Electronic Journal of Statistics
Vol. 6 (2012) 2125–2149
ISSN: 1935-7524
DOI: 10.1214/12-EJS740

The graphical lasso: New insights and alternatives

Rahul Mazumder[*]

Massachusetts Institute of Technology
Cambridge, MA 02139
and
Department of Statistics
Stanford University
Stanford, CA 94305
e-mail: rahul.mazumder@gmail.com

and

Trevor Hastie[†]

Departments of Statistics and Health Research and Policy
Stanford University
Stanford, CA 94305
e-mail: hastie@stanford.edu

Abstract: The graphical lasso [5] is an algorithm for learning the structure in an undirected Gaussian graphical model, using ℓ_1 regularization to control the number of zeros in the precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ [2, 11]. The R package GLASSO [5] is popular, fast, and allows one to efficiently build a path of models for different values of the tuning parameter. Convergence of GLASSO can be tricky; the converged precision matrix might not be the inverse of the estimated covariance, and occasionally it fails to converge with warm starts. In this paper we explain this behavior, and propose new algorithms that appear to outperform GLASSO.

By studying the "normal equations" we see that, GLASSO is solving the *dual* of the graphical lasso penalized likelihood, by block coordinate ascent; a result which can also be found in [2]. In this dual, the target of estimation is $\mathbf{\Sigma}$, the covariance matrix, rather than the precision matrix $\mathbf{\Theta}$. We propose similar primal algorithms P-GLASSO and DP-GLASSO, that also operate by block-coordinate descent, where $\mathbf{\Theta}$ is the optimization target. We study all of these algorithms, and in particular different approaches to solving their coordinate sub-problems. We conclude that DP-GLASSO is superior from several points of view.

AMS 2000 subject classifications: Primary 62H99, 62-09; secondary 62-04.
Keywords and phrases: Graphical lasso, sparse inverse covariance selection, precision matrix, convex analysis/optimization, positive definite matrices, sparsity, semidefinite programming.

Received August 2012.

[*] Rahul Mazumder was supported by grant DMS-1007719 from the National Science Foundation.

[†] Trevor Hastie was partially supported by grant DMS-1007719 from the National Science Foundation, and grant RO1-EB001988-15 from the National Institutes of Health.

Contents

1. Introduction

Consider a data matrix $\mathbf{X}_{n\times p}$, a sample of n realizations from a p-dimensional Gaussian distribution with zero mean and positive definite covariance matrix $\mathbf{\Sigma}$. The task is to estimate the unknown $\mathbf{\Sigma}$ based on the n samples — a challenging problem especially when $n \ll p$, when the ordinary maximum likelihood estimate does not exist. Even if it does exist (for $p \leq n$), the MLE is often poorly behaved, and regularization is called for. The Graphical Lasso [5] is a regularization framework for estimating the covariance matrix $\mathbf{\Sigma}$, under the assumption that its inverse $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ is sparse [2, 11, 8]. $\mathbf{\Theta}$ is called the precision matrix; if an element $\theta_{jk} = 0$, this implies that the corresponding variables X_j and X_k are conditionally independent, given the rest. Our algorithms focus either on the restricted version of $\mathbf{\Theta}$ or its inverse $\mathbf{W} = \mathbf{\Theta}^{-1}$. The graphical lasso problem minimizes a ℓ_1-regularized negative log-likelihood:

$$\underset{\mathbf{\Theta}\succ\mathbf{0}}{\text{minimize}}\; f(\mathbf{\Theta}) := -\log\det(\mathbf{\Theta}) + \operatorname{tr}(\mathbf{S}\mathbf{\Theta}) + \lambda\|\mathbf{\Theta}\|_1. \tag{1.1}$$

Here $\mathbf{S}$ is the sample covariance matrix, $\|\mathbf{\Theta}\|_1$ denotes the sum of the absolute values of $\mathbf{\Theta}$, and λ is a tuning parameter controlling the amount of ℓ_1 shrinkage. This is a semidefinite programming problem (SDP) in the variable $\mathbf{\Theta}$ [4].

In this paper we revisit the GLASSO algorithm proposed by Friedman, Hastie and Tibshirani [5] for solving (1.1); we analyze its properties, expose problems and issues, and propose alternative algorithms more suitable for the task.

Some of the results and conclusions of this paper can be found in [2], both explicitly and implicitly. We re-derive some of the results and derive new results, insights and algorithms, using a unified and more elementary framework.

Notation We denote the entries of a matrix $\boldsymbol{A}_{n\times n}$ by a_{ij}. $\|\boldsymbol{A}\|_1$ denotes the sum of its absolute values, $\|\boldsymbol{A}\|_\infty$ the maximum absolute value of its entries, $\|\boldsymbol{A}\|_F$ is its Frobenius norm, and $\text{abs}(\boldsymbol{A})$ is the matrix with elements $|a_{ij}|$. For a vector $\mathbf{u} \in \Re^q$, $\|\mathbf{u}\|_1$ denotes the ℓ_1 norm, and so on.

From now on, unless otherwise specified, we will assume that $\lambda > 0$.

2. Review of the GLASSO algorithm

We use the frame-work of "normal equations" as in [6, 5]. Using sub-gradient notation, we can write the optimality conditions (aka "normal equations") for a solution to (1.1) as

$$-\boldsymbol{\Theta}^{-1} + \mathbf{S} + \lambda\boldsymbol{\Gamma} = \mathbf{0}, \tag{2.1}$$

where $\boldsymbol{\Gamma}$ is a matrix of component-wise signs of $\boldsymbol{\Theta}$:

$$\begin{array}{rcl} \gamma_{jk} & = & \text{sign}(\theta_{jk}) \text{ if } \theta_{jk} \neq 0 \\ \gamma_{jk} & \in & [-1, 1] \text{ if } \theta_{jk} = 0 \end{array} \tag{2.2}$$

(we use the notation $\gamma_{jk} \in \text{Sign}(\theta_{jk})$). Since the global stationary conditions of (2.1) require θ_{jj} to be positive, this implies that

$$w_{ii} = s_{ii} + \lambda,\ i = 1, \ldots, p, \tag{2.3}$$

where $\mathbf{W} = \boldsymbol{\Theta}^{-1}$.

GLASSO uses a block-coordinate method for solving (2.1). Consider a partitioning of $\boldsymbol{\Theta}$ and $\boldsymbol{\Gamma}$:

$$\boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{\Theta}_{11} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{21} & \theta_{22} \end{pmatrix}, \quad \boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}_{21} & \gamma_{22} \end{pmatrix} \tag{2.4}$$

where $\boldsymbol{\Theta}_{11}$ is $(p-1)\times(p-1)$, $\boldsymbol{\theta}_{12}$ is $(p-1)\times 1$ and θ_{22} is scalar. $\mathbf{W}$ and $\mathbf{S}$ are partitioned the same way. Using properties of inverses of block-partitioned matrices, observe that $\mathbf{W} = \boldsymbol{\Theta}^{-1}$ can be written in two equivalent forms:

$$\begin{pmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & w_{22} \end{pmatrix} = \begin{pmatrix} (\boldsymbol{\Theta}_{11} - \frac{\boldsymbol{\theta}_{12}\boldsymbol{\theta}_{21}}{\theta_{22}})^{-1} & -\mathbf{W}_{11}\frac{\boldsymbol{\theta}_{12}}{\theta_{22}} \\ \cdot & \frac{1}{\theta_{22}} - \frac{\boldsymbol{\theta}_{21}\mathbf{W}_{11}\boldsymbol{\theta}_{12}}{\theta_{22}^2} \end{pmatrix} \tag{2.5}$$

$$= \begin{pmatrix} \boldsymbol{\Theta}_{11}^{-1} + \frac{\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}\boldsymbol{\theta}_{21}\boldsymbol{\Theta}_{11}^{-1}}{(\theta_{22}-\boldsymbol{\theta}_{21}\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12})} & -\frac{\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}}{\theta_{22}-\boldsymbol{\theta}_{21}\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}} \\ \cdot & \frac{1}{(\theta_{22}-\boldsymbol{\theta}_{21}\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12})} \end{pmatrix} \tag{2.6}$$

GLASSO solves for a row/column of (2.1) at a time, holding the rest fixed. Considering the pth column of (2.1), we get

$$-\mathbf{w}_{12} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}. \tag{2.7}$$

Reading off $\mathbf{w}_{12}$ from (2.5) we have

$$\mathbf{w}_{12} = -\mathbf{W}_{11}\boldsymbol{\theta}_{12}/\theta_{22} \tag{2.8}$$

and plugging into (2.7), we have:

$$\mathbf{W}_{11}\frac{\boldsymbol{\theta}_{12}}{\theta_{22}} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}. \tag{2.9}$$

GLASSO operates on the above gradient equation, as described below.

As a variation consider reading off $\mathbf{w}_{12}$ from (2.6):

$$\frac{\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}}{(\theta_{22} - \boldsymbol{\theta}_{21}\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12})} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}. \tag{2.10}$$

The above simplifies to

$$\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}w_{22} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}, \tag{2.11}$$

where $w_{22} = 1/(\theta_{22} - \boldsymbol{\theta}_{21}\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12})$ is fixed (by the global stationary conditions (2.3)). We will see that these two apparently similar estimating equations (2.9) and (2.11) lead to *very* different algorithms.

The GLASSO algorithm solves (2.9) for $\boldsymbol{\beta} = \boldsymbol{\theta}_{12}/\theta_{22}$, that is

$$\mathbf{W}_{11}\boldsymbol{\beta} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}, \tag{2.12}$$

where $\boldsymbol{\gamma}_{12} \in \operatorname{Sign}(\boldsymbol{\beta})$, since $\theta_{22} > 0$. (2.12) is the stationarity equation for the following ℓ_1 regularized quadratic program:

$$\underset{\boldsymbol{\beta}\in\Re^{p-1}}{\text{minimize}} \;\; \left\{\tfrac{1}{2}\boldsymbol{\beta}'\mathbf{W}_{11}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{s}_{12} + \lambda\|\boldsymbol{\beta}\|_1\right\}, \tag{2.13}$$

where $\mathbf{W}_{11} \succ 0$ is assumed to be fixed. This is analogous to a lasso regression problem of the last variable on the rest, except the cross-product matrix $\mathbf{S}_{11}$ is replaced by its current estimate $\mathbf{W}_{11}$. This problem itself can be solved efficiently using elementwise coordinate descent, exploiting the sparsity in $\boldsymbol{\beta}$. From $\hat{\boldsymbol{\beta}}$, it is easy to obtain $\hat{\mathbf{w}}_{12}$ from (2.8). Using the lower-right element of (2.5), $\hat{\theta}_{22}$ is obtained by

$$\frac{1}{\hat{\theta}_{22}} = w_{22} - \hat{\boldsymbol{\beta}}'\hat{\mathbf{w}}_{12}. \tag{2.14}$$

Finally, $\hat{\boldsymbol{\theta}}_{12}$ can now be recovered from $\hat{\boldsymbol{\beta}}$ and $\hat{\theta}_{22}$. Notice, however, that having solved for $\boldsymbol{\beta}$ and updated $\mathbf{w}_{12}$, GLASSO can move onto the next block; disentangling $\boldsymbol{\theta}_{12}$ and θ_{22} can be done at the end, when the algorithm over all blocks has converged. The GLASSO algorithm is outlined in Algorithm 1. We show in Lemma 3 in Section 8 that the successive updates in GLASSO keep $\mathbf{W}$ positive definite.

Figure 1 (left panel, black curve) plots the objective $f(\boldsymbol{\Theta}^{(k)})$ for the sequence of solutions produced by GLASSO on an example. Surprisingly, the curve is not monotone decreasing, as confirmed by the middle plot. If GLASSO were solving (1.1) by block coordinate-descent, we would not anticipate this behavior.

A closer look at steps (2.8) and (2.9) of the GLASSO algorithm leads to the following observations:

Algorithm 1 GLASSO algorithm [5]

```
1. Initialize $\mathbf{W} = \mathbf{S} + \lambda\mathbf{I}$.
2. Cycle around the columns repeatedly, performing the following steps till convergence:
    (a) Rearrange the rows/columns so that the target column is last (implicitly).
    (b) Solve the lasso problem (2.13), using as warm starts the solution from the previous round for this column.
    (c) Update the row/column (off-diagonal) of the covariance using $\hat{\mathbf{w}}_{12}$ (2.8).
    (d) Save $\hat{\boldsymbol{\beta}}$ for this column in the matrix $\mathbf{B}$.
3. Finally, for every row/column, compute the diagonal entries $\hat{\theta}_{jj}$ using (2.14), and convert the $\mathbf{B}$ matrix to $\boldsymbol{\Theta}$.
```

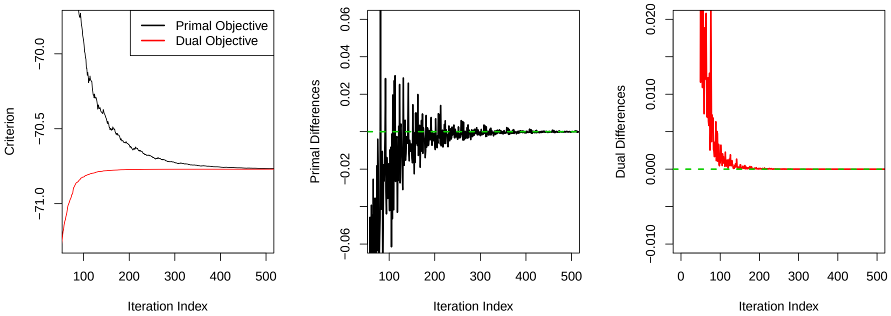

FIG 1. *[Left panel] The objective values of the primal criterion (1.1) and the dual criterion (4.1) corresponding to the covariance matrix* $\mathbf{W}$ *produced by* GLASSO *algorithm as a function of the iteration index (each column/row update). [Middle Panel] The successive differences of the primal objective values — the zero crossings indicate non-monotonicity. [Right Panel] The successive differences in the dual objective values — there are no zero crossings, indicating that* GLASSO *produces a monotone sequence of dual objective values.*

(a) We wish to solve (2.7) for $\boldsymbol{\theta}_{12}$. However $\boldsymbol{\theta}_{12}$ is entangled in $\mathbf{W}_{11}$, which is (incorrectly) treated as a constant.

(b) After updating $\boldsymbol{\theta}_{12}$, we see from (2.6) that the entire (working) covariance matrix $\mathbf{W}$ changes. GLASSO however updates only $\mathbf{w}_{12}$ and $\mathbf{w}_{21}$.

These two observations explain the non-monotone behavior of GLASSO in minimizing $f(\boldsymbol{\Theta})$. Section 3 shows a corrected block-coordinate descent algorithm for $\boldsymbol{\Theta}$, and Section 4 shows that the GLASSO algorithm is actually optimizing the dual of problem (1.1), with the optimization variable being $\mathbf{W}$.

3. A corrected GLASSO block coordinate-descent algorithm

Recall that (2.11) is a variant of (2.9), where the dependence of the covariance sub-matrix $\mathbf{W}_{11}$ on $\boldsymbol{\theta}_{12}$ is explicit. With $\boldsymbol{\alpha} = \boldsymbol{\theta}_{12} w_{22}$ (with $w_{22} \geq 0$ fixed),

Algorithm 2 P-GLASSO Algorithm

```
1. Initialize $\mathbf{W} = \mathrm{diag}(\mathbf{S}) + \lambda\mathbf{I}$, and $\boldsymbol{\Theta} = \mathbf{W}^{-1}$.
2. Cycle around the columns repeatedly, performing the following steps till convergence:
    (a) Rearrange the rows/columns so that the target column is last (implicitly).
    (b) Compute $\boldsymbol{\Theta}_{11}^{-1}$ using (3.3).
    (c) Solve (3.1) for $\boldsymbol{\alpha}$, using as warm starts the solution from the previous round of
        row/column updates. Update $\hat{\boldsymbol{\theta}}_{12} = \hat{\boldsymbol{\alpha}}/w_{22}$, and $\hat{\theta}_{22}$ using (3.2).
    (d) Update $\boldsymbol{\Theta}$ and $\mathbf{W}$ using (2.6), ensuring that $\boldsymbol{\Theta}\mathbf{W} = \mathbf{I}_p$.
3. Output the solution $\boldsymbol{\Theta}$ (precision) and its exact inverse $\mathbf{W}$ (covariance).
```

$\boldsymbol{\Theta}_{11} \succ 0$, (2.11) is equivalent to the stationary condition for

$$\underset{\boldsymbol{\alpha}\in\Re^{p-1}}{\text{minimize}} \left\{ \tfrac{1}{2}\boldsymbol{\alpha}'\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\alpha} + \boldsymbol{\alpha}'\mathbf{s}_{12} + \lambda\|\boldsymbol{\alpha}\|_1 \right\}. \tag{3.1}$$

If $\hat{\boldsymbol{\alpha}}$ is the minimizer of (3.1), then $\hat{\boldsymbol{\theta}}_{12} = \hat{\boldsymbol{\alpha}}/w_{22}$. To complete the optimization for the entire row/column we need to update θ_{22}. This follows simply from (2.6)

$$\hat{\theta}_{22} = \frac{1}{w_{22}} + \hat{\boldsymbol{\theta}}_{21}\boldsymbol{\Theta}_{11}^{-1}\hat{\boldsymbol{\theta}}_{12}, \tag{3.2}$$

with $w_{22} = s_{22} + \lambda$.

To solve (3.1) we need $\boldsymbol{\Theta}_{11}^{-1}$ for each block update. We achieve this by maintaining $\mathbf{W} = \boldsymbol{\Theta}^{-1}$ as the iterations proceed. Then for each block

- we obtain $\boldsymbol{\Theta}_{11}^{-1}$ from
$$\boldsymbol{\Theta}_{11}^{-1} = \mathbf{W}_{11} - \mathbf{w}_{12}\mathbf{w}_{21}/w_{22}; \tag{3.3}$$
- once $\boldsymbol{\theta}_{12}$ is updated, the *entire* working covariance matrix $\mathbf{W}$ is updated (in particular the portions $\mathbf{W}_{11}$ and $\mathbf{w}_{12}$), via the identities in (2.6), using the known $\boldsymbol{\Theta}_{11}^{-1}$.

Both these steps are simple rank-one updates with a total cost of $O(p^2)$ operations.

We refer to this as the primal graphical lasso or P-GLASSO, which we present in Algorithm 2.

The P-GLASSO algorithm requires slightly more work than GLASSO, since an additional $O(p^2)$ operations have to be performed before and after each block update. In return we have that after every row/column update, $\boldsymbol{\Theta}$ and $\mathbf{W}$ are positive definite (for $\lambda > 0$) and $\boldsymbol{\Theta}\mathbf{W} = \mathbf{I}_p$.

4. What is GLASSO actually solving?

Building upon the framework developed in Section 2, we now proceed to establish that GLASSO solves the convex dual of problem (1.1), by block coordinate ascent. We reach this conclusion via elementary arguments, closely aligned with the framework we develop in Section 2. The approach we present here is intended for an audience without much of a familiarity with convex duality theory [4].

Figure 1 illustrates that GLASSO is an ascent algorithm on the dual of the problem 1.1. The red curve in the left plot shows the dual objective rising monotonely, and the rightmost plot shows that the increments are indeed positive. There is an added twist though: in solving the block-coordinate update, GLASSO solves instead the dual of *that* subproblem.

4.1. Dual of the ℓ_1 regularized log-likelihood

We present below the following lemma, the conclusion of which also appears in [2], but we use the framework developed in Section 2.

Lemma 1. *Consider the primal problem (1.1) and its stationarity conditions (2.1). These are equivalent to the stationarity conditions for the box-constrained SDP*

$$\underset{\tilde{\mathbf{\Gamma}}:\ \|\tilde{\mathbf{\Gamma}}\|_\infty \le \lambda}{\text{maximize}}\ g(\tilde{\mathbf{\Gamma}}) := \log\det(\mathbf{S} + \tilde{\mathbf{\Gamma}}) + p \tag{4.1}$$

under the transformation $\mathbf{S} + \tilde{\mathbf{\Gamma}} = \mathbf{\Theta}^{-1}$.

Proof. The (sub)gradient conditions (2.1) can be rewritten as:

$$-(\mathbf{S} + \lambda\mathbf{\Gamma})^{-1} + \mathbf{\Theta} = \mathbf{0} \tag{4.2}$$

where $\mathbf{\Gamma} = \text{sgn}(\mathbf{\Theta})$. We write $\tilde{\mathbf{\Gamma}} = \lambda\mathbf{\Gamma}$ and observe that $\|\tilde{\mathbf{\Gamma}}\|_\infty \le \lambda$. Denote by $\text{abs}(\mathbf{\Theta})$ the matrix with element-wise absolute values.

Hence if $(\mathbf{\Theta}, \mathbf{\Gamma})$ satisfy (4.2), the substitutions

$$\tilde{\mathbf{\Gamma}} = \lambda\mathbf{\Gamma}; \quad \mathbf{P} = \text{abs}(\mathbf{\Theta}) \tag{4.3}$$

satisfy the following set of equations:

$$\begin{array}{rcl} -(\mathbf{S} + \tilde{\mathbf{\Gamma}})^{-1} + \mathbf{P} * \text{sgn}(\tilde{\mathbf{\Gamma}}) & = & \mathbf{0} \\ \mathbf{P} * (\text{abs}(\tilde{\mathbf{\Gamma}}) - \lambda\mathbf{1}_p\mathbf{1}_p') & = & \mathbf{0} \\ \|\tilde{\mathbf{\Gamma}}\|_\infty & \le & \lambda. \end{array} \tag{4.4}$$

In the above, $\mathbf{P}$ is a symmetric $p \times p$ matrix with non-negative entries, $\mathbf{1}_p\mathbf{1}_p'$ denotes a $p \times p$ matrix of ones, and the operator '$*$' denotes element-wise product. We observe that (4.4) are the KKT optimality conditions for the box-constrained SDP (4.1). Similarly, the transformations $\mathbf{\Theta} = \mathbf{P} * \text{sgn}(\tilde{\mathbf{\Gamma}})$ and $\mathbf{\Gamma} = \tilde{\mathbf{\Gamma}}/\lambda$ show that conditions (4.4) imply condition (4.2). Based on (4.2) the optimal solutions of the two problems (1.1) and (4.1) are related by $\mathbf{S} + \tilde{\mathbf{\Gamma}} = \mathbf{\Theta}^{-1}$. □

Notice that for the dual, the optimization variable is $\tilde{\mathbf{\Gamma}}$, with $\mathbf{S} + \tilde{\mathbf{\Gamma}} = \mathbf{\Theta}^{-1} = \mathbf{W}$. In other words, the dual problem solves for $\mathbf{W}$ rather than $\mathbf{\Theta}$, a fact that is suggested by the GLASSO algorithm.

Remark 1. The equivalence of the solutions to problems (4.1) and (1.1) as described above can also be derived via convex duality theory [4], which shows that (4.1) is a dual function of the ℓ_1 regularized negative log-likelihood (1.1). Strong duality holds, hence the optimal solutions of the two problems coincide [2].

We now consider solving (4.4) for the last block $\tilde{\boldsymbol{\gamma}}_{12}$ (excluding diagonal), holding the rest of $\tilde{\boldsymbol{\Gamma}}$ fixed. The corresponding equations are

$$\begin{array}{rcl} -\boldsymbol{\theta}_{12} + \mathbf{p}_{12} * \operatorname{sgn}(\tilde{\boldsymbol{\gamma}}_{12}) & = & \mathbf{0} \\ \mathbf{p}_{12} * (\operatorname{abs}(\tilde{\boldsymbol{\gamma}}_{12}) - \lambda \mathbf{1}_{p-1}) & = & \mathbf{0} \\ \|\tilde{\boldsymbol{\gamma}}_{12}\|_\infty & \leq & \lambda. \end{array} \tag{4.5}$$

The only non-trivial translation is the $\boldsymbol{\theta}_{12}$ in the first equation. We must express this in terms of the optimization variable $\tilde{\boldsymbol{\gamma}}_{12}$. Since $\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}_{12} = \mathbf{w}_{12}$, using the identities in (2.5), we have $\mathbf{W}_{11}^{-1}(\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}_{12}) = -\boldsymbol{\theta}_{12}/\theta_{22}$. Since $\theta_{22} > 0$, we can redefine $\tilde{\mathbf{p}}_{12} = \mathbf{p}_{12}/\theta_{22}$, to get

$$\begin{array}{rcl} \mathbf{W}_{11}^{-1}(\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}_{12}) + \tilde{\mathbf{p}}_{12} * \operatorname{sgn}(\tilde{\boldsymbol{\gamma}}_{12}) & = & \mathbf{0} \\ \tilde{\mathbf{p}}_{12} * (\operatorname{abs}(\tilde{\boldsymbol{\gamma}}_{12}) - \lambda \mathbf{1}_{p-1}) & = & \mathbf{0} \\ \|\tilde{\boldsymbol{\gamma}}_{12}\|_\infty & \leq & \lambda. \end{array} \tag{4.6}$$

The following lemma shows that a block update of GLASSO solves (4.6) (and hence (4.5)), a block of stationary conditions for the dual of the graphical lasso problem. Curiously, GLASSO does this not directly, but by solving the dual of the QP corresponding to this block of equations.

Lemma 2. *Assume $\mathbf{W}_{11} \succ \mathbf{0}$. The stationarity equations*

$$\mathbf{W}_{11}\hat{\boldsymbol{\beta}} + \mathbf{s}_{12} + \lambda\hat{\boldsymbol{\gamma}}_{12} = 0, \tag{4.7}$$

where $\hat{\boldsymbol{\gamma}}_{12} \in Sign(\hat{\boldsymbol{\beta}})$, correspond to the solution of the ℓ_1-regularized QP:

$$\underset{\boldsymbol{\beta} \in \Re^{p-1}}{\text{minimize}} \ \tfrac{1}{2}\boldsymbol{\beta}'\mathbf{W}_{11}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{s}_{12} + \lambda\|\boldsymbol{\beta}\|_1. \tag{4.8}$$

Solving (4.8) is equivalent to solving the following box-constrained QP:

$$\underset{\boldsymbol{\gamma} \in \Re^{p-1}}{\text{minimize}} \ \tfrac{1}{2}(\mathbf{s}_{12} + \boldsymbol{\gamma})'\mathbf{W}_{11}^{-1}(\mathbf{s}_{12} + \boldsymbol{\gamma}) \ \text{ subject to } \ \|\boldsymbol{\gamma}\|_\infty \leq \lambda, \tag{4.9}$$

with stationarity conditions given by (4.6), where the $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\gamma}}_{12}$ are related by

$$\hat{\boldsymbol{\beta}} = -\mathbf{W}_{11}^{-1}(\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}_{12}). \tag{4.10}$$

Proof. (4.7) is the KKT optimality condition for the ℓ_1 regularized QP (4.8). We rewrite (4.7) as

$$\hat{\boldsymbol{\beta}} + \mathbf{W}_{11}^{-1}(\mathbf{s}_{12} + \lambda\hat{\boldsymbol{\gamma}}_{12}) = 0. \tag{4.11}$$

Observe that $\hat{\beta}_i = \operatorname{sgn}(\hat{\beta}_i)|\beta_i| \ \forall i$ and $\|\hat{\boldsymbol{\gamma}}_{12}\|_\infty \leq 1$. Suppose $\hat{\boldsymbol{\beta}}, \ \hat{\boldsymbol{\gamma}}_{12}$ satisfy (4.11), then the substitutions

$$\tilde{\boldsymbol{\gamma}}_{12} = \lambda\hat{\boldsymbol{\gamma}}_{12}, \quad \tilde{\mathbf{p}}_{12} = \operatorname{abs}(\hat{\boldsymbol{\beta}}) \tag{4.12}$$

in (4.11) satisfy the stationarity conditions (4.6). It turns out that (4.6) is equivalent to the KKT optimality conditions of the box-constrained QP (4.9). Similarly, we note that if $\tilde{\boldsymbol{\gamma}}_{12}, \tilde{\mathbf{p}}_{12}$ satisfy (4.6), then the substitution

$$\hat{\boldsymbol{\gamma}}_{12} = \tilde{\boldsymbol{\gamma}}_{12}/\lambda; \quad \hat{\boldsymbol{\beta}} = \tilde{\mathbf{p}}_{12} * \operatorname{sgn}(\tilde{\boldsymbol{\gamma}}_{12})$$

satisfies (4.11). Hence the $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\gamma}}_{12}$ are related by (4.10). □

Remark 2. The above result can also be derived via convex duality theory[4], where (4.9) is actually the Lagrange dual of the ℓ_1 regularized QP (4.8), with (4.10) denoting the primal-dual relationship. [2, Section 3.3] interpret (4.9) as an ℓ_1 penalized regression problem (using convex duality theory) and explore connections with the set up of [8].

Note that the QP (4.9) is a (partial) optimization over the variable $\mathbf{w}_{12}$ only (since $\mathbf{s}_{12}$ is fixed); the sub-matrix $\mathbf{W}_{11}$ remains fixed in the QP. Exactly one row/column of $\mathbf{W}$ changes when the block-coordinate algorithm of GLASSO moves to a new row/column, unlike an explicit full matrix update in $\mathbf{W}_{11}$, which is required if $\boldsymbol{\theta}_{12}$ is updated. This again emphasizes that GLASSO is operating on the covariance matrix instead of $\boldsymbol{\Theta}$. We thus arrive at the following conclusion:

Theorem 4.1. GLASSO *performs block-coordinate ascent on the box-constrained SDP (4.1), the Lagrange dual of the primal problem (1.1). Each of the block steps are themselves box-constrained QPs, which* GLASSO *optimizes via their Lagrange duals.*

In our annotation perhaps GLASSO should be called DD-GLASSO, since it performs dual block updates for the dual of the graphical lasso problem. Banerjee, Ghaoui and d'Aspremont [2], the paper that inspired the original GLASSO article [5], also operates on the dual. They however solve the block-updates directly (which are box constrained QPs) using interior-point methods.

5. A new algorithm — DP-GLASSO

In Section 3, we described P-GLASSO, a primal coordinate-descent method. For every row/column we need to solve a lasso problem (3.1), which operates on a quadratic form corresponding to the square matrix $\boldsymbol{\Theta}_{11}^{-1}$. There are two problems with this approach:

- the matrix $\boldsymbol{\Theta}_{11}^{-1}$ needs to be constructed at every row/column update with complexity $O(p^2)$;
- $\boldsymbol{\Theta}_{11}^{-1}$ is dense.

We now show how a simple modification of the ℓ_1-regularized QP leads to a box-constrained QP with attractive computational properties.

The KKT optimality conditions for (3.1), following (2.11), can be written as:

$$\boldsymbol{\Theta}_{11}^{-1}\boldsymbol{\alpha} + \mathbf{s}_{12} + \lambda \operatorname{sgn}(\boldsymbol{\alpha}) = 0. \tag{5.1}$$

Algorithm 3 DP-GLASSO algorithm

```
1. Initialize $\mathbf{\Theta} = \mathrm{diag}(\mathbf{S} + \lambda\mathbf{I})^{-1}$.
2. Cycle around the columns repeatedly, performing the following steps till convergence:
    (a) Rearrange the rows/columns so that the target column is last (implicitly).
    (b) Solve (5.3) for $\tilde{\gamma}$ and update
            $\hat{\boldsymbol{\theta}}_{12} = -\mathbf{\Theta}_{11}(\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}})/w_{22}$
    (c) Solve for $\theta_{22}$ using (5.5).
    (d) Update the working covariance $\mathbf{w}_{12} = \mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}$.
```

Along the same lines of the derivations used in Lemma 2, the condition above is equivalent to

$$\begin{aligned} \tilde{\mathbf{q}}_{12} * \mathrm{sgn}(\tilde{\boldsymbol{\gamma}}) + \mathbf{\Theta}_{11}(\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}) &= \mathbf{0} \\ \tilde{\mathbf{q}}_{12} * (\mathrm{abs}(\tilde{\boldsymbol{\gamma}}) - \lambda\mathbf{1}_{p-1}) &= 0 \\ \|\tilde{\boldsymbol{\gamma}}\|_\infty &\leq \lambda \end{aligned} \tag{5.2}$$

for some vector (with non-negative entries) $\tilde{\mathbf{q}}_{12}$. (5.2) are the KKT optimality conditions for the following box-constrained QP:

$$\underset{\boldsymbol{\gamma}\in\Re^{p-1}}{\mathrm{minimize}} \ \tfrac{1}{2}(\mathbf{s}_{12} + \boldsymbol{\gamma})'\mathbf{\Theta}_{11}(\mathbf{s}_{12} + \boldsymbol{\gamma}); \ \text{subject to} \ \|\boldsymbol{\gamma}\|_\infty \leq \lambda. \tag{5.3}$$

The optimal solutions of (5.3) and (5.1) are related by

$$\hat{\boldsymbol{\alpha}} = -\mathbf{\Theta}_{11}(\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}}), \tag{5.4}$$

a consequence of (5.1), with $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{\theta}}_{12} \cdot w_{22}$ and $w_{22} = s_{22} + \lambda$. The diagonal θ_{22} of the precision matrix is updated via (2.6):

$$\hat{\theta}_{22} = \frac{1 - (\mathbf{s}_{12} + \tilde{\boldsymbol{\gamma}})'\hat{\boldsymbol{\theta}}_{12}}{w_{22}} \tag{5.5}$$

By strong duality, the box-constrained QP (5.3) with its optimality conditions (5.2) is equivalent to the lasso problem (3.1). Now both the problems listed at the beginning of the section are removed. The problem matrix $\mathbf{\Theta}_{11}$ is sparse, and no $O(p^2)$ updating is required after each block.

The solutions returned at step 2(b) for $\hat{\boldsymbol{\theta}}_{12}$ need not be exactly sparse, even though it purports to produce the solution to the primal block problem (3.1), which is sparse. One needs to use a tight convergence criterion when solving (5.3). In addition, one can threshold those elements of $\hat{\boldsymbol{\theta}}_{12}$ for which $\tilde{\boldsymbol{\gamma}}$ is away from the box boundary, since those values are known to be zero.

Note that DP-GLASSO does to the primal formulation (1.1) what GLASSO does to the dual. DP-GLASSO operates on the precision matrix, whereas GLASSO operates on the covariance matrix.

6. Computational costs in solving the block QPs

The ℓ_1 regularized QPs appearing in (2.13) and (3.1) are of the generic form

$$\underset{\mathbf{u}\in\Re^q}{\text{minimize}} \quad \tfrac{1}{2}\mathbf{u}'\mathbf{A}\mathbf{u} + \mathbf{a}'\mathbf{u} + \lambda\|\mathbf{u}\|_1, \tag{6.1}$$

for $\mathbf{A} \succ \mathbf{0}$. In this paper, we choose to use cyclical coordinate descent for solving (6.1), as it is used in the GLASSO algorithm implementation of Friedman, Hastie and Tibshirani [5]. Moreover, cyclical coordinate descent methods perform well with good warm-starts. These are available for both (2.13) and (3.1), since they both maintain working copies of the precision matrix, updated after every row/column update. There are other efficient ways for solving (6.1), capable of scaling to large problems — for example first-order proximal methods [3, 9], but we do not pursue them in this paper.

The box-constrained QPs appearing in (4.9) and (5.3) are of the generic form:

$$\underset{\mathbf{v}\in\Re^q}{\text{minimize}} \quad \tfrac{1}{2}(\mathbf{v}+\mathbf{b})'\tilde{\mathbf{A}}(\mathbf{v}+\mathbf{b}) \ \text{ subject to } \|\mathbf{v}\|_\infty \le \lambda \tag{6.2}$$

for some $\tilde{\mathbf{A}} \succ \mathbf{0}$. As in the case above, we will use cyclical coordinate-descent for optimizing (6.2).

In general it is more efficient to solve (6.1) than (6.2) for larger values of λ. This is because a large value of λ in (6.1) results in sparse solutions $\hat{\mathbf{u}}$; the coordinate descent algorithm can easily detect when a zero stays zero, and no further work gets done for that coordinate on that pass. If the solution to (6.1) has κ non-zeros, then on average κ coordinates need to be updated. This leads to a cost of $O(q\kappa)$, for one full sweep across all the q coordinates.

On the other hand, a large λ for (6.2) corresponds to a weakly-regularized solution. Cyclical coordinate procedures for this task are not as effective. Every coordinate update of $\mathbf{v}$ results in updating the gradient, which requires adding a scalar multiple of a column of $\tilde{\mathbf{A}}$. If $\tilde{\mathbf{A}}$ is dense, this leads to a cost of $O(q)$, and for one full cycle across all the coordinates this costs $O(q^2)$, rather than the $O(q\kappa)$ for (6.1).

However, our experimental results show that DP-GLASSO is more efficient than GLASSO, so there are some other factors in play. When $\tilde{\mathbf{A}}$ is sparse, there are computational savings. If $\tilde{\mathbf{A}}$ has κq non-zeros, the cost per column reduces on average to $O(\kappa q)$ from $O(q^2)$. For the formulation (5.3) $\tilde{\mathbf{A}}$ is $\mathbf{\Theta}_{11}$, which is sparse for large λ. Hence for large λ, GLASSO and DP-GLASSO have similar costs.

For smaller values of λ, the box-constrained QP (6.2) is particularly attractive. Most of the coordinates in the optimal solution $\hat{\mathbf{v}}$ will pile up at the boundary points $\{-\lambda, \lambda\}$, which means that the coordinates need not be updated frequently. For problem (5.3) this number is also κ, the number of non-zero coefficients in the corresponding column of the precision matrix. If κ of the coordinates pile up at the boundary, then one full sweep of cyclical coordinate descent across all the coordinates will require updating gradients corresponding to the remaining $q-\kappa$ coordinates. Using similar calculations as before, this will cost $O(q(q-\kappa))$ operations per full cycle (since for small λ, $\tilde{\mathbf{A}}$ will be dense).

For the ℓ_1 regularized problem (6.1), no such saving is achieved, and the cost is $O(q^2)$ per cycle.

Note that to solve problem (1.1), we need to solve a QP of a particular type (6.1) or (6.2) for a certain number of outer cycles (ie full sweeps across rows/columns). For every row/column update, the associated QP requires varying number of iterations to converge. It is hard to characterize all these factors and come up with precise estimates of convergence rates of the overall algorithm. However, we have observed that with warm-starts, on a relatively dense grid of λs, the complexities given above are pretty much accurate for DP-GLASSO (with warmstarts) specially when one is interested in solutions with small / moderate accuracy. Our experimental results in Section 9.1 and Appendix Section B support our observation.

We will now have a more critical look at the updates of the GLASSO algorithm and study their properties.

7. GLASSO: Positive definiteness, sparsity and exact inversion

As noted earlier, GLASSO operates on $\mathbf{W}$ — it does *not* explicitly compute the inverse $\mathbf{W}^{-1}$. It does however keep track of the estimates for $\boldsymbol{\theta}_{12}$ after every row/column update. The copy of $\boldsymbol{\Theta}$ retained by GLASSO along the row/column updates is not the exact inverse of the optimization variable $\mathbf{W}$. Figure 2 illustrates this by plotting the squared-norm $\|(\boldsymbol{\Theta} - \mathbf{W}^{-1})\|_F^2$ as a function of the iteration index. Only upon (asymptotic) convergence, will $\boldsymbol{\Theta}$ be equal to $\mathbf{W}^{-1}$. This can have important consequences.

In many real-life problems one only needs an approximate solution to (1.1):

- for computational reasons it might be impractical to obtain a solution of high accuracy;
- from a statistical viewpoint it might be sufficient to obtain an approximate solution for $\boldsymbol{\Theta}$ that is *both* sparse and positive definite

It turns out that the GLASSO algorithm is not suited to this purpose.

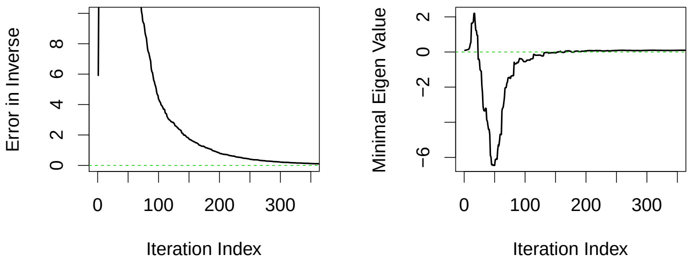

Fig 2. *Figure illustrating some negative properties of* GLASSO *using a typical numerical example. [Left Panel] The precision matrix produced after every row/column update need not be the exact inverse of the working covariance matrix — the squared Frobenius norm of the error is being plotted across iterations. [Right Panel] The estimated precision matrix* $\boldsymbol{\Theta}$ *produced by* GLASSO *need not be positive definite along iterations; plot shows minimal eigen-value.*

Since the GLASSO is a block coordinate procedure on the covariance matrix, it maintains a positive definite covariance matrix at every row/column update. However, since the estimated precision matrix is not the exact inverse of $\mathbf{W}$, it need not be positive definite. Although it is relatively straightforward to maintain an exact inverse of $\mathbf{W}$ along the row/column updates (via simple rank-one updates as before), this inverse $\mathbf{W}^{-1}$ need *not* be sparse. Arbitrary thresholding rules may be used to set some of the entries to zero, but that might destroy the positive-definiteness of the matrix. Since a principal motivation of solving (1.1) is to obtain a sparse precision matrix (which is also positive definite), returning a dense $\mathbf{W}^{-1}$ to (1.1) is not desirable.

Figures 2 illustrates the above observations on a typical example.

The DP-GLASSO algorithm operates on the primal (1.1). Instead of optimizing the ℓ_1 regularized QP (3.1), which requires computing $\mathbf{\Theta}_{11}^{-1}$, DP-GLASSO optimizes (5.3). After every row/column update the precision matrix $\mathbf{\Theta}$ is positive definite. The working covariance matrix maintained by DP-GLASSO via $\mathbf{w}_{12} := \mathbf{s}_{12} + \hat{\boldsymbol{\gamma}}$ need not be the exact inverse of $\mathbf{\Theta}$. Exact covariance matrix estimates, if required, can be obtained by tracking $\mathbf{\Theta}^{-1}$ via simple rank-one updates, as described earlier.

Unlike GLASSO, DP-GLASSO (and P-GLASSO) return a sparse and positive definite precision matrix even if the row/column iterations are terminated prematurely.

8. Warm starts and path-seeking strategies

Since we seldom know in advance a good value of λ, we often compute a sequence of solutions to (1.1) for a (typically) decreasing sequence of values $\lambda_1 > \lambda_2 > \cdots > \lambda_K$. Warm-start or continuation methods use the solution at λ_i as an initial guess for the solution at λ_{i+1}, and often yield great efficiency. It turns out that for algorithms like GLASSO which operate on the dual problem, not all warm-starts necessarily lead to a convergent algorithm. We address this aspect in detail in this section.

The following lemma states the conditions under which the row/column updates of the GLASSO algorithm will maintain positive definiteness of the covariance matrix $\mathbf{W}$.

Lemma 3. *Suppose* $\mathbf{Z}$ *is used as a warm-start for the* GLASSO *algorithm. If* $\mathbf{Z} \succ \mathbf{0}$ *and* $\|\mathbf{Z} - \mathbf{S}\|_\infty \leq \lambda$, *then every row/column update of* GLASSO *maintains positive definiteness of the working covariance matrix* $\mathbf{W}$.

Proof. Recall that the GLASSO solves the dual (4.1). Assume $\mathbf{Z}$ is partitioned as in (2.4), and the pth row/column is being updated. Since $\mathbf{Z} \succ \mathbf{0}$, we have both

$$\mathbf{Z}_{11} \succ \mathbf{0} \text{ and } \big(z_{22} - \mathbf{z}_{21}(\mathbf{Z}_{11})^{-1}\mathbf{z}_{12}\big) > 0. \tag{8.1}$$

Since $\mathbf{Z}_{11}$ remains fixed, it suffices to show that after the row/column update, the expression $(\hat{w}_{22} - \hat{\mathbf{w}}_{21}(\mathbf{Z}_{11})^{-1}\hat{\mathbf{w}}_{12})$ remains positive. Recall that, via standard optimality conditions we have $\hat{w}_{22} = s_{22} + \lambda$, which makes $\hat{w}_{22} \geq z_{22}$ (since by assumption, $|z_{22} - s_{22}| \leq \lambda$ and $z_{22} > 0$). Furthermore, $\hat{\mathbf{w}}_{21} = \mathbf{s}_{21} + \hat{\boldsymbol{\gamma}}$, where $\hat{\boldsymbol{\gamma}}$ is the optimal solution to the corresponding box-QP (4.9). Since the starting solution $\mathbf{z}_{21}$ satisfies the box-constraint (4.9) i.e. $\|\mathbf{z}_{21} - \mathbf{s}_{21}\|_\infty \leq \lambda$, the optimal solution of the QP (4.9) improves the objective:

$$\hat{\mathbf{w}}_{21}(\mathbf{Z}_{11})^{-1}\hat{\mathbf{w}}_{12} \leq \mathbf{z}_{21}(\mathbf{Z}_{11})^{-1}\mathbf{z}_{12}$$

Combining the above along with the fact that $\hat{w}_{22} \geq z_{22}$ we see

$$\hat{w}_{22} - \hat{\mathbf{w}}_{21}(\mathbf{Z}_{11})^{-1}\hat{\mathbf{w}}_{12} > 0, \tag{8.2}$$

which implies that the new covariance estimate $\widehat{\mathbf{W}} \succ \mathbf{0}$. □

Remark 3. If the condition $\|\mathbf{Z} - \mathbf{S}\|_\infty \leq \lambda$ appearing in Lemma 3 is violated, then the row/column update of GLASSO need not maintain PD of the covariance matrix $\mathbf{W}$.

We have encountered many counter-examples that show this to be true, see the discussion below.

The R package implementation of GLASSO allows the user to specify a warm-start as a tuple $(\boldsymbol{\Theta}_0, \mathbf{W}_0)$. This option is typically used in the construction of a path algorithm.

If $(\widehat{\boldsymbol{\Theta}}_\lambda, \widehat{\mathbf{W}}_\lambda)$ is provided as a warm-start for $\lambda' < \lambda$, then the GLASSO algorithm is not guaranteed to converge. It is easy to find numerical examples by choosing the gap $\lambda - \lambda'$ to be large enough. Among the various examples we encountered, we briefly describe one here. Details of the experiment/data and other examples can be found in the Appendix A.1. We generated a data-matrix $\mathbf{X}_{n \times p}$, with $n = 2, p = 5$ with iid standard Gaussian entries. $\mathbf{S}$ is the sample covariance matrix. We solved problem (1.1) using GLASSO for $\lambda = 0.9 \times \max_{i \neq j} |s_{ij}|$. We took the estimated covariance and precision matrices: $\widehat{\mathbf{W}}_\lambda$ and $\widehat{\boldsymbol{\Theta}}_\lambda$ as a warm-start for the GLASSO algorithm with $\lambda' = \lambda \times 0.01$. The GLASSO algorithm failed to converge with this warm-start. We note that $\|\widehat{\mathbf{W}}_\lambda - \mathbf{S}\|_\infty = 0.0402 \nleq \lambda'$ (hence violating the sufficient condition in Lemma 4) and after updating the first row/column via the GLASSO algorithm we observed that "covariance matrix" $\mathbf{W}$ has negative eigen-values — leading to a non-convergent algorithm. The above phenomenon is not surprising and easy to explain and generalize. If the warm-start fails to satisfy $\|\widehat{\mathbf{W}}_\lambda - \mathbf{S}\|_\infty$, then during the course of the row/column updates the working covariance matrix may lose positive definiteness. In such a case, the block problems (QPs) may not correspond to valid convex programs (due to the lack of the postive-definiteness of the quadratic forms). This seems to be the fundamental reason behind the non-convergence of the algorithm. Since $\widehat{\mathbf{W}}_\lambda$ solves the dual (4.1), it is necessarily of the form $\widehat{\mathbf{W}}_\lambda = \mathbf{S} + \tilde{\boldsymbol{\Gamma}}$, for $\|\tilde{\boldsymbol{\Gamma}}\|_\infty \leq \lambda$. In the light of Lemma 3 and also Remark 3, the warm-start needs to be dual-feasible in order to guarantee that the iterates $\widehat{\mathbf{W}}$ remain PD and hence for the sub-problems to be well defined convex programs. Clearly $\widehat{\mathbf{W}}_\lambda$ does not satisfy the box-constraint $\|\widehat{\mathbf{W}}_\lambda - \mathbf{S}\|_\infty \leq \lambda'$, for $\lambda' < \lambda$. However, in practice the GLASSO algorithm is usually seen to converge (numerically) when λ' is quite *close* to λ. This is probably because the working covariance matrix remains positive definite and the block QPs are valid convex programs. If the difference between λ' and λ is large then the algorithm may very likely get into trouble.

The following lemma establishes that any PD matrix can be taken as a warm-start for P-GLASSO or DP-GLASSOto ensure a convergent algorithm.

Lemma 4. *Suppose $\mathbf{\Phi} \succ \mathbf{0}$ is a used as a warm-start for the* P-GLASSO *(or* DP-GLASSO*) algorithm. Then every row/column update of* P-GLASSO *(or* DP-GLASSO*) maintains positive definiteness of the working precision matrix $\mathbf{\Theta}$.*

Proof. Consider updating the pth row/column of the precision matrix. The condition $\mathbf{\Phi} \succ \mathbf{0}$ is equivalent to both

$$\mathbf{\Phi}_{11} \succ \mathbf{0} \text{ and } \left(\phi_{22} - \mathbf{\Phi}_{21}(\mathbf{\Phi}_{11})^{-1}\mathbf{\Phi}_{12}\right) > 0.$$

Note that the block $\mathbf{\Phi}_{11}$ remains fixed; only the pth row/column of $\mathbf{\Theta}$ changes. $\boldsymbol{\phi}_{21}$ gets updated to $\hat{\boldsymbol{\theta}}_{21}$, as does $\hat{\boldsymbol{\theta}}_{12}$. From (2.6) the updated diagonal entry $\hat{\theta}_{22}$ satisfies:

$$\hat{\theta}_{22} - \hat{\boldsymbol{\theta}}_{21}(\mathbf{\Phi}_{11})^{-1}\hat{\boldsymbol{\theta}}_{12} = \frac{1}{(s_{22} + \lambda)} > 0.$$

Thus the updated matrix $\hat{\mathbf{\Theta}}$ remains PD. The result for the DP-GLASSO algorithm follows, since both the versions P-GLASSO and DP-GLASSO solve the same block coordinate problem. □

Remark 4. A simple consequence of Lemmas 3 and 4 is that the QPs arising in the process, namely the ℓ_1 regularized QPs (2.13), (3.1) and the box-constrained QPs (4.9) and (5.3) are all valid convex programs, since all the respective matrices $\mathbf{W}_{11}$, $\mathbf{\Theta}_{11}^{-1}$ and $\mathbf{W}_{11}^{-1}$, $\mathbf{\Theta}_{11}$ appearing in the quadratic forms are PD.

As exhibited in Lemma 4, both the algorithms DP-GLASSO and P-GLASSO are guaranteed to converge from any positive-definite warm start. This is due to the unconstrained formulation of the primal problem (1.1).

GLASSO really only requires an initialization for $\mathbf{W}$, since it constructs $\mathbf{\Theta}$ on the fly. Likewise DP-GLASSO only requires an initialization for $\mathbf{\Theta}$. Having the other half of the tuple assists in the block-updating algorithms. For example, GLASSO solves a series of lasso problems, where $\mathbf{\Theta}$ play the role as parameters. By supplying $\mathbf{\Theta}$ along with $\mathbf{W}$, the block-wise lasso problems can be given starting values close to the solutions. The same applies to DP-GLASSO. In neither case do the pairs have to be inverses of each other to serve this purpose.

If we wish to start with inverse pairs, and maintain such a relationship, we have described earlier how $O(p^2)$ updates after each block optimization can achieve this. One caveat for GLASSO is that starting with an inverse pair costs $O(p^3)$ operations, since we typically start with $\mathbf{W} = \mathbf{S} + \lambda\mathbf{I}$. For DP-GLASSO, we typically start with a diagonal matrix, which is trivial to invert.

9. Experimental results & timing comparisons

We compared the performances of algorithms GLASSO and DP-GLASSO (both with and without warm-starts) on different examples with varying (n, p) values. While most of the results are presented in this section, some are relegated to the Appendix B. Section 9.1 describes some synthetic examples and Section 9.2 presents comparisons on a real-life micro-array data-set.

9.1. Synthetic experiments

In this section we present examples generated from two different covariance models — as characterized by the covariance matrix $\mathbf{\Sigma}$ or equivalently the precision matrix $\mathbf{\Theta}$. We create a data matrix $\mathbf{X}_{n\times p}$ by drawing n independent samples from a p dimensional normal distribution $\text{MVN}(\mathbf{0}, \mathbf{\Sigma})$. The sample covariance matrix is taken as the input $\mathbf{S}$ to problem (1.1). The two covariance models are described below:

Type-1 The population concentration matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ has uniform sparsity with approximately 77 % of the entries zero.
We created the covariance matrix as follows. We generated a matrix $\mathbf{B}$ with iid standard Gaussian entries, symmetrized it via $\frac{1}{2}(\mathbf{B} + \mathbf{B}')$ and set approximately 77% of the entries of this matrix to zero, to obtain $\tilde{\mathbf{B}}$ (say). We added a scalar multiple of the p dimensional identity matrix to $\tilde{\mathbf{B}}$ to get the precision matrix $\mathbf{\Theta} = \tilde{\mathbf{B}} + \eta \mathbf{I}_{p\times p}$, with η chosen such that the minimum eigen value of $\mathbf{\Theta}$ is one.

Type-2 This example, taken from [11], is an auto-regressive process of order two — the precision matrix being tri-diagonal:

$$\theta_{ij} = \begin{cases} 0.5, & \text{if } \ |j-i| = 1, \ i = 2, \ldots, (p-1); \\ 0.25, & \text{if } \ |j-i| = 2, \ i = 3, \ldots, (p-2); \\ 1, & \text{if } \ i = j, \ i = 1, \ldots, p; \text{and} \\ 0 & \text{otherwise.} \end{cases}$$

For each of the two set-ups Type-1 and Type-2 we consider twelve different combinations of (n, p):

(a) $p = 1000$, $n \in \{1500, 1000, 500\}$.
(b) $p = 800$, $n \in \{1000, 800, 500\}$.
(c) $p = 500$, $n \in \{800, 500, 200\}$.
(d) $p = 200$, $n \in \{500, 200, 50\}$.

For every (n, p) we solved (1.1) on a grid of twenty λ values linearly spaced in the log-scale, with $\lambda_i = 0.8^i \times \{0.9\lambda_{\max}\}$, $i = 1, \ldots, 20$, where $\lambda_{\max} = \max_{i\neq j} |s_{ij}|$, is the off-diagonal entry of $\mathbf{S}$ with largest absolute value. $\lambda_{\max}$ is the smallest value of λ for which the solution to (1.1) is a diagonal matrix.

Since this article focuses on the GLASSO algorithm, its properties and alternatives that stem from the main idea of block-coordinate optimization, we present here the performances of the following algorithms:

Dual-Cold GLASSO with initialization $\mathbf{W} = \mathbf{S} + \lambda\mathbf{I}_{p\times p}$, as suggested in [5].

Dual-Warm The path-wise version of GLASSO with warm-starts, as suggested in [5]. Although this path-wise version need not converge in general, this was not a problem in our experiments, probably due to the fine-grid of λ values.

Primal-Cold DP-GLASSO with diagonal initialization $\mathbf{\Theta} = (\text{diag}(\mathbf{S}) + \lambda\mathbf{I})^{-1}$.

Primal-Warm The path-wise version of DP-GLASSO with warm-starts.

We did not include P-GLASSO in the comparisons above since P-GLASSO requires additional matrix rank-one updates after every row/column update, which makes it more expensive. None of the above listed algorithms require matrix inversions (via rank one updates). Furthermore, DP-GLASSO and P-GLASSO are quite similar as both are doing a block coordinate optimization on the dual. Hence we only included DP-GLASSO in our comparisons. We used our own implementation of the GLASSO and DP-GLASSO algorithm in R. The entire program is written in R, except the inner block-update solvers, which are the real work-horses:

- For GLASSO we used the lasso code crossProdLasso written in FORTRAN by [5];
- For DP-GLASSO we wrote our own FORTRAN code to solve the box QP.

An R package `dpglasso` that implements DP-GLASSO is available on CRAN.

In the figure and tables that follow below, for every algorithm, at a fixed λ we report the *total time* taken by *all* the QPs — the ℓ_1 regularized QP for GLASSO and the box constrained QP for DP-GLASSO till convergence All computations were done on a Linux machine with model specs: Intel(R) Xeon(R) CPU 5160 @ 3.00GHz.

Convergence Criterion Since DP-GLASSO operates on the the primal formulation and GLASSO operates on the dual — to make the convergence criteria comparable across examples we based it on the relative change in the primal objective values i.e. $f(\mathbf{\Theta})$ (1.1) across two successive iterations:

$$\frac{f(\mathbf{\Theta}_k) - f(\mathbf{\Theta}_{k-1})}{|f(\mathbf{\Theta}_{k-1})|} \leq \text{TOL}, \tag{9.1}$$

where one iteration refers to a full sweep across p rows/columns of the precision matrix (for DP-GLASSO) and covariance matrix (for GLASSO); and TOL denotes the tolerance level or level of accuracy of the solution. To compute the primal objective value for the GLASSO algorithm, the precision matrix is computed from $\widehat{\mathbf{W}}$ via direct inversion (the time taken for inversion and objective value computation is not included in the timing comparisons).

Computing the objective function is quite expensive relative to the computational cost of the iterations. In our experience convergence criteria based on a relative change in the precision matrix for DP-GLASSO and the covariance matrix for GLASSO seemed to be a practical choice for the examples we considered. However, for reasons we described above, we used criterion 9.1 in the experiments.

Observations Figure 3 presents the times taken by the algorithms to converge to an accuracy of $\text{TOL} = 10^{-4}$ on a grid of λ values.

The figure shows eight different scenarios with $p > n$, corresponding to the two different covariance models Type-1 (left panel) and Type-2 (right panel). It is quite evident that DP-GLASSO with warm-starts (Primal-Warm) outperforms all the other algorithms across all the different examples. All the algorithms converge quickly for large values of λ (typically high sparsity) and become slower with decreasing λ. For large p and small λ, convergence is slow; however for $p > n$, the non-sparse end of the regularization path is really not that interesting from a statistical viewpoint. Warm-starts apparently do *not* always help in speeding up the convergence of GLASSO ; for example see Figure 3 with $(n, p) = (500, 1000)$ (Type 1) and $(n, p) = (500, 800)$ (Type 2). This probably further validates the fact that warm-starts in the case of GLASSO need to be carefully designed, in order for them to *speed-up* convergence. Note however, that GLASSO with the warm-starts prescribed is not even guaranteed to converge — we however did not come across any such instance among the experiments presented in this section.

Based on the suggestion of a referee we annotated the plots in Figure 3 with locations in the regularization path that are of interest. For each plot, two vertical dotted lines are drawn which correspond to the λs at which the distance of the estimated precision matrix $\widehat{\mathbf{\Theta}}_\lambda$ from the population precision matrix is minimized wrt to the $\|\cdot\|_1$ norm (green) and $\|\cdot\|_F$ norm (blue). The optimal λ corresponding to the $\|\cdot\|_1$ metric chooses sparser models than those chosen by $\|\cdot\|_F$; the performance gains achieved by DP-GLASSO seem to be more prominent for the latter λ.

Table 1 presents the timings for all the four algorithmic variants on the twelve different (n, p) combinations listed above for Type 1. For every example, we report the total time till convergence on a grid of twenty λ values for two different tolerance levels: $\text{TOL} \in \{10^{-4}, 10^{-5}\}$. Note that the DP-GLASSO returns positive definite and sparse precision matrices even if the algorithm is terminated at a relatively small/moderate accuracy level — this is not the case in GLASSO. The rightmost column presents the proportion of non-zeros averaged across the entire path of solutions $\widehat{\mathbf{\Theta}}_\lambda$, where $\widehat{\mathbf{\Theta}}_\lambda$ is obtained by solving (1.1) to a high precision i.e. 10^{-6}, by algorithms GLASSO and DP-GLASSO and averaging the results.

Again we see that in all the examples DP-GLASSO with warm-starts is the clear winner among its competitors. For a fixed p, the total time to trace out the path generally decreases with increasing n. There is no clear winner between GLASSO with warm-starts and GLASSO without warm-starts. It is often seen that DP-GLASSO without warm-starts converges faster than both the variants of GLASSO (with and without warm-starts).

Table 2 reports the timing comparisons for Type 2. Once again we see that in all the examples Primal-Warm turns out to be the clear winner.

For $n \leq p = 1000$, we observe that Primal-Warm is generally faster for Type-2 than Type-1. This however, is reversed for smaller values of $p \in \{800, 500\}$. Primal-Cold is has a smaller overall computation time for Type-1 over Type-2.

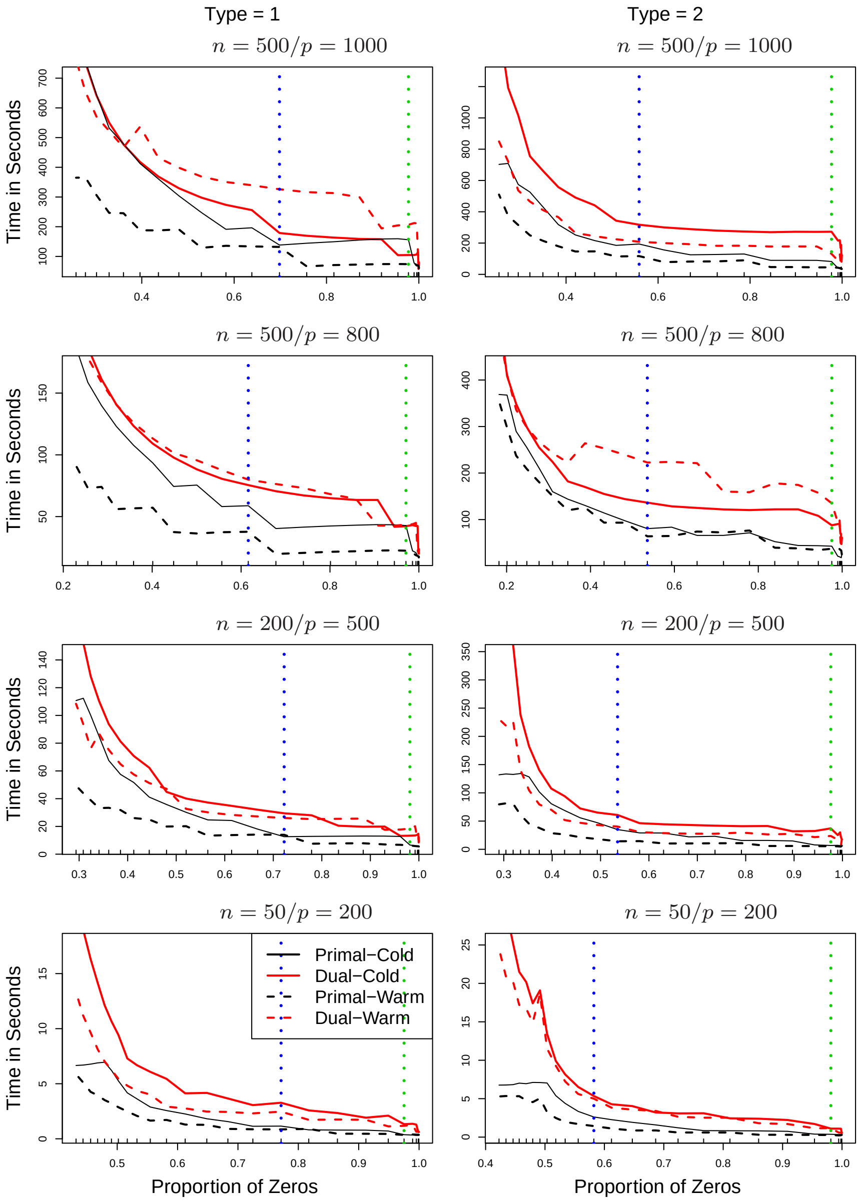

FIG 3. *The timings in seconds for the four different algorithmic versions:* GLASSO *(with and without warm-starts) and* DP-GLASSO *(with and without warm-starts) for a grid of λ values on the log-scale. [Left Panel] Covariance model for Type-1, [Right Panel] Covariance model for Type-2. The horizontal axis is indexed by the proportion of zeros in the solution. The vertical dashed lines correspond to the optimal λ values for which the estimated errors $\|\widehat{\mathbf{\Theta}}_\lambda - \mathbf{\Theta}\|_1$ (green) and $\|\widehat{\mathbf{\Theta}}_\lambda - \mathbf{\Theta}\|_F$ (blue) are minimum.*

Table 1

Table showing the performances of the four algorithms glasso *(Dual-Warm/Cold) and* dp-glasso *(Primal-Warm/Cold) for the covariance model Type-1. We present the times (in seconds) required to compute a path of solutions to (1.1) (on a grid of twenty λ values) for different (n, p) combinations and relative errors (as in (9.1)). The rightmost column gives the averaged sparsity level across the grid of λ values.* dp-glasso *with warm-starts is consistently the winner across all the examples*

p / n	relative	Total time (secs) to compute a path of solutions				Average %
	error (TOL)	Dual-Cold	Dual-Warm	Primal-Cold	Primal-Warm	Zeros in path
1000 / 500	10^{-4}	3550.71	6592.63	2558.83	**2005.25**	80.2
	10^{-5}	4706.22	8835.59	3234.97	**2832.15**	
1000 / 1000	10^{-4}	2788.30	3158.71	2206.95	**1347.05**	83.0
	10^{-5}	3597.21	4232.92	2710.34	**1865.57**	
1000 / 1500	10^{-4}	2447.19	4505.02	1813.61	**932.34**	85.6
	10^{-5}	2764.23	6426.49	2199.53	**1382.64**	
800 / 500	10^{-4}	1216.30	2284.56	928.37	**541.66**	78.8
	10^{-5}	1776.72	3010.15	1173.76	**798.93**	
800 / 800	10^{-4}	1135.73	1049.16	788.12	**438.46**	80.0
	10^{-5}	1481.36	1397.25	986.19	**614.98**	
800 / 1000	10^{-4}	1129.01	1146.63	786.02	**453.06**	80.2
	10^{-5}	1430.77	1618.41	992.13	**642.90**	
500 / 200	10^{-4}	605.45	559.14	395.11	**191.88**	75.9
	10^{-5}	811.58	795.43	520.98	**282.65**	
500 / 500	10^{-4}	427.85	241.90	252.83	**123.35**	75.2
	10^{-5}	551.11	315.86	319.89	**182.81**	
500 / 800	10^{-4}	359.78	279.67	207.28	**111.92**	80.9
	10^{-5}	416.87	402.61	257.06	**157.13**	
200 / 50	10^{-4}	65.87	50.99	37.40	**23.32**	75.6
	10^{-5}	92.04	75.06	45.88	**35.81**	
200 / 200	10^{-4}	35.29	25.70	17.32	**11.72**	66.8
	10^{-5}	45.90	33.23	22.41	**17.16**	
200 / 300	10^{-4}	32.29	23.60	16.30	**10.77**	66.0
	10^{-5}	38.37	33.95	20.12	**15.12**	

In some cases (for example $n \leq p = 1000$), we see that Primal-Warm in Type-2 converges much faster than its competitors on a relative scale than in Type-1 — this difference is due to the variations in the structure of the covariance matrix.

9.2. Micro-array example

We consider the data-set introduced in [1] and further studied in [10, 7]. In this experiment, tissue samples were analyzed using an Affymetrix Oligonucleotide array. The data was processed, filtered and reduced to a subset of 2000 gene expression values. The number of Colon Adenocarcinoma tissue samples is $n = 62$. For the purpose of the experiments presented in this section, we pre-screened the genes to a size of $p = 725$. We obtained this subset of genes using the idea of *exact covariance thresholding* introduced in our paper [7]. We thresholded the sample correlation matrix obtained from the 62×2000 microarray data-matrix into connected components with a threshold of 0.00364[1] — the genes belonging to the largest connected component formed our pre-screened gene pool of size

[1] this is the largest value of the threshold for which the size of the largest connected component is smaller than 800

Table 2

Table showing comparative timings of the four algorithmic variants of GLASSO *and* DP-GLASSO *for the covariance model in Type-2. This table is similar to Table 1, displaying results for Type-1.* DP-GLASSO *with warm-starts consistently outperforms all its competitors*

p / n	relative	Total time (secs) to compute a path of solutions				Average %
	error (TOL)	Dual-Cold	Dual-Warm	Primal-Cold	Primal-Warm	Zeros in path
1000 / 500	10^{-4}	6093.11	5483.03	3495.67	**1661.93**	75.6
	10^{-5}	7707.24	7923.80	4401.28	**2358.08**	
1000 / 1000	10^{-4}	4773.98	3582.28	2697.38	**1015.84**	76.70
	10^{-5}	6054.21	4714.80	3444.79	**1593.54**	
1000 / 1500	10^{-4}	4786.28	5175.16	2693.39	**1062.06**	78.5
	10^{-5}	6171.01	6958.29	3432.33	**1679.16**	
800 / 500	10^{-4}	2914.63	3466.49	1685.41	**1293.18**	74.3
	10^{-5}	3674.73	4572.97	2083.20	**1893.22**	
800 / 800	10^{-4}	2021.55	1995.90	1131.35	**618.06**	74.4
	10^{-5}	2521.06	2639.62	1415.95	**922.93**	
800 / 1000	10^{-4}	3674.36	2551.06	1834.86	**885.79**	75.9
	10^{-5}	4599.59	3353.78	2260.58	**1353.28**	
500 / 200	10^{-4}	1200.24	885.76	718.75	**291.61**	70.5
	10^{-5}	1574.62	1219.12	876.45	**408.41**	
500 / 500	10^{-4}	575.53	386.20	323.30	**130.59**	72.2
	10^{-5}	730.54	535.58	421.91	**193.08**	
500 / 800	10^{-4}	666.75	474.12	373.60	**115.75**	73.7
	10^{-5}	852.54	659.58	485.47	**185.60**	
200 / 50	10^{-4}	110.18	98.23	48.98	**26.97**	73.0
	10^{-5}	142.77	133.67	55.27	**33.95**	
200 / 200	10^{-4}	50.63	40.68	23.94	**9.97**	63.7
	10^{-5}	66.63	56.71	31.57	**14.70**	
200 / 300	10^{-4}	47.63	36.18	21.24	**8.19**	65.0
	10^{-5}	60.98	50.52	27.41	**12.22**	

Table 3

Comparisons among algorithms for a microarray dataset with $n = 62$ and $p = 725$, for different tolerance levels (TOL). We took a grid of fifteen λ values, the average % of zeros along the whole path is 90.8

relative	Total time (secs) to compute a path of solutions			
error (TOL)	Dual-Cold	Dual-Warm	Primal-Cold	Primal-Warm
10^{-3}	515.15	406.57	462.58	**334.56**
10^{-4}	976.16	677.76	709.83	**521.44**

$p = 725$. This (subset) data-matrix of size $(n, p) = (62, 725)$ is used for our experiments.

The results presented below in Table 3 show timing comparisons of the four different algorithms: Primal-Warm/Cold and Dual-Warm/Cold on a grid of fifteen λ values in the log-scale. Once again we see that the Primal-Warm outperforms the others in terms of speed and accuracy. Dual-Warm performs quite well in this example.

10. Conclusions

This paper explores some of the apparent mysteries in the behavior of the GLASSO algorithm introduced in [5]. These have been explained by leveraging the fact that the GLASSO algorithm is solving the dual of the graphical lasso problem (1.1), by block coordinate ascent. Each block update, itself the solution to a convex program, is solved via its own dual, which is equivalent to a lasso problem. The optimization variable is $\mathbf{W}$, the covariance matrix, rather than the target precision matrix $\mathbf{\Theta}$. During the course of the iterations, a working version of $\mathbf{\Theta}$ is maintained, but it may not be positive definite, and its inverse is not $\mathbf{W}$. Tight convergence is therefore essential, for the solution $\hat{\mathbf{\Theta}}$ to be a proper inverse covariance. There are issues using warm starts with GLASSO, when computing a path of solutions. Unless the sequence of λs are sufficiently close, since the "warm start"s are not dual feasible, the algorithm can get into trouble.

We have also developed two primal algorithms P-GLASSO and DP-GLASSO. The former is more expensive, since it maintains the relationship $\mathbf{W} = \mathbf{\Theta}^{-1}$ at every step, an $O(p^3)$ operation per sweep across all row/columns. DP-GLASSO is similar in flavor to GLASSO except its optimization variable is $\mathbf{\Theta}$. It also solves the dual problem when computing its block update, in this case a box-QP. This box-QP has attractive sparsity properties at *both* ends of the regularization path, as evidenced in some of our experiments. It maintains a positive definite $\mathbf{\Theta}$ throughout its iterations, and can be started at any positive definite matrix. Our experiments show in addition that DP-GLASSO is faster than GLASSO.

An R package `dpglasso` that implements DP-GLASSO is available on CRAN.

Acknowledgements

We would like to thank Robert Tibshirani and his research group at Stanford Statistics for helpful discussions. We are also thankful to the anonymous referees whose comments led to improvements in this presentation.

Appendix A: Additional numerical illustrations and examples

This section complements the examples provided in the paper with further experiments and illustrations.

A.1. Examples: Non-convergence of GLASSO *with warm-starts*

This section illustrates with examples that warm-starts for the GLASSO need not converge. This is a continuation of examples presented in Section 8.

Example 1. We took $(n, p) = (2, 5)$ and setting the seed of the random number generator in R as set.seed(2008) we generated a data-matrix $\mathbf{X}_{n\times p}$ with iid standard Gaussian entries. The sample covariance matrix $\mathbf{S}$ is given below:

$$\begin{pmatrix} 0.03597652 & 0.03792221 & 0.1058585 & -0.08360659 & 0.1366725 \\ 0.03597652 & 0.03792221 & 0.1058585 & -0.08360659 & 0.1366725 \\ 0.10585853 & 0.11158361 & 0.3114818 & -0.24600689 & 0.4021497 \\ -0.08360659 & -0.08812823 & -0.2460069 & 0.19429514 & -0.3176160 \\ 0.13667246 & 0.14406402 & 0.4021497 & -0.31761603 & 0.5192098 \end{pmatrix}$$

With q denoting the maximum off-diagonal entry of $\mathbf{S}$ (in absolute value), we solved (1.1) using GLASSO at $\lambda = 0.9 \times q$. The covariance matrix for this λ was taken as a warm-start for the GLASSO algorithm with $\lambda' = \lambda \times 0.01$. The smallest eigen-value of the working covariance matrix $\mathbf{W}$ produced by the GLASSO algorithm, upon updating the first row/column was: -0.002896128, which is clearly undesirable for the convergence of the algorithm GLASSO. This is why the algorithm GLASSO breaks down.

Example 2. The example is similar to above, with $(n, p) = (10, 50)$, the seed of random number generator in R being set to set.seed(2008) and $\mathbf{X}_{n \times p}$ is the data-matrix with iid Gaussian entries. If the covariance matrix $\widehat{\mathbf{W}_\lambda}$ which solves problem (1.1) with $\lambda = 0.9 \times \max_{i \neq j} |s_{ij}|$ is taken as a warm-start to the GLASSO algorithm with $\lambda' = \lambda \times 0.1$ — the algorithm fails to converge. Like the previous example, after the first row/column update, the working covariance matrix has negative eigen-values.

Appendix B: More examples and comparisons

This section is a continuation to Section 9, in that it provides further examples comparing the performance of algorithms GLASSO and DP-GLASSO. The experimental data is generated as follows. For a fixed value of p, we generate a matrix $\mathbf{A}_{p \times p}$ with random Gaussian entries. The matrix is symmetrized by $\mathbf{A} \leftarrow (\mathbf{A} + \mathbf{A}')/2$. Approximately half of the off-diagonal entries of the matrix are set to zero, uniformly at random. All the eigen-values of the matrix $\mathbf{A}$ are lifted so that the smallest eigen-value is zero. The noiseless version of the precision matrix is given by $\boldsymbol{\Theta} = \mathbf{A} + \tau \mathbf{I}_{p \times p}$. We generated the sample covariance matrix $\mathbf{S}$ by adding symmetric positive semi-definite random noise $\mathbf{N}$ to $\boldsymbol{\Theta}^{-1}$; i.e. $\mathbf{S} = \boldsymbol{\Theta}^{-1} + \mathbf{N}$, where this noise is generated in the same manner as $\mathbf{A}$. We considered four different values of $p \in \{300, 500, 800, 1000\}$ and two different values of $\tau \in \{1, 4\}$.

For every p, τ combination we considered a path of twenty λ values on the geometric scale. For every such case four experiments were performed: Primal-Cold, Primal-Warm, Dual-Cold and Dual-Warm (as described in Section 9). Each combination was run 5 times, and the results averaged, to avoid dependencies on machine loads. Figure 4 shows the results. Overall, DP-GLASSO with warm starts performs the best, especially at the extremes of the path. We gave some explanation for this in Section 6. For the largest problems ($p = 1000$) their performances are comparable in the central part of the path (though DP-GLASSO dominates), but at the extremes DP-GLASSO dominates by a large margin.

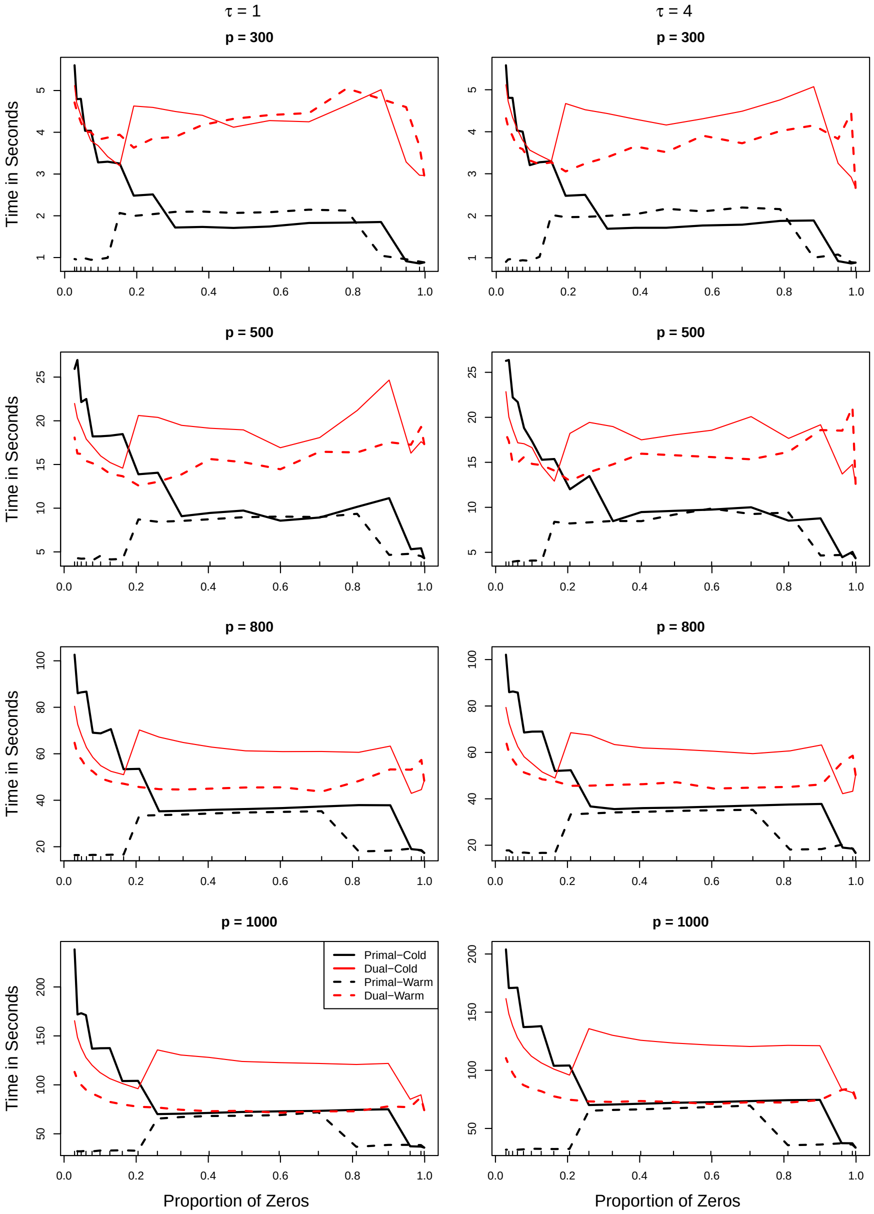

FIG 4. *The timings in seconds for the four different algorithmic versions* GLASSO *(with and without warm-starts) and* DP-GLASSO *(with and without warm-starts) for a grid of twenty λ values on the log-scale. The horizontal axis is indexed by the proportion of zeros in the solution.*

References

[1] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America* **96** 6745–6750.

[2] Banerjee, O., Ghaoui, L. E. and 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. MR2417243

[3] Beck, A. and Teboulle, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sciences* **2** 183-202. MR2486527

[4] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press. MR2061575

[5] Friedman, J., Hastie, T. and Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9** 432–441.

[6] Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction (Springer Series in Statistics)*, 2 ed. Springer New York. MR2722294

[7] Mazumder, R. and Hastie, T. (2012). Exact Covariance Thresholding into Connected Components for Large-Scale Graphical Lasso. *Journal of Machine Learning Research* **13** 781–794. MR2913718

[8] Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *Annals of Statistics* **34** 1436–1462. MR2278363

[9] Nesterov, Y. (2007). Gradient methods for minimizing composite objective function Technical Report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain. Tech. Rep, 76.

[10] Rothman, A. J., Bickel, P. J., Levina, E. and Zhu, J. (2008). Sparse Permutation Invariant Covariance Estimation. *Electronic Journal of Statistics* **2** 494-515. MR2417391

[11] Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika* **94** 19-35. MR2367824