Collaborative filtering (CF) is to predict the preferences of an active user from the preference data of his/her similar users. Thus, it is critical to exactly compute the similarity between users. In developing recommendation systems, one of the most successful and popular methodologies is 'collaborative filtering'.

However, despite its success and popularity, conventional CF methods have two drawbacks. One is ‘item sparsity’ problem, which means that as the number of items in database increases, the density of each user’s records with respect to these items will decrease [5]. When similar users have rated different items, it is hard to figure out that those users have similar preferences. The other one is ‘new item’ or ‘cold start’ problem, which means that it is impossible to recommend new or recently added items until they are rated by other users.

Our paper focuses on a probabilistic learning framework for solving the ‘item sparsity’ problem while describing a conceptual collaborative filtering method that associates users and items with concepts. The ‘item sparsity’ problem cannot be avoided as long as a CF system uses the similarity measures such as Pearson correlation coefficient and cosine similarity. This is because such a CF system requires an exact match to compute the similarity between users only with explicit feedback on items such as users’ scores and purchase logs [4]. In contrast, in our work, items and users are semantically represented as a set of domain concepts with the probabilistic framework, and we thus achieve to avoid the item sparsity problem and to improve the precision of recommendation.

2. Conceptual collaborative filtering

2.1. Concept representation

For concept representation, the proposed learning framework exploits the concept hierarchy of a comprehensive web directory such as the Open Directory Project (refer to http://dmoz.org). Thus, a concept (or category) can be described with web documents that are classified into the concept.

Definition 1 (Concepts). Let \( C = \{c_1, \ldots, c_n\} \) be a set of concepts, and \( n \) be the total number of concepts. Then, the
concept $c_i$ is defined as the average of web document term vectors under $c_i$ as follows:

$$
c_i = \frac{1}{|D_i|} \sum_{d_j \in D_i} d_j^c \quad \text{where } D_i = \{d_1^c, \ldots, d_j^c, \ldots, d_n^c\}
$$

(1)

where $D_i$ is the set of web documents that belong to the concept $c_i$, and $d_j^c$ is the $j$th document in $D_i$. Each web document $d_j^c$ is represented as the weighted term vector

$$
d_j^c = (w_{m_1}, w_{m_2}, \ldots, w_{m_{V-1}}, w_{m_{V}})
$$

where $V$ denotes the set of index terms and each weight $w_m$ for the index term $t_m$ is calculated by using the traditional information retrieval metrics: term frequency (TF) and inverse document frequency (IDF) [2].

To reduce the number of terms used, we remove the stopwords (such as articles, propositions, and conjunctions) which do not help to conceptualize the concepts, and also use the Porter stemming algorithm [6] to transform inflected words to their stem or root forms.

2.2. Probabilistic learning framework

In terms of probabilistic learning, to recommend the item $i_k$ to the user $u_i$ can be regarded as estimating the conditional probability $p(i_k|u_i)$.

That is, the system recommends several items $\text{argsmax}_{i_k \in I} p(i_k|u_i)$ to the user $u_i$ where $n$ is the number of recommended items, in which $\text{argsmax}_{x \in \mathbb{X}, F(x)} p(x)$ denotes the top-$n$ values of $x$ for sorted values of $F(x)$, assuming that the values of $x$ are discrete. The probability $p(i_k|u_i)$ measures how much an active user $u_i$ likes $i_k \in I$ although the user $u_i$ has not yet accessed the item $i_k$.

The collaborative filtering recommendation is to predict a user's preference for items by using preference information of his/her similar users, and thus, $p(i_k|u_i)$ needs to contain the random variables $u_j$ to represent all the (target) users other than $u_i$. By applying the law of total probability to $p(i_k|u_i)$, it is written as follows:

$$
\text{argmax}_{(i_k, u_i)} p(i_k|u_i) = \text{argmax}_{(i_k, u_i)} \sum_{u_j \in U} p(i_k|u_i \cap u_j) \times p(u_i \cap u_j)
$$

$$
\text{argmax}_{(i_k, u_i)} \sum_{u_j \in U} p(i_k|u_i \cap u_j) \times p(u_i \cap u_j)
$$

(2)

where $U$ is a set of users within which $u_i$ is an active user and $u_k$ is a similar user of $u_i$. $I$ is a set of items that the only similar user (i.e., $u_k$) has accessed. $p(u_i)$ is the user probability that any random user $u_i$ is selected, and for simplicity's sake, it is assumed to be equivalent for all users; that is, $p(u_i) = 1/n$ of total users. Thus, by the definition of $\text{argsmax}$ in Section 2.2, we can remove $p(u_i)$ in Eq. (2). And, $p(i_k|u_i)$ is the probability that a randomly chosen item from items that the user $u_i$ has accessed is the item $i_k$. It is estimated as $p(i_k|u_i) = \text{access}(u_i, i_k)/\text{access}(u_i)$, where $\text{access}(u_i, i_k)$ is the access count of the user $u_i$ for the item $i_k$ and $\text{access}(u_i)$ is the total access counts of the user $u_i$. The access count can be replaced with click-through count, rating score, or purchase counts depending on the properties of items. As for $p(u_i|u_k)$, we interpret it as the probability that any random user from users similar to the user $u_i$ is the user $u_k$, and thus we think that this probability can express the degree of similarity between two users.

As we intend to associate users with concepts, the probability $p(u_i|u_k)$ is re-written with regard to the concept $c_j$ as follows:

$$
p(u_i|u_k) = \sum_{c_j \in c} p(u_i|c_j) \times p(u_k|c_j) \times p(c_j) = p(u_i|u_k)
$$

(3)

where $p(u_i|c_j)$ is the probability that a randomly chosen user from users related to the concept $c_j$ is the user $u_i$. Actually, Eq. (3) can play a role of a similarity function since it is symmetric (i.e., $p(u_i|u_k)$ equals $p(u_k|u_i)$). In the context of collaborative filtering, users can be represented with the items that they selected (or bought), and thus we join the factor of the item $i_k$ into $p(u_i|c_j)$ by applying both the law of total probability and Bayes' theorem as follows:

$$
p(u_i|c_j) = \sum_{i_k \in I} p(i_k|u_i) \times p(u_i|c_j \cap i_k) \times p(i_k)
$$

$$
= \sum_{i_k \in I} p(i_k|u_i) \times p(u_i|c_j) \times p(c_j|p(i_k)) \times p(i_k)
$$

$$
= \sum_{i_k \in I} p(c_j|u_i) \times p(u_i|c_j) \times p(i_k|u_i)
$$

(4)

where $I$ denotes a set of items that an active user $u_i$ has accessed. And, $p(c_j)$, $p(u_i)$, and $p(i_k)$ are the prior probability for the concept $c_j$, the user $u_i$, and the item $i_k$, respectively, and they are assumed to be equivalent for all the value of related random variables. Consequently,

$$
p(u_i|c_j) = \sum_{i_k \in I} p(i_k|u_i) \times p(u_i|c_j)
$$

(5)

where $p(i_k|c_j)$ is the probability that a randomly chosen item from items related to the concept $c_j$ is the item $i_k$.

$p(i_k|u_i)$ is similar to $p(u_i|c_j)$ in Eq. (2). The problem is how to estimate $p(i_k|c_j)$. Suppose an item set $I = \{i_1, \ldots, i_r\}$ is a set of items and $i_k = \{w_{m_1}, w_{m_2}, \ldots, w_{m_{V}}\}$ where $w_{m_i}$ is the weighted value of term $t_m$. We want to associate an item with a concept, and thus each value of $w_{m_i}$ (i.e., $p(t_m|i_k)$) is used to define $p(i_k|c_j)$, which is written as the following proportional equation as in Eq. (5).

$$
p(i_k|c_j) \propto \sum_{v_{m_i} \in i_k} p(t_m|v_{m_i}) \times p(t_m|c_j)
$$

(6)

where $p(t_m|v_{m_i})$ can be estimated by computing $w_{m_i}/w_{t_m}$, where $w_{t_m}$ is the term weight of term $t_m$ in $i_k$, and $w_{t_m}$ denotes the total weight sum of all terms in $i_k$. Similarly, $p(t_m|c_j)$ is estimated by computing $w_{c_j}/w_{t_m}$, where $w_{c_j}$ denotes the total weight sum of all terms in $c_j$. Note that each of concepts is described by documents within each of web categories.

Finally, the probability $p(i_k|u_i)$ for recommending the item $i_k$ to the user $u_i$ is re-written by combining Eqs. (3), (4) and (6) with Eq. (2) as follows:

$$
\text{argmax}_{(i_k, u_i)} p(i_k|u_i) = \text{argmax}_{(i_k, u_i)} \sum_{u_j \in U} p(i_k|u_i \cap u_j) \times p(u_i \cap u_j)
$$

$$
\sum_{u_j \in U} \left\{ \sum_{l_m \in i_k} p(t_m|v_{m_i}) \times p(t_m|c_j) \times p(i_k|u_i) \right\}
$$

$$
\sum_{u_j \in U} \left\{ \sum_{l_m \in i_k} p(t_m|v_{m_i}) \times p(t_m|c_j) \times p(i_k|u_i) \right\}
$$

(7)

As a result, we build the prediction model for recommendation, which is composed of $p(i|u)$, $p(t|i)$, and $p(t|c)$ with the past training data.

3. Experiments

In our experiment, we have extracted 11,584 concepts from the Open Directory Project (ODP) whose domain is limited to music; that is, categories under the category /Top/Arts/Music in ODP. The 9394 users’ click-through logs from Last.fm are...
crawled for experiments. Among the users, we have assigned 50 users to active users. In addition, 22,216 music meta-data are used as items. For evaluation, we introduce two performance measures: Precision_k and InverseRankPrecision_k.

\[
\text{Precision}_k = \frac{\text{number of items the user selects in the top-k items}}{k}
\]

\[
\text{InverseRankPrecision}_k = \frac{\sum_{i=1}^{k} \frac{1}{\text{rank according to user's preference}}}{\sum_{i=1}^{k} \frac{1}{i}}
\]

Precision_k is for examining whether they are relevant to the user’s preferences for top-k items retrieved, and it represents how much the system reflects the user's general preference. And, InverseRankPrecision_k is similar to ‘mean reciprocal rank’ [7], and it represents how much the system reflects the user’s main preferences because it gives more weights to higher ranked items selected. The closer InverseRankPrecision_k reaches 1.0, the more search results are adapted to the user's main preferences. For comparison, we use Pearson correlation coefficient based CF and Cosine similarity based CF, which are denoted as PCF and CCF, respectively, and our proposed method is denoted as SCF.

Fig. 1 shows the effect of the number of users with Precision_{10} and InverseRankPrecision_{10}, where the proportion of training set is 50%. It shows the effect of the number of similar users when recommending items. The number of similar users ranges from 1 to 25. In all cases, this figure shows that the proposed SCF significantly outperforms PCF and CCF. The result confirms that SCF can reduce the sparsity of items by mapping users and items to a set of concepts. In this experiment, Precision_{10} and InverseRankPrecision_{10} are the highest when the numbers of similar users are 5 in PCF and CCF, and 10 in SCF, respectively. The number of distinct items to recommend sharply increases with the numbers of similar users in

Fig. 1. Changes in Precision_k and InverseRankPrecision_k (k = 10).

Fig. 2. Changes in Precision_k and InverseRankPrecision_k for different k.
PCF and CCF, whereas the number of distinct items slowly increases in SCF. Therefore, the numbers of similar users of PCF and CCF are smaller than that of SCF. As for PCF and CCF, even though there is a little difference in Precision_{10} of PCF and CCF, InverseRankPrecision_{10} of CCF is fairly lower than that of PCF in our experiments. This is because the Pearson correlation coefficient of PCF considers the differences in users’ preference scales.

Fig. 2 shows the changes in Precision_{k} and InverseRankPrecision_{k} from varying k when the number of similar users is 10. Since the average number of items that a user has accessed is 50 and the proportion of training set is 50% in our experiments, we increase k from 5 to 25. This figure also shows that the proposed SCF provides much better performance than PCF and CCF. To be noted is that Precision_{k} of SCF, PCF and CCF decrease as the value of k increases, but their InverseRankPrecision_{k} increase as the value of k increases. The bigger value of InverseRankPrecision_{k} means that the users can locate their preferred items among higher ranked items. This result suggests that SCF is more effective method that can accommodate users’ main preferences in developing the collaborative filtering systems.

4. Summary

Our contribution in this paper is the first attempt to a probabilistic learning framework for collaborative learning recommendation. The basic idea behind our approach is to estimate the item recommendation probability, \( p(i|u) \), by applying the law of total probability and Bayes’ theorem. By associating users and items with a set of concepts, we achieve to improve the precision of recommendation and to solve ‘item sparsity’ problem of the conventional CF. Through intensive experiments, we demonstrate that our conceptual CF method is more precise than conventional CF methods. Currently, we are investigating a hybrid collaborative filtering method that combines semantic collaborative filtering with content based filtering.

Acknowledgments

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (Grant number NIPA-2010-C1090-1031-0002).

References


Jae-woon Lee received the B.S. degree in Information and Telecommunication Engineering from Soongsil University, Seoul, Korea, in 2006. He is currently a Ph.D. candidate at School of Computer Science and Engineering at Seoul National University, Seoul, Korea. His research interests include machine learning, ontology, and personalized search.

Han-joon Kim received the B.S. and M.S. degrees in Computer Science and Statistics from Seoul National University, Seoul, Korea, in 1994 and the Ph.D. degree in Computer Science and Engineering from Seoul National University, Seoul, Korea in 2002, respectively. He is currently an associate professor at the School of Electrical and Computer Engineering, University of Seoul, Korea. His current research interests include machine learning, ontology, and intelligent information retrieval.

Sang-goo Lee received the B.S. degree in Computer Science from Seoul National University at Seoul, Korea, in 1985. He also received the M.S. and Ph.D. in Computer Science from Northwestern University at Evanston, Illinois, USA, in 1987 and 1990, respectively. He is currently a professor in School of Computer Science and Engineering at Seoul National University, Seoul, Korea. He has been the director of Center for e-Business Technology at Seoul National University, since 2001. His current research interests include e-Business technology, database systems, product information management, and ontology.