Image Categorization Using Directed Graphs

Hua Wang, Heng Huang, and Chris Ding

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

Abstract. Most existing graph-based semi-supervised classification methods use pairwise similarities as edge weights of an *undirected graph* with images as the nodes of the graph. Recently several new graph construction methods produce, however, *directed graph* (asymmetric similarity between nodes). A simple symmetrization is often used to convert a directed graph to an undirected one. This, however, loses important structural information conveyed by asymmetric similarities. In this paper, we propose a novel symmetric co-linkage similarity which captures the essential relationship among the nodes in the directed graph. We apply this new co-linkage similarity in two important computer vision tasks for image categorization: object recognition and image annotation. Extensive empirical studies demonstrate the effectiveness of our method.

Keywords: Directed graph learning, Co-linkage similarity, Multi-label classification, Object Recognition, Image annotation.

1 Introduction

Many computer vision problems, such as image categorization including object recognition and image annotation, are often solved by semi-supervised classification algorithms due to the lack of enough labeled training data. Most, if not all, existing graph-based semi-supervised classification algorithms learn on an *undirected graph* [21,20,6,19,16] with images as the nodes and pairwise image similarities as edge weights. The edge weights of the graph are typically assessed using traditional graph construction functions, such as Gaussian kernel similarity. Such graph-based semi-supervised classification performance is sensitive to parameter variances, and there is no reliable way to determine the optimal parameter value especially when the amount of labeled data is small [20,15,3,18].

In order to tackle this parameter sensitivity problem, several robust graph construction methods have been proposed recently such as [15,3,18,8]. A common aspect of these methods, however, is that their immediate output is a *directed graph* (asymmetric similarity between nodes). In order to make use of existing graph-base semi-supervised classification algorithms, they convert the directed graph to an undirected one through a simple symmetrization step. To be more specific, the weight of an edge on the converted undirected graph is assigned by the average of the two weights on both directions of the same edge in the original

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 762–775, 2010.

directed graph. Consequently, the important structural information conveyed by edge directions are simply discarded, and the benefits of these methods are impaired.

In this work, employing four basic second order processes of directed graph, *co-citation*, *co-reference*, and *passage* as shown in Fig. 2, we propose a novel co-linkage similarity (CS) to measure the pairwise similarity between any two vertices in a directed graph, by which the valuable structural information from directed pairwise relationship between vertices is preserved. Besides, motivated by Hypertext Induced Topic Selection (HITS) algorithm [10] and PageRank algorithm [13] but different from them, our CS symmetrically normalizes both in-links and out-links of a directed graph in a balanced manner, such that effective mutual link reinforcement can be achieved. As a result, the pairwise relationships among the vertices are enhanced and the topological structure turns out more lucid, such that the performance of subsequent classifications on the induced undirected graph is improved.

We first show an example that demonstrates the effectiveness of the proposed CS. Fig. 1 shows part of the directed graph generated by the method from [18] using UIUC car data [1]. Vertices e and l are positive training samples, *i.e.*, a car exists in the picture; vertices b and c are negative training samples, *i.e.*, no car exists in the picture; the rest are testing images. The goal is to predict the label of testing image z. Previous methods symmetrize the directed graph to an undirected one. Because the immediate training in-neighbors of z is c, which dominates the label assignment of z, the cars in picture z can not be detected. In our method, the positive co-reference between e and z, together with the positive co-citation between l and z, overwhelms the negative co-reference between b and z, therefore z is correctly assigned with a positive label.

We apply the proposed CS to two important computer vision tasks for image categorization, object recognition and image annotation, using semi-supervised classification algorithms. Promising experimental results demonstrate the effectiveness of our proposed CS.

2 Backgrounds

We first define necessary notations used in the discussions in the sequel.

Semi-supervised learning notations. Suppose we have $n = n_l + n_u$ data points $\{\mathbf{x}_i\}_{i=1}^{n}$ corresponding to n images, where the first n_l data points are already labeled by $\{\mathbf{y}_i\}_{i=1}^{n_l}$ 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 those unlabeled data.

Graph notations. Given the input data as above, we may construct either an undirected graph or a directed graph to capture the pairwise relationships among the data points for succeeding graph-based semi-supervised classification.

Pairwise similarities between data objects are usually described as an undirected graph $\mathcal{G}^u$ with a *symmetric* weight matrix $W \in \mathbb{R}^{n \times n}$.

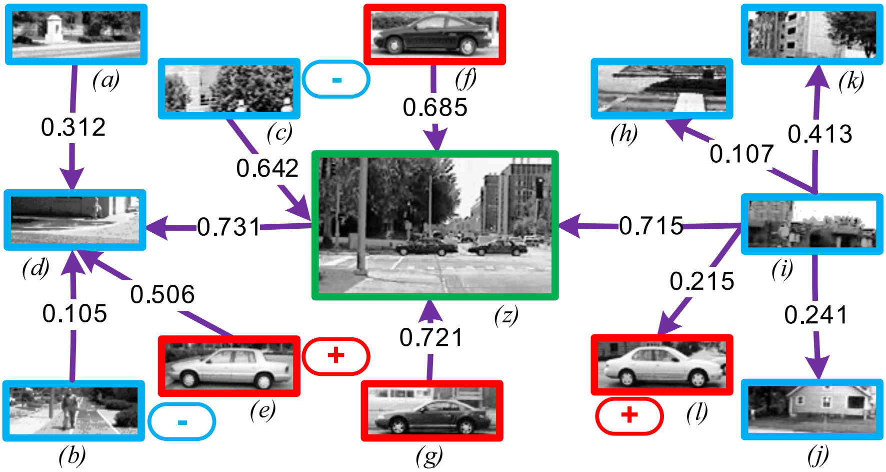


Fig. 1. Part of directed graph generated by L1G method [18]. Vertices e and l are positive training samples, *i.e.*, a car exists in the picture; vertices b and c are negative training samples, *i.e.*, no car exists in the picture; the rest are testing samples. For the test picture z, only our CS measure can correctly detect the existence of the cars.

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 the *asymmetric* adjacency matrix $L = \{0,1\}^{n \times n}$, such that $|\mathcal{V}| = n$, and $L_{ij} = 1$ if there is an edge $i \rightarrow j$ from vertex i to vertex j, and $L_{ij} = 0$ otherwise. The edge $i \rightarrow j$ 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 $d_i^+ = \sum_k L_{ik}$. Similarly, the number of in-neighbors of j is the *in-degree* of j, given by $d_j^- = \sum_k L_{kj}$. Let D_{out} be a diagonal matrix and $D_{\text{out}}(i,i) = d_i^+$, and D_{in} be a diagonal matrix and $D_{\text{in}}(i,i) = d_i^-$.

A weighted directed graph is described by a weight matrix $R \in \mathbb{R}^{n \times n}$ when there exists a function $r : \mathcal{E} \rightarrow \mathbb{R}^+$, which associates a positive value $R_{ij} = r(i,j)$ with every edge $i \rightarrow j \in \mathcal{E}$. 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 $d_i^+ = \sum_k R_{ik}$, and the in-degree is defined as $d_i^- = \sum_k R_{kj}$.

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

2.1 Traditional Undirected Graph for Semi-supervised Learning

Traditional graph construction scheme comprises two steps: adjacency construction and graph weight calculation. For the former, there exist two widely used methods [2]: ϵ ball and K-Nearest Neighbor methods. For the latter, although there exist other methods, Gaussian kernel similarity (GKS) defined as following is the most frequently used one [2]:

$$W_{ij} = \begin{cases} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right), & \forall\ (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}, \\ 0, & \text{otherwise}\ . \end{cases} \tag{1}$$

Unfortunately, classification performance using GKS in Eq. (1) heavily depends on the selection of σ, and satisfactory results are not easy to achieve [15].

2.2 Directed Graph for Robust Semi-supervised Learning

Recently, based on sparsity representations, several robust graph construction methods have been proposed [3,18,8]. Graph is a gathering of pairwise relations, while the relation among visual images is essentially an estimation by human cognition system. It has been proved [14] in neural science that the human vision system seeks a sparse coding for the incoming image using a few words in a feature vocabulary. Wright et al. [17] demonstrated that the ℓ_1 linear reconstruction error minimization can naturally lead to a sparse representation for human facial images. In other words, a graph for image data can be naturally constructed in a parameter free way through sparse representation. During the ℓ_1 minimization for the optimal sparse representation, the direction of edges and the graph weights are generated [18].

Sparse representation assumes that [3] any feature vector in a class can be represented as a linear combination of some other feature vectors in the same class. Also, given a feature vector, its sparsest representation is achieved when all the basis feature vectors belong to the same class as the feature vector. Formally, let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote all data vectors (images). For any given $\mathbf{x}_i \in \mathcal{X}$, it can be decomposed as a sparse linear combination of the rest of the feature vectors in $\mathcal{X}$. We denote $F_{\hat{i}} = [\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n]$, thus the circumflex notation $\hat{i}$ means "not i". Then the linear representation of $\mathbf{x}_i$ can be written as:

$$\mathbf{x}_i = F_{\hat{i}}\mathbf{s}_i \quad \in \mathbb{R}^m, \tag{2}$$

where $\mathbf{s}_i \in \mathbb{R}^{n-1}$ is a coefficient vector whose nonzero entries are excepted to be as few as possible. Because in image categorization, $m \ll n-1$, the system equation $\mathbf{x}_i = F_{\hat{i}}\mathbf{s}_i$ is underdetermined, its solution is not unique. Conventionally, this difficulty can resolved by choosing the minimum ℓ_2 norm solution:

$$(P_1) \qquad \arg\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_{\ell_2}, \quad \text{subject to} \quad \mathbf{x}_i = F_{\hat{i}}\mathbf{s}_i, \tag{3}$$

While this optimization problem can be easily solved (via the pseudo-inverse of $F_{\hat{i}}$), the solution $\mathbf{s}_i$ is not especially informative for categorizing $\mathbf{x}_i$. $\mathbf{s}_i$ obtained by Eq. (3) is generally dense, with large nonzero entries corresponding to many training samples [17]. To tackle this problem, we seek the sparsest solution of $\mathbf{x}_i = F_{\hat{i}}\mathbf{s}_i$ by solving the following optimization problem:

$$(P_1) \qquad \arg\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_{\ell_0}, \quad \text{subject to} \quad \mathbf{x}_i = F_{\hat{i}}\mathbf{s}_i, \tag{4}$$

where $\|\cdot\|_{\ell_0}$ denotes the ℓ_0 norm, which counts the number of nonzero entries in a vector. Although P_1 is NP-hard and even difficult to approximate, in light of the emerging theory of compressed sensing in signal processing, it can solved as following ℓ_1 minimization problem if $\mathbf{s}_i$ is sparse enough [17]:

$$(P_2) \qquad \arg\min_{\mathbf{s}_i} \|\mathbf{s}_i\|_{\ell_1}, \quad \text{subject to} \quad \mathbf{x}_i = F_{\hat{i}}\mathbf{s}_i, \tag{5}$$

where $\|\cdot\|_{\ell_1}$ denotes the ℓ_1 norm. $\mathbf{s}_i$ can be considered as the descriptor of image $\mathbf{x}_i$ using the rest images as basis. Directed graphs can be constructed using two following methods: Sparsity Induced Similarity (SIS) [3] or ℓ_1 graph (L1G) [18]:

$$\text{SIS graph: } R_{ij} = \frac{\max\{\mathbf{s}_i(j), 0\}}{\sum_{k=1,k\neq i}^{N} \max\{\mathbf{s}_i(k), 0\}}, \qquad \text{L1G graph: } R_{ij} = |\mathbf{s}_i(j)| \ . \tag{6}$$

Other directed graph construction methods are also available such as [8,17]. Note that, by the construction in Eq. (6), $R_{ij} \neq R_{ji}$, *i.e.*, a directed graph $\mathcal{G}^d$ is constructed with R as weight matrix. Apparently, R is parameter free.

Besides using sparsity representation, there also exist other mechanisms for robust graph construction, such as linear neighborhood propagation method [15], which also yields a directed graph as immediate output.

In order to work with existing graph-based semi-supervised learning methods, a simple symmetrization step has been used to convert the directed graph R to an undirected graph W [15,3,18] using

$$W_{ij} = (R_{ij} + R_{ji})/2 \ . \tag{7}$$

Obviously, structural information conveyed by edge directions are discarded.

3 Co-linkage Similarity of a Directed Graph

In this section, we will introduce a method to measure similarities in a directed graph. We consider the second order process on a directed graph. We first study the four basic processes and later emphasize the importance of edge weight normalization. This work is motivated by previous works in link analysis [10,13,5,7].

3.1 Co-linkage Similarity from Second Order Random Walk Processes

On a directed graph, we consider the random walk process. If from node i, the random walker has a large probability walks to j, we say there is a large similarity between i, j. We consider the second order process as shown in Fig. 2. There are four type of processes: co-citation, co-reference, passage ($i \rightarrow j$) and passage ($j \rightarrow i$) to as illustrated in Fig. 2. It is sufficient to use these four basic process to describe a directed graph.

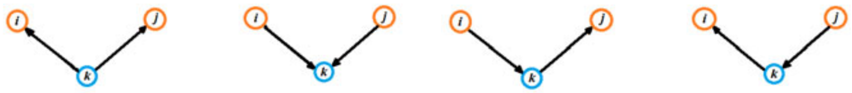


Fig. 2. Four fundamental second order processes on a directed graph. 1st figure: vertices i and j are co-cited by vertex k. 2nd figure: vertices i and j co-reference vertex k. 3rd figure: passage from vertices i to j. 4th figure: passage from vertices j to i.

Co-citation. If two vertices i and j are co-cited by many other vertices, such as k in Fig. 2 (1st figure), 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:

$$W_{ij}^{(c)} = \sum_k L_{ki} L_{kj} = \left(L^T L\right)_{ij} . \tag{8}$$

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

$$W_{ij}^{(r)} = \sum_k L_{ik} L_{jk} = (L L^T)_{ij} . \tag{9}$$

Passage. If there is a path between two vertices i and j through a vertex (such as k), i and j can commute each other and have certain similarity. Because the path could have two different directions, there are two types of passage links as shown in 3rd and 4th figures of Fig. 2. Passage($i \rightarrow j$) is computed as

$$W_{i \rightarrow j}^{(p)} = \sum_k L_{ik} L_{kj} = (LL)_{ij} \tag{10}$$

Passage($j \rightarrow i$) is computed as

$$W_{j \rightarrow i}^{(p)} = \sum_k L_{kj} L_{ki} = (L^T L^T)_{ij} \tag{11}$$

All co-citation, co-reference, and passage define the similarity between vertex pairs on a directed graph, therefore we define the co-linkage similarity as following:

$$W = L^T L + L L^T + LL + L^T L^T . \tag{12}$$

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

3.2 Link Normalization

On the web, a vertex/webpage with bigger out-degree has greater influence than another vertex/webpage with smaller out-degree. However, since these out-links can be arbitrarily added by the webpage designer, and the importance of this webpage 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. 3. Basically, if a webpage has a large out-degree, the significance/uniqueness of its cocitation is reduced. This points the necessity of out-degree normalization.

Generally speaking, the indegree of a document is not easily manipulated and is therefore a good indicator of the importance of the webpage. But, when counting co-reference between two webpages (see Fig. 3) as similarity between the webpages, in-degree should also be normalized, because a webpage i with large indegree lose the specificity of the those webpages pointing to i.

For the passage between i, j through k, the normalization is also needed. If k has large in-degree and out-degree, then the information flow through k is not *special* or *specific*, because many other pairs also pass through k. In other words, the information flow between any two nodes has a large probability to pass through k. Therefore, the passage through k is not statistically significant as compared to another k' with small in-degree and out-degree. A proper link weight normalization on k will render the mediating power of k constant, independent of the in-degree and out-degree of k.

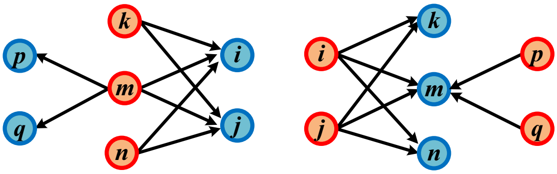


Fig. 3. Importance of hyperlink normalization. Left: vertices i and j are co-cited by vertices k, m and n. However, since vertex m also cites vertices p and q, the co-citation of i and j by m is not as significant as that by either k or n. This fact can be compensated by normalizing the weights on the 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. Right: 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.

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

$$L \rightarrow D_{\text{out}}^{-1} L, \tag{13}$$

$$L \rightarrow L D_{\text{in}}^{-1}, \tag{14}$$

$$L \rightarrow D_{\text{out}}^{-1/2} L D_{\text{in}}^{-1/2} \tag{15}$$

Normalization of Eq. (13) uses the out-degree and is used in the PageRank algorithm 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 webpages/nodes pointing to them. Normalization using out-degree will balance the importance of each of these nodes.

Normalization of Eq. (14) 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 webpages/nodes pointing to them. Normalization using in-degree will balance the importance of each of these nodes.

Normalization of Eq. (15) can be viewed as a compromise between the above two normalization. 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, co-reference, and passage, we adopt this symmetric normalization in our work.

Replacing L with the symmetric normalization, we obtain the **effective similarity** between two nodes on the directed graph

$$\begin{aligned} W = {} & D_{\text{in}}^{-\frac{1}{2}} L^T D_{\text{out}}^{-1} L D_{\text{in}}^{-\frac{1}{2}} + D_{\text{out}}^{-\frac{1}{2}} L D_{\text{in}}^{-1} L^T D_{\text{out}}^{-\frac{1}{2}} \\ & + D_{\text{out}}^{-\frac{1}{2}} L D_{\text{in}}^{-\frac{1}{2}} D_{\text{out}}^{-\frac{1}{2}} L D_{\text{in}}^{-\frac{1}{2}} + D_{\text{in}}^{-\frac{1}{2}} L^T D_{\text{out}}^{-\frac{1}{2}} D_{\text{in}}^{-\frac{1}{2}} L^T D_{\text{out}}^{-\frac{1}{2}} \ . \end{aligned} \tag{16}$$

where the first term is for co-citation the second one is for co-reference and the third and fourth are for the passages. It should be noted that co-citation and co-reference are inherently symmetric as emphasized in [5] and the passages are directional and naturally symmetrized by including both passage($i \rightarrow j$) and passage($j \rightarrow i$). Eq. (16) is the main result of this work, and we call W as co-linkage similarity (CS).

4 Empirical Studies

We apply our proposed co-linkage similarity (CS) defined in Eq. (16) into two important computer vision applications for image categorization, object recognition and image annotation, which are often solved by graph-based semi-supervised classification algorithms.

We evaluate our CS and compare to the following three graph construction schemes. (1) Traditional GKS method as in Eq. (1). This is the most frequently used graph construction method and produces a fully connected undirected graph. In our experiments, σ is fine tuned upon data set to get best performance. (2) Sparsity Induced Similarity (SIS) [3] and (3) ℓ_1 graph (L1G) [18] as in Eq. (6) are two recently published works and have demonstrated better performance than other related methods. The immediate output of these two methods is a directed graph, therefore the original papers used a simple symmetrization in Eq. (7) before classification. For our method, we induce two undirected graphs from the immediate outputs of SIS and L1G methods respectively, which are denoted as CS-SIS and CS-L1G.

4.1 Object Recognition Using Single-Label Classification

Data sets. We use the following three data sets in our evaluations, which are commonly used for semi-supervised learning and object recognition experiments.

Cedar Buffalo binary digit data set [9]. The digits are preprocessed to reduce the size of each image down to a 16×16 by down-sampling and Gaussian smoothing, and the value of each pixel ranges from 0 to 255. Each digit is thus represented by a 256-dimensional vector.

UIUC car training data set [1] consists of 1050 images of cars in side views with resolution of $40(\mathrm{H}) \times 100(\mathrm{W})$ pixels. For this data set, we use dense grids of histogram-of-gradient features to represent each image [4], where 20×20 pixel blocks, block stride of 10 pixels, and 8 orientation bins are used to obtain the feature vector of 240 dimensions for each image.

ETH-80 data set [11] contains 8 object categories. In each category there are 10 different objects, and for each object there are 41 different poses. There are $8 \times 10 \times 41 = 3280$ images in total. Here, similar to [11], we use the histogram of the first derivatives $D_x D_y$ with 48 dimensions over 3 different scales to represent each image, and then all features are normalized.

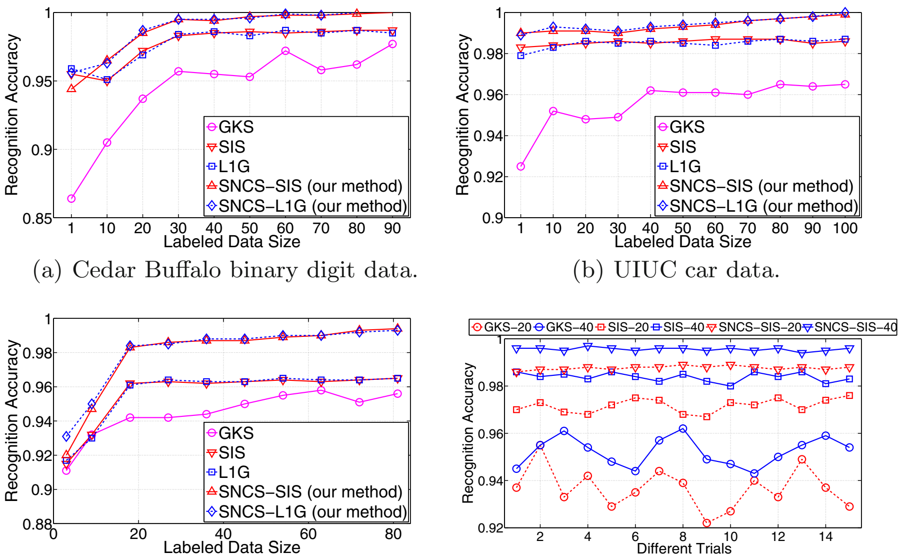
Labeled and unlabeled samples. Similar to [21], for a given class, we randomly pick up samples as labeled data, and the rest are used as unlabeled data.

Classification algorithm. We use the Green's function semi-supervised learning framework [6] for classification, which is a state-of-the-art graph-based semi-supervised classification method and has demonstrated superior performance than several representative competing methods such as [21,20].

Evaluation criterion. We employ recognition accuracy to evaluate the performance of the proposed CS in classification. Each recognition accuracy curve is obtained by averaging the results over 10 different trials. For each trial, again, we randomly select the labeled and unlabeled samples.

Classification performance. We compare the semi-supervised classification performance on the five graphs using the three data sets. We use the same experimental setup as in [3]. For Cedar Buffalo binary digit data set, we use digits "1" and "2" for classification. For UIUC car training data set, two classes, 1050 images of cars and backgrounds, are used in evaluation. These two data sets are used to evaluate binary classification performance. Then we use ETH-80 data set for multi-class classification performance. Three types of objects, apples, pears and tomatoes, are used in our experiments since it is comparatively difficult to distinguish these three categories in this data set [11].

Fgcap 4(a)–4(c) show the classification performance of five compared graph construction methods. The x-axis is the number of labeled samples, and the y-axis is the recognition accuracy. From these figures, we can see that SIS and L1G always have similar performance. This is consistent with the theoretical background as they are derived in a similar way using similar mechanism, *i.e.*, sparsity representation, as detailed in Section 2.2. Both of them are better than GKS



(a) Cedar Buffalo binary digit data.

(b) UIUC car data.

(c) ETH-80 data.

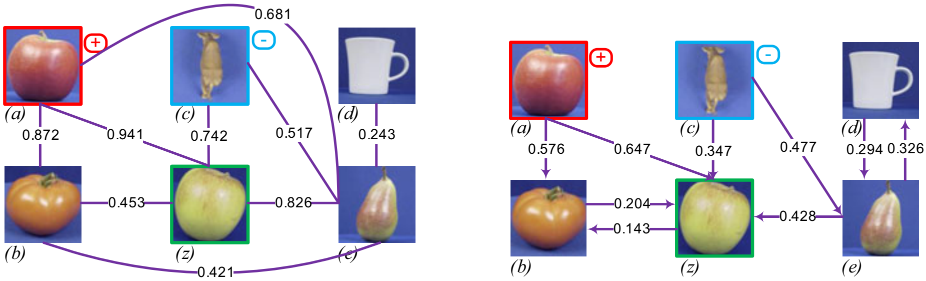
(d) Performance stability evaluation.

Fig. 4. 4(a)–4(c): Recognition accuracy on three data sets. 4(d): The performance stability of different graph construction schemes for different number of labeled samples

method. Our method, as shown by "CS-SIS" and "CS-L1G" in the figures, generally outperforms other methods, sometimes significantly. These results clearly demonstrate the effectiveness of our method.

Effectiveness of directed graph and CS. A careful examination at the classification results show that, many objects can be only correctly detected on our CS induced graph, *e.g.*, the apple of picture z in Fig. 5 for ETH-80 data and the cars of picture z in Fig. 1 for UIUC car training data set. In these figures, the training samples in the same class as the target testing sample (vertex z) are labeled as positive, while other training samples are labeled as negative.

The undirected graph generated by GKS method for a small part of ETH80 data is first shown in Fig. 5(a), where the minor edges (weight less 0.1) are removed for clear illustration. On this graph, picture z can not acquire a positive label by the label propagation algorithm, *i.e.*, the apple can not be detected. In Fig. 5(b), the directed graph generated by L1G method shows that, although the label of vertex z can be correctly inferred from its training neighbors a and c, after the simple symmetrization in Eq. (7), it can only acquire an incorrect negative label. By our CS method, because the co-reference between a and z is stronger than the relationship between c and z due to mutual link reinforcement and link normalization, picture z finally obtains a correct positive label.



(a) Part of undirected graph (by GKS).

(b) Part of directed graph (by L1G).

Fig. 5. Part of the graphs built from ETH-80 data set. Vertex a is a positive training sample and c is a negative training sample, the rest are testing samples. Test picture z can always be correctly classified on our CS induced graph. All other compared methods fail to assign correct label.

In Fig. 1, we do not show the undirected graph generated by GKS method, because the car in picture z can never be detected on GKS graphs. Considering the fact that GKS method assigns an edge weight essentially upon the image similarity using the Euclidean metric, and the sizes of the cars are relatively small and the picture has a very strong background with trees and buildings, this result is reasonable. For the directed graph generated by L1G method as in Fig. 1, the immediate training in-neighbors of z is c, which dominates the label assignment of z, therefore the cars again can not be detected. However, on our CS induced graph, the positive co-reference between e and z together with the positive co-citation between l and z overwhelm the negative co-reference between b and z, hence z is correctly assigned with a positive label.

Both examples in Fig. 5 and Fig. 1 concretely demonstrate the advantage of our method.

Classification stability. Same as in [15,3], we also study the performance stability when the amount of labeled samples varies. We compare the stability of our CS method to those of GKS method and SIS method. We do not show the stability of L1G method as it is very similar to SIS method theoretically and empirically. We use digits “1” and “2” in Cedar Buffalo binary digit data set for illustration. We compare with two different labeled data size: 20 and 40. We repeat the experiments for 15 times, and the x-axis of Fig. 4(d) is the index of different trials. The curve “GKS-20” denotes the recognition accuracy obtained by classification on GKS graph with 20 labeled samples, similar for the other curves. From Fig. 4(d), we can see that the proposed CS and SIS method have very stable performance for different labeling data sizes, while GKS exhibits relatively large performance fluctuations. Again, these results are consistent with our theoretical analysis.

4.2 Image Annotation Using Multi-label Classification

We also evaluate our proposed CS in image annotation applications. We use standard 5-fold cross validation to evaluate the performance of compared graph construction methods. Because image annotation is a multi-label classification task, we use multi-label correlated Green's function (MCGF) method [16] for classification, which is an extension of the Green's function semi-supervised learning framework [6] to deal with multi-label problems. We set the parameter α in the MCGF method as 0.1, which is same as in the original work.

Data sets. We use the following three benchmark multi-label image data sets in our evaluations.

TRECVID 2005[1] data set contains 61901 images and labeled with 39 concepts (labels). As in most previous works, we randomly sample the data such that each concept has at least 100 images, same as in [16].

MSRC[2] data set is provided by the computer vision group at Microsoft Research Cambridge, which has 591 images annotated by 22 classes.

Barcelona data set[3] contains 139 images with 4 categories, *i.e.*, "building", "flora", "people" and "sky".

In order for a complete evaluation, we first use simple (natural) features of these three image data sets. 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. In addition, we also evaluate our methods using advanced image features. For MSRC data set and Barcelona data set, 100-dimensional SIFT features (denoted as "SIFT features" in Table 1) are extracted for classification. For TRECVID 2005 data set, SIFT features can not be extracted on typical personal computers due to its big size.

Evaluation criteria. The conventional 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 definition 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 [12].

Classification performance. Table 1 presents the classification performance comparison for five methods using 5-fold cross validation on three data sets. From the results, we can see that our method is constantly better than the other compared methods. We achieve about 10.56% improvements on average for TRECVID 2005 data set, about 11.32% improvements on average for MSRC data set and about 6.60% improvements on average for Barcelona data set. Again, these results quantitatively support the theoretical advantages of our proposed method.

[1] http://www-nlpir.nist.gov/projects/trecvid

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

[3] http://mlg.ucd.ie/content/view/61

Table 1. Performance evaluations of compared methods by 5-fold cross validations

Data sets	Evaluation metrics		Compared methods				
			GKS	SIS	L1G	CS-SIS	CS-L1G
TRECVID 2005	Macro average	Precision	0.249	0.258	0.260	0.292	0.293
		F1 score	0.272	0.281	0.283	0.301	0.303
	Micro average	Precision	0.237	0.249	0.248	0.287	0.289
		F1 score	0.270	0.279	0.280	0.295	0.298
MSRC	Macro average	Precision	0.216	0.224	0.221	0.258	0.257
		F1 score	0.285	0.297	0.296	0.322	0.320
	Micro average	Precision	0.214	0.221	0.220	0.254	0.253
		F1 score	0.281	0.296	0.293	0.316	0.315
MSRC (SIFT features)	Macro average	Precision	0.341	0.361	0.360	0.420	0.418
		F1 score	0.391	0.402	0.404	0.457	0.453
	Micro average	Precision	0.324	0.351	0.350	0.424	0.423
		F1 score	0.371	0.406	0.403	0.456	0.455
Barcelona	Macro average	Precision	0.784	0.815	0.813	0.864	0.861
		F1 score	0.724	0.757	0.753	0.811	0.810
	Micro average	Precision	0.781	0.812	0.811	0.859	0.853
		F1 score	0.720	0.751	0.750	0.807	0.806
Barcelona (SIFT features)	Macro average	Precision	0.795	0.836	0.835	0.885	0.883
		F1 score	0.731	0.769	0.768	0.825	0.824
	Micro average	Precision	0.792	0.823	0.823	0.882	0.884
		F1 score	0.730	0.767	0.763	0.827	0.825

5 Conclusions

In this paper, we presented a novel co-linkage similarity (CS) to describe a directed graph in an undirected way. Besides preserving structural directionality information of a directed graph, our CS method also enhances the pairwise relationships among the data objects by taking into account both mutual link reinforcement and symmetric in-links and out-links normalization. As a result, directed graph data can be used in existing graph-based semi-supervised classification algorithms with improved classification performance, which by design can only work with undirected graph data. By applying our proposed CS into two important computer vision problems for image categorization, object recognition for single-label classification and image annotation for multi-label classification, we conducted extensive empirical studies on six benchmark data sets to evaluate various aspects of our method. Clear improvements demonstrated in all experimental results validate the performance of our proposed method.

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

References

1. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 113–127. Springer, Heidelberg (2002)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation 15(6), 1373–1396 (2003)
3. Cheng, H., Liu, Z., Yang, J.: Sparsity Induced Similarity Measure for Label Propagation. In: IEEE ICCV (2009)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR (2005)
5. Ding, C., He, X., Husbands, P., Zha, H., Simon, H.: PageRank, HITS and a unified framework for link analysis. In: ACM SIGIR (2002)
6. 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)
7. Ding, C., Zha, H., He, X., Husbands, P., Simon, H.: Link analysis: hubs and authorities on the World Wide Web. SIAM Review (2004)
8. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: IEEE CVPR (2009)
9. Hull, J.: A database for handwritten text recognition research. IEEE TPAMI 16(5), 550–554 (1994)
10. Kleinberg, J.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
11. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: IEEE CVPR (2003)
12. Lewis, D., Yang, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Stanford Digital Library Technologies Project (1998)
14. Rao, R., Olshausen, B., Lewicki, M.: Probabilistic models of the brain: Perception and neural function (2002)
15. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: ICML (2006)
16. Wang, H., Huang, H., Ding, C.: Image Annotation Using Multi-label Correlated Greens Function. In: IEEE ICCV (2009)
17. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE TPAMI, 210–227 (2009)
18. Yan, S., Wang, H.: Semi-supervised learning by sparse representation. In: SDM (2009)
19. Yang, W., Zhang, S., Liang, W.: A Graph Based Subspace Semi-supervised Learning Framework for Dimensionality Reduction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 664–677. Springer, Heidelberg (2008)
20. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: NIPS (2004)
21. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML (2003)