RC24812 (W0906-086) June 18, 2009
Computer Science

IBM Research Report

Sparse MRF Learning with Priors on Regularization Parameters

Katya Scheinberg
Columbia University

Narges Bani Asadi
Stanford University

Irina Rish
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Sparse MRF Learning with Priors on Regularization Parameters

Abstract

In this paper, we consider the sparse inverse covariance selection problem which is equivalent to structure recovery of a Markov Network over Gaussian variables. The problem of regularization parameter(s) selection is addressed in a Bayesian way, by assuming a prior on the parameter(s) and by using MAP optimization to find both the inverse covariance matrix and the unknown parameters. Our general formulation extends prior art by allowing a vector of regularization parameters and is well-suited for learning structured graphs such as scale-free networks where the sparsity of nodes varies significantly. We also introduce a novel and efficient approach to solving the sparse inverse covariance problem that compares favorably to the state-of-art. Our empirical results demonstrate advantages of our approach on structured (scale-free) networks.

1 INTRODUCTION

We address the problem of learning the structure of a sparse Markov network (Markov Random Field, or MRF) over Gaussian variables, which is equivalent to learning the zero-pattern of the inverse covariance matrix. The accuracy of the structure reconstruction can be very sensitive to the choice of this regularization parameter(s), and the problem of the "proper" selection of this parameter in practical settings remains open, despite theoretical advances that analyze asymptotic behavior [1].

[1] As mentioned in [6], "the general issue of selecting a proper amount of regularization for getting a right-sized structure or model has largely remained a problem with unsatisfactory solutions".

Our approach suggests an automated way of selecting the regularization parameter(s) in sparse MRF learning, and generalizes a previously proposed approach of [1] to the vector of regularization parameters. In our framework, the regularization parameter(s) controlling the sparsity of solution is considered to be a random variable with certain prior, and the objective is to find a maximum a posteriory probability (MAP) solution (Θ, Λ), where Θ is the set of model parameters and Λ is the set of regularization parameters. Our algorithm is based on alternating optimization over Θ and Λ, respectively.

Our general formulation is well-suited for learning structured networks with potentially very different node degrees (and thus different sparsity of the columns in the inverse covariance matrix). One common practical example of such networks are networks with heavy-tail (power-law) degree distributions, also called scale-free networks. Examples of such networks include social networks, protein interaction networks, Internet, world wide web, correlation networks between active brain areas in fMRI studies [8], and many other real-life networks (see [3] for a survey). Our empirical results compare a wide variety of approaches to the regularization parameter selection and vector-based approaches appear to be an attractive choice.

Moreover, we propose a novel algorithm (SINCO) for sparse inverse covariance matrix reconstruction given a regularization parameter. SINCO solves the primal problem (unlike its predecessors such as COVSEL of [7], using coordinate descent, is very efficient and naturally preserving the sparsity of the solution. As is seen from our computational results SINCO has better capability in reducing the false positives error than *glasso* [4], since it rarely introduces unnecessary nonzero elements.

2 MAP APPROACH TO SPARSE GAUSSIAN MRF LEARNING

Let $X = \{X_1, ..., X_p\}$ be a set of p random variables, and let $G = (V, E)$ be a Markov network (a Markov Random Field, or MRF) representing the conditional independence structure of the joint distribution $P(X)$. The set of vertices $V = \{1, ..., p\}$ is in a one-to-one correspondence with the set of variables in X. The edge set E contains an edge (i, j) if and only if X_i is conditionally dependent on X_j given all remaining variables; the lack of edge between X_i and X_j means that the two variables are conditionally independent given all remaining variables [**?**].

We will assume a multivariate Gaussian probability density function over $X = \{X_1, ..., X_p\}$:

$$p(\mathbf{x}) = (2\pi)^{-p/2} \det(\Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\mu)} \quad (1)$$

where μ is the mean and $\mathbf{\Sigma}$ is the covariance matrix of the distribution, respectively, and $\mathbf{x}^T$ denotes the transpose of the column-vector $\mathbf{x}$. Without loss of generality we will assume that the data are normalized to have zero mean ($\mu = \mathbf{0}$), and we only need to estimate the parameter Σ (or Σ^{-1}). Since $det(\Sigma)^{-1} = det(\Sigma^{-1})$, we can now rewrite eq. 1, assuming $C = \Sigma^{-1}$ and $\mu = 0$:

$$p(\mathbf{x}) = (2\pi)^{-p/2} \det(C)^{\frac{1}{2}} e^{-\frac{1}{2}\mathbf{x}^T C \mathbf{x}}. \quad (2)$$

Missing edges in the above graphical model correspond to zero entries in the inverse covariance matrix $C = \Sigma^{-1}$, and thus the problem of structure learning for the above probabilistic graphical model is equivalent to the problem of learning the zero-pattern of the inverse-covariance matrix. Note that the inverse of the maximum-likelihood estimate of the covariance matrix Σ (i.e. the empirical covariance matrix $A = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{x}_i$ where $\mathbf{x}_i$ is the i-th sample, $i = 1, ..., n$), even if it exists, does not typically contain any elements that are exactly zero. Therefore an explicit sparsity-enforcing constraint needs to be added to the estimation process.

A common approach is to include as penalty the l_1-norm of C, which is equivalent to imposing a Laplace prior on C in maximum-likelihood framework [7, 4, 10]. Formally, the entries C_{ij} of the inverse covariance matrix C are assumed to be independent random variables, each following a Laplace distribution $p(C_{ij}) = \frac{\lambda_{ij}}{2} e^{-\lambda_{ij}|C_{ij} - \alpha_{ij}|}$ with zero location parameter (mean) α_{ij} and common scale parameter $\lambda_{ij} = \lambda$, yielding $p(C) = \prod_{i=1}^{p}\prod_{j=1}^{p} p(C_{ij}) = (\lambda/2)^{p^2} e^{-\lambda||C||_1}$, where $||C||_1 = \sum_{ij}|C_{ij}|$ is the (vector) l_1-norm of C. Then the objective is to find the maximum-likelihood solution $\arg\max_{C \succ 0} p(C|\mathbf{X})$, where $\mathbf{X}$ is the $n \times p$ data matrix, or equivalently, since $p(C|\mathbf{X}) = P(\mathbf{X}, C)/p(\mathbf{X})$ and $p(\mathbf{X})$ does not include C, to find $\arg\max_{C \succ 0} P(\mathbf{X}, C)$, over positive definite matrices C. This yields the following optimization problem considered in [7, 4, 10]:

$$\max_{C \succ 0} \ln\det(C) - \mathrm{tr}(AC) - \lambda||C||_1 \quad (3)$$

where $\det(Z)$ and $\mathrm{tr}(Z)$ denote the determinant and the trace (sum of the diagonal elements) of a matrix Z, respectively.

Herein, we make a more general assumption about $p(C)$, allowing different rows in C to have different parameters λ_i, i.e., $p(C_{ij}) = \frac{\lambda_i}{2} e^{-\lambda_i|C_{ij}|}$ This reflects our desire to model structured networks with potentially very different node degrees (i.e., row densities in C). This yields $p(C) = \prod_{i=1}^{p}\prod_{j=1}^{p} \frac{\lambda_i}{2} e^{-\lambda_i|C_{ij}|} = = \prod_{i=1}^{p} \frac{\lambda_i^p}{2^p} e^{-\lambda_i \sum_{j=1}^{p}|C_{ij}|}$.

Moreover, we will take Bayesian approach and assume that parameters λ_i are also random variables following some joint distribution $p(\{\lambda_i\})$. Given a dataset X of n samples (rows) of vector $\mathbf{X}$, the joint log-likelihood can be then written as

$$\ln L(X, C, \{\lambda_i\}) = \ln\{p(X|C)p(C|\{\lambda_i\})p(\{\lambda_i\})\} =$$
$$const + \frac{n}{2}\ln\det(C) - \frac{1}{2}\sum_{i=1}^{n}\mathbf{x}_i^T C \mathbf{x}_i + p\sum_{i}^{p}\ln\frac{\lambda_i}{2} -$$
$$- \sum_{i}^{p}\lambda_i\sum_{j=1}^{p}|C_{ij}| + \ln p(\{\lambda_i\}),$$

where *const* does not depend on C or $\{\lambda_i\}$.

We can also rewrite $\sum_{i=1}^{n}\mathbf{x}_i^T C \mathbf{x}_i = n\mathrm{tr}(AC)$ where tr denotes the trace of a matrix, and $A = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T\mathbf{x}_i$ is the empirical covariance matrix.

We will use the maximum a posteriori probability (MAP) approach that requires maximization of the above joint log-likelihood, rewritten as

$$\max_{C \succ 0, \{\lambda_i\}} \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)] - \sum_{i}^{p}\lambda_i\sum_{j=1}^{p}|C_{ij}| +$$
$$+ p\sum_{i}^{p}\ln\lambda_i + \ln p(\{\lambda_i\}),$$

where $C \succ 0$ constraint ensures the solution C (inverse covariance matrix) is positive definite.

We will consider independent λ_i following *exponential priors* $p(\lambda_i) = b_i e^{-b_i\lambda_i}$. This yields:

$$\max_{C \succ 0, \lambda \in \mathbf{R}^p} \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)] - \sum_{i}^{p}\lambda_i\sum_{j=1}^{p}|C_{ij}| +$$
$$p\sum_{i}^{p}\ln\lambda_i - \sum_{i}^{p} b_i\lambda_i. \quad (4)$$

Rather than taking a more expensive, fully Bayesian approach here and integrating out C in order to obtain the maximum-likelihood type II estimate of b_i, we will use an approximate estimate $b_i = ||A(i)_r^{-1}||_1/p$, where $A_r = A + \epsilon I$ is the empirical covariance matrix[2], and $A_r(i)$ denotes its i-th row. In other words, b_i is estimated as an average l_1-norm per element of i-th row. We also considered the truncated (to exclude negative values of λ) unit-variance Gaussian prior which replaces $\sum_i^p b_i\lambda_i$ in the equation above with $\sum_i^p(\lambda_i - b_i)^2/2$.

3 SINCO: NEW METHOD FOR SPARSE INVERSE-COVARIANCE MATRIX SELECTION

Let S be some given $p \times p$ symmetric matrix with non-negative entries. We are considering the following optimization problem:

$$\max_{C \succ 0} \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)] - \lambda\|C\|_S \qquad (5)$$

Here by $\|C\|_S$ we denote the sum of absolute values of some elements of the matrix $S \cdot C$, where $\cdot$ denotes the element-wise product. For example if S is a matrix of all ones, then $\|\cdot\|_S$ equals to the sum of the absolute values of all the element of a matrix and the problem reduces to the problem addressed in [7] (i.e., eq. 3 with λ replaced by $(2/n)\lambda$).

The dual of this problem can be written similarly to the dual in [7]

$$\max_{W \succ 0}\{\frac{n}{2}\ln\det(W) - np/2 : \text{ s.t. } -S \le \frac{n}{2}(W - A) \le S\}, \qquad (6)$$

here the inequalities involving matrices W, A and S are element-wise.

The optimality conditions for this pair of primal and dual problems imply that $W = C^{-1}$ and that $(n/2)W_{ij} - A_{ij} = S_{ij}$ if $C_{ij} > 0$ and $(n/2)W_{ij} - A_{ij} = -S_{ij}$ if $C_{ij} < 0$. Hence, $\|C\|_S = \mathrm{tr}(((n/2)W - A)C) = np/2 - \mathrm{tr}(AC)$.

3.1 OPTIMIZATION ALGORITHM

We now consider possible optimization algorithms for solving problems 5 or 6. These problems are convex and the interior point method can be applied, however, the per-iteration cost of the interior point method or any other second-order method prohibits their use for many practical instances. Several other methods were proposed, such as *glasso* [4] and block coordinate descent [7], Nesterov's smoothing method [9], [7] and a projected gradient [**?**]. With the exception of *glasso*, the other proposed methods solve the dual problem and hence do not exploit the potential sparsity of the dual solution. Here we describe another version of coordinate descent applied to the primal problems, with a small cost per iteration. We refer to this method as SINCO, for Sparse INverse COvariance problem. The advantage of this method is that it works directly on the primal matrix and updates it two elements at a time, hence, naturally preserving the sparsity of the solution. As is seen from our computational results SINCO has better capability in reducing the false positives error than *glasso* [4], since it rarely introduces unnecessary nonzero elements to the primal matrix C. Here we briefly describe the method.

First let us consider a reformulation of the problem 5:

$$\max_{C',C''} \quad \frac{n}{2}[\ln\det(C' - C'') - \mathrm{tr}(A(C' - C''))] - \\ -\mathrm{tr}(S(C' + C'')), \qquad \text{s.t.} C' \ge 0,\ C'' \ge 0.$$

The method we propose works as follows:

0. Initialize $C' = I,\ C'' = 0$
1. find $W = (C' - C'')^{-1}$;
2. Form the gradient $G' = \frac{n}{2}(W - A) - S$ and $G'' = -S - \frac{n}{2}(W + A)$
3. For each pair (i, j) such that
   $G'_{ij} > 0, C''_{ij} = 0$, compute the maximum by updating C' along the direction $e_i e_j^T + e_j e_i^T$
   $G'_{ij} < 0, C'_{ij} > 0$, compute the maximum by updating C' along the direction $-e_i e_j^T - e_j e_i^T$
   $G''_{ij} > 0, C'_{ij} = 0$, compute the maximum by updating C'' along the direction $e_i e_j^T + e_j e_i^T$
   $G''_{ij} < 0, C''_{ij} > 0$, compute the maximum by updating C'' along the direction $-e_i e_j^T - e_j e_i^T$
4. Choose the step which provide the maximum function improvement
5. Update W^{-1} and the function value and repeat.
6. end

The key to this very simple-minded coordinate descent algorithm is the fact that the maximum the one-dimensional function in Step 3 is available in a closed form. Indeed, consider the step $\bar{C}' = C' + \theta(e_i e_j^T + e_j e_i^T)$.

The inverse W, then, is updated, according to the Sherman-Morrison-Woodbury formula [**?**], as follows

$$\begin{aligned}
\bar{W} &= W - \theta(\kappa_1 W_i W_j^T + \kappa_2 W_i W_i^T + \kappa_3 W_j W_j^T + \kappa_1 W_j W_i^T) \\
\kappa_1 &= -(1 + \theta W_{ij})/(\theta^2(W_{ii} * W_{jj} - W_{ij}^2) - 1 - 2 * \theta * W_{ij}) \\
\kappa_2 &= \theta W_{jj}/(\theta^2(W_{ii} * W_{jj} - W_{ij}^2) - 1 - 2\theta W_{ij}) \\
\kappa_3 &= \theta W_{ii}/(\theta^2(W_{ii} * W_{jj} - W_{ij}^2) - 1 - 2\theta * W_{ij}).
\end{aligned}$$

[2] slightly regularized with small $\epsilon = 10^{-3}$ on the diagonal to obtain an invertible matrix when A is not invertible.

Let us consider the derivative of the objective function with respect to θ

$$f'(\theta) = (nW_{ij} - nA_{ij} - S_{ij} - S_{ji}) + N\theta(W_{ii}W_{jj} + W_{ij}^2)/(\theta^2(W_{ii}W_{jj} - W_{ij}^2) - 1 - 2\theta W_{ij}) + 2\theta W_{ij}(W_{ii}W_{jj} - W_{ij}^2)) \quad ;$$

To find the value of θ for which the derivative of the objective function equals zero we need to solve the following quadratic equation

$$ab\theta^2 + (na - 2W_{ij}b)\theta - (nW_{ij} - nA_{ij} - S_{ij} - S_{ji}) = 0,$$

where $a = W_{ii}W_{jj} - W_{ij}^2$ and $b = -nA_{ij} - S_{ij} - S_{ji}$. Notice that a is always nonnegative, because matrix W is positive definite, and it equals to zero only when $i = j$. We know that at $\theta = 0$ $f'(0) > 0$. Let us investigate what happens when θ grows. The discriminant of the quadratic equation is

$$D = (Na + 2W_{ij}b)^2 + 4ab^2 \geq 0,$$

hence the quadratic equation always has a solution. It is possible to show, by analyzing the quadratic equation that the maximum of the function of θ can be found from the solution of this quadratic equations.

The other possible steps listed in Step 3 can be analyzed analogously. The objective function value is easy to update using the formula

$$det(C' - C'' + \theta(e_i e_j^T + e_j e_i^T)) = det(C' - C'')(1 + 2\theta W_{ij} - \theta^2 a));$$

Each step can be computed by a constant number of arithmetic operations, hence to find the step that provided maximum function value improvement it takes $O(n^2)$ operations - the same amount of work (up to a constant) that it takes to update W and the gradient after one iteration. Hence the per-iteration complexity is small. Moreover, this algorithms lends itself readily to massive parallelization.

The convergence of the method follows from the convergence of a block-coordinate descent method on a strictly convex objective function. The only constraints are box constraints (nonnegativity) and they do not hinder the convergence.

4 FIXED-POINT METHOD FOR λ SELECTION

We shall now address the optimization problem arising in selection of parameter λ as discussed in Section 2.

4.1 SCALAR λ

We consider the following optimization problems:

$$\max_{C,\lambda} \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)] - \lambda\|C\|_S + p^2\ln\lambda - \theta(\lambda), \quad (7)$$

where $\theta(\lambda)$ is some given function of λ derived from the prior introduced in Sections 2 and 3. By $\|C\|_S$ we denote the sum of absolute values of the elements of the matrix $S \cdot C$, where $\cdot$ denotes the element-wise product, where S is a given $p \times p$ matrix with nonnegative entries.

Let $f(C) = \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)]$, and let us consider the following function:

$$\phi(\lambda) = \max_C f(C) - \lambda\|C\|_S.$$

The function $\frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)] - \lambda\|C\|_S$ is strictly concave and, hence, has a unique maximizer $C(\lambda)$ for any value of λ. From general theory of convex optimization in [**?**] we know that $\phi(\lambda)$ is a differentiable convex function whose derivative for any given λ equals $\|C(\lambda)\|_S$. The proof of this simple fact can be found in the Appendix.

Lemma 4.1 *$\phi(\lambda)$ is a differentiable convex function whose derivative for any given λ equals $-\|C(\lambda)\|_S$.*

Now let us consider the following optimization problem

$$\max_\lambda \psi(\lambda) = \max_\lambda(\phi(\lambda) + p^2\ln\lambda - \theta(\lambda). \quad (8)$$

Clearly, the optimal solution to this problem is also optimal for problem (7). To find $\phi(\lambda)$ one needs to solve the the sparse inverse covariance selection problem with a fixed value of λ. This can be done by SINCO method described in the previous section or by several other methods, see e.g., [7, 4].

Notice that $\psi(\lambda)$ is a sum of a convex and a concave functions, hence is neither convex nor concave and may have multiple local optima. In our experiments with $\theta(\lambda) = b\lambda$ we observed that the maximum was unique in most of the cases. In the rare case when it appeared to be not unique, it was not clear if such was the true nature of $\psi(l)$ or a result of inaccuracies in the solution of the convex subproblems.[3]

[3]The derivative of $\psi(\lambda)$ is $p^2/\lambda - \|C(\lambda)\|_S - b$. Here are some observations which help explain why a unique stationary point is typical for the case when $\theta(\lambda) = b\lambda$. We are considering all point for which $\psi(\lambda) = 0$. Multiplying the expression for the gradient by nonnegative λ we have that

$$\lambda\psi'(\lambda) = p^2 - \lambda\|C(\lambda)\|_S - b\lambda$$

If the quantity $\lambda\|C(\lambda)\|_S$ increases as λ grows (which is expected since the decrease of λ usually slows down with the growth of λ) then the right hand side of the last equality is a decreasing function of λ. Hence the equality to zero can only be achieved for a single value of λ which would imply unique maximum. We are not yet aware of any theoretical result that guarantees that $\lambda\|C(\lambda)\|_S$ increases monotonically with λ but we have consistently observed it in the experiments. Note also that for sufficiently small λ the quantity $\lambda\psi'(\lambda)$ is positive, while for sufficiently large λ it becomes negative. Hence an existence of at least one local maximum is always guaranteed.

A similar analysis and an update rule can be derived for the Gaussian prior on λ.

We will describe the optimization scheme to solve problem 8 in the next section.

4.2 VECTOR λ

Now let us consider a similar problem to (7), but with λ - a vector of weights for the S-norm of the columns of C. Hence we will now consider a vector norm $\|C_i\|_{S_i}$ which is the same the matrix norm we discussed before, applied to columns (or rows) of C and S.

$$\max_{C,\lambda} \quad \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)] - \sum_{i=1}^{p}\lambda_i\|C_i\|_{S_i} + \quad p\sum_{i=1}^{p}\ln\lambda_i - \sum_{i=1}^{p} b_i\lambda_i$$

As before, let $f(C) = \frac{n}{2}[\ln\det(C) - \mathrm{tr}(AC)]$, and

$$\phi(\lambda) = \max_{C} f(C) - \lambda\|C\|_S.$$

Notice that, for any fixed $\bar{\lambda}$, $\sum_i \bar{\lambda}_i\|C_i\|_{S_i} = \|C\|_{\lambda\cdot S}$, where $\lambda\cdot S$ is a matrix whose i-th column equals $\lambda_i S_i$ for all i. This implies that for any fixed and given λ and $\bar{\lambda}$ function $\phi(\theta) = \phi(\lambda+\theta\bar{\lambda})$ reduces to the case of the univariate $\phi(\lambda)$ described in the previous section. This implies, for instance, that the multivariate function $\phi(\lambda)$ is convex in any direction, hence is convex in general. Also from the analysis in the previous section it is easy to see that the i-th element of the gradient of $\phi(\lambda)$ equals $-\|C_i(\lambda)\|_{S_i}$.

Now we consider $\psi(\lambda) = \phi(\lambda) + p\sum_{i=1}^{p}\ln\lambda_i - \sum_{i=1}^{p} b_i\lambda_i$. This function is again neither concave nor convex. Its gradient is

$$(\nabla\psi(\lambda))_i = -\|C_i(\lambda)\|_{S_i} + p/\lambda_i - b_i, \quad i = 1,\ldots,p,$$

where $C(\lambda)$ is, again, the maximizer of $f(C) - \lambda\|C\|_S$ for a given λ. Hence for λ^* which maximizes $\psi(\lambda)$ we have

$$\|C_i(\lambda^*)\|_{S_i} + b_i = p/\lambda_i^*, \quad i = 1,\ldots,p,$$

or, equivalently,

$$\lambda_i^* = \frac{p}{\|C_i(\lambda^*)\|_{S_i} + b_i} \quad , i = 1,\ldots,p.$$

Hence λ^* is a fixed point of the following operator $T(\lambda) = p/(\|C(\lambda)\|_S + b)$, where by $p/(\|C(\lambda)\|_S + b)$ we mean a p-dimensional vector with entries $p/(\|C_i(\lambda)\|_{S_i} + b_i)$. To solve this problem we consider applying the following fixed point algorithm

0. Initialize λ^1;
1. find $C(\lambda^k)$ and $\phi(\lambda^k)$;
2. If $\sum_i (p/\lambda_i - \|C_i(\lambda^k)\|_{S_i} - b_i)^2 < \epsilon$ go to step 4.
3. $\lambda_i^{k+1} = p/(\|C_i(\lambda^k)\|_{S_i} + b_i)$; go to step 1.
4. end

Note that in Step 1 we perform a standard inverse covariance selection optimization problem with fixed λ such as is done in the previous section.

In our experiments the fixed point algorithm presented above converged in every experiment. While we do not have theoretical guarantees of the convergence of the algorithm[4], we will present a modification of the algorithm which invokes a line search algorithm in case the fixed point iteration fails to provide sufficient improvement in the objective function $\psi(\lambda)$.

We apply the following optimization algorithm.

0. Initialize λ^1;
1. find $C(\lambda^k)$ and $\phi(\lambda^k)$;
2. If $\sum_i (p/\lambda_i - \|C_i(\lambda^k)\|_{S_i} - b_i)^2 < \epsilon$ go to step 5.
3. $\lambda_i^{k+1} = p/(\|C_i(\lambda^k)\|_{S_i} + b_i)$; (9)
4. find $C_i(\lambda^{k+1})$ and $\psi(\lambda^{k+1})$;
   if $\psi(\lambda^{k+1}) > \psi(\lambda^k)$ $k = k+1$, go to step 3.
   else $\lambda^{k+1} = (\lambda^k + \lambda^{k+1})/2$. Go to step 4.
5. end

The proposed algorithm performs a line search along the direction d defined by $d_i = p/(\|C_i(\lambda)\|_{S_i} + b_i) - \lambda_i$, while the gradient of $\psi(\lambda)$ equals g such that $g_i = p/\lambda_i - \|C_i(\lambda)\|_{S_i} - b_i$. If we consider the inner product,

[4]The fixed point algorithm is known to converge when the Lipschitz constant of operator T is less than 1. The Lipschitz constant of T at a given λ can be bounded by $\frac{pL_C(\lambda)}{(\|C(\lambda)\|_S + \min_i b_i)^2}$, where $L_C(\lambda)$ is the upper bound of Lipschitz constant of $\|C(\lambda)\|_S$ at λ. If our earlier observation, that $\|C(\lambda)\|_S$ typically reduces slower than λ grows, holds, then $L_C \leq 1$ and we can estimate the Lipschitz constant of T during the fixed point algorithm. Moreover, we can see that, since $\|C(\lambda)\|_S$ is monotonically decreasing and is bounded from below, then $L_C(\lambda) \to 0$ as $\lambda \to \infty$ and hence for large enough λ there is always at least one fixed point for operator T. Hence we can conclude that the maximum of $\psi(\lambda)$ is achieved. Note also that $\lambda_i^* \leq p/b_i$, $i = 1,\ldots,p$ and due to the fact that we are only interested in solutions $C(\lambda)$ such that $\|C(\lambda)\|_S$ is bounded from above by some predefined constant, we can assume w.l.o.g that we consider only $\lambda_i \geq \delta$ for all $i = 1,\ldots,p$ and for some $\delta > 0$.

we have $d^T g =$

$$\sum_i (p^2/(\lambda_i(\|C_i(\lambda)\|_{S_i} + b_i)) + \lambda_i(\|C_i(\lambda)\|_{S_i} + b_i) - 2p) =$$

$$\frac{1}{p}\sum_i (p/\lambda_i(\|C_i(\lambda)\|_{S_i} + b_i) + \lambda_i(\|C_i(\lambda)\|_{S_i} + b_i)/p - 2) \geq 0.$$

Hence, unless $p = \lambda_i\|C_i(\lambda)\|_{S_i}$ for all i, then we know the the direction d makes and obtuse angle with the gradient and, thus, is an ascent direction. In the case when $p = \lambda_i\|C_i(\lambda)\|_{S_i}$ for all i, then the gradient of $\psi(\lambda)$ is zero and the algorithm have converged to a local stationary point. In fact we can show that

$$d^T g/\|d\|\|g\| \geq const > 0,$$

for all cases when $\|d\|\|g\| > 0$, which means that the cosine of angle between the gradient and the direction d remains bounded away from zero, which will in turn imply that sufficient ascent can always be achieved by a line search along direction d. Indeed, from $\|d\|\|g\| \leq \frac{\|d\|^2+\|g\|^2}{2}$ we have $d^T g/\|d\|\|g\| \geq$

$$2\frac{\sum_{i=1}^p (p^2/(\lambda_i(\|C_i(\lambda)\|_{S_i} + b_i) + \lambda_i(\|C_i(\lambda)\|_{S_i} + b_i) - 2p)}{\sum_{i=1}^p ((\frac{p}{\lambda_i} - (\|C_i(\lambda)\|_{S_i} + b_i))^2 + (\frac{p}{(\|C_i(\lambda)\|_{S_i}+b_i)} - \lambda_i)^2)} \geq$$

$$\sum_{i=1}^p \lambda_i(\|C_i(\lambda)\|_{S_i} + b_i) \geq const > 0.$$

The last inequality comes from the facts that $(\|C_i(\lambda)\|_{S_i} + b_i) > b_i)$ and that $\lambda_i \geq \delta > 0$ for all $i = 1, \ldots, p$.

The advantage of the Algorithm (9) is that, while theoretically convergent to the optimum solution, it only resorts to line search if the initial fixed point iteration fails. Hence, in practice, no extra work is necessary to apply this algorithm. In our experiment the number of fixed point iterations was small compared to the dimension p and the algorithm worked very efficiently. The work of each iteration is essentially the same as the work taken by a single solve of the inverse covariance problem, but since the consecutive solves are related, one can successfully utilize warm starts.

5 EMPIRICAL EVALUATION

We performed experiments on semi-realistic synthetic data generated from "structured" random networks that followed a power-law degree distribution over the variables. The networks were generated using the preferential attachment (Barabasi-Albert) model [3] [5], that produces "scale-free" (power-law) networks containing (relatively few) very highly connected "hubs" besides a large number of sparsely connected nodes. Although such networks are sparse in terms of total number of edges, their power-law structure is a natural candidate for using vector rather than scalar sparsity parameter.

We generated power-law networks with density $5\%, 21\%$ and 31%, measured by the % of non-zero off-diagonal entries. For each density level, we generated 5 different power-law networks over $p = 100$ variables, that defined the structure of the "ground-truth" inverse covariance matrix, and for each of them, we generated 5 matrices with randomly generated covariances corresponding to the non-diagonal non-zero entries (bounded by 0.1 in order to ensure the resulting matrix is positive-definite)[6] We then sampled $n = 50, 100, 200, 500, 1000, 10000$ instances from the corresponding multivariate Gaussian distribution over $p = 100$ variables.

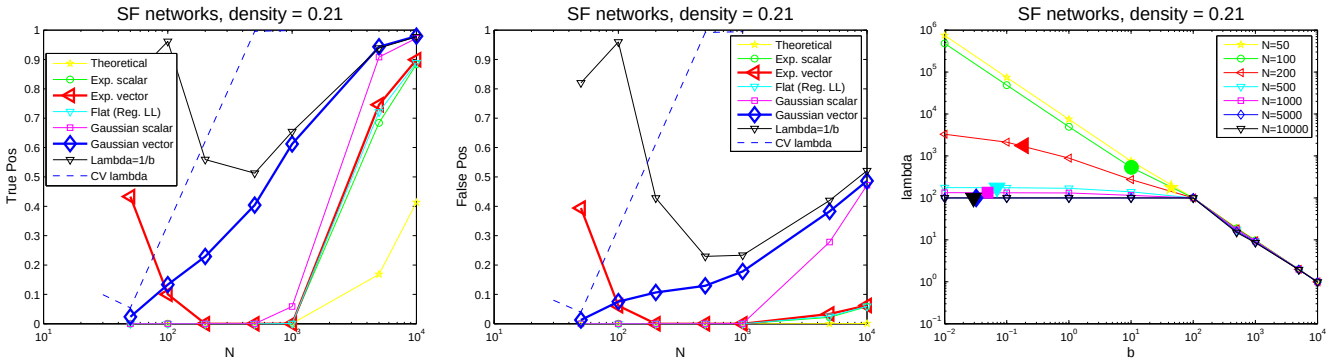
We evaluated the following methods of selecting λ when reconstructing the sparse MRF structure from data: (1) *theoretical* λ - theoretically derived λ in [2][7]; (2) *cross-validation* λ is selected as one giving best average prediction (i.e. minimizing the sum-squared prediction error) over the network nodes; (3) "exponential scalar" and (4) "exponential vector" correspond to parameter selected using our method with exponential prior, its scalar and vector versions, respectively; (5) Gaussian scalar and (6) Gaussian vector are defined similarly for Gaussian prior; (7) *fixed* $\lambda = 1/b$ simply assigns to λ the mean of the exponential distribution estimated directly from the data as mentioned in section 2; finally (8) *flat ("regularized likelihood")* corresponds to the flat-prior version described in [1] along with the other scalar priors. We report the *off-diagonal* true positive (TP) and false positive (FP) rates.

Figure 1 summarizes the results on scale-free networks with density 21%, comparing vector-lambda approach to the scalar approach and to a wide variety of other methods mentioned above (we observed similar type of results for other densities). We observed that:
1. cross-validation (CV) for prediction often selects nearly-zero λ, and thus is similar to unregularized ML estimate, selecting too many edges and having very high false-positive rate; this is not surprising as it is well known that cross-validated λ for the prediction objective can be a very bad choice for the structure/model selection in l_1-regularized setting (e.g., see [5] for examples);

[5] We used the open-source Matlab code available at http://www.mathworks.com/matlabcentral/fileexchange/11947

[6] The variance over the results was quite small.

[7] Theoretically derived λ has asymptotic guarantee of correct recovery of the *connectivity components* (rather than edges), which correspond to *marginal* rather than conditional independencies, i.e. to the entries in covariance rather then the inverse covariance matrix. Although such approach is asymptotically consistent, for finite number of samples it tends to miss many edges.



(a) TP (b) FP (c) lambda vs b

Figure 1: Results on scale-free networks (21% density).

2. theoretical (Banerjee's) λ is another extreme: its edge selection is too conservative in order to bound the false-positive rate of the covariance matrix entries asymptotically, and thus its true positive rate is close to zero unless the number of samples becomes very large;
3. *our approaches are in between the two extremes, for both exponential and Gaussian priors.*
4. *vector-λ approaches seem preferable in the relatively low-sample regime (especially Gaussian-vector-method), since their scalar counterparts tend to be too conservative and yield TP=0 in that regime*;
5. regularized likelihood behaves very similar to scalar-exponential method, but does not require parameter tuning;
6. simply setting $\lambda = 1/b$, i.e. to the mean of the exponential distribution, does not seem to work well as its FP rate is very high in both small-sample and large-sample regimes, only going somewhat doing in the mid-sample regime.
7. *Figure 1(c) shows that λ is not very sensitive to the choice of b for a wide range of bs when N gets sufficiently high.*

Figure 2 compares SINCO versus *glasso* in terms of their solution accuracy. Recall that SINCO solves the primal problem rather than dual, and its incremental updates naturally preserve the sparsity of the solution. Indeed, when we use SINCO vs *glasso* with the same λ (e.g., $\lambda = 1/b$ as shown in the first row of Figure 2), the false-positive error drops to zero as the number of samples grows, while *glasso* yields an *increasing* error that approaches 100%! This behavior remains to be understood better. Apparently, for the same gorwth rate of λ, the amount of regularization is enough for SINCO to reach 0%, but for *glasso* a higher λ may be required, otherwise the first term in 5 that grows linearly in n pushes the result towards unregularized (and dense) ML estimate. On the other hand, since eventually ML estimate must converge to the true inverse covariance matrix, the FP error must dropped, but apparently it happens later with *glasso*. Similarly, the second row of Figure 2 compares the two methods when used as subroutines at each iteration of our fixed-point method, for a particular value of the hyperparameter b. Apparently, SINCO leads to lower λ, and thus higher sparsity and (much) lower FP rate, while its true positive (TP) rate is very similar to the one of *glasso*.

References

[1] N. Bani Asadi, I. Rish, K. Scheinberg, D. Kanevsky, and B. Ramabhadran. A MAP Approach to Learning Sparse Gaussian Markov Networks. In *ICASSP*. 2009.

[2] O. Banerjee, L. El Ghaoui, A. d'Aspremont, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *ICML*, pages 89–96. 2006.

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

[4] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 2007.

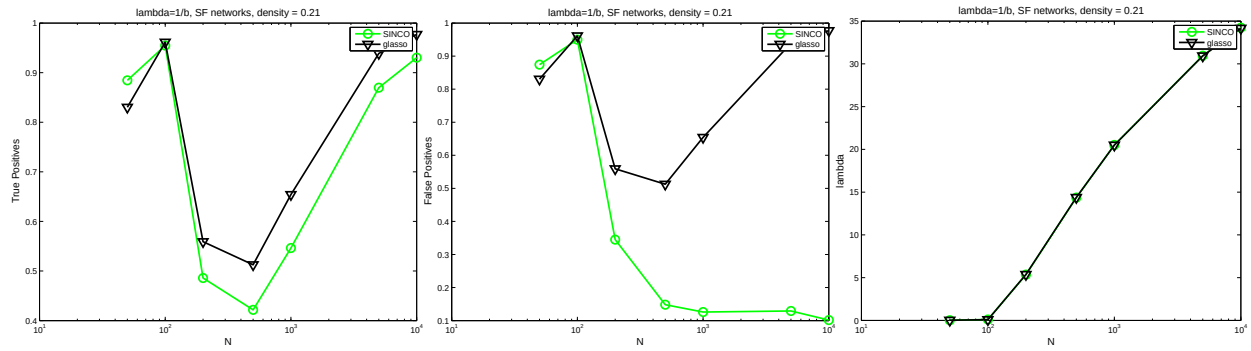
[5] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

[6] Nicolai Meinshausen and Peter Buehlmann. Stability selection, 2008.

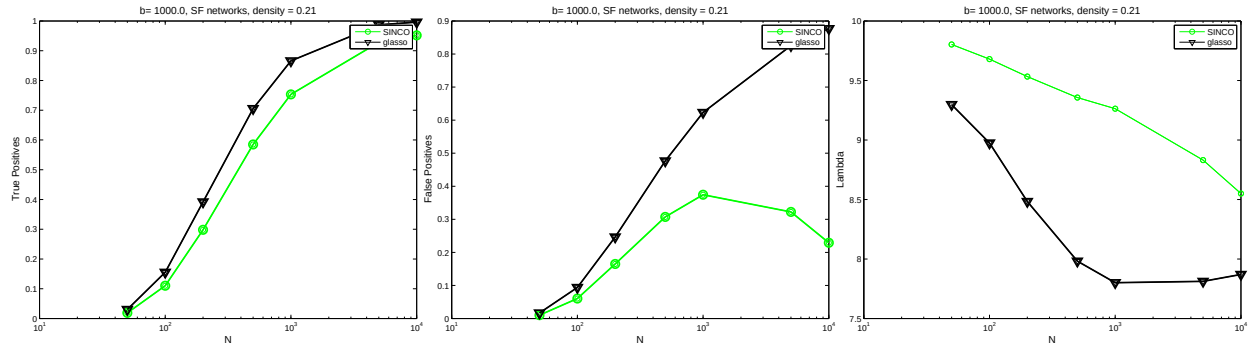
[7] O.Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, March 2008.

[8] V.M. Eguiluz and D.R. Chialvo and G.A. Cecchi and M. Baliki and A.V. Apkarian. Scale-free functional brain networks. *Physical Review Letters*, 94:018102, 2005.

[9] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Prog.*, 103(1):127–152, 2005.



Results for fixed $\lambda = 1/b$: (a) TP (b) FP (c)lambda



Results for b=10: (a) TP (b)FP (c)lambda

Figure 2: SINCO vs glasso on scale-free networks (21% density).

[10] M. Yuan and Y. Lin. Model Selection and Estimation in the Gaussian Graphical Model. *Biometrika*, 94(1):19–35, 2007.

6 Appendix

Proof of Lemma 4.1

Proof. Consider $\frac{\phi(\lambda+d\lambda)-\phi(\lambda)}{d\lambda} =$

$$= \frac{f(\lambda+d\lambda) - (\lambda+d\lambda)\|C(\lambda+d\lambda)\|_S - f(\lambda) + \lambda\|C(\lambda)\|_S}{d\lambda} =$$
$$\frac{f(\lambda+d\lambda) - \lambda\|C(\lambda+d\lambda)\|_S - f(\lambda) + \lambda\|C(\lambda)\|_S}{d\lambda} - \|C(\lambda+d\lambda)\|_S$$

We will now show that

$$\lim_{d\lambda\to 0} \frac{f(\lambda+d\lambda) - \lambda\|C(\lambda+d\lambda)\|_S - f(\lambda) + \lambda\|C(\lambda)\|_S}{d\lambda} = 0 \qquad (10)$$

Let us consider only $d\lambda > 0$ for a moment. Assume that

$$\limsup_{d\lambda\to 0} \frac{f(\lambda+d\lambda) - \lambda\|C(\lambda+d\lambda)\|_S - f(\lambda) + \lambda\|C(\lambda)\|_S}{d\lambda} \geq 2\epsilon > 0.$$

This means that there is an infinite sequence $d\lambda_k \to +0$ such that

$$f(\lambda+d\lambda_k) - \lambda\|C(\lambda+d\lambda_k)\|_S \geq f(\lambda) - \lambda\|C(\lambda)\|_S + \epsilon d\lambda_k.$$

Since $\epsilon d\lambda_k > 0$ this means that for some small enough $d\lambda_k$ $C(\lambda+d\lambda_k)$ is a better solution that $C(\lambda)$ for the given λ. Since by assumption $C(\lambda)$ is the maximizer, then we have reached a contradiction and the above lim sup equals to zero.

Now assume

$$\liminf_{d\lambda\to 0} \frac{f(\lambda+d\lambda) - \lambda\|C(\lambda+d\lambda)\|_S - f(\lambda) + \lambda\|C(\lambda)\|_S}{d\lambda} \leq 2\epsilon < 0.$$

Again we have a sequence $d\lambda_k \to +0$ for which

$$f(\lambda+d\lambda_k) - \lambda\|C(\lambda+d\lambda_k)\|_S \leq f(\lambda) - \lambda\|C(\lambda)\|_S + \epsilon d\lambda_k.$$

or

$$f(\lambda+d\lambda_k) - (\lambda+d\lambda_k)\|C(\lambda+d\lambda_k)\|_S \leq f(\lambda) - (\lambda+d\lambda_k)\|C(\lambda)\|_S$$
$$+\epsilon d\lambda_k + d\lambda(\|C(\lambda)\|_S - \|C(\lambda+d\lambda_k)\|_S).$$

Since $\|C(\lambda)\|_S - \|C(\lambda+d\lambda_k)\|_S) \to 0$ as $d\lambda_k \to 0$, then for large enough k we have

$$f(\lambda+d\lambda_k) - (\lambda+d\lambda_k)\|C(\lambda+d\lambda_k)\|_S < f(\lambda) - (\lambda+d\lambda_k)\|C(\lambda)\|_S,$$

which contradicts the fact that $C(\lambda+d\lambda_k)$ is the optimal solution for $\lambda+d\lambda_k$. The proof can be repeated almost identically for $d\lambda < 0$, hence we have shown (10).

It is now trivial to conclude that the derivative

$$\begin{aligned}\phi'(\lambda) &= \lim_{d\lambda\to 0} \frac{\phi(\lambda+d\lambda) - \phi(\lambda)}{d\lambda} \\ &= \lim_{d\lambda\to 0} -\|C(\lambda+d\lambda)\|_S = -\|C(\lambda)\|_S\end{aligned}$$

The convexity follows from the simple fact that as λ increases $\|C(\lambda)\|_S$ has to decrease, hence the derivative of $\phi(\lambda)$ increases.

■