Solving Nonconvex QCQP: Convex-Relaxation then Local Refinement

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

HQCQP: Homogeneous Case

Homogeneous Quadratically Constrained Quadratic Programming: Given symmetric matrices $Q,\ A_i's$, find global optimal solution of the following optimization problem

$$
\text{(HQCQP)} \quad \begin{array}{lll} z^* := & \text{Maximize} & \mathbf{x}^T Q\mathbf{x} \ (= Q \bullet (\mathbf{x}\mathbf{x}^T)) \\ & \text{s.t.} & \mathbf{x}^T A_i \mathbf{x} \ (\leq, =, \geq) \ b_i, \ \forall i = 1, ..., m. \end{array}
$$

Note that if $\mathbf{x}^*$ is a (global) optimal solution, so is $-\mathbf{x}^*$.

Rank-Constrained SDP Formulation:

$$
\begin{array}{lll} z^* := & \text{Maximize}_{X \in \mathcal{S}^n} & Q \bullet X \\ & \text{s.t.} & A_i \bullet X \ (\leq, =, \geq) \ b_i, \ \forall i = 1, ..., m, \\ & & X \succeq \mathbf{0}, \ \text{rank}(X) = 1. \end{array}
$$

SDP Relaxation: remove the rank constraint.

QCQP: General Case

Quadratically Constrained Quadratic Programming:

$$\begin{array}{llll} & z^* := & \text{Maximize} & \mathbf{x}^T Q\mathbf{x} + 2\mathbf{q}^T\mathbf{x} \\ \text{(QCQP)} & & & \\ & & \text{s.t.} & \mathbf{x}^T A_i\mathbf{x} + 2\mathbf{a}_i^T\mathbf{x}\ (\leq, =, \geq)\ b_i,\ \forall i = 1, ..., m. \end{array}$$

that can be homogenized by adding an auxiliary variable:

$$\begin{array}{llll} & z^* := & \text{Maximize} & \mathbf{x}^T Q\mathbf{x} + 2x_{n+1}\mathbf{q}^T\mathbf{x} \\ \text{(HQCQP)} & & & \\ & & \text{s.t.} & \mathbf{x}^T A_i\mathbf{x} + 2x_{n+1}\mathbf{a}_i^T\mathbf{x}\ (\leq, =, \geq)\ b_i,\ \forall i = 1, ..., m, \\ & & & x_{n+1}^2 = 1. \end{array}$$

GUROBI “solves” nonconvex QCQP by a “branching and cut” algorithm.

QCQP SDP Relaxation

$$
\begin{array}{rll}
z^{SDP} := & \text{Maximize} & \begin{pmatrix} Q & \mathbf{q} \\ \mathbf{q}^T & 0 \end{pmatrix} \bullet X \\
& \text{s.t.} & \begin{pmatrix} A_i & \mathbf{a}_i \\ \mathbf{a}_i^T & 0 \end{pmatrix} \bullet X = b_i, \ \forall i = 1, ..., m, \\
& & \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \bullet X = 1, \\
& & X \succeq 0,
\end{array}
$$

where $X \in \mathcal{S}^{n+1}$.

QCQP Application: Binary Least Squares

$$
\text{(BLS)} \quad
\begin{array}{lll}
z^* := & \text{Minimize} & \|A\mathbf{x} - \mathbf{c}\|^2 \\
& \text{s.t.} & x_j(1 - x_j) = 0, \ \forall i = 1, ..., n.
\end{array}
$$

In the following, we describe a general hybrid scheme to “solve” QCQP:

- Solve the SDP relaxation and construct a vector solution $\mathbf{x}^{SDP}$.
- Use $\mathbf{x}^{SDP}$ as an initial solution to start an iterative optimization or local search solver/method, such as SDM, to find a “local” optimal solution (probably near $\mathbf{x}^{SDP}$).

If the original QCQP is a feasibility problem, then the optimization problem in the second step is a nonconvex least-squares problem.

Example I: Ball-Constrained Quadratic Optimization

$$
\text{(BQP)} \qquad \begin{array}{lll} z^* := & \text{Maximize} & \mathbf{x}^T Q\mathbf{x} + 2\mathbf{q}^T\mathbf{x} \\ & \text{s.t.} & \|\mathbf{x}\|^2 = 1. \end{array} \qquad (1)
$$

Here, the given matrix $Q \in \mathcal{S}^n$ (the set of n-dimensional symmetric matrices), vector $\mathbf{q} \in R^n$; and $\|.\|$ is the Euclidean norm.

Let $X \in \mathcal{S}^{n+1}$,

$$
Q' = \begin{pmatrix} Q & \mathbf{q} \\ \mathbf{q}^T & 0 \end{pmatrix}, \; I_{1:n} = \begin{pmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix}, \text{ and } \; I_{n+1} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.
$$

Then we can write its SDP relaxation...

BQP SDP Relaxation

$$
\begin{array}{lrl}
z^{SDP} = & \text{Maximize} & Q' \bullet X \\
\text{(SDP)} & & \\
& \text{s.t.} & I_{1:n} \bullet X = 1, \\
& & I_{n+1} \bullet X = 1, \\
& & X \succeq \mathbf{0}.
\end{array}
$$

The dual of (SDP) can be written as:

$$
\begin{array}{lrl}
z^{SDP} = & \text{Minimize} & y_1 + y_2 \\
\text{(DSDP)} & & \\
& \text{s.t.} & y_1 I_{1:n} + y_2 I_{n+1} - S = Q', \\
& & S \succeq \mathbf{0}.
\end{array}
$$

Exactness Result

X^* is an optimal solution matrix to SDP if and only if there exist a feasible dual variables (y_1^*, y_2^*) such that

$$S^* = y_1^* I_{1:n} + y_2^* I_{n+1} - Q' \succeq \mathbf{0}$$

$$S^* \bullet X^* = 0.$$

Observation: $z^{SDP} \geq z^*$.

Theorem 1 *The SDP relaxation is exact for (BQP), meaning $z^{SDP} = z^*$. Moreover, there is a rank-1 SDP solution $X^* = \mathbf{x}\mathbf{x}^T$ such that*

$$\mathbf{x}^* = \mathbf{x}(1:n)/x_{n+1}$$

is an optimal solution to (BQP), where such a rank-one solution can be produced by the null-space reduction in Lecture Note #5.

No need to continue the second iterative step!

Example II: the Singular Value Problem

Given matrix $A \in R^{m\times n}$, the singular value of the matrix is the optimization problem:

$$
\begin{array}{lll}
z^* := & \text{Maximize} & \mathbf{y}^T A\mathbf{x} \\
\text{SVP} & & \\
& \text{s.t.} & \|\mathbf{x}\|^2 = 1, \\
& & \|\mathbf{y}\|^2 = 1.
\end{array} \tag{2}
$$

Let $Z \in \mathcal{S}^{n+m}$,

$$
Q' = \begin{pmatrix} \mathbf{0} & A^T \\ A & \mathbf{0} \end{pmatrix}, \; I_x = \begin{pmatrix} I_{1:n} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{0} \end{pmatrix}, \text{ and } \; I_y = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & I_{1:m} \end{pmatrix}.
$$

Then we can write its SDP relaxation...

SVP SDP Relaxation

$$
\begin{array}{lrll}
 & z^{SDP} = & \text{Maximize} & Q' \bullet Z \\
\text{(SDP)} & & & \\
 & & \text{s.t.} & I_x \bullet Z = 1, \\
 & & & I_y \bullet Z = 1, \\
 & & & Z \succeq \mathbf{0}.
\end{array}
$$

The dual of (SDP) can be written as:

$$
\begin{array}{lll}
 & \text{Minimize} & y_1 + y_2 \\
\text{(DSDP)} & & \\
 & \text{s.t.} & y_1 I_x + y_2 I_y - S = Q', \\
 & & S \succeq \mathbf{0}.
\end{array}
$$

Exactness Result

Theorem 2 *The SDP relaxation is exact for (SVP), meaning* $z^{SDP} = z^*$*. Moreover, there is a rank-1 SDP solution* $Z^* = (\mathbf{x}^*; \mathbf{y}^*)(\mathbf{x}^*; \mathbf{y}^*)^T$ *such that* $\mathbf{x}^*$ *and* $\mathbf{y}^*$ *is an optimal solution to (SVP), where such a rank-one solution can be produced by the null-space reduction in Lecture Note #5.*

No need to continue the second iterative step!

Example III: Statistical Regret Sum Minimization

Consider a fractional optimization problem

$$\begin{array}{ll} \min_{(\alpha\in R^p,\beta\in R^d)} & \sum_{i=1}^n \frac{(y_i-\mathbf{x}_i^T\beta)^2}{\mathbf{z}_i^T\alpha} \\ \text{s.t.} & \sum_{j=1}^p \alpha_j = 1,\ \alpha_j \geq 0\ \forall j = 1,...,p. \end{array}$$

Here, given data $\mathbf{z}_i$ is a non-negative and non-zero vector for all i.

For each term $\frac{(y_i-\mathbf{x}_i^T\beta)^2}{\mathbf{z}_i^T\alpha}$, one may introduce three scalar variables $(\gamma_i, \sigma_i, q_i)$ and reformulates the problem as a CLP.

A CLP Reformulation

$$
\begin{array}{ll}
\min_{(\alpha\in R^p,\beta\in R^d,\gamma,\sigma,\mathbf{q})} & \sum_{i=1}^n \gamma_i \\
\text{s.t.} & \sum_{j=1}^p \alpha_j = 1,\ \alpha_j \ge 0,\ \forall j, \\
& \mathbf{z}_i^T\alpha - q_i = 0,\ \forall i, \\
& \mathbf{x}_i^T\beta - \sigma_i = y_i,\ \forall i, \\
& \begin{pmatrix} \gamma_i & \sigma_i \\ \sigma_i & q_i \end{pmatrix} \succeq \mathbf{0},\ \forall i.
\end{array}
$$

This is a standard (dual) conic linear optimization problem with a product of many 2×2 positive semi-definite matrix cones, and a non-negative cones on α.

Validation of the Reformulation

The positive semi-definiteness of the 2×2 matrix means $\gamma_i q_i \geq \sigma_i^2$. $\gamma_i \geq 0,\ q_i \geq 0$ so that

$$\gamma_i \geq \frac{\sigma_i^2}{q_i} = \frac{(y_i - \mathbf{x}_i^T \beta)^2}{\mathbf{z}_i^T \alpha}.$$

Since γ_i is a upper bound variable on each term, minimizing $\sum_i \gamma_i$ would make the bound tight so that it is equivalent to minimizing

$$\sum_i \frac{(y_i - \mathbf{x}_i^T \beta)^2}{\mathbf{z}_i^T \alpha}.$$

The new formulation has $2n+1$ equality constraints with two-block structures (one involving $(\alpha, \mathbf{q})$ and the other (β, σ)). One may add more linear constraints to link α and β, and it becomes a standard conic linear optimization problem.

Standard Dual CLP Form

$$
\begin{aligned}
\max_{(\alpha\in R^p,\beta\in R^d,\gamma\in R^n)} \quad & -\sum_{i=1}^n \gamma_i \\
\text{s.t.} \quad & \begin{pmatrix} -\gamma_i & \mathbf{x}_i^T\beta \\ \mathbf{x}_i^T\beta & -\mathbf{z}_i^T\alpha \end{pmatrix} \preceq \begin{pmatrix} 0 & y_i \\ y_i & 0 \end{pmatrix}, \ \forall i, \\
& \sum_{j=1}^p \alpha_j \le 1, \\
& -\alpha_j \le 0, \ \forall j.
\end{aligned}
$$

Note that the left-hand-side of the matrix inequality can be written as

$$
\sum_{j=1}^p \alpha_j \begin{pmatrix} 0 & 0 \\ 0 & -z_{ij} \end{pmatrix} + \sum_{j=1}^d \beta_j \begin{pmatrix} 0 & x_{ij} \\ x_{ij} & 0 \end{pmatrix} + \gamma_i \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}.
$$

It would be solved by a standard conic linear programming solvers.

Exactness Result: the CLP relaxation is exact and no need for the second iterative step.

Example IV: Binary Quadratic Maximization

$$\text{(BQM)} \quad z^* := \quad \text{Maximize} \quad \mathbf{x}^T Q \mathbf{x}$$
$$\text{subject to} \quad (x_j)^2 = 1,\ j = 1, ..., n.$$

where Q is positive semidefinite.

Even if the original Q is not psd, when can still formulate an equivalent problem by

$$Q := Q + |\lambda| \cdot I \succeq \mathbf{0}$$

where λ is the minimal eigenvalue of Q.

Semidefinite Relaxation for (BQM)

$$
\begin{array}{rll}
z^{SDP} := & \text{Maimize} & Q \bullet X \\
 & \text{s.t.} & I_j \bullet X = 1,\ j = 1, ..., n, \\
 & & X \succeq \mathbf{0}.
\end{array}
$$

$$
\begin{array}{rll}
z^{SDP} = & \text{Minimize} & \mathbf{e}^T \mathbf{y} \\
 & \text{s.t.} & \text{Diag}(\mathbf{y}) \succeq Q.
\end{array}
$$

Approximation Ratio by Randomized Rank-Reduction

Let $\bar{X}$ be an optimal matrix solution of the SDP relaxation, and let random vector

$$\mathbf{u} \in N(\mathbf{0}, \bar{X}) \quad \text{and} \quad \hat{\mathbf{x}} = \mathsf{Sign}(\mathbf{u})$$

where $\mathsf{Sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$. Clearly, $\hat{\mathbf{x}}$ is binary, and

$$\mathsf{E}[\hat{\mathbf{x}}^T Q \hat{\mathbf{x}}] = Q \bullet \mathsf{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^T] = Q \bullet \frac{2}{\pi} \arcsin[\bar{X}] \geq \frac{2}{\pi} Q \bullet \bar{X}.$$

Theorem 3 *For solving (BQM), we have an approximation ratio $\frac{2}{\pi}$, that is, one can find a feasible solution $\hat{\mathbf{x}}$ such that*

$$\mathsf{E}[\hat{\mathbf{x}}^T Q \hat{\mathbf{x}}] \geq \frac{2}{\pi} z^{SDP} \geq \frac{2}{\pi} z^*.$$

One can start from $\hat{\mathbf{x}}$ and apply any binary QP solver in the second iterative step.

Example V: SNL

Given a graph $G = (V, E)$ and sets of non–negative weights, say $\{d_{ij} : (i, j) \in E\}$, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d, where the distance information is preserved.

More precisely: given anchors $\mathbf{a}_k \in \mathbf{R}^d$, $d_{ij} \in N_x$, and $\hat{d}_{kj} \in N_a$, find $\mathbf{x}_i \in \mathbf{R}^d$ such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{ij}^2, \ \forall\, (i, j) \in N_x, \ i < j,$$
$$\|\mathbf{a}_k - \mathbf{x}_j\|^2 = \hat{d}_{kj}^2, \ \forall\, (k, j) \in N_a,$$

This is a QCQP feasibility problem.

Matrix Representation and SDP Relaxation

Let $X = [\mathbf{x}_1\ \mathbf{x}_2\ \dots\ \mathbf{x}_n]$ be the $d \times n$ matrix that needs to be determined and $\mathbf{e}_j$ be the vector of all zero except 1 at the jth position. The SDP relaxation is also an SDP feasibility problem:

$$
\begin{aligned}
&(\mathbf{e}_i - \mathbf{e}_j)^T(\mathbf{e}_i - \mathbf{e}_j) \bullet Y = d_{ij}^2,\ \forall\, i, j \in N_x,\ i < j,\\
&(\mathbf{a}_k; -\mathbf{e}_j)^T(\mathbf{a}_k; -\mathbf{e}_j) \bullet \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} = \hat{d}_{kj}^2,\ \forall\, k, j \in N_a,\\
&Y \succeq X^T X.
\end{aligned}
$$

First-Step: Solve the SDP feasibility problem and let X^{SDP} be the position solution of the optimal matrix solution. If the problem is "anchor-free", let X^{SDP} be the eigenvectors of Y^{SDP} corresponding to the d largest eigenvalues.

Second-Step: Using X^{SDP} as the initial solution and apply SDM in minimizing nonlinear least-squares function:

$$
\min_X \sum_{(i<j,j)\in N_x} (\|\mathbf{x}_i - \mathbf{x}_j\|^2 - d_{ij}^2)^2 + \sum_{(k,j)\in N_a} (\|\mathbf{a}_k - \mathbf{x}_j\|^2 - \hat{d}_{kj}^2)^2.
$$

Example VI: The Kissing Number Problem

Given a unit ball centered at the origin in dimension d, the maximum number of other unit balls can touch or kiss the centered unit-ball?

The Kissing Problem as a QCQP: Given n unit-balls, could we find their center locations for all of them such that the following problem is feasible:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &\geq 4, \ \forall 1 \leq i \neq j \leq n, \\ \|\mathbf{x}_i\|^2 &= 4, \ \forall i \\ \mathbf{x}_i &\in R^d \end{aligned}$$

Equivalent SDP relaxation:

$$\begin{aligned} (\mathbf{e}_i - \mathbf{e}_j)^T Y (\mathbf{e}_i - \mathbf{e}_j) &\geq 4, \ \forall i \neq j, \\ \mathbf{e}_i^T Y \mathbf{e}_i &= 4, \ \forall i \\ Y &\succeq \mathbf{0}; \quad (\text{Rank}(Y) = d) \end{aligned}$$

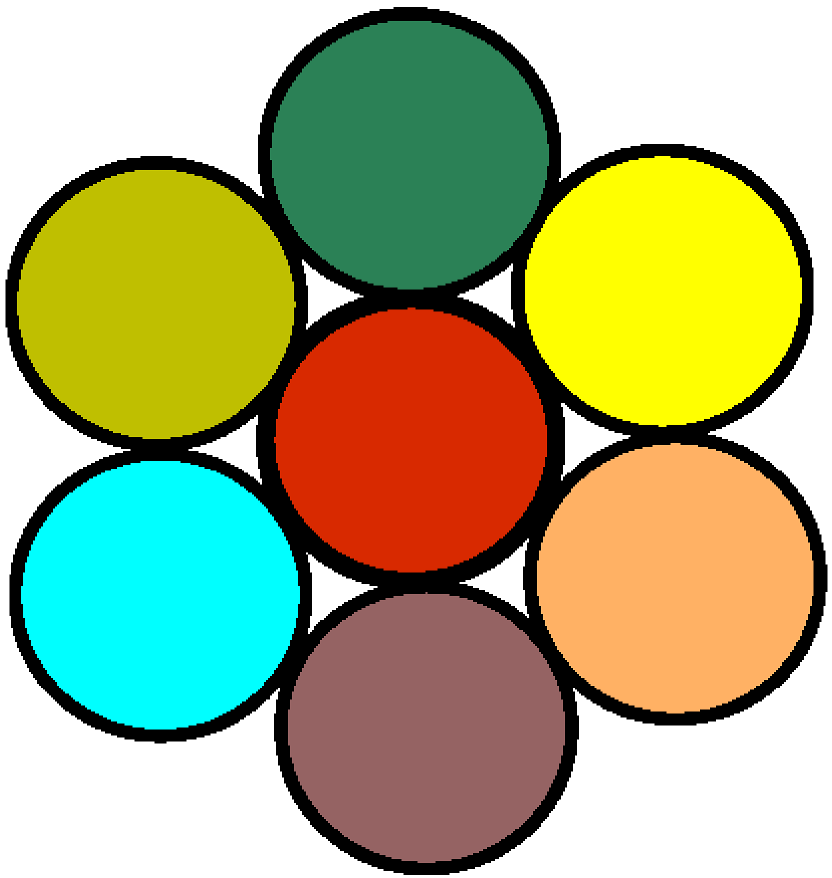

Figure 1: Kissing Localization in 2D

Optimization Algorithm Summaries and Remarks

- Optimization algorithms tend to be iterative procedures. Starting from a given point $\mathbf{x}^0$, they generate a sequence $\{\mathbf{x}^k\}$ of iterates (or trial solutions) that converge to a "solution" – or at least they are designed to be so.

- We study algorithms that produce iterates according to

  - well determined rules–Deterministic Algorithm

  - random selection process–Randomized Algorithm.

  The rules to be followed and the procedures that can be applied depend to a large extent on the characteristics of the problem to be solved.

- Most algorithms Descent Direction in nature, but some of them may not monotonically decrease the objective and more "strategic", such as the BB stepsize in SDM or the potential-reduction algorithm for LP.

- Algorithm convergence and speed

  - Finite versus infinite convergence. For some classes of optimization problems there are algorithms that obtain an exact solution—or detect the unboundedness–in a finite number of iterations.

– Polynomial-time versus exponential-time. The solution time grows, in the worst-case, as a function of problem sizes (number of variables, constraints, accuracy, etc.).

– Convergence order and rate.

- Algorithm Classes: Depending on information of the problem being used to create a new iterate, we have

(a) Zero-order algorithms. Popular when the gradient and Hessian information are difficult to obtain, e.g., no explicit function forms are given, functions are not differentiable, etc.

Golden-Section Method for one-dimensional search - linear convergence rate 0.618

(b) First-order algorithms. Most popular now-days, suitable for large scale data optimization with low accuracy requirement, e.g., Machine Learning, Statistical Predictions...

Bi-Section Method for one-dimensional search - linear convergence rate 0.5

(c) Second-order algorithms. Popular for optimization problems with high accuracy need, e.g., some scientific computing, etc.

Newton's method: superior local quadratic (order-two) convergence, but may fail globally.

Most algorithm analyses are based on solving various Lipschitz conditions/constants.

All algorithms allow some inexactness in numerical computation.

- Algorithm customization becomes a trend, that is, general-purpose algorithms needs to be adapted by exploiting specific problem structures and domain knowledges/heuristics where Machine Learning could play an important role.
- Algorithm for LP (even with integer variables) are matured and algorithms for most convex optimization are well understood now, but search for the global optimal solution of general nonconvex optimization remains illusive, most of which are in computation classes such as NP-Complete, NP-Hard, PPAD, #P-Hard, etc.. Common approaches:
  - Starting from multiple initial solutions
  - Adding randomness/inexactness to escape local trap
  - Solving convex relaxation (if you have one), then applying a nonconvex solver to refine the solution obtained from the relaxation.