Expert Systems with Applications 36 (2009) 6181–6186

Contents lists available at ScienceDirect

Expert Systems with Applications

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

An iterative semi-explicit rating method for building collaborative recommender systems

Buhwan Jeong [a], Jaewook Lee [b,*], Hyunbo Cho [b]

[a] Data Mining Team, Daum Communications Corp, 1730-8 Odeung, Jeju 690-150, South Korea
[b] Department of Industrial and Management Engineering, Pohang University of Science and Technology (POSTECH), San 31 Hyoja Pohang, Kyungbuk 790-784, South Korea

ARTICLE INFO

Keywords:
Collaborative filtering
Data sparsity
Explicit rating
Recommender system
Semi-explicit rating

ABSTRACT

Collaborative filtering plays the key role in recent recommender systems. It uses a user-item preference matrix rated either explicitly (i.e., explicit rating) or implicitly (i.e., implicit feedback). Despite the explicit rating captures the preferences better, it often results in a severely sparse matrix. The paper presents a novel iterative semi-explicit rating method that extrapolates unrated elements in a semi-supervised manner. Extrapolation is simply an aggregation of neighbor ratings, and iterative extrapolations result in a dense preference matrix. Preliminary simulation results show that the recommendation using the semi-explicit rating data outperforms that of using the pure explicit data only.

1. Introduction

Recommender systems have gained more importance ever before as the increasing popularity of Internet and social networking, e.g., electronic commerce, Web 2.0, and web personalization. Over the last decade, they are ones of the most successful applications both in academia and in industry. Success stories can be found in recommending books and CDs at Amazon.com (Linden, Smith, & York, 2003), movies by MovieLens (Miller, Albert, Lam, Konstan, & Riedl, 2003), news by GroupLens (Konstan et al., 1997) and by MONERs (Lee & Park, 2007), ESL reading lessons (Hsu, 2008) and so forth. Nonetheless, current state-of-the-art shows that they require further improvements to make them more effective and applicable to a broader range of real-life applications. For example, developments of better methods for representing user behavior and the information about the items to be recommended, more advanced recommendation methods that incorporate various contextual information into the recommendation process and utilize multi-criteria ratings, and less intrusive and more flexible recommendation methods require to be further enhanced (Adomavicius & Tuzhilin, 2005). The paper particularly concentrates on an improvement of capturing better user behaviors, i.e., rating the user preference.

Rating for recommender systems (or collaborative filtering in particular) results in a user-item preference matrix by means of either explicit rating or implicit rating. In the explicit rating, each user examines items and assigns them rating values on a rating scale, while in the implicit rating the rating values are presumed based on the user's behaviors such as purchase of the item, access to the information content, time duration to read the content, actions (e.g., save, print, delete) applied to the content, etc. It is reported that the explicit rating captures user preferences to items more accurately than implicit rating does (Nichols, 1998). However, the latent problem of the explicit rating, i.e., data sparsity (which is usually severer than that of the implicit rating), makes it hard to manipulate the rating matrix – i.e., recommending items to an active user – in a pragmatic sense.

The paper aims to propose a novel rating method, namely semi-explicit rating (SER), to overcome the sparsity problem. The proposed method extrapolates the rating scores of unrated elements in the principle of semi-supervised learning (Jeong, Lee, Cho, & Lee, 2008; Lee & Lee, 2005, 2006, 2007), in that by manipulating a few labeled/rated elements mathematically a number of the rest unlabeled/unrated elements are estimated. Especially to enhance the recommendation accuracy, the proposed method iteratively updates the user-item preference matrix until it becomes stabilized.

The remainder of the paper is organized as follows: Section 2 addresses previous works on recommender systems, especially on collaborative filtering. Section 3 presents the details of the proposed method, followed by preliminary validations via numerical experiments in Section 4. Finally, the concluding remarks and future works are given in Section 5.

2. Related works

Due to massive diversity in algorithms and applications, this section briefly reviews the key research branches of the recommender systems and collaborative filtering relevant to this paper. For more comprehensive reviews and comparison, see references

* Corresponding author.
E-mail address: jaewookl@postech.ac.kr (J. Lee).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.07.085

such as Adomavicius and Tuzhilin (2005), Deshpande and Karypis (2004) and Candillier, Meyer, and Boullé (2007).

The recommendation problem is to maximize an active user's satisfaction by suggesting him/her a set of items from many. According to the definition by Adomavicius and Tuzhilin (2005), the user satisfaction can be formulated as a utility function u that measures the usefulness of an item g to the user c, i.e., u: $\mathbf{C} \times \mathbf{G} \rightarrow \mathbf{R}$, where $\mathbf{C}$ is the set of all users, $\mathbf{G}$ is the set of all possible items that can be recommended, and $\mathbf{R}$ is a totally ordered set in nonnegative real numbers within a certain range. Note that the sizes of both $\mathbf{C}$ and $\mathbf{G}$ are very large – up to more than millions in some cases. Then, for each user $c \in \mathbf{C}$, the objective is to choose an item $g' \in \mathbf{G}$ such that maximizes the user's utility, more formally, $\forall c \in \mathbf{C}, g'_c = \arg\max_{g \in \mathbf{G}} u(c, g)$.

The recommender systems can be commonly classified into the following three types based on how recommendations are made: *content-based recommendations*, in which the user will be recommended items similar to the ones the user preferred in the past; *collaborative recommendations*, in which the user will be recommended items that people with similar preferences liked in the past; and *hybrid approaches*, in which collaborative and content-based recommendations are mixed. First, the content-based methods utilize *user profiles* that contain information about users' tastes, preferences, and needs, and *item profiles* that are a set of attributes characterizing an item g. The techniques used in information retrieval/text mining such as *vector space model* and *term frequency/inverse document frequency (TF-IDF)* are used for these recommender systems. Second, the collaborative methods (or *collaborative filtering*) predict the utility of items for a particular user-based on the items previously rated by other users. The underlying assumption is that similar users have similar preferences. A user-item rating matrix $\mathbf{R} \subset \mathscr{R}^{|\mathbf{C}| \times |\mathbf{G}|}$ is augmented for collaborative filtering systems. According to Breese, Heckerman, and Kadie (1998), algorithms in this type can be classified into *memory-based* and *model-based* ones. The memory-based algorithms, the mathematical details of which will be provided in the next section, estimate the value of unknown rating r_{cg} for user c and item g as an aggregate of the ratings of some other users for the same item g. On the other hand, the model-based algorithms make a classifier trained from the collection of ratings, and then predict future ratings. Finally the hybrid methods are nothing but an integration of collaborative and content-based methods to avoid each other's limitations. See Adomavicius and Tuzhilin (2005) for the detailed survey and exemplary recommender systems.

Specifically the collaborative methods can be categorize into a user-based approach and an item-based approach according to the searching order. The former user-based approach, more popular at present, first finds a small group of users having similar preferences (i.e., nearest neighbors to the active user) and then suggests the items the group commonly shares (e.g., purchase, access, read, etc.). Despite its popularity, the user-based approaches have some problems in practice – data sparsity, scalability, and real-time performance (Grcar, Mladenic, Fortuna, & Grobelnik, 2006; Herlocker, Konstan, Terveen, & Riedl, 2004; Sarwar, Karypis, Konstan, & Reidl, 2001). On the other hand, the recent item-based approach directly looks for a set of items similar to an active item. It roughly consists of measuring similarity between items and then predicting a recommendation item. The item similarity is often computed in terms of cosine, correlation, and conditional probability as the user similarity, whereas the prediction employs weighted sum and regression (Herlocker et al., 2004; Lee, Jun, Lee, & Kim, 2005; Sarwar et al., 2001).

One of the most important issues in collaborative filtering for recommendation accuracy is how to prepare the user-item preference matrix. The matrix can be filled either explicitly or implicitly, and hybrid rating is also possible. The explicit rating constructs the user-item matrix with users' explicit rating scores on a certain rating scale, so that it can exactly express users' tastes and preferences. However, it has some crucial weaknesses: ambiguity in the use of appropriate scales, difficulty in providing motivation and incentives for evaluators, detecting biased and malicious evaluators, and achieving a critical mass of users to avoid data sparsity (Nichols, 1998). Users tend to rate an item more frequently if they feel it is good, and not to rate otherwise. On the other hand, the implicit rating constructs the user-item matrix by observing users' behaviors such as whether or not an action (e.g., purchase, access, save, print, reply) is performed to the item, how long they spend time on reading, for example, the item, and how many times they have browsed the item, and so on (Lee et al., 2005; Nichols, 1998). The resulting matrix is usually less sparse, but the scores are assumed/implicit thereby less informative. The explicit rating provides a better user-item matrix for plausible predictions about the interests of a user, provided that every user is even, rational, unbiased, and correct.

The focus of this paper is to overcome the data sparsity problem in the user-item matrix. Widely used ways to deal with this problem are to use dimension reduction techniques such as a naïve method to select relevant users and/or items only (e.g., eliminate sparse rows/columns from the user-item matrix), or a more sophisticated method based on linear algebra and statistical analysis such as the singular value decomposition (SVD, or named as LSA/LSI (Latent Semantic Analysis/Indexing) in many applications) and principal components analysis (PCA) (Grcar et al., 2006). These dimension reduction techniques not only resolve the data sparsity and scalability problems, but also improve recommendation accuracy. In addition, the item-based collaborative filtering is known to be very effective in dealing with such sparse data (Grcar et al., 2006; Sarwar et al., 2001). Other approaches include horting, clustering, and Bayesian networks (Grcar et al., 2006). Nonetheless, the original matrix still remains sparse.

3. Semi-explicit rating and recommendation prediction

This section presents a novel extrapolation method, namely semi-explicit rating (SER), that estimates unrated elements in the user-item preference matrix. The method is based on the semi-supervised learning principle, in that a number of unrated elements are filled by numerical inference of a few (sparse) explicit ratings.

3.1. Basic idea to extrapolate unrated elements

The user-item preference matrix $\mathbf{R}(=[r_{ij}]) \subset \mathscr{R}^{N \times M}$ contains N users' preferences to M items, i.e., an element r_{ij} represents user i's rating of item j, as shown in Fig. 1. To extrapolate an unrated element r_{ij}, we employ the memory-based approaches that infer the rating from neighbor users' ratings r_{lj} by a formulation of $r_{ij} = f(r_{lj}, Sim^{\mathrm{U}}(i,l))$, where $l(\neq i \,\&\, \leqslant N)$ is the index of the users who rated the active item j, $f(\cdot)$ is an aggregation function, and $Sim^{\mathrm{U}}(i,l)$ is the similarity between users i and l. Some examples of the aggregation function are

$$r_{ij} = \kappa_c \sum_l Sim^{\mathrm{U}}(i,l) \times r_{lj} \tag{1}$$

$$r_{ij} = \bar{r}_{i,} + \kappa_c \sum_l Sim^{\mathrm{U}}(i,l) \times (r_{lj} - \bar{r}_{i,}), \tag{2}$$

where multiplier κ_c serves as a normalizing factor and is usually selected as $\kappa_c = 1/\sum_l |Sim^{\mathrm{U}}(i,l)|$, and where $\bar{r}_{i,}$ in (2) is the average rating of the active user i. Eq. (1) is the most common aggregation function where the similarity measure $Sim^{\mathrm{U}}(i,l)$ is used as a weight, but it has a shortcoming in that different users may use different