Practical Optimization Algorithms and Applications

Chapter XI: Quadratic Programming

Lingfeng NIU

Research Center on Fictitious Economy & Data Science,
University of Chinese Academy of Sciences

niulf@ucas.ac.cn

Outline

Outline

Example - Portfolio Optimization

Every investor knows that there is a tradeoff between risk and return:
To increase the expected return on investment, an investor must be willing to tolerate greater risks.

Example - Portfolio Optimization

Every investor knows that there is a tradeoff between risk and return:
To increase the expected return on investment, an investor must be willing to tolerate greater risks.
Portfolio theory studies how to model this tradeoff given a collection of n possible investments with returns r_i, $i = 1, 2, \cdots, n$. The returns r_i are usually not known in advance, and are often assumed to be random variables that follow a normal distribution. We can characterize these variables by

- the expected value $\mu_i = E[r_i]$;
- the variance $\sigma_i^2 = E[(r_i - \mu_i)^2]$.

The variance measures the fluctuations of the variable r_i about its mean, so that larger values of σ_i indicate riskier investments.

Example - Portfolio Optimization

The returns are not in general independent, and we can define correlations between pairs of returns as follows:

$$\rho_{ij} = \frac{E[(r_i - \mu_i)(r_j - \mu_j)]}{\sigma_i \sigma_j}, \qquad \forall i, j = 1, 2, \cdots, n.$$

The correlation measures the tendency of the return on investments i and j to move in the same direction.

- Two investments whose returns tend to rise and fall together have a positive covariance.
- The nearer ρ_{ij} is to 1, the more closely the two investments track each other.
- Investments whose returns tend to move in opposite directions have negative covariance.

Example - Portfolio Optimization

An investor constructs a portfolio by putting a fraction x_i of the available funds into investment i, for $i = 1, 2, \cdots, n$. Assuming that all available funds are invested and that short-selling is not allowed, the constraints are $\sum_{i=1}^{n} x_i = 1$ and $x \geq 0$. The return on the portfolio is given by

$$R = \sum_{i=1}^{n} x_i r_i. \tag{1}$$

Example - Portfolio Optimization

To measure the desirability of the portfolio, we need to obtain measures of its expected return variance. The expected return is simply

$$E(R) = E\left[\sum_{i=1}^{n} x_i r_i\right] = \sum_{i=1}^{n} x_i E[r_i] = x^T \mu. \tag{2}$$

while the variance is given by

$$\mathrm{Var}[R] = E[(R - E[R])^2] = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \sigma_i \sigma_j \rho_{ij} = x^T G x,$$

where the $n \times n$ symmetric positive semidefinite matrix G defined by $G_{ij} = \rho_{ij}\sigma_i\sigma_j$ is called the *covariance matrix.*

Example - Portfolio Optimization

We are interested in portfolios for which the expected return $x^T\mu$ is large while the variance $x^T Gx$ is small. Combine these two aims into a single objective function with the aid of a "risk tolerance parameter" denoted by κ, and solve the following problem to find the "optimal" portfolio:

$$\max x^T\mu - \kappa x^T Gx, \text{ s.t. } \sum_{i=1}^{n} x_i = 1, x \geq 0. \tag{3}$$

The parameter κ lies in the range $[0, \infty)$, and its chosen value depends on the preferences of the individual investor.

- Conservative investors would place more emphasis on minimizing risk in their portfolio, so they would choose a large value of κ to increase the weight of the variance measure in the objective function.
- More daring investors are prepared to take on more risk in the hope of a higher expected return, so their value of κ would be closer to zero.

Example - Portfolio Optimization

The difficulty in applying this portfolio optimization technique to real-life investing lies in defining the expected returns, variances, and covariances for the investments in question. One possibility is to use historical data, defining the quantities μ_i, σ_i, and ρ_{ij} to be equal to their historical values between the present day and, say, five years ago. It is not wise to assume that future performance will mirror the past, of course. Moreover, historical data will not be available for many interesting investments (such as start-up companies based on new technology). Financial professionals often combine the historical data with their own insights and expectations to produce values of μ_i, σ_i, and ρ_{ij}.

Introduction

An optimization problem with a quadratic objective function and linear constraints is called a *quadratic program*(QP).

- QPs are important in their own right;
- QPs arise as subproblems in methods for general constrained optimization, such as sequential quadratic programming, augmented Lagrangian methods, and interior-point methods.

Introduction

The general QP can be stated as

$$\min_x \quad q(x) = \tfrac{1}{2} x^T G x + x^T c \tag{4a}$$

$$s.t. \quad a_i^T x = b_i, \qquad i \in \mathcal{E}, \tag{4b}$$

$$a_i^T x \geq b_i, \qquad i \in \mathcal{I}, \tag{4c}$$

where G is a symmetric $n \times n$ matrix, $\mathcal{E}$ and $\mathcal{I}$ are finite sets of indices, and d, x, and $\{a_i\}$, $i \in \mathcal{E} \cup \mathcal{I}$, are vectors with n elements.

Introduction

QPs can always be solved (or can be shown to be infeasible) in a finite number of iterations, but the effort required to find a solution depends strongly on the characteristics of the objective function and the number of inequality constraints.

- If the Hessian matrix G is positive semidefinite, we say that (4) is a convex QP, and in this case the problem is sometimes not much more difficult to solve than a linear program.
- Nonconvex QPs, in which G is an indefinite matrix, can be more challenging, since they can have several stationary points and local minima.

We limit ourselves to studying algorithms that **finding the solution of a convex quadratic program or a stationary point of a general (nonconvex) quadratic program**.

Outline

Equality-Constrained Quadratic Programs

We begin our discussion of algorithms for QP by considering the case where only equality constraints are present. Techniques for this special case are applicable also to problem with inequality constraints. Active set methods for general QP solve an equality-constrained QP at each iteration.

- properties of equality-constrained QPs
- direct solution of the KKT system
- iterative solution of the KKT system

Properties of Equality-Constrained QPs

Let us denote the number of constraints by m, assume that $m \leq n$, and write the quadratic program as

$$\min q(x) \equiv \frac{1}{2} x^T G x + x^T c \text{ s.t. } Ax = b, \tag{5}$$

where A is the $m \times n$ Jacobian of constraints (with $m \leq n$) whose rows are a_i^T, $i \in \mathcal{E}$ and b is the vector in $\Re^m$ whose components are b_i, $i \in \mathcal{E}$. For the present, we assume that A has full row rank so that the constraints are consistent.

Properties of Equality-Constrained QPs

The first-order necessary conditions for x^* to be a solution of (5) state that there is a vector λ^* such that the following system of equations is satisfied:

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}. \tag{6}$$

This system can be rewritten in a form that is useful for computation by expressing x^* as $x^* = x + p$, where x is some estimate of the solution and p is the desired step. By introducing this notation and rearranging the equations, we obtain

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \tag{7}$$

where

$$h = Ax - b, \qquad g = c + Gx, \qquad p = x^* - x.$$

The matrix in (7) is called the Karush-Kuhn-Tucker (KKT) matrix.

Properties of Equality-Constrained QPs

We use Z to denote the $n \times (n-m)$ matrix whose columns are a basis for the null space of A. That is, Z has full rank and satisfies $AZ = 0$.

Theorem

Let A have full row rank, and assume that the reduced-Hessian matrix $Z^T GZ$ is positive definite. Then the KKT matrix

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \tag{8}$$

is nonsingular, and there is a unique vector pair (x^*, λ^*) satisfying (16) and x^* is the unique global solution of (5).

When the reduced Hessian matrix $Z^T GZ$ is positive semidefinite with zero eigenvalues, the vector x^* satisfying (16) is a local minimizer but not a strict local minimizer. If the reduced Hessian has negative eigenvalues, then x^* is only a stationary point, but not a local minimizer.

Direct Solution of the KKT System

The first important observation is that if $m \geq 1$, the KKT matrix (8) is always indefinite. We define the inertia of a symmetric matrix to be the scalar triple that indicates the numbers n_+, n_-, and n_0 of positive, negative, and zero eigenvalues, respectively, that is,

$$\text{inertia}(K) = (n_+, n_-, n_0).$$

The following result characterizes the inertia of the KKT matrix.

Theorem

Let K be defined by (8), and suppose that A has rank m. Then

$$\textit{inertia}(K) = \textit{inertia}(Z^T G Z) + (m, m, 0).$$

Therefore, if $Z^T G Z$ is positive definite, $\textit{inertia}(K) = (n, m, 0)$.

Knowing that the KKT system is indefinite, we now describe the main direct techniques used to solve (7).

Factoring the Full KKT System

One option for solving (7) is to **perform a triangular factorization on the full KKT matrix and then perform backward and forward substitution with the triangular factors**.

- We cannot use the Cholesky factorization algorithm because the KKT matrix is indefinite.
- We could use Gaussian elimination with partial pivoting to obtain the L and U factors, but this approach has the disadvantage that it ignores the symmetry.
- Currently, the most effective strategy in this case is to use a *symmetric indefinite factorization*.

Factoring the Full KKT System

For a general symmetric matrix K, there exist a permutation matrix P, a unit lower triangular L, and a block-diagonal B with either 1×1 or 2×2 blocks, such that

$$P^T K P = L B L^T,$$

The computational cost of this symmetric indefinite factorization is typically about half the cost of sparse Gaussian elimination.

The approach of factoring the full $(n+m) \times (n+m)$ KKT matrix (8) is quite effective on some problems. It may be expensive, however, when the heuristics for choosing the permutation matrix P are not able to do a very good job of maintaining sparsity in the L factor, so that L becomes much more dense than the original coefficient matrix.

Range-Space Method

Assuming that G is positive definite, we can multiply the first equation in (7) by AG^{-1} and then subtract the second equation to obtain a linear system in the vector λ^* alone:

$$(AG^{-1}A^T)\lambda^* = (AG^{-1}g - h). \tag{9}$$

We solve this symmetric positive definite system for λ^* and then recover p from the first equation in (7) by solving

$$Gp = A^T\lambda^* - g. \tag{10}$$

This approach requires us to perform operation with G^{-1}, as well as to compute the factorization of the $m \times m$ matrix $AG^{-1}A^T$. Therefore, it is most useful when:

- G is well conditioned and easy to invert;
- G^{-1} is known explicitly through a quasi-Newton updating formula;
- the number of equality constraints m is small, so that the number of backsolves needed to form the matrix $AG^{-1}A^T$ is not too large.

Null-Space Method

The null-space method does not requires nonsingularity of G and therefore has wider applicability than the range-space method. It assumes only that A have full row rank, and assume that the reduced-Hessian matrix $Z^T GZ$ is positive definite. However, it requires knowledge of the null-space basis matrix Z. Like the range-space method, it exploits the block structure in the KKT system to decouple (8) into smaller systems.
Suppose that we partition the vector p in (7) into two components, as follows:

$$p = Yp_Y + Zp_Z, \tag{11}$$

where Z is the $n \times (n - m)$ null-space matrix, Y is any $n \times m$ matrix such that $[Y|Z]$ is nonsingular, p_Y is an m-vector, and p_Z is an $(n - m)$-vector. YP_Y is a particular solution of $Ax = b$, and ZP_Z is a displacement along these constraints.

Null-Space Method

By substituting p into the second equation of (7) and recalling that $AZ = 0$, we obtain

$$(AY)p_Y = -h.$$

Since A has rank m and $[Y|Z]$ is $n \times n$ nonsingular, the product $A[Y|Z] = [AY|0]$ has rank m. Therefore, AY is a nonsingular $m \times m$ matrix, and p_Y is well determined by the above equations. Meanwhile, we can substitute (11) into the first equation of (7) to obtain p_Z by solving

$$(Z^T GZ)P_Z = -Z^T Gp_Y - Z^T g. \tag{12}$$

To obtain the Lagrange multiplier, we multiply the first equation of (7) by Y^T to obtain the linear system

$$(AY)^T \lambda^* = Y^T (g + Gp),$$

which can be solved for λ^*.

Null-Space Method

- The null-space approach can be effective when the number of degrees of freedom $n - m$ is small.
- Its main drawback is the need for the null-space matrix Z, which can be expensive to compute in many large problems. The matrix Z is not uniquely defined, and if it is poorly chosen, the reduced system (12) may become ill conditioned.
  - If we choose Z to have orthonormal columns, as is normally done in software for small and medium-sized problems, then the conditioning of $Z^T GZ$ is at least as good as that of G itself.
  - When A is large and sparse, however, this choice of Z is relatively expensive to compute, so for practical reasons we are often forced to use one of the less reliable choices of Z.

Iterative Solution of the KKT System

An alternative to the direct factorization techniques is to use an iterative method to solve the KKT system (7). **Iterative methods are suitable for solving very large systems and often lend themselves well to parallelization.**

- The conjugate gradient method is not recommended for solving the full system (7), because it can be unstable on systems that are not positive definite.
- Better options are Krylov methods for general linear or symmetric indefinite systems. Candidates include the GMRES, QMR, and LSQR methods.
- Other iterative methods can be derived from the null-space approach by applying the conjugate gradient method to the reduced system (12).

CG Applied to the Reduced System

Expressing the solution of the quadratic program (5) as

$$x^* = Yx_Y + Zx_Z, \tag{13}$$

for some vectors $x_z \in \Re^{n-m}$, $x_Y \in \Re^m$, the constraints $Ax = b$ yield

$$AYx_Y = b,$$

which determines the vector x_Y. Substituting (13) into (5), we see that x_Z solves the unconstrained reduced problem

$$\min_{x_Z} \frac{1}{2} x_Z^T Z^T G Z x_Z + x_Z^T c_Z,$$

where

$$c_Z = Z^T G Y x_Y + Z^T c.$$

The solution x_Z satisfies the linear system

$$Z^T G Z x_Z = -c_Z.$$

Since $Z^T GZ$ is positive definite, we can apply the CG method to this linear system and substitute x_Z into (13) to obtain a solution of (5).

Outline

Inequality-Constrained Quadratic Programs

We discuss several classes of algorithms for solving QPs that contain inequality constraints and possibly equality constraints.

- **Classical active-set methods** can be applied both to convex and nonconvex problems, and they have been the most widely used methods since the 1970s.
- **Gradient-projection methods** attempt to accelerate the solution process by allowing rapid changes in the active set, and are most efficient when the only constraints in the problem are bounds on the variables.
- **Interior-point methods** have recently been shown to be effective for solving large convex quadratic programs.

Optimality Conditions for Inequality-Constrained Problems

The Lagrangian for (4) is

$$\mathcal{L} = \frac{1}{2} x^T G x + x^T c - \sum_{i \in \mathcal{I} \cup \mathcal{E}} \lambda_i (a_i^T x - b_i). \tag{14}$$

The active set $\mathcal{A}(x^*)$ consists of the indices of the constraints for which equality holds at x^*:

$$\mathcal{A}(x^*) = \{ i \in \mathcal{E} \cup \mathcal{I} | a_i^T x^* = b_i \}. \tag{15}$$

Optimality Conditions for Inequality-Constrained Problems

Any solution x^* satisfies the following first-order conditions, for some Lagrangian multiplier λ_i^*, $i \in \mathcal{A}(x^*)$:

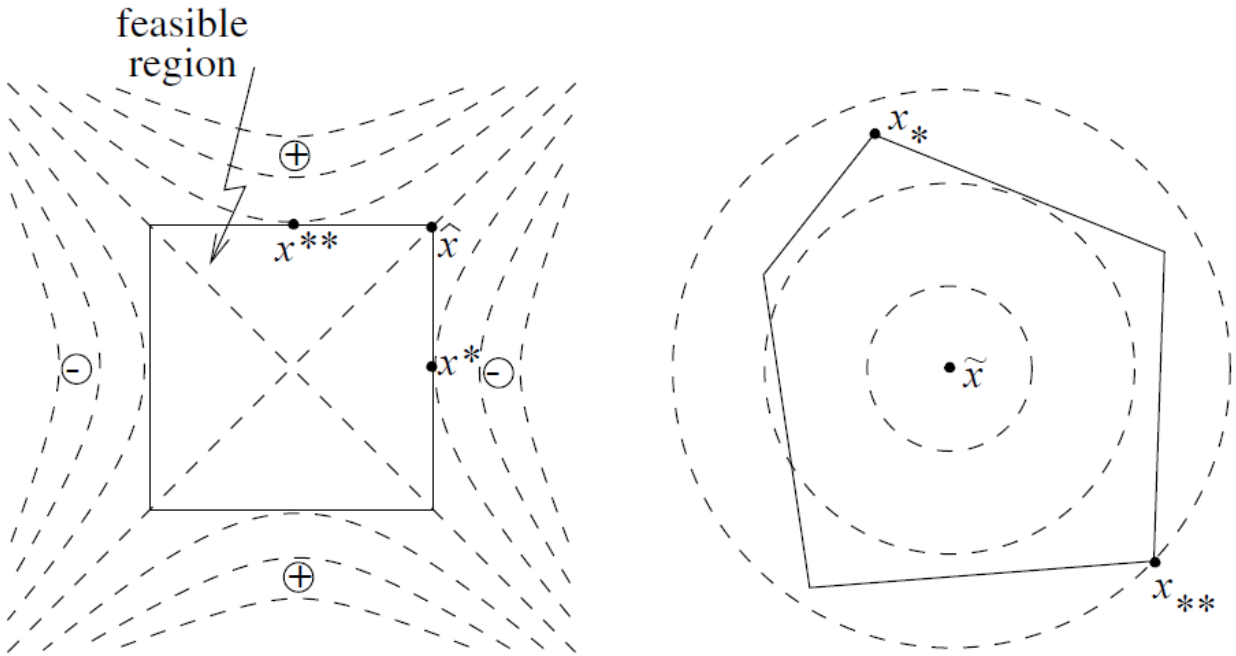
$$
\begin{aligned}
Gx^* + c - \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* a_i &= 0, && && (16a)\\
a_i^T x^* &= b_i, && \forall i \in \mathcal{A}(x^*), && (16b)\\
a_i^T x^* &\geq b_i, && \forall i \in \mathcal{I} \backslash \mathcal{A}(x^*), && (16c)\\
\lambda_i^* &\geq 0, && \forall i \in \mathcal{I} \cap \mathcal{A}(x^*). && (16d)
\end{aligned}
$$

Theorem

If x^* satisfies the conditions (16) for some λ_i^*, $i \in \mathcal{A}(x^*)$, and G is positive semidefinite, then x^* is global solution of (4). When G is positive definite, x^* is actually the unique global solution.

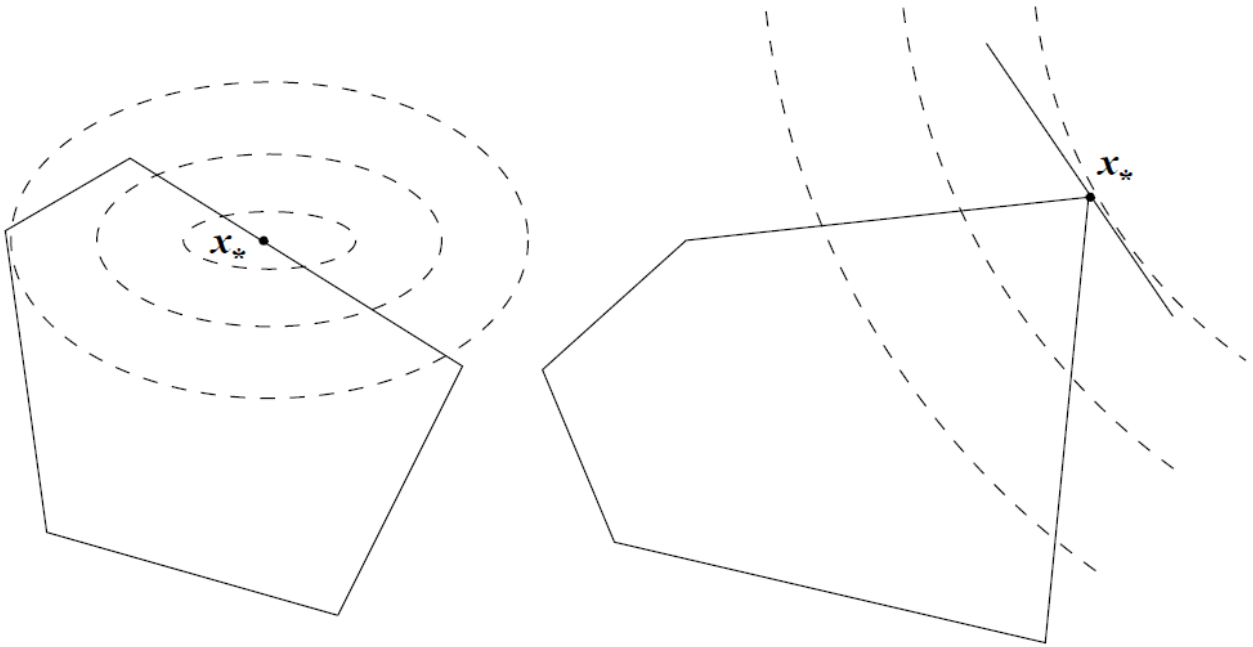
Nonconvex/Indefinite QPs

When G is not positive definite, the general problem (4) may have more than one strict local minimizer at which the second-order necessary conditions are satisfied. Such problems are referred to as being "nonconvex" or "indefinite", and they cause some complication for algorithms. Examples of indefinite quadratic programs are illustrated in the following figure.



Degeneracy

A second property that causes difficulties for some algorithms is *degeneracy* which essentially refers to situations in which either
(a) the active constraint gradients a_i, $i \in \mathcal{A}(x^*)$, are linearly dependent at the solution x^*, and/or
(b) the strict complementarity condition fails to hold, that is, there is some index $i \in \mathcal{A}(x^*)$ such that all Lagrange multipliers satisfying (16) have $\lambda^* = 0$.



Degeneracy

Degeneracy can cause problems for algorithms for two main reasons.

- First, **linear independence** of the active constraint gradients can cause numerical difficulties in the step computation because certain matrices that we need to factor become **rank deficient**.
- Second, when the problem contains **weakly active constraints**, it is difficult for the algorithm to determine whether these constraints are active at the solution. In the case of active-set methods and gradient projection methods, this indecisiveness can cause the **algorithm to zigzag** as the iterates move on and off the weakly active constraints on successive iterations. Safeguards must be used to prevent such behavior.

Active-Set Methods for Convex QPs

We now describe active-set methods, which are generally the most effective methods for small- to medium-scale problems.

- We consider only the convex case, in which the matrix G is positive semidefinite. Since the feasible region is a convex set, any local solution of the QP is a global minimizer.
- The case in which G is an indefinite matrix raises complications in the algorithms and is outside the scope of in our class. We refer to N. I. M. Gould, D. Orban, and P.L. Toint, *Numerical methods for large-scale nonlinear optimization*, Acta Numerica, 14(2005), pp. 299-361. for a discussion of nonconvex QPs.

Active-Set Methods for Convex QPs

If the contents of the *optimal* active sets were know in advance, we could find the solution x^* by applying one of the techniques for equality-constrained QP to the problem

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \text{ s.t. } a_i^T x = b_i, \ i \in \mathcal{A}(x^*). \tag{17}$$

Of course, we usually don't have prior knowledge of $\mathcal{A}(x^*)$, and determination of this set is the main challenge facing algorithms for inequality-constrained QP.

Active-Set Methods for Convex QPs

An active-set method starts by making a guess of the optimal active set, and if this guess turns out to be incorrect, it repeatedly uses gradient and Lagrange multiplier information to drop one index from the current estimate of $\mathcal{A}(x^*)$ and add a new index.

- We have already encountered an active-set approach for linear programming, namely the simplex method.
- Active-set methods for QP differ from the simplex method, in that the iterates (and the solution x^*) are not necessarily vertices of the feasible region.

Active-set methods for QP come in three varieties, known as *primal*, *dual*, and *primal-dual*. We restrict our discussion to primal methods, which generate iterates that remain feasible with respect to the primal problem (4) while steadily decreasing the primal objective function $q(\cdot)$.

Active-Set Methods for Convex QPs

Primal active-set methods usually start by computing a feasible initial iterate x_0, and then ensure that all subsequent iterates remain feasible.

They find a step from one iterate to the next by solving a quadratic subproblem in which a subset of the constraints is imposed as equalities. This subset is referred to as the *working set* and is denoted at the k-th iterate x_k by $\mathcal{W}_k$. It consists of all the equality constraints $i \in \mathcal{E}$ together with some - but not necessarily all - of the active inequality constraints.

An important requirement we impose on $\mathcal{W}_k$ is that the gradients a_i of the constraints in the working set be linearly independent, even when the full set of active constraints at that point has linearly dependent gradients.

Active-Set Methods for Convex QPs

Given an iterate x_k and the working set $\mathcal{W}_k$, we first check whether x_k minimizes the quadratic q in the subspace defined by the working set. If not, we compute a step p by solving an equality-constrained QP subproblem in which the constraints corresponding to the working set $\mathcal{W}_k$ are regarded as equalities and all other constraints are temporarily disregarded. To express this subproblem in terms of the step p, we define

$$p = x - x_k, \qquad g_k = Gx_k + c.$$

By substituting for x into the objective function in (4), we find that

$$q(x) = q(x_k + p) = \frac{1}{2} p^T G p + g_k^T p + \rho_k,$$

where $\rho_k = \frac{1}{2} x_k^T G x_k + c^T x_k$ is independent of p. We can write the QP subproblem to be solved at the kth iteration as follows:

$$\min_p \quad \tfrac{1}{2} p^T G p + g_k^T p \tag{18a}$$

$$s.t. \qquad a_i^T p = 0, \qquad i \in \mathcal{W}_k. \tag{18b}$$

Denote the solution of this subproblem by p_k.

Active-Set Methods for Convex QPs

Suppose for the moment that the optimal p_k from (18) is nonzero. We need to decide how far to move along this direction.

- If $x_k + p_k$ is feasible with respect to all the constraints, set

$$x_{k+1} = x_k + p_k$$

.

- Otherwise, choose step-length parameter α_k is chosen to be the largest value in the range $[0, 1)$ for which all constraints are satisfied and set

$$x_{k+1} = x_k + \alpha_k p_k.$$

Active-Set Methods for Convex QPs

- For $i \in \mathcal{W}_k$, since $a_i^T(x_k + \alpha p_k) = a_i^T x_k = b$ for all α, the corresponding constraints will certainly be satisfied regardless of the choice of α_k.
- For $i \notin \mathcal{W}_k$,
  - if $a_i^T p_k \geq 0$, since $a_i^T(x_k + \alpha_k p_k) \geq a_i^T x_k \geq b_i$ for all $\alpha_k \geq 0$, this constraint will be satisfied for all nonnegative choices of α;
  - if $a_i^T p_k < 0$, $a_i^T(x_k + \alpha_k p_k) \geq b_i$ only can be satisfied when

$$\alpha_k \leq \frac{b_i - a_i^T x_k}{a_i^T p_k}.$$

Since we want α_k to be as large as possible in $[0, 1]$ subject to retaining feasibility, we have the following definition:

$$\alpha_k \equiv \min\left(1, \min_{i \notin \mathcal{W}_k, a_i^T p_k < 0} \frac{b_i - a_i^T x_k}{a_i^T p_k}\right). \tag{19}$$

The constraints i for which the minimum is achieved the *blocking constraints*.

Active-Set Methods for Convex QPs

- If $\alpha_k = 1$ and no new constraints are active at $x_k + \alpha_k p_k$, then there are no blocking constraints on this iteration.
- If $\alpha_k < 1$, that is, the step along p_k was blocked by some constraint not in $\mathcal{W}_k$, a new working set $\mathcal{W}_{k+1}$ is constructed by adding one of the blocking constraints to $\mathcal{W}_k$.
- Note that it is quite possible for α_k to be zero, since we could have $a_i^T p_k < 0$ for some constraint i that is active at x_k but not a member of the current working set $\mathcal{W}_k$.

We continue to iterate in this manner, adding constraints to the working set until we reach a point $\hat{x}$ that minimizes the quadratic objective function over its current working set $\hat{\mathcal{W}}$. It is easy to recognize such a point because of the subproblem (18) has solution $p = 0$.

Since $p = 0$ satisfies the optimality conditions for (18), we have that

$$\sum_{i \in \hat{\mathcal{W}}} a_i \hat{\lambda}_i = g = G\hat{x} + c \tag{20}$$

for some Lagrange multipliers $\hat{\lambda}_i$, $i \in \hat{\mathcal{W}}$. It follows that $\hat{x}$ and $\hat{\lambda}$ satisfy the first KKT condition, if we define the multipliers corresponding to the inequality constraints that are not in the working set to be zero.

Because of the control imposed on the step-length, $\hat{x}$ is also feasible with respect to all the constraints, so the second and third KKT conditions are satisfied at this point.

Active-Set Methods for Convex QPs

We now examine the signs of the multipliers corresponding to the inequality constraints in the working set, that is, the indices $i \in \mathcal{W} \cup \mathcal{I}$.

- If these multipliers are all nonnegative, the fourth KKT condition is also satisfied, so we conclude that $\hat{x}$ is a KKT point for the original problem (4). In fact, since G is positive semidefinite, we can show that $\hat{x}$ is a local minimizer. When G is positive definite, $\hat{x}$ is a strict local minimizer.
- If, on the other hand, one of the multipliers $\hat{\lambda}_j$, $j \in \hat{\mathcal{W}} \cup \mathcal{I}$, is negative, the fourth KKT condition is not satisfied, and the objective function $q(\cdot)$ may be decreased by dropping this constraint. We then remove an index j corresponding to one of the negative multipliers from the working set and solve a new subproblem (18) for the new step. We show in the following theorem that this strategy produces a direction p at the next iteration that is feasible with respect to the dropped constraint.

Theorem

Suppose $\hat{x}$ satisfies first-order conditions for the equality-constrained subproblem with working set $\hat{\mathcal{W}}$; that is, equation (20) is satisfied along with $a_i^T \hat{x} = b_i$ for all $i \in \hat{\mathcal{W}}$. Suppose, too, that the constraint gradients a_i, $i \in \hat{\mathcal{W}}$, are linearly independent, and that there is an index $j \in \hat{\mathcal{W}}$ such that $\hat{\lambda}_j < 0$. Finally, let p be the solution obtained by dropping the constraint j and solving the following subproblem:

$$\min_p \quad \tfrac{1}{2} p^T G p + (G\hat{x} + d)^T p, \tag{21a}$$

$$s.t. \quad a_i^T p = 0, \forall i \in \hat{\mathcal{W}} \text{ with } i \neq j. \tag{21b}$$

Then p is a feasible direction for constraint j, that is, $a_j^T p \geq 0$. Moreover, if p satisfies second-order sufficient conditions for (21), then we have that $a_j^T p > 0$, and p is a descent direction for $q(\cdot)$.

Active-Set Methods for Convex QPs

While any index j for which $\hat{\lambda}_j < 0$ usually will give directions along which the algorithm can make progress, the most negative multiplier is often chosen in practice. This choice is motivated by the sensitivity analysis, which shows that the rate of decrease in the objective function when one constraint is removed is proportional to the magnitude of the Lagrange multiplier for that constraint. However, the step along the resulting direction in q is not guaranteed to be greater than for other possible choices of j.

Theorem

Suppose that the solution p_k of (18) is nonzero and satisfies the second-order sufficient conditions for optimality for that problem. Then the function $q(\cdot)$ is strictly decreasing along the direction p_k.

Specially, when G is positive definite-the *strictly* convex case-the second-order sufficient conditions are satisfied for all feasible subproblems of the form (18), so that we obtain a strict decrease in $q(\cdot)$ whenever $p_k \neq 0$. This fact is significant when we discuss finite termination of the algorithm.

Compute a feasible starting point x_0;
Set $\mathcal{W}_0$ to be a subset of the active constraints at x_0;
for $k = 0, 1, 2, \ldots$
  Solve (18) to find p_k;
  if $p_k = 0$
    Compute Lagrange multipliers $\hat{\lambda}_i$ that satisfy (20),
      set $\hat{\mathcal{W}} = \mathcal{W}_k$;
    if $\hat{\lambda}_i \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$;
      STOP with solution $x^* = x_k$;
    else
      Set $j = \arg\min_{j \in \mathcal{W}_k \cap \mathcal{I}} \hat{\lambda}_j$;
      $x_{k+1} = x_k$; $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \backslash \{j\}$;
  else (* $p_k \neq 0$ *)
    Compute α_k from (19);
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
    if there are blocking constraints
      Obtain $\mathcal{W}_{k+1}$ by adding one of the blocking
        constraints to $\mathcal{W}_{k+1}$;
    else
      $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k$;
end (for)

Further Remarks on the Active-Set Method

- **Techniques for determining an initial feasible point**. We can derive a "Phase I" approach. In fact, no significant modifications are needed to generalize this method from linear programming to quadratic programming;
- **Updating Factorizations.** Since the working set can change by just one index at every iteration, the KKT matrix differs in at most one row and one column from the previous iteration's KKT matrix. Indeed, G remains fixed, whereas the matrix A of constraint gradients corresponding to the current working set may change through addition or deletion of a single row. It follows from this observation that we can compute the matrix factors needed to solve (18) at the current iteration by updating the factors computed at the previous iteration, rather than recomputing them from scratch. These updating techniques are crucial to the efficiency of active-set methods.
- **Finite Termination of Active-Set Algorithm on Strictly Convex QPs.**

Finite Termination of Active-Set Algorithm on Strictly Convex QPs

If the solution of (18) is $p_k = 0$, the current point x_k is the unique global minimizer of $q(\cdot)$ for the working set $\mathcal{W}_k$.

If it is not the solution of the original problem (4) (that is, at least one of the Lagrange multipliers is negative), The step p_{k+1} computed after a constraint is dropped will be a strict decrease direction for $q(\cdot)$.

If $\alpha_k > 0$, the value of q is lower than $q(x_k)$ at all subsequent iterations. It follows that the algorithm can never return to the working set $\mathcal{W}_k$, since subsequent iterates have values of q that are lower than the global minimizer for this working set.

Finite Termination of Active-Set Algorithm on Strictly Convex QPs

The algorithm encounters an iterate k for which $p_k = 0$ solves (18) at least on every nth iteration.

To show this claim, note that whenever we have an iteration for which $p_k \neq 0$,

- either $\alpha_k = 1$ (in which case we reach the minimizer of q on the current working set $\mathcal{W}_k$, so that the next iteration will yield $p_{k+1} = 0$),
- or else a constraint is added to the working set $\mathcal{W}_k$.

If the latter situation occurs repeatedly, then after at most n iterations the working set will contain n indices, which correspond to n linearly independent vectors. The solution of (18) will then be $p_k = 0$, since only the zero vector will satisfy the constraints.

Finite Termination of Active-Set Algorithm on Strictly Convex QPs

It is not difficult to show that the algorithm just mentioned converges for strictly convex QPs under certain assumptions, that is, it identifies the solution x^* in a finite number of iterations. This claim is certainly true if we assume that the method always takes a nonzero step length α_k whenever the direction p_k computed from (18) is nonzero. Our argument proceeds as follows:

- If the solution of (18) is $p_k = 0$, the current point x_k is the unique global minimizer of $q(\cdot)$ for the working set $\mathcal{W}_k$;
- The algorithm encounters an iterate k for which $p_k = 0$ solves (18) at least on every nth iteration.
- Taken together, the two statements above indicate that the algorithm finds the global minimum of q on its current working set periodically (at least once every n iterations) and that having done so, it never visits this particular working set again.

Interior-Point Methods

The primal-dual interior-point approach can be applied to convex quadratic programs through a simple extension of the linear-programming algorithms. The resulting algorithms are simple to describe, relatively easy to implement, and quite efficient on certain types of problems.
For simplicity, we restrict our attention to convex quadratic programs with inequality constraints, which we write as follows:

$$\min_x \quad q(x) \equiv \tfrac{1}{2} x^T G x + x^T c \tag{22a}$$

$$s.t. \qquad Ax \geq b, \tag{22b}$$

where G is symmetric and positive semidefinite, and where the $m \times n$ matrix A and righthand-side b are defined by

$$A = [a_i]_{i \in \mathcal{I}}, \qquad b = [b_i]_{i \in \mathcal{I}}, \qquad \mathcal{I} = \{1, 2, \cdots, m\}.$$

The KKT conditions for problem (22) can be written as

$$
\begin{aligned}
Gx - A^T\lambda + c &= 0, && && (23a)\\
Ax - y - b &= 0, && && (23b)\\
y_i\lambda_i &= 0, && i = 1, 2, \cdots, m, && (23c)\\
(y, \lambda) &\geq 0. && && (23d)
\end{aligned}
$$

Since we assume that G is positive semidefinite, these KKT conditions are not only necessary but also sufficient, so we can solve the convex program (22) by finding solutions of the system (23).

Interior-Point Methods

Given a current iterate (x, y, λ) that satisfies $(y, \lambda) > 0$, we can define a complementarity measure

$$\mu = \frac{y^T \lambda}{m}.$$

The path-following, primal-dual methods by considering the *perturbed* KKT conditions given by

$$F(x, y, \lambda, \sigma, \mu) = \begin{bmatrix} Gx - A^T\lambda + c \\ Ax - y - b \\ \mathcal{Y}\Lambda e - \sigma\mu e \end{bmatrix} = 0, \tag{24}$$

where

$$\mathcal{Y} = \mathrm{diag}(y_1, \cdots, y_m),\ \Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_m),\ e = (1, \cdots, 1)^T,$$

and $\sigma \in [0, 1]$. The solutions of (24) for all possible values of σ and μ define the *central path*, which is a trajectory that leads to the solution of the quadratic program as $\sigma\mu$ tends to zero.

By fixing μ and applying Newton's method to (24), we obtain the linear system

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & \mathcal{Y} \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle y \\ \triangle \lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -\Lambda \mathcal{Y} e + \sigma \mu e \end{bmatrix}, \tag{25}$$

where

$$r_d = Gx - A^T \lambda + c, \qquad r_p = Ax - y - b.$$

We obtain the next iterate by setting

$$(x^+, y^+, \lambda^+) = (x, y, \lambda) + \alpha(\triangle x, \triangle y, \triangle \lambda),$$

where α is chosen to retain the inequality $(y^+, \lambda^+) > 0$ and possibly to satisfy various other conditions.

The most popular interior-point method for convex QP is based on Mehrotra'a predictor-corrector algorithm, originally developed for LP. The extension to QP is straightforward .

First, we compute an affine scaling step $(\triangle x^{aff}, \triangle y^{aff}, \triangle \lambda^{aff})$ by setting $\sigma = 0$ in (25). We improve upon this step by computing a corrector step. Next, we compute the centering parameter σ. The total step is obtained by solving the following system

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & \mathcal{Y} \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle y \\ \triangle \lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -\Lambda \mathcal{Y} e - \triangle \Lambda^{aff} \triangle \mathcal{Y}^{aff} e + \sigma \mu e \end{bmatrix}, \quad (26)$$

Predictor-Correct Interior-Point Algorithm for QP

Compute (x_0, y_0, λ_0) with $(y_0, \lambda_0) > 0$;
for $k = 0, 1, 2, \cdots$
Set $(x, y, \lambda) = (x_k, y_k, \lambda_k)$ and solve (25) with $\sigma = 0$ for $(\triangle x^{aff}, \triangle y^{aff}, \triangle \lambda^{aff})$;
Calculate $\mu = y^T \lambda / m$;
Calculate $\hat{\alpha}_{aff} = \max\{\alpha \in (0, 1] | (y, \lambda) + \alpha(\triangle y^{aff}, \triangle \lambda^{aff}) \geq 0\}$;
Calculate $\mu_{aff} = (y + \hat{\alpha}^{aff} \triangle y^{aff})^T (\lambda + \hat{\alpha}^{aff} \triangle \lambda^{aff}) / m$;
Set centering parameter to $\sigma = (\mu_{aff} / \mu)^3$;
Solve (26) for $(\triangle x, \triangle y, \triangle \lambda)$;
Choose $\tau_k \in (0, 1)$ and set $\hat{\alpha} = \min(\alpha^{pri}_{\tau_k}, \alpha^{dual}_{\tau_k})$;
Set $(x_{k+1}, y_{k+1}, \lambda_{k+1}) = (x_k, y_k, \lambda_k) + \hat{\alpha}(\triangle x, \triangle y, \triangle \lambda)$
end(for)

The Gradient Projection Method

- In the classical active-set method, the active set and working set change slowly, usually by a single index at each iteration. As a result, this method may require many iterations to converge on large-scale problems.
- The gradient projection method is designed to make rapid changes to the active set. It is most efficient when the constraints are simple in form.

We will restrict our attention to the bound-constrained problem.

The Bound-constrained Problem

Consider the bound-constrained problem:

$$\min_x \quad q(x) = \tfrac{1}{2} x^T G x + x^T c \tag{27a}$$

$$s.t. \qquad l \leq x \leq u, \tag{27b}$$

where G is symmetric and l and u are vectors of lower and upper bounds on the components of x. The feasible region is sometimes called a "box" because of its rectangular shape. Some components of x may lack an upper or a lower bound; we handle these cases by setting the appropriate components of l and u to $-\infty$ and $+\infty$, respectively. We do not make any positive definiteness assumptions on G, since the gradient projection approach can be applied to both convex and nonconvex problems.

The Gradient Projection Method

Each iteration of the gradient projection algorithm consists of two stages.

- In the first stage, we search along the steepest descent direction from the current point x, that is, the direction $-g$, where $g = Gx + c$. When a bound is encountered, the search direction is "bent" so that it stays feasible. We search along this piecewise path and locate the first local minimizer of q, which we denote by x^c and refer to as the Cauchy point. The working set is now defined to be the set of bound constraints that are active at the Cauchy point, denoted by $\mathcal{A}(x^c)$.
- In the second stage of each gradient projection iteration, we "explore" the face of the feasible box on which the Cauchy point lies by solving a subproblem in which the active components x_i for $i \in \mathcal{A}(x^c)$ are fixed at the values x_i^c.

Cauchy Point Computation

We now derive an explicit expression for the piecewise linear path obtained by projecting the steepest descent direction onto the feasible box, and outline the search procedure for identifying the first local minimum of q along this path.

We define the projection of an arbitrary point x onto the feasible region (27b) as follows: The ith component is given by

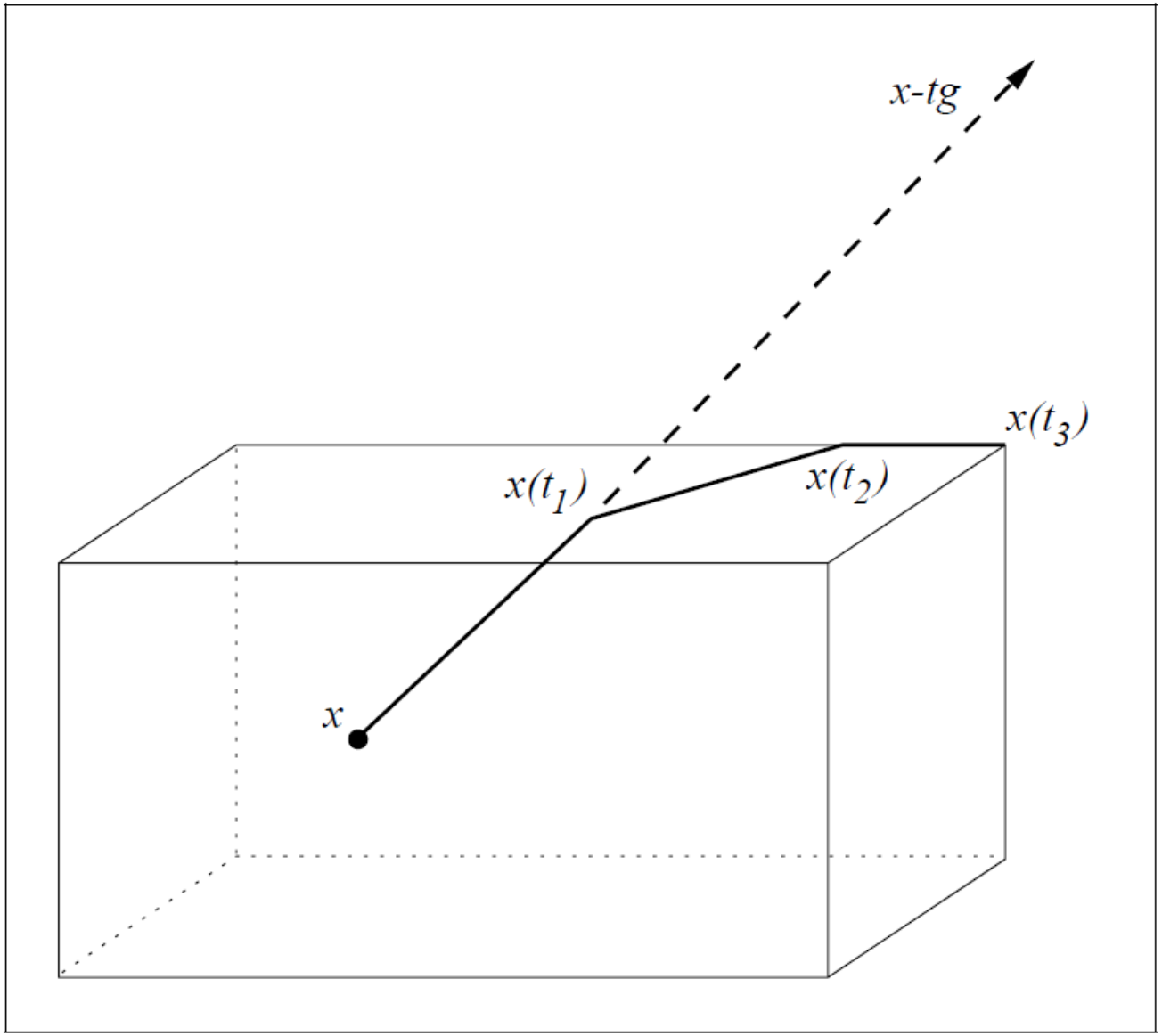
$$P(x, l, u)_i = \begin{cases} l_i & \text{if} \quad x_i < l_i, \\ x_i & \text{if} \quad x_i \in [l_i, u_i], \\ u_i & \text{if} \quad x_i > u_i. \end{cases} \tag{28}$$

The piecewise linear path $x(t)$ starting at the reference point x and obtained by projecting the steepest descent direction at x onto the feasible region (27b) is thus given by

$$x(t) = P(x - tg, l, u), \tag{29}$$

where $g = Gx + c$.

Cauchy Point Computation



The piecewise linear path $x(t)$, for an example in $\Re^3$.

Cauchy Point Computation

The Cauchy point x^c, which is defined as the first local minimizer of the univariate, piecewise quadratic $q(x(t))$, for $t \geq 0$. This minimizer is obtained by examining each of the line segments that make up $x(t)$. To perform this search, we need to determine the values of t at which the kinks in $x(t)$, or *breakpoints*, occur. We first identify the values of t for which each component reaches its bound along the chosen direction $-g$. These values $\bar{t}_i$ are given by the following explicit formulae:

$$\bar{t}_i = \begin{cases} (x_i - u_i)/g_i & \text{if} g_i < 0 \text{ and } u_i < +\infty, \\ (x_i - l_i)/g_i & \text{if} g_i > 0 \text{ and } l_i > -\infty, \\ \infty & \text{otherwise.} \end{cases} \tag{30}$$

The components of $x(t)$ for any t are therefore

$$x_i(t) = \begin{cases} x_i - tg_i & \text{if} t \leq \bar{t}_i, \\ x_i - \bar{t}_i g_i & \text{otherwise.} \end{cases} \tag{31}$$

This has a simple geometrical interpretation: A component $x_i(t)$ will move at the rate given by the projection of the gradient along this direction, and will remain fixed once this variable reaches one of its bounds.

Cauchy Point Computation

To search for the first local minimizer along $P(x - tg, l, u)$, we eliminate the duplicate values and zero values of $\bar{t}_i$ from the set $\{\bar{t}_1, \bar{t}_2, \cdots, \bar{t}_n\}$, to obtain a sorted, reduced set of breakpoints $\{t_1, t_2, \cdots, t_l\}$ with $0 < t_1 < t_2 < \cdots$. We now examine the intervals $[0, t_1]$, $[t_1, t_2]$, $[t_2, t_3]$, $\cdots$ in turn. Suppose we have examined the intervals up to t_{j-1} and have not yet found a local minimizer. For the interval $[t_{j-1}, t_j]$, we have that

$$x(t) = x(t_{j-1}) + (\triangle t)p^{j-1},$$

where

$$\triangle t = t - t_{j-1} \in [0, t_j - t_{j-1}],$$

and

$$p_i^{j-1} = \begin{cases} -g_i & \text{if } t_{j-1} < \bar{t}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{32}$$

Cauchy Point Computation

We can then write the quadratic (27a) on the line segment $[x(t_{j-1}), x(t_j)]$ as follows:

$$q(x(t)) = c^T(x(t_{j-1})+(\triangle t)p^{j-1})+\frac{1}{2}(x(t_{j-1})+(\triangle t)p^{j-1})^T G(x(t_{j-1})+(\triangle t)p^{j-1}).$$

By expanding and grouping the coefficients of 1, $\triangle t$, and $(\triangle t)^2$, we find that

$$q(x(t)) = f_{j-1} + f'_{j-1}\triangle t + \frac{1}{2}f''_{j-1}(\triangle t)^2, \qquad \triangle t = t - t_{j-1} \in [0, t_j - t_{j-1}] \quad (33)$$

where the coefficients f_{j-1}, f'_{j-1} and f''_j are defined by

$$\begin{aligned} f_{j-1} &= c^T x(t_{j-1}) + \frac{1}{2}x(t_{j-1})^T G x(t_{j-1}), \\ f'_{j-1} &= c^T p^{j-1} + x(t_{j-1})^T G p^{j-1}, \\ f''_{j-1} &= (p^{j-1})^T G p^{j-1}. \end{aligned}$$

Cauchy Point Computation

By differentiating (33) with respect to $\triangle t$ and equating to zero, we obtain $\triangle t^* = -f'_{j-1}/f''_{j-1}$. The following cases can occur.
(i) If $f''_{j-1} > 0$ and $\triangle t^* \in [0, t_j - t_{j-1})$ there is a minimizer at $t = t_{j-1} + \triangle t^*$.
(ii) If $f''_{j-1} \geq 0$ and $f'_{j-1} \geq 0$ there is a local minimizer of $q(x(t))$ at $t = t_{j-1}$;
(iii) In all other cases we move on to the next interval $[t_j, t_{j+1}]$ and continue the search.

For the next search interval, we need to calculate the new direction p^j from (32), and we use this new value to calculate f_j, f'_j, and f''_j. Since p^j differs from p^{j-1} typically in just one component, computational savings can sometimes be made by updating these coefficients rather than computing them from scratch.

Subspace Minimization

After the Cauchy point x^c has been computed, the components of x^c that are at their lower or upper bounds define the active set

$$\mathcal{A}(x^c) = \{i | x_i^c = l_i \text{ or } x_i^c = u_i\}.$$

In the second stage of the gradient projection iteration, we approximately solve the QP obtained by fixing these components at the values x_i for $i \in \mathcal{A}(x^c)$ at the value x_i^c. The remaining components are determined from the subproblem

$$\min_x \quad q(x) = \tfrac{1}{2} x^T G x + x^T c \tag{34a}$$

$$s.t. \qquad x_i = x_i^c, \qquad i \in \mathcal{A}(x^c), \tag{34b}$$

$$l_i \leq x_i \leq u_i, \qquad i \notin \mathcal{A}(x^c). \tag{34c}$$

Subspace Minimization

It is not necessary to solve problem (34) exactly.

- Nor is it desirable, since the subproblem maybe almost as difficult as the original problem (27).
- In fact, to obtain global convergence all we require of the approximate solution x^+ of (34) is that $q(x^+) \leq q(x^c)$ and that x^+ be feasible with respect to the constraints (27).

A strategy that is intermediate between choosing $x^+ = x^c$ as the approximate solution (on the one hand) and solving (34) exactly (on the other hand) is to compute an approximate solution of (34) by using the conjugate gradient iteration.

Gradient-Projection Method for QP

Compute a feasible starting point x^0;
for $k = 0, 1, 2, \cdots$
    if x^k satisfies the KKT conditions for (27)
        STOP with solution $x^* = x^k$;
    Set $x = x^k$ and find the Cauchy point x^c;
    Find an approximate solution x^+ of (34) such that $q(x^+) \leq q(x^c)$
        and x^+ is feasible;
    $x^{k+1} \leftarrow x^+$;
end(for)

Subspace Minimization

- When applied to problems that satisfy strict complementarity (that is, problems for which all Lagrange multipliers associated with an active bound at the optimizer x^* are nonzero), the active sets $\mathcal{A}(x^c)$ generated by the gradient projection algorithm eventually settle down. That is, constraint indices cannot repeatedly enter and leave the active set on successive iterations.
- When the problem is degenerate, the active set may not settle down, but various devices have been proposed to prevent this undesirable behavior from taking place.

Extension to QP with general linear constraints

While gradient projection methods can be applied in principle to problems with general linear constraints, significant computation may be required to perform the projection onto the feasible set in such case.

For example, if the constraint set is defined as $a_i^T x \geq b_i$, $i \in \mathcal{I}$, we must solve the following convex quadratic program to compute the projection of a given point $\hat{x}$ onto the feasible set:

$$\min_x \|x - \hat{x}\|^2, \text{ s.t. } a_i^T x \geq b_i, i \in \mathcal{I}.$$

The expense of solving this "projection subproblem" may approach the cost of solving the original quadratic program, so it is usually not economical to apply gradient projection to this case.

Extension to QP with general linear constraints

When we use duality to replace a strictly convex quadratic program with its dual, the gradient projection method may be useful in solving the bound-constrained dual problem. which is formulated in terms of the Lagrangian multiplier λ as follows:

$$\max_x \hat{q}(\lambda) = -\frac{1}{2}(A^T\lambda - c)^T G^{-1}(A^T\lambda - c)^T + b^T\lambda, \text{ s.t. } \lambda \geq 0.$$

This approach is most useful when G has a simple form, for example, a diagonal or block-diagonal matrix.

Nonlinear Gradient Projection

It is not difficult to extend the gradient projection method to the problem

$$\min f(x) \qquad \text{s.t.} l \leq x \leq u,$$

where f is a nonlinear function and l and u are vectors of lower and upper bounds, respectively.

At the current iterate x_k, we form the quadratic model

$$q_k(x) = f_k + \nabla f_k^T (x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k),$$

where B_k is a approximation to $\nabla^2 f(x_k)$.

Nonlinear Gradient Projection

In line search methods, we require B_k is a positive definite approximation to $\nabla^2 f(x_k)$.

We then use the gradient projection method for QP to find an approximate solution $\hat{x}$ of the subproblem

$$\min q_k(x) \qquad \text{s.t.} l \leq x \leq u.$$

The search direction is defined as $p_k = x_k - \hat{x}$ and the new iterate is given by $x_{k+1} = x_k + \alpha_k p_k$, where the steplength α_k is chosen to satisfy

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \eta \alpha_k \nabla f_k^T p_k$$

for some parameter $\eta \in (0, 1)$.

If the projected gradient is nonzero, since B_k is assumed to be positive definite, the inequality

$$f_k = q_k(x_k) > q_k(x^c) \geq q_k(\hat{x}) = f_k + \nabla f_k^T p_k + \frac{1}{2} p_k^T B_k p_k$$

implies that $\nabla f_k^T p_k < 0$ and the search direction p_k is indeed a descent direction for the objective function f.

Nonlinear Gradient Projection

In trust region methods, there is no requirement for q_k to be convex, we can define B_k to be the Hessian $\nabla^2 f(x_k)$ or a quasi-Newton approximation obtained from the BFGS or SR1 formulas.

The step p_k is obtained by solving the subproblem

$$\min q_k(x) \qquad \text{s.t. } l \leq x \leq u, \qquad \|x - x_k\|_\infty \leq \Delta_k,$$

for some $\Delta_k > 0$. This problem can be posed as a bound-constrained quadratic program as follows:

$$\min q_k(x) \qquad \text{s.t. } \max(l, x_k - \Delta_k e) \leq x \leq \min(u, x_k + \Delta_k e),$$

where $e = (1, \cdots, 1)^T$. The gradient projection method for QP can be used to solve this subproblem.

The step p_k is accepted or rejected following standard trust-region strategies, and the radius Δ_k is updated according to the agreement between the change in f and the change in q_k produced by the step p_k.

Outline

Perspective and Software

Practical numerical comparisons show that

- The interior-point methods are generally much faster on large problems;
- If a warm start is required, active-set methods are generally preferable;
- Practical projected gradient methods are currently only available for several kinds of QPs whose feasible regions have special structure.

Perspective and Software

- Active-set methods for convex QP: QPOPT, VE09, BQPD, QPA ...
- Interior point method:
  - Commercial Solvers: CPLEX, XPRESS-MP and MOSEK.
  - Open Source Package: OOPS and OOQP(objective-oriented codes that allow the user to customize the linear algebra techniques to the particular structure of the data for an application).
  - Some nonlinear programming interior-point packages, such as LOQO, and KNITRO, are also effective for convex and nonconvex QP.

Thanks for your attention!