# Sparse inverse covariance estimation with the graphical lasso

Jerome Friedman *
Trevor Hastie †
and Robert Tibshirani‡

November 17, 2007

**Abstract**

We consider the problem of estimating sparse graphs by a lasso penalty applied to the inverse covariance matrix. Using a coordinate descent procedure for the lasso, we develop a simple algorithm— the *Graphical Lasso*— that is remarkably fast: it solves a 1000 node problem ($\sim 500,000$ parameters) in at most a minute, and is 30 to 4000 times faster than competing methods. It also provides a conceptual link between the exact problem and the approximation suggested by Meinshausen & Bühlmann (2006). We illustrate the method on some cell-signaling data from proteomics.

## 1 Introduction

In recent years a number of authors have proposed the estimation of sparse undirected graphical models through the use of $L_1$ (lasso) regularization. The basic model for continuous data assumes that the observations have a multivariate Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$. If the $ij$th component of $\Sigma^{-1}$ is zero, then variables $i$ and $j$ are conditionally

*Dept. of Statistics, Stanford Univ., CA 94305, jhf@stanford.edu
†Depts. of Statistics, and Health, Research & Policy, Stanford Univ., CA 94305, hastie@stanford.edu
‡Depts. of Health, Research & Policy, and Statistics, Stanford Univ, tibs@stanford.edu

independent, given the other variables. Thus it makes sense to impose an $L_1$ penalty for the estimation of $\Sigma^{-1}$, to increase its sparsity.

Meinshausen & Bühlmann (2006) take a simple approach to this problem; they estimate a sparse graphical model by fitting a lasso model to each variable, using the others as predictors. The component $\hat{\Sigma}^{-1}_{ij}$ is then estimated to be non-zero if either the estimated coefficient of variable $i$ on $j$, or the estimated coefficient of variable $j$ on $i$, is non-zero (alternatively they use an AND rule). They show that asymptotically, this consistently estimates the set of non-zero elements of $\Sigma^{-1}$.

Other authors have proposed algorithms for the exact maximization of the $L_1$-penalized log-likelihood; Yuan & Lin (2007), Banerjee et al. (2007) and Dahl et al. (2007) adapt interior point optimization methods for the solution to this problem. Both papers also establish that the simpler approach of Meinshausen & Bühlmann (2006) can be viewed as an approximation to the exact problem.

We use the blockwise coordinate descent approach in Banerjee et al. (2007) as a launching point, and propose a new algorithm for the exact problem. This new procedure is extremely simple, and is substantially faster competing approaches in our tests. It also bridges the "conceptual gap" between the Meinshausen & Bühlmann (2006) proposal and the exact problem.

# 2 The proposed method

Suppose we have $N$ multivariate normal observations of dimension $p$, with mean $\mu$ and covariance $\Sigma$. Following Banerjee et al. (2007), let $\Theta = \Sigma^{-1}$, and let $S$ be the empirical covariance matrix, the problem is to maximize the log-likelihood

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \rho||\Theta||_1, \tag{1}$$

over non-negative definite matrices $\Theta$ * Here tr denotes the trace and $||\Theta||_1$ is the $L_1$ norm— the sum of the absolute values of the elements of $\Sigma^{-1}$. Expression (1) is the Gaussian log-likelihood of the data, partially maximized with respect to the mean parameter $\mu$. Yuan & Lin (2007) solve this problem

*We note that while most authors use this formulation, Yuan & Lin (2007) omit the diagonal elements from the penalty.

using the interior point method for the "maxdet" problem, proposed by Vandenberghe et al. (1998). Banerjee et al. (2007) develop a different framework for the optimization, which was the impetus for our work.

Banerjee et al. (2007) show that the problem (1) is convex and consider estimation of $\Sigma$ (rather than $\Sigma^{-1}$), as follows. Let $W$ be the estimate of $\Sigma$. They show that one can solve the problem by optimizing over each row and corresponding column of $W$ in a block coordinate descent fashion. Partitioning $W$ and $S$

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}, \tag{2}$$

they show that the solution for $w_{12}$ satisfies

$$w_{12} = \mathrm{argmin}_y \{ y^T W_{11}^{-1} y : ||y - s_{12}||_\infty \leq \rho \}. \tag{3}$$

This is a box-constrained quadratic program which they solve using an interior point procedure. Permuting the rows and columns so the target column is always the last, they solve a problem like (3) for each column, updating their estimate of $W$ after each stage. This is repeated until convergence. If this procedure is initialized with a positive definite matrix, they show that the iterates from this procedure remains positive definite and invertible, even if $p > N$.

Using convex duality, Banerjee et al. (2007) go on to show that solving (3) is equivalent to solving the dual problem

$$\min_\beta \Big\{ \tfrac{1}{2} ||W_{11}^{1/2} \beta - b||^2 + \rho ||\beta||_1 \Big\}, \tag{4}$$

where $b = W_{11}^{-1/2} s_{12}$;[†] if $\beta$ solves (4), then $w_{12} = W_{11}\beta$ solves (3). Expression (4) resembles a lasso regression, and is the basis for our approach.

First we verify the equivalence between the solutions to (1) and (4) directly. Expanding the relation $W\Theta = I$ gives an expression that will be useful below:

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix} \begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix}. \tag{5}$$

Now the sub-gradient equation for maximization of the log-likelihood (1) is

$$W - S - \rho \cdot \Gamma = 0, \tag{6}$$

[†] The corresponding expression in Banerjee et al. (2007) does not have the leading $\frac{1}{2}$ and has a factor of $\frac{1}{2}$ in $b$. We have written it in this equivalent form to avoid factors of $\frac{1}{2}$ later.

using the fact that the derivative of $\log\det\Theta$ equals $\Theta^{-1} = W$, given in e.g Boyd & Vandenberghe (2004), page 641. Here $\Gamma_{ij} \in \text{sign}(\Theta_{ij})$; i.e. $\Gamma_{ij} = \text{sign}(\Theta_{ij})$ if $\Theta_{ij} \neq 0$, else $\Gamma_{ij} \in [-1, 1]$ if $\Theta_{ij} = 0$.

Now the upper right block of equation (6) is

$$w_{12} - s_{12} - \rho \cdot \gamma_{12} = 0. \tag{7}$$

On the other hand, the sub-gradient equation from (4) works out to be

$$W_{11}\beta - s_{12} + \rho \cdot \nu = 0, \tag{8}$$

where $\nu \in \text{sign}(\beta)$ element-wise. Now suppose $(W, \Gamma)$ solves (6), and hence $(w_{12}, \gamma_{12})$ solves (7). Then $\beta = W_{11}^{-1} w_{12}$ and $\nu = -\gamma_{12}$ solves (8). The equivalence of the first two terms is obvious. For the sign terms, since $W_{11}\theta_{12} + w_{12}\theta_{22} = 0$ from (5), we have that $\theta_{12} = -\theta_{22} W_{11}^{-1} w_{12}$. Since $\theta_{22} > 0$, it follows that $\text{sign}(\theta_{12}) = -\text{sign}(W_{11}^{-1} w_{12}) = -\text{sign}(\beta)$. This proves the equivalence. We note that the solution $\beta$ to the lasso problem (4) gives us (up to a negative constant) the corresponding part of $\Theta$: $\theta_{12} = -\theta_{22}\beta$.

Now to the main point of this paper. Problem (4) looks like a lasso ($L_1$-regularized) least squares problem. In fact if $W_{11} = S_{11}$, then the solutions $\hat{\beta}$ are easily seen to equal the lasso estimates for the $p$th variable on the others, and hence related to the Meinshausen & Bühlmann (2006) proposal. As pointed out by Banerjee et al. (2007), $W_{11} \neq S_{11}$ in general and hence the Meinshausen & Bühlmann (2006) approach does not yield the maximum likelihood estimator. They point out that their block-wise interior-point procedure is equivalent to recursively solving and updating the lasso problem (4), but do not pursue this approach. We do, to great advantage, because fast coordinate descent algorithms (Friedman et al. 2007) make solution of the lasso problem very attractive.

In terms of inner products, the usual lasso estimates for the $p$th variable on the others take as input the data $S_{11}$ and $s_{12}$. To solve (4) we instead use $W_{11}$ and $s_{12}$, where $W_{11}$ is our current estimate of the upper block of $W$. We then update $w$ and cycle through all of the variables until convergence.

Note that from (6), the solution $w_{ii} = s_{ii} + \rho$ for all $i$, since $\theta_{ii} > 0$, and hence $\Gamma_{ii} = 1$. For convenience we call this algorithm the *graphical lasso.* Here is the algorithm in detail:

*Graphical Lasso Algorithm*

1. Start with $W = S + \rho I$. The diagonal of $W$ remains unchanged in what follows.

2. For each $j = 1, 2, \ldots p, 1, 2, \ldots p, \ldots$, solve the lasso problem (4), which takes as input the inner products $W_{11}$ and $s_{12}$. This gives a $p - 1$ vector solution $\hat{\beta}$. Fill in the corresponding row and column of $W$ using $w_{12} = W_{11}\hat{\beta}$.

3. Continue until convergence

There is a simple, conceptually appealing way to view this procedure. Given a data matrix $\mathbf{X}$ and outcome vector $\mathbf{y}$, we can think of the linear least squares regression estimates $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ as functions not of the raw data, but instead the inner products $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{y}$. Similarly, one can show that the lasso estimates are functions of these inner products as well. Hence in the current problem, we can think of the lasso estimates for the $p$th variable on the others as having the functional form

$$\texttt{lasso}(S_{11}, s_{12}, \rho). \tag{9}$$

But application of the lasso to each variable does not solve problem (1); to solve this via the graphical lasso we instead use the inner products $W_{11}$ and $s_{12}$. That is, we replace (9) by

$$\texttt{lasso}(W_{11}, s_{12}, \rho). \tag{10}$$

The point is that problem (1) is not equivalent to $p$ separate regularized regression problems, but to $p$ coupled lasso problems that share the same $W$ and $\Theta = W^{-1}$. The use of $W_{11}$ in place of $S_{11}$ shares the information between the problems in an appropriate fashion.

Note that each iteration in step (2) implies a permutation of the rows and columns to make the target column the last. The lasso problem in step (2) above can be efficiently solved by coordinate descent (Friedman et al. 2007, Wu & Lange 2007). Here are the details. Letting $V = W_{11}$ and $u = s_{12}$, then the update has the form

$$\hat{\beta}_j \leftarrow S(u_j - \sum_{k \neq j} V_{kj}\hat{\beta}_k, \rho)/V_{jj} \tag{11}$$

for $j = 1, 2, \ldots, p, 1, 2, \ldots p, \ldots$, where $S$ is the soft-threshold operator:

$$S(x, t) = \text{sign}(x)(|x| - t)_+. \tag{12}$$

We cycle through the predictors until convergence. In our implementation, the procedure stops when the average absolute change in $W$ is less than $t \cdot \text{ave}|S^{-\text{diag}}|$ where $S^{-\text{diag}}$ are the off-diagonal elements of the empirical covariance matrix $S$, and $t$ is a fixed threshold, set by default at 0.001.

Note that $\hat{\beta}$ will typically be sparse, and so the computation $w_{12} = W_{11}\hat{\beta}$ will be fast; if there are $r$ non-zero elements, it takes $rp$ operations.

Although our algorithm has estimated $\hat{\Sigma} = W$, we can recover $\hat{\Theta} = W^{-1}$ relatively cheaply. Note that from the partitioning in (5), we have

$$\begin{aligned} W_{11}\theta_{12} + w_{12}\theta_{22} &= 0 \\ w_{12}^T\theta_{12} + w_{22}\theta_{22} &= 1, \end{aligned}$$

from which we derive the standard partitioned inverse expressions

$$\theta_{12} = -W_{11}^{-1}w_{12}\theta_{22} \tag{13}$$

$$\theta_{22} = 1/(w_{22} - w_{12}^T W_{11}^{-1} w_{12}). \tag{14}$$

But since $\hat{\beta} = W_{11}^{-1}w_{12}$, we have that $\hat{\theta}_{22} = 1/(w_{22} - w_{12}^T\hat{\beta})$ and $\hat{\theta}_{12} = -\hat{\beta}\hat{\theta}_{22}$. Thus $\hat{\theta}_{12}$ is a simply rescaling of $\hat{\beta}$ by $-\hat{\theta}_{22}$, which is easily computed. Although these calculations could be included in step 2 of the *graphical lasso algorithm*, they are not needed till the end; hence we store all the coefficients $\beta$ for each of the $p$ problems in a $p \times p$ matrix $\hat{B}$, and compute $\hat{\Theta}$ after convergence.

Interestingly, if $W = S$, these are just the formulas for obtaining the inverse of a partitioned matrix. That is, if we set $W = S$ and $\rho = 0$ in the above algorithm, then one sweep through the predictors computes $S^{-1}$, using a linear regression at each stage.

**Remark.** In some situations it might make sense to specify different amounts of regularization for each variable, or even allow each inverse covariance element to be penalized differently. Thus we maximize the log-likelihood

$$\log\det\Theta - \text{tr}(S\Theta) - ||\Theta * P||_1, \tag{15}$$

where $P = \{\rho_{jk}\}$ with $\rho_{jk} = \rho_{kj}$, and $*$ indicates componentwise multiplication. It is easy to show that (15) is maximized by the preceding algorithm, with $\rho$ replaced by $\rho_{jk}$ in the soft-thresholding step (11). Typically one might take $\rho_{jk} = \sqrt{\rho_j\rho_k}$ for some values $\rho_1, \rho_2, \ldots \rho_p$, to allow different amounts of regularization for each variable

| $p$ | Problem Type | (1) Graphical Lasso | (2) Approx | (3) COVSEL | Ratio of (3) to (1) |
|---|---|---|---|---|---|
| 100 | sparse | .014 | .007 | 34.7 | 2476.4 |
| 100 | dense | .053 | .018 | 2.2 | 40.9 |
| 200 | sparse | .050 | .027 | $> 205.35$ | $> 4107$ |
| 200 | dense | .497 | .146 | 16.9 | 33.9 |
| 400 | sparse | 1.23 | .193 | $> 1616.7$ | $> 1314.3$ |
| 400 | dense | 6.2 | .752 | 313.0 | 50.5 |

Table 1: *Timings (seconds) for graphical lasso, Meinhausen-Buhlmann approximation, and COVSEL procedures.*

# 3 Timing comparisons

We simulated Gaussian data from both *sparse* and *dense* scenarios, for a range of problem sizes $p$. The sparse scenario is the AR(1) model taken from Yuan & Lin (2007): $(\Sigma^{-1})_{ii} = 1$, $(\Sigma^{-1})_{i,i-1} = (\Sigma^{-1})_{i-1,i} = 0.5$, and zero otherwise. In the dense scenario, $(\Sigma^{-1})_{ii} = 2$,$(\Sigma^{-1})_{ii'} = 1$ otherwise. We chose the penalty parameter so that the solution had about the actual number of non-zero elements in the sparse setting, and about half of total number of elements in the dense setting. The graphical lasso procedure was coded in Fortran, linked to an R language function. All timings were carried out on a Intel Xeon 2.80GH processor.

We compared the graphical lasso to the COVSEL program provided by Banerjee et al. (2007). This is a Matlab program, with a loop that calls a C language code to do the box-constrained QP for each column of the solution matrix. To be as fair as possible to COVSEL, we only counted the CPU time spent in the C program. We set the maximum number of outer iterations to 30, and following the authors code, set the the duality gap for convergence to 0.1.

The number of CPU seconds for each trial is shown in Table 1. The algorithm took between 2 and 8 iterations of the outer loop. In the dense scenarios for $p = 200$ and 400, COVSEL had not converged by 30 iterations. We see that the graphical lasso is 30 to 4000 times faster than COVSEL, and only about two to ten times slower than the approximate method.

Figure 1 shows the number of CPU seconds required for the graphical lasso procedure, for problem sizes up to 1000. The computation time is $O(p^3)$

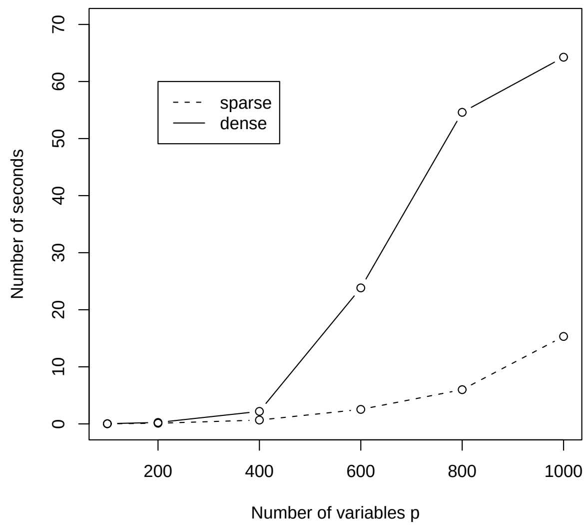


Figure 1: *Number of CPU seconds required for the graphical lasso procedure.*

for dense problems, and considerably less than that for sparse problems. Even in the dense scenario, it solves a 1000 node problem ($\sim 500,000$ parameters) in about a minute. However the computation time depends strongly on the value of $\rho$, as illustrated in Table 2.

# 4 Analysis of cell signalling data

For illustration we analyze a flow cytometry dataset on $p = 11$ proteins and $n = 7466$ cells, from Sachs et al. (2003). These authors fit a directed acyclic graph (DAG) to the data, producing the network in Figure 2.

The result of applying the graphical lasso to these data is shown in Figure 3, for 12 different values of the penalty parameter $\rho$. There is moderate agreement between, for example, the graph for $L_1$ norm $= 0.00496$ and the DAG: the former has about half of the edges and non-edges that appear in the DAG. Figure 4 shows the lasso coefficients as a function of total $L_1$ norm of the coefficient vector.

In the left panel of Figure 5 we tried two different kinds of 10-fold cross-

| $\rho$ | Fraction non-zero | CPU time (sec.) |
|---|---|---|
| 0.01 | .96 | 26.7 |
| 0.03 | .62 | 8.5 |
| 0.06 | .36 | 4.1 |
| 0.60 | .00 | 0.4 |

Table 2: *Timing results for dense scenario, $p = 400$, for different values of the regularization parameter $\rho$. The middle column is the number of non-zero coefficients.*

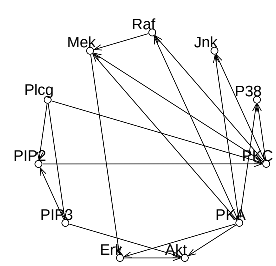


Figure 2: *Directed acylic graph from cell-signaling data, from Sachs et al. (2003).*

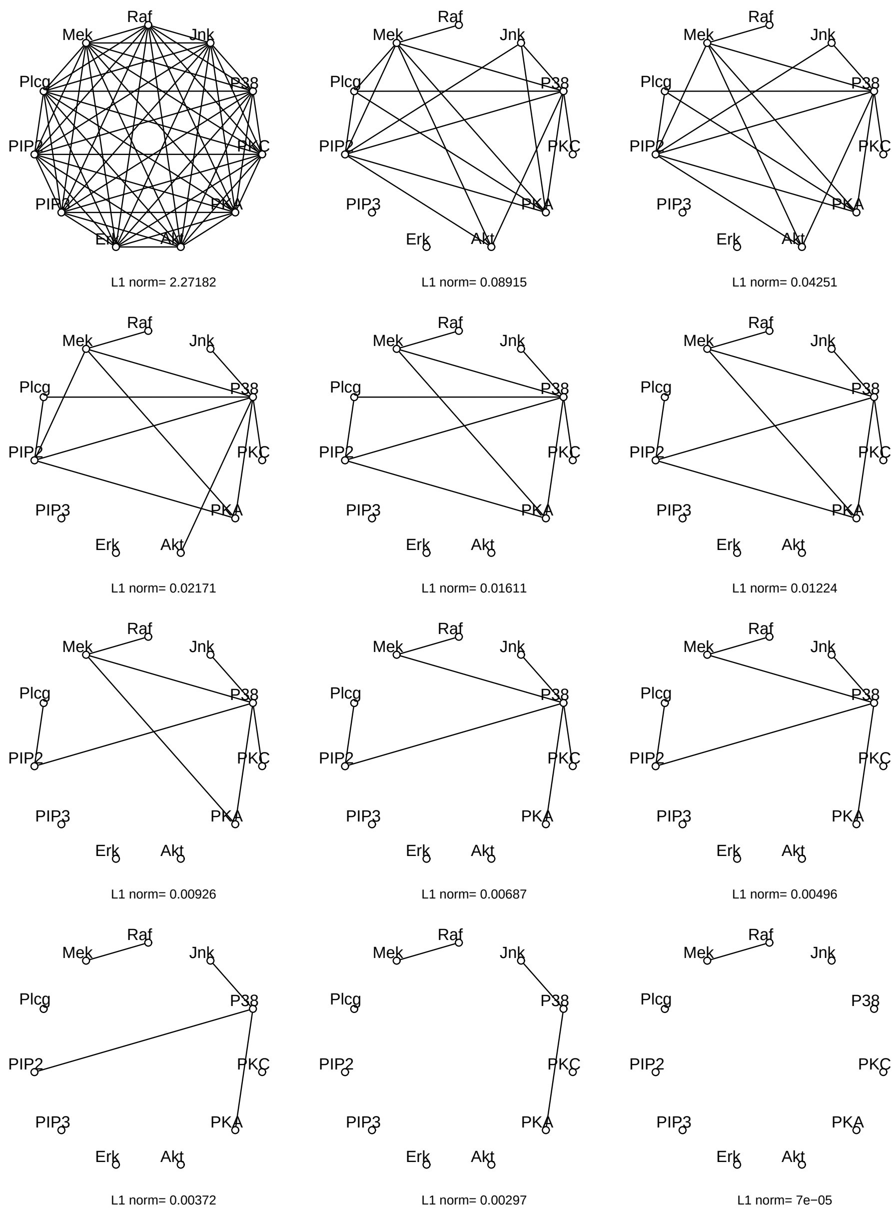


Figure 3: *Cell-signaling data: undirected graphs from graphical lasso with different values of the penalty parameter $\rho$.*

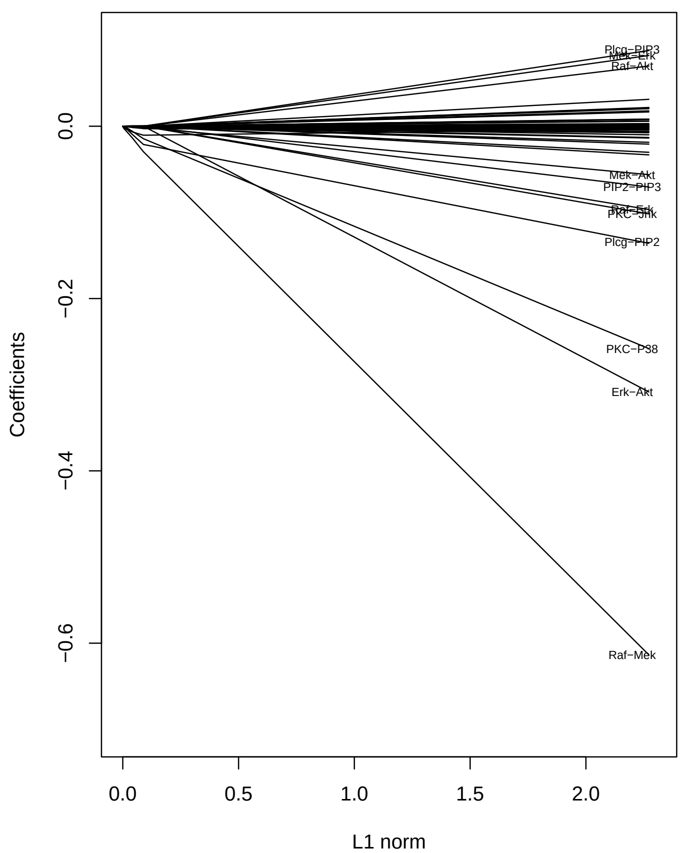


Figure 4: *Cell-signaling data: profile of coefficients as the total $L_1$ norm of the coefficient vector increases, that is, as $\rho$ decreases. Profiles for the largest coefficients are labeled with the corresponding pair of proteins.*

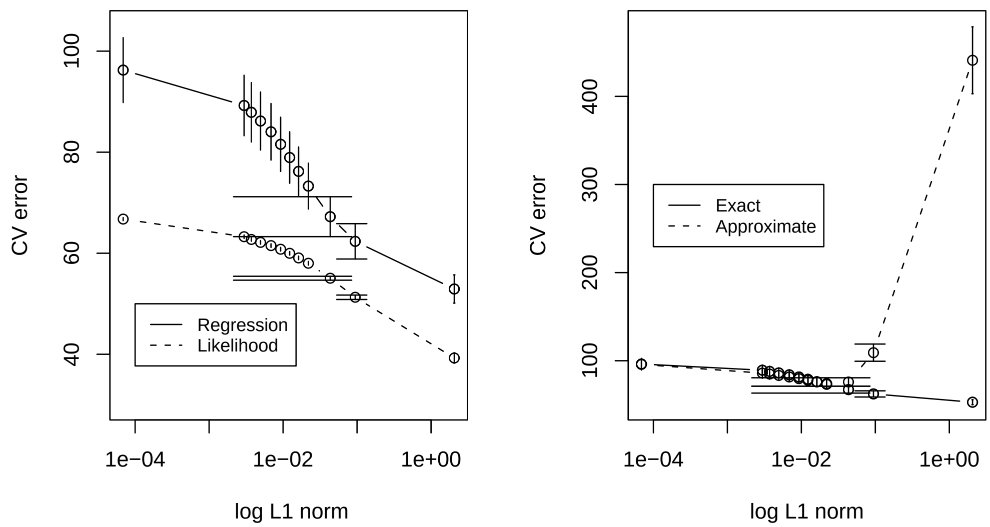


Figure 5: *Cell-signaling data. Left panel shows tenfold cross-validation using both Regression and Likelihood approaches (details in text). Right panel compares the regression sum of squares of the exact graphical lasso approach to the Meinhausen-Buhlmann approximation.*

validation for estimation of the parameter $\rho$. In the "Regression" approach, we fit the graphical lasso to nine-tenths of the data, and used the penalized regression model for each protein to predict the value of that protein in the validation set. We then averaged the squared prediction errors over all 11 proteins. In the "Likelihood" approach, we again applied the graphical lasso to nine-tenths of the data, and then evaluated the log-likelihood (1) over the validation set. The two cross-validation curves indicate that the unregularized model is the best, not surprising given the large number of observations and relatively small number of parameters. However we also see that the likelihood approach is far less variable than the regression method.

The right panel compares the cross-validated sum of squares of the exact graphical lasso approach to the Meinhausen-Buhlmann approximation. For lightly regularized models, the exact approach has a clear advantage.

# 5 Discussion

We have presented a simple and fast algorithm for estimation of a sparse inverse covariance matrix using an $L_1$ penalty. It cycles through the variables, fitting a modified lasso regression to each variable in turn. The individual lasso problems are solved by coordinate descent.

The speed of this new procedure should facilitate the application of sparse inverse covariance procedures to large datasets involving thousands of parameters.

An R language package `glasso` is available on the third author's website.


### Acknowledgments

We thank the authors of Banerjee et al. (2007) for making their COVSEL program publicly available, Larry Wasserman for helpful discussions, and an Editor and two referees for comments that led to improvements in the manuscript. Friedman was partially supported by grant DMS-97-64431 from the National Science Foundation. Hastie was partially supported by grant DMS-0505676 from the National Science Foundation, and grant 2R01 CA 72028-07 from the National Institutes of Health. Tibshirani was partially supported by National Science Foundation Grant DMS-9971405 and National Institutes of Health Contract N01-HV-28183.


# References


Banerjee, O., Ghaoui, L. E. & d'Aspremont, A. (2007), 'Model selection through sparse maximum likelihood estimation', *To appear, J. Machine Learning Research* **101**.

Boyd, S. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

Dahl, J., Vandenberghe, L. & Roychowdhury, V. (2007), Covariance selection for non-chordal graphs via chordal embedding. To appear in Optimization Methods and Software.

Friedman, J., Hastie, T., Hoefling, H. & Tibshirani, R. (2007), 'Pathwise coordinate optimization', *Annals of Applied Statistics, to appear* .

Meinshausen, N. & Bühlmann, P. (2006), ‘High dimensional graphs and variable selection with the lasso’, *Annals of Statistics* **34**, 1436–1462.

Sachs, K., Perez, O., Pe’er, D., Lauffenburger, D. & Nolan, G. (2003), ‘Causal protein-signaling networks derived from multiparameter single-cell data’, *Science* (308 (5721)), 504–6.

Vandenberghe, L., Boyd, S. & Wu, S.-P. (1998), ‘Determinant maximization with linear matrix inequality constraints’, *SIAM Journal on Matrix Analysis and Applications* **19**(2), 499–533.
*citeseer.ist.psu.edu/vandenberghe98determinant.html

Wu, T. & Lange, K. (2007), Coordinate descent procedures for lasso penalized regression.

Yuan, M. & Lin, Y. (2007), ‘Model selection and estimation in the gaussian graphical model’, *Biometrika* **94**(1), 19–35.