Directed Graph Learning via High-Order Co-linkage Analysis

Hua Wang, Chris Ding, and Heng Huang

Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, USA
huawang2007@mavs.uta.edu, chqding@uta.edu, heng@uta.edu

Abstract. Many real world applications can be naturally formulated as a directed graph learning problem. How to extract the directed link structures of a graph and use labeled vertices are the key issues to infer labels of the remaining unlabeled vertices. However, directed graph learning is not well studied in data mining and machine learning areas. In this paper, we propose a novel Co-linkage Analysis (CA) method to process directed graphs in an undirected way with the directional information preserved. On the induced undirected graph, we use a Green's function approach to solve the semi-supervised learning problem. We present a new zero-mode free Laplacian which is invertible. This leads to an Improved Green's Function (IGF) method to solve the classification problem, which is also extended to deal with multi-label classification problems. Promising results in extensive experimental evaluations on real data sets have demonstrated the effectiveness of our approach.

1 Introduction

Different from undirected graphs, which only characterize symmetric pairwise similarity between data objects, directed graphs take into account edge directionality. This additional link structure usually brings useful information, though it makes learning on a directed graph more challenging. As a result, in contrast to a large number of classification methods devised for undirected graphs, classification on directed graphs has been much less studied [29]. In this work, we explore this area and solve the problem to classify unlabeled data on a directed graph by leveraging directed link structures when partially labeled data are given.

Directed graph appears extensively in diverse real world applications. Typical examples of classification on directed graphs include web page categorization [12] and spam host identification [1] on hyperlink networks, document classification or recommendation on citation graphs [10], and many practical problems in other domains such as computational biology [17,15]. Besides these natural real world directed networks, asymmetric pairwise similarities between data objects also generate directed graphs, *e.g.*, the immediate outputs of widely used k-Nearest Neighbor (k-NN) graph construction method [11] and recently proposed sparse representation based graph construction methods [5,25].

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part III, LNAI 6323, pp. 451–466, 2010.

Because most existing graph-based semi-supervised classification algorithms only deal with undirected graphs, directed graphs are routinely converted to undirected ones via symmetrization in different ways prior to usage. For instance, when constructing a k-NN graph [11], an edge is placed between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ when one of them is among the k nearest neighbors of the other one. However, in reality, $\mathbf{x}_i$ is not necessary to be among the k nearest neighbors of $\mathbf{x}_j$, when $\mathbf{x}_j$ is among the k nearest neighbors of $\mathbf{x}_i$. Such symmetrization treatments [11,1,5,25] indeed simply discard the important structural information conveyed by edge directions, which inevitably impair the efficacy of subsequent classifications. For example, it is almost impossible to detect spam host without taking into consideration hyperlink direction — the main mechanism for web spam identification [1]: spam hosts frequently link to genuine hosts, while genuine hosts are rarely observed to link to spam ones. Therefore, there is a great need to develop directed graph based semi-supervised learning algorithms to make use of edge directionality of an input directed graph.

In this work, we focus on semi-supervised learning on a directed graph which classifies unlabeled vertices on a directed graph with partially labeled vertices. Our approach consists of two following steps.

Firstly, we provide an in-depth co-linkage analysis on co-citation and co-reference linkages at second, third and fourth orders. This leads to a novel Co-linkage Analysis (CA) similarity to process a directed graph in an undirected way with the directional information preserved. We also emphasize the importance of link normalization and refine CA similarity by symmetrically normalizing both in-links and out-links in a balanced manner. Once the symmetric pairwise similarity are obtained through this co-linkage analysis process, existing graph based semi-supervised learning methods can be employed.

Secondly, we further develop the Green's function learning framework [8], and present an Improved Green's Function (IGF) method to classify unlabeled data on the induced graph via CA similarity. Here we solve the problem caused by the zero-mode of the combinatorial Laplacian of an input graph. In addition, by incorporating label correlations through the kernel regularization framework derived from the theory of reproducing kernel Hilbert space (RKHS) [23], IGF method is extended to deal with multi-label data.

Related works. Due to the broad usage of directed graphs in numerous real applications, directed graph learning has attracted increasing attention in recent years. F. Chung [6] defined the combinatorial Laplacian of a directed graph, which laid foundation for label propagation on a directed graph. Zhou *et al*. [30] generalized their earlier work [28] for semi-supervised learning on undirected graphs to that on directed graphs by discriminatively normalizing in-links and out-links. They also proposed another method [29] upon the same intuition, in which the regularization on a directed graph has a similar form to the combinatorial Laplacian of a directed graph defined in [6]. Shin *et al*. considered learning on an artificial directed graph derived from an undirected graph through an interesting method — "graph sharpening" [18], which removes the direction from an unlabeled datum to a labeled one on all edges. Besides label propagation,

various other mechanisms have also been used to devise learning methods on a directed graph to take advantage of its asymmetric nature [17,15,31,1,27].

Notations. Pairwise similarities between data objects are usually described as an undirected graph $\mathcal{G}^u$ with a *symmetric* weight matrix $W \in \mathbb{R}^{n\times n}$. $D = \text{diag}\,(W\mathbf{e})$, where $\mathbf{e} = \{1,\ldots,1\}^T$, and $(D-W)$ is the graph Laplacian.

Suppose $\mathcal{G}^d = (\mathcal{V},\mathcal{E})$ is an unweighted directed graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$. $\mathcal{G}^d$ is described by an *asymmetric* adjacency matrix $L = \{0,1\}^{n\times n}$, such that $|\mathcal{V}| = n$, and $L_{ij} = 1$ if there exists an edge $i \to j$ from vertex i to vertex j, and $L_{ij} = 0$ otherwise. The edge $i \to j$ is an ordered pair, and we say j is the *out-neighbor* of i, or i is the *in-neighbor* of j. The number of out-neighbors of i is the *out-degree* of i, given by $d^i_{\text{out}} = \sum_k L_{ik}$. Similarly, the number of in-neighbors of j is the *in-degree* of j, given by $d^j_{\text{in}} = \sum_k L_{kj}$. Let D_{out} be a diagonal matrix and $D_{\text{out}}\,(i,i) = d^i_{\text{out}}$, and D_{in} be a diagonal matrix and $D_{\text{in}}\,(i,i) = d^i_{\text{in}}$. When $i \to i \in \mathcal{E}$, the edge is called as a *loop*. A graph is *simple* if it has no loop. In this work, we only consider simple directed graphs, which are also strongly connected and aperiodic [2].

A weighted directed graph is described by a weight matrix $R \in \mathbb{R}^{n\times n}$ when there exists a function $r : \mathcal{E} \to \mathbb{R}^+$, which associates a nonnegative value $R_{i\to j}$ with every edge $i \to j \in \mathcal{E}$. Here we use R for directed graph to distinguish from W for undirected graph. An unweighted directed graph is a special case of weighted directed graphs when $R = L$. For a weighted directed graph, the out-degree is defined as $d^i_{\text{out}} = \sum_k R_{ik}$, and the in-degree is defined as $d^i_{\text{in}} = \sum_k R_{kj}$.

When it is clear from context, we use W and $\mathcal{G}^u$ interchangeably, and the same for R (or L) and $\mathcal{G}^d$.

2 Challenges of Semi-supervised Learning on A Directed Graph

The semi-supervised learning problem on a directed graph is as following. On a small subset of the vertices, the class labels are known. The task is to classify the rest vertices on the graph.

On an undirected graph, this problem is easy to understand. However, on a directed graph, this problem can be very intriguing. A semi-supervised learning problem on a simple unweighted directed graph is shown in Fig. 1(a). On this graph, the final class labels on the unlabeled vertices are not obvious. Fig. 1 illustrates three possible solutions.

Using nearest neighbor classification. If we use the nearest neighbor classification (NNC), the results are shown in Fig. 1(b). The NNC algorithm is the following iterative algorithm. It computes the label $(y_1,\cdots,y_n)$ on all unlabeled vertices with y_i fixed to their signs on all labeled vertices while $y_j^{(t=0)} = 0$ for all unlabeled vertices. We iterate with $y_j^{(t+1)} = \text{sign}\left(\sum_i L_{ij} y_i^{(t)}\right)$ until convergence. Vertex f will be labeled as "$-$" due to the the incoming neighbor a. Vertex e will be labeled as "$-$" due to the the incoming neighbor f. Repeating this, vertices d and c will be labeled as "$-$".

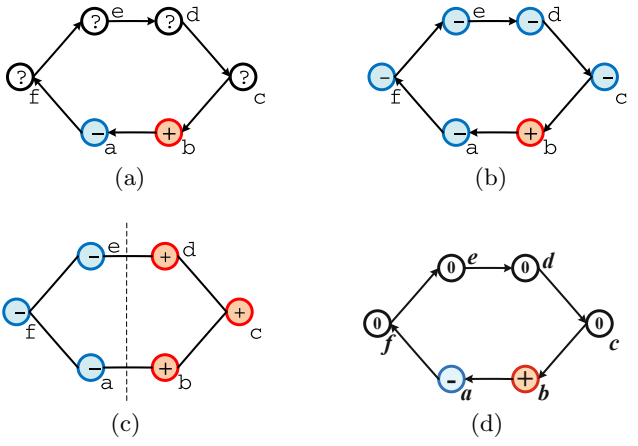


Fig. 1. (a) Semi-supervised learning on a simple directed graph. Vertex a is positively labeled and vertex b is negatively labeled. The task is to classify the rest vertices.
(b) Solution of the problem in (a) via nearest neighbor method.
(c) Solution of the problem in (a) via symmetrization and label propagation method.
(d) Solution of the problem in (a) via random walk method.

Using symmetrization. If we symmetrize the directed graph into an undirected graph by $W = L + L^T$, the results are shown in Fig. 1(c). In this case, the problem becomes the semi-supervised learning on an undirected graph. It is now obvious that the final class labels are assigned as shown in Fig. 1(c).

Using random walk. If we use information propagation via random walks, the results are shown in Fig. 1(d), *i.e.*, class labels on the unlabeled vertices are undetermined. The reason is as following. A random walker starting from vertex a will carry negative class information. This walker will walk to vertex f with probability 1. It then will walk to vertex e with probability 1, *etc*. At time tends to infinity, this walker will reach all vertices with equal probability of 1/6, passing on a negative label.

On the other hand, a random walker starting from vertex b will carry positive class information. It will visit each vertex with 1/6 probability as time tends to infinity, passing on a positive label. Thus on each unlabeled vertex, the probability of positive label is equal to the probability of negative label. Therefore, the final labeling is undetermined.

Note that the situation will be very different if the graph is **undirected** as shown in Fig. 1(c). On the undirected graph, the random walker starting from vertex a (call it walker-a) will have a higher probability reaching f than reaching e, because after reaching f, instead of going to e (as required by the directed graph), it has the choice of **walking back** to a. Thus the farther-away from a, the smaller probability walker-a will reach. The same holds for the random walker starting from vertex b (call it walker-b). Therefore, the probability for walker-a reaching f is higher than the probability for walker-b reaching f, leading to a "$-$" label for f.

Challenges of learning on a directed graph The above discussions show that semi-supervised learning on a directed graph is rather intriguing. Different approaches lead to very different results (while on an undirected graph, different approaches lead to the same results). Our analysis also shows that simple symmetrization of the adjacency matrix (link matrix L), *i.e.*, $W = L + L^T$, loses critical information and results in very different outcomes.

We point out without elaboration that unsupervised learning such as clustering on a directed graph also has very similar intriguing problems. In general, research on directed graphs learning is lacking.

In this paper, we attempt to solve this learning problem by building a symmetric pairwise similarity from a directed graph. Once this symmetric similarity is constructed, the problem becomes learning on an undirected graph, and we may solve the problem using any existing algorithm for undirected graphs.

3 Co-linkage Analysis of A Directed Graph

In this section, we propose a novel Co-linkage Analysis (CA) method to process a directed graph in an undirected way. We first study the two fundamental co-linkages: co-citation and co-reference [9,7], and extend them to higher orders. Then we emphasize the importance of edge weight normalization. In our previous work [24], we use only second-order processes to describe a directed graph. In this work, we induce a symmetric similarity from a directed graph using both second-order co-linkages and their high-order extensions.

3.1 Pairwise Similarity via Co-linkage Analysis

Second-order co-citation and co-reference processes. On a directed graph, we consider the following two second-order fundamental processes: *co-citation* [19] as shown in Fig. 2(a) and *co-reference* [13] as shown in Fig. 2(b).

If two vertices i and j are co-cited by many other vertices, such as vertex k in Fig. 2(a), i and j are likely to be related in some sense. Thus co-citation is a similarity measure and defined as the number of vertices that co-cite i and j:

$$W_{ij}^{(c)} = \sum_k L_{ki} L_{kj} = \left(L^T L\right)_{ij} \quad . \tag{1}$$

On the other hand, if two vertices i and j co-reference several other vertices, such as vertex k in Fig. 2(b), i and j are supposed to have certain commonality. Co-reference also measures similarity between vertices:

$$W_{ij}^{(r)} = \sum_k L_{ik} L_{jk} = (LL^T)_{ij} \quad . \tag{2}$$

Combining $W^{(c)}$ and $W^{(r)}$, we define the second-order similarity as:

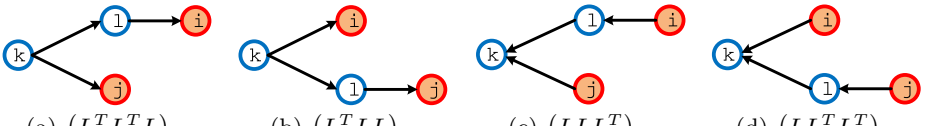
$$W^{(\mathrm{2nd})} = L^T L + LL^T, \tag{3}$$

where we assume co-citation and co-reference are equally important.

(a) Co-citation (b) Co-reference

Fig. 2. Two fundamental second-order processes on a directed graph

Third-order co-citation and co-reference processes. Now we extend the co-citation and co-reference processes to the third-order. Specifically, for the co-citation between vertices i and j with respect to vertex k as in Fig. 2(a), an intermediate vertex can be inserted between k and i as in Fig. 3(a) or between k and j as in Fig. 3(b). We call them as *third-order co-citations*. Similarly, *third-order co-references* are defined as in Fig. 3(c) and Fig. 3(d). Same as the original second-order co-citation and co-reference, they also measure the similarities between vertices i and j.



(a) $\left(L^T L^T L\right)_{ij}$ (b) $\left(L^T LL\right)_{ij}$ (c) $\left(LLL^T\right)_{ij}$ (d) $\left(LL^T L^T\right)_{ij}$

Fig. 3. Third-order processes on a directed graph. (a)—(b): third-order co-citation; (c)—(d): third-order co-reference.

For the third-order co-citation in Fig. 3(a), the similarity between vertices i and j can be easily counted by $\sum_k \sum_l L_{li} L_{kl} L_{kj} = \left(L^T L^T L\right)_{ij}$. Following the same way for the rest three processes, the third-order similarity is defined as:

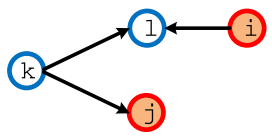
$$\begin{aligned} W^{(3\text{rd})} &= L^T L^T L + L^T LL + LLL^T + LL^T L^T \\ &= L\left(L + L^T\right) L^T + L^T \left(L + L^T\right) L, \end{aligned} \tag{4}$$

where we assume the four third-order processes in Fig. 3 are equally important.

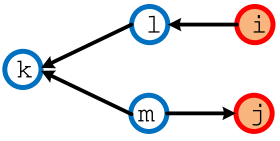
Note that, on a directed graph, other third-order processes also exist, such as the one shown in Fig. 4(a). However, because this process forms neither co-citation nor co-reference, it is not taken into account.

Fourth-order co-citation and co-reference processes. We further extend the co-citation and co-reference processes to the fourth-order, which are illustrated in Fig. 5. Again, we do not consider the processes not forming either co-citation or co-reference such as the one shown in Fig. 4(b). Thus, the fourth-order similarity is defined as:

$$\begin{aligned} W^{(4\text{th})} &= L^T LLL + L^T L^T L^T L + L^T L^T LL + LLLL^T + LL^T L^T L^T + LLL^T L^T \\ &= L\left(LL + L^T L^T + LL^T\right) L^T + L^T \left(LL + L^T L^T + L^T L\right) L \ . \end{aligned} \tag{5}$$

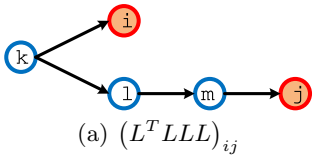


(a) Invalid 3rd-order process: $\left(LL^TL\right)_{ij}$

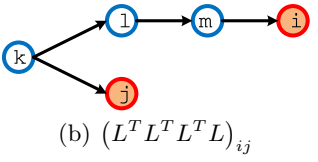


(b) Invalid 4th-order process: $\left(LLL^TL\right)_{ij}$

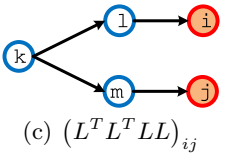
Fig. 4. Invalid third-order and fourth-order processes on a directed graph



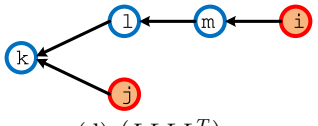
(a) $\left(L^TLLL\right)_{ij}$



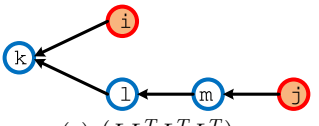
(b) $\left(L^TL^TL^TL\right)_{ij}$



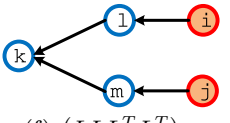
(c) $\left(L^TL^TLL\right)_{ij}$



(d) $\left(LLLL^T\right)_{ij}$



(e) $\left(LL^TL^TL^T\right)_{ij}$



(f) $\left(LLL^TL^T\right)_{ij}$

Fig. 5. Fourth-order processes on a directed graph. (a)—(c): fourth-order co-citation; (d)—(e): fourth-order co-reference.

Combining $W^{(2nd)}$, $W^{(3rd)}$ and $W^{(4th)}$, we obtain the proposed Co-linkage Analysis (CA) similarity as following:

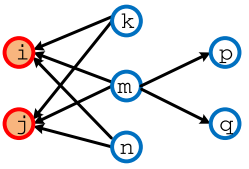
$$W = W^{(2nd)} + \mu W^{(3rd)} + \nu W^{(4th)}, \tag{6}$$

where μ and ν are the parameters to balance the relative importance of the third-order and fourth-order similarities, which are empirically selected as $\mu = \left(\sum_{i\neq j} W_{ij}^{(2nd)}\right) / \left(\sum_{i\neq j} W_{ij}^{(3rd)}\right)$ and $\nu = \left(\sum_{i\neq j} W_{ij}^{(2nd)}\right) / \left(\sum_{i\neq j} W_{ij}^{(4th)}\right)$.

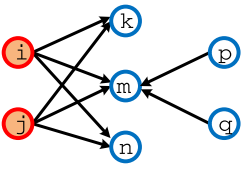
3.2 Link Normalization

On the web, a vertex/web page with bigger out-degree has greater influence than another one with smaller out-degree. However, since these out-links can be arbitrarily added by the web page designer, and the importance of this web page can be arbitrarily increased.

In PageRank algorithm, every out-going hyperlinks from a vertex is inversely weighted by its out-degree, thereby every vertex has the same total out-going weight. This can be stated as *Internet Democracy* : every web site has a total of one vote. The hyperlink normalization and its importance are illustrated in Fig. 6(a). Basically, if a web page has a large out-degree, the significance/uniqueness of its co-citation is reduced. This points the necessity of out-degree normalization.



(a) Out-degree normalization.



(b) In-degree normalization.

Fig. 6. Importance of link normalization. (a): vertices i and j are co-cited by vertices k, m and n. However, since vertex m also cites vertices p and q, the co-citation of i and j by m is not as significant as that by either k or n. This fact can be compensated by normalizing the weights on the out-bound links of a vertex, *i.e.*, the co-citation of i and j by m is then $2/4 = 50\%$ as important as that by either k or n. (b): vertices i and j co-reference vertices k, m and n. However, since vertex m is also referenced by p and q, the co-reference of i and j by m is not as significant as that to either k or n. This fact can be similarly compensated by normalizing the in-bound links of a vertex.

Generally speaking, the in-degree of a document is not easily manipulated and is therefore a good indicator of the importance of the web page. But, when counting co-reference between two web pages as in Fig. 6(b) as similarity between the web pages, in-degree should also be normalized, because a web page i with large in-degree lose the specificity of the those web pages pointing to i.

With these discussions, the reasonable choices of link normalizations are:

$$L \to D_{\text{out}}^{-1} L, \tag{7}$$

$$L \to L D_{\text{in}}^{-1}, \tag{8}$$

$$L \to D_{\text{out}}^{-1/2} L D_{\text{in}}^{-1/2} \ . \tag{9}$$

Normalization of Eq. (7) uses the out-degree and is used in the PageRank algorithm [3,16], which is essentially the transition probability of a random walk. Normalization using out-degree is related to the concept of co-citation since co-citation uses out-links from those web pages/vertices pointing to them. Normalization using out-degree will balance the importance of each of these vertices.

Normalization of Eq. (8) uses the in-degree and can be viewed as the transition probability of a random walk on the inverse direction of the directed graph. Normalization using in-degree is related to the concept of co-reference since co-reference uses in-links from those web pages/vertices pointing to them. Normalization using in-degree will balance the importance of each of these vertices.

Normalization of Eq. (9) can be viewed as a compromise between the above two normalizations. This is also symmetric among the in-degree and out-degree. Considering the balance of in-degree and out-degree normalization and the balance among co-citation and co-reference, we adopt this symmetric normalization in our work.

Replacing L in Eq. (3), Eq. (4) and Eq. (5) by the symmetrically normalized $D_{\text{out}}^{-1/2} L D_{\text{in}}^{-1/2}$ defined in Eq. (9), we can compute normalized CA through

Eq. (6), which is used in all our empirical evaluations. When a weighted directed graph is used, L is replaced by R.

4 Semi-supervised Learning via Improved Green's Function Method

With the symmetric CA similarity induced from a directed graph, we may use any existing graph-based semi-supervised learning algorithm for undirected graphs to classify the unlabeled data points. In this paper, we further develop the Green's function learning framework [8], and present a Improved Green's Function (IGF) method for classification. In this method, we solve the problem caused by the zero-mode of the combinatorial Laplacian of an input graph.

4.1 A Brief Review of the Green's Function Learning Framework

Suppose we have $n = n_l + n_u$ data points $\{\mathbf{x}_i\}_{i=1}^{n}$, where the first n_l data points are labeled with $\{\mathbf{y}_i\}_{i=1}^{n_l}$ for K target classes. Here, $\mathbf{x}_i \in \mathbb{R}^p$ and $\mathbf{y}_i \in \{-1, +1\}^K$, such that $\mathbf{y}_i(k) = +1$ if $\mathbf{x}_i$ belongs to the k-th class, and -1 otherwise. Our task is to learn the classification $\{\mathbf{y}_i\}_{i=n_l+1}^{n}$ for the unlabeled data. For the unlabeled data points, we set $\mathbf{y}_i(k) = 0$. We write $Y = [\mathbf{y}_1, \cdots, \mathbf{y}_n]^T$.

Given a graph with edge weight W among the data points $\{\mathbf{x}_i\}_{i=1}^{n}$, we wish to learn the mapping function $F = \mathbb{R}^{n \times K}$ such that $|F - Y|$ is minimized, where $|\cdot|$ stands for the Frobenious norm of a matrix. Adding a penalty (regularization) term to ensure smoothness with respect to the underlying data manifold, the Green's function learning framework minimizes the following objective [8]:

$$J(F) = |F - Y| + \alpha F^T \mathcal{K}^{-1} F, \tag{10}$$

where $\mathcal{K}$ is a kernel in RKHS, and $\mathcal{K}^{-1} = (D - W)$. Here α is a parameter to balance the relative importance of the regularization term.

Taking the derivative of J with respect to F and set it as 0, we obtain $F = [I + \alpha(D - W)]^{-1} Y$. At large α limit, F is computed as following:

$$F = GY = (D - W)^{-1} Y, \tag{11}$$

where $G = (D - W)^{-1}$ is the Green's function of the input graph. However, G is not well defined due the existence of the zero-mode of $(D - W)$.

Let $(D - W)\mathbf{v}_k = \lambda_k \mathbf{v}_k$, where $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of $(D - W)$ and $\mathbf{v}_k$ are the corresponding eigenvectors. Because we consider connected graphs, the first eigenvector is a constant vector $\mathbf{v}_1 = \mathbf{e}/\sqrt{n}$ with zero eigenvalue and multiplicity one. Thus, G is not well defined because $\mathbf{v}_1 \mathbf{v}_1^T / \lambda_1 = \mathbf{e}\mathbf{e}^T / n\lambda_1$. The analysis in [8] shows that this zero-mode of $(D - W)$ is a consequence of the Von Neumann boundary condition (derivatives are continuous at the boundary) and thus the solution is undetermined up to an overall constant.

This overall constant is removed in [8] by explicitly discarding the zero-mode of $(D - W)$ and the Green's function is computed as follows:

$$G = \frac{1}{(D-W)_+} = \sum_{i=2}^{n} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\lambda_i} \ . \tag{12}$$

4.2 Zero-Mode Free Laplacian

In this paper, we propose a zero-mode free Laplacian. The graph Laplacian is usually defined as the embedding of $q_1, \cdots, q_n$ by solving

$$\min_q \frac{1}{2} \sum_{ij} (q_i - q_j)^2 W_{ij}, \ s.t. \ \sum_i q_i^2 = 1, \sum_i q_i = 0 \ . \tag{13}$$

Now, we propose to modify this to the following

$$\min_q \frac{1}{2} \sum_{ij} (q_i - q_j)^2 W_{ij} + \frac{W_{++}}{n^2} (\sum_i q_i)^2, \ s.t. \ \sum_i q_i^2 = 1, \sum_i q_i = 0, \tag{14}$$

where $W_{++} = \sum_{ij} W_{ij}$. Clearly, the optimal solution for Eq. (14) is identical to that for Eq. (13). Note that

$$\frac{1}{2} \sum_{ij} (q_i - q_j)^2 W_{ij} + \frac{W_{++}}{n^2} (\sum_i q_i)^2 = \mathbf{q}^T L_+ \mathbf{q}, \tag{15}$$

where the **zero-mode free Laplacian** L_+ is defined as

$$L_+ = D - W + \frac{W_{++}}{n^2} \mathbf{e}^T \mathbf{e} \ . \tag{16}$$

Some properties of L_+ are:

(1) $\mathbf{v}_1 = \mathbf{e}/n^{1/2}$ is an eigenvector of L_+ with eigenvalue $\lambda_1(L_+) = W_{++}/n$.
(2) L_+ and $L = D - W$ have the same eigenvectors $\mathbf{v}_2, \cdots, \mathbf{v}_n$ with same eigenvalues.
(3) L_+ is **positive definite** and its inverse is well defined.

The new Green's function becomes the following:

$$F = \frac{1}{D - W + \frac{W_{++}}{n^2} E} Y, \tag{17}$$

where $E = \mathbf{e}^T \mathbf{e}$. We call Eq. (17) as Improved Green's Function (IGF) method.

4.3 Kernel Regularized Correlative Multi-label Classification

Multi-label data present a new opportunity to improve classification accuracy through label correlations, which is absent in single-label data. Typically, label correlations of a multi-label data set is captured by a correlation matrix $C \in$

$\mathbb{R}^{K\times K}$, which can be computed as in [23]. Adding a penalty for label correlations to impose smoothness, we minimize the following objective:

$$J(F) = \beta|F - Y|^2 + \mathbf{tr}\left(F^T\mathcal{K}^{-1}F - \gamma\mathcal{K}^{-\frac{1}{2}}FCF^T\mathcal{K}^{-\frac{1}{2}}\right), \tag{18}$$

where $\mathcal{K} = G = \left(D - W + \frac{W_{++}}{n^2}E\right)^{-1}$, β and γ are two small nonnegative constants to balance the two regularization terms.

When $0 < \gamma < \min\{1, 1/\max(\zeta_k)\}$ where $\zeta_k (0 < k < K)$ are the eigenvalues of C, following the same derivation as in [23], the solution to the optimization problem in Eq. (18) when β is small is obtained as:

$$F = GY\left(I - \gamma C\right)^{-1} \quad . \tag{19}$$

We call Eq. (19) as Multi-Label Improved Green's Function (ML-IGF) method, which solves multi-label classification problems.

5 Experiments

We evaluate the effectiveness of the proposed CA similarity, and the classification performances of IGF method on single-label data and ML-IGF method on multi-label data through classification tasks on directed graphs.

Single-label data sets. Because web data naturally generate directed graphs, we use the **WebKB** data set[1] for single-label classification. We consider a subset of the WebKB data set containing the pages from four universities, Cornell, Texas, Washington and Wisconsin, from which we remove the isolated pages, *i.e.*, those have no incoming and outgoing links, resulting in 858, 825, 1195 and 1238 pages respectively, for a total of 4116. These pages have been manually classified into the following seven categories: "student", "faculty", "staff", "department", "course", "project" and "other". We treat the extracted directed graphs as unweighted directed graphs and conduct classification on them.

Multi-label data sets. The following multi-label data sets are used to evaluate multi-label classification performance.

MSRC[2] has 591 images annotated by 22 classes. We divide each image into 64 blocks by a 8×8 grid and compute the first and second moments (mean and variance) of each color band to obtain a 384-dimensional vector as features.

Mediamill [20] includes 43907 sub-shots with 101 classes, where each image is characterized by a 120-dimensional vector. Eliminating the classes containing less than 1000 samples, we have 27 classes. We randomly select 2609 sub-shots such that each class has at least 100 labeled data points.

Music emotion [21] comprises 593 songs with 6 emotions (labels). The dimensionality of the data points is 72.

[1] `http://www-2.cs.cmu.edu/~webkb/`

[2] `http://research.microsoft.com/en-us/projects/objectclassrecognition/default.htm`

Yahoo data described in [22] came from the "yahoo.com" domain. We use the "science" topic as it has maximum number of labels, which contains 6345 web pages with 22 labels.

Because these data sets are supplied in format of feature vectors, we construct directed graphs using k-NN graph construction method. Different from [11], we place a directed edge $i \to j$ if vertex $\mathbf{x}_j$ is a k-Nearest Neighbor of vertex $\mathbf{x}_i$. In our evaluations, we set $k = 3$ ($k = 1$ and $k = 5$ lead to similar experimental results, which are not shown due to space limit).

5.1 Effectiveness of Co-linkage Analysis

We first evaluate the effectiveness of the proposed CA similarity defined in Eq. (6) in processing a directed graph in an undirected way.

A special benefit to use a separate graph construction step lies in that, existing graph-based semi-supervised learning methods can also benefit from the additional information contained in edge directions of a directed graph. Therefore we evaluate the effectiveness of the induced undirected graph by the proposed CA when it is used in the following three representative graph-based semi-supervised learning methods: (1) Gaussian fields and harmonic functions (GFHF) [32] method, (2) local and global consistency (LGC) [28] method, and (3) our previous work, *i.e.*, the Green's function (GF) [8] method. Because these classification methods only work on undirected graphs, given a directed graph L, a simple symmetrization broadly used in existing works is as following: $W_{ij} = 1$ if $L(i \to j) = 1$ or $L(j \to i) = 1$. This graph is denoted as "Symmetrized graph" in Table 1, and compared against the undirected graph induced by the proposed CA which is denoted as "CA graph".

We use the WebKB data set for evaluation. For each category of web pages from each university, a binary classification is conducted, *e.g.*, we classify "student" web pages *vs.* non-student web pages from Cornell university, denoted as "Cornell (student)". Ignoring the "other" category, we perform $4 \times 6 = 24$ binary classifications by every compared classification method. Because web pages within a same university are well-linked, and cross links between different universities are rare, we can imagine that a small number of training samples are sufficient to exactly classify web pages based on only link information. Therefore, in each binary classification, we randomly draw 4 pages as training examples, under the constraint that there is at least one labeled instance for each class. For each binary classification, we repeat 50 independent trials and the average test errors are reported in Table 1.

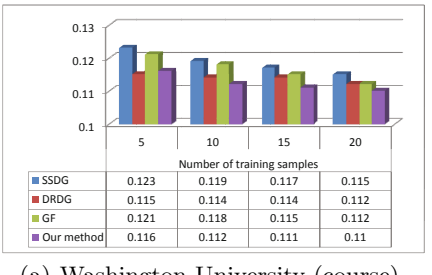
From Table 1 we can see that, the classification performances measured by "test error" on CA graphs always outperform those on symmetrized graphs. Due to space limit, we cannot list all classification results, and pick up one binary classification from each university as in Table 1, which are similar to those not shown. Therefore, we conclude that the proposed CA method is more effective to characterize a directed graph than the simple symmetrization methods that do not consider edge directions.

Table 1. Improved classification performance (test error) of three existing representative graph-based semi-supervised classification methods by using CA graph

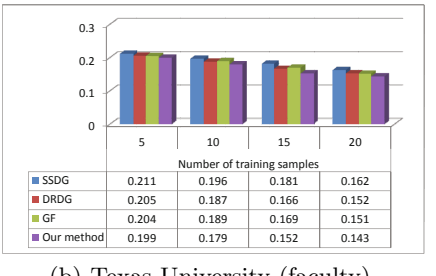
	Cornell (student)			Wisconsin (student)		
	GFHF	LGC	GF	GFHF	LGC	GF
Symmetrized graph	0.246	0.238	0.225	0.207	0.205	0.196
CA graph	0.223	0.212	0.173	0.195	0.191	0.183
	Washington (course)			Texas (faculty)		
	GFHF	LGC	GF	GFHF	LGC	GF
Symmetrized graph	0.142	0.140	0.136	0.228	0.227	0.218
CA graph	0.137	0.135	0.121	0.221	0.215	0.204

5.2 Single-Label Classification Using IGF Method

We evaluate single-label classification performance of IGF method by conducting 2-class classification to distinguish "course" *vs*. non-course web pages in Washington University and "faculty" *vs*. non-faculty web pages in Texas University in WebKB data set. We compare the classification results of our method against two state-of-the-art classification algorithms on directed graphs: (1) Semi-Supervised learning on Directed Graph (SSDG) [30] method, and (2) Distribution Regularized classification on Directed Graph (DRDG) [29] method. We also report the results by the Green's Function (GF) [8] method, where a simple symmetrization of $W = \left(L + L^T\right)/2$ is used to form the undirected graph. The classification performance comparison measured by average test error over 50 independent trials are listed in Fig. 7, which demonstrate the superiority of our method and thereby confirm its usefulness.



(a) Washington University (course).



(b) Texas University (faculty).

Fig. 7. Test errors to classify "course" *vs*. non-course web pages in Washington University and "faculty" *vs*. non-faculty web pages in Texas University in WebKB data set by four compared methods

Table 2. Performance evaluations of the compared methods by 5-fold cross validations

Data sets	Evaluation metrics		Compared methods					
			SSDG	DRDG	MLSI	SMSE	ML-IGF-S	ML-IGF
MSRC	Macro average	Precision	0.215	0.224	0.252	0.248	0.281	**0.311**
		F1 score	0.223	0.238	0.287	0.279	0.288	**0.319**
	Micro average	Precision	0.201	0.223	0.253	0.247	0.279	**0.317**
		F1 score	0.267	0.278	0.301	0.298	0.324	**0.338**
MediaMill	Macro average	Precision	0.201	0.203	0.207	0.210	0.252	**0.274**
		F1 score	0.289	0.292	0.301	0.312	0.352	**0.391**
	Micro average	Precision	0.203	0.206	0.207	0.215	0.259	**0.282**
		F1 score	0.332	0.334	0.341	0.347	0.368	**0.406**
Music emotion	Macro average	Precision	0.313	0.317	0.329	0.331	0.392	**0.404**
		F1 score	0.305	0.308	0.323	0.331	0.399	**0.415**
	Micro average	Precision	0.308	0.311	0.328	0.332	0.395	**0.412**
		F1 score	0.310	0.314	0.339	0.354	0.401	**0.420**
Yahoo (Science)	Macro average	Precision	0.367	0.372	0.396	0.398	0.421	**0.443**
		F1 score	0.278	0.282	0.296	0.305	0.361	**0.379**
	Micro average	Precision	0.369	0.375	0.395	0.402	0.448	**0.470**
		F1 score	0.202	0.203	0.209	0.215	0.236	**0.256**

5.3 Multi-label Classification Using Multi-label IGF Method

We use standard 5-fold cross validation to evaluate multi-label classification performance of ML-IGF method. We empirically selected $\gamma = \min\{0.1, 1/\max(\zeta_k)\}$. We compare our method with (1) SSDG method and (2) DRDG method as in Section 5.2, which, however, are designed for single label classifications. Therefore, for every class, we conduct a binary classification. We also compare our method to two recent multi-label classification methods: (3) Multi-label informed Latent Semantic Indexing (MLSI) [26] method, and (4) Semi-supervised learning by Sylvester Equation (SMSE) [4] method. The classification by these two methods are directly conducted on original data. Because, to our best knowledge, ML-IGF method presented in this work is the first one to exploit the information conveyed by both link directionality and label correlations, we cannot find a counterpart method for comparison.

We also evaluate the effectiveness of link normalization discussed in Section 3.2, and conduct classification using ML-IGF method on the induced graph when no normalization is used. We denote these results as ML-IGF-S in Table 2.

The widely used classification performance metrics in statistical learning, *precision* and *F1 score*, are used to evaluate the compared methods. Precision and F1 score are computed for every class following the standard definitions for a binary classification problem. To address multi-label classification, macro average and micro average are used to assess the overall performance across multiple labels [14].

Table 2 presents the classification performance comparisons by 5-fold cross validation, which show that ML-IGF method generally outperforms all other methods, sometimes significantly. These results quantitatively demonstrate the effectiveness of our method, and justify the utility of the CA similarity and label correlations. Besides, the classification performances of ML-IGF is always better than those of ML-IGF-S method, which provide a concrete evidence that link normalization is an indispensable part of the proposed CA similarity.

6 Conclusions

This paper explored the usage of directed graphs to solve semi-supervised learning problems. We proposed a novel Co-linkage Analysis (CA) method to transform a directed graph to an undirected one, which is built upon the co-linkage processes on directed graphs. With the induced symmetric CA similarity, a Improved Green's Function (IGF) method was presented to solve the classification problem, which is also generalized to deal with multi-label classification problems. Extensive experimental evaluations on real data sets have demonstrated that the performance of the proposed approach outperforms other related previous methods in literature.

Acknowledgments. This research is supported by NSF-CCF 0830780, NSF-CCF 0939187, NSF-CCF 0917274, NSF-DMS 0915228, NSF-CNS 0923494.

References

1. Abernethy, J., Chapelle, O., Castillo, C.: Web spam identification through content and hyperlinks. In: Proc. of International Workshop on Adversarial Information Retrieval on the Web (2008)
2. Bang-Jensen, J.: Digraphs: theory, algorithms and applications. Springer, Heidelberg (2008)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: WWW (1998)
4. Chen, G., Song, Y., Wang, F., Zhang, C.: Semi-supervised Multi-label Learning by Solving a Sylvester Equation. In: SDM (2008)
5. Cheng, H., Liu, Z., Yang, J.: Sparsity Induced Similarity Measure for Label Propagation. In: IEEE ICCV (2009)
6. Chung, F.: Laplacians and the Cheeger inequality for directed graphs. Annals of Combinatorics 9(1), 1–19 (2005)
7. Ding, C., He, X., Husbands, P., Zha, H., Simon, H.: PageRank, HITS and a unified framework for link analysis. In: ACM SIGIR (2002)
8. Ding, C., Simon, H., Jin, R., Li, T.: A learning framework using Green's function and kernel regularization with application to recommender system. In: ACM SIGKDD (2007)
9. Ding, C., Zha, H., He, X., Husbands, P., Simon, H.: Link analysis: hubs and authorities on the World Wide Web. SIAM Review 256 (2004)
10. Giles, C., Bollacker, K., Lawrence, S.: CiteSeer: An automatic citation indexing system. In: Proc. of ACM Conf. on Digital libraries (1998)

11. Hein, M., Maier, M.: Manifold denoising. In: NIPS (2007)
12. Joachims, T., Cristianini, N., Shawe-Taylor, J.: Composite kernels for hypertext categorisation. In: ICML (2001)
13. Kessler, M.: Bibliographic coupling between scientific papers. American documentation 14(1), 10–25 (1963)
14. Lewis, D., Yang, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research (2004)
15. Meila, M., Pentney, W.: Clustering by weighted cuts in directed graphs. In: SDM (2007)
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Stanford Digital Library Technologies Project (1998)
17. Pentney, W., Meila, M.: Spectral clustering of biological sequence data. In: AAAI (2005)
18. Shin, H., Hill, N., Ratsch, G.: Graph based semi-supervised learning with sharper edges. In: ECML (2006)
19. Small, H.: Co-citation in the scientific literature: A new measure of the relationship between two documents. J. Am. Soc. for Info. Sci. Tech. 24(4), 265–269 (1973)
20. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: ACM Multimedia (2006)
21. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: ISMIR
22. Ueda, N., Saito, K.: Single-shot detection of multiple categories of text using parametric mixture models. In: ACM SIGKDD (2002)
23. Wang, H., Huang, H., Ding, C.: Image Annotation Using Multi-label Correlated Greens Function. In: IEEE ICCV (2009)
24. Wang, H., Huang, H., Ding, C.: Image Categorization Using Directed Graphs. In: ECCV (2010)
25. Yan, S., Wang, H.: Semi-supervised learning by sparse representation. In: SDM (2009)
26. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: ACM SIGIR (2005)
27. Zhang, D., Mao, R.: Classifying networked entities with modularity kernels. In: ACM CIKM (2008)
28. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: NIPS (2004)
29. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: ICML (2005)
30. Zhou, D., Schölkopf, B., Hofmann, T.: Semi-supervised learning on directed graphs. In: NIPS (2005)
31. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. In: ACM SIGIR (2007)
32. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML (2003)