Physica A 490 (2018) 690–701

Contents lists available at ScienceDirect

Physica A

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

A reconsideration of negative ratings for network-based recommendation

Liang Hu [a], Liang Ren [b], Wenbin Lin [c,d,*]

[a] Department of Mathematics, Sichuan University, Chengdu 610065, China
[b] Southwestern Institute of Physics, Chengdu 610041, China
[c] School of Mathematics and Physics, University of South China, Hengyang, 421001, China
[d] School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

HIGHLIGHTS

- Incorporated the negative ratings into the NBI algorithm.
- Clarified the role of negative ratings in detail.
- Proposed an approach to solve the problem about how to exploit negative ratings in a simple way, and achieved very promising results.

ARTICLE INFO

Article history:
Received 13 November 2016
Received in revised form 3 June 2017

Keywords:
Recommender systems
Bipartite network
Negative ratings

ABSTRACT

Recommendation algorithms based on bipartite networks have become increasingly popular, thanks to their accuracy and flexibility. Currently, many of these methods ignore users' negative ratings. In this work, we propose a method to exploit negative ratings for the network-based inference algorithm. We find that negative ratings play a positive role regardless of sparsity of data sets. Furthermore, we improve the efficiency of our method and compare it with the state-of-the-art algorithms. Experimental results show that the present method outperforms the existing algorithms.

1. Introduction

The information age makes our life easier in various ways, but we also increasingly suffer from the information overload problem. Search engines help people to find desired information from huge data. However, when the users do not know the appropriate keywords to describe what they want, search engines will be incapable of action [1]. This means that current search engines completely ignore the differences among users' interests or background, and consequently they are unable to provide users with personalized services. In contrast, recommender systems can satisfy idiosyncratic needs of users, so they are considered as an effective solution to the information overload problem [2–4].

By analyzing users' records, recommender systems infer their preferences and recommend relevant items to them accordingly. So far recommender systems have been widely used in many fields [5,6]. For instances, *Google News* uses click histories of active users to recommend news [7], and *Amazon.com* suggests books to users according to their purchase records [8]. Similarly, *YouTube* employs log files of users to recommend videos [9], and 60% of DVDs rented by *Netflix* are selected based on personalized recommendations [5]. Given demands like these, the design of recommendation algorithms has drawn increasing attention from engineers and scholars. Various kinds of recommendation algorithms have been

* Corresponding author.
E-mail addresses: lwb@usc.edu.cn, wl@swjtu.edu.cn (W. Lin).

http://dx.doi.org/10.1016/j.physa.2017.08.119
0378-4371/

proposed. Traditional methods include collaborative filtering [10–14], content-based analysis [15], spectral analysis [16], latent semantic models [5,17], and among others [6,18,19]. Recently, there are some important developments including social filtering [20–26], matrix factorizations [27–29], and network-based algorithms [30–33]. In particular, network-based algorithms have been demonstrated to be both highly accurate and efficient, and hence become increasingly popular [34–39].

These algorithms typically transform users' evaluations into discrete ratings, which indicate the degree of users' preferences. Take *Amazon.com* as an example, its users vote items with one to five stars representing "I hate it" to "I love it". The more stars users give to an item, the more users like that item. The number of stars is the rating value. In general, ratings lower than the scale median play a negative role when the similarity between two users is defined as the Pearson correlation on all their commonly voted items [40,41], hence we label ratings lower than the median as negative ratings. For instance, ratings of 1 and 2 are considered as negative ratings in a system with discrete ratings ranging from 1 to 5. According to Ref. [41], negative ratings contain richer information than disfavor. For instance, if a user has quite high standard toward items of her interest, she may give negative ratings as well. In this case, negative ratings mean both disfavor and relevance [42], with the former representing negative effects and the latter indicating positive effects.

However, negative ratings are ignored by many network-based algorithms. For example, Refs. [31,35] delete negative ratings from data sets. Two main reasons are responsible for this neglect. First, negative ratings play an ambivalent role. As a result, it is difficult to identify their effects to be negative or positive. Second, some variables have to be introduced to assign weight and resource for negative ratings. This increases the computational complexity of recommendation algorithms considerably.

In this paper, we firstly analyze the weighted network-based recommendation algorithm proposed in Ref. [41], which can distinguish the contributions of positive ratings from those of negative ratings, and we find there exists one flaw in that algorithm. Then, we modify the weighted network-based recommendation algorithm to overcome this flaw, and eliminate the redundant correlation information which may lead to some distortions [43]. With these improvements in place, we can study the contributions of negative ratings. Our simulation results based on two benchmark data sets, *MovieLens* and *Netflix*, indicate that negative ratings play a positive role regardless of sparsity of data sets, after the redundant correlations have been eliminated. Finally we improve the efficiency of our algorithm by removing two variables, and validate its performance. Experimental results show that this new algorithm can not only be more accurate, but also generates more diverse and novel recommendations, compared with other network-based algorithms.

2. Algorithm

A recommender system can be described by a user–item bipartite network $G(U, O, E)$ [44], which consists of a set of users $U = \{u_1, u_2, \ldots, u_N\}$, a set of items $O = \{o_1, o_2, \ldots, o_M\}$, and a link set E [45]. Here Latin letters represent users and Greek letters denote items. Consider a system with N users and M items, and denote the adjacent matrix as $W = \{w_{i\alpha}\} \in \mathbf{R}^{N,M}$. For a specific user u_i, we assign an initial resource vector $\vec{f_i}$ with its element $f_{i\alpha}$ being the resource of item o_α. The degree of user u_i, defined as the number of items that user u_i has collected [46], is denoted as $k(u_i)$. Similarly, the degree of item o_α, defined as the number of users who have collected o_α, is denoted as $k(o_\alpha)$. The aim of a recommender system is to generate a ranking list of target users' uncollected items, based on observed information.

The standard network-based inference (NBI) [45], which works on unweighted bipartite networks, is the simplest one. In this case, every edge has the same meaning, and every collected item has the same initial resource. However, negative ratings are not taken into account. The adjacent matrix W is defined as:

$$w_{i\alpha} = \begin{cases} 1 & o_\alpha \text{ is collected by } u_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In addition, $k(u_i)$ equals to the sum of the ith row of W, and $k(o_\alpha)$ equals to the sum of the αth column of W. The initial resource is set as:

$$f_{i\alpha} = w_{i\alpha}. \tag{2}$$

Given a target user u_I, one unit of resource will be allocated to each of her collected items, and zero to the uncollected ones. The standard NBI works according to a two-step resource allocation process [43,45]:

step i From item side to user side. The resource owned by an arbitrary item o_α is equally distributed to all the neighboring users who have selected this item. If user u_j is one of these users, the final resource she received can be written as:

$$g_j = \sum_{\alpha=1}^{M} \frac{w_{j\alpha} f_{I\alpha}}{k(o_\alpha)}. \tag{3}$$

step ii From user side to item side. The resource of each user is equally allocated to all of her neighboring items. The item o_β's resource, which is obtained from its neighboring users, can be described by

$$f'_{I\beta} = \sum_{j=1}^{N} \frac{w_{j\beta} g_j}{k(u_j)}, \tag{4}$$