Quadratic Programming

November 2024

1 Introduction

1.1 QP problem

Quadratic Programming (QP) is a type of optimization problem where the objective function is quadratic, and the constraints are linear. It can be formulated as follows:

$$\begin{aligned} \min_{x} \quad & \varphi(x), \quad \text{where} \quad \varphi(x) := g^T x + \frac{1}{2} x^T H x \\ s.t. \quad & Ax + b \in \mathcal{C} \end{aligned}$$

where $x \in \mathbb{R}^n$ is the decision variable, $g \in \mathbb{R}^n$ is a vector, $H \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $A \in \mathbb{R}^{m \times n}$ is a matrix, $b \in \mathbb{R}^m$ is a vector, $\mathcal{C} \subseteq \mathbb{R}^m$ is a nonempty, closed set.

In simple terms, the constraint $Ax + b \in \mathcal{C}$ includes both the equality constraints ($a_i^T x + b_i = 0$) and inequality constraints ($a_i^T x + b_i \leq 0$). Hence the problem can also be written as follows:

$$\begin{aligned} \min_{x} \quad & g^T x + \frac{1}{2} x^T H x \\ s.t. \quad & a_i^T x + b_i = 0, \ \forall i \in \mathcal{I}_1 := \{1, 2, \cdots, l\} \\ & a_i^T x + b_i \leq 0, \ \forall i \in \mathcal{I}_2 := \{l+1, \cdots, m\} \end{aligned}$$

where for any $i \in \mathcal{I} := \mathcal{I}_1 \cap \mathcal{I}_2$, $a_i^T \in \mathbb{R}^n$ is the i-th row of A and $b_i \in \mathbb{R}$ is the i-th element of b. Specifically, we will refer to it as convex QP if the matrix H is positive semi-definite.

1.2 Exact penalty subproblem

The central focus of the following algorithms is the numerical solution of exact penalty subproblems, which we define to be any problem of the form:

$$\min_{x \in \mathcal{X}} \quad J(x), \quad \text{where} \quad J(x) := g^T x + \frac{1}{2} x^T H x + \sum_{i \in \mathcal{I}_1} |a_i^T x + b_i| + \sum_{i \in \mathcal{I}_2} \max\{a_i^T x + b_i, 0\} \tag{1}$$

2 Algorithms

2.1 Iterative reweighting algorithm (IRWA)

We now describe an iterative algorithm for minimizing the function J in (1), where in each iteration one solves a subproblem whose objective is the sum of φ and a weighted linear least-squares term.

Here, we define our local approximation to J at a given point $\tilde{x}$ and with a given *relaxation vector* $\epsilon \in \mathbb{R}^m_{++}$ by

$$\begin{aligned} \hat{G}_{(\tilde{x},\epsilon)}(x) &:= g^T x + \frac{1}{2} x^T H x + \frac{1}{2} \left(\sum_{i \in \mathcal{I}_1} w_i(\tilde{x}, \epsilon) |a_i^T x + b_i|^2 + \sum_{i \in \mathcal{I}_2} w_i(\tilde{x}, \epsilon) (a_i^T x + b_i - \min\{a_i^T \tilde{x} + b_i, 0\})^2 \right) \\ &= \frac{1}{2} x^T (H + A^T W A) x + (g^T + v^T W A) x + \frac{1}{2} v^T W v \end{aligned}$$

where

$$\begin{aligned} W \in \mathbb{R}^{m\times m} \quad &:= \operatorname{diag}(w_1(\tilde{x},\epsilon), \cdots, w_m(\tilde{x},\epsilon)) \\ v \in \mathbb{R}^m \quad &:= \begin{bmatrix} b_1 & \cdots & b_l & \max\{-a_{l+1}\tilde{x},\ b_{l+1}\} & \cdots & \max\{-a_m\tilde{x},\ b_m\} \end{bmatrix}^T \end{aligned}$$

and for any $x \in \mathbb{R}^n$, we define

$$w_i(x,\epsilon) := \begin{cases} \left(|a_i^T x + b_i|^2 + \epsilon_i^2\right)^{-1/2} & i \in \mathcal{I}_1 \\ \left(\max\{(a_i^T x + b_i), 0\}^2 + \epsilon_i^2\right)^{-1/2} & i \in \mathcal{I}_2 \end{cases}$$

We now state the algorithm.

Step 0. (Initialization) Choose an initial point $x^{(0)} \in \mathcal{X}$, an initial relaxation vector $\epsilon^{(0)} \in \mathbb{R}^l_{++}$ and scaling parameters $\eta \in (0,1), \gamma > 0$ and $M > 0$. Let $\sigma \geq 0$ and $\sigma' \geq 0$ be two scalars which serve as termination tolerances for step-size and relaxation parameter, respectively. Set $k := 0$.

Step 1. (Solve the reweighted subproblem for $x^{(k+1)}$) Compute a solution $x^{(k+1)}$ to the problem

$$\mathcal{G}(x^{(k)}, \epsilon^{(k)}): \quad \min_{x\in\mathcal{X}} \ \hat{G}_{(x^{(k)},\epsilon^{(k)})}(x)$$

i.e. Solve the linear system

$$(H + A^T W A)x + (g + A^T W v) = 0$$

Step 2. (Set the new relaxation vector $\epsilon^{(k+1)}$) Set

$$q_i^{(k)} := a_i^T(x^{(k+1)} - x^{(k)}) \quad \text{and} \quad r_i^{(k)} := (1 - v_i)(a_i^T x^{(k)} + b_i) \quad \forall i \in \mathcal{I}$$

If

$$|q_i^{(k)}| \leq M \left[|r_i^{(k)}|^2 + (\epsilon_i^{(k)})^2\right]^{\frac{1}{2}+\gamma} \quad \forall i \in \mathcal{I}$$

Then choose $\epsilon^{(k+1)} \in (0, \eta\epsilon^{(k)}]$, else set $\epsilon^{(k+1)} := \epsilon^{(k)}$

Step 3. (Check stopping criteria) If $\|x^{(k+1)} - x^{(k)}\|_2 \leq \sigma$ and $\|\epsilon^{(k)}\|_2 \leq \sigma'$, then stop; else, set $k := k+1$ and go back to Step 1.

2.2 Alternating direction augmented Lagrangian (ADAL)

Defining

$$\hat{J}(x,p) := \varphi(x) + \sum_{i\in\mathcal{I}_1} |p_i| + \sum_{i\in\mathcal{I}_2} \max\{p_i, 0\},$$

the problem (1) has the equivalent form

$$\begin{aligned} \min_{x\in\mathcal{X},p} \quad & \hat{J}(x,p) \\ \text{subject to} \quad & Ax + b = p \end{aligned}$$

where $p = (p_1, \cdots, p_m)^T$. In particular, the function in (1) can be expressed as $J(x) = \hat{J}(x, Ax+b)$. Defining dual variables $u = (u_1, \cdots, u_m)^T$ and a penalty parameter $\mu > 0$, an augmented Lagrangian is given by

$$L(x,p,u) = g^T x + \frac{1}{2}x^T H x + \sum_{i\in\mathcal{I}_1} |p_i| + \sum_{i\in\mathcal{I}_2} \max\{p_i, 0\} + u^T(Ax + b - p) + \frac{\mu}{2}\|Ax + b - p\|_2^2$$

We now state the algorithm.

Step 0. (Initialization) Choose an initial point $x^{(0)}$, dual vector $u_i^{(0)} \in \mathbb{R}^{m_i}$ for $i \in \mathcal{I}$, and penalty parameter $\mu > 0$. Let $\sigma \geq 0$ and $\sigma'' \geq 0$ be two scalars which serve as termination tolerances for step-size and constrained residual, respectively. Set $k := 0$.

Step 1. (Solve the augmented Lagrangian subproblems for $(x^{(k+1)}, p^{(k+1)})$) Compute a solution $x^{(k+1)}$ to the problem

$$\min_x \quad L(x, p^{(k)}, u^{(k)})$$

and a solution $p^{(k+1)}$ to the problem

$$\min_p \quad L(x^{(k+1)}, p, u^{(k)})$$

Step 2. (Set the new multipliers $u^{(k+1)}$) Set

$$u^{(k+1)} := u^{(k)} + \frac{1}{\mu}\left(Ax^{(k+1)} + b - p^{(k+1)}\right)$$

Step 3. (Check stopping criteria) If $\|x^{(k+1)} - x^{(k)}\|_2 \leq \sigma$ and $\sup_{i\in\mathcal{I}}\left\{|a_i x^{(k+1)} + b_i - p_i^{(k+1)}|\right\} \leq \sigma''$, then stop; else, set $k := k+1$ and go back to Step 1.