Quadratic Programming: Gradient projection and Interior point methods

Oscar S. Dalmau Cedeño
dalmau@cimat.mx

23 de octubre de 2018

Quadratic Programming with box constraints:

$$
\begin{aligned}
\min q(\boldsymbol{x}) &= \frac{1}{2}\boldsymbol{x}^T\mathbf{G}\boldsymbol{x} - \boldsymbol{c}^T\boldsymbol{x} \quad (1)\\
s.a.:\ l \leq &\ \boldsymbol{x} \leq u \quad (2)
\end{aligned}
$$

where $\mathbf{G} \in \mathbb{R}^{n\times n}$ is symmetric and $\boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^n$.

The *gradient projection algorithm* is a two stage iterative method

First stage: (Cauchy Point)

- Use the steepest descent direction $-\mathrm{g}$ from the current point $\boldsymbol{x}$, If a bound is found, the search direction changes is in order to maintain feasibility.
- We search along the resulting piecewise-linear path and locate the first local minimizer $\boldsymbol{x}^c$ (Cauchy point) of $q(\cdot)$.
- The *working set* $\mathcal{A}(\boldsymbol{x}^c)$ is defined as the set of bound constraints that are active at $\boldsymbol{x}_c$.

Second stage: (Subspace minimization) In each iteration, we solve a subproblem subject to the active components, ie, we fix $x_i = x_i^c$, $i \in \mathcal{A}(\boldsymbol{x}^c)$ and solve for the remainder variables.

Cauchy Point

- First, we build a feasible piecewise-linear path.
- Then, we minimize $q(\cdot)$ along this path

Cauchy Point: piecewise-linear path

- We can generate the piecewise-linear path by projecting the steepest descent direction onto the feasible box
- Projection operator onto the feasible region. The component i is defined as follows:

$$P(\boldsymbol{x}; l, u)_i = \begin{cases} l_i & \text{if } x_i < l_i \\ x_i & \text{if } l_i \leq x_i \leq u_i \\ u_i & \text{if } x_i > u_i \end{cases}$$

where $l \prec u$.

Cauchy Point: piecewise-linear path

The piecewise-linear path $x(t)$ of $\boldsymbol{x} - t\mathbf{g}$ for $t \geq 0$, ie, the ray that start $\boldsymbol{x}$ in the direction $-\mathbf{g}$, is

$$x(t) \;\; = \;\; P(\boldsymbol{x} - t\mathbf{g}; l, u), \; t \geq 0$$

where $l \prec u$ and $\mathbf{g} = \mathbf{G}\boldsymbol{x} - \boldsymbol{c}$

Cauchy Point

The Cauchy point $\boldsymbol{x}^c$ is defined as the first minimizer of $q()$ along the piecewise-linear path $x(t)$, ie, $\boldsymbol{x}^c := x(t^*)$ where

$$t^* = \min\{\arg\min_{t\geq 0} q(x(t))\} \tag{3}$$

Cauchy Point: piecewise-linear path

- The minimizer is obtained by analyzing the sequence of line segments of $x(t)$
- Then, we need to compute the sequence of breakpoints $0 < t_1 < t_2 < \cdots$

Piecewise-linear path: breakpoints

- The general idea is to find the bounds $\bar{t}_i$ for each component along the direction $-\mathbf{g}$, then the duplicates are deleted and the remaining values are sorted.

Piecewise-linear path: breakpoints

Bounds $\bar{t}_i$ for each component: from $x_i - tg_i \in [l_i, u_i]$, ie,

$$l_i \quad \leq x_i - tg_i \leq u_i$$

note that $r_i(t) = x_i - tg_i$, $t \geq 0$ is a ray with slope $-g_i$

1. If $-g_i > 0$ the ray is increasing and we obtain an upper bound $x_i - \bar{t}_i g_i = u_i$, if $u_i < \infty$; ie, $\bar{t}_i = \frac{x_i - u_i}{g_i}$
2. If $-g_i < 0$ the ray is decreasing and we obtain a lower bound $x_i - \bar{t}_i g_i = l_i$, if $l_i > -\infty$; ie, $\bar{t}_i = \frac{x_i - l_i}{g_i}$
3. If ($-g_i > 0$ and $u_i = \infty$) or ($-g_i < 0$ and $l_i = -\infty$) then $\bar{t}_i = \infty$
4. If $g_i = 0$ then $\bar{t}_i$ can take any value, in particular $\bar{t}_i = \infty$ to indicate that there is not breakpoint

Piecewise-linear path: breakpoints

$$\bar{t}_i = \begin{cases} \frac{x_i - u_i}{g_i} & \text{if } -g_i > 0 \text{ and } u_i < \infty \\ \frac{x_i - l_i}{g_i} & \text{if } -g_i < 0 \text{ and } l_i > -\infty \\ \infty & \text{otherwise} \end{cases}$$

To obtain the breakpoints, we remove duplicates and zeros from the set $\{\bar{t}_1, \bar{t}_2, \cdots\}$, finally the reduced set is sorted and we obtain the set of breakpoints $\{t_1, t_2, \cdots\}$ with

$$0 < t_1 < t_2 < \cdots$$

Cauchy Point computation

For computing the Cauchy Point, we analyze the sequence of intervals $[0, t_1]$, $[t_1, t_2]$, $[t_2, t_3]$, $\cdots$ and compute the first minimizer of $q(x(t))$ for $t \in [t_{j-1}, t_j]$, $j = 0, 1, 2, 3, \cdots$

Cauchy Point computation

- Suppose we have analyzed up to the breakpoint t_{j-1} and have not yet found a local minimizer (at the beginning $t_{-1} = 0$)
- Then, the next step is to analyze the interval $[t_{j-1}, t_j]$.

Cauchy Point computation

- As $t \in [t_{j-1}, t_j]$ then $x(t) \in [x(t_{j-1}), x(t_j)]$.
- The previous segment line can reparametrized, ie, subtracting side by side t_{j-1}, $\tau \stackrel{def}{=} t - t_{j-1} \in [t_{j-1} - t_{j-1}, t_j - t_{j-1}]$ in order to simplify notation as follows:

$$x(\tau) \quad = \quad x(t_{j-1}) + \tau p^{j-1} \tag{4}$$

where $\tau := t - t_{j-1} \in [0, t_j - t_{j-1}]$ and

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

The first condition corresponds to the feasible region, otherwise the path is on the face of the feasible box.

Cauchy Point computation

We can write the function $q(\cdot)$ on the segment $[x(t_{j-1}), x(t_j)]$ by substituting $x(\tau) = x(t_{j-1}) + \tau p^{j-1}$, ie

$$
\begin{aligned}
q(x(\tau)) &= \frac{1}{2}\alpha_{j-1}\tau^2 + \beta_{j-1}\tau + \gamma_{j-1}; \tau \in [0, t_j - t_{j-1}] \\
\gamma_{j-1} &= \frac{1}{2}x(t_{j-1})^T \mathbf{G} x(t_{j-1}) - \boldsymbol{c}^T x(t_{j-1}) \\
\beta_{j-1} &= x(t_{j-1})^T \mathbf{G} p^{j-1} - \boldsymbol{c}^T p^{j-1} \\
\alpha_{j-1} &= (p^{j-1})^T \mathbf{G} p^{j-1}
\end{aligned}
$$

Cauchy Point computation

Then $\tau^* = -\frac{b}{a}$ and we can consider three cases:

a) If $\tau^* < 0$ then there is a minimizer at $\tau^* = 0$, ie $t = t_{j-1}$

b) If $\tau^* \in [0, t_j - t_{j-1})$ then there is a minimizer at $t = t_{j-1} + \tau^*$

c) If $\tau^* \geq t_j - t_{j-1}$ then we analyze the next interval $t \in [t_j, t_{j+1}]$

Subspace minimization

After computing the Cauchy point $\boldsymbol{x}^c$ we compute the active set, ie, the components $\boldsymbol{x}^c$ equal to the lower or upper bounds

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

In the second stage of the algorithm, we approximately solve the QP problem obtained by setting $x_i = x_i^c$ for $i \in \mathcal{A}(\boldsymbol{x}^c)$

Subspace minimization

Then we approximately solve

$$
\begin{aligned}
\text{min } q(\boldsymbol{x}) &= \frac{1}{2}\boldsymbol{x}^T\mathbf{G}\boldsymbol{x} - \boldsymbol{c}^T\boldsymbol{x} \\
\text{s.t:} \quad x_i &= x_i^c,\ i \in \mathcal{A}(\boldsymbol{x}^c) \\
l_i \leq x_i &\leq u_i,\ i \notin \mathcal{A}(\boldsymbol{x}^c)
\end{aligned}
$$

Subspace minimization

- Since the previous subproblem may be as difficult as the original, for example, in large scale problems, we do not solve this problem completely.
- In order to obtain global convergence of the gradient projection procedure, we require only to find an approximate solution $\boldsymbol{x}^+$ such that $q(\boldsymbol{x}^+) \leq q(\boldsymbol{x}^c)$, ie, the objective function value is not worse than that obtained at $\boldsymbol{x}^c$
- We can simply use $\boldsymbol{x}^+ = \boldsymbol{x}^c$ or we can apply conjugate gradient to the problem and terminate as soon as a bound $l \leq \boldsymbol{x} \leq u$ is encountered

Gradient Projection Method for QP

Compute $\boldsymbol{x}_0$
for $k = 0, 1, 2, \cdots$ **do**
  if $\boldsymbol{x}^k$ satisfies the KKT conditions of the original problem **then**
    Stop $\boldsymbol{x}^* = \boldsymbol{x}^k$
  end if
  Set $\boldsymbol{x} = \boldsymbol{x}^k$ and compute the Cauchy point $\boldsymbol{x}^c$
  Find an approximate solution $\boldsymbol{x}^+$ of the problem such that $q(\boldsymbol{x}^+) \leq q(\boldsymbol{x}^c)$ and $\boldsymbol{x}^+$ is feasible
  $\boldsymbol{x}^{k+1} = \boldsymbol{x}^+$
end for

Métodos de Punto Interior para QP

- Consideremos el siguiente problema de programación cuadrática :

$$\begin{aligned} \min q(\boldsymbol{x}) &= \frac{1}{2}\boldsymbol{x}^T\mathbf{G}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} && (5) \\ s.a.:\ A\boldsymbol{x} &\succeq \boldsymbol{b}, && (6) \end{aligned}$$

donde $\mathbf{G} \in \mathbb{R}^{n\times n}$ es una matriz simétrica semi positiva definida (caso convexo)

Las condiciones de KKT para $(\boldsymbol{x}, \boldsymbol{\lambda})$ son

$$
\begin{aligned}
\nabla \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) &= \mathbf{G}\boldsymbol{x} + \boldsymbol{c} - A^T \boldsymbol{\lambda} = 0 && (7)\\
A\boldsymbol{x} - \boldsymbol{b} &\succeq 0 && (8)\\
\lambda_i (A\boldsymbol{x} - \boldsymbol{b})_i &= 0 && (9)\\
\boldsymbol{\lambda} &\succeq 0 && (10)
\end{aligned}
$$

Añadiendo variables de holgura $\boldsymbol{y}$, las KKT's se reescriben

$$
\begin{aligned}
\mathbf{G}\boldsymbol{x} + \boldsymbol{c} - A^T\boldsymbol{\lambda} &= 0 \quad (11)\\
A\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{b} &= 0 \quad (12)\\
\lambda_i y_i &= 0 \quad (13)\\
(\boldsymbol{\lambda}, \boldsymbol{y}) &\succeq 0 \quad (14)
\end{aligned}
$$

Método Primal-Dual

O simplemente usemos la siguiente notación

$$F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}) = \begin{bmatrix} \mathbf{G}\boldsymbol{x} - A^T\boldsymbol{\lambda} + \boldsymbol{c} \\ A\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{b} \\ Y\Lambda e \end{bmatrix} = 0 \tag{15}$$

$$(\boldsymbol{\lambda}, \boldsymbol{y}) \succeq 0 \tag{16}$$

donde $F(x, \lambda, s) : \mathbb{R}^{(n+2m)} \to \mathbb{R}^{(n+2m)}$ y

$$Y = diag\{y_1, y_2, \ldots, y_m\} \tag{17}$$

$$\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_m\} \tag{18}$$

$$e = [1, 1, \ldots, 1]^T \tag{19}$$

que podemos resolver usando Newton, seleccionando el tamaño de paso de modo que $(\boldsymbol{\lambda}, \boldsymbol{y}) \succeq 0$

Version Central Path Following

O podemos usar la version *perturbada* del camino central

$$F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\lambda}; \sigma\mu) = \begin{bmatrix} \mathbf{G}\boldsymbol{x} - A^T\boldsymbol{\lambda} + \boldsymbol{c} \\ A\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{b} \\ Y\Lambda e - \sigma\mu \end{bmatrix} = 0 \tag{20}$$

donde $\mu = \frac{\boldsymbol{y}^T\boldsymbol{\lambda}}{m}$ y $\sigma \in [0, 1]$.

Podemos resolver usando el método de Newton.

Ver la versión del Central Path Following para programación Lineal.

Algoritmo Predictor-Corrector para QP

Otra forma es usando la estrategia Predictor-Corrector para QP

Paso Predictor: Primero se calcula el paso afin y el tamaño de paso que garantice factibilidad, luego se calculan μ, μ_{aff} y el parámetro de centralidad σ.

Paso Corrector: Se calcula el paso corrector, el tamaño de paso que garantice factibilidad y se actualiza la próxima iteración.

Paso Predictor

Se calcula el paso predictor $(\Delta x^{aff}, \Delta y^{aff}, \Delta \lambda^{aff})$ considerando $\sigma = 0$. Para lo cual se resuelve el sistema

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta x^{aff} \\ \Delta y^{aff} \\ \Delta \lambda^{aff} \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -Y\Lambda e \end{bmatrix} \tag{21}$$

con $r_c = \mathbf{G}\boldsymbol{x} - A^T\boldsymbol{\lambda} + \boldsymbol{c}$ y $r_b = A\boldsymbol{x} - \boldsymbol{y} - \boldsymbol{b}$

Tamaño de paso: Paso Predictor

Luego se determina a calidad de dicha dirección, a través de α_{aff}^{pri} y α_{aff}^{dua}

$$\alpha_{aff}^{pri} = \text{mín}\left(1, \min_{i:\Delta y_i^{aff}<0} -\frac{y_i}{\Delta y_i^{aff}}\right) \tag{22}$$

$$\alpha_{aff}^{dua} = \text{mín}\left(1, \min_{i:\Delta \lambda_i^{aff}<0} -\frac{\lambda_i}{\Delta \lambda_i^{aff}}\right) \tag{23}$$

Es decir, se calculan los maximos tamaños de pasos permitidos a lo largo de la dirección affine scaling.

Paso Predictor

Luego se calcula la *medida de dualidad* μ_{aff} (affine duality measure), que es el paso que lleva a la frontera

$$\mu_{aff} = (y + \alpha_{aff}^{pri} \Delta y^{aff})^T (\boldsymbol{\lambda} + \alpha_{aff}^{dua} \Delta \lambda^{aff})/m \tag{24}$$

y el *parametro de centrado* (centering parameter)

$$\sigma = (\frac{\mu_{aff}}{\mu})^3 \tag{25}$$

Tamaño de paso: Paso Predictor

O simplemente se puede deteminar un solo tamaño de paso α_{aff}

$$\alpha_{aff} \quad = \quad \text{máx}\{\alpha \in (0,1] | (\boldsymbol{y}, \boldsymbol{\lambda}) + \alpha(\Delta y^{aff}, \Delta\lambda^{aff}) \succeq 0\} \tag{26}$$

y en este caso la *medida de dualidad* es

$$\mu_{aff} = (y + \alpha_{aff}\Delta y^{aff})^T(\boldsymbol{\lambda} + \alpha_{aff}\Delta\lambda^{aff})/m \tag{27}$$

y el *parametro de centrado*

$$\sigma = (\frac{\mu_{aff}}{\mu})^3 \tag{28}$$

Paso Corrector

Para el paso corrector se resuelve el sistema de ecuaciones, ie se calcula la direccion $(\Delta x, \Delta y, \Delta\lambda)$

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -Y\Lambda e - \Delta Y^{aff} \Delta\Lambda^{aff} e + \sigma\mu e \end{bmatrix}$$

Tamaño de paso: Paso Correcto

Deteminar un solo tamaño de paso $\hat{\alpha}$

$$\hat{\alpha} = \max\{\alpha \in (0,1] | (\boldsymbol{y}, \boldsymbol{\lambda}) + \alpha(\Delta y, \Delta\lambda) \succeq 0\} \tag{29}$$

Otra forma $\hat{\alpha} = \min(\alpha_\tau^{prim}, \alpha_\tau^{dual})$, donde

$$\alpha_\tau^{prim} = \max\{\alpha \in (0,1] | \boldsymbol{y} + \alpha\Delta y \succeq (1-\tau)\boldsymbol{y}\} \tag{30}$$
$$\alpha_\tau^{dual} = \max\{\alpha \in (0,1] | \boldsymbol{\lambda} + \alpha\Delta\lambda \succeq (1-\tau)\boldsymbol{\lambda}\} \tag{31}$$

Algoritmo Predictor-Corrector para QP

Calcular $(\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{\lambda}_0)$ con $(\boldsymbol{y}_0, \boldsymbol{\lambda}_0) \succ 0$
for $k = 0, 1, 2, \cdots$ **do**
  Calcular el paso predictor o afin $(\Delta x^{aff}, \Delta y^{aff}, \Delta \lambda^{aff})$
  Calcular $\mu = \boldsymbol{y}^T \boldsymbol{\lambda} / m$
  Calcular el tamaño de paso predictor
  $\alpha_{aff} = \text{máx}\{\alpha \in (0, 1] | (\boldsymbol{y}, \boldsymbol{\lambda}) + \alpha(\Delta y^{aff}, \Delta \lambda^{aff}) \succeq 0\}$
  Calcular $\mu_{aff} = (\boldsymbol{y} + \alpha_{aff} \Delta y^{aff})^T (\boldsymbol{\lambda} + \alpha_{aff} \Delta \lambda^{aff}) / m$
  Calcular el parametro de centrado $\sigma = (\frac{\mu_{aff}}{\mu})^3$
  Calcular el paso corrector $(\Delta x, \Delta y, \Delta \lambda)$
  Calcular el tamaño de paso corrector $\hat{\alpha}$
  Actualiza $(\boldsymbol{x}_{k+1}, \boldsymbol{y}_{k+1}, \boldsymbol{\lambda}_{k+1}) = (\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{\lambda}_k) + \hat{\alpha}(\Delta x, \Delta y, \Delta \lambda)$
end for

Comentarios

- Punto Inicial: Se puede usar cualquiera de las estrategias usadas en Programación lineal
- Criterio de Paro

$$
\begin{aligned}
\|\mathbf{G}\boldsymbol{x}_{k+1} + \boldsymbol{c} - A^T\boldsymbol{\lambda}_{k+1}\|_2^2 &\leq \epsilon && (32)\\
\|A\boldsymbol{x}_{k+1} - \boldsymbol{y}_{k+1} - \boldsymbol{b}\|_2^2 &\leq \epsilon && (33)\\
\boldsymbol{y}_{k+1}^T\boldsymbol{\lambda}_{k+1} &\leq \epsilon && (34)
\end{aligned}
$$