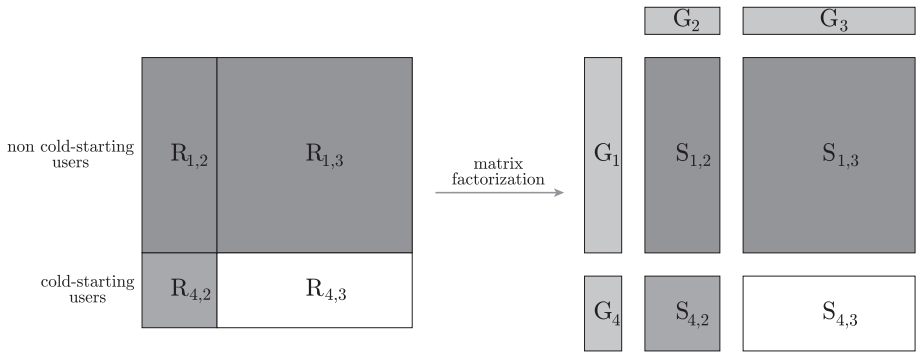

Fig. 3. Left: the input matrix, rearranged to form four sub-matrices and used with the data fusion matrix factorization. Right: the result of matrix tri-factorization based on data fusion.

factorization. The RRSE estimate was calculated by dividing examples into 10 subsets and averaging the RRSE values of the 10 runs.

To statistically compare all our combinations, we used a Friedman test with a post hoc Nemenyi test (Demšar, 2006). The Friedman test is a non-parametric test that compares the average ranks of methods to reject the null hypothesis that all methods are equivalent; the post hoc Nemenyi test evaluates if the performance of two methods is significantly different by at least the critical difference (CD).

Datasets. For the purpose of experimenting with the proposed approach, we designed a synthetic recommendation dataset named *ArtificialDS*. The design goal for this dataset was to contain some items of which ratings correlate positively, and some items of which ratings correlate negatively; additionally, some noise was also added to the data. The resulting dataset contains 15 integer attributes that correspond to ratings of users. The first three attributes (A1 – A3) are generated randomly, and the remaining 12 attributes (A4 – A15) are the following functions of the first three attributes:

- A_4, A_7, A_{10}, A_{13} equal to $A_1 + rand(\{-1, 0, 1\})$,
- A_5, A_8, A_{11}, A_{14} equal to $6 - A_2 + rand(\{-1, 0, 1\})$,
- A_6, A_9, A_{12}, A_{15} equal to $A_{i\in\{6,9,12,15\}} = \frac{A_{i-1}+A_{i-2}}{2} + rand(\{-1, 0, 1\})$,

where the function *rand* randomly chooses one of the elements in the given set. We also added 10% of examples with all attributes' values randomly generated (noise). Table 2 displays the characteristics of the generated dataset.

In addition to the synthetic data set, we also evaluated our approach using two real datasets, which are presented in Section 4.4.

4.1. Handling users in an absolute cold start

In each experiment we simulated different cold start scenarios for test users by removing all attributes, except a few. With this approach we artificially simulated an absolute cold start (denoted with CS0), and five different levels of a partial cold start – where only 1 to 5 ratings are available (denoted with CS1, CS2, CS3, CS4 and CS5).

Just to confirm our assumption that the number of given initial ratings increases the matrix factorization performance, we initially evaluated the average performances of different levels of cold start problems (levels denoted with CS0 to CS5) using our data. The results in Fig. 4 confirm this assumption and show that one given rating already decreases the RRSE of the absolute cold start problem (CS0).

Table 2
The basic properties of tested synthetic dataset.

Dataset	Users	Items	Ratings	Sparsity	Range of ratings
ArtificialDS	200	15	3000	0	1–5

As mentioned in Section 3.1, we transform an absolute cold start situation (CS0) into a relative cold start situation by imputing one of the attributes. Note that in order to do this, we are only limited to those attribute selection and imputation approaches from Sections 3.2 and 3.3 that work with single attributes (other methods cannot be used as they model dependencies between more attributes). To select the most relevant attribute, we therefore use either standard deviation (SD) or the most frequent values (FR) approaches; then we impute a missing value of a chosen attribute by using either its mean value (ME) or its most frequent value (MF).

Fig. 5 displays performances of four matrix factorization methods when used on users in an absolute cold start state, compared to performances when an absolute cold start was converted into a relative cold start with one given rating, as described above. The comparison shows that the procedure significantly decreases recommendation error in two of the matrix factorization methods (NG and SN), while in the other two the error slightly increases. Although the Friedman test reports no statistical significance in this overall change, we consider this initial result as a good starting point for further improvement, as described in the following sections.

4.2. Performance of different parameters in imputation steps

In the main part of our evaluation we observed how the choice of parameters in each of the imputation steps (shown in Fig. 2) influences the matrix factorization performance. After computing the matrix factorization performance for all 320 different combinations of parameters, we averaged the results to observe the average influence of individual parameters. Additionally, we used the Friedman test to determine if the performance differences are statistically significant. If the test indicated that there is a statistical difference between the observed approaches with a significance level $p \leqslant 0.05$, we used the post hoc Nemenyi test to further compare differences between individual parameter selections. These differences are shown using critical difference (CD) charts (Demšar, 2006).

The evaluation, how different percentages of nearest neighbors (the first step of the method) influence performance on the average, is shown in Fig. 6; and its statistical analysis is shown in Fig. 7. The results in these two figures show that, on the average, the imputation methods using any of the five percentage parameters (within the first step) are ranked higher than the best performi-

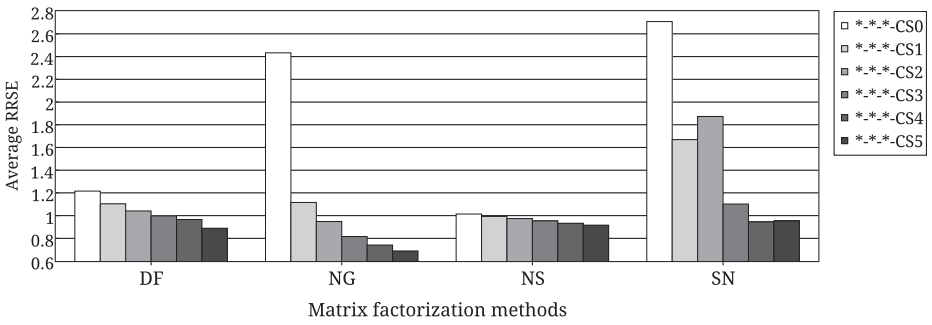

Fig. 4. Performance evaluation of raw matrix factorization methods (no imputation is used) with *different levels of the cold start* (denoted with CS0 to CS5, depending on how many initial ratings are given, from 0 to 5, respectively). The average performance was computed by averaging values of the remaining parameters that are denoted with *. The results show that the greater number of the given initial ratings improves the matrix factorization performance.

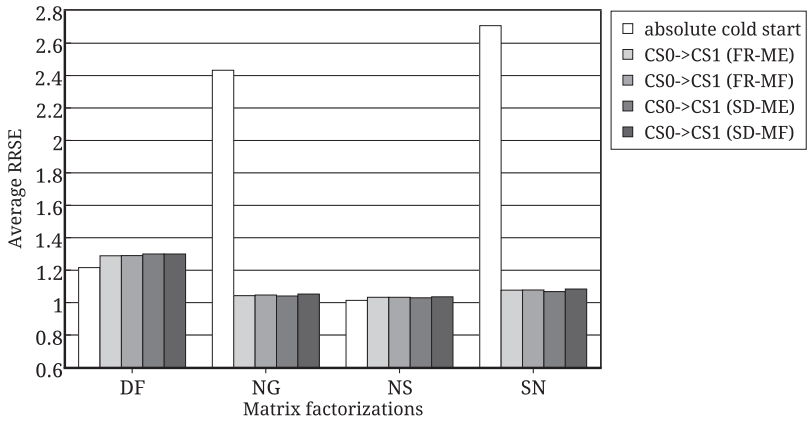

Fig. 5. Comparison of matrix factorization methods when used on users in an absolute cold start and users that have been transformed to a partial cold start with one given rating. Four transformation methods are compared, depending on the approach for selection of the best attribute to be imputed (standard deviation – SD or highest frequency – FR), and the imputation method (mean – ME or most frequent value – MF).

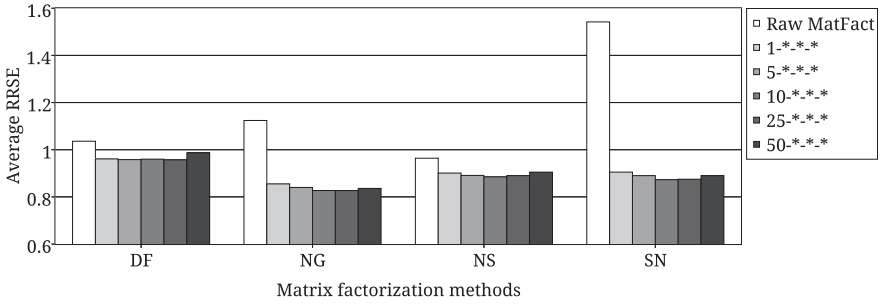

Fig. 6. Performance evaluation of raw matrix factorization methods (no imputation is used) with different imputation approaches. The results show the average performance according to various values of the *first parameter – percentage of the considered nearest neighbors*. The average was computed by averaging values of the remaining parameters that are denoted with *.

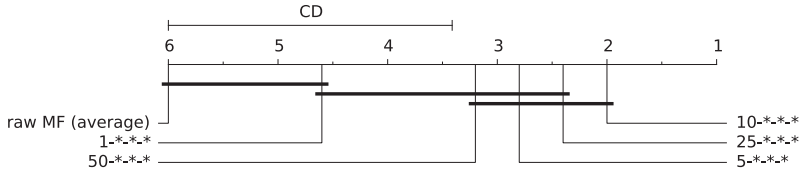

Fig. 7. Statistical evaluation of average performances of various matrix factorization methods. The methods are averaged among parameters that are denoted with * (e.g. 5-*-*-* represents all methods that consider 5% of similar users). The figure compares the best performing matrix factorization method with averaged results, achieved with *different percentages of the considered nearest neighbors*. Groups of methods that are not significantly different (at $p \leqslant 0.05$) are connected with a horizontal line. The figure shows that three parameter settings (5%, 10%, 25% and 50%) yield improvement in recommendation accuracy compared with raw matrix factorization.

ng raw matrix factorization method. The second figure also shows that all parameters except 1% yield statistically significant improvements.

Likewise, evaluation of the average performance of different attribute selection approaches (the second step of the method) is shown in Fig. 8; its statistical analysis is shown in Fig. 9. These averaged results also confirm that the use of any attribute selection method contributes to higher matrix factorization performance. The performances of attribute selection using frequency (FR) and RReliefF (RR) are also significantly better than the raw matrix factorization.

Further, evaluation of different aggregation strategies (the third step of the method) is shown in Fig. 10; its statistical analysis is shown in Fig. 11. Again, these two figures also confirm the benefits of imputing missing values, and indicate that aggregation by mean value (ME) or using regression trees (RT) yield significant improvements in performance.

Based on these results, we chose several method combinations, which are expected to work well, and will compare them with other strategies on real domains in Section 4.4. These methods are combinations of parameter values that achieved the best averaged results, such as: 10-FR–ME-*, 25-FR–ME-*, 10-RR–RT-*, and 25-RR–RT-*.

4.3. Performance ranking of individual experiments

In the next step, we compare the ranks of all 320 different combinations of parameters in each of the imputation steps (shown in Fig. 2). We also include four raw matrix factorization methods in the comparison, to observe the relative performance, as shown in Fig. 12. According to the results, some combinations are much better performing than the raw matrix factorization methods.

Among all approaches, shown in Fig. 12, we chose the best performing method (50-RR–LR-*) for further evaluation on real

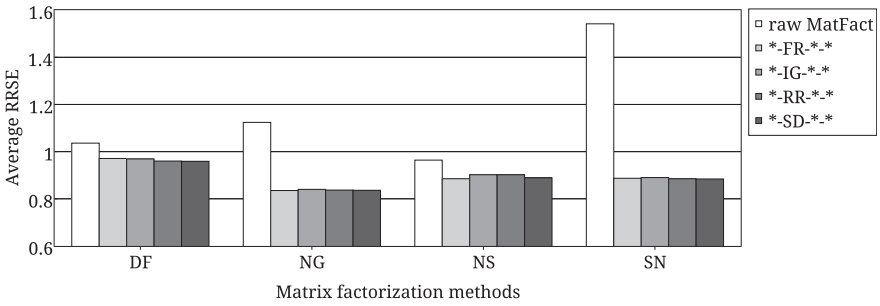

Fig. 8. Performance evaluation of raw matrix factorization methods (no imputation is used) with different imputation approaches. The results show the average performance according to various values of the *second parameter – the attribute selection approach*. The average was computed by averaging values of remaining parameters that are denoted with *.

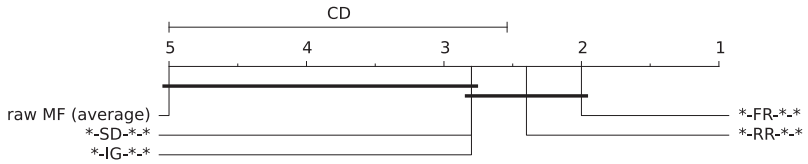

Fig. 9. Statistical evaluation of average performances of various matrix factorization methods. The methods are averaged among parameters that are denoted with * (e.g. *-SD-*-* represents all methods that select an attribute according to minimal standard deviation). The figure compares the best performing matrix factorization method with the averaged results, achieved with *different approaches for attribute selection*. Groups of methods that are not significantly different (at $p \leqslant 0.05$) are connected with a horizontal line. The figure shows that two approaches (FR and RR) yield improvements in recommendation accuracy compared with the raw matrix factorization.

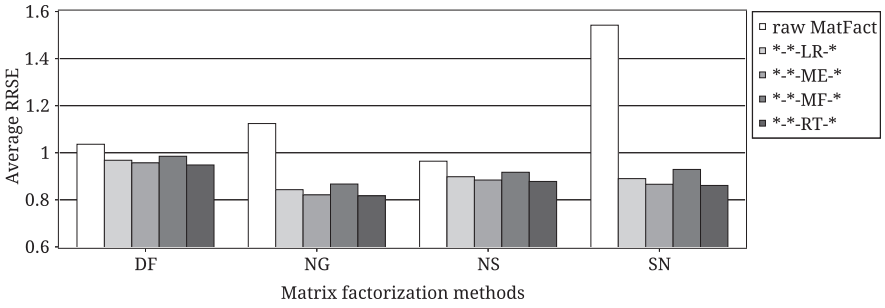

Fig. 10. Performance evaluation of raw matrix factorization methods (no imputation is used) with different imputation approaches. The results show the average performance according to various values of the *third parameter – value aggregation strategy*. The average was computed by averaging values of remaining parameters that are denoted with *.

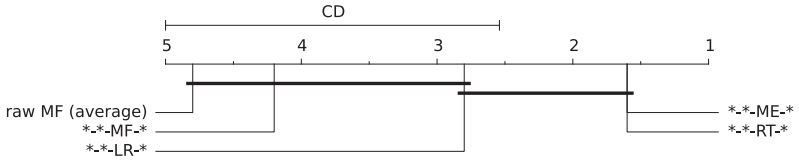

Fig. 11. Statistical evaluation of average performances of various matrix factorization methods. The methods are averaged among parameters that are denoted with * (e.g. *-*-ME-* represents all methods that impute the mean value). The figure compares the best performing matrix factorization method with averaged results, achieved with *different value aggregation strategies*. Groups of methods that are not significantly different (at $p \leqslant 0.05$) are connected with a horizontal line. The figure shows that two approaches (ME and RT) yield improvements in recommendation accuracy compared to the raw matrix factorization.

domains. Together with selected methods from Section 4.2 (10-FR–ME-*, 25-FR–ME-*, 10-RR–RT-*, and 25-RR–RT-*) we performed their evaluation on two real domains, as follows in the next subsection.

4.4. Evaluation on two real domains

To confirm our findings, we decided to verify how different imputation strategies work on two independent real datasets. We selected methods with parameter values that performed best on the average (Section 4.2) and the best-ranked method on the synthetic dataset (Section 4.3), as follows: 50-RR–LR-*, 10-FR–ME-*, 25-FR–ME-*, 10-RR–RT-*, and 25-RR–RT-*. We experimented with two real datasets: the Jester dataset (Goldberg, Roeder, Gupta, & Perkins, 2001), which contains ratings of 100 jokes, and our own dataset with students' preferences about multimedia learning materials (Ocepek, Bosnić, Nančovska Šerbec, & Rugelj, 2013). Table 3 displays their properties.

The experiments on real domains follow the same experimental protocol as experiments on our synthetic domain. To enable recommendation performance of users in a cold start we simulated cold start scenarios by removing all attributes except 1 to 5

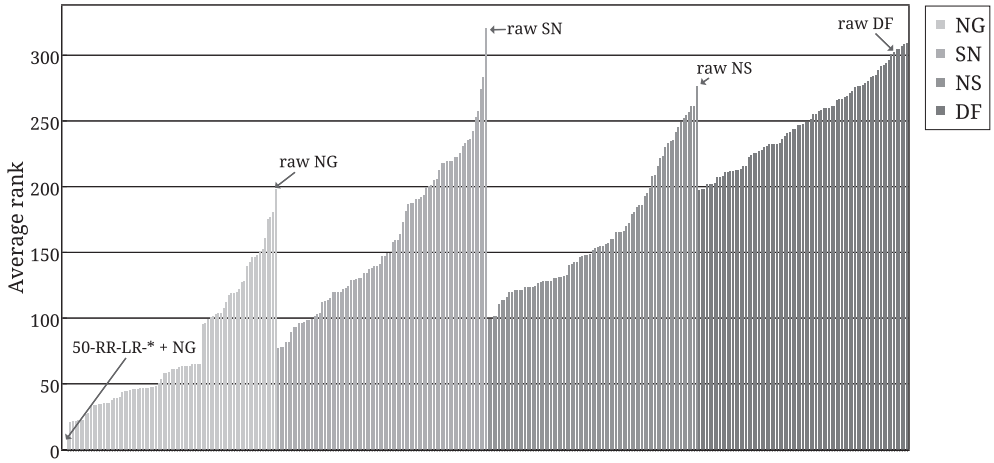

Fig. 12. Comparison of all 320 combinations of imputation methods with raw matrix factorization methods. The results are grouped by the particular matrix factorization method (NG, SN, NS, DF).

Table 3
The basic properties of two real datasets: Jester and PEFbase.

Dataset	Users	Items	Ratings	Sparsity	Range of ratings
Jester	300	100	30,000	0	1–5
PEFbase	286	35	10,010	0	1–4

(achieving scenarios denoted with CS1, CS2, CS3, CS4, and CS5). To confirm our assumption that the number of given initial ratings increases the matrix factorization performance, we started by evaluating average performances of different levels of cold start problems. The results in Fig. 13 confirm our previous results, and show that one given rating already decreases the RRSE of the absolute cold start problem (CS0).

Fig. 14 finally shows the comparison of the chosen imputation approaches (applied with all four matrix factorization algorithms) with raw matrix factorization approaches. The figure shows that the selected approaches are very applicable on real domains, as well. Namely, the figure shows that the raw matrix factorization approaches mostly benefit from using the imputation approaches; the exceptions are DF on the Jester set and NS on the PEFbase, where some of the approaches increase recommendation accuracy, while the other decrease it.

Fig. 15 also shows the statistical analysis of the former approaches, where their performance on both real data sets is considered. The evaluation shows that two methods (25-FR–ME-* and 10-FR–ME-*) perform statistically better than the raw matrix factorization algorithms. The other three methods (50-RR–LR-*, 25-RR–RT-* and 10-RR–RT-*) still reduce prediction error, but not significantly.

Our approach provides recommendations to new users without any additional inquiry and is suitable for recommender systems that have no prior data about the user. It is applicable to an arbitrary recommender system that is based on matrix factorization and allows recommending without initially separating CS-users from the other users. At the same time, the approach is also suitable for solving a new item problem, which is similar to the new user problem. A disadvantage of this approach includes increased computational time, which is especially evident if we use the non-negative matrix factorization with stochastic gradient descent. Also note, that due to the combinatorial nature of our experiment, we have limited our study only to the basic sets of parameters for each step; in our future work we shall deeper analyze the impact of the chosen parameters on the resulting recommendation performance.

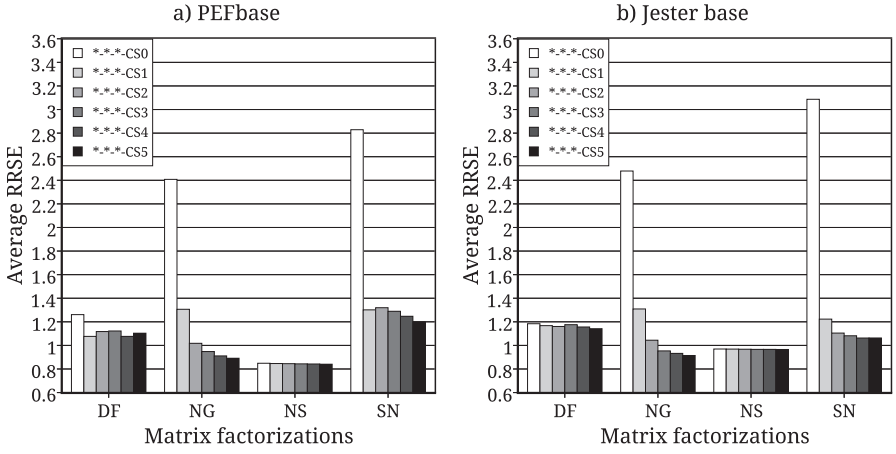

Fig. 13. Performance evaluation (on two real datasets) of raw matrix factorization methods (no imputation is used) with different levels of the cold start. The average performance was computed by averaging values of the remaining parameters that are denoted with *. The results support the previous results that the greater number of given initial ratings improves the matrix factorization performance.

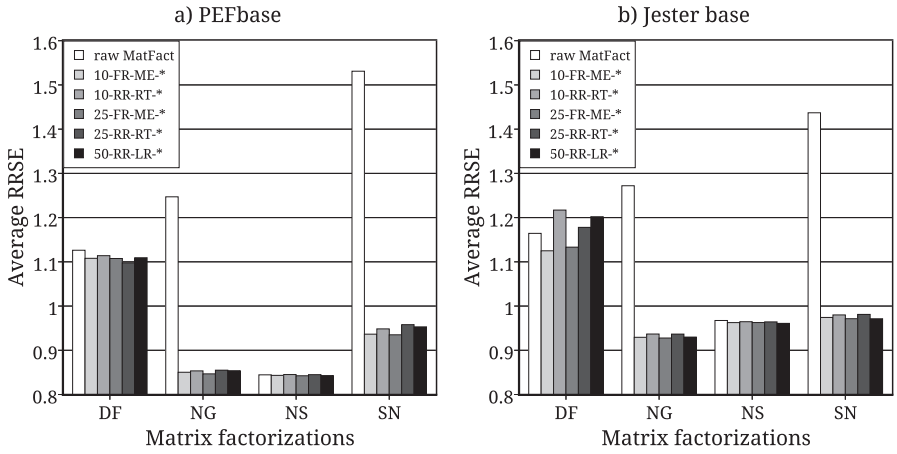

Fig. 14. Comparison of raw matrix factorizations and matrix factorizations using 50-RR–LR-* combination for aggregating missing value (on two real datasets).

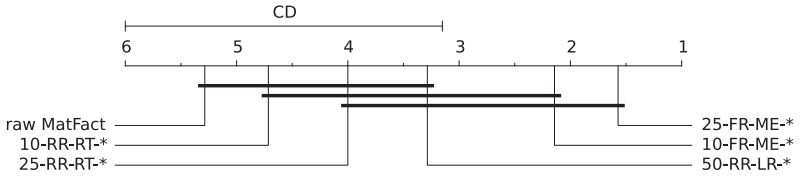

Fig. 15. Comparison the best three imputation approaches with four raw matrix factorization approaches. Groups of methods that are not significantly different (at $p \leqslant 0.05$) are connected with a horizontal line. The results show that the best imputation approach performs significantly better than all four raw matrix factorization approaches.

5. Conclusion

In this paper, we proposed a framework for the imputation of missing values into the ratings matrix. Our framework consists of three steps that determine how to select the nearest neighbors, how to select relevant attributes, and how to aggregate attribute values. We have evaluated the proposed framework on one synthetic and two real datasets, using four different matrix factorization algorithms. The results showed that the imputation of missing values can significantly reduce prediction error. Two methods, 25-FR–ME-* and 10-FR–ME-*, that were combined by using parameter values that performed best on the average, have performed statistically better than the raw matrix factorization algorithms on real domains, as well.

Our novel approach combines local learning, attribute selection and value aggregation into a single approach. The motivation for such an approach stems from the field of semi-supervised machine learning, where unlabeled examples are used to optimize classification performance. With this study, we pointed to a promising research field of selective imputing missing values.

Our future work in this field will be focused on exploring alternative approaches for selection of similar users, attribute selection, and imputation of missing values. In the first step, where we find the most similar users to a given CS user, we shall explore different similarity measures (cosine distance, PIP etc.) in addition to the used Euclidean distance. In the second step, where we choose the most relevant and reliable attributes (recommendation items), we shall study the impact of different attribute selection approaches on the algorithm's performance. In the third step, where we impute missing values for users in the cold start state, we shall experiment with other supervised learning models (e.g. support vector machines, random forest etc.) and association rules. Our future challenges also include adapting other matrix factorization algorithms for recommender systems.

As an application of our work we plan to implement an e-learning recommender system for recommending learning objects based on students' performance and learning characteristics; such systems strongly suffer from the cold start problem due to numerous, continuously incoming new students. We hope that our results will positively contribute to the development of the recommender systems field.

References

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, 178*, 37–51.

Bell, R. M., & Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Seventh IEEE international conference on data mining, ICDM 2007* (pp. 43–52). IEEE.

Bobadilla, J., Ortega, F., & Hernando, A. (2012). A collaborative filtering similarity measure based on singularities. *Information Processing & Management, 48*, 204–217.

Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012b). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems, 26*, 225–238.

Chapelle, O., Schölkopf, B., Zien, A., et al. (2006). *Semi-supervised learning* (Vol. 2). Cambridge: MIT press.

Chen, C. C., Wan, Y. H., Chung, M. C., & Sun, Y. C. (2013). An effective recommendation method for cold start new users using trust and distrust networks. *Information Sciences, 224*, 19–36.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1–30.

Ding, C., Li, T., & Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*, 45–55.

Eckhardt, A. (2012). Similarity of users (content-based) preference models for collaborative filtering in few ratings scenario. *Expert Systems with Applications, 39*, 11511–11516.

Formoso, V., Fernãndez, D., Cacheda, F., & Carneiro, V. (2013). Using profile expansion techniques to alleviate the new user problem. *Information Processing & Management, 49*, 659–672.

Funk, S., (2011). Netflix update: Try this at home, 2006. url: <http://sifter.org/~simon/journal/20061211.html>.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4*, 133–151.

Gunawardana, A., & Meek, C. (2008). Tied boltzmann machines for cold start recommendations. In *Proceedings of the 2008 ACM conference on recommender systems* (p. 19).

Guo, G., Zhang, J., & Yorke-Smith, N. (2015). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems, 74*, 14–27.

Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., & Botstein, D. (1999). Imputing missing data for gene expression arrays.

Kent, J. T. (1983). Information gain and a general measure of correlation. *Biometrika, 70*, 163–173.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, 42–49.

Leung, C. W., Chan, S. C., & Chung, F. (2008). An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems, 21*, 515–529.

Li, Y., (2014). Sparse machine learning models in bioinformatics.

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications, 41*, 2065–2073.

Mo, Q., & Draper, B. A. (2012). Semi-nonnegative matrix factorization for motion segmentation with missing data. In *Computer vision–ECCV 2012* (pp. 402–415). Springer.

Ocepek, U., Bosnić, Z., Nančovska Šerbec, I., & Rugelj, J. (2013). Exploring the relation between learning style models and preferred multimedia types. *Computers & Education, 69*, 343–355.

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

Patra, B. K., Launonen, R., Ollikainen, V., & Nandi, S. (2015). A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data. *Knowledge-Based Systems*.

Pereira, A. L. V., & Hruschka, E. R. (2015). Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems*.

Rashid, A. M., Karypis, G., & Riedl, J. (2008). Learning preferences of new users in recommender systems: An information theoretic approach. *ACM SIGKDD Explorations Newsletter, 10*, 90–100.

Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning, 53*, 23–69.

Rosli, A., You, T., Ha, I., Chung, K., & Jo, G. (2014). Alleviating the cold-start problem by incorporating movies facebook pages. *Cluster Computing*, 1–11.

Said, A., Jain, B. J., & Albayrak, S. (2012). Analyzing weighting schemes in collaborative filtering: Cold start, post cold start and power users. In *Proceedings of the 27th annual ACM symposium on applied computing* (pp. 2035–2040). ACM.

Schein, A., & Popescul, A. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 253–260).

Son, L. H. (2014). Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*.

Takács, G., & Pilászy, I. (2008). Investigation of various matrix factorization methods for large recommender systems. *Data mining workshops, ICDMW'08* (Vol. 1). IEEE International.

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

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., et al. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics, 17*, 520–525.

Wang, F., Li, T., & Zhang, C. (2008). Semi-supervised clustering via matrix factorization. In *SDM* (pp. 1–12). SIAM.

Wang, G., & Wang, Y. (2014). Probabilistic attribute mapping for cold-start recommendation. In *Foundations and applications of intelligent systems*. In F. Sun, T. Li, & H. Li (Eds.). *Advances in intelligent systems and computing* (Vol. 213, pp. 421–431). Berlin, Heidelberg: Springer.

Zhang, X., Cheng, J., Qiu, S., Zhu, G., & Lu, H. (2015). Dualds: A dual discriminative rating elicitation framework for cold start recommendation. *Knowledge-Based Systems, 73*, 161–172.

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.

Zhu, X. (2005). Semi-supervised learning literature survey. Technical Report 1530 Computer Sciences, University of Wisconsin-Madison.

Žitnik, M., & Zupan, B. (2013). Data fusion by matrix factorization. Available at: arXiv preprint arXiv:1307.0803.