Expert Systems With Applications 89 (2017) 404–412

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Content-based filtering for recommendation systems using multiattribute networks

Jieun Son, Seoung Bum Kim*

Department of Industrial Management Engineering, Korea University, 145 Anam-Ro, Seoungbuk-Gu, Anam-dong, Seoul 136-713, South Korea

ARTICLE INFO

Article history:
Received 21 April 2017
Revised 1 August 2017
Accepted 2 August 2017
Available online 3 August 2017

Keywords:
Content-based filtering
Recommender system
Movie recommendation
Network analysis

ABSTRACT

Content-based filtering (CBF), one of the most successful recommendation techniques, is based on correlations between contents. CBF uses item information, represented as attributes, to calculate the similarities between items. In this study, we propose a novel CBF method that uses a multiattribute network to effectively reflect several attributes when calculating correlations to recommend items to users. In the network analysis, we measure the similarities between directly and indirectly linked items. Moreover, our proposed method employs centrality and clustering techniques to consider the mutual relationships among items, as well as determine the structural patterns of these interactions. This mechanism ensures that a variety of items are recommended to the user, which improves the performance. We compared the proposed approach with existing approaches using MovieLens data, and found that our approach outperformed existing methods in terms of accuracy and robustness. Our proposed method can address the sparsity problem and over-specialization problem that frequently affect recommender systems. Furthermore, the proposed method depends only on ratings data obtained from a user's own past information, and so it is not affected by the cold start problem.

1. Introduction

The development of the Internet and web technologies has expanded the range of items available in various areas such as entertainment, e-commerce, and education. However, it is becoming difficult to find suitable items that match the preferences of users. Thus, recommender systems are expected to guide users to items that might be of interest to them. Various approaches have been introduced to improve the recommendation performance of recommender systems (Adomavicius & Tuzhilin, 2005; Park, Kim, Choi, & Kim, 2012), where bestseller recommendation is one of the simplest mechanisms, based on sales frequencies. In addition, ratings data, purchase history records, user demographic information, social information, and product descriptions are used to improve customer satisfaction in personalized recommender systems (Al-Shamri, 2016; Nunes, 2012). Recommender systems are typically categorized into collaborative filtering (CF) and content-based filtering (CBF) systems. CF algorithms attempt to predict the utility of an item for a particular user based on the similarity to the user's past ratings (Mahmoud & John, 2015). These methods are used widely, but the main limitation of CF systems is that they are vulnerable to fraud or profile injection attacks, which might have significant negative effects on the robustness of such systems (Lee et al., 2012). In particular, the performance of CF degrades as the numbers of customers and products increase, and thus CF algorithms must be updated continually over time. Therefore, new approaches are required that can rapidly produce high-quality recommendations, even for very large-scale problems (Cai et al., 2014).

CBF algorithms recommend suitable items to users based on the descriptions of items and user preferences. Furthermore, CBF algorithms employ profile information or ratings solely for the active user, and thus they can generate accurate recommendations even if the number of ratings from other users is not sufficiently large. Nevertheless, CBF has several drawbacks. CBF cannot generate suitable suggestions if the content analyzed for an item does not contain appropriate information for categorization. To address this limitation, feature weighting (FW) has been proposed for CBF, which assigns different levels of importance to different features (Debnath, Ganguly, & Mitra, 2008). For example, when choosing a cellular phone, the price may be more important than its color. However, this approach tends to provide biased results to users when items are recommended by computing the similarity repeatedly based on only a few attributes, which is usually called the over-specialization problem. To address these problems, some studies have attempted to include ontological information in semantic analyses. However, the scalability and the sparsity problem may occur when enormous amounts of data are calculated using matrix-based approaches (Di Noia, Mirizzi, Romito, Zanker (2012)).

* Corresponding author.
E-mail addresses: jieunson@korea.ac.kr (J. Son), sbkim1@korea.ac.kr (S.B. Kim).

http://dx.doi.org/10.1016/j.eswa.2017.08.008
0957-4174/

In the present study, we propose a type of CBF that uses a multiattribute network (MN), which comprises entire attribute information for different items. Many possible attributes of CBF can play significant roles in determining the quality of the recommendation results, because they may provide sufficient information for measuring sophisticated similarities. In the network analysis, we measure the similarities between directly and indirectly linked items. Moreover, centrality techniques in the network analysis can simultaneously consider the mutual relationships among items that are indirectly connected as well as determining the structural patterns of these interactions. This approach can address the sparsity problem and the over-specialization problem that frequently affect recommender systems. Furthermore, our proposed method depends only on ratings data obtained from a user's own past information, and so it is not affected by the cold start problem that is prevalent in CF (Lika, Kolomvatsos, & Hadjiefthymiades, 2014).

The remainder of this paper is organized as follows. In the related work section, we review CBF-based recommendation approaches. The proposed method section presents the proposed recommender system. The experimental evaluation section presents the experimental results and discussion. In the final section, we present our conclusions.

2. Related work

2.1. Multiattribute recommender systems

2.1.1. Pure CBF

CBF attempts to recommend items similar to those that a given user has liked in the past. The basic process is performed by matching user preferences with item attributes. Therefore, these systems require appropriate techniques for representing items and determining user preferences, as well as strategies for comparing user preferences with item representations (Choi, Kang, & Jeon, 2006; Lops, De Gemmis, & Semeraro, 2011).

Various machine learning techniques have been applied to CBF, including decision trees, K-means, neural networks, and naïve Bayes. For example, the basic concept employed by a naïve-Bayes classifier aims to determine whether an item is preferable by examining attribute information (Lew, Sebe, Djeraba, & Jain, 2006; Li, Lu, & Xuefeng, 2005). This method is used to estimate the probability that an item belongs to a class C_i The rating prediction is computed using the following probability function:

$$P(C_i|X) = \prod_{k=1}^{n} P(x_k|C_i), \quad (1)$$

where each item instance X is described by a conjunction of item attribute values $\langle x_1, x_2, \ldots x_k \rangle$. However, CBF cannot provide suitable recommendation results if the content does not contain sufficient information for classifying items. In some cases, domain knowledge or an ontology is required to determine the attributes that play important roles in recommendation.

2.1.2. Feature weighting

In CBF, feature weighting (FW) assigns different levels of importance to different features (Debnath et al., 2008), where the weight values obtained from a social network graph are used for predicting user preferences. The similarity S between objects O_i and O_j is calculated as

$$S(O_i, O_j) = \omega_1 f(A_{1i}, A_{1j}) + \omega_2 f(A_{2i}, A_{2j}) + \cdots + \omega_n f(A_{ni}, A_{nj}), \quad (2)$$

where ω_n is the weight for attribute A_n between object O_i and O_j, and f depends on the type of attribute. However, FW algorithms constitute hybrid recommender systems, because they require preference information about other users. Thus, they cannot be built on the basis of the ratings provided by "active users."

2.2. Semantic analysis

Some studies have aimed to improve the recommendation performance by including ontological information. Di Noia et al. (2012) demonstrated how the linked open data (LOD) cloud can be used as the main information source for a semantic CBF, where this method employs a three-dimensional vector space model comprising movies, movie properties, and the values of properties extracted from DBpedia and LinkedMDB. Semantic analysis using LOD is more suitable in comparison with existing methods. However, although the content analyzed for items contains sufficient information, the scalability problem still needs to be resolved. Furthermore, when a large number of attributes are included, sparse matrix issues can frequently occur.

2.3. Network analysis

Networks are patterns in the relationships that connect objects. A network analysis involves exploring, describing, and understanding numerous relational and structural aspects of networks. Important studies of network analyses have been conducted in an extremely diverse range of fields, including sociology, psychology, business, computer science, mathematics, and statistics (Anderson & Vongpanitlerd, 2006). The nodes in a network represent objects, and the links denote the relationships or flows between nodes. The distinguishing feature of a network analysis compared with more traditional connectivity measures is that it explicitly quantifies how the connectivity varies according to the characteristics of individuals. One of its advantages is that very complex and otherwise incomprehensible relationships can be clarified and structured. Thus, conclusions can be reached regarding large-scale connections that might otherwise appear irrelevant or excessively complicated. Moreover, meaningful information can be extracted by moving beyond individual perception when relatively little input data is available, because a network analysis does not necessarily focus on a specific location, but instead can encompass all types of relationships (Carrer-Neto, Hernández-Alcaraz, Valencia-García, & García-Sánchez, 2012).

Network analyses have also been used to improve the performances of recommender systems. Online social networking services can provide additional social relationship information about users (Kwon et al., 2009). Clustering in networks reduces the likelihood of information overload and improves the scalability of systems, because the similarity can only be calculated for users in the target cluster. Some studies have proposed clustering approaches based on social networks of users in order to derive CF recommendations (Colace, De Santo, Greco, Moscato, & Picariello, 2015; Seth et al., 2008), which can provide a rich source of information regarding user groups and their correlations. In addition, experimental evaluations have shown that using clustering techniques improves the prediction accuracy of user preferences (Pereira & Hruschka, 2015). Analyzing user activities in social network services (e.g., Facebook or LinkedIn) can provide e-commerce with the opportunity to create more personalized offers, and to help users cope with huge information overload problems (Zhou, Xu, Li, Josang, & Cox, 2012). In hybrid systems that combine CF and CBF methods, linear regression equations obtained from a social network graph are used to estimate the weight values of attributes (Debnath et al., 2008; Kushwaha, Mehrotra, Kalia, Kumar, & Vyas, 2016).

Table 1
Item-attribute matrix.

	Attribute 1	Attribute 2	...	Attribute P
Item 1	A, B	F	...	Z
Item 2	B, C	G	...	Y
Item 3	A, C	G	...	Z
...			...	
Item N	B, C	F	...	Y

Table 2
Binary version of the item-attribute matrix.

	Attribute 1			Attribute 2		...	Attribute P	
	A	B	C	F	G	...	Y	Z
Item 1	1	1	0	1	0	...	0	1
Item 2	0	1	1	0	1	...	1	0
Item 3	1	0	1	0	1	...	0	1
...	...	...	...	...	...	...	...	...
Item N	0	1	1	1	0	...	1	0

3. Proposed method

The proposed CBF recommendation method, using an MN (CBF–MN) algorithm, considers the relationships among items over a broad range, and evaluates the importance of each item based on various features. The overall process of the CBF–MN algorithm is described as follows: (1) acquiring item attributes, (2) calculating item similarities, (3) generating an MN containing all the items, (4) network clustering to group the items, and (5) calculating a score that reflects the importance of each item selected in the cluster, and using this score for recommendation.

3.1. Data acquisition

Attributes characterize each item, e.g., the attributes of a movie include actors, actresses, the director, language, country, and budget. The proposed CBF–MN uses as many attributes as possible to efficiently represent an item, because each user has different preference criteria regarding item selection. When the attributes are unstructured, an appropriate preprocessing algorithm is required to transform them into structured data (Liao, Chu, & Hsiao, 2012). Table 1 shows the format of an item–attribute matrix.

3.2. Item similarity computation

After constructing the item–attribute matrix, the similarities are computed to determine the closeness between paired items. The Dice similarity can be used to measure the similarity between two samples when the attribute is categorical, and assign the same weight to each matching attribute. In this study, we employ the Dice similarity, which calculates the similarity scores of nodes based on their connecting patterns (Choi, Cha, & Tappert, 2010; Song, Kawabata, Itoh, Watanabe, & Yokota, 2013). A binary matrix is required to calculate the Dice similarity. Table 2 presents a binary version of the item–attribute matrix shown in Table 1.

The Dice similarity (DS) can be calculated as follows:

$$DS_{ij} = \frac{2n_{ij}}{n_i + n_j}, \tag{3}$$

where n_i, n_j, and n_{ij} are the number of values in items i, the number of values in item j, and the number of common values between items i and j, respectively.

We define the multiattribute similarity (MAS) by taking the average of the Dice similarities for all attributes:

$$MAS_{ij} = \frac{\sum_{n=1}^{P} DS_{ij}(n)}{p}, \tag{4}$$

where P is the number of attributes, and $DS_{ij}(n)$ is the Dice similarity between items i and j in an attribute n. One may think that MAS is a naïve measure that just consider the average of the Dice similarities for all attributes without taking into account the relevance they have. In this study, we assume that the attributes of the movie are independent.

An item-item similarity matrix is generated by calculating the MAS. Obviously, the range of values in this similarity matrix is between zero and one.

3.3. Generating an MN

Matrix-based MAS only considers the direct relationships between paired items. A binary matrix (Table 2) tends to be sparse, especially when items are attributed many attributes and their values. Therefore, it is difficult to determine the structural and indirect relationships, even after the item similarities have been generated. However, a network analysis based on the relational information among items can address this issue. According to graph theory, a network analysis can improve the search efficiency in a complex structure (Newman & Girvan, 2004). In the proposed method, we conduct a network analysis based on the item similarities, to determine the overall structure of the relationships and hidden information. The network graph can be generated based on an item–item similarity matrix. A one-mode weighted graph G comprises a set of nodes V and weighted edges E. The weight of a graph is equal to the item similarity score calculated using Eq. (4). Each weighted edge has two end points, n_1 and n_2, such that $n_1, n_2 \in V$. The node and weighted edge sets are typically denoted by capital letters. The size of the network is determined by the number of nodes.

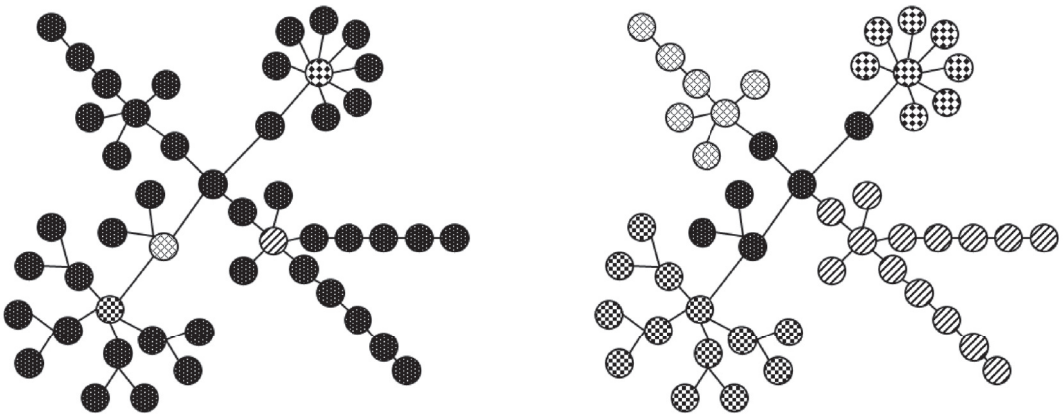
In a movie recommender system, a node of the network represents a movie, and a weighted edge represents the degree of relevance between two movies. For example, if all the attributes of two movies match, then they are connected by a thick edge, and if only one attribute matches, they are connected by a thin edge. These movie networks can identify not only direct associations between movies, but also indirect associations and structural characteristics.

3.4. Item clustering

Item clustering improves the accuracy of user rating predictions, and substantially increases the speed of online calculations. The aim of this step is to locate neighboring items for all the items in a network. Network clustering algorithms are designed to measure the strength of the division of a network into clusters. The principal aim of this technique is to discover the natural divisions of social networks as groups. Modularity analysis is one of the most popular methods for detecting community structures in a network (Amiri, Hossain, Crawford, & Wigand, 2013; Santos, Carvalho, & Nascimento, 2016). Networks with high modularity have dense connections between the nodes within clusters, but sparse connections between nodes in different clusters. The community structure based on the concept of modularity Q was introduced by Newman (Newman & Girvan, 2006), and Q is calculated as follows:

$$Q = \sum_{r} \left(e_{rr} - a_r^2 \right), \tag{5}$$

where e_{rr} is the fraction of links that connect two nodes inside the community r, and a_r is the fraction of links that have one or both vertices inside r. A higher value of Q indicates a stronger community structure in a network. We demonstrate the applicability of the modularity technique through a hypothetical example with 48 observations. Fig. 1 compares two clustering methods (K-means and modularity) in a non-weighted network, where each method generates five clusters by adjusting the parameters. According to



(a) Result of K-means clustering. (b) Result of modularity clustering.

Fig. 1. Comparison of two clustering methods.

Fig. 1(a), K-means cannot determine correct communities, because it only considers the direct relationships between paired items. In contrast, as shown in Fig. 1(b), the modularity analysis identifies the proper number of clusters by considering the link structures between observations.

Modularity clustering is one of the best performing algorithms among network-based clustering techniques. Some researchers have demonstrated that modularity analysis also performs well in recommendation systems. The modularity metric, which is well-known in the area of complex networks as a measure for quantifying how well groups of nodes are defined in a network, has been used to form groups that contain users and movies that are closely related (Demovic et al., 2013; Sibaldo, de Carvalho, Ren, & Cavalcanti, 2014). In our movie recommender system, although movies are not directly linked, they can be grouped into one group. In other words, through a modularity analysis, it is possible to understand which films share similar characteristics and are structurally clustered.

3.5. Recommendation

The items in a clustered group of user preferences are candidates for recommendation when the user provides information on their past experiences with these items in the form of ratings. First, we calculate the centrality of items in the MN to determine the structural similarity relationships among items. Second, for personalized recommendations, the system incorporates the distance from the user's preferred item to other items in the centralities.

3.5.1. Centrality of items in the MN

We use three centrality measures to determine the most similar items. A central node can be considered important, because it has an advantageous and influential position in the network. The degree centrality (D_C^w) is the number of connections between a node and other nodes. A given node has a greater influence when it has more neighbors (Opsahl, Agneessens, & Skvoretz, 2010). For a node i, D_C is defined as

$$D_C^w(i) = \frac{\sum_j^N w_{ij}}{n-1}, \tag{6}$$

where w is the weighted adjacency matrix, and w_{ij} is the weight of the edge that connects node i to j. Here, n is the total number of nodes in the network. A high value of D_C^w indicates that an item shares attributes with other items.

The calculation of D_C^w is limited by the number of nodes that are directly connected to the target node, whereas indirectly connected nodes are not included in the measurement.

The closeness centrality (C_C^w) is a measure of the average shortest path between the node and all other nodes in the graph (Okamoto, Chen, & Li, 2008), which allows the identification of nodes that can be reached quickly from other nodes:

$$C_C^w(i) = \frac{n-1}{\sum_{j=1}^n d^w(i,j)}, \tag{7}$$

where $d^w(i,j)$ is the weighted shortest path from i to j, and n is the number of items in the network. A node with a high value of C_C^w is located close to the center, because it is close to all other nodes in the network (Newman, 2008).

The betweenness centrality (B_C^w) is defined based on the number of shortest paths passing through a node (Freeman, 1978), and measures the influence of a node over the flow of information between every pair of nodes in the network. Nodes with high centrality are in a special position, because most of the other nodes must channel their communications through them.

$$B_C^w(i) = \sum_{j<k} \frac{g_{jk}^w(i)}{g_{jk}^w}, \tag{8}$$

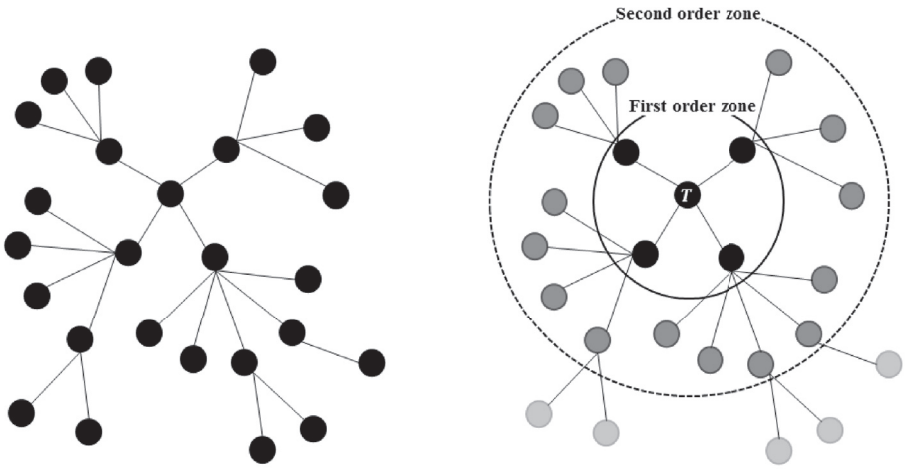
where g_{jk} is the number of all weighted geodesic paths connecting nodes j and k, and $g_{jk}(i)$ is the number of the weighted geodesic paths connecting j and k that pass through node i. While the range of the previous two formulas is between 0 and 1, the range of the betweenness centrality can exceed 1, because the betweenness centrality of a node scales with the number of pairs of nodes. Therefore, we divide each value of the betweenness centrality by the maximum betweenness centrality, so that its range is represented as a value between 0 and 1.

Communities form a cluster if there are subtle differences in the structures among items. An item that lies on communication paths can control the communication flow among communities, and thus is important. In a recommender system, this improves the diversity of the items recommended to the user.

To combine the values of the centralities obtained using Eqs. (6), (7), and (8), we simply use the average centrality (AC):

$$AC(i) = \frac{\sum_{k=1}^M C_k(i)}{M}, \tag{9}$$

where M is the number of centralities, and $C_k(i)$ is the centrality value of node i in terms of the centrality measure C_k. In our case, the value of M is 3, because we use three centralities: degree centrality, closeness centrality, and betweenness centrality.



(a) Sociocentric network (b) Egocentric network

Fig. 2. Examples of sociocentric and egocentric networks.

3.5.2. Ego-focused centrality

Two principal approaches are employed in a network analysis: sociocentric and egocentric (Chung, Hossain, & Davis, 2005; Marsden, 2002). Existing centrality measures, such as those in Eqs. (4), (5), and (6), are generated in a sociocentric network. Therefore, the greater the average centrality value of Eq. (9), the more likely it is that the movie has popular and common attributes. This approach is appropriate when recommending movies that have the most common and popular attributes to users who do not have a past history. However, for personalized recommendations, items that are considered to match the user's preference should be recommended. Therefore, we conduct a centrality analysis in an egocentric network for personalized recommendations (Biswas & Biswas, 2015). When a user picks an item of interest, this becomes a target item (ego node), and recommender systems must find items relevant to the target item.

Fig. 2. presents examples of sociocentric and egocentric networks. A sociocentric network analysis focuses (Fig. 2(a)) on measuring the structural patterns of the interactions among all nodes in a network. Fig. 2(b) shows an example of an egocentric network, which focuses on an individual. In an egocentric network analysis, the neighboring nodes of the target node T are considered as the subjects of an analysis with order zones. A first order zone includes all individuals that are directly linked to T. A second order zone is a group of nodes connected to at least one node within a first order zone. However, existing centrality measures for egocentric designs provide no information associated with the properties of neighboring nodes. Thus, a loss of information occurs, because the measures cannot consider other nodes located outside of an order zone (Everett & Borgatti, 2005; Newman, 2003).

In this study, we propose a centrality measure for egocentric networks that considers all of the nodes in a network. Thus, the ego-focused centrality (C_{EF}) can be calculated as follows:

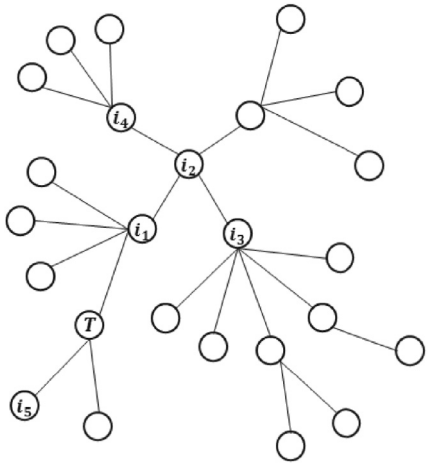


Fig. 3. An egocentric network in which T is the ego-node.

$$C_{EF}(i,T) = \frac{AC(i)}{d(i,T)}, \tag{10}$$

where the numerator is the average centrality value calculated in Eq. (9), and the denominator is a distance that can be quantified by the number of links between item i and T.

Fig. 3 and Table 3 show the values of C_{EF}s for i_1, i_2, i_3, i_4, and i_5 in the egocentric network in which T is the ego-node. For example, the C_{EF} of i_1 is high, although the average centrality of i_1 is lower than those of i_2 and i_3. This is because i_1 is closer to T than i_2 and i_3. In contrast, when i_3 and i_4 are at the same distance, i_4 has a

Table 3
The calculation of ego-focused centrality values for nodes i_1, i_2, i_3, i_4, and i_5.

	Degree Centrality	Closeness Centrality	Betweenness Centrality	Average Centrality	Link Distance	Ego-focused Centrality
i_1	0.833	0.817	0.567	0.739	1	0.739
i_2	0.667	1.000	1.000	0.889	2	0.445
i_3	1.000	0.875	0.719	0.865	3	0.288
i_4	0.667	0.742	0.163	0.524	3	0.175
i_5	0.167	0.480	0.000	0.216	1	0.216

Table 4
Example showing the calculation of an ego-focused recommendation score.

	Ego-focused Centrality of Preference Item T			Ego-focused Centrality of Non-preference Item T			RS_{EF}
	T1	T2	T3	T4	T5	T6	
i_1	0.999	0.764	0.262	0	0.121	0.319	1.585
i_2	0.223	0.562	0.751	0.112	0.098	0.216	1.11
i_3	0.551	0.674	0.495	0.251	0.132	0.311	1.026
i_4	0	0	0	0	0	0	0
i_5	0.465	0.131	0.516	0.432	0.333	0.347	0
i_6	0.525	0.778	0.423	0.991	0.246	0.546	-0.057
i_7	0.213	0.364	0.105	0.779	0.528	0.468	-1.093

Table 5
The complete list of attributes used in our movie recommendation experiment.

Attribute	Number of values	Type	Number of categories	Domain
Release	1	Year	79	YYYY
Genre	1–5	String	18	Comedy, etc.
Director	1	String	974	<Name>
Writer	1	String	1002	<Name>
Actor	3	String	2817	<Name>
Keyword	5	String	3952	Love etc.
Viewing Class	1	String	4	All, 12, 15, 18
Color	1	String	2	Color, B/W

lower ego-focused centrality value, because the average centrality of i_4 is lower than that of i_3. Similarly, all items are assigned a C_{EF} value. Consequently, all items are sorted based on C_{EF}, and the top n items in the list are recommended to a user.

3.6. Extensions

In general, a user's history contains an abundance of information regarding preferences and non-preferences. In this section, we explain a method for creating an item list arranged in order of preference priority for recommendation when the user has a number of preference and non-preference items in the database as a training dataset. First, the items purchased personally by a user are divided into two groups based on ratings information: a preference item set and a non-preference item set. For each item set, the C_{EF} values are calculated for each target item. Next, all the calculated C_{EF} values are combined for each group. The ego-focused recommendation score (RS_{EF}) is the sum of the C_{EF} values in the preference item set minus the sum of the C_{EF} values in the non-preference item set:

$$RS_{EF}(i) = \left(\sum_{j=1} C_{EF-P}(i, T_j)\right) - \left(\sum_{k=1} C_{CF-NP}\ (i, T_k)\right), \tag{11}$$

where C_{EF-P} is the C_{EF} of a preference item, and C_{EF-NP} is the C_{EF} of a non-preference item. Moreover, T_j is the preference item set, and T_k is the non-preference item set for a personal user. The first term of Eq. (11) represents the degree of preference based on a target item i, and the second term denotes the degree of non-preference based on a target item i. Therefore, a positive RS_{EF} indicates that item i is relevant to the user's interests. However, a negative RS_{EF} indicates that the item is irrelevant to the user's interest. Table 4 shows an example of an RS_{EF} calculation.

The system creates an item list arranged in order of preference priority for recommending items to the user, who has three preference items and three non-preference items as a training dataset. For example, the sum of C_{EF} values in the preference group for i_1 is high, whereas the sum of C_{EF} values in the non-preference group is low, i.e., i_1 is strongly related to the preference items but has no relationship with the non-preference items. In conclusion, it is highly likely that i_1 will be recommended to the user. In the case of i_5, the sum of the C_{EF} values is high in both the preference and non-preference groups, which is highly ambiguous for predicting the preference priority, because i_5 is related to the preference items as well as the non-preference items. Items i_6 and i_7, which have negative RS_{EF} values, might only rarely be recommended to a user.

4. Experimental evaluation

4.1. Experimental data and evaluation measure

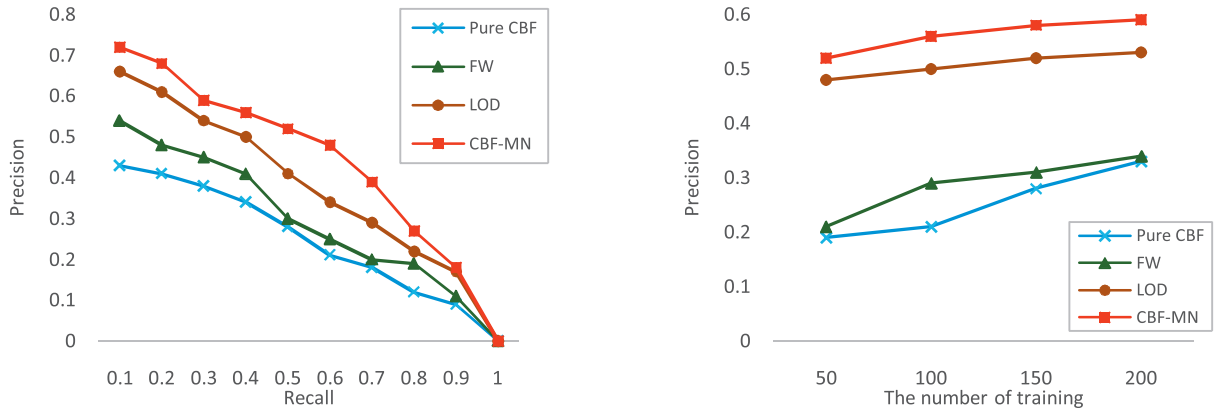
In this section, we present the results of experiments conducted to evaluate the proposed CBF–MN system, and comparisons with other methods. We used movie ratings data provided by the online movie recommender service MovieLens (Herlocker, Konstan, & Riedl, 2000) for movie recommendations. The 1 M MovieLens dataset contains 100,000,029 ratings provided by 6040 users for 3952 movies, as well as movie information such as the movie title, release date, and genre. A number of attributes are required by CBF–MN to implement the recommender system. Therefore, we gathered additional attribute information from the internet movie database (IMDB) website, which contains movie information such as the names of the director(s), writer(s), and actor(s), as well as keywords, viewing classes, and color/black-and-white. Table 5 shows the complete list of attributes used in our experiment for movie recommendation. For example, the genre attribute is the genre of the movie content. Each movie has between one and five genres, such as comedy or action. The number of genre categories is 18, and the data type is string.

Table 6 shows a binary version of the item-attribute matrix generated using Table 5. For example, the genres of Toy Story, released in 1995, are comedy, adventure, and animation. John Lasseter is the director as well as the writer. The stars of Toy Story are Tom Hanks, Tim Allen, and Don Rickles. The keywords are rivalry, toy, cowboy, jealousy, and claw crane, and the viewing class is all.

The MovieLens ratings range between one and five points, where one indicates a strong non-preference and five indicates a strong preference. However, the preferences of users are different, and so their scales can be different, i.e., one user might consider assigning a rating of three points whereas another might assign the same movie a different rating. Therefore, a standardized procedure is required for each user (Chung & Jo, 2008). A standardized

Table 6
Binary version of the item-attribute matrix on movie domain.

	Release			Genre				Viewing Class			Color	
	1950	...	1995	Comedy	...	Animation	...	All	...	18	Color	B/W
Toy Story	0	...	1	1	...	1	...	1	...	0	1	0
Heat	0	...	1	0	...	0	...	0	...	1	1	0
Powder	0	...	1	0	...	0	...	0	...	0	1	0
...	...	...	...	...	...	...	...		...		...	...
Clueless	0	...	1	1	...	0	...	0	...	0	1	0



(a) Precision and recall curves (b) N precision

Fig. 4. Performance comparison using Pure CBF, FW, LOD, and CBF-MN in terms of the precision and recall.

procedure involves determining the expected rating, which can be calculated from the following equation:

$$E = \sum_{c=min}^{max} c \times P(c), \tag{12}$$

where c represents the rating points, P(c) is the proportion of c for a specific user, min is the minimum grade point, and max is the maximum grade point in a ratings data set. If a user's rating experience is positive, then the expected rating is higher than that of other users. We classified the training items into two groups. If the rating of an item was greater than or equal to the user's own expected rating E, then the item was classified into the preference group. By contrast, an item was classified into the non-preference group when the rating of an item was lower than the expected rating E.

To evaluate the performance of the proposed approach, we compared it with Pure CBF, FW, and LOD, as described in Section 2.1. For pure CBF, no user-defined parameters exist, because we used a naïve Bayes algorithm (Lew, Sebe, Djeraba, & Jain, 2006). For FW, we used a linear regression framework from the literature (Debnath et al., 2008). In Eq. (2), the similarity S between the objects O_i and O_j is taken as the number of users who are interested in both objects. Thus, solving Eq. (2) using a linear regression equation provides estimates for the weights $\omega_1, \omega_2, \cdots, \omega_n$. To generate the matrix for LOD solutions, we followed the method described in Di Noia et al. (2012). The LOD graph is represented as a three-dimensional adjacency matrix, where each dimension refers to an ontology property. A cell in the matrix is not null if there is a property that relates a subject in a row to an object in a column. For a given dimension, the similarity between two movies is the cosine similarity between the two vectors. In this manner, all of the nodes in the graph are represented both in the rows and the columns of the matrix.

We used two evaluation measures, i.e., precision and recall, which are widely employed as information retrieval evaluation metrics (Schröder, Thiele, & Lehner, 2011).

4.2. Experimental results

The dataset comprised 100 users, which we randomly selected from the MovieLens database. The final results were obtained by averaging the results of 10 repetitions of these random selections.

Fig. 4(a) presents the precision and recall curves for the three algorithms. For an individual trial of an experiment, we randomly selected 200 rated movies as a training set, and 50 movies as a testing set. Fig. 4(a) shows that the proposed CBF–MN performed better than the other three algorithms. The Pure CBF, FW, and LOD algorithms only consider the direct relationships between two items, and thus it is highly likely that a recommender system would recommend similar items, with specific attributes being provided to the users repeatedly. This phenomenon is called concentration bias or over-specialization (Mahmoud & John, 2015). In particular, FW is more suitable for users with certain attributes that reflect their preferences. However, more advanced strategies consider the fact that various items are required for personalized recommendations, which may lead to the cold start problem in practical applications. The ontological information in LOD improves the quality of CBF, but the scalability and sparsity problem must also be addressed.

As shown in Fig. 4(b), we performed experiments with training sets of various sizes to examine the impact on the performance of changing the amount of training data. For validation, the 50 most recently rated movies were used as testing data. The training data was varied by increasing the number of movies rated in the past from 50 to 200 in increments of 50. For all four methods, a smaller training set yielded an inferior performance. CBF–MN and LOD were more robust in terms of the training set size than the other two algorithms, because a network analysis or ontology

analysis can provide more information even with small amounts of training data. Overall, the proposed CBF–MN method outperformed the other algorithms in terms of accuracy.

5. Conclusions

The vast amount of information that is currently available makes it difficult for users to find items in e-commerce systems. A recommender system can help users to address the issue of information overload, and find objects that are relevant to them. CBF systems are the most widely employed filtering systems, which aim to recommend items that are similar to those a given user has liked in the past. The basic process involves matching user preferences with item attributes. However, these systems have limitations, because they only consider direct relationships among items. In particular, the performance is degraded when the system has many attributes or standards of similarity. Some studies have aimed to select or weight features in order to overcome these limitations, but these have combined CF with a large number of ratings given by other users.

To address these shortcomings, we have proposed a CBF system using an MN that includes all of the attribute information for items. Many possible attributes of CBF can play significant roles in determining the quality of recommendation results, because they may provide information that is suitable for measuring sophisticated similarities.

We compared the proposed CBF-MN recommender system with Pure CBF, FW, and LOD using MovieLens data, which demonstrated that our system performed better than the existing methods in terms of accuracy and robustness. The experimental results and analysis highlighted the following interesting findings and practical implications. Existing CBF approaches suffer from over-specialization, because they do not employ suitable information for measuring item similarities. Furthermore, the scalability and the sparsity problem should be addressed, because enormous amounts of data are used in calculations with matrix-based approaches. In contrast, CBF–MN allows various items to be recommended to a user based on a network analysis, and this improves the system's performance. CBF–MN solves the issue of over-specialization, because a number of attributes are used as criteria for characterizing items. Ultimately, it is highly desirable that a recommender system should not recommend excessively similar items, but rather diverse items by considering diverse criteria. Furthermore, CBF–MN adopts a network analysis that considers the relationships between all items, and examines the structural and indirect relationships among them. This approach provides a wealth of information, and helps to address the sparsity problem.

In future research, we would like to use additional text features. Eight attributes were used in our recommendations, but this might be insufficient for recommending appropriate movies. Additional features related to scenarios or review comments could be incorporated into each node to construct more informative networks. Further, we will consider the relevance of attribute for movie recommendation systems. We found that some research attempted to determine the relevance of attributes for recommendation systems. One of them is feature weighting used in our experiments as a comparison algorithm (Debnath et al., 2008). Feature weighting assigns different levels of importance to different attributes in which the weight values obtained from a social network graph are used for predicting user preferences. They found that some of the attributes have stable weight values, while some attributes like director, rating, vote, year, and color have unstable or negative weight. We believe if the relevance information for attributes is included, our proposed recommendation method assuming all attributes are equally important can be improved and more meaningful. We will keep this issue that reflects the relevance of attributes as the priority for the future study.

Acknowledgments

The authors would like to thank the editor and reviewers for their useful comments and suggestions, which were greatly help in improving the quality of the paper. This work was supported by Brain Korea PLUS, Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning (NRF-2016R1A2B1008994) and Ministry of Trade, Industry & Energy under Industrial Technology Innovation Program (R1623371).

References

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

Al-Shamri, M. Y. H. (2016). User profiling approaches for demographic recommender systems. *Knowledge-Based Systems, 100*, 175–187.

Amiri, B., Hossain, L., Crawford, J. W., & Wigand, R. T. (2013). Community detection in complex networks: Multi–objective enhanced firefly algorithm. *Knowledge-Based Systems, 46*, 1–11.

Anderson, B., & Vongpanitlerd, S. (2006). *Network analysis and synthesis*. Dover.

Biswas, A., & Biswas, B. (2015). Investigating community structure in perspective of ego network. *Expert Systems with Applications, 42*(20), 6913–6934.

Cai, Y., Leung, H. F., Li, Q., Min, H., Tang, J., & Li, J. (2014). Typicality-based collaborative filtering recommendation. *IEEE Transactions on Knowledge and Data Engineering, 16*(3), 766–779.

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R., & García-Sánchez, F. (2012). Social knowledge-based recommender system. Application to the movies domain.. *Expert Systems with applications, 39*(12), 10990–11000.

Choi, S. S., Cha, S. H., & Tappert, C. C. (2010). A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics, 8*(1), 43–48.

Chung, K. K., Hossain, L., & Davis, J. (2005). Exploring sociocentric and egocentric approaches for social network analysis. In *Proceedings of the 2nd international conference on knowledge management in Asia Pacific* (pp. 1–8).

Chung, K. Y., & Jo, S. M. (2008). Discovery of preference through learning profile for content-based filtering. *The Journal of the Korea Contents Association, 8*(2), 1–8.

Colace, F., De Santo, M., Greco, L., Moscato, V., & Picariello, A. (2015). A collaborative user-centered framework for recommending items in Online Social Networks. *Computers in Human Behavior, 51*, 694–704.

Choi, S. H., Kang, S., & Jeon, Y. J. (2006). Personalized recommendation system based on product specification values. *Expert Systems with Applications, 31*(3), 607–616.

Debnath, S., Ganguly, N., & Mitra, P. (2008). Feature weighting in content based recommendation system using social network analysis. In *Proceedings of the 17th international conference on World Wide Web* (pp. 1041–1042).

Demovic, L., Fritscher, E., Kriz, J., Kuzmik, O., Proksa, O., Vandlikova, D., et al. (2013). Movie recommendation based on graph traversal algorithms. In *Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on* (pp. 152–156). IEEE.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., & Zanker, M. (2012). Linked open data to support content-based recommender systems. In *Proceedings of the 8th international conference on semantic systems* (pp. 1–8).

Everett, M., & Borgatti, S. P. (2005). Ego network betweenness. *Social Networks, 27*(1), 31–38.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks, 1*(3), 215–239.

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

Kushwaha, N., Mehrotra, S., Kalia, R., Kumar, D., & Vyas, O. P. (2016). Inclusion of semantic and time-variant information using matrix factorization approach for implicit rating of Last. Fm dataset. *Arabian Journal for Science and Engineering, 41*(12), 5077–5092.

Kwon, O. (2009). A social network approach to resolving group-level conflict in context-aware services. *Expert Systems with Applications, 36*(5), 8967–8974.

Lee, J. S., & Zhu, D. (2012). Shilling attack detection-a new approach for a trustworthy recommender system. *INFORMS Journal on Computing, 24*(1), 117–131.

Lew, M. S., Sebe, N., Djeraba, C., & Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2*(1), 1–19.

Li, Y., Lu, L., & Xuefeng, L. (2005). A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce. *Expert Systems with Applications, 28*(1), 67–77.

Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications–A decade review from 2000 to 2011. *Expert Systems with Applications, 39*(12), 11303–11311.

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications, 41*(4), 2065–2073.

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

Mahmoud, D. S., & John, R. I. (2015). Enhanced content-based filtering algorithm using Artificial Bee Colony optimization. In *SAI intelligent systems conference (IntelliSys)* (pp. 155–163).

Marsden, P. V. (2002). Egocentric and sociocentric measures of network centrality. *Social Networks, 24*(4), 407–422.

Newman, M. E. (2003). Ego-centered networks and the ripple effect. *Social Networks, 25*(1), 83–95.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E, 69*(2), 026113.

Newman, M. E. (2006). Modularity and community structure in networks. *In Proceedings of the National Academy of Sciences, 103*(23), 8577–8582.

Newman, M. E. (2008). The mathematics of networks. *The New Palgrave Encyclopedia of Economics*.

Nunes, M. A. S., & Hu, R. (2012, September). Personality-based recommender systems: An overview. In *Proceedings of the sixth ACM conference on recommender systems* (pp. 5–6).

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks, 32*(3), 245–251.

Okamoto, K., Chen, W., & Li, X. Y. (2008). Ranking of closeness centrality for large-scale social networks. In *International workshop on frontiers in algorithmics* (pp. 186–195). Springer Berlin Heidelberg.

Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications, 39*(11), 10059–10072.

Pereira, A. L. V., & Hruschka, E. R. (2015). Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowledge-Based Systems, 82*, 11–19.

Schröder, G., Thiele, M., & Lehner, W. (2011). Setting goals and choosing metrics for recommender system evaluations. *UCERSTI2 workshop at the 5th ACM conference on recommender systems* (Vol. 23, p. 53). Chicago, USA.

Santos, C. P., Carvalho, D. M., & Nascimento, M. C. (2016). A consensus graph clustering algorithm for directed networks. *Expert Systems with Applications, 54*, 121–135.

Seth, A., & Zhang, J. (2008). A social network based approach to personalized recommendation of participatory media content. In *ICWSM*.

Sibaldo, M. A., de Carvalho, T., Ren, T. I., & Cavalcanti, G. (2014). Recommender system based on modularity. In *Proceedings of the 2014 recommender systems challenge (p. 58)*. ACM.

Song, Q., Kawabata, T., Itoh, F., Watanabe, Y., & Yokota, H. (2013). A file recommendation method based on task workflow patterns using file-access logs. In *International conference on database and expert systems applications* (pp. 410–417). Springer Berlin Heidelberg.

Zhou, X., Xu, Y., Li, Y., Josang, A., & Cox, C. (2012). The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review, 37*(2), 119–132.