Information Processing and Management 49 (2013) 688–697

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

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

Cluster searching strategies for collaborative recommendation systems

Ismail Sengor Altingovde [a,*,1], Özlem Nurcan Subakan [b,1], Özgür Ulusoy [c]

[a] L3S Research Center, Hannover, Germany
[b] Department of Computer & Information Science & Engineering, University of Florida, USA
[c] Computer Engineering Department, Bilkent University, Ankara, Turkey

ARTICLE INFO

Article history:
Available online 3 November 2012

Keywords:
Collaborative filtering
Clustering
Inverted index

ABSTRACT

In-memory nearest neighbor computation is a typical collaborative filtering approach for high recommendation accuracy. However, this approach is not scalable given the huge number of customers and items in typical commercial applications. Cluster-based collaborative filtering techniques can be a remedy for the efficiency problem, but they usually provide relatively lower accuracy figures, since they may become over-generalized and produce less-personalized recommendations. Our research explores an individualistic strategy which initially clusters the users and then exploits the members within clusters, but not just the cluster representatives, during the recommendation generation stage. We provide an efficient implementation of this strategy by adapting a specifically tailored cluster-skipping inverted index structure. Experimental results reveal that the individualistic strategy with the cluster-skipping index is a good compromise that yields high accuracy and reasonable scalability figures.

1. Introduction

Collaborative filtering (CF) is one of the most widely encountered techniques in generating recommendations, which formalizes the notion of "word of mouth" in daily life. Ranging from non-profit Web sites to highly competitive e-commerce giants, players of digital society investigate a lot on this technology, so that they can identify and present the most interesting, relevant or enjoyable items, say a movie, album, book, news story, radio/television show, research paper or even friends (in social networking applications), for their users.

In a nutshell, a CF system is based on determining and aggregating the "votes" of like-minded users for an "active user", so that it can make useful recommendations to the active user (Adomavicius & Tuzhilin, 2005). It works over a database of ratings for the items provided by users, either explicitly (e.g., by voting for a movie) or implicitly (e.g., by clicking on a particular hyperlink). The logic behind collaborative filtering systems is that each user belongs to a community of like-minded people; hence the items favored by these users can be used to form predictions or suggestions. A prediction is the system's opinion for an item that is explicitly asked by the user (e.g., should I go to the movie "Harry Potter and the Deathly Hallows"?) and usually expressed in the form of a score in the same scale with the users' ratings. A suggestion list is a ranked list of items that the system believes the user may be interested in. Collaborative filtering has been successfully used in domains such as recommending movies or songs, where the information content is not easily parse-able and traditional information filtering techniques are difficult to apply (see Adomavicius & Tuzhilin, 2005 for an exhaustive survey).

[*] Corresponding author.
E-mail address: ismaila@cs.bilkent.edu.tr (I.S. Altingovde).
[1] Work done while the author was at Bilkent University.

0306-4573/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ipm.2012.07.008

In the literature, CF methods using clustering are well-understood to improve scalability, however the findings in terms of the recommendation accuracy are somewhat inconclusive. In many works (e.g., Breese, Heckerman, & Kadie, 1998; Linden, Smith, & York, 2003; Sarwar, Karypis, Konstan, & Riedl, 2002), once the clusters of users are formed, neighborhood computation stage compares the active user with each of the cluster representatives (i.e., a virtual user including aggregated votes of all users in the cluster) to obtain the most similar clusters. In the next step, the entire clusters (or, equivalently the representatives) are used to generate the recommendation. However, the results reported in some other (e.g., Xue et al., 2005) studies reveal that, once most similar clusters are determined, it is better – in terms of accuracy – to "look into" these clusters to retrieve the most similar users from the clusters. And then, the recommendations can be generated by aggregating the votes of these "real" users but not the cluster representatives. In this paper, we call the former approach *aggregating strategy* and the latter *individualistic strategy*, to denote how clusters are actually used for recommendation generation. Being a more accurate approach, the individualistic strategy requires user-by-user comparisons for each similar cluster, which can hurt the scalability improvements provided by clustering, if done in a straightforward manner. Note that, an alternative to clustering users is constructing item clusters for the item-similarity based CF approach (Sarwar, Karypis, Konstan, & Riedl, 2001). Methods described in our study are potentially applicable to this case, as well.

In this paper, we focus our attention on *cluster-based* CF. As a contribution; we present a specially tailored inverted index structure originally proposed for information retrieval over clustered collections (Altingovde, Can, & Ulusoy, 2006; Altingovde, Demir, Can, & Ulusoy, 2008; Can, Altingovde, & Demir, 2004) and adapt it into the CF framework. The, so-called, *cluster-skipping* inverted index is intended to allow applying the individualistic strategy without too much degradation in the efficiency. That is, the proposed index structure makes it possible to look into the user clusters during the neighborhood formation stage so that the "actual users" can form the neighborhood and subsequently be used during recommendation generation, instead of the virtual ones, i.e., cluster representatives. We envision that this strategy would improve accuracy of cluster-based CF methods, while retaining their advantages in terms of scalability. We further investigate the performance of the proposed approach for cluster-based *hybrid filtering* (HF) systems, which makes use of content-based features, as well. Our experiments based on a large publicly available dataset justify the use of cluster-skipping index for cluster-based CF scenarios.

The rest of the paper is organized as follows: Section 2 presents related work on CF, with special emphasis on the works about cluster-based CF. In Section 3, we describe a baseline CF system, which is based on the *k*-nearest-neighbor approach and employs an inverted index for improving scalability (as proposed in Cöster and Svensson (2002)). We discuss our adaptation of a special inverted index structure for efficient and accurate cluster-based CF in Section 4. Next, in Section 5, we present a cluster-based HF system, which combines content-based filtering with CF by a two-stage clustering, i.e., first clustering the items and then imposing user clusters on top of the item clusters. Experimental results are presented in Section 6. Section 7 concludes the paper and points future work directions.

2. Related work

Collaborative filtering (CF) and content-based recommendation are two important classes of recommender systems. In a recommender system, a set of users are expected to rate a set of items (Su & Khoshgoftaar, 2009; Zhan et al., 2010). CF takes the rating values into account to make recommendation, while content-based recommender systems use the features of users and items (Li & Jin, 2003; Su & Khoshgoftaar, 2009). Hybrid approaches also exist combining the features of CF and content-based recommendation (Burke, 2002; Choi, Jeong, & Jeong, 2010; de Campos, Fernández-Luna, Huete, & Rueda-Morales, 2010; Li, Myaeng, & Kim, 2007). More recently, in addition to the users and items, external ratings (Umyarov & Tuzhilin, 2011) (aggregated from the external sources) and contextual information (such as time and place) that can be obtained either explicitly, implicitly or via data mining techniques are also incorporated into recommendation process (Adomavicius, Mobasher, & Ricci, 2011; Palmisano, Tuzhilin, & Gorgoglione, 2008).

Traditional CF algorithms can be classified into memory-based and model-based algorithms (Breese et al., 1998). For memory-based CF systems, generating a recommendation for an active user involves two stages. During neighborhood formation stage, the system finds the other users (typically at most *k* of them) that are most similar to the active user with respect to their votes on common items, i.e., similar likes and dislikes. During the actual recommendation stage, the system aggregates the information from the user's neighborhood and either provides a prediction score for some particular item or outputs a recommendation (suggestion) list of promising unseen items for the active user. Subsequently, memory based approaches can be equivalently called *k*-nearest-neighbor (*k*-NN) collaborative filters.

k-NN computation is dynamic and immediately reacts to changes in the user database. Every new rating added to the user database is included in the neighborhood calculation, since similarities between users are calculated in memory when needed. On the other hand, this dynamicity leads to an important drawback for these systems. The necessity of comparing all users to an active user on the fly creates a significant efficiency bottleneck for large systems with hundreds of thousands of users, tens of thousands of items and millions of rating scores. So, scalability is an important handicap of pure *k*-NN approaches. Several optimizations for improving *k*-NN performance were proposed, such as precomputing neighborhoods and sampling (Adomavicius & Tuzhilin, 2005).

As an alternative approach to solve the scalability problem, model-based collaborative filtering algorithms develop an internal model of the available user ratings database by using approaches that are widely employed in statistics and machine