Applications of the lasso and grouped lasso to the estimation of sparse graphical models

Jerome Friedman *
Trevor Hastie †
and Robert Tibshirani‡

March 10, 2010

Abstract

We propose several methods for estimating edge-sparse and node-sparse graphical models based on lasso and grouped lasso penalties. We develop efficient algorithms for fitting these models when the numbers of nodes and potential edges are large. We compare them to competing methods including the graphical lasso and SPACE (Peng, Wang, Zhou & Zhu 2008). Surprisingly, we find that for edge selection, a simple method based on univariate screening of the elements of the empirical correlation matrix usually performs as well or better than all of the more complex methods proposed here and elsewhere.

Running title: Applications of the lasso and grouped lasso

1 Introduction

A number of authors have proposed the estimation of sparse undirected graphical models through the use of ℓ_1 (lasso) regularization. The basic

*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

model for continuous data assumes that the observations have a multivariate Gaussian distribution with mean μ and covariance matrix Σ. With this assumption, if the ijth component of Σ^{-1} is zero, then variables i and j are conditionally independent, given the other variables. Moreover, the ij element of Σ^{-1} is, up to a positive scalar, the regression coefficient of variable j in the multiple regression of variable i on the rest, and vice-versa (Hastie et al. 2009, for example). Thus it makes sense to impose an ℓ_1 penalty for the estimation of Σ^{-1}, to impose sparsity.

Meinshausen & Bühlmann (2006) take a simple approach to this problem; they estimate a sparse graphical model by fitting a collection of lasso regression models, using in turn each variable as the response, and 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 Σ^{-1}. Following the approach of Banerjee et al. (2008), Friedman et al. (2007) proposed the *graphical lasso* algorithm which uses the blockwise coordinate descent strategy, fitting a modified lasso problem in each descent step. Their new procedure is extremely simple, and is substantially faster than many competing approaches.

In this paper we propose a symmetrized version of the Meinshausen & Bühlmann (2006) method, and also adapt the grouped lasso method of Yuan & Lin (2007*a*) to the estimation of sparse graphical models. The resulting procedure provides fast approximations to the exact penalized maximum likelihood estimate. In addition, we propose a different penalty which groups all of the edges that are connected to a given node. The resulting graph is sparse not in its edges but in its nodes: that is, some nodes have no edges connecting them to the remaining nodes. In Section 2 we review some existing methods for estimating sparse graphical models and propose some new ones. We carry out a comparative study of the accuracy of these procedures in Section 3, and come to the surprising conclusion that simple correlation-screening is competitive with the best for edge detection. In Section 4 we propose some methods for estimating node-sparse graphs, in contrast to the edge-sparse graphs that are the focus of Section 2. Section 5 discusses the computational complexity of each of the methods, and finally Section 6 contains some discussion.

2 Estimation of sparse undirected graphs

The first approach that we discuss for sparse graphical modelling uses an ℓ_1 (lasso) penalty and was suggested by Yuan & Lin (2007*b*). Suppose that we have N multivariate normal observations of dimension p, with mean μ and covariance Σ. Following Banerjee et al. (2008), let $\Theta = \Sigma^{-1}$, and let S be the empirical covariance matrix, the problem is to maximize the penalized log-likelihood

$$\log\det\Theta - \mathrm{tr}(S\Theta) - \rho||\Theta||_{\ell_1}, \tag{1}$$

over non-negative definite matrices Θ. Here tr denotes the trace and $||\Theta||_{\ell_1}$ is the ℓ_1 norm — the sum of the absolute values of the elements of $\Theta = \Sigma^{-1}$. The first two terms in (1) gives, up to a constant, the Gaussian log-likelihood of the data, partially maximized with respect to the mean parameter μ (also the Wishart log-likelihood for Σ). Friedman et al. (2008) propose the "graphical lasso" procedure for this problem, an efficient implementation of blockwise coordinate descent.

Peng, Wang, Zhou & Zhu (2008) take a symmetric regression approach, called "SPACE", in response to the asymmetry of Meinshausen & Bühlmann (2006). They fit a model of the form:

$$\hat{\mathbf{X}} = \mathbf{X}\hat{\mathbf{B}} \tag{2}$$

where $\mathbf{X}$ is the $N \times p$ data matrix and $\mathbf{B}$ is $p \times p$ with zeros on the diagonal. Assume the rows of $\mathbf{X}$ are multivariate normal, and let β_{ij} be the population regression coefficient of X_i on X_j (in the multiple regression of X_i on the rest). Then

$$\beta_{ij} = \rho_{ij}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}} \tag{3}$$

where ρ_{ij} is a partial correlation and σ^{ii} are the residual variances. $\mathbf{B}$ is filled with these β_{ij}, except the diagonal is zero. The SPACE method reparametrizes β_{ij} in terms of the symmetric ρ_{ij} and σ^{ii}, and then minimizes

$$\tfrac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}||_F^2 + \lambda\sum_{i\neq j}|\rho_{ij}| \tag{4}$$

Their algorithm alternates between estimating the σ^{ii} and the ρ_{ij}. In principle they need also to ensure that $-1 \leq \hat{\rho}_{ij} \leq 1$, but they say that this seems not to be a problem in practice.

2.1 The symmetric lasso procedure

Here we also formulate and implement a method for symmetrizing the Meinshausen-Bühlmann lasso approach. This method is closely related to the SPACE procedure (Peng, Wang, Zhou & Zhu 2008) described above, which was the catalyst for the work in this section.

Recall that if $X = (X_1, X_2, \ldots, X_p)$ has a multivariate Gaussian distribution with mean-vector 0 (for convenience) with covariance Σ, then $\Theta = \Sigma^{-1}$ captures the conditional distributions of each X_j given the rest. Namely

$$X_j | X_{-j} \sim N(\sum_{i \neq j} X_i \beta_{ij}, \sigma^{jj}), \tag{5}$$

where

$$\beta_{ij} = -\frac{\theta_{ij}}{\theta_{jj}} \quad \text{and} \tag{6}$$

$$\sigma^{jj} = \frac{1}{\theta_{jj}}. \tag{7}$$

On the other hand, if we assume that the conditional distribution of each variable on the rest is linear, then if we fill in Θ according to the prescription above, its inverse must be the covariance matrix of the variables.

We can express the negative log-product-likelihood for all these conditional distributions as

$$l(\Theta) = \frac{1}{2} \sum_{j=1}^{p} \left[N \log \sigma^{jj} + \frac{1}{\sigma^{jj}} ||\mathbf{x}_j - \mathbf{X} B_j||_2^2 \right], \tag{8}$$

where B_j is a p-vector with elements β_{ij}, except a 0 in the jth position. This is also known as a *pseudo log likelihood* (Besag 1975).

From the symmetry of Θ this means that $\sigma^{jj}\beta_{ij} = \sigma^{ii}\beta_{ji}$, which is a requirement in this joint linear model for the means. Alternatively we can write

$$l(\tilde{\Theta}) = \frac{1}{2} \sum_{j=1}^{p} \left[N \log \sigma^{jj} + \frac{1}{\sigma^{jj}} ||\mathbf{x}_j + \mathbf{X} \tilde{\Theta}_j \sigma^{jj}||_2^2 \right], \tag{9}$$

where $\tilde{\Theta}$ is symmetric with zero on the diagonal. We propose to estimate a sparse $\tilde{\Theta}$ by solving

$$\min_{\tilde{\Theta}, \{\sigma^{ii}\}_1^p} \tfrac{1}{N} l(\tilde{\Theta}) + \lambda \sum_{i<j} |\tilde{\theta}_{ij}| \quad \text{s.t. } \tilde{\theta}_{ij} = \tilde{\theta}_{ji}. \tag{10}$$

The algorithm we propose is coordinate descent for a decreasing sequence of values for λ on the log scale, starting with a large enough value so that $\tilde{\Theta} = 0$. Details are given in Appendix A.

Given σ^{jj}, $j = 1, \ldots, p$ and for fixed λ, there is a simple coordinate update for $\tilde{\theta}_{ij}$:

$$\tilde{\theta}_{ij} = \frac{S(-(s_{ij} + s_{ji}), \lambda)}{\sigma^{ii} + \sigma^{jj}}, \tag{11}$$

where $s_{ij} = \mathbf{x}_i^T \mathbf{r}_j^{-i}/N$, and $\mathbf{r}_j^{-i}$ is the partial residual for X_i in the regression of X_j on the rest. $S(\cdot, \lambda)$ is the *soft-thresholding* operator. This derivation assumes that the X_j have mean zero and variance 1, but can be modified to accommodate other cases. Given $\tilde{\Theta}$, solving for the σ^{jj} amounts to finding roots of quadratic equations, one for each j. Hence for a fixed λ, iteration would be needed to solve for both $\tilde{\Theta}$ and the σ^{jj}. Full details of both steps are given in Appendix A. We call this procedure the *symmetric lasso*.

As an approximation and potential speedup, we form a path of solutions, and use the σ^{jj} from the previous λ_{k-1} as the values for λ_k. The starting values would be $\sigma^{jj} = 1$ (the estimates when $\tilde{\Theta} = 0$). On the other hand, the estimates might not change much if the exact solutions for the σ^{jj} were iterated at each λ_k. We call this latter procedure the *approximate symmetric lasso*.

2.2 The paired group lasso

In this section we propose another, more direct modification to the Meinshausen-Bühlmann procedure, based on the grouped lasso.

2.2.1 Review of the grouped lasso

Suppose that we have a regression problem with N observations and p features, and an N-vector of outcomes $\mathbf{y}$. Let $\mathbf{X}_j$ be the feature *matrix* for the p_j features in the jth group. Then the grouped lasso minimizes

$$\tfrac{1}{2}||\mathbf{y} - \sum_{j=1}^{J} \mathbf{X}_j \beta_j||^2 + \lambda \sum_{j=1}^{J} ||\beta_j|| \tag{12}$$

where β_j is the coefficient vector for the jth group, and $||\cdot|| = ||\cdot||_{\ell_2}$ is the Euclidean norm. The actual expression in Yuan & Lin (2007*a*) has factors for

different group sizes. In our generalization in the next section, the effective group sizes are equal, so we omit the factors here.

The subgradient equations are

$$-\mathbf{X}_j^T(\mathbf{y} - \sum_k \mathbf{X}_k\beta_k) + \lambda t_j = 0,$$

where $t_j = \beta_j/||\beta_j||$ if $\beta_j \neq 0$, and t_j is a vector with $||t_j|| \leq 1$ otherwise. It is natural to solve these equations by blockwise coordinate descent. We now focus on the solution for one block, holding the other coefficients fixed.

Let $\mathbf{r}_j = \mathbf{y} - \sum_{k\neq j} \mathbf{X}_k\hat{\beta}_k$ be the partial residual for the jth group, and let $s_j = \mathbf{X}_j^T\mathbf{r}_j$. If $||s_j|| \leq \lambda$ then the solution for $\hat{\beta}_j$ is zero; otherwise the solution satisfies

$$\hat{\beta}_j = (\mathbf{X}_j^T\mathbf{X}_j + \frac{\lambda}{||\beta_j||}\mathbf{I})^{-1}\mathbf{X}_j^T\mathbf{r}_j. \tag{13}$$

This is like a ridge regression, with the ridge parameter depending on $||\beta_j||$.

As shown by Yuan & Lin (2007*a*), if $\mathbf{X}_j^T\mathbf{X}_j = \mathbf{I}$, then this solution has a simple form in terms of soft-thresholded least squares estimates:

$$\hat{\beta}_j = (1 - \lambda/||s_j||)s_j. \tag{14}$$

In the general case, a scalar equation can be derived for $||\beta_j||$ from (13); then substituting into the right-hand side of (13) gives the solution. However, this can lead to an unstable algorithm because of potential division by small norms. Instead we find that coordinate descent within the block is more stable; details are given in Friedman et al. (2010).

Yuan & Lin (2007*a*) assume this blockwise-orthonormality to simplify the computational procedure.

2.2.2 Application to sparse graph estimation—the paired grouped lasso

Here we propose a different method for sparse graph estimation that uses the grouped lasso. Assume the columns of $\mathbf{X}$ are standardized to have mean zero and unit norm. We start with the regression model (2), and solve

$$\min_{\mathbf{B}} \tfrac{1}{2}||\mathbf{X} - \mathbf{XB}||_F^2 + \lambda \sum_{j<i} ||(\beta_{ij}, \beta_{ji})|| \tag{15}$$

with the diagonal elements β_{ii} of $\mathbf{B}$ zero.

The gradient equations are:

$$\begin{aligned} -\mathbf{x}_i^T(\mathbf{x}_j - \sum_{k\neq j}\beta_{kj}\mathbf{x}_k) + \lambda\frac{\beta_{ij}}{||(\beta_{ij},\beta_{ji})||} &= 0 \\ -\mathbf{x}_j^T(\mathbf{x}_i - \sum_{k\neq i}\beta_{ki}\mathbf{x}_k) + \lambda\frac{\beta_{ji}}{||(\beta_{ij},\beta_{ji})||} &= 0 \end{aligned} \tag{16}$$

Let $r_{ij} = \mathbf{x}_i^T(\mathbf{x}_j - \sum_{k\neq i}\beta_{kj}\mathbf{x}_k)$ and $r_{ji} = \mathbf{x}_j^T(\mathbf{x}_i - \sum_{k\neq j}\beta_{ki}\mathbf{x}_k)$, each regression coefficients in light of the normalization of $\mathbf{X}$. Then it can be shown that the solutions $(\hat{\beta}_{ij},\hat{\beta}_{ji}) = 0$ if $||(r_{ij},r_{ji})|| < \lambda$ and otherwise we have

$$(\hat{\beta}_{ij},\hat{\beta}_{ji}) = \left(1 - \frac{\lambda}{||(r_{ij},r_{ji})||}\right)(r_{ij},r_{ji}). \tag{17}$$

Thus the algorithm cycles through all symmetric pairs (β_{ij},β_{ji}), either setting them to zero or soft-thresholding them as in (17). We call this the *paired group lasso*.

If θ_{ij} is the ijth element of the multivariate Gaussian inverse covariance matrix, then as in (6) and (7) $\beta_{ij} = -\theta_{ij}\sigma^{jj}$ and hence

$$||(\beta_{ij},\beta_{ji})|| = |\theta_{ij}|\sqrt{\sigma^{ii2} + \sigma^{jj2}}$$

Thus $||(\beta_{ij},\beta_{ji})|| = 0 \leftrightarrow |\theta_{ij}| = 0$. The penalty is just a weighted lasso for the parameters θ_{ij}. The weights make sense: pairs i,j with larger residual variances get larger penalty weights.

Alternatively, from (3)

$$||(\beta_{ij},\beta_{ji})|| = |\rho_{ij}|\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}} + \frac{\sigma^{ii}}{\sigma^{jj}}} \tag{18}$$

where ρ_{ij} is the partial correlation between variables i and j. Hence the paired-group-lasso penalty is also a weighted lasso for the parameters ρ_{ij}; compare with the SPACE criterion (4).

2.3 Timing comparisons

In this section we compare the aforementioned procedures in a small simulation study. There are 3 scenarios with varying sample size N and number of

Method	$N = 500$ $p = 500$	$N = 100$ $p = 1000$	$N = 1000$ $p = 100$
Graphical Lasso	183.8	767.5	0.40
Meinshausen-Bühlmann Lasso	12.3	245.0	0.15
Paired Group Lasso	3.0	32.0	0.03
SPACE	416.3	280.6	3.70
Symmetric Lasso	10.3	66.0	0.06
Symmetric Lasso—approximate	7.0	42.3	0.02

Table 1: *Timings for six different methods on three problems. Timings are in seconds, averaged over 3 runs.*

variables p, as shown in Table 1. In the first scenario the data were generated from $N(0, \Sigma)$ and about 20% of the entries of Σ were non-zero. In the other two scenarios, we chose $\Sigma = I$. Timings were computed over 30 values of the corresponding regularization parameter, and the range of regularization parameters was chosen so that each method produced approximately the same number of non-zero estimates. The convergence threshold was chosen to be 0.001 for all methods, except for SPACE, which does not offer control of this parameter. All programs were coded in double precision Fortran, called from the R language, except for SPACE which was coded in C and R. This latter program did its loop over regularization parameters in R, which puts it at a slight speed disadvantage. The results shown in Table 1 show that the paired group lasso, symmetric lasso, and approximate symmetric lasso procedures are much faster than the competitors. By coupling together the models for each symmetric pair (β_{ij}, β_{ji}), they both achieve speedups over the simple lasso (Meinshausen-Bühlmann) approach. We do not know why SPACE is so slow in our experiments. In principle it should have similar speed to the symmetric lasso; perhaps the extensive use of updating formulas in our implementation produced a substantial gain in efficiency. In the next section we study the accuracy of these methods and some others, for edge detection in sparse graphs.

3 A comparative study of accuracy

3.1 Data generation

Here we used $p = 400$ variables and $N = 200$ observations generated from a multivariate normal distribution with mean zero and covariance matrix $\Sigma = \Theta^{-1}$. The inverse matrix Θ was taken to be very sparse with approximately p out of the $p(p-1)/2$ off–diagonal elements having non zero value $\theta_{ij} = \theta$, and the rest being zero valued.

The non zero elements of Θ had three patterns: random, hubs, and cliques.

Random. Here each off–diagonal element was randomly set to $a \neq 0$ with probability 0.005 resulting in 447 non zero elements.

Hubs. The rows/columns are partitioned into disjoint groups $\{G_k\}_1^K$ each associated with a "central" row k in that group. The non zero off–diagonal elements of Θ are taken to be $\theta_{ik} = \theta$ for $i \in G_k$ and $\theta_{ik} = 0$ otherwise. Here there were $K = 20$ groups each with 20 members resulting in 380 non zero off–diagonal elements of Θ.

Cliques. The rows are partitioned into disjoint groups and $\theta_{ij} = \theta$ $(i \neq j)$ for $i, j \in G_k$. Here there were 20 groups each with 7 members resulting in 420 off–diagonal elements of Θ.

3.2 Performance measures

The goal of the exercise is to correctly identify the non zero elements of Θ given the empirical correlation matrix $\hat{\Sigma}$ derived from multivariate normal data ($p = 400$, $N = 200$) generated from $\Sigma = \Theta^{-1}$. Suppose there are nz nonzero off-diagonal elements in Θ, and $z = p(p-1)/2 - nz$ zero elements. The figure of merit we use is the fractional area under the ROC curve starting from zero false positives up to nz false positives, relative to perfect classification (all true positives correctly identified before any false positives). Specifically,

$$\mathrm{AUC}_f = \frac{\int_0^{nz/z} t(f)\, df}{nz/z} \tag{19}$$

where f is the false positive rate specifying a point on the curve and $t(f)$ is the true positive rate at f. By construction $\mathrm{AUC}_f = 1$ for perfect selection ($t(f) = 1\ \forall f > 0$). For random selection of positives ($t(f) = f$), $\mathrm{AUC}_f = nz/z$.

3.3 Methods

Six methods were considered; in addition to the methods discussed earlier, we consider a few simple alternative approaches. The *univariate correlation* method simply ranks the off–diagonal elements of the empirical correlation matrix $\hat{\Sigma}$ on their absolute values in descending order and identifies nonzero elements of Θ in that order. For the graphical lasso, the Meinshausen & Bühlmann (2006) approach using the AND criterion, the symmetric lasso, and the paired group lasso, the positives are identified in the order they become non zero as the regularization parameter λ is relaxed from $\infty \geq \lambda \geq 0$.

The *statewise* approach is derived from the symmetric lasso log-likelihood criterion (8). Here each successive element is identified to be non zero in turn as the one whose corresponding component of the gradient of the log-likelihood is largest in absolute value. The log-likelihood is then minimized with respect to all non zero elements including the newly added one. *Statewise* is an approximation to forward statewise regression and the most aggressive selector among those being considered. The least aggressive is *univariate correlation* since it corresponds to the symmetric elastic net using the ridge penalty

$$\frac{1}{2}\ell(\Theta) + \lambda \sum_{i \neq j} (\theta_{ij}^2 + \epsilon \cdot |\theta_{ij}|) \text{ with } \epsilon \to 0. \tag{20}$$

3.4 Results and Summary

Tables 2–5 show the average values of AUC_f (19) for each of the six methods for different configurations of the inverse matrix Θ as averaged over 20 trials. The quantities in parentheses are the standard deviation of the mean as estimated over the 20 trials. The caption for each table summarizes the results found for that configuration.

In terms of speed *univariate correlation* is by far the fastest method followed by *paired group lasso* and *symmetric lasso* which are considerably slower. *Graphical lasso* is by far the slowest method with *statewise* and *Meinshausen & Bühlmann* being somewhat faster than *graphical lasso*.

Method	AUC_f	(std. error)
univariate correlation	0.554	(0.0051)
graphical lasso	0.558	(0.0051)
Meinshausen & Bühlmann	0.555	(0.0050)
symmetric lasso	0.550	(0.0051)
paired group lasso	0.550	(0.0051)
statewise	0.494	(0.0049)

Table 2: RANDOM. Results for the random configuration with the 447 randomly selected non zero elements in Θ set to $\theta = -0.2$ resulting in positive correlations in $\Sigma = \Theta^{-1}$. The results for $\theta = 0.2$ (negative correlations) are the same within uncertainty. Here one sees that all methods except *statewise* do equally well.

Method	AUC_f	(std. error)
univariate correlation	0.700	(0.0065)
graphical lasso	0.704	(0.0067)
Meinshausen & Bühlmann	0.710	(0.0068)
symmetric lasso	0.609	(0.0060)
paired group lasso	0.622	(0.0061)
statewise	0.409	(0.0048)

Table 3: HUB. Results for the 20×20 hub configuration where each of the 380 non zero elements was set to $\theta = -0.175$. Results for positive values $\theta = 0.175$ are again the same within uncertainty. Here *univariate correlation*, *graphical lasso* and *Meinshausen & Bühlmann* do equally well, *symmetric lasso* and *paired group lasso* are somewhat inferior, and *statewise* has the worst performance.

Thus, the dominating methods over the situations considered here are *univariate correlation* and *statewise*. The former provides the best performance, or very close to it, in all settings except positive cliques where *statewise* dominates. The positive clique setting might be considered somewhat pathological, in that all coefficients in a clique are positive, but at the same time, the positive variables within each clique are all negatively correlated with each other. The *statewise* method wins in that case because it is the most aggressive.

Method	AUC_f	(std. error)
univariate correlation	0.409	(0.0082)
graphical lasso	0.392	(0.0077)
Meinshausen & Bühlmann	0.339	(0.0064)
symmetric lasso	0.324	(0.0063)
paired group lasso	0.323	(0.0064)
statewise	0.186	(0.0025)

Table 4: CLIQUE (−). Results for the 20×7 clique configuration with each of the 420 non zero elements set to $\theta = -0.1$ (positive correlations). Here the comparative performance of the methods is similar to that of the random configuration (Table 2) except that *Meinshausen & Bühlmann* is here similar to *paired group lasso* and *symmetric lasso*.

Method	AUC_f	(std. error)
univariate correlation	0.146	(0.0030)
graphical lasso	0.146	(0.0030)
Meinshausen & Bühlmann	0.159	(0.0032)
symmetric lasso	0.156	(0.0032)
paired group lasso	0.156	(0.0032)
statewise	0.664	(0.0086)

Table 5: CLIQUE (+). Results for 20×7 cliques with the 420 non zero elements set to $\theta = 0.5$ (negative correlations). Here *statewise* dominates with the other methods being equally inferior.

4 Estimation of node-sparse graphs

In this section we propose several methods for estimation of graphs that are sparse in a different way than those described earlier. In the previous section we considered methods that deleted edges from the graph. Here we propose methods for deleting all of the edges that connect to a given node, by applying the grouped-lasso penalty to entire rows and columns of the correlation matrix.

4.1 The graphical grouped lasso

Here we assume the same notation as in (1) in Section 2, and propose maximizing the penalized log-likelihood

$$J(\Theta) = \log\det\Theta - \mathrm{tr}S\Theta - \lambda \sum_i ||\Theta_{-i,i}||_{\ell_2} \tag{21}$$

over non-negative definite matrices Θ. Here $||\Theta_{-i,i}||_{\ell_2}$ is a group-lasso penalty applied to the ith row of Θ, but omitting the diagonal element. By the symmetry of Θ, it applies to the ith column as well. Expression (21) is the penalized Gaussian log-likelihood of the data, partially maximized with respect to the mean parameter μ.

Let $W = \hat{\Theta}^{-1}$ be the solution to this convex optimization problem, and θ_i be the ith row of Θ with θ_{ii} set to zero.

The subgradient equation is

$$W - S - (\lambda/2)R$$

where the components of the $p \times p$ matrix R are defined by $r_{ij} = (u_i)_j + (v_j)_i$, $u_i = d||\theta_i||/d\theta_i = (\theta_i/||\theta_i||)$ if $\theta_i \neq 0$ and $||u_i|| < 1$ otherwise; $v_j = d||\theta_j||/d\theta_j = (\theta_j/||\theta_j||)$ if $\theta_j \neq 0$ and $||v_j|| < 1$ otherwise.

The subgradient equation for one row/col can be written as

$$w_{i,-i} - s_{i,-i} - (\lambda/2)\theta_{i,-i}\{1/||\theta_{i,-i}|| + 1/||\theta_{j,-j}||\}_{j\neq i} = 0 \tag{22}$$

Using the relation $W_{-i,-i}\theta_{i,-i} + w_{i,-i}\theta_{ii} = 0$, this can be written as

$$W_{-i,-i}\beta_{i,-i} - s_{i,-i} + (\lambda/2)\beta_{i,-i}\{1/||\beta_{i,-i}|| + \frac{\theta_{i,i}}{\theta_{j,j}}(1/||\beta_{-j,j}||)\}_{j\neq i} = 0 \tag{23}$$

(1) and (23) are equivalent if we set $w_{i,-i} = W_{ii}\beta_{i,-i}$, since $\theta_{i,-i} = -\hat{\beta}_{i,-i}\theta_{ii}$ and thus $\frac{\beta_{i,-i}}{||\beta_{i,-i}||} = -\frac{\theta_{i,-i}}{||\theta_{i,-i}||}$ It turns out that the solution to (23) may have $\beta_{-j,j} \approx 0$ so we need to check for this explicitly. The algorithm is detailed below.

Graphical Grouped Lasso Algorithm

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

2. For each $i = 1, 2, \ldots p, 1, 2, \ldots p, \ldots$, iteratively solve the equation
$$(W_{-i,-i} + (\lambda/2) \cdot D)\beta_{i,-i} = s_{i,-i}$$
for $\beta_{i,-i}$, where $D = \text{diag}(\{1/||\beta_{i,-i}|| + \frac{\theta_{i,i}}{\theta_{j,j}}(1/||\beta_{-j,j}||)\}_{j \neq i})$. Let the overall solution matrix be $\hat{\Theta}$. Let $\hat{\Theta}_0$ be the solution with $\theta_{-i,i} = \theta_{-i,i} = 0$. If $J(\hat{\Theta}_0) < J(\hat{\Theta})$, set $\beta_{-i,i} = 0$
Fill in the corresponding row and column of W using $w_{ii} = W_{-i,-i}\hat{\beta}_{-i,-i}$.

3. Continue until convergence

Unfortunately this algorithm is slow for large problems, since it requires an iterative solution for each row and column. For this reason we explore next some alternative models.

4.2 Principal components: *edge-in* model

This approach is a regression-based method that estimates a kind of sparse principal components. Using the notation of Section 2, we assume

$$\hat{\mathbf{X}} = \mathbf{X}\hat{\mathbf{B}} \tag{24}$$

where $\mathbf{X}$ is $N \times p$ and $\mathbf{B}$ is $p \times p$ with zeros on the diagonal. We minimize

$$\tfrac{1}{2}||\mathbf{X} - \mathbf{X}\mathbf{B}||_F^2 + \lambda \sum_{j=1}^{p} ||\beta_{j,j}|| \tag{25}$$

with $\beta_{j,j}$ the jth column of $\mathbf{B}$ and $\beta_{jj} = 0$ for all j. Since this model predicts each column $\mathbf{x}_j$ from the other columns, the grouping defined by the penalty consists of all coefficients for a given response variable. That is, it penalizes all edges pointing into a given node j, hence we call it the "edge-in" model.

To optimize this criterion, we apply the regression version of grouped lasso using each column of $\mathbf{X}$ as a response. This has one group $\mathbf{X}_{-j}$ ($\mathbf{X}$ with the jth column removed) and response $\mathbf{x}_j$. The condition for a column to be zero is $||\mathbf{X}_{-j}^T \mathbf{x}_i|| < \lambda$. Unfortunately this again requires an iterative solution for each row.

4.3 Principal components: *edge-out* model

Here we minimize

$$\frac{1}{2}||\mathbf{X} - \mathbf{XB}||_F^2 + \lambda \sum_{i=1}^{p} ||\beta_{i,-i}|| \tag{26}$$

with $\beta_{i,-i}$ the ith row of $\mathbf{B}$ and $\beta_{ii} = 0$ for all i. This model penalizes all edges coming from a given node i. A related idea is explored in Peng, Bergamaschi, Han, Noh, Pollack & Wang (2008), where it is applied to two different sets of genomic measurements.

It turns out that there is a simple, fast algorithm for the edge-out model, due to the implicit orthogonality between the different outcome variables for a given predictor i. The gradient equations are:

$$-\mathbf{x}_i^T(\mathbf{x}_j - \sum_{k \neq i} \beta_{kj}\mathbf{x}_k) + \lambda \frac{\beta_{i,-i}}{||\beta_{i,-i}||} = 0 \tag{27}$$

for $i, j = 1, \ldots p$. Let $r_{i,-i} = \mathbf{x}_i^T(\mathbf{x}_j - \sum_{k \neq i} \beta_{kj}\mathbf{x}_k)$, Then $\beta_{i,-i} = 0$ if $||r_{i,-i}|| < \lambda$ and otherwise we have $\beta_{i,-i} = (1 - \lambda/||r_{i,-i}||)r_{i,-i}$, Thus we simply cycle through rows $i = 1, 2, \ldots, p, 1, 2, \ldots$, zeroing out or soft-thresholding the non-diagonal elements of that row. The relevant quantities can be updated to speed up the computation.

4.4 Examples

Among these three procedures, we have implemented a fast version of just the edge-out algorithm. For the three problems of Table 1, total elapsed time for the edge-out procedure over a path of 30 λ values were 6.3, 12.1, and 0.1 seconds, respectively. Hence its speed is competitive with the fastest methods for the sparse graph problem.

We next apply the graphical grouped lasso and edge-out model to a flow cytometry dataset on $p = 11$ proteins and $n = 7466$ cells, from Sachs et al. (2003). The results for the graphical grouped lasso and the edge-out model are shown in Figures 1 and 2 respectively. We see that the graphical grouped lasso has a very narrow range of sparsity, as λ varies, while the edge-out model seems to produce more potentially interpretable groups. The latter graphs might suggest a controlling role for proteins P38 and PKC.

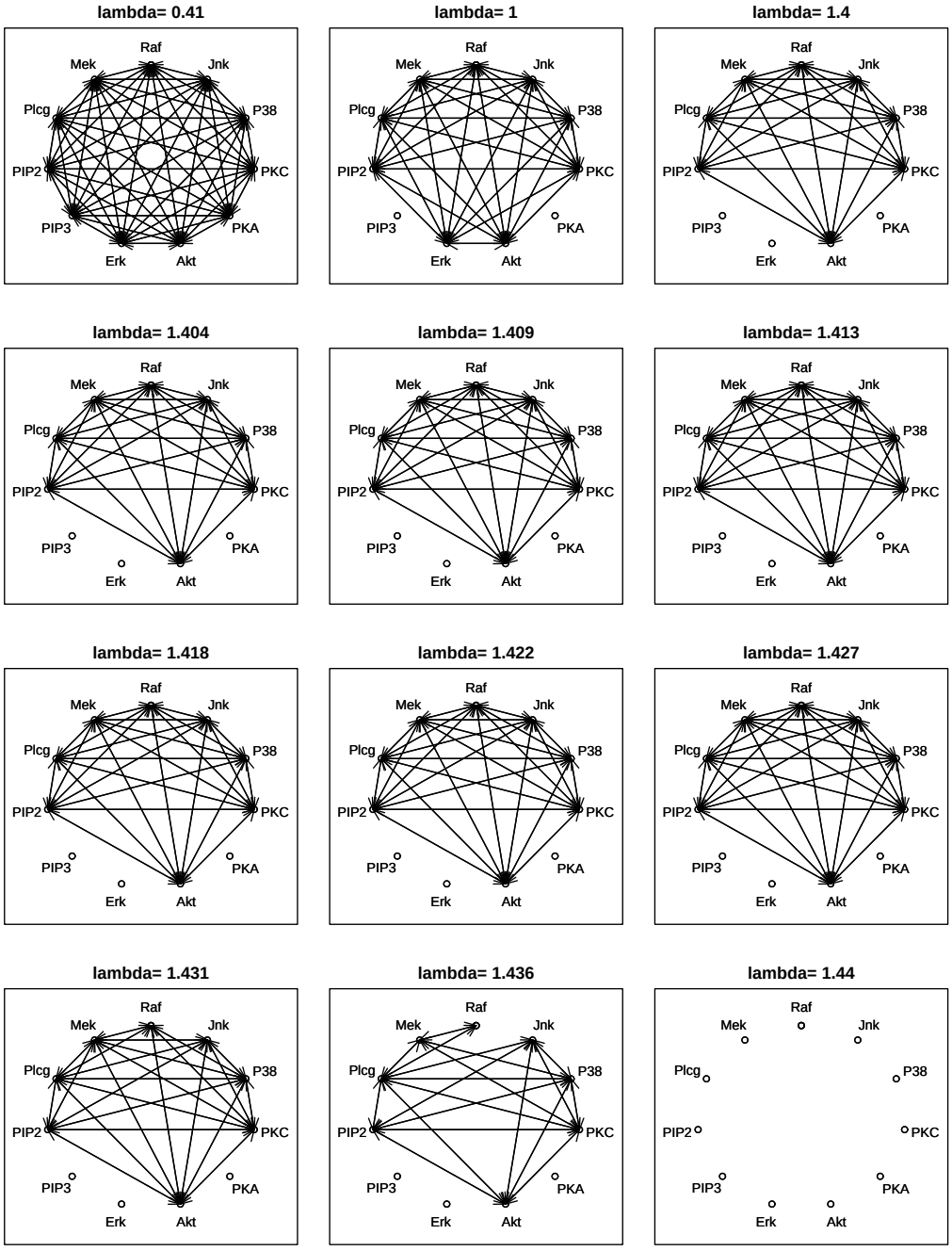

Figure 1: *Networks derived from protein data using the graphical grouped lasso.*

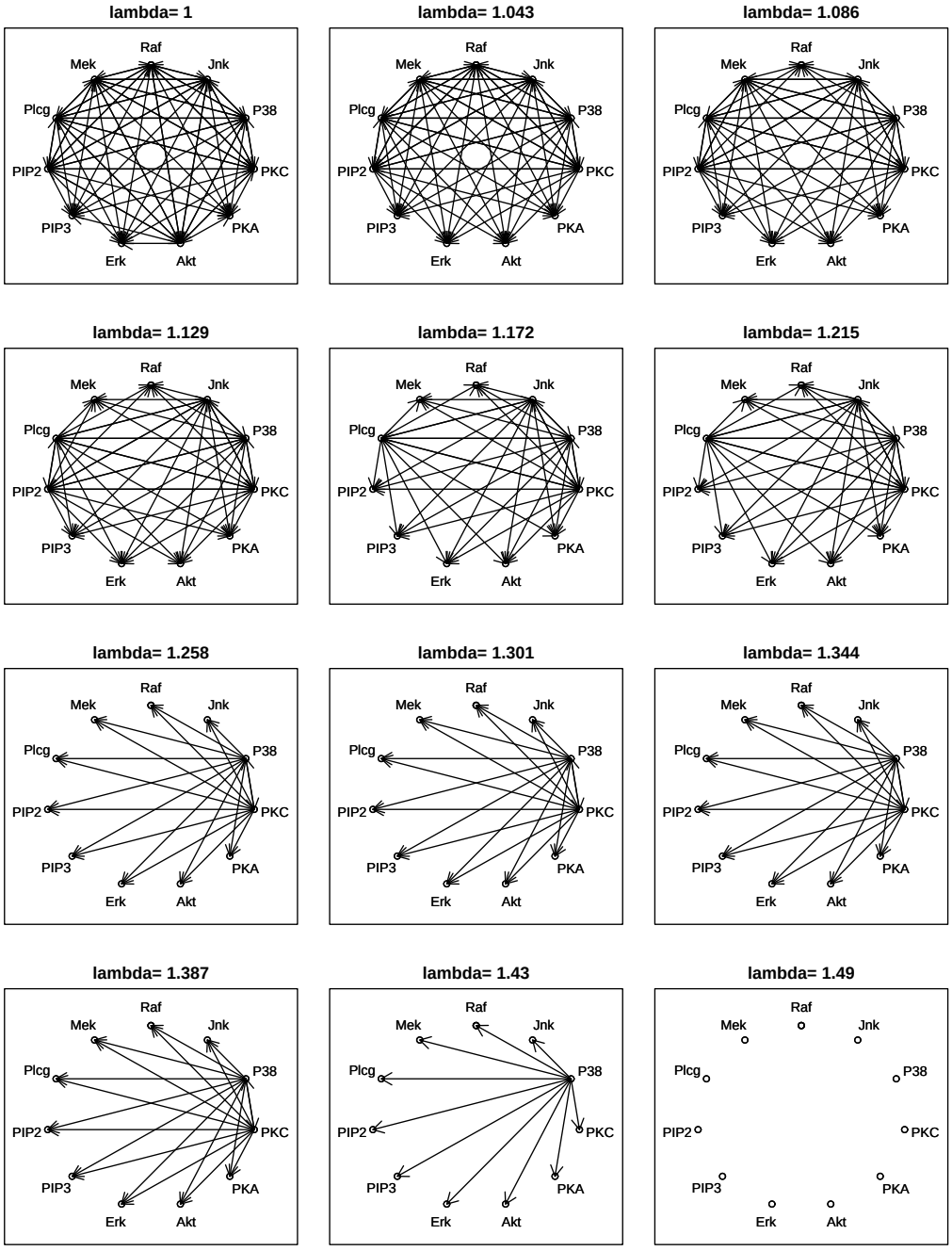

Figure 2: *Networks derived from protein data using the edge-out model.*

Algorithm	Sparsity Type	Computational Complexity	
		Correlation	Data matrix
1. Graphical lasso	Edge-sparse	$O(p^3)$	$O(p^3)$
2. Symmetric graphical lasso	Edge-sparse	$O(p^2) + O(kp)$	$O(p^2N) + O(kN)$
3. Paired group lasso	Edge-sparse	$O(p^2) + O(kp)$	$O(p^2N) + O(kN)$
4. Graphical grouped lasso	Node-sparse	$O(p^3)$	
5. Edge-out	Node-sparse	$O(p^2) + O(kp^2)$	$O(p^2N) + O(kNp)$

Table 6: *Summary of the algorithms proposed in this paper, along with their computational scaling. n is the number of observations, p is the number of variables, and k is the number of non-zero variables in the estimated model.*

5 Computational complexity

We summarize in Table 6 the new procedures proposed in this paper. For two of the methods we have implemented separate versions, that take as input either a $p \times p$ correlation matrix or an $N \times p$ data matrix.

In procedures 2,3 and 5, updating formulae are used to great advantage, dramatically reduce the computation. These three procedures will be added to the R graphical lasso package `glasso` and will be available in the CRAN collection.

6 Discussion

In paper we have proposed some new techniques for the estimation of edge-sparse and node-sparse graphical models based on lasso and grouped lasso penalties. In the edge-sparse setting, there was a surprising finding: we presented a simulation study showing that for detection of the presence or absence of edges, a simple method based on univariate screening of the correlations performs as well or better than all of the more complex competing methods. However for estimation of the correlation matrix and its inverse, only the graphical lasso produces positive definite estimates of both matrices. SPACE and the symmetric lasso yield estimates of the inverse correlation matrix that are usually positive definite: an additional $O(p^3)$ operation would be required to obtain an estimate of the correlation matrix.

Acknowledgments

We thank the authors of Peng, Wang, Zhou & Zhu (2008) for making their SPACE program publicly available, and helpful discussion about these problems. 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.

A Derivation of symmetric lasso

We calculate the gradient of $\ell(\tilde{\Theta})$ in (8) with respect to $\tilde{\theta}_{ij}$. Bear in mind that both $\beta_{ij} = -\tilde{\theta}_{ij}\sigma^{jj}$ and $\beta_{ji} = -\tilde{\theta}_{ji}\sigma^{ii}$ involve $\tilde{\theta}_{ij}$ (by the symmetry), so the gradient is

$$\begin{aligned}
\frac{\partial \ell(\tilde{\Theta})}{\tilde{\theta}_{ij}} &= \frac{\partial \frac{1}{2\sigma^{jj}}||\mathbf{x}_j + \mathbf{X}\tilde{\Theta}_j\sigma^{jj}||_2^2}{\partial \tilde{\theta}_{ij}} + \frac{\partial \frac{1}{2\sigma^{ii}}||\mathbf{x}_i + \mathbf{X}\tilde{\Theta}_i\sigma^{ii}||_2^2}{\partial \tilde{\theta}_{ji}} && (28)\\
&= \mathbf{x}_i^T(\mathbf{x}_j + \sum_{\ell \notin [i,j]} \mathbf{x}_\ell \tilde{\theta}_{\ell j}\sigma^{jj}) + \mathbf{x}_j^T(\mathbf{x}_i + \sum_{\ell \notin [i,j]} \mathbf{x}_\ell \tilde{\theta}_{\ell i}\sigma^{ii}) && (29)\\
&= x_i^T \mathbf{r}_j^{-i} + N\tilde{\theta}_{ij}\sigma^{jj} + x_j^T \mathbf{r}_i^{-j} + N\tilde{\theta}_{ji}\sigma^{ii} && (30)\\
&= N \cdot \left[s_{ij} + s_{ji} + (\sigma^{jj} + \sigma^{ii})\tilde{\theta}_{ij} \right]. && (31)
\end{aligned}$$

Define

$$C(\tilde{\Theta}) = \frac{1}{N}\ell(\tilde{\Theta}) + \lambda \sum_{i<j} |\tilde{\theta}_{ij}|. \quad (32)$$

Hence

$$\frac{\partial C(\tilde{\Theta})}{\partial \tilde{\theta}_{ij}} = s_{ij} + s_{ji} + (\sigma^{jj} + \sigma^{ii})\tilde{\theta}_{ij} + \lambda \cdot \mathrm{Sign}(\tilde{\theta}_{ij}). \quad (33)$$

Setting (33) equal to zero results in the soft-thresholding in (11). The term

s_{ij} can be written as

$$\begin{aligned} s_{ij} &= \mathbf{x}_i^T \mathbf{r}_j^{-i}/N && (34)\\ &= c_{ij} + \sum_{\substack{\ell=1\\ \ell\neq i}}^{p} c_{i\ell}\tilde{\theta}_{\ell j}\sigma^{jj} && (35)\\ &= c_{ij} + \sum_{\ell=1}^{p} c_{i\ell}\tilde{\theta}_{\ell j}\sigma^{jj} - \tilde{\theta}_{ij}\sigma^{jj} && (36)\\ &= \mathbf{x}_i^T \mathbf{r}_j/N - \tilde{\theta}_{ij}\sigma^{jj} && (37)\\ &= c_{ij} + z_{ij}\sigma^{jj} - \tilde{\theta}_{ij}\sigma^{jj}. && (38) \end{aligned}$$

Here $c_{ij} = \mathbf{x}_i^T \mathbf{x}_j/N$ is the ijth entry of the correlation matrix, and we have defined the type of fitted value $z_{ij} = \sum_{\ell=1}^{p} c_{i\ell}\tilde{\theta}_{\ell j}$. Note that the $\tilde{\theta}_{ij}s$ in (34)–(37) are the *old* values, whereas in (31) it is the *new* value about to be updated. Hence each time a coefficient changes, we have to update $r_{\ell j}$ for all ℓ ($O(p)$ operations).

Given $\tilde{\Theta}$, we can estimate σ^{jj} again by minimizing (32), separately for each j.

Hence we solve

$$\min_{\sigma} \log\sigma + \frac{1}{N\sigma}||\mathbf{x}_j + \mathbf{X}\tilde{\Theta}_j\sigma||_2^2. \tag{39}$$

Expanding the RHS we get

$$\begin{aligned} & \log\sigma + \frac{1}{\sigma}\left[c_{jj} + \frac{2}{N}\langle \mathbf{x}_j, \mathbf{X}\tilde{\Theta}_j\rangle\sigma + \frac{1}{N}||\mathbf{X}\tilde{\Theta}_j||_2^2\sigma^2\right] && (40)\\ = \;& \log\sigma + \frac{1}{\sigma} + C + q_j\sigma, && (41) \end{aligned}$$

where

$$q_j = \sum_{\ell=1}^{p}\sum_{\ell'=1}^{p} c_{\ell\ell'}\ \tilde{\theta}_{\ell j}\tilde{\theta}_{\ell' j} \tag{42}$$

Setting the derivative to zero we get

$$\frac{1}{\sigma} - \frac{1}{\sigma^2} + q_j = 0, \tag{43}$$

or

$$q_j\sigma^2 + \sigma - 1 = 0, \tag{44}$$

with only possible solution

$$\sigma^{jj} = \frac{-1 + \sqrt{1 + 4q_j}}{2q_j}. \tag{45}$$

Since from (38) $z_{ij} = \sum_{\ell=1}^{p} c_{i\ell}\tilde{\theta}_{\ell j}$, we see that q_j is given by

$$q_j = \sum_{\ell=1}^{p} \tilde{\theta}_{\ell j} z_{\ell j}. \tag{46}$$

Updating

When $\tilde{\theta}_{ij}$ changes, the jth column of $\{z_{\ell j}\}$ changes:

$$z_{\ell j} \leftarrow z_{\ell j} + c_{i\ell}\Delta_{ij}, \tag{47}$$

where $\Delta_{ij} = \tilde{\theta}_{ij}^{new} - \tilde{\theta}_{ij}^{old}$. This is an $O(p)$ operation, although it occurs only k times, where k is the number of non-zero $\tilde{\theta}_{ij}$s.

From the symmetry of c_{ij}, and the definitions of z_{ij} and q_j, it can be shown that the change in q_j can be computed in $O(1)$ operations:

$$q_j \leftarrow q_j + 2z_{ij}\Delta_{ij} + \Delta_{ij}^2. \tag{48}$$

References

Banerjee, O., Ghaoui, L. E. & d'Aspremont, A. (2008), 'Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data', *Journal of Machine Learning Research* **9**, 485–516.

Besag, J. (1975), 'Statistical analysis of non-lattice data', *The Statistician* **24**(3).

Friedman, J., Hastie, T., Hoefling, H. & Tibshirani, R. (2007), 'Pathwise coordinate optimization', *Annals of Applied Statistics* **2**(1), 302–332.

Friedman, J., Hastie, T. & Tibshirani, R. (2008), 'Regularization paths for generalized linear models via coordinate descent', *Submitted* .

Friedman, J., Hastie, T. & Tibshirani, R. (2010), A note on the group lasso and the sparse group lasso, Technical report, Statistics Department, Stanford University.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning: Prediction, Inference and Data Mining*, second edn, Springer Verlag, New York.

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

Peng, J., Bergamaschi, A., Han, W., Noh, D.-Y., Pollack, J. & Wang, P. (2008), 'Regularized multivariare regression for identifying master predictors with application to intregative genomics study of breast cancer', *submitted* .

Peng, J., Wang, P., Zhou, N. & Zhu, J. (2008), 'Partial correlation estimation by joint sparse regression models', *submitted* .

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.

Yuan, M. & Lin, Y. (2007*a*), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society, Series B* **68**(1), 49–67.

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