Expert Systems With Applications 92 (2018) 507–520

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques

Mehrbakhsh Nilashi[a,b,*], Othman Ibrahim[a], Karamollah Bagherifard[c]

[a] Faculty of Computing, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[b] Department of Computer Engineering, Lahijan Branch, Islamic Azad University, Lahijan, Iran
[c] Young Researchers and Elite Club, Yasooj Branch, Islamic Azad University, Yasooj, Iran

ARTICLE INFO

Article history:
Received 22 July 2017
Revised 23 September 2017
Accepted 24 September 2017
Available online 29 September 2017

Keywords:
Recommender systems
Ontology
Clustering
Dimensionality reduction
Scalability
Sparsity

ABSTRACT

Improving the efficiency of methods has been a big challenge in recommender systems. It has been also important to consider the trade-off between the accuracy and the computation time in recommending the items by the recommender systems as they need to produce the recommendations accurately and meanwhile in real-time. In this regard, this research develops a new hybrid recommendation method based on Collaborative Filtering (CF) approaches. Accordingly, in this research we solve two main drawbacks of recommender systems, sparsity and scalability, using dimensionality reduction and ontology techniques. Then, we use ontology to improve the accuracy of recommendations in CF part. In the CF part, we also use a dimensionality reduction technique, Singular Value Decomposition (SVD), to find the most similar items and users in each cluster of items and users which can significantly improve the scalability of the recommendation method. We evaluate the method on two real-world datasets to show its effectiveness and compare the results with the results of methods in the literature. The results showed that our method is effective in improving the sparsity and scalability problems in CF.

1. Introduction

Finding information in large-scale websites is a difficult and time-consuming process. Artificial Intelligence (AI) approaches are appearing at the forefront of research in information retrieval and information filtering systems. Recommender systems are a good example of one such AI approach. They have emerged in the e-commerce domain and are one way to address this issue. Such systems have been developed to actively recommend relevant information to users (Jugovac, Jannach, & Lerche, 2017; Nilashi, Jannach, bin Ibrahim, Esfahani, & Ahmadi, 2016a), typically without the need for an explicit search query. The history of recommender systems dates back to 1979 in relation to cognitive science (Rich, 1979). These systems have been important tools among other application areas such as, information retrieval (Salton, 1989), tourism (Kabassi, 2010), management science (Murthi & Sarkar, 2003), approximation theory (Powell, 1981), consumer choice modeling in business and marketing (Lilien, Kotler, & Moorthy, 1992), and forecasting theories (Armstrong, 2001).

Collaborative Filtering (CF) systems are information retrieval systems that operate under the assumption a user will like the same data items that other users have liked in the past. These systems are particularly popular and have been applied in many online shopping websites (Nilashi, Jannach et al. 2016; Nilashi, Salahshour et al., 2016b). CF algorithms mainly aggregate feedback for items from different users and use the similarities between items and items (item-based) or between users and users (user-based) to provide recommendations to a target user (Nilashi, Jannach, bin Ibrahim, & Ithnin, 2015).

Basically, CF recommendation algorithms are based on two main categories which are model-based and memory-based methods (Adomavicius & Tuzhilin, 2005). Memory-based (or heuristic-based) methods, such as correlation analysis and vector similarity, search the user database for user profiles that are similar to the profile of the active user that the recommendation is made for. In this type of recommender systems, it is important that the user and item databases remain in system memory during the algorithm's runtime. Because heuristic-based approaches can make predictions based on the local neighbourhood of the active user, or can base their predictions on the similarities between items, these systems can also be classed into the user-based and item-based approaches (Sarwar et al., 2001).

* Corresponding author.
E-mail addresses: nilashi@liau.ac.ir (M. Nilashi), othmanibrahim@utm.my (O. Ibrahim), k.bagherifard@iauyasooj.ac.ir (K. Bagherifard).

https://doi.org/10.1016/j.eswa.2017.09.058
0957-4174/

Memory- and model-based approaches have some advantages and disadvantages for item recommendation. Sparsity has been one of the main difficulties associated with these approaches, whereas recommendation with high accuracy has been one of the important advantages of the memory-based approach. However, this approach is not scalable for current recommendation systems as their databases include huge numbers of items and users. In addition, memory-based methods are heuristics and the prediction and recommendations are based on the whole ratings provided by the users to the items. Hence, all ratings are required to be maintained in memory. This method is a typical approach for high recommendation accuracy based on CF, but is not scalable for large-scale websites that use the huge number of users and items in recommendation systems (Sarwar, Karypis, Konstan, & Riedl, 2002). According to Goldberg, Roeder, Gupta, and Perkins (2001), model-based methods for learning a model utilize the group selection of ratings which is then applied to provide rating predictions. In addition, model-based CF algorithms have been an alternative approach to *k*-NN to solve the scalability problem of memory-based method (Nilashi, Esfahani et al., 2016c). A probabilistic method is utilized for these systems and the unrated value of a user prediction is measured based on the ratings the user has given to other items. Model-based algorithms do not suffer from memory-based drawbacks and can create prediction over a shorter period of time compared to memory-based algorithms because model-based algorithms perform off-line computation for training. These techniques regularly make a concise rating pattern off-line. Model-based CF (e.g., Singular Value Decomposition (SVD)-based CF) improves the scalability and the efficiency problem (Koren, Bell, & Volinsky, 2009, Liu et al., Nilashi et al., 2015; Nilashi, bin Ibrahim, & Ithnin, 2014), but may lead to some problems such as decreasing the accuracy performance (Linden, Smith, & York, 2003).

Hence, in this study a new method is proposed based on CF method to overcome the sparsity and scalability problems in CF algorithms accordingly to improve the performance of recommender systems. In fact, the performance improvement is achieved using ontology (Shambour & Lu, 2012) and dimensionality reduction (Koren, 2008; Koren et al., 2009) techniques. At the moment, there is no implementation of recommender systems by the use of combining ontology and dimensionality reduction techniques to solve the scalability and sparsity issues of CF recommender systems. Accordingly, this research tries to develop a new recommendation system based on CF using ontology and dimensionality reduction techniques. In order to enhance the prediction accuracy and overcome the scalability issue of recommender systems, we propose to use ontology and SVD. Specifically, we develop the method for user- and item based CF. We use the items' ontology for the item-based semantic similarity calculation and SVD for the item- and user-based CF recommendation part. In comparison with the previous studies, in this research we:

- develop a new recommendation system using ontology and dimensionality reduction techniques.
- improve the accuracy of recommendation systems by alleviating the sparsity issue in item-based CF using ontology.
- improve the scalability of recommendation systems using dimensionality reduction techniques.

The remainder of this paper is organized as follows: In Section 2, we briefly introduce the related subjects for the development of the proposed recommender system. In Section 3, problem statement and our research contributions are presented. Section 4 presents research methodology. Section 5 provides method evaluation results. In Section 6, we provide the discussion. Finally, conclusion is provided in Section 7.

2. Background theories

In the following sub-sections, we present the related subjects for the development of the proposed recommender system. Since, the ontology, clustering, dimensionality reduction and CF are important components of the proposed method, a short introduction of them is presented.

2.1. CF recommendation methods

The recommendation systems generally are divided into three categories: CF, Content-Based Filtering (CBF) and hybrid method. CF techniques in recommender systems are particularly popular and have been applied in many online shopping websites (Liu et al., 2011; Nilashi et al., 2014). The key to successful collaborative recommendation lies in the ability to make meaningful associations between people and their product preferences, in order to assist the end-user in future transactions. Similarities between past experiences and preferences are exploited to form neighbourhoods of like-minded people from which to draw recommendations or predictions for a given individual user. Based on the genuine process of CF strategy (Schafer, Frankowski, Herlocker, & Sen, 2007), a target user in the website will receive recommendation list of items that other users, with similar tastes, liked in the past. All CF methods require the past ratings of users in order to predict and accordingly recommend items to the target user (Cheng & Wang, 2014). To do so, similarities between the users and items are calculated using the distance measures. As CF can be classified as user-based and item-based, accordingly the similarity calculation for these approaches will be different (Nilashi et al., 2014).

2.2. Clustering methods and CF

CF is one of the methods widely used in recommender systems using two different techniques, memory-based and model-based (Breese, Heckerman, & Kadie, 1998; Su & Khoshgoftaar, 2009). The memory-based depends on the entire rating which exists in the user-item matrix for forming neighbors of the active user to generate recommendation tailored to his/her preferences. In contrast, the model-based methods learn the models of recommendations from the entire ratings to generate the recommendation for the target user. The well-known machine learning techniques for this approach is clustering (Sarwar et al., 2002), probabilistic Latent Semantic Analysis (pLSA) (Hofmann & Puzicha, 2004), matrix factorization (e.g. SVD) (Koren et al., 2009) and machine learning on the graph (Zhou et al., 2008).

Since memory-based techniques are easy to understand, implement and be successfully utilized in the real world application, they are considered suitable methods in recommender systems. However, this method often fails in large-scale applications. The sparsity of user-item matrix that is resulted since the user only rates few items throughout a large database of items is one of the issues in this technique that cause this failure. Thus, calculated similarity between users/items is unreliable value because of the few overlapping ratings between them. Efficiency is another issue in memory–based CF because similarity between pairs of items or users is needed to be measured for finding their neighborhood. A line of studies has been conducted for overcoming this drawback of memory-based techniques by a model-based clustering approach for enhancing efficiency (Gong, 2010; He, Yang, & Jiao, 2011; Sadaei, Enayatifar, Lee, & Mahmud, 2016; Shinde & Kulkarni, 2012; Wang, 2012). Clustering method groups similar items or users into separate clusters to identify neighborhood Clustering techniques have been used either directly or as a preprocessing stage in recommender systems (Adomavicius & Tuzhilin, 2005; Gong, 2010; Nilashi et al., 2014; Pham, Cao, Klamma, & Jarke,

2011; Truong, Ishikawa, & Honiden, 2007; Zhang, Zhou, & Zhang, 2011). For example, in the work conducted by Ungar and Foster, the authors developed a statistical model for CF. They accordingly used clustering methods for estimating model parameters (Ungar & Foster, 1998). They also noted that the clustering can solve overgeneralization (Kushwaha and Vyas, 2014) problem in recommender systems. Breese et al. (1998) further investigated the role of clustering for the accuracy problem of recommender systems. They found that the model based method achieves better accuracy in relation to the memory-based approaches (Breese et al., 1998). Furthermore, Sarwar et al. (2002) found that employing bisecting *k*-means leads to less-personal CF recommendations than other methods. Similarly, Linden et al. (2003) tested a clustering method for item-to-item CF system employed at Amazon.com. They found that the use of clustering improves the scalability issue in CF. Xue et al. (2005) experiments also showed that the combination of memory based and model based methods improve the recommendation efficiency by improving the accuracy and solving scalability problem. They used *k*-means clustering to provide smoothing operations to solve the missing-value problems and evaluated the method on MovieLens and EachMovie datasets.

2.3. Ontology method in recommender system

Modeling the information at the semantic level is one of the main goals of using ontologies (Guarino, Oberle, & Staab, 2009). The original definition of ontology in the computer science was provided by Gruber (1992), and was later refined by Staab and Studer (2009). The notion of an ontology is originally defined by Gruber (1992) as an "explicit specification of a conceptualization" . Borst (1997) defines an ontology as a "formal specification of a shared conceptualization" . In addition, Taniar and Rahayu (2006) defined ontology "as a knowledge domain conceptualization into a computer processable format which models entities, attributes, and axioms" . Ontology is typically made up of a vocabulary and relationships between the concepts. According to Antoniou and Van Harmelen, (2004), ontologies are concept properties, disjointness statements, value restrictions, and specification of logical relationships between the objects. Ontology has been a tool to formally model the structure of a system based on the relationships which are emerged from its observation.

In recommender systems, the semantic information of an item includes the attributes, the relationships among the items, and the relationship between meta-information and items. In recent years, ontologies have been successfully adopted in recommender systems for overcoming the shortcomings of these systems (Martín-Vicente, 2014; Lopez-Nores et al., 2010). Porcel, Martinez-Cruz, Bernabé-Moreno, Tejeda-Lorente, and Herrera-Viedma (2015), focused on the accuracy improvement of recommender systems by incorporating fuzzy ontology in their method. Many researchers involve domain ontologies in the recommender systems to in measuring the preferences of users to the items of the content (Middleton, De Roure, & Shadbolt, 2009). Some researchers develop the semantic recommendation approach using with combining item-based CF and item-based semantic similarity techniques. In Daramola, Adigun, and Ayo (2009), authors develop a prototype e-tourism recommendation system using ontology for tourism services. In Wang and Kong (2007)), authors propose a semantic enhanced collaborative recommendation system using the usage data and semantic information. Moreover, using knowledge about items and users help to produce a recommendation based on knowledge and reasoning about which item meet the needs of users (Trewin, 2000). The present study aims to use a hybrid method based on knowledge.

3. Related work

Before giving details of the techniques incorporated in the proposed method and our experimental evaluation, in this section we summarize other existing approaches of recommender systems in the literature.

Lee and Olafsson (2009) proposed a cooperative prediction scheme for CF recommender systems. They evaluated the method on EachMovie and MovieLens datasets. De Campos, Fernández-Luna, Huete, and Rueda-Morales (2010) presented a new Bayesian network model to deal with the problem of hybrid recommendation by combining content-based and collaborative features. They used MovieLens and The Internet Movie Database (IMDB) data sets to show the effectiveness of the method. Fan et al. (2014) developed a recommendation method of user-based CF based on predictive value padding. Their method predicts the empty values in user-item matrix by the integration of content-based recommendation algorithm and user activity level before calculating user similarity. They used MovieLens dataset to show the accuracy improvement of the method.

Shambour and Lu (2012) developed a recommendation method to solve the sparsity and cold-start issues. They incorporated additional information from the users' social trust network and the items' semantic domain knowledge for improving the recommendation accuracy and coverage. They used Yahoo! Webscope R4 and MovieLens datasets for their experiments. The results of their study showed that the method is effective in solving the sparsity and cold-start issues. Tsai and Hung (2009) used clustering ensembles for CF and content based recommendation methods. They used *k*-means and Self-Organizing Maps (SOM) for clustering task. They used MovieLens dataset for their experiments and showed that the ensembles of clustering methods outperform the recommendation methods which do not rely on ensemble learning. Wu, Chang, and Liu (2014) developed a hybrid approach that combines content-based approach with CF under a unified model called Co-Clustering with Augmented Matrices (CCAM). They evaluated the method on two MovieLens datasets (100 k and 1 M datasets). They showed that content-based information can help reduce the sparsity problem through minimizing the mutual information loss of the three data matrices based on CCAM.

Lee, Chun, Shim, and Lee (2006) developed an ontology-based product recommender system for Business-to-Business (B2B) marketplaces. Their method was keyword-based and independent of the underlying physical structure of product ontology. Specifically, their method was based on content-based recommendation technique which represented product data in ontological graphs. Zhuhadar et al. (2009) developed a multi-model ontology-based framework for semantic search of educational content in e-learning context. They combined the content-based with the rule-based approaches to provide the user the hybrid recommendations. They evaluated the method using Top-N precision and Top-N recall metrics.

Liao, Kao, Liao, and Chen (2009) implemented a library recommender system to provide service for the users of National Chung Hsing University (NCHU) in Taiwan. They used ontology for improving the prediction accuracy of the recommender system. Specifically, the Classification for Chinese Libraries (CCL) (Liao, Liao, Kao, & Harn, 2006) was adopted as reference ontology. Liao, Hsu, Chen, and Chen (2010) developed a recommendation system by incorporating CF techniques with the Personal Ontology Model of PORE to recommend English collections. They used Dewey Decimal Classification (DDC) as the reference ontology to build a personal ontology for each patron. Moreno, Valls, Isern, Marin, and Borràs (2013) used ontology-based model for the construction and exploitation of user profiles. The author developed a web-based

system, SigTur/E-Destination, to provide personalized recommendations of touristic activities in the region of Tarragona.

Hawalah and Fasli (2014) utilized contextual ontological user profiles for personalized recommendations. Also, they developed a method to compute semantic relatedness between concepts in rich and complex ontological structures. They conducted a user-centered study to assess the effectiveness of the recommendations by the method and used precision metric for the method evaluation. Martinez-Cruz, Porcel, Bernabé-Moreno, and Herrera-Viedma (2015) developed a recommender system by incorporating ontologies to improve the representation of user profiles. Specifically, they used fuzzy linguistic modeling to facilitate the representation of different concepts. The proposed method could able to reveal the relationships between users and their preferences about the items by incorporating domain ontology to the system. The authors used Mean Absolute Error (MAE) and coverage metrics to evaluate the method. Al-Hassan, Lu, and Lu (2015) used semantic knowledge of items to enhance the recommendation quality. Accordingly, they developed a hybrid semantic enhanced recommendation method by combining the Inferential Ontology-based Semantic Similarity (IOBSS) measure and the standard item-based CF approach. They evaluated the method on Australian tourism services using MAE metric with ten-fold cross validation technique.

Lv, Hu, and Chen (2016) developed a recommendation system based on relational data using the relational data in the domain ontology. They used genetic algorithm for recommendation process. They used MovieLens dataset for the method evaluation using recall, diversity and precision metrics. Pham, Jung, Nguyen, and Kim (2016) proposed an ontology-based multilingual recommendation system for the movie domain. The aim of their work was to discover the relationships among multilingual concepts for searching on a movie domain and ontological user preferences. Celdrán, Pérez, Clemente, and Pérez (2016) developed a hybrid recommender system that combined content-based, CF, and context-aware approaches. In addition, they used semantic web techniques to model the information of the recommender system. Specifically, the recommender ontology was defined with the Ontology Web Language 2 (OWL 2). They evaluated the method on MovieLens dataset using F1, recall and precision metrics. Moreno, Segrera, López, Muñoz, and Sánchez (2016) proposed a complete framework to deal jointly with the scalability, sparsity, first rater and cold start problems with combining web mining methods and domain specific ontologies. They evaluated the method on MovieLens dataset using precision metric.

Bassiliades, Symeonidis, Meditskos, Kontopoulos, Gouvas, and Vlahavas (2017) developed recommendation algorithm using ontology for providing the application developer with recommendations about the best matching Cloud Platform as a Service (PaaS) offering. The results of their work demonstrated that the method was effective in solving scalability issue. Tarus, Niu, and Yousif (2017) proposed a hybrid knowledge-based recommender system based on ontology and Sequential Pattern Mining (SPM) for recommendation of e-learning resources to learners. The researcher used SPM to discover the learners' sequential learning patterns and ontology to model and represent the domain knowledge about the learner and learning resources. Kermany and Alizadeh (2017) proposed a recommender system using Adaptive Neuro-Fuzzy Interference System (ANFIS) for multi-criteria recommender systems by incorporating demographic information of users and ontological item-based semantic information. They evaluated the method using the data from the Yahoo!Movies platform by F1, MAE and precision metrics. The results of their work revealed that the use of semantic information enhances the predictive accuracy of multi-criteria recommender systems.

4. Proposed hybrid recommender system

In Fig. 1, the proposed recommender system is presented. The proposed method aims to produce accurate and scalable recommendations. Two main phases are considered for the method. In the first phase, the recommendation models are constructed. In this phase several tasks are performed which are clustering the rating, dimensionality reduction using SVD and producing the similarities matrices of the items and users. In the first step, we cluster the users' ratings on movies using Expectation Maximization (EM) algorithm. Then for each cluster we provide the semantic similarity calculation matrices from the movie ontology repository.

Meanwhile, on each cluster we perform SVD to obtain the decomposition matrices. From the figure, it can be seen that we develop the SVD models for users and items. Hence, after matrices decomposition task, the similarities calculations can be effectively performed on each matrix. In the second phase, after performing initial trains of the models in the offline phase, the prediction and accordingly recommendations tasks are performed for a given user (target user). In fact, a ranked list of items is provided to be recommended by the recommender system to the target user. To do so, the target user is assigned to one of the clusters determined in the first phase. Then SVD calculation is performed based on the past ratings to find the target user similarities to the other users (finding the neighbors of the target user). For item-based recommendation, we also perform same procedure for the items. We finally combine user- and item-based predictions in a weighted approach as presented by Liu, Hu et al. (2014).

Ontology. The movie ontology is constructed using the Movie Ontology (MO) which can be accessed through (http://www.movieontology.org/) (see Fig. 2). MO semantically describes movie related concepts. In addition, we used MO to establish a correspondence between classes in the ontology and database genres. Movie' genre is a multi-valued attribute whereas origin country is a mono-valued attribute (Ticha et al., 2012). The URL of the movie in IDMb (The Internet Movie Database) is a unique key that can show every movie item in the system. By implementing a Web Crawler (see Fig. 3) and using these unique keys, the system can extract content information from the IMDb for each movie item. This content information is saved in the database so that they can be used for generating ontology-based metadata. In fact, the web crawler analyzes the IMDb web pages based on the predefined features of each movie and extracts feature-values. Each extracted feature-value belongs to a feature.

Calculating the similarity between the two items based on their semantic descriptions is an important task. In this research, we use binary Jaccard similarity coefficient for the item-based semantic similarity (Kermany & Alizadeh, 2017; Shambour & Lu, 2012). To do so, for an item taxonomy $\aleph$ with m categories that items may fall into, we consider each item as a binary vector ($\vec{E} = (e_{x,1}, e_{x,2}, \ldots, e_{x,m})$ where a binary variable $e_{x,p}$ ($p = 1, \ldots, m$) is defined as:

$$e_{x,p} = \left\{ \begin{array}{l} 1 \text{ If Item } \times \text{ belong to category p} \\ 0 \text{ If Item } \times \text{ does not belong to category p} \end{array} \right\} \quad (1)$$

Then, the semantic similarity of movies x and y is presented as follow (Kermany & Alizadeh, 2017; Shambour & Lu, 2012):

$$SemSim(x, y) = \frac{K_{11}}{K_{01} + K_{10} + K_{11}} \quad (2)$$

In Eq. (2), K_{01}, K_{10} and K_{11} respectively indicate the total number of genres for $(e_{x,j} = 0;\ e_{y,j} = 1), (e_{x,j} = 1;\ e_{y,j} = 0)$ and $(e_{x,j} = 1;\ e_{y,j} = 1)$.

Clustering. We consider both content-based features and user rating data for clustering since considering only one of them will lead to low accuracy, overgeneralization, and overlapping of the

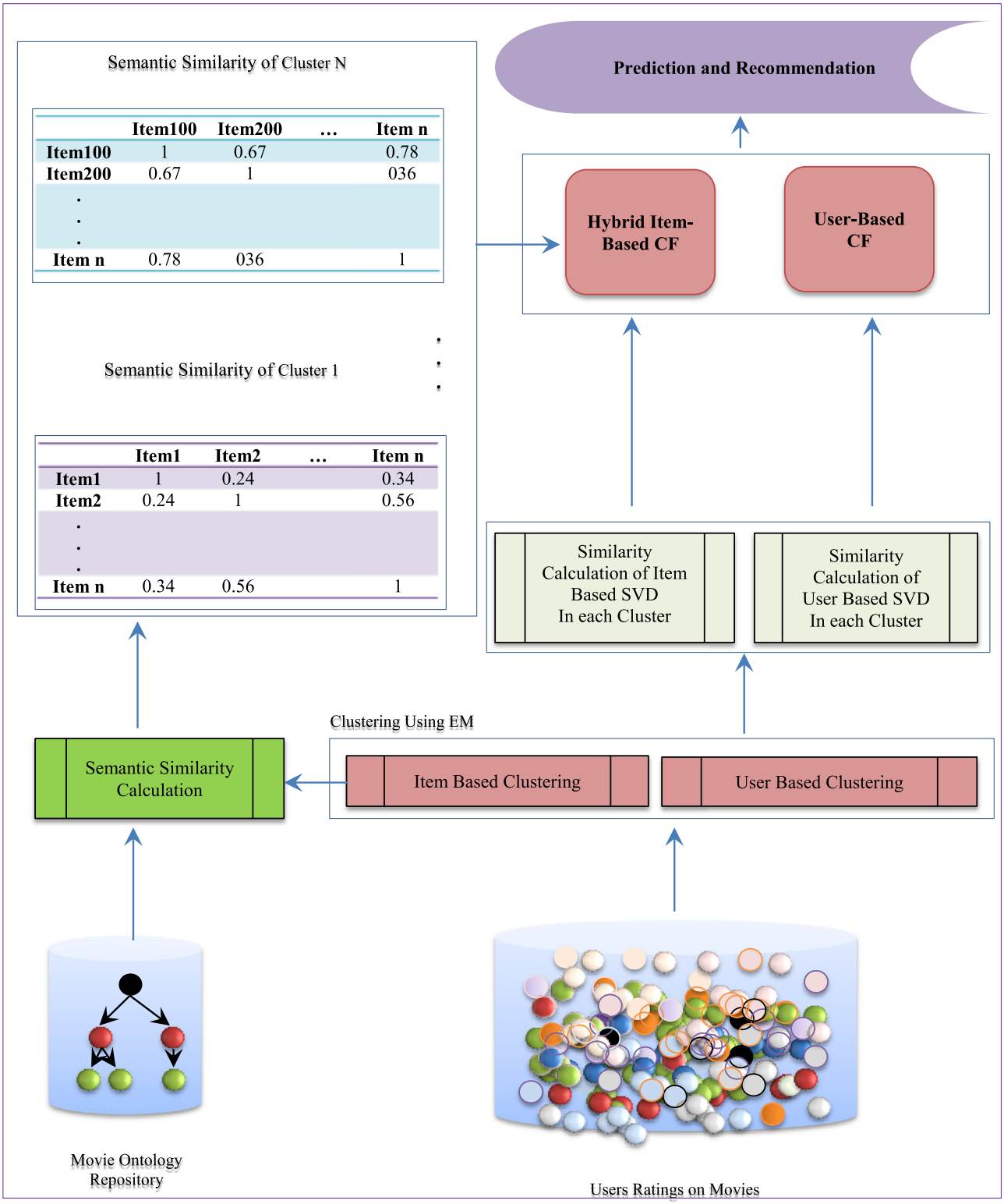

Fig. 1. System Framework.

clusters. In this study, we use EM clustering algorithm. EM algorithm, proposed by Dempster, Laird, and Rubin (1977), has been widely used in the prior studies as an unsupervised learning method. In Algorithm 1, the EM algorithm is presented. From the EM algorithm, it can be seen that EM is mainly divided into two main steps which are E-step and M-step. In the E-step of EM algorithm, EM proceeds by estimating to which component each data point belongs. In M-step, EM proceeds re-estimating the parameters on the basis of the estimation in E-step. Hence, after each iteration of EM, it is guaranteed that the re-estimated parameters give at least as high a log-likelihood as the previous parameter values. In the last step of EM, we should check for convergence. In

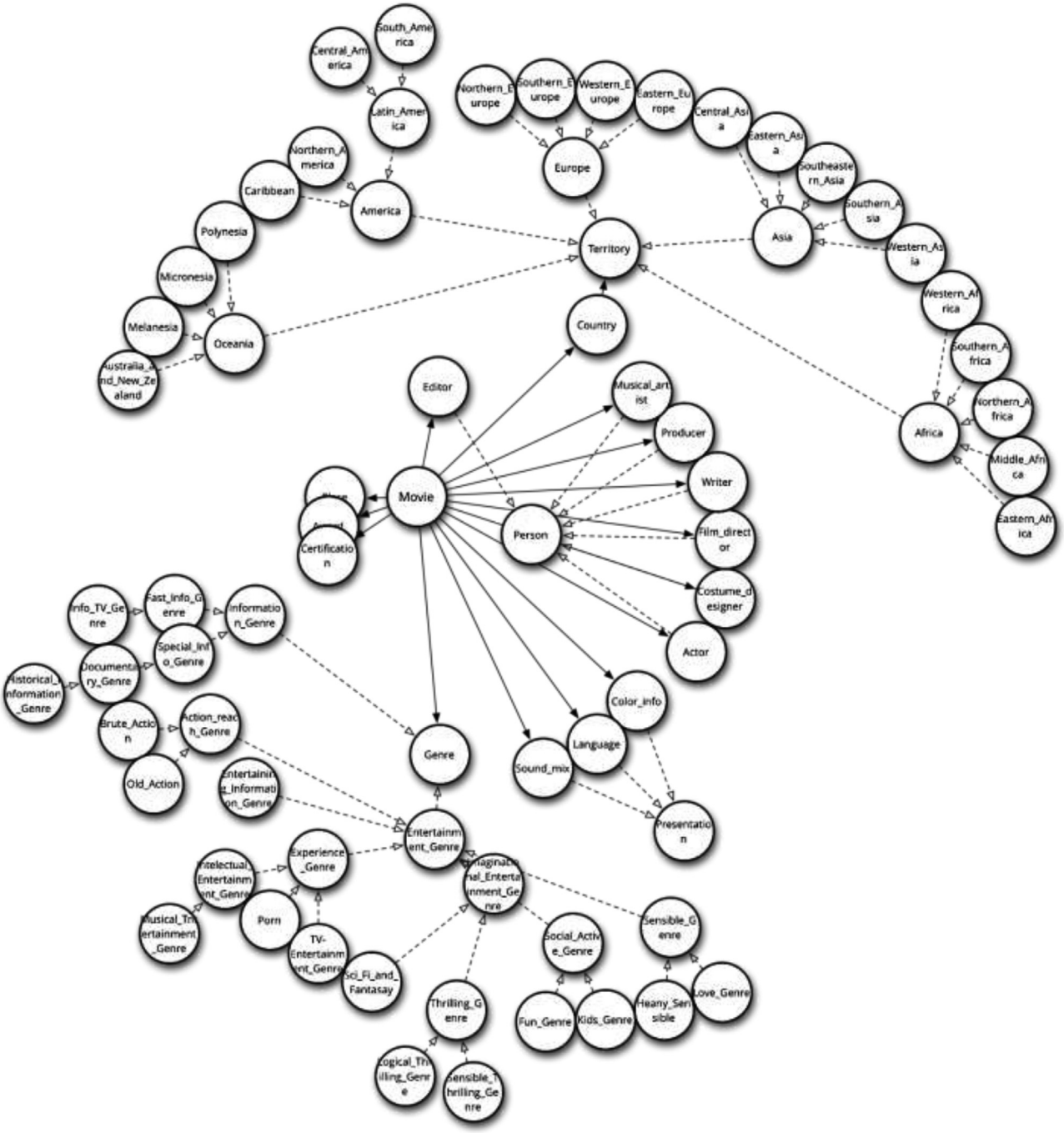

Fig. 2. Overview of the movie ontology domain.

addition, E-step and M-step need to be repeated until convergence is met which in EM this can be performed based on convergence of the parameters or on the log likelihood function.

A. User Clustering

In user clustering, users are clustered based on similar preferences according to their rating. After creating the clusters, the aggregation of opinions in each cluster is used to perform the prediction task for the target user. Thus, it results in improving performance since the cluster that should be analyzed includes much fewer users compared to the number of all users (since the size of the group that must be analyzed is much smaller) (Gong, 2010; Sarwar et al., 2002).

In Fig. 4a, m indicates the number of all users, a_{ij} is the average rating of user cluster center i given to item j, R_{ij}defines the rating that has been provided by user i for item j, and n and c respectively denote the number of all items and the number of user centers.

B. Item Clustering

In item clustering, items are clustered based on similar ratings provided by users. After creating the clusters, the aggregation of opinions of the other items in any clusters is used for prediction task for the target item. Thus, it results in improving performance since the cluster that should be analyzed includes much fewer items compared to the number of all items (Gong, 2010).

In Fig. 4b, m denotes the number of all users, $a_{ij}$is the average rating of user i to *item cluster center* j, n implies the number of

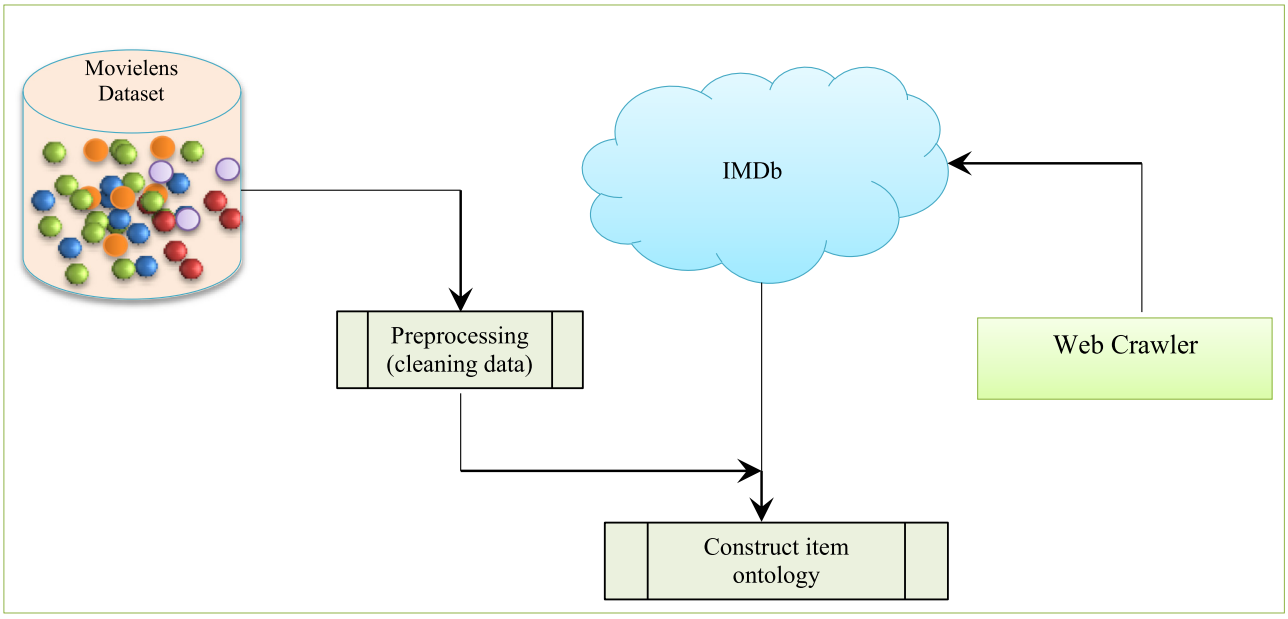

Fig. 3. Constructing ontologies.

Algorithm 1 EM Algorithm.

Variables
z is an unknown hidden variable.
μ_j means of the model.
π_j distribution function.
Σ_j variances of the model.
$\{x_i\}_{i=1}^{N}$ dataset with N data points.
z a random variable.
$p(x|z=j) \sim N(\mu_j, \sigma_j I_j)$ a Gaussian distribution. $l(\theta, D)$ likelihood function.
Steps of EM
Initialize: Initialize means and variances of the model $\{\mu_j^{(0)}, \sum_j^{(0)}, \pi_j^{(0)}\}$.
Step 1. Expectation: Using the estimates of $\theta^{(t)} = \{\mu_j^{(t)}, \sum_j^{(t)}, \pi_j^{(t)}\}$, parameters compute the estimate of w_{ij}

$$w_{ij}^{(t)} = p(z=j|x_i, \theta^{(t)}) = \frac{\pi_j^{(t)} p(x_i|z_i=j, \theta^{(t)})}{\sum_{m=1}^{k} \pi_m^{(t)} p(x_i|z_i=k, \theta^{(t)})}$$

Step 2. Maximization: Using estimates of $w_{ij}^{(i)}$, update the estimates of the model parameters

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(t)} x_i}{\sum_{i=1}^{N} w_{ij}^{(t)}}$$

$$\sigma_j^{(t+1)} = \frac{\sum_{i=1}^{N} w_{ij}^{(t)} ||x_i - \mu_i||^2}{\sum_{i=1}^{N} w_{ij}^{(t)}}$$

$$\pi_i^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} w_{ij}^{(t)}$$

Step 3. Check for convergence: This can be performed based on convergence of the parameters or on the log likelihood function. If the convergence criterion is not satisfied return to Step 1.

all items,R_{ij} indicates the rating of user i to *item* j, and k is the number of item centers.

Singular Value Decomposition. SVD (Golub & Reinsch, 1970) has been one of the robust data dimensionality reduction techniques in real matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a powerful computation tool for solving data analysis problems in numerical linear algebra. Using SVD a matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$ with the rank of $r \leq min(N, M)$, can be decomposed as: $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$where,

$$\mathbf{U} = \left[\underbrace{u_1, u_2, ..., u_r}_{\mathbf{U}_r}, \underbrace{u_{r+1}, ..., u_N}_{\mathbf{U}_r} \right] = [\mathbf{U}_r|\mathbf{U}_r] \tag{3}$$

$$\mathbf{V} = \left[\underbrace{v_1, v_2, ..., v_r}_{\mathbf{V}_r}, \underbrace{v_{r+1}, ..., v_M}_{\mathbf{V}_r} \right] = [\mathbf{V}_r|\mathbf{V}_r] \tag{4}$$

$$A = [\mathbf{U}_r|\mathbf{U}_r] \begin{bmatrix} \mathbf{\Sigma}_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^T \\ \mathbf{V}_r^T \end{bmatrix} = \mathbf{U}_r \mathbf{\Sigma}_r {\mathbf{V}_r}^T \tag{5}$$

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i u_i u_i^T = u_1 v_1^T \sigma_1 + u_2 v_2^T \sigma_2 + u_3 v_3^T \sigma_3 + u_4 v_4^T \sigma_4 + \ldots + u_r v_r^T \sigma_r, \quad rank(\mathbf{A}) = r$$

$$\mathbf{B} = \sum_{i=1}^{k} \sigma_i u_i u_i^T = u_1 v_1^T \sigma_1 + u_2 v_2^T \sigma_2 + u_3 v_3^T \sigma_3 + u_4 v_4^T \sigma_4 + \ldots + u_k v_k^T \sigma_k \quad rank(\mathbf{B}) = k$$

$$\mathbf{A} - \mathbf{B} = \mathbf{U}\mathbf{\Sigma}_\mathbf{A}\mathbf{V}^T - \mathbf{U}\mathbf{\Sigma}_\mathbf{B}\mathbf{V}^T = \mathbf{U}^T[\mathbf{\Sigma}_\mathbf{A} - \mathbf{\Sigma}_\mathbf{B}]\mathbf{V}^T \rightarrow ||\mathbf{A} - \mathbf{B}|| = \sigma_{k+1} \tag{6}$$

Definition 1. Let $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be a SVD of $\mathbf{A} = \mathbb{R}^{m \times n}$ and $r = \text{rank}(\mathbf{A})$. If for $k < r$ define $\mathbf{A}_k = \sum_{j=1}^{k} u_j \sigma_j, \quad u_j^T$, then

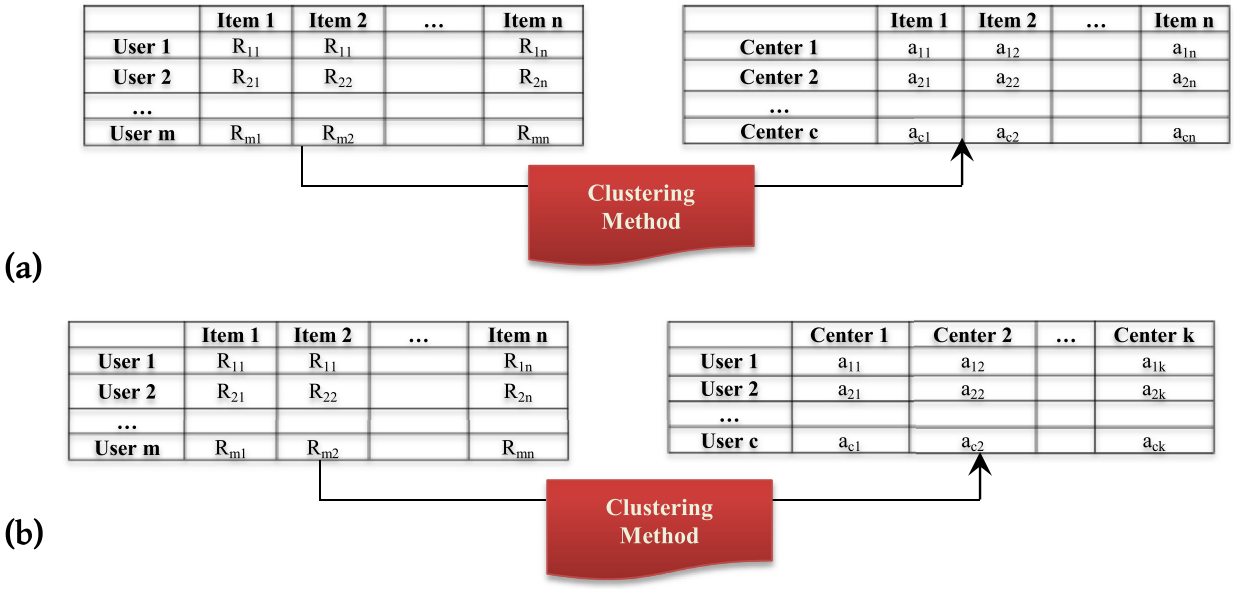

Fig. 4. Forming (a) User Clusters in CF Based and (b) Item Clusters in CF Based.

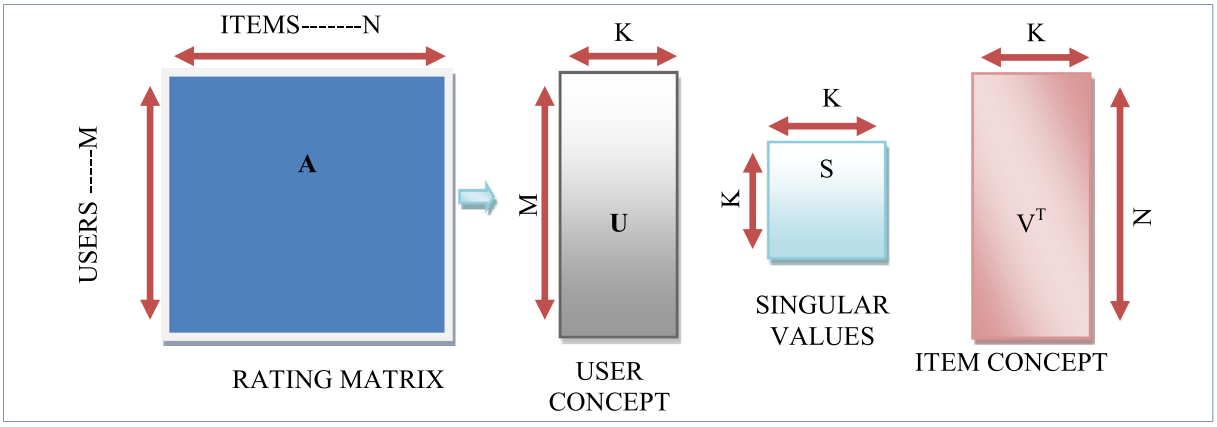

Fig. 5. Illustrating the basic SVD theorem.

$\|\mathbf{A}-\mathbf{A}_k\|_F = \min_{rank\ (\mathbf{B})\le k} \|\mathbf{A}-\mathbf{B}_k\|_F = \sqrt{\sum_{j=k+1}^{r}\sigma_j^2}$, where $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j}|x_{ij}|^2}$ is the Frobenius-norm of a matrix **X**.

According to the nature of SVD and its linearity, it is possible that we apply it on a matrix which has two dimensions. Fig. 5 shows the general procedure of SVD for dimensionality reduction in user-item matrix in CF that **A** implies the rating matrix of user to items, **U** refers to user concepts matrix, **S** indicates singular values and $\mathbf{V}^T$ that is a reprehensive of item concepts. Therefore, using SVD algorithm, it is possible to convert a given matrix **A** into $\mathbf{A} = \mathbf{USV}^T$.

Therefore, the matrix **A** can be decomposed with rank *r* and by considering the matrix **A** with rank *k*, it can be obtained the matrix **B** which gives the approximation of **A** based on defined *k*. In addition, in CF the user-item matrix for example in Table 1 can be reduced in two dimensions as shown in Fig. 6 to get latent relationships between its objects. In the following example, the ratings are presented in Table 1. Thus, **U, V** and **S** can be calculated as follows:

Table 1
User-item matrix.

	Item 1	Item 2	Item 3	Item 4
User 1	5	5	0	5
User 2	5	0	3	4
User 3	3	4	0	3
User 4	0	0	5	3
User 5	5	4	4	5
User 6	5	4	5	5

$$U = \begin{pmatrix} 0.45 & 0.54 & 0.01 & 0.50 & -0.50 & 0.11 \\ 0.36 & -0.25 & -0.86 & 0.15 & 0.21 & 0.06 \\ 0.29 & 0.40 & 0.23 & 0.10 & 0.83 & 0.08 \\ 0.21 & -0.67 & 0.40 & 0.59 & 0.07 & 0.02 \\ 0.51 & -0.06 & 0.11 & -0.29 & -0.07 & -0.80 \\ 0.53 & -0.19 & 0.19 & -0.53 & -0.14 & 0.58 \end{pmatrix} \cdots$$

$$V = \begin{pmatrix} 0.57 & 0.22 & -0.67 & -0.41 \\ 0.43 & 0.52 & 0.69 & -0.26 \\ 0.38 & -0.82 & 0.25 & -0.33 \\ 0.59 & -0.05 & -0.01 & 0.81 \end{pmatrix} \cdots$$

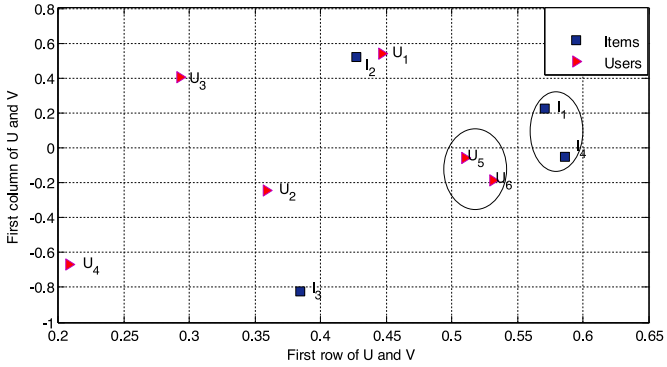

Fig. 6. Two-dimensional space of applying SVD for users and items.

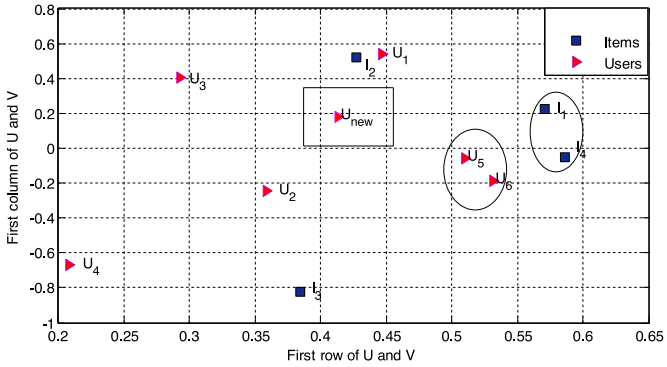

Fig. 7. Two-dimensional space of applying SVD for users, items and new user.

$$S = \begin{pmatrix} 17.71 & 0.00 & 0.00 & 0.00 \\ 0.00 & 6.39 & 0.00 & 0.00 \\ 0.00 & 0.00 & 3.10 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.33 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

For a better understanding of this decomposition, the result of applying SVD on user rating matrix is projected in 2-dimensional space and plotted in Fig. 6. The first column and the second column of **U** is treated as x and y, respectively. In addition, for matrix **V** the first column and the second column is treated as x and y, respectively. To do this, only two singular values 17.71 and 6.39 of matrix **S** are selected as shown in Fig. 6.

In Fig. 7, it is clear that the users 5 and 6 and items 1 and 4 are located very close to each other. Therefore, SVD for dimensionality reduction can be applied effectively to reveal the users that have similar taste and form neighbors for items and users. Furthermore, this decomposition of rating matrix significant reduces the computational complexity for users and items similarities calculation as the first columns of matrices are considered.

For giving similar users and recommendation to the new user, consider new user shares his/her rating as ([5,3,3,2] for items 1–4). For finding the position of new user in the 2 dimensions space, the following calculation is performed as:

$$\mathbf{Newuser}_{2D} = \mathbf{Newuser}^{T} \times \mathbf{V}_2 \times \mathbf{S}_2^{-1}$$

$$\mathbf{Newuser}_{2D} = \begin{bmatrix} 5 \\ 5 \\ 3 \\ 2 \end{bmatrix} \times \begin{bmatrix} 0.57 & 0.22 \\ 0.43 & 0.52 \\ 0.38 & -0.82 \\ 0.59 & -0.05 \end{bmatrix} \times \begin{bmatrix} 17.71 & 0.00 \\ 0.00 & 6.39 \end{bmatrix}^{-1}$$

$$\rightarrow \mathbf{Newuser}_{2D} = [0.4132 \quad 0.1784]$$

Algorithm 2 SVD in the prediction task.

Step 1: The user-item matrix $R^{m,n}$ with raw data is converted to the dense matrix $\mathbf{B}^{m,n}$.

Step 2: Matrix $\mathbf{B}^{m,n}$ is normalized using Z-score to the matrix $\mathbf{Z}^{m,n}$ by

$$\mathbf{Z}_{ij} = \frac{B_{ij} - \bar{B}^j}{\sigma_j},$$

where $\bar{B}^j$ and σ_j indicate average value and Standard Deviation (SD) for the ratings in the B^j, respectively, that

$$\bar{B}^j = \frac{1}{m}\sum_{i=1}^{m} B_{ij},\ \sigma_j^2 = \frac{1}{m-1}\sum_{i=1}^{m}(B_{ij} - \bar{B}^j)^2$$

Step 3. The SVD method is applied on Z.

Step 4. An approximation of Z is calculated as $\mathbf{Z}_d$.

Step 5. P_{ij} is calculated based on $\bar{B}^j + \sigma_j(\mathbf{Z}_d)_{ij}$.

As can be seen in Fig. 7, by calculation cosine-based similarity, it can found the users close to the new user for forming k-nearest neighbors. This method can be applied for items and we can form the similar items (item-based similarity).

The SVD approach approximates the missing rating values based on the matrix factorization ($\hat{r} = (\mathbf{U}_k\mathbf{S}_k{}^{1/2})_u.(\mathbf{S}_k{}^{1/2}\mathbf{V}'_k)_i)$). The following steps are done by an example to estimate an unknown rating to the active user. Let $\mathbf{Y} = \{a_{ij}\} \in \mathbb{R}^{m,n}$ be the user-item matrix contains the ratings users $U = \{u_1, u_2, \ldots, u_m\}$ to the items $I = \{i_1, i_2, \ldots, i_n\}$. The goal is to predict unknown ratings in this matrix. The algorithm for this task is presented in Algorithm 2. As an example, let we have the rating matrix **R,** thus p_{23} can be calculated using the Algorithm 2:

$$\mathbf{R} = \begin{array}{c} \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{array} \begin{array}{c} \begin{array}{cccccc} I_1 & I_2 & I_3 & I_4 & I_5 & I_6 \end{array} \\ \begin{pmatrix} 2 & 3 & 5 & ? & 1 & 4 \\ ? & 3 & ? & 4 & ? & 3 \\ 3 & 5 & ? & 2 & 4 & ? \\ 3 & 3 & 4 & ? & 3 & ? \end{pmatrix} \end{array}, \quad p_{23} = ? \quad (\text{Calculating } \bar{B}^i)$$

$$\xrightarrow{Step\,2} \begin{pmatrix} \bar{B}^1 \\ \bar{B}^2 \\ \bar{B}^3 \\ \bar{B}^4 \\ \bar{B}^5 \\ \bar{B}^6 \end{pmatrix} = \begin{pmatrix} 2 \\ 3.5 \\ 2.25 \\ 1.5 \\ 2 \\ 1.75 \end{pmatrix}, \quad (\text{Calculating } \sigma_i)$$

$$\xrightarrow{Step\,2} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} 1.41 \\ 1.00 \\ 2.66 \\ 1.91 \\ 1.83 \\ 2.06 \end{pmatrix} \xrightarrow{Step\,3(Z-scores)} \mathbf{Z_{4\times 6}}$$

$$= \begin{pmatrix} 0.00 & -0.50 & 1.04 & -0.79 & -0.55 & 1.09 \\ -1.42 & -0.50 & -0.85 & 1.31 & -1.10 & 0.61 \\ 0.71 & 1.50 & -0.85 & 0.26 & 1.10 & -0.85 \\ 0.71 & -2.00 & 0.66 & -0.79 & 0.55 & -0.85 \end{pmatrix}$$

$$\xrightarrow{Step\,4(Z_d, d=2)} \begin{pmatrix} 0.05 & 0.45 \\ -0.05 & -0.71 \\ -0.89 & -0.18 \\ -0.45 & 0.50 \end{pmatrix} \times \begin{pmatrix} 3.17 & 0.00 \\ 0.00 & 2.86 \end{pmatrix}$$

$$\times \begin{pmatrix} -0.32 & 0.31 \\ -0.67 & -0.05 \\ 0.32 & 0.72 \\ 0.09 & -0.58 \\ -0.38 & 0.20 \\ 0.44 & -0.08 \end{pmatrix}$$

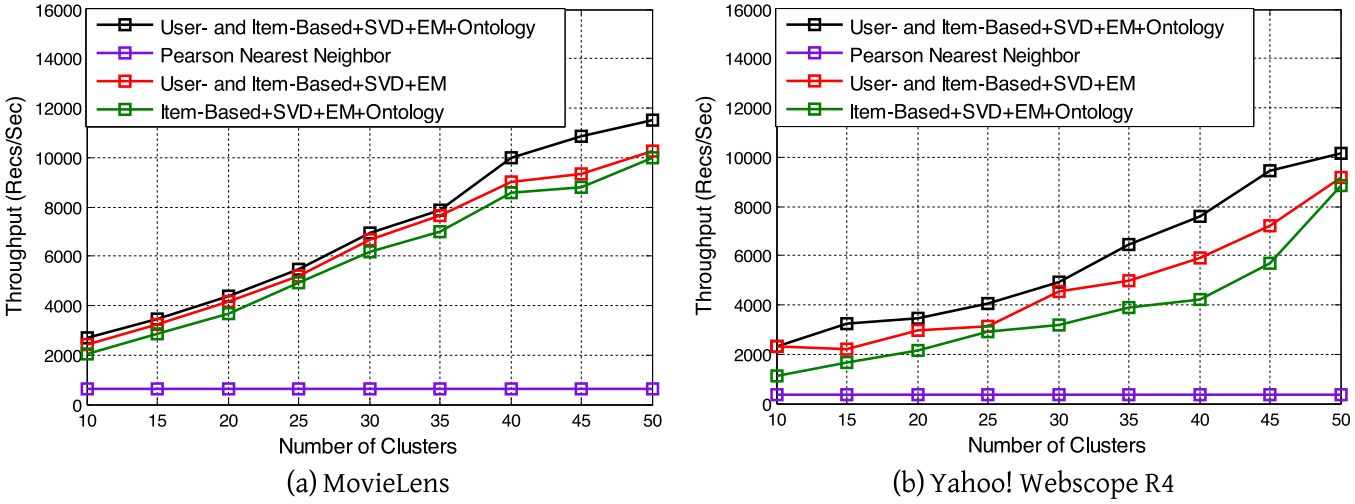

Fig. 8. Throughput of all methods.

$$= \begin{pmatrix} -0.17 & -1.12 & 0.63 & -0.37 & -0.45 & 0.39 \\ -1.35 & 0.01 & -0.47 & 0.99 & -1.24 & 1.02 \\ 0.68 & 1.60 & -0.75 & 0.21 & 1.04 & -0.87 \\ 0.81 & -1.37 & 1.12 & -1.20 & 0.39 & -0.30 \end{pmatrix}$$

$$\xrightarrow{Step\,5} p_{23} = \bar{B}_3 + \sigma_3(\mathbf{Z}_d)_{23} \approx 1$$

5. Experimental results

In this section, the proposed recommender system is evaluated on two real-word datasets. Accordingly, we present the results and compare our method with state-of-the-art recommendation methods.

5.1. Dataset description

To evaluate the proposed method, two datasets, MovieLens and Yahoo! Webscope R4, were considered. The data were cleaned prior to use in the evaluation process. The descriptions of the datasets are as follows:

MovieLens dataset: This dataset (http://www.movieLens.org) is one of the well-known movie datasets that has been used for the evaluation of recommender systems. The numbers of users and movies in the Movielens dataset are 6040 and 3952, respectively. In this dataset, the users have provided ratings on a 5-star scale. We select the users in the dataset who have provided at least 20 ratings. Hence, based on the number of users and movies, this dataset includes 1000,209 anonymous ratings.

Yahoo! Webscope R4 dataset: This dataset (http://webscope.sandbox.yahoo.com) was provided by the Yahoo! Research Alliance Webscope program. In this dataset, the users have provided ratings on a 5-star scale (1 to 5). This dataset is divided into two sets of data, a training set and a test set. The training set includes 7642 users, 11,915 movies and 211,231 ratings. The testing set includes 2309 users, 2380 movies and 10,136 ratings.

In this study, the data collection of items (movies) content is made from the IMDb (http://imdb.com). To do so, we use a Web crawler WebSPHINX available in (http://cs.cmu.edu/~rcm/websphinx/) which has been developed by Rob Miller at Carnegie Mellon University. The collected data is used for constructing and completing item ontology. Furthermore, for testing the model the dataset was split into two groups as training and testing sets. We randomly selected 80% of data for the training set and 20% rest of data for testing set.

5.2. Evaluating the recommender system

The proposed recommender system were developed using MATLAB 7.10 (R2010a) under a 4 GHz processor PC, 8GB RAM and 32-bit Microsoft Windows 7. The proposed algorithm is compared and evaluated with CF recommendation engine that employs the Pearson nearest neighbor algorithm, item-based prediction method with clustering, SVD and ontology, and user- and item-based prediction methods with clustering and SVD but with no contribution of ontology from two perspectives including time throughput (recommendation per second) and accuracy.

Throughput. To show the effectiveness of the proposed method in improving the scalability issue, we evaluate our method on MovieLens and Yahoo! Webscope R4 datasets for throughput which is defined as the number of recommendation per second. In Figs. 8a-b we present the performance results of our experiments for all methods. The throughput of the methods is plotted as a function of the cluster size. We use EM for those methods based on the clustering. We also consider different clustering size for the methods. From the plots we can see that the throughput of those methods that use clustering and dimensionality reduction techniques is substantially higher other methods. In addition, the method which uses the ontology, EM and SVD has a higher throughput than the methods which solely rely on nearest neighbor algorithm for all datasets. This is due to the fact that with the use of clustering a fraction of neighbors is used by the recommendation algorithms. In addition, from these figures it can be found that with the increase of clustering size the throughput of the methods are increased, however as the nearest neighbor algorithm has to scan through all the neighbors, the number of clusters has no impact on its throughput.

Predictive accuracy. With statistical metrics, for example, the MAE between the predicted and the actual ratings is measured. In contrast, decision-support metrics compare the recommended items with the relevant ones, e.g. by counting the overlap. MAE is presented in Eq. (7).

$$\mathbf{MAE}(pred, act) = \sum_{i=1}^{N} \left| \frac{pred_{u,i} - act_{u,i}}{N} \right| \tag{7}$$

where N is the number of items on which a user u has expressed an opinion.

We evaluated the proposed method using MAE for predictive accuracy and compared it with Pearson nearest neighbor algorithm, item-based prediction method with clustering, SVD and on-

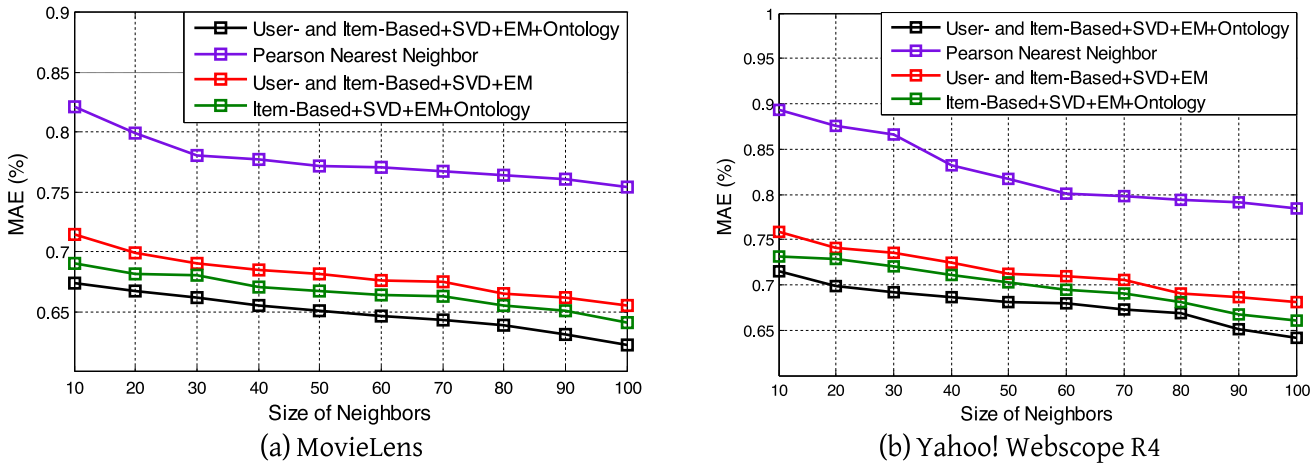

Fig. 9. MAE for all methods on different size of neighbors.

Table 2
F1 and precision for all method on different numbers of Top-N (MovieLens dataset).

Top-*N*	Method A		Method B		Method C		Method D	
	Precision	F1-metric	Precision	F1-metric	Precision	F1-metric	Precision	F1-metric
Top-5	0.787	0.797	0.771	0.773	0.719	0.721	0.564	0.583
Top-10	0.796	0.807	0.782	0.784	0.736	0.739	0.582	0.601
Top-15	0.816	0.827	0.802	0.804	0.747	0.749	0.592	0.615
Top-20	0.821	0.833	0.809	0.811	0.757	0.760	0.601	0.622
Top-25	0.833	0.844	0.823	0.825	0.769	0.770	0.628	0.650
Top-30	0.831	0.840	0.819	0.821	0.757	0.762	0.603	0.605
Top-35	0.823	0.832	0.806	0.808	0.750	0.751	0.581	0.590
Top-40	0.819	0.830	0.801	0.803	0.739	0.741	0.573	0.579
Top-45	0.818	0.827	0.793	0.795	0.733	0.732	0.556	0.558
Top-50	0.813	0.822	0.783	0.785	0.723	0.722	0.541	0.546

Method A = User- and Item-based + SVD + EM + Ontology, Method B = Item-based + SVD + EM + Ontology, Method C = User- *and Item-based + SVD + EM,* Method D = *Nearest Neighbor*.

tology, and user- and item-based prediction methods with clustering and SVD but with no contribution of ontology. Similar to previous studies (Liu, Hu et al., 2014; Koren, 2008), we consider different numbers of neighbors (k) for this evaluation (k = 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100). Figs. 9a-b show the prediction accuracy for different neighborhood size k on datasets. For MAE, it can be found that in the considered neighbor sizes, our method which SVD and ontology help to improve remarkably the prediction accuracy compared with the Pearson nearest neighbor algorithm. When compared to *Item-based + SVD + EM + Ontology* method, *user- and Item-based + SVD + EM + Ontology* method also works better but there is a small difference between them. The superiority of *user- and Item-based + SVD + EM + Ontology* can be explained by the fact that this method for predication uses ontology for the item-based CF part. Although the prediction accuracy of *Item-based + SVD + EM + Ontology* is slightly better than *User- and Item-based + SVD + EM*, however from its through result, it can be seen that this method throughput is lower than *User- and Item-based + SVD + EM* as it does not use SVD.

Decision-support accuracy metrics. Concerning the accuracy measures, in particular the decision-support metrics will play an important role for the multi-criteria recommender evaluations. Many metrics for this purpose are well known from the information retrieval area and will be discussed in the following. The precision Eq. (8) measures the portion of items that are relevant within the received result. In contrast, the recall Eq. (9) measures the portion of relevant items that have been retrieved. Both metrics should be used in common, as with increasing the amount of retrieved items, the recall increases, whereas the precision usually drops with larger result sizes.

$$\textbf{Precision} = \frac{TR}{TR + FR} \tag{8}$$

$$\textbf{Recall} = \frac{TR}{TR + FN} \tag{9}$$

where FN is the number of false non-relevant predictions, TR is the number of true relevant predictions and FR is the number of false relevant predictions.

A metric that considers both values is the F-measure (Tsai & Hung, 2012) (see Eq. (10)), which calculates the mean of the recall and the precision. β can be used to weight the influence of one of both, where $\beta > 1$ increases the importance of the precision and $\beta < 1$, on the opposite, raises the influence of the recall. For a balanced F-measure, $\beta = 1$ is used.

$$\textbf{F1} = \frac{(1 + \beta^2).precision.recall}{\beta^2.precision + recall} \tag{10}$$

For evaluating the proposed method using decision-support accuracy metrics, the precision and F1 were calculated on different numbers of Top-N. In this research, we consider N = 10, 20, 30, 40 and 50 which means that we evaluate the method when recommending the top 10, 20, 30, 40 and 50 movies by the proposed recommender system. Tables 2 and 3 show the precision values for different Top-N. From the table we can see that the precision obtained by our newly method are relatively high in relation to the nearest neighbor algorithm. These tables also show the F1 values for different Top-N. From this table, it can be seen that our method has outperformed the nearest neighbor algorithm in all datasets. For F1, it can be found that in the considered Top-N, the methods

Table 3
F1 and precision for all method on different numbers of Top-N (Yahoo! Webscope R4 dataset).

Top-N	Method A		Method B		Method C		Method D	
	Precision	F1-metric	Precision	F1-metric	Precision	F1-metric	Precision	F1-metric
Top-5	0.757	0.767	0.714	0.727	0.669	0.694	0.494	0.516
Top-10	0.766	0.777	0.737	0.757	0.683	0.707	0.517	0.534
Top-15	0.783	0.792	0.74	0.764	0.688	0.709	0.528	0.549
Top-20	0.786	0.797	0.747	0.771	0.692	0.711	0.536	0.557
Top-25	0.788	0.797	0.749	0.776	0.697	0.714	0.542	0.565
Top-30	0.789	0.801	0.757	0.779	0.704	0.724	0.551	0.571
Top-35	0.791	0.803	0.761	0.785	0.715	0.727	0.564	0.582
Top-40	0.799	0.805	0.764	0.791	0.721	0.734	0.571	0.594
Top-45	0.809	0.814	0.772	0.793	0.727	0.744	0.585	0.608
Top-50	0.815	0.821	0.784	0.798	0.739	0.762	0.617	0.631

Method A = *User- and Item-based + SVD + EM + Ontology*, Method B = *Item-based + SVD + EM + Ontology*, Method C = *User- and Item-based + SVD + EM*, Method D = *Nearest Neighbor*.

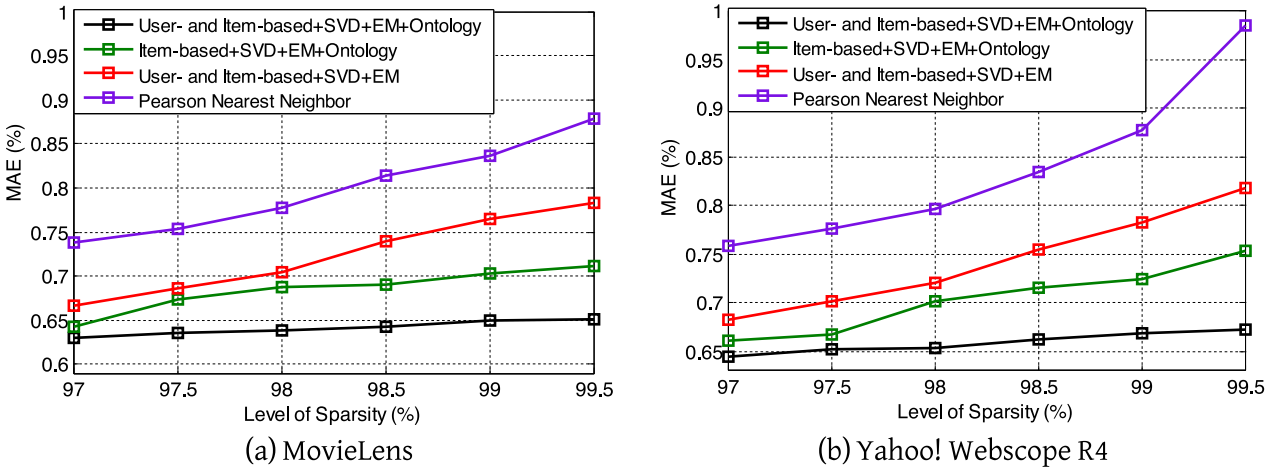

Fig. 10. MAE and different data sparsity levels.

which use ontology work better than the nearest neighbor algorithm. The superiority of our method can be explained by the fact that in our method we use ontology in the item-based CF part. These results are sufficient to support our claim that method is reasonably scalable and accurate in relations to the nearest neighbor algorithm.

We also evaluate the method on the different levels of sparsity and calculated the average MAE. The sparsity level of the MovieLens dataset is 93.7% (sparsity level = $1-(100{,}000/(943\times1682))=0.937$). In addition, the sparsity level of the Yahoo! Webscope R4 dataset is 99.8% (sparsity level = $1-(211{,}231/(7642\times11{,}915))=0.9976$). Accordingly, we create six datasets with different sparsity levels for the MovieLens and Yahoo! Webscope R4 datasets (i.e., 99.5%, 99%, 98.5%, 98%, 97.5% and 97%). We apply the method on the datasets with these sparsity levels and compare the results with the other recommendation algorithms. As can be seen from Figs. 10a-b, the MAE values of two methods *User- and Item-based + SVD + EM + Ontology* and *Item-based + SVD + EM + Ontology* for all sparsity levels of dataset are lower than *User- and Item-based + SVD + EM* and *Nearest Neighbor*. In addition, from these figures it can be seen that the increasing ratio of the MAE for *Nearest Neighbor* is very high compared to the other methods. The results also reveal that the methods which use ontology have better prediction accuracy compared to the other methods for the dataset that is sparser. This is because of the fact that the methods which used ontology are more effective in solving the sparsity issue and accordingly are more accurate.

Table 4
The MAE results of methods.

Method	MAE
User- and Item-based + SVD + EM + Ontology	0.6342 ± 0.0061
User-based CF	0.8080 ± 0.0095
Item-based CF	0.8370 ± 0.0083
CCAM	0.7520 ± 0.0053

In order to compare our work with the methods developed in the literature, we evaluated our method on the actual sparsity level of the MovieLens dataset (sparsity level = 93.7%) using MAE metric. The results are presented in Table 4. We report the average MAE of User- and Item-based + SVD + EM + Ontology, CCAM (Wu et al., 2014), User-based CF (Sarwar et al., 2001) and Item-based CF (Sarwar et al., 2001) methods. From the results presented in Table 3, we can see that the proposed method which uses ontology and dimensionality reduction techniques help to improve the MAE of recommendation over the CCAM (Wu et al., 2014), User-based CF (Sarwar et al., 2001) and Item-based CF (Sarwar et al., 2001) methods.

6. Discussion

Scalability and sparsity are two main issues in the design of recommender systems. Accordingly, in this research, attempts have been made to solve these issues to improve the performance of recommender systems. The method developed in this study uses ontology in the CF with the aid of clustering and dimensionality reduction techniques. To evaluate the method, two movie

datasets, Yahoo! Webscope R4 dataset and MovieLens, were used. The results provided by MAE, Recall, Precision and F1 showed that the use of ontology with the aid of clustering and dimensionality reduction techniques was effective in improving the performance of the CF recommender systems. The results of our analysis demonstrated that the hybrid recommendation method can be used to solve the scalability and sparsity issues of recommender systems.

With regard to the scalability issue, the proposed method enhanced the scalability of the CF recommender systems through throughput which is defined as the number of recommendation per second. The results showed that the throughput of those methods that use clustering and dimensionality reduction techniques is substantially higher other methods. In addition, we found that the method which uses the ontology, EM and SVD has a higher throughput than the methods which solely rely on nearest neighbor algorithm. With regard to sparsity issue, the hybrid method outperformed the nearest neighbor method in all datasets. In addition, the results also revealed that the methods which use ontology have better prediction accuracy compared to the other methods for the dataset that is sparser. The improvement in the accuracy of the recommendations by the proposed method is because the recommendation method uses semantic similarity relations for items in item-based CF. The use of semantic similarity accordingly improves the recommendation accuracy of item-based CF in the hybrid method.

Overall, it is worth mentioning that the proposed hybrid method is a significant improvement with respect to the throughput, prediction and recommendation accuracy. This demonstrates its effectiveness in alleviating the sparsity and improving the scalability of CF recommender systems. Because in recommendation systems the trade-off between the computation time (improving the scalability) and the accuracy (alleviating the sparsity) is important, our method can be a promising and effective intelligent system for movie recommendation.

7. Conclusions

The present study proposed a recommendation method based on CF using ontology and dimensionality reduction techniques to improve the sparsity and scalability problems in CF. We analyzed the predictive accuracy and time complexity (scalability) of proposed method on real-world datasets in the domain of movie recommendation provided by Movielens and Yahoo! Research Alliance Webscope program. The proposed method was evaluated using precision, MAE and F1 metrics to be comparable with the algorithms in previous studies. Our experiments confirmed that the proposed method has improved both the predictive accuracy and throughput of movie recommendations.

In this study, we have used solely EM clustering for clustering and non-incremental SVD for dimensionality reduction tasks in the proposed method. The use of incremental SVD may help the recommender system to provide recommendations with good scalability in relation to the non-incremental SVD. In addition, in this study the proposed method has been evaluated on movie recommendation domain. Hence, in our future work we plan to consider other clustering methods especially clustering ensembles methods to be incorporated into the proposed recommendation method. Furthermore, in our future studies we will extend the method by incorporating the incremental SVD into the predictions models for improving the scalability issue of CF. Moreover, in our future work we plan to further improve the proposed method and evaluate it using additional metrics such as diversity and novelty on other types of datasets.

References

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

Al-Hassan, M., Lu, H., & Lu, J. (2015). A semantic enhanced hybrid recommendation approach: A case study of e-Government tourism service recommendation system. *Decision Support Systems, 72*, 97–109.

Antoniou, G., & Van Harmelen, F. (2004). *Semantic Web Primer*. Cambridge, MA, USA: The MIT Press.

Armstrong, J. S. (2001). *Principles of forecasting: a handbook for researchers and practitioners*: Vol. 30. Boston: Kluwer Academic Publishers.

Bassiliades, N., Symeonidis, M., Meditskos, G., Kontopoulos, E., Gouvas, P., & Vlahavas, I. (2017). A semantic recommendation algorithm for the PaaSport platform-as-a-service marketplace. *Expert Systems with Applications, 67*, 203–227.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Paper presented at the Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*.

Celdrán, A. H., Pérez, M. G., Clemente, F. J. G., & Pérez, G. M. (2016). Design of a recommender system based on users' behavior and collaborative location and tracking. *Journal of Computational Science, 12*, 83–94.

Cheng, L.-C., & Wang, H.-A. (2014). A fuzzy recommender system based on the integration of subjective preferences and objective information. *Applied Soft Computing, 18*, 290–301.

Daramola, O., Adigun, M., & Ayo, C. (2009). Building an ontology-based framework for tourism recommendation services. *Information and communication technologies in tourism, 2009*, 135–147.

De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning, 51*(7), 785–799.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. *Series B (methodological), 39*, 1–38.

Fan, J., Pan, W., & Jiang, L. (2014 January). An improved collaborative filtering algorithm combining content-based algorithm and user activity. In *Big Data and Smart Computing (BIGCOMP), 2014 International Conference on* (pp. 88–91).

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval, 4*(2), 133–151.

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische mathematik, 14*(5), 403–420.

Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. *JSW, 5*(7), 745–752.

Gruber, T. R. (1992). *Ontolingua: a mechanism to support portable ontologies*: Vol. 27. Stanford: Stanford University, Knowledge Systems Laboratory.

Guarino, N., Oberle, D., & Staab, S. (2009). What is an Ontology? In *Handbook on ontologies* (pp. 1–17). Berlin Heidelberg: Springer.

Hawalah, A., & Fasli, M. (2014). Utilizing contextual ontological user profiles for personalized recommendations. *Expert Systems with Applications, 41*(10), 4777–4797.

He, Y., Yang, S., & Jiao, C. (2011). A hybrid collaborative filtering recommendation algorithm for solving the data sparsity. Paper presented at the Computer Science and Society (ISCCS), *2011 International Symposium on*.

Hofmann, T., & Puzicha, J. C. (2004). System and method for personalized search, information filtering, and for generating recommendations utilizing statistical latent class models: Google Patents.

Jugovac, M., Jannach, D., & Lerche, L. (2017). Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications, 81*, 321–331.

Kabassi, K. (2010). Personalizing recommendations for tourists. *Telematics and Informatics, 27*(1), 51–66.

Kermany, N. R., & Alizadeh, S. H. (2017). A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. *Electronic Commerce Research and Applications*, 50–64.

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

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

Kushwaha, N., & Vyas, O. P. (2014 October). SemMovieRec: Extraction of semantic features of DBpedia for recommender system. In *Proceedings of the 7th ACM India Computing Conference* (p. p. 13).

Lee, J. S., & Olafsson, S. (2009). Two-way cooperative prediction for collaborative filtering recommendations. *Expert Systems with Applications, 36*(3), 5353–5361.

Lee, T., Chun, J., Shim, J., & Lee, S. G. (2006). An ontology-based product recommender system for B2B marketplaces. *International Journal of Electronic Commerce, 11*(2), 125–155.

Liao, I., Hsu, W., Chen, M.-S., & Chen, L. (2010). A library recommender system based on a personal ontology model and collaborative filtering technique for English collections. *Emerald Group Publishing, 28*(3), 386–400.

Liao, I.-E., Liao, S.-C., Kao, K.-F., & Harn, I.-F. (2006). A personal ontology model for library recommendation system, ICADL'06. In *Proceedings of the Ninth International Conference on Asian Digital Libraries* (pp. 173–182).

Liao, S., Kao, K., Liao, I., & Chen, H. (2009). PORE: A personal ontology recommender system for digital libraries. *Emerald, 27*(3), 496–508.

Lilien, G. L., Kotler, P., & Moorthy, K. S. (1992). *Marketing models*. Prentice-Hall Englewood Cliffs.
Linden, G., Smith, B., & York, J. (2003). In *Amazon. com recommendations: Item–to-item collaborative filtering*. Internet Computing: *7* (pp. 76–80). IEEE.
Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014a). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems, 56*, 156–166.
Liu, L., Mehandjiev, N., & Xu, D. L. (2011 October). Multi-criteria service recommendation based on user criteria preferences. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 77–84).
López-Nores, M., Pazos-Arias, J. J., García-Duque, J., Blanco-Fernández, Y., Martín-Vicente, M. I., Fernández-Vilas, A., et al. (2010). MiSPOT: Dynamic product placement for digital TV through MPEG-4 processing and semantic reasoning. *Knowledge and Information Systems, 22*(1), 101–128.
Lv, G., Hu, C., & Chen, S. (2016). Research on recommender system based on ontology and genetic algorithm. *Neurocomputing, 187*, 92–97.
Martinez-Cruz, C., Porcel, C., Bernabé-Moreno, J., & Herrera-Viedma, E. (2015). A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling. *Information Sciences, 311*, 102–118.
Martín-Vicente, M. I., Gil-Solla, A., Ramos-Cabrer, M., Pazos-Arias, J. J., Blanco-Fernández, Y., & López-Nores, M. (2014). A semantic approach to improve neighborhood formation in collaborative recommender systems. *Expert Systems with Applications, 41*(17), 7776–7788.
Middleton, S. E., De Roure, D., & Shadbolt, N. R. (2009). Ontology-based recommender systems. In *Handbook on ontologies* (pp. 779–796). Berlin Heidelberg: Springer.
Moreno, A., Valls, A., Isern, D., Marin, L., & Borràs, J. (2013). Sigtur/e-destination: Ontology-based personalized recommendation of tourism and leisure activities. *Engineering Applications of Artificial Intelligence, 26*(1), 633–651.
Moreno, M. N., Segrera, S., López, V. F., Muñoz, M. D., & Sánchez, Á. L. (2016). Web mining based framework for solving usual problems in recommender systems. *A case study for movies' recommendation*. *Neurocomputing, 176*, 72–80.
Murthi, B., & Sarkar, S. (2003). The role of the management sciences in research on personalization. *Management Science, 49*(10), 1344–1362.
Nilashi, M. (2016). An Overview of Data Mining Techniques in Recommender Systems. *Journal of Soft Computing and Decision Support Systems, 3*(6), 16–44.
Nilashi, M., bin Ibrahim, O., & Ithnin, N. (2014). Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications, 41*(8), 3879–3900.
Nilashi, M., Jannach, D., bin Ibrahim, O., & Ithnin, N. (2015). Clustering-and regression-based multi-criteria collaborative filtering with incremental updates. *Information Sciences, 293*, 235–250.
Nilashi, M., Jannach, D., bin Ibrahim, O., Esfahani, M. D., & Ahmadi, H. (2016a). Recommendation quality, transparency, and website quality for trust-building in recommendation agents. *Electronic Commerce Research and Applications, 19*, 70–84.
Pham, M. C., Cao, Y., Klamma, R., & Jarke, M. (2011). A clustering approach for collaborative filtering recommendation using social network analysis. *Journal of Universal Computer Science, 17*(4), 583–604.
Pham, X. H., Jung, J. J., Nguyen, N. T., & Kim, P. (2016). Ontology-based multilingual search in recommendation systems. *Acta Polytechnica Hungarica, 13*(2), 195–207.
Porcel, C., Martinez-Cruz, C., Bernabé-Moreno, J., Tejeda-Lorente, Á., & Herrera-Viedma, E. (2015). Integrating ontologies and fuzzy logic to represent user-trustworthiness in recommender systems. *Procedia Computer Science, 55*, 603–612.
Powell, M. J. D. (1981). *Approximation theory and methods*. Cambridge university press.
Rich, E. (1979). User modeling via stereotypes. *Cognitive science, 3*(4), 329–354.
Sadaei, H. J., Enayatifar, R., Lee, M. H., & Mahmud, M. (2016). A hybrid model based on differential fuzzy logic relationships and imperialist competitive algorithm for stock market forecasting. *Applied Soft Computing, 40*, 132–149.
Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of:*. Addison-Wesley.
Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Paper presented at the Proceedings of the fifth international conference on computer and information technolog*.
Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001,). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285–295).
Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). *Collaborative filtering recommender systems the adaptive web* (pp. 291–324). Springer.
Shambour, Q., & Lu, J. (2012). A trust-semantic fusion-based recommendation approach for e-business applications. *Decision Support Systems, 54*(1), 768–780.
Shinde, S. K., & Kulkarni, U. (2012). Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications, 39*(1), 1381–1387.
Staab, S., & Studer, R. (2009). *Handbook on ontologies*. Springer.
Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence, 2009*(4), 1–19.
Taniar, D., & Rahayu, J. W. (2006). *Web semantics and ontology*. Idea Group Pub.
Tarus, J. K., Niu, Z., & Yousif, A. (2017). A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems, 72*, 37–48.
Ticha, S. B., Roussanaly, A., Boyer, A., & Bsaies, K. (2012 September). User semantic preferences for collaborative recommendations. In *International Conference on Electronic Commerce and Web Technologies* (pp. 203–211). Berlin Heidelberg: Springer.
Trewin, S. (2000). Knowledge-based recommender systems. *Encyclopedia of Library and Information Science: Supplement, 69*(32), 180–200.
Truong, K., Ishikawa, F., & Honiden, S. (2007). Improving accuracy of recommender system by item clustering. *IEICE Transactions on Information and Systems, 90*(9), 1363–1373.
Tsai, C.-F., & Hung, C. (2012). Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing, 12*(4), 1417–1425.
Ungar, L. H., & Foster, D. P. (1998). Clustering methods for collaborative filtering. *Paper presented at the AAAI Workshop on Recommendation Systems*.
Wang, P. (2012). A Personalized Collaborative Recommendation Approach Based on Clustering of Customers. *Physics Procedia, 24*, 812–816.
Wang, R. Q., & Kong, F. S. (2007 August). Semantic-enhanced personalized recommender system. In *Machine Learning and Cybernetics, 2007 International Conference on: 7* (pp. 4069–4074).
Wu, M. L., Chang, C. H., & Liu, R. Z. (2014). Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. *Expert Systems with Applications, 41*(6), 2754–2761.
Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., et al. (2005). Scalable collaborative filtering using cluster-based smoothing. In *Paper presented at the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*.
Zhang, Z.-K., Zhou, T., & Zhang, Y.-C. (2011). Tag-aware recommender systems: A state-of-the-art survey. *Journal of Computer Science and Technology, 26*(5), 767–777.
Zhou, D., Zhu, S., Yu, K., Song, X., Tseng, B. L., Zha, H., et al. (2008). Learning multiple graphs for document recommendations. In *Paper presented at the Proceedings of the 17th international conference on World Wide Web*.
Zhuhadar, L., Nasraoui, O., Wyatt, R., & Romero, E. (2009). Multi-model ontology-based hybrid recommender system in e-learning domain. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on: 3* (pp. 91–95). IEEE.