Link Prediction: the Power of Maximal Entropy Random Walk

Rong-Hua Li
Chinese Univ. of Hong Kong
rhli@se.cuhk.edu.hk

Jeffrey Xu Yu
Chinese Univ. of Hong Kong
yu@se.cuhk.edu.hk

Jianquan Liu
University of Tsukuba
ljq@dblab.is.tsukuba.ac.jp

ABSTRACT

Link prediction is a fundamental problem in social network analysis. The key technique in unsupervised link prediction is to find an appropriate similarity measure between nodes of a network. A class of wildly used similarity measures are based on random walk on graph. The traditional random walk (TRW) considers the link structures by treating all nodes in a network equivalently, and ignores the centrality of nodes of a network. However, in many real networks, nodes of a network not only prefer to link to the similar node, but also prefer to link to the central nodes of the network. To address this issue, we use maximal entropy random walk (MERW) for link prediction, which incorporates the centrality of nodes of the network. First, we study certain important properties of MERW on graph G by constructing an eigen-weighted graph $\mathcal{G}$. We show that the transition matrix and stationary distribution of MERW on G are identical to the ones of TRW on $\mathcal{G}$. Based on $\mathcal{G}$, we further give the maximal entropy graph Laplacians, and show how to fast compute the hitting time and commute time of MERW. Second, we propose four new graph kernels and two similarity measures based on MERW for link prediction. Finally, to exhibit the power of MERW in link prediction, we compare 27 various link prediction methods over 3 synthetic and 8 real networks. The results show that our newly proposed MERW based methods outperform the state-of-the-art method on most datasets.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; G.2.2 [**Discrete mathematics**]: Graph theory—*Graph algorithms*

General Terms

Algorithm, Theory, Experimentation

Keywords

Maximal entropy random walk, graph kernels, similarity measures, link prediction

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

1. INTRODUCTION

Link prediction has been recognized as a fundamental problem in social network analysis that aims to infer which unobserved links will appear in the near future by a given snapshot of a network [24]. Many problems in social computing and data mining can be modeled as a link prediction problem, such as the friends suggestion problem [32] in social network and the product recommendation problem [22] in online shopping system.

Due to the broad applications, the link prediction problem has been attracted much attention in research communities [24, 11, 25]. The key challenges of the link prediction problem are owing to the sparsity and huge size of the networks. Recently, a large number of approaches have been proposed to address this issue. The methods can be classified into two categories: supervised methods and unsupervised methods. The supervised method for link prediction is identified as the state-of-the-art method, which predict the unobserved links by a binary classifier. However, the supervised methods, such as the method proposed in [25], typically suffer from the so-called class imbalance and feature selection problem [2]. Moreover, most classifiers are based on the class distribution of the training data, thus they could perform poorly in some datasets that do not meet the prior assumptions. Instead, the unsupervised methods work in an agnostic way, thus they can naturally avoid this problem. In addition, unsupervised methods do not need to decide which node features and edge features to use for link prediction, thus they also avoid the feature selection problem. In this paper, we focus on unsupervised methods for link prediction.

The key point of the unsupervised methods for link prediction is to find an appropriate similarity measure between nodes of a graph. The widely applied methods to measure similarities between nodes of a graph are based on random walk on graph. We call this random walk the traditional random walk (TRW) in this paper. In TRW, the transition probabilities from a start node to any of its neighbors are equivalent, i.e., the reciprocal of the out-degree of the start node. TRW considers the link structures by treating all nodes of a graph as equivalent entities. In other words, TRW ignores the centrality of nodes of a graph. However, in the context of link prediction, we argue that the centrality of the nodes play an important role. The reason is because the nodes in a network not only prefer to link to the similar nodes, but also prefer to link to the central nodes of the network. Consequently, it is highly desirable to take the centrality of nodes into consideration in the random walk for link prediction task.

To address the issue of node centrality, we resort to maximal entropy random walk (MERW), which incorporates the centrality of nodes of a graph. This is because the transition probabilities of MERW are proportional to the eigenvector centrality of the nodes [5], which is wildly applied to measure the importance of the nodes over a graph. Unlike other centrality measures based on degree, betweenness, and closeness [30], the eigenvector centrality measure is based on the idea that the important of a node in a graph is larger if it has many important neighbors. In other words, the eigenvector centrality measure captures the structural context of a node. In addition to capture the centrality of the nodes, MERW has two nice properties. First, the same length paths between two given nodes have the same probabilities. Second, the longer path between two given nodes has lower probability, which consists with the intuition for measuring the similarity between nodes. As a result, we believe MERW can probably perform much better than TRW in link prediction.

MERW has been studied recently. Burda et al. study the MERW design [6]. Sinatra et al. present the evidence of constructing approximate MERW to make it more practical [34]. Devenne and Libert propose centrality measures for complex networks based on MERW, but they do not systematically analyze the properties of MERW [9]. To the best of our knowledge, we are the first group to investigate a series of important properties of MERW in depth, and systematically design a set of similarity measures based on MERW for link prediction task. Our extensive experiments exhibit the power of MERW in link prediction.

The main contributions of this paper are summarized as follows.

- We propose a set of unsupervised MERW based methods for link prediction. In particular, we propose four new graph kernels and two new similarity measures based on MERW on graph. The new graph kernels and similarity measures capture the centrality of nodes, which is important in link prediction as the nodes in real networks prefer to link to the central nodes.
- We establish a foundation to study MERW over an unweighted and undirected graph by constructing an eigen-weighted graph, whose edges are weighted by the dominant eigenvalue and eigenvector of the adjacency matrix of the graph. Based on the construction of eigen-weighted graph, we study certain important properties of MERW. We define three new graph Laplacians based on MERW, and derive new formula to compute hitting time and commute time of MERW by utilizing pseudo-inverse of the combinatorial graph Laplacian based on MERW. We show that the hitting and commute time of MERW can be deemed as two new dissimilarity measures between nodes of a graph.
- Extensive experiments confirm the power of MERW in link prediction. Specifically, we compare 27 diverse link prediction methods over 11 real and synthetic datasets. Our newly proposed MERW based approach (NMEDK) outperforms the state-of-the-art link prediction algorithm on most datasets.

The rest of the paper is organized as follows. We introduce maximal entropy random walk (MERW) in comparison with traditional random walk (TRW) in Section 2. We give a new construction of eigen-weighted graph as basis to study the important properties of MERW in Section 3. And then we propose new graph kernels and similarity measures based on MERW for link prediction in Section 4. We show the extensive performance results on link prediction in Section 5, and discuss related work in Section 6. Finally, We conclude this work in Section 7.

2. MAXIMAL ENTROPY RANDOM WALK

In this section, we introduce traditional random walk followed by discussions on maximal entropy random walk on graphs. As mentioned in [9], maximal entropy random walk cannot be applied to weighted graphs. We concentrate ourselves on unweighted and undirected graphs in this paper.

Consider an unweighted and undirected graph $G(V, E)$, with a set of nodes V and a set of edges E, where the size of nodes is $n = |V|$. Below, we use V_i to denote a node in a graph and v to denote a vector representation. The graph G can be represented as a symmetric adjacency matrix, $\mathbf{A}$, where $A_{ij} = 1$ if nodes $(V_i, V_j) \in E$ otherwise $A_{ij} = 0$.

The degree of a node $V_i \in G$ is denoted as $d_i = \sum_{j=1}^{n} A_{ij}$. Let $\mathbf{D} = diag(d_1, d_2, \cdots, d_n)$ be a diagonal matrix of node degrees, a random walk on G can be defined using a transition matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, with entries $p_{ij} = \frac{A_{ij}}{d_i}$. We call it traditional random walk (TRW) in order to distinguish from the maximal entropy random walk discussed below. It is well known that TRW on an undirected graph forms a reversible Markov chain and reaches a unique stationary distribution π [26]. The stationary distribution π satisfies the so-called detailed balance equation $\pi_i p_{ij} = \pi_j p_{ji}$, where $\pi_i = \frac{d_i}{\sum_{k=1}^{n} d_k}$, for $1 \le i \le n$.

Maximal entropy random walk: We review the maximal entropy random walk (MERW) on graphs [6, 31, 33]. First, we introduce the entropy rate of random walk on graph which is well known in information theory [8]. Consider a path τ_{ij}^t generated by the random walk with length t from V_i to V_j. Suppose the path τ_{ij}^t passes through the nodes $V_i, V_{i_1}, \cdots, V_{i_{t-1}}, V_j$, then the probability $p(\tau_{ij}^t)$ of the path τ_{ij}^t is defined as $p(\tau_{ij}^t) = p_{ii_1}p_{i_1i_2}\cdots p_{i_{t-2}i_{t-1}}p_{i_{t-1}j}$ The Shannon entropy [8] of all paths with length t generated by the walker is $E_t = -\sum_{\forall \tau_{ij}^t} p(\tau_{ij}^t) \ln p(\tau_{ij}^t)$ and the entropy rate of random walk is defined as $\eta = \lim_{t\to\infty} E_t/t$.

A well known result in information theory [8] indicates that the maximal entropy rate of random walk on a graph can be computed from the transition matrix $\mathbf{P}$ and the stationary distribution π as follows.

$$\eta = -\sum_i \pi_i \sum_j p_{ij} \ln p_{ij} \tag{1}$$

On the other hand, the maximal entropy rate of random walk on a graph is bounded by $\ln \lambda$ [31] (In [9, 33], this quantity is called *topological entropy* of a graph.), where λ is the largest eigenvalue (or called dominant eigenvalue) of the adjacency matrix $\mathbf{A}$. As shown in [6], this quantity can be obtained by the following asymptotic value.

$$\eta_{\max} = \lim_{t\to\infty} \frac{\ln \sum_{i,j} (A^t)_{ij}}{t} = \ln \lambda \tag{2}$$

Unlike TRW, MERW aims to maximize the entropy rate of a walk by carefully constructing a probabilistic transition matrix as follows. Let $v = (v_1, v_2, \cdots, v_n)$ be the normalized eigenvector ($\sum_{i=1}^{n} v_i^2 = 1$) with respect to the largest eigen-

value λ of the adjacency matrix $\mathbf{A}$. Obviously, v_i is positive as guaranteed by Forbenius-Perron theorem [20]. Then, the transition probability of MERW becomes

$$p_{ij} = \frac{A_{ij} v_j}{\lambda v_i} \tag{3}$$

We can reformulate it into a matrix form as follows.

$$\mathbf{P_v} = \frac{\mathbf{D_v}^{-1} \mathbf{A} \mathbf{D_v}}{\lambda} \tag{4}$$

where $\mathbf{D_v} = diag(v_1, v_2, \cdots, v_n)$ denotes the diagonal matrix with respect to v. The stationary distribution of MERW becomes $\pi^* = (v_1^2, v_2^2, \cdots, v_n^2)$ [6]. Together with π^* and p_{ij}, it can be easily confirmed that MERW maximizes the entropy rate. In addition, all paths τ_{ij}^t with length t between nodes V_i and V_j have the same probability $p(\tau_{ij}^t) = \frac{v_j}{\lambda^t v_i}$, which is independent of the intermediate nodes in the path. Obviously, the longer path has the smaller probability.

3. IMPORTANT PROPERTIES OF MERW

In this section, we identify certain useful properties of MERW which are crucial to construct MERW based similarity measures on graph. First, we propose an eigen-weighted graph as a basic tool for studying important properties of MERW. Then, we define three new graph Laplacians based on MERW, and show that the hitting time and commute time of MERW can be efficiently computed by the new graph Laplacian.

3.1 Eigen(vector)-weighted graph

TRW has been well studied, but MERW is not yet. In order to investigate the properties of MERW, we propose a new method to construct a family of eigenvector-weighted graphs $\mathbb{G}$ from the original unweighted and undirected graph G. With this construction, we show that the transition matrix and stationary distribution of MERW on G are identical to those of TRW on $\mathbb{G}$, respectively. We further define a special eigenvector-weighted graph, called an eigen-weighted graph $\mathcal{G}$. Based on the special graph, we simplify our investigation of MERW on G by analyzing and deriving from TRW on $\mathcal{G}$.

Definition 3.1: (**Eigenvector-weighted graph**) Given an unweighted and undirected graph $G(V, E)$, its adjacency matrix $\mathbf{A}$, the largest eigenvalue λ of $\mathbf{A}$, and the normalized eigenvector $v = (v_1, v_2, \cdots, v_n)$ w.r.t. λ, an eigenvector-weighted graph is $\mathbb{G}(V, E, \mathbb{W})$, where $\mathbb{W}$ is an *eigenvector-weighting* set, and is constructed by $\mathbb{W}_{ij} = \gamma v_i v_j$, if edge $(V_i, V_j) \in E$, where γ is a real parameter. □

We give two theorems regarding relationships between MERW on G and TRW on $\mathbb{G}$.

Theorem 3.1: *The transition matrix of MERW on graph G is identical to the transition matrix of TRW on the eigenvector-weighted graph $\mathbb{G}$.* □

Proof Sketch: Consider TRW on eigenvector-weighted graph $\mathbb{G}$. Deriving from the transition probability on an edge (V_i, V_j), we have

$$p_{ij} = \frac{A_{ij}\gamma v_i v_j}{\sum_j A_{ij}\gamma v_i v_j} = \frac{A_{ij} v_j}{\sum_j A_{ij} v_j} = \frac{A_{ij} v_j}{\lambda v_i},$$

that is identical to Eq. (3). Hence, the transition matrix of MERW on G is the same as that of TRW on $\mathbb{G}$. □

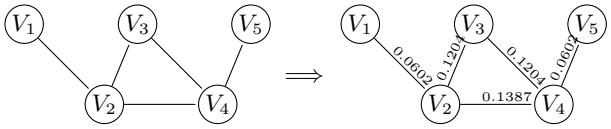

Figure 1: An Example: G (left) and $\mathcal{G}$ (right)

Theorem 3.2: *The stationary distribution of MERW on graph G is identical to the stationary distribution of TRW on the eigenvetor-weighted graph $\mathbb{G}$.* □

Proof Sketch: Consider TRW on a n-node eigenvetor-weighted graph $\mathbb{G}$. Let the stationary distribution of TRW be $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$, we have

$$\pi_i = \frac{\sum_j W_{ij}}{\sum_{i,j} W_{ij}} = \frac{\sum_j A_{ij}\gamma v_i v_j}{\sum_{i,j} A_{ij}\gamma v_i v_j} = \frac{v_i^2}{\sum_i v_i^2} = v_i^2$$

where $i = 1, 2, \cdots, n$. That is identical to the stationary distribution of MERW on graph G. □

In order to define maximal entropy graph Laplacians and to deduce important properties of MERW, we define an eigen-weighted graph which uses the dominant eigenvalue and eigenvector as the weighting set, by letting $\gamma = 1/\lambda$.

Definition 3.2: (**Eigen-weighted graph**) Given an unweighted and undirected graph $G(V, E)$, its adjacency matrix $\mathbf{A}$, the largest eigenvalue λ of $\mathbf{A}$, and the normalized eigenvector $v = (v_1, v_2, \cdots, v_n)$ w.r.t. λ, an eigen-weighted graph is defined as $\mathcal{G}(V, E, W)$, where W is an *eigen-weighting* set, and is constructed by $W_{ij} = \frac{v_i v_j}{\lambda}$, if edge $(V_i, V_j) \in E$. □

It is worth noting that for any connected, non-bipartite, undirected and unweighted graph G, there exists only one eigen-weighted graph $\mathcal{G}$ corresponding to G, based on the uniqueness properties of the dominant eigenvalue and eigenvector of the adjacency matrix of G. On the other hand, TRW on a general weighted graph typically cannot be converted to MERW on the corresponding unweighted graph. This is because it is almost impossible to decompose the weight of a given edge of the weighted graph into the product form of two corresponding elements of the dominant eigenvector. Note that both Theorem 3.1 and Theorem 3.2 hold for eigen-weighted graphs, because the eigen-weighted graph is a special eigenvector-weighted graph.

Fig. 1 shows an example. For the given original graph G (left), the largest value (λ) of $\mathbf{A}$ and the normalized eigenvector v w.r.t. λ can be obtained using the power iteration algorithm. Here, $\lambda = 2.3028$, and $v = (0.2454, 0.5651, 0.4908, 0.5651, 0.2454)$. The resulting eigen-weighted graph $\mathcal{G}$ is shown on the right.

3.2 Graph Laplacians

Graph Laplacian in spectral graph theory [7] is widely used to analyze important parameters of random walk on graph. Here we first study the graph Laplacians on the eigen-weighted graph, and then we propose three new graph Laplacians on the original un-weighted and undirected graph based on MERW.

Laplacian of eigen(vector)-weighted graph: We introduce two graph Laplacians. The first graph Laplacian, which we call general graph Laplacian, is characterized by

the transition matrix and stationary distribution of the walk [7, 4]. Let π be the stationary distribution of the walk, $\mathbf{P}$ be the transition matrix, and $\mathbf{\Pi} = \mathbf{diag}(\pi)$, the general graph Laplacian $\mathbb{L}$ can be defined as

$$\mathbb{L} = \mathbf{\Pi}(\mathbf{I} - \mathbf{P}) \tag{5}$$

where $\mathbf{I}$ is an identity matrix. In eigenvector-weighted graph, we have

$$\mathbb{L} = {\mathbf{D_v}}^2 - \frac{\mathbf{D_v A D_v}}{\lambda} \tag{6}$$

where $\mathbf{D_v} = diag(v_1, v_2, \cdots, v_n)$ and $\mathbf{A}$ is the adjacency matrix of G. Together with Theorem 3.1 and Theorem 3.2, we can conclude that the Laplacian matrix $\mathbb{L}$ defined on the eigenvector-weighted graph $\mathbb{G}$ is identical to the general graph Laplacian defined on the original G based on MERW. The second is the combinatorial graph Laplacian [7]. For any weighted graph, the combinatorial graph Laplacian $\mathcal{L}$ is defined by

$$\mathcal{L} = \mathbf{D} - \mathbf{W}$$

where $\mathbf{W}$ is the adjacency matrix of the weighted graph and $\mathbf{D}$ is a diagonal matrix of row sums of $\mathbf{W}$.

It is easy to verify that, for eigen-weighted graphs, the combinatorial graph Laplacian $\mathcal{L}$ is identical to the general graph Laplacian $\mathbb{L}$. In effect, the eigen-weighted graph is only the case that satisfies $\mathcal{L} = \mathbb{L}$.

Maximal entropy graph Laplacians: Based on the general graph Laplacian (Eq. (6)) and the combinatorial graph Laplacian on the eigen-weighted graph, we introduce three new graph Laplacians based on MERW.

Given an unweighted and undirected graph G, its adjacency matrix $\mathbf{A}$, the largest eigenvalue λ, and the normalized eigenvector $v = (v_1, v_2, \cdots, v_n)$ w.r.t λ. Let $\mathbf{D_v} = diag(v_1, v_2, \cdots, v_n)$. The maximal entropy combinatorial Laplacian (MECL) of G is defined as follows.

$$\mathbf{L} = {\mathbf{D_v}}^2 - \frac{\mathbf{D_v A D_v}}{\lambda} \tag{7}$$

$\mathbf{L}$ is equal to the combinatorial graph Laplacian defined on the eigen-weighted graph, thus we refer $\mathbf{L}$ to maximal entropy combinatorial Laplacian (MECL). Similar to the normalized Laplacians based on TRW, we can define the normalized maximal entropy Laplacians based on MERW, namely, the symmetric normalized maximal entropy Laplacian $\mathbf{L}_{sym}$, and the asymmetric normalized maximal entropy Laplacian $\mathbf{L}_{rw}$ as follows.

$$\mathbf{L}_{sym} = {\mathbf{D_v}}^{-1}\mathbf{L}{\mathbf{D_v}}^{-1} = \mathbf{I} - \mathbf{A}/\lambda \tag{8}$$

$$\mathbf{L}_{rw} = {\mathbf{D_v}}^{-2}\mathbf{L} = \mathbf{I} - {\mathbf{D_v}}^{-1}\mathbf{A}\mathbf{D_v}/\lambda \tag{9}$$

3.3 Hitting time and commute time of MERW

Hitting time and commute time are two important parameters of TRW. The hitting time $h(i,j)$ defines the average number of steps that start from node V_i and first arrive at node V_j in TRW [26]. It can be computed in an iterative fashion as follows [10, 26].

$$h(i,j) = \begin{cases} 1 + \sum_{k=1}^{n} p_{ik} h(k,j), & if\ i \neq j \\ 0, & otherwise \end{cases} \tag{10}$$

The commute time $c(i,j)$ is defined as the average number of steps that the walker starts at node V_i, reaching node $V_j (i \neq j)$ for the first time, and then goes back to node V_i such that $c(i,j) = h(i,j) + h(j,i)$. As shown in [10, 16], the commute time of TRW on an undirected graph can be used as a distance measure.

It is well known that the hitting time and commute time of TRW can be computed by the pseudo-inverse of the graph Laplacian. We show that the hitting and commute time of MERW can also be computed by the pseudo-inverse of the MECL. Assume the Moore-Penrose pseudo-inverse [10] of MECL is denoted by $\mathbf{L}^+$, with entries L^+_{ij}. We have Theorem 3.3.

Theorem 3.3:

$$h(i,j) = \sum_{k=1}^{n} (L^+_{ik} - L^+_{ij} - L^+_{jk} + L^+_{jj}) v_k^2 \tag{11}$$

$$c(i,j) = L^+_{ii} + L^+_{jj} - 2L^+_{ij} \tag{12}$$

□

We give the proof in Appendix A. It is not hard to verify that the hitting and commute time of MERW on an unweighted and undirected graph is equal to that of TRW on the corresponding eigen-weighted graph. Based on the construction of eigen-weighted graph, the commute time of MERW on graph G is also a distance measure.

4. NEW KERNELS AND SIMILARITY MEASURES

The key technique in unsupervised link prediction is to define a similarity measure between nodes of a graph. In this section, we give a class of new graph kernels and similarity measures based on MERW for link prediction.

4.1 Maximal entropy graph kernels

In TRW on graph, the pseudo-inverse of the Laplacian matrix is called commute time kernel [10]. We showed that the commute time of MERW can be computed by the pseudo-inverse of the MECL. Likewise, we can define a new maximal entropy commute time kernel based on MERW. The maximal entropy commute time kernel (CK) and its normalized kernel (CK_N) are given below.

$$CK = \mathbf{L}^+ \tag{13}$$

$$CK_N = \mathbf{L}^+_{sym} \tag{14}$$

Since the pseudo-inverse of $\mathbf{L}$ (Eq. (7)) and $\mathbf{L}_{sym}$ (Eq. (8)) are positive semidefinite, CK and CK_N are valid kernels.

As shown in [17], the heat diffusion kernel is closely related to TRW on graph. The heat diffusion kernel is constructed based on heat equation and can be defined as matrix exponentiation. Motivated by this, we define new maximal entropy heat diffusion kernels. The maximal entropy heat diffusion kernel DK and the normalized maximal entropy heat diffusion kernel DK_N are given below.

$$DK = \exp(-\alpha \mathbf{L}) \tag{15}$$

$$DK_N = \exp(-\alpha \mathbf{L}_{sym}) \tag{16}$$

where α is a positive real parameter.

The regularized Laplacian kernel of TRW on graph was first presented in [35] based on the regularization operators. By regularization on MECL, we define new maximal entropy regularized Laplacian kernels of MERW on graph. The max-

imal entropy regularized Laplacian kernel RK and the normalized counterpart RK_N are given below.

$$RK = (\mathbf{I} + \alpha\mathbf{L})^{-1} \quad (17)$$

$$RK_N = (\mathbf{I} + \alpha\mathbf{L}_{sym})^{-1} \quad (18)$$

where α is a positive real parameter, and $\mathbf{I}$ is an identity matrix.

With the help of the eigen-weighted graph, we can also define new maximal entropy Neumann kernels on the eigen-weighted graph $\mathcal{G}$. Firstly, we give a theorem.

Theorem 4.1: *Given a graph G, and its adjacency matrix $\mathbf{A}$, let $\mathbf{D_v}$ be a diagonal matrix defined in Eq. (7), then the matrix $\mathbf{I} - \alpha\mathbf{D_v}\mathbf{A}\mathbf{D_v}/\lambda$ is a positive definite matrix, for $\alpha \in (0, 1)$.* □

Proof Sketch: Obviously, the matrix $\mathbf{I} - \alpha\mathbf{D_v}\mathbf{A}\mathbf{D_v}/\lambda$ is symmetric. Let ρ be the spectral radius of matrix $\mathbf{D_v}\mathbf{A}\mathbf{D_v}/\lambda$, then we have

$$\rho \leq ||\frac{\mathbf{D_v}\mathbf{A}\mathbf{D_v}}{\lambda}||_\infty = \max_i\{\sum_j A_{ij}v_iv_j/\lambda\} = \max_i\{v_i^2\} \leq 1$$

First, because v is a normalized vector, the last inequality holds. Second, because $\alpha \in (0, 1)$, all eigenvalues of matrix $\mathbf{I} - \alpha\mathbf{D_v}\mathbf{A}\mathbf{D_v}/\lambda$ is positive. This completes the proof. □

Based on the theorem, a maximal entropy Neumann kernel NK and the corresponding normalized version NK_N can be defined as follows.

$$NK = (\mathbf{I} - \alpha\mathbf{D_v}\mathbf{A}\mathbf{D_v}/\lambda)^{-1} \quad (19)$$

$$NK_N = (\mathbf{I} - \alpha\mathbf{A}/\lambda)^{-1} \quad (20)$$

where $\alpha \in (0, 1)$ is a real parameter.

The normalized maximal entropy Neumann kernel NK_N is closely related to the Katz index when it is applied for link prediction [15]. In addition, it is not hard to show that the normalized maximal entropy regularized Laplacian kernel is equivalent to the maximal entropy Neumann kernel with a different real parameter, i.e., $RK_N = (1-\beta)(\mathbf{I} - \beta\mathbf{A}/\lambda)^{-1}$, where $\beta = \alpha/(1+\alpha)$.

4.2 Maximal entropy inverse P-distance

In [12], inverse P-distance (P_D) is defined to compute the proximity between nodes, V_i and V_j, on a graph.

$$P_D(i,j) = \sum_{\tau_{ij}: V_i \rightsquigarrow V_j} p(\tau_{ij})\alpha^{l(\tau_{ij})} \quad (21)$$

Here, the summation is taken over all paths τ_{ij} that start at node V_i and ends at node V_j, where $l(\tau_{ij})$ denotes the length of path τ_{ij}. $P_D(i,j)$ measures distances inversely: it is larger for nodes V_i "closer" to V_j. In MERW, all paths with equal length have the same probabilities. The maximal entropy inverse P-distance can be written in a more compact form.

$$P_D(i,j) = \sum_{l=1}^{\infty}(\frac{\alpha}{\lambda})^l\frac{v_j}{v_i}(A^l)_{ij} \quad (22)$$

Here, Eq. (22) can be put in a matrix form as follows.

$$\mathbf{P_D} = \frac{\alpha\mathbf{A}}{\lambda}{\mathbf{D_v}}^{-1}(\mathbf{I} - \frac{\alpha\mathbf{A}}{\lambda})^{-1}\mathbf{D_v} \quad (23)$$

By replacing α^l by $\frac{\alpha^l}{l!}$ in Eq. (22), we have a matrix exponentiation, denoted as $\mathbf{P_D}'$ as follows.

$$\mathbf{P_D}' = \frac{\alpha\mathbf{A}}{\lambda}{\mathbf{D_v}}^{-1}\exp(\frac{\alpha\mathbf{A}}{\lambda})\mathbf{D_v} \quad (24)$$

4.3 Maximal entropy SimRank

The original SimRank is based on the idea that two nodes are similar if they are joined to similar neighbors [13]. The SimRank is closely related to the random walk on a product graph. Motivated by this, we propose a new SimRank based maximal entropy random walk on graph, denoted as $S(x,y)$ for two nodes V_x and V_y in Eq. (25).

$$S(x,y) = \begin{cases} \alpha\lambda^{-2}v_xv_y\sum_{a\in N(x)}\sum_{b\in N(y)}\frac{S(a,b)}{v_av_b}, & \text{if } x \neq y \\ 1, & \text{if } x = y \end{cases} \quad (25)$$

where $\alpha \in (0, 1)$ and $N(x)$ denotes to the neighbor node set of node V_x. We call Eq. (25) the maximal entropy SimRank equation. The existence and uniqueness of the solution to the maximal entropy SimRank equation is guaranteed by the following theorem.

Theorem 4.2: *The maximal entropy SimRank equation defined in Eq. (25) has a unique solution.* □

Proof Sketch: This can be proved in the similar as to prove the original SimRank equation in [13]. □

The maximal entropy SimRank can be computed in an iterative fashion as follows.

$$R_{k+1}(x,y) = \alpha\lambda^{-2}v_xv_y\sum_{a\in N(x)}\sum_{b\in N(y)}\frac{R_k(a,b)}{v_av_b} \quad (26)$$

where the initial point is defined as $R_0(x,y) = 0$ if $x \neq y$, otherwise $R_0(x,y) = 1$. Since the maximal entropy SimRank equation has an unique solution, Eq. (26) has the same form as Eq. (25), the iterative computation can be reached by a fixed-point. More formally, $S(x,y) = \lim_{k\to\infty} R_k(x,y)$.

Note that SimRank can be computed by the so-called "expected-f meeting distance" of two surfers in a random surfer-pairs model [13]. The random surfer pairs model is identical to a random surfer on a so-called product graph induced by the original graph. Each node in the product graph is a node pair of the original graph. Thus, two random surfers starting at nodes V_x and V_y, respectively, meeting at V_u, is equal to the random surfer starting from node (V_x, V_y) and ending at node (V_u, V_u) in the product graph. The maximal entropy SimRank can also be modeled by a maximal entropy random surfer-pairs model. Below, we first give a maximal entropy expected-f meeting distance and then establish the equivalent relationship between it and maximal entropy SimRank. The maximal entropy expected-f meeting distance $dist(x,y)$ between two nodes V_x and V_y is defined as

$$dist(x,y) = \sum_{\tau:(x,y)\rightsquigarrow(u,u)} p(\tau)\alpha^{l(\tau)} \quad (27)$$

where τ denotes a path generated by the maximal entropy random surfer starting at node (x,y) and ending at node (u,u) in the product graph. $p(\tau)$ denotes the probability of path τ, and $l(\tau)$ is the length of the τ. Based on Eq. (27), we have the following theorem.

Theorem 4.3: *The maximal entropy SimRank between a node pair (x,y) is their maximal entropy expected-f meeting distance traveling back-edges.* □

Proof Sketch: We prove it by splitting the path τ into two parts: the first step $(a,b) \rightsquigarrow (x,y)$, where $a \in N(x), b \in$

$N(y)$; and the remaining path τ'.

$$\begin{aligned} dist(x,y) &= \sum_{\tau:(u,u)\rightsquigarrow(x,y)} p(\tau)\alpha^{l(\tau)} \\ &= \sum_{a\in N(x)}\sum_{b\in N(y)}\sum_{\tau':(u,u)\rightsquigarrow(a,b)} \frac{v_x v_y}{\lambda^2 v_a v_b} p(\tau')\alpha^{l(\tau')+1} \\ &= \frac{\alpha v_x v_y}{\lambda^2}\sum_{a\in N(x)}\sum_{b\in N(y)}\sum_{\tau':(u,u)\rightsquigarrow(a,b)} \frac{p(\tau')\alpha^{l(\tau')+1}}{v_a v_b} \\ &= \frac{\alpha v_x v_y}{\lambda^2}\sum_{a\in N(x)}\sum_{b\in N(y)} \frac{dist(a,b)}{v_a v_b} \end{aligned} \tag{28}$$

Note that $l(\tau) = l(\tau') + 1$ and $p(\tau) = \frac{v_x v_y}{\lambda^2 v_a v_b} p(\tau')$. Since $dist(x, y)$ is equal to the maximal entropy SimRank (Eq. (25)), the maximal entropy expected-f meeting distance between two nodes is equivalent to their maximal entropy SimRank. □

5. EXPERIMENTS

In this section, we evaluate the effectiveness of MERW based similarity measures for link prediction. In the following, we first introduce the experimental setup, and then report our results.

5.1 Experimental setup

Datasets: We conduct our experiments on 3 synthetic and 8 real networks. Specifically, for synthetic networks, we generate three networks with 1,000 nodes using three classic random graph generators: **ER** (Erods-Renyi random graph [30]), **BA** (Barabasi-Albert random graph (scale free graph) [3]), and **SW** (small word random graph [30]). For real networks, we test the proposed methods on 8 representative datasets, which are widely used for link prediction both in computer science community and physics community. Specifically, the first five are **USAir** (network of US air transportation system), **C.elegans** (neural network of the nematode worm [36]), **Yeast** (protein-protein interaction network [14]), **Power** (network of power grid of the western US [36]), and **NetScience** (collaboration network of researchers [29], who work on complex network theory). The second three are collaboration networks collected form Arxiv e-print archive [21], including three different areas of physics. In particular, they are **Gr-Qc** (General Relativity and Quantum Cosmology), **Hep-ph** (High Energy Physics-Phenomenology), and **Hep-Th** (High Energy Physics-Theory). Table 1 shows the numbers of nodes and edges of the networks as well as the numbers of nodes and edges of the giant components (GC) of the networks.

Networks	Nodes	Edges	Nodes of GC	Edges of GC
ER	1000	9997	1000	9997
BA	1000	11964	1000	11964
SW	1000	1995	1000	1995
USAir	332	2126	332	2126
C.elegans	453	2298	453	2298
Yeast	1870	4480	1458	1993
Power	4941	6594	4941	6594
NetScience	1461	2742	397	914
Gr-Qc	5242	14490	4158	13428
Hep-ph	12008	118505	11204	117649
Hep-Th	9877	25748	8638	24806

Table 1: Datasets statistics

Evaluation metrics: We employ two widely used metrics to evaluate the link prediction methods: the Area under the ROC curve (AUC)[25, 27] and precision [27, 2]. The first metric evaluates the overall ranking yielded by the algorithms, while the second metric focuses on top-K predictive results. In our experiments, the AUC is computed by a standard method described in [27]. A larger AUC value indicates a better link prediction performance. The precision is defined as the ratio of relevant number of items over all selected items using $\frac{k}{K}$. Here, we set $K = 30$, and k is the number of links that successfully predicted by the algorithms. Obviously, the larger precision means the higher predictive accuracy.

Baselines: We compare 27 various link prediction algorithms involving 11 MERW based methods, 10 TRW based methods, common neighbor, Adamic/Ada [1], supervised link prediction method [25], and supervised random walk [2]. In particular, the similarity measures we tested include: Commute time of TRW (CTT), Commute time of MERW (CTME), Commute time kernel (CK), Maximal entropy commute time kernel (MECK), Normalized commute time kernel (NCK), Normalized maximal entropy commute time kernel (NMECK), Heat diffusion kernel (DK), Maximal entropy heat diffusion kernel (MEDK), Normalized heat diffusion kernel (NDK), Normalized maximal entropy heat diffusion kernel (NMEDK), Regularized Laplacian kernel (RK), Maximal entropy regularized Laplacian kernel (MERK), Normalized regularized Laplacian kernel (NRK), Normalized maximal entropy regularized Laplacian kernel (NMERK), Neumann kernel (NK), Maximal entropy Neumann kernel (MENK), Normalized Neumann kernel (NNK), Normalized maximal entropy Neumann kernel (NMENK), Inverse P-distance (PD), Maximal entropy inverse P-distance (MEPD), Inverse P-distance with matrix exponentiation (PDM), Maximal entropy inverse P-distance with matrix exponentiation (MEPDM), SimRank (SR), Maximal entropy SimRank (MESR), and Common Neighbor (CN), Adamic/Adar (AA), supervised link prediction method (HPLP+), and supervised random walk (SRW).

Link prediction methodology: For each dataset given in Table 1, first, we extract the giant component of the graph, and randomly split the edges into a training set and a test set. The test set contains 10% of all edges in the giant component. In particular, we conduct 10 times random partitions of training and test sets on the datasets, and the link prediction results are the average over this partitions. Second, we perform the link prediction algorithms on the datasets. Specifically, we consider two cases: the unsupervised methods and supervised methods. For the unsupervised algorithms, which involve MERW and TRW based method, CN, and AA, we compute the similarity matrix on the training set using the similarity measures described in the previous sections. Then, we use the similarity matrix to calculate the predictions for the edges in the test set. For the supervised algorithms, including HPLP+ and SRW, we perform the corresponding learning algorithms on the training set, and then compute predictions on the test set. It is worth mentioning that we implement a naive Bayes classifer using the full feature set (HPLP+) defined in [25] for supervised link prediction, as it obtains the best performance in [25]. Finally, we compare the link prediction performance using the evaluation metrics described above.

Parameter settings: In MERW or TRW based similarity measures, there is only one parameter: the damping factor

α. We set $\alpha = 0.5$, as it is not very sensitive in our experiments. In addition, we set all parameters of other baselines as the same as their original papers respectively.

Experimental environment: All experiments are conducted on the Linux workstation with 2xQuad-Core Intel Xeon 3.06 GHz CPU, 48 Gb memory, and running CentOS 5.5. All the algorithms are implemented by MATLAB 2009 and Visual C++ 6.0.

5.2 Experimental results

Table 2 and Table 3 show the results of 27 various link prediction algorithms on 11 datasets under AUC and precision metric respectively. We can clearly see that the MERW based link prediction methods achieve better predictions than all other unsupervised methods under both AUC and precision metric. Moreover, on most datsets, the MERW based methods perform better than supervised methods. Generally, the normalized maximal entropy graph kernels outperform the non-normalized ones. And the normalized maximal entropy diffusion kernel achieves the best performance among all the MERW based methods. We show the detail analysis as follows.

Commute time kernels: By comparing commute time of TRW (CTT), commute time of MERW (CTME, Eq. (12)), commute time kernel (CK), maximal entropy commute time kernel (MECK, Eq. (13)) and their corresponding normalized kernels (Eq. (14)), we observe that the normalized maximal entropy commute time kernel outperforms the others on most datasets. We also see that both CTT and CTME perform poorly. This consists with a recent result reported in [28], in which the authors claim that the commute time distance converges to a meaningless distance measure, thus it results in poor precision for link prediction.

Heat diffusion/regularized Laplacian kernels/Neumann kernels: Comparing among heat diffusion kernel (DK), maximal entropy heat diffusion kernel (MEDK, Eq. (15)), and their normalized counterparts (NMEDK, Eq. (16)), the results show that NMEDK achieves significant improvement over DK, MEDK, and NDK. For instance, in Yeast dataset (column 7, row 8-11), NMEDK achieves near-optimal AUC, and obtain 67.6%, 517.4%, and 135.9% relative improvement on DK, MEDK, and NDK in terms of precision metric, respectively. The similar comparisons can also be observed between the regularized Laplacian kernels and maximal entropy regularized Laplacian kernels (Eq. (17) and Eq. (18)), and also between the Neumann kernels and maximal entropy Neumann kernels. In addition, it is worth mentioning that the normalized maximal entropy heat diffusion, normalized maximal entropy regularized Laplacian kernel, and the normalized maximal entropy Neumann kernel get the similar performance on most datasets, and they outperform the corresponding non-normalized kernels. The AUC of the maximal entropy graph kernels achieve near-optimal value on most datasets. Moreover, the precision of these graph kernels is close to 1 on USAir, C.elegances, Yeast, and NetScience, as well as the three synthetic graphs. This results indicate that the MERW based similarity measures can probably capture the nature of link formation process of these networks, thus the link predictive precision is close to 1.

Inverse P-distance and SimRank: Among inverse P-distance (PD), inverse P-distance with matrix exponentiation (PDM), SimRank (SR), and their maximal entropy counterparts (Eq. (23), Eq. (24), and Eq. (25)), the maximal entropy P-distances with matrix exponentiation (MEPDM) outperforms others in terms of both AUC and precision. Over all the datasets, the maximal entropy P-distances with matrix exponentiation (MEPDM) improves AUC over PD, MEPD, and PDM by 11.8%, 1%, 16.6% on average, respectively. And also it boosts precision on PD, MEPD, and PDM by 52.8%, 0.7%, 362.6% on average respectively. SimRank measures perform poorly than the graph kernels. More worse, the time complexity of SimRank measures is $O(n^4)$, thus we cannot obtain all the experimental results.

Comparison with supervised methods: Here we compare the MEPDM with supervised link prediction (HPLP+), and supervised random walk (SRW), as MEPDM achieves the best performance over the unsupervised methods in our experiments. We can clearly see that the performance of MEPDM is better than supervised methods (HPLP+ and SRW) over USAir, C.ele, Yeast, NetScience, GrQc, HepTh, and three synthetic networks under both AUC and precision metric. Moreover, on the rest datasets, MEPDM achieves competitive performance with supervised methods. It is worth mentioning that the SRW slightly outperforms HPLP+ under both AUC and precision metric on most datasets. The AUC of MEPDM, HPLP+, and SRW falls into the range (0.72, 0.99), and the precision of this algorithms is roughly between (0.12, 0.96). This results suggest that the MEPDM, as supervised methods do, exhibits very good performance in link prediction.

To summarize, the experimental results highlight the power of MERW in link prediction. Also, the results empirically confirm that the centrality of nodes is very important in link prediction. Since the MERW based methods inherently capture the centrality of nodes, they can yield much better performance than those methods that do not consider the centrality of nodes in link prediction. Indeed, as observed in many real networks [3, 30], the link formation typically consists with a preferential attachment process. That is to say, the nodes in the network tend to link to the central nodes, which results in a so-called "rich-get-richer" phenomenon. In effect, the MERW implicitly incorporates this phenomenon into the random walk process. This is because the transition probability in each step of MERW refers to the eigenvector centrality of the nodes, thus it makes the walk greedily moves to the important nodes. Besides, as opposed to the supervised methods, the MERW based methods work in a agnostic manner, thus they can naturally avoid the class distribution and feature selection problem [2].

6. RELATED WORK

Link prediction and similarity measure on graph: After the seminal work by Liben-Nowell and Kleinberg [24], the link prediction problem has attracted considerable attention in recent years both from computer science and physics community [27, 11, 19, 25, 18, 2]. The existing link prediction approaches can be classified into two categories: unsupervised and supervised methods. Most unsupervised link prediction algorithms are based on the similarity measure between the nodes of a graph. A recent survey can be found in [27]. Below, we focus on the random walk based similarity measures. In [13], SimRank is proposed, based on the idea that two nodes are similar if they are joined to similar neighbor nodes. The complexity of this algorithm is

SM	ER	BA	SW	USAir	C.ele	Yeast	Power	NetSci	GrQc	HepPh	HepTh
CTT	0.710	0.750	0.791	0.847	0.784	0.709	0.713	0.917	0.520	0.523	0.525
CTME	0.720	0.746	0.745	0.855	0.798	0.501	0.501	0.866	0.556	0.645	0.534
CK	0.805	0.883	0.804	0.856	0.809	0.715	0.501	0.799	0.513	0.501	0.513
MECK	0.940	0.981	0.845	0.936	0.856	0.757	0.501	0.975	0.517	0.501	0.503
NCK	0.502	0.501	0.501	0.708	0.706	0.501	0.501	0.501	0.503	0.508	0.501
NMECK	0.903	0.983	0.982	0.931	0.969	0.710	0.501	0.971	0.623	0.750	0.675
DK	0.835	0.813	0.983	0.836	0.838	0.829	0.764	0.965	0.501	0.605	0.593
MEDK	0.999	0.983	0.998	0.991	0.971	0.749	0.812	0.963	0.739	0.735	0.746
NDK	0.786	0.711	0.956	0.920	0.778	0.731	0.857	0.908	0.531	0.530	0.530
NMEDK	**0.999**	**0.983**	**0.998**	**0.997**	**0.978**	**0.970**	0.857	**0.996**	**0.739**	0.755	**0.758**
RK	0.851	0.907	0.973	0.898	0.887	0.803	0.864	0.624	0.632	0.608	0.561
MERK	0.999	0.983	0.998	0.981	0.949	0.812	0.812	0.963	0.618	0.745	0.735
NRK	0.504	0.501	0.501	0.719	0.501	0.703	0.806	0.501	0.501	0.508	0.504
NMERK	0.999	0.983	0.998	0.983	0.975	0.968	0.857	0.986	0.739	0.755	0.756
MENK	0.999	0.983	0.998	0.936	0.975	0.799	0.812	0.963	0.618	0.730	0.746
NNK	0.503	0.501	0.501	0.819	0.501	0.705	0.806	0.501	0.501	0.508	0.504
NMENK	0.999	0.983	0.998	0.983	0.965	0.965	0.857	0.996	0.739	0.755	0.752
PD	0.926	0.974	0.953	0.971	0.866	0.887	0.857	0.722	0.666	0.618	0.628
MEPD	0.999	0.976	0.998	0.993	0.964	0.968	0.857	0.913	0.739	0.755	0.758
PDM	0.805	0.764	0.957	0.972	0.798	0.886	0.857	0.874	0.616	0.660	0.530
MEPDM	0.999	0.983	0.998	0.990	0.976	0.970	0.857	0.996	0.739	0.755	0.758
SR	–	–	–	0.905	0.860	–	–	0.955	–	–	–
MESR	–	–	–	0.960	0.876	–	–	0.963	–	–	–
CN	0.884	0.782	0.501	0.386	0.971	0.752	0.802	0.961	0.617	0.623	0.635
AA	0.886	0.781	0.501	0.409	0.975	0.793	0.806	0.969	0.623	0.630	0.638
HPLP+	0.983	0.971	0.978	0.979	0.974	0.965	**0.886**	0.984	0.725	0.753	0.732
SRW	0.991	0.977	0.989	0.983	0.972	0.967	0.863	0.983	0.731	**0.760**	0.754

Table 2: Comparison of AUC value among 27 various algorithms.

$O(n^4)$. Pei, et al. [23] propose an efficient single-pair SimRank algorithm. In [38], a family of dissimilarity measures are developed based on a biased random walk on graph, which generalize both the commute-time and the shortest-path distances. In [12], a unified distance function, namely inverse P-distance, is proposed. There exists a strong connection between the personalized pagerank and the inverse P-distance. In [11], it uses graph kernels for link prediction task. In [19], a learning framework is proposed for link prediction, which generalizes several graph kernels based methods. Later, in [18], the same authors proposes a spectral evolution model and develops a spectral extrapolation algorithm for link prediction, which is based on the observation that the large networks change over time result in a change of a the graph's spectrum and keeping the eigenvectors unchanged. More recently, the supervised link prediction has attracted much attention [25, 2]. For instance, in [25], the authors propose a supervised method using classic classifier such as naive Bayes, decision tree, and bagging for link prediction. In [2], a supervised random walk is designed for link prediction, which performs random walk on a weighted graph with the weights learned by a supervised learning algorithms. The supervised methods typically suffer from the class imbalance and feature selection problems. However, our MERW based methods are unsupervised approaches, thus they can naturally avoid the feature selection problem.

Maximal entropy random walk on graph: The maximal entropy random walk was first proposed by Ruelle and Bowens in [33] and it was also called Ruelle-Bowens random walk [9]. This random walk on an unweighted graph chooses the transition probabilities proportional to the importance of the nodes measured by its eigenvector centrality, which is a well known centrality measure in sociology [5]. As shown in [31, 6, 9, 34], this process makes all paths between two given nodes with same length have the same probabilities. That is to say, the transition probabilities of this random walk are chosen to maximize the entropy rate of the walk.

In [6], a maximal entropy random walk is designed on an unweighted and undirected graph according to the dominant eigenvector of the adjacency matrix. In [31], it shows that there is one and only one random walk that can achieve the maximal entropy rate. The main problem of designing maximal entropy random walk is that it requires to know the dominant eigenvector of the adjacency matrix, which means one should have global knowledge of the graph. However, such global knowledge is unavailable. To make MERW more practical, [34] shows that one can construct approximate maximal entropy random walk with the degree of the nodes on graph. The most related to our work is [9], where a centrality measure, namely entropy rank, is proposed based on the maximal entropy random walk on an unweighted and directed graph. The entropy rank uses the stationary distribution of the maximal entropy random walk on an unweighted and directed graph as the measure of centrality. As shown in [9], the stationary distribution of MERW on a directed graph is identical to the element-wise product of the dominant left and right singular vector of the adjacency matrix. The major drawback of this measure is that it can make the centrality of the nodes with zero in-degree or out-degree equal to zero. In this work, we explore the power of MERW on undirected graphs in the context of link prediction.

Graph kernel: Graph kernel is a powerful tool. We focus on graph kernels defined as a similarity measure between nodes of a given graph. Kondor and Lafferty [17] first propose to construct kernel on graph. They propose an exponential kernel, namely diffusion kernel, based on the heat equation. This kernel can be naturally constructed by matrix exponentiation. Smola and Kondor extend it based on the regularization operators, which produce several regularized graph kernels [35]. Recently, Fouss, et al. [10] show

SM	ER	BA	SW	USAir	C.ele	Yeast	Power	NetSci	GrQc	HepPh	HepTh
CTT	0.01	0.05	0.21	0.35	0.14	0.11	0.08	0.44	0.02	0.02	0.02
CTME	0.02	0.05	0.05	0.36	0.14	≈0	≈0	0.06	0.11	0.15	0.03
CK	0.71	0.74	0.58	0.06	0.20	0.02	≈0	0.10	0.01	≈0	0.01
MECK	0.74	0.95	0.44	0.60	0.69	0.06	≈0	0.02	0.02	≈0	0.03
NCK	≈0	≈0	≈0	0.01	0.01	≈0	≈0	≈0	≈0	0.01	≈0
NMECK	0.70	0.95	0.79	0.53	0.90	0.01	≈0	0.13	0.12	0.15	0.18
DK	0.22	0.21	0.94	0.14	0.24	0.55	0.1	0.53	≈0	0.11	0.09
MEDK	0.96	0.95	0.95	0.95	0.92	0.15	0.01	0.26	0.16	0.14	0.16
NDK	0.09	0.03	0.86	0.32	0.08	0.39	0.12	0.02	0.03	0.03	0.03
NMEDK	**0.96**	**0.95**	**0.95**	**0.96**	**0.93**	**0.92**	0.12	**0.94**	**0.21**	0.22	**0.24**
RK	0.82	0.79	0.89	0.20	0.39	0.60	0.11	0.62	0.03	0.10	0.06
MERK	0.96	0.95	0.95	0.90	0.85	0.20	0.01	0.26	0.12	0.20	0.20
NRK	≈0	≈0	≈0	0.02	≈0	0.01	0.01	≈0	≈0	0.01	0.004
NMERK	0.96	0.95	0.95	0.92	0.91	0.91	0.12	0.92	0.21	0.22	0.21
MENK	0.96	0.95	0.95	0.94	0.91	0.10	0.01	0.26	0.12	0.21	0.20
NNK	≈0	≈0	≈0	0.02	≈0	0.01	0.01	≈0	≈0	0.01	≈0
NMENK	0.96	0.95	0.95	0.95	0.91	0.91	0.12	0.94	0.21	0.22	0.21
PD	0.93	0.82	0.84	0.58	0.52	0.66	0.12	0.72	0.10	0.09	0.13
MEPD	0.96	0.94	0.95	0.95	0.89	0.91	0.12	0.91	0.21	0.22	0.24
PDM	0.10	0.06	0.85	0.37	0.20	0.58	0.12	0.27	0.12	0.16	0.03
MEPDM	0.96	0.95	0.95	0.94	0.92	0.92	0.12	0.94	0.21	0.22	0.24
SR	–	–	–	0.21	0.14	–	–	0.26	–	–	–
MESR	–	–	–	0.48	0.28	–	–	0.50	–	–	–
CN	0.24	0.04	≈0	0.39	0.16	0.10	0.01	0.36	0.12	0.12	0.14
AA	0.25	0.04	≈0	0.41	0.21	0.09	0.01	0.61	0.12	0.13	0.14
HPLP+	0.92	0.86	0.88	0.81	0.71	0.81	**0.21**	0.73	0.18	0.22	0.20
SRW	0.95	0.87	0.87	0.82	0.70	0.83	0.20	0.75	0.18	**0.24**	0.21

Table 3: Comparison of link predictive precision among 27 various algorithms.

that the Moore-Penrose pseudo-inverse of the graph Laplacian is a kernel, also called commute time kernel. This kernel is wildly used as a similarities measure between nodes of a graph [10, 37]. However, very recently, Luxburg, et al. [28] show that the commute time distance does not consider the structure of the graph and it will converge to a meaningless distance measure on a graph. This results indicate that the commute time kernel is not very well for measure the similarity between nodes of a graph. The other graph kernel is Neumann kernel [11], which can be expressed as infinite series of matrix powers. The Neumann kernel is closely related to the random walk with restart, which is well known for measure the importance of nodes on a graph. It is important to note that all the graph kernels mentioned above are closely related to the transitional random walk on graph. We study new graph kernels based on MERW.

7. CONCLUSION

In this paper, we propose a set of unsupervised link prediction methods that incorporate the centrality of nodes of the graph based on MERW. We first study certain important properties of MERW by constructing a new eigen-weighted graph. Specifically, based on the eigen-weighted graph, we give a class of new graph Laplacians, namely maximal entropy graph Laplacians, and show that the hitting and commute time of MERW can be computed using the pseudo-inverse of the maximal entropy combinatorial Laplacian. Then, we define four types of graph kernels and two similarity measures on graph based on MERW for link prediction. Finally, we compare 27 various link prediction algorithms over 11 diverse datasets, and show our newly proposed MERW based method (NMEDK) outperforms all the other unsupervised approaches as well as the supervised methods on most datasets. Future work includes generalizing the MERW based methods to directed graphs and exploring MERW for other data mining and machine learning applications.

Appendix A

We claim some properties of pseudo-inverse of MECL as follows: (1) $\mathbf{L}^+$ is a positive semidefinite matrix. (2)$\mathbf{L}^+$ has rank $n-1$. (3)$\mathbf{L}^+$ is doubly centered. Since MECL is equal to the graph Laplacian of the corresponding eigen-weighted graph, MECL shares the properties of the ordinary weighted graph Laplacian. Hence, the claims hold. In addition, we give a lemma [20] in order to prove the Theorem 3.3.

Lemma 7.1: *For any irreducible matrix $\mathbf{A} \in R^{n\times n}$ with spectral radius $\rho(\mathbf{A}) = r$, if $\mathbf{A_i}$ is a principal submatrix of $\mathbf{A}$ obtained by removing the i-th row and column of blocks, then matrix $r\mathbf{I} - \mathbf{A_i}$ is nonsingular.* □

With the properties and Lemma 7.1, we can prove Theorem 3.3. First, we reformulate Eq. (10) into a vector form as $h = e + \mathbf{P}h$, where e is a vector with elements all one, h is the hitting time vector, and $\mathbf{P}$ is the transition matrix. Without loss of generality, we remove the n-th row of vector h and e, and remove the n-th row and n-th column of matrix $\mathbf{A}$, and denote them by $\tilde{h}$, $\tilde{e}$, and $\tilde{\mathbf{A}}$ respectively. We can obtain $\tilde{h} = \tilde{e} + \tilde{\mathbf{P}}\tilde{h}$, where $\tilde{\mathbf{P}} = \tilde{\mathbf{D}}_{\mathbf{v}}^{-1}\tilde{\mathbf{A}}\tilde{\mathbf{D}}_{\mathbf{v}}/\lambda$. Furthermore, let $\tilde{\mathbf{L}} = \tilde{\mathbf{D}}_{\mathbf{v}}^2 - \tilde{\mathbf{D}}_{\mathbf{v}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}_{\mathbf{v}}/\lambda$, we have $\tilde{\mathbf{L}}\tilde{h} = \tilde{\mathbf{D}}_{\mathbf{v}}^2\tilde{e}$. According to Lemma 7.1, $\tilde{\mathbf{L}}$ is nonsingular. We thus get $\tilde{h} = \tilde{\mathbf{L}}^+\tilde{\mathbf{D}}_{\mathbf{v}}^2\tilde{e} = \tilde{\mathbf{L}}^{-1}\tilde{\mathbf{D}}_{\mathbf{v}}^2\tilde{e}$. With the same techniques used in [10], we can efficiently solve the pseudo-inverse of $\tilde{\mathbf{L}}$ ($\tilde{\mathbf{L}}^+$) by computing the pseudo-inverse of $\mathbf{L}$ ($\mathbf{L}^+$). We have

$$h(i,n) = \sum_{j=1}^{n} (L_{ij}^+ - L_{in}^+ - L_{nj}^+ + L_{nn}^+)v_j^2$$

More generally, we have

$$h(i,k) = \sum_{j=1}^{n} (L_{ij}^+ - L_{ik}^+ - L_{kj}^+ + L_{kk}^+)v_j^2$$

We can further compute the commute time as follows:

$$c(i,k) = h(i,k) + h(k,i) = (L^+_{ii} + L^+_{kk} - 2L^+_{ik}) \sum_{j=1}^{n} v_j^2$$
$$= (L^+_{ii} + L^+_{kk} - 2L^+_{ik})$$

Since we normalize the dominate eigenvector v, the last equality holds.

Acknowledgments

The work was supported by grants of the Research Grants Council of the Hong Kong SAR, China No. 419008 and 419109.

8. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3), 2003.

[2] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *WSDM' 11*.

[3] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[4] D. Boley, G. Ranjan, and Z.-L. Zhang. Commute times for a directed graph using an asymmetric laplacian. Technical report, Computer Science and Engineering, University of Minnesota, 2010.

[5] P. Bonacich. Power and centrality: A family of measures. *The American Journal of Sociology*, 92(5):1170–1182, 1987.

[6] Z. Burda, J. Duda, J. M. Luck, and B. Waclaw. Localization of the maximal entropy randomwalk. *Physical Review Letters*, 102(16), 2009.

[7] F. R. K. Chuang. *Spectral Graph Theory*. Regional Conference Series in mathematics. American Mathematical Society, 1997.

[8] T. M. Cover and J. A. Thomas. *Elements of Information Theory 2nd Edition*. Wiley-Interscience, 2006.

[9] J.-C. Delvenne and A.-S. Libert. Centrality measures and thermodynamic formalism for complex networks. 2010.

[10] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *TKDE*, 19(3), 2007.

[11] T. Ito, M. Shimbo, T. Kudo, and Y. Matsumoto. Application of kernels to link analysis. In *KDD' 05*.

[12] G. Jeh and J. Widom. Scaling personalized web search. In *WWW' 03*.

[13] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD' 02*.

[14] H. Jeong, S. P. Mason, A.-L. Barabĺćsi1, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411, 2001.

[15] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 1953.

[16] D. J. Klein and M. Randic. Resistance distance. *Journal of Mathematical Chemistry*, 12(1):81–95, 1993.

[17] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML' 02*.

[18] J. Kunegis, D. Fay, and C. Bauckhage. Network growth and the spectral evolution model. In *CIKM' 10*.

[19] J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *ICML' 10*.

[20] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[21] J. Leskovec. Standford network analysis project. *http://snap.standford.edu*, 2010.

[22] M. Li, B. M. Dias, I. Jarman, W. El-Deredy, and P. J. Lisboa. Grocery shopping recommendations based on basket-sensitive random walk. KDD' 09.

[23] P. Li, H. Liu, J. X. Yu, J. He, and X. Du. Fast single-pair simrank computation. In *SDM' 10*.

[24] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[25] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *KDD' 10*.

[26] L. Lovasz. Random walks on graphs: A survey. *Combinatorics*, 2:1–46, 1996.

[27] L. Lu and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2010.

[28] U. v. Luxburg, A. Radl, and M. Hein. Getting lost in space: Large sample analysis of the commute distance. In *NIPS'10*.

[29] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(3), 2006.

[30] M. E. J. Newman. *Networks: An Introduction*. OXFORD University Press, 2010.

[31] W. PARRY. Intrinsic markov chains. *Transactions of the American Mathematical Society*, 112:55–66, 1964.

[32] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. KDD '10, 2010.

[33] D. Ruelle. *Thermodynamic Formalism*. Addison-Wesley, Reading Mass., 1978.

[34] R. Sinatra, J. Gomez-Gardenes, R. Lambiotte, V. Nicosia, and V. Latora. Optimal random walks in complex networks with limited information. *arXiv:1007.4936v1*, 2010.

[35] A. J. Smola and R. I. Kondor. Kernels and regularization on graphs. In *COLT' 03*.

[36] D. J. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:409–410, 1998.

[37] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens. Graph nodes clustering with the sigmoid commute-time kernel: A comparative study. *Data and Knowledge Engineering 68*, 68:338–361, 2009.

[38] L. Yen, A. Mantrach, and M. Shimbo. A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances. In *KDD' 08*.