Open access • Journal Article • DOI:10.1137/S0895479802413856

A Dual Approach to Semidefinite Least-Squares Problems — Source link

Jérôme Malick

Published on: 01 Jan 2005 - SIAM Journal on Matrix Analysis and Applications (Society for Industrial and Applied Mathematics)

Topics: Conic optimization, Semidefinite programming, Convex optimization, Convex set and Affine space

A dual approach to semidefinite least-squares problems

Jérôme Malick

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 272–284

A DUAL APPROACH TO SEMIDEFINITE LEAST-SQUARES PROBLEMS*

JÉRÔME MALICK†

Abstract. In this paper, we study the projection onto the intersection of an affine subspace and a convex set and provide a particular treatment for the cone of positive semidefinite matrices. Among applications of this problem is the calibration of covariance matrices. We propose a Lagrangian dualization of this least-squares problem, which leads us to a convex differentiable dual problem. We propose to solve the latter problem with a quasi-Newton algorithm. We assess this approach with numerical experiments which show that fairly large problems can be solved efficiently.

Key words. Lagrangian duality, semidefinite optimization, calibration of covariance matrices

AMS subject classifications. 65K05, 65F99, 90C22, 91B28

DOI. 10.1137/S0895479802413856

1. Introduction.

1.1. To find the best approximation. We propose a method to solve the following problem: to project a point, in a Euclidean space, onto the intersection of a closed convex set $\mathcal{K}$ and of an affine subspace. We are particularly interested in the case where $\mathcal{K}$ is a cone, more specifically the cone of symmetric positive semidefinite matrices. We call this latter problem *semidefinite least-squares (sdls)*.

Semidefinite least-squares problems arise in different fields of numerical and applied mathematics. For instance, a "good" approximation of a covariance matrix between n assets, which plays a key role in portfolio risk analysis, could be obtained from a first estimate by solving a semidefinite least-squares (this is developed in subsection 5.4). Semidefinite least-squares also occur in robust quadratic optimization and numerical linear algebra (preconditioning of linear system and error analysis of Jacobi methods for the symmetric eigenvalue problem; see [DH00]).

Our aim is to propose an algorithm based on Lagrangian duality to solve the above-mentioned least-squares problem. This paper is organized as follows. We focus, in section 2, on the case where there are no affine constraints: using tools from convex analysis, we recover known properties of distance functions. In section 3, we introduce affine constraints and we show that their dualization yields a dual problem which is convex and differentiable. A quasi-Newton algorithm is proposed in section 4 to solve this last problem. Computational results, comparison with existing methods, and applications of the semidefinite version of this algorithm are presented in section 5.

1.2. Basic notation. The general framework of this paper is a Euclidean space, say $\mathbb{R}^p$, equipped with a scalar product $\langle \cdot, \cdot \rangle$. We will denote by $|| \cdot ||$ the associated norm. We consider, in particular, the space of $n \times n$ symmetric matrices $\mathcal{S}_n$, equipped, for instance, with the Frobenius scalar product

$$\forall X, Y \in \mathcal{S}_n \quad \langle X, Y \rangle \,=\, \mathrm{tr}(XY) \,=\, \sum_{i,j=1}^{n} X_{ij} Y_{ij}\,,$$

*Received by the editors September 3, 2002; accepted for publication (in revised form) by M. L. Overton November 14, 2003; published electronically September 14, 2004.
http://www.siam.org/journals/simax/26-1/41385.html
†INRIA, 655 av. de l'Europe, Montbonnot, 38334 Saint Ismier, France (jerome.malick@inria.fr).

where $\mathrm{tr}(X)$ is the trace of the matrix X. We give below a short glossary of symbols:

- the closed convex cone of positive semidefinite matrices is denoted by $\mathcal{S}_n^+$; we use the notation $X \succeq 0$ to express that X lies in $\mathcal{S}_n^+$;
- the adjoint of a linear mapping $\mathcal{A}$ is denoted by $\mathcal{A}^*$;
- for any vector x in $\mathbb{R}^n$, $\mathrm{Diag}\, x$ denotes the diagonal matrix with the vector x on the main diagonal; its adjoint operator diag: $\mathcal{S}_n \to \mathbb{R}^n$ is $\mathrm{diag}(A) = [a_{11}, \ldots, a_{nn}]^\top$.

1.3. Formulation. The problems we will focus on can be expressed as follows. Let $\mathcal{K}$ be a closed convex set of $\mathbb{R}^p$. Let a vector $b \in \mathbb{R}^m$ and a linear operator $\mathcal{A} : \mathbb{R}^p \longrightarrow \mathbb{R}^m$ be given. We want to compute the projection of a vector $c \in \mathbb{R}^p$ onto the closed convex subset of $\mathbb{R}^p$ formed by the intersection of $\mathcal{K}$ and the affine subspace defined by $\mathcal{A}$ and b. Our goal is to design an algorithm to solve

$$\begin{cases} \inf \quad \frac{1}{2}||x - c||^2, \\ \mathcal{A}\, x = b\,, \\ x \in \mathcal{K}\,. \end{cases} \tag{1.1}$$

Each component function of $\mathcal{A}$ can be expressed as a scalar product: there exist m elements $a_i \in \mathbb{R}^p$ such that $\mathcal{A}(x) = [\langle a_1, x\rangle, \ldots, \langle a_m, x\rangle]^\top$. Therefore an equivalent formulation is (for b, a_i, and c given)

$$\begin{cases} \inf \quad \frac{1}{2}||x - c||^2, \\ \langle a_i, x\rangle = b_i\,, \quad i = 1, \ldots, m, \\ x \in \mathcal{K}\,. \end{cases}$$

The first remark is that, if the feasible domain is nonempty, there exists a unique $x^\star$ which achieves the above infimum. In what follows, we assume this to be the case; therefore we use the notation min rather than inf for this least-squares problem.

To end this introduction, we specify the framework of this paper. Our first motivation is to solve efficiently semidefinite least-squares (i.e., when $\mathcal{K} = \mathcal{S}_n^+$), which section 5 is devoted to. Although the material of this paper can be developed with a general closed convex set $\mathcal{K}$ (see Remarks 2.3 and 4.3(ii)), we restrict ourselves to the case where $\mathcal{K}$ is a *closed convex cone*. This allows us to introduce adapted tools, to simplify calculus and to stay closer to semidefinite least-squares.

2. Projection onto a closed convex cone. To begin with, we isolate the problem of computing the projection $p_{\mathcal{K}}(c)$ of a fixed $c \in \mathbb{R}^p$ onto a closed convex cone $\mathcal{K}$, with a special study for $\mathcal{K} = \mathcal{S}_n^+$. The aim of this section is twofold:

(1) to recall results we will need;

(2) to draw connections between these results and tools from convex analysis.

2.1. Moreau theorem and Moreau regularization. The projection onto a cone $\mathcal{K}$ enjoys properties which come close to those of the projection onto a subspace. The set playing the role of the orthogonal subspace is the *polar cone* $\mathcal{K}^o$ of $\mathcal{K}$:

$$\mathcal{K}^o := \{s \in \mathbb{R}^p : \quad \langle s, x\rangle \leq 0 \quad \text{for all } x \in \mathcal{K}\}\,.$$

A first observation is that $\mathcal{K}^o$ is also closed and convex. There is a decomposition result which generalizes the decomposition of a vector space as the direct sum of a (closed) subspace and its orthogonal (see [HUL01, Chap. A]).

THEOREM 2.1 (Moreau decomposition). *Let $\mathcal{K}$ be a closed convex cone. For the three elements x, x_1, and x_2 in $\mathbb{R}^p$, the two properties below are equivalent:*

(i) $x = x_1 + x_2$ *with* $x_1 \in \mathcal{K}$, $x_2 \in \mathcal{K}^o$ *and* $\langle x_1, x_2 \rangle = 0$,
(ii) $x_1 = p_{\mathcal{K}}(x)$ *and* $x_2 = p_{\mathcal{K}^o}(x)$.

We turn now to variational properties of the half-squared distance to $\mathcal{K}$, which will be needed in section 3:

$$\begin{array}{rccl} d_{\mathcal{K}} & : \mathbb{R}^p & \longrightarrow & \mathbb{R}, \\ & x & \longmapsto & \min\limits_{y \in \mathcal{K}} \ \frac{1}{2}||x-y||^2 . \end{array} \tag{2.1}$$

We start by observing that there is another useful expression of $d_{\mathcal{K}}$. By definition, the above minimum is reached at the unique point $p_{\mathcal{K}}(x)$. Then Theorem 2.1 yields

$$d_{\mathcal{K}}(x) \ = \ \frac{1}{2}||x - p_{\mathcal{K}}(x)||^2 \ = \ \frac{1}{2}||p_{\mathcal{K}^o}(x)||^2 . \tag{2.2}$$

The following properties are not new, and can be proved with basic tools. Here, we show that they are straightforward applications of properties of *Moreau–Yosida regularization* [HUL93, Chap. XV]. For a convex function f on $\mathbb{R}^p$, we define the Moreau–Yosida regularization of f to be the function

$$x \longmapsto \min_{y \in \mathbb{R}^p} \ \left\{ f(y) + \frac{1}{2}||x-y||^2 \right\} .$$

THEOREM 2.2. *Let $\mathcal{K}$ be a closed convex cone in $\mathbb{R}^p$. Then the function $d_{\mathcal{K}}(x)$ defined by* (2.1) *is a convex differentiable function from $\mathbb{R}^p$ to $\mathbb{R}$, whose gradient is*

$$\nabla d_{\mathcal{K}}(x) = p_{\mathcal{K}^o}(x) . \tag{2.3}$$

Furthermore the gradient function is 1*-Lipschitz continuous.*

Proof. Let $\mathrm{I}_{\mathcal{K}}$ be the indicator function of $\mathcal{K}$ (whose values are 0 on $\mathcal{K}$ and $+\infty$ elsewhere). The theorem is just Theorem 4.1.4 of [HUL93, Chap. XV] written in our case, since $d_{\mathcal{K}}$ can be interpreted as the Moreau–Yosida regularization of $\mathrm{I}_{\mathcal{K}}$:

$$d_{\mathcal{K}}(x) = \min_{y \in \mathbb{R}^p} \ \left\{ \mathrm{I}_{\mathcal{K}}(y) + \frac{1}{2}||x-y||^2 \right\} .$$

We get, in particular, $\nabla d_{\mathcal{K}}(x) = x - p_{\mathcal{K}}(x) = p_{\mathcal{K}^o}(x)$ (by Theorem 2.1). The Lipschitz property is clear here since the gradient is a projection. □

Remark 2.3. Note that the above result is valid when k is a general closed convex set, but then the expression (2.3) of the gradient is replaced by $\nabla d_{\mathcal{K}}(x) = x - p_{\mathcal{K}}(x)$ which is again 1-Lipschitz.

2.2. Projection onto $\mathcal{S}_n^+$. In this subsection, we consider the semidefinite least-squares problem without any affine constraint. We recall a crucial theorem for our purposes: an explicit formula for the projection onto $\mathcal{S}_n^+$.

We need more notation. We denote by $\lambda_i(C)$ the (real) eigenvalues of $C \in \mathcal{S}_n$, and rank them in nonincreasing order

$$\lambda_1(C) \geq \lambda_2(C) \geq \cdots \geq \lambda_n(C);$$

$\lambda(C)$ will stand for $[\lambda_1(C), \ldots, \lambda_n(C)]^\top$. The symmetric matrix C is diagonalizable in an orthonormal basis of $\mathbb{R}^n$ formed by eigenvectors of C: $C = P_C(\mathrm{Diag}\lambda(C))P_C^\top$.

We will often drop the dependence on C from our notation. We denote by C_+ the "positive semidefinite part" of C (negative eigenvalues are set to zero):

$$C_+ := P \begin{bmatrix} \max\{\lambda_1, 0\} & & \\ & \ddots & \\ & & \max\{\lambda_n, 0\} \end{bmatrix} P^\top ; \tag{2.4}$$

likewise the "negative semidefinite part" is denoted by C_-.

THEOREM 2.4. *Let $C \in \mathcal{S}_n$. Then the projection $p_{\mathcal{S}_n^+}(C)$ of C onto $\mathcal{S}_n^+$ is the matrix C_+, defined by* (2.4). *Likewise the projection $p_{\mathcal{S}_n^-}(C)$ of C onto the polar cone $(\mathcal{S}_n^+)^o = \mathcal{S}_n^-$ is C_-.*

A direct proof of this result is proposed in [Hig88]. It is worth mentioning that this theorem is also a straightforward application of Theorem 2.1 (see [HUL01, Exercise A.15]), the key being that $C_+ \in \mathcal{S}_n^+$, $C_- \in \mathcal{S}_n^-$ and $\langle C_+, C_- \rangle = 0$.

Remark 2.5. The space $\mathcal{S}_n$ is frequently equipped with a weighted version of the Frobenius norm

$$||X||_W = ||W^{1/2} X W^{1/2}||,$$

where W is a positive definite matrix. It is easy to express the projection (in the sense of the weighted scalar product) of $C \in \mathcal{S}_n$ onto $\mathcal{S}_n^+$ as

$$W^{-1/2}(W^{1/2} C W^{1/2})_+ W^{-1/2}.$$

3. Lagrangian duality. We propose in this section a Lagrangian dualization of (1.1). The idea is to treat in two different ways the two different kinds of constraints: on one hand affine constraints in $\mathbb{R}^p$ and on the other hand convex constraints. The technique is to dualize only affine constraints, forming a partial Lagrangian.

All the present paper relies upon the next statement. It motivates the developments of previous sections and will give birth to computational methods.

THEOREM 3.1. *Consider the following least-squares problem in $(\mathbb{R}^p, ||\cdot||)$:*

$$(primal) \begin{cases} \min & \frac{1}{2}||x - c||^2, \\ x \in \mathcal{K}, & \mathcal{A}\, x = b\,, \end{cases}$$

which is our primal problem. Form the partial Lagrangian depending on two variables (the primal variable x which lies in $\mathcal{K} \subset \mathbb{R}^p$ and the dual variable y which lies in the constraint space $\mathbb{R}^m$)

$$L(x; y) := \frac{1}{2}||c - x||^2 - y^\top (\mathcal{A}x - b)\,. \tag{3.1}$$

Define the corresponding dual function

$$\theta(y) := \min_{x \in \mathcal{K}} L(x; y) \tag{3.2}$$

and the dual problem on the constraint space $\mathbb{R}^m$

$$(dual) \begin{cases} \sup & \theta(y), \\ y \in \mathbb{R}^m\,. \end{cases}$$

The dual function has the following expressions:

$$\begin{aligned} \theta(y) &= -\tfrac{1}{2}||p_{\mathcal{K}}(c + \mathcal{A}^* y)||^2 + \tfrac{1}{2}||c||^2 + y^\top b \\ &= -d_{\mathcal{K}^o}(c + \mathcal{A}^* y) + \tfrac{1}{2}||c||^2 + y^\top b\,. \end{aligned} \tag{3.3}$$

Proof. Let us transform the partial Lagrangian to isolate the function $d_{\mathcal{K}}$ of (2.1):

$$\begin{aligned} L(x;y) &= \tfrac{1}{2}||c-x||^2 - \langle \mathcal{A}^* y, x\rangle + y^\top b \\ &= \tfrac{1}{2}||(c+\mathcal{A}^* y) - x||^2 - (\tfrac{1}{2}||\mathcal{A}^* y||^2 + \langle c, \mathcal{A}^* y\rangle) + y^\top b \\ &= \tfrac{1}{2}||(c+\mathcal{A}^* y) - x||^2 - (\tfrac{1}{2}||\mathcal{A}^* y + c||^2 - \tfrac{1}{2}||c||^2) + y^\top b\,. \end{aligned}$$

Now get an expression of θ. For any fixed $y \in \mathbb{R}^m$:

$$\begin{aligned} \theta(y) &:= \min_{x\in\mathcal{K}} L(x;y) \\ &= d_{\mathcal{K}}(c+\mathcal{A}^* y) - \tfrac{1}{2}||\mathcal{A}^* y + c||^2 + \tfrac{1}{2}||c||^2 + y^\top b\,. \end{aligned}$$

Simplify with (2.2):

$$\begin{aligned} \theta(y) &= \tfrac{1}{2}||p_{\mathcal{K}^o}(c+\mathcal{A}^* y)||^2 - \tfrac{1}{2}||\mathcal{A}^* y + c||^2 + \tfrac{1}{2}||c||^2 + y^\top b \\ &= -\tfrac{1}{2}||p_{\mathcal{K}}(c+\mathcal{A}^* y)||^2 + \tfrac{1}{2}||c||^2 + y^\top b \\ &= -d_{\mathcal{K}^o}(c+\mathcal{A}^* y) + \tfrac{1}{2}||c||^2 + y^\top b\,. \end{aligned}$$

We therefore obtain the expected formulations of θ. □

Notice that we know the unique point in $\mathcal{K}$ which achieves the minimum in (3.2) for $y \in \mathbb{R}^p$. In the remainder of the paper, we denote it by $x(y)$:

$$x(y) := \operatorname*{argmin}_{x\in\mathcal{K}} L(x;y) = p_{\mathcal{K}}(c+\mathcal{A}^* y)\,. \tag{3.4}$$

In other words there holds

$$\theta(y) = L(x(y);y)\,. \tag{3.5}$$

The dual function θ inherits the properties of $d_{\mathcal{K}^o}$ studied in section 2.

THEOREM 3.2. *The function θ of* (3.3) *satisfies the properties below:*

(i) *θ is concave,*

(ii) *θ is differentiable,*

(iii) *$\nabla\theta$ is Lipschitz continuous and is given by*

$$\nabla\theta(y) = -\mathcal{A}\{p_{\mathcal{K}}(c+\mathcal{A}^* y)\} + b \tag{3.6}$$

Proof. The dual function, as a minimum of affine functions of y, is concave by construction. Besides, with equation (3.3), according to results on $d_{\mathcal{K}^o}$ (Theorem 2.2 for $\mathcal{K}^o$), θ is differentiable, its gradient is

$$\begin{aligned} \nabla\theta(y) &= -\mathcal{A}\{\nabla d_{\mathcal{K}^o}(c+\mathcal{A}^* y)\} + b \\ &= -\mathcal{A}\{p_{\mathcal{K}}(c+\mathcal{A}^* y)\} + b\,, \end{aligned}$$

which is the required result. □

The dual function has a strong structure which will be used for algorithmic perspectives. The dual problem reduces to the *convex-differentiable* optimization problem

$$\begin{cases} \inf\ \tfrac{1}{2}||p_{\mathcal{K}}(c+\mathcal{A}^* y)||^2 - y^\top b, \\ y \in \mathbb{R}^m\,. \end{cases}$$

Example 1 (semidefinite least-squares). In the case $\mathcal{K} = \mathcal{S}_n^+$, the key point is that we have an easy-to-compute formulation of the projection (Theorem 2.4). The dual problem is here

$$\begin{cases} \min\ ||(C+\mathcal{A}^* y)_+||^2 - b^\top y \\ y \in \mathbb{R}^m\,. \end{cases}$$

4. A dual algorithm. We want to solve the primal problem, i.e., to find $x^\star \in \mathcal{K}$ closest to $c \in \mathbb{R}^p$ while satisfying affine constraints. The structure of the primal is not easy to use directly. On the other hand, its dual problem is more strongly structured (Theorem 3.2) and thus opens the way to a possible resolution procedure.

We assume in this section that there is a solution $y^\star$ to the dual problem: the dual function is bounded from above and its supremum is actually a maximum, achieved at $y^\star$. We are therefore in the following primal-dual situation

$$(4.1) \qquad (primal)\ \begin{cases} \min \quad \frac{1}{2}||x-c||^2 \\ x \in \mathcal{K},\ \ \mathcal{A}\,x = b \end{cases} \qquad (dual)\ \begin{cases} \max \quad \theta(y) \\ y \in \mathbb{R}^m\,. \end{cases}$$

with θ expressed by (3.3).

4.1. From dual to primal solution. In this subsection, we suppose that we are able to get efficiently $y^\star$. We show that in this case the primal problem is indeed solved. We start by mentioning that each value of the dual function gives a lower bound on the primal objective function: from the weak duality theorem (see [HUL93, Chap. XII]), there holds

$$(4.2) \qquad \theta(y) \leq \frac{1}{2}||c-x||^2$$

for all dual-feasible points (i.e., $y \in \mathbb{R}^m$) and for all primal-feasible points (i.e., $x \in \mathcal{K}$ such that $\mathcal{A}\,x = b$).

THEOREM 4.1. *Assume the existence of a dual solution $y^\star$. Then the solution $x^\star$ of the primal problem is given by*

$$(4.3) \qquad x^\star = p_{\mathcal{K}}(c + \mathcal{A}^* y^\star)\,.$$

Proof. From Theorem 3.2, θ is concave and differentiable, then at $y^\star$ which achieves its maximum, its gradient is zero. By equations (3.4) and (3.6), this results in $\mathcal{A}\,x(y^\star) = b$, i.e., $x(y^\star)$ is primal-feasible. Then we have by (3.5)

$$(4.4) \qquad \theta(y^\star) = L(x(y^\star); y^\star) = \frac{1}{2}||c - x(y^\star)||^2.$$

By (4.2), $\theta(y^\star)$ is a lower bound of the objective function of the primal. Equation (4.4) means that this lower bound is reached at the primal-feasible $x(y^\star)$. Thus that point is the minimum and

$$x^\star = x(y^\star) = p_{\mathcal{K}}(c + \mathcal{A}^* y^\star)\,,$$

which ends the proof. ∎

This theorem says that there is *no duality gap* between the primal and the dual. This is expressed by equation (4.4): the values of the primal function at its minimum and of the dual at its maximum are the same.

A particular case yielding both existence of $y^\star$ and absence of a duality gap is the primal *Slater condition* (see [HUL93, Chap. XII]), expressing that feasibility of the primal constraints is preserved despite perturbations of b. It corresponds to the existence of a point *strictly feasible* of the primal: there exists x satisfying $\mathcal{A}\,x = b$ and lying in the interior of $\mathcal{K}$, assumed nonempty.

4.2. Computing a dual solution. The regularity properties of θ allow the use of any classical algorithm to minimize it; for instance, a quasi-Newton algorithm is considered as most efficient.

ALGORITHM 1. *Consider the pair of primal-dual problems* (4.1). *Let a black-box perform the following task:*

(i) *compute* $\mathcal{A}^* y$ *for given* $y \in \mathbb{R}^m$;
(ii) *compute* $\mathcal{A}\, x$ *for given* $x \in \mathbb{R}^p$;
(iii) *compute* $p_{\mathcal{K}}(z)$ *for given* $z \in \mathbb{R}^p$.

Use a quasi-Newton optimization code to maximize θ *on* $\mathbb{R}^m$. *With the help of the above black-box, this code generates a maximizing sequence* $(y_k)_k$ *together with the corresponding:*

(i) $x_k = p_{\mathcal{K}}(c + \mathcal{A}^* y_k)$;
(ii) $\nabla\theta(y_k) = -\mathcal{A}\, x_k + b$;
(iii) $\theta(y_k) = -\frac{1}{2}||x_k||^2 + y^\top b$.

To implement the above algorithm the only thing we basically need is to compute $p_{\mathcal{K}}$. In other words, the key point to solve our problem (i.e., to compute the projection onto the intersection of $\mathcal{K}$ with an affine hyperplane) is to know how to solve the problem without affine constraints (i.e., to compute the projection onto $\mathcal{K}$). This means the algorithm is efficient when the difficulty is due to the addition of affine constraints. For instance, this is the case for semidefinite least-squares, where we have the easy-to-compute expression (2.4) of the projection.

An instance of quasi-Newton known to be convergent when the objective function is convex and has a Lipschitz gradient is the so-called BFGS with Wolfe line-search (Theorem 4.9 of [BGLS03]). Here is a convergence result.

THEOREM 4.2. *Let* $\mathcal{A}$ *be surjective and the Slater assumption hold. Then Algorithm* 1 *gives an approximation of* $x^\star$: *for any* $\epsilon > 0$, *there is* k *such that* $||x_k - x^\star|| \leq \epsilon$.

Proof. From the Slater assumption and the surjectivity of $\mathcal{A}$, the dual optimal set is bounded [HUL93, Chap. VII], and then each level-set is bounded [HUL93, Chap. IV]. The sequence (y_k) is thus bounded. Take $\epsilon > 0$. Since θ is continuous on $\mathbb{R}^m$, there exists a dual solution $y^\star$ and k large enough such that $||y_k - y^\star|| \leq \epsilon / ||A^*||$. Now from Lipschitzian property of the projection we can write

$$||x^\star - x_k|| = ||p_{\mathcal{K}}(c + \mathcal{A}^* y^\star) - p_{\mathcal{K}}(c + \mathcal{A}^* y_k)|| \leq ||\mathcal{A}^*||\ ||y^\star - y_k|| \leq \epsilon\,.$$

This ends the proof. ∎

We mention that the so-called *limited memory* quasi-Newton method can also be used. It avoids the need to store an $m \times m$ matrix, thus accommodating very large values (see [BGLS03, sects. 1.2.2 and 6.3]).

Remark 4.3. To conclude this section, we mention two possible extensions.

(i) Observe that the dual of

$$\begin{cases} \min & \frac{1}{2}||x - c||^2, \\ x \in \mathcal{K}, & \mathcal{A}\, x \leq b, \end{cases}$$

is (by an easy adaptation of the proof of Theorem 3.1)

$$\begin{cases} \max & \theta(y), \\ y \in \mathbb{R}^m, & y_i \geq 0 \quad \text{for all } i = 1, \ldots, m. \end{cases}$$

Thus Algorithm 1 can solve such problems with inequality constraints, whenever the quasi-Newton algorithm accepts box-constraints.

(ii) We also add that this approach can even be used for problems when $\mathcal{K}$ is a general closed convex set. The method can be adapted to treat the projection on the intersection of a convex set $\mathcal{K}$ and an affine subspace, if one knows how to project onto $\mathcal{K}$. In fact, θ is always a concave differentiable function, whose gradient is

$$\nabla\theta(y) = -\mathcal{A}x(y) + b$$

with $x(y) = \operatorname{argmin}_{x\in\mathcal{K}} L(x; y)$. All these results come from [HUL01, Chap. D.4.4].

5. Semidefinite least-squares. In this section, we focus on the case where $\mathcal{K} = \mathcal{S}_n^+$. Semidefinite least-squares is a very important subclass of the general least-squares problem (1.1). For instance, the computation of a "good" approximation of the covariance matrix of n assets can be expressed as a semidefinite least-squares problem (see subsection 5.4). Recall that an optimization program is written under a semidefinite least-squares form if there are a matrix $C \in \mathcal{S}_n$, a vector $b \in \mathbb{R}^m$ and a linear operator $\mathcal{A} : \mathcal{S}_n \longrightarrow \mathbb{R}^m$ such that

$$\text{(sdls)} \quad \begin{cases} \min & \frac{1}{2}||X - C||^2, \\ \mathcal{A}\,X = b, \\ X \succeq 0\,. \end{cases}$$

If $\mathcal{A}$ is expressed via its m component functions, (sdls) can be formulated with the help of m symmetric matrices:

$$\begin{cases} \min & \frac{1}{2}||X - C||^2, \\ \langle A_i, X\rangle = b_i\,, & i = 1, \ldots, m, \\ X \succeq 0\,. \end{cases}$$

We present, in subsections 5.1 and 5.2, known methods to solve (sdls): interior points and alternative projections. We also give a dual interpretation of the latter. We then show that Algorithm 1 is a good alternative to these methods.

5.1. Semidefinite approach. A natural idea to attack (sdls) directly is to phrase it as a semidefinite program. The problem can actually be seen as a quadratic-semidefinite program and then efficient interior-points methods for SDP programming are available (see [Tod01] for a review).

The (nonlinear) objective function can actually be pushed into constraints:

$$\text{(5.1)} \quad \begin{cases} \min & t, \\ ||X - C|| \leq t, \\ \mathcal{A}\,X = b, \\ X \succeq 0, \end{cases}$$

a problem expressed as a quadratic-semidefinite program [BTN01]. Thus powerful interior-points methods solvers can be used. However the number of variables is $\mathcal{O}(n^2)$ and this approach is presented as impractical for large n in [Hig02, subsect. 3.3]. Tests that we ran with SEDUMI [Stu99] confirm this point.

However, it should be mentioned that putting (5.1) in SEDUMI format requires the introduction of artificial variables and constraints. Adapted interior-points variants may exist. Note that [Tak03] provides one for sparse matrices.

5.2. Alternating projections method. Another interesting method is proposed in [Hig02] to solve particular instances of semidefinite least-squares (and it could be easily generalized to any semidefinite least-squares). Here $\mathcal{A}$ is the diag-operator: we want to solve

$$\begin{cases} \min \quad \frac{1}{2}||X-C||^2\,, \\ X_{ii}=1\,, \quad i=1,\dots,n\,, \\ X \succeq 0\,. \end{cases} \tag{5.2}$$

Introducing the notation

$$\mathcal{U}=\{X\in\mathcal{S}_n : X_{ii}=1\}, \tag{5.3}$$

the idea of the so-called alternating projection method is to repeat the operation

$$X \leftarrow p_{\mathcal{S}_n^+}(p_{\mathcal{U}}(X)). \tag{5.4}$$

Besides, the so-called Dykstra's correction [Dyk83] is used in [Hig02]. All together the algorithm is as follows.

ALGORITHM 2 (Algorithm 3.3 of [Hig02]). *For $C\in\mathcal{S}_n$, this algorithm solves* (5.2):

$\Delta S_0=0, Y_0=C$
for $k=1,2,\dots$
  $R_k=Y_{k-1}-\Delta S_{k-1}$ *% Dykstra's correction*
  $X_k=p_{\mathcal{S}_n^+}(R_k)$
  $\Delta S_k=X_k-R_k$
  $Y_k=p_{\mathcal{U}}(X_k)$
end

The alternating projection method has actually a dual interpretation in this case. Theorem 5.1 below says that it is just the standard gradient optimization algorithm applied to the dual of (5.2), namely $y_{k+1}=y_k+\nabla\theta(y_k)$. In fact, recalling formula (3.6), the gradient algorithm can be expressed as follows.

ALGORITHM 3. *This algorithm maximizes $\theta(y)=-||p_{\mathcal{S}_n^+}(C+\mathcal{A}^*y)||^2+b^\top y$ on $\mathbb{R}^n$, by the gradient method (with constant stepsize equal to 1):*

$y_1=0$,
for $k=1,2,\dots$
  $\bar{X}_k=p_{\mathcal{S}_n^+}(C+\mathcal{A}^*y_k)$
  $y_{k+1}=y_k+(-\mathcal{A}\bar{X}_k+b)$
end

In view of (5.2), we have here $b=[1,\dots,1]^\top$, $\mathcal{A}=\mathrm{diag}$, $\mathcal{A}^*=\mathrm{Diag}$, and we observe that $\mathcal{A}\mathcal{A}^*=I_m$.

THEOREM 5.1. *The sequence $(\bar{X}_k)$ generated by Algorithm 3 is the same as (X_k) generated by Algorithm 2.*

Proof. First, observe that the projection of $X\in\mathcal{S}_n$ on $\mathcal{U}$ is

$$p_{\mathcal{U}}(X)=X-\mathcal{A}^*(\mathcal{A}X-b). \tag{5.5}$$

Let us prove by recurrence that

$$\bar{X}_k=X_k \ \text{ and } \ R_k=C+\mathcal{A}^*y_k \quad \text{for all } k\geq 0\,. \tag{5.6}$$

This is true for $k = 1$ since $R_1 = C$, $y_1 = 0$, and $\bar{X}_1 = X_1 = p_{\mathcal{S}_n^+}(C)$. Suppose now that it holds for k; then

$$\begin{aligned} R_{k+1} &= Y_k - \Delta S_k && \text{[definition of } R_{k+1}\text{]} \\ &= Y_k - X_k + R_k && \text{[definition of } \Delta S_k\text{]} \\ &= -\mathcal{A}^*(\mathcal{A}X_k - b) + R_k && \text{[(5.5) and definition of } Y_k\text{]} \\ &= -\mathcal{A}^*(\mathcal{A}\bar{X}_k - b) + C + \mathcal{A}^* y_k && \text{[recurrence assumptions]} \\ &= C + \mathcal{A}^*(y_k - (\mathcal{A}\bar{X}_k - b)) \\ &= C + \mathcal{A}^* y_{k+1}\,. && \text{[definition of } y_{k+1}\text{]} \end{aligned}$$

Hence $X_{k+1} = p_{\mathcal{S}_n^+}(R_{k+1}) = p_{\mathcal{S}_n^+}(c + \mathcal{A}^* y_{k+1}) = \bar{X}_{k+1}$, and the theorem is proved. □

As a result, the method of alternative projections (Algorithm 2) and our proposal (Algorithm 1) are well comparable: both are optimization algorithms to maximize the dual function, the former does this by the (simple) gradient method with constant step size while the latter uses the (sophisticated) quasi-Newton approach.

5.3. Numerical results. For illustration, Algorithm 1 has been applied to instances of (5.2). We ran three types of experiments:

(i) We solve (5.2) with random dense matrices C (random $C_{ij} \in [-1, 1]$ and $C_{ii} = 1$) of sizes from 100×100 to 3000×3000.

(ii) We take a matrix $X^\star$ in $\mathcal{U} \cap \mathcal{S}_n^+$ (where $\mathcal{U}$ is defined by (5.3)) of size 1000×1000 and we perturb it to create a matrix C such that $X^\star$ is the projection of C. We then test Algorithm 1 with this C.

(iii) We fix the size (500×500) and we take matrices with increasing entries on the diagonal.

The algorithm has been coded in Fortran and we use the LAPACK library for numerical algebra. Note that the computation of the projection onto $\mathcal{S}_n^+$ is nothing more than an eigensystem computation: we use symmetric QR algorithm of LAPACK. In our experiments, the stopping test is

$$\frac{1}{\sqrt{n}}||\nabla\theta(y_k)|| = \frac{1}{\sqrt{n}}||\mathcal{A}X_k - b|| \leq 10^{-7}\,.$$

The performance measures have been obtained on a machine of the Intel P4 2 GHz processor family with 512 Mbytes of memory. The system runs under Linux Redhat 8.0 and uses the gnu compilation chain.

First experiment. The results with random matrices are as follows.

matrix sizes	cpu time	nb of iterations
100×100	0.2 s	14
300×300	3.3 s	14
500×500	16.3 s	17
800×800	1 min 10 s	17
1000×1000	2 min 05 s	18
1500×1500	7 min 35 s	18
2000×2000	17 min 41 s	19
3000×3000	1h 08 min 14 s	19

Some observations are worth mentioning:

- Computation on matrices up to 200×200 takes less than one second and the algorithm copes very well with larger matrices (one hour for a 3000×3000 dense matrix).

- For these kinds of matrices (with $C_{ij} \in [-1, 1]$), the typical number of iterations ranges between 10 and 20, almost independently of the problem size. The experimental computational cost is $\mathcal{O}(n^3)$.
- The bulk of the work is the spectral decomposition. Better cpu time could be obtained along the lines of Corollary 3.5 of [Hig02], by avoiding the computation of the full eigensystem. We have not implemented this idea.

Second experiment. Now we give an idea of the behavior of the algorithm by tracing the error between the exact solution and the current iterate. To construct a synthetic example, one can proceed as follows. Take a matrix $X^\star$ in $\mathcal{U} \cap \mathcal{S}_n^+$. Then set $C := X^\star + \delta X$, where δX lies in the normal cone to $\mathcal{U} \cap \mathcal{S}_n^+$ at $X^\star$, so that $X^\star$ solves (5.2).

By calculus rules of [HUL01, Chap. A], the normal cone to $\mathcal{U} \cap \mathcal{S}_n^+$ at $X^\star$ is the sum of $\mathcal{U}^\perp$ (normal "cone" to the subspace $\mathcal{U}$) and of $\mathcal{S}_n^- \cap X^\perp$ (normal cone to $\mathcal{S}_n^+$ at $X^\star$). It suffices to choose an $X^\star$ such that constructing a matrix in $\mathcal{S}_n^- \cap X^\perp$ is easy. For example, take $1 \leq \ell \leq n$ and consider the matrix

$$X^\star := \begin{bmatrix} E_\ell & \\ & I_{n-\ell} \end{bmatrix},$$

where $I_{n-\ell}$ is the $(n-\ell)\times(n-\ell)$ identity matrix and E_ℓ is the $\ell \times \ell$ matrix with all entries equal to 1. Then it is easy to see that a suitable matrix is

$$C := \begin{bmatrix} \frac{\ell}{\ell-1} E_\ell & \\ & I_{n-\ell} \end{bmatrix} + D,$$

where D is an arbitrary diagonal matrix.

Figure 5.1 shows the evolution of the distance of the current iterate X_k to the solution (which is $X^\star$ by construction) for an instance where $n = 1000$, $\ell = 500$, and D_{ii} is a random number in $[-10, 10]$. The algorithm converges in 56 iterations (and 58 diagonalizations). Note that the run takes more iterations than the 1000×1000-instance of the first experiment (18 iterations). This fact is underlined by the third experiment.

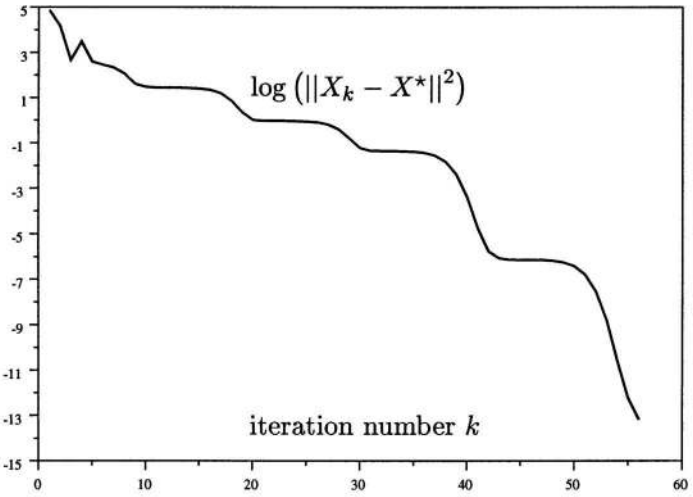

FIG. 5.1. *Evolution of* $||X_k - X^\star||^2$.

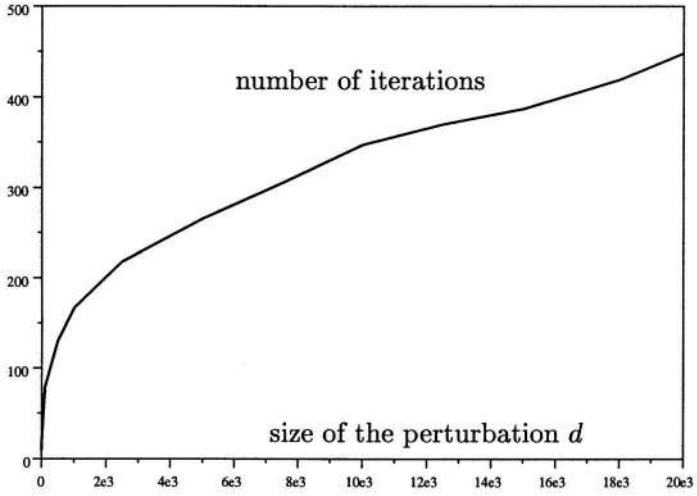

FIG. 5.2. *Influence of the remoteness of C.*

Third experiment. When the entries of C get larger, C gets more remote from the target $\mathcal{U} \cap \mathcal{S}_n^+$. One can think that computing the projection of C then takes more iterations. The aim of this experiment is to quantify the phenomenon, which turns out to be fairly significant. In Figure 5.2, C is constructed as in the second experiment with $n = 500$, $\ell = 250$, and random $D_{ii} \in [-d, d]$ (we increase d from 0 to 20000).

5.4. Calibration of covariance matrices. In this subsection we outline the problem of finding a "good" approximation $\bar{Q}$ of the theoretical covariance matrix Q between n assets: it turns out to be a semidefinite least-squares problem. Another problem of that kind is computing the nearest correlation matrix [Hig02]. To have a good approximation $\bar{Q}$ is important in portfolio management: this matrix is used to have a robust estimation of the "ex-ante" risk of any possible portfolio among these n assets.

For instance, portfolio managers often look for portfolios minimizing the financial risk while having a fixed return. They want to solve a portfolio selection problem of the following type:

$$\begin{cases} \min \quad x^\top Q x\,, \\ x^\top r \geq \beta\,, \\ x_i \in [0,1], \quad \sum_{i=1}^n x_i = 1. \end{cases} \tag{5.7}$$

This is the famous portfolio selection problem of Markowitz (Nobel prize winner in 1990). The covariance matrix Q (which is positive semidefinite) is used to estimate the risk. Under the classical economic assumption that there is no rewarding risk-less investments (no-arbitrage assumption), Q is definite positive. Let $\tilde{Q}$ be a first estimate of the true covariance matrix Q: for instance $\tilde{Q}$ can be the empirical estimate after k days. The point is that $\tilde{Q}$ has a bad condition number:

- When the number of observations k is too small, $\tilde{Q}$ is rank-deficient (some investments are considered with no risk; it is not consistent with the no-arbitrage assumption).
- When there are different levels of risks in the portfolio, $\tilde{Q}$ is ill conditioned (the condition number of $\tilde{Q}$ is typically greater than 10^7 if there are stocks,

options, and monetary products in the portfolio; it reaches 10^{17} for hedge funds).

We may impose $X \succeq \alpha I_n$ for some selected $\alpha > 0$ to avoid too low-risk portfolios: we thus guarantee "cautious" risk evaluations, and stability of the portfolio selected by (5.7). Eventually we are led to the so-called *calibration of covariance matrix* problem, which is a shifted (sdls):

$$\begin{cases} \min \quad \frac{1}{2}||X - \tilde{Q}||^2, \\ X \succeq \alpha I_n\,, \\ \langle I_n, X\rangle = \mathrm{tr}(\tilde{Q})\,, \\ \langle A_i, X\rangle = \sigma_i^2\,, \end{cases}$$

where σ_i^2 represent "ex-post" volatilities of well-chosen portfolios. The constraint $\langle I_n, X\rangle = \mathrm{tr}(\tilde{Q})$ enforces the conservation of the empirical total risk. We solve real-life instances of this problem (provided to us by RAISE PARTNER) with Algorithm 1; the results are quite similar to those of subsection 5.3.

We end with a remark. The material developed in this section can be easily extended to the Frobenius norm with weights (see Remark 2.5). This is of little impact in theory but more in practice: for instance, the covariance between some assets are sometimes more relevant than others, so we want to ensure in the calibration process of the covariance matrix that the relevance is properly emphasized.

Acknowledgments. My primary thanks go to Claude Lemaréchal for his advice and his everyday help. I am indebted to François Oustry and RAISE PARTNER (especially Nabil Layaïda). I also want to thank the editor and the two referees for their numerous suggestions.

REFERENCES

[BGLS03] J. F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical Optimization*, Springer-Verlag, Berlin, 2003.

[BTN01] R. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, MPS-SIAM Ser. Optim., SIAM, Philadelphia, 2001.

[DH00] P. Davies and N. Higham, *Numerically stable generation of correlation matrices and theirs factors*, BIT, 40 (2000), pp. 640–651.

[Dyk83] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842.

[Hig88] N. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[Hig02] N. Higham, *Computing the nearest symmetric correlation matrix—a problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[HUL01] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer-Verlag, Berlin, 2001.

[Stu99] J. F. Sturm, *Using Sedumi* 1.02, *a Matlab toolbox for optimization over symmetric cones*, Optm. Meth. Soft., 11/12 (1999), pp. 625–653.

[Tak03] P. L. Takouda, *Problèmes d'approximation matricielle linéaires coniques: Approches par projections et via Optimisation sous contraintes de semidéfinie positivité*, Ph.D. thesis, Université Paul Sabatier–Toulouse III, 2003.

[Tod01] M. J. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.