Chapter

A Bipartite Graph-Based Recommender for Crowdfunding with Sparse Data

Hongwei Wang and Shiqin Chen

Abstract

It is a common problem facing recommender to sparse data dealing, especially for crowdfunding recommendations. The collaborative filtering (CF) tends to recommend a user those items only connecting to similar users directly but fails to recommend the items with indirect actions to similar users. Therefore, CF performs poorly in the case of sparse data like Kickstarter. We propose a method of enabling indirect crowdfunding campaign recommendation based on bipartite graph. PersonalRank is applicable to calculate global similarity; as opposed to local similarity, for any node of the network, we use PersonalRank in an iterative manner to produce recommendation list where CF is invalid. Furthermore, we propose a bipartite graph-based CF model by combining CF and PersonalRank. The new model classifies nodes into one of the following two types: user nodes and campaign nodes. For any two types of nodes, the global similarity between them is calculated by PersonalRank. Finally, a recommendation list is generated for any node through CF algorithm. Experimental results show that the bipartite graph-based CF achieves better performance in recommendation for the extremely sparse data from crowdfunding campaigns.

Keywords: crowdfunding, recommender, bipartite graph, network structure

1. Introduction

As the largest crowdfunding platform in the world, Kickstarter has attracted 8,604,863 users who participated in 230,850 campaigns with 22,525,091 investment behaviors (www.kickstarter.com). However, about 60% of the campaigns are unsuccessfully financed. The main reason is that many campaigns failed to find enough investors, rather than the ideas were not good enough [1]. Therefore, a recommender for crowdfunding is the key to solving this problem.

A survey has shown that the sparseness of user behaviors in Kickstarter is about 99.99%, leading to the commonly used recommendation algorithms inefficient. For example, collaborative filtering (CF) algorithm based on cosine similarity aims to find users who have the same preference, then calculates interest similarity, and produces recommendation list. However, it is difficult for the algorithm to find similar users on a sparse data, which is one of the main problems faced by recommender systems [2].

Faced with large-scale sparse data, network analysis algorithms are effective approaches to overcome the problem. For example, the PageRank algorithm is applicable to calculate the weight of web nodes. As a global iterative algorithm, PageRank does not distinguish the types of nodes, making it hard to improve the recommendation performance. However, an improved algorithm based on PageRank (i.e., bipartite graph model) provides ideas for us. Using bipartite graph model, we divide the network into an item-user structure, where there is no direct edge between items or between users. Then, the global similarity is calculated by bipartite graph analysis, as opposed to local similarity calculated by cosine function, and can better deal with the problem of sparse data.

Experiments show that bipartite graph model can effectively produce recommendation lists with sparse data. Furthermore, in the global iterative process of bipartite graph model, the similarity between items or between users is also calculated, in addition to the similarity between items and users. Compared with cosine function, which can only calculate adjacent users, this kind of similarity is extracted from the network, thus it is able to solve the computation problem caused by sparse data. Therefore, we propose a bipartite graph-based CF model by combining the similarity calculated by bipartite graph model with CF algorithm.

2. Literature review

2.1 Graph model

PageRank is a classic algorithm to calculate the node's weight [3, 4]. PageRank determines the importance of all web pages based on the assumption that web pages linked from high-quality pages are also high-quality. A page is given a higher weight if more high-ranking pages point to it. Prior studies have raised improved PageRank algorithms, e.g., topic-sensitive PageRank [5]; the algorithm where the linked pages are content relevant but nondirectly adjacent pages, instead of directly adjacent pages [6]. PageRank is a computing-consuming and time-consuming algorithm, and its computational efficiency can be improved by some improved algorithms [7, 8].

When the node's weight is calculated by PageRank, the link weight and the content weight are not distinguished [9]. HITS algorithm separates the quality of nodes into link authority (Hub) and content authority (Authority) [10]. Based on content authority of pages, link authority of pages is determined, and then overall evaluation of web pages is given. A good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs. This kind of mutually reinforcing relationship between hubs and authorities is applicable for the discovery of authoritative pages and automatic identification of the web structure and resources. Since there are problems of topic drift and irrelevant links in HITS algorithm, some improved methods are proposed [11, 12].

A bipartite graph is an extension of network theory and has attracted lots of attention, such as social network analysis [13]. A bipartite graph divides network nodes into two types, which is different from PageRank that treats nodes as homogeneous. Only nodes in different types are directly connected, while nodes in the same type are indirectly connected [14, 15]. The crowdfunding network can be abstracted as a bipartite graph, where one group of nodes is investors and the other group is items. The bipartite graph model can calculate the distance between nodes, such as Laplacian distance [16], though appropriate algorithms. The Laplacian matrix can measure the reachability of nodes in graph models. Since the distance between nodes is calculated in bipartite graphs, they can be transformed into the similarity between nodes [17]. Typical algorithms include mean similarity [18], and subsequent research has shown the upper and lower bounds of bipartite approximations [19]. On this basis, combined with hierarchical subgraphs, Hausdorff edit distance is proposed that can improve calculation accuracy and reduce computational complexity [20]. Visualization methods are also suggested [21]. In practice, the bipartite graph is applied to image segmentation [22]. In terms of recommender system, researcher uses aggregated bipartite graph model to reduce computational complexity of graph models, while recommendation accuracy is decreased [23].

2.2 Collaborative filtering

Collaborative filtering (CF) techniques are widely used in recommender systems [24]. Relaying on historical behaviors of users, similarity between users is calculated, and then products purchased by similar users are recommended. CF techniques are classified into item-based CF and user-based CF. User-based CF algorithm firstly identifies the user preference profile [25], next calculates the similarity based on the user preference profile, and finally applies the distance of user similarity to recommendation algorithms [26]. Since most users only have purchase behaviors for a few products, sparse data problem hinders the efficiency of recommendation [27, 28]. One solution is data clustering, which solves the problem to some extent.

How to evaluate the performance of recommender systems is a complex topic. In general, recommender systems more tend to provide a narrow recommendation list. Inspired by the Gini index, directed weighted conduction (DWC) is proposed. DWC is an evaluation metric based on bipartite graph model, which can effectively avoid recommendation congestion and greatly improve the novelty and diversity of recommendation [29].

3. Research gaps and problem definitions

Take **Figure 1** as an example, where black nodes A, B, C, and D denote users and gray nodes e, f, g and h denote items. If using the user-based cosine similarity CF algorithm, user A has adjacent users C and B. Item f is impossibly recommended to user A, because the adjacent users of A have no direct link to f. Similarly, f is also impossibly recommended to A in the item-based CF algorithm. Cosine similarity algorithm is a local algorithm, which cannot calculate the similarities of global nodes in a sparse network structure. The recommendation accuracy should be guaranteed using local similarity with dense data, but it is hard to get ideal performance in the case of sparse data.

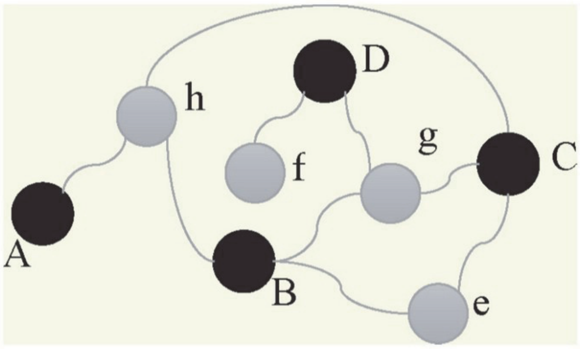

Figure 1.
The diagram of the application of CF algorithm in the network structure.

In bipartite graph algorithms, such as PersonalRank, the distance between items and users can be obtained directly. Therefore, the direct recommendation results can be obtained by transforming the distance into the similarity. For instance, the network in **Figure 1** is transformed into a bipartite graph as shown in **Figure 2**.

PersonalRank is used to calculate the bipartite graph in **Figure 2**. If a recommendation is provided to user A, iterative calculation starts at A. After 62 iterations, the calculation result converges, and the similarities between A and each item are obtained:

$$\begin{aligned} s(A,\mathrm{e}) &= 0.07709, \quad s(A,f) = 0.01791 \\ s(A,g) &= 0.09499, \quad s(A,h) = 0.26949 \end{aligned} \tag{1}$$

Except the node h with direct action to A, the recommended order of the remaining three items is g ranks firstly, e followed, and f lastly. In fact, in the calculation process, PersonalRank also repeatedly iterates to generate user similarities, but explicit output does not exist. In **Figure 2**, the implicit similarities between A and other users are:

$$s(A,B) = 0.13602, \quad s(A,C) = 0.13602, \quad s(A,D) = 0.04213 \tag{2}$$

The above similarities are different from the similarities based on cosine function or Pearson function. Local similarity between users is obtained by cosine or Pearson function (i.e., only the nodes directly adjacent to the user are calculated), while global similarity between users is obtained by bipartite graph algorithm. Taking **Figure 2** as an example, since A and D have no common actions, their similarity cannot be calculated by cosine or Pearson similarity function, or $s(A, D) = 0$. It is effective with dense data, since there are enough users with common actions and a neighborhood with a sufficient width is able to be obtained. However, in the case of sparse data, globally calculating user similarity is apparently more effective.

The research progress related to this paper is summarized in **Table 1**. For the personalized recommender for crowdfunding campaigns, although graph models have been used in present research, bipartite graph model is rarely used, especially focusing on solving the problem of sparse data in crowdfunding communities.

Crowdfunding platforms represented by Kickstarter use an *all-or-nothing* funding model, and the funding success rate is about 40%. Founders spend a lot in

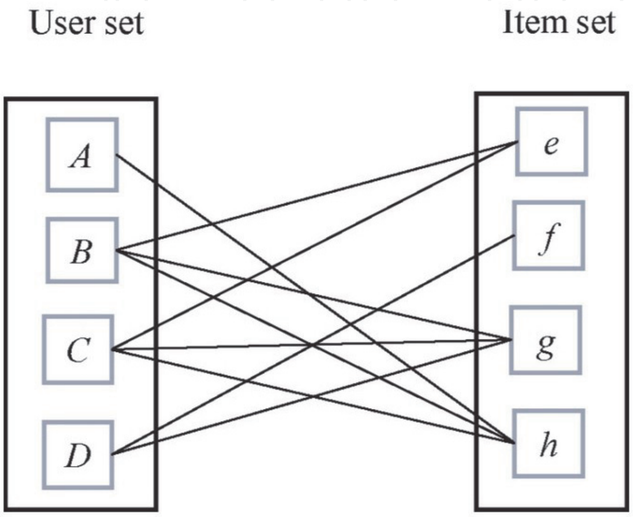

Figure 2.
The diagram of bipartite graph transformation.

Authors	Year	Conclusions	Comments
Rakesh and Choo [30]	2015	• Four research dimensions: temporal traits, personal traits, geo-location traits, and network traits • The backing habits of investors are influenced by their social circle	• Analysis is focused on project features, while recommender is just an application • Supervised machine learning lacks reporting on sparse data
An et al. [1]	2014	• Social networks can identify user preference more accurately • Different recommender strategies are adopted for different types of projects	• It is not aimed at the current situation of sparse data in crowdfunding communities • Information in social networks is needed, making operability reduced
Lu et al. [31]	2014	• Social networks can identify user preference	• There are similarities in research questions, but it is not aimed at sparse data • Identifying a large number text from social media has high cost
Stone et al. [32]	2013	• Traditional collaborative filtering algorithm is not suitable in the field of venture capital • The recommendation performance is improved by hierarchy information (group, segment, code)	• VC and crowdfunding have similarities and also many differences • There are differences between KNN and the method used in this paper
Zhou et al. [33]	2010	• Global recommendation algorithm overcomes some shortcomings of local recommendation algorithm • Nodes of "weak ties" have value in identifying user preference • Bipartite graph model can improve the diversity of recommendation	• The goal of the study is to achieve the balance between diversity and accuracy of recommendation, without considering the processing of sparse data

Table 1.
The main research progress related to this paper.

energy maintaining campaigns. A survey found that during the preparation period, it took an average of 30 minutes a day and 11 hours on weekends; during the fundraising period, it took 2–11 hours a day lasting 0.5–2 months [34]. Once the funding failed, the founder would get nothing. The reasons for failure might be the quality of the campaigns was poor or right investors had not been found. In the latter case, designing a reasonable personalized recommender system will increase the funding success rate. Therefore, taking advantages of PersonalRank in computing of bipartite graphs, combined with advantages of CF algorithms, the following research questions are proposed to investigate the recommender for crowdfunding campaigns:

1. In view of the extremely sparse data in crowdfunding communities, we extract user behaviors into the bipartite graph structure and calculate the global similarity between nodes in the graph model.

2. Depending on the node similarity matrix in bipartite graph model, we propose a bipartite graph-based CF model combined with CF algorithm to generate recommendation list for crowdfunding campaigns.

3. We conduct experiments on the dataset from Kickstarter to evaluate the effectiveness of the bipartite graph-based recommender algorithm, comparing differences between algorithms and suggesting feasible solutions.

4. Model overview

PageRank is an algorithm that measures the weight of a specific web page relative to other web pages, which is often used in page ranking. PageRank assumes that a user randomly selects a page to visit from all pages and then jumps to other pages through hyperlinks. After reaching each page, the user has two options: end here, or continue visiting by selecting a link randomly. Let *d* be the probability of continuing visiting. The user selects a hyperlink at random with the same probability from the current page to continue visiting, which is a random walk process. After many rounds of walks, the probability of visiting each page will converge to a stable value. This value is the weight of a web page. The algorithm is shown in Eq. (3):

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in in(i)} \frac{PR(j)}{|out(j)|} \tag{3}$$

$PR(i)$ is the probability of visiting page *i, d* is the probability of continuing visiting pages (i.e., the damping coefficient), *N* is the total number of pages, *in (i)* is the page set pointing to page *i* (i.e., in-links), and *out (j)* is the page set pointed by page j (i.e., out-links).

PageRank is a global algorithm, which does not distinguish the types of nodes. However, the recommender system for crowdfunding campaigns is faced with both user nodes and campaign nodes. We can only obtain the weight of nodes themselves by PageRank, rather than the similarity between nodes. Based on PageRank, the improved algorithm PersonalRank is a bipartite graph algorithm [6], which can generate personalized item list for users, as shown in the Eq. (4):

$$\begin{gathered} PR(i) = (1-d)r_i + d \sum_{j \in in(i)} \frac{PR(j)}{|out(j)|} \\ r_i = \begin{cases} 1 & , \text{ if } \; i = u \\ 0 & , \text{ if } \; i \neq u \end{cases} \end{gathered} \tag{4}$$

The difference between Eqs. (3) and (4) is that 1/*N* is replaced by r_i. In other words, initial probabilities vary in different nodes. In bipartite graph model, *u* is the target user, and Eq. (4) actually calculates the similarity of all nodes relative to node *u*.

Specifically, unlike PageRank randomly selecting a node to walk, PersonalRank starts from the special node *u* and can only walk to different types of nodes. Taking crowdfunding as an example, user nodes can only walk to campaign nodes, while campaign nodes can only walk to user nodes. After reaching a new node, the walk stops and restarts from *u* with a probability of 1-*d* or continues walking to a node in the other type with a probability of *d.* After many rounds of walk, the probabilities of visiting each node tend to be stable. Therefore, before running PersonalRank algorithm, an initial probability must be set for each node. In PageRank, if *u* is the user, let the initial probability of visiting node *u* be 1 and other nodes be 0. But in PageRank, initial probability of visiting each node is equal, and the initial probability is *1/N.*

A bipartite graph is a graph model composed of two groups of nodes with different properties, and the nodes in the same group are not connected. A bipartite graph can be defined as a network structure *G* = <*U*, *I*, *E*>, where *U* denotes the user set; *I* denotes the item set; and *E* denotes the edges of bipartite graph model.

Figure 2 is a typical bipartite graph structure, containing four users and four items. The actions between users and items are mapped as edges in the graph. For simplicity, the weights of edges are assumed the same. Take crowdfunding as an example, where U denotes investors; I denotes crowdfunding campaigns; and the edges denote users' investment behaviors in campaigns. G is actually a matrix structure, which can be calculated to obtain the global similarity by PersonalRank.

The core idea of CF is the calculation of similarity between users (user-based) or between items (item-based). The similarity algorithm commonly used is cosine function, as shown in Eq. (5):

$$similarity(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{5}$$

In bipartite graph model, the similarities between all nodes are calculated, which can be integrated with CF algorithm, and may achieve better performance than directly recommendation by bipartite graphs.

5. Experimental data and experimental settings

5.1 Experimental data

The research data was collected from Kickstarter, which contains 32,226 investment behaviors from 14,506 users which invest on 787 campaigns. This paper used an offline evaluation method to evaluate the recommender system, dividing the dataset into a training set and a test set. If a user has only one investment behavior, the recommendation list cannot be produced because if the behavior is classified into the training set, the accuracy of the recommendation list cannot be evaluated; if classified into the test set, preference similarity cannot be obtained through the user's behavior.

Data sparseness is defined as the probability of matrix elements without data, which is calculated by Eq. (6). The sparseness of the experimental data is 96–99%, that is, about 96–99% of the matrix elements in the users' behavior matrix lack values:

$$sparsity = 1 - \frac{|Behavior|}{|User| \times |Item|} \tag{6}$$

In the dataset, most users support less than five campaigns, also leading to the extremely sparseness of the dataset. Many campaigns have a small number of supporters, while popular campaigns have won a large number of supporters. Statistics show that campaigns in the dataset have one supporter at least, 9046 supporters at most, and 41 supporters on average.

5.2 Experimental settings

Firstly, the parameters are setting. PersonalRank has two parameters:

1. Convergence coefficient. According to the present research, it is set to 0.85.

2. Number of iterations. There is no fixed value, and it needs to be set depending on the data by following two methods: (a) specify the number of iterations

forcibly; and (2) judge whether the global computing result has converged, and stop iteration if converged. We integrate these two methods and use the following method for iteration setting.

Algorithm 1. Iteration setting of PersonalRank Algorithm

```
Input: network structure G
Output: computing results of PersonalRank
Define G; #Construct the network
Define max_iteration; # Define a maximum number of iterations
Define item; #Define the starting point of PersonalRank walking
Define previous_iteration = [Null]; # Predefine iteration result
For iteration in range(0,max_iteration)
    For I in G.nodes():
        Pr[i] = PersonalRank(G);
    End For
    If previous_iteration == Pr:
        Break; #Converged
    End If
    previous_iteration = Pr;
End For
Output Pr;
```

The time complexity of Algorithm 1 is $O(max\_iteration^{*}|item|)$, where *max_iteration* is the predefined number of iterations and $|item|$ is the number of nodes of items. The complexity of the algorithm means the complexity of the number of iterations, not the complexity of the complete algorithm. We tried to calculate the network, showing that the PersonalRank converges after 100 iterations.

All of the CF in experiments use item-based algorithm, for the following reasons: (1) The number of items is much smaller than the number of users so that the computing cost of the similarity between items is much lower than between users. (2) Item-based methods are used more often in practical applications due to computing convenience, such as Amazon recommender system.

The compared algorithms in this study are summarized in **Table 2**. The content-based recommender is based on the similarity of items. For instance, if a user has supported a “music” campaign, the content-based recommender algorithm assumes that the user has a greater preference for “music” campaigns. In the compared experiments, we chose six indicators to measure the similarity of campaigns: campaign category, social network of founders, funding status, number of pledge levels, minimum pledge money, and average funding amount.

Popularity-based recommender means the most popular items are directly recommended to users (user-based) or the most popular users are recommended to items (item-based). Popularity-based recommender is independent of neighbor nodes, which means the recommendation lists are the same for any users.

Two parameters need to be set in CF algorithms:

1. The number of neighbors K. K similar users (items) are selected as the source for producing recommendation lists, and the items which the users are interested in are recommended to target users.

2. The list length N. N items are recommended to target users (or N users are recommended to target items). Generally, N is set to 5 or 10, which is widely used in present studies.

	Compared algorithms	Description
1	Cosine-based CF	Collaborative filtering algorithm based on cosine similarity
2	PersonalRank	Recommendation directly using PersonalRank to calculate bipartite graphs
3	Bipartite graph-based CF	Using PersonalRank to calculate node similarity then using CF to recommend
4	Content-based recommender	Recommendation according to the content
5	Popularity-based recommender	The most popular items (users) are recommended to users (items)

Table 2.
Compared algorithms and description.

In addition to cosine similarity function, other similarity functions have also been tried. The results show that cosine similarity function performs best in the recommender for crowdfunding campaigns. Therefore, cosine-based CF is used as one of the benchmarks for comparing.

6. Experimental result

The sparseness of user behaviors is larger than 99%, and many users cannot find similar users. As a local similarity method, cosine function hardly produces recommendation lists in this situation. Thus, the similarity between any users without intersection is set to 0. **Tables 3** and **4** show the recommendation performance of CF algorithm based on cosine similarity.

From **Tables 3** and **4**, when $K = 40$ and $K = 55$, the best performances are achieved, respectively. However, on the whole, the accuracy is extremely low, which has large room for improvement. The reason is that on the extremely sparse dataset, users have few intersections, which makes it difficult to find users with similar interests, resulting in the low accuracy of recommendation.

Table 5 shows the performance of using PersonalRank to produce recommendation lists. Compared to cosine similarity CF algorithm, the accuracy of recommendation by PersonalRank has at least doubled, which indicates that on sparse data network, the global similarity algorithm can effectively solve the computing problem of node similarity and improve the recommendation accuracy.

Then we use bipartite graph-based CF algorithm. The recommendation result for $N = 5$ is shown in **Table 6**, where the algorithm achieves the best performance when $K = 30$. The recommendation result for $N = 10$ is shown in **Table 7**, where the algorithm achieves the best performance when $K = 30$. However, compared to the result of recommendation directly by bipartite graph model, bipartite graph-based CF algorithm does not perform better. It indicates that on this dataset, the accuracy of recommendation calculated by bipartite graph model is higher.

Comparing **Tables 5**–**7**, we can get the conclusion that the result of recommendation by bipartite graph model is superior to bipartite graph-based CF algorithm. The possible reasons are as follows. (1) Although bipartite graph-based CF algorithm can obtain the similarity between items (users) and generate neighbor items (users), which cannot be done by cosine similarity algorithm, the CF algorithm cannot extract enough items from the neighborhood for recommendation due to the extremely sparse data (e.g., A and B are very similar, but if B has few actions, the accuracy of the recommendation list is still quite low). Therefore, the recommender

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
5	0.07	0.25	2.60	0.871
10	0.08	0.28	2.58	0.935
15	0.09	0.30	2.52	0.96
20	0.09	0.31	2.46	1.013
25	0.09	0.31	2.38	1.035
30	0.10	0.32	2.36	1.043
35	0.10	0.33	2.34	1.048
40	**0.10**	**0.35**	**2.33**	**1.052**
45	0.10	0.33	2.33	1.054
50	0.10	0.33	2.33	1.057
55	0.10	0.34	2.32	1.054
60	0.10	0.34	2.32	1.055
65	0.10	0.34	2.32	1.056
70	0.10	0.33	2.32	1.057
75	0.10	0.33	2.32	1.057
80	0.10	0.33	2.32	1.057
85	0.10	0.34	2.32	1.057
90	0.10	0.34	2.32	1.057
95	0.10	0.34	2.32	1.058
100	0.10	0.34	2.32	1.058

Table 3.
Performance of CF algorithm based on cosine similarity (N = 5).

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
5	0.13	0.22	4.97	0.842
10	0.13	0.22	4.9	0.9
15	0.15	0.25	4.79	0.922
20	0.16	0.26	4.64	0.958
25	0.16	0.27	4.5	0.97
30	0.16	0.27	4.45	0.976
35	0.17	0.28	4.43	0.981
40	0.17	0.29	4.42	0.985
45	0.17	0.29	4.41	0.988
50	0.18	0.29	4.41	0.99
55	**0.18**	**0.3**	**4.4**	**0.99**
60	0.18	0.3	4.4	0.992
65	0.18	0.29	4.4	0.992
70	0.18	0.29	4.4	0.993
75	0.18	0.3	4.4	0.993
80	0.18	0.3	4.4	0.992

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
85	0.18	0.3	4.4	0.993
90	0.18	0.3	4.4	0.993
95	0.18	0.3	4.4	0.993
100	0.18	0.29	4.4	0.993

Table 4.
*Performance of CF algorithm based on cosine similarity (*N = *10).*

N	Recall (%)	Precision (%)	Coverage (%)	Popularity
1	0.16	1.17	1.72	1.586
2	0.28	1.02	3.17	1.545
3	0.41	0.99	4.48	1.476
4	0.52	0.93	5.65	1.434
5	0.59	0.85	6.75	1.425
6	0.67	0.81	7.81	1.415
7	0.72	0.75	8.82	1.401
8	0.80	0.72	9.80	1.382
9	0.85	0.68	10.78	1.372
10	0.90	0.66	11.76	1.365

Table 5.
Result of recommendation by PersonalRank.

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
5	0.39	0.56	7.72	1.298
10	0.38	0.55	6.22	1.468
15	0.39	0.57	5.62	1.572
20	0.39	0.56	5.40	1.607
25	0.40	0.58	5.27	1.627
30	**0.41**	**0.59**	**5.19**	**1.641**
35	0.41	0.59	5.16	1.653
40	0.40	0.59	5.10	1.654
45	0.41	0.59	5.09	1.663
50	0.40	0.59	5.08	1.655
55	0.39	0.57	5.07	1.638
60	0.40	0.58	5.04	1.643
65	0.40	0.58	5.03	1.645
70	0.40	0.58	5.02	1.647
75	0.40	0.58	5.01	1.649
80	0.40	0.58	5.00	1.65
85	0.40	0.58	5.00	1.652

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
90	0.40	0.58	5.00	1.654
95	0.40	0.58	4.99	1.655
100	0.40	0.58	4.99	1.656

Table 6.
*Result of bipartite graph-based CF algorithm (*N = *5).*

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
5	0.56	0.41	14.11	1.235
10	0.58	0.42	11.51	1.406
15	0.59	0.43	10.32	1.466
20	0.60	0.43	9.91	1.495
25	0.62	0.45	9.67	1.521
30	**0.64**	**0.47**	**9.54**	**1.532**
35	0.64	0.47	9.49	1.532
40	0.64	0.47	9.41	1.536
45	0.63	0.45	9.39	1.544
50	0.62	0.45	9.41	1.541
55	0.62	0.45	9.39	1.536
60	0.61	0.44	9.34	1.539
65	0.60	0.44	9.29	1.541
70	0.60	0.44	9.26	1.543
75	0.60	0.44	9.24	1.546
80	0.60	0.44	9.23	1.547
85	0.60	0.44	9.23	1.548
90	0.60	0.44	9.22	1.55
95	0.60	0.44	9.21	1.551
100	0.61	0.44	9.20	1.552

Table 7.
*Result of bipartite graph-based CF algorithm (*N = *10).*

performance of CF algorithm is poor. (2) Local algorithms only produce local optimal solutions, also resulting in the poor performance of bipartite graph-based CF algorithm, whereas recommendation directly by bipartite graph model is a global recommendation algorithm, which can overcome the shortage of sparse matrix.

The results of content-based recommender are shown in **Tables 8** and **9**, where the best performances are achieved when *K* = 90 and *K* = 100, respectively. The accuracy of content-based recommender is the lowest, which might be determined by the investment preference of investors on crowdfunding campaigns. For example, many investors have participated in multiple categories of campaigns, rather than focusing on one or several categories.

The comprehensive comparison result of various algorithms is summarized in **Table 10**. On this dataset, PersonalRank is the most effective in computing the node distance of bipartite graph model and converting it into similarity, followed by CF

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
5	0.01	0.04	1.82	0.846
10	0.03	0.09	1.69	0.927
15	0.03	0.11	1.54	1.029
20	0.04	0.12	1.45	1.121
25	0.04	0.14	1.36	1.243
30	0.04	0.15	1.24	1.337
35	0.05	0.16	1.14	1.427
40	0.05	0.17	1.05	1.52
45	0.05	0.17	0.95	1.607
50	0.05	0.17	0.88	1.647
55	0.05	0.17	0.83	1.685
60	0.06	0.19	0.77	1.729
65	0.06	0.20	0.72	1.763
70	0.06	0.20	0.68	1.801
75	0.06	0.20	0.65	1.839
80	0.06	0.22	0.62	1.867
85	0.07	0.22	0.59	1.901
90	**0.08**	**0.25**	**0.55**	**1.933**
95	0.07	0.25	0.53	1.957
100	0.07	0.25	0.50	1.993

Table 8.
Result of content-based recommender (N = 5).

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
5	0.02	0.03	3.45	0.819
10	0.04	0.06	3.25	0.876
15	0.05	0.08	3.02	0.936
20	0.05	0.09	2.91	0.999
25	0.06	0.10	2.76	1.082
30	0.07	0.11	2.56	1.152
35	0.07	0.11	2.38	1.232
40	0.07	0.11	2.20	1.316
45	0.07	0.12	2.03	1.393
50	0.07	0.12	1.85	1.452
55	0.08	0.13	1.71	1.507
60	0.08	0.14	1.56	1.549
65	0.09	0.15	1.45	1.581
70	0.09	0.15	1.36	1.608
75	0.10	0.16	1.29	1.645
80	0.10	0.17	1.23	1.673

K	Recall (%)	Precision (%)	Coverage (%)	Popularity
85	0.10	0.17	1.17	1.698
90	0.11	0.18	1.11	1.72
95	0.12	0.19	1.06	1.739
100	**0.12**	**0.20**	**1.01**	**1.76**

Table 9.
*Result of content-based recommender (*N = *10).*

Recommender	List length *N*	Number of neighbors *K*	Recall (%)	Precision (%)	Coverage (%)	Popularity
Cosine-based CF	5	40	0.10	0.35	2.33	1.052
Cosine-based CF	10	55	0.18	0.3	4.4	0.99
PersonalRank	5	Global	0.59	0.85	6.75	1.425
PersonalRank	10	Global	0.90	0.66	11.76	1.365
Bipartite graph-based CF	5	30	0.41	0.59	5.19	1.641
Bipartite graph-based CF	10	30	0.64	0.47	9.54	1.532
Content-based	5	90	0.08	0.25	0.55	1.933
Content-based	10	100	0.12	0.20	1.01	1.76
Popularity-based	5	—	0.335	0.528	0.035	2.90
Popularity-based	10	—	0.509	0.400	0.069	2.729

Table 10.
Comprehensive comparison result of various algorithms.

algorithm using global similarity distance, while content-based recommender has the worst performance. Popularity-based recommender algorithm is superior to cosine-based CF and content-based recommender in precision, but its coverages are too low (0.035 and 0.069), since popularity-based algorithm always recommends those most popular users to the target campaign.

7. Conclusion and prospects

The sparseness in crowdfunding platform Kickstarter is more than 99% [35]. With such a high sparseness, cosine-based CF obtains poor recommendation performance. Therefore, we use the bipartite graph-based network structure to describe users' behaviors and use PersonalRank to calculate the distance between campaigns and users to directly produce recommendation lists. Next, we integrate bipartite graph model and CF algorithm, and the correlation among the items set (the users set) is obtained by PersonalRank as the measurement of interest similarity. Experimental results show that recommender based on bipartite graph model achieves better performance on a sparse dataset. This paper proposes a method to solve the problem of sparse data, providing a new idea for generating recommendation list in crowdfunding platforms.

Directions for future works are as follows. (1) In terms of bipartite graph model, PersonalRank is not the only algorithm, while other network algorithms are applicable to calculate the node similarity, such as SimRank [36]. Other graph models could be applied to recommendation for crowdfunding campaigns in the future. (2) Due to computing complexity, all of CF algorithms used in this paper are item-based, rather than user-based. However, we have to use user-based recommender in some cases. For example, when a new user enters the system, user-based method is more suitable in recommendation. Future research could make a comparison with user-based recommender algorithms. (3) The datasets are all from Kickstarter, but there are other crowdfunding platforms, such as Indiegogo [37]. Research could use other crowdfunding platforms to verify the applicability of bipartite graph model. (4) Based on the data from the crowdfunding platform, we have verified the usefulness of bipartite graph model. However, not all the information in crowdfunding communities is used. For example, some research found the home bias is a common phenomenon in investment [38], that is, offline relationships between founders and investors may have already been established, such as friends, classmates, acquaintances, colleagues, etc. Consequently, there is a psychological and cultural convergence between founders and investors, and the physical distance is relatively close. Therefore, in personalized recommender, the physical distance in graph model could be considered, and the physical distance between users could be modeled into binary graph model to improve the performance of recommender.

Acknowledgements

This work is partially supported by the NSFC Grant (71771177), the University Innovation Fund from the Science and Technology Development Center of the Ministry of Education (2019J01012), and Standardization of Trade in Service Fund (FMBZH-1947).

Conflict of interest

The authors declare no conflict of interest.

Author details

Hongwei Wang* and Shiqin Chen
School of Economics and Management, Tongji University, Shanghai, China

*Address all correspondence to: hwwang@tongji.edu.cn

References

[1] An J, Quercia D, Crowcroft J. Recommending investors for crowdfunding projects. In: Proceedings of the 23rd International Conference on World Wide Web (WWW '14). New York, US: ACM Press; 2014. pp. 261-270

[2] Chen L, Chen G, Wang F. Recommender systems based on user reviews: The state of the art. User Modeling and User-Adapted Interaction. 2015;**25**(2):99-154

[3] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. In: Proceedings of the Seventh International Conference on World Wide Web (WWW '7). Amsterdam, NLD: Elsevier Science Publishers B. V.; 1998. pp. 107-117

[4] Page L, Brin S, Motwani R, et al. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab; Stanford, CA; 1999

[5] Richardson M, Domingos P. The intelligent surfer: Probabilistic combination of link and content information in PageRank. In: Neural Information Processing Systems 14. Cambridge, MA: MIT Press; 2001. pp. 1441-1448

[6] Haveliwala TH. Topic-sensitive PageRank. In: Proceedings of the 11th International Conference on World Wide Web (WWW '02). New York, US: ACM Press; 2002. pp. 517-526

[7] Kohlschutter C, Chirita P, Nejdl W. Efficient parallel computation of PageRank. In: Proceedings of the 28th European conference on Advances in Information Retrieval (ECIR '06). Berlin, Heidelberg: Springer-Verlag; 2006. pp. 241-252

[8] Arasu A, Novak J, Tomkins A, et al. PageRank computation and the structure of the web: Experiments and algorithms. In: Proceedings of the 11th International Conference on World Wide Web (WWW '02). New York, US: ACM Press; 2002. pp. 107-117

[9] Massucci FA, Docampo D. Measuring the academic reputation through citation networks via PageRank. Journal of Informetrics. 2019;**13**(1):185-201

[10] Kleinberg J. Authoritative sources in a hyperlinked environment. Journal of the ACM. 1999;**46**(5):604-632

[11] Bharat K, Henzinger M. Improved algorithms for topic distillation in a hyperlinked environment. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98). New York, US: ACM Press; 1998. pp. 104-111

[12] Lempel R, Moran S. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. Computer Network. 2000;**33**(1-6):387-401

[13] Tay DBH, Lin Z. Design of near orthogonal graph filter banks. IEEE Signal Processing Letters. 2015;**22**(6): 701-704

[14] Hammack RH, Puffenberger O. A prime factor theorem for bipartite graphs. European Journal of Combinatorics. 2015;**47**:123-140

[15] Kaya B. Hotel recommendation system by bipartite networks and link prediction. Journal of Information Science. 2020;**46**(1):53-63

[16] Niu A, Fan D, Wang G. On the distance Laplacian spectral radius of bipartite graphs. Discrete Applied Mathematics. 2015;**186**:207-213

[17] Gharibshah J, Jalili M. Connectedness of users-items networks

and recommender systems. Applied Mathematics and Computation. 2014; **243**:578-584

[18] Riesen K, Bunke H. Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing. 2009;**27** (7):950-959

[19] Riesen K, Fischer A, Bunke H. Estimating graph edit distance using lower and upper bounds of bipartite approximations. International Journal of Pattern Recognition and Artificial Intelligence. 2015;**29**(02):1550011

[20] Fischer A, Uchida S, Frinken V, et al. Improving hausdorff edit distance using structural node context. In: International Workshop on Graph-Based Representations in Pattern Recognition. Berlin, Heidelberg: Springer-Verlag; 2015. pp. 148-157

[21] Zhou H, Xu P, Qu H. Visualization of bipartite relations between graphs and sets. Journal of Visualization. 2015; **18**(2):159-172

[22] Wang X, Tang Y, Masnou S, et al. A global/local affinity graph for image segmentation. IEEE Transactions on Image Processing. 2015;**24**(4):1399-1411

[23] Lee S, Kahng M, Lee S. Constructing compact and effective graphs for recommender systems via node and edge aggregations. Expert Systems with Applications. 2015;**42**(7):3396-3409

[24] Billsus D, Pazzani MJ. Learning collaborative information filters. In: Proceedings of the 15th International Conference on Machine Learning (ICML ’98). San Francisco, CA: Morgan Kaufmann Publishers; 1998. pp. 46-54

[25] Lin C-C, Tsai C-C. Applying social bookmarking to collective information searching (CIS): An analysis of behavioral pattern and peer interaction for co-exploring quality online resources. Computers in Human Behavior. 2011;**27**(3):1249-1257

[26] Dalcanale F, Fontane DG, Csapo J. A general framework for a collaborative water quality knowledge and information network. Environmental Management. 2011;**47**(3):443-455

[27] Liu H, He J, Wang T, et al. Combining user preferences and user opinions for accurate recommendation. Electronic Commerce Research and Applications. 2013;**12**(1):14-23

[28] Xu J, Zheng X, DingW. Personalized recommendation based on reviews and ratings alleviating the sparsity problem of collaborative filtering. In: Proceedings of the 2012 IEEE Ninth International Conference on e-Business Engineering (ICEBE ’12). Piscataway, N.J: IEEE Press; 2012. pp. 9-16

[29] Ren X, Lu L, Liu R, et al. Avoiding congestion in recommender systems. New Journal of Physics. 2014;**16**(6): 063057

[30] Rakesh V, Choo J, Reddy CK. Project recommendation using heterogeneous traits in crowdfunding. In: Proceedings of the Ninth International AAAI Conference on Web and Social Media. Palo Alto, CA: AAAI Press; 2015. pp. 337-346

[31] Lu C-T, Shuai H-H, Yu PS. Identifying your customers in social networks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM ’14). New York, USA: ACM Press; 2014. pp. 391-400

[32] Stone T, Zhang W, Zhao X. An empirical study of top-n recommendation for venture finance. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge

Management. New York, USA: ACM Press; 2013. pp. 1865-1868

[33] Zhou T, Kuscsik Z, Liu J-G, Medo M, Wakeling JR, Zhang Y-C. Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences of the United States of America. 2010;**107**(10): 4511-4515

[34] Hui JS, Gerber E, Greenberg M. Easy Money? The Demands of Crowdfunding Work. Segal Design Institute, Northwestern University; Gerber, Greenberg; 2012. pp. 1-11

[35] Testa S, Nielsen KR, Bogers M, et al. The role of crowdfunding in moving towards a sustainable society. Technological Forecasting and Social Change. 2019;**141**:66-73

[36] Jeh G,Widom J. SimRank: A measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02). Association for Computing Machinery. New York, US: ACM Press; 2002. pp. 538-543

[37] Simons A, Kaiser LF, Vom Brocke J. Enterprise crowdfunding: Foundations, applications, and research findings. Business and Information Systems Engineering. 2019;**61**(1):113-121

[38] Lin M, Viswanathan S. Home bias in online investments: An empirical study of an online crowdfunding market. Management Science. 2016;**62**(5): 1393-1414