2015-07

Simultaneous co-clustering and learning to address the cold start problem in recommender systems

Knowledge-Based Systems, Amsterdam, v. 82, p. 11-19, Jul. 2015
http://www.producao.usp.br/handle/BDPI/50927

Knowledge-Based Systems 82 (2015) 11–19

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Simultaneous co-clustering and learning to address the cold start problem in recommender systems

Andre Luiz Vizine Pereira [a,b,*], Eduardo Raul Hruschka [a]

[a] University of São Paulo, São Carlos, Brazil
[b] Rubens Lara College of Technology, FATEC-RL, Santos, Brazil

ARTICLE INFO

Article history:
Received 6 September 2014
Received in revised form 17 February 2015
Accepted 18 February 2015
Available online 3 March 2015

Keywords:
Recommender system
Cold starting
Co-clustering
Predictive modeling

ABSTRACT

Recommender Systems (RSs) are powerful and popular tools for e-commerce. To build their recommendations, RSs make use of varied data sources, which capture the characteristics of items, users, and their transactions. Despite recent advances in RS, the cold start problem is still a relevant issue that deserves further attention, and arises due to the lack of prior information about new users and new items. To minimize system degradation, a hybrid approach is presented that combines collaborative filtering recommendations with demographic information. The approach is based on an existing algorithm, SCOAL (Simultaneous Co-Clustering and Learning), and provides a hybrid recommendation approach that can address the (pure) cold start problem, where no collaborative information (ratings) is available for new users. Better predictions are produced from this relaxation of assumptions to replace the lack of information for the new user. Experiments using real-world datasets show the effectiveness of the proposed approach.

1. Introduction

Recommender Systems (RSs) are important components for e-commerce systems [29]. More recently, RSs have also been used to recommend movies [6], songs [17], videos [12], research resources in digital libraries [40,41], and people one may know from social networks [10,32]. To build their recommendations, RSs use varied data sources, which define the characteristics of items, users, and their transactions, and are categorized by the data sources and techniques used, such as, Content Based Filtering (CBF), Demographic Filtering (DF), and Collaborative Filtering (CF). CBF based RSs analyze a set of documents and/or descriptions of items previously evaluated by the user to build a user model or profile, which will then be used by the system for future recommendations of new items [4,38,51]. DF based RSs use attributes such as age, sex, occupation, and educational level to construct a demographic profile of the user, and different recommendations are generated for different demographic niches [30,37]. CF based RSs select items (documents, films, etc.) based on opinions that other users associate with them. Users share information and views about particular items, assigning scores that serve as reference benchmarks to other users.

CBF, DF, and CF RSs have been available for several years, and their advantages, performances, and limitations are well understood [1]. In particular, several CF based systems have been proposed and evaluated in recent years, providing satisfactory results in various commercial applications [9,43,25,34,26]. On the other hand, rather than adopting CF, DF, and CBF as standalone approaches, hybrid systems can combine the strengths better represent user needs [50,4,11,49,39,57,5].

An important issue for RSs is the cold start problem [45,28,2], which occurs due to the lack of prior information about new users and items. For instance, historical data about user profiles and their shopping preferences may not be available [1,42]. Similarly, characteristics of items might be unknown [21,36]. In many situations, the cold start problem leads to loss of new users who decide to stop using the system due to the low accuracy in the first recommendations made by the RS [8]. To minimize the system degradation caused by cold starting, a hybrid approach is proposed that combines collaborative filtering recommendations with demographic information and implements an iterative divide and conquer approach interleaving clustering and learning tasks to construct prediction models. An existing algorithm, SCOAL (Simultaneous Co-Clustering and Learning) [16], is used.

Our contributions are as follows: (i) we propose a hybrid RS based on SCOAL that addresses the (pure) cold start problem, where no collaborative information is available for new users; (ii) we show that better predictions can be built by relaxing the

* Corresponding author at: University of São Paulo, São Carlos, Brazil.
E-mail addresses: avizine@gmail.com (A.L. Vizine Pereira), erh@icmc.usp.br (E.R. Hruschka).

http://dx.doi.org/10.1016/j.knosys.2015.02.016
0950-7051/© 2015 Elsevier B.V. All rights reserved.

assumption on lack of user information. Although this is the expected behavior, quantitative analyses are not common in the literature and, as such, are a complementary contribution to our work.

Section 2 discusses recent advances related to the cold start problem. Section 3 reviews the SCOAL algorithm, and Section 4, presents our approach, based on the SCOAL algorithm, to address the cold start problem. The proposed approach is experimentally evaluated in Section 5, and compared to existing techniques. Section 6 reports our main conclusions and provides some directions for future work.

2. Related work

Many approaches have been proposed to address the cold start problem. A probabilistic model for combining CF and CBF has been proposed by Schein et al. [45]. It uses latent variables, i.e., aspect models between users/movies and movies/actors associations, to make recommendations for new movies. The well-known Expectation Maximization (EM) algorithm [13] is used to estimate the parameters of the model. A hybrid RS proposed by Wang and Wang [53] exploits the probabilistic latent semantic analysis [27] to model the relationship between user attributes and item attributes, and the EM algorithm to fit the model. Gantner et al. [18] use a method that maps item or user attributes to the latent features of a matrix factorization model. With such mappings, one can tackle the new-user and the new-item problems, thereby enhancing speed and predictive accuracy. In an RS based on trust networks [52], users are connected by means of trust scores and receive recommendations for items rated by people belonging to a web of trust network. In a related approach, a recommendation algorithm based on tripartite graphs (users-items-tags) Zhang et al. [58] considers social tags as a bridge between users and items. Tags based on social systems enable users to employ arbitrary tags to label the items of interest. In this setting, predictions are based on both the frequency of tags (such as personal preferences) and the semantic relationships between tags and items (such as global information). Another algorithm based on social networks data was proposed by Sahebi & Cohen [44]. Information from social networks in different dimensions (e.g. interaction between users, ratings, level of popularity/friendship, etc.), is used to detect latent communities of users. Thus, a new user can be inserted into a community that is more similar to his/her profile. The RS can then make recommendations based on ratings from users of that community, which are the closest according to the user's profile. Guo [22,23] merges the ratings of trusted neighbors and thus form a new rating profile for the active users. This new rating profile is then assessed through a Bayesian similarity measure that takes into account both direction and length of rating profiles.

A similarity measure, proximity impact popularity (PIP), was proposed by Ahn [2] as part of his approach to address cold start problems. PIP computes similarities between users and employs these to make recommendations. This approach has shown superior accuracy compared to other measures widely used in CF based RSs, particularly when a small number of ratings is available for computing similarities. Bobadilla et al. [8] proposed a similarity measure that considers not only the numerical information contained in the ratings, such as measures traditionally used in CF, e.g. Pearson correlation, but also uses information about the distribution and number of ratings obtained by each pair of users to be compared. The authors argue that it is more reasonable to set greater similarity between users who have positively evaluated a similar number of items than between users for which the number of items is very different. Basiri et al. [3] apply all the available information for each user to create an ordered weighted averaging operator [56] that is used to make recommendations. The operator uses a set of weights associated with each recommendation technique (CF, CBF, and DF) and their possible combinations to make predictions. Lin et al. [33] used information extracted from Twitter data to make recommendations of mobile applications. Their approach is based on latent Dirichlet allocation [7] which generates latent groups. A new user is mapped to previously defined latent groups using transitive relationships between them and applications.

A hybrid fuzzy linguistic RS based on the quality of items was proposed by Tejeda-Lorente et al. [50]and uses a recommendation strategy that switches between CBF and CF approaches to share user experiences in a university digital library. With this dual perspective, the cold start problem is minimized, because the system switches from one approach to another according to the situation of the system at any given moment.

Shaw et al. [47] use association rules to expand system user profiles based on patterns and associations of items, topics, and categories, thereby giving more information to a recommender system. An algorithm that combines association rules and data clustering has been presented by Sobhanam & Mariappan [48]. The association rules are used to create and expand the user profile to increase the number of ratings made by the user, thus minimizing the cold start problem. Clustering is used to group items that later are used to make predictions for new items. Another approach based on association rules proposed by Leung et al. [31] makes use of cross-level association rules (CLARE) to integrate content information about domain items into collaborative filters. The CLARE algorithm operates on a preference model, comprising user-item and item-item relationships, and infers user preferences for items from the attributes they possess using associations, item attributes, and other domain items, when no recommendations for that item can be generated using CF.

In the predictive model proposed by Park & Chu [35], attributes of movies (e.g. year of production, genre, cast, etc.) and user characteristics (e.g. demographic information, and history behavior) are used as independent variables for linear regression.

In summary, there are different approaches to address the cold start problem. Prediction models typically make use of information about the characteristics of the items and user demographics. There is also a need to identify clusters of users and similar items to improve prediction accuracy. The next section presents the SCOAL algorithm [16], which is a key component of our approach. This algorithm explores evaluations made by users (CF setting), as well as attributes that characterize items and users (CBF and DF settings), to learn a set of models that best represent similar clusters of users and items. Such clusters are automatically identified as part of the model learning process.

3. Review of the SCOAL algorithm

The SCOAL algorithm [16,14] implements an iterative divide and conquer approach that interleaves clustering and learning tasks. A mathematical optimization problem is formulated and an objective function is minimized through an iterative process, until a local minimum is found. A prediction model is generated for each co-cluster using sets of attributes comprising the rows and columns of a matrix, **Y**, where, for example, rows represent users and columns represent items (see Table 1). In this setting, the similarity between users is indirectly calculated by considering the predictions made by the learning models, and there is one learning model per co-cluster. For regression problems, such as those addressed in this paper, data clustering is based on the continuous outputs of the prediction models. This simultaneous

Table 1
Pedagogical example of a matrix, **Y**, used in recommender systems. In practice, users and items are described by means of attributes, which have been omitted here for simplicity.

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	5	3	4	4	?
User 2	3	1	2	3	3
User 3	4	3	4	3	4
User 4	3	3	1	5	4
User 5	1	5	5	2	1

approach for clustering and learning produces better results than clustering data first and then building regression models independently [16,14].

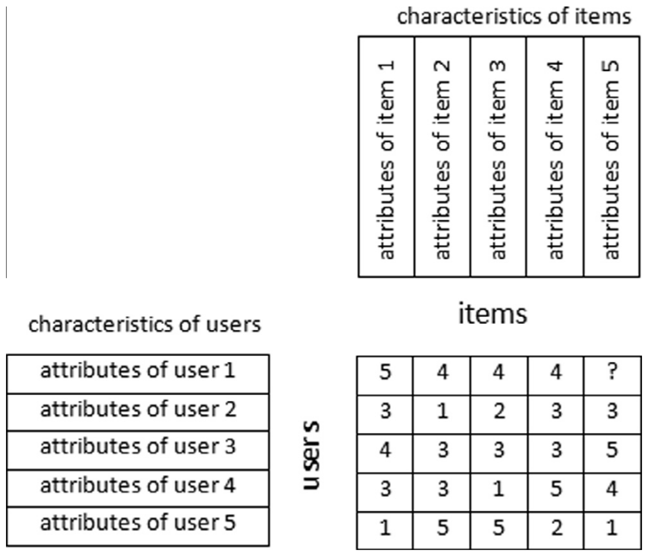
To illustrate the mode of operation, consider the scenario shown in Table 1. In addition to considering the values of each element, y_{ij}, of matrix **Y**, where rows represent users and columns represent items, SCOAL uses attributes that describe the specific characteristics of users and items in the process of learning the overall prediction model. Fig. 1 shows the data set used by SCOAL to build predictive models. For example, in the context of movie recommendations, user attributes could be name, age, gender, and profession, while movie attributes could be genre, date of release, director, and lead actor. Following standard notation for (linear) regression models, these attributes are the independent variables and y_{ij} are the respective values of the dependent variable.

Given the large amount of data involved in recommendation problems, it is unlikely that all user-movie ratings can be well represented by a single model. Rather, it is usually better to use learning models that represent the ratings of a subset of users for a subset of films. In this regard, SCOAL partitions the matrix of data into clusters of users and movies, so that each co-cluster can be well characterized by a particular predictive model. The similarity between two objects is not determined by the ratings, but by the results of their predictive models. Fig. 2 illustrates a hypothetical result obtained by SCOAL on the data in Table 1. There are four co-clusters represented by different colors, each one associated with its likely predictive model. Note that this solution involves partitioning the data matrix into two clusters of rows and two clusters of columns.

The attributes of the i-th user are represented by vector, $\mathbf{u}_i$, and the attributes of the j-th item are represented by vector, $\mathbf{v}_j$. As usual for a linear regression model, each element of matrix, Y (Table 1), can be written as $y_{ij} = \boldsymbol{\beta}_{ij}^T\mathbf{x}_{ij} + \mathcal{N}(0,\sigma^2)$, where $\mathbf{x}_{ij}$ is the vector that concatenates the attributes of users and items, $\mathbf{x}_{ij} = [1, \mathbf{u}_i, \mathbf{v}_j]$. From the observed data, one can then estimate $\hat{\boldsymbol{\beta}}_{ij}^T$ and later make predictions for any user-item (i,j) pair,

$$\hat{y}_{ij} = \hat{\boldsymbol{\beta}}_{ij}^T\mathbf{x}_{ij}, \tag{1}$$

where $\hat{\boldsymbol{\beta}}_{ij}^T$ concatenates the parameters of the model, i.e., $\hat{\boldsymbol{\beta}}_{ij}^T = [\hat{\beta}^0, \hat{\boldsymbol{\beta}}_i^T, \hat{\boldsymbol{\beta}}_j^T]$.



5	4	4	4	?
3	1	2	3	3
4	3	3	3	5
3	3	1	5	4
1	5	5	2	1

Fig. 1. Data used by SCOAL to build prediction models (adapted from [15]).

The SCOAL algorithm obtains a set of U clusters of rows (users), $\mathbb{C}^u = \{C_1^u, C_2^u, \ldots, C_U^u\}$, where $C_r^u \in \mathbb{C}^u, r = 1, \ldots, U$, and a set of V clusters of columns (items), $\mathbb{C}^v = \{C_1^v, C_2^v, \ldots, C_V^v\}$, where $C_c^v \in \mathbb{C}^v, c = 1, \ldots, V$, simultaneously, such that the values for each co-cluster, $C_{r,c}^{u,v}$, may be predicted using the same regression model specific to each co-cluster. The objective function to be minimized is

$$\sum_{ij} w_{ij}(y_{ij} - \hat{y}_{ij})^2, \tag{2}$$

where $\hat{y}_{ij} = \hat{\boldsymbol{\beta}}_{rc}^T\mathbf{x}_{ij}, \hat{\boldsymbol{\beta}}_{rc}^T$ is the set of parameters for co-cluster $C_{r,c}^{u,v}$, $u_i \in C_r^u$, $v_j \in C_c^v$; and

$$w_{ij} = \begin{cases} 1 \ \textit{if} \ y_{ij} \neq ? \\ 0 \ \textit{if} \ y_{ij} = \ ? \end{cases}.$$

The main steps of SCOAL are summarized in Algorithm 1. Note that SCOAL, as used in our work, is based on the single-cluster assumption. A discussion about cluster assumptions in the context of RSs is outside the scope of our work and can be found in Xu et al. [55].

Algorithm 1. SCOAL (adapted from [16]).

```
Inputs: $\mathbf{Y}_{I\times J}, \mathbf{W}_{I\times J}, \mathbf{x}_{ij} = [1, \mathbf{u}_i, \mathbf{v}_j], \forall i, j, U, V$.
Output: Co-clustering, $(\mathbb{C}^u, \mathbb{C}^v)$, and the respective co-cluster
  models, $(\hat{\boldsymbol{\beta}}$s).
Initialize a random co-clustering. $(\mathbb{C}^u, \mathbb{C}^v)$;
  repeat
    Step 1: Build a regression model for each co-cluster;
    Step 2: Find row (user) clusters by assigning each row to
  the cluster that minimizes the prediction error;
    Step 3: Find column (item) clusters by assigning each
  column to the cluster that minimizes the prediction error;
  until Convergence criterion is met.
Return $\mathbb{C}^u, \mathbb{C}^v$, and $\hat{\boldsymbol{\beta}}$.
```

4. Tackling the cold start problem

Concepts from the SCOAL algorithm are used to build the hybrid approach to address cold start problems. We note that SCOAL was not originally conceived to deal with cold start problems and our approach is of broad scope, not limited to SCOAL, i.e., it can be used as a cold start tool for other RSs provided the required data is available. Fig. 3 outlines our contributions (cold start recommendations on the right hand side) by considering SCOAL as the main engine of an RS system. There are two main modules:

Simultaneous co-clustering and learning models: The purpose of this module is to find groups of users with common interests and simultaneously build prediction models that are specific to each group. In principle, we are mostly interested in considering

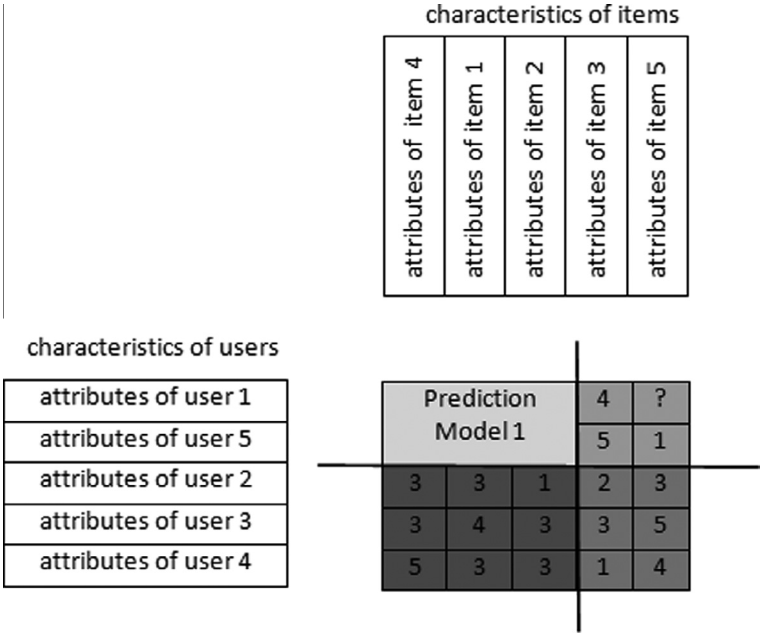


Fig. 2. Solution found by SCOAL after rearranging the rows and columns from Table 1. The different colors represent the induced clusters. The first rectangle represents a prediction model (adapted from [15]).

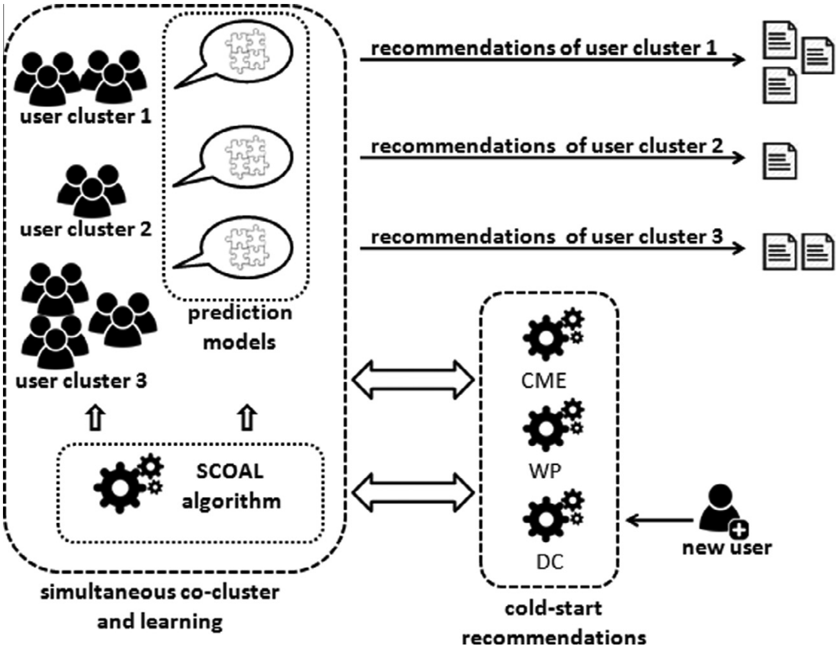

user cluster 1
user cluster 2
user cluster 3
prediction models
SCOAL algorithm
simultaneous co-cluster and learning
recommendations of user cluster 1
recommendations of user cluster 2
recommendations of user cluster 3
CME
WP
DC
new user
cold-start recommendations

Fig. 3. Overview of the proposed tools for addressing cold start as components of a recommender system.

the cold start problem for new users, but our proposal also makes it possible to address the analogous problem for new items, because the algorithm is based on clustering the data matrix in both dimensions, rows (users) and columns (items). Once convergence is met (Algorithm 1), the output provided by SCOAL is the matrix **Y** partitioned in blocks of row clusters, $\mathbb{C}^u$, and column clusters, $\mathbb{C}^v$. Considering the example of Fig. 2, after rearranging the rows and columns of the input matrix **Y** (Table 1), the SCOAL output is two row clusters, $(C_1^u = \{user1, user5\}, C_2^u = \{user2, user3, user4\})$, and two column clusters, $(C_1^v = \{item4, item1, item2\}, C_2^v = \{item3, item5\})$.

Cold start recommendations: Our proposed hybrid approach is used to identify which group of users (generated by SCOAL) the new user should be allocated to, so that predictions can be made based on the applicable prediction models of that group. This module allows the system not only to address cold starting, where a minimum number of ratings made by the new user is provided, but also for situations where no ratings are available for the new user. The remaining subsections describe the components of this module.

4.1. Cluster with Minimum Error (CME)

The CME approach is similar to the computation in Step 2 of the SCOAL Algorithm proposed by Deodhar & Ghosh [14]. It is based on the premise that a minimum number of ratings made by the new user is required for the system to build his/her profile, as in the Netflix RS. A small number of initial ratings made by the user is employed to find the row cluster, r^*, whose models minimize the mean squared error (MSE) for these initial ratings,

$$r^* = \arg_r \min\{\sum_{n=1}^{N}(y_{new,n} - \hat{y}_{new,n})^2\}, \tag{3}$$

where N is the number of initial ratings made by the new user; $n = 1, \ldots, N$ are indices that represent the items initially rated by the user; $\hat{y}_{new,n} = \hat{\beta}_{rc}^T \mathbf{x}_{new,n}, \mathbf{x}_{new,n} = [1, \mathbf{u}_{new}, \mathbf{v}_n], \mathbf{u}_{new} \in C_r^u$; and $\mathbf{v}_n \in C_c^v$. After identifying r^*, the new user is assigned to this row cluster and all the predictions for them will be made based on the models from this cluster,

$$\hat{y}_{new,j} = \hat{\beta}_{r^*c}^T \mathbf{x}_{new,j}, \mathbf{u}_{new} \in C_{r^*}^u, \mathbf{v}_j \in C_c^v. \tag{4}$$

4.2. Pure cold start problems

As discussed in Section 2, most of the approaches described in the literature [45,52,58,47,36] are not capable of dealing with the pure cold start problem, for which no ratings are available for a new user. A few exceptions tackle this problem by computing similarities between users, e.g. using Pearson correlation [8], cosine measure [58], and PIP [2], or making factorization models attribute-aware by plugging learnable mapping functions into them [18]. In contrast. we use the row clusters found by SCOAL (see Fig. 2) to build a classification model. A dataset is generated where the instances are the existing users, described by their attributes and includes their respective cluster labels, which will be later treated as class labels. Table 2 illustrates an example of such a dataset, where the user attributes are name, age, gender, and occupation.

The dataset obtained from SCOAL is used to build a classifier to indicate the best prediction models for a new user. Our goal is to find r^*, the best row cluster according to the conditional probability of the class credited to the new user, $P(Class|\mathbf{u}_{new})$. Here, class refers to a row cluster label that, in turn, provides regression models to be used to make predictions for a new user. Thus, it is expected to find,

$$r^* = \arg_r \max P(C_r^u|\mathbf{u}_{new}). \tag{5}$$

Once r^* is found, recommendations for the new user are made following Eq. (4).

We chose three popular classifiers to illustrate the approach, Naive Bayes, J48 Decision Tree, and Logistic Regression, which are available in Weka's API [24]. We anticipate that these standard classifiers will provide good results (as described in Section 5). However, predictions can be improved by smoothing the classifier outputs, as described in Sections 4.2.1 and 4.2.2.

4.2.1. Weighted prediction

The weighted prediction (WP), as discussed above, devises a classifier and estimates the class probability distribution for each row cluster. Then, rather than choosing a single row cluster to make predictions for a new user, all clusters are used, and a weighted average is calculated from the individual predictions of every row cluster, where the weights are the class probabilities,

$$\hat{y}_{new,j} = \sum_{r=1}^{U}\left[\frac{P(C_r^u|\mathbf{x}_{new})\cdot\hat{\beta}_{r,c}^T\mathbf{x}_{new,j}}{\sum_{r=1}^{U}P(C_r^u|\mathbf{x}_{new})}\right]. \tag{6}$$

4.2.2. Dynamic classification

The Dynamic Classification (DC) approach is a variant of WP (described in Section 4.2.1). In contrast to WP, DC considers one item at a time and builds a specific classification model for every item for which a rating has to be estimated. The dataset used to build the classification model is assembled dynamically, one item at a time, and considering only the users who have rated that particular item of interest. Thus, the dataset used to estimate the class probabilities is updated dynamically, taking into account the different items to be rated for a new user. Except for this relevant detail, the predictions are made as given by Eq. (6).

Table 2
Dataset used to build a classification model for pure cold start problems. The cluster labels obtained from SCOAL become the class labels, which are used to build a classification model capable of finding the best SCOAL based prediction models for new users.

User	Age	Gender	Occupation	Cluster
1	57	Male	Admin	1
5	36	Male	Admin	1
2	24	Male	Operator	2
3	53	Female	Professor	2
4	23	Female	Writer	2

Clearly, DC requires computationally efficient classifiers. Therefore, we illustrate its use by applying a Naive Bayes classifier. Depending on the computer resources available to the user, more computationally demanding classifiers can be used.

5. Experimental evaluation

We evaluated the proposed approaches in two scenarios: (i) cold start, with a small number of ratings per user available; and (ii) pure cold start, with no ratings available for a new user. Three real-world datasets traditionally used for assessing RSs were used, as described in the following sections.

5.1. Datasets

5.1.1. Movielens

The Movielens dataset comprises 100,000 ratings made by 943 users on 1682 movies. Ratings vary from 1 to 5. Each user has rated at least 20 movies. The dataset is available through the GroupLens Research Project from the University of Minnesota.[1] In addition to the ratings, the dataset also provides demographic information of the users and information about the movies, from which a set of 23 attributes was used to train the prediction models. There are 19 different genres (e.g. drama, action, adventure, romance, etc.), which were captured through a binary vector of size 19. To improve the learning model, features derived from the data matrix ($\mathbf{Y}$) were used,

- Number of movies rated by each user.
- Average of the ratings made by each user.
- Variance of the ratings made by each user.
- Number of users that rated each movie.
- Overall average of the ratings for each movie.
- Overall variance of the ratings for each movie.

5.1.2. Jester

Th Jester data set was extracted from an online jokes recommender system called Jester [20]. The dataset gathers ratings from 24,983 users for 100 jokes. Each user rated a minimum of 36 jokes,[2] from −10 to +10. Information about users and jokes is not originally available but, as with the Movielens dataset, one can derive them from the data matrix ($\mathbf{Y}$). In our experiments, we used

[1] Download link: http://www.grouplens.org/datasets/movielens/.

[2] Download link: http://eigentaste.berkeley.edu/dataset/.

the same six features listed above for the Movielens dataset, but now the items are jokes rather than movies.

The computation of (additional) user/item features allows the application of our proposed approach in a variety of real-world problems. Most importantly, these features are likely to improve prediction models and, as such, their use is encouraged.

5.1.3. Netflix

The Netflix dataset has been widely used to assess the performance of RSs. It was originally built to support the Netflix Prize,[3] and has more than 100,000,000 ratings made by 480,000 users. Data were collected from October 1998 to December 2005 on a scale from 1 to 5. For this experiment, the original dataset was sampled by randomly choosing 2000 users and 17,770 movies. A user should have rated at least 20 movies to be included in the dataset. Similar to the Jester and Movielens datasets, six additional features from the data matrix were used.

5.2. Experimental setup

It is not our goal to evaluate the performance of SCOAL [16] as a standalone algorithm. This has been extensively covered in its original and subsequent references. Rather, we want to show the efficacy of our proposed approach to deal with cold start problems. Therefore, we have not performed any optimization on SCOAL parameters. We arbitrarily adopted $U = 4$ (number of row clusters) and $V = 4$ (number of column clusters). In practice, these parameters could be optimized in a number of ways, but such study is out of the scope of our work. The convergence criterion was defined as the maximum relative difference of MSE in two consecutive iterations of SCOAL to be less than or equal to 10^{-3}. As usual, we employed 10-fold cross-validation [54]. Following standard and well-established methodology, we evaluated the results using the normalized mean absolute error (NMAE) [20]. For WP, we used three classifiers (Naive Bayes, J48 decision tree, and Logistic Regression), whereas for DC we used only the Naive Bayes classifier.

5.3. Partial cold start problems

Consider the partial cold start case where a small number of ratings is available for each new user. We simulated artificial new users by varying the number of ratings initially available for each new user from 1 to 20. For each level of ratings (120), a recommendation was performed for each dataset and each similarity measure. The proposed CME (Section 4.1) was assessed for each scenario and compared to other approaches widely used in the literature.

For comparison purposes we computed similarities between users by adopting three measures widely used in RS, Pearson correlation, cosine measure, and constrained Pearson [46]. The Pearson correlation measures the linear correlation between two vectors of ratings; the cosine measure looks at the angle between two vectors of ratings where a smaller angle is regarded as implying greater similarity; and the constrained Pearson is a modified version of the Pearson correlation that allows only pairs of ratings on the same side (e.g. positive or negative ratings) to contribute to the correlation. Although widely used, these measures are vulnerable to data scarcity and are not appropriate for cold start problems. We also compare our proposed approach to PIP [2], which focuses on improving RS performance under partial cold start conditions where only a small number of ratings are available.

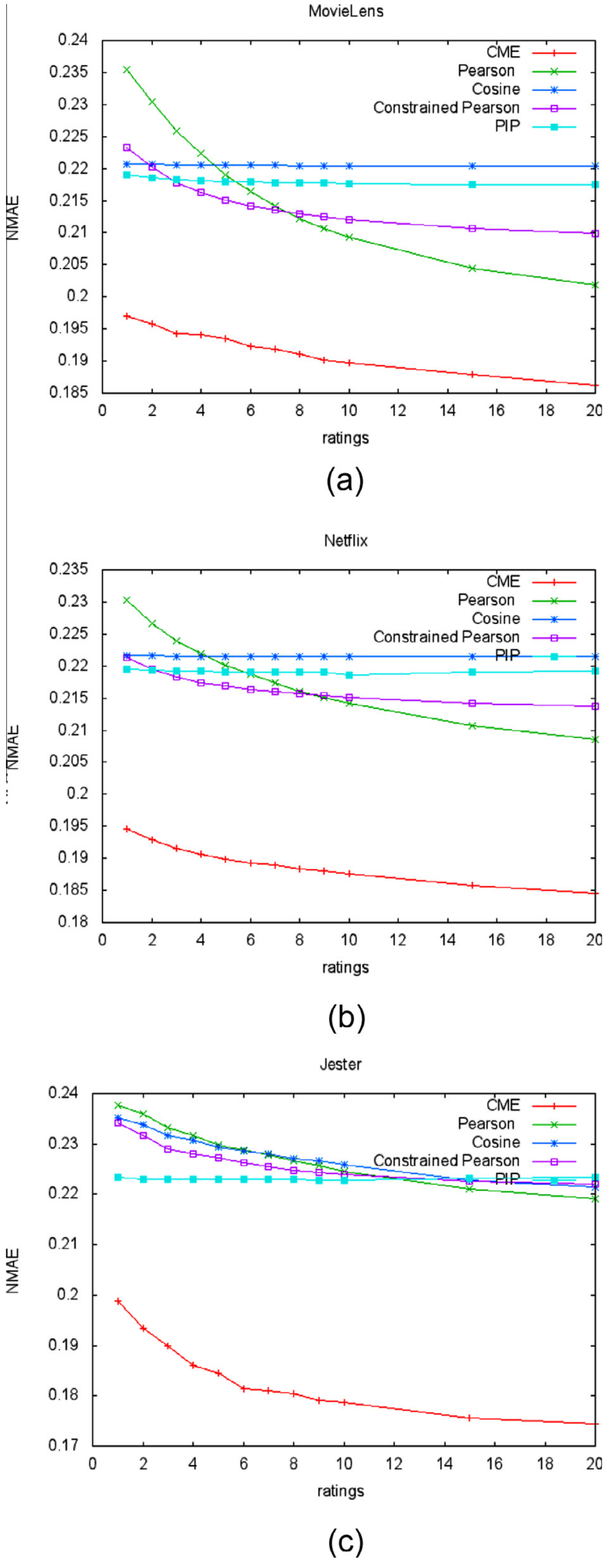


Fig. 4. Partial cold start scenarios: normalized mean absolute error (NMAE) for different levels of available ratings.

Fig. 4 summarizes the results. CME provided the best results for all the assessed settings, whereas PIP obtains better results than the baseline approaches only for particular scenarios. PIP improves on simpler approaches only if there are less than 4 initial ratings for Movielens and Netflix, and less than 10 for Jester.

[3] http://www.netflixprize.com/.

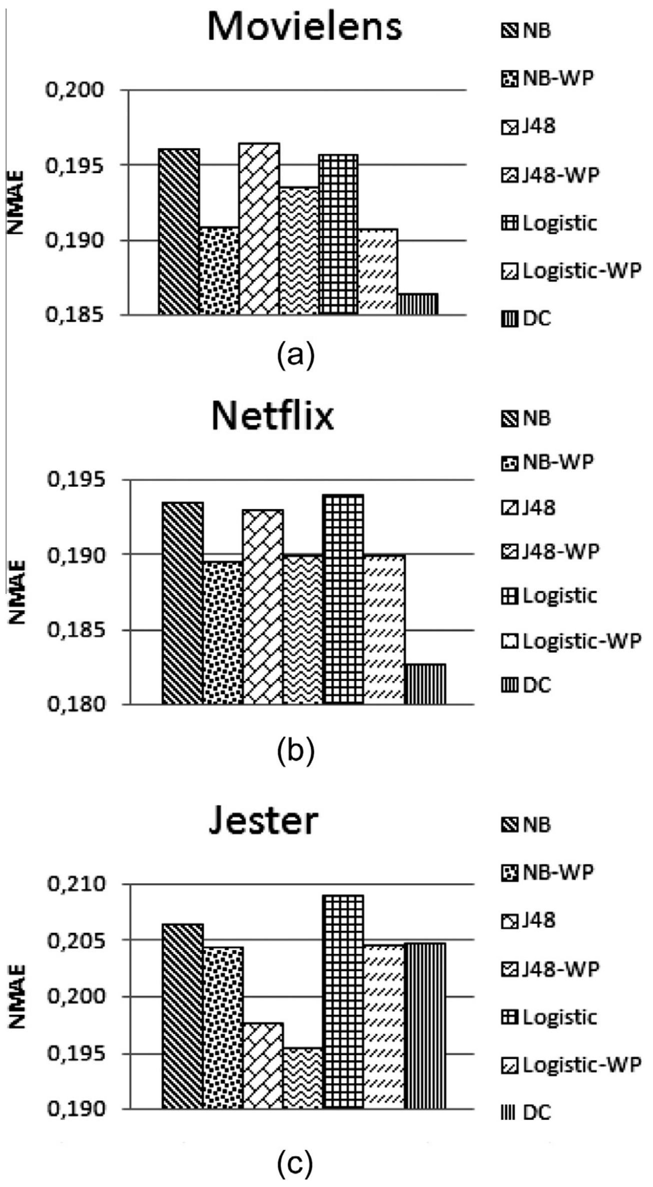


Fig. 5. Pure cold start scenarios: normalized mean absolute error (NMAE).

5.4. Pure cold start problems

For pure cold start problems, where no previous ratings are available for a new user, measures that capture similarities between users cannot be computed and, as a consequence, most of the approaches described in the literature are inappropriate. However, our proposed WP (Section 4.2.1) and DC (Section 4.2.2) approaches address this issue, and the outcomes are summarized in Fig. 5. All recommendations were performed for each dataset with test sets of new users only, i.e., without any ratings. In most cases, DC achieved a superior performance to WP. However, WP based on J48 outperformed DC on the Jester dataset. This is because of the better accuracy of the J48 classifier in this dataset, as shown in Fig. 6, that shows the classification accuracies obtained by each classifier built to address the cold start problem. Recall from the example in Table 2, that the cluster labels obtained from SCOAL become class labels, which are then used to build a classification model capable of finding the best SCOAL based prediction models for new users.

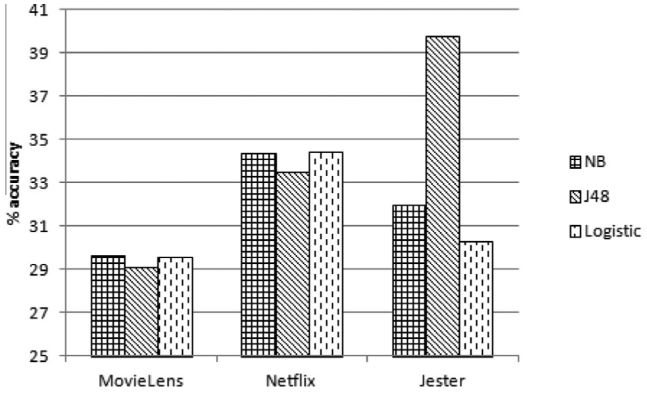


Fig. 6. Classification accuracies obtained by considering that each cluster label found by SCOAL is a class label for the cold start problem.

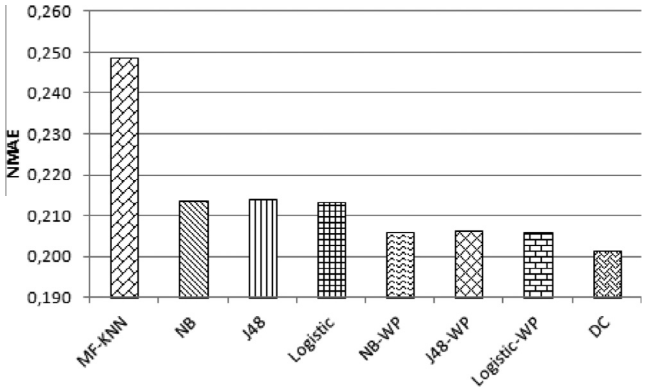


Fig. 7. Movielens recommender system by MF-KNN, NB, J48, Logistic, WP, and DC: normalized mean absolute error.

5.4.1. Comparisons with the MF-KNN method

We used a matrix factorization K-nearest neighbors (MF-KNN) method, proposed in Gantner et al. [18], to compare to our proposed approach. MF-KNN uses traditional matrix factorization to derive a set of latent features from **Y** to characterize users and items. The latent features of a user or item can only be set if the user/item occurs in the training set. In a cold start scenario this is not the case. However, it is possible to make use of the factorization model for new users by estimating their latent features from the side information (e.g. age, occupation, gender).

MF-KNN consists of:

1. Training the factorization model using **Y**.
2. Learning mapping functions from attribute to factor space. For simplicity, following the description provided in [18], we make use of a k-nearest neighbor, with $k = 1$ and using the nearest neighbor as the most similar user, where the cosine of the $\mathbf{u}$ vectors (user attributes) is the similarity measure. That is, predictions for a new user were performed using the latent features of the most similar user in the training set (1-NN).

The algorithms used in step 1 (matrix factorization) and step 2 (user attribute KNN) are available in the MyMediaLite RS algorithm library [19]. As suggested in [19], the matrix factorization parameters were set as $\lambda = 0.015$, $\alpha = 0.01$, iterations = 100, StDev = 0.1 and numFactors = 32.

Comparisons with MF-KNN can only be performed in datasets for which side information not derived from the ratings matrix is available. Movielens was the only dataset that meets this

Table 3
Elapsed CPU Time for recommender systems on the Movielens datset.

Method	NB	WP-NB	J48	WP-J48	Logistic	WP-Logistic	DC	MF-KNN
CPU time (s)	0.017	0.031	0.014	0.027	0.016	0.028	4.789	0.545

Table 4
Normalized mean absolute error (standard deviation). Bold text shows the best results.

		Movielens	Netflix	Jester
Incremental cold start	CME	**0.18624(0.003)**	**0.18458(0.002)**	**0.17435(0.003)**
	Pearson	0.20186(0.010)	0.20854(0.009)	0.21921(0.003)
	Cosine	0.22040(0.009)	0.22147(0.008)	0.22152(0.003)
	Constrained Pearson	0.20997(0.010)	0.21368(0.009)	0.22194(0.003)
	PIP	0.21746(0.009)	0.21913(0.008)	0.22328(0.003)
Pure cold start	Naive Bayes (NB)	0.19602(0.006)	0.19348(0.001)	0.20643(0.009)
	NB-WP	0.19089(0.007)	0.18953(0.001)	0.20447(0.007)
	J48	0.19651(0.008)	0.19291(0.000)	0.19772(0.005)
	J48-WP	0.19347(0.008)	0.18991(0.001)	**0.19553(0.008)**
	Logistic	0.19575(0.007)	0.19405 (0.002)	0.20893(0.006)
	Logistic-WP	0.19076(0.007)	0.18988(0.002)	0.20457(0.007)
	DC	**0.18647(0.004)**	**0.18275(0.002)**	0.20476(0.006)

requirement. We used gender, occupation, and age (segmented to teenager, adult, and old) as inputs for the user attribute KNN. The item attributes were 19 different genres. As usual, these nominal attributes were encoded using binary representations.

Following standard procedures, we performed 10-fold cross-validation. Fig. 7 shows the NMAE, and Table 3 shows the mean execution times achieved using a Macbook Pro 2 GHz Intel Core I7.

5.5. Discussion

For incremental cold start scenarios, CME provides the best results for all the assessed datasets. For pure cold start scenarios, DC shows the best results for most cases.

We did not consider the relative accuracy of results from DC and CME methods. The latter incorporates partial information (in the form of a small number of ratings for new users), whereas the former does not. However, Table 4 shows the best results were obtained by CME, for the case of 20 initial ratings. For Movielens and Netflix, DC is comparable to CME. For Jester, where CME is superior to DC, the DC results are still acceptable.

In summary, the proposed approach for cold start problems is practical and effective.

6. Final remarks and future work

Four approaches were described to address cold start problems frequently encountered in recommender systems, originating from the lack of prior information of new users and/or items. In real-world applications, the inability to appropriately tackle cold start problems commonly leads to churn issues. Therefore, developing better tools to address these problems is of paramount importance. Our proposed approaches have shown to be very useful, yielding superior outcomes for three real-world datasets compared to state of the art methods widely used to assess the performance of RSs. In particular, the DC approach is effective and very promising although computationally demanding.

Considering future work, we need to investigate the use of regression ensembles, which usually yield better predictions. However, we expect that computational issues may limit the advantages, which may be possibly addressed by high performance computation solutions particularly adapted to this problem.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, EEE Trans. Knowl. Data Eng. 17 (2005) 734–749.

[2] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold starting problem, Inform. Sci. 178 (2008) 37–51.

[3] J. Basiri, A. Shakery, B. Moshiri, M.Z. Hayat, Alleviating the cold start problem of recommender systems using a new hybrid approach, in: 2010 5th International Symposium Telecommunications (IST), IEEE, 2010, pp. 962–967.

[4] C. Basu, H. Hirsh, W. Cohen, et al., Recommendation as classification: using social and content-based information in recommendation, in: AAAI/IAAI, 1998, pp. 714–720.

[5] P. Bedi, P. Vashisth, Empowering recommender systems using trust and argumentation, Inform. Sci. 279 (2014) 569–586.

[6] J. Bennett, L. Stan, The netflix prize, in: Proceedings of KDD Cup and Workshop, 2007, p. 35.

[7] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Machine Learning Res. 3 (2003) 993–1022.

[8] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowl.-Based Syst. 26 (2012) 225–238.

[9] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.

[10] J. Chen, W. Geyer, C. Dugan, M. Muller, I. Guy, Make new friends, but keep the old: recommending people on social networking sites, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2009, pp. 201–210.

[11] M.K. Condliff, D.D. Lewis, D. Madigan, C. Posse, Bayesian mixed-effects models for recommender systems, in: ACM SIGIR, Citeseer vol. 99, 1999, pp. 23–30.

[12] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, D. Sampath, The youtube video recommendation system, in: Proceedings of the Fourth ACM Conference on Recommender Systems, 2010, pp. 293–296.

[13] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. Roy. Stat. Soc. 39 (1977) 1–38.

[14] M. Deodhar, J. Ghosh, A framework for simultaneous co-clustering and learning from complex data, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 250–259.

[15] M. Deodhar, J. Ghosh, Simultaneous partitioning and learning for prediction from complex data – tutorial, in: Second International Conference on Pattern Recognition and Machine Learning, 2007.

[16] M. Deodhar, J. Ghosh, Scoal: a framework for simultaneous co-clustering and learning from complex data, ACM Trans. Knowl. Discov. Data 4 (2010) 11.

[17] G. Dror, N. Koenigstein, Y. Koren, M. Weimer, The yahoo! music dataset and kdd-cup11, J. Machine Learning Res. – Proc. Track 18 (2012) 8–18.

[18] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, L. Schmidt-Thieme, Learning attribute-to-feature mappings for cold start recommendations, in: 2010 IEEE 10th International Conference on Data Mining (ICDM), IEEE, 2010, pp. 176–185.

[19] Z. Gantner, S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, MyMediaLite: a free recommender system library, in: Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011), 2011.

[20] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Inform. Retrieval 4 (2001) 133–151.

[21] A. Gunawardana, C. Meek, Tied boltzmann machines for cold start recommendations, in: Proceedings of the 2008 ACM Conference on Recommender Systems, 2008, pp. 19–26.
[22] G. Guo, Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 451–454.
[23] G. Guo, J. Zhang, D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, Knowl.-Based Syst. 57 (2014) 57–68.
[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, ACM SIGKDD Explorat. Newslett. 11 (2009) 10–18.
[25] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inform. Syst. (TOIS) 22 (2004) 5–53.
[26] A. Hernando, J. Bobadilla, F. Ortega, J. Tejedor, Incorporating reliability measurements into the predictions of a recommender system, Inform. Sci. 218 (2013) 1–16.
[27] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1999, pp. 50–57.
[28] Z. Huang, H. Chen, D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Trans. Inform. Syst. (TOIS) 22 (2004) 116–142.
[29] P.B. Kantor, L. Rokach, F. Ricci, B. Shapira, Recommender Systems Handbook, Springer, 2011.
[30] B. Krulwich, Lifestyle finder: intelligent user profiling using large-scale demographic data, AI Mag. 18 (1997) 37.
[31] C.W.-k. Leung, S.C.-f. Chan, F.-l. Chung, An empirical study of a cross-level association rule mining approach to cold start recommendations, Knowl.-Based Syst. 21 (2008) 515–529.
[32] Y.-M. Li, C.-L. Chou, L.-F. Lin, A social recommender mechanism for location-based group commerce, Inform. Sci. 274 (2014) 125–142.
[33] J. Lin, K. Sugiyama, M.-Y. Kan, T.-S. Chua, Addressing cold start in app recommendation: latent user models constructed from twitter followers, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2013, pp. 283–292.
[34] F. Ortega, J.-L. SáNchez, J. Bobadilla, A. Gutiérrez, Improving collaborative filtering-based recommender systems results using pareto dominance, Inform. Sci.: Int. J. 239 (2013) 50–61.
[35] S.T. Park, W. Chu, Pairwise preference regression for cold start recommendation, in: Proceedings of the Third ACM Conference on Recommender Systems, 2009, pp. 21–28.
[36] S.T. Park, D. Pennock, O. Madani, N. Good, D. DeCoste, Naive filterbots for robust cold start recommendations, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 699–705.
[37] M.J. Pazzani, A framework for collaborative, content-based and demographic filtering, Artif. Intell. Rev. 13 (1999) 393–408.
[38] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, 2007, pp. 325–341.
[39] A. Popescul, D.M. Pennock, S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc, 2001, pp. 437–444.
[40] C. Porcel, E. Herrera-Viedma, Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries, Knowl.-Based Syst. 23 (2010) 32–39.
[41] C. Porcel, J.M. Moreno, E. Herrera-Viedma, A multi-disciplinar recommender system to advice research resources in university digital libraries, Expert Syst. Appl. 36 (2009) 12520–12528.
[42] A.M. Rashid, I. Albert, D. Cosley, S.K. Lam, S.M. McNee, J.A. Konstan, J. Riedl, Getting to know you: learning new user preferences in recommender systems, in: Proceedings of the 7th International Conference on Intelligent User Interfaces, 2002, pp. 127–134.
[43] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, 1994, pp. 175–186.
[44] S. Sahebi, W. Cohen, Community-based recommendations: a solution to the cold start problem, in: Workshop on Recommender Systems and the Social Web, RSWEB, 2011.
[45] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold start recommendations, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2002, pp. 253–260.
[46] U. Shardanand, P. Maes, Social information filtering: algorithms for automating word of mouth, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co, 1995, pp. 210–217.
[47] G. Shaw, Y. Xu, S. Geva, Using association rules to solve the cold start problem in recommender systems, in: Advances in Knowledge Discovery and Data Mining, Springer, 2010, pp. 340–347.
[48] H. Sobhanam, A. Mariappan, Addressing cold start problem in recommender systems using association rules and clustering technique, in: 2013 International Conference on Computer Communication and Informatics (ICCCI), IEEE, 2013, pp. 1–5.
[49] I. Soboroff, C. Nicholas, Combining content and collaboration in text filtering, in: Proceedings of the IJCAI, vol. 99, 1999, pp. 86–91.
[50] I. Tejeda-Lorente, C. Porcel, E. Peis, R. Sanz, E. Herrera-Viedma, A quality based recommender system to disseminate information in a university digital library, Inform. Sci. 261 (2014) 52–69.
[51] R. Van Meteren, M. Van Someren, Using content-based filtering for recommendation, in: Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, 2000.
[52] P. Victor, M. De Cock, C. Cornelis, A. Teredesai, Getting cold start users connected in a recommender systems trust network, Comput. Intell. Decis. Control 1 (2008) 877–882.
[53] G. Wang, Y. Wang, Probabilistic attribute mapping for cold start recommendation, in: Foundations and Applications of Intelligent Systems, Springer, 2014, pp. 421–431.
[54] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.
[55] B. Xu, J. Bu, C. Chen, D. Cai, An exploration of improving collaborative recommender systems via user-item subgroups, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 21–30.
[56] R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decisionmaking, IEEE Trans. Syst. Man Cybernet. 18 (1988) 183–190.
[57] Z. Zhang, H. Lin, K. Liu, D. Wu, G. Zhang, J. Lu, A hybrid fuzzy-based personalized recommender system for telecom products/services, Inform. Sci. 235 (2013) 117–129.
[58] Z.-K. Zhang, C. Liu, Y.-C. Zhang, T. Zhou, Solving the cold start problem in recommender systems with social tags, EPL (Europhys. Lett.) 92 (2010) 28002.