

On formulating quadratic functions in optimization models.

Author: Erling D. Andersen

Convex quadratic constraints quite frequently appear in optimization problems and hence it is important to know how to formulate them efficiently. In this note it is argued that reformulating them to separable form is advantageous because it makes the convexity check trivial and usually leads to a faster solution time requiring less storage. The suggested technique is particularly applicable in portfolio optimization where a factor model for the covariance matrix is employed.

Moreover, we discuss how to formulate quadratic constraints using quadratic cones and the benefits of doing that.

1 Introduction

Consider the quadratic optimization problem

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \frac{1}{2} x^T Q x + a^T x + b \leq 0 \end{aligned} \quad (1)$$

where Q is symmetric

$$Q = Q^T.$$

Normally there are some structure in the Q matrix e.g. Q may be a low rank matrix or have a factor structure. The problem (1) is said to be separable if Q is a diagonal matrix. Subsequently it is demonstrated how the problem (1) always can be reformulated to have a separable structure.

For simplicity it is assumed that the problem (1) only has one constraint. However, it should be obvious how to extend the suggested techniques to a problem with an arbitrary number of constraints. Moreover, it is important to note that the techniques outlined are applicable to problems with a quadratic objective as well. Indeed the problem

$$\text{minimize} \quad c^T x + \frac{1}{2} x^T Q x \quad (2)$$

is equivalent to

$$\begin{aligned} & \text{minimize} && c^T x + t \\ & \text{subject to} && \frac{1}{2} x^T Q x \leq t \end{aligned} \quad (3)$$

which has the form (1).

2 The convexity assumption

The problem (1) is an easy problem to solve if it is a convex problem. It is well known that the problem (1) is convex if and only if the quadratic function

$$f(x) = x^T Q x \quad (4)$$

is convex. Furthermore, the following statements are equivalent.



Function	Storage	Operational
	cost	cost
f	$\frac{1}{2}n^2$	$\frac{1}{2}n^2$
g	np	np

Table 1: Storage and evaluation costs.

- i) f is convex.
- ii) Q positive semidefinite.
- iii) There exist a matrix H such that $Q = HH^T$.

Observe using $Q = HH^T$ we have

$$\begin{aligned} x^T Q x &= x^T H H^T x \\ &= \|H^T x\|^2 \\ &\geq 0. \end{aligned}$$

Note that H is not unique in general, for instance H may be the Cholesky factor or $Q^{\frac{1}{2}}$. Moreover, in practice usually the matrix H is known and not Q and this precisely the reason why it can be concluded Q positive semidefinite.

Since optimization software typically is not informed about H but is only given Q then the software checks convexity assumption by computing a Cholesky factorization of Q . Unfortunately this is not a robust convexity check as the following example demonstrates: assume that

$$Q = \begin{bmatrix} 1 & \sqrt{\alpha} \\ \sqrt{\alpha} & \alpha \end{bmatrix}.$$

This matrix is by construction positive semidefinite. Next assume computations are done in finite precision using 3 digits of accuracy and $\alpha = 5$ so the problem is to check whether

$$\begin{bmatrix} 1 & 2.24 \\ 2.24 & 5 \end{bmatrix}$$

is positive semidefinite. However, it is not positive semidefinite! Hence, if rounding errors are present as they are on a computer then the rounded Q matrix may not positive semidefinite. In practice computers employs about 16 figures of accuracy so wrong conclusions about the convexity does not appear often but cannot be ruled out.

The conclusion is that a convexity check is not fool proof in practice. Nevertheless for the special case where the matrix Q is a diagonal matrix then check is simple and fool proof, since the check consist of checking whether all the diagonal elements are nonnegative.

3 Separable form is the winner (usually)

In the previous section it was demonstrated that if Q is a diagonal matrix then it is easy to check the convexity assumption. The purpose of this section is to demonstrate that the problem (1) always can be made separable given the convexity assumption. In addition we demonstrate that the reformulation likely leads to a faster solution time.

First define

$$g(x) = \|H^T x\|^2 \tag{5}$$

and clearly

$$f(x) = g(x) = x^T Q x$$

holds. Therefore, we can use (5) instead of (4) if deemed worthwhile.

Assuming that $H \in \mathbb{R}^{n \times p}$ and Q is a dense matrix, then Table 3 list how much storage that is required to store f and g on a computer respectively. Moreover, the table list how many operations that is required to evaluate the two functions respectively. By operations we mean the number of basic arithmetic operations like $+$ that is needed to evaluate the function.



Table 3 shows that if p is much smaller than n , then using the formulation (5) saves a huge amount of storage and work. This observation can be used to reformulate (1) to

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \frac{1}{2}y^T y + a^T x + b \leq 0, \\ & && H^T x - y = 0. \end{aligned} \quad (6)$$

The formulation (6) is larger than (1) because p additional linear equalities and variables have been added. However, if p is much smaller than n , then the formulation (6) requires much less storage which in most cases leads to a much faster solution times.

Before continuing then let us take a look at an important special case. Assume a vector v is known such that

$$Hv = a$$

For instance if H is nonsingular, then v exists and can be computed as $H^{-1}a$. In many practical applications a v will be trivially known.

Now if we let

$$H^T x - y = -v$$

then

$$\begin{aligned} \|y\|^2 &= \|H^T x + v\|^2 \\ &= x^T H H^T x + v^T v + 2x^T H v \\ &= x^T Q x + 2a^T x + v^T v \end{aligned}$$

Therefore, problem (6) and

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && 0.5y^T y - 0.5v^T v + b \leq 0, \\ & && H^T x - y = v. \end{aligned} \quad (7)$$

is equivalent. The reformulation (7) is sparser than (6) in the constraint matrix because the problem no longer contains a and may therefore be preferable.

Now let us consider a slight generalization i.e. let us assume that

$$Q = D + H H^T \quad (8)$$

where D is positive semidefinite matrix. This implies Q is positive semidefinite. Furthermore, D is assumed to be simple e.g. a diagonal matrix. Using the structure in (8) then (1) can be cast as

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \frac{1}{2}(x^T D x + y^T y) + a^T x + b \leq 0, \\ & && H^T x - y = 0. \end{aligned} \quad (9)$$

In portfolio optimization arising in finance n may be 1000 and p is less than, say 50. In that case (9) will require about 10 times less storage compared to (1) assuming Q is dense. This will translate into dramatic faster solution times.

4 Conic reformulation

It is always possible and often worthwhile to reformulate convex quadratic and quadratically constrained optimization problems on conic form. In this section we will discuss that possibility.

We will use the definitions

$$\mathcal{K}^q := \{x \in \mathbb{R}^n \mid x_1 \geq \|x_{2:n}\|\}$$

and

$$\mathcal{K}^r := \{x \in \mathbb{R}^n \mid 2x_1 x_2 \geq \|x_{3:n}\|^2, \quad x_1, x_2 \geq 0\}.$$

Hence, \mathcal{K}^q is the quadratic cone and \mathcal{K}^r is the rotated quadratic cone.

Let us first consider the conic reformulation of (6) which is

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && t + a^T x + b = 0, \\ & && H^T x - y = -h, \\ & && s = 1, \\ & && \begin{bmatrix} s \\ t \\ y \end{bmatrix} \in \mathcal{K}^r. \end{aligned} \quad (10)$$





or more compactly

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && t + a^T x + b = 0, \\ & && \begin{bmatrix} 1 \\ t \\ H^T x + h \end{bmatrix} \in \mathcal{K}^r. \end{aligned} \quad (11)$$

Next consider (7) which is infeasible if

$$0.5v^T v - b < 0.$$

Now assume that is not the case then (7) can be stated on the form

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && H^T x - y = -v, \\ & && t = \sqrt{0.5v^T v - b}, \\ & && \begin{bmatrix} t \\ y \end{bmatrix} \in \mathcal{K}^q \end{aligned} \quad (12)$$

or compactly

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \begin{bmatrix} \sqrt{0.5v^T v - b} \\ H^T x + v \end{bmatrix} \in \mathcal{K}^q. \end{aligned} \quad (13)$$

Next let us consider the problem (9) which can be reformulated as a conic quadratic optimization problem as follows. First we scale the x variable by $D^{-\frac{1}{2}}$ i.e. we replace x by $D^{-\frac{1}{2}}\bar{x}$ to obtain

$$\begin{aligned} & \text{minimize} && c^T D^{-\frac{1}{2}} x \\ & \text{subject to} && \frac{1}{2}(x^T x + y^T y) + a^T x + b \leq 0, \\ & && H^T D^{-\frac{1}{2}} x - y = 0. \end{aligned} \quad (14)$$

which equivalent to the conic quadratic problem

$$\begin{aligned} & \text{minimize} && c^T D^{-\frac{1}{2}} \bar{x} \\ & \text{subject to} && t + a^T D^{-\frac{1}{2}} \bar{x} + b = 0, \\ & && H^T D^{-\frac{1}{2}} \bar{x} - y = 0, \\ & && \begin{bmatrix} 1 \\ t \\ \bar{x} \\ y \end{bmatrix} \in \mathcal{K}^r. \end{aligned} \quad (15)$$

When the optimal solution to (15) is computed then original solution can obtained from

$$x = D^{-\frac{1}{2}} \bar{x}.$$

5 Linear least squares problems with inequalities

An example of generalized linear least square problem is

$$\begin{aligned} & \text{minimize} && \|H^T x + h\| \\ & \text{subject to} && Ax \geq b. \end{aligned} \quad (16)$$

Here we will discuss how to reformulate that as a conic optimization problem.

The problem (16) can be seen as a quadratic optimization problem because minimizing

$$\|H^T x + h\|$$

or

$$\|H^T x + h\|^2 = x^T H H^T x + 2h^T H x + h^T h$$

is equivalent. Therefore, the problem (16) may be stated as the quadratic optimization problem

$$\begin{aligned} & \text{minimize} && x^T H H^T x + 2h^T H x + h^T h \\ & \text{subject to} && Ax \geq b. \end{aligned} \quad (17)$$



The conic quadratic reformulation of (16) is trivial because it is

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} t \\ Ax \\ H^T x + h \end{bmatrix} \in \mathcal{K}^q, \end{array} \geq b, \quad (18)$$

whereas the conic reformulation of (17) is

$$\begin{array}{ll} \text{minimize} & t + 2h^T Hx + h^T h \\ \text{subject to} & \begin{bmatrix} t \\ 0.5 \\ Ax \\ H^T x \end{bmatrix} \in \mathcal{K}^r, \end{array} \geq b, \quad (19)$$

Now the question is should (18) or (19) be preferred? Consider the following problem

$$\begin{array}{ll} \text{minimize} & \|x\| \\ \text{subject to} & \sum_{j=1}^n x_j \geq \alpha, \\ & x \geq 0. \end{array} \quad (20)$$

For $n = 10$ and $\alpha = 10^4$, then MOSEK version 7 requires 22 iterations to solve (19) whereas only 6 iterations are required to solve the formulation (18) where the accuracy of the reported solution is about the same in both cases. If α is reduced to 1 the two methods formulation are equally good in terms of the number iterations. This confirms our experience that the formulation (18) usually leads to the best results. The reason for this might be that the norm is a nicer function than the squared norm in the sense that the norm of something is closer to one than the squared norm.

We therefore offer the advice that a least square objective as in (16) is not converted to a quadratic problem which then is converted to a conic problem. Rather it should directly be stated on its natural conic quadratic form (18).

6 On the numerical benefits of a conic reformulation

In this section we will demonstrate that a conic reformulation of quadratic constraint often leads to a better scaling.

Consider the quadratic constraints

$$x^T x \leq 10^{-12}$$

and

$$x^T x \leq 10^{12}.$$

The conic reformulations are

$$\begin{array}{ll} t & = 10^{-6}, \\ \begin{bmatrix} t \\ x \end{bmatrix} & \in \mathcal{K}^q \end{array}$$

and

$$\begin{array}{ll} t & = 10^6, \\ \begin{bmatrix} t \\ x \end{bmatrix} & \in \mathcal{K}^q. \end{array}$$

respectively. Observe, the numbers appearing in the conic reformulation is much closer to one and hence the problems are much better scaled.

Next consider the quadratic constraint

$$10^{-2}x_1^2 + 10^{-4}x_2^2 + 10^{-8}x_3^2 \leq 10^{-8}$$

which has the conic quadratic reformulation

$$\begin{array}{ll} t & = 10^{-4}, \\ y_1 & = 10^{-1}x_1, \\ y_2 & = 10^{-3}x_2, \\ y_3 & = 10^{-4}x_3, \\ \begin{bmatrix} t \\ y \end{bmatrix} & \in \mathcal{K}^q \end{array}$$





Observe that the conic reformulation is better scaled. Indeed the relative difference between the biggest and smallest number is reduced by 3 orders of magnitude by doing the conic reformulation.

Finally, we may use the substitution

$$y = 10^{-2}\bar{y}$$

to obtain

$$\begin{aligned} t &= 10^{-2}, \\ \bar{y}_1 &= 10^1 x_1, \\ \bar{y}_2 &= 10^{-1} x_2, \\ \bar{y}_3 &= 10^{-2} x_3, \\ \begin{bmatrix} t \\ \bar{y} \end{bmatrix} &\in \mathcal{K}^q \end{aligned}$$

which improves the scaling further.

Clearly, the conic reformulation is bigger because there are more variables and hence may take longer time to solve. However, the additional constraints and variables are very sparse and normally this means only slightly higher computational costs per iteration. On other the reformulation may lead to fewer so in many cases the solution time will be shorter after reformulation.

7 Conclusion

In this note we have showed that when a quadratic function occur in an optimization problem then there might be different ways of representing them. Moreover, given a suitable structure in the quadratic term then a reformulation to separable form may lead to much more effective representation. In addition the reformulation leads to a much simpler and fool proof convexity check.

Finally, we have discussed how to formulate quadratic constraints using quadratic cones. In particular we argued when a least least square objective or least squares type constraints occur then they should not be converted to quadratic form and then converted to conic form. Rather the least square terms should be represented naturally in conic framework without squaring the term.





“the fast path to optimum”

MOSEK ApS provides optimization software which help our clients to make better decisions. Our customer base consists of financial institutions and companies, engineering and software vendors, among others.

The company was established in 1997 by Erling D. Andersen and Knud D. Andersen and it specializes in creating advanced software for solution of mathematical optimization problems. In particular, the company focuses on solution of large-scale linear, quadratic, and conic optimization problems.

Mosek ApS

Fruebjergvej 3
2100 Copenhagen
Denmark

www.mosek.com
info@mosek.com

