Knowledge-Based Systems 81 (2015) 148–158

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A link prediction approach for item recommendation with complex number

Feng Xie [a,b,*], Zhen Chen [b,c], Jiaxing Shang [a], Xiaoping Feng [d], Jun Li [b,c]

[a] Department of Automation, Tsinghua University, Beijing 100084, China
[b] Research Institute of Information Technology, Tsinghua University, Beijing 100084, China
[c] Tsinghua National Lab for Information Science and Technology, Beijing 100084, China
[d] AppChina Web Search Group, Beijing 100190, China

ARTICLE INFO

Article history:
Received 29 May 2014
Received in revised form 23 December 2014
Accepted 11 February 2015
Available online 18 February 2015

Keywords:
Recommender systems
Link prediction
Complex numbers
Data sparsity
Collaborative filtering

ABSTRACT

Recommendation can be reduced to a sub-problem of link prediction, with specific nodes (users and items) and links (similar relations among users/items, and interactions between users and items). However, previous link prediction approaches must be modified to suit recommendation instances because they neglect to distinguish the fundamental relations *similar* vs. *dissimilar* and *like* vs. *dislike*. Here, we propose a novel and unified way to cope with this deficiency, modeling the relational dualities using complex numbers. Previous works can still be used in this representation. In experiments with the MovieLens dataset and the Android software website AppChina.com, the proposed Complex Representation-based Link Prediction method (CORLP) achieves significant performance in accuracy and coverage compared with state-of-the-art methods. In addition, the results reveal several new findings. First, performance is improved, when the user and item degrees are taken into account. Second, the item degree plays a more important role than the user degree in the final recommendation. Given its notable performance, we are preparing to use the method in a commercial setting, AppChina.com, for application recommendation.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Information overload creates difficulties for users. Information filtering tools, such as search engines, can help find items of interest to users, but the requirement that users specify in advance what they are looking for is still challenging [1,2]. Fortunately, recommender systems, which attempt to predict interests by mining data on past user-item interactions, can be used to identify what users need [3,4]. Consequently, recommender systems provide users with items that they are not aware of or cannot access using traditional keyword searching approaches. Recommender systems have been successfully deployed in many application settings, e.g., book, video, music, and friend recommendations on Amazon, Youtube, Pandora, and Facebook, respectively. Most of these have a client–server architecture with a centralized control mode. Some work has paved the way for developing recommender systems for personal knowledge management in collaborative environments in a distributed mode, widely used in connection with traditional knowledge management methods and tools [5,6].

An efficient recommender system can help customers find what they want quickly, and thereby save their time, improving the customer experience [7], and promoting sales [8]. As the core of a recommender system, recommendation algorithms typically take user and item attributes and user-item interactions (such as explicit ratings, and implicit browsing, purchasing or clicking-through activities) as input to anticipate user interests [9]. One of the most popular and promising recommendation algorithms, collaborative filtering (CF) provides recommendations using only user-item interactions [10,11], which can be classified as user based [12] or item based [13], depending on whether the cluster of recommendations are derived by identifying similar users based on their overlapping interactions or on similar items based on the common users who ever have expressed interests in them [14–16]. Despite its success, CF still suffers from data sparsity [17,18], where sparse user-item interactions lead to invalid user or item clustering. Some variants have been proposed to alleviate this problem [19–21]. Furthermore, this approach involves the risk that

* Corresponding author at: Department of Automation, Tsinghua University, China. Tel.: +86 15801438286.
E-mail address: xiefeng10@gmail.com (F. Xie).

http://dx.doi.org/10.1016/j.knosys.2015.02.013
0950-7051/© 2015 Elsevier B.V. All rights reserved.

increasing numbers of users will be exposed to a narrowing selection of popular items, while unpopular items that might be very relevant to users will be overlooked [22]. Several attractive solutions have been proposed to overcome these disadvantages. One is to explore the structures of user-item interaction graphs to improve recommendation performance [23–25]. More specifically, users and items are regarded as nodes in a bipartite graph, with their interactions represented as links. In this representation, the recommendation problem is converted to finding future links for each user node, and thus can be converted into a link prediction problem. Link prediction is a fundamental problem that attempts to estimate the likelihood of the existence of a link between two nodes based on observed links and node attributes [26,27]. In a typical link prediction scenario, the nodes are symmetric, and the question of which node is the subject or object is neglected. However, there are two types of nodes in a user-item graph, users and items. In addition, three types of links (user-user, user-item, and item-item) depending on different endpoint combinations coexist. We further define the type of links between two users or items as *similar* or *dissimilar* and that between users and items as *like* or *dislike*. In this classical setting, it is much more interesting to predict *like* or *dislike* links since we typically would not recommend users to users or items to items.

In this paper, we propose a novel and unified model based on complex-number representation to address this task. The *similar* or *dissimilar* links are weighted by real numbers, while the *like* or *dislike* ones are weighted by complex numbers. Since complex number j has the property that $j^2 = -1$, complex numbers provide a natural way to model the particularities of item recommendations, when the recommendation problem is being reduced to a link prediction problem. Consequently, previous link prediction algorithms can still be used conveniently without modification. We evaluate the validity and efficiency of this representation and demonstrate the performance of this recommendation approach on two real-world datasets. One of the datasets is collected from our commercial platform, where the proposed method will be implemented in the near future.

The rest of this paper is organized as follows. Section 2 provides a detailed description of the proposed algorithm. Section 3 describes experiments on two real-world datasets and discusses the experimental results. This is followed by a final section, which summarizes the findings and proposes future research directions.

2. Proposed algorithm

The method proposed in this paper is based on abstracting recommendation to a link prediction problem. Firstly, the subjects (or users) and objects (or items) in a recommender system are regarded as nodes in a graph, while the links of the graph are taken to represent the relations between different types of nodes, such as user-user or item-item similarities and user-item interactions. Then, interest prediction between a particular user and an item can be reduced to evaluating the likelihood of existence of a link between the nodes corresponding to them in the graph. Since previous link prediction methods work by taking just one type of nodes into account, we need to modify them before using them in a recommendation scenario. This can be addressed efficiently with the proposed method by introducing complex numbers into graph theory.

2.1. Basic notation

In the typical link prediction approach based recommendation scenario, input data are modelled as a directed graph $G = (V, E, \omega)$ where the set of nodes V consists of all users U and items I present in the system ($V = U \cup I$), E is the set of links that represent various relations among these nodes ($E = U \times U \cup U \times I \cup I \times I$), and ω contains all of the links' weights. Furthermore, any path is denoted by $(a_1, a_2, \ldots, a_{k+1})(a_i \in V$, where $i = 1,2,\ldots, k+1$ and k is the length of the path), a_1 and a_{k+1} are two endpoints, while $a_i(i = 2,3,\ldots, k)$ is the inner node, and there are k links along this path ($(a_i, a_{i+1}) \in E$, where $i = 1,2,\ldots, k$). When $k = 1$, the length of the path is equal to one, and it is reduced to a link with no inner nodes. In addition, we define $N_i(u)$ as the set of items that are rated by user u and $N_u(i)$ as the set of users who have expressed interest in item i. That is, $N_i(u) = \{i \,|(u, i)\in E, i \in I\}$ and $N_u(i) = \{u \,|(u, i) \in E, u \in U\}$. If two nodes are connected, this node-pair is always connected by two links, one in each direction. Then, the recommendation is reduced to predicting whether a link will exist between an item and a particular user in the graph. In this paper, we calculate an estimated score that expresses how relevant any item is to a particular user using the link prediction algorithm.

2.2. Triangle closing

There are two types of relations among nodes in a user-item bipartite graph. First, there is the similarity, $\omega_{similar}$, between two users or items, including both user-user and item-item links. Second, there is the preference, ω_{like} and $-\omega_{like}$, of the user of an item, including user-item links and item-user links, respectively, because of the need to recognize the asymmetry between user and item. That is, when there is a link with weight ω_{like} from user u to item i, there is always a reverse link with weight $-\omega_{like}$ from item i to user u and vice versa. Here, ω_{like} and $\omega_{similar}$ are normalized values just for the weights. The principle of triangle closing in this model can be illustrated as in Fig. 1.

The principle is twofold: users who have expressed the same interest in (perhaps many) common items might be similar (see Fig. 1a), similar users will have a similar interest in the same item (see Fig. 1b), and user similarity is transitive among users (see Fig. 1c). Analogously, items liked by (perhaps many) common users might be similar (see Fig. 1d), users tend to be interested in similar items (see Fig. 1e), and item similarity is also transitive among items (see Fig. 1f). These are the core ideas of CF from another perspective. Consequently, these rules can be expressed mathematically as follows:

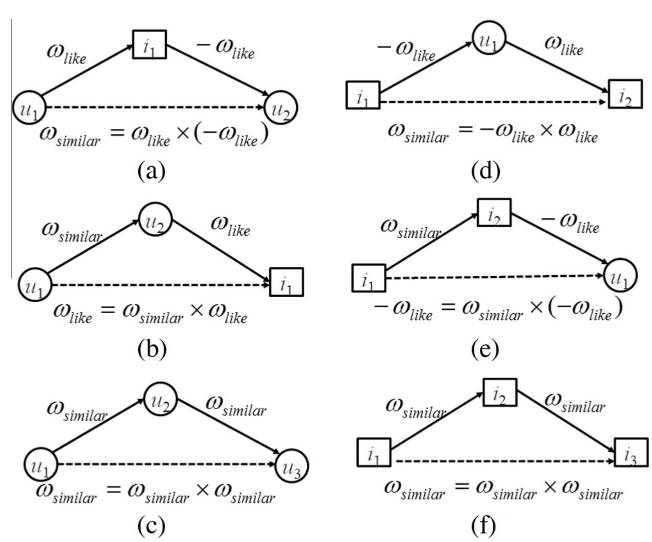

Fig. 1. The multiplication rules lead to triangle closing between the *like* and *similar* relations.

Download English Version:

https://daneshyari.com/en/article/403501

Download Persian Version:

https://daneshyari.com/article/403501

Daneshyari.com