

Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data

Onureena Banerjee
Laurent El Ghaoui
EECS Department
University of California, Berkeley
Berkeley, CA 94720 USA

ONUREENA@EECS.BERKELEY.EDU
ELGHAOUI@EECS.BERKELEY.EDU

Alexandre d'Aspremont
ORFE Department
Princeton University
Princeton, NJ 08544 USA

ASPREMON@PRINCETON.EDU

Editor: John Lafferty

Abstract

We consider the problem of estimating the parameters of a Gaussian or binary distribution in such a way that the resulting undirected graphical model is sparse. Our approach is to solve a maximum likelihood problem with an added ℓ_1 -norm penalty term. The problem as formulated is convex but the memory requirements and complexity of existing interior point methods are prohibitive for problems with more than tens of nodes. We present two new algorithms for solving problems with at least a thousand nodes in the Gaussian case. Our first algorithm uses block coordinate descent, and can be interpreted as recursive ℓ_1 -norm penalized regression. Our second algorithm, based on Nesterov's first order method, yields a complexity estimate with a better dependence on problem size than existing interior point methods. Using a log determinant relaxation of the log partition function (Wainwright and Jordan, 2006), we show that these same algorithms can be used to solve an approximate sparse maximum likelihood problem for the binary case. We test our algorithms on synthetic data, as well as on gene expression and senate voting records data.

Keywords: model selection, maximum likelihood estimation, convex optimization, Gaussian graphical model, binary data

1. Introduction

Undirected graphical models offer a way to describe and explain the relationships among a set of variables, a central element of multivariate data analysis. The principle of parsimony dictates that we should select the simplest graphical model that adequately explains the data. In this paper we consider practical ways of implementing the following approach to finding such a model: given a set of data, we solve a maximum likelihood problem with an added ℓ_1 -norm penalty to make the resulting graph as sparse as possible.

Many authors have studied a variety of related ideas. In the Gaussian case, model selection involves finding the pattern of zeros in the inverse covariance matrix, since these zeros correspond to conditional independencies among the variables. Traditionally, a greedy forward-backward search algorithm is used to determine the zero pattern (e.g., Lauritzen, 1996). However, this is computationally infeasible for data with even a moderate number of variables. Li and Gui (2005) introduce

a gradient descent algorithm in which they account for the sparsity of the inverse covariance matrix by defining a loss function that is the negative of the log likelihood function. Speed and Kiiveri (1986) and, more recently, Dahl et al. (Revised 2007) proposed a set of large scale methods for problems where a sparsity pattern for the inverse covariance is given and one must estimate the nonzero elements of the matrix.

Another way to estimate the graphical model is to find the set of neighbors of each node in the graph by regressing that variable against the remaining variables. In this vein, Dobra and West (2004) employ a stochastic algorithm to manage tens of thousands of variables. There has also been a great deal of interest in using ℓ_1 -norm penalties in statistical applications. d'Aspremont et al. (2004) apply an ℓ_1 norm penalty to sparse principle component analysis. Directly related to our problem is the use of the Lasso of Tibshirani (1996) to obtain a very short list of neighbors for each node in the graph. Meinshausen and Bühlmann (2006) study this approach in detail, and show that the resulting estimator is consistent, even for high-dimensional graphs. A related approach is the graphical lasso, explored by Friedman et al. (2007).

Our objective here is to both estimate the sparsity pattern of the underlying graph and to obtain a regularized estimate of the covariance matrix. The difficulty of the problem formulation we consider lies in its computation. Although the problem is convex, it is non-smooth and has an unbounded constraint set. As we shall see, the resulting complexity for existing interior point methods is $O(p^6)$, where p is the number of variables in the distribution. In addition, interior point methods require that at each step we compute and store a Hessian of size $O(p^2)$. The memory requirements and complexity are thus prohibitive for $O(p)$ higher than the tens. Specialized algorithms are needed to handle larger problems.

The remainder of the paper is organized as follows. We begin by considering Gaussian data. In Section 2 we set up the problem, derive its dual, discuss properties of the solution and how heavily to weight the ℓ_1 -norm penalty in our problem. In Section 3 we present a provably convergent block coordinate descent algorithm that can be interpreted a sequence of iterative ℓ_1 -norm penalized regressions. In Section 4 we present a second, alternative algorithm based on Nesterov's recent work on non-smooth optimization, and give a rigorous complexity analysis with better dependence on problem size than interior point methods. In Section 5 we show that the algorithms we developed for the Gaussian case can also be used to solve an approximate sparse maximum likelihood problem for multivariate binary data, using a log determinant relaxation for the log partition function given by Wainwright and Jordan (2006). In Section 6, we test our methods on synthetic as well as gene expression and senate voting records data.

2. Problem Formulation

In this section we set up the sparse maximum likelihood problem for Gaussian data, derive its dual, and discuss some of its properties.

2.1 Problem Setup

Suppose we are given n samples independently drawn from a p -variate Gaussian distribution: $y^{(1)}, \dots, y^{(n)} \sim \mathcal{N}(\mu, \Sigma_p)$, where the covariance matrix Σ is to be estimated. Let S denote the second moment matrix about the mean:

$$S := \frac{1}{n} \sum_{k=1}^n (y^{(k)} - \mu)(y^{(k)} - \mu)^T.$$

Let $\hat{\Sigma}^{-1}$ denote our estimate of the inverse covariance matrix. Our estimator takes the form:

$$\hat{\Sigma}^{-1} = \arg \max_{X \succ 0} \log \det X - \text{trace}(SX) - \lambda \|X\|_1. \tag{1}$$

Here, $\|X\|_1$ denotes the sum of the absolute values of the elements of the positive definite matrix X .

The scalar parameter λ controls the size of the penalty. The penalty term is a proxy for the number of nonzero elements in X , and is often used—albeit with vector, not matrix, variables—in regression techniques, such as the Lasso.

In the case where $S \succ 0$, the classical maximum likelihood estimate is recovered for $\lambda = 0$. However, when the number of samples n is small compared to the number of variables p , the second moment matrix may not be invertible. In such cases, for $\lambda > 0$, our estimator performs some regularization so that our estimate $\hat{\Sigma}$ is always invertible, no matter how small the ratio of samples to variables is.

Even in cases where we have enough samples so that $S \succ 0$, the inverse S^{-1} may not be sparse, even if there are many conditional independencies among the variables in the distribution. By trading off maximality of the log likelihood for sparsity, we hope to find a very sparse solution that still adequately explains the data. A larger value of λ corresponds to a sparser solution that fits the data less well. A smaller λ corresponds to a solution that fits the data well but is less sparse. The choice of λ is therefore an important issue that will be examined in detail in Section 2.3.

2.2 The Dual Problem and Bounds on the Solution

By expressing the ℓ_1 norm in (1) as

$$\|X\|_1 = \max_{\|U\|_\infty \leq 1} \text{trace}(XU),$$

where $\|U\|_\infty$ denotes the maximum absolute value element of the symmetric matrix U , we can write (1) as

$$\max_{X \succ 0} \min_{\|U\|_\infty \leq \lambda} \log \det X - \text{trace}(X, S + U).$$

This corresponds to seeking an estimate with the maximum worst-case log likelihood, over all additive perturbations of the second moment matrix S . A similar robustness interpretation can be made for a number of estimation problems, such as support vector machines for classification.

We can obtain the dual problem by exchanging the max and the min. The resulting inner problem in X can be solved analytically by setting the gradient of the objective to zero and solving for X . The result is

$$\min_{\|U\|_\infty \leq \lambda} -\log \det(S + U) - p$$

where the primal and dual variables are related as: $X = (S + U)^{-1}$. Note that the log determinant function acts a log barrier, creating an implicit constraint that $S + U \succ 0$.

To write things neatly, let $W = S + U$. Then the dual of our sparse maximum likelihood problem is

$$\hat{\Sigma} := \max \{ \log \det W : \|W - S\|_\infty \leq \lambda \}. \tag{2}$$

Observe that the dual problem (2) estimates the covariance matrix while the primal problem estimates its inverse. We also observe that the diagonal elements of the solution are $\Sigma_{kk} = S_{kk} + \lambda$ for all k .

The following theorem shows that adding the ℓ_1 -norm penalty regularizes the solution.

Theorem 1 *For every $\lambda > 0$, the optimal solution to (1) is unique, with bounded eigenvalues:*

$$\frac{p}{\lambda} \geq \|\hat{\Sigma}^{-1}\|_2 \geq (\|S\|_2 + \lambda p)^{-1}.$$

Here, $\|A\|_2$ denotes the maximum eigenvalue of a symmetric matrix A . The proof is contained in the appendix.

The dual problem (2) is smooth and convex, albeit with $p(p+1)/2$ variables. For small values of p , the problem can be solved by existing interior point methods (e.g., Vandenberghe et al., 1998). The complexity to compute an ε -suboptimal solution using such second-order methods, however, is $O(p^6 \log(1/\varepsilon))$, making them infeasible when p is larger than the tens.

A related problem, solved by Dahl et al. (Revised 2007), is to compute a maximum likelihood estimate of the covariance matrix when the sparsity structure of the inverse is known in advance. This is accomplished by adding constraints to (1) of the form: $X_{ij} = 0$ for all pairs (i, j) in some specified set. Our constraint set is unbounded as we hope to uncover the sparsity structure automatically, starting with a dense second moment matrix S .

2.3 Choice of Penalty Parameter

Consider the true, unknown graphical model for a given distribution. This graph has p nodes, and an edge between nodes k and j is missing if variables k and j are independent conditional on the rest of the variables. For a given node k , let C_k denote its connectivity component: the set of all nodes that are connected to node k through some chain of edges. In particular, if node $j \notin C_k$, then variables j and k are independent.

Meinshausen and Bühlmann (2006) show that the Lasso, when used to estimate the set of neighbors for each node, yields an estimate of the sparsity pattern of the graph in a way that is asymptotically consistent. Consistency, they show, hinges on the choice of the regularization parameter. In addition to asymptotic results, they prove a finite sample result: by choosing the regularization parameter in a particular way, the probability that two distinct connectivity components are falsely joined in the estimated graph can be controlled.

Following the methods used by Meinshausen and Bühlmann (2006), we can derive a corresponding choice for the regularization parameter. Let \hat{C}_k^λ denote our estimate of the connectivity component of node k . In the context of our optimization problem, this corresponds to the entries of row k in $\hat{\Sigma}$ that are nonzero.

Let α be a given level in $[0, 1]$. Consider the following choice for the penalty parameter in (1):

$$\lambda(\alpha) := (\max_{i>j} \hat{\sigma}_i \hat{\sigma}_j) \frac{t_{n-2}(\alpha/2p^2)}{\sqrt{n-2 + t_{n-2}^2(\alpha/2p^2)}} \tag{3}$$

where $t_{n-2}(\alpha)$ denotes the $(100 - \alpha)\%$ point of the Student's t-distribution for $n - 2$ degrees of freedom, and $\hat{\sigma}_i$ is the empirical variance of variable i . Then we can prove the following theorem:

Theorem 2 Using $\lambda(\alpha)$ the penalty parameter in (1), for any fixed level α ,

$$P(\exists k \in \{1, \dots, p\} : \hat{C}_k^\lambda \not\subseteq C_k) \leq \alpha.$$

Observe that, for a fixed problem size p , as the number of samples n increases to infinity, the penalty parameter $\lambda(\alpha)$ decreases to zero. Thus, asymptotically we recover the classical maximum likelihood estimate, S , which in turn converges in probability to the true covariance Σ .

Meinshausen and Bühlmann (2006) prove that Lasso recovers the underlying sparsity pattern consistently, in a scheme where the number of variables is allowed to grow with the number of samples: $p \in O(n^\gamma)$. Although such an analysis is beyond the scope of this paper, a similar result seems to hold here provided we assume certain conditions on the true covariance matrix.

From the point of view of estimating the covariance matrix itself, when the number of variables is allowed to grow with the number of samples, it may be better to use estimates such as those described by Bickel and Levina (2008), which are shown to be consistent in the operator norm as long as $(\log p)^2/n \rightarrow 0$.

3. Block Coordinate Descent Algorithm

In this section we present an algorithm for solving (2) that uses block coordinate descent.

3.1 Algorithm Description

We begin by detailing the algorithm. For any symmetric matrix A , let $A_{\setminus k \setminus j}$ denote the matrix produced by removing column k and row j . Let A_j denote column j with the diagonal element A_{jj} removed. The plan is to optimize over one row and column of the variable matrix W at a time, and to repeatedly sweep through all columns until we achieve convergence.

Initialize: $W^{(0)} := S + \lambda I$

For $k \geq 0$

1. **For** $j = 1, \dots, p$

(a) Let $W^{(j-1)}$ denote the current iterate. Solve the quadratic program

$$\hat{y} := \arg \min_y \{y^T (W_{\setminus j \setminus j}^{(j-1)})^{-1} y : \|y - S_j\|_\infty \leq \lambda\}. \quad (4)$$

(b) Update rule: $W^{(j)}$ is $W^{(j-1)}$ with column/row W_j replaced by \hat{y} .

2. Let $\hat{W}^{(0)} := W^{(p)}$.

3. After each sweep through all columns, check the convergence condition. Convergence occurs when

$$\text{trace}((\hat{W}^{(0)})^{-1} S) - p + \lambda \|(\hat{W}^{(0)})^{-1}\|_1 \leq \epsilon.$$

3.2 Convergence and Property of Solution

Using Schur complements, we can prove convergence:

Theorem 3 *The block coordinate descent algorithm described above converges, achieving an ε -suboptimal solution to (2). In particular, the iterates produced by the algorithm are strictly positive definite: each time we sweep through the columns, $W^{(j)} \succ 0$ for all j .*

The proof of Theorem 3 sheds some interesting light on the solution to problem (1). In particular, we can use this method to show that the solution has the following property:

Theorem 4 *Fix any $k \in \{1, \dots, p\}$. If $\lambda \geq |S_{kj}|$ for all $j \neq k$, then column and row k of the solution $\hat{\Sigma}$ to (2) are zero, excluding the diagonal element.*

This means that, for a given second moment matrix S , if λ is chosen such that the condition in Theorem 4 is met for some column k , then the sparse maximum likelihood method estimates variable k to be independent of all other variables in the distribution. In particular, Theorem 4 implies that if $\lambda \geq |S_{kj}|$ for all $k > j$, then (1) estimates all variables in the distribution to be pairwise independent.

Using the work of Luo and Tseng (1992), it may be possible to show that the local convergence rate of this method is at least linear. In practice we have found that a small number of sweeps through all columns, independent of problem size p , is sufficient to achieve convergence. In each iteration, we must solve one quadratic program (QP). In a standard primal-dual interior point method for solving QPs, the computational cost is dominated by the cost of finding a search direction, which involves inverting matrices of size p . The cost of each iteration is therefore $O(p^3)$. For a fixed number of K sweeps, the cost of the method is $O(Kp^4)$.

3.3 Interpretation as Iterative Penalized Regression

The dual of (4) is

$$\min_x x^T W_{\setminus j \setminus j}^{(j-1)} x - S_j^T x + \lambda \|x\|_1. \quad (5)$$

Strong duality obtains so that problems (5) and (4) are equivalent. If we let Q denote the square root of $W_{\setminus j \setminus j}^{(j-1)}$, and $b := \frac{1}{2} Q^{-1} S_j$, then we can write (5) as

$$\min_x \|Qx - b\|_2^2 + \lambda \|x\|_1. \quad (6)$$

The problem (6) is a penalized least-squares problems, known as the Lasso. If $W_{\setminus j \setminus j}^{(j-1)}$ were the j -th principal minor of the sample covariance S , then (6) would be equivalent to a penalized regression of variable j against all others. Thus, the approach is reminiscent of the approach explored by Meinshausen and Bühlmann (2006), but there are two differences. First, we begin by adding a multiple of the identity to the sample covariance matrix, so we can guarantee that each penalized regression problem has a unique solution, without requiring the data or solutions to satisfy any special conditions (see Osborne et al., 2000, for conditions that guarantee uniqueness of the Lasso). Second, and more importantly, we update the problem data after each regression: except at the very first update, $W_{\setminus j \setminus j}^{(j-1)}$ is never a minor of S . In this sense, the coordinate descent method can be interpreted as a sequence of iterative Lasso problems. The approach is pursued to great advantage by Friedman et al. (2007), who also provide a conceptual link to the approach of Meinshausen and Bühlmann (2006).

4. Nesterov's First Order Method

In this section we apply the recent results due to Nesterov (2005) to obtain a first order algorithm for solving (1) with lower memory requirements and a rigorous complexity estimate with a better dependence on problem size than those offered by interior point methods. Our purpose is not to obtain another algorithm, as we have found that the block coordinate descent is fairly efficient; rather, we seek to use Nesterov's formalism to derive a rigorous complexity estimate for the problem, improved over that offered by interior-point methods. In particular, although we have found in practice that a small, fixed number K of sweeps through all columns is sufficient to achieve convergence using the block coordinate descent method, we have not been able to compute a bound on K . In what follows, we compute a guaranteed theoretical upper bound on the complexity of solving (1).

As we will see, Nesterov's framework allows us to obtain an algorithm that has a complexity of $O(p^{4.5}/\varepsilon)$, where $\varepsilon > 0$ is the desired accuracy on the objective of problem (1). This is in contrast to the complexity of interior-point methods, $O(p^6 \log(1/\varepsilon))$. Thus, Nesterov's method provides a much better dependence on problem size and lower memory requirements at the expense of a degraded dependence on accuracy.

4.1 Idea of Nesterov's Method

Nesterov's method applies to a class of non-smooth, convex optimization problems of the form

$$\min_x \{f(x) : x \in Q_1\} \tag{7}$$

where the objective function can be written as

$$f(x) = \hat{f}(x) + \max_u \{ \langle Ax, u \rangle_2 : u \in Q_2 \}.$$

Here, Q_1 and Q_2 are bounded, closed, convex sets, $\hat{f}(x)$ is differentiable (with a Lipschitz-continuous gradient) and convex on Q_1 , and A is a linear operator. The challenge is to write our problem in the appropriate form and choose associated functions and parameters in such a way as to obtain the best possible complexity estimate, by applying general results obtained by Nesterov (2005).

Observe that we can write (1) in the form (7) if we impose bounds on the eigenvalues of the solution, X . To this end, we let

$$\begin{aligned} Q_1 &:= \{x : aI \preceq X \preceq bI\}, \\ Q_2 &:= \{u : \|u\|_\infty \leq \lambda\} \end{aligned}$$

where the constants a, b are given such that $b > a > 0$. By Theorem 1, we know that such bounds always exist. We also define $\hat{f}(x) := -\log \det x + \langle S, x \rangle$, and $A := I$.

To Q_1 and Q_2 , we associate norms and continuous, strongly convex functions, called prox-functions, $d_1(x)$ and $d_2(u)$. For Q_1 we choose the Frobenius norm, and a prox-function $d_1(x) = -\log \det x + p \log b$. For Q_2 , we choose the Frobenius norm again, and a prox-function $d_2(x) = \|u\|_F^2/2$.

The method applies a smoothing technique to the non-smooth problem (7), which replaces the objective of the original problem, $f(x)$, by a penalized function involving the prox-function $d_2(u)$:

$$\tilde{f}(x) = \hat{f}(x) + \max_{u \in Q_2} \{ \langle Ax, u \rangle - \mu d_2(u) \}. \tag{8}$$

The above function turns out to be a smooth uniform approximation to f everywhere. It is differentiable, convex on Q_1 , and has a Lipschitz-continuous gradient, with a constant L that can be computed as detailed below. A specific gradient scheme is then applied to this smooth approximation, with convergence rate $O(L/\varepsilon)$.

4.2 Algorithm and Complexity Estimate

To detail the algorithm and compute the complexity, we must first calculate some parameters corresponding to our definitions and choices above. First, the strong convexity parameter for $d_1(x)$ on Q_1 is $\sigma_1 = 1/b^2$, in the sense that

$$\nabla^2 d_1(X)[H, H] = \text{trace}(X^{-1}HX^{-1}H) \geq b^{-2}\|H\|_F^2$$

for every symmetric H . Furthermore, the center of the set Q_1 is $x_0 := \arg \min_{x \in Q_1} d_1(x) = bI$, and satisfies $d_1(x_0) = 0$. With our choice, we have $D_1 := \max_{x \in Q_1} d_1(x) = p \log(b/a)$.

Similarly, the strong convexity parameter for $d_2(u)$ on Q_2 is $\sigma_2 := 1$, and we have

$$D_2 := \max_{u \in Q_2} d_2(U) = p^2/2.$$

With this choice, the center of the set Q_2 is $u_0 := \arg \min_{u \in Q_2} d_2(u) = 0$.

For a desired accuracy ε , we set the smoothness parameter $\mu := \varepsilon/2D_2$, and set $x_0 = bI$. The algorithm proceeds as follows:

For $k \geq 0$ **do**

1. Compute $\nabla \tilde{f}(x_k) = -x^{-1} + S + u^*(x_k)$, where $u^*(x)$ solves (8).
2. Find $y_k = \arg \min_y \{ \langle \nabla \tilde{f}(x_k), y - x_k \rangle + \frac{1}{2}L(\varepsilon)\|y - x_k\|_F^2 : y \in Q_1 \}$.
3. Find $z_k = \arg \min_x \{ \frac{L(\varepsilon)}{\sigma_1}d_1(X) + \sum_{i=0}^k \frac{i+1}{2} \langle \nabla \tilde{f}(x_i), x - x_i \rangle : x \in Q_1 \}$.
4. Update $x_k = \frac{2}{k+3}z_k + \frac{k+1}{k+3}y_k$.

In our case, the Lipschitz constant for the gradient of our smooth approximation to the objective function is

$$L(\varepsilon) := M + D_2\|A\|^2/(2\sigma_2\varepsilon)$$

where $M := 1/a^2$ is the Lipschitz constant for the gradient of \tilde{f} , and the norm $\|A\|$ is induced by the Frobenius norm, and is equal to λ .

The algorithm is guaranteed to produce an ε -suboptimal solution after a number of steps not exceeding

$$\begin{aligned} N(\varepsilon) &:= \\ &4\|A\| \sqrt{\frac{D_1 D_2}{\sigma_1 \sigma_2}} \cdot \frac{1}{\varepsilon} + \sqrt{\frac{M D_1}{\sigma_1 \varepsilon}} \\ &= (\kappa \sqrt{\log \kappa})(4p^{1.5} a \lambda / \sqrt{2} + \sqrt{\varepsilon p}) / \varepsilon. \end{aligned} \tag{9}$$

where $\kappa = b/a$ is a bound on the condition number of the solution.

Now we are ready to estimate the complexity of the algorithm. For Step 1, the gradient of the smooth approximation is computed in closed form by taking the inverse of x . Step 2 essentially

amounts to projecting on Q_1 , and requires that we solve an eigenvalue problem. The same is true for Step 3. In fact, each iteration costs $O(p^3)$. The number of iterations necessary to achieve an objective with absolute accuracy less than ϵ is given in (9) by $N(\epsilon) = O(p^{1.5}/\epsilon)$. Thus, if the condition number κ is fixed in advance, the complexity of the algorithm is $O(p^{4.5}/\epsilon)$.

5. Binary Variables: Approximate Sparse Maximum Likelihood Estimation

In this section, we consider the problem of estimating an undirected graphical model for multivariate binary data. Recently, Wainwright et al. (2006) applied an ℓ_1 -norm penalty to the logistic regression problem to obtain a binary version of the high-dimensional consistency results of Meinshausen and Bühlmann (2006). We apply the log determinant relaxation of Wainwright and Jordan (2006) to formulate an approximate sparse maximum likelihood (ASML) problem for estimating the parameters in a multivariate binary distribution. We show that the resulting problem is the same as the Gaussian sparse maximum likelihood (SML) problem, and that we can therefore apply our previously-developed algorithms to sparse model selection in a binary setting.

Consider a distribution made up of p binary random variables. Using n data samples, we wish to estimate the structure of the distribution. An Ising model for this distribution is

$$p(x; \theta) = \exp\left\{\sum_{i=1}^p \theta_i x_i + \sum_{i=1}^{p-1} \sum_{j=i+1}^p \theta_{ij} x_i x_j - A(\theta)\right\} \tag{10}$$

where

$$A(\theta) = \log \sum_{x \in \mathcal{X}^p} \exp\left\{\sum_{i=1}^p \theta_i x_i + \sum_{i=1}^{p-1} \sum_{j=i+1}^p \theta_{ij} x_i x_j\right\} \tag{11}$$

is the log partition function.

The sparse maximum likelihood problem in this case is to maximize (10) with an added ℓ_1 -norm penalty on terms θ_{kj} . Specifically, in the undirected graphical model, an edge between nodes k and j is missing if $\theta_{kj} = 0$.

A well-known difficulty is that the log partition function has too many terms in its outer sum to compute. However, if we use the log determinant relaxation for the log partition function developed by Wainwright and Jordan (2006), we can obtain an approximate sparse maximum likelihood (ASML) estimate.

In their paper, Wainwright and Jordan (2006) consider the problem of computing marginal probabilities over subsets of nodes in a graphical model for a discrete-valued Markov random field. This problem is closely related to the one we examine here, and exactly solving the problem for general graphs is intractable. Wainwright and Jordan (2006) propose a relaxation by using a Gaussian bound on the log partition function (11) and a semidefinite outer bound on the polytope of marginal probabilities. In the next section, we use their results to obtain an approximate estimate of undirected graphical model for binary variables, and we show that, when using the simplest semidefinite outer bound on the constraint set, we obtain a problem that is almost exactly the same as that considered in previous sections, for the Gaussian case. In particular, this means that we can reuse the block coordinate descent or Nesterov algorithms to approximately estimate graphical models in the case of binary data.

5.1 Problem Formulation

Let's begin with some notation. Letting $d := p(p + 1)/2$, define the map $R : \mathbf{R}^d \rightarrow S^{p+1}$ as follows:

$$R(\theta) = \begin{pmatrix} 0 & \theta_1 & \theta_2 & \dots & \theta_p \\ \theta_1 & 0 & \theta_{12} & \dots & \theta_{1p} \\ \vdots & & & & \\ \theta_p & \theta_{1p} & \theta_{2p} & \dots & 0 \end{pmatrix}$$

Suppose that our n samples are $z^{(1)}, \dots, z^{(n)} \in \{-1, +1\}^p$. Let \bar{z}_i and \bar{z}_{ij} denote sample mean and second moments. The sparse maximum likelihood problem is

$$\hat{\theta}_{\text{exact}} := \arg \max_{\theta} \frac{1}{2} \langle R(\theta), R(\bar{z}) \rangle - A(\theta) - \lambda \|\theta\|_1. \tag{12}$$

Finally define the constant vector $m = (1, \frac{4}{3}, \dots, \frac{4}{3}) \in \mathbf{R}^{p+1}$. Wainwright and Jordan (2006) give an upper bound on the log partition function as the solution to the following variational problem:

$$\begin{aligned} A(\theta) &\leq \max_{\mu} \frac{1}{2} \log \det(R(\mu) + \text{diag}(m)) + \langle \theta, \mu \rangle \\ &= \frac{1}{2} \cdot \max_{\mu} \log \det(R(\mu) + \text{diag}(m)) + \langle R(\theta), R(\mu) \rangle. \end{aligned} \tag{13}$$

If we use the bound (13) in our sparse maximum likelihood problem (12), we won't be able to extract an optimizing argument $\hat{\theta}$. Our first step, therefore, will be to rewrite the bound in a form that will allow this.

Lemma 5 *We can rewrite the bound (13) as*

$$A(\theta) \leq \frac{p}{2} \log\left(\frac{e\pi}{2}\right) - \frac{1}{2}(p + 1) - \frac{1}{2} \cdot \left\{ \max_{\mathbf{v}} \mathbf{v}^T m + \log \det(-(R(\theta) + \text{diag}(\mathbf{v}))) \right\}. \tag{14}$$

Using this version of the bound (13), we have the following theorem.

Theorem 6 *Using the upper bound on the log partition function given in (14), the approximate sparse maximum likelihood problem has the following solution:*

$$\begin{aligned} \hat{\theta}_k &= \bar{\mu}_k \\ \hat{\theta}_{kj} &= -(\hat{\Gamma})_{kj}^{-1} \end{aligned} \tag{15}$$

where the matrix $\hat{\Gamma}$ is the solution to the following problem, related to (2):

$$\hat{\Gamma} := \arg \max \{ \log \det W : W_{kk} = S_{kk} + \frac{1}{3}, |W_{kj} - S_{kj}| \leq \lambda \}. \tag{16}$$

Here, S is defined as before:

$$S = \frac{1}{n} \sum_{k=1}^n (z^{(k)} - \bar{\mu})(z^{(k)} - \bar{\mu})^T$$

where $\bar{\mu}$ is the vector of sample means \bar{z}_i .

In particular, this means that we can reuse the algorithms developed in Sections 3 and 4 for problems with binary variables. The relaxation (13) is the simplest one offered by Wainwright and Jordan (2006). The relaxation can be tightened by adding linear constraints on the variable μ .

5.2 Penalty Parameter Choice for Binary Variables

For the choice of the penalty parameter λ , we can derive a formula analogous to (3). Consider the choice

$$\lambda(\alpha)_{\text{bin}} := \frac{(\chi^2(\alpha/2p^2, 1))^{\frac{1}{2}}}{(\min_{i>j} \hat{\sigma}_i \hat{\sigma}_j) \sqrt{n}} \quad (17)$$

where $\chi^2(\alpha, 1)$ is the $(100 - \alpha)\%$ point of the chi-square distribution for one degree of freedom. Since our variables take on values in $\{-1, 1\}$, the empirical variances are of the form:

$$\hat{\sigma}_i^2 = 1 - \bar{\mu}_i^2.$$

Using (17), we have the following binary version of Theorem 2:

Theorem 7 *With (17) chosen as the penalty parameter in the approximate sparse maximum likelihood problem, for a fixed level α ,*

$$P(\exists k \in \{1, \dots, p\} : \hat{C}_k^\lambda \not\subseteq C_k) \leq \alpha.$$

6. Numerical Results

In this section we present the results of some numerical experiments, both on synthetic and real data. The synthetic experiments were performed on continuous data, and tests only the original problem formulation. Both continuous and discrete real data sets are used, however, testing the Gaussian and binary formulations respectively.

6.1 Synthetic Experiments

Synthetic experiments require that we generate underlying sparse inverse covariance matrices. To this end, we first randomly choose a diagonal matrix with positive diagonal entries. A given number of nonzeros are inserted in the matrix at random locations symmetrically. Positive definiteness is ensured by adding a multiple of the identity to the matrix if needed. The multiple is chosen to be only as large as necessary for inversion with no errors.

6.2 Sparsity and Thresholding

A very simple approach to obtaining a sparse estimate of the inverse covariance matrix would be to apply a threshold to the inverse empirical covariance matrix, S^{-1} . However, even when S is easily invertible, it can be difficult to select a threshold level. We solved a synthetic problem of size $p = 100$ where the true concentration matrix density was set to $\delta = 0.1$. Drawing $n = 200$ samples, we plot in Figure 1 the sorted absolute value elements of S^{-1} on the left and $\hat{\Sigma}^{-1}$, the solution to (1), on the right.

It is clearly easier to choose a threshold level for the sparse maximum likelihood estimate. Applying a threshold to either S^{-1} or $\hat{\Sigma}^{-1}$ would decrease the log likelihood of the estimate by an unknown amount. One way to compute the largest possible threshold t that preserves positive definiteness in S can be computed by solving the minimization problem:

$$t \leq \min_{\|v\|_1=1} v^T S^{-1} v.$$

The condition (6.2) is equivalent to the condition that $S^{-1} + L \succ 0$ for all matrices L such that $\|L\|_\infty \leq t$.

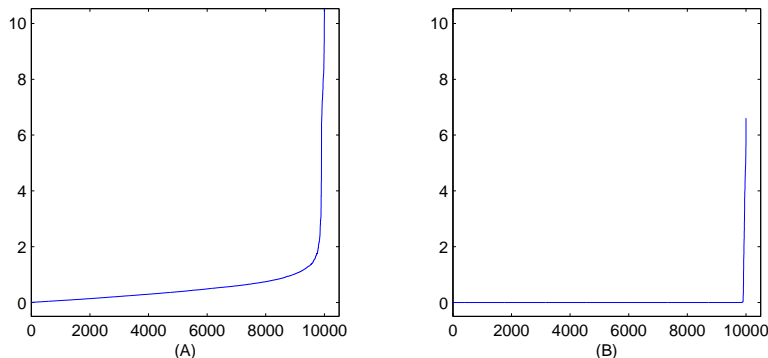


Figure 1: Sorted absolute value of elements of (A) S^{-1} and (B) $\hat{\Sigma}^{-1}$. The solution $\hat{\Sigma}^{-1}$ to (1) is un-thresholded.

6.3 Recovering Structure

We begin with a small experiment to test the ability of the method to recover the sparse structure of an underlying covariance matrix. Figure 2 (A) shows a sparse inverse covariance matrix of size $p = 30$. Figure 2 (B) displays a corresponding S^{-1} , using $n = 60$ samples. Figure 2 (C) displays the solution to (1) for $\lambda = 0.1$. The value of the penalty parameter here is chosen arbitrarily, and the solution is not thresholded. Nevertheless, we can still pick out features that were present in the true underlying inverse covariance matrix.

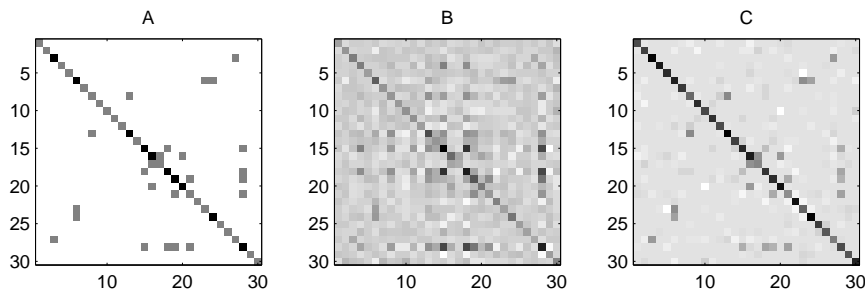


Figure 2: Recovering the sparsity pattern. We plot (A) the original inverse covariance matrix Σ^{-1} , (B) the noisy sample inverse S^{-1} , and (C) the solution to problem (1) for $\lambda = 0.1$.

Using the same underlying inverse covariance matrix, we repeat the experiment using smaller sample sizes. We solve (1) for $n = 30$ and $n = 20$ using the same arbitrarily chosen penalty parameter value $\lambda = 0.1$, and display the solutions in Figure 3. As expected, our ability to pick out features of the true inverse covariance matrix diminishes with the number of samples. This is an added reason to choose a larger value of λ when we have fewer samples, as in (3).

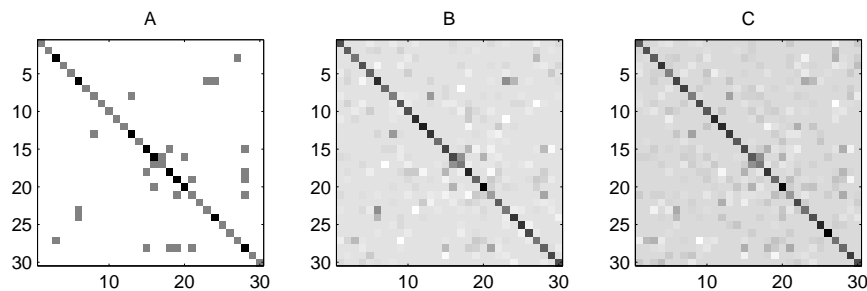


Figure 3: Recovering the sparsity pattern for small sample size. We plot (A) the original inverse covariance matrix Σ^{-1} , (B) the solution to problem (1) for $n = 30$ and (C) the solution for $n = 20$. A penalty parameter of $\lambda = 0.1$ is used for (B) and (C).

6.4 CPU Times Versus Problem Size

For a sense of the practical performance of the block coordinate descent method, we implemented it in Matlab, using Mosek to solve the quadratic program at each step. For problem sizes ranging from $p = 400$ to $p = 1000$, we solved the problem using $n = 2p$ samples. In all cases, one sweep through all the columns was sufficient to achieve an absolute accuracy of $\epsilon = 1e - 7$. The Nesterov algorithm, implemented in C, was found to have a comparable computation time provided one chooses a large value for the desired accuracy. Since the Nesterov algorithm aims for a lower accuracy and requires the setting of bounds a and b on the eigenvalues of the solution, it was not included in this experiment.

In Figure 4 we plot the average CPU time to achieve convergence, along with CPU times for the Lasso for comparison. CPU times were computed using an AMD Athlon 64 2.20Ghz processor with 1.96GB of RAM. Using this very simple implementation, the block coordinate descent method solves a problem of size $p = 1000$ in about an hour and a half. The computation time could possibly be cut down by using a fast Lasso implementation to solve the optimization problem, as described in Section 3.3, and by using a suitable method to compute the square root matrix Q from (6).

6.5 Path Following Experiments

Figure 6.5 shows two path following examples. We solve two randomly generated problems of size $p = 5$ and $n = 100$ samples. The red lines correspond to elements of the solution that are zero in the true underlying inverse covariance matrix. The blue lines correspond to true nonzeros. The vertical lines mark ranges of λ for which we recover the correct sparsity pattern exactly. Note that, by Theorem 4, for λ values greater than those shown, the solution will be diagonal.

On a related note, we observe that (1) also works well in recovering the sparsity pattern of a matrix masked by noise. The following experiment illustrates this observation. We generate a sparse inverse covariance matrix of size $p = 50$ as described above. Then, instead of using an empirical covariance S as input to (1), we use $S = (\Sigma^{-1} + V)^{-1}$, where V is a randomly generated uniform noise of size $\sigma = 0.1$. We then solve (1) for various values of the penalty parameter λ .

In Figure 6, for each value of λ shown, we randomly selected 10 sample covariance matrices S of size $p = 50$ and computed the number of misclassified zeros and nonzero elements in the solution

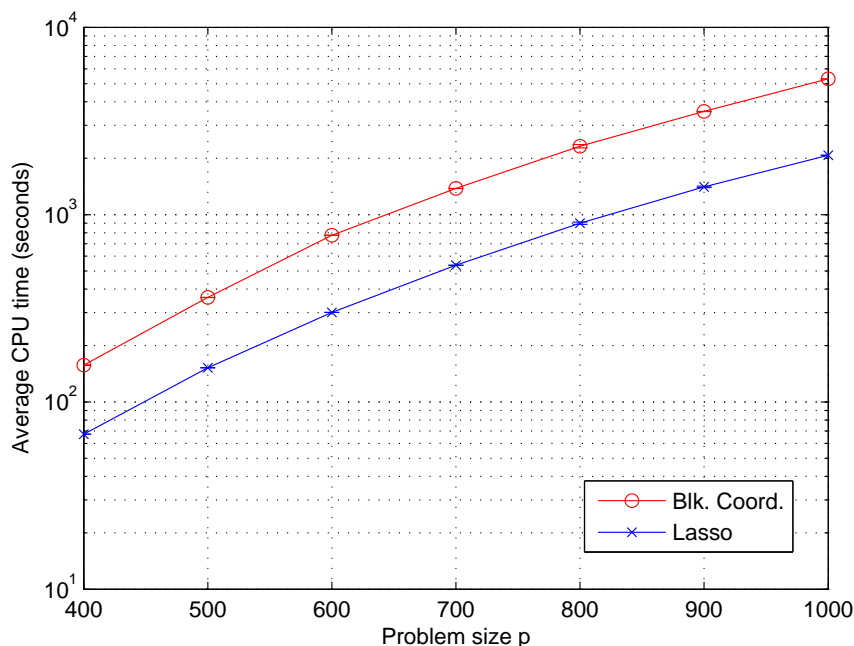


Figure 4: Average CPU times vs. problem size using block coordinate descent. We plot the average CPU time (in seconds) versus problem size. The standard deviations for the block coordinate descent method are: 0.6077, 0.2652, 0.1989, 1.0165, 48.0833, 2.7732, and 6.2314 seconds respectively. For the Lasso, the standard deviations are: 0.0221, 0.3646, 0.1878, 0.5966, 12.7279, 13.1257, and 0.6187 seconds respectively.

to (1). We plot the average percentage of errors (number of misclassified zeros plus misclassified nonzeros divided by p^2), as well as error bars corresponding to one standard deviation. As shown, the error rate is nearly zero on average when the penalty is set to equal the noise level σ .

6.6 Estimating the Stability of the Solution Using Cross-validation

Viewing our estimator as a machine that classifies elements of the concentration matrix as either zero or nonzero, we next turn to the question of the stability of the classification. Classifier instability is defined here as the probability that the classification of an arbitrary element of the matrix is changed by some small disturbance in the data. In other words, once we obtain a concentration graph by treating our estimated matrix $\hat{\Sigma}$ as an adjacency matrix, we would like to measure the stability of our results.

To empirically estimate the instability of our classifier, we can use the following simple procedure. First, we use all the available data to obtain an estimate of the concentration matrix $\hat{\Sigma}_0^{-1}$. Then we perform K -fold cross validation: randomly dividing the available data into K subsamples, we compute K different estimates $\hat{\Sigma}_i^{-1}$ by successively leaving out one subsample each time. After each estimate $\hat{\Sigma}_i^{-1}$ is computed, we count the number of matrix entries with classifications that are

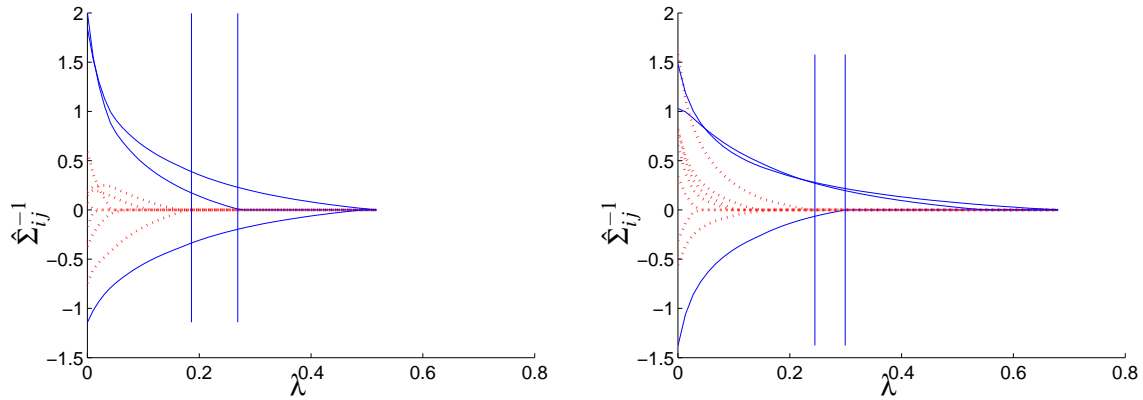


Figure 5: Path following: elements of solution to (1) as λ increases. Dashed lines correspond to elements that are zero in the true inverse covariance matrix; solid lines correspond to true nonzeros. Vertical lines mark a range of λ values using which we recover the sparsity pattern exactly.

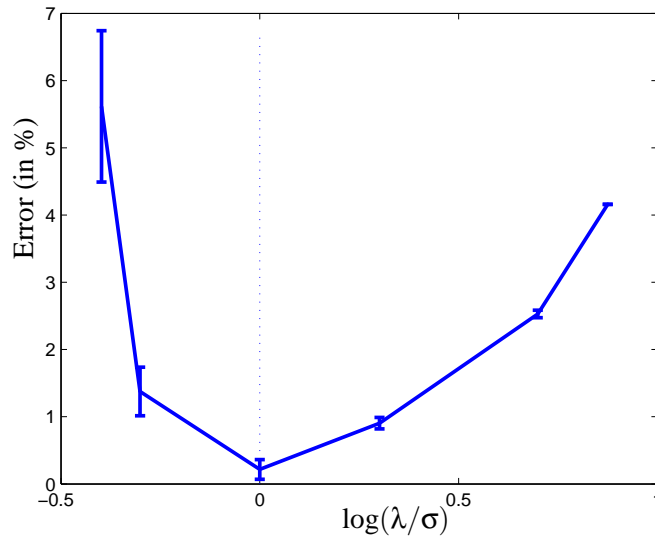


Figure 6: Recovering sparsity pattern in a matrix with added uniform noise of size $\sigma = 0.1$. We plot the average percentage of misclassified entries as a function of $\log(\lambda/\sigma)$.

different from those of $\hat{\Sigma}_0^{-1}$. Our estimate of the instability is then the average number of entries, taken over the K estimates, that were classified differently from those of $\hat{\Sigma}_0^{-1}$.

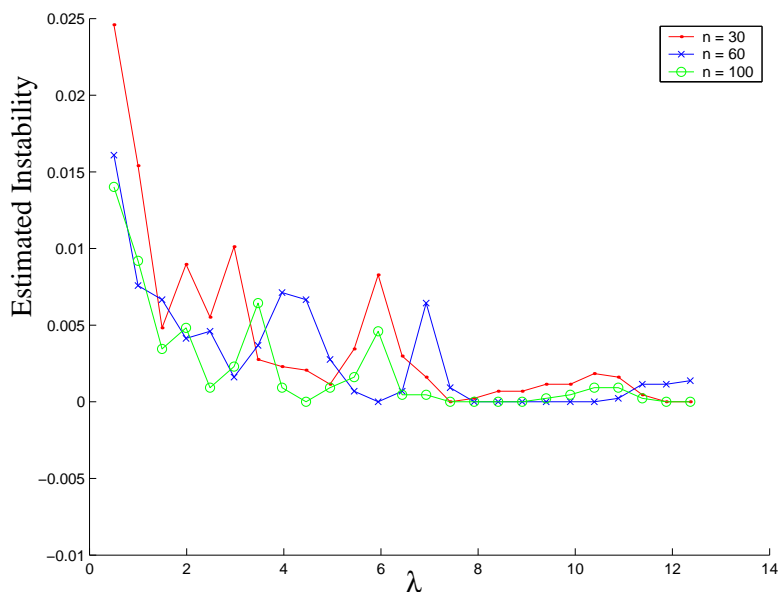


Figure 7: Estimating stability of the solution using cross validation. We plot estimated instability, the probability that the classification of an arbitrary element is changed by a small disturbance in the data, for a range of values of λ using data sets of size $n = 30, 60$, and 100 samples for a problem with $p = 30$ variables.

Below we show the results of some experiments using synthetic data. First, we randomly generated an underlying concentration matrix of size $p = 30$. Let $\delta = 0.25$ denote the fraction of nonzero elements in Σ^{-1} . Then, from the corresponding covariance matrix Σ , we generated three data sets of sizes $n = 30, 60$, and 100.

In Figure 7 we plot estimated instability versus a range of values for the penalty parameter λ , using 10-fold cross validation. We applied Theorem 4 to compute a maximum value of λ ; beyond this value, the estimated graph is empty for all three data sets. As shown, even for a small number of samples, the estimated graph is fairly stable for all values of λ .

We can also see how our estimate of the instability changes as we divide the available data into a greater number of subsamples. In Figure 8 we repeat the experiment described above, this time fixing $\lambda = 0.2$ and instead varying the number of folds K in the cross validation. In these examples, for each value of n , dividing the available data into more subsamples for cross validation decreases our estimate of the instability.

6.7 Performance as a Binary Classifier

In this section we numerically examine the ability of the sparse maximum likelihood (SML) method to correctly classify elements of the inverse covariance matrix as zero or nonzero. For comparison, we will use the Lasso estimate of Meinshausen and Bühlmann (2006), which has been shown to perform extremely well. The Lasso regresses each variable against all others one at a time. Upon obtaining a solution $\theta^{(k)}$ for each variable k , one can estimate sparsity in one of two ways: either by

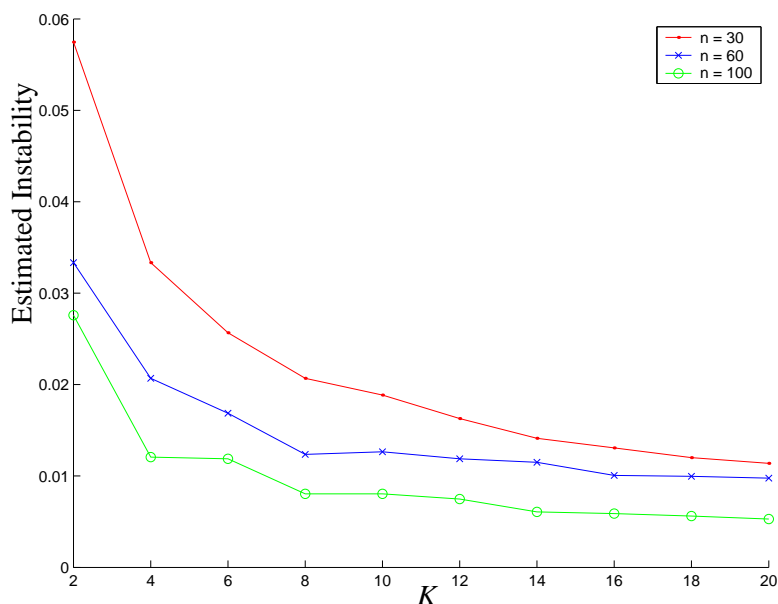


Figure 8: Estimating instability of the solution using cross validation. We plot the estimate of the instability obtained using K samples, for various values of K . Once again, we use data sets of size $n = 30, 60$, and 100 samples for a problem with $p = 30$ variables.

declaring an element $\hat{\Sigma}_{ij}$ nonzero if both $\theta_i^{(k)} \neq 0$ and $\theta_j^{(k)} \neq 0$ (Lasso-AND) or, less conservatively, if either of those quantities is nonzero (Lasso-OR).

As noted previously, Meinshausen and Bühlmann (2006) have also derived a formula for choosing their penalty parameter. Both the SML and Lasso penalty parameter formulas depend on a chosen level α , which is a bound on the same error probability for each method. For these experiments, we set $\alpha = 0.05$.

In the following experiments, we fixed the problem size p at 30 and generated sparse underlying inverse covariance matrices as described above. We varied the number of samples n from 10 to 310. For each value of n shown, we ran 30 trials in which we estimated the sparsity pattern of the inverse covariance matrix using the SML, Lasso-OR, and Lasso-AND methods. We then recorded the average number of nonzeros estimated by each method, and the average number of entries correctly identified as nonzero (true positives).

We show two sets of plots. Figure 6.7 corresponds to experiments where the true density was set to be low, $\delta = 0.05$. We plot the power (proportion of correctly identified nonzeros), positive predictive value (proportion of estimated nonzeros that are correct), and the density estimated by each method. Figure 6.7 corresponds to experiments where the true density was set to be high, $\delta = 0.40$, and we plot the same three quantities.

Meinshausen and Bühlmann (2006) report that, asymptotically, Lasso-AND and Lasso-OR yield the same estimate of the sparsity pattern of the inverse covariance matrix. At a finite number of samples, the SML method seems to fall in between the two methods in terms of power, positive predictive value, and the density of the estimate. It typically offers, on average, the lowest total

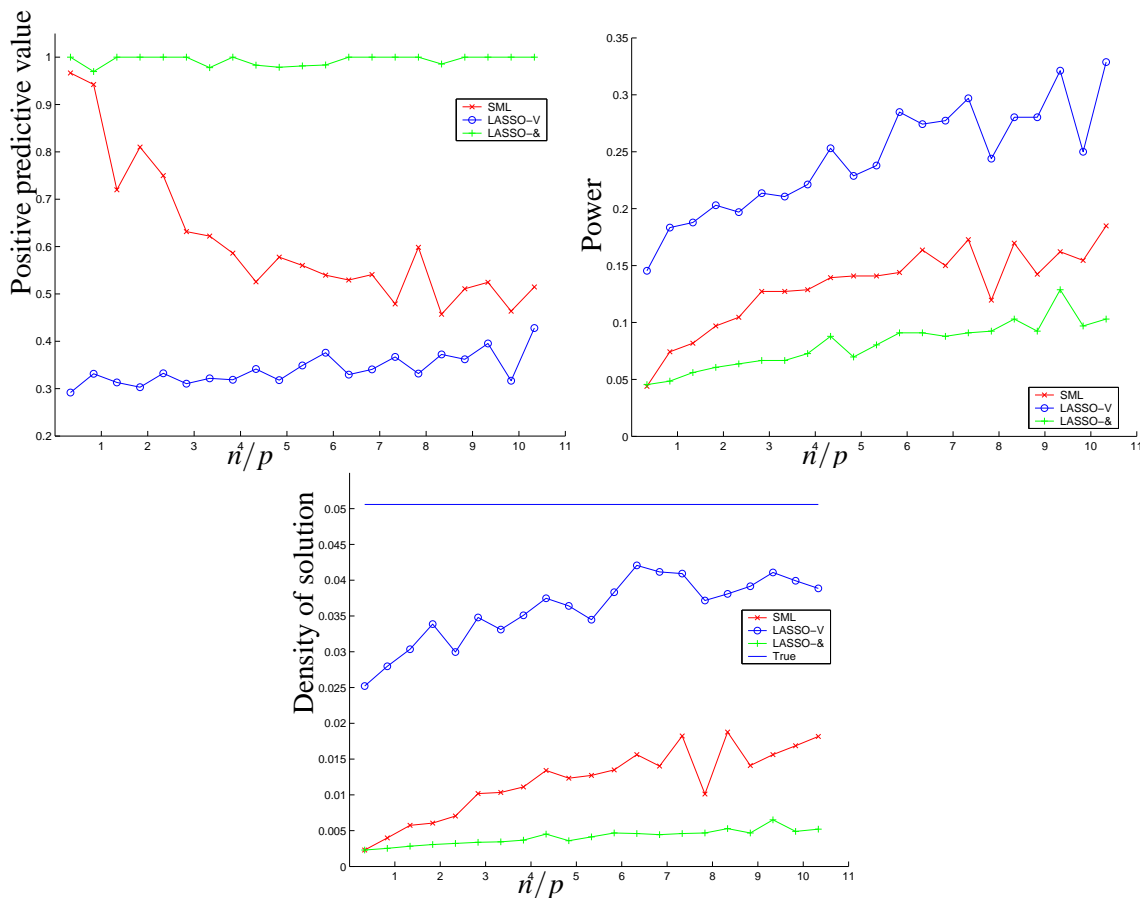


Figure 9: Classifying zeros and nonzeros for a true density of $\delta = 0.05$. We plot the positive predictive value, the power, and the estimated density using SML, Lasso-OR and Lasso-AND.

number of errors, tied with either Lasso-AND or Lasso-OR. Among the two Lasso methods, it would seem that if the true density is very low, it is slightly better to use the more conservative Lasso-AND. If the density is higher, it may be better to use Lasso-OR. When the true density is unknown, we can achieve an accuracy comparable to the better choice among the Lasso methods by computing the SML estimate. Figure 11 shows one example of sparsity pattern recovery when the true density is low.

The Lasso and SML methods have a comparable computational complexity. However, unlike the Lasso, the SML method is not parallelizable. Parallelization would render the Lasso a more computationally attractive choice, since each variable can be regressed against all other variables separately, at an individual cost of $O(p^3)$. In exchange, SML can offer a more accurate estimate of the sparsity pattern, as well as a well-conditioned estimate of the covariance matrix itself.

6.8 Recovering Structure from Binary Data

Next we turn our attention to the binary version of the algorithm, as described in Section 5. For $p = 100$ variables, we randomly generated Ising models (10) such that the maximum degree of the

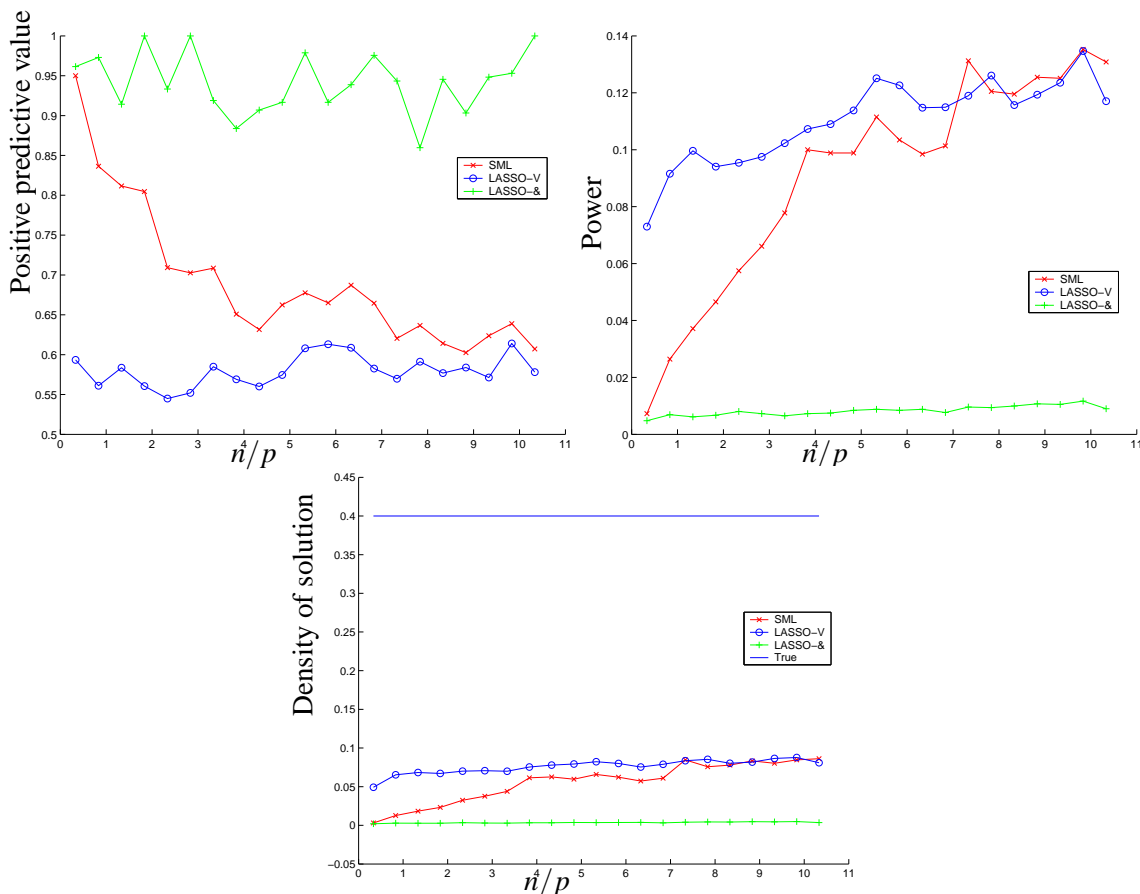


Figure 10: Classifying zeros and nonzeros for a true density of $\delta = 0.40$. We plot the positive predictive value, the power, and the estimated density using SML, Lasso-OR and Lasso-AND.

resulting graph was $d = 4$. We drew n i.i.d. samples and then attempted to recover the underlying sparsity pattern using both the approximate sparse maximum likelihood method and ℓ_1 -norm penalized logistic regression as described by Wainwright et al. (2006).

In Figure 12 we plot the average sensitivity (the fraction of actual nonzeros that are correctly identified as nonzeros) as well as the average specificity (the fraction of actual zeros that are correctly identified as zeros) for a range of sample sizes. We used a significance level of $\alpha = 0.05$ for the approximate sparse maximum likelihood method, and a regularization parameter choice of $\lambda_n = 0.04 * (\log p)^3 / \sqrt{n}$ for the penalized logistic regression AND method, as described by Wainwright et al. (2006).

7. Gene Expression and U.S. Senate Voting Records Data

We tested our algorithms on three sets of data: two gene expression data sets, as well as US Senate voting records. Gene expression data is often assumed to be approximately normally distributed,

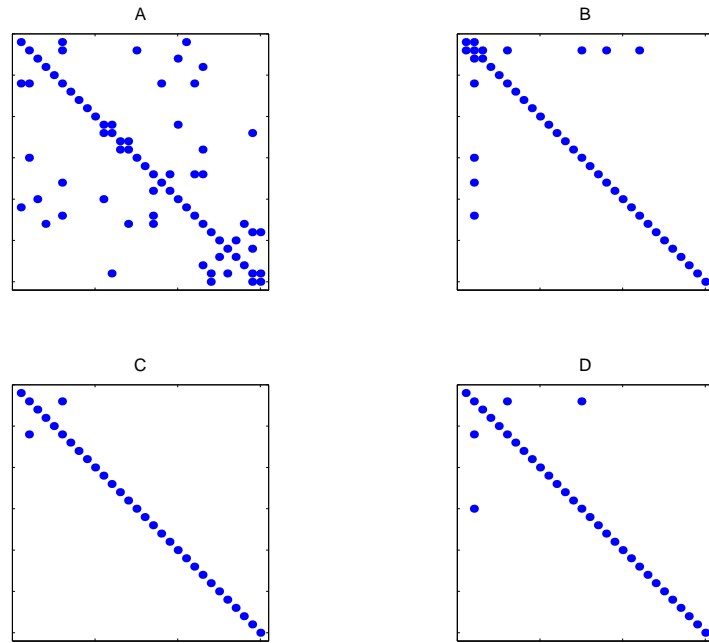


Figure 11: Comparing sparsity pattern recovery to the Lasso. (A) true covariance (B) Lasso-OR (C) Lasso-AND (D) SML.

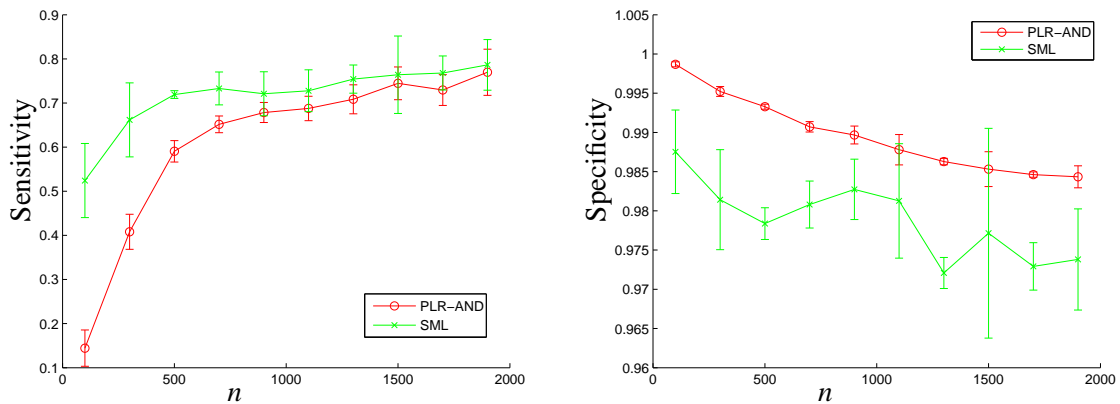


Figure 12: Comparing sparsity pattern recovery to the ℓ_1 -norm penalized logistic regression (PLR). We show the sensitivity (left) as well as the specificity (right) as the number of samples is increased.

while the voting records data is here considered binary. In this section we briefly explore their respective estimated graphical models.

7.1 Rosetta Inpharmatics Compendium

We applied our algorithms to the Rosetta Inpharmatics Compendium of gene expression profiles described by Hughes et al. (2000). The 300 experiment compendium contains $n = 253$ samples with $p = 6136$ variables. With a view towards obtaining a very sparse graph, we replaced $\alpha/2p^2$ in (3) by α , and set $\alpha = 0.05$. The resulting penalty parameter is $\lambda = 0.0313$.

This is a large penalty for this data set, and by applying Theorem 4 we find that all but 270 of the variables are estimated to be independent from all the rest, clearly a very conservative estimate. Figure 13 displays the resulting graph.

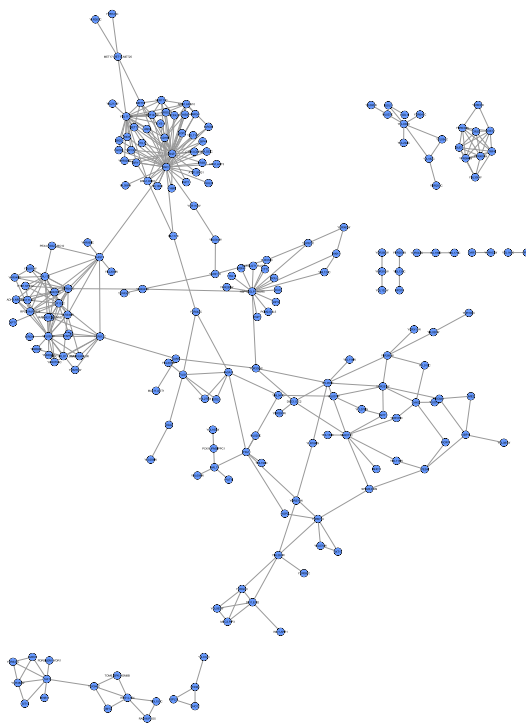


Figure 13: Application to Hughes compendium. The above graph results from solving (1) for this data set with a penalty parameter of $\lambda = 0.0313$.

Figure 14 focuses on a region of Figure 13, a cluster of genes that is unconnected to the remaining genes in this estimate. According to Gene Ontology (see Ashburner et al., 2000), these genes are associated with iron homeostasis. The probability that a gene has been false included in this cluster is at most 0.05.

As a second example, in Figure 15, we show a subgraph of genes associated with cellular membrane fusion. All three graphs were rendered using Cytoscape.

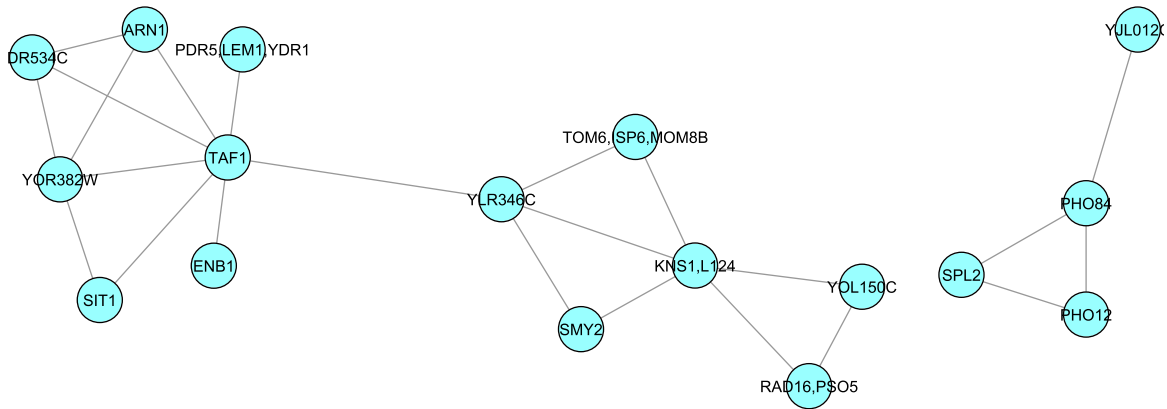


Figure 14: Application to Hughes data set (closeup of Figure 13). These genes are associated with iron homeostasis.

7.2 Iconix Microarray Data

Next we analyzed a subset of a 10,000 gene microarray data set from 160 drug treated rat livers (Natsoulis et al., 2005). In this study, rats were treated with a variety of fibrates, statin, or estrogen receptor agonist compounds. Taking the 500 genes with the highest variance, we once again replaced $\alpha/2p^2$ in (3) by α , and set $\alpha = 0.05$. The resulting penalty parameter is $\lambda = 0.0853$.

By applying Theorem 4 we find that all but 339 of the variables are estimated to be independent from the rest. This estimate is less conservative than that obtained in the Hughes case since the ratio of samples to variables is 160 to 500 instead of 253 to 6136.

The first order neighbors of any node in a Gaussian graphical model form the set of predictors for that variable. In the estimated obtained by solving (1), we found that LDL receptor had one of the largest number of first-order neighbors in the Gaussian graphical model. The LDL receptor is believed to be one of the key mediators of the effect of both statins and estrogenic compounds on LDL cholesterol. Table 1 lists some of the first order neighbors of LDL receptor.

It is perhaps not surprising that several of these genes are directly involved in either lipid or steroid metabolism (K03249, AI411979, AI410548, NM_013200, Y00102). Other genes such as Cbp/p300 are known to be global transcriptional regulators. Finally, some are un-annotated ESTs. Their connection to the LDL receptor in this analysis may provide clues to their function.

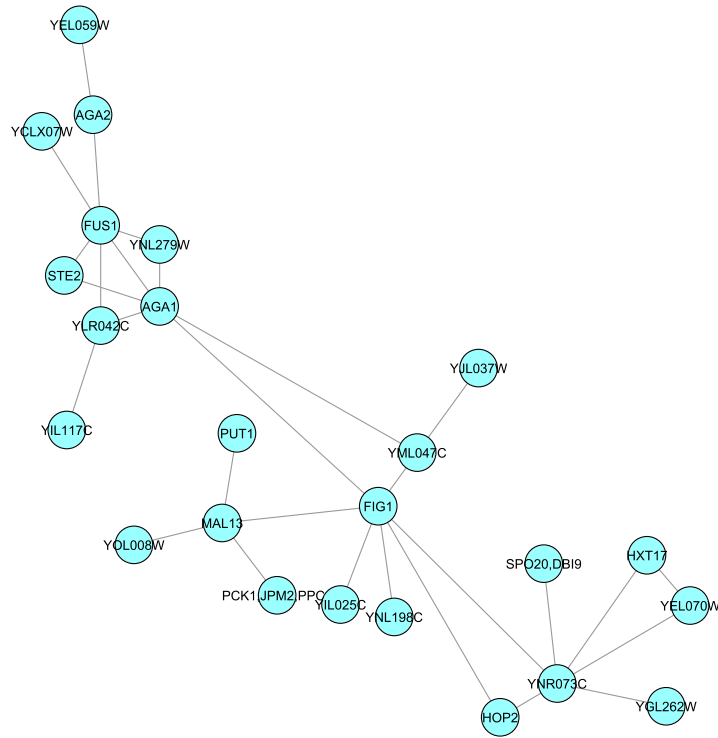


Figure 15: Application to Hughes data set (subgraph of Figure 13). These genes are associated with cellular membrane fusion.

ACCESSION	GENE
BF553500	CBP/P300-INTERACTING TRANSACTIVATOR
BF387347	EST
BF405996	CALCIUM CHANNEL, VOLTAGE DEPENDENT
NM_017158	CYTOCHROME P450, 2C39
K03249	ENOYL-CoA, HYDRATASE/3-HYDROXYACYL CO A DEHYDROG.
BE100965	EST
AI411979	CARNITINE O-ACETYLTRANSFERASE
AI410548	3-HYDROXYISOBUTYRYL-Co A HYDROLASE
NM_017288	SODIUM CHANNEL, VOLTAGE-GATED
Y00102	ESTROGEN RECEPTOR 1
NM_013200	CARNITINE PALMITOYLTRANSFERASE 1B

Table 1: Predictor genes for LDL receptor.

7.3 Senate Voting Records Data

We conclude our numerical experiments by testing our approximate sparse maximum likelihood estimation method on binary data. The data set consists of US senate voting records data from the 109th congress (2004 - 2006). There are one hundred variables, corresponding to 100 senators.

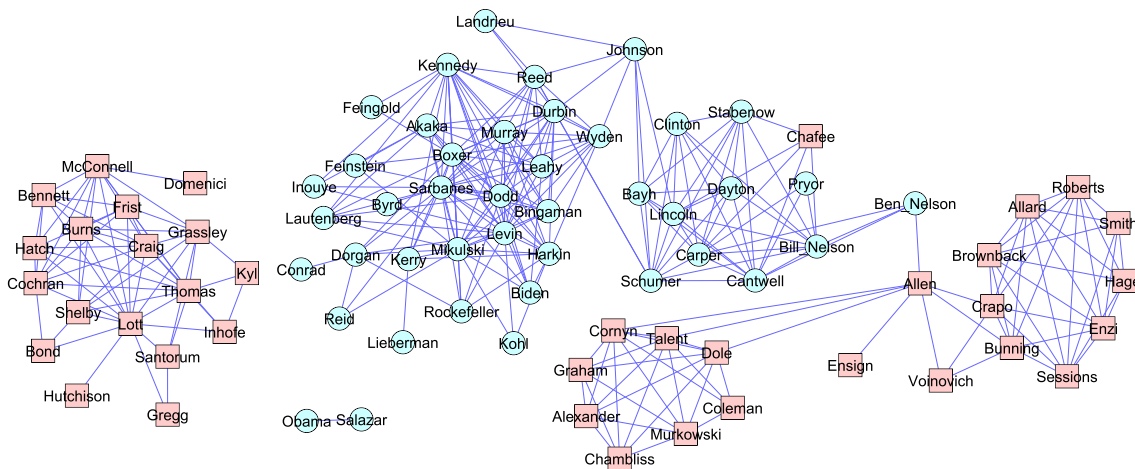


Figure 16: US Senate, 109th Congress (2004-2006). The graph displays the solution to (12) obtained using the log determinant relaxation to the log partition function of Wainwright and Jordan (2006). Democratic senators are represented by round nodes and Republican senators are represented by square nodes.

Each of the 542 samples is bill that was put to a vote. The votes are recorded as -1 for no and 1 for yes.

There are many missing values in this data set, corresponding to missed votes. Since our analysis depends on data values taken solely from $\{-1, 1\}$, it was necessary to impute values to these. For this experiment, we replaced all missing votes with noes (-1). We chose the penalty parameter $\lambda(\alpha)$ according to (17), using a significance level of $\alpha = 0.05$. Figure 16 shows the resulting graphical model, rendered using Cytoscape. Red nodes correspond to Republican senators, and blue nodes correspond to Democratic senators.

We can make some tentative observations by browsing the network of senators. As neighbors most Democrats have only other Democrats and Republicans have only other Republicans. Senator Chafee (R, RI) has only Democrats as his neighbors, an observation that supports media statements made by and about Chafee during those years. Senator Allen (R, VA) unites two otherwise separate groups of Republicans and also provides a connection to the large cluster of Democrats through Ben Nelson (D, NE), which also supports media statements made about him prior to his 2006 re-election campaign. Thus, although we obtained this graphical model via a relaxation of the log partition function, the resulting picture is supported by conventional wisdom. Figure 17 shows a subgraph consisting of neighbors of degree three or lower of Senator Allen.

Finally, we estimated the instability of these results using 10-fold cross validation, as described in Section 6.6. The resulting estimate of the instability is 0.00376, suggesting that our estimate of the graphical model is fairly stable.

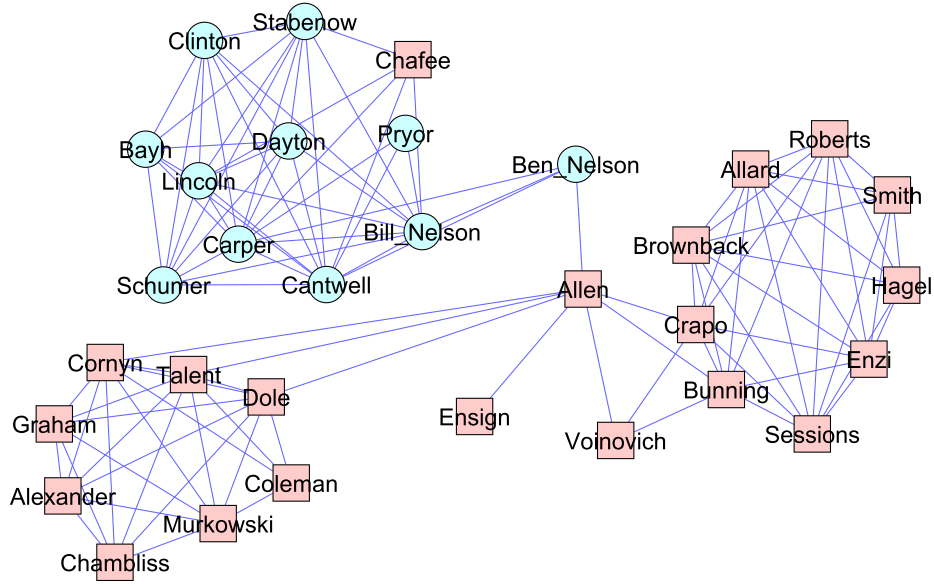


Figure 17: US Senate, 109th Congress. Neighbors of Senator Allen (degree three or lower).

Acknowledgments

We are indebted to Georges Natsoulis for his interpretation of the Iconix data set analysis and for gathering the senate voting records. We also thank Martin Wainwright, Bin Yu, Peter Bartlett, and Michael Jordan for many enlightening conversations.

Appendix A. Proof of Solution Properties and Block Coordinate Descent Convergence

In this section, we give short proofs of the two theorems on properties of the solution to (1), as well as the convergence of the block coordinate descent method.

Proof of Theorem 1:

Since $\hat{\Sigma}$ satisfies $\hat{\Sigma} = S + \hat{U}$, where $\|U\|_\infty \leq \lambda$, we have:

$$\begin{aligned} \|\hat{\Sigma}\|_2 &= \|S + \hat{U}\|_2 \\ &\leq \|S\|_2 + \|\hat{U}\|_2 \leq \|S\|_2 + \|U\|_\infty \leq \|S\|_2 + \lambda p \end{aligned}$$

which yields the lower bound on $\|\hat{\Sigma}^{-1}\|_2$. Likewise, we can show that $\|\hat{\Sigma}^{-1}\|_2$ is bounded above. At the optimum, the primal dual gap is zero:

$$\begin{aligned} &-\log \det \hat{\Sigma}^{-1} + \text{trace}(S\hat{\Sigma}^{-1}) + \lambda \|\hat{\Sigma}^{-1}\|_1 - \log \det \hat{\Sigma} - p \\ &= \text{trace}(S\hat{\Sigma}^{-1}) + \lambda \|\hat{\Sigma}^{-1}\|_1 - p = 0 \end{aligned}$$

We therefore have

$$\begin{aligned} \|\hat{\Sigma}^{-1}\|_2 &\leq \|\hat{\Sigma}^{-1}\|_F \leq \|\hat{\Sigma}^{-1}\|_1 \\ &= p/\lambda - \text{trace}(S\hat{\Sigma}^{-1})/\lambda \leq p/\lambda \end{aligned}$$

where the last inequality follows from $\text{trace}(S\hat{\Sigma}^{-1}) \geq 0$, since $S \succeq 0$ and $\hat{\Sigma}^{-1} \succ 0$. \blacksquare

Next we prove the convergence of block coordinate descent:

Proof of Theorem 3:

To see that optimizing over one row and column of W in (2) yields the quadratic program (4), let all but the last row and column of W be fixed. Since we know the diagonal entries of the solution, we can fix the remaining diagonal entry as well:

$$W = \begin{pmatrix} W_{\setminus p \setminus p} & w_p \\ w_p^T & W_{pp} \end{pmatrix}.$$

Then, using Schur complements, we have that

$$\det W = \det W_{\setminus p \setminus p} \cdot (W_{pp} - w_p^T (W_{\setminus p \setminus p})^{-1} w_p)$$

which gives rise to (4).

By general results on block coordinate descent algorithms (e.g., Bertsekas, 1998), the algorithm converges if (4) has a unique solution at each iteration. Thus it suffices to show that, at every sweep, $W^{(j)} \succ 0$ for all columns j . Prior to the first sweep, the initial value of the variable is positive definite: $W^{(0)} \succ 0$ since $W^{(0)} := S + \lambda J$, and we have $S \succeq 0$ and $\lambda > 0$ by assumption.

Now suppose that $W^{(j)} \succ 0$. This implies that the following Schur complement is positive:

$$w_{jj} - W_j^T (W_{\setminus j \setminus j}^{(j)})^{-1} W_j > 0$$

By the update rule we have that the corresponding Schur complement for $W^{(j+1)}$ is even greater:

$$w_{jj} - W_j^T (W_{\setminus j \setminus j}^{(j+1)})^{-1} W_j > w_{jj} - W_j^T (W_{\setminus j \setminus j}^{(j)})^{-1} W_j > 0$$

so that $W^{(j+1)} \succ 0$. \blacksquare

Finally, we apply Theorem 3 to prove the second property of the solution.

Proof of Theorem 4:

Suppose that column j of the second moment matrix satisfies $|S_{ij}| \leq \lambda$ for all $i \neq j$. This means that the zero vector is in the constraint set of (4) for that column. Each time we return to column j , the objective function will be different, but always of the form $y^T A y$ for $A \succ 0$. Since the constraint set will not change, the solution for column j will always be zero. By Theorem 3, the block coordinate descent algorithm converges to a solution, and so therefore the solution must have $\hat{\Sigma}_j = 0$. \blacksquare

Appendix B. Proof of Error Bounds

Next we shall show that the penalty parameter choice given in (3) yields the error probability bound of Theorem 2. The proof is nearly identical to that of (Meinshausen and Bühlmann, 2006, Theorem 3). The differences stem from a different objective function, and the fact that our variable is a matrix of size p rather than a vector of size p . Our proof is only an adaptation of their proof to our problem.

B.1 Preliminaries

Before we begin, consider problem (1), for a matrix S of any size:

$$\hat{X} = \arg \min -\log \det X + \text{trace}(SX) + \lambda \|X\|_1$$

where we have dropped the constraint $X \succ 0$ since it is implicit, due to the log determinant function. Since the problem is unconstrained, the solution \hat{X} must correspond to setting the subgradient of the objective to zero:

$$\begin{aligned} S_{ij} - X_{ij}^{-1} &= -\lambda & \text{for } X_{ij} > 0, \\ S_{ij} - X_{ij}^{-1} &= \lambda & \text{for } X_{ij} < 0, \\ |S_{ij} - X_{ij}^{-1}| &\leq \lambda & \text{for } X_{ij} = 0. \end{aligned} \tag{18}$$

Recall that by Theorem 1, the solution is unique for λ positive.

B.2 Proof of Error Bound for Gaussian Data

Now we are ready to prove Theorem 2.

Proof of Theorem 2:

Sort columns of the covariance matrix so that variables in the same connectivity component are grouped together. The correct zero pattern for the covariance matrix is then block diagonal. Define

$$\Sigma^{\text{correct}} := \text{blk diag}(C_1, \dots, C_\ell) \tag{19}$$

The inverse $(\Sigma^{\text{correct}})^{-1}$ must also be block diagonal, with possible additional zeros inside the blocks. If we constrain the solution to (1) to have this structure, then by the form of the objective, we can optimize over each block separately. For each block, the solution is characterized by (18).

Now, suppose that

$$\lambda > \max_{i \in N, j \in N \setminus C_i} |S_{ij} - \Sigma_{ij}^{\text{correct}}|. \tag{20}$$

Then, by the subgradient characterization of the solution noted above, and the fact that the solution is unique for $\lambda > 0$, it must be the case that $\hat{\Sigma} = \Sigma^{\text{correct}}$. By the definition of Σ^{correct} , this implies that, for $\hat{\Sigma}$, we have $\hat{C}_k = C_k$ for all $k \in N$.

Taking the contrapositive of this statement, we can write:

$$\begin{aligned} &P(\exists k \in N : \hat{C}_k \not\subseteq C_k) \\ &\leq P(\max_{i \in N, j \in N \setminus C_i} |S_{ij} - \Sigma_{ij}^{\text{correct}}| \geq \lambda) \\ &\leq p^2(n) \cdot \max_{i \in N, j \in N \setminus C_i} P(|S_{ij} - \Sigma_{ij}^{\text{correct}}| \geq \lambda) \\ &= p^2(n) \cdot \max_{i \in N, j \in N \setminus C_i} P(|S_{ij}| \geq \lambda). \end{aligned} \tag{21}$$

The equality at the end follows since, by definition, $\Sigma_{ij}^{\text{correct}} = 0$ for $j \in N \setminus C_i$. It remains to bound $P(|S_{ij}| \geq \lambda)$.

The statement $|S_{kj}| \geq \lambda$ can be written as:

$$|R_{kj}|(1 - R_{kj}^2)^{-\frac{1}{2}} \geq \lambda(s_{kk}s_{jj} - \lambda^2)^{-\frac{1}{2}}$$

where R_{kj} is the correlation between variables k and j , since

$$|R_{kj}|(1 - R_{kj}^2)^{-\frac{1}{2}} = |S_{kj}|(S_{kk}S_{jj} - S_{kj}^2)^{-\frac{1}{2}}.$$

Furthermore, the condition $j \in N \setminus C_k$ is equivalent to saying that variables k and j are independent: $\Sigma_{kj} = 0$. Conditional on this, the statistic

$$R_{kj}(1 - R_{kj}^2)^{-\frac{1}{2}}(n-2)^{\frac{1}{2}}$$

has a Student's t-distribution for $n-2$ degrees of freedom. Therefore, for all $j \in N \setminus C_k$,

$$\begin{aligned} & P(|S_{kj}| \geq \lambda | S_{kk} = s_{kk}, S_{jj} = s_{jj}) \\ &= 2P(T_{n-2} \geq \lambda(s_{kk}s_{jj} - \lambda^2)^{-\frac{1}{2}}(n-2)^{\frac{1}{2}} | S_{kk} = s_{kk}, S_{jj} = s_{jj}) \\ &\leq 2\tilde{F}_{n-2}(\lambda(\hat{\sigma}_k^2\hat{\sigma}_j^2 - \lambda^2)^{-\frac{1}{2}}(n-2)^{\frac{1}{2}}) \end{aligned} \quad (22)$$

where $\hat{\sigma}_k^2$ is the sample variance of variable k , and $\tilde{F}_{n-2} = 1 - F_{n-2}$ is the CDF of the Student's t-distribution with $n-2$ degree of freedom. This implies that, for all $j \in N \setminus C_k$,

$$P(|S_{kj}| \geq \lambda) \leq 2\tilde{F}_{n-2}(\lambda(\hat{\sigma}_k^2\hat{\sigma}_j^2 - \lambda^2)^{-\frac{1}{2}}(n-2)^{\frac{1}{2}})$$

since $P(A) = \int P(A|B)P(B)dB \leq K \int P(B)dB = K$. Putting the inequalities together, we have that:

$$\begin{aligned} & P(\exists k : \hat{C}_k^\lambda \not\subseteq C_k) \\ &\leq p^2 \cdot \max_{k,j \in N \setminus C_k} 2\tilde{F}_{n-2}(\lambda(\hat{\sigma}_k^2\hat{\sigma}_j^2 - \lambda^2)^{-\frac{1}{2}}(n-2)^{\frac{1}{2}}) \\ &= 2p^2\tilde{F}_{n-2}(\lambda((n-2)/((\max_{i>j} \hat{\sigma}_i\hat{\sigma}_j)^2 - \lambda^2))^{\frac{1}{2}}). \end{aligned}$$

For any fixed α , our required condition on λ is therefore

$$\tilde{F}_{n-2}(\lambda((n-2)/((\max_{i>j} \hat{\sigma}_i\hat{\sigma}_j)^2 - \lambda^2))^{\frac{1}{2}}) = \alpha/2p^2$$

which is satisfied by choosing λ according to (3). ■

B.3 Proof of Bound for Binary Data

We can reuse much of the previous proof to derive a corresponding formula for the binary case.

Proof of Theorem 7:

The proof of Theorem 7 is identical to the proof of Theorem 2, except that we have a different null distribution for $|S_{kj}|$. The null distribution of

$$nR_{kj}^2$$

is chi-squared with one degree of freedom. Analogous to (22), we have:

$$\begin{aligned} & P(|S_{kj}| \geq \lambda | S_{kk} = s_{kk}, S_{jj} = s_{jj}) \\ &= 2P(nR_{kj}^2 \geq n\lambda^2 s_{kk}s_{jj} | S_{kk} = s_{kk}, S_{jj} = s_{jj}) \\ &\leq 2\tilde{G}(n\lambda^2\hat{\sigma}_k^2\hat{\sigma}_j^2) \end{aligned}$$

where $\hat{\sigma}_k^2$ is the sample variance of variable k , and $\tilde{G} = 1 - G$ is the CDF of the chi-squared distribution with one degree of freedom. This implies that, for all $j \in N \setminus C_k$,

$$P(|S_{kj}| \geq \lambda) \leq 2\tilde{G}((\lambda\hat{\sigma}_k\hat{\sigma}_j\sqrt{n})^2).$$

Putting the inequalities together, we have that:

$$\begin{aligned} & P(\exists k : \hat{C}_k^\lambda \not\subseteq C_k) \\ & \leq p^2 \cdot \max_{k, j \in N \setminus C_k} 2\tilde{G}((\lambda \hat{\sigma}_k \hat{\sigma}_j \sqrt{n})^2) \\ & = 2p^2 \tilde{G}((\min_{i>j} \hat{\sigma}_k \hat{\sigma}_j)^2 n \lambda^2) \end{aligned}$$

so that, for any fixed α , we can achieve our desired bound by choosing $\lambda(\alpha)$ according to (17). ■

Appendix C. Proof of Connection Between Gaussian SML and Binary ASML

We end with a proof of Theorem 6, which connects the exact Gaussian sparse maximum likelihood problem with the approximate sparse maximum likelihood problem obtained by using the log determinant relaxation of Wainwright and Jordan (2006). First we must prove Lemma 5.

Proof of Lemma 5:

The conjugate function for the convex normalization $A(\theta)$ is defined as

$$A^*(\mu) := \sup_{\theta} \{ \langle \mu, \theta \rangle - A(\theta) \}. \quad (23)$$

Wainwright and Jordan derive a lower bound on this conjugate function using an entropy bound:

$$A^*(\mu) \geq B^*(\mu). \quad (24)$$

Since our original variables are spin variables $x \in \{-1, +1\}$, the bound given in the paper is

$$B^*(\mu) := -\frac{1}{2} \log \det(R(\mu) + \text{diag}(m)) - \frac{p}{2} \log\left(\frac{e\pi}{2}\right) \quad (25)$$

where $m := (1, \frac{4}{3}, \dots, \frac{4}{3})$.

The dual of this lower bound is $B(\theta)$:

$$B^*(\mu) := \max_{\theta} \langle \theta, \mu \rangle - B(\theta) \leq \max_{\theta} \langle \theta, \mu \rangle - A(\theta) =: A^*(\mu). \quad (26)$$

This means that, for all μ, θ ,

$$\langle \theta, \mu \rangle - B(\theta) \leq A^*(\mu) \quad (27)$$

or

$$B(\theta) \geq \langle \theta, \mu \rangle - A^*(\mu) \quad (28)$$

so that in particular

$$B(\theta) \geq \max_{\mu} \langle \theta, \mu \rangle - A^*(\mu) =: A(\theta) \quad (29)$$

Using the definition of $B(\theta)$ and its dual $B^*(\mu)$, we can write

$$\begin{aligned}
 B(\theta) &:= \max_{\mu} \langle \theta, \mu \rangle - B^*(\mu) \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) + \max_{\mu} \frac{1}{2} \langle R(\theta), R(\mu) \rangle + \frac{1}{2} \log \det(R(\mu) + \text{diag}(m)) \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) + \frac{1}{2} \cdot \max\{\langle R(\theta), X - \text{diag}(m) \rangle + \log \det(X) : X \succ 0, \text{diag}(X) = m\} \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) + \frac{1}{2} \cdot \{\max_{X \succ 0} \min_{\mathbf{v}} \langle R(\theta), X - \text{diag}(m) \rangle + \log \det(X) + \mathbf{v}^T (\text{diag}(X) - m)\} \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) + \frac{1}{2} \cdot \{\max_{X \succ 0} \min_{\mathbf{v}} \langle R(\theta) + \text{diag}(\mathbf{v}), X \rangle + \log \det(X) - \mathbf{v}^T m\} \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) + \frac{1}{2} \cdot \{\min_{\mathbf{v}} -\mathbf{v}^T m + \max_{X \succ 0} \langle R(\theta) + \text{diag}(\mathbf{v}), X \rangle + \log \det(X)\} \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) + \frac{1}{2} \cdot \{\min_{\mathbf{v}} -\mathbf{v}^T m - \log \det(-(R(\theta) + \text{diag}(\mathbf{v}))) - (p+1)\} \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) - \frac{1}{2}(p+1) + \frac{1}{2} \cdot \{\min_{\mathbf{v}} -\mathbf{v}^T m - \log \det(-(R(\theta) + \text{diag}(\mathbf{v})))\} \\
 &= \frac{p}{2} \log\left(\frac{e\pi}{2}\right) - \frac{1}{2}(p+1) - \frac{1}{2} \cdot \{\max_{\mathbf{v}} \mathbf{v}^T m + \log \det(-(R(\theta) + \text{diag}(\mathbf{v}\lambda))\}.
 \end{aligned} \tag{30}$$

■

Now we use Lemma 5 to prove the main result of Section 5.1. Having expressed the upper bound on the log partition function as a constant minus a maximization problem will help when we formulate the sparse approximate maximum likelihood problem.

Proof of Theorem 6:

The approximate sparse maximum likelihood problem is obtained by replacing the log partition function $A(\theta)$ with its upper bound $B(\theta)$, as derived in Lemma 5:

$$\begin{aligned}
 &n \cdot \{\max_{\theta} \frac{1}{2} \langle R(\theta), R(\bar{z}) \rangle - B(\theta) - \lambda \|\theta\|_1\} \\
 &= n \cdot \{\max_{\theta} \frac{1}{2} \langle R(\theta), R(\bar{z}) \rangle - \lambda \|\theta\|_1 + \frac{1}{2}(p+1) - \frac{p}{2} \log\left(\frac{e\pi}{2}\right) \\
 &\quad + \frac{1}{2} \cdot \{\max_{\mathbf{v}} \mathbf{v}^T m + \log \det(-(R(\theta) + \text{diag}(\mathbf{v})))\}\} \\
 &= \frac{n}{2}(p+1) - \frac{np}{2} \log\left(\frac{e\pi}{2}\right) + \frac{n}{2} \cdot \max_{\theta, \mathbf{v}} \{\mathbf{v}^T m + \langle R(\theta), R(\bar{z}) \rangle \\
 &\quad + \log \det(-(R(\theta) + \text{diag}(\mathbf{v}))) - 2\lambda \|\theta\|_1\}.
 \end{aligned} \tag{31}$$

We can collect the variables θ and \mathbf{v} into an unconstrained symmetric matrix variable $Y := -(R(\theta) + \text{diag}(\mathbf{v}))$.

Observe that

$$\begin{aligned}
 \langle R(\theta), R(\bar{z}) \rangle &= \langle -Y - \text{diag}(\mathbf{v}), R(\bar{z}) \rangle \\
 &= -\langle Y, R(\bar{z}) \rangle - \langle \text{diag}(\mathbf{v}), R(\bar{z}) \rangle = -\langle Y, R(\bar{z}) \rangle
 \end{aligned} \tag{32}$$

and that

$$\begin{aligned}
 \mathbf{v}^T m &= \langle \text{diag}(\mathbf{v}), \text{diag}(m) \rangle = \langle -Y - R(\theta), \text{diag}(m) \rangle \\
 &= -\langle Y, \text{diag}(m) \rangle - \langle R(\theta), \text{diag}(m) \rangle = -\langle Y, \text{diag}(m) \rangle.
 \end{aligned} \tag{33}$$

The approximate sparse maximum likelihood problem can then be written in terms of Y :

$$\begin{aligned}
 &\frac{n}{2}(p+1) - \frac{np}{2} \log\left(\frac{e\pi}{2}\right) + \frac{n}{2} \cdot \max_{\theta, \mathbf{v}} \{\mathbf{v}^T m + \langle R(\theta), R(\bar{z}) \rangle \\
 &\quad + \log \det(-(R(\theta) + \text{diag}(m))) - 2\lambda \|\theta\|_1\} \\
 &= \frac{n}{2}(p+1) - \frac{np}{2} \log\left(\frac{e\pi}{2}\right) + \frac{n}{2} \cdot \max\{\log \det Y - \langle Y, R(\bar{z}) + \text{diag}(m) \rangle \\
 &\quad - 2\lambda \sum_{i=2}^p \sum_{j=i+1}^{p+1} |Y_{ij}|\}.
 \end{aligned} \tag{34}$$

If we let $M := R(\bar{z}) + \text{diag}(m)$, then:

$$M = \begin{pmatrix} 1 & \bar{\mu}^T \\ \bar{\mu} & Z + \frac{1}{3}I \end{pmatrix}$$

where $\bar{\mu}$ is the sample mean and

$$Z = \frac{1}{n} \sum_{k=1}^n z^{(k)} (z^{(k)})^T.$$

Due to the added $\frac{1}{3}I$ term, we have that $M \succ 0$ for any data set.

The problem can now be written as:

$$\hat{Y} := \arg \max \{ \log \det Y - \langle Y, M \rangle - 2\lambda \sum_{i=2}^p \sum_{j=i+1}^{p+1} |Y_{ij}| : Y \succ 0 \}. \quad (35)$$

Since we are only penalizing certain elements of the variable Y , the solution \hat{X} of the dual problem to (35) will be of the form:

$$\hat{X} = \begin{pmatrix} 1 & \bar{\mu}^T \\ \bar{\mu} & \tilde{X} \end{pmatrix}$$

where

$$\tilde{X} := \arg \max \{ \log \det V : V_{kk} = Z_{kk} + \frac{1}{3}, |V_{kj} - Z_{kj}| \leq \lambda \}.$$

We can write an equivalent problem for estimating the covariance matrix. Define a new variable:

$$\Gamma = V - \bar{\mu} \bar{\mu}^T.$$

Using this variable, and the fact that the second moment matrix about the mean, defined as before, can be written

$$S = \frac{1}{n} \sum_{k=1}^n z^{(k)} (z^{(k)})^T - \bar{\mu} \bar{\mu}^T = Z - \bar{\mu} \bar{\mu}^T$$

we obtain the formulation (16). Using Schur complements, we see that our primal variable is of the form:

$$Y = \begin{pmatrix} * & * \\ * & \hat{\Gamma}^{-1} \end{pmatrix}.$$

From our definition of the variable Y , we see that the parameters we are estimating, $\hat{\theta}_{kj}$, are the negatives of the off-diagonal elements of $\hat{\Gamma}^{-1}$, which gives us (15). ■

References

- M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: Tool for the unification of biology. *Nature Genet.*, 25:25–29, 2000.
- D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1998.
- P. J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Annals of Statistics*, 36(1):199–227, 2008.
- J. Dahl, V. Roychowdhury, and L. Vandenberghe. Covariance selection for non-chordal graphs via chordal embedding. *To appear in Optimization Methods and Software*, Revised 2007.

- A. d'Aspremont, L. El Ghaoui, M.I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *Advances in Neural Information Processing Systems*, 17, 2004.
- A. Dobra and M. West. Bayesian covariance selection. Technical Report, 2004.
- J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 2007.
- T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraborty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.
- S. Lauritzen. *Graphical Models*. Springer Verlag, 1996.
- H. Li and J. Gui. Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *University of Pennsylvania Technical Report*, 2005.
- Z. Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.
- N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of statistics*, 34:1436–1462, 2006.
- G. Natsoulis, L. El Ghaoui, G. Lanckriet, A. Tolley, F. Leroy, S. Dunlea, B. Eynon, C. Pearson, S. Tugendreich, and K. Jarnagin. Classification of a large microarray data set: algorithm comparison and analysis of drug signatures. *Genome Research*, 15:724–736, 2005.
- Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Prog.*, 103(1):127–152, 2005.
- M. R. Osborne, B. Presnell, and B. A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.
- T. P. Speed and H. T. Kiiveri. Gaussian markov distributions over finite graphs. *Annals of Statistics*, 14(1):138–150, 1986.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal statistical society, series B*, 58(267-288), 1996.
- L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(4):499–533, 1998.
- M. Wainwright and M. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing*, 2006.
- M. Wainwright, P. Ravikumar, and J. D. Lafferty. High-dimensional graphical model selection using ℓ_1 -regularized logistic regression. *Proceedings of Advances in Neural Information Processing Systems*, 2006.