Learning Scale-Free Networks by Dynamic Node-Specific Degree Prior

Qingming Tang[1] QMTANG@TTIC.EDU
Toyota Technological Institute at Chicago, 6045 S. Kenwood Ave., Chicago, Illinois 60637, USA

Siqi Sun[1] SIQI@TTIC.EDU
Toyota Technological Institute at Chicago, 6045 S. Kenwood Ave., Chicago, Illinois 60637, USA

Jinbo Xu JINBO.XU@GMAIL.COM
Toyota Technological Institute at Chicago, 6045 S. Kenwood Ave., Chicago, Illinois 60637, USA

Abstract

Learning network structure underlying data is an important problem in machine learning. This paper presents a novel degree prior to study the inference of scale-free networks, which are widely used to model social and biological networks. In particular, this paper formulates scale-free network inference using Gaussian Graphical model (GGM) regularized by a node degree prior. Our degree prior not only promotes a desirable global degree distribution, but also exploits the estimated degree of an individual node and the relative strength of all the edges of a single node. To fulfill this, this paper proposes a ranking-based method to dynamically estimate the degree of a node, which makes the resultant optimization problem challenging to solve. To deal with this, this paper presents a novel ADMM (alternating direction method of multipliers) procedure. Our experimental results on both synthetic and real data show that our prior not only yields a scale-free network, but also produces many more correctly predicted edges than existing scale-free inducing prior, hub-inducing prior and the l_1 norm.

1. Introduction

Graphical models are widely used to describe the relationship between variables (or features). Estimating the structure of an undirected graphical model from a dataset has been extensively studied (Meinshausen & Bühlmann, 2006; Yuan & Lin, 2007; Friedman et al., 2008; Banerjee et al., 2008; Wainwright et al., 2006). Gaussian Graphical Models (GGMs) are widely used to model the data and l_1 penalty is used to yield a sparse graph structure. GGMs assume that the observed data follows a multivariate Gaussian distribution with a covariance matrix. When GGMs are used, the network structure is encoded in the zero pattern of the precision matrix. Accordingly, structure learning can be formulated as minimizing the negative log-likelihood (NLL) with an l_1 penalty. However, the widely-used l_1 penalty assumes that any pair of variables is equally likely to form an edge. This is not suitable for many real-world networks such as gene networks, protein-protein interaction networks and social networks, which are scale-free and contain a small percentage of hub nodes.

A scale-free network (BARABÁSI & Bonabeau, 2003) has node degree following the power-law distribution. In particular, a scale-free network may contain a few hub nodes, whose degrees are much larger than the others. That is, a hub node more likely forms an edge than the others. In real-world applications, a hub node is usually functionally important. For example, in a gene network, a gene playing functions in many biological processes (Zhang & Horvath, 2005; Goh et al., 2007) tends to be a hub. A few methods have been proposed to infer scale-free networks by using a reweighed l_1 norm. For example, (Peng et al., 2009) proposed a joint regression sparse method to learn scale-free networks by setting the penalty proportional to the estimated degrees. (Candes et al., 2008) proposed a method that iteratively reweighs the penalty in the l_1 norm based on the inverse of the previous estimation of precision matrix. This method can suppress the large bias in the l_1 norm when the magnitude of the non-zero entries vary a lot and is also closer to the l_0 norm. Some recent papers follow the idea of node-based learning using group lasso (Friedman et al., 2010b; Meier et al., 2008; Tan et al., 2014; Friedman et al., 2010a; Mohan et al., 2014). Since group lasso penal-

[1]Equal Contribution.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

ty promotes similar patterns among variables in the same group, it tends to strengthen the signal of hub nodes and suppress that of non-hub nodes. As such, group lasso may only produce edges adjacent to nodes with a large degree (i.e., hubs).

In order to capture the scale-free property, (Liu & Ihler, 2011) minimizes the negative log-likelihood with a scale-free prior, which approximates the degree of each variable by the l_1 norm. However, the objective function is non-convex and thus, is hard to optimize. (Defazio & Caetano, 2012) approximates the global node degree distribution by the submodular convex envelope of the node degree function. The convex envelope is the Lovasz extension (Lovász, 1983; Bach, 2010) of the sum of logarithm of node degree. These methods consider only the global node degree distribution, but not the degree of an individual node. However, the approximation function used to model this global degree distribution may not induce a power-law distribution, because there are many possible distributions other than power-law that optimizing the approximation function. See **Theorem** 1 for details.

To further improve scale-free network inference, this paper introduces a novel node-degree prior, which not only promotes a desirable global node degree distribution, but also exploits the estimated degree of an individual node and the relative strengths of all the edges adjacent to the same node. To fulfill this, we use node and edge ranking to dynamically estimate the degree of an individual node and the relative strength of one potential edge. As dynamic ranking is used in the prior, the objective function is very challenging to optimize. This paper presents a novel ADMM (alternating direction method of multipliers) (Boyd et al., 2011; Fukushima, 1992) method for this.

2. Model

2.1. Notation & Preliminaries

Let $G = (V, E)$ denote a graph, where V is the vertex set ($p = |V|$) and E the edge set. We also use E to denote the adjacency matrix of G, i.e., $E_{ij} = 1$ if and only if $(i, j) \in E$. In this paper we assume that the variable set V has a Gaussian distribution with a precision matrix X. Then, the graph structure is encoded in the zero pattern of the estimated X, i.e. the edge formed by X is $E(X) = \{(u, v), X_{uv} \neq 0\}$. Let $F(X)$ denote the objective function. For GGMs, $F(X) = tr(XS) - \log\det(X)$, where S is the empirical covariance matrix. To induce a sparse graph, a l_1 penalty is applied to obtain the following objective function.

$$\hat{X} = \arg\min_X F(X) + \lambda\|X\|_1 \tag{1}$$

Here we define two ranking functions. For a given i, let $(.)$ denote the permutation of $(1, 2, ..., p-1)$ such that $|X_{i,(1)}| \geq |X_{i,(2)}| \geq ... \geq |X_{i,(p-1)}|$ where $(X_{i,(1)}, X_{i,(2)}, ..., X_{i,(p-1)})$ is a shuffle of a row vector X_i excluding the element $X_{i,i}$. Let M be a $p-1$ dimension positive vector and V a p dimension vector. We define $X_i \circ M = \sum_{u=1}^{p-1} |X_{i,(u)}| M_u$. Let $[X, M, V]$ be a permutation of $(1, 2, ..., p)$ such that $X_{[1,X,M,V]} \circ M + V_1 \geq X_{[2,X,M,V]} \circ M + V_2 \geq ... \geq X_{[p,X,M,V]} \circ M + V_p$, where $[j, X, M, V]$ denote the j^{th} element of $[X, M, V]$. When V is a zero vector, we also write $[X, M, V]$ as $[X, M]$ and $[j, X, M, V]$ as $[j, X, M]$, respectively.

To promote a sparse graph, a natural way is to use a l_1 penalty to regulate $F(X)$. However, l_1 norm cannot induce a scale-free network. This is because that a scale-free network has node degree following a power-law distribution, i.e., $P(d) \propto d^{-\gamma}$ where d is the degree and γ is a constant ranging from 2 to 3, rather than a uniform distribution. A simple prior for scale-free networks is the sum of the logarithm of node degree as follows.

$$\lambda \sum_{v=1}^{p} \gamma \log(d_v + 1), \tag{2}$$

where $d_v = \sum_{u=1}^{p} I[X_{vu} \neq 0, u \neq v]$ and I is the indicator function. The constant 1 is added to handle the situation when $d_v = 0$. The l_0 norm in (2) is non-differentiable and thus, hard to optimize. One way to resolve this problem is to approximate d_v by the l_1 norm of X_v as shown in (Liu & Ihler, 2011). Since the logarithm of l_1 approximation is non-convex, a convex envelope of (2) based on submodular function and Lovasz extension (Lovász, 1983) is introduced recently in the papers (Bach, 2010; Defazio & Caetano, 2012) as follows.

$$\sum_{v=1}^{p} \sum_{u=1}^{p-1} |X_{v,(u)}| \Big(\log(u+1) - \log(u) \Big). \tag{3}$$

Let $h(u) = (\log(u+1) - \log(u))$. The convex envelope of Eq. 2 can be written as

$$\sum_{v=1}^{p} \sum_{u=1}^{p-1} h(u) |X_{v,(u)}|. \tag{4}$$

2.2. Node-Specific Degree Prior

Although Eq. 2 and its approximations can be used to induce a scale-free graph, they still have some issues, as explained in **Theorem** 1.

Theorem 1. *Let $S(G)$ denote the sum of logarithm of node degree of a graph G. For any graph G satisfying the scale-free property, there exists another graph G' with the same number of edges but not satisfying the scale-free property such that $S(G') < S(G)$.*

Since we would like to use (2) as a regularizer, a smaller value of (2), denoted as $S(G)$, is always favorable. Here we briefly prove **Theorem** 1. Since G is scale-free, a large percentage of its nodes have small degrees. It is reasonable to assume that G has two nodes u' and v' with the same degree. Let a denote their degree. Without loss of generality, there exists a node x connecting to u' but not to v'. We construct a graph $G^{(1)}$ from G by connecting x to v' and disconnecting x from u'. Since $\log$ is a concave function, we have

$$\begin{aligned} S(G^{(1)}) &= S(G) - 2\log(a+1) + \log(a) + \log(a+2) \\ &< S(G) \end{aligned} \tag{5}$$

If there are two nodes with the same degree in $G^{(i)}$, we can construct $G^{(i+1)}$ from $G^{(i)}$ such that $S(G^{(+1)}) < S(G^{(i)}) < .... < S(G^{(1)}) < S(G)$. By this procedure, we may finally obtain a non-scale-free graph with a smaller $S(G)$ value. So, **Theorem** 1 is proved.

Since **Theorem** 1 shows that a global description of scale-free property is not the ideal choice for inducing a scale-free graph, we propose a new prior to describe the degree of each individual node in the target graph.

Intuitively, given a node u, if we rank all the potential edges adjacent to u in descending order by $|X_{u,v}|$ where $v \in V - u$, then (u, v) with a higher rank is more likely to be a true edge than those with a lower rank and should receive less penalty. To account for this intuition, we introduce a non-decreasing positive sequence H and the following prior for node u

$$H \circ X_u = \sum_{v=1}^{p-1} H_v |X_{u,(v)}|. \tag{6}$$

Here (v) represents the node ranked in the v^{th} position. When all the elements in H are same, this prior simply is l_1 norm. In our work, we use a moderately increasing positive sequence H (e.g. $H_v = \log(v+1)^\alpha$ for $v = 1, 2, ..., p-1$), such that a pair (u, v) with a larger estimated $|X_{u,v}|$ has a smaller penalty. On the other hand, we do not want to penalize all the edges very differently since the penalty is based upon only rough estimation of the edge strength.

Let d_u denote the degree of node u, we propose the following prior based on Eq. 6.

$$\sum_{u=1}^{p} \frac{H \circ X_u}{H_{d_u}} \tag{7}$$

Comparing to the l_1 norm, our prior in Eq.7 is non-uniform since each edge has a different penalty, depending on its ranking and the estimated node degrees. This prior has the following two properties.

- When $v \leq d_u$, the penalty for $X_{u,(v)}$ is less than or equal to 1; otherwise, the penalty is larger than 1.
- If node i has a higher degree than node j, then $X_{i,(u)}$ has a smaller penalty than $X_{j,(u)}$.

The first property implies that Eq.(7) favors a graph following a given degree distribution. Suppose that in the true graph, node (or variable) i ($i = 1, 2, ..., p$) has degree d_i. Let $E^{(1)}$ and $E^{(2)}$ denote two edge sets of the same size. If $E^{(1)}$ has the degree distribution $(d_1, d_2, ..., d_p)$ but $E^{(2)}$ does not, then we can prove the following result (see Supplementary for proof).

$$\sum_{u=1}^{p} \frac{H \circ E_u^{(1)}}{H_{d_u}} \leq \sum_{u=1}^{p} \frac{H \circ E_u^{(2)}}{H_{d_u}} \tag{8}$$

The second property implies that Eq.(7) is actually robust. Even if we cannot exactly estimate the node degree $(d_1, d_2, ..., d_p)$, Eq.(7) may still work well as long as we can accurately rank nodes by their degrees.

2.3. Dynamic Node-Specific Degree Prior

We have introduced a prior (7) that favors a graph with a given degree distribution. Now the question is how to use (7) to produce a scale-free graph without knowing the exact node degree.

Let $\#_k$ denote the expected number of nodes with degree k. We can estimate $\#_k$ from the prior degree distribution $p_0(d) \propto d^{-\gamma}$ when γ is given. Supposing that all the nodes are ranked in descending order by their degree, we use τ_i to denote the estimated degree of the i^{th} ranked node based on the power-law distribution, such that $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_p$. Further, if we know the desired number of predicted edges, say we are told to output a edges, then the expected degree τ'_v of node v is assumed to be proportional to $a \times \frac{\tau_v}{\sum_{i=1}^{p} \tau_i}$. In the following content, we would just use τ_v to denote the expected degree of node v.

Now the question is how to rank nodes by their degrees? Although the exact degree d_v of node v is not available, we can approximate $\log(d_v + 1)$ by Lovasz Extenion (Defazio & Caetano, 2012; Bach, 2010; Lovász, 1983), i.e., $\sum_{u=1}^{p-1} |X_{v,(u)}| \Big(\log(u+1) - \log(u) \Big)$ (see Eq. 2). That is, we can rank nodes by their Lovasz Extension. Note that although we use Lovasz Extension in our implementation, other approximations such as l_1 norm can also be used to rank nodes without losing accuracy.

We define our dynamic node-specific prior as follows.

$$\Omega(X) = \sum_{v=1}^{p} \frac{X_{[v,X,h]} \circ H}{H_{\tau_v}} \tag{9}$$

Note that $h(u) = \log(u+1) - \log(u)$ for $1 \leq u \leq p$ and $[X, h]$ defines the ranking based on Lovasz Extension. The i-th ranked node is assigned a node penalty $\frac{1}{H_{\tau_i}}$, denoted as $g(i)$. Note that the ranking of nodes by their degrees is not static. Instead, it is determined dynamically by our optimization algorithm. Whenever there is a new estimation of X, the node and edge ranking may be changed. That is, our node-specific degree prior is dynamic instead of static.

3. Optimization

With prior defined in (9), we have the following objective function:

$$F(x) + \beta\Omega(X) \tag{10}$$

Here β is used to control sparsity level. The challenge of minimizing Eq.(10) lies in the fact that we do not know the ranking of both nodes and edges in advance. Instead we have to determine the ranking dynamically. We use an ADMM algorithm to solve it by introducing dual variables Y and U as follows.

$$X^{l+1} = \arg\min_X F(X) + \frac{\rho}{2}\|X - Y^l + U^l\|_F^2 \tag{11}$$
$$Y^{l+1} = \arg\min_Y \beta\Omega(Y) + \frac{\rho}{2}\|X^{l+1} - Y + U^l\|_F^2 \tag{12}$$
$$U^{l+1} = U^l + X^{l+1} - Y^{l+1} \tag{13}$$

We can use a first-order method such as gradient descent to optimize the first subproblem since $F(X)$ is convex. In this paper, we assume that $F(x)$ is a Gaussian likelihood, in which case (11) can be solved by eigen-decomposition. Here we describe a novel algorithm for (12).

Let $A = X^{l+1} + U^l$ and $\lambda = \frac{\beta}{\rho}$. Minimizing (12) is equivalent to

$$\min_Y \frac{1}{2}\|Y - A\|_F^2 + \lambda\Omega(Y), \tag{14}$$

which can be relaxed as

$$\min_Y \frac{1}{2}\|Y - A\|_F^2 + \lambda\sum_{v=1}^{p} g(v)Y_{[v]} \circ H$$
$$s.t. \quad g([v]) = g([v, Y, h]) \quad 1 \le v \le p. \tag{15}$$

Here, we simply use $[\cdot]$ to denote a permutation of $\{1, 2, ..., p-1\}$, and use $[v]$ to denote the v^{th} element of this permutation. The reason we introduce (15) is, given Y and without the constraint of (15), the optimal $[.]$ is $[Y, H]$. Adding the constraint $g([v]) = g([v, Y, h])$, (15) can be relaxed by solving the following problem until $g([v, Y, H, \delta]) = g([v, Y, h]) \quad 1 \le v \le p$.

$$\min_{Y,\delta} \frac{1}{2}\|Y - A\|_F^2 + \lambda\sum_{v=1}^{p} g(v)\{Y_{[v,Y,H,\delta]} \circ H\} \tag{16}$$

Here δ is the dual vector and can be updated by $\delta(v) = \mu \cdot (g([v, Y, H, \delta]) - g([v, Y, h])$ for $1 \le v \le p$, where μ is the step size. Actually, as only a small percentage of nodes have large degrees, we may speed up by using the condition $g([v, Y, H, \delta]) = g([v, Y, h])$ for $1 \le v \le k$ where k is much smaller than p. That is, instead of ranking all the nodes, we just need to select top k of them, which can be done much faster. We now propose **Algorithm 1** to solve (16), which in spirit is a dual decomposition (Sontag et al., 2011) algorithm.

Algorithm 1 Update of node ranking

```
1: Randomly generate $Y^0$. Set $t = 0, \delta^0 = 0$ and compute $[Y^0, H, \delta^0]$
2: while TRUE do
3:     $Y^{t+1} = \arg\min_Y \frac{1}{2}\|Y - A\|_F^2$
4:         $+\lambda\sum_v g(v)Y_{[v,Y^t,H,\delta^t]} \circ H$
5:     if $[Y^{t+1}, H, \delta^t] = [Y^t, H, \delta^t] = [Y^{t+1}, h]$ then
6:         break
7:     else
8:         $\delta^{t+1} = \delta^t + \mu(g([Y^{t+1}, H, \delta^t]) - g([Y^{t+1}, h]))$
9:     end if
10:    $t = t + 1$
11: end while
```

Theorem 2. *The output Y' of algorithm 1 satisfies the following condition.*

$$Y' = \arg\min_Y \{\frac{1}{2}\|Y - A\|_F^2 + \lambda\sum_{v=1}^{p} g(v)Y_{[v,Y',h]} \circ H\} \tag{17}$$

See supplementary for the proof of **Theorem** 2.

Solving line 3-4 in **Algorithm** 1 is not trivial. We can reformulate it as follows.

$$\min_Y \sum_{v=1}^{p} \{\frac{1}{2}\|Y_v - A_v\|_F^2 + Y_v \circ H'(v)\}. \tag{18}$$

Here $H'(v)$ is a $p-1$ dimension vector and $H'_u(v) = \lambda g(v)H_u$ for $1 \le u \le p-1$. Since Y is symmetric, (18) can be reformulated as follows.

$$\min_Y \sum_{v=1}^{p} \{\frac{1}{2}\|Y_v - A_v\|_F^2 + Y_v \circ H'(v)\}$$
$$s.t. \quad Y = Y^T. \tag{19}$$

Problem (19) can be solved iteratively using dual decomposition by introducing the Lagrangian term $tr(\sigma(Y - Y^T))$, where σ is a p by p matrix which would be updated by $\sigma = \sigma + \mu(Y - Y^T)$. Notice that $tr(\sigma(Y - Y^T)) =$

$tr((\sigma - \sigma^T)Y)$, (19) can be decomposed into p independent sub-problems as follows.

$$\min_{Y_v} \frac{1}{2}\|Y_v - (A + \sigma^T - \sigma)_v\|_F^2 + Y_v \circ H'(v) \qquad (20)$$

Let $B = A + \sigma^T - \sigma$. Obviously $Y_{v,v} = B_{v,v}$ holds, so we do not consider $Y_{v,v}$ in the remaining section. Let $\hat{Y}_v$=$\{\hat{Y}_{v,(1)}, \hat{Y}_{v,(2)}, ..., \hat{Y}_{v,(p-1)}\}$ be a feasible solution of (20). We define the cluster structure of $\hat{Y}_v$ as follows.

Definition 3 Let $\{\hat{Y}_{v,(1)}, \hat{Y}_{v,(2)}, ..., \hat{Y}_{v,(p-1)}\}$ be a ranked feasible solution. Supposing that $|\hat{Y}_{v,(1)}| = |\hat{Y}_{v,(2)}| = ... = |\hat{Y}_{v,(t)}| > |\hat{Y}_{v,(t+1)}|$, we say $\{\hat{Y}_{v,(1)}, \hat{Y}_{v,(2)}, ..., \hat{Y}_{v,(t)}\}$ form cluster 1 and denote it as $C|_{\hat{Y}_v}(1)$. Similarly, we can define $C|_{\hat{Y}_v}(2)$, $C|_{\hat{Y}_v}(3)$ and so on. Assume that $\{\hat{Y}_{v,(1)}, \hat{Y}_{v,(2)}, ..., \hat{Y}_{v,(p-1)}\}$ is clustered into $T(\hat{Y}_v)$ groups. Let $\left|C|_{\hat{Y}_v}(k)\right|$ denote the size of $C|_{\hat{Y}_v}(k)$ and $y|_{\hat{Y}_v}(k)$ the absolute value of its element.

Assuming that Y_v^* is the optimal solution of (20), by Definition 3, for $1 \le k \le T(Y_v^*)$, Y_v^* has the following property.

$$y|_{Y_v^*}(k) = \max\{0, \frac{\sum_{i \in C|_{Y_v^*}(k)} \{|B_{v,(i)}| - H_i'(v)\}}{\left|C|_{Y_v^*}(k)\right|}\} \qquad (21)$$

See (Defazio & Caetano, 2012) and our supplementary material for detailed proof of (21). Based on (21), we propose a novel dynamic programming algorithm to solve (20), which can be reduced to the problem of finding the constrained optimal partition of $\{B_{v,(1)}, B_{v,(2)}, ..., B_{v,(p-1)}\}$.

Let $Y_{v,1:t}^* = \{C|_{Y_{v,1:t}^*}(1), C|_{Y_{v,1:t}^*}(2), ..., C|_{Y_{v,1:t}^*}(m = T(Y_{v,1:t}^*))\}$ be the optimal partition for $\{B_{v,(1)}, B_{v,(2)}, ..., B_{v,(t)}\}$. Let $C|_{Y_{v,1:t}^*}(m+1) = \{|B_{v,(t+1)}| - H_{t+1}'(v)\}$ and $C|_{Y_{v,1:t}^*}(0) = \{\infty\}$. Then we have the following theorem.

Theorem 3. $Y_{v,1:t+1}^* = \{C|_{Y_{v,1:t}^*}(1), ..., C|_{Y_{v,1:t}^*}(k-1), C_k\}$, *where* C_k *is a set with* $\Sigma_{s=k}^{m+1}|C|_{Y_{v,1:t}^*}(s)|$ *elements which are equal to* y_k.

$$y_k = \max\{0, \frac{\sum_{i \in \bigcup_{s=k}^{m+1} C|_{Y_{v,1:t}^*}(s)} \{|B_{v,(i)}| - H_i'(v)\}}{\Sigma_{s=k}^{m+1}|C|_{Y_{v,1:t}^*}(s)|}\} \qquad (22)$$

and k *is the largest value such that* $y|_{Y_{v,1:t}^*}(k-1) > y_k$.

Theorem 3 clearly shows that this problem satisfies the optimal substructure property and thus, can be solved by dynamic programming. A $O(p\log(p))$ algorithm (**Algorithm** 2) to solve (20) is proposed. See supplementary material for the proof of substructure property and the correctness of the $O(p\log(p))$ algorithm. In **Algorithm 2**, $Rep(x, p)$ duplicates x by p times.

Algorithm 2 Edge rank updating

```
1: Input: $B_v$ and $H'(v)$
2: Output: $Y_v^*$
3: Sort $B_v$, get $\{B_{v,(1)}, B_{v,(2)}, ..., B_{v,(p-1)}\}$
4: Initialize $t = 0$, $sum(0) = 0$ and $Y_{1..0}^* = \emptyset$
5: $t = t + 1$
6: while $t < p$ do
7:     $sum(t) = sum(t-1) + |B_{v,(t)}| - H_t'(v)$
8:     $m = T(Y_{1..t-1}^*)$, $C|_{Y_{v,1:t}^*}(m+1) = \{|B_{v,(t+1)}| - H_{t+1}'(v)\}$ and $C|_{Y_{v,1:t}^*}(0) = \{\infty\}$
9:     Binary search to find out the largest index $b$ among $1, ..., m+1$, such that $y|_{Y_{v,1:t}^*}(b-1) > \max\{0, \frac{\sum_{i \in \bigcup_{s=b}^{m+1} C|_{Y_{v,1:t}^*}(s)}\{|B_{v,(i)}| - H_i'(v)\}}{\Sigma_{s=b}^{m+1}|C|_{Y_{v,1:t}^*}(s)|}\}$
10:    $S = |\cup_{s=1}^{b-1} C|_{Y_{1..t-1}^*}(s)|$
         Set $C|_{Y_{1..t}^*}(b) =$
           $\{Rep(max\{\frac{sum(i) - sum(S)}{t - S}, 0\}, t - S)$
11:    $Y_{1..t}^* = \cup_{s=1}^{b-1} C|_{Y_{1..t-1}^*}(s) + C|_{Y_{1..t}^*}(b)$
12: end while
```

4. Related Work & Hyper Parameters

To introduce scale-free prior ($p(d) \propto d^{-\alpha}$), (Liu & Ihler, 2011) proposed to approximate the degree of node i by $\|X_{-i}\|_1 = \sum_{j \neq i}|X_{i,j}|$ and then use the following objective function.

$$F(X) + \alpha \sum_{i=1}^{p} \log(\|X_{-i}\|_1 + \epsilon_i) + \beta \sum_{i=1}^{p} |X_{i,i}|, \qquad (23)$$

where ϵ_i is the parameter for each node to smooth out the scale-free distribution. Without prior knowledge, it is not easy to determine the value of ϵ_i. Note that the above objective function is not convex even when $F(x)$ is convex because of the log-sum function involved. The objective is optimized by reweighing the penalty for each $X_{i,j}$ at each step, and the method (denoted as RW) is guaranteed to converge to a local optimal. The parameter ϵ is set as diagonal of estimated X in previous iteration, and β as $2\frac{\alpha}{\epsilon_i}$, as suggested by the authors. They use α to control the sparsity, i.e. the number of predicted edges.

Recently (Defazio & Caetano, 2012) proposed a Lovasz extension approach to approximate node degree by a convex function. The convex function is a reweighed l_1 with larger penalty applied to edges with relatively larger strength. It turns out that such kind of convex function prior does not work well when we just need to predict a few edges, as shown in the experiments. Further, both (Liu & Ihler, 2011) and (Defazio & Caetano, 2012) consider only the global degree distribution instead of the degree of each node.

(Tan et al., 2014) proposes a method (denoted as Hub) specifically for a graph with a few hubs and applies a group lasso penalty. In particular, they decompose X as $Z + V + V^T$, where Z is a sparse symmetric matrix and V is a matrix whose column are almost entirely zero or non-zero. Intuitively, Z describes the relationship between non-hubs and V that between hubs. They formulate the problem as follows.

$$\min_{V,Z} \; F(X) + \lambda_1 \|Z\|_1 + \lambda_2 \|V\| + \lambda_3 \sum_{j=1}^{p} \|V_j\|_2$$
$$s.t. \;\; X = Z + V + V^T \tag{24}$$

An ADMM algorithm is used to solve this problem. In our test, we use $\lambda_3 = 0.01$ to yield the best performance. Besides,we set $\lambda = \lambda_1 = \lambda_2$ to produce a graph with a desired level of sparsity.

Our method uses 2 hyper-parameters: γ and β. Meanwhile, γ is the hyper parameter for the power-law distribution and λ controls sparsity.

5. Results

We tested our method on two real gene expression datasets and two types of simulated networks: (1) a scale-free network generated by Barabasi-Albert (BA) model (Albert & Barabási, 2002) and (2) a network with a few hub nodes. We generated the data for the simulated scale-free network by its corresponding Multivariate Gaussian distribution. We compared our method (denoted as "DNS", short for "Dynamic Node-Specific Prior") with graphical lasso ("Glasso") (Friedman et al., 2008), neighborhood selection ("NS") (Meinshausen & Bühlmann, 2006), reweighted l_1 regularization ("RW") (Liu & Ihler, 2011), Lovasz extenion ("Lovasz") (Defazio & Caetano, 2012) and a recent hub detection method ("Hub") (Tan et al., 2014).

5.1. Performance on Scale-Free Networks

We generated a network of 500 nodes (p = 500) from the BA model. Each entry Ω_{uv} of the precision matrix is set to 0.3 if (u, v) forms an edge, and 0 otherwise. To make the precision matrix positive definite, we set the diagonal of Ω to the minimum eigenvalue of Ω plus 0.2. In total we generate 250 data samples from the corresponding multivariate Gaussian distribution ($i.e., n = 250$). The hyper-parameters of all the methods are set as described in the last section.

Our method is not sensitive to the hyper-parameter γ. As shown in Figure 2, a few different γ values (2.1, 2.5, and 2.9) yield almost the same result. Hence we use $\gamma = 2.5$ in the following experiments.

Moreover, as shown in Figure 2, our method DNS outperforms the others in terms of prediction accuracy. It is not surprising that both RW and Hub outperform Glasso and NS since the former two methods are specifically designed for scale-free or hub networks. Lovasz, which is also designed for scale-free networks, would outperform Glasso as the number of predicted edges increase.

Figure 1 displays the log-log degree distribution of the true network and the networks estimated by DNS, RW and Glasso. Both DNS and RW yield networks satisfying the power-law distribution while Glasso does not, which confirms that both DNS and RW indeed favor scale-free networks.

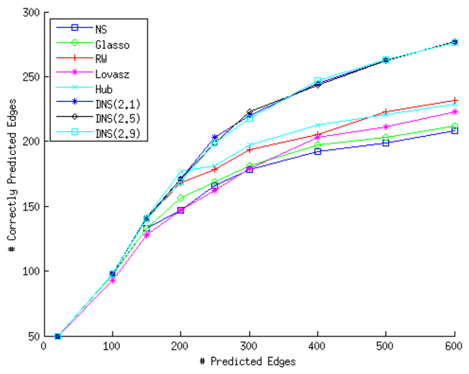

Figure 2. Simulation results on a scale free network. Gaussian Graphical Model is used with $n = 250$ and $p = 500$. X-axis is the number of predicted edges and Y-axis is the number of correctly predicted edges.

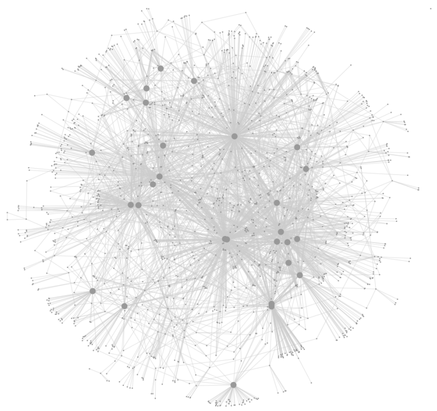

Figure 3. Visualization of the hub network with 1643 nodes and 3996 edges. For visualization purpose we ignore a connected component with less than or equal to 4 nodes. We also highlight the hub nodes, whose degree is at least 30.

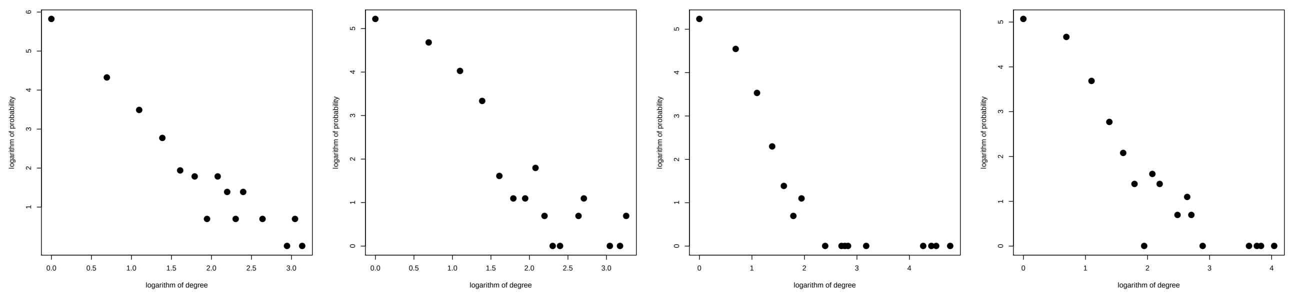

Figure 1. From left to right, the log-log degree distribution of (1) the true network and the estimated networks by (2) DNS (3) RW and (4) Glasso, respectively. Linear relationship is expected since the true network is scale-free. The network yielded by Glasso violates the power-law distribution most, as evidenced by a point close to (2,0).

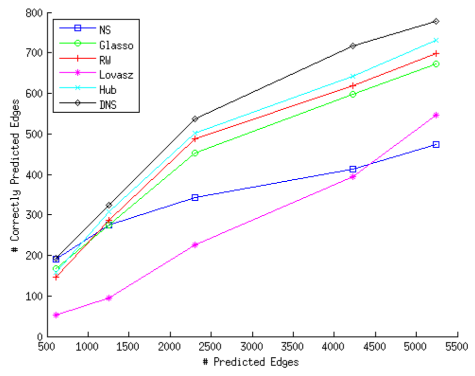

Figure 4. Simulation results of a hub network. Gaussian Graphical Model is used with $n = 806$ and $p = 1643$. X-axis is the number of predicted edges and Y-axis is the number of correctly predicted edges.

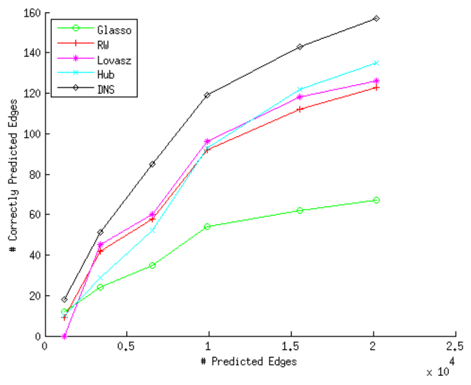

Figure 5. The performance of all the tested methods on a real gene expression data set (DREAM5 dataset 3). X-axis is the number of predicted edges and Y-axis is the number of correctly predicted edges.

5.2. Performance on Hub Networks

We also tested our method on a hub network, which contains a few nodes with very large degrees but not strictly follows the scale-free property. See Figure 3 for a visualization, where larger dots indicate the hub nodes. Here we use the DREAM5 Network Inference Challenge dataset 1, which is a simulated gene expression data with 806 samples. DREAM5 also provides a ground truth network for this dataset. See (Marbach et al., 2012) for more details.

The result in Figure 4 shows that our method outperforms all the others, although our method is not specifically designed for hub networks. This shows that DNS also performs well in a graph with non-uniform degree distribution but without strict scale-free property.

5.3. Gene Expression Data

To further test our method, we used DREAM5 dataset 3 and 4 respectively. Dataset3 contains 805 samples and 4511 genes and its ground truth consists of 3902 edges. Dataset4 contains 536 samples and 5950 genes and its ground truth consists of 3940 edges. The two datasets are very challenging as the data is noisy and without Gaussian and scale-free property. For each dataset, up to $100,000$ predicted edges are allowed for submission for each team competing in the contest. See (Marbach et al., 2012) for a detailed description of the two data sets. We determine the hyper parameters of all the methods such that they output exactly the same number of edges. As shown in Figure 5, our method obtains much higher accuracy than the others on DREAM5 dataset 3. To compare the degree distribution of different methods, we chose the values of the hyper-parameters such

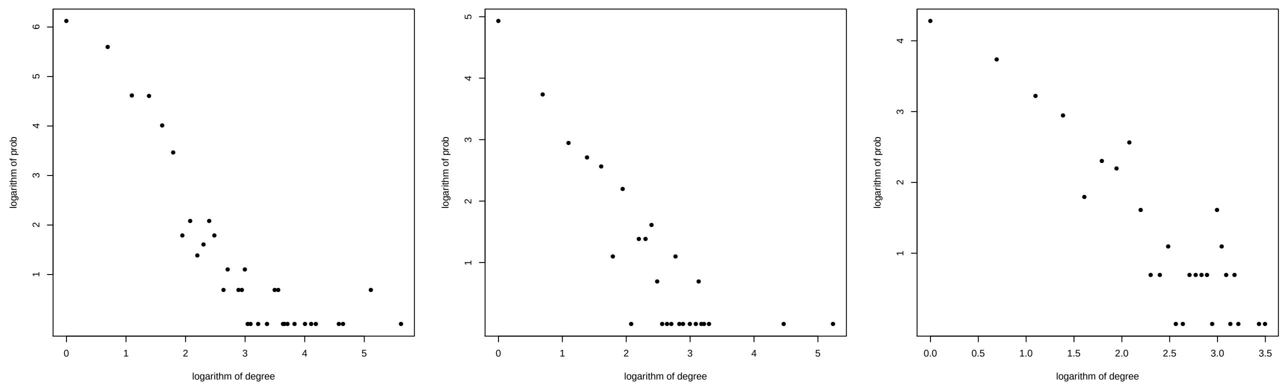

Figure 6. From left to right, the log-log degree distribution plot for (1) the true network and the estimated networks by (2) Glasso and (3) DNS, respectively.

that each method yields 4000 edges. The degree distributions of the resultant networks are shown in Figure 6. As shown in this figure, our estimated network is more similar to a scale-free network.

We also tested our result on DREAM5 dataset 4. As most algorithms are time consuming, we just run our method DNS and Glasso. According to our test, both our algorithm and Glasso perform not very good on this dataset (Actually, all DREAM5 competitors do not do well on this dataset (Marbach et al., 2012)). But our algorithm still outperforms Glasso in terms of accuracy. Actually, the accuracy of DNS is about two times of Glassso (e.g. when predicting about 19000 edges, Glasso generate 9 correct edges while DNS find 18 correct edges).

6. Conclusions

We have presented a novel node-specific degree prior to study the inference of scale-free networks, which are widely used to model social and biological networks. Our prior not only promotes a desirable global node degree distribution, but also takes into consideration the estimated degree of each individual node and the relative strength of all the possible edges adjacent to the same node. To fulfill this, we have developed a ranking-based algorithm to dynamically model the degree distribution of a given network. The optimization problem resulting from our prior is quite challenging. We have developed a novel ADMM algorithm to solve it.

We have demonstrated the superior performance of our prior using simulated data and three DREAM5 datasets. Our prior greatly outperforms the others in terms of the number of correctly predicted edges, especially on the real gene expression data.

The idea presented in this paper can potentially be useful to other degree-constrained network inference problem. In particular, it might be applied to infer protein residue-residue interaction network from a multiple sequence alignment, for which we may predict the degree distribution of each residue using a supervised machine learning method.

References

Albert, Réka and Barabási, Albert-László. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

Bach, Francis R. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems*, pp. 118–126, 2010.

Banerjee, Onureena, El Ghaoui, Laurent, and d'Aspremont, Alexandre. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

BARABÁSI, BY ALBERT-LÁSZLÓ and Bonabeau, Eric. Scale-free. *Scientific American*, 2003.

Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Candes, Emmanuel J, Wakin, Michael B, and Boyd, Stephen P. Enhancing sparsity by reweighted 1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

Defazio, Aaron and Caetano, Tiberio S. A convex formulation for learning scale-free networks via submodular relaxation. In *Advances in Neural Information Processing Systems*, pp. 1250–1258, 2012.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Rob. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33 (1):1, 2010a.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010b.

Fukushima, Masao. Application of the alternating direction method of multipliers to separable convex programming problems. *Computational Optimization and Applications*, 1(1):93–111, 1992.

Goh, Kwang-Il, Cusick, Michael E, Valle, David, Childs, Barton, Vidal, Marc, and Barabási, Albert-László. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

Liu, Qiang and Ihler, Alexander T. Learning scale free networks by reweighted l1 regularization. In *International Conference on Artificial Intelligence and Statistics*, pp. 40–48, 2011.

Lovász, László. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pp. 235–257. Springer, 1983.

Marbach, Daniel, Costello, James C, Küffner, Robert, Vega, Nicole M, Prill, Robert J, Camacho, Diogo M, Allison, Kyle R, Kellis, Manolis, Collins, James J, Stolovitzky, Gustavo, et al. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804, 2012.

Meier, Lukas, Van De Geer, Sara, and Bühlmann, Peter. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

Meinshausen, Nicolai and Bühlmann, Peter. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pp. 1436–1462, 2006.

Mohan, Karthik, London, Palma, Fazel, Maryam, Witten, Daniela, and Lee, Su-In. Node-based learning of multiple gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488, 2014.

Peng, Jie, Wang, Pei, Zhou, Nengfeng, and Zhu, Ji. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104 (486), 2009.

Sontag, David, Globerson, Amir, and Jaakkola, Tommi. Introduction to dual decomposition for inference. *Optimization for Machine Learning*, 1:219–254, 2011.

Tan, Kean Ming, London, Palma, Mohan, Karthik, Lee, Su-In, Fazel, Maryam, and Witten, Daniela. Learning graphical models with hubs. *The Journal of Machine Learning Research*, 15(1):3297–3331, 2014.

Wainwright, Martin J, Lafferty, John D, and Ravikumar, Pradeep K. High-dimensional graphical model selection using l_1-regularized logistic regression. In *Advances in neural information processing systems*, pp. 1465–1472, 2006.

Yuan, Ming and Lin, Yi. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Zhang, Bin and Horvath, Steve. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.