Data & Knowledge Engineering xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Data & Knowledge Engineering

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

From biological to social networks: Link prediction based on multi-way spectral clustering

Panagiotis Symeonidis *, Nantia Iakovidou, Nicolaos Mantas, Yannis Manolopoulos

Aristotle University, Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 25 February 2012
Received in revised form 16 May 2013
Accepted 16 May 2013
Available online xxxx

Keywords:
Cross-disciplinary applications
Information filtering
Clustering
Bioinformatics databases

ABSTRACT

Link prediction in protein–protein interaction networks (PPINs) is an important task in biology, since the vast majority of biological functions involve such protein interactions. Link prediction is also important for online social networks (OSNs), which provide predictions about who is a friend of whom. Many link prediction methods for PPINs/OSNs are local-based and do not exploit all network structure, which limits prediction accuracy. On the other hand, there are global approaches to detect the overall path structure in a network, being computationally prohibitive for huge-size PPINs/OSNs. In this paper, we enhance a previously proposed *multi-way spectral clustering* method by introducing new ways to capture node proximity in both PPINs/OSNs. Our new enhanced method uses information obtained from the top few eigenvectors of the normalized Laplacian matrix. As a result, it produces a less noisy matrix, which is smaller and more compact than the original one. In this way, we are able to provide faster and more accurate link predictions. Moreover, our new spectral clustering model is based on the well-known Bray–Curtis coefficient to measure proximity between two nodes. Compared to traditional clustering algorithms, such as k-means and DBSCAN, which assume globular (convex) regions in Euclidean space, our approach is more flexible in capturing the non-connected components of a social graph and a wider range of cluster geometries. We perform an extensive experimental comparison of the proposed method against existing link prediction algorithms and *k*-means algorithm, using two synthetic data sets, three real social networks and three real human protein data sets. Our experimental results show that our SpectralLink algorithm outperforms the local approaches, the *k*-means algorithm and another spectral clustering method in terms of effectiveness, whereas it is more efficient than the global approaches.

1. Introduction

Online social networks (OSNs) such as Facebook.com,[1] Myspace,[2] Hi5.com,[3] etc. contain gigabytes of data that can be mined to make predictions about who is a friend of whom. OSNs gather information on users' social contacts, construct a large interconnected social network, and recommend other people to users based on the network structure. *Link Prediction* in social networks, tries to infer new interactions among members of a social network that are likely to occur in the near future. There are two main approaches [24] that handle it. The first approach is based on local features of a network, focusing mainly on the nodes structure; the second approach is based on global features, detecting the overall path structure in a network. For instance, as an

* Corresponding author.
E-mail addresses: symeon@csd.auth.gr (P. Symeonidis), niakovid@csd.auth.gr (N. Iakovidou), nmantas@csd.auth.gr (N. Mantas), manolopo@csd.auth.gr (Y. Manolopoulos).

[1] http://www.facebook.com.
[2] http://www.myspace.com.
[3] http://www.hi5.com.

0169-023X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.datak.2013.05.008

example of a local approach, as shown in Fig. 1, Facebook.com or Hi5.com uses the following style of recommendation for recommending new friends to a target user U_1: "People you may know: (i) user U_7 because you have three common friends (users U_5, U_6, and U_{10}) (ii) user U_4 because you have two common friends (users U_2 and U_3) (iii) user U_9 because you have one common friend (user U_8) …". The list of recommended friends is ranked based on the number of common friends each candidate friend has with the target user.

Moreover, inspired from the recent surge of research on large, complex networks and their properties, we also study protein–protein interaction networks (PPINs) — structures whose nodes represent proteins and whose edges represent interaction, or influence between them. Interactions between proteins are important for numerous – if not all – biological functions. Given a natural example from the area of biology, signals from the exterior of a cell are mediated to the inside of that cell by protein–protein interactions of the signaling molecules. This process, called signal transduction, plays a fundamental role in many biological processes and in many diseases (e.g. cancers). Thus, we study a basic computational problem underlying protein networks, the link prediction problem, i.e. given a part of a protein network we seek to accurately predict the rest of the network's edges, by performing multiway spectral clustering analysis.

In this paper, we provide link predictions in both OSNs and PPINs, by performing *multi-way spectral clustering*, which uses information obtained from the top few eigenvectors and eigenvalues of the normalized Laplacian matrix and computes a multi-way partition of the data.

Compared to approaches based on local features of a network (i.e. Common Neighbors index or else known as FOAF algorithm, Adamic/Adar index, Jaccard Coefficient, etc. — for more details see Related work section), we provide link predictions, by exploiting the normalized Laplacian matrix of the graph, which captures the overall network structure. Instead, local approaches consider only pathways of maximum length 2 between a target user/protein and his candidate friends/interacting proteins, which results to lower accuracy prediction as will be shown experimentally later.

Compared to global approaches (i.e. Katz status index, RWR algorithm, SimRank algorithm etc.), which also operate on the overall structure of a network (i.e. initial adjacency matrix), our method is more efficient. The reason is that, our method is based on the top few eigenvectors and eigenvalues of the normalized Laplacian matrix, requiring less time and space complexity than the global algorithms, as will be shown in Section 4.4. Solving a standard eigenvalue problem for all eigenvectors takes $O(n^3)$ operations, where n is the number of nodes in a graph. This becomes impractical for applications with n on the order of millions. However, real social and protein–protein interaction networks have often the following properties [38]: 1) The graphs are often only locally connected and the resulting eigensystem is very sparse, and 2) only the top few eigenvectors are needed for graph partitioning. These special properties of our problem can be fully exploited by an eigensolver called the Lanczos method [11], resulting to faster time complexity than global algorithms.

Compared to traditional clustering algorithms, such as k-means and DBSCAN, which make explicit or implicit assumptions that clusters form globular (convex) regions in Euclidean space, the normalized Laplacian matrix has some desirable properties that make it more suitable for real OSNs and PPINs, which often have non-connected components with non-globular shapes. Firstly, it is positive semi-definite with k non-negative real-valued eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_k$. The number of 0 eigenvalues equals the number of the connected components in a graph. Thus, spectral clustering is more flexible than k-means, in capturing (i) the non-connected components of a graph, and (ii) a wider range of cluster geometries and shapes [47].

The contributions of our approach are summarized as follows: (i) For the first time spectral clustering has been used for providing link prediction in both OSNs/PPINs. (ii) We provide more accurate friend recommendations and protein link predictions than local approaches and k-means, by detecting a wider range of network structure and cluster geometries. This reveals the latent associations between users/proteins of OSNs and PPINs respectively, as will be shown experimentally later. (iii) We provide higher efficiency than the global approaches. Our approach, by performing dimensionality reduction of the normalized Laplacian matrix, results to a smaller and more compact graph matrix than the original one, as will be also shown experimentally. (iv) We define two new node similarity measures that exploit local and global characteristics of a network. In particular, we calculate the similarity between nodes that belong in the same cluster and similarity between nodes that belong in different clusters by exploiting triangular inequality between the two nodes and the center of a cluster. (v) Compared to the bulk of research on social networks that has focused almost exclusively on positive interpretations of links between people, we also study the interplay between positive and negative relationships.

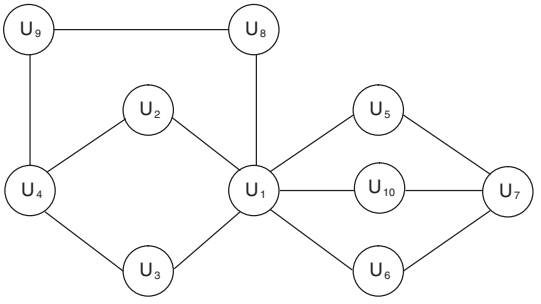

Fig. 1. Network example.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

Respectively, we also apply our proposed scheme to a PPIN with positive and negative links. (vi) We perform extensive experimental comparison of the proposed method against existing link prediction algorithms, another spectral clustering algorithm and *k*-means, using synthetic and real data sets.

The rest of this paper is organized as follows. Section 2 summarizes the related work, whereas Section 3 briefly reviews preliminaries in graphs employed in our approach. A motivating example, the proposed approach, an extension for signed networks and its complexity analysis, are described in Section 4. Experimental results are given in Section 5. Section 6 discusses solutions to problems in the link prediction task. Finally, Section 7 concludes this paper.

2. Related work

The research for link prediction in OSNs consists of local and global approaches [24]. The local approaches focus mainly on the local node's structure, whereas the global approaches, detect the overall path structure in a network.

There is a variety of local-based similarity measures [24,25,49], which are node-dependent (i.e. Common Neighbors index or else known as FOAF [6] algorithm, Adamic/Adar [2] index, Jaccard Coefficient, etc.) for analyzing the "proximity" of nodes in a network. Common Neighbors index, also known as Friend of a Friend algorithm (FOAF) [6], is adopted by many popular OSNs, such as facebook.com and hi5.com for the friend recommendation task. FOAF is based on the common sense that two nodes v_x, v_y are more likely to form a link in the future, if they have many common neighbors. In addition to FOAF algorithm, there are also more complicated local-based measures such as Jaccard Coefficient and Adamic/Adar index. Jaccard Coefficient [24] is a commonly used similarity metric in Information Retrieval. To measure proximity between two nodes v_x and v_y, Jaccard Coefficient measures the ratio of the number of common neighbors between v_x and v_y to the number of non-common neighbors. Adamic/Adar index [2], which is based on Jaccard Coefficient, measures how strongly "related" two web pages are. To do this, it exploits features of the web pages and defines a similarity measure between them, by refining the simple counting of common features (Jaccard Coefficient) by weighting rarer features more heavily.

There are a variety of global approaches [24], which are path-dependent (i.e. Katz [17] status index, RWR [33] algorithm, SimRank [16] algorithm, the commute time [8] algorithm etc.). Leo Katz [17] introduced a status index derived from sociometric analysis. His method computes the important and influential nodes in a social network. He also used the concept of attenuation in influence transmitted through intermediary nodes. RWR algorithm [33] (Random Walk with Restart algorithm) is based on a Markov-chain model of random walk through a graph. RWR considers a random walker that starts from node v_x and chooses randomly among the available edges every time, except that before making a choice, with probability c he goes back to node v_x (restart). Thus, the relevance score of node v_x with respect to node v_y is defined as the steady-state probability r_{v_x,v_y} that the random walker will finally stay at node v_y.

As far as PPINs are concerned, there are a lot of biochemical and biophysical methods to detect interactions in such networks [19,26]. However, since molecular biology techniques are quite expensive and very often time-consuming, it is by far preferable to apply graph theory techniques to study such kind of problems. Authors in [36] use sequence data to apply spectral clustering techniques. They prove that their algorithm offers competitive performance on the clustering of biological sequence data. Authors in [13] also present a simple and unified derivation of the spectral algorithms and they apply it to microarray datasets. They illustrate the performance of spectral algorithms by providing numerous experimental results. Stelzl et al. [40] also studied a human protein–protein interaction network and they developed a tool for the identification of PPINs, which can be used to detect interactions across the entire proteome of an organism. Algorithms for reducing the noise presented in PPI networks [23] and predicting protein functions from weighted PPIs [18] have also been proposed. Another tool, named Local Protein Community Finder has also been developed from the authors in [44]. This tool finds a community close to a queried protein in any network specified by the user. Generally, a variety of computational methods have been investigated so far for the protein network inference problem [4,5]. Authors in [27] present a local path index to estimate the likelihood of the existence of a link between two nodes. Authors in [46] introduce a method based on a variant of kernel canonical correlation analysis to predict the protein network of a yeast. Other methods try to predict protein interactions from evolutionary similarities [35], while others combine different sources of data to infer the network [30].

Spectral clustering, is one of the most popular modern clustering algorithms. Its efficacy is mainly based on the fact that it does not make any assumptions on the form of the clusters [38,47]. This property comes from the mapping of the original space to an eigensystem. Due to this virtue, Spectral clustering is applied in many different research areas, such as bioinformatics [13] for clustering biological sequence data and computer imaging [38] for image segmentation.

There are two main categories of spectral clustering algorithms based on the number of eigenvectors they use. The first category [29,38,39] uses a matrix of affinities between nodes and clusters the nodes based on the second smallest eigenvector of the Laplacian matrix. Then, recursively uses the second smallest eigenvector to further partition these clusters. A representative example of this category is the two-way Ncut algorithm [29,38,39]. The second category, which is similar to our new enhanced method, directly computes a multi-way partition of the data [32]. In particular, it selects the largest k eigenvectors and their corresponding eigenvalues. Then, it extracts the clusters by finding the approximate equal elements in the selected eigenvectors using any clustering algorithm e.g. *k*-means.

Recently, Yan et al. [47] proposed a general framework for fast approximate spectral clustering in which a distortion-minimizing local transformation is first applied to the data. This framework is based on a theoretical analysis that provides a statistical characterization of the effect of distortion on the mis-clustering rate. Moreover, Abbassi and Mirronki [1] proposed a spectral method for designing a recommender system for blogs. However, the fact that they do not weight differently the similarities between nodes that belong in the same cluster and nodes that belong in different clusters is questionable. Jerome Kunegis and Andreas Lommatzsch [21] proposed a unified framework for learning link prediction and edge weight prediction functions in large networks, based on the

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

transformation of a graph's algebraic spectrum. Kunegis et al. [20] also introduced a link prediction algorithm based on the extrapolation of a network's spectral evolution, a method which generalizes several common graph kernels that can be expressed as spectral transformations. In addition, Kunegis et al. [22] studied the problem of signed global networks as well, by identifying unpopular users and predicting the sign of links. Finally, Yen et al. [48] addressed the problem of clustering the nodes of a weighted and undirected graph by using the sigmoid commute-time kernel, a measure for detecting similarity between nodes of a graph.

The novelty of our new proposed method compared to existing approaches is as follows:

- Recently, extensive empirical analysis has demonstrated that FOAF [6] algorithm, performs better than other complicated variants [25,49] such as Adamic/Adar index and Jaccard Coefficient. Thus, we compare our method against FOAF algorithm as representative of the local-based measures, and as will be experimentally shown later, our method outperforms FOAF algorithm in terms of accuracy.
- In contrast to global algorithms, such as the Katz index [17] and the Random Walk with Restart (RWR) algorithm [33], our method is more efficient. This means that our method, which is based on the top few eigenvectors and eigenvalues of the normalized Laplacian matrix, requires less time and space complexity than global algorithms. We compare our method against RWR, as representatives of the global algorithms, and as will be shown experimentally later, our method outperforms RWR in terms of accuracy and time complexity.
- In contrast to traditional clustering algorithms, such as *k*-means and DBSCAN, our method is more flexible, because it captures (i) the non-connected components of a graph, and (ii) a wider range of cluster geometries and shapes [47]. Thus, it results to better friend recommendations and protein link predictions. We have compared our method against *k*-means, as representative of the clustering algorithms, and as will be shown experimentally later, our method is more effective than *k*-means.

Besides the aforementioned link prediction algorithms that are based solely on the graph structure, there are alternative methods that exploit other data sources such as messages among users, co-authored paper, common tagging etc. For instance, Ido Guy et al. [12], proposed a novel user interface widget for providing users with recommendations of people. Their people recommendations were based on aggregated information collected from various sources across IBM organization (i.e. common tagging, common link structure, common co-authored papers etc.). Chen et al. [6] evaluated four recommender algorithms (Content Matching, Content-plus-Link, FOAF algorithm and, SONAR) to help users discover new friends on IBM's OSN. TidalTrust [10] and MoleTrust [31] are also hybrid approaches combining the rating data of collaborative filtering systems with the link data of trust-based social networks (i.e. Epinions.com) to improve the prediction accuracy. In contrast to the above methods, we focus only on predictions based on the link structure of an OSN and a PPIN and thus, we will exclude them from our experimental comparison.

3. Preliminaries in graphs

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of edges such that an edge joins a pair of vertices. In this paper, $\mathcal{G}$ will always be a general undirected and unvalued graph as shown in Fig. 1. $\mathcal{G}$ can express friendships among users of an OSN or interactions among proteins of a PPIN and will be used as our running example, throughout the paper. Notice that our running example concerns a friendship network.

The adjacency matrix $\mathcal{A}$ of graph $\mathcal{G}$ is a matrix with rows and columns labeled by graph vertices, with a 1 or 0 in position (v_i, v_j) according to whether v_i and v_j are connected or not. For an undirected graph, the adjacency matrix is symmetric. In Table 1, we present the resulting adjacency matrix $\mathcal{A}$ of graph $\mathcal{G}$. Notice that in $\mathcal{A}$ we use zeros along the diagonals, to depict that a node is not connected to itself. In case of a large graph $\mathcal{G}$, it is important to note that its adjacency matrix $\mathcal{A}$ can be characterized by high dimensionality and sparsity.

The spectral algorithms are based on eigenvectors of Laplacians, which are a combination of the adjacency and the degree matrix. The normalized Laplacian matrix of graph $\mathcal{G}$ is computed by equation $\mathcal{L} = \mathcal{D}^{-\frac{\alpha}{\in}} \times (\mathcal{D} - \mathcal{A}) \times \mathcal{D}^{-\frac{\alpha}{\in}}$, where $\mathcal{D}$ is the degree matrix of graph $\mathcal{G}$. The normalized Laplacian matrix $\mathcal{L}$ is positive semi-definite and has n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_n$. Moreover, the number of 0 eigenvalues equals the number of the connected components in a graph.

Table 2 presents the most important symbols and their corresponding definitions, which are used frequently in the sequel.

Table 1
Adjacency matrix $\mathcal{A}$ of graph $\mathcal{G}$.

	$\mathbf{U}_1$	$\mathbf{U}_2$	$\mathbf{U}_3$	$\mathbf{U}_4$	$\mathbf{U}_5$	$\mathbf{U}_6$	$\mathbf{U}_7$	$\mathbf{U}_8$	$\mathbf{U}_9$	$\mathbf{U}_{10}$
$\mathbf{U}_1$	0	1	1	0	1cp	1	0	1	0	1
$\mathbf{U}_2$	1	0	0	1	0	0	0	0	0	0
$\mathbf{U}_3$	1	0	0	1	0	0	0	0	0	0
$\mathbf{U}_4$	0	1	1	0	0	0	0	0	1	0
$\mathbf{U}_5$	1	0	0	0	0	0	1	0	0	0
$\mathbf{U}_6$	1	0	0	0	0	0	1	0	0	0
$\mathbf{U}_7$	0	0	0	0	1	1	0	0	0	1
$\mathbf{U}_8$	1	0	0	0	0	0	0	0	1	0
$\mathbf{U}_9$	0	0	0	1	0	0	0	1	0	0
$\mathbf{U}_{10}$	1	0	0	0	0	0	1	0	0	0

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

t2.1 **Table 2**
t2.2 Symbols used throughout the study.

t2.3 Symbol	Description
$\mathcal{G}$	Undirected and unvalued graph
$\mathcal{V}$	Set of graph nodes (vertices)
$\mathcal{E}$	Set of graph edges
$\mathcal{A}$	Adjacency matrix of graph $\mathcal{G}$
$\mathcal{D}$	Degree matrix of graph $\mathcal{G}$
$\mathcal{L}$	Normalized Laplacian matrix of graph $\mathcal{G}$
t2.10 u_i	Eigenvector of $\mathcal{L}$
t2.11 λ_i	Eigenvalue of $\mathcal{L}$
t2.12 v_i	Graph node
t2.13 e_i	Graph edge
t2.14 $sim(v_i,v_j)$	Similarity between nodes v_i and v_j
t2.15 n	Number of vertices in graph $\mathcal{G}$

4. The proposed approach

In this section, through a motivating example we first provide the outline of our approach, named SpectralLink. Next, we analyze the steps of the proposed algorithm.

4.1. Outline

Here, we describe how SpectralLink is applied on OSNs/PPINs and how the link prediction is performed according to the detected associations.

When using an OSN, users explicitly declare their friends so that they are able to share information items with them (i.e. photos, news etc.). After some time, the social network accumulates a set of connection data (graph of friendships), which can be represented by an undirected graph similar to that of Fig. 1.

Our SpectralLink approach finds similarities between nodes in an undirected graph constructed from these connection data. The SpectralLink algorithm uses as input the connections of a graph $\mathcal{G}$ and outputs a similarity matrix between any two nodes in $\mathcal{G}$. Therefore, friends can be recommended to a target user u according to their weights in the similarity matrix.

In the following, to illustrate how our approach works, we apply the SpectralLink algorithm to our running example. As illustrated in Fig. 1, 10 users are connected in a graph. If we have to recommend a new friend to U_1, then there is no direct indication for this task in the original adjacency matrix $\mathcal{A}$, as shown in Table 1. However, after performing the SpectralLink algorithm, we can get a similarity matrix between any two nodes of graph $\mathcal{G}$ and recommend friends according to their weights.

Firstly, SpectralLink computes the first k eigenvectors $u_1, \ldots, u_k$ with the corresponding $\lambda_1, \ldots, \lambda_k$ eigenvalues of $\mathcal{L}$ based on equation $\mathcal{L} \times \mathcal{U} = \lambda \times \mathcal{U}$, where $\mathcal{U}$ matrix has columns, the eigenvectors $u_1, \ldots, u_k$ and nodes $v_i \in R$, with $i = 1, \ldots, n$, corresponding to the i-row of $\mathcal{U}$. In our running example, we compute the first $k = 2$ eigenvectors and $\lambda = 2$ of $\mathcal{L}$, respectively, as shown in Tables 3 and 4.

Secondly, we cluster nodes v_i of $\mathcal{U}$ with the k-means algorithm into clusters $C_1, \ldots, C_k$. In our running example, k is equal to 2. Thus, we will partition data in 2 clusters. In Table 5, we present vector IDX with $i = 1, \ldots, n$ rows, which correspond to the assignment of a node v_i in one of the two clusters. Thus, node U_1 is assigned in cluster C_1, node U_2 is assigned in cluster C_2, etc. Moreover, based on the k-means algorithm, we can compute the centroids of each cluster. This information is shown, in Table 7. Based on the distances of each node from each cluster centroid we can define matrix D, which is shown in Table 6. Vector IDX and matrix D will be used in the next step of SpectralLink to calculate the similarity between nodes that belong in the same cluster and similarity between nodes that belong in different clusters.

t3.1 **Table 3**
t3.2 The first $k = 2$ eigenvectors of $\mathcal{L}$.

t3.3	$\mathbf{u}_1$	$\mathbf{u}_2$
t3.4 1	−0.440	0.072
t3.5 2	−0.291	−0.220
t3.6 3	−0.291	−0.220
t3.7 4	−0.325	−0.426
t3.8 5	−0.295	0.304
t3.9 6	−0.295	0.304
t3.10 7	−0.334	0.453
t3.11 8	−0.285	−0.244
t3.12 9	−0.278	−0.417
t3.13 10	−0.295	0.304

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

Table 4
The first $\lambda = 2$ eigenvalues of $\mathcal{L}$.

λ_1	λ_2
0.892	0.750

Table 5
Vector *IDX* which assigns each u_i node in a specific cluster.

User	Cluster
$\mathbf{U}_1$	1
$\mathbf{U}_2$	2
$\mathbf{U}_3$	2
$\mathbf{U}_4$	2
$\mathbf{U}_5$	1
$\mathbf{U}_6$	1
$\mathbf{U}_7$	1
$\mathbf{U}_8$	2
$\mathbf{U}_9$	2
$\mathbf{U}_{10}$	1

Moreover, in Fig. 2a, we present the 10 nodes of our running example, in the 2-dimensional space based on the first 2 eigenvectors of $\mathcal{L}$ matrix. Additionally, in Fig. 2b, we present the resulting partition of the 10 nodes of graph $\mathcal{G}$ in 2 clusters. Thus, the nodes that are assigned in cluster $C_1 = \{U_2,U_3,U_4,U_8,U_9\}$, whereas the nodes that are assigned in cluster $C_2 = \{U_1,U_5,U_6,U_7,U_{10}\}$. As shown, the partition of *k*-means is in accordance with the visual representation in the 2-dimensional space of the nodes in Fig. 2a.

Thirdly, in contrast to a previous proposed work published in [15], in order to quantify the similarity between nodes, we are based on triangle inequality which states that for any triangle the sum of the lengths of any two sides must be greater than the length of the remaining side. Since we have calculated matrix *D* with the distances of each node from the centroid of each cluster, based on triangle inequality the distance (i.e. dissimilarity) between any pair of nodes *i* and *j* is bounded in this space: $|D(i, IDX(i)) - D(j, IDX(j))| \leq dist(i,j) \leq D(i, IDX(i)) + D(j, IDX(j))$.

For similarity bounded by 0 and 1, when similarity is one (i.e. exactly similar), the distance (dissimilarity) is zero and when the similarity is zero (i.e. very different), the dissimilarity is one. To quantify the similarity between nodes that belong in the same cluster, we have adapted the Bray–Curtis similarity measure [3], which also ranges in [0,1], using Eq. (1):

$$SimSC(i,j) = \frac{|D(i,IDX(i)) - D(j,IDX(j))|}{D(i,IDX(i)) + D(j,IDX(j))}. \quad (1)$$

Notice that, in contrast to Bray–Curtis similarity measure, our measure does not violate the property of triangular inequality. In our running example the similarity between nodes U_1 and U_7 that belong to same cluster C_1 based on Eq. (1) is: $\frac{|D(1,1)-D(7,1)|}{D(1,1)+D(7,1)} = \frac{|0.144-0.027|}{0.144+0.027} = 0.684$. Moreover, to quantify the similarity between nodes that belong to different clusters we use Eq. (2):

$$SimDC(i,j) = \frac{|D(i,IDX(j)) - D(j,IDX(i))|}{D(i,IDX(j)) + D(j,IDX(i))}. \quad (2)$$

Table 6
Matrix *D* with the distances of each node from the centroid of each cluster.

	$\mathbf{C}_1$	$\mathbf{C}_2$
$\mathbf{U}_1$	0.144	0.412
$\mathbf{U}_2$	0.783	0.009
$\mathbf{U}_3$	0.783	0.0009
$\mathbf{U}_4$	1.053	0.010
$\mathbf{U}_5$	0.005	1.011
$\mathbf{U}_6$	0.005	1.011
$\mathbf{U}_7$	0.027	1.145
$\mathbf{U}_8$	0.843	0.003
$\mathbf{U}_9$	1.115	0.020
$\mathbf{U}_{10}$	0.005	1.011

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

Table 7
The coordinates in the 2-D space of each cluster centroid.

	x	y
Centroid_C_1	−0.734	0.624
Centroid_C_2	−0.703	−0.697

Thus, in our running example the similarity between nodes U_1 and U_4 that belong to different clusters based on Eq. (2) is: $\frac{|D(1,2)-D(4,1)|}{D(1,2)+D(4,1)} = \frac{|0.412-1.053|}{0.412+1.053} = 0.437$. It is obvious that Eq. (1) promotes similarity between nodes that belong to the same cluster. In contrast, Eq. (2) penalizes similarities between nodes that belong to different clusters.

In Table 8, we present the node similarity matrix of graph $\mathcal{G}$. For readability reasons, we put zero values to already adjacent nodes. In our running example, as shown in Table 8, node U_1 would receive node U_7 as recommendation. The resulting recommendation is reasonable, because U_1 has 3 common interactors with node U_7. In contrast, U_1 has only 2 common interactors with node U_4. That is, the SpectralLink approach is able to capture the associations among the graph data objects. The associations can then be used to improve the friend/protein recommendation procedure, as will be verified by our experimental results.

4.2. The SpectralLink algorithm

In this section, we describe our new SpectralLink algorithm in detail. Our SpectralLink algorithm computes node similarity for a node v_i with each node v_j in a graph $\mathcal{G}$.

As shown in Fig. 3, our SpectralLink algorithm is based on matrix $\mathcal{L}$ of a graph $\mathcal{G}$. It takes the first k eigenvectors $u_1, \ldots, u_k$ of $\mathcal{L}$. Then, based on these eigenvectors it clusters nodes $v_1 \ldots v_n$ of graph $\mathcal{G}$ with k-means algorithm. Next, based on the distances of each node v_i from the nearest cluster centroid it calculates similarities between a test node and the other nodes in graph $\mathcal{G}$. Finally, for a test node we rank the calculated similarities with other nodes and predict the top ranked ones as his possible friends or interacting proteins.

4.3. Extending SpectralLink for signed networks

In this Section, we derive variants of SpectralLink that apply to directed networks and networks with weighted edges, including the case of edges with negative weights (signed networks).

Signed networks edges have positive (+1) as well as negative (−1) weights. Such signed graphs arise for instance in social networks (i.e. Epinions.com, Shashdot Zoo, etc.) where negative edges denote distrust instead of trustiness. In biology, proteins in cells tend to form complex signaling networks that respond to various signals, ranging from environmental conditions, hormones or neurotransmitters to ions, and perform a series of tasks such as cell growth, maintenance of cell survival, proliferation, differentiation, development and apoptosis [40]. In such signed graphs, SpectralLink algorithm, can be adjusted accordingly. Firstly, we can use an alternative definition of diagonal degree matrix [14,21] by using the absolute diagonal degree matrix $\mathcal{D}_{ii} = \sum_{j=1}^{n} |\mathcal{A}_{ij}|$. Then, we can define the signed normalized Laplacian matrix, by giving $\mathcal{L} = \mathcal{D}^{-\frac{1}{2}} \times (\mathcal{D} - \mathcal{A}) \times \mathcal{D}^{-\frac{1}{2}}$.

As the unsigned normalized Laplacian matrix, the signed normalized Laplacian matrix is positive semi-define. However, when each connected component of the graph contains a cycle with an odd number of negatively weighted edges, then the signed normalized Laplacian matrix can be positive-definite.

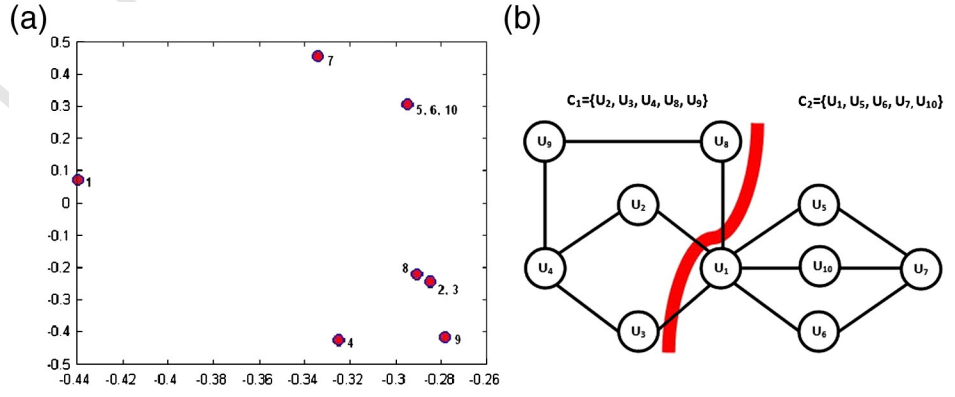

Fig. 2. For our running example, we present the (a) 2-D space plot of nodes of graph $\mathcal{G}$ based on the second eigenvector of $\mathcal{L}$ and (b) the resulting partition of the 10 nodes of graph $\mathcal{G}$ in 2 clusters.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

Table 8
Node similarity matrix. It presents the possibility of interaction between two nodes.

	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}
U_1	0.000	0.000	0.000	0.437	0.000	0.000	0.684	0.000	0.460	0.000
U_2	0.000	0.000	0.000	0.000	0.127	0.127	0.187	0.500	0.379	0.127
U_3	0.000	0.000	0.000	0.000	0.127	0.127	0.187	0.500	0.379	0.127
U_4	0.437	0.000	0.000	0.000	0.020	0.020	0.041	0.538	0.000	0.020
U_5	0.000	0.127	0.127	0.020	0.000	0.000	0.000	0.090	0.048	0.000
U_6	0.000	0.127	0.127	0.020	0.000	0.000	0.000	0.090	0.048	0.000
U_7	0.684	0.187	0.187	0.041	0.000	0.000	0.000	0.151	0.013	0.000
U_8	0.000	0.500	0.500	0.538	0.090	0.090	0.151	0.000	0.000	0.090
U_9	0.460	0.379	0.379	0.000	0.048	0.048	0.013	0.000	0.000	0.048
U_{10}	0.000	0.127	0.127	0.020	0.000	0.000	0.000	0.090	0.048	0.000

4.4. Complexity analysis

Social and protein–protein interaction networks are large and contain a significant amount of information. Global based algorithms that can be used for link prediction, such as Random Walk with Restart (RWR) [33,43] is computationally prohibitive for large graphs. In particular, RWR's main computational cost consists of a large matrix inversion, which has $O(n^3)$ time complexity. It is important to mention here that Tong et al. [43] proposed a faster version of RWR. However, it preserves almost 90% quality of the original RWR, which is a questionable solution for the link prediction problem, where accuracy is the most important parameter. Moreover, space complexity is another limitation of the RWR algorithms, since they require $O(n^2)$ memory space.

Friend of a Friend algorithm (FOAF), as a representative of the local-based methods, considers very small paths between any pair of nodes in $\mathcal{G}$. In particular, for each v_x node, FOAF traverses all its neighbors and then traverses the neighbors of each of v_x's neighbor. Since the time complexity to traverse the neighborhood of a node is simply h (h is the average node degree in a network) and our graph $\mathcal{G}$ is sparse, it holds that $h << n$. Thus, the time complexity of FOAF is $O(n \times h^2)$. The space complexity for FOAF is $O(n \times h)$.

Solving a standard eigenvalue problem for all eigenvectors takes $O(n^3)$ operations, where n is the number of graph nodes. This becomes impractical for applications with n on the order of millions. However, real social and protein–protein interaction networks have often the following properties: 1) The graphs are often only locally connected and the resulting eigensystems are very sparse, and 2) only the top few eigenvectors are needed for graph partitioning. These special properties of our problem can be fully exploited by an eigensolver called the Lanczos method [11]. The time complexity of a Lanczos algorithm is $O(m \times n) + O(m \times M(n))$, where m is a usually small constant number of matrix–vector computations required, n is the number of nodes in a graph, and $M(n)$ is the cost of a matrix–vector computation of $\mathcal{L} \times x$, where $\mathcal{L}$ is the normalized Laplacian matrix and x is an eigenvector.

Algorithm SpectralLink $(\mathcal{G}, \mathcal{A}, n, k)$
Input
$\mathcal{G}$: an undirected and unweighted graph
$\mathcal{A}$: adjacency matrix of graph $\mathcal{G}$,
n: number of nodes of graph $\mathcal{G}$,
k: number of clusters
Output
$sim(v_i, v_j)$: similarity between node v_i with each node v_j in $\mathcal{G}$

1. Compute the diagonal degree matrix $\mathcal{D}$ with elements:
$$\mathcal{D}_{ii} = \sum_{j=1}^{n} \mathcal{A}_{ij}$$
2. Compute the normalized Laplacian matrix:
$$\mathcal{L} = \mathcal{D}^{-\frac{\infty}{\in}} \times (\mathcal{D} - \mathcal{A}) \times \mathcal{D}^{-\frac{\infty}{\in}}$$
3. Find the first k eigenvectors $u_1, \ldots, u_k$ of L
4. Let matrix $\mathcal{U} \in \mathcal{R}$ contain $u_1, \ldots, u_k$ eigenvectors as columns and nodes $v_i \in R$, with $i = 1, \ldots, n$, correspond to the i-row of $\mathcal{U}$
5. Cluster the nodes v_i with k-means algorithm into clusters $C_1, \ldots, C_k$
6. For a node v_i compute its similarity with each node v_j that belongs in the same cluster based on Equation 1
7. For a node v_i compute its similarity with each node v_j that belongs to a different cluster based on Equation 2

Fig. 3. The SpectralLink algorithm.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

The running time of the two-way Ncut algorithm is $O(m \times n)$, where n is the number of nodes and m is the number of steps Lanczos takes to converge. Moreover, the time complexity of k-means is $O(n \times k \times i \times d)$, where n is the number of nodes, k is the number of clusters, i is the number of iterations until k-means converge, and d is the number of attributes, where each node can be expressed as a d-dimensional real vector. The space complexity of k-means is $O((n + k) \times m)$, where m is the number of centroids, that are stored in each iteration.

Recently Yan et al. [47] proposed a k-means-based approximate spectral clustering algorithm (KASP), which applies firstly k-means to cluster the nodes of a graph and then applies spectral clustering only on the cluster centroids (representative nodes of graph). By using this implementation, the overall computation cost of SpectraLink is $O(k^3) + O(n \times k \times i \times d)$.

5. Experimental evaluation

In this section, we experimentally compare our new approach SpectralLink, with k-means [28] algorithm, the two-way normalized cut algorithm [29,38,39], the Random Walk with Restart [33] algorithm, and the Friend of a Friend [6] algorithm, denoted as k-means, two-way Ncut, RWR, and FOAF, respectively. Our experiments were performed on a 3 GHz Pentium IV, with 2 GB of memory, running Windows XP. All algorithms were implemented in Matlab.

5.1. Algorithms settings

In this section, we present detailed information of the algorithms that will be compared experimentally with our proposed method:

k-means algorithm: Given a set of nodes (v_1,v_2, …,v_n) of a graph $\mathcal{G}$ and its adjacency matrix, k-means aims to partition the n nodes into k sets ($k < n$) $C = (C_1, C_2 \ldots, C_k)$ to minimize the within-cluster sum of squared error(SSE), as shown by Eq. (3):

$$SSE = \sum_{i=1}^{k} \sum_{v_x \in C_i} dist(v_x, c_i)^2, \tag{3}$$

where v_x is a node in cluster C_i and c_i is the centroid point for cluster C_i. k-means chooses k initial centroids, where k is a user-specified parameter, namely, the number of clusters desired. Each node is then assigned to the closest centroid, and each collection of nodes assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the nodes assigned to the cluster. This procedure is repeated until no node changes cluster, or equivalently, until the centroids remain the same. After the cluster formation, for a node v_i compute its similarity with each node v_j that belongs in the same cluster based on Eq. (1). Moreover, for a node v_i compute its similarity with each node v_j that belongs to a different cluster based on Eq. (2).

Two-way Ncut clustering algorithm: Given a set of nodes (v_1,v_2, …,v_n) of a graph $\mathcal{G}$, and its adjacency matrix, two-way normalized cut (two-way Ncut) algorithm aims to bipartition the n nodes to minimize the Normalized Cut [29,38,39]. In particular, two-way Ncut solves the generalized eigenvalue problem for the second smallest eigenvalue, as shown by Eq. (4):

$$(D-A)y = \lambda Dy \tag{4}$$

where D is a diagonal matrix, A is the adjacency matrix, λ is the second smallest eigenvalue and y is the second smallest eigenvector. Two-way Ncut uses the eigenvector with the second smallest eigenvalue to bipartition the graph, and decides if the current partition should be subdivided again by checking *Ncut* variable stability. In other words, the algorithm decides if the current partition should be subdivided by checking the stability of the *Ncut* and making sure that *Ncut* is below the pre-specified threshold. Finally, it recursively performs repartition of the segmented parts if necessary and gives as a result a number of groups, in which the clustered nodes are contained.

Random Walk with Restart Algorithm: The "random walk with restart" (RWR) algorithm [33] operates as follows: consider a random walker that starts from node v_x. The random walker chooses randomly among the available edges every time, except that, before he makes a choice, with probability c, he goes back to node v_x (restart). Thus, the relevance score of node v_x wrt. node v_y is defined as the steady-state probability r_{v_x,v_y} that the random walker will finally stay at node v_y, as shown by Eq. (5):

$$\vec{r}_{v_x} = c \cdot A \cdot \vec{r}_{v_x} + (1-c) \cdot \vec{e}_{v_x}, \tag{5}$$

where $\vec{e}_{v_x}$ is the $n \cdot 1$ starting vector with the v_x^{th} element equal to 1 and 0 for the other elements of the vector, and A is the adjacency matrix of graph $\mathcal{G}$.

Eq. (5) defines a linear system problem where $\vec{r}_{v_x}$ is a $n \cdot 1$ ranking vector and element r_{v_x,v_y} is the relevance score of node v_y wrt. node v_x, as shown by Eq. (6):

$$\vec{r}_{v_x} = (1-c) \cdot (I - c \cdot A)^{-1} \cdot \vec{e}_{v_x}. \tag{6}$$

In our experiments, we tuned the c parameter and the best results were produced when c is equal to 0.005.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

Friend of a Friend algorithm: The Friend of a Friend (FOAF) algorithm [6] leverages only network information of Friending based on the intuition that "if many of my friends consider Alice a friend, perhaps Alice could be my friend too". The clear intuition behind it, is the primary algorithmic foundation of the "People You May Know" feature on Facebook, which is one of the few known people recommenders deployed on a social networking site. Formally speaking, if we define predicate $F(v_i,v_j)$ to be true if and only if node v_i has a connection with node v_j, the algorithm can be described as follows: for a node v_x being the recipient of a recommendation, its recommendation candidate set is defined as follows [6]:

$$RC(v_x) = \{\text{node } v_c | \exists \textit{ node } v_i \text{ s.t. } F(v_x v_i) \textit{ and } F(v_i v_c)\}.$$

For each candidate node $v_c \epsilon RC(v_x)$, its common interactor node set is $CF(v_x,v_c) = \{\text{node } v_i | F(v_x,v_i) \text{ and } F(v_i,v_c)\}$, which represents the interactors of v_x that connect to v_c and thus serve as a bridge between v_x and v_c. We then define the score of each candidate v_c for recipient v_x as the size of $CF(v_x,v_c)$.

The candidates are recommended to v_x in decreasing order of their score. For a single recommended candidate v_c, we supply the common interactors in $CF(v_x,v_c)$ as the explanation for recommending v_c. Thus, FOAF provides recommendations, considering only pathways of maximum length 2 between an individual and his possible interactors in a social or a protein–protein interaction network. Therefore, users/proteins can be recommended to v_x according to the number on length-2 paths connecting them with him in the network.

5.2. Real and synthetic evaluation data sets

To evaluate the examined algorithms, we have used two synthetic data sets (50K,100K), three real social networks (Facebook, Hi5 and Epinions) and three real human protein data sets (Human, Human Disease and Human Signaling).

5.2.1. Real OSNs datasets

We crawled the graph data from the Facebook and Hi5 web sites at two different time periods. In particular, we crawled the Facebook web site on the 30th of October 2009 and on the 15th of December 2010. Our data crawling method was the following: For each user u, we traverse all his friends and then traverse the friends of each of u's friends etc. From the first crawl of Facebook web site we created a training data set with 3694 users (network size $N = 3.694$, number of edges $E = 13{,}692$), denoted as Facebook 3.7K, where the initial starting node of our crawling was a random user in Germany. From the second crawl of Facebook web site we created the probe data set with the same users by only preserving 3912 new emerged edges among them. We followed the same crawling procedure from the Hi5 web site. From the first crawl of Hi5 web site we created a training data set with 63,329 users and 88,261 edges among them, denoted as Hi5 63K, where the initial starting node of our crawling was a random user in the US. From the second crawl of Hi5 web site we created the probe data set with the same users by only preserving 16,512 new emerged edges connecting them. The graph data from the first crawl are used to predict the new links emerging in the second crawl. Moreover, we use in our comparison the Epinions[4] 132K data set, which is a who-trusts-whom social network that consist of positive and negative edges. A positive edge implies trust whereas a negative edge implies distrust.

5.2.2. Real PPINs datasets

The first protein data set[5] used in this paper contains a total of 3269 unique interactions between 1925 different human proteins and is denoted as Human Data-set. The second protein data set[6] is a part of the Human Disease Network [9] containing 1200 interactions between 868 proteins and is denoted as Human Disease Data-set. Finally, the third network[7] tested here, called Human Signaling Data-set contains 2938 interactions between 1221 proteins.

5.2.3. Synthetic datasets

The size of real online social networks is huge. For instance, Facebook has over 500 million users with an average of roughly 100 friends each. To study the algorithms' computational complexity performance, we used synthetic network models of different sizes. Although real networks have many complex structural properties [7], such as degree heterogeneity, the rich-club phenomenon, etc., as a start point for generating synthetic data sets, we consider a very simple model.

In contrast to purely random (i.e., Erdos-Renyi) graphs, where the connections among nodes are completely independent random events, our synthetic model follows similar directions with [34,42]. It ensures dependency among the connections of nodes, by characterizing each node with a ten-dimensional vector with each element a randomly selected real number in the interval $[-1,1]$. This vector represents the node's intrinsic features such as the profile of a person. Two nodes are considered to be similar and thus of high probability to connect to each other if they share many close attributes. Given a network size n and the degree k of each node, we start with an empty network with n nodes. At each time step, a node with the smallest degree is randomly selected. Among all other nodes whose degrees are smaller than k, this selected node will connect to the most similar node with probability $1 - p$, while a

[4] http://www.trustlet.org/wiki/Downloaded_Epinions_dataset.
[5] http://www.cell.com/cgi/content/full/122/6/957/DC1/.
[6] http://www.pnas.org/content/suppl/2007/05/03/0701361104.DC1.
[7] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2174632/?tool=pubmed.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

randomly chosen one with probability p. This process will be terminated when all nodes are of degree k. The parameter $p \in [0,1]$ represents the strength of randomness in generating links, which can be understood as noise or irrationality that exists in almost every real system. Based on the above procedure, we have created 2 synthetic data sets based on different network sizes n (50,000, 100,000), by keeping an identical m nodes degree equal to 50 and for both data sets (p is fixed to 0.2).

5.2.4. Topological properties of all datasets

Table 9 presents several calculated topological properties for all the aforementioned data sets. In Table 9, N represents the total number of nodes, E the total number of edges, ASD the average shortest path distance between node pairs, ADEG is the average node degree, LCC is the average local clustering coefficient and GD is the graph diameter (maximum shortest path distance). Regarding real PPINs, in the first protein data set denoted as Human, the average shortest path distance (ASD) between any two proteins of the network is 5.34. This means that most proteins are very closely linked, a phenomenon that has been described as small world property of networks [41]. According to a definition introduced in [45], a small-world network is defined to be a network where the typical distance between two randomly chosen nodes (ASD) grows proportionally to the logarithm of the number of nodes N in the network. Small-world networks have sub-networks that are characterized by the presence of connections between almost any two nodes within them i.e. high local clustering coefficient (LCC). Moreover, most pairs of nodes are connected by at least one short path (i.e. small ASD). On the other hand, the second protein network denoted as Human Disease, does not consist a small world network since its average shortest path distance is equal to 7.97 though its logarithm of N is equal to 2.9.

Regarding real OSNs, Hi5 63K has a very small LLC (0.02) and a quite big ASD (7.18). Thus, Hi5 data set cannot be considered as a small-world network. In contrast, Facebook 3.7K presents (i) a large clustering coefficient (LCC) equal to 0.11, and (ii) a small average shortest path length (ASD) equal to 4.23 and it can be also considered as a "small world" network.

5.3. Experimental protocol and evaluation metrics

As already described in Section 5.2, in our evaluation we consider the division of Facebook 3.7K, Hi5 63K, Human and Human Disease data sets into two sets, according to the exact time stamp of the links downloaded: (i) the training set $\mathcal{E}^{\mathcal{T}}$ is treated as known information and, (ii) the probe set $\mathcal{E}^{\mathcal{P}}$ is used for testing. No information in the probe set is allowed to be used for prediction. It is obvious that $\mathcal{E}^{\mathcal{T}} \cap \mathcal{E}^{\mathcal{P}} = \emptyset$. For each user/protein that has at least one new friend/interacting protein in $\mathcal{E}^{\mathcal{P}}$ we generate recommendations based on his interactors in $\mathcal{E}^{\mathcal{T}}$. Then, we average the results for each user/protein and compute the final performance of each algorithm.

Epinions, Synthetic, Human and Human Disease data sets do not have time stamps of the edges. The performance of the algorithms is evaluated by applying double cross-validation (internal and external). Each data set was divided into 10 subsets. Each subset ($\mathcal{E}^{\mathcal{P}}$) was in turn used for performance estimation in the external cross-validation. The 9 remaining subsets ($\mathcal{E}^{\mathcal{T}}$) were used for the internal cross-validation. In particular, we performed an internal 9-fold cross-validation to determine the best values of the algorithms' needed parameters. We chose as values for the parameters those providing the best performance on the internal 9-fold cross-validation. Then, their performance is averaged on the external 10-fold cross-validation. The presented results, based on two-tailed t-test, are statistically significant at the 0.05 level.

We use the classic precision/recall metric as performance measure for friend/protein recommendations. For a test user/protein receiving a list of n recommended friends/proteins (top-n list), precision and recall are defined as follows:

Precision is the ratio of the number of relevant users/proteins in the top-n list (i.e., those in the top-n list that belong in the future set of users/proteins of the target user/protein) to n.

Recall is the ratio of the number of relevant users/proteins in the top-n list to the total number of relevant users/proteins (all users/proteins in the future set of the target user/protein).

5.4. Sensitivity analysis for the SpectralLink algorithm

In this Section, we study the sensitivity of SpectralLink accuracy performance in a synthetic, a real social network and in two real human protein data sets (i) with different similarity measures that capture proximity between nodes, (ii) with different k number of clusters and (iii) with different controllable sparsity.

Table 9
Topological properties of the synthetic and the real data sets.

Data-Set	N	E	ASD	ADEG	LCC	GD	Network type
Hi5 63K	63,329	88,261	7.18	2.78	0.02	19	Unsigned
Facebook 3.7K	3694	13,692	4.23	7.21	0.11	10	Unsigned
Epinions 132K	131,828	841,372	1.78	6.38	0.24	14	Signed
Synthetic 50K	50,000	1,250,000	5.65	50	0.11	12	Unsigned
Synthetic 100K	100,000	2,500,000	8.72	50	0.05	15	Unsigned
Human	1925	3269	5.34	3.4	0.02	14	Unsigned
Human Disease	868	1200	7.97	1.9	0.011	18	Unsigned
Human Signaling	1221	2938	3.02	4.56	0.046	10	Signed

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

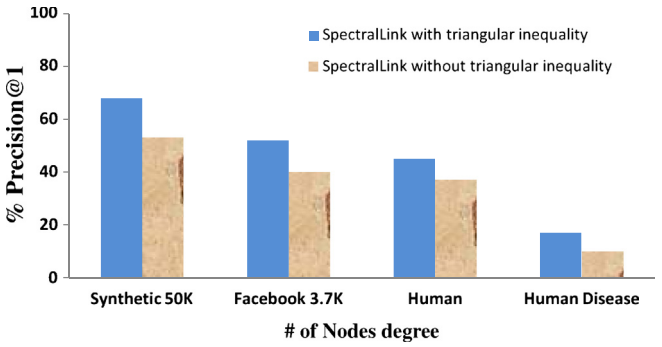

Fig. 4. Precision diagram comparing SpectralLink with and without triangular similarity for all data sets.

As discussed in Section 4.1, we have introduced two new similarity equations (Eqs. (1) and (2)) to quantify the similarity between nodes that belong to same clusters and between nodes that belong to different clusters. Here, we perform experiments by considering also the following similarity measures [15]:

$$SimSC(i,j) = 1 - |min(D(i)) - min(D(j))| \quad (7)$$

$$SimDC(i,j) = \frac{1}{D(i, IDX(j)) + D(j, IDX(i))}. \quad (8)$$

Eqs. (7) and (8) capture the proximity between two nodes by taking into consideration their distance from the cluster centroid. However, they do not consider the triangular inequality between the two nodes and the cluster centroid.

Fig. 4 summarizes the precision performance of the examined data sets, using SpectralLink with triangular inequality (Eqs. (1) and (2)) and SpectralLink without triangular inequality (Eqs. (7) and (8)). It is obvious that our new proposed equations outperform in all cases and this happens because we take full advantage of the upper and lower bound of the triangular inequality, as shown in Section 4.1. Thus, henceforth we will use Eqs. (1) and (2) for all experiments.

In Section 2, one of the required input values for the SpectralLink algorithm is the number k of clusters. To improve our recommendations in terms of effectiveness, it is important to fine-tune the k variable. For the synthetic 50K data set, we examine the performance of precision metric when we recommend a top 1 friend (i.e. %precision@1) vs. different values of k. Fig. 5a illustrates precision for varying k values in the synthetic 50K data set.

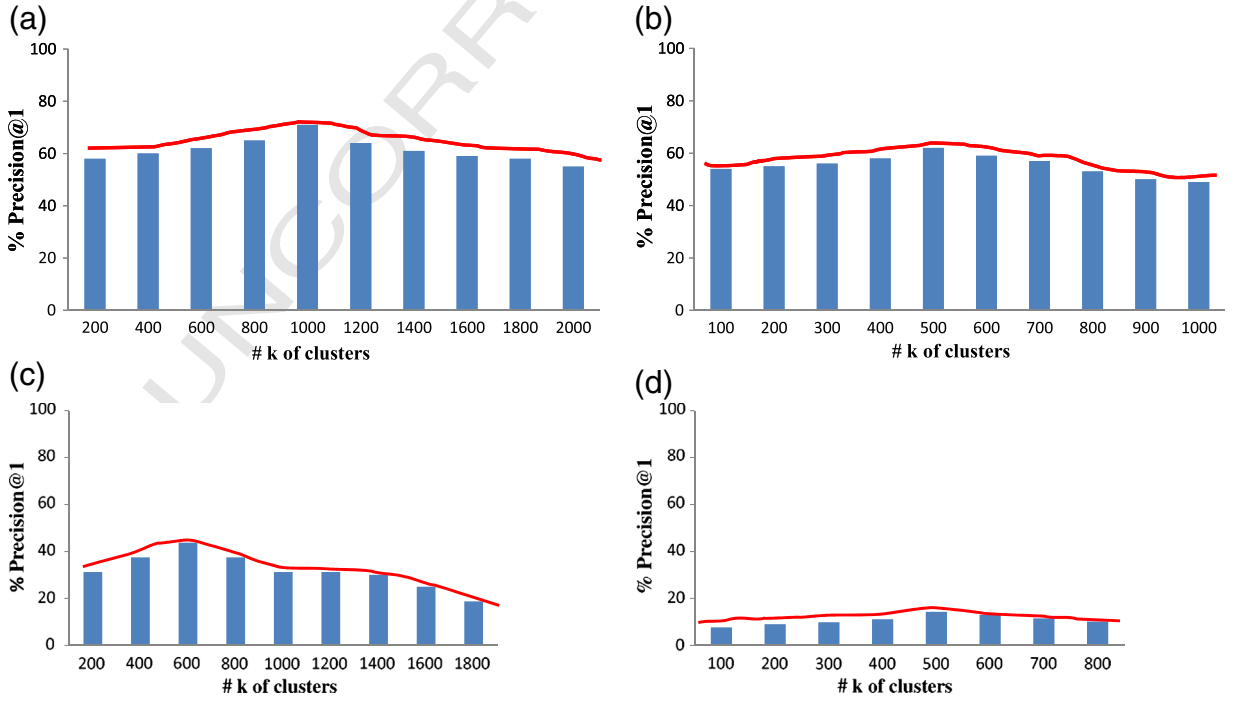

Fig. 5. Precision vs. number k of clusters diagrams for: (a) Synthetic 50K, (b) Facebook 3.7K, (c) Human and (d) Human Disease data sets.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

As expected, the best precision performance of SpectralLink is attained with $k = 1000$ clusters. The reason is the average node degree (ADEG) of the 50K data set, which is equal to 50. Thus, with a number $k = 1000$ of clusters, we get an average cluster size, which corresponds to ADEG of this data set. In the following, we keep $k = 1000$ as the default initial value for the SpectralLink algorithm for this data set. For the Facebook 3.7K data set, we follow the same tuning procedure. Fig. 5b illustrates precision for varying number k of clusters. The best result is attained for $k = 500$. Once again, the initial k number is analogous to ADEG (i.e. 7.21) and the network size (i.e. 3694/7.21 = 512). We also tuned the k variable for the two protein data sets. As shown in Fig. 5c and d, for the Human Protein data set, best performance is obtained when k equals 600 and for the Human Disease Protein data set when k is equal to 500. These numbers are also accordant to ADEG and the size of each network. We have to notice that the number of selected clusters could reduce the gains over the predicting accuracy. This is why our method requires a fine-tuning on the number of selected clusters. However, the final number of selected clusters can be easily estimated by dividing the N number of nodes in a graph with ADEG, as already shown above.

Next, we measure the accuracy that SpectralLink attains, with different controllable sparsity. To examine the accuracy performance of SpectralLink in terms of different network sparsity, we have created for the 50K synthetic data set 5 different sparsity cases, by changing the m number of friends that a node has (50, 60, 70, 80, 90), as shown in Fig. 6a.

As expected, with k increasing, the precision increases too. For the Facebook 3.7K data set, we also examine 5 different sparsity cases, by changing the m number of friends that a node has (i.e. 3, 4, 5, 6, 7), as shown in Fig. 6b. As expected, the best precision value is attained when we consider more adjacent nodes (i.e. m equal to 7). This is reasonable since the ADEG of Facebook 3.7K data set is equal to 7.21. SpectralLink can predict more effectively new friends for larger m values, since in such cases the network density is increased. Fig. 6c and d show precision diagrams for the two protein data sets and present the increase in precision when a larger amount of protein-neighbors is known. As expected, with increasing the percentage of observed links, the precision increases too. Thus, SpectralLink can predict more effectively new links between proteins for larger node degree values, since in such cases the network density is increased.

5.5. Accuracy comparison of SpectralLink with other methods

We proceed with the comparison of SpectralLink with k-means, two-way Ncut, RWR, and FOAF algorithms, in terms of precision and recall. We examine the ranked list, which is recommended to a target user/protein, starting from the top one. For the Facebook 3.7K data set, in Fig. 7a we plot a precision versus recall curve for all five algorithms.

As shown, SpectralLink outperforms k-means, because it takes into consideration also the degree of connectivity of a graph. Moreover, SpectralLink is more flexible than k-means, because it captures a wider range of cluster geometries and shapes and not only cyclic clusters. Although the two-way Ncut algorithm benefits from the advantages of spectral clustering, it fails to provide good link predictions. That is, SpectralLink outperforms Two-way Ncut because the latter relies only on the second eigenvector,

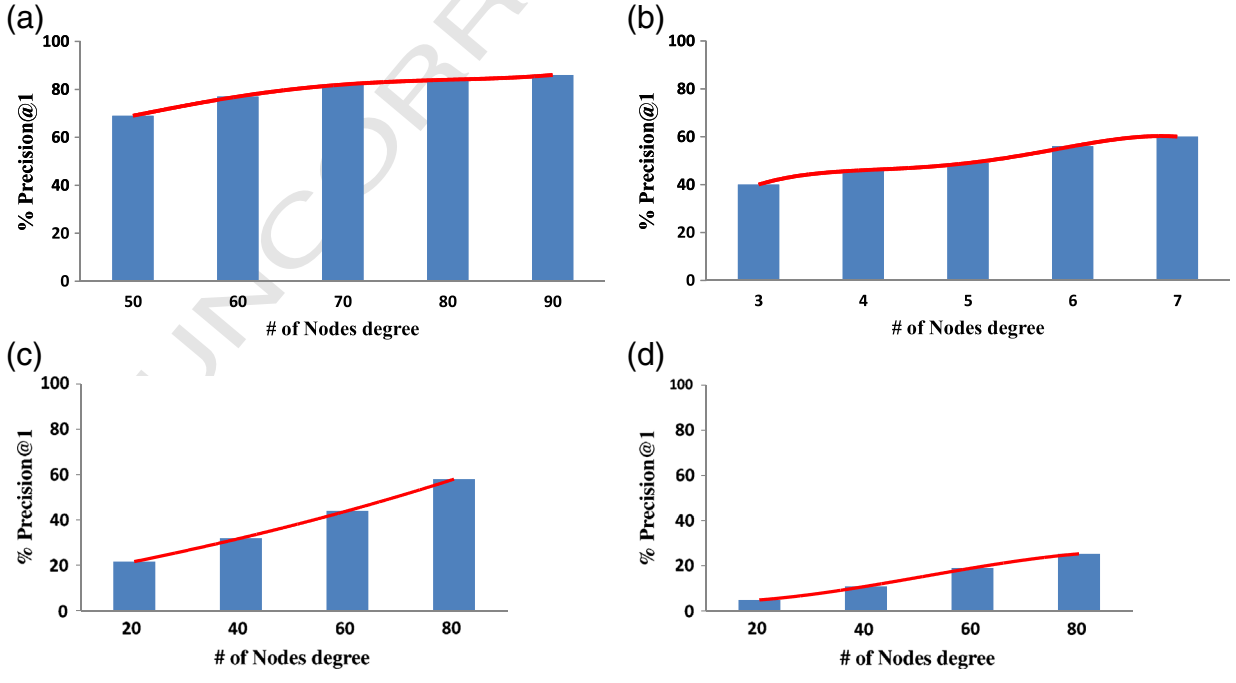

Fig. 6. Precision diagram presenting the increase in precision when a larger amount of neighbors is known for a data set. The data sets represented are: (a) Synthetic 50K, (b) Facebook 3.7K, (c) Human and (d) Human Disease.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

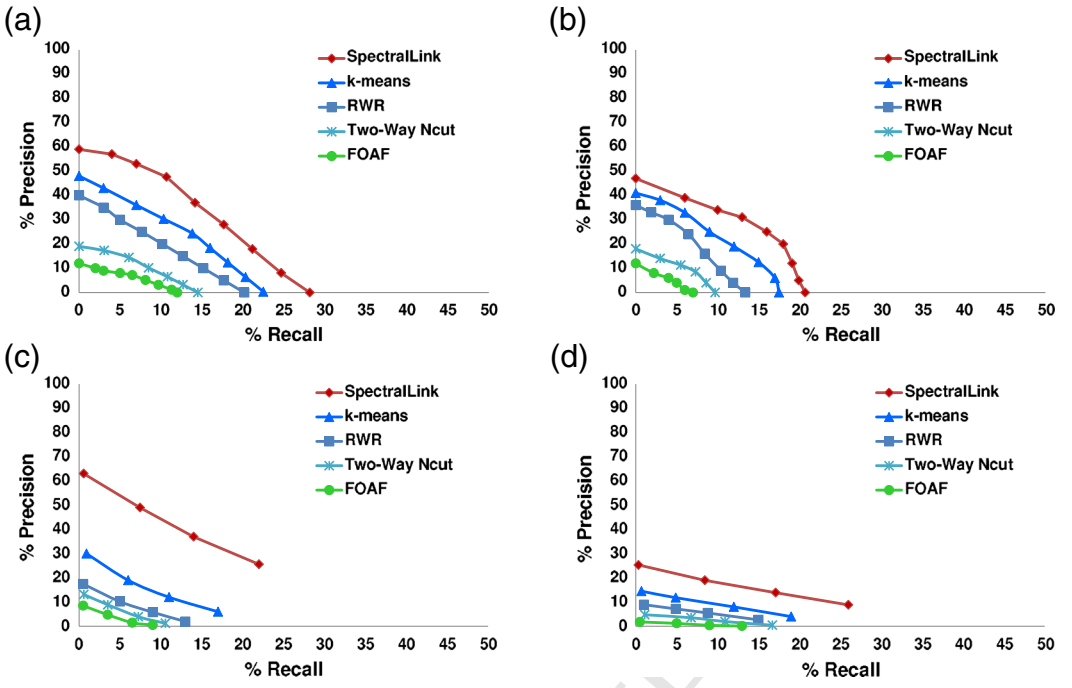

Fig. 7. Comparison of SpectralLink, *k*-means, RWR, Two-Way Ncut and FOAF algorithms for the: (a) Facebook 3.7K, (b) Hi5 63K, (c) Human and (d) Human Disease data sets.

cutting the subsequent eigenvectors, which might be perfect partitioning vectors. RWR traverses globally the social/protein network, failing to capture adequately the local characteristics of the graph. FOAF cannot provide accurate recommendations because it exploits only length-2 paths, failing to capture the notion of the global characteristic of a graph.

We also plot a precision versus recall diagram for the Hi5 63K data set, depicted in Fig. 7. The precision of SpectralLink is decreased in this specific data set. The main reason is the topological characteristics of Hi5 63K data set (i.e. high ASD = 7.18 and small ADEG = 2.78) [45]. Based on these characteristics, Hi5 63K cannot be considered as a small-world network. Thus, it is not well-connected and results to lower recommendation accuracy.

Finally, we plot a precision versus recall curve for each Human Protein data set as shown in Fig. 7c and d. The recall and precision vary as we increase the number of recommended proteins. These experiments show that SpectralLink and *k*-means are more robust in predicting relevant proteins and the reason is that SpectralLink and *k*-means, identify clusters with high within-cluster nodes similarity and low between-cluster similarity. Thus, the high within-cluster node similarity captures effectively the notion of the local characteristics of a graph, whereas the low between-cluster dissimilarity captures effectively the notion of the global characteristics of a graph.

As already mentioned in Section 5.2 the Human data set consists a small-world network, while the Human Disease data-set does not possess this property. This is why it results to low recommendation accuracy.

5.6. Time comparison of SpectralLink with other methods

In this section, we compare SpectralLink, against *k*-means, RWR, two-way Ncut and FOAF algorithms in terms of efficiency using two synthetic, two real social and three real human protein data sets. We have created 2 synthetic data sets based on different network sizes *n* (50,000, 100,000), by keeping an *m* node degree equal to 50 for all data sets. Then, we measured the clock time for the

t10.1 **Table 10**
t10.2 Time performance (in s) of RWR, *k*-means, SpectralLink, FOAF and Two-Way Ncut algorithms for all data sets.

t10.3		Algorithms				
t10.4	Data-set	RWR	*k*-means	SpectralLink	FOAF	Two-Way Ncut
t10.5	Hi5 63K	1.106	0.745	0.562	0.179	0.516
t10.6	Facebook 3.7K	0.135	0.105	0.085	0.029	0.058
t10.7	Human	0.542	0.391	0.153	0.028	0.109
t10.8	Human Disease	0.438	0.312	0.129	0.021	0.068
t10.9	Human Signaling	0.496	0.364	0.138	0.025	0.083

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

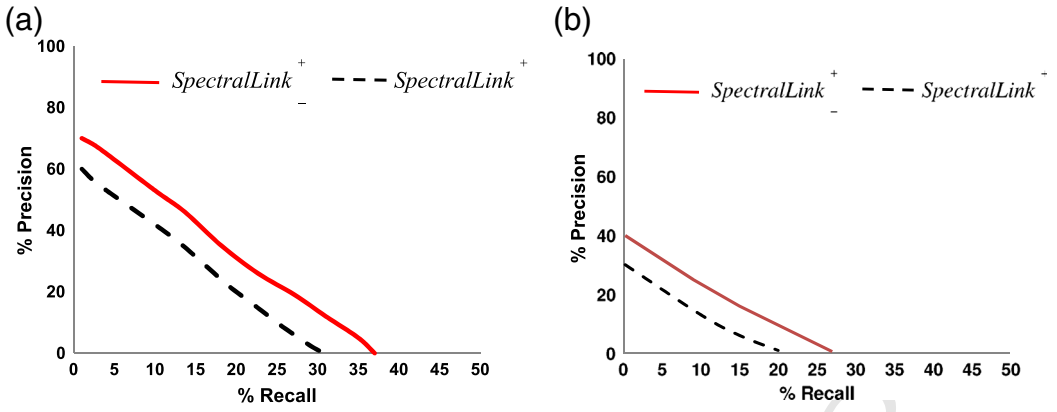

Fig. 8. Accuracy performance of SpectralLink in terms of precision/recall in: (a) Epinions 132K and (b) Human Signaling data sets.

off-line parts of all algorithms. The off-line part refers to the average computation time for calculating the similarities for a target node. The results are presented in Table 10.

As shown, SpectralLink outperforms RWR, which presents the worst time complexity because it calculates the inverse of an $n \times n$ matrix, whereas SpectralLink performs calculations on the decomposed normalized Laplacian matrix. Moreover, SpectralLink outperforms *k*-means because it requires only the top few eigenvectors for graph partitioning. Furthermore, SpectralLink outperforms two-way Ncut because it is computationally wasteful, since it is a recursive algorithm and only the second eigenvector is used in each bipartition.

As expected, FOAF outperforms the other algorithms due to its simpler complexity. However, as already shown in Section 5.5, FOAF performs the worst results in terms of accuracy prediction. This means, that it is not suitable for the link prediction task, even if FOAF presents small time complexity. Finally, notice that these experimental results correspond to the algorithms' complexities, as previously discussed in Section 4.4.

5.7. SpectralLink accuracy in signed networks

In this section, we present the accuracy performance of SpectralLink when we take into account positive and negative links of a signed network, i.e. Epinions 132K data set. We have two different variants of SpectralLink: The first variation considers only positive links and is denoted as *SpectralLink*$^{+}$. The second variation considers both positive and negative links and is denoted as *SpectralLink*$^{\pm}$. Fig. 8 presents the precision and recall diagram for both versions of SpectralLink. As shown, *SpectralLink*$^{\pm}$ outperforms *SpectralLink*$^{+}$. The reason is that *SpectralLink*$^{\pm}$ exploits positive and negative links. This means that if we use information about negative edges for predicting the presence of positive edges we get an accuracy improvement of SpectralLink predictions. These results clearly demonstrate that there is, in some settings, a significant improvement to be gained by using information about negative edges, even to predict the presence or absence of positive edges.

6. Discussion

There are many difficulties in the study of the link prediction problem. A basic problem is the data sparsity [37] of OSNs/PPINs. That is, the prior probability of a link is typically quite small for building a statistical model. To overcome this limitation, we studied a synthetic network model with controllable density.

Real networks have many complex structural properties [7], such as degree heterogeneity, the rich-club phenomenon, the mixing pattern, etc. These network properties are not considered by our synthetic network model, since they are out of the scope of this paper. However, our synthetic network model can be easily extended to better resemble real networks. For example, by applying the degree heterogeneity index [7] with a probability *p*, a synthetic network with different level of degree heterogeneity can be composed.

This paper concerns unweighted and undirected networks. However, our algorithm can be easily extended to more general cases. For example, we can handle the directed networks by replacing the original adjacency matrix *A* by an asymmetric one. Also, this paper concerns the prediction problem in static networks. In reality, many networks are continuously evolving, and the links created in different times should be assigned with different weights. Our algorithm could deal with weighted networks by replacing *A* by a weighted matrix.

7. Conclusions

In this paper, we introduced a framework that uses an enhanced multi-way spectral clustering method, which is based on triangular inequality to measure node proximity in OSNs/PPINs. We compared our method with previous related work, *k*-means,

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

two-way Ncut spectral clustering algorithm, and other well-known link prediction algorithms, using two synthetic, three real social networks and three real human protein data sets. We have shown that our SpectralLink algorithm provides more accurate and faster link predictions. In future, we intend to improve link prediction by combining unipartite with bipartite social/biological networks. Bipartite social networks can also provide valuable information by also exploiting users' co-commenting on written posts, co-rating products and co-participating in groups. Bipartite protein–gene networks can provide valuable information based on the information of proteins with genes interactions.

References

[1] Z. Abbassi, V. Mirrokni, A recommender system based on local random walks and spectral methods, Proceedings Workshop on Knowledge Discovery on the Web (WebKDD) in conjuction with the 1st International Workshop on Social Networks Analysis (SNA-KDD), 2007, pp. 139–153, (Philadelphia, PA).
[2] L. Adamic, E. Adar, How to search a social network, Social Networks 27 (3) (2005) 187–203.
[3] J.R. Bray, J.T. Curtis, An ordination of the upland forest communities of Southern Wisconsin, Ecological Monographs 27 (4) (1957) 325–349.
[4] S. Brohee, J. van Helden, Evaluation of clustering algorithms for protein–protein interaction networks, BMC Bioinformatics 7 (2006) 488–506.
[5] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, B. Jacq, Functional classification of proteins for the prediction of cellular function from a protein–protein interaction network, Genome Biology 5 (2003) R6.1–R6.13.
[6] J. Chen, W. Geyer, C. Dugan, M. Muller, I. Guy, Make new friends, but keep the old: recommending people on social networking sites, Proceedings 27th International Conference on Human factors in Computing Systems (CHI), 2009, pp. 201–210, (Boston, MA).
[7] L. Costa, F. Rodrigues, G. Travieso, P. Boas, Characterization of complex networks: a survey of measurements, Advances in Physics 56 (1) (2007) 167–242.
[8] F. Fouss, A. Pirotte, J.M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, IEEE Transactions on Knowledge and Data Engineering 19 (3) (2007) 355–369.
[9] K. Goh, M.E. Cusick, D. Valle, B. Childs, M. Vidal, A.L. Barabasi, The human disease network, Proceedings of the National Academy of Sciences 104 (2007) 8685–8690.
[10] J. Golbeck, Personalizing applications through integration of inferred trust values in semantic web-based social networks, Proceedings Semantic Network Analysis Workshop in conjunction with the 4th International Semantic Web Conference (ISWC), Galway, Ireland, 2005.
[11] G. Golub, C. Van Loan, Matrix computations, 1983.
[12] I. Guy, I. Ronen, E. Wilcox, Do you know?: recommending people to invite into your social network, Proceedings 13th International Conference on Intelligent User Interfaces (IUI), 2009, pp. 77–86, (Sanibel Island, FL).
[13] D. Higham, G. Kalna, M. Kibble, Spectral clustering and its use in bioinformatics, Journal of Computational and Applied Mathematics 204 (2007) 25–37.
[14] Y. Hou, Bounds for the least Laplacian eigenvalue of a signed graph, Acta Mathematica Sinica 21 (2005) 955–960.
[15] N. Iakovidou, P. Symeonidis, Y. Manolopoulos, Multiway spectral clustering link prediction in protein–protein interaction networks, Proceedings 10th IEEE International Conference on Information Technology and Applications in Biomedecine (ITAB), 2010, pp. 1–4, (Corfu, Greece).
[16] G. Jeh, J. Widom, SimRank: a measure of structural-context similarity, Proceedings 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2002, pp. 538–543, (Edmonton, Canada).
[17] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.
[18] T. Kim, M. Li, K.H. Ryu, J. Shin, Prediction of protein function from protein–protein interaction network by weighted graph mining, Proceedings 4th International Conference on Bioinformatics and Biomedical Technology (ICBBT), 2012, pp. 150–154, (Singapore).
[19] T. Kocher, G. Superti-Furga, Mass spectrometry based functional proteomics: from molecular machines to protein networks, Nature Methods 4 (2007) 807–815.
[20] J. Kunegis, D. Fay, C. Bauckhage, Network growth and the spectral evolution model, Proceedings 19th ACM International Conference on Information and Knowledge Management (CIKM), 2010, pp. 739–748, (New York, USA).
[21] J. Kunegis, A. Lommatzsch, Learning spectral graph transformations for link prediction, Proceedings International Conference in Machine Learning (ICML), 2009, p. 71, (Montreal, Canada).
[22] J. Kunegis, A. Lommatzsch, C. Bauckhage, The Slashdot Zoo: mining a social network with negative edges, Proceedings 18th International Conference on World Wide Web (WWW), 2009, pp. 741–750, (New York, USA).
[23] C. Lei, J. Ruan, A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity, Bioinformatics 29 (3) (2013) 355–364.
[24] D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, Proceedings 12th International Conference on Information and Knowledge Management (CIKM), 2003, pp. 556–559, (New Orleans, LO).
[25] L. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, Journal of the American Society for Information Science and Technology 58 (7) (2007) 1019–1031.
[26] L. Liua, Y. Caic, W. Lua, K. Fenge, C. Penga, B. Niu, Prediction of protein–protein interactions based on PseAA composition and hybrid feature selection, Biochemical and Biophysical Research Communications 380 (2009) 318–322.
[27] L. Lu, C. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Physical Review E 80 (2009) 1–9.
[28] J. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1967, pp. 281–297, (Berkeley, CA).
[29] M. Maila, J. Shi, A random walks view of spectral segmentation, Proceedings International Conference on AI and Statistics (AISTAT), 2001.
[30] E. Marcotte, M. Pellegrini, M. Thomson, T. Yeates, D. Eisenberg, A combined algorithm for genome-wide prediction of protein function, Nature 14 (1999) 849–856.
[31] P. Massa, P. Avesani, Trust-aware collaborative filtering for recommender systems, Proceedings Federated International Conference On The Move to Meaningful Internet (CoopIS, DOA, ODBASE), 2004, pp. 492–508, (Larnaca, Cyprus).
[32] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Advances in Neural Information Processing Systems, 14, 2001, pp. 849–856.
[33] J. Pan, H. Yang, C. Faloutsos, P. Duygulu, Automatic multimedia cross-modal correlation discovery, Proceedings 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2004, pp. 653–658, (Seattle, WA).
[34] A. Papadimitriou, P. Symeonidis, Y. Manolopoulos, Friendlink: link prediction in social networks via bounded local path traversal, Proceedings International Conference on Computational Aspects of Social Networks (CASoN), 2011, pp. 66–71, (Salamanca, Spain).
[35] F. Pazos, A. Valencia, Similarity of phylogenetic trees as indicator of protein–protein interaction, Protein Engineering 14 (2001) 609–614.
[36] W. Pentney, M. Meila, Spectral clustering of biological sequence data, Proceedings 12th National Conference on Artificial Intelligence, 2005, pp. 845–850.
[37] M. Rattigan, D. Jensen, The case for anomalous link discovery, ACM SIGKDD Explorations 7 (2) (2005) 41–47.
[38] J. Shi, J. Malik, Normalized cuts and image segmentation, Proceedings Conference on Computer Vision and Pattern Recognition (CVPR), 1997, p. 731, (San Juan, Puerto Rico).
[39] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[40] U. Stelzl, U. Worm, M.e.a. Lalowski, A human protein–protein interaction network: a resource for annotating the proteome, Cell 122 (2005) 957–968.
[41] S. Strogatz, Exploring complex networks, Nature 410 (2001) 268–276.
[42] P. Symeonidis, E. Tiakas, Y. Manolopoulos, Transitive node similarity for link prediction in social networks with positive and negative links, Proceedings 4th ACM Conference on Recommender Systems (RecSys), 2010, pp. 183–190, (Barcelona, Spain).
[43] H. Tong, C. Faloutsos, J. Pan, Fast random walk with restart and its applications, Proceedings 6th International Conference on Data Mining (ICDM), 2006, pp. 613–622, (Hong Kong, China).

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008

[44] K. Voevodski, S. Teng, Y. Xia, Finding local communities in protein networks, BMC Bioinformatics 10 (2009) 297–310.
[45] D. Watts, S. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1997) 440–442.
[46] Y. Yamanishi, J. Vert, M. Kanehisa, Protein network inference from multiple genomic data: a supervised approach, Bioinformatics 20 (2004) i363–i370.
[47] D. Yan, L. Huang, M. Jordan, Fast approximate spectral clustering, Proceedings 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 907–916, (Paris, France).
[48] L. Yen, F. Fouss, C. Decaestecker, P. Francq, M. Saerens, Graph nodes clustering with the sigmoid commute-time kernel: a comparative study, Data and Knowledge Engineering 68 (3) (2009) 338–361.
[49] T. Zhou, L. Lu, Y. Zhang, Predicting missing links via local information, The European Physical Journal B 71 (4) (2009) 623–630.

Panagiotis Symeonidis received his bachelor degree in applied informatics in 1996, and the MSc degree in information systems in 2004, from Macedonia University, Greece. He received the PhD degree in web mining from Aristotle University of Thessaloniki, Greece, in 2008. Currently, he is working as a postdoctoral researcher at Aristotle University of Thessaloniki, Greece. He is the coauthor of more than 35 articles in international journals and conference proceedings. His articles have received more than 450 citations from other scientific publications. He teaches courses on databases, data mining and data warehousing in a postgraduate program in Aristotle University of Thessaloniki. He is also the director of 1st SEK (School Laboratory Center) of Stavroupolis, Thessaloniki. His research interests include web mining (usage mining, content mining and graph mining), information retrieval and filtering, recommender systems, social media in Web 2.0 and online social networks.

Nantia Iakovidou was born in Greece, in 1983. She received a B.Sc. in Computer Science (2005) and a M.Sc. in Information Systems (2007) from the Department of Informatics, Aristotle University of Thessaloniki, Greece. Currently, she is a PhD Student at Department of Informatics (Aristotle University) and her research interests include: data mining in bioinformatics, string matching in biomedical data, discovery of graph patterns from brain networks.

Nikolaos Mantas received in 2008 his BSc in Computer Science from the Department of Informatics, Aristotle University of Thessaloniki, Greece. In 2010, he received the MSc in Informatics and Management from the Department of Informatics, Aristotle University of Thessaloniki, Greece. He is currently a PhD. Student in the Department of informatics and Telecommunication Engineering, University of West Macedonia.

Yannis Manolopoulos was born in Greece. He received a Diploma (5-year) in Electrical Engineering, Aristotle University (1981) and a Doctorate degree in Computer Engineering from Aristotle University (1986). He is now a Professor in Informatics Department of Aristotle University, Greece. Also he is a staff member at the Aristotle University of Thessaloniki since 1982, Visiting Researcher at the Computer Science Department, University of Toronto, 1984-1985, Visiting Associate Professor at the Computer Science Department, University of Maryland at College Park, 1993-1994 and Visiting Professor at the Computer Science Department, University of Cyprus, 2000-2001. He has published over 280 papers in journals and conference proceedings and he has received over 6000 citations from over 800 institutional groups.

Please cite this article as: P. Symeonidis, et al., From biological to social networks: Link prediction based on multi-way spectral clustering, Data & Knowledge Engineering (2013), http://dx.doi.org/10.1016/j.datak.2013.05.008