Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Learning Latent Networks via Matrix Decomposition by Solving Lyapunov Equations With Applications in Gene Regulatory Networks

Yunpeng Zhao [1,*] and Shuo Chen [2]

[1]Department of Statistics, George Mason University, Fairfax, 22030, USA and
[2]Department of Epidemiology and Biostatistics, University of Maryland, College Park, MD 20742, USA.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: One common difficulty in the study of biological networks is the lack of certainty about the presence or absence of edges between nodes due to technical limitations in biological experiments. As a fundamental question, learning the true latent networks, or link prediction has attracted increasing attention in bioinformatics. We propose a new criterionbased method utilizing observed network topology as well as information on nodes. The key ingredient of this work is the decomposition of the latent probability matrix as a product of two low-rank matrices, which leads to a very efficient algorithm based on solving Lyapunov equations.

Results: The algorithm is computationally efficient and performs well under different simulation setups for recovery of latent networks. And the proposed method has been applied to a gene regulatory network of E. coli.

Contact: yzhao15@gmu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A network is a data structure composed of objects (called *vertices* or *nodes*) as well as relations (called *vertices* or *nodes*) between them. For example, a gene regulatory network contains a collection of molecular regulators and interactions between these regulators in gene expression. Biological networks, for instance, protein-protein interaction networks, metabolic networks, neural networks, among others, are fundamental to understand underlying biological mechanisms. In the past decades, modeling and analyzing network structure have attracted increasing attention in many research fields, including biology (especially bioinformatics), computer science, social sciences, statistics, among others (see Newman, 2010; Getoor and Diehl, 2005; Goldenberg *et al.*, 2010 for general reviews).

A common difficulty in the study of biological networks is the lack of certainty about the presence or absence of edges between nodes. That is, some observed edges and non-edge status can be false positives or false negatives, respectively. Lack of negative examples is typical in biological networks due to technical limitations in biological experiments Lu and Zhou, 2010. For instance, in a protein–protein interaction network, a pair of proteins with no observed edge may not imply that there is no interaction between the two proteins. Instead, it may indicate that this interaction has not been detected by experiment, that the current techniques did not have enough sensitivity to detect the interaction. Positive examples could sometimes also be uncertain. For example, a large number of false positive interactions may be generated by high-throughput experiments (von Mering *et al.*, 2002).

Therefore, learning the true latent networks, or so-called link prediction has become a fundamental question in network science, particularly in the study of biological networks. The goal of link prediction is to estimate the likelihood of the existence of a link between two nodes, based on the observed network topology as well as additional attributes of nodes, called node covariates, if applicable (see Getoor and Diehl, 2005, Lu and Zhou, 2010 and Liben-Nowell and Kleinberg, 2007 for reviews from different research fields).

Rigorously speaking, there are two different settings for link prediction. In the first setting, the network is assumed to be dynamic, and a

snapshot of the network at time t, or a sequence of snapshots at time $1, ..., t$, is observed. And the task is to predict new links that are likely to emerge in the near future, like at time $t + 1$. In the second setting, the network is static but is only partially observed, or contains observation errors, and the task is to recover the entire latent network. These two tasks are related in practice since a missing link may appear in the future. For example, a missing interaction due to technical limitations in a protein-protein interaction network may become detectable by more accurate experiments in the future. Or two real-life friends connected on Facebook so far but may build a link later. However, these two settings are quite different from the analysis of point of view. We focus on the second setting, i.e., static networks in this article and do not consider dynamics networks.

Further, we consider two subtypes of problems. The first subtype is what we introduced in the beginning – the observed network contains false positive/negative examples and the task is to recover the true latent network. To the best of our knowledge, this problem was first explicitly studied in Zhao *et al.*, 2017. Their method does not estimate the numerical values of the link probabilities, which in fact cannot be estimated since the false positive/negative rate is unknown. Instead, it provides relative rankings of probabilities of the existence of links between nodes. The performance of these rankings can be evaluated by the ROC curve. The second subtype is a matrix completion problem, i.e., the observed adjacency matrix contains missing components and the task is to fill in these blanks. Our method are motivated from the first subtype. And the proposed algorithm and method can also be applied to the second subtype with some modification. See Section 3.2 for details.

Existing link prediction methods in the literature can be loosely classified into unsupervised and supervised approaches. Unsupervised approaches are based on various types of node similarity measures. In this type of approaches, each pair of nodes is assigned a similarity score, and pairs with higher similarity scores are assumed to be linked with higher probabilities. Typical choices of such 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 whole network, such as the Katz index (Katz, 1953) and the Leicht-Holme-Newman Index (Leicht *et al.*, 2006). See Liben-Nowell and Kleinberg, 2007; Lu and Zhou, 2010 for comprehensive reviews of unsupervised approaches.

In supervised approaches, link prediction is reformulated as a binary classification problem, where the responses ($\{1, 0\}$) indicate whether there exists a link for a pair, and the predictors are covariates on each pair, constructed from node attributes. A number of popular supervised learning methods have been used in the link prediction problem, including the support vector machine (Ben-Hur and Noble, 2005; Bleakley *et al.*, 2007), semi-supervised learning (Kashima *et al.*, 2009; Raymond and Kashima, 2010), optimizing the area under the curve (AUC) (Menon and Elkan, 2011). Hasan *et al.*, 2006 evaluated the performance of several supervised learning methods. Further, methods based on probabilistic models for incomplete network can also be categorized into supervised approaches. Popular examples of model-based supervised approaches include latent space models (Hoff *et al.*, 2002), latent variable models (Hoff, 2007; Miller *et al.*, 2010), stochastic relational models (Yu *et al.*, 2007), and hierarchical structure models (Clauset *et al.*, 2008).

Most of these supervised approaches are designed for the second subtype problem introduced earlier – filling in missing values in an adjacency matrix. Zhao *et al.*, 2017 considered the first subtype, allowing for uncertainty of positive and negative examples. Zhao *et al.*, 2017 treated link probabilities as unknown parameters and made no structural assumption on networks. Thus this method needs to estimate an order of n^2 parameters where n is the size of the network, which may result in high computational cost for large networks.

To address this challenge, we propose a new criterion function by assuming that the latent probability matrix can be decomposed as a product of two low-rank matrices. When the network is directed, the rows of the two matrices represent the latent "features" of each node as a link sender or a link receiver, respectively. When the network is undirected, these two matrices are set to be identical so that there is no distinction between senders and receivers. If the attributes of two nodes are similar, it is natural to make the assumption that these nodes also behave similar as senders/receivers when building connections. It is worth noticing that this assumption is not the same as the assumption in unsupervised approaches: two nodes with high similarity are not necessarily more likely to be connected as in unsupervised approaches. Instead, they play a similar role in building connections. When a similarity matrix of nodes is available, we include a penalty term into the criterion function in order to penalize the difference between latent features of two nodes when the similarities between these nodes are high. This idea is a generalization of the classical graph-based semi-supervised learning for regression or classification (one-dimensional case) to the link prediction in networks (two-dimensional case). See Chapelle *et al.*, 2006 for a comprehensive review of graph-based learning and other semi-supervised learning methods.

We develop an efficient algorithm by iteratively updating the two matrices. The key ingredient of this algorithm is that each update is equivalent to solving two Lyapunov equations, which can be computed efficiently by eigen-decomposition. With some modification, our method can also be applied to the second subtype problem. The modified criterion function is solved by a similar algorithm with an imputation technique.

The rest of this article is organized as follows. In Section 2, we introduce the model assumptions, and propose novel link prediction criteria via low-rank matrices decomposition. In Section 3, we develop the algorithm for optimizing those criteria based on solving Lyapunov equations iteratively. In Section 4, we evaluate the performance of proposed criteria on simulated networks under various settings. In Section 5, we apply our method to predict links a regulatory network of E. coli. Section 6 concludes with a summary and discussion of future works.

2 Methods

We begin with basic notation. A network with n nodes can be represented by an $n \times n$ adjacency matrix $Y = [Y_{ij}]$, where

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

Here we use Y to denote the adjacency matrix instead of the more standard notation A, because we consider the links of the network as responses, and the characteristics of nodes as explanatory variables. In our work, node covariates will not explicitly appear in the criterion function. Instead, a similarity matrix constructed by node covariates will be used. Further, our method can be applied into both directed networks and undirected networks. In fact, the case of directed networks is more straightforward. We focus on the link prediction problem for directed networks and then show that our method can be easily adapted to the undirected case. Therefore, in general Y can be either symmetric (for undirected networks) or asymmetric (for directed networks).

2.1 Model assumptions

We briefly reiterate the model assumption in Zhao *et al.*, 2017 and will follow the same assumption in this paper. Assume that there is an adjacency matrix of the true latent network, denoted by Y^{True}, and this network is observed with errors, denoted by Y. Each Y_{ij}^{True} follows a Bernoulli

distribution with success rate $\mathbb{P}(Y_{ij}^{True} = 1) = P_{ij}$. And the observed network is generated by

$$\mathbb{P}(Y_{ij} = 1|Y_{ij}^{True} = 1) = \alpha, \quad \mathbb{P}(Y_{ij} = 0|Y_{ij}^{True} = 0) = \beta,$$

where α and β are the probabilities of correctly recording a true edge and an absent edge, respectively. Here we assume α and β to be constant.

Note that $\beta = 1$ and $\alpha = 1$ are two important special cases. In the former case, all observed links are true positives. For example, in a protein-protein interaction network where all observed links have been verified by experiments, we only need to estimate the likelihood of links for node pairs without observed links. Similarly, in the latter case, all unobserved links are true negatives and the task is to investigate the reliability of observed links.

Under the model setting, we have

$$\tilde{P}_{ij} \triangleq \mathbb{P}(Y_{ij} = 1) = (\alpha + \beta - 1)P_{ij} + (1 - \beta).$$

And by Bayes' rule,

$$\mathbb{P}(Y_{ij}^{True} = 1|Y_{ij} = 1) = \frac{\alpha P_{ij}}{\tilde{P}_{ij}}, \tag{1}$$

$$\mathbb{P}(Y_{ij}^{True} = 1|Y_{ij} = 0) = \frac{(1 - \alpha)P_{ij}}{1 - \tilde{P}_{ij}}. \tag{2}$$

It is worth noticing that both (1) and (2) are monotone increasing functions of $\tilde{P}_{ij}$ if $\alpha + \beta > 1$. Zhao *et al.*, 2017 then made a crucial observation that it is sufficient to estimate the link probabilities for the observed network – $\tilde{P}_{ij}$ to provide rankings of potential links. We follow this idea and focus on the estimation of $\tilde{P}_{ij}$.

2.2 Criteria for link prediction via matrix decomposition

For the purpose of link prediction, in addition to the observed network Y, we assume that there is a symmetric matrix $W = [W_{ij}]$ with $0 \leq W_{ij} \leq 1$ available, which describes the similarity between nodes i and j. In this article, we only consider the similarity matrix W as external information on nodes, that is, W is generated from external node covariates, not from the topology of the network. In general, W can also be obtained from network topology, or a combination of both sources (see Liben2007 for some popular choices of topology based similarity measures).

In this subsection, we first propose the criterion for estimating $\tilde{P}_{ij}$ in directed networks:

$$\begin{aligned}\arg\min_{U,V} \ Q_1 = &\sum_{ij}^{n}(Y_{ij} - F_{ij})^2 \\ &+ \lambda \sum_{i<j} W_{ij}\|U_{i\cdot} - U_{j\cdot}\|^2 + \lambda \sum_{i<j} W_{ij}\|V_{i\cdot} - V_{j\cdot}\|^2,\end{aligned} \tag{3}$$

where $U, V \in \mathbb{R}^{n\times K}$, $F = UV^T$ and λ is a tuning parameter. For any matrix M, we use $M_{i\cdot}$ to denote the i-th row of M and $M_{\cdot j}$ to denote the j-th column of M.

The criterion function above is an improvement of the method in Zhao *et al.*, 2017. For comparison, we give the criterion function for directed networks in that article,

$$\begin{aligned}\arg\min_{F} \ &\sum_{ij}^{n}(Y_{ij} - F_{ij})^2 \\ &+ \lambda \sum_{ii'jj'}^{n} W_{ii'}W_{jj'}(F_{ij} - F_{i'j'})^2.\end{aligned} \tag{4}$$

The key difference between the two criteria is that F in (4) is a matrix containing n^2 free parameters. Therefore, the computation costs of (4) may become very high for large networks. By contrast, criterion (3) imposes a structure of the latent probability matrix F, that is, F can be decomposed into two low-rank matrices, U and V. If $K \ll n$, the model complexity can be significantly reduced. Matrices U and V have the following natural interpretation. $U_{i\cdot}$ can be considered as the feature of node i as a link sender, and $V_{j\cdot}$ can be consider as the feature of node j as a link receiver. And the probability of link existence between node i and j is high if the features of i and j match.

The first term of both (3) and (4) is the widely used squared error loss connecting the link probabilities with the observed network. The motivation of this loss function is that the minimizer of its population version, i.e., $\mathbb{E}(Y_{ij} - F_{ij})^2$ is $\tilde{P}_{ij}$. One can also choose other loss functions such as the hinge loss or the negative log-likelihood. The main reason for choosing the squared error loss here is computational efficiency, since it makes the first term of (3) a quadratic form.

The penalty term in (3) is based on the following assumption: if node i and j are similar according to the similarity matrix W, then node i and j will have similar behavior as link senders, i.e., $U_{i\cdot}$ is close to $U_{j\cdot}$, and also have similar behavior as link receivers, i.e., $V_{i\cdot}$ is close to $V_{j\cdot}$. Since $F_{ij} = U_{i\cdot}V'_{j\cdot}$, this assumption is equivalent to saying if both pairs of endpoints are similar, the link probabilities on these two pairs should also be close, which is the key assumption made in (4).

It is worth noticing that this "pair similarity" assumption is different from the "node similarity" assumption used by many unsupervised link prediction approaches and is more general. The node similarity assumption is that a link is more likely to exist if two nodes are similar. By assuming pair similarity, two nodes can possibly have high link probabilities even they are not similar according to W. We believe that the pair similarity assumption is in particular more valid for biological networks. For instance, predators are not similar to their preys in a food web. But a lion and a tiger may have similar preys since these two animals have similar characteristics and both are at the top of the food chain.

Note that (4) is also based on the pair similarity assumption, in a more explicit way in fact. The difference is that since F is decomposable in (3), we can penalize the differences of two endpoints $U_{i\cdot}$ and $V_{j\cdot}$ separately, avoiding the complicated summation term in (4).

Another advantage of (3) is that we can safely remove the penalty unless the similarity matrix W does provide extra useful information. Unlike (4), (3) is still a valid criterion function for link prediction even without the penalty term, i.e., setting $\lambda = 0$. That is also part of the reason that we only use W based on external information on nodes in this article, since W created by network topology is not very reliable if the missing rate of links is high. Therefore, we would suggest not include the penalty term in (3) if no useful node covariates are available.

By modifying (5), we can also propose a criterion for the second subtype problem in link prediction – fill in the missing components of a partially observed adjacency matrix Y. Let $S = [S_{ij}]$ be an $n \times n$ matrix, where $S_{ij} = 1$ if Y_{ij} is observed, and $S_{ij} = 0$ otherwise. In this case, we propose the following criterion:

$$\begin{aligned}\arg\min_{U,V} \ Q_2 = &\sum_{ij}^{n} S_{ij}(Y_{ij} - F_{ij})^2 \\ &+ \lambda \sum_{i<j} W_{ij}\|U_{i\cdot} - U_{j\cdot}\|^2 + \lambda \sum_{i<j} W_{ij}\|V_{i\cdot} - V_{j\cdot}\|^2,\end{aligned} \tag{5}$$

Since (5) only involves a partial sum of the loss function terms, we will refer to (5) as the partial-sum criterion and (3) as the full-sum criterion for the rest of the article, as in Zhao *et al.* (2017).

The partial sum criterion is also closely related to the first subtype problem. Suppose that some examples are certainly true positives or true negatives, that is, some Y_{ij} may be known to be true 1's and true 0's,

while others may be uncertain. If there exist sufficient amount of true positives and true negatives, it makes sense to only include such certain information in the criterion. Therefore, when using the partial sum criterion, we require that the observed Y_{ij} contain no error. That is, $S_{ij} = 1$ only if it is known that $Y_{ij} = Y_{ij}^{True}$, and 0 otherwise. If this requirement is not satisfied, then the second subtype problem can be easily put into the framework of the first subtype by setting $Y_{ij} = 0$ for all (i, j) with $S_{ij} = 0$. Therefore, due to the availability of extra information, the partial-sum criterion (5) presumably performs better than the full-sum criterion (3). On the other hand, the terms S_{ij} bring extra difficulty to the algorithm design and we will explain the details in Section 3.

So far we focus on the link prediction criteria for directed networks. The link prediction criteria for undirected networks can be easily obtained if we make F symmetric, i.e., $U = V$ in (3) and (5). Then the full-sum criterion for undirected networks is

$$\arg\min_{U} \; Q_3 = \sum_{ij}^{n} (Y_{ij} - F_{ij})^2 + 2\lambda \sum_{i<j} W_{ij} \|U_{i\cdot} - U_{j\cdot}\|^2, \quad (6)$$

where $U \in \mathbb{R}^{n\times K}$ and $F = UU^T$. Similarly, the partial-sum criterion for undirected network is

$$\arg\min_{U} \; Q_4 = \sum_{ij}^{n} S_{ij}(Y_{ij} - F_{ij})^2 + 2\lambda \sum_{i<j} W_{ij} \|U_{i\cdot} - U_{j\cdot}\|^2, \quad (7)$$

where $S_{ij} = 1$ if it is known that $Y_{ij} = Y_{ij}^{True}$, and 0 otherwise.

3 Algorithm

3.1 Optimizing full-sum criteria by solving Lyapunov equations

We begin with the full-sum criterion for directed networks. By some simple matrix algebra, Q_1 can be written as the following matrix form,

$$\begin{aligned} Q_1 =& \text{Tr}(VU^TUV^T - 2Y^TF + Y^TY) + \lambda \text{Tr}(U^TDU - U^TWU) \\ &+ \lambda \text{Tr}(V^TDV - V^TWV), \end{aligned}$$

where D is a diagonal matrix with $d_{ii} = \sum_j W_{ij}$. By taking the partial derivatives with respect to U and V, one can show that the optimal solution $(\hat{U}, \hat{V})$ of the full-sum criterion (3) must satisfy

$$\hat{U}\hat{V}^T\hat{V} - Y\hat{V} + \lambda(D - W)\hat{U} = 0, \quad (8)$$

$$\hat{V}\hat{U}^T\hat{U} - Y^T\hat{U} + \lambda(D - W)\hat{V} = 0. \quad (9)$$

Naturally, we optimize the objective function by solving (8) and (9) iteratively. Specifically, we update $\hat{U}$ in (8) with $\hat{V}$ fixed and update $\hat{V}$ in (9) with $\hat{U}$ fixed. Both (8) and (9) can be written as a general form called Lyapunov equation (Bartels and Steward, 1972) as follows,

$$LZ + ZR = C, \quad (10)$$

where L, R and C and known matrices and Z is unknown. For example, for (8), $L = \lambda(D - W)$, $R = \hat{V}^T\hat{V}$ and $C = Y\hat{V}$.

First, note that (10) is essentially a system of linear equations of Z. By vectorizing both sides of (10), we obtain

$$(I_K \otimes L + R^T \otimes I_n)\text{vec}(Z) = \text{vec}(C), \quad (11)$$

where $\text{vec}(\cdot)$ denotes the transformation of a matrix column by column to a column vector, and $\otimes$ is the Kronecker product. The solution of (10) is the matrix form of the solution of this system of linear equations.

However, directly solving (11) may be time-consuming when n is large, especially as one iteration in the whole procedure. A commonly-used approach to solve Lyapunov equation is based on eigen-decomposition, which we describe in details as follows.

Since both L and R are symmetric matrices, they can be diagonalized. Let $L = P\Lambda_1P^T$ and $R = Q\Lambda_2Q^T$, where Λ_1 and Λ_2 are the diagonal matrices. Then (10) becomes

$$P\Lambda_1P^TZ + ZQ\Lambda_2Q^T = C.$$

By multiplying P^T from left and Q from right on both sides of the above equation, we obtain

$$\Lambda_1P^TZQ + P^TZQ\Lambda_2 = P^TCQ.$$

Let $\tilde{Z} = P^TZQ$. Then $\tilde{Z}$ can be solved elementwisely by the formula $\tilde{Z}_{ij} = [P^TCQ]_{ij}/(\Lambda_{1ii} + \Lambda_{2jj})$, since Λ_1 and Λ_2 are diagonal. Finally we can easily get Z back by $Z = P\tilde{Z}Q^T$.

In general, the method of eigen-decomposition is much more faster than solving a system of linear equations directly. More interestingly and importantly, this method can be further accelerated in our scenario according to the following observation. The most time-consuming part of this procedure is eigen-decomposition of L and R. In our problem, $\hat{U}^T\hat{U}$, $\hat{V}^T\hat{V}$ and $D - W$ require eigen-decomposition. But $D - W$ remains unchanged for all the iterations, and thus requires eigen-decomposition only once. On the other hand, both $\hat{U}^T\hat{U}$ and $\hat{V}^T\hat{V}$ are $K \times K$ matrix. Typically, K is chosen to be a small number compared with n, in which case eigen-decomposition of these matrix is trivial.

The full-sum criterion for undirected networks (6) can be solved by a similar approach. Note that (6) is equivalent to the following constrained optimization problem,

$$\begin{aligned} \arg\min_{U,V} \; & \sum_{ij}^{n} (Y_{ij} - F_{ij})^2 \\ & + \lambda \sum_{i<j} W_{ij} \|U_{i\cdot} - U_{j\cdot}\|^2 + \lambda \sum_{i<j} W_{ij} \|V_{i\cdot} - V_{j\cdot}\|^2, \end{aligned} \quad (12)$$

subject to $U = V$.

Empirically, we found that if we start from an initial value satisfying the symmetry constraint, then the solutions (8) and (9) for each iteration also satisfy that constraint. Therefore, (6) can be optimized by directly applying the method above with an initial value $U^{(0)} = V^{(0)}$.

3.2 Optimizing partial-sum criteria by imputation

With terms S_{ij}, the derivatives of the partial-sum criterion (5) with respect to U and V cannot be written as a form of (8) and (9), and thus cannot be optimized directly by solving Lyapunov equations. However, (5) can be solved by optimizing a sequence of full-sum criteria (3), if we apply the idea of imputation.

Clearly, the partial-sum criterion (5) is equivalent to the following optimization problem with augmented variables $\hat{Y}$,

$$\begin{aligned} \arg\min_{F,\hat{Y}} \; & \sum_{ij}^{n} S_{ij} \sum_{ij}^{n} (Y_{ij} - F_{ij})^2 + \sum_{ij}^{n} (1 - S_{ij})(\hat{Y}_{ij} - F_{ij})^2 \\ & + \lambda \sum_{i<j} W_{ij} \|U_{i\cdot} - U_{j\cdot}\|^2 + \lambda \sum_{i<j} W_{ij} \|V_{i\cdot} - V_{j\cdot}\|^2, \end{aligned} \quad (13)$$

subject to $\hat{Y}_{ij} = Y_{ij}$, for $S_{ij} = 1$.

We can update F and $\hat{Y}$ iteratively. With $\hat{Y}$ being fixed, (13) now becomes a standard full-sum criterion, and thus can be solved by the method

introduced in the previous subsection. With F fixed, $\hat{Y}$ can be obtained by imputing the values of F_{ij} for all $S_{ij} = 0$. In summary, the augmented optimization problem (13) can be solved by the following steps iteratively.

- Optimization: Optimizing a full-sum criterion with $\hat{Y}$ as the observed network.
- Imputation:

$$\hat{Y}_{ij} = \begin{cases} Y_{ij} & \text{if } S_{ij} = 1, \\ F_{ij} & \text{if } S_{ij} = 0. \end{cases}$$

4 Simulation studies

In this section, we evaluate the performance of the full-sum and partial-sum criteria on several simulated scenarios. In all simulation studies, the size of each network is fixed with $n = 500$. And node i contains covariates X_i which are independently generated from a multivariate normal distribution $N_p(0, I_p)$ with $p = 5$. Each Y_{ij}^{True} is generated independently, with logit $P_{ij} = f(X_i, X_j)$. We consider the following four functions $f(X_i, X_j)$ in simulation studies.

$$(a) \quad \sum_k (X_{ik} - X_{jk}), \qquad (a') \quad \sum_k (X_{ik} - X_{jk}) - 8,$$

$$(b) \quad 2X_i^T X_j / \|X_j\|, \qquad (b') \quad 2X_i^T X_j / \|X_j\| - 4,$$

Among them (a) is a linear function; (b) is the well-known projection model proposed in Hoff *et al.* (2002), under which the logit of link probability Y_{ij} is the projection of X_i onto the direction of X_j. (a') and (b') are obtained by subtracting a constant within the logit link in (a) and (b), respectively. (a') and (b') will generate sparser networks, which will allows us to evaluate the performance of our methods for both dense and sparse networks.

Next we generate indicators S_{ij}'s as independent Bernoulli variables with success probability 0.5, and define $Y_{ij} = S_{ij} Y_{ij}^{True}$. In this setting, all the observed edges are true positives but the missing edges may or may not be true negatives. The partial-sum criterion only uses correct information, i.e., Y_{ij}'s on the pairs with $S_{ij} = 1$. By contrast, the full-sum criterion use all the Y_{ij}'s as responses, so it will include half of absent edges as false negatives.

We define the similarity matrix W by the Gaussian Kernel,

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

where we choose $\sigma = \frac{1}{4}\text{median}\{\|X_i - X_j\|, i = 1, ..., n, j = 1, ..., n\}$. We optimize the full-sum and partial-sum criteria with $K = 5$ and λ chosen by 5-fold cross-validation.

The performance of link prediction is evaluated by ROC curves. Now we give the details of the evaluation procedure, because our setting is slightly different from the classical supervised learning, especially for the full-sum criterion. In each simulation, we evaluate the performance of both criteria on the same test set $\{(i, j) : S_{ij} = 0\}$ for fair comparison. The ROC curves are determined by the rankings of $\hat{F}_{ij}$ estimated by the full-sum or the partial-sum criterion on the test set. Specifically, let R_{ij} be the ranking of $\hat{f}_{ij}$ on the test set in descending order. For any integer k, false positives are pairs (i, j) ranked within top k but without links in the true network ($Y_{ij}^{True} = 0$), and true positives are pairs ranked within top k with $Y_{ij}^{True} = 1$. Then the true positive rate (TPR) and the false positive rate (FPR) are defined by

$$\text{TPR}(k) = \frac{|\{(i, j) : S_{ij} = 0, R_{ij} \le k, Y_{ij}^{True} = 1\}|}{|\{(i, j) : S_{ij} = 0, Y_{ij}^{True} = 1\}|},$$

$$\text{FPR}(k) = \frac{|\{(i, j) : S_{ij} = 0, R_{ij} \le k, Y_{ij}^{True} = 0\}|}{|\{(i, j) : S_{ij} = 0, Y_{ij}^{True} = 0\}|}.$$

Figure 1 shows the performance of the full-sum criterion and partial-sum criterion for the four simulation settings by ROC curves and the corresponding AUCs (Area under a curve). Each curve and the corresponding AUC are the average of 50 replicates. We also provide the ROC curves constructed from true P_{ij}'s as a benchmark for comparison.

Both the full-sum and the partial-sum criteria perform well in all simulation settings. As expected, the partial-sum criterion always provides better results since it assumes more information and only uses correct true positives and true negatives. But the performance of these two criteria is quite comparable; sometimes, the gaps between ROC curves cannot be clearly seen from the figures and can only be identified by the numerical values of AUCs. The criteria performs better for linear models than latent space models, which is also expected because of the simplicity of linear models.

5 Applications to a gene regulatory network of E. coli

In this section, we analyze a gene regulation dataset of E. coli, compiled by Faith *et al.*, 2007. This dataset includes of a gene regulatory network and expression data. The regulatory network consists of 3293 experimentally confirmed links between 1209 genes, which makes the network very sparse (average degree ≈ 2.72). Figure 2 shows this network. The expression data consist of 445 E. coli Affymetrix Antisense2 microarray expression profiles for those genes. The similarity matrix W is created from the gene expression data by the same Gaussian kernel as in simulation studies.

To evaluate the performance of our prediction methods, we generate indicators S_{ij}'s as independent Bernoulli variables with success probability α, and set $Y_{ij} = S_{ij} Y_{ij}^{True}$. We test our criteria on two different values of α, $\alpha = 0.2, 0.5$, corresponding to different proportions of the available true positives. We fix $K = 5$ and select λ by cross-validation. ROC curves are plotted to evaluate the performance of the full-sum and partial-sum criterion on pairs with $S_{ij} = 0$.

From Figure 3, both criteria perform quite well even for a small sample rate $\alpha = 0.2$. The distinction between the full-sum and partial-sum criteria become more apparent compared with the simulation studies, since this gene regulatory network is a challenging dataset for the purpose of link prediction due to its sparsity. Moreover, the similarity matrix W plays an important role in link prediction for this example. In fact, setting $\lambda = 0$ will result in low accuracy of prediction for both full-sum and partial-sum criterion, that is, the AUCs for both criteria drop below 0.7 without W. This suggests that the expression data are not only highly relevant to link existence in the regulatory network, but also provide auxiliary information beyond the matrix decomposition based on observed topology.

6 Summary and future work

In this article, we propose a novel link prediction approach for both directed and undirected networks based on matrix decomposition. This method naturally incorporates structural information of the network as well as node covariates. By assuming that the latent probability matrix

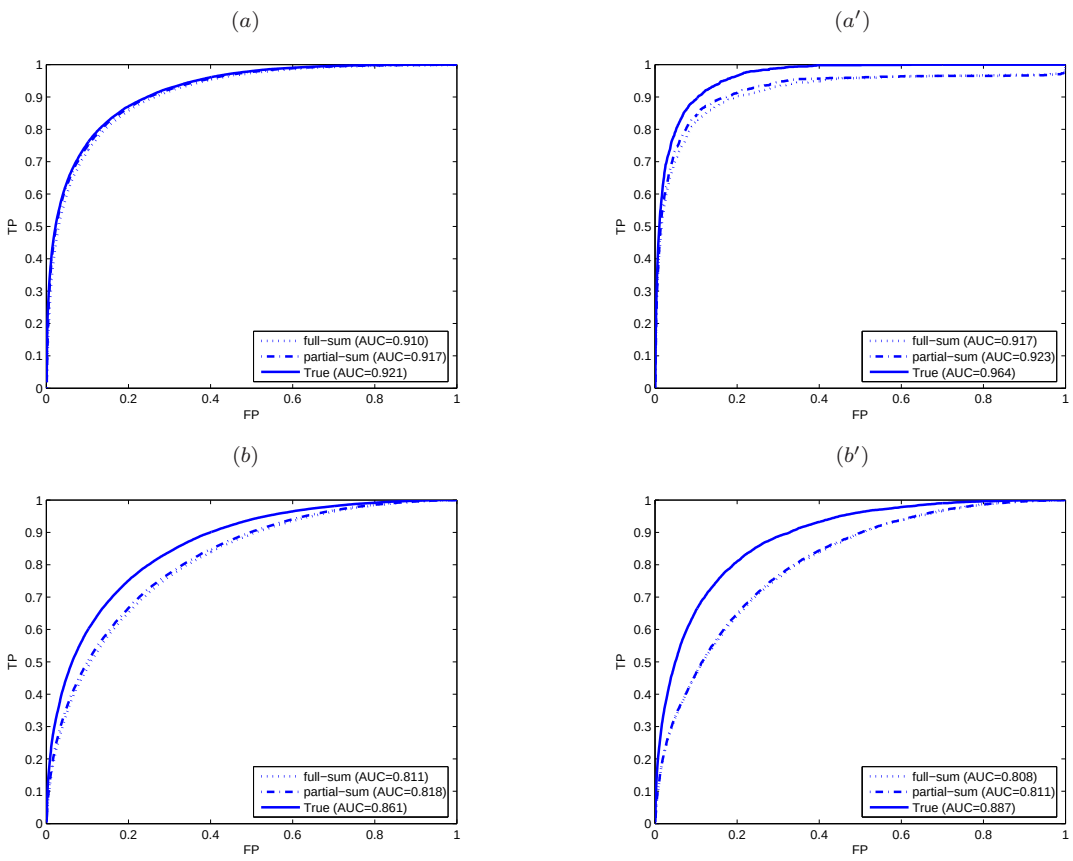

Fig. 1. ROC curves over 50 replicates.

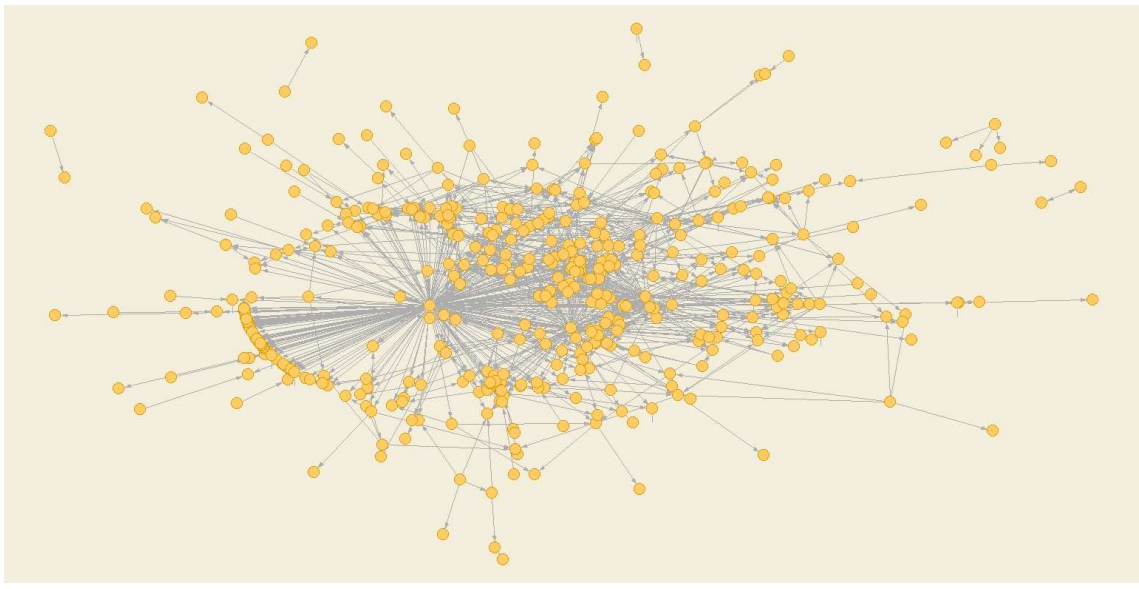

Fig. 2. Gene regulatory network of E. coli, prepared by Pajek.

can be decomposed into two low-rank matrices and applying eigen-decomposition for Lyapunov equations, this new approach significantly reduces the model complexity, and thus improves algorithmic efficiency and reliability of the existing methods.

In the future, we would like to investigate the theoretical properties of link prediction by our method. Recently, Du and Zhao, 2017 studied the consistency of the classical semi-supervised learning and suggested the estimator of the regression function is inconsistent when the tuning parameter λ is non-zero. Similarly, we conjecture that it would be unrealistic

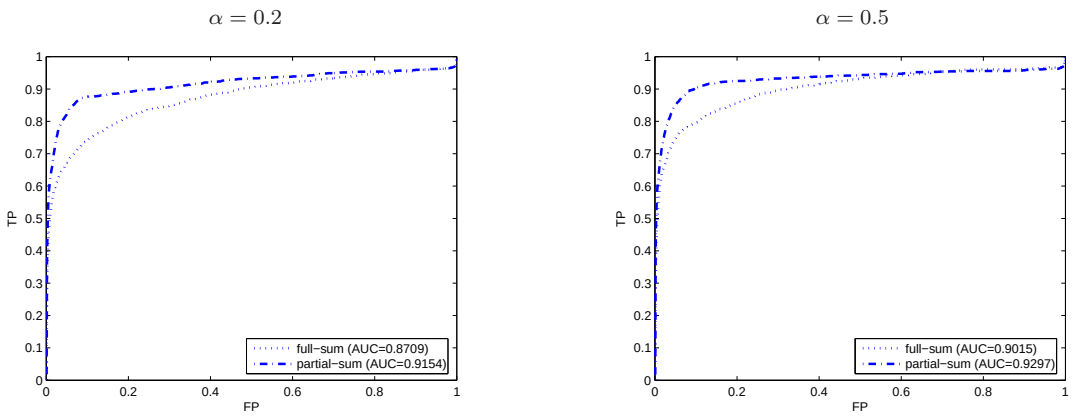

Fig. 3. ROC curves for gene regulatory network of E. coli.

to expect the consistency of the estimator of the probability matrix in the link prediction problem since it can be viewed as a generalization of the classical semi-supervised learning. On the other hand, since the performance of link prediction is usually evaluated by ROC curves, it would be more natural to investigate the asymptotic behavior of the corresponding AUCs (Area under the curve), which is potentially challenging because it requires novel theoretical tools for rank consistency.

We also plan to further study the algorithm for the partial-sum criterion. Unlike the full-sum version, the partial-sum criterion cannot be directly optimized by solving Lyapunov equations iteratively, and thus the proposed algorithm relies on an imputation technique. More analytic work is needed to better understand the principle of imputation used in the current algorithm.

Acknowledgements

Funding

This work has been supported by NSF DMS 1513004.

References

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, **25**(3), 211–230.

Bartels, R. H. and Steward, G. W. (1972). Solution of the matrix equation ax+xb=c. *Communications of the ACM*, **15**, 930–953.

Ben-Hur, A. and Noble, W. S. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics*, **21**(Suppl. 1), i38–i46.

Bleakley, K., Biau, G., and Vert, J.-P. (2007). Supervised reconstruction of biological networks with local models. *Bioinformatics*, **23**(13), i57–i65.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-supervised Learning*. The MIT Press.

Clauset, A., Moore, C., and Newman, M. E. J. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, **453**(7191), 98–101.

Du, C. and Zhao, Y. (2017). On consistency of graph-based semi-supervised learning.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLOS Biology*, **5**, 54–66.

Getoor, L. and Diehl, C. P. (2005). Link mining: A survey. *ACM SIGKDD Explorations Newsletter*, **7**(2), 3–12.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, **2**, 129–233.

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)*.

Hoff, P. D. (2007). Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, Cambridge, MA.

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.

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 2009 SIAM International Conference on Data Mining*.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, **18**(1), 39–43.

Leicht, E. A., Holme, P., and Newman, M. E. J. (2006). Vertex similarity in networks. *Physical Review E*, **73**, 026120.

Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, **58**(7), 1019–1031.

Lu, L. and Zhou, T. (2010). Link prediction in complex networks: A survey. arXiv:1010.0725v1.

Menon, A. and Elkan, C. (2011). Link prediction via matrix factorization. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6912 of *Lecture Notes in Computer Science*, pages 437–452. Springer Berlin Heidelberg.

Miller, K., Griffiths, T., and Jordan, M. (2010). Nonparametric latent feature models for link prediction. In Y. Bengio, D. Schuurmans, J. Lafferty, and C. Williams, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 22.

Newman, M. E. J. (2010). *Networks: An introduction*. Oxford University Press.

Raymond, R. and Kashima, H. (2010). Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. In J. Balczar, F. Bonchi, A. Gionis, and M. Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 131–147. Springer Berlin Heidelberg.

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.

Yu, K., Chu, W., Yu, S., Tresp, V., and Xu, Z. (2007). Stochastic relational models for discriminative link prediction. In *Proceedings of Neural Information Precessing Systems*, pages 1553–1560. MIT Press, Cambridge MA.

Zhao, Y., Wu, Y.-J., Levina, E., and Zhu, J. (2017). Link prediction for partially observed networks. *Journal of Computational and Graphical Statistics*, (To appear).