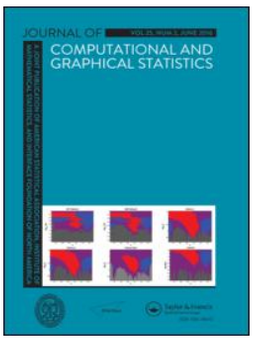

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: http://amstat.tandfonline.com/loi/ucgs20

Link Prediction for Partially Observed Networks

Yunpeng Zhao, Yun-Jhong Wu, Elizaveta Levina & Ji Zhu

To cite this article: Yunpeng Zhao, Yun-Jhong Wu, Elizaveta Levina & Ji Zhu (2017): Link Prediction for Partially Observed Networks, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2017.1286243

To link to this article: http://dx.doi.org/10.1080/10618600.2017.1286243

Accepted author version posted online: 26 Jan 2017.
Published online: 26 Jan 2017.

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2017, VOL. 0, NO. 0, 1–9
https://doi.org/10.1080/10618600.2017.1286243

Taylor & Francis
Taylor & Francis Group

Link Prediction for Partially Observed Networks

Yunpeng Zhao[a], Yun-Jhong Wu[b], Elizaveta Levina[b], and Ji Zhu[b]

[a]Department of Statistics, George Mason University, Fairfax, VA; [b]Department of Statistics, University of Michigan, Ann Arbor, MI

ABSTRACT

Link prediction is one of the fundamental problems in network analysis. In many applications, notably in genetics, a partially observed network may not contain any negative examples, that is, edges known for certain to be absent, which creates a difficulty for existing supervised learning approaches. We develop a new method that treats the observed network as a sample of the true network with different sampling rates for positive (true edges) and negative (absent edges) examples. We obtain a relative ranking of potential links by their probabilities, using information on network topology as well as node covariates if available. The method relies on the intuitive assumption that if two pairs of nodes are similar, the probabilities of these pairs forming an edge are also similar. Empirically, the method performs well under many settings, including when the observed network is sparse. We apply the method to a protein–protein interaction network and a school friendship network.

ARTICLE HISTORY
Received September 2015
Revised September 2016

KEYWORDS
Link prediction; Ranking; Social networks

1. Introduction

A variety of data in many different fields can be described by networks. Examples include friendship and social networks, food webs, protein–protein interaction and gene regulatory networks, the World Wide Web, and many others. One of the fundamental problems in network science is link prediction, where the goal is to predict the existence of a link between two nodes based on observed links between other nodes as well as additional information about the nodes (node covariates) when available (see Getoor and Diehl 2005; Liben-Nowell and Kleinberg 2007; Lu and Zhou 2010; Hasan and Zaki 2011 for recent reviews). Link prediction has wide applications. For example, recommendation of new friends or connections for members is an important service in online social networks such as Facebook. In biological networks, such as protein–protein interaction and gene regulatory networks, it is usually time-consuming and expensive to test existence of links by comprehensive experiments; link prediction in these biological networks can provide specific targets for future experiments.

There are two different settings under which the link prediction problem is commonly studied. In the first setting, a snapshot of the network at time t, or a sequence of snapshots at times $1, \ldots, t$, is used to predict new links that are likely to appear in the near future (at time $t + 1$). In the second setting, the network is treated as static but not fully observed, and the task is to fill in the missing links in such a partially observed network. These two tasks are related in practice, since a network evolving over time can also be partially observed and a missing link may emerge in the future. From the analysis point of view, however, these settings are quite different; in this article, we focus on the partially observed setting and do not consider networks evolving over time.

There are several types of methods for the link prediction problem in the literature. The first class of methods consists of unsupervised approaches based on various types of node similarities. These methods assign a similarity score $s(i, j)$ to each pair of nodes i and j, and higher similarity scores are assumed to imply higher probabilities of a link. Similarities can be based either on node attributes or solely on the network structure, such as the number of common neighbors; the latter are known as structural similarities. Typical choices of structural similarity measures include local indices based on common neighbors, such as the Jaccard index (Liben-Nowell and Kleinberg 2007) or the Adamic–Adar index (Adamic and Adar 2003), and global indices based on the ensemble of all paths, such as the Katz index (Katz 1953) and the Leicht–Holme–Newman Index (Leicht, Holme, and Newman 2006). Comprehensive reviews of such similarity measures can be found in Liben-Nowell and Kleinberg (2007) and Lu and Zhou (2010).

Another class of approaches to link prediction applies supervised learning methods that use both network structures and node attributes. These methods treat link prediction as a binary classification problem, where the responses are $\{1, 0\}$ indicating whether there exists a link for a pair, and the predictors are covariates for each pair, which are constructed from node attributes. A number of popular supervised learning methods have been applied to the link prediction problem. For example, Ben-Hur and Noble (2005) and Bleakley, Biau, and Vert (2007) used the support vector machine with pairwise kernels, and Hasan et al. (2006) compared the performance of several supervised learning methods. Menon and Elkan (2011) proposed a criterion optimizing AUC (area under the curve) to address the imbalance problem of positive and negative examples. Kashima et al. (2009) and Raymond and Kashima (2010) used

CONTACT Yunpeng Zhao yzhao15@gmu.edu Department of Statistics, George Mason University, Fairfax, VA 22030, USA.

semisupervised learning approaches for link prediction. Other supervised methods use probabilistic models for incomplete networks to do link prediction, for example, the hierarchical structure models (Clauset, Moore, and Newman 2008), latent space models (Hoff, Raftery, and Handcock 2002), latent variable models (Hoff 2007; Miller, Griffiths, and Jordan 2009), stochastic relational models (Yu et al. 2007), multi-relational influence propagation (Yang et al. 2012), etc.

Our approach falls in the supervised learning category, in the sense that we make use of both the node similarities and observed links. However, one difficulty in treating link prediction as a straightforward classification problem is the lack of certainty about the negative and positive examples. This is particularly true for negative examples (when we know for certain no edge exists between a pair of nodes. In biological networks in particular, there may be no negative examples at all (Ben-Hur and Noble 2006). For instance, in a protein–protein interaction network, a pair with no observed edge may not mean that there is no interaction between the two proteins—instead, it may indicate that the experiment to test that interaction has not been done, or that it did not have enough sensitivity to detect the interaction. Positive examples could sometimes also be spurious—for example, high-throughput experiments can yield a large number of false positive protein–protein interactions (von Mering et al. 2002). Here, we propose a new link prediction method that allows for the presence of both false positive and false negative examples. More formally, we assume that we observe the true network with independent observation errors, that is, with some true edges missing and other edges recorded erroneously. The error rates for both kinds of errors are assumed unknown, and in fact cannot be estimated under this framework. However, we can rank potential links in order of their estimated probabilities, for node pairs with observed links as well as for node pairs with no observed links. These relative rankings rather than absolute probabilities of edges are sufficient in many applications. For example, pairs of proteins without observed interactions that rank highly could be given priority in subsequent experiments. To obtain these rankings, we use node covariates when available, and/or network topology based on observed links.

We note that statistical analysis of networks with edge errors has also recently been studied in other settings. For instance, Priebe et al. (2015) considered the node classification problem on networks with observation errors. Balachandran, Airoldi, and Kolaczyk (2013) and Balachandran, Kolaczyk, and Viles (2014) focused on inference problems of certain summary statistics on noisy networks. To the best of our knowledge, the current article is the first one to study the link prediction problem on networks with observation errors.

The rest of the article is organized as follows. In Section 2, we specify our (rather minimal) model assumptions for the network and the edge errors. We propose link ranking criteria for both directed and undirected networks in Section 3. The algorithms used to optimize these criteria are discussed in Section 4. In Section 5, we test the performance of proposed criteria on simulated networks. In Section 6, we apply our methods to link prediction in a protein–protein interaction network and a school friendship network. Section 7 concludes with a summary and discussion of future directions.

2. The Network Model

A network with n nodes (vertices) can be represented by an $n \times n$ adjacency matrix $A = [A_{ij}]$, where

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j, \\ 0 & \text{otherwise.} \end{cases}$$

We will consider the link prediction problem for both undirected and directed networks. Therefore, A can be either symmetric (for undirected networks) or asymmetric (for directed networks).

In our framework, we distinguish between the adjacency matrix of the true underlying network A, and its observed version $\widetilde{A}$. We assume that each A_{ij} follows a Bernoulli distribution with $\mathbb{P}(A_{ij} = 1) = P_{ij}$. Given the true network, we assume that the observed network is generated by

$$\mathbb{P}(\widetilde{A}_{ij} = 1 | A_{ij} = 1) = \alpha_{ij}, \quad \mathbb{P}(\widetilde{A}_{ij} = 0 | A_{ij} = 0) = \beta_{ij},$$

where α_{ij} and β_{ij} are the probabilities of correctly recording a true edge and an absent edge, respectively. Then we have

$$\widetilde{P}_{ij} \equiv \mathbb{P}(\widetilde{A}_{ij} = 1) = (\alpha_{ij} + \beta_{ij} - 1)P_{ij} + (1 - \beta_{ij}). \tag{2.1}$$

2.1. The Simple Case: Constant Sampling Rates

We first assume that $\alpha_{ij} = \alpha$ and $\beta_{ij} = \beta$ are constants and do not depend on i, j, or P_{ij}. Note that if the values of α, β, and P_{ij} were known, then the probabilities of true edges conditional on the observed adjacency matrix could have been estimated as

$$\mathbb{P}(A_{ij} = 1 | \widetilde{A}_{ij} = 1) = \frac{\alpha P_{ij}}{\widetilde{P}_{ij}}, \tag{2.2}$$

$$\mathbb{P}(A_{ij} = 1 | \widetilde{A}_{ij} = 0) = \frac{(1 - \alpha)P_{ij}}{1 - \widetilde{P}_{ij}}. \tag{2.3}$$

It is easy to check that both (2.2) and (2.3) are monotone increasing functions of P_{ij}. Taking (2.1) into account implies that they are also increasing functions of $\widetilde{P}_{ij}$ as long as $\alpha + \beta > 1$. This gives us a crucial observation: if the goal is to obtain relative rankings of potential links, it is sufficient to estimate $\widetilde{P}_{ij}$, and it is not necessary to know α, β, and P_{ij}.

An important special case in this setting is $\beta = 1$. Then all the observed links are true positive examples, for example, in a protein–protein interaction network, all observed links have been verified by experiments. In that case, we only need to provide a ranking for node pairs without observed links. This can be applied in recommender systems, for example, for recommending possible new friends in a social network. Another special case is when $\alpha = 1$, which corresponds to all unobserved edges being true negatives. This setting can be used in the problem of investigating reliability of observed links, for example, in a gene regulatory network inferred from high-throughput gene expression data.

2.2. The General Case: Nonconstant Sampling Rates

Note that (2.2) and (2.3) still hold when α and β are nonconstant, with α and β replaced with α_{ij} and β_{ij}, respectively. If both (2.2) and (2.3) are monotone increasing functions of P_{ij}, and $\widetilde{P}_{ij}$ is also monotone increasing in P_{ij}, then again, conditional on the

observed adjacency matrix, relative rankings of potential links can be obtained from an estimate of $\widetilde{P}_{ij}$ without knowing α_{ij} and β_{ij}.

Now suppose that α_{ij} and β_{ij} are differentiable functions of P_{ij}. By taking derivatives with respect to P_{ij} in (2.1), (2.2), and (2.3), it is straightforward to show that the following conditions ensure the required monotonicities:

$$\alpha_{ij} + \beta_{ij} - 1 + \alpha'_{ij}P_{ij} - \beta'_{ij}(1 - P_{ij}) \geq 0,$$
$$\beta_{ij}(1 - \alpha_{ij}) + (\alpha_{ij}\beta'_{ij} - \alpha'_{ij}\beta_{ij} - \beta'_{ij})P_{ij}(1 - P_{ij}) \geq 0,$$
$$(1 - \beta_{ij})\alpha_{ij} + (\alpha_{ij}\beta'_{ij} - \alpha'_{ij}\beta_{ij} + \alpha'_{ij})P_{ij}(1 - P_{ij}) \geq 0,$$

where α'_{ij} and β'_{ij} denote derivatives of α_{ij} and β_{ij} with respect to P_{ij}, respectively. These conditions seem complicated, but they cover a wide range of possibilities. For example, when β_{ij} is a constant, for example, $\beta_{ij} = 1$, and α_{ij} is a power function of P_{ij}, that is, $\alpha_{ij} = u + vP_{ij}^{w}$ with $u, v, w \geq 0$ and $u + v \leq 1$ (which implies that links with high probability of appearing in the true network are also more likely to be observed, a fairly natural assumption), the above conditions are satisfied. Similarly, when α_{ij} is a constant, for example, $\alpha_{ij} = 1$, and β_{ij} is a power function of P_{ij}, that is, $\beta_{ij} = u + v(1 - P_{ij})^{w}$ with $u, v, w \geq 0$ and $u + v \leq 1$ (which implies that links with low probability of appearing in the true network are more likely to be recorded as absent edges), the above conditions are also satisfied.

These results imply that for a wide range of scenarios, conditional on the observed adjacency matrix, an estimate of the matrix $\widetilde{P}$ provides rankings for potential links, and thus we focus on estimating $\widetilde{P}$ for the rest of the article.

3. Link Prediction Criteria

In this section, we propose criteria for estimating the probabilities of edges in the observed network, $\widetilde{P}_{ij}$, for both directed and undirected networks. The criteria rely on a symmetric matrix $W = [W_{ij}]$, $0 \leq W_{ij} \leq 1$, which describes the similarity between nodes i and j. The similarity matrix W can be obtained from different sources, including node information, network topology, or a combination of the two. We discuss possible choices for W in Section 3.3.

3.1. Link Prediction for Directed Networks

First, we consider directed networks. The key assumption we make is that if two pairs of nodes are similar to each other, the probability of links within these two pairs are also similar. Specifically, in Figure 1, P_{ij} and $P_{i'j'}$ are assumed close in value if node i is similar to node i' and node j is similar to node j'.

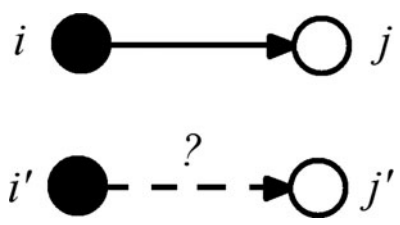

Figure 1. Pair similarity for directed networks.

For directed networks, we measure similarity of node pairs (i, i') and (j, j') by the product $W_{ii'}W_{jj'}$, which implies two pairs are similar only if both pairs of endpoints are similar. This assumption should not to be confused with a different assumption made by many unsupervised link prediction methods, which assume that a link is more likely to exist between similar nodes (i.e., high W_{ij} corresponds to high P_{ij}, instead of our assumption of high $W_{ii'}W_{jj'}$ corresponding to small $|P_{ij} - P_{i'j'}|$). This assumption is applicable to networks with assortative mixing, which are common—a typical example is a social network, where people tend to be friends with those of similar age, income level, race, etc. However, there are also networks with disassortative mixing, in which the assumption that similar pairs are more likely to be connected is no longer valid—for example, predators do not typically feed on each other in a food web. Our assumption, in contrast, is equally plausible for both assortative and disassortative networks, as well as more general settings, as it does not assume anything about the relationship between P_{ij} and W_{ij}.

Motivated by this assumption of similar probabilities of links for similar node pairs, we propose to estimate $\widetilde{P}_{ij} = E(\widetilde{A}_{ij})$ by

$$\widehat{F} = \arg\min_{F} \frac{1}{n^2}\sum_{ij}^{n}(\widetilde{A}_{ij} - F_{ij})^2 + \frac{\lambda}{n^4}\sum_{ii'jj'}^{n} W_{ii'}W_{jj'}(F_{ij} - F_{i'j'})^2, \tag{3.1}$$

where F is a real-valued $n \times n$ matrix, and λ is a tuning parameter. The first term is the usual squared error loss connecting the parameters with the observed network. The minimizer of its population version $\mathbb{E}(\widetilde{A}_{ij} - F_{ij})^2$ is $\widetilde{P}_{ij}$. The second term enforces our key assumption, penalizing the difference between F_{ij} and $F_{i'j'}$ proportionately to the similarity between node pairs (i, i') and (j, j'). The choice of the squared error loss is not crucial, and other commonly used loss functions could be considered instead, for example, the hinge loss or the negative log-likelihood. The main reason for choosing the squared error loss is computational efficiency, since it makes (3.1) a quadratic problem; see more details on this in Section 4.

In some applications, we may have additional information about true positive and negative examples, that is, some $\widetilde{A}_{ij}$'s may be known to be true 1's and true 0's, while others may be uncertain. This could happen, for example, when validation experiments have been conducted on a subset of a gene or protein network inferred from expression data. If such information is available, it makes sense to use it, and we can then modify criterion (3.1) as follows:

$$\arg\min_{F} \frac{1}{\sum_{ij}^{n} E_{ij}}\sum_{ij}^{n} E_{ij}(\widetilde{A}_{ij} - F_{ij})^2 + \frac{\lambda}{n^4}\sum_{ii'jj'}^{n} W_{ii'}W_{jj'}(F_{ij} - F_{i'j'})^2, \tag{3.2}$$

where $E_{ij} = 1$ if it is known that $\widetilde{A}_{ij} = A_{ij}$, and 0 otherwise. This is similar to a semisupervised criterion proposed in Kashima et al. (2009). However, Kashima et al. (2009) did not consider the uncertainty in positive and negative examples, nor did they consider the undirected case which we discuss next. Since (3.2) only involves a partial sum of the loss function terms, we will refer to

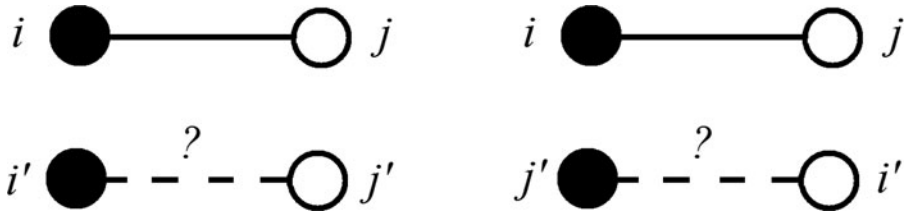

Figure 2. Pair similarity for undirected networks.

(3.2) as the partial-sum criterion and (3.1) as the full-sum criterion for the rest of the article. It is worth noting that the partial-sum criterion (3.2) can be further generalized for networks with epistemic certainty on edges, that is, E_{ij} can take a continuous value from 0 to 1, which reflects the certainty about the edge or nonedge status between i and j.

3.2. Link Prediction for Undirected Networks

For undirected networks, our key assumption that P_{ij} and $P_{i'j'}$ are close if two pairs (i, i') and (j, j') are similar needs to take into account that the direction no longer matters; thus the pairs are similar if either i is similar to i' and j is similar to j', or if i is similar to j' and j is similar to i' (see Figure 2). Thus, we need a new pair similarity measure that combines $W_{ii'}W_{jj'}$ and $W_{ij'}W_{ji'}$. There are multiple options, for example, two natural combinations are

$$S_1 = W_{ii'}W_{jj'} + W_{ij'}W_{ji'} \text{ and } S_2 = \max(W_{ii'}W_{jj'}, W_{ij'}W_{ji'}).$$

Empirically, we found that S_2 performs better than S_1 for a range of real and simulated networks. The reason for this can be easily illustrated on the stochastic block model. The stochastic block model is a commonly used model for networks with communities, where the probability of a link only depends on the community labels of its two endpoints. Specifically, given community labels $c = \{c_1, \ldots, c_n\}$, A_{ij}'s are independent Bernoulli random variables with

$$P_{ij} = B_{c_i c_j}, \tag{3.3}$$

where $B = [B_{ab}]$ is a $K \times K$ symmetric matrix, and K is the number of communities in the network. Suppose we have the best similarity measure we can possibly hope to have based on the truth, $W_{ij} = I(c_i = c_j)$, where I is the indicator function. In that case, (3.3) implies $P_{ij} = P_{i'j'}$ if $\max(W_{ii'}W_{jj'}, W_{ij'}W_{ji'}) = 1$, whereas the sum of the weights would be misleading.

Using S_2 as the measure of pair similarity, we propose estimating $\widetilde{P}_{ij}$ for undirected networks by

$$\arg\min_F \frac{1}{n^2}\sum_{i<j}^{n}(\widetilde{A}_{ij} - F_{ij})^2 + \frac{\lambda}{n^4}\sum_{i<j,i'<j'}^{n}\max(W_{ii'}W_{jj'}, W_{ij'}W_{ji'})(F_{ij} - F_{i'j'})^2. \tag{3.4}$$

Similarly to the directed case, if we have information about true positive and negative examples, we can use a partial-sum criterion instead,

$$\frac{1}{\sum_{i<j}^{n} E_{ij}}\sum_{i<j}^{n} E_{ij}(\widetilde{A}_{ij} - F_{ij})^2 + \frac{\lambda}{n^4}\sum_{i<j,i'<j'}^{n}\max(W_{ii'}W_{jj'}, W_{ij'}W_{ji'})(F_{ij} - F_{i'j'})^2, \tag{3.5}$$

where $E_{ij} = 1$ if it is known that $\widetilde{A}_{ij} = A_{ij}$, otherwise $E_{ij} = 0$.

3.3. Node Similarity Measures

The last component of the method we need to specify is the node similarity matrix W. If we have available node covariates and reasons to believe that they are related to the structure of the network, it is natural to use covariate information to construct W_{ij}. Though more complicated formats do exist, node covariates are typically represented by an $n \times p$ matrix X where X_{ik} is the value of variable k for node i. Then W_{ij} can be taken to be some similarity measure between the ith and jth rows of X. For example, if X contains only numerical variables and has been standardized, we can use the exponential decay kernel,

$$W_{ij} = \exp\left\{-\frac{\|X_{i\cdot} - X_{j\cdot}\|^2}{\sigma^2}\right\},$$

where $\|\cdot\|$ is the Euclidean vector norm.

When node covariates are not available, node similarity W_{ij} is usually obtained from the topology of the observed network $\widetilde{A}$, that is, W_{ij} is large if i and j have a similar pattern of connections with other nodes. For undirected networks, a simple choice of W_{ij} could be

$$W_{ij} = \frac{|\{k : \widetilde{A}_{ik} = \widetilde{A}_{jk}\}|}{n}, \tag{3.6}$$

where $|\cdot|$ denotes cardinality of a set. This particular measure turns out to be not very useful: since most real networks are sparse, most entries of any kth column will be 0, and thus most of $W_{ii'}$'s would be large. A more informative measure is the Jaccard index (Liben-Nowell and Kleinberg 2007),

$$W_{ij} = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}, \tag{3.7}$$

where $N(i) = \{k : \widetilde{A}_{ik} = 1\}$ is the set of neighbors of node i. Note that the Jaccard index is particularly useful in sparse networks, since when using the Jaccard index, two nodes i and j would have a high similarity only if they share many neighbors,

that is, $A_{i\cdot}$ and $A_{j\cdot}$ share many 1's. By contrast, i and j will not be considered as similar only because the corresponding rows of the adjacency matrix share many 0's, which is often the case in sparse networks.

The directed network case is similar, except we need to count the "in" and the "out" links separately. The formulas corresponding to (3.6) and (3.7) become

$$W_{ij} = \frac{|\{k : \widetilde{A}_{ik} = \widetilde{A}_{jk}\}|}{2n} + \frac{|\{k : \widetilde{A}_{ki} = \widetilde{A}_{kj}\}|}{2n},$$
$$W_{ij} = \frac{|N_1(i) \cap N_1(j)|}{2|N_1(i) \cup N_1(j)|} + \frac{|N_2(i) \cap N_2(j)|}{2|N_2(i) \cup N_2(j)|}, \quad (3.8)$$

where $N_1(i) = \{k : \widetilde{A}_{ik} = 1\}$ and $N_2(i) = \{k : \widetilde{A}_{ki} = 1\}$.

4. Optimization Algorithms

The proposed link prediction criteria are convex and quadratic in parameters, and thus optimization is fairly straightforward. The obvious approach is to treat the matrix f as a long vector with n^2 elements (or $n(n-1)/2$ in the undirected case), and solve the linear system obtained by taking the first derivative of any criterion above with respect to this vector. However, solving a system of linear equations could be challenging for large-scale problems (Boyd and Vandenberghe 2004); the number of parameters here is $O(n^2)$, and so the linear system requires $O(n^4)$ memory. However, if W is sparse, or sparsified by applying thresholding or some other similar method, then solving the linear system is the efficient choice.

If the W matrix is not sparse, an iterative algorithm with sequential updates that only requires $O(n^2)$ memory would be a better choice than solving the linear system. We propose an iterative algorithm following the idea of block coordinate descent (Hildreth 1957; Warga 1963). A block coordinate descent algorithm partitions the coordinates into blocks and iteratively optimizes the criterion with respect to each block while holding the other blocks fixed.

First, we derive the update equations for directed networks. Note that the objective functions in (3.1) and (3.2) can be written in the general form

$$Q = \frac{1}{n^2}\sum_{ij}^{n} V_{ij}(\widetilde{A}_{ij} - F_{ij})^2 + \frac{\lambda}{n^4}\sum_{ii'jj'}^{n} W_{ii'}W_{jj'}(F_{ij} - F_{i'j'})^2, \quad (4.1)$$

where $V_{ij} \equiv 1$ for (3.1) and $V_{ij} = E_{ij}$ for (3.2). Note that for simplicity of further derivations, we replaced the denominators in (3.1) and (3.2) by n_2 and n_4, respectively, absorbing the terms with V_{ij} into the tuning parameters. For any matrix M, let $M_{i\cdot}$ be the ith row of M. We treat $F_{i\cdot}$ as a block, and update $F_{i\cdot}$ iteratively. Define $V_i = \mathrm{diag}(V_{i\cdot})$. Then

$$\sum_{ij} V_{ij}(F_{ij} - \widetilde{A}_{ij})^2 = \sum_{i}(F_{i\cdot} - \widetilde{A}_{i\cdot})^T V_i (F_{i\cdot} - \widetilde{A}_{i\cdot}). \quad (4.2)$$

Let D be the $n \times n$ diagonal matrix with $D_{ii} = \sum_j W_{ij}$. Then

$$\sum_{jj'} W_{jj'}(F_{ij} - F_{i'j'})^2 = F_{i\cdot}^T D F_{i\cdot} - 2F_{i\cdot}^T W F_{i'\cdot} + F_{i'\cdot}^T D F_{i'\cdot}. \quad (4.3)$$

Plugging (4.2) and (4.3) into (4.1), and taking the first derivative of Q with respect to $F_{i\cdot}$, we obtain

$$\frac{\partial Q}{\partial F_{i\cdot}} = \frac{2}{n^2}V_i(F_{i\cdot} - \widetilde{A}_{i\cdot}) + \lambda\frac{4}{n^4}\left[W_{ii}(DF_{i\cdot} - WF_{i\cdot}) + \sum_{i'\neq i} W_{ii'}(DF_{i\cdot} - WF_{i'\cdot})\right]. \quad (4.4)$$

Setting $\frac{\partial Q}{\partial F_{i\cdot}} = 0$ and solving for $F_{i\cdot}$, we obtain the updating formula

$$F_{i\cdot}^{(t+1)} \leftarrow \left(n^2 V_i + 2\lambda\sum_{i'} W_{ii'}D - 2\lambda W_{ii}W\right)^{-1} \times \left(n^2 V_i \widetilde{A}_{i\cdot} + 2\lambda\sum_{i'\neq i} W_{ii'}WF_{i'\cdot}^{(t)}\right), \quad (4.5)$$

where $F_{i\cdot}^{(t)}$ is the value of $F_{i\cdot}$ at iteration t.

This update is fast to compute but its derivation relies on the product form of $W_{ii'}$ and $W_{jj'}$, and thus is not directly applicable in the undirected case, where S_2 is used as the similarity measure. However, we can still approximate S_2 with a product, using the fact that for $x \geq 0, y \geq 0$, $\lim_{q\to\infty}\sqrt[q]{x^q + y^q} = \max(x, y)$. Thus, for sufficiently large q, we have

$$[\max(W_{ii'}W_{jj'}, W_{ij'}W_{ji'})]^q \approx (W_{ii'}W_{jj'})^q + (W_{ij'}W_{ji'})^q. \quad (4.6)$$

Further, W^q is a monotone transformation of W and can also serve as a similarity measure. Based on (4.6), we propose to substitute the following approximate objective function for undirected networks,

$$Q = \frac{1}{n^2}\sum_{i<j}^{n} V_{ij}(\widetilde{A}_{ij} - F_{ij})^2 + \frac{\lambda}{n^4}\sum_{i<j,i'<j'}^{n}((W_{ii'}W_{jj'})^q + (W_{ij'}W_{ji'})^q)(F_{ij} - F_{i'j'})^2, \quad (4.7)$$

where $V_{ij} \equiv 1$ for the full sum criterion and $V_{ij} = E_{ij}$ for the partial sum criterion. By symmetry,

$$\sum_{i<j,i'<j'}^{n}((W_{ii'}W_{jj'})^q + (W_{ij'}W_{ji'})^q)(F_{ij} - F_{i'j'})^2 = \frac{1}{2}\sum_{i\neq j,i'\neq j'}^{n} W_{ii'}^q W_{jj'}^q (F_{ij} - F_{i'j'})^2.$$

This is now in the same form as (4.1), with each term in the sum containing a product of $W_{ii'}$ and $W_{jj'}$, and therefore (4.7) can be minimized by block coordinate descent with an analogous updating equation as that in the directed network case. Note that since in general Q is strictly convex, the proposed coordinate descent algorithm has a linear convergence rate (Luo and Tseng 1992), that is,

$$\|F^{(k)} - F\|_F \leq \|F^{(0)} - F\|_F \gamma^k,$$

where $\gamma \in (0, 1)$ and $F^{(k)}$ denotes the estimate after k updating cycles.

In practice, we found that when W is sparse or truncated to be sparse, solving the linear system can be much faster than

the block coordinate descent method; however, when W is dense and the number of nodes is reasonably large, the block coordinate descent method dominates directly solving linear equations.

5. Empirical Evaluation on Simulated Networks

In this section, we test the performance of our link prediction methods on simulated networks. We focus on the setting where $\beta = 1$, that is, there is no false positive link in the network. In all cases, each network consists of $n = 1000$ nodes, and node i's covariates X_i are generated from a Gaussian mixture distribution with 20 components. Specifically, we first generated 20 independent centers μ_k, $k = 1, \ldots, 20$, with $\mu_k \sim N(0, \eta^2 I_p)$, $p = 5$. Then for each center μ_k, we generated 50 independent nodes, with $X_i \sim N(\mu_k, I_p)$. We generated each A_{ij} independently from the distance model in (Hoff, Raftery, and Handcock 2002), which has the form

$$\text{logit}(P_{ij}) = -\|X_i - X_j\| + c, \tag{5.1}$$

where $\|\cdot\|$ is the Euclidean norm, and c (depending on the value of η) was chosen such that the average degree of the generated network is about 20. The above generative model is based on the intuition that nodes close to each other (in terms of their positions, that is, X_i's in the "social space") are more likely to become linked. Note that the proposed method does not need such an assumption. Nevertheless, the above formula can be used to generate both directed and undirected networks; for undirected networks, we set $A_{ij} = A_{ji}$.

We also generated indicators E_{ij}'s as independent Bernoulli variables taking values 1 and 0, with $\mathbb{P}(E_{ij} = 1) = \alpha_{ij}$, and set $\widetilde{A}_{ij} = E_{ij}A_{ij}$. This setup corresponds to $\beta_{ij} = 1$ and the "partially observed" network of the title, where all the observed edges are true but the unobserved edges may or may not be true 0's. We consider two scenarios for the sampling rate: (1) $\alpha_{ij} = 0.8$, which is a constant, and (2) $\alpha_{ij} = P_{ij}^w$, where w was chosen such that the overall sampling rate is about 0.8. Note that the second scenario implies that if two nodes are likely to be linked, then the chance that the true edge is correctly recorded is also high.

Since we have node covariates affecting the probabilities of links in this case, we define the similarity matrix W by

$$W_{ij} = \exp\left\{-\frac{\|X_i - X_j\|^2}{\sigma^2}\right\},$$

where we choose $\sigma = \frac{1}{4}\text{median}\{\|X_i - X_j\|, i, j = 1, \ldots, n\}$. Further, we also include the case of the network-based similarity matrix W, as in (3.7) and (3.8), which is the only option if node covariates are not available. We optimize all criteria by solving linear equations, with λ chosen by five-fold cross-validation. Specifically, in each fold, we set 20% of A_{ij}'s to 0. The tuning parameter λ is then selected such that the overall predictive AUC on the set of selected pairs is maximized.

The performance of link prediction is evaluated on the "test" set $\mathcal{S} = \{(i, j) : E_{ij} = 0\}$. We report the area under the ROC curve (AUC), which only depends on the rankings of the estimates $\widehat{F}_{ij}$ rather than their numerical values. The AUC is defined as

$$\text{AUC} = \frac{\sum_{(i,j),(i',j')\in\mathcal{S}} \left(1\left(\widehat{F}_{ij} > \widehat{F}_{i'j'}\right) + \frac{1}{2}1\left(\widehat{F}_{ij} = \widehat{F}_{i'j'}\right)\right) 1\left(A_{ij} = 1, A_{i'j'} = 0\right)}{\sum_{(i,j),(i',j')\in\mathcal{S}} 1\left(A_{ij} = 1, A_{i'j'} = 0\right)}.$$

The results are shown in Tables 1 and 2. Overall, there is little difference between the partial-sum and full-sum criteria. As expected, as η increases, the centers of the node covariates, that is, μ_k's, become more separated, and the nodes become more "clustered," consequently the difficulty of the problem decreases, and the performance of the proposed method improves. Also as expected, using node covariates to define the similarity matrix always performs better than using the observed network topology, as according to the simulation setup, the link probabilities are directly affected by node covariates. Further, when node covariates are used to compute the similarity matrix, there is little difference between directed network models and their

Table 1. Directed network: mean and standard error of predictive AUC for model (5.1) based on 20 replications.

		W_{ij} (using node covariates)		W_{ij} (using network topology)	
(η^2, c)	α_{ij}	Partial-sum	Full-sum	Partial-sum	Full-sum
$(1, -0.3)$	0.8	0.745 (0.0053)	0.748 (0.0056)	0.595 (0.1073)	0.589 (0.1006)
	$P_{ij}^{0.05}$	0.761 (0.0066)	0.761 (0.0059)	0.591 (0.1156)	0.586 (0.1093)
$(4, 1)$	0.8	0.864 (0.0021)	0.852 (0.0020)	0.549 (0.1157)	0.543 (0.1023)
	$P_{ij}^{0.04}$	0.875 (0.0029)	0.870 (0.0024)	0.740 (0.1571)	0.720 (0.1441)
$(9, 2)$	0.8	0.929 (0.0017)	0.923 (0.0018)	0.791 (0.1851)	0.784 (0.1809)
	$P_{ij}^{0.03}$	0.933 (0.0026)	0.930 (0.0022)	0.907 (0.0052)	0.899 (0.0044)
$(16, 3)$	0.8	0.965 (0.0013)	0.964 (0.0014)	0.944 (0.0014)	0.948 (0.0016)
	$P_{ij}^{0.025}$	0.965 (0.0029)	0.963 (0.0033)	0.944 (0.0064)	0.944 (0.0065)

Table 2. Undirected network: mean and standard error of predictive AUC for generative model (5.1) based on 20 repetitions.

		W_{ij} (using node covariates)		W_{ij} (using network topology)	
(η^2, c)	α_{ij}	Partial-sum	Full-sum	Partial-sum	Full-sum
$(1, -0.3)$	0.8	0.753 (0.0033)	0.746 (0.0043)	0.501 (0.0046)	0.502 (0.0046)
	$P_{ij}^{0.05}$	0.761 (0.0046)	0.760 (0.0047)	0.499 (0.0061)	0.499 (0.0061)
$(4, 1)$	0.8	0.863 (0.0022)	0.849 (0.0020)	0.539 (0.0813)	0.547 (0.0852)
	$P_{ij}^{0.04}$	0.874 (0.0032)	0.868 (0.0026)	0.526 (0.0786)	0.526 (0.0768)
$(9, 2)$	0.8	0.928 (0.0017)	0.921 (0.0019)	0.615 (0.0155)	0.632 (0.0183)
	$P_{ij}^{0.03}$	0.933 (0.0026)	0.930 (0.0025)	0.818 (0.0112)	0.832 (0.0146)
$(16, 3)$	0.8	0.966 (0.0014)	0.963 (0.0014)	0.942 (0.0017)	0.944 (0.0017)
	$P_{ij}^{0.025}$	0.966 (0.0036)	0.965 (0.0037)	0.942 (0.0062)	0.941 (0.0070)

Table 3. Directed network: mean and standard error of predictive AUC for model (5.1) based on 20 replications (with 10% missing rate in the W matrix).

		W_{ij} (using node covariates)		W_{ij} (using network topology)	
(η^2, c)	α_{ij}	Partial-sum	Full-sum	Partial-sum	Full-sum
(1, −0.3)	0.8	0.741 (0.0077)	0.744 (0.0075)	0.581 (0.1142)	0.579 (0.1093)
	$P_{ij}^{0.05}$	0.755 (0.0080)	0.752 (0.0079)	0.590 (0.1145)	0.584 (0.1089)
(4, 1)	0.8	0.853 (0.0021)	0.840 (0.0020)	0.539 (0.1157)	0.540 (0.1023)
	$P_{ij}^{0.04}$	0.852 (0.0031)	0.855 (0.0027)	0.721 (0.1652)	0.718 (0.1503)
(9, 2)	0.8	0.915 (0.0028)	0.911 (0.0025)	0.782 (0.1835)	0.765 (0.1842)
	$P_{ij}^{0.03}$	0.918 (0.0024)	0.915 (0.0032)	0.883 (0.0073)	0.880 (0.0065)
(16, 3)	0.8	0.945 (0.0017)	0.942 (0.0014)	0.923 (0.0016)	0.925 (0.0020)
	$P_{ij}^{0.025}$	0.942 (0.0032)	0.950 (0.0030)	0.945 (0.0066)	0.942 (0.0071)

Table 4. Undirected network: mean and standard error of predictive AUC for generative model (5.1) based on 20 repetitions (with 10% missing rate in the W matrix).

		W_{ij} (using node covariates)		W_{ij} (using network topology)	
(η^2, c)	α_{ij}	Partial-sum	Full-sum	Partial-sum	Full-sum
(1, −0.3)	0.8	0.741 (0.0045)	0.723 (0.0048)	0.503 (0.0052)	0.499 (0.0057)
	$P_{ij}^{0.05}$	0.761 (0.0046)	0.760 (0.0047)	0.499 (0.0061)	0.499 (0.0061)
(4, 1)	0.8	0.850 (0.0048)	0.852 (0.0045)	0.522 (0.0884)	0.520 (0.0901)
	$P_{ij}^{0.04}$	0.874 (0.0052)	0.868 (0.0056)	0.514 (0.0862)	0.511 (0.0853)
(9, 2)	0.8	0.910 (0.0031)	0.905 (0.0032)	0.620 (0.0194)	0.628 (0.0238)
	$P_{ij}^{0.03}$	0.915 (0.0034)	0.923 (0.0032)	0.824 (0.0152)	0.820 (0.0177)
(16, 3)	0.8	0.952 (0.0020)	0.954 (0.0026)	0.925 (0.0023)	0.919 (0.0025)
	$P_{ij}^{0.025}$	0.955 (0.0044)	0.960 (0.0041)	0.939 (0.0072)	0.935 (0.0076)

undirected versions. However, when network topology is used to define the similarity matrix, the performance for directed networks is better than for undirected ones, especially when η is relatively small; this is probably because the network topology-based similarity is more reliable for the directed networks than for the undirected ones. Comparing the two sampling rates, the performance under nonconstant sampling rate is slightly better. This is probably because with this nonconstant sampling rate, recorded links are more reliable than in the constant scenario (the higher the probability, the more likely a true edge is correctly recorded). Overall, we observe that with enough links the proposed method performs well, and using covariate-based similarity generally improves prediction.

Further, since in many applied contexts one would expect to see missing values with node similarities, we also test our proposed method when entries of the similarity matrix W are missing. Specifically, we randomly sample 10% entries of the W matrix and replace them by 0. The results are shown in Tables 3 and 4. Comparing with the results when W is complete, we see that the average performance of the proposed method is not significantly affected; however, the standard error of the AUC increases due to the missingness in W. In addition, we have also considered the case when 20% entries of W are missing at random (results not shown in the article). Again, we observed that the average AUC decreases a little, but not much, while the standard error of the AUC increases.

6. Applications

6.1. The Protein–Protein Interaction Network

Our first application is to an undirected network containing yeast protein–protein interactions from von Mering et al. (2002). This network was edited to contain only highly reliable interactions supported by multiple experiments (Bleakley, Biau, and Vert 2007), resulting in 984 protein nodes and 2438 edges, with the average node degree about 5. We take this verified network to be the true underlying network A. Bleakley, Biau, and Vert (2007) also constructed a matrix measuring similarities between proteins based on gene expression, protein localization, phylogenetic profiles, and yeast two-hybrid data, which we use as the node similarity matrix W for link prediction.

Here, we compare the full-sum criterion (3.4), the partial-sum criterion (3.5), and the latent variable model proposed and implemented in the R package `eigenmodel` by Hoff (2007). Specifically, the latent variable model assumes

$$P(A_{ij} = 1 \mid x_{ij}, u_i, u_j) = \Phi(\mu + \beta^\top x_{ij} + u_i^\top \Lambda u_j),$$

where Φ is the CDF of the standard normal distribution, Λ is a $k \times k$ diagonal matrix, and $u_i \in \mathbb{R}^k$ are unobserved latent variables, and the model is fitted using Markov chain Monte Carlo (MCMC). Note that the model is flexible and can be fitted either with or without node similarities. To test prediction, we generate indicators E_{ij}'s as independent Bernoulli variables taking value 1 with probability α_{ij}, and set $\widetilde{A}_{ij} = E_{ij}A_{ij}$. Similarly to the simulation study, we consider two scenarios for α_{ij}: (1) α_{ij} is a constant, and we set $\alpha_{ij} = 0.5$ and 0.8, corresponding to different amounts of available information, and (2) $\alpha_{ij} = \widehat{f}_{ij}^{\,\omega}$, where $\widehat{f}_{ij}$ were obtained by applying the proposed method to the complete data and ω was chosen such that the overall sampling rate is about 0.5 or 0.8.

We use the block coordinate descent algorithm proposed in Section 4 to approximately optimize (3.4) and (3.5), with $q = 10$. The tuning parameter λ is selected by five-fold cross-validation as in Section 5.

The latent variable model depends on a tuning parameter K, the dimension of the latent space. We fix $K = 5$ since larger values of K do not significantly change the performance in this example. We again use predictive AUC to evaluate the link prediction performance on the set $\{(i, j) : E_{ij} = 0\}$.

The results are reported in Table 5. The latent variable model performs better when using node similarities, but overall is outperformed by our method in most cases. Both models perform better under the nonconstant sampling rate, but the latent variable model not using W is more sensitive to the sampling rate

Table 5. Mean and standard error of predictive AUC for the protein–protein interaction data based on 20 repetitions.

α_{ij}	Partial-sum	Full-sum	Latent model (without W)	Latent model (using W)
0.8	0.875 (0.0085)	0.834 (0.0086)	0.831 (0.0121)	0.868 (0.0086)
$\widehat{f}_{ij}^{0.03}$	0.904 (0.0512)	0.937 (0.0357)	0.822 (0.0807)	0.930 (0.0709)
0.5	0.856 (0.0046)	0.823 (0.0054)	0.790 (0.0099)	0.880 (0.0069)
$\widehat{f}_{ij}^{0.08}$	0.899 (0.0326)	0.935 (0.0247)	0.825 (0.0480)	0.914 (0.0334)

Table 6. Mean and standard error of predictive AUC for the protein–protein interaction data based on 20 repetitions (with 10% missing rate in the W matrix).

α_{ij}	Partial-sum	Full-sum	Latent model (without W)	Latent model (using W)
0.8	0.863 (0.0092)	0.836 (0.0086)	0.831 (0.0121)	0.862 (0.0090)
$\widehat{f}_{ij}^{0.03}$	0.897 (0.0532)	0.932 (0.0370)	0.822 (0.083)	0.910 (0.0733)
0.5	0.844 (0.0061)	0.833 (0.0075)	0.790 (0.0099)	0.861 (0.0095)
$\widehat{f}_{ij}^{0.08}$	0.871 (0.0351)	0.926 (0.0235)	0.825 (0.0480)	0.924 (0.0345)

α_{ij}, since a low sampling rate may substantially distort the overall network topology. On the other hand, when the node similarity matrix W is used, the impact of the sampling rate is virtually gone. Similar as in simulation studies, we also test the proposed method when 10% entries of the W matrix are missing, and the results are shown in Table 6. Once again, we observe that the average performance of the proposed method is not significantly affected, while the standard error of the AUC increases.

6.2. The School Friendship Network

This dataset is a school friendship network from the National Longitudinal Study of Adolescent Health, previously studied by Hunter, Goodreau, and Handcock (2008). This network contains 1011 high school students and 5459 directed links connecting students to their friends, as reported by the students themselves. The average degree of this network is also around 5. In addition to the link information, the dataset also includes information on the student's grade, gender, and race. We construct two node similarity matrices, one using the Jaccard index (3.7) and the other based on node covariates,

$$W_{ij} = (1_{\{\text{grade}_i=\text{grade}_j\}} + 1_{\{\text{gender}_i=\text{gender}_j\}} + 1_{\{\text{race}_i=\text{race}_j\}})/3.$$

To test our two link prediction criteria, we use the same settings for E_{ij} as in the previous example, and apply block coordinate descent to minimize the criteria with λ chosen by cross-validation, and report the average predictive AUC over 20 realizations of E_{ij}'s. We also compare with a latent variable model for directed networks by Hoff (2009) (implemented in the R package `amen`). The model is flexible and can be fitted either with or without node similarities.

The results are reported in Tables 7 and 8. First, it can be seen that node covariates are not very helpful for link prediction in this case; this is probably because the node covariates here are discrete and the similarities they provide are too coarse. Further, note that when covariates are taken to define node similarities,

Table 7. Mean and standard error of predictive AUC for the school friendship data based on 20 repetitions.

	W_{ij} (using network topology)		W_{ij} (using node covariates)			
α_{ij}	Partial-sum	Full-sum	Partial-sum	Full-sum	Latent model (without W)	Latent model (using W)
0.8	0.922 (0.0033)	0.914 (0.0041)	0.551 (0.0826)	0.540 (0.0810)	0.854 (0.0117)	0.856 (0.0090)
$\widehat{f}_{ij}^{0.02}$	0.974 (0.0277)	0.976 (0.0255)	0.509 (0.0826)	0.507 (0.0810)	NA*	NA*
0.5	0.879 (0.0045)	0.861 (0.0045)	0.553 (0.0090)	0.545 (0.0056)	0.791 (0.0102)	0.808 (0.0076)
$\widehat{f}_{ij}^{0.065}$	0.971 (0.0176)	0.972 (0.0178)	0.539 (0.0585)	0.569 (0.0538)	NA*	NA*

NOTE: *The MCMC iterations were interrupted and "NAs produced" were returned.

Table 8. Mean and standard error of predictive AUC for the school friendship data based on 20 repetitions (with 10% missing rate in the W matrix).

	W_{ij} (using network topology)		W_{ij} (using node covariates)			
α_{ij}	Partial-sum	Full-sum	Partial-sum	Full-sum	Latent model (without W)	Latent model (using W)
0.8	0.910 (0.0045)	0.903 (0.0060)	0.522 (0.0826)	0.543 (0.0810)	0.854 (0.0117)	0.843 (0.0088)
$\widehat{f}_{ij}^{0.02}$	0.958 (0.0305)	0.969 (0.0281)	0.503 (0.0852)	0.489 (0.0861)	NA*	NA*
0.5	0.879 (0.0062)	0.861 (0.0060)	0.543 (0.0104)	0.540 (0.0110)	0.791 (0.0102)	0.812 (0.0095)
$\widehat{f}_{ij}^{0.065}$	0.973 (0.0226)	0.965 (0.0204)	0.539 (0.0631)	0.569 (0.0612)	NA*	NA*

NOTE: *The MCMC iterations were interrupted and "NAs produced" were returned.

a simple average of the three covariates is used, while ideally, the performance may be improved if a weighted average is used to construct node similarities. However, this approach would introduce weights as unknown parameters and it is beyond the scope of this article. When the network topology is used to construct node similarities, both the partial-sum and the full-sum criteria perform fairly well and outperform the latent variable model, which failed to converge in the nonconstant sampling rate case, whereas our method improves when the sampling rate is nonconstant.

7. Summary and Future Work

We have proposed a new framework for link prediction that allows uncertainty in observed links and nonlinks of a given network. Our method can provide relative rankings of potential links for pairs with and without observed links. The proposed link prediction criteria are fully nonparametric and essentially model-free, relying only on the assumption that similar node pairs have similar link probabilities, which is valid for a wide range of network models. One direction we would like to explore in the future is to combine more specific parametric network models with our nonparametric approach, with the goal of achieving both robustness and efficiency. We are also investigating consistency properties of our method, which is challenging because it requires developing a novel theoretical framework for evaluating consistency of rankings. Ultimately, we would also like to incorporate the general framework of link uncertainty into other network problems, for example, community detection.

Acknowledgments

Yunpeng Zhao gratefully acknowledges the support from NSF DMS 1513004. Elizaveta Levina gratefully acknowledges the support from NSF DMS 01106772, NSF DMS 1159005, and NSF DMS 1521551. Ji Zhu gratefully acknowledges the support from NSF DMS 1407698, KLAS 130026507, and KLAS 130028612.

References

Adamic, L. A., and Adar, E. (2003), "Friends and Neighbors on the Web," *Social Networks*, 25, 211–230. [1]

Balachandran, P., Airoldi, E., and Kolaczyk, E. (2013), "Inference of Network Summary Statistics Through Network Denoising," *Annals of Statistics*, arXiv:1310.0423. [2]

Balachandran, P., Kolaczyk, E. D., and Viles, W. (2014), "On the Propagation of Low-Rate Measurement Error to SubgraphCounts in Large," *Sparse Networks*, arXiv:1409.5640. [2]

Ben-Hur, A., and Noble, W. S. (2005), "Kernel Methods for Predicting Protein-Protein Interactions," *Bioinformatics*, 21, i38–i46. [1]

——— (2006), "Choosing Negative Examples for the Prediction ofProtein-Protein Interactions," *BMC Bioinformatics*, 7, (Suppl 1):S2. [2]

Bleakley, K., Biau, G., and Vert, J.-P. (2007), "Supervised Reconstruction of Biological Networks With Local Models," *Bioinformatics*, 23, i57–i65. [1,7]

Boyd, S., and Vandenberghe, L. (2004), *Convex Optimization*, New York: Cambridge University Press. [5]

Clauset, A., Moore, C., and Newman, M. E. J. (2008), "Hierarchical Structure and the Prediction of Missing Links in Networks," *Nature*, 453, 98–101. [2]

Getoor, L., and Diehl, C. P. (2005), "Link Mining: A Survey," *ACM SIGKDD Explorations Newsletter*, 7, 3–12. [1]

Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. (2006), "Link Prediction Using Supervised Learning," in *Workshop on Link Analysis, Counter-terrorism and Security (at SIAM Data Mining Conference)*. [1]

Hasan, M. A., and Zaki, M. J. (2011), "A Survey of Link Prediction in Social Networks," in *Social Network Data Analytics*, ed. C. C. Aggarwal, New York: Springer, pp. 243–275. [1]

Hildreth, C. (1957), "A Quadratic Programming Procedure," *Naval ResearchLogistics Quarterly*, 4, 79–85. [5]

Hoff, P. D. (2007), "Modeling Homophily and Stochastic Equivalence in Symmetric Relational Data," in *Advances in Neural InformationProcessing Systems* (vol. 19), Cambridge, MA: MIT Press, pp. 657–664. [2,7]

——— (2009), "Multiplicative Latent Factor Models for Description and Prediction of Social Networks," *Computational & Mathematical Organization Theory*, 15, 261–272. [8]

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), "Latent Space Approaches to Social Network Analysis," *Journal of the American Statistical Association*, 97, 1090–1098. [2,6]

Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008), "Goodness of Fit of Social Network Models," *Journal of the American Statistical Association*, 103, 248–258. [8]

Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., and Tsuda, K. (2009), "Link Propagation: A Fast Semi-Supervised Learning Algorithm for Link Prediction," in *Proceedings of the 2009SIAM International Conference on Data Mining*, 1100–1111. [1,3]

Katz, L. (1953), "A New Status Index Derived From Sociometric Analysis," *Psychometrika*, 18, 39–43. [1]

Leicht, E. A., Holme, P., and Newman, M. E. J. (2006), "Vertex Similarity in Networks," *Physical Review E*, 73, 026120. [1]

Liben-Nowell, D., and Kleinberg, J. (2007), "The Link-Prediction Problem for Social Networks," *Physica A: Statistical Mechanics and its Applications*, 390, 1150–1170. [1,4]

Lu, L., and Zhou, T. (2010), "Link Prediction in Complex Networks: A Survey," arXiv:1010.0725v1. [1]

Luo, Z.-Q., and Tseng, P. (1992), "On the Convergence of the Coordinate Descent Method for Convex Differentiable Minimization," *Journal of Optimization Theory and Applications*, 72, 7–35. [5]

Menon, A., and Elkan, C. (2011), "Link Prediction via Matrix Factorization," in *Machine Learning and Knowledge Discovery in Databases*, (*Lecture Notes in Computer Science*, Vol. 6912), eds. D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, Berlin Heidelberg: Springer, pp. 437–452. [1]

Miller, K., Griffiths, T., and Jordan, M. (2009), "Nonparametric Latent Feature Models for Link Prediction," in *Advances in Neural Information Processing Systems (NIPS)* (Vol. 22), eds. Y. Bengio, D. Schuurmans, J. Lafferty, and C. Williams, Red Hook, NY: Curran Associates, pp. 1276–1284. [2]

Priebe, C. E., Sussman, D. L., Tang, M., and Vogelstein, J. T. (2015), "Statistical Inference on Errorfully Observed Graphs," *Journal of Computational and Graphical Statistics*, 24, 930–953. [2]

Raymond, R., and Kashima, H. (2010), "Fast and Scalable Algorithms for Semi-Supervised Link Prediction on Static and Dynamic Graphs," in *Machine Learning and Knowledge Discovery in Databases* (*Lecture Notes in Computer Science*, Vol. 6323), eds. J. Balczar, F. Bonchi, A. Gionis, and M. Sebag, Berlin Heidelberg: Springer, pp. 131–147. [1]

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., and Bork, P. (2002), "Comparative Assessment of Large-Scale Data Sets of Protein-Protein Interactions," *Nature*, 417, 399–403. [2,7]

Warga, J. (1963), "Minimizing Certain Convex Functions," *Journal of the Society for Industrial and Applied Mathematics*, 11, 588–593. [5]

Yang, Y., Chawla, N., Sun, Y., and Hani, J. (2012), "Predicting Links in Multi-Relational and Heterogeneous Networks," in *2012 IEEE 12th International Conference on Data Mining (ICDM)*, pp. 755–764. [2]

Yu, K., Chu, W., Yu, S., Tresp, V., and Xu, Z. (2007), "Stochastic Relational Models for Discriminative Link Prediction," in *Proceedings of Neural Information Processing Systems*, Cambridge MA: MIT Press, pp. 1553–1560. [2]