Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 4430087, 9 pages
https://doi.org/10.1155/2022/4430087

Research Article

Efficient Rational Community Detection in Attribute Bipartite Graphs

Chen Yang,[1] Hao Ji ,[2] and Yanping Wu[3]

[1]*China Tower Corporation Limited, Zhejiang Branch, Hangzhou, China*
[2]*Hangzhou Medical College, Hangzhou, China*
[3]*University of Technology Sydney, Sydney, Australia*

Correspondence should be addressed to Hao Ji; jihaobest11@163.com

Received 9 February 2022; Revised 17 March 2022; Accepted 4 April 2022; Published 28 April 2022

Academic Editor: Chunlai Chai

Bipartite graph is widely used to model the complex relationships among two types of entities. Community detection (CD) is a fundamental tool for graph analysis, which aims to find all or top-k densely connected subgraphs. However, the existing studies about the CD problem usually focus on structure cohesiveness, such as (α, β)-core, but ignore the attributes within the relationships, which can be modeled as attribute bipartite graphs. Moreover, the returned results usually suffer from rationality issues. To overcome the limitations, in this paper, we introduce a novel metric, named *rational score*, which takes both preference consistency and community size into consideration to evaluate the community. Based on the proposed rational score and the widely used (α, β)-core model, we propose and investigate the rational (α, β)-core detection in attribute bipartite graphs (RCD-ABG), which aims to retrieve the connected (α, β)-core with the largest rational score. We prove that the problem is NP-hard and the object function is nonmonotonic and non-submodular. To tackle RCD-ABG problem, a basic greedy framework is first proposed. To further improve the quality of returned results, two optimized strategies are further developed. Finally, extensive experiments are conducted on 6 real-world bipartite networks to evaluate the performance of the proposed model and techniques. As shown in experiments, the returned community is significantly better than the result returned by the traditional (α, β)-core model.

1. Introduction

A bipartite graph is composed of two disjoint vertex sets, and there are only edges connecting vertices from different sets. Due to its proliferation applications like fraudsters detection [1] and collaboration group maintenance [2], many fundamental problems have been investigated to analyze the bipartite graphs. Among these problems, community detection (CD) aims to find all or top-k communities by leveraging different models like (α, β)-core [3], bitruss [4], and so on. Due to its unique feature, the (α, β)-core model is widely adopted in different domains. Given a bipartite graph, the (α, β)-core is the maximal subgraph where the degree of each vertex in the upper layer is at least α and the degree of each vertex in the lower layer is at least β. Nonetheless, previous models mainly focus on the cohesiveness structure of the graphs but neglect the attribute properties with community.

In real applications, the relationships between different entities often have certain characteristics, which can be modeled as attribute bipartite graphs. For example, in the user-movie network of Figure 1, the upper layer denotes a set of users and the lower layer are the set of movies. Each edge is associated with a number denoting the score assigned from a user to a movie. For a discussion group in the platform, it will have a more harmonious atmosphere if users have high consistency of preference (e.g., rating the same score or tag for the same movie). Besides, small discussion group is more conducive to frequent communication among users. However, the existing research cannot capture those properties. Motivated by this, in this paper, we introduce a novel metric, named rational score, which takes both

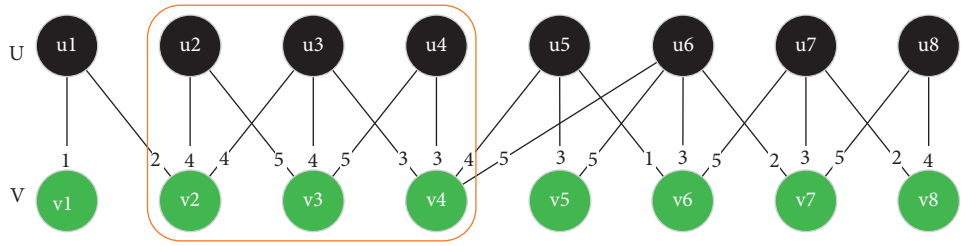

Figure 1: A user-movie network.

preference consistency and community size into consideration to evaluate a community. Furthermore, we formally define the problem of rational community detection over attribute bipartite graphs (RCD-ABG), which attempts to find the connected (α, β)-core with the largest rational score. The following is a motivation example.

Example 1. Reconsider the user-movie network in Figure 1, where the number on the edge denotes the corresponding rating for the movie. Note that the scoring mechanism adopts a five-point system, so the score varies from 1 to 5 in the network. Suppose $\alpha = 2$ and $\beta = 2$ here. Based on the definition, the subgraph induced by vertex set $\{u_2, u_3, \ldots, u_8, v_2, v_3, \ldots, v_8\}$ is a $(2, 2)$-core, where the degree of each vertex is at least 2. However, in the $(2, 2)$-core, many users have distinct scoring schemes for the same movie. For example, users u_6, u_7, and u_8 gave three different scores to the movie v_7. Moreover, the community size is too large to facilitate communication between users. For instance, users u_2 and u_6 even have not watched the same movie ever. Given $\alpha = 2$ and $\beta = 2$, the vertices in the orange rectangle are our identified rational (α, β)-core community. Note that, due to the complex equation involved, the detailed definition of rational (α, β)-core community can be found in preliminaries section. As we can observe, in this community, most users share the same movie taste and the number of people in the group is more reasonable.

1.1. Applications. The RCD-ABG problem can find many real-world applications. We list some examples as follows.

(i) *Discussion Group Mining.* In some real-world bipartite graphs such as BookCrossing, edges denote rating relationships between users and books. There are many discussion groups with these platforms. For users, they are more likely to stay active in a discussion group if the users inside share the same taste. Besides, users will prefer to discuss different topics in a group with appropriate size. This is because too many users can make them uncomfortable and too few will make the discussion difficult to carry on. Hence, by retrieving the rational group, the platform can provide group recommendation more precisely, which is helpful for better user experience.

(ii) *Personalized Product Recommendation.* In customer-movie bipartite networks, the customers will rate the movies based on their personal preference and movie performance. By retrieving the rational (α, β)-core, the personalized movie recommendation can be provided to customers in the rational community. For instance, in the community found in the orange rectangle in Figure 1, the platform can recommend movie v_4 for user u_2. This is because v_4 is given the common score from other customers (i.e., u_3 and u_4). Similarly, movie v_2 can be recommended for user u_4.

1.2. Challenges. To our best knowledge, we are the first to investigate the rational (α, β)-core detection problem in attribute bipartite graphs. We prove the problem is NP-hard and we adopt the greedy framework to remove the best vertex iteratively. However, removing a vertex from the graph may make many other vertices drop from the result, which limits the effectiveness of the algorithm. Hence, it is necessary to develop optimized techniques to address these challenges.

1.3. Our Solution. Due to the NP-hardness of the problem, a basic greedy framework is proposed by adopting the greedy framework. In general, we remove the vertex with the smallest marginal gain at each iteration and calculate the remaining (α, β)-core with its rational score. We stop this process until there is no (α, β)-core and return the (α, β)-core with the largest rational score as the result. To address the discussed drawbacks of our basic greedy framework, we further develop two improved strategies, namely, 2-hop neighbors-based optimization and followers-based optimization. Specifically, in 2-hop neighbors-based optimization, we approximate the marginal score by considering the 2-hop neighbors of the removed vertex in the same layer. In our followers-based optimization, we consider the followers of the removed vertex and modify the marginal rational score.

1.4. Contributions. The contributions of this paper are summarized as follows.

(i) To better capture the properties within bipartite graph community, we conduct the first research to propose and investigate the rational community detection problem over attribute bipartite graphs by leveraging the novel rational score metric developed.

(ii) Theoretically, we prove that the problem is NP-hard, and the rational score function is non-monotonic and non-submodular.

(iii) The basic greedy framework is first presented. To further improve the quality of returned results, two optimized strategies are proposed, namely, 2-hop neighbors-based optimization and followers-based optimization.

(iv) Experiments over 6 real-world bipartite graphs are conducted to show the superiority of proposed techniques. Compared with the traditional (α, β)-core model, our model is much more effective.

1.5. Roadmap. We organize the rest of this paper as follows. We first review the related work. Then, we introduce the problem investigated and the corresponding problem properties. Next, we will present the basic greedy framework and two optimized strategies. Finally, we report the performance of our algorithms over real datasets and conclude the paper.

2. Related Work

In this paper, we conduct the first attempt to propose and investigate the rational (α, β)-core problem. Thus, we will present the related work from the following two aspects.

Cohesive Subgraphs Mining. In different domains, graphs are widely used to model the complex relationships among different entities. As a key problem in graph analysis, community search has been widely studied in the literature and different models have been proposed to measure the cohesiveness of community, such as k-core, k-truss, and clique. In many real-world applications, both graph structures and attribute information are considered. For attribute graph processing, community search problem used both link relationship and attributes because the attributes usually can make communities more meaningful and easy to interpret [5]. In [5], Fang et al. proposed attributed community query (or ACQ) problem, which returned an attributed community (AC) for an attributed graph. The returned community should satisfy both structure cohesiveness constraint and keyword cohesiveness constraint. In [6], Huang and Lakshmanan considered communities based on topics of interest and proposed attributed truss communities (ATC) search problem. They aimed to find connected k-truss subgraphs that contained query vertices with the largest attribute relevance score. In [7], Zhang et al. proposed a keyword-centric community search (KCCS) problem over attribute graphs. They tried to find a community, where the degree of each vertex should be at least k, and the distance between the vertex and all query keywords is minimized. Influential community search has also been studied in [8], where each vertex is associated with a number denoting its influence. Its goal was to find communities with the largest influence.

Bipartite Graph Analysis. Recently, the bipartite graph has attracted much attention due to its proliferate applications like online group recommendation and fraudsters' detection [2]. In [9], Borgatti and Everett were the first to investigate the cohesive communities in bipartite graphs for network analysis. To analyze the properties of bipartite networks, numerous models have been investigated, such as (α, β)-core [10], bitruss [11], and biclique [12]. In [13], the significant (α, β)-community search problem was proposed and studied on weighted bipartite graphs, where each edge is associated with a weight. They aimed to find the significant (α, β)-community that contained query vertex and maximized the minimum edge weight within community. In [4], Wang et al. studied the bitruss model in bipartite graphs. Given a bipartite graph, the bitruss is the maximal subgraph where each edge is contained in at least k butterflies. In the literature, considering the fairness constraints, the fair clustering problems [14–16] were investigated to find communities on bipartite graphs. However, none of the previous studies take the rationality of communities into consideration.

Table 1: Summary of notations.

Notation	Definition
$G = (U, L, E, \mathscr{A})$	An attribute bipartite graph
U/L	The vertex set
E	The edge set
$\mathscr{A} = \{a_1, \ldots, a_t\}$	The attribute set of edges
$S = (U_S, L_S, E_S, \mathscr{A}_S)$	An induced subgraph of G
n	Number of vertices in G
m	Number of edges in G
u, v	Vertex in G
$N_S(u)$	The set of u's neighbors in S
$d_S(u)$	The degree of u in S
α, β	The degree constraint
$x_G(v)$	Consensus score of vertex $v \in L$
x_S	Consensus score of subgraph S
$f(S)$	Rational score of subgraph S

3. Preliminaries

In this section, we first introduce some necessary concepts and present the formal definition of the rational community detection problem over attribute bipartite graphs. Table 1 summarizes the notations that are frequently used in this paper.

3.1. Problem Definition. We consider an attribute bipartite graph $G = (U, L, E, \mathscr{A})$ as an undirected graph without multiple edges and self-loops. U and L are the two disjoint and independent vertex sets in G; that is, $U \cap L = \varnothing$. E is the edge set and each edge $e = (u, v) \in E$ connects one vertex $u \in U$ and one vertex $v \in L$; that is, $E \subseteq U \times L$. $\mathscr{A} = \{a_1, a_2, \ldots, a_t\}$ is the attribute set. Each edge $e \in E$ is

associated with an attribute (e.g., number/tag) $a(e) \in \mathscr{A}$. We use n and m to denote the number of vertices and edges in G, respectively. Given an attribute bipartite graph G, a subgraph $S = (U_S, L_S, E_S, \mathscr{A}_S)$ is an induced subgraph of G; if $U_S \subseteq U, L_S \subseteq L, E_S = E \cap (U_S \times L_S)$ and $\mathscr{A}_S \subseteq \mathscr{A}$. For a vertex $u \in S$, the set of u's neighbors is denoted by $N_S(u)$ (i.e., the adjacent vertices of u). $d_S(u) = |N_S(u)|$ denotes the degree of u in S (i.e., the number of u's neighbor vertices).

Definition 1 ((α, β)-core). Given a bipartite graph G, a subgraph S is the (α, β)-core of G, denoted by $C_{\alpha,\beta}$, if it satisfies the following: (1) degree constraint (i.e., $d_S(u) \geq \alpha$ for each vertex $u \in U_S$ and $d_S(v) \geq \beta$ for each vertex $v \in L_S$); (2) S is maximal; that is, any supergraph S'^S is not a (α, β)-core.

To compute the (α, β)-core, in our paper, we iteratively remove the vertices in two layers violating the corresponding degree constraint until there are no unsatisfied vertices in the graph, the details of which are shown in Algorithm 1. The time complexity is $O(m)$ [17]. As discussed before, the people in a rational discussion group are cohesive and have consistent preference. In the following, we first introduce the consensus score of vertex and community, respectively. Note that we only consider the consensus score of the vertex in lower (e.g., movie) layer. The rational (α, β)-core model is further developed based on the rational score consisting of the consensus score and community size. Then, we present the formal definition of our problem.

Definition 2 (Consensus score). Given an attribute bipartite graph G, the consensus score of each vertex $v \in L$, denoted by $x_G(v)/d_G(v)$, where $x_G(v)$ is the maximum number of its adjacent edges in G with the same attribute number. For a subgraph S of G, its consensus score is defined as $x_S = \sum_{v \in L_S} x_S(v)/d_S(v)/|L_S|$, where $\sum_{v \in L_S} x_S(v)/d_S(v)$ is the sum of consensus score of all vertices in L_S and $|L_S|$ is the number of vertices in the lower layer of S.

Example 2. Considering the vertices in the orange line of the bipartite graph in Figure 1, the consensus score of v_3 is 2/3. The consensus score of community in the orange line is 8/9.

To judge a community, we not only want to consider the consensus but also want to consider the size constraint of it. This is because that the traditional study group with not very large size can facilitate people there to discuss and analyze problem. So, we also combine the size constraint into our rational score function, which is expressed as follows:

$$f(S) = \lambda \frac{\sum_{v \in L_S} x_S(v)/d_S(v)}{|L_S|} + (1 - \lambda) \frac{1}{|U_S||L_S|}, \quad (1)$$

where λ is a parameter to make the trade-off between the consensus score and the community size. Based on this rational score function, we give the definition of rational community.

Definition 3 (rational (α, β)-core). Given an attribute bipartite graph G and two positive integers α and β, a subgraph S is a attribute (α, β)-core of G, denoted by $RC_{\alpha,\beta}$, if it meets the following three criteria:

(i) Connectivity: S is connected

(ii) Cohesiveness: S is a (α, β)-core

(iii) Rationality: S has the largest rational score $f(S)$ among subgraphs satisfying the above criteria

3.1.1. Problem Statement. Given an attribute bipartite graph G and two positive integers α and β, we aim to develop efficient algorithms to find the rational (α, β)-core (i.e., the (α, β)-core with the largest rational score).

3.2. Problem Properties. As shown in Theorem 1, the problem studied is NP-hard. Besides, the rational score function is nonmonotonic and non-submodular, whose details are in Theorem 2.

Theorem 1. *Given an attribute bipartite graph G, the problem of computing the rational (α, β)-core is NP-hard.*

Proof. When $\alpha > 0$ and $\beta > 0$, we reduce the biclique problem [17] to RCD-ABG problem. Given an attribute bipartite graph $G = (V = (U \cup L), E, \mathscr{A})$, where for each vertex in lower layer L, its adjacent edges have distinct attribute. This means that given a subgraph S of G, the consensus score of each vertex v in L_S is $1/d_S(v)$. Hence, our score function is converted to $f(S) = \lambda \sum_{v \in L_S} 1/d_S(v)/|L_S| + (1 - \lambda)1/|U_S||L_S|$. In order to make the rational score large, for the first term of function, namely, $\lambda \sum_{v \in L_S} 1/d_S(v)/|L_S|$, we need to make the numerator be largest and the denominator be smallest. Due to the degree constraint of lower layer, the lower bound of $d_S(v)$ is β. So, the rational score function is $f = \lambda|L_S|1/\beta/|L_S| + (1 - \lambda)1/|U_S||L_S| = \lambda 1/\beta + (1 - \lambda)1/|U_S||L_S|$. Given the parameter α, β, and λ, to find rational (α, β)-core with largest f, $|U_S|$ and $|L_S|$ need to be minimized, which means that $|U_S|$ and $|L_S|$ should be equal to β and α, respectively. As discussed, each vertex $u \in U_S$ (resp. $u \in L_S$) should satisfy $d_S(u) \geq \alpha$ (resp. $d_S(u) \geq \beta$). This is a biclique that each vertex in different layers is connect, which is NP-hard [17]. Therefore, our problem is NP-hard. □ □

Theorem 2. *The objective score function $f(S)$ is nonmonotonic and non-submodular.*

Proof. Nonmonotonic. By considering the example in Figure 1, we first prove its nonmonotonicity. Note that we only keep two decimal places in the following. Suppose $\lambda = 0.5$; we can see that in subgraph denoted by solid line, that is, $S = \{u_2, u_3, u_4, v_2, v_3, v_4\}$, $f(S) = 0.5$. After deleting vertex u_2, $f(S/\{u_2\}) = 0.53$. While, by further deleting vertex u_4, the present score is $f(S \backslash \{u_2\} \backslash \{v_2\}) = 0.5$. Therefore, the function is nonmonotonic.

Non-Submodular. Given two sets A and B, $f(x)$ is submodular if $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$. We show the inequality does not hold by a counterexample in Figure 1. Suppose $A = \{(u_2, u_3, v_2, v_3)\}$ and $B = \{(u_3, u_4, v_3, v_4)\}$.

We have $f(A) = 0.5$, $f(B) = 0.5$, $f(A \cup B) = 0.5$, and $f(A \cap B) = 0.75$. Thus, the equation does not hold and f is not submodular. □

4. Solution

In this section, a greedy framework is firstly developed to find the result, which is based on the concept of score function and marginal gain that we define. Considering the limitations of the basic method, we further propose two novel strategies with better quality.

4.1. A Basic Greedy Framework (BGF). Intuitively, to find the (α, β)-core with largest score, we can delete those vertices whose deletion will increase the score. Based on this, we present our basic greedy framework by introducing the rational marginal gain as follows.

Definition 4. (rational marginal score). Given an attribute bipartite graph G and a vertex $u \in G$, the rational marginal gain is defined as

$$\triangle_G(u) = \begin{cases} f(G) - f\left(\dfrac{G}{(N_G'(u) \cup \{u\})}\right) u \in U. \\ \\ f(G) - f\left(\dfrac{G}{\{u\}}\right) u \in L, \end{cases} \tag{2}$$

where $N_G'(u)$ is the set of u's neighbors in L that violate the degree constraint after removing vertex u.

4.1.1. The Basic Greedy Framework (BGF). The details of BGF are illustrated in Algorithm 2, which includes three main steps. We use $\mathscr{G}$ to denote the set of all connected (α, β)-cores. *Step 1.* We find all (α, β)-cores of G and store them into $\mathscr{G}$ in Line 2. We use $\mathscr{G}_i$ to denote the current processing (α, β)-core. *Step 2.* At each iteration, we greedily peel the vertex v in graph $\mathscr{G}_i$ providing the smallest marginal gain $\triangle_{\mathscr{G}_i}(v) = f(\mathscr{G}_i) - f(\mathscr{G}_i/\{v\})$ (Line 5), which is called the best vertex. After removing best vertex, we calculate the (α, β)-core in the remaining graph. If there are many connected (α, β)-cores, we push back them into $\mathscr{G}$ (Lines 6–8). We continue this process until there is no (α, β)-core in the graph. Note that the rational marginal gain may be a negative number, which means the score of function increase. *Step 3.* We output the result with the largest score among obtained attribute (α, β)-cores (Line 9).

Example 3. Considering the user-movie network in Figure 1. Suppose $\alpha = 2, \beta = 2$. According the BGF, vertex v_6 is removed firstly and the rational score of the remained $(2, 2)$-core is 0.415476. Similarly, we remove vertices v_7, v_4, and v_3, iteratively. The corresponding rational score is 0.481667, 0.6, and 0. Therefore, the returned result is $\{u_2, u_3, v_2, v_3\}$ with rational score of 0.6.

4.2. Optimized Strategies. The basic greedy framework is simple but suffers from the following drawback. When removing a vertex v from the subgraph S, it may make the support of some other vertices decrease and lead them to drop from the community in succession. Note that these vertices are called the followers of v including v itself, denoted as $\mathscr{F}_S(v)$. If the removal vertex has a large number of followers, it can severely limit the effectiveness of the algorithm. Hence, we need to consider the effect of each removal vertex. In the following section, we propose two improved strategies to handle the limitation.

4.2.1. 2-Hop Neighbors Optimization (OS-I). As observed, if the removal vertex is in the lower layer, its 2-hop neighbors in the same layer may violate the degree constraint and be deleted, which significantly affect the rational score. Based on this, we use the following equation to approximate marginal score function $\triangle_G(u)$ by $\widehat{\triangle}_G(u)$,

$$\widehat{\triangle}_G(u) = f(G) - f\left(\frac{G}{(N_G'(u) \cup \{u\})}\right) u \in U, f(G) - f\left(\frac{G}{(H2_G(u) \cup \{u\})}\right), \tag{3}$$

where $H2_G(v)$ is the 2-hop neighbors of v in the lower layer. Therefore, the best vertex is adjusted as $u \longleftarrow \text{argmin}_{v \in \mathscr{G}_i} \widehat{\triangle}_{\mathscr{G}_i}(v)$ in Line 5 of algorithm 1 and other steps are the same.

Example 4. Reconsider the user-movie network in Figure 1. Suppose $\alpha = 2$, $\beta = 2$. According to the OS-I, we remove vertex v_7 firstly and obtain the rational score of the remained $(2, 2)$-core. Then, we remove u_6 and obtain the corresponding score of 0.6556. After removing u_3, the obtained score is 0. So, we return the result by $\{u_2, u_3, u_4, v_2, v_3, v_4\}$ with a score of 0.6556.

4.2.2. Followers-Based Optimization (OS-II). The second idea is motivated by the followers of each removal vertex. Generally, instead of removing one vertex and calculating the rational marginal gain, we remove a vertex with its all followers from the current candidate graph that have the smallest attribute marginal gain. Hence, the marginal score is modified as the following equation:

$$\widetilde{\triangle}_G(u) = f(G) - f\left(\frac{G}{\mathscr{F}_G(u)}\right), \tag{4}$$

and the other steps are the same as BGF.

Input: G: a bipartite graph, α, β: degree constraints
Output: The (α, β)-core of G
(1) **While** exists $u \in U$ with $d(u) < \alpha$ or $u \in V$ with $d(u) < \beta$ **do**
(2) $G \longleftarrow G/\{u\}$
(3) **return** G

Algorithm 1: Compute (α, β)-core.

Input: G: attribute bipartite graph, α: degree constraint in upper layer, β: degree constraint in lower layer
Output: H: the connected (α, β)-core with the largest rational score
(1) $i \longleftarrow 1$
(2) $\mathcal{G} \longleftarrow$ an empty vector
//Step 1
(3) $\mathcal{G} \longleftarrow$ all connected $C_{\alpha,\beta}(G)$
//Step 2
(4) **While:** $\mathcal{G} \neq \emptyset$ **do**
(5) $u \longleftarrow \text{argmin}_{v \in \mathcal{G}_i} \triangle_{\mathcal{G}_i}(v)$
(6) **for each** connected $C_{\alpha,\beta}(\mathcal{G}_i/u)$ in $\mathcal{G}_i \backslash u$ **do**
(7) push back $C_{\alpha,\beta}(\mathcal{G}_i/u)$ into $\mathcal{G}$
(8) $i \longleftarrow i + 1$;
//Step 3
(9) $H \longleftarrow \text{arg}G'_{G_i \in \mathcal{G}} f(G')$;

Algorithm 2: A basic greedy framework (BGF).

Example 5. Reconsider the example in Figure 1. Suppose $\alpha = 2$, $\beta = 2$. According to the OS-II, vertex u_7 is removed firstly and the obtained score is 0.455333. Then, we remove v_5 and calculate the score with 0.65556. After deleting u_2, the score is 0. Therefore, we return the result $\{u_2, u_3, u_4, v_2, v_3, v_4\}$ with a score of 0.65556.

4.2.3. Analysis. The main difference between BGF and optimized algorithm is the best vertex. In BGF, calculating the marginal score of a vertex is $O(1)$ time. In OS-II, the time complexity of identifying the followers of the vertex is $O(m)$, which may significantly increase the running time.

5. Experiments

5.1. Algorithms. To the best of our knowledge, there is no existing work for RCD-ABG problem. In the experiments, we implement and evaluate the following algorithms.

(i) *BGF*. The baseline greedy framework is presented in Algorithm 2, which iteratively peels the graph and returns the best result during the search

(ii) *OS-I*. OS-I leverages the baseline framework BGF and further integrates the proposed 2-hop neighbor-based optimization

(iii) *OS-II*. OS-II leverages the baseline framework BGF and further integrates the proposed follower-based optimization

(iv) *ORI*. To evaluate the advantage of proposed model, we also implement the traditional (α, β)-core search method [10], which iteratively removes the vertex that violates the degree constraints and returns the final subgraph

Table 2: Statistics of datasets.

Dataset	$\lvert U\rvert$	$\lvert L\rvert$	$\lvert E\rvert$	$\lvert\mathcal{A}\rvert$
HetRec (HR)	2,101	18,746	92,835	5
CiaoDVD (CD)	17,615	16,121	72,345	5
TripAdvisor (TA)	145,316	1,759	175,655	5
MovieLens (ML)	71,535	65,134	855,598	5
BookCrossing (BC)	278,855	270,981	941,148	10
Personality (PY)	1,822	198,118	1,028,751	5

5.2. Datasets and Workloads. We employ 6 real-world bipartite graphs. Among these datasets, CiaoDVD and TripAdvisor can be obtained on KONECT (https://konect.uni-koblenz.de). Other datasets are publicly available on GroupLens (https://grouplens.org/datasets/). The statistics of datasets are shown in Table 2, where $|\mathcal{A}|$ is the number of attributes in bipartite graphs. HetRec (HR) [18] is a user-artists network, where the attribute of edges denotes the number of time that user listens to the music by the artist. CiaoDVD (CD) and MovieLens (ML) [18] are user-movie networks of which the attributes of relationships represent the ratings for movie. TripAdvisor is a user-hotel bipartite graphs and the attribute of its edges denotes the rating taken by users. The BookCrossing (BC) is a user-book network and the edges of it denote the book-rating taken by user. Due to the density of graphs, $\alpha = \beta$ vary from 5 to 25 in HetRec, CiaoDVD, and TripAdvisor, vary from 15 to 35 in BookCrossing and vary from 50 to 250 in MovieLens and Personality. λ is set as 0.7 because the density of community will

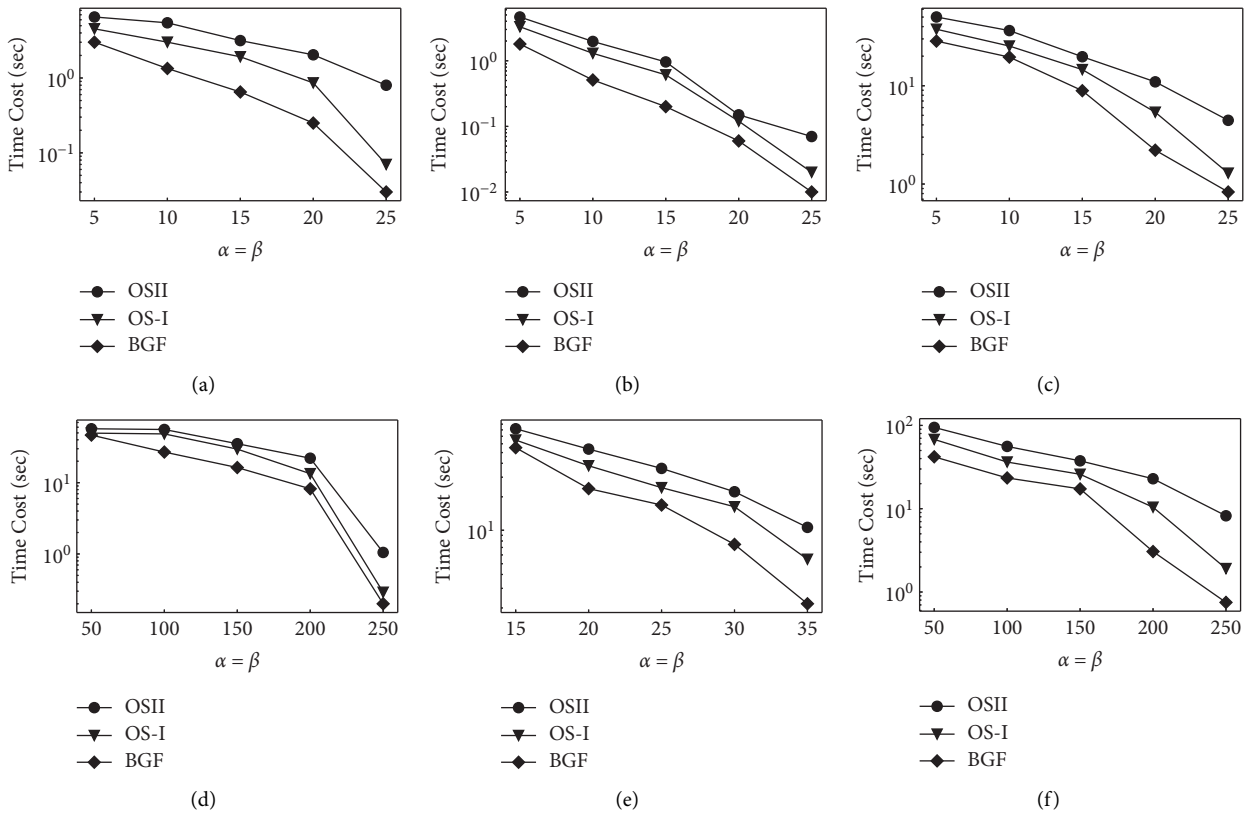

Figure 2: Efficiency evaluation by varying parameters α and β. (a) HetRec. (b) CiaoDVD. (c) TripAdvisor. (d) MovieLens. (e) BookCrossing. (f) Personality.

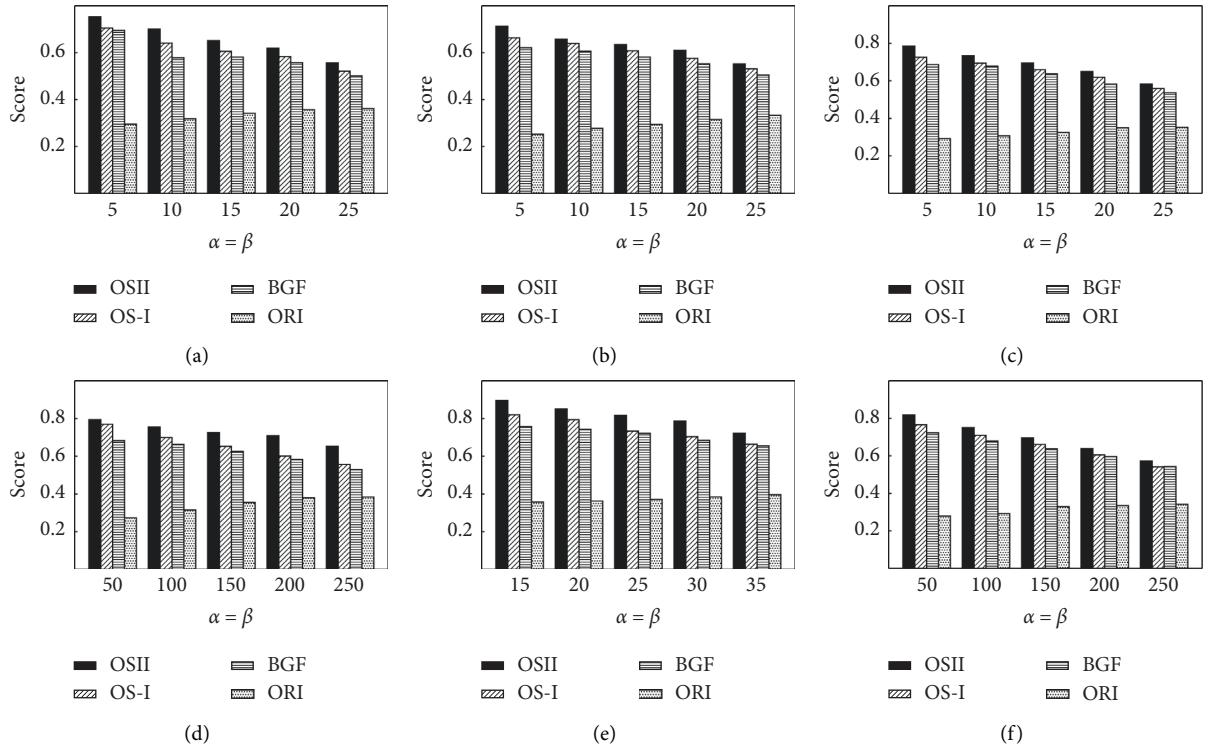

Figure 3: Effectiveness evaluation by varying parameters α and β. (a) HetRec. (b) CiaoDVD. (c) TripAdvisor. (d) MovieLens. (e) BookCrossing. (f) Personality.

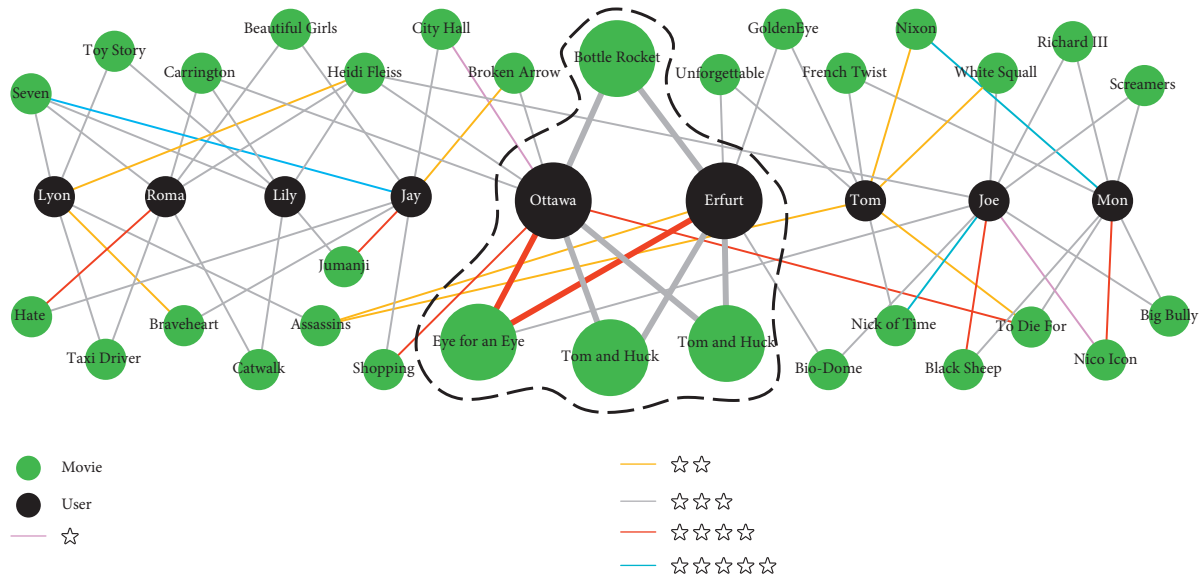

Figure 4: Case study.

strengthen with the continuous deletion of vertices; thus, we focus on consensus score. All the programs are implemented in standard C++. All the experiments are performed on a server with an Intel Xeon 2.4 GHz CPU and 128 GB main memory.

5.3. Efficiency Evaluation. To evaluate the efficiency, we report the response time of algorithms by varying α and β in Figures 2(a)–2(f). As observed, the time cost of OS-I and OS-II is more than BGF. This is because OS-I needs to calculate 2-hop neighbors of vertex in the lower layer and OS-II needs to calculate followers of vertex. Although the time complexity of OS-I and OS-II is more complex than BGF, there is not much difference of response time between them. We can observe that when α and β increase, the response time decreases for all methods. This is because the community size decreases.

5.4. Effectiveness Evaluation. To evaluate the effectiveness, we compare BGF, OS-I, and OS-II with ORI and report the rational score of the returned community. ORI is based on the traditional (α, β)-core model. It first computes the (α, β)-core of the graph and then directly returns the connected component with the largest rational score. The results are shown in Figures 3(a)–3(f). We can observe that original (α, β)-core has very small rational score. OS-I and OS-II significantly outperform BGF over all the datasets, namely, find community with higher score than BGF. The score returned by OS-I is at least 0.01 higher than the one returned by BGF in all datasets. Due to the feature of the consensus score, the improvement of OS-I is already significant for the overall performance. The rational score decreases when α and β increase because of tighter degree constraint.

5.5. Case Study. To further evaluate the advantage of the proposed model, we conduct a case study on HetRec dataset. The results are shown in Figure 4. As shown, the movie and user are marked with different colors. The different-color edges denote different scores. The community in the solid line that consists of enlarged vertices and bold edges is the returned result. As we can see, it can find a more rational community with a high preference and density structure.

6. Conclusion and Future Work

In this paper, we propose and investigate the rational (α, β)-core detection problem in attribute bipartite graphs. We formally define the problem and prove its NP-hardness. To solve this problem, a basic greedy framework is first presented, which iteratively removes the best vertex with the smallest marginal gain and calculate the remaining (α, β)-core. Two optimized strategies, namely, 2-hop neighbor-based optimization and follower-based optimization, are proposed to improve the performance. Experiments are conducted on real bipartite graphs to demonstrate the advantages of proposed model and techniques. As shown in the experience, the proposed model significantly outperforms the traditional (α, β)-core model. In real-world applications, there are also attributes within the vertices of the graphs. In the further work, we will consider more complex scenario to design the model and the corresponding approaches.

Data Availability

The datasets in this paper are publicly available at https://konect.uni-koblenz.de and https://grouplens.org/datasets/.

Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents conflicts of interest in connection with the work submitted.

Acknowledgments

This work was supported by ZJSSF 21NDQN247YB, ZJNSF Y202045024, ZJNSF LQ20F020007, and ZJNSF LY21F020012.

References

[1] B. Liu, L. Yuan, X. Lin, L. Qin, W. Zhang, and J. Zhou, "Efficient (a,β) core computation: an index-based approach," in *Proceedings of The World Wide Web Conference*, pp. 1130–1141, San Francisco, CA, USA, May 2019.

[2] E. Ntoutsi, K. Stefanidis, K. Rausch, and H. P. Kriegel, "Strength lies in differences": diversifying friends for recommendations through subspace clustering," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014*, pp. 729–738, ACM, Shanghai, China, November 2014.

[3] H. Yang, M. Zhang, X. Wang, and C. Chen, "Cohesive subgraph detection in large bipartite networks," in *Proceedings of the International Conference on Scientific and Statistical Database Management*, pp. 1–4, Bolzano, Italy, July2020.

[4] K. Wang, X. Lin, Q. Lu, W. Zhang, and Y. Zhang, "Efficient bitruss decomposition for large-scale bipartite graphs," in *Proceedings of the 36th IEEE International Conference on Data Engineering, ICDE 2020*, pp. 661–672, Dallas, TX, USA, April 2020.

[5] Y. Fang, R. Cheng, Y. Chen, S. Luo, and J. Hu, "Effective and efficient attributed community search," *The VLDB Journal*, vol. 26, no. 6, pp. 803–828, 2017.

[6] X. Huang and L. V. S. Lakshmanan, "Attribute-driven community search," *Proceedings of the VLDB Endowment*, vol. 10, no. 9, p. 949, 2017.

[7] Z. Zhang, X. Huang, J. Xu, B. Choi, and Z. Shang, "Keyword-centric community search," in *Proceedings of the 35th IEEE International Conference on Data Engineering*, pp. 422–433, Macao, China, April 2019.

[8] R. Li, L. Qin, J. X. Yu, and R. Mao, "Influential community search in large networks," *Proceedings of the VLDB Endowment*, vol. 8, no. 5, pp. 509–520, 2015.

[9] S. P. Borgatti and M. G. Everett, "Network analysis of 2-mode data," *Social Networks*, vol. 19, no. 3, pp. 243–269, 1997.

[10] D. Ding, H. Li, Z. Huang, and N. Mamoulis, "Efficient fault-tolerant group recommendation using alpha-beta-core," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, November 2017.

[11] Z. Zou, "Bitruss decomposition of bipartite graphs," in *Proceedings of the Database Systems for Advanced Applications - 21st International Conference*, pp. 218–233, DASFAA, Dallas, TX, USA, April 2016.

[12] Y. Zhang, C. A. Phillips, G. L. Rogers, E. J Baker, E. J Chesler, and M. A Langston, "On finding bicliques in bipartite graphs: a novel algorithm and its application to the integration of diverse biological data types," *BMC Bioinformatics*, vol. 15, no. 110, p. 110, 2014.

[13] K. Wang, W. Zhang, X. Lin, Y. Zhang, L. Qin, and Y. Zhang, "Efficient and effective community search on large-scale bipartite graphs," in *Proceedings of the 37th IEEE International Conference on Data Engineering*, pp. 85–96, ICDE, Chania, Greece, April 2021.

[14] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 5029–5037, Long Beach, CA, USA, December 2017.

[15] S. Ahmadian, A. Epasto, R. Kumar, and M. Mahdian, "Fair correlation clustering," in *Proceedings of the The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, pp. 4195–4205, PMLR, sicily, Italy, August 2020.

[16] H. Larochelle, M. A. Ranzato, R. Hadsell, M. F. Balcan, and H. T. Lin, "Fair hierarchical clustering," in *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, virtual, La Jolla CA, USA, December 2020.

[17] B. Lyu, L. Qin, X. Lin, Y. Zhang, Z. Qian, and J. Zhou, "Maximum biclique search at billion scale," *Proceedings of the VLDB Endowment*, vol. 13, no. 9, pp. 1359–1372, 2020.

[18] I. Cantador, B. Peter, and T. Kuflik, "2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011)," in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec '11*, ACM, Chicago, IL, USA, October 2011.