SIAM J. Matrix Anal. Appl.
Vol. 27, No. 2, pp. 532–546

LEAST-SQUARES COVARIANCE MATRIX ADJUSTMENT[*]

STEPHEN BOYD[†] AND LIN XIAO[‡]

Abstract. We consider the problem of finding the smallest adjustment to a given symmetric $n \times n$ matrix, as measured by the Euclidean or Frobenius norm, so that it satisfies some given linear equalities and inequalities, and in addition is positive semidefinite. This least-squares covariance adjustment problem is a convex optimization problem, and can be efficiently solved using standard methods when the number of variables (i.e., entries in the matrix) is modest, say, under 1000. Since the number of variables is $n(n+1)/2$, this corresponds to a limit around $n = 45$. Malick [*SIAM J. Matrix Anal. Appl.,* 26 (2005), pp. 272–284] studies a closely related problem and calls it the semidefinite least-squares problem. In this paper we formulate a dual problem that has no matrix inequality or matrix variables, and a number of (scalar) variables equal to the number of equality and inequality constraints in the original least-squares covariance adjustment problem. This dual problem allows us to solve far larger least-squares covariance adjustment problems than would be possible using standard methods. Assuming a modest number of constraints, problems with $n = 1000$ are readily solved by the dual method. The dual method coincides with the dual method proposed by Malick when there are no inequality constraints and can be obtained as an extension of his dual method when there are inequality constraints. Using the dual problem, we show that in many cases the optimal solution is a low rank update of the original matrix. When the original matrix has structure, such as sparsity, this observation allows us to solve very large least-squares covariance adjustment problems.

Key words. matrix nearness problems, covariance matrix, least-squares, semidefinite least-squares

AMS subject classifications. 15A18, 90C06, 90C22, 90C46, 93E24

DOI. 10.1137/040609902

1. The problem.

This paper concerns the following problem. We are given a symmetric matrix G, and seek X, the nearest (in the least-squares sense) symmetric matrix that is positive semidefinite and in addition satisfies some given linear equalities and inequalities. This can be expressed as the optimization problem

$$\begin{array}{ll} \text{minimize} & (1/2)\|X-G\|_F^2 \\ \text{subject to} & X \succeq 0, \\ & \mathbf{Tr}\, A_i X = b_i, \quad i = 1,\dots,p, \\ & \mathbf{Tr}\, C_j X \le d_j, \quad j = 1,\dots,m. \end{array} \tag{1.1}$$

Here the matrices X, G, A_i, and C_j are $n \times n$ and symmetric (and real). The (real) matrix X is the optimization variable; the matrices G, A_i, and C_j, and the scalars b_i and d_j, are the problem data. We note that the optimization variable X has dimension $n(n+1)/2$, i.e., it contains $n(n+1)/2$ independent scalar variables. In (1.1), $\mathbf{Tr}$ denotes the trace of a matrix, $\succeq$ denotes matrix inequality, and $\|\cdot\|_F$

[*] Received by the editors June 11, 2004; accepted for publication (in revised form) by L. Vandenberghe April 4, 2005; published electronically November 22, 2005.
http://www.siam.org/journals/simax/27-2/60990.html

[†] Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510 (boyd@stanford.edu).

[‡] Center for the Mathematics of Information, California Institute of Technology, Pasadena, CA 91125-9300 (lxiao@caltech.edu).

denotes the Frobenius norm, i.e.,

$$\|U\|_F = (\mathbf{Tr}\, U^T U)^{1/2} = \left(\sum_{i,j=1}^{n} U_{ij}^2 \right)^{1/2}.$$

Since any real-valued linear function f on the set of $n \times n$ symmetric matrices can be expressed as $f(U) = \mathbf{Tr}\, AU$ for some symmetric $n \times n$ matrix A, we see that the constraints in (1.1) are a general set of p linear equality constraints and m linear inequality constraints. For reasons to be explained below, we refer to problem (1.1) as the *least-squares covariance adjustment problem* (LSCAP).

LSCAP is a type of *matrix nearness problem*, i.e., the problem of finding a matrix that satisfies some property and is nearest to a given one. In the least-squares covariance adjustment problem, we use the Frobenius norm, the given matrix is symmetric, and the property is that the matrix belongs to a polyhedron, i.e., the solution set of some linear equalities and inequalities. See Higham [6] for a recent survey of matrix nearness problems. A related geometric interpretation of the LSCAP (1.1) is as follows: We are given a symmetric matrix G, and wish to compute the (Frobenius norm) projection of G onto the intersection of a general polyhedron, described by linear equalities and inequalities, and the cone of positive semidefinite matrices.

LSCAP (1.1) comes up in several contexts. One is in making adjustments to a symmetric matrix so that it is consistent with prior knowledge or assumptions, and is a valid covariance matrix. We interpret X as the covariance matrix of a zero mean random n-vector z, and suppose that the variances of some linear functions of z are known, e.g.,

$$\mathbf{E}\, \|F_i^T z\|^2 = \mathbf{Tr}\, F_i^T X F_i = b_i, \qquad i = 1, \ldots, p.$$

We are given G, which is some approximation of X (e.g., a sample covariance matrix), and wish to find the covariance matrix nearest G that is also consistent with the prior information. This results in problem (1.1) with $A_i = F_i F_i^T$ (using $\mathbf{Tr}\, \|F_i X\|^2 = \mathbf{Tr}\, X(F_i F_i^T)$). If we are also given some lower or upper bounds on the variances of some linear combinations of z, we obtain an LSCAP with inequality constraints.

We will also consider a special case of the basic problem (1.1), in which there is only one equality constraint, defined by a matrix of rank one:

$$\begin{array}{rl} \text{minimize} & (1/2)\|X - G\|_F^2 \\ \text{subject to} & X \succeq 0, \quad c^T X c = b. \end{array} \tag{1.2}$$

This is the problem of finding the covariance matrix closest to G that matches the given variance of one given linear function of the underlying random variable.

1.1. Previous work.

Some special cases of the LSCAP have simple solutions, or have been addressed in the literature. The simplest case is when there are no constraints on X other than positive semidefiniteness. In this case the LSCAP reduces to finding the projection of a symmetric matrix onto the positive semidefinite cone, which has a well-known solution based on the eigenvalue decomposition of G (given in section 1.3). Another example is the problem of finding the nearest correlation matrix, i.e., the nearest symmetric positive semidefinite matrix with unit diagonal elements. This particular problem is addressed in Higham [7], where an alternating projections method is proposed for solving the problem.

There is a large literature on matrix nearness problems; see, for example, the references in the survey [6]. In much of this work, the original matrix (which we denote G) is nonsymmetric, which complicates the problem a bit. For example, in [5], Higham considers the problem of finding the nearest positive semidefinite matrix to a given (nonsymmetric) matrix with no other constraints.

Our approach and methods are related to others that have been proposed for other matrix nearness problems. In particular, we find ideas related to the dual of the original problem in, e.g., [7]; the idea of exploiting structure to evaluate the projection of a matrix onto the positive semidefinite cone also appears in [7]. Our method for the special case (1.2) involves a one parameter search, which can be done by a bisection algorithm, or a guarded Newton method. This is similar to methods proposed in [4, 5] and several of the papers cited in Higham's survey paper for (other) matrix nearness problems.

By far the closest previous work is the recent paper [11] by Malick, which appeared while this paper was under review. Malick considers the general problem of computing the Euclidean projection of a point onto the intersection of a cone and an affine set, and the special case in which the cone is the positive semidefinite cone, which he calls *semidefinite least-squares* (SDLS). His SDLS problem coincides with our LSCAP, with no equality constraints; the inequality constraints can, however, be handled by introducing slack variables, and including the slack nonnegativity constraints in the cone, so SDLS can be said to be more general than LSCAP. Our approach and Malick's approach are very similar with some differences that we will point out in what follows. In fact, some of our ideas and Malick's can be combined to create a method that can very efficiently solve large-scale SDLS problems.

1.2. Basic properties of LSCAP. The LSCAP (1.1) is a *convex optimization problem*, since the objective is convex, the constraint functions are linear, and the semidefiniteness inequality $X \succeq 0$ is convex. (The constraint $X \succeq 0$ is called a generalized or cone constraint, or more specifically, a linear matrix inequality.) If the problem (1.1) is feasible, then it has a unique solution, since the objective function is strictly convex and has bounded sublevel sets. We will denote the solution as $X^\star$.

The observation that the LSCAP is a convex optimization problem have several important implications. The first is that the LSCAP is readily solvable, by standard interior-point methods, at least when the number of variables is modest, say, under 1000 (which corresponds to n around 45). General purpose interior-point solvers (that do not exploit problem structure) have a complexity that scales at least as fast as the number of variables cubed, so the overall complexity of solving the LSCAP problem, using this approach, scales at least as fast as n^6. For background on convex optimization and interior-point methods, see, e.g., [3].

The second consequence of convexity of LSCAP is that its associated *dual problem* (defined below) is sharp (provided a technical condition holds) and can be used to solve the original LSCAP. This is the central idea of our method, and is described in more detail in section 2.

The LSCAP problem is closely related to *semidefinite programming* (SDP) [18, 17, 3]. Like LSCAP, SDP is an optimization problem with a symmetric matrix variable, a semidefiniteness constraint, and linear equality and inequality constraints. In an SDP, the objective is a linear function; in LSCAP the objective is (a simple) convex quadratic function. In fact, it is possible to express the LSCAP problem as an SDP, using a standard trick for converting a quadratic objective into a linear matrix inequality and a linear objective (see, e.g., [18]). In principal, this allows us to solve the

LSCAP using standard algorithms for solving SDPs. Without exploiting any particular structure in the resulting SDP, however, this approach will be quite inefficient. The observation that LSCAP (in the SDLS form) is convex, and can be formulated as an SDP, but one that is difficult to solve using the standard methods can be found in Malick [11] and Higham [7].

1.3. Projection on the positive semidefinite cone.

Here we introduce some notation and standard material that we will use in what follows. For a symmetric $n \times n$ matrix X, we sort the eigenvalues in decreasing order,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

and use the notation $\lambda_i(X)$, for example, to specify the matrix if it is not clear.

The positive and negative semidefinite parts of X, denoted by X_+ and X_-, respectively, are defined implicitly by the conditions

$$X = X_+ - X_-, \quad X_+ = X_+^T \succeq 0, \quad X_- = X_-^T \succeq 0, \quad X_+ X_- = 0.$$

The positive semidefinite part X_+ is the projection of X onto the positive semidefinite cone, i.e., we have

$$\|X - X_+\|_F = \|X_-\|_F \leq \|X - Z\|_F$$

for any positive semidefinite Z. In a similar way, $\|X + Z\|_F$ is minimized, over all positive semidefinite matrices Z, by the choice $Z = X_-$ (see, e.g., [3, section 8.1.1] or [8]).

We can express the positive and negative semidefinite parts explicitly as

$$X_+ = \sum_{\lambda_i > 0} \lambda_i q_i q_i^T, \qquad X_- = \sum_{\lambda_i < 0} -\lambda_i q_i q_i^T,$$

where $X = \sum_{i=1}^n \lambda_i q_i q_i^T$ is an eigendecomposition of X, i.e., $q_1, \ldots, q_n$ is a set of orthonormal eigenvectors of X with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. We also note, for future reference, that to compute X_+ and X_-, it suffices to find the negative eigenvalues of X and the associated eigenvectors; X_+ can then be computed as $X_+ = X + X_-$.

2. The dual problem.

2.1. Lagrangian and dual function.

We introduce the Lagrange multipliers $\nu_1, \ldots, \nu_p$ associated with the equality constraints $\mu_1, \ldots, \mu_m$ associated with the inequality constraints, and the symmetric $n \times n$ matrix Z associated with the matrix inequality $X \succeq 0$ (which we write as $-X \preceq 0$). The Lagrangian of problem (1.1) is then

$$\begin{aligned} L(X, Z, \nu, \mu) &= (1/2)\|X - G\|_F^2 - \mathbf{Tr}\, ZX + \sum_{i=1}^p \nu_i (\mathbf{Tr}\, A_i X - b_i) + \sum_{j=1}^m \mu_j (\mathbf{Tr}\, C_j X - d_j) \\ &= (1/2)\|X - G\|_F^2 + \mathbf{Tr}\, X \left(-Z + \sum_{i=1}^p \nu_i A_i + \sum_{j=1}^m \mu_j C_j \right) - \nu^T b - \mu^T d \\ &= (1/2)\|X - G\|_F^2 + \mathbf{Tr}\, X \left(-Z + A(\nu) + C(\mu) \right) - \nu^T b - \mu^T d, \end{aligned}$$

where we use the notation

$$A(\nu) = \sum_{i=1}^{p} \nu_i A_i, \qquad C(\nu) = \sum_{j=1}^{m} \mu_j C_j.$$

To form the dual function, we must minimize the Lagrangian over X, which we can do by setting the gradient with respect to X to zero (since L is convex quadratic in X). This yields

$$X - G - Z + A(\nu) + C(\mu) = 0,$$

so the X that minimizes L is given by

$$X = G - A(\nu) - C(\mu) + Z. \tag{2.1}$$

Substituting this expression for X back into the Lagrangian we obtain the dual function:

$$\begin{aligned} g(Z,\nu,\mu) &= \inf_X L(X,Z,\nu,\mu) \\ &= (1/2)\| - A(\nu) - C(\mu) + Z\|_F^2 - \nu^T b - \mu^T d \\ &\quad + \mathbf{Tr}\,(G - A(\nu) - C(\nu) + Z)\,(-Z + A(\nu) + C(\mu)) \\ &= -(1/2)\| - A(\nu) - C(\mu) + Z + G\|_F^2 + (1/2)\|G\|_F^2 - \nu^T b - \mu^T d. \end{aligned}$$

This is a concave quadratic function of the dual variables Z, ν, and μ.

2.2. Dual problem and properties. The dual problem is

$$\begin{array}{ll} \text{maximize} & -(1/2)\| - A(\nu) - C(\mu) + Z + G\|_F^2 + (1/2)\|G\|_F^2 - \nu^T b - \mu^T d \\ \text{subject to} & Z \succeq 0, \qquad \mu \succeq 0, \end{array} \tag{2.2}$$

where the symbol $\succeq$ between vectors means elementwise, and the (dual) variables are Z, ν, and μ. This problem is similar to the original LSCAP: it has a symmetric matrix variable (as well as two vector variables), a quadratic objective, a positive semidefiniteness constraint, and a set of (scalar) nonnegativity constraints.

Weak duality always holds for the dual problem (2.2): if Z, ν, and μ are dual feasible, i.e., $Z \succeq 0$ and $\mu \succeq 0$, then the dual objective is a lower bound on the optimal value of the LSCAP (1.1). If the LSCAP (1.1) is *strictly feasible*, i.e., there exists an $X \succ 0$ that satisfies the linear equalities and inequalities in (1.1), then *strong duality* holds: there exist $Z^\star$, $\nu^\star$, $\mu^\star$ that are optimal for the dual problem (2.2) with dual objective equal to the optimal value of the LSCAP (1.1). Moreover, we can recover the optimal solution of the LSCAP (1.1) from the dual optimal variables using

$$X^\star = G - A(\nu^\star) - C(\mu^\star) + Z^\star \tag{2.3}$$

(which is the choice of X that minimizes the Lagrangian, when $Z = Z^\star$, $\nu = \nu^\star$, and $\mu = \mu^\star$). This follows from convexity of the original LSCAP, and strict convexity of the Lagrangian with respect to the primal variable X; see, e.g., [3].

Although we will end up deriving this result directly, general convex optimization theory tells us that $X^\star$ and $Z^\star$ satisfy the *complementarity condition* $Z^\star X^\star = 0$. Since $X^\star$ and $Z^\star$ are both positive semidefinite, this means that the intersection of their ranges is $\{0\}$.

2.3. Simplified dual problem. It is possible to analytically maximize over the variable Z in (2.2), using the fact (from section 1.3) that

$$\|-A(\nu) - C(\mu) + Z + G\|_F^2$$

is minimized over the choice of $Z \succeq 0$ by choosing

$$Z = (G - A(\nu) - C(\mu))_-\,, \tag{2.4}$$

which gives

$$\begin{aligned}\|-A(\nu) - C(\mu) + Z + G\|_F^2 &= \left\|(G - A(\nu) - C(\mu))_+\right\|_F^2 \\ &= \sum_{\lambda_i > 0} \lambda_i^2\,(G - A(\nu) - C(\mu))\,.\end{aligned}$$

Using the result (2.4), we can simplify the dual problem (2.2) to

$$\begin{array}{ll}\text{maximize} & -(1/2)\,\|(G - A(\nu) - C(\mu))_+\|_F^2 + (1/2)\|G\|_F^2 - \nu^T b - \mu^T d \\ \text{subject to} & \mu \succeq 0,\end{array} \tag{2.5}$$

where the variables are ν and μ. We will refer to this as the *simplified dual LSCAP problem*. The simplified dual problem has no matrix variable or inequality and has a number of (scalar) variables equal to the number of equality and inequality constraints in the original LSCAP.

The simplified dual problem is, of course, a convex optimization problem (since the objective is concave, and is to be maximized). We also note that the objective function is differentiable, but not twice differentiable. To see this we define ϕ, for symmetric W, as

$$\phi(W) = (1/2)\|W_+\|_F^2 = (1/2)\sum_i \left(\max\{0, \lambda_i(W)\}\right)^2,$$

the sum of the squares of the positive eigenvalues. Its gradient can be expressed as

$$\nabla\phi(W) = W_+, \tag{2.6}$$

by which we mean the following: for symmetric V, we have

$$\phi(W + V) = \phi(W) + \mathbf{Tr}\, V W_+ + o(V).$$

This can be verified several ways, for example, using standard perturbation theory for the eigenvalues of a symmetric matrix, or results from Borwein and Lewis [2, section 5.2].

From the formula (2.6), we conclude that $\nabla\phi(W)$ is continuous, indeed, with Lipschitz constant one: for any symmetric V and W,

$$\|\nabla\phi(V) - \nabla\phi(W)\|_F \le \|V - W\|_F.$$

Whenever W has no zero eigenvalues, $\nabla\phi(W)$ is differentiable; but when W has a zero eigenvalue, $\nabla\phi(W)$ is not differentiable. In other words, ϕ is twice differentiable, except at points W with a zero eigenvalue.

Using (2.6) we can find expressions for the gradient of the objective of the simplified dual (2.5), which we denote

$$\psi(\nu,\mu) = -(1/2)\,\|(G - A(\nu) - C(\mu))_+\|_F^2 + (1/2)\|G\|_F^2 - \nu^T b - \mu^T d.$$

We have

$$\frac{\partial\psi}{\partial\nu_i} = \mathbf{Tr}(G - A(\nu) - C(\mu))_+ A_i - b_i, \qquad \frac{\partial\psi}{\partial\mu_j} = \mathbf{Tr}(G - A(\nu) - C(\mu))_+ C_j - d_j.$$

Defining $X(\nu,\mu)$ as $X(\nu,\mu) = (G - A(\nu) - C(\mu))_+$, we see that the gradients of the dual objective can be expressed as

$$\frac{\partial\psi}{\partial\nu_i} = \mathbf{Tr}\, A_i X(\nu,\mu) - b_i, \qquad \frac{\partial\psi}{\partial\mu_j} = \mathbf{Tr}\, C_j X(\nu,\mu) - d_j,$$

which are exactly the residuals for the constraints in the primal LSCAP with $X = X(\nu,\mu)$.

The simplified dual (2.5) coincides with the dual derived by Malick for the case when there are no inequality constraints in the LSCAP, and is obtained as a simple extension of his dual, when inequality constraints are included (see [11, Remark 4.3]). He also observes that the dual objective is differentiable, and has a gradient with Lipschitz constant one.

From (2.3) and (2.4) we find that

$$X^\star = (G - A(\nu^\star) - C(\mu^\star))_+. \tag{2.7}$$

The formula (2.7) shows that the optimal solution of the LSCAP problem is always obtained by adding a linear combination of the constraint data matrices to the original matrix, and then projecting the result onto the positive semidefinite cone.

More generally, suppose that ν and μ are *any* feasible values of the dual variables (i.e., $\mu_j \geq 0$), not necessarily optimal. Then the dual objective $\psi(\nu,\mu)$ gives a lower bound on the optimal value of the primal LSCAP. The matrix $X(\nu,\mu)$ is always positive semidefinite (since it is the positive semidefinite part of $G - A(\nu) - C(\mu)$), but does not satisfy the original equality and inequality constraints, unless ν and μ are optimal.

2.4. Rank of optimal adjustment. Consider $X^\star - G$, which is the optimal adjustment made to G. We have

$$\begin{aligned} X^\star - G &= (G - A(\nu^\star) - C(\mu^\star))_+ - G \\ &= -A(\nu^\star) - C(\mu^\star) + (G - A(\nu^\star) - C(\mu^\star))_-. \end{aligned} \tag{2.8}$$

This shows that the optimal adjustment made to G is a linear combination of the constraint matrices, plus the negative part of $G - A(\nu^\star) - C(\mu^\star)$.

We can draw many interesting conclusions from this observation. We start by observing that the number of negative eigenvalues of $G - A(\nu^\star) - C(\mu^\star)$ is no more than the number of negative eigenvalues of G, plus the sum of the ranks of A_i, plus the sum of the ranks of C_j. In other words, we have

$$\mathbf{Rank}(G - A(\nu^\star) - C(\mu^\star))_- \leq \mathbf{Rank}(G_-) + \sum_i \mathbf{Rank}(A_i) + \sum_j \mathbf{Rank}(C_j). \tag{2.9}$$

We also have

$$\mathbf{Rank}(-A(\nu^\star) - C(\mu^\star)) \le \sum_i \mathbf{Rank}(A_i) + \sum_j \mathbf{Rank}(C_j),$$

so, using the result (2.8) above, we conclude

$$\mathbf{Rank}(X^\star - G) \le \mathbf{Rank}(G_-) + 2\sum_i \mathbf{Rank}(A_i) + 2\sum_j \mathbf{Rank}(C_j).$$

This shows that if G is positive semidefinite, and there are only a small number of constraints, with low rank data matrices, then the optimal adjustment is low rank, i.e., $X^\star$ is a low rank update of G.

As another example, suppose that G is a *factor model covariance matrix*; i.e., it can be expressed as a sum of a nonnegative diagonal matrix and a positive semidefinite matrix of low rank, say, k. (This means that the underlying random variables can be explained by k common factors, plus some independent noise in each component.) We also assume that the constraint matrices A_i and C_j are rank one and positive semidefinite. For this example, the rank of the optimal adjustment $X^\star - G$ cannot exceed $2(p + m)$, so the optimal adjusted covariance matrix $X^\star$ will also be a factor model (indeed, with the same diagonal part as G, and a number of additional factors not exceeding twice the number of equalities and inequalities).

3. Solution via simplified dual.

In many cases, we can solve the LSCAP (1.1) efficiently via the simplified dual problem (2.5), which has differentiable (but not twice differentiable) objective, a number of variables equal to the number of constraints in the original LSCAP, and nonnegativity constraints on some of the variables.

Several methods can be used to solve the general simplified dual problem. One very simple method is the projected (sub)gradient method, described in detail in section 3.1; another possibility is a conjugate gradient method that can handle nonnegativity constraints. Various special cases of the simplified dual can be handled by more specialized methods. For example, if there is only one equality constraint, the dual objective function can be shown to be twice differentiable, so Newton's method can be used. (This is described in more detail in section 3.4.3.)

If there are no inequality constraints in the original LSCAP (1.1), the simplified dual is an unconstrained problem, so any method for unconstrained minimization of a differentiable (but not twice differentiable) function can be used. For this case, Malick proposes a quasi-Newton method (BFGS) for the simplified dual problem [11]. Such a method will yield faster convergence than a simple gradient method for LSCAP problems with a few thousand equality constraints. For problems with more than a few thousand constraints, methods that do not require forming or storing a $p \times p$ matrix, such as conjugate gradients, might be preferable.

3.1. Dual projected gradient algorithm.

In this section we describe in detail one simple method that can be used to solve the simplified dual problem (2.5). The projected (sub)gradient method repeats the following steps:

1. *Update X.* Set $X := (G - A(\nu) - C(\mu))_+$.
2. *Projected gradient update for μ and ν.*
   (a) Evaluate primal residuals/dual gradients,

$$\frac{\partial \psi}{\partial \nu_i} = \mathbf{Tr}\, A_i X - b_i, \qquad \frac{\partial \psi}{\partial \mu_j} = \mathbf{Tr}\, C_j X - d_j.$$

(b) Set

$$\nu_i := \nu_i + \alpha(\mathbf{Tr}\, A_i X - b_i), \qquad \mu_j := \left(\mu_j + \alpha(\mathbf{Tr}\, C_j X - d_j)\right)_+ .$$

Here $\alpha > 0$ is a step size parameter. The iteration is stopped when all ν_i and μ_j, the equality constraint residuals and the positive parts of the inequality constraint residuals, are small enough.

Assuming the original LSCAP is strictly feasible, this algorithm is guaranteed to converge, i.e., ν, μ, and X converge to their optimal values, provided α is small enough. This follows from standard analysis of the subgradient algorithm, specialized to the case of differentiable objective; see, for example, [15, 1, 13, 10, 16].

We can give a specific bound on α that guarantees convergence. In the general case, convergence is guaranteed whenever $\alpha < 2/L$, where L is a Lipschitz constant for the gradient of the dual objective, i.e., the mapping from (ν, μ) to $(\partial\psi/\partial\nu, \partial\psi/\partial\mu)$. We can derive a Lipschitz constant for this mapping by noting that it is the composition of three mappings:

- The affine mapping from (ν, μ) to $G - A(\nu) - C(\mu)$.
- Projection onto the positive semidefinite cone, which yields X.
- The affine mapping from X to $(\partial\psi/\partial\nu, \partial\psi/\partial\mu)$.

The first affine mapping has a Lipschitz constant

$$\left(\sum_{i=1}^{p} \|A_i\|^2 + \sum_{j=1}^{m} \|C_j\|^2\right)^{1/2},$$

where the norm is the spectral norm (i.e., maximum singular value). The projection has a Lipschitz constant one. The final affine mapping has a Lipschitz constant

$$\left(\sum_{i=1}^{p} \|A_i\|^2 + \sum_{j=1}^{m} \|C_j\|^2\right)^{1/2}.$$

Thus, the gradient has a Lipschitz constant

$$L = \sum_{i=1}^{p} \|A_i\|^2 + \sum_{j=1}^{m} \|C_j\|^2.$$

While we are guaranteed that the algorithm converges for $\alpha < 2/L$, such a choice usually yields slow convergence.

We mention another step size strategy that guarantees convergence, and does not depend on the problem data at all. We choose the step size as a function of the iteration, i.e., we use step size α_k in iteration k. Convergence is guaranteed provided the step size sequence α_k satisfies the simple conditions

$$\alpha_k \geq 0, \qquad \lim_{k\to\infty} \alpha_k = 0, \qquad \sum_k \alpha_k = \infty.$$

For example, the algorithm is guaranteed to converge, for any problem data, with the universal choice $\alpha_k = 1/k$. (See, e.g., [15, 16] for more on step size strategies for projected subgradient methods.)

3.2. Complexity analysis. No matter what method is used to solve the simplified dual problem, each iteration requires evaluating the gradient of the dual objective, i.e., steps 1 and 2(a) of the projected gradient algorithm described above. For the projected gradient algorithm, this evaluation dominates the effort per step; for other methods, such as conjugate gradients or quasi-Newton methods, the cost of an iteration is at least the cost of evaluating the dual objective gradient. In this section and the next we analyze the cost of evaluating the dual objective gradient.

We first assume no structure in the problem data, i.e., G, A_i, C_j are given as full, high rank $n \times n$ matrices. We compute X by first forming $A(\nu)$ and $C(\mu)$, which costs order $n^2(p+m)$ flops, then computing the complete eigendecomposition of $G - A(\nu) - C(\mu)$, which is order n^3 flops, and finally forming X (which is order n^3 flops). The cost of step 2(a) is $n^2(p+m)$, so the overall cost of one dual objective gradient evaluation is therefore order $n^2 \max\{n, p+m\}$.

If there are few constraints (compared to n), the cost is dominated by the complete eigendecomposition of $G - A(\nu) - C(\mu)$, which has a cost around $10n^3$. One dual objective gradient evaluation (and therefore one iteration of the dual projected gradient algorithm) can be carried out on a current personal computer, for $n = 1000$, in at most a few seconds; assuming on the order of several hundred iterations, we can solve such an LSCAP in minutes. This dimension is far beyond the size of an LSCAP that can be solved using a standard interior-point method directly, since the problem has order 10^6 variables (entries in X).

3.3. Exploiting structure. In many interesting cases we can exploit structure in the LSCAP data to reduce the complexity of a dual objective gradient evaluation far below order n^3. In this section we describe a generic example that illustrates the basic idea.

We assume that G is a positive definite matrix for which we can carry out matrix-vector multiplication efficiently, such as a sparse matrix, block diagonal matrix, or a diagonal matrix plus a matrix of low rank, given in factored form. We also assume that the data matrices A_i and C_j are all low rank and given in factored form (such as LDL^T). We let k denote the total of the ranks of the constraint data matrices, and we assume that $k \ll n$. Following (2.9) we see that for any ν and μ, we have $\mathbf{Rank}(G - A(\nu) - C(\mu))_- \leq k$.

The matrix-vector multiplication $y \to (G - A(\nu) - C(\mu))y$ can be carried out efficiently, since Gy is efficiently computed, and $A_i y$ and $C_j y$ are efficiently computed using their factored forms. Therefore, we can use a Lanczos or subspace iteration method to compute the k eigenvalues $\lambda_{n-k-1}, \ldots, \lambda_n$, as well as the associated eigenvectors, of $G - A(\nu) - C(\mu)$ (see, e.g., [12, 14]). This can be done very efficiently, since these methods require only that we multiply a given vector by $G - A(\nu) - C(\mu)$. Choosing the negative eigenvalues from these, and using the associated eigenvectors, we form $(G - A(\nu) - C(\mu))_-$, storing it in factored form. We then "form" $X = (G - A(\nu) - C(\mu))_+ = (G - A(\nu) - C(\mu)) + (G - A(\nu) - C(\mu))_-$, storing it as G (or really, the subroutine that computes Gy) along with the low rank, factored representation of $(G - A(\nu) - C(\mu))_-$.

To evaluate the gradient of the dual objective, we must compute $\mathbf{Tr}\, A_i X$ and $\mathbf{Tr}\, C_j X$. But these are readily computed using the factored form of A_i and C_j, and using the fact that $y \to Xy$ can be efficiently computed.

We note that this observation applies for *any* method for solving the simplified dual problem, and not just the dual projected gradient approach described above. In particular, it can be used with Malick's BFGS method for the SDLS problem.

3.4. Special case: One rank one constraint. In this section we consider the special case with one rank one equality constraint (1.2),

$$\begin{array}{ll} \text{minimize} & (1/2)\|X - G\|_F^2 \\ \text{subject to} & X \succeq 0, \quad c^T X c = b, \end{array}$$

where $G \succeq 0$. Without loss of generality we can assume that $\|c\| = 1$. We assume that $b > 0$, so this problem is strictly feasible, which implies that the solution has the form

$$X^\star = (G - \nu^\star cc^T)_+,$$

where $\nu^\star$ satisfies $c^T(G - \nu^\star cc^T)_+ c = b$ (see (2.7)). The residual $c^T(G - \nu cc^T)_+ c - b$ is monotone nonincreasing in ν. We now find a lower and upper bound on $\nu^\star$.

We define $\underline{\nu}$ as the value of ν which satisfies $c^T(G - \nu cc^T)c = b$, i.e., $\underline{\nu} = c^T Gc - b$. If $G - \underline{\nu} cc^T \succeq 0$, then $\nu^\star = \underline{\nu}$. Otherwise,

$$c^T(G - \underline{\nu} cc^T)_+ c - b \geq c^T(G - \underline{\nu} cc^T)c - b = 0,$$

which implies that $\nu^\star \geq \underline{\nu}$. Thus, $\underline{\nu}$ is a lower bound on the optimal solution $\nu^\star$.

Now we derive an upper bound $\overline{\nu}$ on $\nu^\star$. We first observe that

$$\psi(\nu) = -(1/2)\|(G - \nu cc^T)_+\|_F^2 + (1/2)\|G\|_F^2 - \nu b \leq (1/2)\|G\|_F^2 - \nu b.$$

Since the primal objective function is nonnegative, the optimal dual variable $\nu^\star$ must have a nonnegative dual objective value. Using the inequality above, we have

$$0 \leq \psi(\nu^\star) \leq (1/2)\|G\|_F^2 - \nu^\star b.$$

From this we obtain

$$\nu^\star \leq \bar{\nu} = \|G\|_F^2/(2b).$$

3.4.1. Bisection. We can now find $\nu^\star$ by bisection, starting from the initial bounds $\underline{\nu}$ and $\overline{\nu}$. At each iteration we must compute λ_n, the unique negative eigenvalue of $G - \nu cc^T$, and the associated unit eigenvector q, and then evaluate

$$\begin{aligned} c^T(G - \nu cc^T)_+ c - b &= c^T(G - \nu cc^T)c - \lambda_n (c^T q)^2 - b \\ &= c^T Gc - b - \nu - \lambda_n (c^T q)^2 \\ &= \tilde{\nu} - \nu - \lambda_n (c^T q)^2. \end{aligned}$$

We have observed that in most cases, $\nu^\star$ is much closer to the lower bound $\underline{\nu}$ than to the upper bound $\overline{\nu}$. This suggests carrying out the bisection using the geometric mean of the current bounds as the next iterate, instead of the usual arithmetic mean. This was suggested by Byers for another matrix nearness problem [4].

3.4.2. Exploiting structure. We can exploit structure in G to compute the initial bounds and carry out the bisection iterations efficiently. Suppose, for example, that G is large, but it is easy to compute the matrix-vector product Gy. Then we can compute λ_n by a power method, for example, with a shift to ensure the method converges to the negative eigenvalue. If we can efficiently compute $(G - \nu cc^T)^{-1} y$ (e.g., if G is sparse, and we are able to find a sufficiently sparse Cholesky factor), we can compute the eigenvalue λ_n and eigenvector q using inverse iteration with shifts, which requires only a handful of iterations.

The upper bound requires computing $\|G\|_F^2$, which can also be done efficiently in many cases, by exploiting structure in G. This is clear when G is sparse, since $\|G\|_F^2$ is just the sum of the squares of its nonzero entries. As a more interesting example, suppose G is a factor matrix, given as $D_1 + LD_2L^T$, where L has $k \ll n$ columns and D_1 and D_2 are diagonal. Even though G is full, its Frobenius norm can be efficiently computed using the formula

$$\begin{aligned}\|G\|_F^2 &= \|D_1\|_F^2 + 2\,\mathbf{Tr}\,D_1LD_2L^T + \mathbf{Tr}\,LD_2L^TLD_2L^T\\ &= \|D_1\|_F^2 + 2\,\mathbf{Tr}(LD_2)^T(D_1L) + \mathbf{Tr}(D_2L^TL)(D_2L^TL)\\ &= \|D_1\|_F^2 + 2\,\mathbf{Tr}(LD_2)^T(D_1L) + \|D_2L^TL\|_F^2.\end{aligned}$$

The right-hand side can be calculated in order k^2n flops (as compared to kn^2, if we form the matrix G and then the sum of the squares of it entries).

3.4.3. Newton's method. For this special case, the residual $r(\nu) = c^T(G - \nu cc^T)_+c - b$ is differentiable (in fact, analytic), so we can use Newton's method to find its root $\nu^\star$. (This is the same as using Newton's method to maximize the dual objective function, since r is the derivative of the dual objective.) The derivative of the residual (i.e., the second derivative of the dual objective) is given by

$$\dot{r} = -1 - \dot{\lambda}_n(c^Tq)^2 - 2\lambda_n(c^Tq)(c^T\dot{q}),$$

where we use the superscript dot to denote the derivative with respect to ν.

Now we recall some results from standard perturbation theory (see, e.g., [9]). Suppose the matrix A is symmetric, depends on a scalar parameter ν, and has isolated eigenvalue λ with associated unit eigenvector q. Then we have

$$\dot{\lambda} = q^T\dot{A}q, \qquad \dot{q} = -(A - \lambda I)^\dagger \dot{A}q,$$

where $U^\dagger$ denotes the Moore–Penrose pseudoinverse. Applying these results here, we have

$$\dot{\lambda}_n = -(c^Tq)^2, \qquad \dot{q} = (c^Tq)(G - \nu cc^T - \lambda_n I)^\dagger c.$$

Substituting these expressions into the equation above, we obtain

$$\dot{r} = -1 + (c^Tq)^4 - 2\lambda_n(c^Tq)^2c^T(G - \nu cc^T - \lambda_n I)^\dagger c.$$

Now we describe a simple guarded Newton method, which maintains an interval $[l, u]$ known to contain $\nu^\star$ at each step, and whose width decreases by at least a fixed fraction each step (thus guaranteeing convergence). For an interval $[l, u]$ and $\alpha > 0$, we let $\alpha[l, u]$ denote the interval scaled by α about its center, i.e.,

$$\alpha[l, u] = [(u + l)/2 - \alpha(u - l)/2,\ (u + l)/2 + \alpha(u - l)/2)].$$

The guarded Newton method starts with the initial interval $[l, u] = [\underline{\nu}, \overline{\nu}]$, an initial value ν inside the interval (such as $\nu = (u + l)/2$), and a guard parameter value α that satisfies $0 \le \alpha < 1$. The following steps are then repeated, until $r(\nu)$ is small enough:

1. *Calculate Newton update.* Find the (pure) Newton update $\nu_{\text{nt}} = \nu - r(\nu)/\dot{r}(\nu)$.
2. *Project onto guard interval.* Set ν to be the projection of ν_{nt} onto the interval $\alpha[l, u]$.
3. *Update guard interval.* If $r(v) > 0$, set $l = \nu$; otherwise set $u = \nu$.

When $\alpha = 0$ this guarded Newton algorithm reduces to bisection. For $\alpha > 0$, the algorithm has quadratic terminal convergence. As a result, it converges to high accuracy very quickly; on the other hand, it requires computing $(G - \lambda_n cc^T)^\dagger c$ at each iteration. If no structure is exploited, this does not increase the order of the

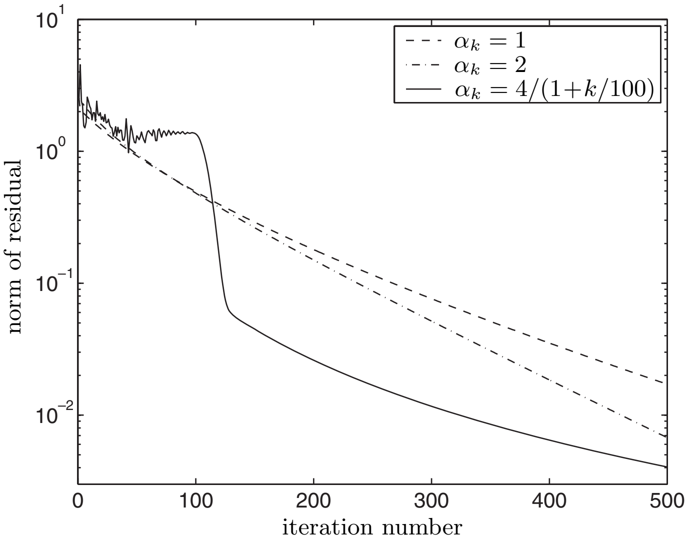

FIG. 1. *Norm of residual for the large example.*

computational cost (which is still n^3). But this method makes it difficult or impossible to exploit structure such as sparsity in G.

4. Examples. Our first example is a large problem. The matrix G is a sparse positive semidefinite matrix with dimension $n = 10^4$ and around 10^6 nonzero entries. It was generated by choosing a random sparse matrix Z, with unit Gaussian nonzero entries, and setting $G = Z^T Z$. (The density of Z was chosen so that G has around 10^6 nonzero entries.) The problem has $p = 50$ equality constraints and $m = 50$ inequality constraints. The coefficient matrices A_i and C_j are all rank one, $A_i = a_i a_i^T$, $C_j = c_j c_j^T$ with a_i and c_j chosen randomly and normalized. The coefficients b_i and d_j were chosen as $b_i = \eta_i a_i^T G a_i$, where η_i are random with uniform distribution on $[0.5, 1.5]$.

The method described in section 3.1 was used with constant step $\alpha = 1$, constant step size $\alpha = 2$, and diminishing step size rule with $\alpha_k = 4/(1 + k/100)$. The convergence of the norm of the residual,

$$\left(\sum_{i=1}^{p} (\mathbf{Tr}\, A_i X - b_i)^2 + \sum_{j=1}^{m} (\max\, \{0,\ \mathbf{Tr}\, C_j X - d_j\})^2 \right)^{1/2},$$

is shown in Figure 1, and the convergence of the dual objective is given in Figure 2. Convergence to modest accuracy occurs in about 400 iterations.

Our second example has the special form (1.2), with G a dense randomly chosen positive semidefinite 100×100 matrix, and the vector c randomly generated and normalized. We solved this problem using the bisection method, using both the arithmetic and geometric means, and the guarded Newton method with parameter $\alpha = 0.9$. The convergence of the residual magnitude is shown in Figure 3. As expected, the Newton method converges to very high accuracy in fewer steps.

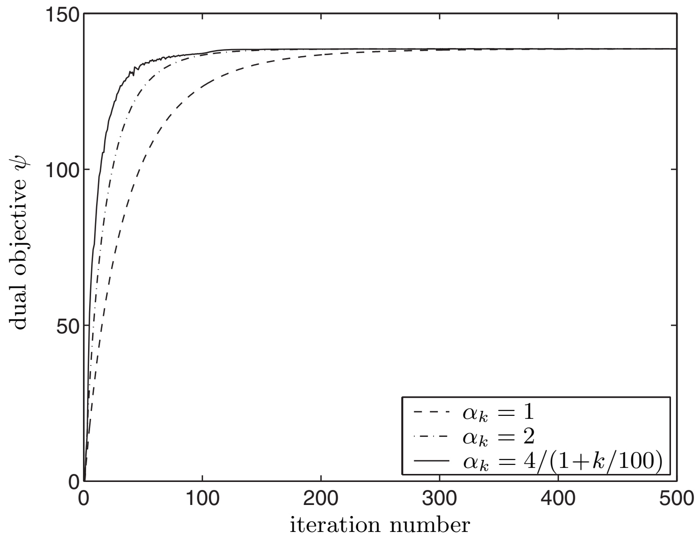

FIG. 2. *Dual objective for the large example.*

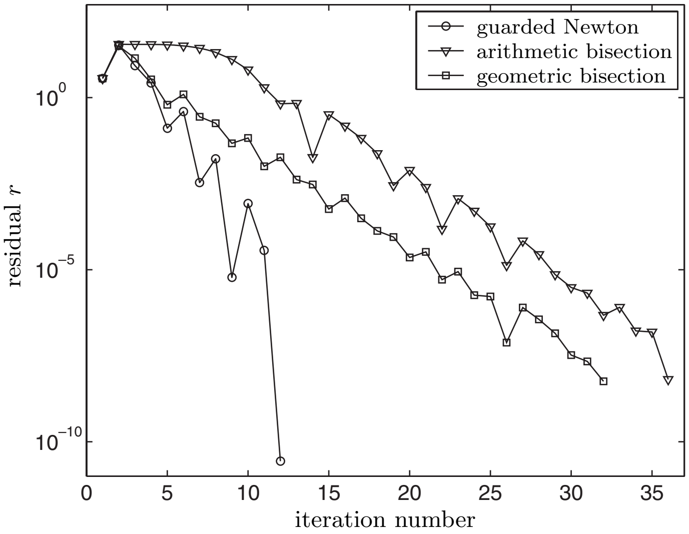

FIG. 3. *Residual magnitude for the small example with one rank-one equality constraint.*

Acknowledgments. We are grateful to Andrew Ng for suggesting the problem to us and to two anonymous reviewers for very useful suggestions and comments.

REFERENCES

[1] D. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[2] J. Borwein and A. Lewis, *Convex Analysis and Nonlinear Optimization*, Canad. Math. Soc. Books Math., Springer-Verlag, New York, 2000.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004. Available online from www.stanford.edu/~boyd/cvxbook.html.

[4] R. Byers, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.

[5] N. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[6] N. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. Gover and S. Barnett, eds., Oxford University Press, Oxford, 1989, pp. 1–27.

[7] N. Higham, *Computing the nearest correlation matrix—A problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.

[8] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer-Verlag, New York, 2001.

[9] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.

[10] E. Levitin and B. Polyak, *Constrained minimization methods*, USSR Comput. Math. Math. Phys., 6 (1966), pp. 1–50.

[11] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 272–284.

[12] B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[13] B. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.

[14] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

[15] N. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer Ser. Comput. Math., Springer-Verlag, New York, 1985.

[16] N. Shor, *Nondifferentiable Optimization and Polynomial Problems*, Kluwer, Norwell, MA, 1998.

[17] M. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.

[18] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.