Maximum Consistency Preferential Random Walks

Deguang Kong, Chris Ding

Dept. of Computer Science & Engineering, University of Texas at Arlington, TX, 76013
Email:doogkong@gmail.com, chqding@uta.edu

Abstract. Random walk plays a significant role in computer science. The popular PageRank algorithm uses random walk. Personalized random walks force random walk to "personalized views" of the graph according to users' preferences. In this paper, we show the close relations between different preferential random walks and label propagation methods used in semi-supervised learning. We further present a maximum consistency algorithm on these preferential random walk/label propagation methods to ensure maximum consistency from labeled data to unlabeled data. Extensive experimental results on 9 datasets provide performance comparisons of different preferential random walks/label propagation methods. They also indicate that the proposed maximum consistency algorithm clearly improves the classification accuracy over existing methods.

1 Introduction

Random walk model [1] is a mathematical formalization of the paths that consist of taking successive random steps, i.e., at each step the walk jumps to another site according to some probability distribution. The random walk model plays an important role in computer science, and it has many applications in information retrieval [2], social network [3], etc. PageRank [4] is a link analysis algorithm, which uses the idea of random walk to measure the webpages' relative importances. Personalized Page Rank [5] is presented to create "personalized views" of the web searching results based on redefining importances according to users' preferences.

Semi-supervised learning(SSL) has connections with random walks on graphs. In SSL, only a small number of data points are labeled while a large number of data points are unlabeled. The goal of SSL is to classify the unlabeled data based on labeled data. SSL has attracted more attention because the acquisition of labeled data is quite expensive and time-consuming, while large amount of unlabeled data are easier to obtain. Many different methods have been proposed to solve SSL problems [6, 7], e.g., classification-based method [8], clustering-based method [9], graph-based method [10, 11, 12], etc. Among all these methods, the graph-based method is the most popular way to model the whole dataset as undirected weighted graph with pairwise similarities($\mathbf{W}$), and the semi-supervised learning can be viewed as label propagation from labeled data to unlabeled data, like a random walk on a similarity-graph $\mathbf{W}$. Our work is inspired by previous graph-based semi-supervised methods, especially by the works of consistency labeling [11] and Green's function [12].

In this paper, we *first* show the close relations between preferential random walks and label propagation methods. We show that the labeled data points act as the preferential/personalized bias vectors in the personalized random walks. This provides much

insight to the existing label propagation methods, and suggest ways to improve these methods. In addition, we perform extensive experiments to compare the performances of different methods used in preferential random walks.

Furthermore, we observe that current label-propagation approach may not achieve best available results, especially when the propagation operator, inferred from both labeled and unlabeled data points, does not exactly reveal the intrinsic structure of data. Many label propagation methods are done in one shot from source (labeled data) to all unlabeled data. This can not guarantee many newly-labeled data, which lie far-away in the data manifold of both labeled and unlabeled data, are labeled reliably. Motivated by this observation, in this paper, we present a novel maximum consistency approach to improve the performance of existing label propagation methods. Our approach focuses on propagating labels from source to *nearby* unlabeled data points only, and thus reliably labeling these data points. This propagation expands progressively to all unlabeled data, to ensure maximum consistency from labeled data to unlabeled data. Maximum consistency algorithm leverages existing propagation methods and repeatedly utilizes it, which incurs almost the same computational complexity as other existing propagation methods.

Here we summarize the contribution of our paper.

- We show the direct relations between preferential random walks and existing label propagation methods. Extensive experiments on 9 datasets are performed to demonstrate the performance of different methods.
- We present a maximum consistency algorithm to improve existing label-propagation methods. Extensive experiments performed on 9 datasets indicate clear performance improvement.

The rest of this paper is organized as follows. §2 gives a brief overview of personalized random walk. Next in §3, we establish the connections between the preferential random walks and label propagation methods. In §4, we emphasize the concept of score distribution in semi-supervised learning methods. In §5, we propose maximum consistency label propagation method. §6 reviews the related work to our paper. In §7, extensive experiments on 9 datasets are performed to provide the performance comparisons of both different preferential random walks/label propagation methods and proposed maximum consistency algorithm. Finally, we conclude the paper.

2 A brief view of personalized random walk

On an undirected graph with edge weights $\mathbf{W}$, let $\mathbf{D}$ be the diagonal matrix with $\mathbf{D} = \mathrm{diag}(\mathbf{We}), \mathbf{e} = (1, \ldots, 1)^T$, then $\mathbf{P} = (\mathbf{P}_{ij})$ is the transition probability from node i to node j,

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \tag{1}$$

Let $\mathbf{f}_i$ be the stationary probability of one random walker on site i. The following propagation

$$\mathbf{f} = (1-\alpha)\mathbf{y} + \alpha\mathbf{P}^T\mathbf{f}, \tag{2}$$

governs the random walker. The converged stationary distribution gives the score.

Here $\mathbf{y}$ is the personalized (bias) probability distribution; this fixed vector represents the personal interest or other preferential treat of different nodes. In PageRank [4], $\mathbf{y} = (1, \ldots, 1)^T/n$, $\alpha = 0.9$. In personalized random walk [5], $\mathbf{y}$ encodes the personalized preferences. For example, for a random walker who prefers to visit sites i_1, i_2. Then $\mathbf{y}_i = 1/2$ if $i = i_1, i_2$; $\mathbf{y}_i = 0$ otherwise.

2.1 Personalized random walk for 2-class semi-supervise learning

To do classification for partially labeled data for 2-class, we divide the data into $\mathbf{X}_+$, $\mathbf{X}_-$, and $\mathbf{X}_u$ for positively labeled, negatively labeled, and unlabeled datasets. We do two random walks: (1) one for the positive class with preferential vector $\mathbf{y}^{(+)}$ where $\mathbf{y}_i^{(+)} = 1/|\mathbf{X}_+|$ if $i \in \mathbf{X}_+$; $\mathbf{y}_i^{(+)} = 0$ otherwise. The converged score of Eq.(2) gives $\mathbf{f}^{(+)}$. (2) one for the negative class with preferential vector $\mathbf{y}^{(-)}$ where $\mathbf{y}_i^{(-)} = 1/|\mathbf{X}_-|$ if $i \in \mathbf{X}_-$; $\mathbf{y}_i^{(-)} = 0$ otherwise. The converged score of Eq.(2) gives $\mathbf{f}^{(-)}$. We then assign for each unlabeled data $\mathbf{x}_i \in \mathbf{X}_u$ the class with higher stationary distribution: $k = \max(\mathbf{f}_i^{(+)}, \mathbf{f}_i^{(-)})$.

Note that because the propagation of Eq.(2) is linear, we can do the semi-supervised learning using only *one* random walk with the preferential vector $\mathbf{y} = \frac{1}{2}(\mathbf{f}^{(+)} - \mathbf{f}^{(-)})$. We then assign for each unlabeled data $\mathbf{x}_i$ the class with $k = \text{sign}(\mathbf{f}_i)$. This is a simple algorithm. Note that here $\sum_i \mathbf{y}_i = 0$, since $\sum_i \mathbf{y}_i^{(+)} = 1$ and $\sum_i \mathbf{y}_i^{(-)} = 1$. This will be useful in deriving the Green's function method below.

2.2 Generalized Preferential Random Walk for multi-class

In multi-person random walks, there are K random walkers. Each random walker $k (1 \leq k \leq K)$has a distribution vector $\mathbf{f}_k$ and a personalized preference vector $\mathbf{y}_k$,

$$\mathbf{f}_k = (1-\alpha)\mathbf{y}_K + \alpha \mathbf{P}^T \mathbf{f}_K. \tag{3}$$

Let $\mathbf{F} = (\mathbf{f}_1, \cdots, \mathbf{f}_K)$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_K)$, from Eq.(2), we obtain the transition

$$\mathbf{F} = (1-\alpha)\mathbf{Y} + \alpha \mathbf{P}^T \mathbf{F}. \tag{4}$$

The solution for the final stationary distributions of the K random walkers are

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{P}^T} \mathbf{Y}. \tag{5}$$

Method 1:

Here we use standard random walk transition probability of Eq.(1) and obtain the stationary distributions of the K random walkers:

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{W}\mathbf{D}^{-1}} \mathbf{Y} = \frac{1-\alpha}{(\mathbf{D} - \alpha \mathbf{W})\mathbf{D}^{-1}} \mathbf{Y} = \mathbf{D} \frac{1-\alpha}{(\mathbf{D} - \alpha \mathbf{W})} \mathbf{Y}. \tag{6}$$

Method 2:

If we use the "pseudo transition probability" $\mathbf{P} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$, we obtain the stationary distributions of the K random walkers as:

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}} \mathbf{Y}. \tag{7}$$

Method3:

If we use another "pseudo transition probability" $\mathbf{P} = \mathbf{W}\mathbf{D}^{-1}$, we obtain the stationary distributions of the K random walkers as:

$$\mathbf{F} = \frac{1-\alpha}{\mathbf{I} - \alpha \mathbf{D}^{-1} \mathbf{W}} \mathbf{Y} = \frac{1-\alpha}{\mathbf{D}^{-1}(\mathbf{D} - \alpha \mathbf{W})} \mathbf{Y} = \frac{1-\alpha}{(\mathbf{D} - \alpha \mathbf{W})} \mathbf{D}\mathbf{Y}. \tag{8}$$

So far, we have discussed random walks on a graph. Next, we make connections to semi-supervised learning. The significance of relation analysis between preferential random walks and label propagations is to help to capture the essence of these algorithms and better interpret the experiment results. To our knowledge, so far there is a lack of systematic study to explore the commonalities and differences of these algorithms, and their relations to label propagation algorithms.

3 Relations between preferential random walks and Label Propagations

In semi-supervised learning, we have $n = n_\ell + n_u$ data points $\{\mathbf{x}_i\}_{i=1}^{n}$, where first n_ℓ data points are already labeled with $\{y_i\}_{i=1}^{n_\ell}$ for c target classes. Here, $\mathbf{x}_i \in \Re^p$ and $y_i \in \{1, 2, ..., K\}$, such that $y_i = k$ if x_i belongs to the k-th class. The last n_u data are unlabeled. The goal of semi-supervised learning is to learn their class labels: $\{y_i\}_{i=n_\ell+1}^{n}$. Let $\mathbf{Y} \in \Re^{n \times K}$ be a class indicator matrix, $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ is labeled as class $y_i = j$; and $\mathbf{Y}_{ij} = 0$ otherwise.

3.1 Local - Global Consistency method(LGC)

Local and global consistency(LGC) [13] utilizes sufficiently smooth assumptions with respect to the intrinsic structure collectively revealed by known labeled and unlabeled data points. Given the graph edge matrix $\mathbf{W}$, LGC constructs the normalized matrix $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$, where D is a diagonal matrix with $\mathbf{D} = \mathrm{diag}(\mathbf{W}\mathbf{e})$. Then the predicted label matrix $\mathbf{F}$ is,

$$\mathbf{F} = \mathbf{Q}\mathbf{Y}, \qquad \mathbf{Q} = \beta(\mathbf{I} - \alpha \mathbf{S})^{-1}, \tag{9}$$

where $\mathbf{Q}$ is the label propagation operator, $\alpha = \frac{1}{1+\mu}$, $\beta = \frac{\mu}{1+\mu}$, and μ is a parameter.

Relations with preferential random walk Compared with method 2 in generalized preferential random walk of Eq.(7), we can see LGC is *identical* to it. This is because constant β will not change the classification results.

3.2 Green's function method(GF)

Green's function for semi-supervised learning and label propagation is first presented in [12]. GF is defined as the inverse of graph laplacian $\mathcal{L} = \mathbf{D} - \mathbf{W}$ with zero-mode discarded. Using the eigenvectors of $\mathcal{L}$: $\mathcal{L}\mathbf{v}_k = \lambda_k \mathbf{v}_k$, where $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ are the eigenvalues. Green's function computes the predicted label matrix $\mathbf{F}$,

$$\mathbf{F} = \mathbf{Q}\mathbf{Y}, \qquad \mathbf{Q} = \mathcal{L}_+^{-1} = \frac{1}{(\mathbf{D} - \mathbf{W})_+} = \sum_{i=2}^{n} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\lambda_i}, \tag{10}$$

where $\mathbf{Q}$ is label propagation operator, $(\mathbf{D} - \mathbf{W})_+$ indicates zero eigen-mode is discarded.

Relations with preferential random walk From Method 1 of generalized preferential random walk, the stationary distribution $\mathbf{F}$ of Eq.(6) is related to Q in Eq.(10). As $\alpha \longrightarrow 1$, we have

$$(\mathbf{D} - \alpha\mathbf{W})^{-1} \longrightarrow (\mathbf{D} - \alpha\mathbf{W})^{+} = \sum_{i=2}^{n} \frac{\mathbf{v}_i \mathbf{v}_i^T}{\lambda_i}. \tag{11}$$

Indeed, for classification purpose, the GF approach is the limit of Method 1 of generalized preferential random walk of Eq.(6). This is further explained below:

(1) In semi-supervised learning, the classification result for object i is determined by the location of the largest element in i-th row(See Eq.12).

(2) Given a distribution $\mathbf{A}$ and a diagonal matrix $\mathbf{D} = \mathrm{diag}(d_1 \cdots d_n)$, $\mathbf{DA}$ will multiply the i-th row of $\mathbf{A}$ by d_i. The relative distribution of this row does not change. Thus $\mathbf{D}$ applied to distribution $\mathbf{A}$ does not change the classification results.

(3) The multiplicative constant $(1 - \alpha)$ does not change the classification too.

(4) The physical reason of discarding zero mode is the use of the Von Neumann boundary condition. Algorithmically, this is also consistent: First, the discarded zero mode in Eq.(11) is $\mathbf{v}_1\mathbf{v}_1^T/\lambda_1 = \mathbf{e}\mathbf{e}^T/(n\lambda_1)$ where $\lambda_1 = 0$. As discussed in §2.1, the multi-class random walk can be equivalently viewed as a single random walk with preference vector $\mathbf{y} = \frac{1}{2}(\mathbf{y}^{(k)} - \mathbf{y}^{(\bar{k})})$, where $\mathbf{y}^{(k)}$ is the preference vector for class k, and $\mathbf{y}^{(\bar{k})}$ is the preference vector for other classes $\bar{k}$. Note $\sum_i \mathbf{y}_i = 0$, since $\sum_i \mathbf{y}_i^{(k)} = 1$ and $\sum_i \mathbf{y}_i^{(\bar{k})} = 1$. Thus we have $(\mathbf{v}_1\mathbf{v}_1^T/\lambda_1)\mathbf{y} = 0$, indicating including the zero mode in Eq.(11) does not alter the final results of label propagation.

3.3 Comparison of preferential random walk results

In label propagation of Eq.(9) or Eq.(10), once the distribution score (a.k.a propagation score) $\mathbf{F}$ are obtained, each unlabeled data point $\mathbf{x}_i$ is assign a class label according to

$$k = \arg\max_{1 \leq j \leq c} \mathbf{F}_{ij} \tag{12}$$

Note the key difference of LGC with GF is the computation of propagation operator $\mathbf{Q}$: LGC uses Eq.(9) while GF uses Eq.(10), which leads to different label propagation results. Another popular label propagation method is Harmonic function [10], which emphasizes harmonic nature of the label diffusive process. It is very different from LGC and Green's function, thus we did not discuss it here.

We have done extensive experiments to compare the above discussed methods for semi-supervised learning. We defer the presentation of these results in the experiment §7. We next discuss another contribution of this paper, i.e., the maximum consistency algorithm on these preferential random walk/label propagation methods.

4 Score Distribution: Confidence of Label Assignment

We begin the presentation of our maximum consistency with analysis of the distribution scores of the propagation. Our approach is motivated by careful examinations of experiment results. One conclusion is that although label propagation methods are effective, their current achieved results can be improved significantly. Below we illustrate the reasons.

In both LGC (Eq.9) and GF (Eq.10) methods, the propagation is done in one shot. All unlabeled data obtain their class labels immediately. However, some unlabeled data points may lie near labeled data in the data manifold (embedding subspace), while many other unlabeled data lie far-away from the labeled data. Therefore, the **reliability** or confidence of the class labels obtained in propagation vary from high (for those lie near labeled data) to low (for those lie far-away from labeled data).

However, in the *currently standard* class assignment procedure of Eq.(12), the class decision is simply the largest one among the c classes in the propagation score distribution across c classes. For example, for $\mathbf{x}_i$, the score distribution maybe

$$(\mathbf{F}_{i1} \cdots \mathbf{F}_{ic}) = (0.1,\ 0.2,\ 0.8,\ 0.3,\ 0.05),$$

in a data with $c = 5$ classes. For $\mathbf{x}_j$, the score distribution maybe

$$(\mathbf{F}_{j1} \cdots \mathbf{F}_{jc}) = (0.2,\ 0.35,\ 0.38,\ 0.05,\ 0.3).$$

Even though both $\mathbf{x}_i, \mathbf{x}_j$ are assigned class label=3, the confidence of the assignments are different. Clearly, $\mathbf{x}_i$ is assigned with higher confidence because $\mathbf{F}_{i3} = 0.8$ is much higher than other classes. $\mathbf{x}_j$ is assigned with lower confidence because $\mathbf{F}_{j3} = 0.38$ is marginally higher than some other classes. In other words, for $\mathbf{x}_i$ the propagation score distribution has a sharp peak while for $\mathbf{x}_j$ the propagation score distribution has a rather flat peak.

There could be many reasons that $\mathbf{x}_i$'s score distribution is much sharper than the score distribution for $\mathbf{x}_j$. $\mathbf{x}_i$ could lie much closer to class$= 3$ labeled data point than $\mathbf{x}_j$. It could also be that there are more class$= 3$ labeled data near $\mathbf{x}_i$ than near $\mathbf{x}_j$. It is also possible that there are many unlabeled points near $\mathbf{x}_i$ such that they mutually enhance the class$= 3$ probability than those near $\mathbf{x}_j$. More possibilities exist. Fortunately, it is not necessary to dig out these details — they are *collectively* reflected in the propagation score distribution.

In existing label propagation approaches, both $\mathbf{x}_i, \mathbf{x}_j$ are assigned labels in one shot. Now consider a different approach where we break the actual label assignment into several rounds. We first assign class label for $\mathbf{x}_i$ and move it to the pool of already-labeled data, while defer the decision for $\mathbf{x}_j$ in later rounds. As the pool of already-labeled data expands to the neighborhood of $\mathbf{x}_j$, the propagation score distribution for

$\mathbf{x}_j$ is likely to become sharper. At this time/round, we assign class label to $\mathbf{x}_j$. Thus the class label assignment is always occurring at the situation where the assignment is done with high confidences, i.e., the assignment is done such that the data point is the most consistent with other members of the same class, both globally and locally, as reflected by the sharp score distribution. From these observations and discussions, we design a maximum consistent(MC) label propagation algorithm, which uses the label propagation operator $\mathbf{Q}$ defined in both LGC and GF methods. We call our approach as MC-LGC and MC-GF. Detailed algorithm is presented in next section.

5 Maximum Consistency Label Propagation

5.1 Design of the algorithm

Our algorithm design is guided by maximum consistency assumption, which consists of multiple label propagations,

$$\begin{aligned} \mathbf{F}^1 &= \mathbf{Q}\mathbf{Y}^0, \\ \mathbf{F}^2 &= \mathbf{Q}\mathbf{Y}^1, \\ &\cdots \\ \mathbf{F}^t &= \mathbf{Q}\mathbf{Y}^{t-1}, \end{aligned} \tag{13}$$

where $\mathbf{Q}$ is the propagation operator which can be computed from Eq.(9) or Eq.(10), and $\mathbf{F}^t$ is the label prediction matrix during each propagation. In each label propagation process, we use the current labeled data matrix $\mathbf{Y}^t$ to update the label prediction matrix $\mathbf{F}^t$.

At the end of each propagation, only those unlabeled data points whose class labels are reliably predicted are actually assigned class labels and moved into the pool of labeled data (Lpool). The rest of unlabeled data points remain in the pool of unlabeled data (Upool). Thus the pool of unlabeled data decreases with each propagation, and the pool of labeled data expands with each propagation. At last propagation, all remaining unlabeled data are assigned class labels.

Because of class balance consideration, the pool of labeled data should get approximately the same number of new members for each class. In our algorithm, each class gets one new member after each propagation. We call this procedure as "balanced class expansion (BCE)". The number of unlabeled data are shrinking while the number of labeled data are increasing during this repeated BCE procedure. The critical issue in this BCE procedure is how to select this new member for each class. i.e., how to decide "reliably predicted" data points in each BCE. As analyzed in above section, the reliability of label propagation is reflected in score distribution. Thus, in our algorithm, we use the score distribution to decide the most "reliable predicted" data points from the data points in Upool in each BCE. We will illustrate more details in the next section.

Discussion If we add different number of new members to different classes, it will produce unbalance. Even if the discriminant scores of one class are much higher than those of another class, we still consider add one number for each class. Although it is inefficient, we believe this conservative way will result in selection of more "reliable" data points.

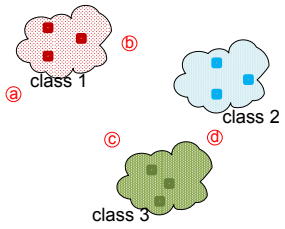


Fig. 1: Selection of discriminative data in balanced class expansion. Data points: a, b, c, d.

5.2 Normalization on the distribution score

Although data in Lpool expands in a class-balanced way, there are always the situation where classes become unbalanced. In the label propagation, we need to properly normalize the contributions from each class.

Suppose, a subset of data are labeled and there exists a class prior probability π_k. Let $\pi = \mathrm{diag}(\pi_1 \cdots \pi_k)$, and $\mathbf{Z}$ be the multi-class label assignment matrix from labeled data, i.e.,

$$\mathbf{Z}_{ik} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ belongs to class k} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

then the balanced source of propagation is defined as

$$\mathbf{Y} = \mathbf{Z}\pi = \begin{pmatrix} \pi_1\mathbf{Z}_{1,1} & \cdots & \pi_c\mathbf{Z}_{1,c} \\ \cdots & \cdots & \cdots \\ \pi_1\mathbf{Z}_{n,1} & \cdots & \pi_c\mathbf{Z}_{n,c} \end{pmatrix}. \tag{15}$$

In our algorithm, we set the prior to $\pi_k = \frac{1}{\sum_i \mathbf{Z}_{ik}}$. therefore, each class contributes the same total weight to the propagation: $\sum_i \mathbf{Y}_{ik} = \sum_i \mathbf{Y}_{i\ell}$ for any two class k, ℓ. In our algorithm the initial label matrix $\mathbf{Y}^0$ is constructed as

$$\mathbf{Y}^0 = \mathbf{Z}^0\pi^0, \tag{16}$$

where $\mathbf{Z}^0$ is the initial label assignment matrix constructed as Eq.(14) from the initially labeled data in Lpool. In the t-th iteration, let $\mathbf{Z}^t$ be the label assignment matrix constructed from current data in Lpool,

$$\mathbf{Y}^t = \mathbf{Z}^t\pi^t. \tag{17}$$

5.3 Reliable assigning class labels with score distribution

After obtaining the assignment score $\mathbf{F}_{ik}$ for all data in Upool, our goal is to pick up the "reliable" assigned data points, one for each class, and add them to the Lpool whereas remove them from the Upool. Afterwards in the actual label assignment for each class, we (1) find out all the currently unlabeled data assigned to this class, (2) pick the one with the highest discriminative score and assign it to this class.

A Motivating example to illustrate discriminant score Fig.(1) illustrates the idea of selecting discriminative unlabeled data points. Class 1 selects data a instead of data b, because a is far away from classes 2 and 3; although b is slightly closer to class 1,

but b is also closer to class 2. In other words, a is more class discriminative than b. Similarly, class 2 selects data c instead of d, because c is more discriminative than d.

Now we discuss the discriminative score computation. For each unlabeled data point $\mathbf{x}_i$, it has been assigned to k scores($\mathbf{F}_{ik}, 1 \leq k \leq c$). The c scores are then sorted as,

$$\mathbf{F}_{ik_1} \geq \mathbf{F}_{ik_2} \geq \mathbf{F}_{ik_3} \geq ... \tag{18}$$

3 classes with the highest scores are recorded as the three closest classes for $\mathbf{x}_i$: $\mathbf{F}_{k_1}$; $\mathbf{F}_{k_2}$, $\mathbf{F}_{k_3}$. As discussed above, even two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ have been assigned to the same class c_k, they may have different discriminant scores depending on the scores which how $\mathbf{x}_i, \mathbf{x}_j$ may be assigned to other classes. Here we consider the **target** class the data points will be assigned to and other two **competing** classes which we wish to be discriminant against. The discriminative scores for the 1st choice target class are defined as (if there is only 2 classes, we do not need c_{k3}),

$$\mathbf{E}(i, c_{k1}) = \mathbf{F}^2_{ic_{k1}} \frac{|\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k2}}| + |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k3}}|}{\sqrt{\mathbf{F}_{ic_{k1}} + \mathbf{F}_{ic_{k2}} + \mathbf{F}_{ic_{k3}}}}. \tag{19}$$

The score difference achieves the discriminative affects. The denominator provides a mild scale normalization. Without this term, the class with largest $\mathbf{F}_{ik}$ scale may dominate the score computation process. Note that these scores are computed once for each balanced class expansion. For each unlabeled data point $\mathbf{x}_i$ in Upool, it is assigned to class k, which has the largest $\mathbf{F}_{ik}$scores among all class k. For each class k, we select the data points $\mathbf{x}_i$, which has the largest discriminative score $\mathbf{E}(x_i, c_k)$ among all data points in Upool assigned to class k. This procedure is designed to maximize the label assignment consistency, which is consistent with LGC/GF approach.

Discussion on the discriminant score Actually, we can define other formulations of discriminant score. (1) Without the denominator of Eq.(19), discriminant score can be written as,

$$\mathbf{E}_2(i, c_{k1}) = \mathbf{F}^2_{ic_{k1}}(|\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k2}}| + |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k3}}|). \tag{20}$$

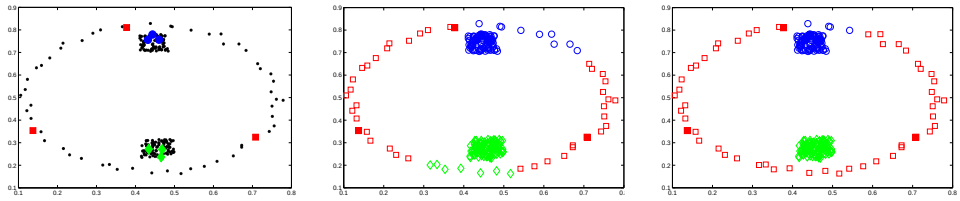
(2) Without the square for the 1st term of Eq.(19), discriminant score can be written as,

$$\mathbf{E}_3(i, c_{k1}) = \mathbf{F}_{ic_{k1}} \frac{|\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k2}}| + |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{k3}}|}{\sqrt{\mathbf{F}_{ic_{k1}} + \mathbf{F}_{ic_{k2}} + \mathbf{F}_{ic_{k3}}}}. \tag{21}$$

(3) Select more top (e.g., $4, 5, 6, 7, \cdot\cdot$) classes to compute the discriminant score, then discriminant score for T classes is given by,

$$\mathbf{E}_4(i, c_{k1}) = \mathbf{F}^2_{ic_{k1}} \frac{\sum_{t=1}^{T} |\mathbf{F}_{ic_{k1}} - \mathbf{F}_{ic_{kt}}|}{\sqrt{\sum_{t=1}^{T} \mathbf{F}_{ic_{kt}}}}. \tag{22}$$

Our experiments results(see §7.4) show Eq.(19) achieves slightly better results than other discriminant scores defined in Eqs.(20,21,22). For Eq.(20), the denominator is removed. When some $\mathbf{F}_{ic_k}$ has very large values, it may dominator the score. For Eq.(21), square of score $\mathbf{F}_{ic_k}$ is removed, which makes the score less sharper than that of Eq.(19). For Eq.(22), more top classes are fetched to achieve discriminant effect. In our experiments, we find when we select 3 classes, we can get very good results. When we select more classes, the results change slightly, but sometimes even worse.



(a) Data distribution. (b) LGC result (c) MC-LGC result

Fig. 2: Illustration of maximum consistency approach on a synthetic dataset. Labeled data shown in thick symbols: red squares, green diamonds, blue circles for 3 classes. Initially unlabeled data are shown in black stars and, after obtaining labels, shown in open symbols.

Algorithm 1 Maximum consistency label propagation algorithm (MC algorithm)

Input: labeled data $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$, unlabeled data $U = \{\mathbf{x}_j\}_{j=\ell+1}^{\ell+u}$, MaxIter
Output: predicted class labels for unlabeled data
Procedure:
1: compute propagation operator $\mathbf{Q}$ with Eq.(9) or Eq.(10), compute initial label matrix $\mathbf{Y}^0$ using Eq.(16), $t = 1$
2: **while** $t <$ MaxIter & U is not empty **do**
3: $\mathbf{F}^t = \mathbf{Q}\mathbf{Y}^{t-1}$
4: **for** each unlabeled data **do**
5: compute its corresponding discriminative score using Eq.(19)
6: **end for**
7: **for** $k = 1$ to c **do**
8: search all unlabeled data whose 1st choice target class is k. {Balanced class expansion}
9: **if** not empty **then**
10: pick the one with the largest discriminative score, add it to class k, remove it from U
11: **end if**
12: **end for**
13: Update $\mathbf{Y}^t$ with Eq.(17) using current label assignment $\mathbf{Z}^t$ {new labeled data added to Lpool}
14: $t = t + 1$
15: **end while**

Demonstration of algorithm performance on toy data. We illustrate the advantage of the MC approach (on LGC methods) in Fig.2. A 3-class synthetic dataset is displayed in Fig.(2a). For each class, three data points are labeled while the rest of data points are unlabeled. Results of standard LGC methods and MC-LGC methods are shown in Figs.(2b, 2c). It is clear that MC approaches achieves better results. One can get similar results if making the comparisons of GF against MC-GF methods.

Complete algorithm is listed in Algorithm 1. This algorithm wraps around the label propagation operator $\mathbf{Q}$, and it can also use other label propagation operators.

Time complexity analysis Note we only need to compute propagation operator $\mathbf{Q}$ (through Eq.10 or Eq.9) once as in standard LGC or GF, and the extra time cost is the iteration cost in balanced class expansion(BCE) process, which includes (1) the iteration time of BCE process which is proportional to number of iteration t; (2) the discriminant score table computation in lines $7-13$ of Algorithm 1, which is proportional to the number of *current* unlabeled data points n_l and the number of class label c. In our experiment, we find that the extra time cost is very limited as compared to the propagation operator computation in step 1.

6 Related Works

Here we discuss the previous works related to our algorithm. The related methods can be roughly divided into three categories, (1) personalized random walk (RW); (2) semi-supervised learning(SSL); (3) belief propagation (BP).

Random Walk is a popular technique widely used for PageRank algorithm [4]. Many variations of random walk methods are proposed, including personalized page rank [5], lazy random walks [14], fast random walk with restart [15], center-piece subgraph discovery [16], using ghost edge for classification [17], analysis [18] and so on.

Semi-Supervised Learning methods are widely used in real applications. Graph-based semi-supervised methods are the most popular and effective methods in semi-supervised learning. The key-idea of graph-based semi-supervised methods is to estimate a (label propagation) function on a graph, which maximizes (1) consistency with the label information; (2) the smoothness over the whole graph. Several representative methods include harmonic function [10], local and global consistency [11] and Green's function [12].

Belief Propagation [19] is widely used for inference in probability graphical model. Belief propagation methods can be used for collective classification for network data [20], grouping nodes into regions for graphs [21] and so on. However, the computational cost for BP method is usually very high.

Maximum consistency label propagation is an improvement of state-of-the-art semi-supervised learning methods, which can be extended for collective classification [20] and community detection [22]. Due to space limit, we omit the discussions here.

7 Experiments

In this section, we perform two groups of experiments. One group is to compare three different methods in preferential random walks of Eqs.(6-8), and the other group is to evaluate the effectiveness of maximum consistency (MC) algorithm.

7.1 Datasets

We adopt 9 data sets in our experiments, including two face datasets AT&T and umist, three digit datasets mnist [23], binalpha and digit[1], two image scene datasets Caltec101 [24, 25] and MSRC [25], and two text datasets Newsgroup[2] and Reuters[3]. Table 1 summarizes the characteristics of the datasets.

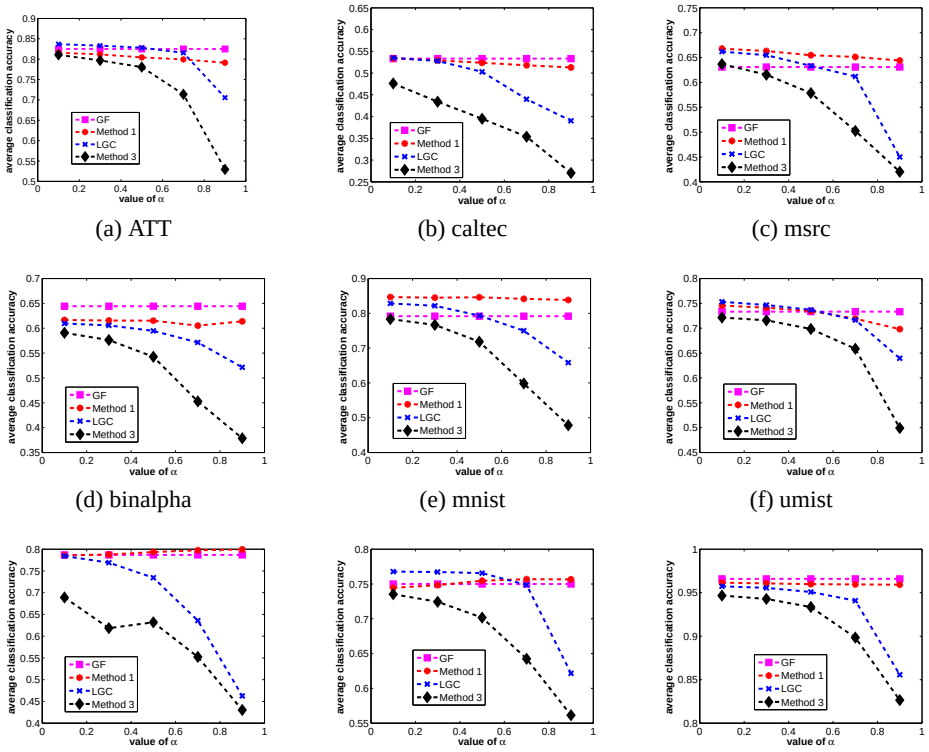
7.2 Experiments results on 3 methods of Generalized Preferential Random Walks of Eqs.(6-8)

In §2, we give three methods for generalized preferential random walks. We show method 2 is equivalent to LGC method. When $\alpha = 0.1$, GF method is the limit

[1] http://www.kyb.tuebingen.mpg.de/ssl-book/ benchmarks.html
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/
[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 1: Descriptions of datasets

Dataset	#Size	#Dimension	#Class
AT&T	400	644	40
Caltech	600	432	20
MSRC	210	432	7
Binalpha	1014	320	36
Mnist	150	784	10
Umist	360	644	20
Newsgroup	499	500	5
Reuters	900	1000	10
digit	1500	241	2



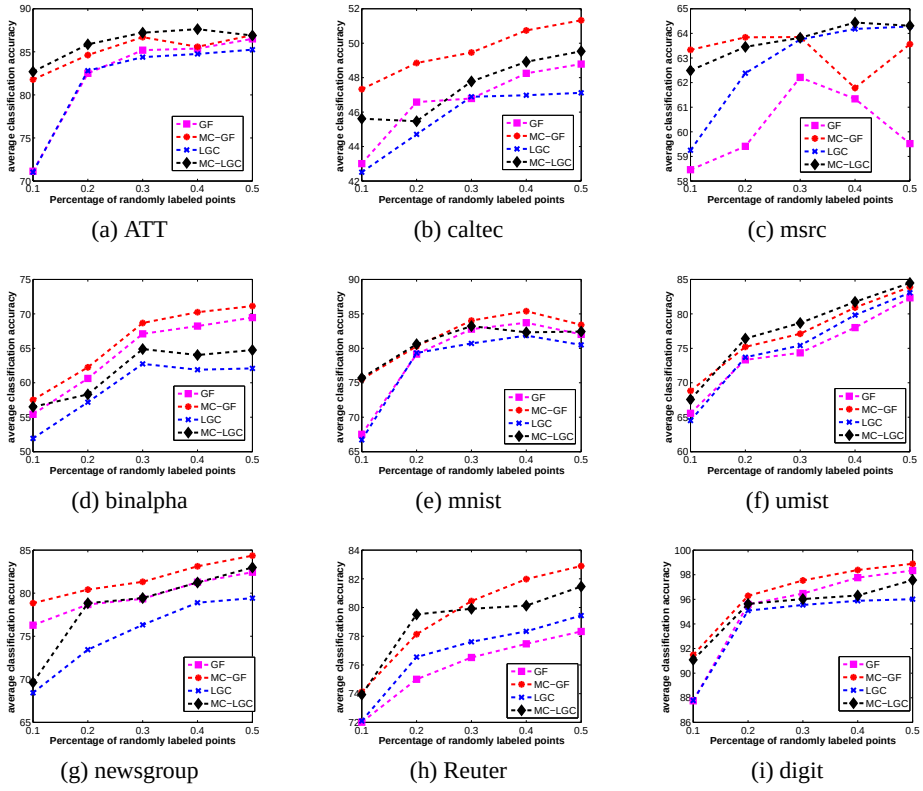
(a) ATT (b) caltec (c) msrc

(d) binalpha (e) mnist (f) umist

(g) newsgroup (h) Reuter (i) digit

Fig. 3: Experiments results on 4 methods of Generalized Preferential Random Walks: GF, method1, method2(=LGC), method3. x-axis represents the different α settings($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$), y-axis is the average classification accuracy over 10 independent runs.

of method 1. In all the methods except in GF, parameter α will influence the semi-supervised classification results. For image datasets, we use Gaussian kernel to construct the graph edge weights $\mathbf{W}_{ij} = e^{-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2}$, where γ is fine tuned according to [10]. For text datasets, we use linear kernel to compute graph similarity. We randomly select 20% of all data as the training data. In Fig.3, we show the average classification



(a) ATT (b) caltec (c) msrc

(d) binalpha (e) mnist (f) umist

(g) newsgroup (h) Reuter (i) digit

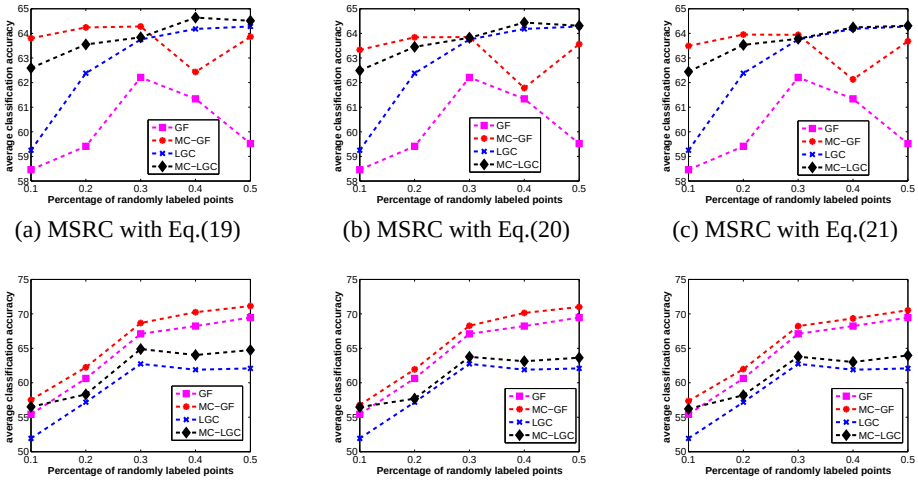
Fig. 4: Experiments results on 4 methods of label propagation: GF, MC-GF, LGC, MC-LGC. x-axis represents the different percentage of labeled data, y-axis is the average classification accuracy over 10 independent runs

results on 4 methods (GF, method1, method2(=LGC), method3) by using 5-fold cross-validation. In Fig.3, x-axis represents different α settings($\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$), y-axis is the average classification accuracy over 10 independent runs.

Experiment result analysis From Fig. 3, we can observe: (1) method 1 and GF perform well on all the datasets; (2) parameter α does not influence very much for the classification results obtained from method 1; (3) method 2 and 3 perform reasonably well when $\alpha \leq 0.5$, but their performances degrade much when α is approaching 1.

7.3 Experiment results on maximum consistency algorithm

We compare maximum consistency algorithm with standard LGC and GF methods. The α in LGC and MC-LGC methods are set to $\alpha = 0.5$ as suggested in [13]. We use Eq.(19) as the discriminant score in the balanced class expansion process. The maximum iteration time T is set according to the number of data points in the unlabeled



(a) MSRC with Eq.(19) (b) MSRC with Eq.(20) (c) MSRC with Eq.(21)

(d) binalpha with Eq.(19) (e) binalpha with Eq.(20) (f) binalpha with Eq.(21)

Fig. 5: Experiments results on 4 methods of label propagation: GF, MC-GF, LGC, MC-LGC using different discriminant score computations of Eqs.(19,20 and 21) on datasets MSRC and binalpha. x-axis represents the different percentage of labeled data, y-axis is the average classification accuracy over 10 independent runs

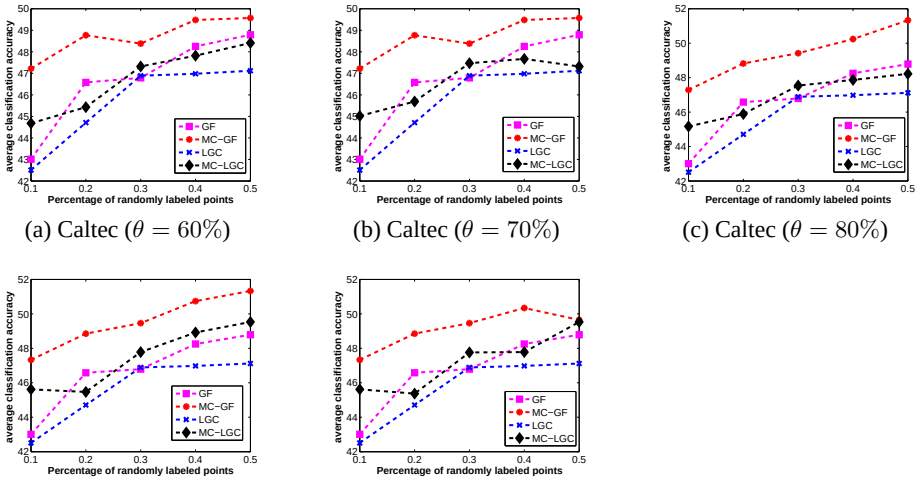
pool. If there are more than $\theta = 90\%$ of the whole data labeled, we stop the proposed maximum consistency algorithm, and do one-shot label propagation.

We show the classification results of 4 methods (LGC, MC-LGC, GF, MC-GF) by randomly selecting different percentages of labeled data in Fig.4, where x-axis represents different percentages of labeled data (i.e., $10\%, 20\%, \cdots$), and y-axis is the average classification accuracy over 10 independent runs.

Experiment results analysis From Fig. 4, we observe, (1) MC-LGC consistently performs better than LGC especially when the percentage of labeled data is very small (e.g., 10%); (2) MC-GF performs much better than GF; (3) on text dataset, MC-GF's superiority is much more significant (more than 5% improvement). Next, we discuss maximum consistency algorithm experiment results with different parameter settings.

7.4 Discussion on the parameter settings of maximum consistency algorithm

Discussion on discriminant score computation Discriminant score computation is very important for the decision of data to be propagated. The first issue is how to compute the discriminant score. Here we show the experiment results of classification when alternative discriminant score computation formulations of Eq.(20, 21) are used. The other settings of the experiments are the same as those described in §7.3. Fig. 5 shows the classification results of 4 methods of label propagation (GF, MC-GF, LGC, MC-LGC) by using different discriminant score computations of Eqs.(19, 20, 21) on datasets MSRC and binalpha. We observe that, most of the time, the classification results ob-



(a) Caltec ($\theta = 60\%$) (b) Caltec ($\theta = 70\%$) (c) Caltec ($\theta = 80\%$)

(d) Caltec ($\theta = 90\%$) (e) Caltec ($\theta = 100\%$)

Fig. 6: Experiments results on 4 methods of label propagation: GF, MC-GF, LGC, MC-LGC by using different parameter θ on dataset Caltec. x-axis represents the different percentage of labeled data, y-axis is the average classification accuracy over 10 independent runs

tained from Eq.(19) are slightly better on both datasets for both MC-GF and MC-LGC methods. These experiment results suggest us to use Eq.(19) in our algorithm.

Discussion on the iteration number Another key parameter is related to the extent to which the procedure is designed for maximizing the label assignment consistency. As described in §7.3, we use the number of labeled data points in labeled pool as a criteria to stop our algorithm. We use parameter θ to represent the percentage of *currently* labeled data of the whole dataset. In §7.3, we set $\theta = 0.9$. We try different settings of $\theta = \{60\%, 70\%, 80\%, 90\%, 100\%\}$ and report the experiment results on dataset Caltec in Fig. 6. The other settings of the experiments are the same as those described in §7.3. We find, on most of the datasets, if we set $\theta = 90\%$, we can achieve the best results. Thus we set $\theta = 90\%$ as the default setting for our maximum consistency algorithm.

8 Conclusion

We analyze the relations between 3 methods of generalized preferential random walks and label propagation methods. A maximum consistency algorithm is presented to improve current label propagation methods. Extensive experiments on 9 datasets show the effectiveness of MC algorithm and different generalized preferential random walks.

Acknowledgments. This work is supported partially by NSF-CCF-0939187, NSF-CCF-0917274, NSF-DMS-0915228.

References

1. Pearson, K.: The problem of the random walk. In: Nature. (1905)

2. Craswell, N., Szummer, M.: Random walks on the click graph. In: SIGIR. (2007)
3. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: WSDM. (2011)
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. In: Technical report, Stanford University Database Group. (1998)
5. Glen, J., Jennifer, W.: Scaling personalized web search. In: Technical report, Stanford University Database Group. (2002)
6. Chapelle, O., Schlkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge, MA, USA (2006)
7. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison (2008)
8. Blum, A., Mitchell, T.M.: Combining labeled and unlabeled sata with co-training. In: COLT. (1998) 92–100
9. Joachims, T.: Transductive learning via spectral graph partitioning. In: ICML. (2003) 290–297
10. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning. (2003) 912–919
11. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Scholkopf, B.: Learning with local and global consistency. Advances in Neural Information Processing Systems **16** (2004) 321–328
12. Ding, C.H.Q., Jin, R., Li, T., Simon, H.D.: A learning framework using green's function and kernel regularization with application to recommender system. In: KDD. (2007) 260–269
13. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schlkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, MIT Press (2004) 321–328
14. Minkov, E., Cohen, W.W.: Learning to rank typed graph walks: Local and global approaches. In: WebKDD and SNA-KDD joint workshop. (2007)
15. Tong, H., Faloutsos, C., Pan, J.Y.: Fast random walk with restart and its applications. In: ICDM. (2006)
16. Tong, H., Faloutsos, C.: Center-piece subgraphs: Problem definition and fast solutions. In: KDD. (2006)
17. Gallagher, B., Tong, H., Eliassi-Rad, T., Faloutsos, C.: Using ghost edges for classification in sparsely labeled networks. In: KDD. (2008)
18. Koutra, D., Ke, T.Y., Kang, U., Chau, D.H.P., Pao, H.K.K., Faloutsos, C.: Unifying guilt-by-association approaches: Theorems and fast algorithms. In: ECML/PKDD. (2011) 1–8
19. Pearl, J.: Reverend bayes on inference engines: A distributed hierarchical approach. In: AAAI. (1982) 133–136
20. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29** (2008) 93–106
21. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free energy approximations and generalized belief propagation algorithms. IEEE Transactions on Information Theory **51** (2005) 2282–2312
22. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-organization and identification of web communities. IEEE Computer **35** (2002)
23. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. (1998) 2278–2324
24. Dueck, D., Frey, B.J.: Non-metric affinity propagation for unsupervised image categorization. In: ICCV. (2007)
25. Lee, Y.J., Grauman, K.: Foreground focus: Unsupervised learning from partially matching images. International Journal of Computer Vision **85** (2009) 143–166