Bipartite Graphs and Recommendation Systems

Cristina Maier and Dan Simovici
University of Massachusetts Boston, Boston, USA
Email: {cmaier, dsim}@cs.umb.edu

Abstract—**Bipartite graphs are used to model many real-world relationships with applications in several domains, such as: medicine, social networks and marketing. Examples of such relationships include drugs-adverse reactions associations, links between genes and various pathologies, actors and the movies they play in, researchers and the papers they author. We explore several properties of bipartite graphs and propose several notions including the measure of biclique similarity of a set of vertices, the measure of biclique connectivity of a set of vertices, and the notion of chains in bipartite graphs. We introduce the Biclique Similarity Ordering Recommendation (BISOR) algorithm, an application of maximal bicliques of bipartite graphs to recommendation systems that makes use of the notion of biclique similarity of a set of vertices in order to recommend items to users in a certain order of preference. We justify our approach by presenting experimental results that use real-world datasets: Sushi, MovieLens 100k and MovieLens 1Million.**

Index Terms—**bipartite graphs, biclique similarity, polarity, recommendation order**

I. Introduction

Bipartite graphs are graphs whose vertices are partitioned in two disjoint sets, and any existing edge connects a vertex from one set to a vertex from the other set. Real-world relationships from many domains, such as medicine [1] and social networks, could be modeled as bipartite graphs. Examples include drugs-adverse reactions associations, relationships between genes and various pathologies, actors and the movies they play in, researchers and the papers they author, persons and the movies they like.

Recommendation systems focus the attention of users on items in which they might be interested. For a streaming company such as *Netflix* or *Amazon Prime*, it might be important to recommend movies that users might be interested in watching.

The recommendation systems literature is vast. There are different classifications of recommendation systems. One of the most common classification splits the recommendation systems into *content-based filtering*, *collaborative filtering*, and mixed-approaches. Content-based filtering [2], [3] systems recommend items similar to items known to be liked by the user. This method usually requires the creation of profiles for items and/or users. One issue with content-based-filtering is that it will only match other items which share similar properties. This yields no variety in the recommended items, as using this method the user will not be recommended items with different properties. Collaborative-filtering techniques try to solve this issue by looking at the historical interactions between users and items. The term collaborative-filtering was first introduced in [4], where the authors described Tapestry, an experimental mail system which provided support to filter mails using both content-based filtering and collaborative-filtering. *GroupLens* is presented in [5] as an architecture that uses collaborative-filtering to recommend news articles. One collaborative-filtering approach relies on the computations of *nearest-neighborhoods*. The *user-based collaborative-filtering* looks at the nearest neighborhood of the user (i.e. users that share similar likes/dislikes) and recommends items based on the ratings of other users from same nearest neighborhood. This works on the assumption that similar users have similar likes/dislikes. *Item-based collaborative-filtering* approach looks at the nearest neighborhoods of items (i.e. items that share similar likes and dislikes). To predict the rating of a user for a given item, the items that were already rated by the given user are used. One drawback with collaborative-filtering recommender systems is the cold-start problem which appears when new items (or new users) are added to the system, and we do not have information with regards to previous interactions between them and users (items). Data-sparsity can also cause issues with collaborative filtering. Also, when a user has very unique preferences, that are not usually shared across other users, collaborative-filtering might perform poorly. Another collaborative-filtering approach is the *latent factor models*. One of the latent factor models with best performance is the *matrix factorization model* [6], which tries to determine a vector of latent factors between users and items. A survey on auto-encoders recommender systems is presented in [7].

In [8], the authors proposed an algorithm for recommendation that relies on predicting links in a bipartite graph. For any pair of user and item, a kernel function that took into consideration random walks from that user and that item to close neighbors was used. The results from these kernels were fed into a support vector machine algorithm in order to classify the new user-item link as 'possible' or 'impossible'. In [9], authors introduce a recommendation algorithm that uses the notion of one-mode projections.

In [10], the author examines how several graph algebraic analysis methods apply to bipartite graphs.

Manuscript received October 30, 2021; revised April 19, 2022.

doi: 10.12720/jait.13.3.249-258

The recommendation algorithm we propose belongs to the class of collaborative-filtering methods and it uses the structure of the bipartite graph that models users-items likes.

We found that the notion of polarity generated by a relation is useful in exploring bicliques in bipartite graphs. We introduce the measures of biclique connectivity, and biclique similarity of a set of vertices. As an application, we propose a recommendation algorithm that uses the measure of biclique similarity.

In Section II, we discuss bipartite graphs and polarities generated by binary relations. We explore some properties and introduce a special product operation between binary matrices. The notion of maximal biclique generated by a set of vertices is introduced in Section III. In Section IV**,** we propose the measure of biclique similarity of a set of vertices, the measure of biclique connectivity of a set of vertices, and the notion of chains in bipartite graphs. Using some of these notions, we introduce the Biclique Similarity Ordering Recommendation (BISOR) algorithm. In Section V, we present experimental results run on three real-world datasets. Finally, Section VI presents our conclusion and some directions for future work.

The current paper is an extended version of a paper [11] we presented at the ICISDM 2021 conference.

II. Bipartite Graphs and Polarities

The mathematical underpinning of this section is the notion of polarity [12]. The set of subsets of a set U is denoted by *P(U)*.

Let ρ be a binary relation defined on the sets L and R, $\rho \subseteq L \times R$. The *polarity* determined by a relation ρ is a pair of mappings ϕ_ρ: P(L) → P(R) and ψ_ρ: P(R) → P(L) defined as:

$$\phi_\rho(X) = \{y \in R \mid (x, y) \in \rho \text{ for every } x \in X\}$$

$$\psi_\rho(Y) = \{x \in L \mid (x, y) \in \rho \text{ for every } y \in Y\}$$

For any subset $X \subseteq L$ and $Y \subseteq R$.

We say that $\phi_\rho(X)$ is a polar of X and $\psi_\rho(Y)$ is a polar of Y.

The following statements are direct consequences of the definitions of ϕ_ρ and ψ_ρ. Namely, for any binary relation $\rho \subseteq L \times R$ and for every $X \subseteq L$ and $Y \subseteq R$ we have:

$$X \subseteq X_1 \text{ implies } \phi_\rho(X) \supseteq \phi_\rho(X_1) \quad (1)$$
(anti-monotonicity of ϕ_ρ)

$$Y \subseteq Y_1 \text{ implies } \psi_\rho(y) \supseteq \psi_\rho(Y_1) \quad (2)$$
(anti-monotonicity of ψ_ρ)

$$X \subseteq \psi_\rho(\phi_\rho(X)) \quad (3)$$

$$Y \subseteq \phi_\rho(\psi_\rho(Y)) \quad (4)$$

Theorem 2.1. If (ϕ_ρ, ψ_ρ) is a polarity on the sets L and R, then:

$$\phi_\rho(\psi_\rho(\phi_\psi(X))) = \phi(X)$$

and

$$\psi_\rho(\phi_\rho(\psi_\rho(Y))) = \psi_\rho(Y)$$

Proof: As we noted, we have $X \subseteq \psi_\rho(\phi_\rho(X))$, hence $\phi_\rho(X) \supseteq \phi_\rho(\psi_\rho(\phi_\rho(X)))$. The reverse inclusion follows by substituting $\phi_\rho(X)$ for Y in Equality (4), which implies the first equality of the theorem. The proof of the second equality is similar.

If (ϕ_ρ, ψ_ρ) is a polarity on the sets L and R, then for any subsets $X_1, X_2, X \subseteq L$ and $Y_1, Y_2, Y \subseteq R$, we have:

$$\phi_\rho(X_1 \cup X_2) = \phi_\rho(X_1) \cap \phi_\rho(X_2)$$

$$\psi_\rho(Y_1 \cup Y_2) = \psi_\rho(Y_1) \cap \psi_\rho(Y_2)$$

These equalities imply immediately

$$\phi_\rho(X) = \bigcap_{x \in X} \phi_\rho(\{x\})$$

$$\psi_\rho(Y) = \bigcap_{y \in Y} \psi_\rho(\{y\})$$

A *bipartite graph* is a triplet G = $(L, R; \rho)$, where L, R are two disjoint finite sets of vertices and $\rho \subseteq \{\{u, v\} \mid u \in L, v \in R\}$ is the set of edges of G. If $\rho = L \times R$, we say that G is a complete bipartite graph.

Let $\boldsymbol{x}, \boldsymbol{y}$ be two row vectors in $\{0,1\}^k$, where $\boldsymbol{x} = (x_1, \dots, x_k)$ and $\boldsymbol{y} = (y_1, \dots, y_k)$. Define $\mathbf{x} \wedge \mathbf{y}$ as:

$$\boldsymbol{x} \wedge \boldsymbol{y} = (\min\{x_1, y_1\}, \dots, min\{x_k, y_k\})$$

The same operation is defined on column vectors and produces a column vector. The transpose of a vector $\mathbf{x}$ is denoted by $\boldsymbol{x}'$.

For a subset S of a set $U = \{u_1, \dots, u_k\}$ with $|U| = k$ we define its characteristic vector $\boldsymbol{v}_S = (v_1, \dots, v_k)$ as the row vector given by:

$$v_i = \begin{cases} 1 \text{ if } u_i \in S, \\ 0 \text{ otherwise.} \end{cases}$$

Note that if $S, T \subseteq U$ such that $\boldsymbol{v}_S = (v_1, \dots, v_k)$ and $\boldsymbol{v}_T = (w_1, \dots, w_k)$ we have:

$$v_{S \cap T} = (\min\{v_1, w_1\}, \dots, \min\{v_k, w_k\}) = \boldsymbol{v}_S \wedge \boldsymbol{v}_T$$

Let $G = (L, R; \rho)$ be a bipartite graph, where $L = \{u_1, \dots, u_m\}$ and $R = \{v_1, \dots, v_n\}$. The *biadjacency matrix* [13] of G is the matrix $B \in \{0,1\}^{m \times n}$, where $b_{i,j} = 1$ if and only if $(u_i, v_j) \in \rho$. The binary relation $\rho \subseteq L \times R$ defined by the graph G consists of the pairs defined by the edges of G. The component $b_{i,j}$ of the matrix B may be denoted as $b_{u_i v_j}$ when we need to mention explicitly the members of the sets L and R.

The rows of the biadjacency matrix of G are the characteristic vectors of the sets of the form $\phi(\{u_i\})$ while the columns are the characteristic vectors of the sets of the form $\psi(\{v_j\})$.

For a bipartite graph $G = (L, R; \rho)$ and a subset X of L we have $\boldsymbol{v}_{\phi(X)} = \wedge_{x \in X} \boldsymbol{v}_{\phi(\{x\})}$. Therefore, $\boldsymbol{v}_{\phi(X)}$ can be obtained by taking the componentwise minimum of the columns of the form $\boldsymbol{v}_{\phi(\{x_i\})}$ for $x_i \in X$.

Similarly, for a subset Y of R we have $\boldsymbol{v}_{\psi(Y)} = \wedge_{y \in Y} \boldsymbol{v}_{\psi(\{y\})}$ and the row $\boldsymbol{v}_{\psi(Y)}$ equals the componentwise minimum of the rows of the form $\boldsymbol{v}_{\psi(\{y_i\})}$ for $y_i \in Y$.

Example 2.2. For a bipartite graph:

$$G = (\{x_1, x_2, x_3, x_4\}, \{y_1, y_2, y_3, y_4, y_5\}, \rho)$$

Shown in Fig. 1, the biadjacency matrix is:

$$B_\rho = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Thus, a vector $\boldsymbol{v}_{\phi(\{x_i\})}$ is the i^{th} line of the matrix B_ρ:

$$\boldsymbol{v}_{\phi(\{x_1\})} = (1, 0,1, 1,0), \quad \boldsymbol{v}_{\phi(\{x_2\})} = (0, 1, 0, 0,1)$$

$$\boldsymbol{v}_{\phi(\{x_3\})} = (1, 1, 0, 1, 0), \quad \boldsymbol{v}_{\phi(\{x_4\})} = (1, 0, 1, 1,1)$$

and a vector $\boldsymbol{v}_{\psi(\{y_i\})}$ is the transpose of the j^{th} column of B.

$$\boldsymbol{v}_{\phi(\{y_1\})} = (1, 0,1, 1), \boldsymbol{v}_{\phi(\{y_2\})} = (0, 1, 1, 0)$$

$$\boldsymbol{v}_{\phi(\{y_3\})} = (1, 0, 0, 1), \boldsymbol{v}_{\phi(\{y_4\})} = (1, 0, 1, 1)$$

$$\boldsymbol{v}_{\phi(\{y_5\})} = (0, 1, 0, 1)$$

which shows that the biadjacency matrix can be written as:

$$B_\rho = \begin{pmatrix} \boldsymbol{v}_{\phi(x_1)} \\ \vdots \\ \boldsymbol{v}_{\phi(x_4)} \end{pmatrix}$$

$$= \left(\boldsymbol{v}'_{\psi(\{y_1\})}\ \boldsymbol{v}'_{\psi(\{y_2\})}\ \boldsymbol{v}'_{\psi(\{y_3\})}\ \boldsymbol{v}'_{\psi(\{y_4\})}\ \boldsymbol{v}'_{\psi(\{y_5\})}\right)$$

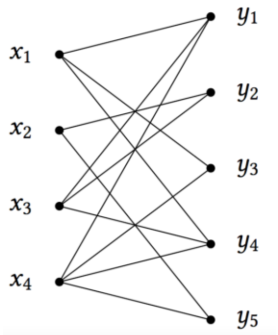

Figure 1. Example of bipartite graph G = $(\{x_1, x_2, x_3, x_4\}, \{y_1, y_{2,y_3,y_4,y_5}\};\ \rho)$.

Example 2.3. For the bipartite graph considered in Example 2.2 we have:

$$\psi(\phi(\{x_1, x_3\})) = \psi(\{y_1, y_4\}) = \{x_1, x_3, x_4\}$$

The characteristic vectors that correspond to the sets involved are:

$$\boldsymbol{v}_{\{x_1,x_3\}} = (1, 0, 1, 0)$$

$$\boldsymbol{v}_{\phi(\{x_1,x_3\})} = \boldsymbol{v}_{\{y_1,y_4\}} = (1, 0, 0, 1, 0)$$

$$\boldsymbol{v}_{\psi(\{y_1,y_4\})} = v_{\{x_1,x_3,x_4\}} = (1, 0, 1, 1)$$

In this paper we use a two-element algebraic structure S_2 defined on the set $\{0, 1\}$ equipped with two operations min and $\rightarrow$, where $\rightarrow$ is the propositional implication defined by:

$$x \rightarrow y = \begin{cases} 1 & \text{if } x = 0, \\ y & \text{if } x = 1. \end{cases}$$

Note that $x \rightarrow y = 1 - x - xy$. This operation is neither associative nor commutative. However, min is distributive over " $\rightarrow$ ", that is:

$$min\{x, y \rightarrow z\} = \min\{x, y\} \rightarrow min\{x, z\}$$

as it can be easily verified. This structure generalized the notion of semi-ring [14].

A *commutative semiring* [14] is a set S together with two binary operations " $+$ " and " $\cdot$ " which satisfy the following three axioms:

i) The operation "+" is associative and commutative and there is an additive neutral element called "0" such that $x + 0 = x$ for all $x \in S$.
ii) The operation " $\cdot$ " is also associative and commutative and there is a multiplicative neutral element called "1" such that $x \cdot 1 = 1$;
iii) The distributive law holds that:

$$(a \cdot b) + (a \cdot c) = a \cdot (b + c)$$

for all triples (a, b, c) from S

Example 2.4. The set $[0, \infty]$ is a semiring relative to the operations $x + y = \min\{x, y\}$ and $x \cdot y = xy$ for $x, y \in [0, \infty]$. The additive neutral element is ∞, while the multiplicative neutral element is 1. Also, we have the distributive equality:

$$min\{ab, ac\} = a\, min\{b, c\}$$

for $a, b, c \in [0, \infty]$.

We introduce a special product operation between binary matrices. If $A \in \{0, 1\}^{\{m\times n\}}$ and $B \in \{0,1\}^{\{n\times p\}}$, then we denote their new product C by $A \star B \in \{0,1\}^{\{m\times p\}}$. The matrix $C \in \{0,1\}^{\{m\times p\}}$ is defined as:

$$c_{ik} = min_{1\le j\le n}\ (a_{ij} \rightarrow b_{jk})$$

For a row vector $\boldsymbol{a} \in \{0,1\}^{\{1\times n\}}$ and $B \in \{0,1\}^{n\times p}$, we have $(\boldsymbol{a}\, B)_k = \min_{1\le j\le n}(a_j \rightarrow b_{jk})$. Similarly, if $\boldsymbol{c} \in \{0,1\}^p$ is a column vector we have $(B\boldsymbol{c})_j = \min_{1\le k\le p}(b_{jk} \rightarrow c_k)$.

Because of the asymmetry of the $\rightarrow$ operation, the transpose of a matrix product is distinct from the product of the transposed matrices in reverse order. Note that the $\star$ product between matrices is not associative, in general. Also, for $B \in \{0,1\}^{m\times n}$:

$$(0, \dots, 0) \star B = (1, \dots, 1)$$
and
$$(1, \dots, 1) \star B = (\min \boldsymbol{b}_1, \dots, \min \boldsymbol{b}_n)$$

where $\boldsymbol{b}_j$ is the j^{th} column of B.

Theorem 2.5. Let (ϕ_ρ, ψ_ρ) be the polarity determined by a relation $\rho \subseteq L \times R$. If $B_\rho \in \{0,1\}^{q\times r}$ is the biadjacency matrix determined by the relation ρ, then:

$$\boldsymbol{v}_{\boldsymbol{X}} \star B_\rho = \boldsymbol{v}_{\phi_\rho(X)}$$
and
$$\boldsymbol{v}_Y \star B'_\rho = \boldsymbol{v}_{\psi_\rho(Y)}$$

for any subsets $X \subseteq L$ and $Y \subseteq R$.

Proof. Let $L = \{x_1, \dots, x_q\}$ and $R = \{y_1, \dots, y_r\}$. If $X \subseteq L$, $X = \{x_{i_1}, \dots, x_n\}$, it follows that $\boldsymbol{x} = (a_1, \dots, a_q) \in \{0,1\}^q$ is a vector that contains 1s in positions $i_1, \dots, i_n$.

We have $\left(\boldsymbol{v}_X \star B_\rho\right)_k = 1$ if and only if $min_{1\leq j\leq n}(a_j \to b_{jk}) = 1$, or $a_j \to b_{jk} = 1$ for all $j \in \{1, \dots, n\}$. This amounts to the fact that $a_j = 1$ implies $b_{jk} = 1$, which is equivalent to saying that for all $x_j \in X$ we have $(x_j, y_k) \in \rho$, that is, $y_k \in \phi_\rho(X)$. This means that $\boldsymbol{v}_X \star B_\rho$ equals $\boldsymbol{v}_{\phi_\rho(X)}$.

Suppose now that $Y \subseteq R$ and $Y = \{y_{k_1}, \dots, y_{k_m}\}$. Then, $\boldsymbol{v}_Y = (c_1, \dots, c_r)$, where $\boldsymbol{v}_Y$ contains 1 in the positions $k_1, \dots, k_m$.

We have $\left(\boldsymbol{v}_Y \star B'_\rho\right)_i = 1$ if and only if $min_{1\leq \ell\leq r}(c_\ell \to b_{i\ell}) = 1$, or $c_\ell \to b_{i\ell} = 1$ for $\ell \in \{1, \dots, r\}$. This means that $c_\ell = 1$ implies $b_{i\ell} = 1$, which is equivalent to saying that for all $y_i \in Y$ we have $(x_\ell, y_i) \in \rho$, that is $x_\ell \in \psi_\rho(Y)$. This means that $\boldsymbol{v}_Y \star B_\rho$ equals $\boldsymbol{v}_{\psi_\rho(Y)}$.

Corollary 2.6. Let (ϕ_ρ, ψ_ρ) be the polarity determined by a relation $\rho \subseteq L \times R$. If $B_\rho \in \{0,1\}^{q\times r}$ is the biadjacency matrix determined by the relation ρ, then:

$$\boldsymbol{v}_{\psi(\phi(X))} = \left(\boldsymbol{v}_X \star B_\rho\right) \star B'_\rho$$

$$\boldsymbol{v}_{\phi(\psi(Y))} = \left(\boldsymbol{v}_Y \star B'_\rho\right) \star B_\rho$$

For anu subsets $X \subseteq L$ and $Y \subseteq R$.

Proof: The corollary follows directly from Theorem 2.5.

Example 2.7. Consider the relation $\rho \subseteq \{x_1, x_2, x_3\} \times \{y_1, y_2\}$ whose bipartite graph is shown in Fig. 2. The biconnectivity matrix is:

$$B_\rho = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

The characteristic vector of $\{x_1\}$ is $\boldsymbol{v}_{\{x_1\}} = (1, 0, 0)$.

Note that:

$$(1, 0, 0) \star B_\rho = (0, 1)$$

hence $\boldsymbol{v}_{\phi(\{x_1\})} = \boldsymbol{v}_{\{x_1\}} \star B_\rho = (0, 1)$.

In other words, $\phi(\{x_1\}) = \{y_2\}$. Since $\boldsymbol{v}_{\{y_2\}} B'_\rho = (1, 1, 1)$, it follows that $\psi_\rho\left(\phi_\rho(\{x_1\})\right) = \{x_1, x_2, x_3\}$.

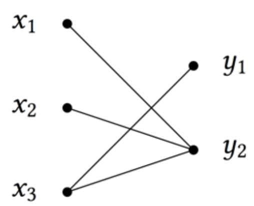

Figure 2. Example of bipartite graph G = $(\{x_1, x_2, x_3\}, \{y_1, y_2\};\ \rho)$.

Example 2.8. For the bipartite graph G=$(\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{y_1, y_2, y_3, y_4, y_5, y_6\}, \rho)$ shown in Fig. 3, the biadjacency matrix is:

$$B_\rho = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Thus, a vector $\boldsymbol{v}_{\phi(\{x_i\})}$ is the i^{th} line of the matrix B_ρ: The characteristic vectors that correspond to the sets involved are:

$$\boldsymbol{v}_{\phi(\{x_1\})} = (1, 0, 0, 0, 0, 0), \quad \boldsymbol{v}_{\phi(\{x_2\})} = (1, 1, 1, 1, 0, 0)$$

$$\boldsymbol{v}_{\phi(\{x_3\})} = (0, 1, 1, 1, 1, 0), \quad \boldsymbol{v}_{\phi(\{x_4\})} = (0, 1, 1, 1, 0, 1)$$

$$\boldsymbol{v}_{\phi(\{x_5\})} = (0, 1, 1, 1, 0, 1), \quad \boldsymbol{v}_{\phi(\{x_6\})} = (0, 0, 0, 0, 0, 1)$$

and a vector $\boldsymbol{v}_{\psi(\{y_j\})}$ is the transpose of the j^{th} column of B:

$$\boldsymbol{v}_{\phi(\{y_1\})} = (1, 1, 0, 0, 0, 0), \quad \boldsymbol{v}_{\phi(\{y\})} = (0, 1, 1, 1, 1, 0)$$

$$\boldsymbol{v}_{\phi(\{y_3\})} = (0, 1, 1, 1, 1, 0), \quad \boldsymbol{v}_{\phi(\{y_4\})} = (0, 1, 1, 1, 1, 0)$$

$$\boldsymbol{v}_{\phi(\{y_5\})} = (0, 0, 1, 0, 0, 0), \quad \boldsymbol{v}_{\phi(\{y_6\})} = (0, 0, 0, 0, 0, 1)$$

Which shows that the be biadjacency matrix can be written as:

$$B_\rho = \begin{pmatrix} \boldsymbol{v}_{\phi(\{x_1\})} \\ \vdots \\ \boldsymbol{v}_{\phi(\{x_6\})} \end{pmatrix}$$

$$= \left(\boldsymbol{v}'_{\psi(\{y_1\})}\ \boldsymbol{v}'_{\psi(\{y_2\})}\ \boldsymbol{v}'_{\psi(\{y_3\})}\ \boldsymbol{v}'_{\psi(\{y_4\})}\ \boldsymbol{v}'_{\psi(\{y_5\})}\ \boldsymbol{v}'_{\psi(\{y_6\})}\right)$$

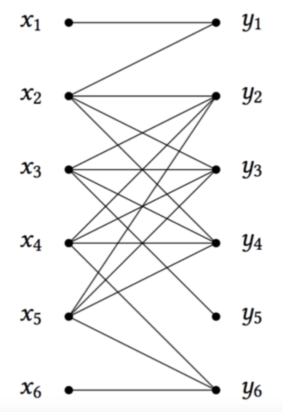

Figure 3. Example of bipartite graph G = $(\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{y_1, y_2, y_3, y_4, y_5, y_6\}, \rho)$.

Example 2.9. For the bipartite graph considered in Example 2.8 we have:

$$\psi\left(\phi(\{x_2, x_3\})\right) = \psi(\{y_2, y_3, y_4\}) = \{x_2, x_3, x_4, x_5\}$$

The characteristic vectors that correspond to the sets involved are:

$$\boldsymbol{v}_{\{x_2, x_3\}} = (0, 1, 1, 0, 0, 0)$$

$$\boldsymbol{v}_{\phi(\{x_1, x_3\})} = \boldsymbol{v}_{\{y_2, y_3, y_4\}} = (0, 1, 1, 1, 0, 0)$$

$$\boldsymbol{v}_{\psi(\{y_2, y_3, y_4\})} = \boldsymbol{v}_{\{x_2, x_3, x_4, x_5\}} = (0, 1, 1, 1, 1, 0)$$

III. Bicliques in Bipartite Graphs

Definition 3.1. A *biclique* in a bipartite graph G=$(L, R;\ \rho)$ is a subgraph of G induced by the sets U, V such that $U \subseteq L, V \subseteq R$ and $U \times V \subseteq \rho$. This biclique is denoted by (U, V).

A bipartite graph G= $(L, R;\ \rho)$ generates a polarity defined by the relation ρ between L and R. Namely, we have:

$$\phi_\rho(X) = \{y \in R \mid \{x, y\} \in \rho \text{ for every } x \in X\}$$

$$\psi_\rho(Y) = \{x \in L \mid \{x, y\} \in \rho \text{ for every } y \in Y\}$$

Theorem 3.2. Let G = $(L, R;\ \rho)$ be a bipartite graph. A pair of sets (U, V), where $U \subseteq L$ and $V \subseteq R$ is a *biclique* in G if and only if $V \subseteq \phi_\rho(U)$ or, equivalently, if $U \subseteq \psi_\rho(V)$.

Proof. If (U, V) is a biclique, $U \times V \subseteq \rho$, and this implies $U \subseteq \psi_\rho(V)$ and $V \subseteq \phi_\rho(U)$. Note that these inclusions are equivalent.

Conversely, suppose that $U \subseteq \psi_\rho(V)$. The definition of $\psi_\rho(V)$ means that for every $x \in U$ we have $x \in \psi_\rho(V)$, hence $U \times V \subseteq \rho$.

Definition 3.3. Let $B_1 = (X_1, Y_1)$ and $B_2 = (X_2, Y_2)$ be two bicliques in a bipartite graph G=$(L, R;\ \rho)$. We write $B_1 \sqsubseteq B_2$ (and we say B_2 contains B_1) if $X_1 \subseteq X_2$ and $Y_1 \subseteq Y_2$. A biclique (B, B') is *maximal* if there is no biclique distinct from (B, B') that contains (B, B').

There are two variants of the maximal biclique problem; the *vertex maximum biclique problem* that seeks to find a biclique (U, V) such that $|U| + |V|$ is maximal, and the *edge maximum biclique problem* that seeks to find a biclique with the largest number of edges. The first problem can be solved in polynomial time (see, for example the problem GT24 in [15]); the second problem is NP-complete as shown in [16].

Theorem 3.4. If (ϕ_ρ, ψ_ρ) is a polarity on the sets L and R, then:

$$\phi_\rho(\psi_\rho(\phi_\rho(X))) = \phi_\rho(X)$$

and

$$\psi_\rho(\phi_\rho(\psi_\rho(Y))) = \psi_\rho(Y)$$

Proof. As we noted, we have $X \subseteq \psi_\rho(\phi_\rho(X))$ hence $\phi_\rho(X) \supseteq \phi_\rho(\psi_\rho(\phi_\rho(X)))$ The reverse inclusion follows by substituting $\phi_\rho(X)$ for Y in Equality (4), which implies the first inequality of the theorem. The proof of the second equality is similar.

Theorem 3.5. Let G= $(L, R;\ \rho)$ be a bipartite graph. A pair of sets of vertices (X, Y) is a maximal biclique if $\phi_\rho(X) = Y$ and $\psi_\rho(Y) = X$.

Proof. Let $\phi_\rho(X) = Y$ and $\psi_\rho(Y) = X$. Clearly, (X, Y) is a biclique. Suppose that (X, Y) is not maximal and let (U, V) be a biclique such that $X \subseteq U$ and $Y \subseteq V$.

Since (U, V) is a biclique, we have $V \subseteq \phi_\rho(U)$. Next, we have $\phi_\rho(U) \subseteq \phi_\rho(X)$ because $X \subseteq U$. Finally, $\phi(X) = Y$ by hypothesis, hence $V \subseteq X$. Since we assumed the reverse inclusion, we have $X = V$. Similarly, $Y = U$, so (X, Y) is indeed a maximal biclique.

Conversely, suppose that (X, Y) is a maximal biclique. Since (X, Y) is a biclique we have $Y \subseteq \phi_\rho(X)$ and $X \subseteq \psi_\rho(Y)$ by Theorem 3.2. Thus, (X, Y) is contained in the biclique $(\phi_\rho(Y), \psi_\rho(X))$. By the maximality of (X, Y) we have $Y = \phi_\rho(X)$ and $X = \psi_\rho(Y)$.

Corollary 3.6. In a bipartite graph G= $(L, R; \rho)$ the maximal biclique generated by a set of vertices $S \subseteq L$ is a pair of sets $B_s = (U, V)$ given by:

$$V = \phi_\rho(S) \text{ and } U = \psi_\rho(\phi_\rho(S))$$

The maximal biclique generated by a set of vertices $T \subseteq R$ is the pair of sets $B^T = (U, V)$ given by:

$$U = \psi_\rho(T) \text{ and } V = \phi_\rho(\psi_\rho(T))$$

Proof. For $S \subseteq L, V = \phi_\rho(S), U = \psi_\rho(\phi_\rho(S))$ we have:

$$\phi_\rho(U) = \phi_\rho(\psi_\rho(\phi_\rho(S))) = \phi_\rho(S) = V$$

(by the first equality of Theorem 3.4)

$$\psi_\rho(V) = \psi_\rho(\phi_\rho(S)) = U$$

which shows that (U, V) is the maximal biclique generated by S by Theorem 3.5. The argument for T is entirely similar.

Corollary 3.7. The following statements are equivalent:

(1) (X, Y) is a maximal biclique

(2) $(\boldsymbol{v}_X \star B_\rho) \star B'_\rho = \boldsymbol{v}_X$ and $\boldsymbol{v}_X \star B_\rho = \boldsymbol{v}_Y$;

(3) $(\boldsymbol{v}_Y \star B'_\rho) \star B_\rho = \boldsymbol{v}_Y$ and $\boldsymbol{v}_Y \star B'_\rho = \boldsymbol{v}_X$

Lemma 3.8. Let G = $(L, R; \rho)$ be a bipartite graph. For any set of sets $\{S_1, S_2, \dots, S_k\}$, $2 \le k \le 2^{|L|}$, where $S_i \subset L$, for any $1 \le i < k$, satisfying $S_i \subset S_j$ for any $1 \le i < j \le k$, we have:

$$\phi_\rho(S_1) \ge \phi_\rho(S_2) \ge \cdots \ge \phi_\rho(S_k)$$

For any set of sets $\{T_1, T_2, \dots, T_m\}, 1 \le m \le 2^{|R|}$, where $T_i \subset R$, for any $1 \le i < m$, satisfying $T_i \subset T_j$ for any $1 \le i < j \le m$, we have:

$$\psi_\rho(T_1) \ge \psi_\rho(T_2) \ge \ \dots \ge \psi_\rho(T_m)$$

Proof. It results from the fact that

$$\phi_\rho(S) = \textstyle\bigcap_{s \in S} \phi_\rho(\{s\})$$

$$\psi_\rho(T) = \textstyle\bigcap_{t \in T} \psi_\rho(\{t\})$$

IV. Recommendation Systems and Bicliques

Assume that for a bipartite graph G = $(L, R; \rho)$, L represents a set of users, R represents a set of items, and that we have an edge between a user $u \in R$ and an item $v \in R$ when user u likes item v. The existence of a large biclique containing user u means the preferences of user u are spread across a large number of users and items. The more preferences the user has and the more users share these preferences, the more central this user is.

An item has a higher connectivity if users that like this item share a large number of preferred items.

Definition 4.1. Let G=$(L, R; \rho)$ be a bipartite graph. The *biclique connectivity of a set of vertices* $S \subseteq L$ is given by:

$$c_S = |\psi_\rho(\phi_\rho(S))| \cdot |\phi_\rho(S)|$$

Similarly, the biclique connectivity of a set of vertices $T \subseteq R$ is:

$$c^T = |\psi_\rho(T)| \cdot |\phi_\rho(\psi_\rho(T))|$$

We refer to $c_{\{u\}}$ as the biclique connectivity of $u \in L$; similarly, $c^{\{v\}}$ is the biclique connectivity of $v \in R$.

Example 4.2. Let G=$(L, R; \rho)$ be the bipartite graph given in Fig. 1. Vertex x_1 generates the maximal biclique

$(\{x_1, x_4\}, \{y_1, y_3, y_4\})$. The biclique connectivity of x_1 is given by:

$$c_{\{x_1\}} = 2 \cdot 3 = 6$$

In the same bipartite graph, the set of vertices $\{x_2, x_4\}$ generates the maximal biclique $(\{x_2, x_4\}, \{y_5\})$. Thus, the biclique connectivity of the set $\{x_2, x_4\}$ is given by:

$$c_{\{x_2,x_4\}} = 2 \cdot 1 = 2$$

The higher the connectivity of a set of users, the more popular a large number of items liked by these users are among a large number of items and users.

Let G=$(L, R; \rho)$ be a bipartite graph where L represents a set of users, R represents a set of items, and ρ is a set of edges representing the likes. A set of users has a large biclique connectivity when the given users share a large number of common likes, or their common liked items are liked by a large number of other users. Similarly, a set of items has a large connectivity value if a large number of users like all these items, or there are many other items that are all liked by the users who like the items from the given set.

There are cases in which we would like to know when a set of users have many common likes, or when a set of items are commonly liked by a large number of users. We can consider that the more common preferences some users have the more alike those users are. And for a given set of items, the more users like all these items, the more things in common these items have. Hence, we introduce a notion of *similarity* between a set of vertices that uses only one side of the maximal biclique generated by that set.

Definition 4.3. Let G=$(L, R; \rho)$ be a bipartite graph. The *biclique similarity* of a set of vertices $S \subseteq L$ is given by:

$$s_S = \left|\phi_\rho(S)\right|$$

The biclique similarity of a set of vertices $T \subseteq R$ is:

$$s^T = \left|\psi_\rho(T)\right|$$

Lemma 4.4. Let G=$(L, R; \rho)$ be a complete bipartite graph. The biclique connectivity of any set of vertices $S \subseteq L, S \neq \emptyset$ is given by:

$$c_S = |L| \cdot |R|$$

Similarly, the biclique connectivity of any set of vertices $T \subseteq R, T \neq \emptyset$ is:

$$c^T = |L| \cdot |R|$$

Proof. It results from the fact that all vertices from L are connected to all vertices from R. This means that the maximal biclique generated by any sets of vertices either in L or in R is the complete bipartite graph itself.

Lemma 4.5. Let G=$(L, R; \rho)$ be a complete bipartite graph. The biclique similarity of any set of vertices $S \subseteq L, S \neq \emptyset$ is given by:

$$s_S = |R|$$

Similarly, the biclique similarity of any set of vertices $T \subseteq R, T \neq \emptyset$ is:

$$s^T = |L|$$

Proof. It results from the fact that all vertices from L are connected to all vertices from R. This means that the maximal biclique generated by any sets of vertices either in L or in R is the complete bipartite graph itself.

Example 4.6. Using the bipartite graph from Fig. 3, we have the following connectivity measures:

$$c_{\{x_2,x_3\}} = |\{x_2, x_3, x_4, x_5\}| \cdot |\{y_2, y_3, y_4\}| = 4 \cdot 3 = 12$$

$$c_{\{x_1,x_2\}} = |\{x_1, x_2\}| \cdot |\{y_1\}| = 2 \cdot 1 = 2$$

$$c_{\{x_2\}} = |\{x_2\}| \cdot |\{x_1, x_2, x_3, x_4\}| = 1 \cdot 4 = 4$$

$$c_{\{x_1,x_3\}} = 0$$

$$c_{\{x_4,x_5\}} = |\{x_4, x_5\}| \cdot |\{y_2, y_3, y_4, y_6\}| = 2 \cdot 4 = 8$$

$$c_{\{x_4\}} = |\{x_4, x_5\}| \cdot |\{y_2, y_3, y_4, y_6\}| = 2 \cdot 4 = 8$$

$$c_{\{x_2,x_4,x_5\}} = |x_2, x_3, x_4, x_5| \cdot |\{y_2, y_3, y_4\}| = 4 \cdot 3 = 12$$

$$c^{\{y_1\}} = |\{x_1, x_2\}| \cdot |\{y_1\}| = 2 \cdot 1 = 2$$

$$c^{\{y_1,y_2\}} = |\{x_2\}| \cdot |\{y_1, y_2, y_3, y_4\}| = 1 \cdot 4 = 4$$

$$c^{\{y_6\}} = |\{x_4, x_5, x_6\}| \cdot |\{y_6\}| = 3 \cdot 1 = 3$$

Some of the similarity measures for vertices in L:

$$s_{\{x_1,x_2\}} = |\{y_1\}| = 1$$

$$s_{\{x_1,x_3\}} = 0$$

$$s_{\{x_1,x_4\}} = 0$$

$$s_{\{x_1,x_5\}} = 0$$

$$s_{\{x_1,x_6\}} = 0$$

$$s_{\{x_2,x_3\}} = |\{y_2, y_3, y_4\}| = 3$$

$$s_{\{x_2,x_4\}} = |\{y_2, y_3, y_4\}| = 3$$

$$s_{\{x_2,x_5\}} = |\{y_2, y_3, y_4\}| = 3$$

$$s_{\{x_2,x_6\}} = 0$$

$$s_{\{x_3,x_4\}} = |\{y_2, y_3, y_4\}| = 3$$

$$s_{\{x_3,x_5\}} = |\{y_2, y_3, y_4\}| = 3$$

$$s_{\{x_3,x_6\}} = 0$$

$$s_{\{x_4,x_5\}} = |\{y_2, y_3, y_4, y_6\}| = 4$$

$$s_{\{x_4,x_6\}} = |\{y_6\}| = 1$$

$$s_{\{x_5,x_6\}} = |\{y_6\}| = 1$$

$$s_{\{x_2,x_4,x_5\}} = |\{y_2, y_3, y_4\}| = 3$$

Some of the similarity measures of vertices in R:

$$s^{\{y_1,y_2\}} = |\{x_2\}| = 1$$

$$s^{\{y_1,y_3\}} = |\{x_2\}| = 1$$

$$s^{\{y_1,y_4\}} = |\{x_2\}| = 1$$

$$s^{\{y_1,y_5\}} = 0$$

$$s^{\{y_1,y_6\}} = 0$$

$$s^{\{y_2,y_3\}} = |\{x_2, x_3, x_4, x_5\}| = 4$$

$$s^{\{y_2,y_4\}} = |\{x_2, x_3, x_4, x_5\}| = 4$$

$$s^{\{y_2,y_5\}} = |\{x_3\}| = 1$$
$$s^{\{y_2,y_6\}} = |\{x_4, x_5\}| = 2$$
$$s^{\{y_3,y_4\}} = |\{x_2, x_3, x_4, x_5\}| = 4$$
$$s^{\{y_3,y_5\}} = |\{x_3\}| = 1$$
$$s^{\{y_3,y_6\}} = |\{x_4, x_5\}| = 2$$
$$s^{\{y_4,y_5\}} = |x_3| = 1$$
$$s^{\{y_4,y_6\}} = |\{x_4, x_5\}| = 2$$
$$s^{\{y_5,y_6\}} = 0$$

We need to determine the order in which to recommend new items to users. For a given user and a new item, we compute the average of similarities between the new item and items we know the given user likes. This will associate a numerical value to each new item. The new items will be sorted according to these values in descending order. This will be the order in which the new items will be recommended to that given user.

Let G be a bipartite graph represented by biadjacency matrix $B \in \{0,1\}^{|L|\times|R|}$. For any pair of $u_i \in L$ and $t_j \in R$ for which $b_{u_i,t_j} = 0$, the *likelihood* that user u_i will like item t_j is given by:

$$\ell(u_i, t_j) = \frac{\sum_{k=1,k\neq j}^{|R|} s^{\{t_k,t_j\}} \cdot b_{u_i t_k}}{\sum_{k=1,k\neq j}^{|R|} b_{u_i,t_k}}$$

A similar version that uses the biclique similarity between users instead of the biclique similarity between items is given below:

$$\ell'(u_i, t_j) = \frac{\sum_{k=1,k\neq i}^{|L|} s_{\{u_k,u_i\}} \cdot b_{u_k t_j}}{\sum_{k=1,k\neq i}^{|L|} b_{u_k,t_j}}$$

Note that for function ℓ, we would need user u_i to have liked at least one item, and for ℓ' we would require the item t_j to be liked by at least one person. Otherwise the denominator of the two fractions would be zero. This is related to the cold start problem. One possible solution could be to temporarily assign a small number of random likes to any new users and new items.

Example 4.7. Using the bipartite graph $G=(L, R; \rho)$ from Fig. 3, where L represents a set of users and R represents a set of items, some of the likelihoods are given below.

Using the similarity between items we get:

$$\ell(x_2, y_6) = \frac{s^{\{y_1,y_6\}} + s^{\{y_2,y_6\}} + s^{\{y_3,y_6\}} + s^{\{y_4,y_6\}}}{4} = \frac{0+2+2+2}{4} = \frac{6}{4} = 1.5$$

$$\ell(x_1, y_6) = \frac{s^{\{y_1,y_6\}}}{1} = \frac{0}{1} = 0$$

This means that between user x_1 and user x_2, user x_2 is more likely to like item y_6. While the likelihood that x_2 will like item y_6 is 1.5, the likelihood that user x_1 likes the same item is 0.

Using the similarity between users, we have:

$$\ell'(x_2, y_6) = \frac{s_{\{x_2,x_4\}} + s_{\{x_2,x_5\}} + s_{\{x_2,x_6\}}}{3} = \frac{3+3+0}{3} = 2$$

$$\ell'(x_1, y_6) = \frac{s_{\{x_1,x_4\}} + s_{\{x_1,x_5\}} + s_{\{x_1,x_6\}}}{3} = \frac{0+0+0}{3} = 0$$

Same as with the similarity between items, when using the similarity between users, there is a likelihood of 0 that user x_1 will like item y_6. The likelihood that user u_2 will like item y_6, is higher than when using the similarity between items.

Definition 4.8. Given a bipartite graph $G=(L, R; \rho)$, where L is a set of users, R is a set of items., and ρ is a set of edges representing likes. ρ induces a set of sets of liked items given by:

$$likes(L) = \{\phi(u) | \, u \in L\}$$

Definition 4.9 Given a bipartite graph $G=(L, R; \rho)$, where L is a set of users, R is a set of items, and ρ is a set of edges representing likes. A *chain* $C = \{C_1, \dots, C_k\}$, where $2 \leq k \leq |likes(L)|$, is an ordered subset of $likes(L)$ with property:

$$C_i \subset C_{i+1}$$
for any $1 \leq i < k$

C is a *maximal chain*, if no other chain $C' \supset C$ can be formed from $likes(L)$.

Note that $C' \supset C$, if $C' \neq C$ and all elements of C are present in C'.

The likes of a user will be part of at least one chain if they are either a proper subset or proper superset of the likes of at least one different user. The set of likes of a user u will be the first element in a chain, if there is at least another user whose likes are a proper superset of the user u's likes. The set of likes of a user u will be placed in the last position of a chain if there is at least one user whose likes are a proper subset of the user u's likes. The likes of a user u will be placed in the middle of a chain, if there is at least one user whose likes are a proper superset of user u's likes, as well as at least one another user whose likes are a proper subset of user u's likes.

Example 4.10. Let $G=(L, R; \rho)$ be the bipartite graph from Fig. 1, where L represents a set of users and R represents a set of items. We have:

$$likes(L) = \{\{y_1, y_3, y_4\}, \{y_2, y_5\}$$
$$\{y_1, y_2, y_4\}, \{y_1, y_3, y_4, y_5\}\}$$

This set contains chain $(\{y_1, y_3, y_4\}, \{y_1, y_3, y_4, y_5\})$, which is a maximal chain.

Example 4.11. Let $G=(L, R; \rho)$ be the bipartite graph from Fig. 2, where L represents a set of users and R represents a set of items.

We have:

$$likes(L) = \{\{y_2\}, \{y_1, y_2\}\}$$

This set contains chain $(\{y_2\}, \{y_1, y_2\})$, which is a maximal chain.

Example 4.12. Let G=$(L,R;\rho)$ be the bipartite graph from Fig. 3, where L represents a set of users and R represents a set of items.

We have:

$$likes(L) = \{\{y_1\},\{y_1,y_2,y_3,y_4\},\{y_2,y_3,y_4,y_5\}$$
$$\{y_2,y_3,y_4,y_6\},\{y_6\}\}$$

The set contains two chains, each of them of length two, given by:

$$(\{y_1\},\{y_1,y_2,y_3,y_4\})$$
$$(\{y_6\},\{y_2,y_3,y_4,y_6\})$$

Both chains are maximal.

This notion of chains might be useful when analyzing a bipartite graph. We plan to investigate this notion and see how it relates to association rules for items.

We introduce next the *Biclique Similarity Ordering Recommendation* (BISOR) Algorithm, a technique that determines the order in which to recommend items to users.

Algorithm 4.1: The BISOR Algorithm.

```
input  : biadjacency matrix B representing likes of users L to items R;
output : A map that contains an ordered set of items for each user.
begin
  usersOrderedRecommendations = empty map of format
    Map < User, OrderedSet < Item >>;
  // get recommendations for each user
  for user in L do
    newItems = getUserNewItems (B, user)
    Map < Item, value > likes = empty map;
    for item in newItems do
      likes.put(item, ℓ(user, item))
      // or likes.put(item, ℓ'(user, item))
    end
    likes = orderByValueDesc(likes);
    usersOrderedRecommendations.put(user,
    likes.keySet())
  end
  return usersOrderedRecommendations;
end
```

V. Experimental Results

We present experimental results for the version of the algorithm that uses the similarity between items. The algorithm was implemented in Java.

Offline validation techniques were used to verify our recommendation algorithm. The steps are straightforward: we remove a number of likes, we run the recommendation algorithm, and then we validate against the likes we have removed.

For the step that selects the likes to be removed we propose two methods:

Sampling-Method 1: We randomly select a percentage of users. For each user from that set, we remove a number of likes (i.e. edges) only if that user contains a given minimum of preferences (i.e. edges). In the experiments presented below, if the selected user had at least five likes, we randomly removed two of them.

Sampling-Method 2: We randomly remove a percentage of edges representing likes.

The recommendation algorithm we propose determines the order in which to recommend other items to users. The recommendation algorithm looks at what positions in the ordered recommendation list are the likes we have previously removed. The position in the list is not enough for validation. Recommending an item at position 5 out of 10 is not the same as recommending an item at position 5 out of 1000. Therefore, we look at the normalized index of the items in the ordered list of recommendations. For example, if we recommend 1,000 items to the user, and a given item is at position 5 in the ordered list, we say that the normalized index of this item is $5/1000 = 0.005$. The smaller the average of the normalized index across all validated likes is, the better the recommendation system. A normalized index, named r, is used in [9] to validate the recommendation result.

We present experimental results run on three real-world datasets. Each experiment was run 10 times. We present the min, max and average values of the measurements across all runs. We compared the validation results obtained by running our BISOR algorithm, with the results obtained by running one of the most widely used Collaborative Filtering (CF) algorithm [17], [18], [9]. We implemented the CF algorithm using their proposed method with similarity between items.

The Sushi dataset [19] contains 5000 users and 100 types of sushi. The original dataset contains the ratings given by users to different types of sushi. Ratings $\in \{0,1,2,3,4\}$, with 0 representing a dislike, and 4 representing a like. We created the biadjacency matrix of likes by taking into account only ratings greater than 2. The resulted matrix contains an edge between a user and a sushi type if that user rated that sushi with a 2, a 3 or a 4. Table I presents experimental results using both Sampling-Method 1 and Sampling-Method 2, which shows BISOR algorithm performed substantially better than the CF algorithm.

TABLE I. Sushi Dataset

Sampling-Method 1 – 10 runs			
	No. Likes Validated	BISOR Avg. Normalized Index	CF Avg Normalized Index
10% Users	avg = 965 min = 952 max = 980	avg = 0.2298 min = 0.22 max = 0.2365	avg = 0.7319 min = 0.7191 max = 0.7445
Sampling-Method 2 – 10 runs			
10% Edges	avg = 4142 min = 4142 max = 4142	avg = 0.2255 min = 0.2193 max = 0.2305	avg = 0.7348 min = 0.7292 max = 0.7402

MovieLens 100k Dataset [20] contains about 100k ratings between 943 users and 1682 movies. Ratings $\in \{1,2,3,4,5\}$, with 1 representing a dislike, and 5 representing a like. We created the biadjacency matrix of likes by taking into account only ratings ≥ 3. Table II presents experimental results, which shows BISOR algorithm performed better than the CF algorithm. When using Sampling-Method 1, the average normalized index across all runs was 0.104 for BISOR and 0.2127 for CF algorithm. For Sampling-Method 2, the averages were 0.1166 for BISOR and 0.2153 for CF.

TABLE II. MOVIELENS 100K DATASET

Sampling-Method 1 – 10 runs			
	No. Likes Validated	BISOR Avg. Normalized Index	CF Avg Normalized Index
10% Users	avg = 188 min = 188 max = 188	avg = 0.104 min = 0.0881 max = 0.1277	avg = 0.2127 min = 0.1908 max = 0.2271
Sampling-Method 2 – 10 runs			
10% Edges	avg = 8252 min = 8252 max = 8252	avg = 0.1166 min = 0.1141 max = 0.119	avg = 0.2153 min = 0.2096 max = 0.2192

MovieLens 1 Million Dataset [20] contains 6040 users, 3952 movie titles and about 1,000,000 ratings. Ratings $\in \{1, 2, 3, 4, 5\}$, with 1 representing a dislike, and 5 representing a like. We created the biadjacency matrix of likes by taking into account only ratings ≥ 3. Table III presents experimental results, which shows BISOR performed substantially better than the CF algorithm.

TABLE III. MOVIELENS 1 MILLION DATASET

Sampling-Method 1 – 10 runs			
	No. Likes Validated	BISOR Avg. Normalized Index	CF Avg Normalized Index
10% Users	avg = 1207.2 min = 1204 max = 1208	avg = 0.0969 min = 0.0864 max = 0.1063	avg = 0.4681 min = 0.4603 max = 0.4786
Sampling-Method 2 – 10 runs			
10% Edges	avg = 57528 min = 57528 max = 57528	avg = 0.0971 min = 0.0965 max = 0.098	avg = 0.4581 min = 0.4571 max = 0.4591

The experimental results run on all three real-world datasets show better results for BISOR algorithm than for one of the most widely used collaborative filtering algorithm. BISOR performed significantly better on the Sushi dataset and MovieLens 1 million dataset and slightly better on the MovieLens 100k dataset. BISOR algorithm showed a more consistent performance across multiple datasets, with an average normalized index in $0.08x$ to $0.23x$ range across all runs. For the CF algorithm, the performance varied substantially from one dataset to another, with a best average normalized index in $0.19x$ to $0.22x$ range for MovieLens 100k, an average normalized index in the range of $0.45x$ to $0.47x$ for MovieLens 1 million, and $0.71x$ to $0.74x$ range for Sushi dataset.

VI. CONCLUSION AND FUTURE WORK

The notion of maximal biclique generated by a set of vertices in a bipartite graph and a measure of biclique similarity are applied to the formulation of the Biclique Similarity Ordering Recommendation (BISOR) Algorithm, a method that leverages the connectivity patterns within a bipartite graph of likes to determine the order in which to recommend items to users. Our approach is using polarities generated by binary relations.

We validated our findings using three real-world datasets: *Sushi*, *MovieLens 100k* and *MovieLens 1 Million*. Compared with one of the most widely used Collaborative Filtering (CF) algorithm, the BISOR algorithm performed substantially better on the Sushi and MovieLens 1 million datasets and slightly better on the MovieLens 100k dataset.

We are seeking to extend this approach to incorporate users' dislikes. We would also like to further investigate whether giving more weight to items/users with high biclique connectivity would improve our algorithm. We also plan to further investigate how the notion of chains in bipartite graphs could be used when analyzing a bipartite graph and the association between different vertices.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Cristina Maier and Dan Simovici contributed to the research. Cristina Maier implemented the algorithms and ran the experiments. All authors approved the final version of the manuscript.

REFERENCES

[1] G. Pavlopoulos, P. Kontou, A. Pavlopoulou, C. Bouyioukos, E. Markou, and P. Bagos, "Bipartite graphs in systems biology and medicine: A survey of methods and applications," *GigaScience*, vol. 7, pp. 1-31, 2018.

[2] M. Balabanović and Y. Shoham, "Fab: Content-Based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[3] R. J. Mooney and L. Roy, "Content-Based book recommending using learning for text categorization," in *Proc. the Fifth ACM Conference on Digital Libraries*, 2000, pp. 195-204.

[4] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61-70, Dec. 1992.

[5] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of Netnews," in *Proc. ACM Conference on Computer Supported Cooperative Work*, 1994, pp. 175-186.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[7] G. Zhang, Y. Liu, and X. Jin, "A survey of autoencoder-based recommender systems," *Frontiers of Computer Science*, vol. 14, pp. 430-450, 2019.

[8] X. Li and H. Chen, "Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach," *Decision Support Systems*, vol. 54, pp. 880-890, 2013.

[9] T. Zhou, J. Ren, M. S. Medo, and Y. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, p. 046115, 2007.

[10] J. Kunegis, "Exploiting the structure of bipartite graphs for algebraic and spectral graph theory applications," *Internet Mathematics*, vol. 11, 2014.

[11] C. Maier and D. Simovici, "On biclique connectivity in bipartite graphs and recommendation systems," in *Proc. the 5th International Conference on Information System and Data Mining*, 2021.

[12] G. Birkhoff, *Lattice Theory*, American Mathematical Society, Colloquium Publications, 1995, vol. 25.

[13] A. S. Asratian, T. M. J. Denley, and R. Häggkvist, *Bipartite Graphs and their Applications*, UK: Cambridge University Press, 1998.

[14] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 325-343, 2000.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability - A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1979.

[16] R. Peeters, "The maximum edge biclique problem is NP-complete," *Discrete Applied Mathematics*, vol. 131, pp. 651-654, 2003.

[17] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53, 2004.
[18] P. Laureti, L. Moret, Y. C. Zhang, and Y. K. Yu, “Information filtering via iterative refinement,” *Europhysics Letters (EPL)*, vol. 75, no. 6, pp. 1006-1012, Sep. 2006.
[19] T. Kamishima and S. Akaho, “Efficient clustering for orders,” in *Proc. Sixth IEEE International Conference on Data Mining - Workshops*, 2006, pp. 274-278.
[20] F. M. Harper and J. A. Konstan, “The MovieLens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, article 19, Dec. 2015.

Cristina Maier studied Computer Science at the University of Massachusetts Boston. Her current research interest is in the area of data mining and machine learning, where she focuses on vertex centralities in graphs, clustering, measurements in bipartite graphs, recommendation systems and optimization problems.

Dr. Dan Simovici is a professor of Computer Science at the University of Massachusetts Boston and an associate of Dana-Farber Cancer Institute in Boston. Dr. Simovici obtained his Ph.D. from the University of Bucharest, Romania. Dr. Simovici served as a visiting professor at the Tohoku University in Japan and at the University of Science and Technology of Lille, France. His current research is in the area of data mining and machine learning and served on program committees of the major data mining conferences. Dr. Simovici’s work includes eighteen books and 193 research papers. His latest book, “Clustering – Theoretical and Practical Aspects” appeared in Summer 2021 at World Scientific.