Expert Systems with Applications 41 (2014) 2754–2761

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices

Meng-Lun Wu, Chia-Hui Chang *, Rui-Zhe Liu

Computer Science and Information Engineering, National Central University, No. 300, Jhongli City, Taoyuan, Taiwan

ARTICLE INFO

Keywords:
Co-clustering
Collaborative filtering
Mutual information
Recommender system
Augmented data

ABSTRACT

Recommender systems have become an important research area because of a high interest from academia and industries. As a branch of recommender systems, collaborative filtering (CF) systems take its roots from sharing opinions with others and have been shown to be very effective for generating high quality recommendations. However, CF often confronts the sparsity problem, caused by fewer ratings against the unknowns that need to be predicted.

In this paper, we consider a hybrid approach that combines content-based approach with collaborative filtering under a unified model called co-clustering with augmented matrices (CCAM). CCAM is based on information-theoretic co-clustering but further considers augmented data matrices like user profile and item description. By presenting results with a reduced error of prediction, we show that content-based information can help reduce the sparsity problem through minimizing the mutual information loss of the three data matrices based on CCAM.

1. Introduction

Recommender systems have become an important research topic since the earliest collaborative filtering algorithm proposed in the mid-1990s. Over the past decade, a lot of work has been done on developing new approaches to recommender systems not only in the industry but also in academia. The interest in this area still remains high because of the needs of practical applications that help users cope with information overload and provide tailored recommendations, contents and services.

As a branch of recommender systems, collaborative filtering systems were planted from what humans have been doing for centuries-sharing opinions with others. In other words, by sharing opinions of similar peers, collaborative filtering (CF) systems predict the utility (rating) of items for a particular user based on the items previously rated by other users. Through identifying the likely preferences of a user based on the known preferences of similar users, CF have been shown to be very effective for generating high quality recommendations.

According to Adomavicius's work (Adomavicius & Tuzhilin, 2005), algorithms for collaborative recommendations can be grouped into two general classes: memory-based (or heuristic-based) and model-based. Model-based algorithms use the collection of ratings to learn a model, which is then used to make rating predictions. In contrast to model-based methods, memory-based algorithms essentially are heuristics that make rating predictions based on the entire collection of previously rated items by the users.

In particular, since collaborative systems rely only on other user recommendations (ratings), they can deal with any kind of content and recommend any items. However, collaborative systems have their own limitations. In many recommender systems, the number of ratings already obtained is usually very small compared to the number of ratings that need to be predicted. As a result, the rating matrix would be extremely sparse, making the rating data insufficient for prediction approaches such as *k*-NN search, probabilistic modeling, or matrix factorization.

A possible solution is the combination of content-based information with rating information to handle the sparsity issue as well as the cold start problem. Therefore, the goal here is to seek a framework that simultaneously considers both rating and content information. In this paper, we integrate content-based filtering with collaborative filtering using co-clustering model. The technique of co-clustering (also called bi-clustering, or two-mode clustering) is derived to cluster a two-dimension tabular data (such as rating matrix) with simultaneous clustering of the rows and columns, which is usually regarded as a good sparsity reduction strategy (Li, Yang, & Xue, 2009).

To be specific, we extend information theoretic co-clustering (Dhillon, Mallela, & Modha, 2003) to augmented matrices such that traditional content-based information can be considered for model construction. We show that co-clustering with augmented

* Corresponding author.
E-mail addresses: menglunwu@gmail.com (M.-L. Wu), chia@csie.ncu.edu.tw (C.-H. Chang), david7539@yahoo.com.tw (R.-Z. Liu).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.10.008

matrices algorithm (CCAM) provides a unified framework for combining content-based approach with collaborative filtering, that is, the augmented matrices provides content-based filtering through user profile and item genres, while the co-clustering result alleviates the sparsity problem for opinion information when applying collaborative filtering framework suggested by George and Merugu (2005).

The rest of this paper is organized as follows. In Section 2, we give an overview of related works. Co-clustering with augmented matrices algorithm will be stated in Section 3. Model-based collaborative filtering based on co-clustering will be described in Section 4. In Section 5, we present our experiments and evaluations. Section 6 concludes the paper and gives some directions for future research.

2. Related work

The research of recommender system has risen since the mid 1990s (Hill, Stead, Rosenstein, & Furnas, 1995). In the last decade, a common way to categorize recommender systems was based on the consideration of user similarity or item similarity. Content-based filtering algorithm (Billsus & Pazzani, 1998) assumes that items which have the same or similar characteristics will be rated in a similar way. As a consequence, how to extract the characteristics that best represent the items is the main challenge for content-based approaches. On the other hand, the assumption of the collaborative filtering (CF) algorithm (Sarwar, Karypis, & Riedl, 2001) is that people with similar preferences would rate items similarly. Thus, the challenge would be how to represent users.

Empirically, CF algorithms are shown to perform well and outperform content-based approaches (Adomavicius & Tuzhilin, 2005; Schafer, Frankowski, Herlocker, & Sen, 2007) when there are many people with similar interests. CF could be broadly classified into memory-based CF approaches and model-based CF algorithms. As the names suggests, memory-based CF approaches often use the nearest neighbor algorithm to find similar peers (Delgado & Ishii, 1999), while model-based CF algorithms rely on the construction of some (probabilistic) models from the user-item rating data (Li et al., 2009). For memory-based CF approaches, the superiority of CF algorithm could be attributed to the direct use of user-item rating matrix for user representation. Thus, we can also represent items based on the transpose of user-item rating matrix. In fact, Chen, Wang, and Zhang (2009) has shown that item-based CF outperforms user-based CF, where both predictions are derived from user-item rating matrix.

Although memory-based CF approaches directly provide information for predicting ratings, a major challenge arises when the provided rating matrix is sparse. That is, if the number of ratings by users are rare, then the amount of information that can be provided to represent users or items will not be enough. In order to cope with the sparsity issue (Sarwar, Karypis, & Riedl, 2000), model-based CF approaches have been proposed in the literature, of which the commonly ones are those based on co-clustering, matrix factorization and user-item rating model.

For example, Dhillon et al. (2003) proposed information theoretic co-clustering algorithm (ITCC), which utilized the information theory (Cover & Thomas, 2006) to model the user-item matrix, and minimize the mutual information loss to generate good co-clusters. Ding, Li, Peng, and Park (2006) proposed orthogonal non-negative matrix tri-factorization method for clustering (ONMTF) by minimizing the error of the estimated user-item rating matrix based on the orthogonal condition to avoid negative values.

There are also hybrid approaches that combine the model-based method with the memory-based method. Chen et al. (2009) combined the model-based method, ONMTF, (Ding et al., 2006) with user-based CF and item-based CF to achieve better performance. The prediction function of a given testing pair ($user_i$, $item_j$) is shown below:

$$r_{ij} = \alpha r_{ij}^{(model)} + (1-\alpha)\beta r_{ij}^{(u)} + (1-\alpha)(1-\beta) r_{ij}^{(a)} \quad (1)$$

where $r_{ij}^{(model)}$ is the prediction results of ONMTF for user i and item j; $r_{ij}^{(u)}$and $r_{ij}^{(a)}$ are user-based and item-based prediction using k-nearest neighbors CF, and the fusion coefficients α and β are weights that lie in the closed interval $[0,1]$.

George and Merugu (2005) studied a special case of the weighted Bregman co-clustering algorithm (Banerjee, Dhillon, Ghosh, Merugu, & Modha, 2007) by minimizing the error of the original rating matrix A with approximated rating matrix $\hat{A}$. The approximated rating of a given ($user_i$, $item_j$) pair in $\hat{A}$ was defined as

$$\hat{A}_{ij} = A_{gh}^{COC} + (A_i^R - A_g^{RC}) + (A_j^C - A_h^{CC}) \quad (2)$$

where g and h are the user cluster index for user i and item group index for item j, A_i^R is the average rating of user i and A_j^C is the average rating of item j. A_{gh}^{COC}, A_g^{RC} and A_h^{CC} are the average ratings of corresponding co-cluster, user cluster and item group respectively. Comparing both frameworks, Chen et al. utilized the linear combination and proper parameter tuning to achieve better performance, while George et al. solved the parameter tuning problem of model-based CF with the minimization of prediction error.

In this paper, we proposed a unified framework for integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. The proposed method can be divided into training phase and testing phase. For the training phase, we use our proposed CCAM algorithm, which considers the content-based approach, to generate the proper co-clustering results. For the testing phase, we combine the co-clustering results with scalable collaborative filtering framework to make prediction. In addition, for unknown users or new items without rating history, the challenge is the cold start problem. For such a problem, the intuitive idea is use the average rating of items (new user-old item) or average rating of users (old user-new item) or global average rating (new user-new item), to assist the prediction.

3. Co-clustering with augmented matrices

In this section, we introduce Co-Clustering with Augmented Matrix (CCAM) to generate a middle layer data format in order to alleviate sparsity problem by incorporating augmented data matrices. Assume that we have the clicking (rating) matrix as well as item feature and user profile data as follows.

Let A, and U be discrete random variables such that A denotes the items, ranging from $\{a_1,\ldots,a_m\}$, and U represents the users, ranging from $\{u_1,\ldots,u_n\}$. We also assume the existence of item and user information: S denotes the item features with categorical attributes, ranging from $\{s_1,\ldots,s_r\}$ and L denotes the user profiles with categorical attributes, ranging from $\{l_1,\ldots,l_v\}$. Let $f(A,U)$ denote the joint probability distribution of the $m \times n$ item-user matrix $\mathbf{R}$, $g(A,S)$ represent the joint distribution of the $m \times r$ item feature matrix $\mathbf{F}$, and $h(U,L)$ indicate the joint probability distribution of the $n \times v$ user profile matrix $\mathbf{P}$. For example, the Movielens data contains movie-user rating matrix $\mathbf{R}$, movie genre matrix $\mathbf{F}$ and user profile matrix $\mathbf{P}$. Note that the all three matrices $f(A,U)$, $g(A,S)$ and $h(U,L)$ are normalized in order to represent joint probability distribution.

We are interested in simultaneously clustering or quantizing A into k disjoint clusters and U into l disjoint clusters via clustering function C_A and C_U which map each item a_i/ user u_j to some item cluster $C_A(a_i)$/ user group $C_U(u_j)$, respectively.

$$\begin{aligned} C_A &: \{a_1,\ldots,a_m\} \longrightarrow \{\hat{a}_1,\ldots,\hat{a}_k\} \\ C_U &: \{u_1,\ldots,u_n\} \longrightarrow \{\hat{u}_1,\ldots,\hat{u}_l\} \end{aligned} \quad (3)$$