Expert Systems with Applications 42 (2015) 3396–3409

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Constructing compact and effective graphs for recommender systems via node and edge aggregations

Sangkeun Lee [a,*], Minsuk Kahng [b], Sang-goo Lee [c]

[a] Computational Science and Engineering Division, Oak Ridge National Laboratory, United States
[b] College of Computing, Georgia Institute of Technology, United States
[c] Department of Computer Science and Engineering, Seoul National University, Republic of Korea

ARTICLE INFO

Article history:
Available online 10 December 2014

Keywords:
Graph
Heterogeneity
Recommendation
Aggregation
Random-walk
Ranking

ABSTRACT

Exploiting graphs for recommender systems has great potential to flexibly incorporate heterogeneous information for producing better recommendation results. As our baseline approach, we first introduce a naïve graph-based recommendation method, which operates with a heterogeneous *log-metadata* graph constructed from user log and content metadata databases. Although the naïve graph-based recommendation method is simple, it allows us to take advantages of heterogeneous information and shows promising flexibility and recommendation accuracy. However, it often leads to extensive processing time due to the sheer size of the graphs constructed from entire user log and content metadata databases. In this paper, we propose node and edge aggregation approaches to constructing compact and effective graphs called '*Factor-Item* bipartite graphs' by aggregating nodes and edges of a *log-metadata* graph. Experimental results using real world datasets indicate that our approach can significantly reduce the size of graphs exploited for recommender systems without sacrificing the recommendation quality.

1. Introduction

Recommender systems aim to help users find unobvious, but preferable information hidden in the long-tail. There have been various research efforts regarding recommender systems by both industry and academia. So far, finding out an effective utility function $u : User \times Item \rightarrow Utility$ which estimates users' preferences on unseen items using based on their prior ratings is the most common problem formulation in the research domain of recommender systems. Thanks to its simplicity and clear formalized definition, it motivated many scientists and practitioners to produce a wide range of outcomes including algorithms, systems, tools and so on; however, it has limitation because it is not trivial to incorporate heterogeneous information such as contextual information, metadata of items or users, and relationships among entities.

To resolve such limitation, several context-aware recommender systems have been proposed (Panniello, Tuzhilin, & Gorgoglione, 2014; Baltrunas & Ricci, 2014; Codina, Ricci, & Ceccaroni, 2013; Adomavicius, Tuzhilin, & Zheng, 2011; Kahng, Lee, & Lee, 2011a; Lee, Park, Kahng, Lee, & Lee, 2010; Adomavicius, Sankaranarayanan, Sen, & Tuzhilin, 2005). On the other hand, several recommender systems which exploit graphs as their underlying data structure also have been proposed (Demovic et al., 2013; Lee, Park, Kahng, & Lee, 2013; Xiang et al., 2010; Cheng, Tan, Sticklen, & Punch, 2007; Gori & Pucci, 2007; Fouss, Pirotte, Renders, & Saerens, 2007; Huang, Chung, Ong, & Chen, 2002). Majority of these existing graph-based approaches mainly focused on improving recommendation quality, but more important advantage of exploiting graph structures for recommender systems is that we can achieve the flexibility of incorporating heterogeneous information into the recommendation process.

In this paper, as our baseline, we introduce a naïve graph-based recommendation method which exploits a *log-metadata* graph as its underlying data structure. A *log-metadata* graph is a heterogeneous graph which is a type of graph composed of multiple types of nodes and edges. To construct a *log-metadata* graph, the naïve method transforms tuples of user log and content metadata tables into nodes and foreign key relationship between tuples into edges, respectively (Aditya et al., 2002). A log table implicitly implies users preferences on items, since a users repeated item consumptions on the same or similar items are valuable information for recommendation. In addition, metadata table of items includes information that can be used to identify the similarity among the items. Note that log and metadata tables are prevalent in many application databases, and there are a number of examples where we can apply the approach (e.g., hotel recommendation based on previous booking history, news recommendation based on news

* Corresponding author.
E-mail addresses: lees4@ornl.gov (S. Lee), kahng@gatech.edu (M. Kahng), sglee@snu.ac.kr (S.-g. Lee).

http://dx.doi.org/10.1016/j.eswa.2014.11.062
0957-4174/

click logs, menu recommendation for a customer using his previous order history, etc.). Graph structure provides great flexibility so that any new type of nodes or edges can be incorporated without modifying the database schema unlike the relational model. Then, recommendation results are generated by ranking nodes in the constructed log-metadata graph, exploiting *Personalized PageRank* (Haveliwala, 2003) for a given recommendation query node (e.g., an active user node).

Although the naïve method is simple and it is mainly based on existing approaches, it enables us to flexibly incorporate heterogeneous information into the recommendation procedure and produce promising recommendation results. However, we noticed that the size of *log-metadata* graph is often very large because the graph is constructed from entire user log and content metadata databases. Processing such large graphs requires extensive computing power and processing time. To resolve this problem, we present a recommendation method based on *Factor-Item* bipartite graphs constructed by aggregating nodes and edges of a *log-metadata* graph.

Fig. 1 shows an example of constructing a *Factor-Item* bipartite graph from a *log-metadata* graph. Performing node and edge aggregation on a *log-metadata* graph will generate a reduced, smaller-sized graph so that we can perform recommendation more efficiently. Recommendation service providers can define *factors* that decide how to aggregate nodes and edges of a *log-metadata* graph. The naïve method's recommendation results generated by the *Personalized PageRank* (Haveliwala, 2003) are deterministic for given queries, meaning that the method will not reflect service providers' different intentions depending on their needs or situations. Like the naïve recommendation method, we use *Personalized PageRank* (Haveliwala, 2003) to define a utility function for scoring and ranking nodes; however, depending on different factors, we can construct various *Factor-Item* bipartite graphs generating recommendations with different semantics (e.g., collaborative filtering, content-based filtering, and context-aware collaborative filtering).

To validate our proposed methods, we compare our two proposed graph-based recommendation methods with the other existing recommendation methods using real datasets. The experimental results show that our proposed methods show better or comparable recommendation quality compared to existing recommendation methods. By constructing *Factor-Item* bipartite graphs, we could significantly reduce the size of exploited graph size compared to the naïve method without sacrificing the recommendation results. In the best case, it even generates better recommendation performance in terms of accuracy.

2. Related work

Most traditional recommender systems (e.g., collaborative filtering (CF) Herlocker, Konstan, Borchers, & Riedl, 1999; Koren, 2008; Cremonesi, Koren, & Turrin, 2010, or content-based filtering (CBF) Pazzani & Billsus, 2007; Whitman et al., 2002) operate in two-dimensional space *User* $\times$ *Item*. However, in many real world applications, it is not sufficient to assume this limited world model that consists of only two types of entities *User* and *Item*. There have been a number of studies on incorporating heterogeneous information into recommender systems.

Since Adomavicius et al. (2005) introduced the concept of multidimensional recommendation by generalizing the recommendation problem to the problem of dealing with multidimensional space $D_1 \times \cdots \times D_n$, several multidimensional recommendation approaches have been proposed. Adomavicius et al. proposed a reduction-based CF (Adomavicius et al., 2005). It first filters out ratings that do not match the current context. Then, conventional two-dimensional CF algorithms are performed on the reduced space. Disjunction-based CF (Lee et al., 2010) was proposed to tackle the sparsity problem which can be caused by the reduction-based CF. In this approach, instead of using all context dimensions at the same time, the algorithm disjunctively uses each contextual dimension and merges the results. Kahng et al. (2011a) proposed a similar approach by taking a Learning-to-Rank approach to compute the weights of contextual effects on recommendation. By getting hints from the matrix factorization based CF approaches, several tensor factorization techniques for multidimensional cases have been introduced (Rendle, 2010; Karatzoglou, Amatriain, Baltrunas, & Oliver, 2010); however, the complexity of the algorithm is exponential to the number of dimensions, which makes the algorithm infeasible in many real world cases.

Apart from the multidimensional approaches, several graph-based recommender systems have been introduced. In detail, Fouss et al. (2007) modeled the recommendation problem as measuring dissimilarities between nodes of a people-movie bipartite graph. The authors compared several random walk based quantities (e.g. L+ Schölkopf & Smola, 2002, average commute time Göbel & Jagers, 1974) for recommendation. Gori and Pucci (2007) presented a graph-based method named *ItemRank* which is a random-walk based scoring algorithm. It operates with a movie network linked by edges whose weights are assigned to be the number of users who watch both movies which the edge connects.

Later, researchers presented several approaches to exploiting graphs for dealing with multidimensional recommendation or combining different recommendation techniques. Xiang et al. (2010) presented a graph-based recommendation which aims to improve recommendation accuracy by mixing users' long-term and short-term preferences. Lee, Song, Kahng, Lee, and Lee (2011) presented how to construct and use a bipartite graph for more general multidimensional recommendation. Demovic et al. (2013) showed graph traversal algorithms designed for movie recommendation using movie graphs.

Flexibility is the capability of enabling easy customization of incorporating heterogeneous information and generate various

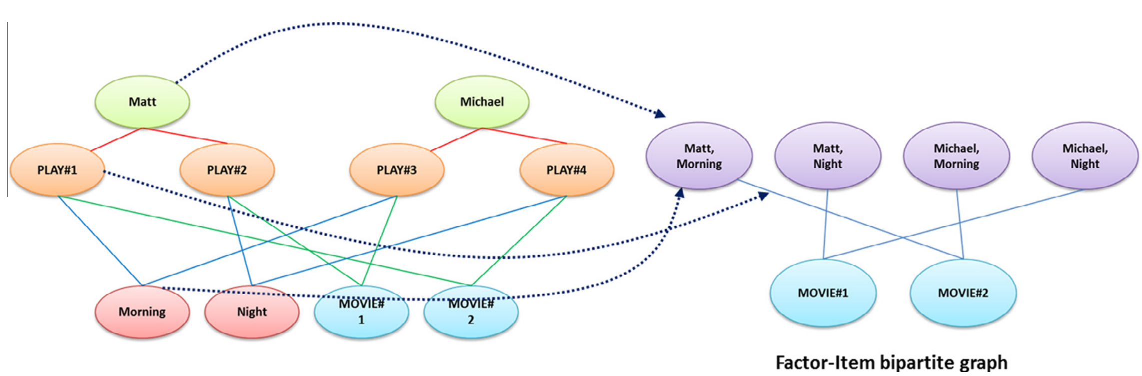

Fig. 1. Constructing a *Factor-Item* bipartite graph by node and edge aggregations.