Using random walks to generate associations between objects

Muhammed A. Yildirim[1,*] Michele Coscia[1]

1 Center for International Development, Harvard University, 79 JFK St, Cambridge MA, US 02138

∗ E-mail: Corresponding muhammed_yildirim@hks.harvard.edu

Abstract

Measuring similarities between objects based on their attributes has been an important problem in many disciplines. Object-attribute associations can be depicted as links on a bipartite graph. A similarity measure can be thought as a unipartite projection of this bipartite graph. The most widely used bipartite projection techniques make assumptions that are not often fulfilled in real life systems, or have the focus on the bipartite connections more than on the unipartite connections. Here, we define a new similarity measure that utilizes a practical procedure to extract unipartite graphs without making *a priori* assumptions about underlying distributions. Our similarity measure captures the relatedness between two objects via the likelihood of a random walker passing through these nodes sequentially on the bipartite graph. An important aspect of the method is that it is robust to heterogeneous bipartite structures and it controls for the transitivity similarity, avoiding the creation of unrealistic homogeneous degree distributions in the resulting unipartite graphs. We test this method using real world examples and compare the obtained results with alternative similarity measures, by validating the actual and orthogonal relations between the entities.

Introduction

An object can be described by its attributes. Given a set of objects, it is often desirable to quantify the similarity between any two objects based on the attributes that they possess. A similarity measure is then used to predict the events in which these two objects behave similarly. For instance, one can ask whether two senators would vote concordantly given the similarity between their voting records. Or one can quantify the likelihood of a person switching occupations based on the task similarities between the occupations.

Here, we think of the object attribute associations as a bipartite graph of two types of nodes (i.e., objects and attributes), where a link is present (often with a weight) between an object and the attribute if the object possesses that attribute. Then, the object-object similarities can be modeled as a unipartite graph. Most of the recent interest in large-scale social, biological, and communication networks has been devoted to unipartite graphs [1, 2]. As a result, unipartite graphs are well understood in literature [3]. An impressive number of tools helps us extracting knowledge from such structures.

The methodology presented in this paper can be thought as a unipartite projection of an underlying bipartite graph. Many complex systems have an underlying bipartite representation: a scientist can be connected to papers that she wrote [4], an actor can be connected to a movie that he/she acted in [5], a country may be connected to the products it exports [6]; flavors can be connected to the food that they are tested in [7], human diseases can be connected to the genes that cause them [8] and many others (e.g., [9–11]). To exploit the richness of the methods developed for analysis of unipartite graphs in recent years, and, therefore, to gain an improved vantage point over the influence or interdependence of entities in bipartite structures, a unipartite projection becomes useful. For instance, projections of the bipartite graphs that we mentioned above has resulted in the scientific collaboration network [4], co-actorship network [5], the product space [6]; flavor network [7] or human disease network (diseasome) [8], respectively.

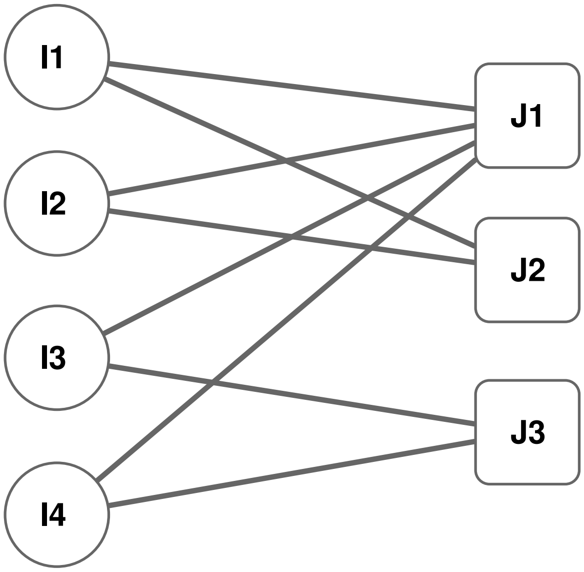

Figure 1. Toy example. This is a simple graph representation of a bipartite graph. Nodes in the class I connect exclusively with nodes in the class J with $I - J$ edges.

In network analysis, techniques aimed at uncovering the actual similarity value between entities in complex networks are popular. Some examples are the network back-boning technique to evaluate the significance of a link in weighted graphs [12]; or the graph deconvolution method to evaluate the direct connections between not directly connected entities [13]. Here, we are focusing on projecting bipartite graphs into unipartite entities. In the following, we refer to the graph projection as the construction of a unipartite graph map exploiting connections in a bipartite graph and allowing us to evaluate the similarity between the objects, for instance predicting which occupations are similar by looking at their common tasks.

Projection techniques make use of distance measures and/or counting common elements [4, 14, 15]. The projection criteria are very important as they affect the usefulness of the graph itself. Suppose we want to create a network map of nodes of class I from Figure 1. Each node of class J can be considered as a vectorial element. Then, the link strength in the bipartite graph would reflect the load of the class I node in that dimension. Using this vectorial representation, then a classical spatial distance can be calculated, such as Euclidean or Cosine distance measures, which have been extensively used in adaptive filtering [16]. We can also represent a I node as a set that contains all the J type nodes which it connects to. Then a set difference measure like the Jaccard measure can be calculated between all possible I node pairings. Issues arise with these approaches. Simpler distance measures like the Jaccard measure cannot cope with what is known as the *saturation effect*: an additional shared J node between two I nodes that share only one J node should count more than between two nodes who already share 100 J nodes [17]. Moreover, in a scale free bipartite graph, the degree of each node decays as $k^{-\alpha}$ in both sets of nodes I and J. Therefore, few hubs from set I connect with many nodes in set J, which on average have a low degree, as a consequence of the scaling of the power-law. If we project the structure to connect nodes from set J, the low degree nodes will connect one with the other, as their few connections cannot outweigh their common I hub connection. Moreover, the similarity is transitive: $i_1 \sim i_2$ and $i_2 \sim i_3$ implies $i_1 \sim i_3$. As a result, unipartite projections using these measures end up having a normal degree distribution, which is different than most real-world scenarios [18].

Some of these drawbacks do not affect other techniques. In [19] and subsequent papers by the same group [20,21], authors propose to overcome the saturation problem by using a resource allocation process. In practice, each node in I is considered to be a bearer of a certain quantity of resources, which it scatters equally to all its J neighbors. Then, each node in J will disseminate equally all the resources it gathered back to I nodes. Using this process, we can quantify the information originated form an I node and ended up in another I node. This amount is the degree of similarity of that node to the other I nodes. This approach, however, belittles crucial structural properties of the graph structure as a whole. The position of a node in the graph and the structural importance of the connections between I and J nodes influence their significance when projecting the graph, beyond what the simple degree can capture. In fact, the focus in [19] is to use bipartite graph projection as a tool for personal recommendation. In other words, the aim is not to predict an $I - I$ edge, but an $I - J$ edge. In this case, the structural properties of the graph as a whole are not important, as the focus is in the direct neighborhood of the node. For example, in a customer-product bipartite graph where customers connect to the products they buy, the method presented in [19] aims to understand which products a customer will likely buy next, given what other customers similar to her purchased.

We have a different aim, namely to predict $I - I$ edges, that is equivalent of building the unipartite graph. In such scenario, we cannot just rely on the immediate neighbors but take into account the overall structure of the complex network. For this reason, we propose an approach that is alternative to [19], with a complementary application. In this method, we let a random walker explore the bipartite structure. In doing so, we can overcome most of the problems of traditional similarity measures. Two nodes from set I are similar if they frequently appear as successive visiting sites of the random walker. Since hubs in J are connected to many nodes in I, their contribution to each node pair in I is low, as the probability of consistently choosing the same endpoints can be considered insignificant. In this way, the random walker gives us information taking into account the overall structural properties of the graph. Random walkers have been extensively used in literature with this precise aim. For example, they are at the basis of the modular organization detection of many community discovery algorithms [22,23]. Other applications include centrality measures, used to rank nodes according to their structural importance [24].

The numerical simulations indicate that this approach is able to predict $I - I$ edges with higher confidence, when the unipartite graph maps extracted from the bipartite graphs are tested against the real world knowledge about the $I - I$ connections. This happens in four different realms, including occupation flows, industry employee flows, political activities in the US Congress and a citation graph between international aid agencies. We also tested our method in ranking $I - J$ edges and it functions equally well as the other methods.

Methods

The proposed approach consists of projecting the bipartite graph into a unipartite graph by creating a weighted edge between two nodes in the unipartite graph from the information present in the bipartite graph. The weight is directly proportional to how often one would observe a random walker on the bipartite graph visiting the two nodes consequently. Formally, let us assume that in the bipartite graph there are two types nodes indexed with i and j, respectively. Assume that there are n (m) i (j) type nodes that form the set $I = \{i_1, ..., i_n\}$ $(J = \{j_1, ..., j_m\})$. An edge in the bipartite graph is defined as a link between an i type and j type node. We can define the $n \times m$ adjacency matrix, B, whose $(i, j)^{th}$ entry represents the strength of the links between node i and node j. In the binary case, $B_{i,j}$ will be 1 if there is an edge between i and j, and 0 otherwise. In general, a bipartite graph can be represented by the triplet $\{I, J, B\}$. The unipartite projection of this bipartite graph onto I domain requires defining an $n \times n$ edge matrix, U, from the bipartite edge matrix B.

Here, we propose to build the U matrix as the number of times a random walker (RW) passes through a pair of I type nodes on the bipartite graph, separated by a single J type node. Suppose the RW is on the node i. Then, the RW would end up to any j type node with probability:

$$P\left(i\rightarrow j\right)=\frac{B_{i,j}}{\sum_{j'\in J}B_{i,j'}} \tag{1}$$

Once on a j type node, the probability that the RW goes to node the i' is:

$$P\left(j\rightarrow i'\right)=\frac{B_{i',j}}{\sum_{i''\in I}B_{i'',j}} \tag{2}$$

Therefore, the probability of moving between nodes i and i' will be the sum of all paths $i\rightarrow j\rightarrow i'$ that pass through $\forall j\in J$:

$$\begin{aligned} P\left(i\rightarrow i'\right) &= \sum_{j}P\left(i\rightarrow j\rightarrow i'\right) \\ &= \sum_{j}P\left(i\rightarrow j\right)P\left(j\rightarrow i'\right) \\ &= \sum_{j}\frac{B_{i,j}}{\sum_{j'\in J}B_{i,j'}}\frac{B_{i',j}}{\sum_{i''\in I}B_{i'',j}} \end{aligned} \tag{3}$$

We can rewrite the transition probabilities in terms of a Markov transition matrix T, such that $T_{i,i'}=P\left(i\rightarrow i'\right)$. The frequency of observing the path $i\rightarrow i'$ also depends on how often the RW visits node i in general. Suppose that $\vec{P}_n$ denotes the probability vector whose ith element is the probability of the RW being on the node i in the nth step of the random walk. We initialize the process with $\vec{P}_0=\frac{1}{|I|}\vec{1}$ where $\vec{1}$ denotes a row vector of ones. Therefore:

$$\vec{P}_n=\frac{1}{|I|}\vec{1}T^{n} \tag{4}$$

Since T is a right-stochastic matrix (i.e., its elements are non-negative and sum of its rows is always 1), the stationary distribution, $\vec{\pi}$ will satisfy:

$$\vec{\pi}T=\vec{\pi} \tag{5}$$

Here, we will assume that the transition matrix is irreducible (i.e., every node can communicate with each other in finite step) and aperiodic (i.e., there is no $\vec{x}$ and integer $m>1$ such that $\vec{x}T^{m}=\vec{x}$ but $\vec{x}\neq T\vec{x}$). If any of these properties are violated, then we will not be able to ensure a unique stationary distribution. In our case, we would ensure that the bipartite graph is connected, which would satisfy the irreducibility property. Moreover, we only work with bipartite graphs with non-directed edges which justifies the aperiodicity property. With these properties at hand, the Perron-Frobenius theorem guarantees the existence of unique stationary distribution, which is the left eigenvector of T matrix with eigenvalue 1.

Given that we calculated the stationary distribution $\vec{\pi}$, then as the RW moves infinitely many times, the random-walk similarity between nodes i and i' is:

$$U_{i,i'}=\pi_{i}T_{i,i'} \tag{6}$$

We would like to remark that Zhou et al. [19] defines a similar metric based on their ProbS methodology but they they do not include π_i in their definition. The π_i element is the one that contributes information about the overall graph structure. It allows the similarity to consider not only the immediate neighbors of a node, but also its position in the graph, enabling T to avoid the saturation and transitivity issues described in the Introduction.

Other Projection Techniques

In this section we provide our implementation of the methods we compared our technique with. In each technique, the entities of the bipartite graph N are considered as binary vectors. Suppose that we have a bipartite graph with two classes of entities $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2, b_3, b_4, b_5\}$. Suppose that a_1 is connected to b_1, b_4 and b_5. Then $a_1 = \{1, 0, 0, 1, 1\}$. In the following discussion, we adopt the convention of always projecting onto nodes of class A.

ProbS

This is the bipartite projecting technique presented in [19]. The assumption at the basis of this measure is the same we implemented, namely that the relatedness of a_1 and a_2 depend on the resource flow from a_1 and a_2 to the B nodes and back. Instead of implementing this idea with random walks, Zhou et al. decided to pass the entire resource equally to all B nodes and back.

So, in the first step, all the resource flows from A to B as:

$$f(b_l) = \sum_{i=1}^{|A|} \frac{N_{il} \times f(a_i)}{k(a_i)},$$

where $k(a_i)$ is the degree of a_i and N is the $|A| \times |B|$ adjacency matrix representing the bipartite graph, containing 1 if a_i is connected to b_l, 0 otherwise. In the next step, all the resource flows back to A, and the final resource located on a_i reads:

$$f'(a_i) = \sum_{l=1}^{|B|} \frac{N_{il} \times f(b_l)}{k(b_l)} = \sum_{l=1}^{|B|} \frac{N_{il}}{k(b_l)} \sum_{j=1}^{|A|} \frac{N_{jl} \times f(a_j)}{k(a_j)}.$$

This can be rewritten as:

$$f'(a_i) = \sum_{j=1}^{|A|} s(a_i, a_j) f(a_j),$$

where:

$$s(a_i, a_j) = \frac{1}{k(a_j)} \sum_{l=1}^{|B|} \frac{N_{il} N_{jl}}{k(b_l)}, \tag{7}$$

which sums the contribution from all two-step paths between a_i and a_j, and it is ultimately the similarity between the two nodes.

Using a standard example that will be adopt from now on, we assume that $a_1 = \{b_1, b_4, b_5\}$ and $a_2 = \{b_1, b_3\}$, and all B nodes do not have any other connection with any other A node. Then, $s(a_1, a_2) = 1/4$.

HeatS

Heats method, introduced by Zhou et al. in [20], is related to the ProbS method but instead of normalizing by column, it is normalized by the row. Mathematically,

$$s(a_i, a_j) = \frac{1}{k(a_i)} \sum_{l=1}^{|B|} \frac{N_{il} N_{jl}}{k(b_l)}, \tag{8}$$

The difference between HeatS (Eq. 8) and ProbS similarity measures (Eq. 7) is the first fraction. For the example introduced above, HeatS similarity would be 1/6, lower than ProbS similarity of 1/4.

Hybrid

The Hybrid methodology, introduced in [21], hybridized ProbS and HeatS, by taking a geometric average of the first normalizing parameters. The similarity in this measure is defined as:

$$s(a_i, a_j) = \frac{1}{k(a_i)^{1-\lambda} k(a_j)^{\lambda}} \sum_{l=1}^{|B|} \frac{N_{il} N_{jl}}{k(b_l)}, \tag{9}$$

Assuming $\lambda = 1/2$, the similarity will be $1/2\sqrt{6}$, a value between ProbS and HeatS similarities.

Jaccard

In this bipartite projecting technique, each class A node is seen as a set of elements. So, if $a_1 = \{1, 0, 0, 1, 1\}$, then we consider it equivalent to $a_1 = \{b_1, b_4, b_5\}$. Then, the similarity between two nodes a_1 and a_2 is equivalent to the Jaccard similarity:

$$s(a_1, a_2) = \frac{|a_1 \cap a_2|}{|a_1 \cup a_2|}.$$

For instance, if $a_2 = \{b_1, b_3, b_5\}$, then $s(a_1, a_2) = 2/4$.

Cosine

This bipartite projecting technique is based on the Cosine similarity. The Cosine distance between two vectors of same length is defined as:

$$s(a_1, a_2) = \frac{a_1 \cdot a_2}{\|a_1\| \|a_2\|} = \frac{\sum\limits_{b \in B} a_1(b) \times a_2(b)}{\sqrt{\sum\limits_{b \in B} (a_1(b))^2} \times \sqrt{\sum\limits_{b \in B} (a_2(b))^2}}.$$

We recall that a_1 and a_2 are both binary vectors. For each $b \in B$ where either (or both) nodes are not attached, the overall contribution to the sum is 0. Only when both $a_1(b)$ and $a_2(b)$ are equal to 1 there is a contribution of 1 to the sum.

Again considering our standard example $a_1 = \{1, 0, 0, 1, 1\}$ and $a_2 = \{1, 0, 1, 0, 1\}$, we obtain $s(a_1, a_2) = 2/3$.

Euclidean

The Euclidean projecting technique takes advantage of the concept of Euclidean distance. The a_1 and a_2 vectors are seen as points in a $|B|$-dimensional space. We then calculate the Euclidean distance between points a_1 and a_2 as follows:

$$d(a_1, a_2) = \sqrt{\sum_{b \in B} (a_1(b) - a_2(b))^2}.$$

The Euclidean similarity is inversely proportional to the Euclidean distance, thus $s(a_1, a_2) = 1/d(a_1, a_2)$. Euclidean similarity gives more weight not only to the co-presence of 1s in a_1 and a_2, but also in co-presence of 0s.

Keeping fixed $a_1 = \{1, 0, 0, 1, 1\}$ and $a_2 = \{1, 0, 1, 0, 1\}$, we obtain $s(a_1, a_2) = 1/\sqrt{2}$.

Pearson

This is the bipartite projecting technique based on the well-known Pearson correlation coefficient. We calculate the correlation coefficient of a_1 and a_2 vectors as follows:

$$s(a_1, a_2) = \frac{\text{cov}(a_1, a_2)}{\sigma_{a_1} \sigma_{a_2}},$$

where $\text{cov}(a_1, a_2)$ is the covariance of a_1 and a_2, and σ_{a_1} and σ_{a_2} are the variance of the a_1 and a_2 vectors, respectively. Just like in the Euclidean case, also the Pearson similarity gives some weight to the co-absence of a connection, not only a co-presence.

We calculate the Pearson similarity for our standard example in which $a_1 = \{1, 0, 0, 1, 1\}$ and $a_2 = \{1, 0, 1, 0, 1\}$, obtaining as result $s(a_1, a_2) = .05/(.3 \times .3)$.

Results

We create two different sets for our experiments. In the first one, we compare the performances of all the methods using the experiments described in [19], on the very same dataset extracted from MovieLens and processed as described there. We then refer to [19] for the details of this experiment. Aim of the first experiment is to test the efficiency of different methods in ranking $I - J$ edges. In a second set of experiments, we study the projection of four real-world bipartite graphs. In this case, we also have unipartite graphs with observed relations between I entities. Aim of this experiment is to show that the $I - I$ edges, as ranked by the proposed methodology, are closer to the observed relations than any other methodology.

In both experiments, we compare the obtained results with the seven alternative projecting techniques presented in the previous section. Four of them are based on distance measures: Jaccard, Cosine, Euclidean and Pearson. The other three alternatives are ProBs [19], HeatS [25] and Hybrid [20][1]. We refer to the proposed method as Bipartite Projection via Random-walk, or “BPR”.

I-J Edges

In this numerical simulation, we have user-to-movie connections if a user (I) liked the movie (J) and the aim is to suggest other movies to the user (a $I - J$ edge). To test the efficiency of the methods, a random subset of connections are removed from the original bipartite graph. Then we calculate movie to movie similarity in the remaining graph using the measures presented above. Finally, for each user i and movie j we average (for Cosine, Euclidean, Jaccard and Pearson) or sum (for ProbS, HeatS, Hybrid and BPR) the movie similarities to j of all movies which are liked by i. At the end of the procedure, for each user i we have a list of J nodes, sorted by the computed value. We calculate the quality of this suggestion list in two ways. First, given a user i and a movie j that was removed from the graph, $r_{i,j}$ is equal to the rank of j in i's suggested list over the length of the list. Second, we shorten the suggested list to different lengths (including 10, 20 and 50 elements) and we record the share of the randomly removed movies that are included in the list - we refer to this measure as Hit Rate (HR-X, where X is the length of the recommendation list). Hybrid method also includes a parameter of choice (λ in Equation 9). We selected $\lambda = 0.2$, which maximized the predictive power.

The results of this numerical simulation are reported in Figures 2 and 3, and Table 1. In Figure 2 we report the cumulative value of r as the recommendation lists grows. A lower value here indicates a better prediction method. In Figure 2, ProbS, Hybrid, BPR and HeatS appear to easily outperform all other methods, with Hybrid performing the best. BPR performed better than HeatS but was slightly

[1] We could not confront with the latest Preferential Diffusion version of the algorithm [21] as we were unable to obtain the code.

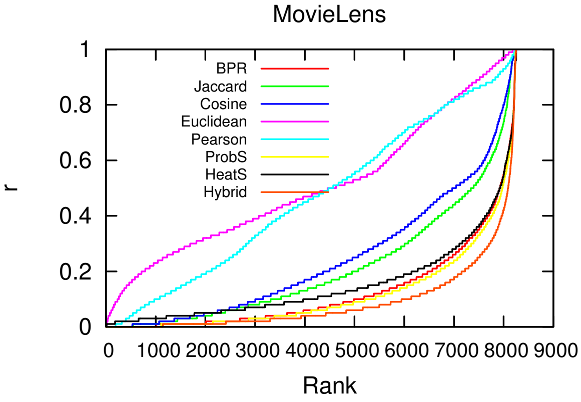

Figure 2. The predicted position of each entry in the probe ranked in the ascending order. On the y-axis, r measures the position of an $I - J$ edge $(i - j)$ in the ordered result of the prediction. For example, if there are 1500 uncollected J connections for i, and $i - j$ is the 30th strongest prediction, we say the position of (i, j) is the top 30/1500, denoted by $r = 0.02$. Lower r values indicate better predictions.

Distance	$< r >$	HR-10	HR-20	HR-50
BPR	0.12336	14.65%	23.02%	38.68%
ProbS	0.11417	15.97%	24.57%	40.99%
Jaccard	0.21086	7.97%	12.30%	22.05%
Cosine	0.24151	7.87%	12.16%	20.35%
Euclidean	0.50581	0.06%	0.16%	0.53%
Pearson	0.46280	2.62%	3.59%	5.68%
HeatS	0.15221	0.21%	1.55%	13.28%
Hybrid	0.08814	17.94%	27.61%	46.08%

Table 1. Average predicted position ($< r >$) and hit rate for recommendation lists of length 10 (HR-10), 20 (HR-20) and 50 (HR-50).

worse than ProbS. In the first column of Table 1 we report the overall average value of r[2]. The ranking of the methodologies remains the same.

In Figure 3 we report the hit rate at different lengths of the recommendation list. Again, the result is confirmed: Hybrid, ProbS and BPR outperform in the task with respective order. The hit rates at different list length are reported in the HR-X columns of Table 1. All these results confirm that BPR works in $I - J$ but there are more efficient methodologies, namely Hybrid and ProbS in this task.

I-I Edges

For the task of predicting $I - I$ edges we consider four different bipartite graphs:

(i) Occupations connected to the tasks they fulfil, from the O-Net database [26, 27] (referred here as "O-Net").

[2] Please note that our numerical simulation results for our implementation of the method presented by Zhou et al. are different from the ones included in [19]. This happens because of a mistake of the authors of [19]: in the paper, they report that the bipartite graph MovieLens contains 85,250 edges, while it actually contains 82,520 edges.

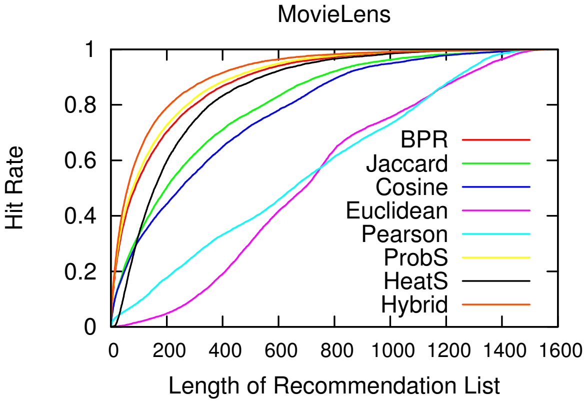

Figure 3. The hitting rate as a function of the length of recommendation list. We shorten the recommendation list to increasing values, in the x-axis, and we report the share of actual $I-J$ edges predicted, on the y-axis. The higher the hit rate, the better the prediction.

(ii) Industries connected to the fields of educations of the people they employ, from the IPUMS dataset [28] (referred here as "IPUMS").

(iii) International aid organizations connected to the countries and the development issues they talk about in their websites [29] (referred here as "Aid").

(iv) Congressmen from the 111th US Congress, connected to the topics they wrote a bill on (referred here as "Congress").

For additional information about how we built the bipartite and the unipartite graphs used, see Supplementary Material.

For each of this bipartite graph we have a corresponding unipartite graph that we use to evaluate the goodness of the projection. The test graphs are:

(i) For O-Net dataset, the occupation-occupation job flows.

(ii) For IPUMS dataset, the job flows across industries.

(iii) For Aid dataset, the mentions of other aid organizations in an organization's website.

(iv) For Congress dataset, the co-sponsorship of bills.

The procedure is the same presented in the previous section: for each pair of I nodes we calculate the similarity using one of the proposed measures. For each node i, we obtain a ordered list of similarities. We use this list to predict actual $I-I$ edges, observed in the corresponding unipartite graphs. In Figure 4 and Table 2 we report the performance in the prediction task for all methods and for all graphs. Figure 4 presents the receiver operating characteristic (ROC) curves of the various methods. We can see that BPR comes as winner or a close second in most cases. Table 2 reports the area under the ROC curve, that summarizes the overall quality of the predictions shown Figure 4. Table 2 confirms that BPR is the best predictor of the $I-I$ edges, based on the observed $I-J$ edges in the test graphs, with the exception of the O-Net graph. However, in that case, BPR is beaten by Pearson, which scores poorly in the other scenarios. The second best predictor is different for each graph, while BPR's performance across all graphs is constantly on top. Since we are dealing with the weighted graphs, we need a threshold, δ, to

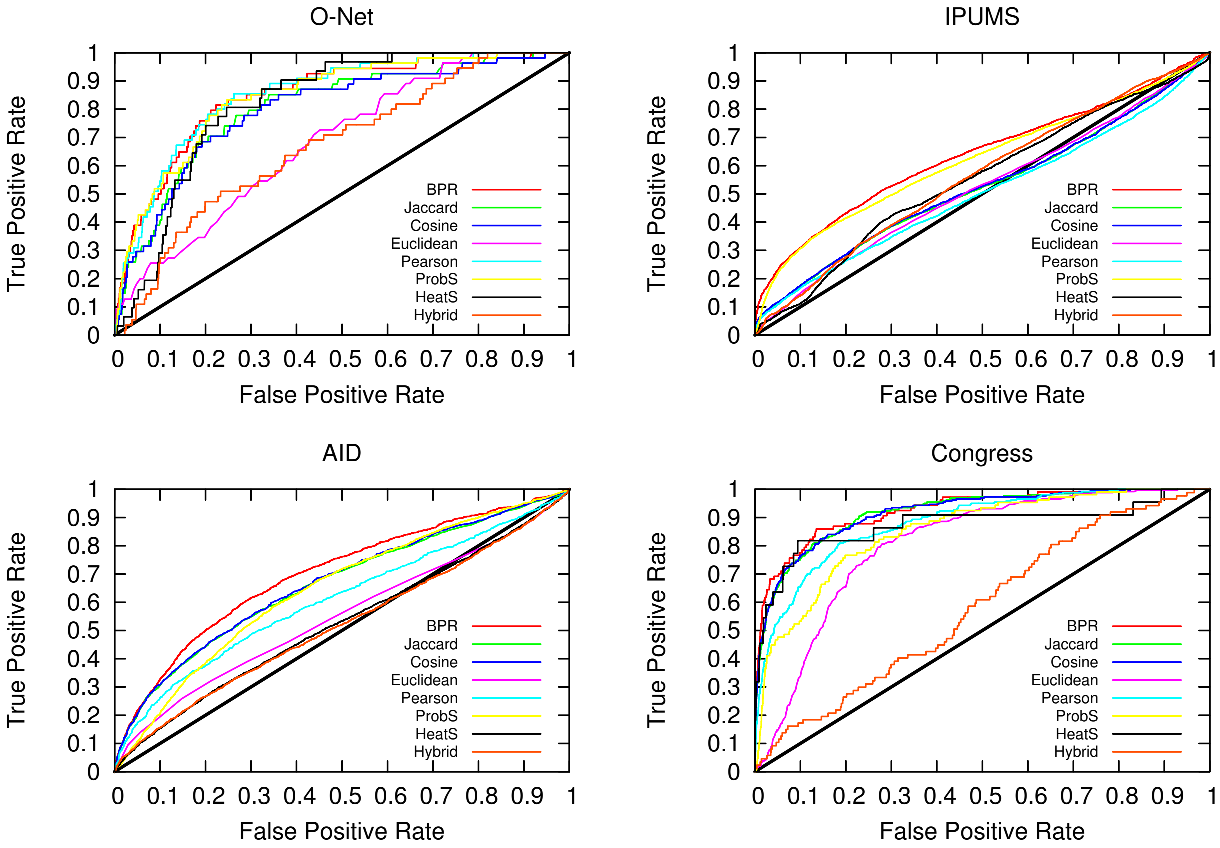

Figure 4. ROC Curves. The receiver operating characteristic (ROC) curves for the four datasets in our experimental set up: O-Net (top left), IPUMS (top right), Aid (bottom left) and Congress (bottom right). Each predicted $I - I$ edge is sorted according to the prediction confidence and it is tested against the observed real-world graph.

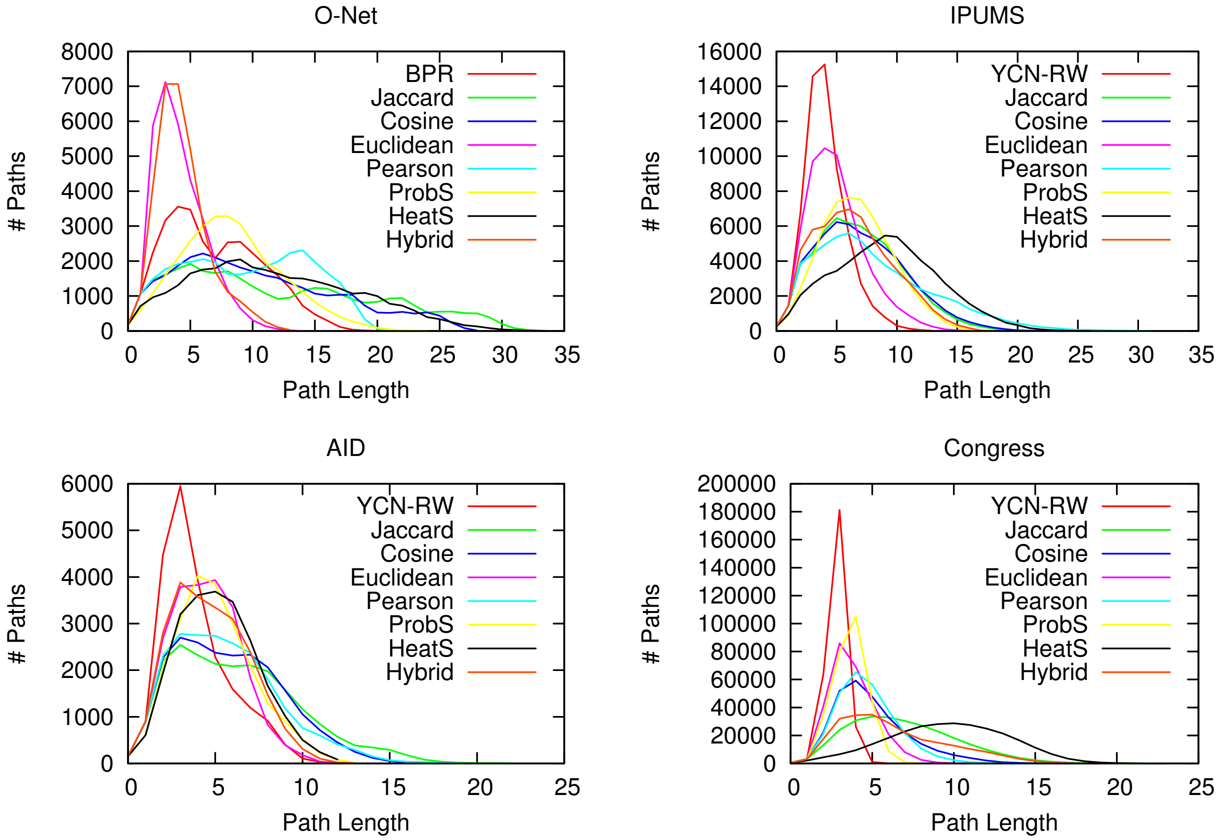

Figure 5. Path distributions. The distribution of the shortest paths in the unipartite graphs generated with each technique, for all datasets: O-Net (top left), IPUMS (top right), Aid (bottom left) and Congress (bottom right). We count the number of paths (y-axis) with a given length in number of edges (x-axis).

Distance	O-Net	IPUMS	Aid	Congress
BPR	0.84627	**0.62831**	**0.69724**	**0.93063**
ProbS	0.84528	0.61966	0.64224	0.90438
Jaccard	0.80507	0.52031	0.66255	0.91908
Cosine	0.79768	0.52101	0.65484	0.91806
Euclidean	0.68452	0.50441	0.53616	0.80787
Pearson	**0.85186**	0.50380	0.65881	0.85145
HeatS	0.78843	0.53862	0.52121	0.88540
Hybrid	0.67216	0.56061	0.51862	0.57986

Table 2. AUC Values. The AUC is the integral of the area below the ROC curve, as shown in Figure 4. If we obtain an AUC equal to .5, then the prediction is said to have a performance equivalent to a random predictor.

determine when an observed weight is significant and when it is not. δ influences prediction scores, but not the performance ranking of the methods (see Supplementary Material and Supplementary Figure 1).

Prediction quality is not the only quality criterion to evaluate the unipartite projections. We also want the unipartite graph map to have topological properties comparable to the real-world complex graphs in the literature. One of such properties is the small-world property [30]: the distribution of shortest paths are normally distributed around a mean much lower than the random Erdös-Renyi graphs, usually $\sim \log n$ where n is the number of nodes in the graph [3]. Figure 5 shows the distribution of the shortest path lengths in different bipartite graph projections. Each graph map has been generated by extracting the maximum spanning tree from all the $I - I$ edge similarities returned by each method, and then adding edges until the average degree reaches 3. We can see that BPR is the only method which constantly generates unipartite graphs with the expected distribution of shortest path lengths. With the exception of the Euclidean method in the O-Net dataset, the other methods have usually either higher averages or distributions more skewed on higher values, or both.

Another property of real-world graphs is a broad degree distribution. Real-world graphs are characterized by few hubs with high degree and many nodes with degree equal to one [18]. However, transitive similarity measures may be prone to boost transitivity beyond what is reasonable, creating large cliques and inflating the degree of most nodes. Therefore, for a similarity measure, higher skewed distribution is a desirable property because it is a signal of the absence of large cliques, that lowers the practicality of the network map. We depict the cumulative degree distributions of the graph projections in Figure 6. We can see that BPR has very broad degree distributions, clearly the broadest in the Aid graph and the broadest in Congress and O-Net after the Euclidean graph. However, we saw that for practical purposes the only contestant was ProbS (Figure 4 and Table 2). Here, ProbS is affected exactly by the problems of very homogeneous degree distribution: in all graphs the nodes with degree lower than 3 are very few (always less than 10%), while the most connected nodes have half or a third the degree they have in BPR.

Discussion

The proposed bipartite projection technique gives differential weights to elements by their commonality. The methodology generates an edge in the graph map whenever the random walker frequently visit the two nodes in the same path, traversing their common elements, which ensures that hubs do not artificially drive up the similarity measure. As a result, the random walk similarity allows the creation of significant and meaningful graph maps, who are structurally very similar to the corresponding real-world graphs. Consequently, the resulting graph projections carry some fundamental properties that are observed in

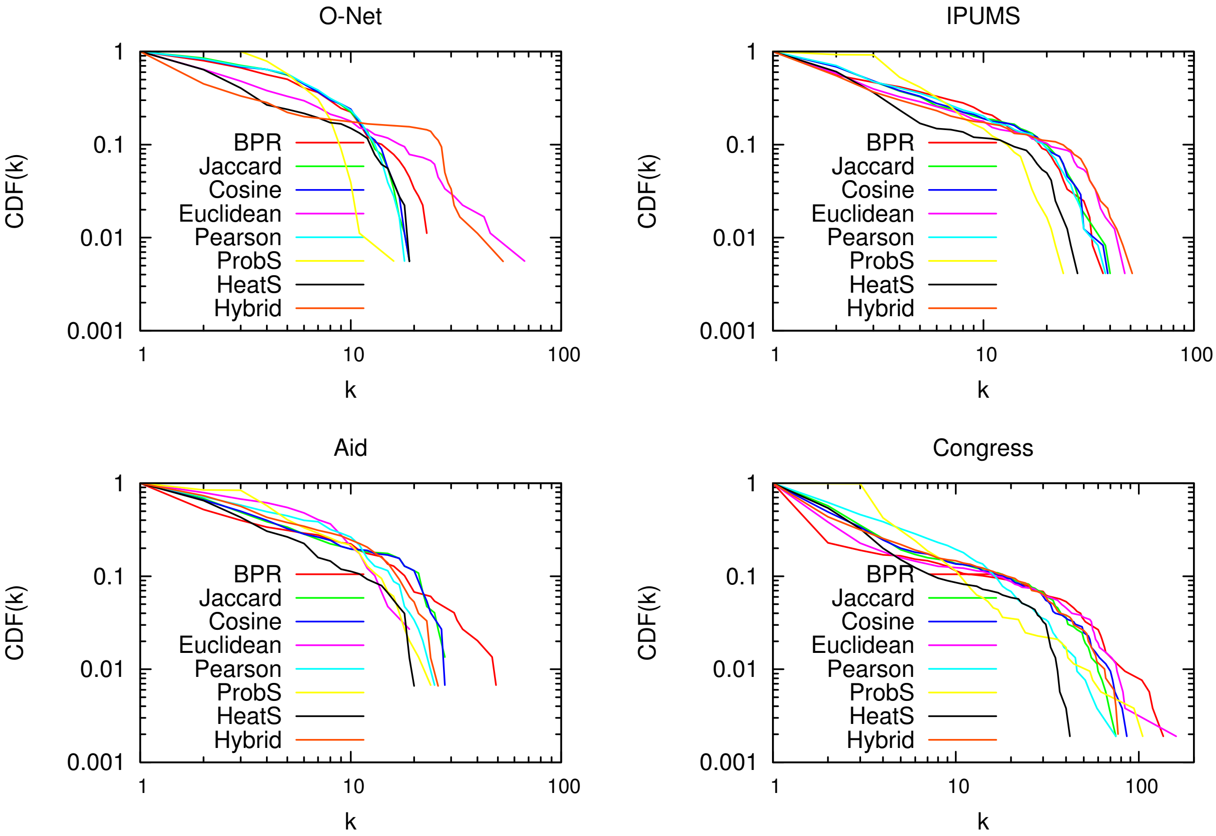

Figure 6. Degree distributions. The cumulative degree distributions of the unipartite graphs generated with each technique, for all datasets: O-Net (top left), IPUMS (top right), Aid (bottom left) and Congress (bottom right). We calculate the probability (y-axis) for a node to have a degree equal to or higher than a given degree (x-axis).

many other naturally occurring graphs.

As a criticism, one could say that it only works in the case of bipartite graphs that exhibits non-overlapping scale free degree distributions, where there are hubs in one or all classes of nodes. In any case, any projecting technique has limitations, and the choice between one algorithm over another has to be made considering the objective of the exercise. We do not exclude the existence of a scenario in which our methodology will not yield significant results. Yet, it has been proved that broad (scale free or exponential) degree distributions are ubiquitous in real world graphs: from social graphs to scientific co-authorship, from the physical Internet infrastructure to the virtual hyperlinks in the World Wide Web, from financial graphs to protein interactions. Therefore, we conclude that our methodology may be applied in this wide range of scenarios.

Acknowledgements

The authors thank Frank Neffke for useful discussions and comments. This material is based upon work supported by the National Science Foundation under Grant No. 1216028. The analysis was performed on an anonymized version of already existing data, in accordance with the Terms of Services of the data sources, for those who involve data about humans (O-Net http://www.onetonline.org/, IPUMS https://www.ipums.org/, and US Census https://www.census.gov/cps/). GovTrack data was not anonymized, but it is publicly available due to the transparency policy of the US Congress. Because the study was performed on already existing and anonymized data, no IRB review was requested.

References

1. Liu YY, Slotine JJ, Barabási AL (2011) Controllability of complex networks. Nature 473: 167–173.

2. Balcan D, Colizza V, Gonçalves B, Hu H, Ramasco JJ, et al. (2009) Multiscale mobility networks and the spatial spreading of infectious diseases. Proceedings of the National Academy of Sciences 106: 21484–21489.

3. Newman ME (2003) The structure and function of complex networks. SIAM review 45: 167–256.

4. Newman ME (2004) Coauthorship networks and patterns of scientific collaboration. Proceedings of the National academy of Sciences of the United States of America 101: 5200–5205.

5. Newman ME, Strogatz SH, Watts DJ (2001) Random graphs with arbitrary degree distributions and their applications. Physical Review E 64: 026118.

6. Hidalgo CA, Klinger B, Barabási AL, Hausmann R (2007) The product space conditions the development of nations. Science 317: 482–487.

7. Ahn YY, Ahnert SE, Bagrow JP, Barabási AL (2011) Flavor network and the principles of food pairing. Scientific reports 1.

8. Goh KI, Cusick ME, Valle D, Childs B, Vidal M, et al. (2007) The human disease detwork. Proceedings of the National Academy of Sciences 104: 8685–8690.

9. Hausmann R, Hidalgo C, Bustos S, Coscia M, Simoes A, et al. (2013) The Atlas of Economic Complexity. MIT Press.

10. Ghoshal G, Zlatić V, Caldarelli G, Newman M (2009) Random hypergraphs and their applications. Physical Review E 79: 066118.

11. Lambiotte R, Ausloos M (2006) Collaborative tagging as a tripartite network. In: Computational Science–ICCS 2006, Springer. pp. 1114–1117.

12. Serrano MÁ, Boguñá M, Vespignani A (2009) Extracting the multiscale backbone of complex weighted networks. Proceedings of the National Academy of Sciences 106: 6483–6488.

13. Feizi S, Marbach D, Médard M, Kellis M (2013) Network deconvolution as a general method to distinguish direct dependencies in networks. Nature Biotechnology .

14. Ramasco JJ, Morris SA (2006) Social inertia in collaboration networks. Physical Review E 73: 016122.

15. Newman ME (2001) Scientific collaboration networks. i. network construction and fundamental results. Physical review E 64: 016131.

16. Zhang Y, Callan J, Minka T (2002) Novelty and redundancy detection in adaptive filtering. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 81–88.

17. Li M, Fan Y, Chen J, Gao L, Di Z, et al. (2005) Weighted networks of scientific communication: the measurement and topological role of weight. Physica A: Statistical Mechanics and its Applications 350: 643–656.

18. Clauset A, Shalizi CR, Newman ME (2009) Power-law distributions in empirical data. SIAM review 51: 661–703.

19. Zhou T, Ren J, Medo M, Zhang YC (2007) Bipartite network projection and personal recommendation. Physical Review E 76: 046115.

20. Zhou T, Kuscsik Z, Liu JG, Medo M, Wakeling JR, et al. (2010) Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences 107: 4511–4515.

21. Lü L, Liu W (2011) Information filtering via preferential diffusion. Physical Review E 83: 066119.

22. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences 105: 1118.

23. Pons P, Latapy M (2005) Computing communities in large networks using random walks. In: Computer and Information Sciences-ISCIS 2005, Springer. pp. 284–293.

24. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: bringing order to the web. Technical Report .

25. Zhou T, Jiang LL, Su RQ, Zhang YC (2008) Effect of initial configuration on network-based recommendation. EPL (Europhysics Letters) 81: 58004.

26. Center OR (2013). About O*NET. `http://www.onetcenter.org/overview.html`. Accessed: 2013-08-17.

27. Center OR (2013). O*NET version 17.0. `http://www.onetcenter.org/download/database?d=db_17_0.zip`. Accessed: 2013-08-17.

28. Ruggles S, Alexander JT, Genadek K, Goeken R, Schroeder MB, et al. (2010) Integrated public use microdata series, ver. 5.0. Minneapolis: University of Minnesota .

29. Coscia M, Hausmann R, Hidalgo CA (2013) The structure and dynamics of international development assistance. Journal of Globalization and Development : 1–42.

30. Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. Nature 393: 440–442.