

最佳化方法與應用二

MA5038-*

Chapter 16. Quadratic Programming

§16.1 Equality-Constrained Quadratic Programs

§16.2 Direct Solution of the KKT System

§16.3 Iterative Solution of the KKT System

§16.4 Inequality-Constrained Problems

§16.5 Active-Set Methods for Convex QPs

§16.6 Interior-Point Methods

§16.7 The Gradient Projection Method

§16.8 Perspectives and Software

Introduction

具有二次目標函數和線性限制式的最佳化問題稱為二次規劃 (quadratic program)。這類問題本身就很重要，並且它們還作為一般受限最佳化方法中的子問題而出現，例如序列二次規劃（第 18 章）、擴增拉格朗日方法（第 17 章）和內點法（第 19 章）。

一般的二次規劃 (QP) 可以陳述為：

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \quad (1a)$$

subject to

$$a_i^T x = b_i \quad \text{if } i \in \mathcal{E}, \quad (1b)$$

$$a_i^T x \geq b_i \quad \text{if } i \in \mathcal{I}, \quad (1c)$$

其中， G 是 $n \times n$ 對稱矩陣， $c, x \in \mathbb{R}^n$ ， \mathcal{E} 和 \mathcal{I} 是有限的 index set，而對所有的 $i \in \mathcal{E} \cup \mathcal{I}$ ， a_i 為 \mathbb{R}^n 向量而 b_i 是實數。

Introduction

二次規劃問題總是可以在有限的計算中解決（或顯示為無解），但尋找解的困難度取決於目標函數的特性和不等式限制式的數量。如果 Hessian 矩陣 G 是半正定的，我們稱 (1) 為凸 QP，在這種情況下，問題在困難度上通常類似於線性規劃。嚴格凸 QP 是指其中 G 是正定的情況，而在 G 是一個不定 (indefinite) 矩陣的情況 (1) 被稱為非凸 QP，非凸 QP 一般來說更具挑戰性，因為它們可能有多個 stationary point 和局部最小值。

在本章中，我們主要聚焦於凸 QP。我們首先考慮二次規劃的一個有趣應用。

Introduction

二次規劃問題總是可以在有限的計算中解決（或顯示為無解），但尋找解的困難度取決於目標函數的特性和不等式限制式的數量。如果 Hessian 矩陣 G 是半正定的，我們稱 (1) 為凸 QP，在這種情況下，問題在困難度上通常類似於線性規劃。嚴格凸 QP 是指其中 G 是正定的情況，而在 G 是一個不定 (indefinite) 矩陣的情況 (1) 被稱為非凸 QP，非凸 QP 一般來說更具挑戰性，因為它們可能有多個 stationary point 和局部最小值。

在本章中，我們主要聚焦於凸 QP。我們首先考慮二次規劃的一個有趣應用。

Introduction

Example (Portfolio optimization)

每位投資者都知道風險和報酬之間存在著一種平衡：為了增加投資的預期報酬，投資者必須願意容忍更大的風險，投資組合理論研究如何在給定 n 種可能的投資的情況下模擬這種平衡。以 $i = 1, 2, \dots, n$ 表示不同的投資項目，對投資 i 的報酬設為 r_i 。報酬 r_i 通常是未知的，通常被假設為依從常態分佈的隨機變量。我們可以通過它們的期望值 $\mu_i = E[r_i]$ 和變異數（方差） $\sigma_i^2 = E[(r_i - \mu_i)^2]$ 來刻畫這些變數。變異數可衡量變數 r_i 對其平均值的波動程度， σ_i 愈大表示投資 i 風險愈高。這些報酬一般來說不是獨立的，我們可以將報酬 i 與 j 之間的相關性定義如下：

$$\rho_{ij} = \frac{E[(r_i - \mu_i)(r_j - \mu_j)]}{\sigma_i \sigma_j} \quad \text{for } i, j = 1, 2, \dots, n.$$

Introduction

Example (cont'd)

相關性可以衡量投資 i 和投資 j 的報酬趨向於朝著相同方向移動的程度。兩個報酬趨向一起上升和下降的投資具有正相關性；當 ρ_{ij} 靠近 1 時，這兩個投資越接近。而報酬趨向於相反方向移動的投資具有負相關性。

投資者通過將可用資金的一部分 x_i 投入到投資 i 中（其中 $i = 1, 2, \dots, n$ ）來構建投資組合。假設所有可用資金都被投資，而且不允許空頭交易，那麼投資的限制條件為：
$$\sum_{i=1}^n x_i = 1 \text{ 且 } x \geq 0.$$

投資組合的報酬則由以下公式給出：

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

Introduction

Example (cont'd)

要衡量投資組合的吸引力，我們需要獲得報酬的平均和變異數等指標。很容易看到報酬的期望值是

$$E[R] = E \left[\sum_{i=1}^n x_i r_i \right] = \sum_{i=1}^n x_i E[r_i] = \mathbf{x}^T \boldsymbol{\mu},$$

而報酬的變異數則由以下公式給出：

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

其中 $n \times n$ 的對稱正半定矩陣 G 並被稱為共變異（協方差）矩陣，其 (i, j) -entry 為 $G_{ij} = \rho_{ij} \sigma_i \sigma_j$.

理想情況下，我們希望找到一個預期回報 $\mathbf{x}^T \boldsymbol{\mu}$ 較大而變異數 $\mathbf{x}^T \mathbf{G} \mathbf{x}$ 較小的投資組合。

Introduction

Example (cont'd)

在 Markowitz [201] 提出的模型中，我們將這兩個目標結合為一個單一的目標函數，並借助一個被表示為 κ 的“風險容忍參數”，通過解決以下問題

$$\max x^T \mu - \kappa x^T G x \quad \text{subject to} \quad \sum_{i=1}^n x_i = 1, x \geq 0.$$

來找到最優投資組合。非負參數 κ 的選擇取決於個別投資者的偏好。保守的投資者更注重在投資組合中降低風險因而選擇較大的 κ 值以增加目標函數中變異數度量的權重。更勇敢的投資者則願意冒更多風險以獲得更高預期報酬，則會選擇較小的 κ 值。在將這種投資組合優化技術應用於現實投資中的困難在於確定所涉及投資的預期回報、方差和相關性。金融專業人士通常會將歷史數據與他們自己的見解和期望相結合，以得出這些數量的值。

§16.1 Equality-Constrained Quadratic Programs

接下來我們開始討論 QP 演算法，首先考慮僅存在等式限制的情況。這個特殊情況的技術同樣適用於具有不等式限制的問題，因為正如我們稍後會在本章中看到的，一些用於一般 QP 的算法在每次迭代中需要解一個等式限制的 QP。

• Properties of equality-constrained QPs

為了簡單起見，我們將等式限制以矩陣形式表示，並將等式限制的 QP 表述如下：

$$\min_x q(x) \equiv \frac{1}{2}x^T Gx + x^T c \quad (2a)$$

$$\text{subject to } Ax = b \quad (2b)$$

其中 A 是限制函數的 $m \times n$ Jacobian 矩陣（其中 $m \leq n$ ）， b 是 \mathbb{R}^m 中的向量，而 A 的第 i 列是 a_i^T 而 b 的第 i 個分量為 b_i 。我們先假設 A 有滿秩 (full rank) – 所以 $\text{rank}(A) = m$ ，因此限制式 (2b) consistent。我們會在 §16.8 討論 A 秩不足的情況。

§16.1 Equality-Constrained Quadratic Programs

接下來我們開始討論 QP 演算法，首先考慮僅存在等式限制的情況。這個特殊情況的技術同樣適用於具有不等式限制的問題，因為正如我們稍後會在本章中看到的，一些用於一般 QP 的算法在每次迭代中需要解一個等式限制的 QP。

• Properties of equality-constrained QPs

為了簡單起見，我們將等式限制以矩陣形式表示，並將等式限制的 QP 表述如下：

$$\min_x q(x) \equiv \frac{1}{2}x^T Gx + x^T c \quad (2a)$$

$$\text{subject to } Ax = b \quad (2b)$$

其中 A 是限制函數的 $m \times n$ Jacobian 矩陣（其中 $m \leq n$ ）， b 是 \mathbb{R}^m 中的向量，而 A 的第 i 列是 a_i^T 而 b 的第 i 個分量為 b_i 。我們先假設 A 有滿秩 (full rank) – 所以 $\text{rank}(A) = m$ ，因此限制式 (2b) consistent。我們會在 §16.8 討論 A 秩不足的情況。

§16.1 Equality-Constrained Quadratic Programs

由課本定理 12.1 (即給出 KKT 條件的定理, 見下頁複習), 若 x_* 是 (2) 的解, 則一階必要條件表明, 存在一個被稱為 Lagrange 乘子向量 (Lagrange multiplier vector) 的向量 λ_* 滿足方程組

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x_* \\ \lambda_* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}. \quad (3)$$

系統 (3) 可以通過將 x_* 表示為 $x_* = x + p$ 的形式進行重寫, 其中 x 是解的某些估計, p 是所需的步進量。引入這些符號並重新排列方程, 我們得到

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

其中

$$h = Ax - b, \quad g = c + Gx, \quad p = x_* - x. \quad (5)$$

將 (3) 改寫為 (4) 對於計算來說是有用的。

Theorem 12.1 – KKT conditions

Theorem (First-Order Necessary Conditions)

Suppose that x_* is a local solution of problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geq 0, i \in \mathcal{I}, \end{cases} \quad (1)_{12}$$

that the functions f and c_i in (1)₁₂ are continuously differentiable, and that the LICQ holds at x_* . Then there is a Lagrange multiplier vector λ_* , with components λ_i^* , $i \in \mathcal{E} \cup \mathcal{I}$, such that the following conditions are satisfied.

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = 0, \quad (32a)_{12}$$

$$c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E}, \quad (32b)_{12}$$

$$c_i(x_*) \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32c)_{12}$$

$$\lambda_i^* \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32d)_{12}$$

$$\lambda_i^* c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}. \quad (32e)_{12}$$

§16.1 Equality-Constrained Quadratic Programs

由課本定理 12.1 (即給出 KKT 條件的定理, 見上頁複習), 若 x_* 是 (2) 的解, 則一階必要條件表明, 存在一個被稱為 Lagrange 乘子向量 (Lagrange multiplier vector) 的向量 λ_* 滿足方程組

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x_* \\ \lambda_* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}. \quad (3)$$

系統 (3) 可以通過將 x_* 表示為 $x_* = x + p$ 的形式進行重寫, 其中 x 是解的某些估計 (經演算法迭代計算所接受的值), p 是所需的步進量。引入這些符號並重新排列方程, 我們得到

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

其中

$$h = Ax - b, \quad g = c + Gx, \quad p = x_* - x. \quad (5)$$

將 (3) 改寫為 (4) 對於計算來說是有用的。

§16.1 Equality-Constrained Quadratic Programs

在 (4) 中的矩陣被稱為 Karush-Kuhn-Tucker (KKT) 矩陣，而下面的結果給出了其為 non-singular 的一個充分條件。跟第 15 章一樣，我們使用 Z 表示“其行向量集構成 A 的 null space 的一組基底”的 $n \times (n - m)$ 矩陣；亦即 Z 滿足 $AZ = 0$ 且有滿秩。

Lemma

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

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \quad (6)$$

is non-singular, and hence for each b and c there is a unique vector pair (x_*, λ_*) satisfying

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x_* \\ \lambda_* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}. \quad (3)$$

§16.1 Equality-Constrained Quadratic Programs

Proof.

Suppose there are vectors w and v such that

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = 0. \quad (7)$$

Then $Aw = 0$; thus from (7) we have that

$$0 = \begin{bmatrix} w \\ v \end{bmatrix}^T \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = w^T G w.$$

Since w lies in the null space of A , it can be written as $w = Zu$ for some vector $u \in \mathbb{R}^{n-m}$. Therefore, we have $0 = w^T G w = u^T Z^T G Z u$, which by positive definiteness of $Z^T G Z$ implies that $u = 0$. Therefore, $w = 0$, and by (7), $A^T v = 0$. Full row rank of A then implies that $v = 0$. We conclude that equation (7) is satisfied only if $w = 0$ and $v = 0$, so the matrix is non-singular, as claimed. \square

§16.1 Equality-Constrained Quadratic Programs

Proof.

Suppose there are vectors w and v such that

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = 0. \quad (7)$$

Then $Aw = 0$; thus from (7) we have that

$$0 = \begin{bmatrix} w \\ v \end{bmatrix}^T \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = w^T G w.$$

Since w lies in the null space of A , it can be written as $w = Zu$ for some vector $u \in \mathbb{R}^{n-m}$. Therefore, we have $0 = w^T G w = u^T Z^T G Z u$, which by positive definiteness of $Z^T G Z$ implies that $u = 0$. Therefore, $w = 0$, and by (7), $A^T v = 0$. Full row rank of A then implies that $v = 0$. We conclude that equation (7) is satisfied only if $w = 0$ and $v = 0$, so the matrix is non-singular, as claimed. \square

§16.1 Equality-Constrained Quadratic Programs

Proof.

Suppose there are vectors w and v such that

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = 0. \quad (7)$$

Then $Aw = 0$; thus from (7) we have that

$$0 = \begin{bmatrix} w \\ v \end{bmatrix}^T \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} = w^T G w.$$

Since w lies in the null space of A , it can be written as $w = Zu$ for some vector $u \in \mathbb{R}^{n-m}$. Therefore, we have $0 = w^T G w = u^T Z^T G Z u$, which by positive definiteness of $Z^T G Z$ implies that $u = 0$. Therefore, $w = 0$, and by (7), $A^T v = 0$. Full row rank of A then implies that $v = 0$. We conclude that equation (7) is satisfied only if $w = 0$ and $v = 0$, so the matrix is non-singular, as claimed. \square

§16.1 Equality-Constrained Quadratic Programs

Example

Consider the quadratic programming problem

$$\begin{aligned} \min q(x) &= 3x_1^2 + \frac{5}{2}x_2^2 + 2x_3^2 + 2x_1x_2 + 1x_1x_3 + 2x_2x_3 \\ &\quad - 8x_1 - 3x_2 - 3x_3 \\ \text{subject to} \quad &x_1 + x_3 = 3, \quad x_2 + x_3 = 0. \end{aligned} \quad (8)$$

We can write this problem in the form (2) by defining

$$G = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}, \quad c = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

The solution x_* and optimal Lagrange multiplier vector λ_* are given by

$$x_* = (2, -1, 1)^T, \quad \lambda_* = (3, -2)^T.$$

§16.1 Equality-Constrained Quadratic Programs

Example

Consider the quadratic programming problem

$$\min q(x) = \frac{6}{2}x_1^2 + \frac{5}{2}x_2^2 + \frac{4}{2}x_3^2 + 2x_1x_2 + 1x_1x_3 + 2x_2x_3 \\ - 8x_1 - 3x_2 - 3x_3$$

$$\text{subject to } x_1 + x_3 = 3, \quad x_2 + x_3 = 0. \quad (8)$$

We can write this problem in the form (2) by defining

$$G = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}, \quad c = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

The solution x_* and optimal Lagrange multiplier vector λ_* are given by

$$x_* = (2, -1, 1)^T, \quad \lambda_* = (3, -2)^T.$$

§16.1 Equality-Constrained Quadratic Programs

Example (cont'd)

In this example, the matrix G is positive definite, and the null-space basis matrix is given by

$$Z = (-1, -1, 1)^T. \quad (9)$$

我們已經看到，當滿足前面 lemma 的條件時，存在唯一的向量 (x_*, λ_*) 滿足 (2) 的一階必要條件。實際上，二階充分條件（定理 12.6 – 見下頁）在 (x_*, λ_*) 也被滿足，因此 x_* 是 (2) 的嚴格局部最小值。實際上，我們可以使用直接的方法來證明 x_* 是 (2) 的 global solution。

Theorem

Let A have full row rank and assume that the reduced-Hessian matrix $Z^T G Z$ is positive definite. Then the vector x_ satisfying (3) is the unique global solution of (2).*

§16.1 Equality-Constrained Quadratic Programs

Example (cont'd)

In this example, the matrix G is positive definite, and the null-space basis matrix is given by

$$Z = (-1, -1, 1)^T. \quad (9)$$

我們已經看到，當滿足前面 lemma 的條件時，存在唯一的向量 (x_*, λ_*) 滿足 (2) 的一階必要條件。實際上，二階充分條件（定理 12.6 – 見下頁）在 (x_*, λ_*) 也被滿足，因此 x_* 是 (2) 的嚴格局部最小值。實際上，我們可以使用直接的方法來證明 x_* 是 (2) 的 global solution。

Theorem

Let A have full row rank and assume that the reduced-Hessian matrix $Z^T G Z$ is positive definite. Then the vector x_ satisfying (3) is the unique global solution of (2).*

Theorem 12.6 – Second-Order Conditions

Theorem (Second-Order Sufficient Conditions)

Suppose that for some feasible point $x_* \in \mathbb{R}^n$ there is a Lagrange multiplier vector λ_* such that the KKT conditions

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = 0, \quad (32a)_{12}$$

$$c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E}, \quad (32b)_{12}$$

$$c_i(x_*) \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32c)_{12}$$

$$\lambda_i^* \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32d)_{12}$$

$$\lambda_i^* c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}. \quad (32e)_{12}$$

are satisfied. Suppose also that

$$w^T \nabla_{xx}^2 \mathcal{L}(x_*, \lambda_*) w > 0 \quad \text{for all } w \in \mathcal{C}(x_*, \lambda_*) \setminus \{0\}. \quad (60)_{12}$$

Then x_* is a strict local solution for

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geq 0, i \in \mathcal{I}. \end{cases} \quad (1)_{12}$$

§16.1 Equality-Constrained Quadratic Programs

Example (cont'd)

In this example, the matrix G is positive definite, and the null-space basis matrix is given by

$$Z = (-1, -1, 1)^T. \quad (9)$$

我們已經看到，當滿足前面 lemma 的條件時，存在唯一的向量 (x_*, λ_*) 滿足 (2) 的一階必要條件。實際上，二階充分條件（定理 12.6 – 見上頁）在 (x_*, λ_*) 也被滿足，因此 x_* 是 (2) 的嚴格局部最小值。實際上，我們可以使用直接的方法來證明 x_* 是 (2) 的 global solution。

Theorem

Let A have full row rank and assume that the reduced-Hessian matrix $Z^T G Z$ is positive definite. Then the vector x_ satisfying (3) is the unique global solution of (2).*

§16.1 Equality-Constrained Quadratic Programs

Proof.

Let x be any other feasible point (satisfying $Ax = b$), and as before, let p denote the difference $x_* - x$. Since $Ax_* = Ax = b$, we have that $Ap = 0$. By substituting into the objective function (2a), we obtain

$$\begin{aligned} q(x) &= \frac{1}{2}(x_* - p)^T G(x_* - p) + c^T(x_* - p) \\ &= \frac{1}{2}p^T Gp - p^T Gx_* - c^T p + q(x_*). \end{aligned} \quad (10)$$

From (3) we have that $Gx_* = -c + A^T \lambda_*$, so from $Ap = 0$ we have

$$p^T Gx_* = p^T(-c + A^T \lambda_*) = -p^T c.$$

By substituting this relation into (10), we obtain

$$q(x) = \frac{1}{2}p^T Gp + q(x_*). \quad \square$$

§16.1 Equality-Constrained Quadratic Programs

Proof (cont'd).

Since p lies in the null space of A , we can write $p = Zu$ for some vector $u \in \mathbb{R}^{n-m}$, so that

$$q(x) = \frac{1}{2}u^T Z^T G Z u + q(x_*).$$

By positive definiteness of $Z^T G Z$, we conclude that $q(x) > q(x_*)$ except when $u = 0$; that is, when $x = x_*$. Therefore, x_* is the unique global solution of (2). \square

當 reduced-Hessian 矩陣 $Z^T G Z$ 是半正定且有零特徵值時，滿足 (3) 的向量 x_* 是一個局部 minimizer，但不是一個嚴格的 minimizer。如果 reduced-Hessian 矩陣有負特徵值，那麼 x_* 只是一個 stationary point，而不是局部 minimizer。

§16.1 Equality-Constrained Quadratic Programs

Proof (cont'd).

Since p lies in the null space of A , we can write $p = Zu$ for some vector $u \in \mathbb{R}^{n-m}$, so that

$$q(x) = \frac{1}{2}u^T Z^T G Z u + q(x_*).$$

By positive definiteness of $Z^T G Z$, we conclude that $q(x) > q(x_*)$ except when $u = 0$; that is, when $x = x_*$. Therefore, x_* is the unique global solution of (2). \square

當 reduced-Hessian 矩陣 $Z^T G Z$ 是半正定且有零特徵值時，滿足 (3) 的向量 x_* 是一個局部 minimizer，但不是一個嚴格的 minimizer。如果 reduced-Hessian 矩陣有負特徵值，那麼 x_* 只是一個 stationary point，而不是局部 minimizer。

§16.2 Direct Solution of the KKT System

在本節中，我們討論求解 KKT 系統 (4) 的高效方法。第一個重要的觀察是，如果 $m \geq 1$ ，那麼 KKT 矩陣 $\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}$ 總是不定的 (indefinite)。我們將對稱矩陣 K 的 inertia 定義為表示正、負和零特徵值數量的三元組，亦即

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

以下結果刻畫了 KKT 矩陣的 inertia。

Theorem

Let K be the KKT matrix defined by (6), and suppose that A has full row rank m . Then

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

因此，如果 $Z^T G Z$ 是正定的，那麼 $\text{inertia}(K) = (n, m, 0)$ 。

§16.2 Direct Solution of the KKT System

在本節中，我們討論求解 KKT 系統 (4) 的高效方法。第一個重要的觀察是，如果 $m \geq 1$ ，那麼 KKT 矩陣 $\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}$ 總是不定的 (indefinite)。我們將對稱矩陣 K 的 inertia 定義為表示正、負和零特徵值數量的三元組，亦即

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

以下結果刻畫了 KKT 矩陣的 inertia。

Theorem

Let K be the KKT matrix defined by (6), and suppose that A has full row rank m . Then

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

因此，如果 $Z^T G Z$ 是正定的，那麼 $\text{inertia}(K) = (n, m, 0)$ 。

§16.2 Direct Solution of the KKT System

這個結果的證明可以在 [111] 中找到（在本課程不予證明）。注意，這個定理的假設在前一個例子中是滿足的。因此，如果我們使用這個例子的數據構造 5×5 KKT 矩陣

$$K = \begin{bmatrix} 6 & 2 & 1 & 1 & 0 \\ 2 & 5 & 2 & 0 & 1 \\ 1 & 2 & 4 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

我們可得到 $\text{inertia}(K) = (3, 2, 0)$ 。

由前述定理可知當 A 有滿秩時，KKT 矩陣 K 是正負未定的（因 $\text{inertia}(K)$ 的前兩個分量必為正），接下來我們描述解決 (4) 的主要直接技術。

§16.2 Direct Solution of the KKT System

這個結果的證明可以在 [111] 中找到（在本課程不予證明）。注意，這個定理的假設在前一個例子中是滿足的。因此，如果我們使用這個例子的數據構造 5×5 KKT 矩陣

$$K = \begin{bmatrix} 6 & 2 & 1 & 1 & 0 \\ 2 & 5 & 2 & 0 & 1 \\ 1 & 2 & 4 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

我們可得到 $\text{inertia}(K) = (3, 2, 0)$ 。

由前述定理可知當 A 有滿秩時，KKT 矩陣 K 是正負未定的（因 $\text{inertia}(K)$ 的前兩個分量必為正），接下來我們描述解決 (4) 的主要直接技術。

§16.2 Direct Solution of the KKT System

這個結果的證明可以在 [111] 中找到（在本課程不予證明）。注意，這個定理的假設在前一個例子中是滿足的。因此，如果我們使用這個例子的數據構造 5×5 KKT 矩陣

$$K = \begin{bmatrix} 6 & 2 & 1 & 1 & 0 \\ 2 & 5 & 2 & 0 & 1 \\ 1 & 2 & 4 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

我們可得到 $\text{inertia}(K) = (3, 2, 0)$ 。

由前述定理可知當 A 有滿秩時，KKT 矩陣 K 是正負未定的（因 $\text{inertia}(K)$ 的前兩個分量必為正），接下來我們描述解決 (4) 的主要直接技術。

§16.2 Direct Solution of the KKT System

- **Factoring the full KKT system**

解 KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

的一種方法是對整個 KKT 矩陣進行三角分解，然後對三角因子進行 backward 和 forward substitution。由於 KKT 矩陣是正負未定的關係，我們不能使用 Cholesky 分解。我們可以使用具有 partial pivoting 的高斯消去法（或其稀疏變體）來進行 LU 分解以獲得 L 和 U 因子，但這種方法的缺點是忽略了對稱性。

§16.2 Direct Solution of the KKT System

在這種情況下最有效的策略是使用 symmetric indefinite factorization。對於一個一般的對稱矩陣 K ，這種因子分解形式為：

$$P^T K P = L B L^T, \quad (11)$$

其中 P 是一個 permutation 矩陣， L 是單位下三角矩陣， B 是帶有 1×1 或 2×2 block 的 block-diagonal 矩陣。矩陣 P 的引入是為了計算的數值穩定性，並且在 K 是大型稀疏矩陣的情況下該分解也可以保持稀疏性。Symmetric indefinite factorization (11) 的計算成本通常約為稀疏高斯消去的一半。

§16.2 Direct Solution of the KKT System

在這種情況下最有效的策略是使用 symmetric indefinite factorization。對於一個一般的對稱矩陣 K ，這種因子分解形式為：

$$P^T K P = L B L^T, \quad (11)$$

其中 P 是一個 permutation 矩陣， L 是單位下三角矩陣， B 是帶有 1×1 或 2×2 block 的 block-diagonal 矩陣。矩陣 P 的引入是為了計算的數值穩定性，並且在 K 是大型稀疏矩陣的情況下該分解也可以保持稀疏性。Symmetric indefinite factorization (11) 的計算成本通常約為稀疏高斯消去的一半。

§16.2 Direct Solution of the KKT System

為了求解 KKT 系統 (4)，我們首先計算 K 的 symmetric indefinite factorization (11)。然後，我們執行以下操作順序來得到解：

- ① solve $Lz = P^T \begin{bmatrix} g \\ h \end{bmatrix}$ to obtain z ;
- ② solve $B\hat{z} = z$ to obtain \hat{z} ;
- ③ solve $L^T\bar{z} = \hat{z}$ to obtain \bar{z} ;
- ④ set $\begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = P\bar{z}$.

由於與排列矩陣 P 和 P^T 的相乘可以通過簡單地重新排列向量分量來執行，所以這部份計算量較少。解系統 $B\hat{z} = z$ 意味著解一些小的 1×1 和 2×2 系統，因此操作次數是系統維度 $(m+n)$ 的一個小倍數，同樣是計算量比較少的。使用 L 和 L^T 進行三角替換的成本較高。它們的確切成本取決於稀疏度的程度，但通常顯著低於執行 symmetric indefinite factorization (11) 的成本。

§16.2 Direct Solution of the KKT System

這種對完整的 $(n + m) \times (n + m)$ KKT 矩陣 (6) 進行分解的方法對許多問題都非常有效。然而，當基於經驗和直覺所選擇排列矩陣 P 無法保持 L 因子的稀疏性時，這種方法的計算量可能會非常大，使得 L 比原始係數矩陣 K 更加密集。

• Schur-Complement method

假設 G 是正定的，我們可以將 (4) 中的第一個方程式乘以 AG^{-1} 然後減去第二個方程式，得到僅包含向量 λ_* 的線性系統：

$$(AG^{-1}A^T)\lambda_* = (AG^{-1}g - h). \quad (12)$$

我們解這個對稱正定系統以獲得 λ_* ，然後通過解 (4) 的第一個方程式

$$Gp = A^T\lambda_* - g \quad (13)$$

而取得 p 。

§16.2 Direct Solution of the KKT System

這種對完整的 $(n + m) \times (n + m)$ KKT 矩陣 (6) 進行分解的方法對許多問題都非常有效。然而，當基於經驗和直覺所選擇排列矩陣 P 無法保持 L 因子的稀疏性時，這種方法的計算量可能會非常大，使得 L 比原始係數矩陣 K 更加密集。

• Schur-Complement method

假設 G 是正定的，我們可以將 (4) 中的第一個方程式乘以 AG^{-1} 然後減去第二個方程式，得到僅包含向量 λ_* 的線性系統：

$$(AG^{-1}A^T)\lambda_* = (AG^{-1}g - h). \quad (12)$$

我們解這個對稱正定系統以獲得 λ_* ，然後通過解 (4) 的第一個方程式

$$Gp = A^T\lambda_* - g \quad (13)$$

而取得 p 。

§16.2 Direct Solution of the KKT System

上述方法要求我們進行 G^{-1} 的操作，並計算 $m \times m$ 矩陣 $AG^{-1}A^T$ 的分解。因此，這種方法在以下情況下最有用：

- G 是 well-conditioned 且容易進行逆運算時（例如，當 G 是 diagonal 或 block-diagonal 時）；或
- 透過 quasi-Newton 更新公式明確知道了 G^{-1} ；或
- 等式限制數量 m 較小，這樣形成矩陣 $AG^{-1}A^T$ 所需的反向求解次數不會太大。

“Schur-Complement 方法”的名稱源於以下事實：矩陣 $-AG^{-1}A^T$ 是 KKT 矩陣 K 中子方陣 G 的 Schur complement，而引進 G 的 Schur complement 後可將 KKT 矩陣 K 拆解成

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ AG^{-1} & I \end{bmatrix} \begin{bmatrix} G & A^T \\ 0 & -AG^{-1}A^T \end{bmatrix}.$$

§16.2 Direct Solution of the KKT System

上述方法要求我們進行 G^{-1} 的操作，並計算 $m \times m$ 矩陣 $AG^{-1}A^T$ 的分解。因此，這種方法在以下情況下最有用：

- G 是 well-conditioned 且容易進行逆運算時（例如，當 G 是 diagonal 或 block-diagonal 時）；或
- 透過 quasi-Newton 更新公式明確知道了 G^{-1} ；或
- 等式限制數量 m 較小，這樣形成矩陣 $AG^{-1}A^T$ 所需的反向求解次數不會太大。

“Schur-Complement 方法”的名稱源於以下事實：矩陣 $-AG^{-1}A^T$ 是 KKT 矩陣 K 中子方陣 G 的 Schur complement，而引進 G 的 Schur complement 後可將 KKT 矩陣 K 拆解成

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ AG^{-1} & I \end{bmatrix} \begin{bmatrix} G & A^T \\ 0 & -AG^{-1}A^T \end{bmatrix}.$$

§16.2 Direct Solution of the KKT System

亦即通過對 (6) 使用 G 作為 pivot 進行 block Gaussian elimination，我們可拆解 KKT 矩陣 K 並將原 KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

改寫為

$$\begin{bmatrix} I & 0 \\ AG^{-1} & I \end{bmatrix} \begin{bmatrix} G & A^T \\ 0 & -AG^{-1}A^T \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}$$

進而反解得到以下的 block upper triangular 系統

$$\begin{bmatrix} G & A^T \\ 0 & -AG^{-1}A^T \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h - AG^{-1}g \end{bmatrix}.$$

(12) 與 (13) 式即是通過將這種 block elimination 技術應用於系統 (4) 並執行 block backsolve 而得到的。

Schur Complement

在線性代數術語中，所謂的 Schur complement 是針對一個方陣的子方陣來說的。給定一個方陣 M ，將其分成四個 blocks 寫成

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

其中 A, D 為方陣，而 B, C^T 為同樣大小的矩陣。針對方陣 M 的兩個子方陣 A 與 D ，我們有：

- ① 若 A 可逆，則 A 的 Schur complement 為 $D - CA^{-1}B$ 。
- ② 若 D 可逆，則 D 的 Schur complement 為 $A - BD^{-1}C$ 。

引進 Schur complement 是為了將具備 Schur complement 子方陣的方陣進行 block triangular 分解。同時若 Schur complement 是可逆的，則原（大）方陣的反矩陣也可以有公式解，如下頁所述。

Schur Complement

在線性代數術語中，所謂的 Schur complement 是針對一個方陣的子方陣來說的。給定一個方陣 M ，將其分成四個 blocks 寫成

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

其中 A, D 為方陣，而 B, C^T 為同樣大小的矩陣。針對方陣 M 的兩個子方陣 A 與 D ，我們有：

- ① 若 A 可逆，則 A 的 Schur complement 為 $D - CA^{-1}B$ 。
- ② 若 D 可逆，則 D 的 Schur complement 為 $A - BD^{-1}C$ 。

引進 Schur complement 是為了將具備 Schur complement 子方陣的方陣進行 block triangular 分解。同時若 Schur complement 是可逆的，則原（大）方陣的反矩陣也可以有公式解，如下頁所述。

Schur Complement

- ① 若 A 可逆時，則

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix}.$$

若更進一步 A 的 Schur complement 可逆，則 M 的反矩陣為

$$\begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}.$$

- ② 若 D 可逆時，則

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A - BD^{-1}C & B \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}$$

若更進一步 D 的 Schur complement 可逆，則 M 的反矩陣為

$$\begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix}.$$

§16.2 Direct Solution of the KKT System

若使用前述 Schur complement 來推導 (4) 中 KKT 矩陣的反矩陣，我們得到

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}^{-1} = \begin{bmatrix} C & E \\ E^T & F \end{bmatrix},$$

其中

$$C = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1},$$

$$E = G^{-1}A^T(AG^{-1}A^T)^{-1},$$

$$F = -(AG^{-1}A^T)^{-1}.$$

KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

的解可以通過將 (4) 中等號兩側左乘以上述的反矩陣後而得。適當地將上述計算分組即為 (12) 與 (13) 二式。

§16.2 Direct Solution of the KKT System

若使用前述 Schur complement 來推導 (4) 中 KKT 矩陣的反矩陣，我們得到

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix}^{-1} = \begin{bmatrix} C & E \\ E^T & F \end{bmatrix},$$

其中

$$C = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1},$$

$$E = G^{-1}A^T(AG^{-1}A^T)^{-1},$$

$$F = -(AG^{-1}A^T)^{-1}.$$

KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

的解可以通過將 (4) 中等號兩側左乘以上述的反矩陣後而得。適當地將上述計算分組即為 (12) 與 (13) 二式。

§16.2 Direct Solution of the KKT System

• Null-Space method

Null-space 方法不需要假設 G 的可逆性，因此比 Schur-complement 方法有更廣泛的應用。它僅假設前述 lemma 的條件成立，即 A 有滿秩且 $Z^T G Z$ 是正定的。然而，它需要知道 null space 基底矩陣 Z 。像 Schur-complement 方法一樣，它利用了 KKT 系統中的 block 結構，將 (4) decouple 為兩個較小的系統。

將 (4) 中的向量 p 寫為

$$p = Y p_Y + Z p_Z, \quad (14)$$

其中 Z 是 $n \times (n - m)$ null-space 基底矩陣， Y 是使得 $[Y:Z]$ 為可逆方陣的任一 $n \times m$ 矩陣， p_Y 是一個 m 維向量， p_Z 是一個 $(n - m)$ 維向量。我們已在 §15.3 中討論過矩陣 Y 和 Z ，其中圖 15.4 (見下頁) 顯示 $Y x_Y$ 是 $Ax = b$ 的一個特解，而 $Z x_Z$ 是沿著這些限制的位移。

§16.2 Direct Solution of the KKT System

• Null-Space method

Null-space 方法不需要假設 G 的可逆性，因此比 Schur-complement 方法有更廣泛的應用。它僅假設前述 lemma 的條件成立，即 A 有滿秩且 $Z^T G Z$ 是正定的。然而，它需要知道 null space 基底矩陣 Z 。像 Schur-complement 方法一樣，它利用了 KKT 系統中的 block 結構，將 (4) decouple 為兩個較小的系統。

將 (4) 中的向量 p 寫為

$$p = Y p_Y + Z p_Z, \quad (14)$$

其中 Z 是 $n \times (n - m)$ null-space 基底矩陣， Y 是使得 $[Y:Z]$ 為可逆方陣的任一 $n \times m$ 矩陣， p_Y 是一個 m 維向量， p_Z 是一個 $(n - m)$ 維向量。我們已在 §15.3 中討論過矩陣 Y 和 Z ，其中圖 15.4 (見下頁) 顯示 $Y x_Y$ 是 $Ax = b$ 的一個特解，而 $Z x_Z$ 是沿著這些限制的位移。

圖 15.4 from Chapter 15

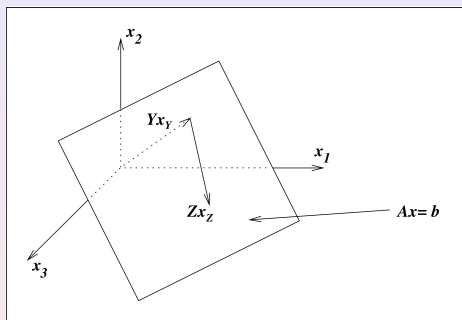


Figure 15.4: General elimination: Case in which $A \in \mathbb{R}^{1 \times 3}$, showing the particular solution and a step in the null space of A .

§16.2 Direct Solution of the KKT System

將形式為 (14) 的 p 代入 (4) 的第二個方程並使用 $AZ=0$ ，可得

$$(AY)p_Y = -h. \quad (15)$$

由於 A 的秩為 m 且 $[Y:Z]$ 是一個 $n \times n$ 可逆矩陣，所以乘積 $A[Y:Z] = [AY:0]$ 的秩為 m 。因此， AY 是一個 $m \times m$ 可逆矩陣，所以 p_Y 可由解方程 (15) 而得。與此同時，我們可以將形式為 (14) 的 p 代入 (4) 的第一個方程而得到

$$-GYp_Y - GZp_Z + A^T\lambda_* = g$$

並進一步在上式等號兩側左乘 Z^T (並注意到 $Z^TA^T=0$) 而得

$$(Z^TGZ)p_Z = -Z^TGYp_Y - Z^Tg. \quad (16)$$

因為 reduced-Hessian 矩陣 Z^TGZ 是正定的，可對其進行 Cholesky 分解來解 (16) 這個系統，從而確定 p_Z 。由此，我們可以計算總步進量 $p = Yp_Y + Zp_Z$ 。

§16.2 Direct Solution of the KKT System

將形式為 (14) 的 p 代入 (4) 的第二個方程並使用 $AZ = 0$ ，可得

$$(AY)p_Y = -h. \quad (15)$$

由於 A 的秩為 m 且 $[Y:Z]$ 是一個 $n \times n$ 可逆矩陣，所以乘積 $A[Y:Z] = [AY:0]$ 的秩為 m 。因此， AY 是一個 $m \times m$ 可逆矩陣，所以 p_Y 可由解方程 (15) 而得。與此同時，我們可以將形式為 (14) 的 p 代入 (4) 的第一個方程而得到

$$-GYp_Y - GZp_Z + A^T\lambda_* = g$$

並進一步在上式等號兩側左乘 Z^T (並注意到 $Z^TA^T = 0$) 而得

$$(Z^TGZ)p_Z = -Z^TGYp_Y - Z^Tg. \quad (16)$$

因為 reduced-Hessian 矩陣 Z^TGZ 是正定的，可對其進行 Cholesky 分解來解 (16) 這個系統，從而確定 p_Z 。由此，我們可以計算總步進量 $p = Yp_Y + Zp_Z$ 。

§16.2 Direct Solution of the KKT System

將形式為 (14) 的 p 代入 (4) 的第二個方程並使用 $AZ = 0$ ，可得

$$(AY)p_Y = -h. \quad (15)$$

由於 A 的秩為 m 且 $[Y:Z]$ 是一個 $n \times n$ 可逆矩陣，所以乘積 $A[Y:Z] = [AY:0]$ 的秩為 m 。因此， AY 是一個 $m \times m$ 可逆矩陣，所以 p_Y 可由解方程 (15) 而得。與此同時，我們可以將形式為 (14) 的 p 代入 (4) 的第一個方程而得到

$$-GYp_Y - GZp_Z + A^T\lambda_* = g$$

並進一步在上式等號兩側左乘 Z^T （並注意到 $Z^TA^T = 0$ ）而得

$$(Z^TGZ)p_Z = -Z^TGYp_Y - Z^Tg. \quad (16)$$

因為 reduced-Hessian 矩陣 Z^TGZ 是正定的，可對其進行 Cholesky 分解來解 (16) 這個系統，從而確定 p_Z 。由此，我們可以計算總步進量 $p = Yp_Y + Zp_Z$ 。

§16.2 Direct Solution of the KKT System

將形式為 (14) 的 p 代入 (4) 的第二個方程並使用 $AZ = 0$ ，可得

$$(AY)p_Y = -h. \quad (15)$$

由於 A 的秩為 m 且 $[Y:Z]$ 是一個 $n \times n$ 可逆矩陣，所以乘積 $A[Y:Z] = [AY:0]$ 的秩為 m 。因此， AY 是一個 $m \times m$ 可逆矩陣，所以 p_Y 可由解方程 (15) 而得。與此同時，我們可以將形式為 (14) 的 p 代入 (4) 的第一個方程而得到

$$-GYp_Y - GZp_Z + A^T\lambda_* = g$$

並進一步在上式等號兩側左乘 Z^T （並注意到 $Z^TA^T = 0$ ）而得

$$(Z^TGZ)p_Z = -Z^TGYp_Y - Z^Tg. \quad (16)$$

因為 reduced-Hessian 矩陣 Z^TGZ 是正定的，可對其進行 Cholesky 分解來解 (16) 這個系統，從而確定 p_Z 。由此，我們可以計算總步進量 $p = Yp_Y + Zp_Z$ 。

§16.2 Direct Solution of the KKT System

對於 Lagrange 乘子向量 λ_* ，我們可以將 KKT 系統 (4) 中的第一個方程左乘 Y^T 而得到方程組

$$(AY)^T \lambda_* = Y^T(g + Gp). \quad (17)$$

由於 AY 可逆，解上述系統便可以得到 λ_* 。

Example

Consider the problem

$$\begin{aligned} \min q(x) = & 3x_1^2 + \frac{5}{2}x_2^2 + 2x_3^2 + 2x_1x_2 + 1x_1x_3 + 2x_2x_3 \\ & -8x_1 - 3x_2 - 3x_3 \end{aligned}$$

$$\text{subject to} \quad x_1 + x_3 = 3, \quad x_2 + x_3 = 0. \quad (8)$$

given in previous example. Recall that

$$Z = (-1, -1, 1)^T. \quad (9)$$

§16.2 Direct Solution of the KKT System

對於 Lagrange 乘子向量 λ_* ，我們可以將 KKT 系統 (4) 中的第一個方程左乘 Y^T 而得到方程組

$$(AY)^T \lambda_* = Y^T(g + Gp). \quad (17)$$

由於 AY 可逆，解上述系統便可以得到 λ_* 。

Example

Consider the problem

$$\begin{aligned} \min q(x) = & 3x_1^2 + \frac{5}{2}x_2^2 + 2x_3^2 + 2x_1x_2 + 1x_1x_3 + 2x_2x_3 \\ & -8x_1 - 3x_2 - 3x_3 \end{aligned}$$

$$\text{subject to} \quad x_1 + x_3 = 3, \quad x_2 + x_3 = 0. \quad (8)$$

given in previous example. Recall that

$$Z = (-1, -1, 1)^T. \quad (9)$$

§16.2 Direct Solution of the KKT System

Example (cont'd)

We can choose

$$Y = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \\ 1/3 & 1/3 \end{bmatrix}$$

so that $[Y;Z]$ is non-singular and $AY = I$.

Suppose we have $x = (0, 0, 0)^T$ in (5). Then

$$h = Ax - b = -b, \quad g = c + Gx = c = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix}.$$

Simple calculation shows that

$$p_Y = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad p_Z = [0],$$

so that

$$p = x_* - x = Yp_Y + Zp_Z = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}.$$

§16.2 Direct Solution of the KKT System

Example (cont'd)

After recovering λ_* from (17), we conclude that

$$x_* = (2, -1, 1)^T, \quad \lambda_* = (3, -2)^T.$$

當 $n - m$ 較小時，null-space 方法可能非常有效。Null-space 方法的主要限制在於需要計算基底矩陣 Z ，而正如我們在第 15 章中所見，其在某些大型問題中可能需要花費很大的計算量才能得到。此外基底矩陣 Z 不是唯一的，若選擇不當，縮減系統

$$(Z^T G Z) p_Z = -Z^T G Y p_Y - Z^T g \quad (16)$$

可能為 ill conditioned。若 Z 的行向量構成 A 之 null space 的 orthonormal 基底，則 $Z^T G Z$ 的條件數通常至少與 G 本身一樣好，但如此的 Z 一般只可能在中小型問題中取得。當 A 大且稀疏時，orthonormal 的 Z 難以計算，導致我們常被迫使用第 15 章中描述的不太可靠的 Z 。

§16.2 Direct Solution of the KKT System

Example (cont'd)

After recovering λ_* from (17), we conclude that

$$x_* = (2, -1, 1)^T, \quad \lambda_* = (3, -2)^T.$$

當 $n - m$ 較小時，null-space 方法可能非常有效。Null-space 方法的主要限制在於需要計算基底矩陣 Z ，而正如我們在第 15 章中所見，其在某些大型問題中可能需要花費很大的計算量才能得到。此外基底矩陣 Z 不是唯一的，若選擇不當，縮減系統

$$(Z^T G Z) p_Z = -Z^T G Y p_Y - Z^T g \quad (16)$$

可能為 ill conditioned。若 Z 的行向量構成 A 之 null space 的 orthonormal 基底，則 $Z^T G Z$ 的條件數通常至少與 G 本身一樣好，但如此的 Z 一般只可能在中小型問題中取得。當 A 大且稀疏時，orthonormal 的 Z 難以計算，導致我們常被迫使用第 15 章中描述的不太可靠的 Z 。

§16.2 Direct Solution of the KKT System

Example (cont'd)

After recovering λ_* from (17), we conclude that

$$x_* = (2, -1, 1)^T, \quad \lambda_* = (3, -2)^T.$$

當 $n - m$ 較小時，null-space 方法可能非常有效。Null-space 方法的主要限制在於需要計算基底矩陣 Z ，而正如我們在第 15 章中所見，其在某些大型問題中可能需要花費很大的計算量才能得到。此外基底矩陣 Z 不是唯一的，若選擇不當，縮減系統

$$(Z^T G Z) p_z = -Z^T G Y p_y - Z^T g \quad (16)$$

可能為 ill conditioned。若 Z 的行向量構成 A 之 null space 的 orthonormal 基底，則 $Z^T G Z$ 的條件數通常至少與 G 本身一樣好，但如此的 Z 一般只可能在中小型問題中取得。當 A 大且稀疏時，orthonormal 的 Z 難以計算，導致我們常被迫使用第 15 章中描述的不太可靠的 Z 。

§16.2 Direct Solution of the KKT System

很難對於 null-space 方法和 Schur-complement 方法的相對效果給出確切的規則，因為諸如計算 Z 的過程中填充等因素在相同維度的問題中也會存在顯著差異。一般來說，如果 G 是正定的且 $AG^{-1}A^T$ 可能因為 G 易於進行逆運算或是因為 m 相對於 n 為小而可以用較小的計算量求得時，則我們可以推薦使用 Schur-complement 方法。否則，通常更喜歡使用 null-space 方法，特別是當計算 G 的分解比計算基底矩陣 Z 以及 $Z^T G Z$ 的分解計算量更大時。

§16.2 Direct Solution of the KKT System

很難對於 null-space 方法和 Schur-complement 方法的相對效果給出確切的規則，因為諸如計算 Z 的過程中填充等因素在相同維度的問題中也會存在顯著差異。一般來說，如果 G 是正定的且 $AG^{-1}A^T$ 可能因為 G 易於進行逆運算或是因為 m 相對於 n 為小而可以用較小的計算量求得時，則我們可以推薦使用 Schur-complement 方法。否則，通常更喜歡使用 null-space 方法，特別是當計算 G 的分解比計算基底矩陣 Z 以及 $Z^T G Z$ 的分解計算量更大時。

§16.3 Iterative Solution of the KKT System

在前一節討論的直接分解技術的替代方法是使用迭代方法來解 KKT 系統 (4)。迭代方法適用於解決非常大的系統，並且通常適於平行化。共軛梯度 (CG) 方法不建議用於解決原始的完整系統 (4)，因為在非正定系統上可能不穩定。更好的選擇是 Krylov 方法，適用於一般的線性或對稱不定系統。候選方法包括 GMRES、QMR 和 LSQR 方法；請參見章末的附註和參考資料。其他迭代方法可以從 null-space 方法中衍生出來，通過將共軛梯度方法應用於縮減系統 (16)。這類方法是第 18 和 19 章算法的關鍵，並在本節的其餘部分進行討論。在接下來的討論中，我們始終假設 $Z^T G Z$ 是正定的。

§16.3 Iterative Solution of the KKT System

• CG applied to the reduced system

我們開始討論

$$\min_x q(x) \equiv \frac{1}{2}x^T Gx + x^T c \quad \text{subject to} \quad Ax = b \quad (2)$$

的迭代 null-space 方法。首先將 (2) 的解表示為

$$x_* = Yx_Y + Zx_Z, \quad (18)$$

其中 $x_Z \in \mathbb{R}^{n-m}$, $x_Y \in \mathbb{R}^m$ 為待求向量，而限制式 $Ax = b$ 可推得

$$AYx_Y = b, \quad (19)$$

這確定了向量 x_Y 。在第 15 章描述了各種實用的 Y 的選擇，其中的一些選擇可讓求解 (19) 較為經濟（計算量較小）。將 (18) 代入 (2) 可得 x_Z 為無受限縮減問題

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

的解，其中

$$c_Z = Z^T GYx_Y + Z^T c.$$

§16.3 Iterative Solution of the KKT System

• CG applied to the reduced system

我們開始討論

$$\min_x q(x) \equiv \frac{1}{2}x^T Gx + x^T c \quad \text{subject to} \quad Ax = b \quad (2)$$

的迭代 null-space 方法。首先將 (2) 的解表示為

$$x_* = Yx_Y + Zx_Z, \quad (18)$$

其中 $x_Z \in \mathbb{R}^{n-m}$, $x_Y \in \mathbb{R}^m$ 為待求向量，而限制式 $Ax = b$ 可推得

$$AYx_Y = b, \quad (19)$$

這確定了向量 x_Y 。在第 15 章描述了各種實用的 Y 的選擇，其中的一些選擇可讓求解 (19) 較為經濟（計算量較小）。將 (18) 代入 (2) 可得 x_Z 為無受限縮減問題

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

的解，其中

$$c_Z = Z^T GYx_Y + Z^T c.$$

§16.3 Iterative Solution of the KKT System

此無受限優化問題的解 x_z 滿足線性系統

$$Z^T G Z x_z = -c_z. \quad (20)$$

由於 $Z^T G Z$ 是正定的，我們可以將 CG 方法應用於這個線性系統，並將 x_z 代入 (18) 中以求得 (2) 的解。

正如第 5 章中所討論的，preconditioning 可以提高 CG 迭代的收斂速度，因此我們假設已經給定一個 preconditioner W_{zz} - 此 preconditioner W_{zz} 是個使 $W_{zz}^{-1/2}(Z^T G Z)W_{zz}^{-1/2}$ 的最大與最小特徵值的比值接近 1 (或說是條件數接近 1) 的 $(n-m) \times (n-m)$ 對稱正定矩陣。應用於 (20) 的 preconditioned CG 方法 (Algorithm 5.3) 如下頁所示 - 在此我們用 d_z 表示 CG 迭代生成的步驟。

§16.3 Iterative Solution of the KKT System

此無受限優化問題的解 x_z 滿足線性系統

$$Z^T G Z x_z = -c_z. \quad (20)$$

由於 $Z^T G Z$ 是正定的，我們可以將 CG 方法應用於這個線性系統，並將 x_z 代入 (18) 中以求得 (2) 的解。

正如第 5 章中所討論的，preconditioning 可以提高 CG 迭代的收斂速度，因此我們假設已經給定一個 preconditioner W_{zz} - 此 preconditioner W_{zz} 是個使 $W_{zz}^{-1/2}(Z^T G Z)W_{zz}^{-1/2}$ 的最大與最小特徵值的比值接近 1 (或說是條件數接近 1) 的 $(n-m) \times (n-m)$ 對稱正定矩陣。應用於 (20) 的 preconditioned CG 方法 (Algorithm 5.3) 如下頁所示 - 在此我們用 d_z 表示 CG 迭代生成的步驟。

Preconditioned CG from Chapter 5

Algorithm 5.3 (Preconditioned CG)

Given x_0 , preconditioner M ;

Set $r_0 \leftarrow Ax_0 - b$, solve $My_0 = r_0$ for y_0 ;

Set $p_0 = -y_0$, $k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k}; \quad (39a)_5$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \quad (39b)_5$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \quad (39c)_5$$

$$\text{Solve } My_{k+1} = r_{k+1}; \quad (39d)_5$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}; \quad (39e)_5$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k; \quad (39e)_5$$

$$k \leftarrow k + 1; \quad (39f)_5$$

end (while)

§16.3 Iterative Solution of the KKT System

Algorithm 16.1 (Preconditioned CG for Reduced Systems).

Choose an initial point x_z ;

Compute $r_z = Z^T G Z x_z + c_z$, $g_z = W_{zz}^{-1} r_z$, and $d_z = -g_z$;

repeat

$$\alpha \leftarrow r_z^T g_z / d_z^T Z^T G Z d_z;$$

$$x_z \leftarrow x_z + \alpha d_z;$$

$$r_z^+ \leftarrow r_z + \alpha Z^T G Z d_z;$$

$$g_z^+ \leftarrow W_{zz}^{-1} r_z^+;$$

$$\beta \leftarrow (r_z^+)^T g_z^+ / r_z^T g_z;$$

$$d_z \leftarrow -g_z^+ + \beta d_z;$$

$$g_z \leftarrow g_z^+; \quad r_z \leftarrow r_z^+;$$

until a termination test is satisfied.

可行的停止法則之一：當 $r_z^T W_{zz}^{-1} r_z$ 足夠小時終止迭代。

§16.3 Iterative Solution of the KKT System

在這個方法中，並不需要真正地形成 reduced-Hessian $Z^T G Z$ ，因為 CG 方法只需要我們計算涉及該矩陣的矩陣與向量乘積。實際上，只要我們能夠計算 Z 和 Z^T 與任意向量的乘積，就不需要 Z 的完整表示。對於某些 Z 的選擇，這些乘積的計算量比算 Z 本身要少得多，正如我們在第 15 章中所見。

理想的 preconditioner 會使得 $W_{ZZ}^{-1/2}(Z^T G Z)W_{ZZ}^{-1/2} = I$ ，亦即 $W_{ZZ} = Z^T G Z$ 。在這個理想的基礎上，我們考慮形如

$$W_{ZZ} = Z^T H Z \quad (21)$$

的 preconditioner，其中 H 是一個使得 $Z^T H Z$ 為正定矩陣的對稱矩陣。下面我們將討論一些 H 的選擇。形如 (21) 的 preconditioner 允許我們在 n 維空間應用 CG 方法，接下來我們將進一步討論。

§16.3 Iterative Solution of the KKT System

在這個方法中，並不需要真正地形成 reduced-Hessian $Z^T G Z$ ，因為 CG 方法只需要我們計算涉及該矩陣的矩陣與向量乘積。實際上，只要我們能夠計算 Z 和 Z^T 與任意向量的乘積，就不需要 Z 的完整表示。對於某些 Z 的選擇，這些乘積的計算量比算 Z 本身要少得多，正如我們在第 15 章中所見。

理想的 preconditioner 會使得 $W_{ZZ}^{-1/2}(Z^T G Z)W_{ZZ}^{-1/2} = I$ ，亦即 $W_{ZZ} = Z^T G Z$ 。在這個理想的基礎上，我們考慮形如

$$W_{ZZ} = Z^T H Z \quad (21)$$

的 preconditioner，其中 H 是一個使得 $Z^T H Z$ 為正定矩陣的對稱矩陣。下面我們將討論一些 H 的選擇。形如 (21) 的 preconditioner 允許我們在 n 維空間應用 CG 方法，接下來我們將進一步討論。

§16.3 Iterative Solution of the KKT System

- The projected CG method

如果使用形如

$$W_{ZZ} = Z^T H Z \quad (21)$$

的 preconditioner 和 $Ax = b$ 的特解，我們可以修改 Algorithm 16.1 設計一種無需操作基底矩陣 Z 的方法。這種方法不受 A 是 ill-conditioned 或 Z 選擇不佳的影響。

為了寫下此形式的 CG 演算法，我們定義了 scaled $n \times n$ 投影矩陣 P 如下：

$$P = Z(Z^T H Z)^{-1} Z^T, \quad (22)$$

其中 H 是形如 (21) 的 preconditioner 中的 preconditioning matrix。在 n 維空間中，CG 迭代如下頁所述。

§16.3 Iterative Solution of the KKT System

Algorithm 16.2 (Projected CG Method).

Choose an initial point x satisfying $Ax = b$;

Compute $r = Gx + c$, $g = Pr$, and $d = -g$;

repeat

$$\alpha \leftarrow r^T g / d^T G d; \quad (23a)$$

$$x \leftarrow x + \alpha d; \quad (23b)$$

$$r^+ \leftarrow r + \alpha G d; \quad (23c)$$

$$g^+ \leftarrow P r^+; \quad (23d)$$

$$\beta \leftarrow (r^+)^T g^+ / r^T g; \quad (23e)$$

$$d \leftarrow -g^+ + \beta d; \quad (23f)$$

$$g \leftarrow g^+; \quad r \leftarrow r^+; \quad (23g)$$

until a convergence test is satisfied.

一個實用的停止法則是在 $r^T g = r^T P r$ 夠小時終止迭代。

§16.3 Iterative Solution of the KKT System

在 Algorithm 16.2 中出現的 g^+ (或是 g) 被稱為 preconditioned residual。由於 g^+ 在 A 的 null space 中。因此，由 Algorithm 16.2 生成的所有搜索方向 d 也都位於 A 的 null space 中，因此迭代 x 都滿足 $Ax = b$ 。如果 $Z^T G Z$ 和 $Z^T H Z$ 都是正定的，那麼 Algorithm 16.2 中的迭代是 well-defined (意即在 (23a) 與 (23e) 中的分母在未達停止法則前都非零)。我們也可以通過

$$x_* = Yx_Y + Zx_Z \quad (18)$$

來驗證由 Algorithm 16.2 生成的迭代 x 與 Algorithm 16.1 的迭代 x_Z 之間的關係。

§16.3 Iterative Solution of the KKT System

在 Algorithm 16.2 中出現的 g^+ (或是 g) 被稱為 preconditioned residual。由於 g^+ 在 A 的 null space 中。因此，由 Algorithm 16.2 生成的所有搜索方向 d 也都位於 A 的 null space 中，因此迭代 x 都滿足 $Ax = b$ 。如果 $Z^T G Z$ 和 $Z^T H Z$ 都是正定的，那麼 Algorithm 16.2 中的迭代是 well-defined (意即在 (23a) 與 (23e) 中的分母在未達停止法則前都非零)。我們也可以通過

$$x_* = Yx_Y + Zx_Z \quad (18)$$

來驗證由 Algorithm 16.2 生成的迭代 x 與 Algorithm 16.1 的迭代 x_Z 之間的關係。

§16.3 Iterative Solution of the KKT System

在 Algorithm 16.2 中出現的 g^+ (或是 g) 被稱為 preconditioned residual。由於 g^+ 在 A 的 null space 中。因此，由 Algorithm 16.2 生成的所有搜索方向 d 也都位於 A 的 null space 中，因此迭代 x 都滿足 $Ax = b$ 。如果 $Z^T G Z$ 和 $Z^T H Z$ 都是正定的，那麼 Algorithm 16.2 中的迭代是 well-defined (意即在 (23a) 與 (23e) 中的分母在未達停止法則前都非零)。我們也可以通過

$$x_* = Yx_Y + Zx_Z \quad (18)$$

來驗證由 Algorithm 16.2 生成的迭代 x 與 Algorithm 16.1 的迭代 x_Z 之間的關係。

§16.3 Iterative Solution of the KKT System

在 Algorithm 16.2 中 null-space 基底矩陣 Z 僅出現在投影矩陣

$$P = Z(Z^T H Z)^{-1} Z^T \quad (22)$$

中，然而我們的目標是不需要用到基底矩陣 Z 的完整表示而計算 Pr^+ 。以下我們針對

- ① $H = I$;
- ② H 的選擇使得 preconditioner $W_{zz} \equiv Z^T H Z$ 正定；

這兩個情況來討論如何在不知 Z 的確切形式下計算 Pr^+ 。一般來說， $H = I$ 和 $H = \text{diag}(|G_{ii}|)$ 是 preconditioning matrix H 的兩個簡單選擇，但在某些應用中，將 H 定義為 G 的 block diagonal 子矩陣也是可行的。

§16.3 Iterative Solution of the KKT System

在 Algorithm 16.2 中 null-space 基底矩陣 Z 僅出現在投影矩陣

$$P = Z(Z^T H Z)^{-1} Z^T \quad (22)$$

中，然而我們的目標是不需要用到基底矩陣 Z 的完整表示而計算 Pr^+ 。以下我們針對

- ① $H = I$;
- ② H 的選擇使得 preconditioner $W_{zz} \equiv Z^T H Z$ 正定；

這兩個情況來討論如何在不知 Z 的確切形式下計算 Pr^+ 。一般來說， $H = I$ 和 $H = \text{diag}(|G_{ii}|)$ 是 preconditioning matrix H 的兩個簡單選擇，但在某些應用中，將 H 定義為 G 的 block diagonal 子矩陣也是可行的。

§16.3 Iterative Solution of the KKT System

首先考慮 $H = I$ 的情況，並以 P_1 來表示這種特殊情況的 P 。這樣 P_1 就是 A 的 null space 的正交投影，即

$$P_1 = Z(Z^T Z)^{-1} Z^T.$$

在 (23d) 式中，preconditioned residual $g^+ = P_1 r^+$ 的計算可以通過兩種方式進行。第一種方式是通過等價公式來表示 P_1 ，即

$$P_1 = I - A^T(AA^T)^{-1}A.$$

由此，我們可以計算 $g^+ = P_1 r^+$ ：我們可以將 g^+ 表示成 $g^+ = r^+ - A^T v^+$ ，其中 v^+ 是系統

$$AA^T v^+ = A r^+ \quad (24)$$

的解。這種計算投影 $g^+ = P_1 r^+$ 的方法稱為 normal equation 方法，而系統 (24) 可以通過對 AA^T 進行 Cholesky 分解來求解。

§16.3 Iterative Solution of the KKT System

首先考慮 $H = I$ 的情況，並以 P_I 來表示這種特殊情況的 P 。這樣 P_I 就是 A 的 null space 的正交投影，即

$$P_I = Z(Z^T Z)^{-1} Z^T.$$

在 (23d) 式中，preconditioned residual $g^+ = P_I r^+$ 的計算可以通過兩種方式進行。第一種方式是通過等價公式來表示 P_I ，即

$$P_I = I - A^T(AA^T)^{-1}A.$$

由此，我們可以計算 $g^+ = P_I r^+$ ：我們可以將 g^+ 表示成 $g^+ = r^+ - A^T v^+$ ，其中 v^+ 是系統

$$AA^T v^+ = Ar^+ \quad (24)$$

的解。這種計算投影 $g^+ = P_I r^+$ 的方法稱為 normal equation 方法，而系統 (24) 可以通過對 AA^T 進行 Cholesky 分解來求解。

§16.3 Iterative Solution of the KKT System

首先考慮 $H = I$ 的情況，並以 P_I 來表示這種特殊情況的 P 。這樣 P_I 就是 A 的 null space 的正交投影，即

$$P_I = Z(Z^T Z)^{-1} Z^T.$$

在 (23d) 式中，preconditioned residual $g^+ = P_I r^+$ 的計算可以通過兩種方式進行。第一種方式是通過等價公式來表示 P_I ，即

$$P_I = I - A^T(AA^T)^{-1}A.$$

由此，我們可以計算 $g^+ = P_I r^+$ ：我們可以將 g^+ 表示成 $g^+ = r^+ - A^T v^+$ ，其中 v^+ 是系統

$$AA^T v^+ = Ar^+ \tag{24}$$

的解。這種計算投影 $g^+ = P_I r^+$ 的方法稱為 normal equation 方法，而系統 (24) 可以通過對 AA^T 進行 Cholesky 分解來求解。

§16.3 Iterative Solution of the KKT System

第二種方法是將

$$g^+ = Pr^+ = Z(Z^T Z)^{-1} Z^T r^+ = r^+ - A^T v^+$$

表示為擴充系統的解

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} g^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} r^+ \\ 0 \end{bmatrix}. \quad (25)$$

如前所述，這個系統可以通過 symmetric indefinite factorization 求解。我們將這種方法稱為擴充系統 (augmented system) 方法。

§16.3 Iterative Solution of the KKT System

現在我們假設 preconditioning matrix H 具有一般形式。

- ① 當 H 可逆時，我們可以使用公式

$$g^+ = Pr^+, \text{ where } P = H^{-1}(I - A^T(AH^{-1}A^T)^{-1}AH^{-1}) \quad (26)$$

計算 g^+ 。然而若 H^{-1} 沒有簡單形式時，(26) 可能不好用。

- ② 若 (H 不可逆但) 對於 A 的 null space 中的所有非零向量 z 都有 $z^THz \neq 0$ 時，我們可以將 g^+ 找作系統的解

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} g^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} r^+ \\ 0 \end{bmatrix}. \quad (27)$$

注意到 (27) 是 (25) 的一個推廣。

通過取 $H = G$ ，我們可以獲得一個“完美”的 preconditioner，但也可以選擇其它使得 Z^THZ 是正定的 H 。(27) 式中的矩陣通常被稱為 constraint preconditioner。

§16.3 Iterative Solution of the KKT System

現在我們假設 preconditioning matrix H 具有一般形式。

- ① 當 H 可逆時，我們可以使用公式

$$g^+ = Pr^+, \text{ where } P = H^{-1}(I - A^T(AH^{-1}A^T)^{-1}AH^{-1}) \quad (26)$$

計算 g^+ 。然而若 H^{-1} 沒有簡單形式時，(26) 可能不好用。

- ② 若 (H 不可逆但) 對於 A 的 null space 中的所有非零向量 z 都有 $z^THz \neq 0$ 時，我們可以將 g^+ 找作系統的解

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} g^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} r^+ \\ 0 \end{bmatrix}. \quad (27)$$

注意到 (27) 是 (25) 的一個推廣。

通過取 $H = G$ ，我們可以獲得一個“完美”的 preconditioner，但也可以選擇其它使得 Z^THZ 是正定的 H 。(27) 式中的矩陣通常被稱為 constraint preconditioner。

§16.3 Iterative Solution of the KKT System

前述計算投影的這些方法中，都沒有使用 null-space 基底矩陣 Z 而只需要進行涉及 A 的矩陣分解。更重要的是，所有這些分解都能幫助我們計算滿足 $Ax = b$ 的初始點。

- ① 計算 $g^+ = P_I r^+$ 的過程倚賴於從中我們可以計算出 $x = A^T(AA^T)^{-1}b$ 的 AA^T 的分解；
- ② 在解 (25) 和 (27) 時所用的矩陣分解使我們可以通過解

$$\begin{bmatrix} I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

找到一個合適的 x 。

因此，我們可以在執行投影運算所需的矩陣分解的基礎上進行一次反解而計算出 Algorithm 16.2 的初始點。不過，這些計算 g^+ 的方法可能會產生顯著的捨入誤差，因此建議使用 iterative refinement 來提高準確性。

§16.4 Inequality-Constrained Problems

在本章的剩餘部分，我們將討論幾類解同時包含不等式和等式限制的凸 QP 的演算法。

- 1 自 1970 年代以來，active-set 方法已被廣泛使用，對於小型和中型問題非常有效。它們允許有效地檢測問題的無界性和不可行性，並且通常給出對最佳 active set 的準確估計。
- 2 內點法是近年來才變得流行，主要流行於 1990 年代。它們非常適合解決大型問題，但在必須解決一系列相關的 QP 問題時可能不是最有效的。

我們也將研究了一種被稱為梯度投影方法 (gradient projection method) 的特殊類型 active-set 方法，這種方法在問題中的唯一限制條件是給定變數的上下界時最為有效。

§16.4 Inequality-Constrained Problems

• Optimality conditions for inequality-constrained problems

進行進一步討論前，我們先簡要回顧不等式限制 QP 的最優條件，然後討論一些解的一些不太明顯的性質。

課本定理 12.1（即給出 KKT 條件的定理）可以應用於問題 (1)：

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \quad (1a)$$

subject to

$$a_i^T x = b_i \quad \text{if } i \in \mathcal{E}, \quad (1b)$$

$$a_i^T x \geq b_i \quad \text{if } i \in \mathcal{I}. \quad (1c)$$

注意到這個問題的 Lagrangian 為

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

Theorem 12.1 – KKT conditions

Theorem (First-Order Necessary Conditions)

Suppose that x_* is a local solution of problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geq 0, i \in \mathcal{I}, \end{cases} \quad (1)_{12}$$

that the functions f and c_i in (1)₁₂ are continuously differentiable, and that the LICQ holds at x_* . Then there is a Lagrange multiplier vector λ_* , with components λ_i^* , $i \in \mathcal{E} \cup \mathcal{I}$, such that the following conditions are satisfied.

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = 0, \quad (32a)_{12}$$

$$c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E}, \quad (32b)_{12}$$

$$c_i(x_*) \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32c)_{12}$$

$$\lambda_i^* \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32d)_{12}$$

$$\lambda_i^* c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}. \quad (32e)_{12}$$

§16.4 Inequality-Constrained Problems

通過將 KKT 條件 (32c)₁₂ 用於 (1) 上，我們發現任何 (1) 的解 x_* 都滿足以下的一階條件：存在 Lagrange 乘子 λ_i^* 使得

$$Gx_* + c - \sum_{i \in \mathcal{A}(x_*)} \lambda_i^* a_i = 0, \quad (28a)$$

$$a_i^T x_* = b_i \quad \text{for all } i \in \mathcal{A}(x_*), \quad (28b)$$

$$a_i^T x_* \geq b_i \quad \text{for all } i \in \mathcal{I} \setminus \mathcal{A}(x_*), \quad (28c)$$

$$\lambda_i^* \geq 0 \quad \text{for all } i \in \mathcal{I} \cap \mathcal{A}(x_*), \quad (28d)$$

其中 active set $\mathcal{A}(x_*)$ 是在 x_* 位置限制式等式成立的所有 index 所形成的集合，亦即

$$\mathcal{A}(x_*) = \{i \in \mathcal{E} \cup \mathcal{I} \mid a_i^T x_* = b_i\}. \quad (29)$$

§16.4 Inequality-Constrained Problems

一個技術上的細節是：在課本定理 12.1 中，我們假設了線性獨立限制資格條件 (LICQ) 得到滿足。如 §12.6 所述，如果我們將 LICQ 替換為其他限制資格條件，例如限制式為線性限制時，則該定理仍然成立。由於在 QP 中限制式皆為線性限制，因此在上面給出的 QP 最優條件中，我們無需假設在解處 active constraint gradients 是線性獨立的。

對於凸 QP，當 G 是半正定時，條件 (28) 實際上是 x_* 成為 global solution 的充分條件，如下定理所示。

Theorem

If x_ satisfies the conditions (28) for some λ_i^* , $i \in \mathcal{A}(x_*)$, and G is positive semi-definite, then x_* is a global solution of (1).*

也就是說，對於凸 QP 而言 KKT point 即為 global solution。

§16.4 Inequality-Constrained Problems

一個技術上的細節是：在課本定理 12.1 中，我們假設了線性獨立限制資格條件 (LICQ) 得到滿足。如 §12.6 所述，如果我們將 LICQ 替換為其他限制資格條件，例如限制式為線性限制時，則該定理仍然成立。由於在 QP 中限制式皆為線性限制，因此在上面給出的 QP 最優條件中，我們無需假設在解處 active constraint gradients 是線性獨立的。

對於凸 QP，當 G 是半正定時，條件 (28) 實際上是 x_* 成為 global solution 的充分條件，如下定理所示。

Theorem

If x_ satisfies the conditions (28) for some λ_i^* , $i \in \mathcal{A}(x_*)$, and G is positive semi-definite, then x_* is a global solution of (1).*

也就是說，對於凸 QP 而言 KKT point 即為 global solution。

§16.4 Inequality-Constrained Problems

Proof.

If x is any other feasible point for (1), we have that $a_i^T x = b_i$ for all $i \in \mathcal{E}$ and $a_i^T x \geq b_i$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Hence, $a_i^T (x - x_*) = 0$ for all $i \in \mathcal{E}$ and $a_i^T (x - x_*) \geq 0$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Using these relationships, together with (28a) and (28d), we have that

$$(x - x_*)^T (Gx_* + c) = \sum_{i \in \mathcal{E}} \lambda_i^* a_i^T (x - x_*) + \sum_{i \in \mathcal{A}(x_*) \cap \mathcal{I}} \lambda_i^* a_i^T (x - x_*) \geq 0. \quad (30)$$

By elementary manipulation, we find that

$$\begin{aligned} q(x) &= q(x_*) + (x - x_*)^T (Gx_* + c) + \frac{1}{2} (x - x_*)^T G (x - x_*) \\ &\geq q(x_*) + \frac{1}{2} (x - x_*)^T G (x - x_*) \geq q(x_*), \end{aligned}$$

where the first inequality follows from (30) and the second inequality follows from positive semi-definiteness of G . We have shown that $q(x) \geq q(x_*)$ for any feasible x , so x_* is a global solution. \square

§16.4 Inequality-Constrained Problems

Proof.

If x is any other feasible point for (1), we have that $a_i^T x = b_i$ for all $i \in \mathcal{E}$ and $a_i^T x \geq b_i$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Hence, $a_i^T (x - x_*) = 0$ for all $i \in \mathcal{E}$ and $a_i^T (x - x_*) \geq 0$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Using these relationships, together with (28a) and (28d), we have that

$$(x - x_*)^T (Gx_* + c) = \sum_{i \in \mathcal{E}} \lambda_i^* a_i^T (x - x_*) + \sum_{i \in \mathcal{A}(x_*) \cap \mathcal{I}} \lambda_i^* a_i^T (x - x_*) \geq 0. \quad (30)$$

By elementary manipulation, we find that

$$\begin{aligned} q(x) &= q(x_*) + (x - x_*)^T (Gx_* + c) + \frac{1}{2} (x - x_*)^T G (x - x_*) \\ &\geq q(x_*) + \frac{1}{2} (x - x_*)^T G (x - x_*) \geq q(x_*), \end{aligned}$$

where the first inequality follows from (30) and the second inequality follows from positive semi-definiteness of G . We have shown that $q(x) \geq q(x_*)$ for any feasible x , so x_* is a global solution. \square

§16.4 Inequality-Constrained Problems

Proof.

If x is any other feasible point for (1), we have that $a_i^T x = b_i$ for all $i \in \mathcal{E}$ and $a_i^T x \geq b_i$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Hence, $a_i^T (x - x_*) = 0$ for all $i \in \mathcal{E}$ and $a_i^T (x - x_*) \geq 0$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Using these relationships, together with (28a) and (28d), we have that

$$(x - x_*)^T (Gx_* + c) = \sum_{i \in \mathcal{E}} \lambda_i^* a_i^T (x - x_*) + \sum_{i \in \mathcal{A}(x_*) \cap \mathcal{I}} \lambda_i^* a_i^T (x - x_*) \geq 0. \quad (30)$$

By elementary manipulation, we find that

$$\begin{aligned} q(x) &= q(x_*) + (x - x_*)^T (Gx_* + c) + \frac{1}{2} (x - x_*)^T G (x - x_*) \\ &\geq q(x_*) + \frac{1}{2} (x - x_*)^T G (x - x_*) \geq q(x_*), \end{aligned}$$

where the first inequality follows from (30) and the second inequality follows from positive semi-definiteness of G . We have shown that $q(x) \geq q(x_*)$ for any feasible x , so x_* is a global solution. \square

§16.4 Inequality-Constrained Problems

Proof.

If x is any other feasible point for (1), we have that $a_i^T x = b_i$ for all $i \in \mathcal{E}$ and $a_i^T x \geq b_i$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Hence, $a_i^T (x - x_*) = 0$ for all $i \in \mathcal{E}$ and $a_i^T (x - x_*) \geq 0$ for all $i \in \mathcal{A}(x_*) \cap \mathcal{I}$. Using these relationships, together with (28a) and (28d), we have that

$$(x - x_*)^T (Gx_* + c) = \sum_{i \in \mathcal{E}} \lambda_i^* a_i^T (x - x_*) + \sum_{i \in \mathcal{A}(x_*) \cap \mathcal{I}} \lambda_i^* a_i^T (x - x_*) \geq 0. \quad (30)$$

By elementary manipulation, we find that

$$\begin{aligned} q(x) &= q(x_*) + (x - x_*)^T (Gx_* + c) + \frac{1}{2} (x - x_*)^T G (x - x_*) \\ &\geq q(x_*) + \frac{1}{2} (x - x_*)^T G (x - x_*) \geq q(x_*), \end{aligned}$$

where the first inequality follows from (30) and the second inequality follows from positive semi-definiteness of G . We have shown that $q(x) \geq q(x_*)$ for any feasible x , so x_* is a global solution. \square

§16.4 Inequality-Constrained Problems

通過對這個證明進行一個微不足道的修改，我們可以看出當 G 是正定時， x_* 實際上是唯一的 global solution。

我們也可以應用 §12.5 的理論來推導 (1) 的二階最優條件。令 Z 為 active constraint Jacobian 矩陣的 null-space 基底矩陣，亦即該矩陣的列是來自所有 $i \in \mathcal{A}(x_*)$ 所對應的 a_i^T 。則 x_* 為局部最小值的二階充分條件滿足於 $Z^T G Z$ 是正定時。在這種情況下，根據定理 12.6（見下頁複習）， x_* 是一個嚴格局部解。

當 G 不是正定時，問題 (1) 可能有不止一個嚴格局部解。如上所述，這樣的問題被稱為“非凸 QP”或“不定 QP”，它們對算法造成了一些困難。不定 QP 的範例如（下下頁的）圖 1 所示。

§16.4 Inequality-Constrained Problems

通過對這個證明進行一個微不足道的修改，我們可以看出當 G 是正定時， x_* 實際上是唯一的 global solution。

我們也可以應用 §12.5 的理論來推導 (1) 的二階最優條件。令 Z 為 active constraint Jacobian 矩陣的 null-space 基底矩陣，亦即該矩陣的列是來自所有 $i \in \mathcal{A}(x_*)$ 所對應的 a_i^T 。則 x_* 為局部最小值的二階充分條件滿足於 $Z^T G Z$ 是正定時。在這種情況下，根據定理 12.6（見下頁複習）， x_* 是一個嚴格局部解。

當 G 不是正定時，問題 (1) 可能有不止一個嚴格局部解。如上所述，這樣的問題被稱為“非凸 QP”或“不定 QP”，它們對算法造成了一些困難。不定 QP 的範例如（下下頁的）圖 1 所示。

Theorem 12.6 – Second-Order Sufficient Conditions

Theorem (Second-Order Sufficient Conditions)

Suppose that for some feasible point $x_* \in \mathbb{R}^n$ there is a Lagrange multiplier vector λ_* such that the KKT conditions

$$\nabla_x \mathcal{L}(x_*, \lambda_*) = 0, \quad (32a)_{12}$$

$$c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E}, \quad (32b)_{12}$$

$$c_i(x_*) \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32c)_{12}$$

$$\lambda_i^* \geq 0 \quad \text{for all } i \in \mathcal{I}, \quad (32d)_{12}$$

$$\lambda_i^* c_i(x_*) = 0 \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}. \quad (32e)_{12}$$

are satisfied. Suppose also that

$$w^T \nabla_{xx}^2 \mathcal{L}(x_*, \lambda_*) w > 0 \quad \text{for all } w \in \mathcal{C}(x_*, \lambda_*) \setminus \{0\}. \quad (60)_{12}$$

Then x_* is a strict local solution for

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geq 0, i \in \mathcal{I}. \end{cases} \quad (1)_{12}$$

§16.4 Inequality-Constrained Problems

通過對這個證明進行一個微不足道的修改，我們可以看出當 G 是正定時， x_* 實際上是唯一的 global solution。

我們也可以應用 §12.5 的理論來推導 (1) 的二階最優條件。令 Z 為 active constraint Jacobian 矩陣的 null-space 基底矩陣，亦即該矩陣的列是來自所有 $i \in \mathcal{A}(x_*)$ 所對應的 a_i^T 。則 x_* 為局部最小值的二階充分條件滿足於 $Z^T G Z$ 是正定時。在這種情況下，根據定理 12.6（見上頁複習）， x_* 是一個嚴格局部解。

當 G 不是正定時，問題 (1) 可能有不止一個嚴格局部解。如上所述，這樣的問題被稱為“非凸 QP”或“不定 QP”，它們對算法造成了一些困難。不定 QP 的範例如（下下頁的）圖 1 所示。

§16.4 Inequality-Constrained Problems

在（下頁）圖 1 的左圖中我們繪製了可行區域和二次目標函數 $q(x)$ 的等高線，其中 G 有一個正特徵值和一個負特徵值。我們用 \oplus 或 \ominus 表示函數在該方向上趨向正無窮大或負無窮大。注意 x^{**} 是一個局部 maximizer， x^* 是一個局部 minimizer，而方框的中心是一個 stationary point。在圖 1 的右圖中， G 的特徵值均為負，而 \tilde{x} 為全域 maximizer 以及兩個局部 minimizer x^* 和 x^{**} 。

§16.4 Inequality-Constrained Problems

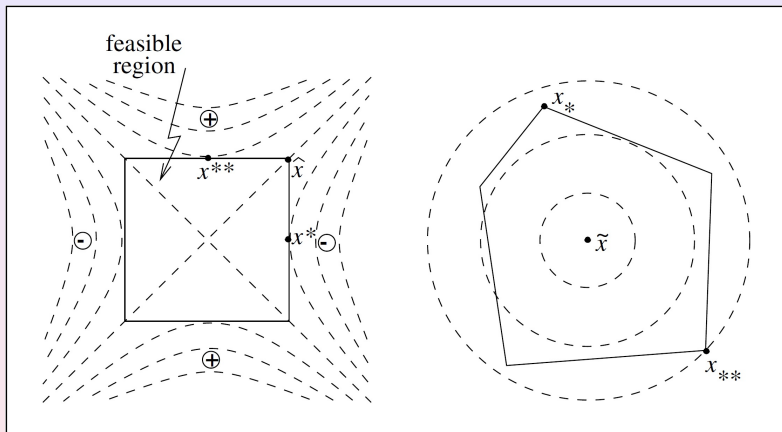


Figure 1: Non-convex quadratic programs.

§16.4 Inequality-Constrained Problems

• Degeneracy

導致一些算法困難的第二個性質是 degeneracy。Degeneracy 這個術語在不同的領域被賦予了各種不同的含義，我們所指的則是以下情況：

- 1 在解 x_* 處，active constraint 的梯度 a_i , $i \in \mathcal{A}(x_*)$ 是線性相依的，以及/或
- 2 定義 12.5 中的嚴格互補條件未能滿足；也就是說，存在某個指標 $i \in \mathcal{A}(x_*)$ ，使得滿足 (28) 的所有 Lagrange 乘子都有 $\lambda_i^* = 0$ （根據定義 12.8，這樣的限制式是 weakly active）。

§16.4 Inequality-Constrained Problems

(下頁) 圖 2 展示了兩個退化的例子。在圖 2 左側的圖中，在解 x_* 處有一個 active constraint，同時也是目標函數的 unconstrained minimizer。根據 (28a) 的符號，我們有 $Gx_* + c = 0$ ，因此獨立的 Lagrange 乘子必須為零。在圖 2 右側的圖中，解 x_* 處有三個限制式為 active。由於這三個 constraints 之梯度都是 \mathbb{R}^2 中的向量，它們必然線性相依。

§16.4 Inequality-Constrained Problems

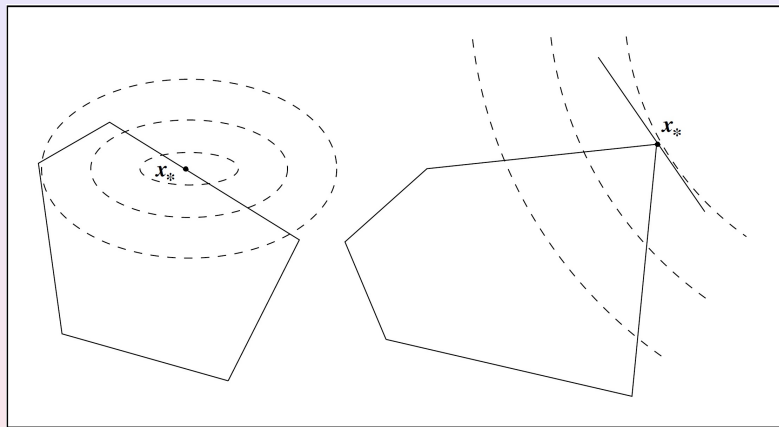


Figure 2: Degenerate solutions of quadratic programs.

§16.4 Inequality-Constrained Problems

缺乏嚴格互補性也可由問題

$$\min x_1^2 + (x_2 + 1)^2 \quad \text{subject to} \quad x \geq 0,$$

來說明，該問題在 $x_* = 0$ 處有解，且兩個限制式都是 active。然而，因為與 active constraint $x_1 \geq 0$ 相關聯的 Lagrange 乘子為零的緣故，在 x_* 處不滿足嚴格互補性。

Degeneracy 會因為兩個主要原因對算法造成問題。首先，active constraint gradient 的線性相依性可能導致步進量計算中的數值困難，因為我們需要分解的某些矩陣變得 rank deficient。其次，當問題包含 weakly active constraint 時，算法難以確定這些限制在解處是否 active。對於 active-set 方法和梯度投影方法（如下所述），這種猶豫不決可能導致算法在迭代過程中在 weakly active constraint 上來回擺動。必須使用安全措施來防止這種行為。

§16.5 Active-Set Methods for Convex QPs

接下來我們描述解決包含等式和不等式限制的二次規劃問題 (1) 的 active-set 方法。我們僅考慮凸 QP 的情況，即在 (1a) 中矩陣 G 是半正定的情況。如果 G 是一個不定矩陣，則會對算法製造複雜性，並且超出了本書的範圍。關於非凸 QP 的討論，可以參考 Gould [147]。

如果最優 active set (29) 的內容事先已知，我們可以通過將問題

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \quad \text{subject to} \quad a_i^T x = b_i, i \in \mathcal{A}(x_*)$$

應用於在 §16.2 和 16.3 中描述的等式限制 QP 的一種技術來找到解 x_* 。

一般而言，我們不會事先知道 $\mathcal{A}(x_*)$ 。事實上，確定這個集合是面對不等式限制 QP 的算法所面臨的主要挑戰。

§16.5 Active-Set Methods for Convex QPs

在第 13 章中，我們已經遇到了線性規劃的一種 active-set 方法，即 simplex 法。基本上，simplex 法從對最優活動集的猜測開始，然後重複使用梯度和 Lagrange 乘子信息來從當前對 $\mathcal{A}(x_*)$ 的估計中刪除一個索引並添加一個新的索引，直到檢測到最優性。對於 QP 的 active-set 方法與 simplex 法不同之處在於，迭代點（以及解 x_* ）不一定是可行區域的頂點。

§16.5 Active-Set Methods for Convex QPs

QP 的 active-set 方法有三種類型：primal, dual 以及 primal-dual。我們僅討論 primal 方法，此方法除了生成原問題 (1) 的可行迭代點外，也同時不斷地降低目標函數 q 的函數值。

Primal active-set 方法通過解決一個二次子問題來找到一個「從一個迭代到下一個迭代」的步進量，在這個二次子問題中，一些不等式限制 (1c) 和所有的等式限制 (1b) 被作為等式限制加入。這些收集等式限制的 index 的集合被稱為工作集，並在第 k 次迭代中被表示為 \mathcal{W}_k 。我們對 \mathcal{W}_k 加諸的一個重要要求是在工作集中的所有 index i 所對應的限制式梯度 a_i 具有線性獨立性，亦即

$\{a_i \mid i \in \mathcal{W}_k\}$ 為線性獨立集。

§16.5 Active-Set Methods for Convex QPs

給定一個迭代點 x_k 和工作集 \mathcal{W}_k ，我們首先檢查 x_k 是否在由工作集所定義的子空間中最小化二次函數 q 。如果不是，我們通過解決一個只具等式限制的 QP 子問題來計算步進量 p ，在這個子問題中，與工作集 \mathcal{W}_k 相對應的限制式被視為等式限制，而其它限制則暫時被忽略。將 $x = x_k + p$ 代入目標函數 (1a) 中得到

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

其中 $g_k = Gx_k + c$ 且 $\rho_k = \frac{1}{2}x_k^T G x_k + c^T x_k$ 。由於 ρ_k 與 p 無關，我們可以在不改變問題的解的情況下扔掉 ρ_k 因而將在第 k 次迭代中要解的 QP 子問題寫成如下形式：

$$\min_p \frac{1}{2}p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

§16.5 Active-Set Methods for Convex QPs

給定一個迭代點 x_k 和工作集 \mathcal{W}_k ，我們首先檢查 x_k 是否在由工作集所定義的子空間中最小化二次函數 q 。如果不是，我們通過解決一個只具等式限制的 QP 子問題來計算步進量 p ，在這個子問題中，與工作集 \mathcal{W}_k 相對應的限制式被視為等式限制，而其它限制則暫時被忽略。將 $x = x_k + p$ 代入目標函數 (1a) 中得到

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

其中 $g_k = Gx_k + c$ 且 $\rho_k = \frac{1}{2}x_k^T G x_k + c^T x_k$ 。由於 ρ_k 與 p 無關，我們可以在不改變問題的解的情況下扔掉 ρ_k 因而將在第 k 次迭代中要解的 QP 子問題寫成如下形式：

$$\min_p \frac{1}{2}p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

§16.5 Active-Set Methods for Convex QPs

我們將子問題

$$\min_p \frac{1}{2} p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

的解表示為 p_k 。注意到因為

$$a_i^T (x_k + \alpha p_k) = a_i^T x_k = b_i \quad \forall \alpha,$$

我們發現對所有的 α 而言，對應於工作集 \mathcal{W}_k 的限制在 $x_k + \alpha p_k$ 處也得到滿足。由於 G 是正定的，因此可以通過 §16.2 中描述的任何技術來計算 (31) 的解。另外我們已假設 x_k 並非由工作集所定義的子空間中最小化二次函數 q 的點，因此從 (31) 中得到的最優解 p_k 必是非零的。

§16.5 Active-Set Methods for Convex QPs

我們將子問題

$$\min_p \frac{1}{2} p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

的解表示為 p_k 。注意到因為

$$a_i^T (x_k + \alpha p_k) = a_i^T x_k = b_i \quad \forall \alpha,$$

我們發現對所有的 α 而言，對應於工作集 \mathcal{W}_k 的限制在 $x_k + \alpha p_k$ 處也得到滿足。由於 G 是正定的，因此可以通過 §16.2 中描述的任何技術來計算 (31) 的解。另外我們已假設 x_k 並非由工作集所定義的子空間中最小化二次函數 q 的點，因此從 (31) 中得到的最優解 p_k 必是非零的。

§16.5 Active-Set Methods for Convex QPs

接下來我們需要決定沿著 p_k 方向移動多遠。如果 $x_k + p_k$ 滿足所有限制式，我們設置 $x_{k+1} = x_k + p_k$ 。否則，我們設置

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

其中步長參數 α_k 被選擇為在範圍 $[0, 1]$ 中滿足所有限制的最大值。因為對應於 $i \in \mathcal{W}_k$ 的限制式都會被滿足，我們只需考慮 $i \notin \mathcal{W}_k$ 所對應的限制來求得 α_k 。

- 1 如果 $i \notin \mathcal{W}_k$ 所對應的 a_i 滿足 $a_i^T p_k \geq 0$ ，那麼對於所有 $\alpha_k \geq 0$ ，我們有 $a_i^T (x_k + \alpha_k p_k) \geq a_i^T x_k \geq b_i$ 。因此，對於所有非負的步長參數的選擇，限制 i 將被滿足。
- 2 如果 $i \notin \mathcal{W}_k$ 所對應的 a_i 滿足 $a_i^T p_k < 0$ ，則只有當

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

才有 $a_i^T (x_k + \alpha_k p_k) \geq b_i$ 。

§16.5 Active-Set Methods for Convex QPs

接下來我們需要決定沿著 p_k 方向移動多遠。如果 $x_k + p_k$ 滿足所有限制式，我們設置 $x_{k+1} = x_k + p_k$ 。否則，我們設置

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

其中步長參數 α_k 被選擇為在範圍 $[0, 1]$ 中滿足所有限制的最大值。因為對應於 $i \in \mathcal{W}_k$ 的限制式都會被滿足，我們只需考慮 $i \notin \mathcal{W}_k$ 所對應的限制來求得 α_k 。

- 1 如果 $i \notin \mathcal{W}_k$ 所對應的 a_i 滿足 $a_i^T p_k \geq 0$ ，那麼對於所有 $\alpha_k \geq 0$ ，我們有 $a_i^T (x_k + \alpha_k p_k) \geq a_i^T x_k \geq b_i$ 。因此，對於所有非負的步長參數的選擇，限制 i 將被滿足。
- 2 如果 $i \notin \mathcal{W}_k$ 所對應的 a_i 滿足 $a_i^T p_k < 0$ ，則只有當

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

才有 $a_i^T (x_k + \alpha_k p_k) \geq b_i$ 。

§16.5 Active-Set Methods for Convex QPs

為了最大化 q 函數值的遞減量，我們希望 α_k 在保持可行性的情況下在 $[0, 1]$ 中盡可能大，因此我們得到了以下定義：

$$\alpha_k \equiv \min \left\{ 1, \min_{i \notin \mathcal{W}_k, a_i^T p_k < 0} \frac{b_i - a_i^T x_k}{a_i^T p_k} \right\}. \quad (32)$$

我們將 (32) 中取得最小值所對應的限制 i 稱為 blocking constraint (如果 $\alpha_k = 1$ ，且在 $x_k + \alpha_k p_k$ 處沒有新的限制變成 active，則在這次迭代中就沒有 blocking constraint)。注意， α_k 可能為零，因為對於某些不是當前工作集 \mathcal{W}_k 的成員的限制 i ，我們可能有 $a_i^T x_k = b_i$ 且 $a_i^T p_k < 0$ 。

§16.5 Active-Set Methods for Convex QPs

為了最大化 q 函數值的遞減量，我們希望 α_k 在保持可行性的情況下在 $[0, 1]$ 中盡可能大，因此我們得到了以下定義：

$$\alpha_k \equiv \min \left\{ 1, \min_{i \notin \mathcal{W}_k, a_i^T p_k < 0} \frac{b_i - a_i^T x_k}{a_i^T p_k} \right\}. \quad (32)$$

我們將 (32) 中取得最小值所對應的限制 i 稱為 blocking constraint (如果 $\alpha_k = 1$ ，且在 $x_k + \alpha_k p_k$ 處沒有新的限制變成 active，則在這次迭代中就沒有 blocking constraint)。注意， α_k 可能為零，因為對於某些不是當前工作集 \mathcal{W}_k 的成員的限制 i ，我們可能有 $a_i^T x_k = b_i$ 且 $a_i^T p_k < 0$ 。

§16.5 Active-Set Methods for Convex QPs

如果 $\alpha_k < 1$ ，亦即沿着 p_k 的步驟被一些不在 \mathcal{W}_k 中的限制所阻擋，那麼一個新的工作集 \mathcal{W}_{k+1} 就會通過將該迭代步驟的 blocking constraint 添加到 \mathcal{W}_k 中來構建。我們繼續以這種方式迭代，將限制添加到工作集中，直到達到一個在其當前工作集 $\widehat{\mathcal{W}}$ 上最小化二次目標函數的點 \hat{x} 。很容易識別這樣的一個點，因為此迭代所考慮的子問題 (31) 的解為 $p = 0$ 。

在 \hat{x} 點由於 $p = 0$ 滿足子問題 (31) 的最優條件 (4)，我們有

$$\sum_{i \in \widehat{\mathcal{W}}} a_i \hat{\lambda}_i = g = G\hat{x} + c, \quad (33)$$

其中對於 $\widehat{\mathcal{W}}$ 中的 i 而言 $\hat{\lambda}_i$ 為 Lagrange 乘子。如果將不在工作集中的不等式限制所對應的乘子定為零，則 \hat{x} 和 $\hat{\lambda}$ 滿足 KKT 條件 (28a)。由於對步長的控制， \hat{x} 必滿足所有的限制，因此在 \hat{x} 上，KKT 條件 (28b) 和 (28c) 也都會得到滿足。

§16.5 Active-Set Methods for Convex QPs

如果 $\alpha_k < 1$ ，亦即沿着 p_k 的步驟被一些不在 \mathcal{W}_k 中的限制所阻擋，那麼一個新的工作集 \mathcal{W}_{k+1} 就會通過將該迭代步驟的 blocking constraint 添加到 \mathcal{W}_k 中來構建。我們繼續以這種方式迭代，將限制添加到工作集中，直到達到一個在其當前工作集 $\widehat{\mathcal{W}}$ 上最小化二次目標函數的點 \hat{x} 。很容易識別這樣的一個點，因為此迭代所考慮的子問題 (31) 的解為 $p = 0$ 。

在 \hat{x} 點由於 $p = 0$ 滿足子問題 (31) 的最優條件 (4)，我們有

$$\sum_{i \in \widehat{\mathcal{W}}} a_i \hat{\lambda}_i = g = G\hat{x} + c, \quad (33)$$

其中對於 $\widehat{\mathcal{W}}$ 中的 i 而言 $\hat{\lambda}_i$ 為 Lagrange 乘子。如果將不在工作集中的不等式限制所對應的乘子定為零，則 \hat{x} 和 $\hat{\lambda}$ 滿足 KKT 條件 (28a)。由於對步長的控制， \hat{x} 必滿足所有的限制，因此在 \hat{x} 上，KKT 條件 (28b) 和 (28c) 也都會得到滿足。

§16.5 Active-Set Methods for Convex QPs

接下來我們來檢查 $\widehat{W} \cap \mathcal{I}$ 中之 index i 所對應於不等式限制的乘子的符號。如果這些乘子都非負，則 KKT 條件 (28d) 也得到滿足，因此可以得出 \hat{x} 是原問題 (1) 的 KKT 點的結論。事實上，由於 G 是正半定的，根據前述定理我們得知 \hat{x} 是 (1) 的 global solution (如果 G 是正定的，則 \hat{x} 是唯一的 global solution)。

另一方面，如果某些 $j \in \widehat{W} \cap \mathcal{I}$ 對應的乘子 $\hat{\lambda}_j$ 是負的，則 KKT 條件 (28d) 不被滿足，目標函數 q 便可能通過刪除其中一個這些限制來降低，如在 §12.3 所示。因此，我們從工作集中刪除一個對應於負乘子的指標 j ，並為新的迭代步驟解決一個新的子問題 (31)。接下來的定理將展示這個策略產生的方向 p 在下次迭代中是對於被刪除的限制是可行的。我們繼續假設工作集中的限制式梯度 a_i 對於工作集中的 i 是線性獨立的。在完整陳述算法之後，我們將討論如何保持這個特性。

§16.5 Active-Set Methods for Convex QPs

接下來我們來檢查 $\widehat{W} \cap \mathcal{I}$ 中之 index i 所對應於不等式限制的乘子的符號。如果這些乘子都非負，則 KKT 條件 (28d) 也得到滿足，因此可以得出 \hat{x} 是原問題 (1) 的 KKT 點的結論。事實上，由於 G 是正半定的，根據前述定理我們得知 \hat{x} 是 (1) 的 global solution (如果 G 是正定的，則 \hat{x} 是唯一的 global solution)。

另一方面，如果某些 $j \in \widehat{W} \cap \mathcal{I}$ 對應的乘子 $\hat{\lambda}_j$ 是負的，則 KKT 條件 (28d) 不被滿足，目標函數 q 便可能通過刪除其中一個這些限制來降低，如在 §12.3 所示。因此，我們從工作集中刪除一個對應於負乘子的指標 j ，並為新的迭代步驟解決一個新的子問題 (31)。接下來的定理將展示這個策略產生的方向 p 在下次迭代中是對於被刪除的限制是可行的。我們繼續假設工作集中的限制式梯度 a_i 對於工作集中的 i 是線性獨立的。在完整陳述算法之後，我們將討論如何保持這個特性。

§16.5 Active-Set Methods for Convex QPs

接下來我們來檢查 $\widehat{\mathcal{W}} \cap \mathcal{I}$ 中之 index i 所對應於不等式限制的乘子的符號。如果這些乘子都非負，則 KKT 條件 (28d) 也得到滿足，因此可以得出 \hat{x} 是原問題 (1) 的 KKT 點的結論。事實上，由於 G 是正半定的，根據前述定理我們得知 \hat{x} 是 (1) 的 global solution (如果 G 是正定的，則 \hat{x} 是唯一的 global solution)。

另一方面，如果某些 $j \in \widehat{\mathcal{W}} \cap \mathcal{I}$ 對應的乘子 $\hat{\lambda}_j$ 是負的，則 KKT 條件 (28d) 不被滿足，目標函數 q 便可能通過刪除其中一個這些限制來降低，如在 §12.3 所示。因此，我們從工作集中刪除一個對應於負乘子的指標 j ，並為新的迭代步驟解決一個新的子問題 (31)。接下來的定理將展示這個策略產生的方向 p 在下次迭代中是對於被刪除的限制是可行的。我們繼續假設工作集中的限制式梯度 a_i 對於工作集中的 i 是線性獨立的。在完整陳述算法之後，我們將討論如何保持這個特性。

§16.5 Active-Set Methods for Convex QPs

Theorem

Let \hat{x} satisfy the first-order conditions

$$\sum_{i \in \widehat{W}} a_i \hat{\lambda}_i = G\hat{x} + c \quad (33)$$

for some $\hat{\lambda}$, and $a_i^T \hat{x} = b_i$ for all $i \in \widehat{W}$. Suppose that **there is an index $j \in \widehat{W} \cap \mathcal{I}$ such that $\hat{\lambda}_j < 0$** . Let p be the solution obtained by dropping the constraint j and solving the following sub-problem:

$$\min_p \frac{1}{2} p^T G p + (G\hat{x} + c)^T p \quad (34a)$$

subject to

$$a_i^T p = 0 \quad \text{for all } i \in \widehat{W} \text{ with } i \neq j. \quad (34b)$$

If constraint gradients $\{a_i\}_{i \in \widehat{W}}$ is linearly independent, 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 (34), then we have that $a_j^T p > 0$, and that p is a descent direction for q .

§16.5 Active-Set Methods for Convex QPs

Theorem

Let \hat{x} satisfy the first-order conditions

$$\sum_{i \in \widehat{\mathcal{W}}} a_i \hat{\lambda}_i = G\hat{x} + c \quad (33)$$

for some $\hat{\lambda}$, and $a_i^T \hat{x} = b_i$ for all $i \in \widehat{\mathcal{W}}$. Suppose that **there is an index $j \in \widehat{\mathcal{W}} \cap \mathcal{I}$ such that $\hat{\lambda}_j < 0$** . Let p be the solution obtained by dropping the constraint j and solving the following sub-problem:

$$\min_p \frac{1}{2} p^T G p + (G\hat{x} + c)^T p \quad (34a)$$

subject to

$$a_i^T p = 0 \quad \text{for all } i \in \widehat{\mathcal{W}} \text{ with } i \neq j. \quad (34b)$$

If constraint gradients $\{a_i\}_{i \in \widehat{\mathcal{W}}}$ is linearly independent, 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 (34), then we have that $a_j^T p > 0$, and that p is a descent direction for q .

§16.5 Active-Set Methods for Convex QPs

Theorem

Let \hat{x} satisfy the first-order conditions

$$\sum_{i \in \widehat{\mathcal{W}}} a_i \hat{\lambda}_i = G\hat{x} + c \quad (33)$$

for some $\hat{\lambda}$, and $a_i^T \hat{x} = b_i$ for all $i \in \widehat{\mathcal{W}}$. Suppose that *there is an index $j \in \widehat{\mathcal{W}} \cap \mathcal{I}$ such that $\hat{\lambda}_j < 0$* . Let p be the solution obtained by dropping the constraint j and solving the following sub-problem:

$$\min_p \frac{1}{2} p^T G p + (G\hat{x} + c)^T p \quad (34a)$$

subject to

$$a_i^T p = 0 \quad \text{for all } i \in \widehat{\mathcal{W}} \text{ with } i \neq j. \quad (34b)$$

If constraint gradients $\{a_i\}_{i \in \widehat{\mathcal{W}}}$ is linearly independent, 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 (34), then we have that $a_j^T p > 0$, and that p is a descent direction for q .

§16.5 Active-Set Methods for Convex QPs

Theorem

Let \hat{x} satisfy the first-order conditions

$$\sum_{i \in \widehat{\mathcal{W}}} a_i \hat{\lambda}_i = G\hat{x} + c \quad (33)$$

for some $\hat{\lambda}$, and $a_i^T \hat{x} = b_i$ for all $i \in \widehat{\mathcal{W}}$. Suppose that *there is an index $j \in \widehat{\mathcal{W}} \cap \mathcal{I}$ such that $\hat{\lambda}_j < 0$* . Let p be the solution obtained by dropping the constraint j and solving the following sub-problem:

$$\min_p \frac{1}{2} p^T G p + (G\hat{x} + c)^T p \quad (34a)$$

subject to

$$a_i^T p = 0 \quad \text{for all } i \in \widehat{\mathcal{W}} \text{ with } i \neq j. \quad (34b)$$

If constraint gradients $\{a_i\}_{i \in \widehat{\mathcal{W}}}$ is linearly independent, 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 (34), then we have that $a_j^T p > 0$, and that p is a descent direction for q .

§16.5 Active-Set Methods for Convex QPs

Proof.

Since p solves (34), we have from the results of Section 16.1 that there are multipliers $\tilde{\lambda}_i$, for all $i \in \widehat{W}$ with $i \neq j$, such that

$$\sum_{i \in \widehat{W}, i \neq j} \tilde{\lambda}_i a_i = Gp + (G\hat{x} + c). \quad (35)$$

In addition, we have by second-order necessary conditions that if Z is a null-space basis vector for the matrix

$$[a_i^T]_{i \in \widehat{W}, i \neq j},$$

then $Z^T G Z$ is positive semi-definite. Clearly, p has the form $p = Z p_Z$ for some vector p_Z , so it follows that $p^T G p \geq 0$. We have made the assumption that \hat{x} and \widehat{W} satisfy the relation (33). By subtracting (33) from (35), we obtain

$$\sum_{i \in \widehat{W}, i \neq j} (\tilde{\lambda}_i - \hat{\lambda}_i) a_i - \hat{\lambda}_j a_j = Gp. \quad (36)$$

§16.5 Active-Set Methods for Convex QPs

Proof.

Since p solves (34), we have from the results of Section 16.1 that there are multipliers $\tilde{\lambda}_i$, for all $i \in \widehat{W}$ with $i \neq j$, such that

$$\sum_{i \in \widehat{W}, i \neq j} \tilde{\lambda}_i a_i = Gp + (G\hat{x} + c). \quad (35)$$

In addition, we have by second-order necessary conditions that if Z is a null-space basis vector for the matrix

$$[a_i^T]_{i \in \widehat{W}, i \neq j},$$

then $Z^T G Z$ is positive semi-definite. Clearly, p has the form $p = Z p_Z$ for some vector p_Z , so it follows that $p^T G p \geq 0$. We have made the assumption that \hat{x} and \widehat{W} satisfy the relation (33). By subtracting (33) from (35), we obtain

$$\sum_{i \in \widehat{W}, i \neq j} (\tilde{\lambda}_i - \hat{\lambda}_i) a_i - \hat{\lambda}_j a_j = Gp. \quad (36)$$

§16.5 Active-Set Methods for Convex QPs

Proof.

Since p solves (34), we have from the results of Section 16.1 that there are multipliers $\tilde{\lambda}_i$, for all $i \in \widehat{W}$ with $i \neq j$, such that

$$\sum_{i \in \widehat{W}, i \neq j} \tilde{\lambda}_i a_i = Gp + (G\hat{x} + c). \quad (35)$$

In addition, we have by second-order necessary conditions that if Z is a null-space basis vector for the matrix

$$[a_i^T]_{i \in \widehat{W}, i \neq j},$$

then $Z^T G Z$ is positive semi-definite. Clearly, p has the form $p = Z p_Z$ for some vector p_Z , so it follows that $p^T G p \geq 0$. We have made the assumption that \hat{x} and \widehat{W} satisfy the relation (33). By subtracting (33) from (35), we obtain

$$\sum_{i \in \widehat{W}, i \neq j} (\tilde{\lambda}_i - \hat{\lambda}_i) a_i - \hat{\lambda}_j a_j = Gp. \quad (36)$$

□

§16.5 Active-Set Methods for Convex QPs

Proof (cont'd).

By taking inner products of both sides with p and using the fact that $a_i^T p = 0$ for all $i \in \widehat{\mathcal{W}}$ with $i \neq j$, we have that

$$-\widehat{\lambda}_j a_j^T p = p^T G p. \quad (37)$$

Since $p^T G p \geq 0$ and $\widehat{\lambda}_j < 0$ by assumption, it follows that $a_j^T p \geq 0$.

If the second-order sufficient conditions of Section 12.5 are satisfied, we have that $Z^T G Z$ defined above is positive definite. From (37), we can have $a_j^T p = 0$ only if

$$p^T G p = p_z^T Z^T G Z p_z = 0,$$

which happens only if $p_z = 0$ and thus $p = 0$. But if $p = 0$, then by substituting into (36) and using linear independence of a_i for $i \in \widehat{\mathcal{W}}$, we must have that $\widehat{\lambda}_j = 0$, which contradicts our choice of j . \square

§16.5 Active-Set Methods for Convex QPs

Proof (cont'd).

By taking inner products of both sides with p and using the fact that $a_i^T p = 0$ for all $i \in \widehat{\mathcal{W}}$ with $i \neq j$, we have that

$$-\widehat{\lambda}_j a_j^T p = p^T G p. \quad (37)$$

Since $p^T G p \geq 0$ and $\widehat{\lambda}_j < 0$ by assumption, it follows that $a_j^T p \geq 0$. If the second-order sufficient conditions of Section 12.5 are satisfied, we have that $Z^T G Z$ defined above is positive definite. From (37), we can have $a_j^T p = 0$ only if

$$p^T G p = p_z^T Z^T G Z p_z = 0,$$

which happens only if $p_z = 0$ and thus $p = 0$. But if $p = 0$, then by substituting into (36) and using linear independence of a_i for $i \in \widehat{\mathcal{W}}$, we must have that $\widehat{\lambda}_j = 0$, which contradicts our choice of j . \square

§16.5 Active-Set Methods for Convex QPs

Proof (cont'd).

We conclude that $p^T G p > 0$ in (37), and therefore $a_j^T p > 0$ whenever p satisfies the second-order sufficient conditions for (34). The claim that p is a descent direction for q is proved in the next theorem. \square

接下來的定理給出了一個結果（並完成上個定理的證明），其顯示只要從 (31) 獲得的 p_k 非零並且滿足當前工作集的二階充分最優條件，則它是 q 的嚴格下降方向。

Theorem

Suppose that the solution p_k of (31) is nonzero and satisfies the second-order sufficient conditions for optimality for that problem. Then the function q is strictly decreasing along the direction p_k .

§16.5 Active-Set Methods for Convex QPs

Proof (cont'd).

We conclude that $p^T G p > 0$ in (37), and therefore $a_j^T p > 0$ whenever p satisfies the second-order sufficient conditions for (34). The claim that p is a descent direction for q is proved in the next theorem. \square

接下來的定理給出了一個結果（並完成上個定理的證明），其顯示只要從 (31) 獲得的 p_k 非零並且滿足當前工作集的二階充分最優條件，則它是 q 的嚴格下降方向。

Theorem

Suppose that the solution p_k of (31) is nonzero and satisfies the second-order sufficient conditions for optimality for that problem. Then the function q is strictly decreasing along the direction p_k .

§16.5 Active-Set Methods for Convex QPs

Proof.

Since p_k satisfies the second-order sufficient conditions; that is, $Z^T G Z$ is positive definite for the matrix Z whose columns are a basis of the null space of the constraints (31b), by the theorem in §16.1 we find that p_k is the unique global solution of (31).

Since $p = 0$ is also a feasible point for (31), its objective value in (31a) must be larger than that of p_k , so we have

$$\frac{1}{2} p_k^T G p_k + g_k^T p_k < 0.$$

Since $p_k^T G p_k \geq 0$ by convexity, this inequality implies that $g_k^T p_k < 0$. Therefore, we have

$$q(x_k + \alpha p_k) = q(x_k) + \alpha g_k^T p_k + \frac{1}{2} \alpha^2 p_k^T G p_k < q(x_k)$$

for all $\alpha > 0$ sufficiently small. □

§16.5 Active-Set Methods for Convex QPs

Proof.

Since p_k satisfies the second-order sufficient conditions; that is, $Z^T G Z$ is positive definite for the matrix Z whose columns are a basis of the null space of the constraints (31b), by the theorem in §16.1 we find that p_k is the unique global solution of (31).

Since $p = 0$ is also a feasible point for (31), its objective value in (31a) must be larger than that of p_k , so we have

$$\frac{1}{2} p_k^T G p_k + g_k^T p_k < 0.$$

Since $p_k^T G p_k \geq 0$ by convexity, this inequality implies that $g_k^T p_k < 0$. Therefore, we have

$$q(x_k + \alpha p_k) = q(x_k) + \alpha g_k^T p_k + \frac{1}{2} \alpha^2 p_k^T G p_k < q(x_k)$$

for all $\alpha > 0$ sufficiently small. □

§16.5 Active-Set Methods for Convex QPs

Proof.

Since p_k satisfies the second-order sufficient conditions; that is, $Z^T G Z$ is positive definite for the matrix Z whose columns are a basis of the null space of the constraints (31b), by the theorem in §16.1 we find that p_k is the unique global solution of (31).

Since $p = 0$ is also a feasible point for (31), its objective value in (31a) must be larger than that of p_k , so we have

$$\frac{1}{2} p_k^T G p_k + g_k^T p_k < 0.$$

Since $p_k^T G p_k \geq 0$ by convexity, this inequality implies that $g_k^T p_k < 0$. Therefore, we have

$$q(x_k + \alpha p_k) = q(x_k) + \alpha g_k^T p_k + \frac{1}{2} \alpha^2 p_k^T G p_k < q(x_k)$$

for all $\alpha > 0$ sufficiently small. □

§16.5 Active-Set Methods for Convex QPs

通常情況下，此算法對任何滿足 $\hat{\lambda}_j < 0$ 的 index j 都會產生一個可以取得進展的方向 p ，但實踐中（以及下文中指定的算法中），通常選擇最負的乘子。這個選擇是由第 12 章中給出的敏感性分析所驅動的，該分析顯示當刪除一個限制時目標函數的減小速率與該限制的 Lagrange 乘子的大小成正比。然而，就像在線性規劃中一樣，沿着這個方向的步長可能會很短（例如當被一個新的限制 block 時），因此 q 的減少量不能保證大於對於其他可能的 j 的選擇。

當 G 是正定的時候 – 即嚴格凸的情況 – 所有形式為 (31) 的可行子問題都滿足二階充分條件。因此，根據上述結果，我們在 $p_k \neq 0$ 時得到了 q 的嚴格下降。這個事實在討論算法的 finite termination 時是非常重要的。

§16.5 Active-Set Methods for Convex QPs

通常情況下，此算法對任何滿足 $\hat{\lambda}_j < 0$ 的 index j 都會產生一個可以取得進展的方向 p ，但實踐中（以及下文中指定的算法中），通常選擇最負的乘子。這個選擇是由第 12 章中給出的敏感性分析所驅動的，該分析顯示當刪除一個限制時目標函數的減小速率與該限制的 Lagrange 乘子的大小成正比。然而，就像在線性規劃中一樣，沿着這個方向的步長可能會很短（例如當被一個新的限制 block 時），因此 q 的減少量不能保證大於對於其他可能的 j 的選擇。

當 G 是正定的時候 – 即嚴格凸的情況 – 所有形式為 (31) 的可行子問題都滿足二階充分條件。因此，根據上述結果，我們在 $p_k \neq 0$ 時得到了 q 的嚴格下降。這個事實在討論算法的 finite termination 時是非常重要的。

§16.5 Active-Set Methods for Convex QPs

- **Specification of the active-set method for convex QP**

Algorithm 16.3 (Active-Set Method for Convex QP).

Compute a feasible starting point x_0 ;

Set \mathcal{W}_0 to be a subset of the active constraints at x_0 ;

for $k = 0, 1, 2, \dots$

 Solve (31) to find p_k ;

if $p_k = 0$

 Compute Lagrange multipliers $\hat{\lambda}_i$ that satisfy (33), with

$$\widehat{\mathcal{W}} = \mathcal{W}_k ;$$

if $\hat{\lambda}_i \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$

stop with solution $x_* = x_k$;

else

$$j \leftarrow \arg \min_{j \in \mathcal{W}_k \cap \mathcal{I}} \hat{\lambda}_j ;$$

$$x_{k+1} \leftarrow x_k ; \mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \setminus \{j\} ;$$

else ($*p_k \neq 0*$)

§16.5 Active-Set Methods for Convex QPs

$$j \leftarrow \arg \min_{j \in \mathcal{W}_k \cap \mathcal{I}} \hat{\lambda}_j;$$

$$x_{k+1} \leftarrow x_k; \mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \setminus \{j\};$$

else ($\star p_k \neq 0 \star$)

Compute α_k from (32);

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

if there are blocking constraints

Obtain \mathcal{W}_{k+1} by adding one of the blocking constraints to \mathcal{W}_k ;

else

$$\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k;$$

end (for)

§16.5 Active-Set Methods for Convex QPs

可以使用各種技術來確定初始可行點。其中一種方法是使用第 13 章中描述的線性規劃的“Phase I”方法。儘管從線性規劃推廣到二次規劃不需要進行重大修改，但在這裡我們描述一種變體，允許用戶提供向量 x 的初始估計 \tilde{x} 。這個估計不需要是可行的，但是如果根據對 QP 的了解做出良好的選擇，可能會減少 Phase I 步驟中所需的工作量。

§16.5 Active-Set Methods for Convex QPs

給定 \tilde{x} ，我們定義以下的線性規劃問題：

$$\begin{aligned} \min_{(x,z)} e^T z \quad \text{subject to} \quad & a_i^T x + \gamma_i z_i = b_i, & i \in \mathcal{E}, \\ & a_i^T x + \gamma_i z_i \geq b_i, & i \in \mathcal{I}, \\ & z \geq 0, \end{aligned}$$

其中 $e = (1, \dots, 1)^T \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$ 且 $\gamma_i = \begin{cases} -\text{sgn}(a_i^T \tilde{x} - b_i) & \text{if } i \in \mathcal{E}, \\ 1 & \text{if } i \in \mathcal{I}. \end{cases}$

這個問題的一個可行初始點為

$$x = \tilde{x}, \quad z_i = |a_i^T \tilde{x} - b_i| \quad (i \in \mathcal{E}), \quad z_i = \max\{b_i - a_i^T \tilde{x}, 0\} \quad (i \in \mathcal{I}).$$

可以驗證如果 \tilde{x} 對於原問題 (1) 是可行的，則 $(x, z) = (\tilde{x}, 0)$ 對於上述子問題是最優解。一般來說，如果原問題有可行點，則上述子問題中的最優目標值為零，而上述子問題的任何解都會產生原問題 (1) 的一個可行點。對於 Algorithm 16.3，初始工作集 \mathcal{W}_0 可以通過找在可行點線性獨立的 active constraint 來取得。

§16.5 Active-Set Methods for Convex QPs

給定 \tilde{x} ，我們定義以下的線性規劃問題：

$$\begin{aligned} \min_{(x,z)} e^T z \quad \text{subject to} \quad & a_i^T x + \gamma_i z_i = b_i, & i \in \mathcal{E}, \\ & a_i^T x + \gamma_i z_i \geq b_i, & i \in \mathcal{I}, \\ & z \geq 0, \end{aligned}$$

其中 $e = (1, \dots, 1)^T \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$ 且 $\gamma_i = \begin{cases} -\text{sgn}(a_i^T \tilde{x} - b_i) & \text{if } i \in \mathcal{E}, \\ 1 & \text{if } i \in \mathcal{I}. \end{cases}$

這個問題的一個可行初始點為

$$x = \tilde{x}, \quad z_i = |a_i^T \tilde{x} - b_i| \quad (i \in \mathcal{E}), \quad z_i = \max \{b_i - a_i^T \tilde{x}, 0\} \quad (i \in \mathcal{I}).$$

可以驗證如果 \tilde{x} 對於原問題 (1) 是可行的，則 $(x, z) = (\tilde{x}, 0)$ 對於上述子問題是最優解。一般來說，如果原問題有可行點，則上述子問題中的最優目標值為零，而上述子問題的任何解都會產生原問題 (1) 的一個可行點。對於 Algorithm 16.3，初始工作集 \mathcal{W}_0 可以通過找在可行點線性獨立的 active constraint 來取得。

§16.5 Active-Set Methods for Convex QPs

給定 \tilde{x} ，我們定義以下的線性規劃問題：

$$\begin{aligned} \min_{(x,z)} e^T z \quad \text{subject to} \quad & a_i^T x + \gamma_i z_i = b_i, & i \in \mathcal{E}, \\ & a_i^T x + \gamma_i z_i \geq b_i, & i \in \mathcal{I}, \\ & z \geq 0, \end{aligned}$$

其中 $e = (1, \dots, 1)^T \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$ 且 $\gamma_i = \begin{cases} -\text{sgn}(a_i^T \tilde{x} - b_i) & \text{if } i \in \mathcal{E}, \\ 1 & \text{if } i \in \mathcal{I}. \end{cases}$

這個問題的一個可行初始點為

$$x = \tilde{x}, \quad z_i = |a_i^T \tilde{x} - b_i| \quad (i \in \mathcal{E}), \quad z_i = \max \{b_i - a_i^T \tilde{x}, 0\} \quad (i \in \mathcal{I}).$$

可以驗證如果 \tilde{x} 對於原問題 (1) 是可行的，則 $(x, z) = (\tilde{x}, 0)$ 對於上述子問題是最優解。一般來說，如果原問題有可行點，則上述子問題中的最優目標值為零，而上述子問題的任何解都會產生原問題 (1) 的一個可行點。對於 Algorithm 16.3，初始工作集 \mathcal{W}_0 可以通過找在可行點線性獨立的 active constraint 來取得。

§16.5 Active-Set Methods for Convex QPs

給定 \tilde{x} ，我們定義以下的線性規劃問題：

$$\begin{aligned} \min_{(x,z)} e^T z \quad \text{subject to} \quad & a_i^T x + \gamma_i z_i = b_i, & i \in \mathcal{E}, \\ & a_i^T x + \gamma_i z_i \geq b_i, & i \in \mathcal{I}, \\ & z \geq 0, \end{aligned}$$

其中 $e = (1, \dots, 1)^T \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$ 且 $\gamma_i = \begin{cases} -\text{sgn}(a_i^T \tilde{x} - b_i) & \text{if } i \in \mathcal{E}, \\ 1 & \text{if } i \in \mathcal{I}. \end{cases}$

這個問題的一個可行初始點為

$$x = \tilde{x}, \quad z_i = |a_i^T \tilde{x} - b_i| \quad (i \in \mathcal{E}), \quad z_i = \max \{b_i - a_i^T \tilde{x}, 0\} \quad (i \in \mathcal{I}).$$

可以驗證如果 \tilde{x} 對於原問題 (1) 是可行的，則 $(x, z) = (\tilde{x}, 0)$ 對於上述子問題是最優解。一般來說，如果原問題有可行點，則上述子問題中的最優目標值為零，而上述子問題的任何解都會產生原問題 (1) 的一個可行點。對於 Algorithm 16.3，初始工作集 \mathcal{W}_0 可以通過找在可行點線性獨立的 active constraint 來取得。

§16.5 Active-Set Methods for Convex QPs

給定 \tilde{x} ，我們定義以下的線性規劃問題：

$$\begin{aligned} \min_{(x,z)} e^T z \quad \text{subject to} \quad & a_i^T x + \gamma_i z_i = b_i, & i \in \mathcal{E}, \\ & a_i^T x + \gamma_i z_i \geq b_i, & i \in \mathcal{I}, \\ & z \geq 0, \end{aligned}$$

其中 $e = (1, \dots, 1)^T \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$ 且 $\gamma_i = \begin{cases} -\text{sgn}(a_i^T \tilde{x} - b_i) & \text{if } i \in \mathcal{E}, \\ 1 & \text{if } i \in \mathcal{I}. \end{cases}$

這個問題的一個可行初始點為

$$x = \tilde{x}, \quad z_i = |a_i^T \tilde{x} - b_i| \quad (i \in \mathcal{E}), \quad z_i = \max \{b_i - a_i^T \tilde{x}, 0\} \quad (i \in \mathcal{I}).$$

可以驗證如果 \tilde{x} 對於原問題 (1) 是可行的，則 $(x, z) = (\tilde{x}, 0)$ 對於上述子問題是最優解。一般來說，如果原問題有可行點，則上述子問題中的最優目標值為零，而上述子問題的任何解都會產生原問題 (1) 的一個可行點。對於 Algorithm 16.3，初始工作集 \mathcal{W}_0 可以通過找在可行點線性獨立的 active constraint 來取得。

§16.5 Active-Set Methods for Convex QPs

另一種方法是使用懲罰（或“big M ”）法，它省去了“Phase I”，而是在目標函數中加入一個保證在真解中為零的違反限制的度量。也就是說，我們在 (1) 中引入一個純量人工變量 η 來衡量違反限制的程度，然後解決以下問題：

$$\min_{(x,\eta)} \frac{1}{2}x^T Gx + x^T c + M\eta \quad \text{subject to} \quad \begin{cases} (a_i^T x - b_i) \leq \eta, & i \in \mathcal{E}, \\ -(a_i^T x - b_i) \leq \eta, & i \in \mathcal{E}, \\ b_i - a_i^T x \leq \eta, & i \in \mathcal{I}, \\ 0 \leq \eta, \end{cases} \quad (38)$$

其中 M 是某個足夠大的正數。通過應用懲罰函數的理論（見第 17 章），可以證明只要原問題 (1) 存在可行點，那麼對於所有足夠大的 M ，(38) 的解將滿足 $\eta = 0$ 且具有作為原問題 (1) 解的分量。

§16.5 Active-Set Methods for Convex QPs

另一種方法是使用懲罰（或“big M ”）法，它省去了“Phase I”，而是在目標函數中加入一個保證在真解中為零的違反限制的度量。也就是說，我們在 (1) 中引入一個純量人工變量 η 來衡量違反限制的程度，然後解決以下問題：

$$\min_{(x,\eta)} \frac{1}{2}x^T Gx + x^T c + M\eta \quad \text{subject to} \quad \begin{cases} (a_i^T x - b_i) \leq \eta, & i \in \mathcal{E}, \\ -(a_i^T x - b_i) \leq \eta, & i \in \mathcal{E}, \\ b_i - a_i^T x \leq \eta, & i \in \mathcal{I}, \\ 0 \leq \eta, \end{cases} \quad (38)$$

其中 M 是某個足夠大的正數。通過應用懲罰函數的理論（見第 17 章），可以證明只要原問題 (1) 存在可行點，那麼對於所有足夠大的 M ，(38) 的解將滿足 $\eta = 0$ 且具有作為原問題 (1) 解的分量。

§16.5 Active-Set Methods for Convex QPs

與先前對問題 (1) 找可行點不同的是，對於子問題 (38) 很容易獲得一個可行點：我們隨便猜測 x 的初始值，並選擇足夠大的 η 使得 (38) 中的所有限制皆滿足。而 M 的選擇一般是基於某些經驗法則來選擇：如果 (38) 的解具有正的 η 值，我們將增加 M 的值並重新嘗試。事實上，這種方法是使用 l_∞ -norm 的懲罰方法；請參見第 17 章。

一個對違反限制的 l_1 -norm 進行懲罰的 (38) 變體如下所示：

$$\begin{aligned} \min_{(x,s,t,v)} \quad & \frac{1}{2}x^T Gx + x^T c + Me_{\mathcal{E}}^T(s+t) + Me_{\mathcal{I}}^T v \\ \text{subject to} \quad & a_i^T x - b_i + s_i - t_i = 0, \quad i \in \mathcal{E}, \\ & a_i^T x - b_i + v_i \geq 0, \quad i \in \mathcal{I}, \\ & s \geq 0, t \geq 0, v \geq 0. \end{aligned}$$

在此， $e_{\mathcal{E}}$ 是長度為 $|\mathcal{E}|$ 的向量 $(1, 1, \dots, 1)^T$ ； $e_{\mathcal{I}}$ 同樣如此。鬆弛變量 s_i 、 t_i 和 v_i 吸收了限制中的任何不可行性。

§16.5 Active-Set Methods for Convex QPs

與先前對問題 (1) 找可行點不同的是，對於子問題 (38) 很容易獲得一個可行點：我們隨便猜測 x 的初始值，並選擇足夠大的 η 使得 (38) 中的所有限制皆滿足。而 M 的選擇一般是基於某些經驗法則來選擇：如果 (38) 的解具有正的 η 值，我們將增加 M 的值並重新嘗試。事實上，這種方法是使用 ℓ_∞ -norm 的懲罰方法；請參見第 17 章。

一個對違反限制的 ℓ_1 -norm 進行懲罰的 (38) 變體如下所示：

$$\begin{aligned} \min_{(x,s,t,v)} \quad & \frac{1}{2} x^T G x + x^T c + M e_{\mathcal{E}}^T (s + t) + M e_{\mathcal{I}}^T v \\ \text{subject to} \quad & a_i^T x - b_i + s_i - t_i = 0, \quad i \in \mathcal{E}, \\ & a_i^T x - b_i + v_i \geq 0, \quad i \in \mathcal{I}, \\ & s \geq 0, t \geq 0, v \geq 0. \end{aligned}$$

在此， $e_{\mathcal{E}}$ 是長度為 $|\mathcal{E}|$ 的向量 $(1, 1, \dots, 1)^T$ ； $e_{\mathcal{I}}$ 同樣如此。鬆弛變量 s_i 、 t_i 和 v_i 吸收了限制中的任何不可行性。

§16.5 Active-Set Methods for Convex QPs

與先前對問題 (1) 找可行點不同的是，對於子問題 (38) 很容易獲得一個可行點：我們隨便猜測 x 的初始值，並選擇足夠大的 η 使得 (38) 中的所有限制皆滿足。而 M 的選擇一般是基於某些經驗法則來選擇：如果 (38) 的解具有正的 η 值，我們將增加 M 的值並重新嘗試。事實上，這種方法是使用 l_∞ -norm 的懲罰方法；請參見第 17 章。

一個對違反限制的 l_1 -norm 進行懲罰的 (38) 變體如下所示：

$$\begin{aligned} \min_{(x,s,t,v)} \quad & \frac{1}{2} x^T G x + x^T c + M e_{\mathcal{E}}^T (s + t) + M e_{\mathcal{I}}^T v \\ \text{subject to} \quad & a_i^T x - b_i + s_i - t_i = 0, \quad i \in \mathcal{E}, \\ & a_i^T x - b_i + v_i \geq 0, \quad i \in \mathcal{I}, \\ & s \geq 0, t \geq 0, v \geq 0. \end{aligned}$$

在此， $e_{\mathcal{E}}$ 是長度為 $|\mathcal{E}|$ 的向量 $(1, 1, \dots, 1)^T$ ； $e_{\mathcal{I}}$ 同樣如此。鬆弛變量 s_i 、 t_i 和 v_i 吸收了限制中的任何不可行性。

§16.5 Active-Set Methods for Convex QPs

在以下示例中，我們使用向量 x 和 p 的下標來表示它們的分量，使用上標來表示迭代的 index。例如， x_1 表示向量 x 的第一個分量，而 x^4 表示向量 x 的第四次迭代。

Example

We apply Algorithm 16.3 to the following simple 2-dimensional problem illustrated in Figure 3.

$$\min_x q(x) = (x_1 - 1)^2 + (x_2 - 2.5)^2 \quad (39a)$$

subject to

$$x_1 - 2x_2 + 2 \geq 0, \quad (39b)$$

$$-x_1 - 2x_2 + 6 \geq 0, \quad (39c)$$

$$-x_1 + 2x_2 + 2 \geq 0, \quad (39d)$$

$$x_1 \geq 0, \quad (39e)$$

$$x_2 \geq 0. \quad (39f)$$

We refer the constraints, in order, by indices 1 through 5.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

To apply Algorithm 16.3, we rewrite the problem in terms of the form of QPs and study the following optimization problem

$$\min_x q(x) = \frac{1}{2}x^T G x + x^T c$$

subject to

$$A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq b,$$

where

$$G = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} -2 \\ -5 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & -2 \\ -1 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} -2 \\ -6 \\ -2 \\ 0 \\ 0 \end{bmatrix}.$$

Suppose at the k -th iterate the working set is \mathcal{W}_k , and $x = x^k$.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Let $[\mathcal{W}_k]$ be the column vector whose entries are elements in \mathcal{W}_k .

Define

$$A_k = A([\mathcal{W}_k], :) \quad \text{and} \quad b_k = b([\mathcal{W}_k], :),$$

and then solve

$$\begin{bmatrix} G & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} -p^k \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} Gx^k + c \\ A_k x^k - b_k \end{bmatrix}. \quad (4')$$

At the $(k+1)$ -th iterate, Algorithm 16.3 provides $x^{k+1} = x^k + \alpha_k p^k$ with α_k given by (32), and update \mathcal{W}_{k+1} if necessary.

For this problem it is easy to determine a feasible initial point; say $x^0 = (2, 0)^T$. Constraints 3 and 5 are active at this point, and we set $\mathcal{W}_0 = \{3, 5\}$. Note that we could just as validly have chosen $\mathcal{W}_0 = \{5\}$ or $\mathcal{W}_0 = \{3\}$ or even $\mathcal{W} = \emptyset$; each choice would lead the algorithm to perform somewhat differently.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Let $[\mathcal{W}_k]$ be the column vector whose entries are elements in \mathcal{W}_k .

Define

$$A_k = A([\mathcal{W}_k], :) \quad \text{and} \quad b_k = b([\mathcal{W}_k], :),$$

and then solve

$$\begin{bmatrix} G & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} -p^k \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} Gx^k + c \\ A_k x^k - b_k \end{bmatrix}. \quad (4')$$

At the $(k+1)$ -th iterate, Algorithm 16.3 provides $x^{k+1} = x^k + \alpha_k p^k$ with α_k given by (32), and update \mathcal{W}_{k+1} if necessary.

For this problem it is easy to determine a feasible initial point; say $x^0 = (2, 0)^T$. Constraints 3 and 5 are active at this point, and we set $\mathcal{W}_0 = \{3, 5\}$. Note that we could just as validly have chosen $\mathcal{W}_0 = \{5\}$ or $\mathcal{W}_0 = \{3\}$ or even $\mathcal{W} = \emptyset$; each choice would lead the algorithm to perform somewhat differently.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

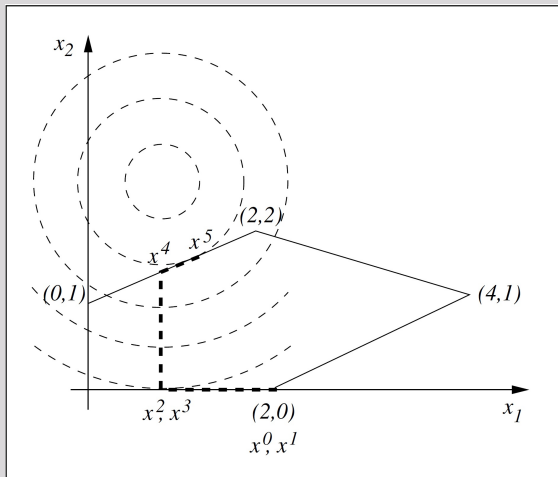


Figure 3: Iterates of the active-set method.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Since x^0 lies on a vertex of the feasible region, it is a minimizer of the objective function q with respect to the working set \mathcal{W}_0 ; that is, the solution of (31) with $k = 0$ is $p = 0$. We can then use (33) to find the multipliers $\hat{\lambda}_3$ and $\hat{\lambda}_5$ associated with the active constraints. Substitution of the data from our problem into (33) yields

$$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \hat{\lambda}_3 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{\lambda}_5 = \begin{bmatrix} 2 \\ -5 \end{bmatrix},$$

which has the solution $(\hat{\lambda}_3, \hat{\lambda}_5) = (-2, -1)$.

We now remove constraint 3 from the working set, because it has the most negative multiplier, and set $\mathcal{W}_1 = \{5\}$. Consider

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -p^1 \\ \hat{\lambda}_5 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 2 \\ 0 \end{bmatrix} + c \\ [0 \ 1] \begin{bmatrix} 2 \\ 0 \end{bmatrix} - 0 \end{bmatrix}. \quad (4'')$$

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Since x^0 lies on a vertex of the feasible region, it is a minimizer of the objective function q with respect to the working set \mathcal{W}_0 ; that is, the solution of (31) with $k = 0$ is $p = 0$. We can then use (33) to find the multipliers $\hat{\lambda}_3$ and $\hat{\lambda}_5$ associated with the active constraints. Substitution of the data from our problem into (33) yields

$$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \hat{\lambda}_3 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{\lambda}_5 = \begin{bmatrix} 2 \\ -5 \end{bmatrix},$$

which has the solution $(\hat{\lambda}_3, \hat{\lambda}_5) = (-2, -1)$.

We now remove constraint 3 from the working set, because it has the most negative multiplier, and set $\mathcal{W}_1 = \{5\}$. Consider

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -p^1 \\ \hat{\lambda}_5 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 2 \\ 0 \end{bmatrix} + c \\ [0 \ 1] \begin{bmatrix} 2 \\ 0 \end{bmatrix} - 0 \end{bmatrix}. \quad (4'')$$

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Since x^0 lies on a vertex of the feasible region, it is a minimizer of the objective function q with respect to the working set \mathcal{W}_0 ; that is, the solution of (31) with $k = 0$ is $p = 0$. We can then use (33) to find the multipliers $\hat{\lambda}_3$ and $\hat{\lambda}_5$ associated with the active constraints. Substitution of the data from our problem into (33) yields

$$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \hat{\lambda}_3 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{\lambda}_5 = \begin{bmatrix} 2 \\ -5 \end{bmatrix},$$

which has the solution $(\hat{\lambda}_3, \hat{\lambda}_5) = (-2, -1)$.

We now remove constraint 3 from the working set, because it has the most negative multiplier, and set $\mathcal{W}_1 = \{5\}$. Consider

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -p^1 \\ \hat{\lambda}_5 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 2 \\ 0 \end{bmatrix} + c \\ [0 \quad 1] \begin{bmatrix} 2 \\ 0 \end{bmatrix} - 0 \end{bmatrix}. \quad (4'')$$

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Solving (31) (or exactly (4'')) for $k = 1$, we obtain the solution $p^1 = (-1, 0)^T$ (and $\hat{\lambda}_5 = -5$). The step-length formula (32) yields $\alpha_1 = 1$, and the new iterate is $x^2 = (1, 0)^T$.

There are no blocking constraints, so that $\mathcal{W}_2 = \mathcal{W}_1 = \{5\}$, and we find at the start of iteration 2 that the solution of

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -p^2 \\ \hat{\lambda}_5 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c \\ [0 \quad 1] \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 0 \end{bmatrix} \quad (4''')$$

is $p^2 = (0, 0)^T$ and $\hat{\lambda}_5 = -5$, so we drop 5 from the working set to obtain $\mathcal{W}_3 = \emptyset$.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Solving (31) (or exactly (4'')) for $k = 1$, we obtain the solution $p^1 = (-1, 0)^T$ (and $\hat{\lambda}_5 = -5$). The step-length formula (32) yields $\alpha_1 = 1$, and the new iterate is $x^2 = (1, 0)^T$.

There are no blocking constraints, so that $\mathcal{W}_2 = \mathcal{W}_1 = \{5\}$, and we find at the start of iteration 2 that the solution of

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -p^2 \\ \hat{\lambda}_5 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c \\ [0 \quad 1] \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 0 \end{bmatrix} \quad (4''')$$

is $p^2 = (0, 0)^T$ and $\hat{\lambda}_5 = -5$, so we drop 5 from the working set to obtain $\mathcal{W}_3 = \emptyset$.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Iteration 3 starts by solving the unconstrained problem, to obtain the solution of

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}(-p^3) = G \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c \quad (4''''')$$

as $p^3 = (0, 2.5)^T$. The formula (32) yields a step length of $\alpha_3 = 0.6$ and a new iterate $x^4 = (1, 1.5)^T$. There is a single blocking constraint (constraint 1), so we obtain $\mathcal{W}_4 = \{1\}$.

The solution of (31) for $k = 4$ is to solve

$$\begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & -2 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} -p^4 \\ \hat{\lambda}_1 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} + c \\ [1 \quad -2] \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} - (-2) \end{bmatrix} \quad (4''''''')$$

to obtain $p^4 = (0.4, 0.2)^T$ (and $\hat{\lambda}_1 = -0.8$), and the new step length is 1. Therefore, $x^5 = (1.4, 1.7)^T$.

§16.5 Active-Set Methods for Convex QPs

Example (cont'd)

Iteration 3 starts by solving the unconstrained problem, to obtain the solution of

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}(-p^3) = G \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c \quad (4''''')$$

as $p^3 = (0, 2.5)^T$. The formula (32) yields a step length of $\alpha_3 = 0.6$ and a new iterate $x^4 = (1, 1.5)^T$. There is a single blocking constraint (constraint 1), so we obtain $\mathcal{W}_4 = \{1\}$.

The solution of (31) for $k = 4$ is to solve

$$\begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & -2 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} -p^4 \\ \hat{\lambda}_1 \end{bmatrix} = \begin{bmatrix} G \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} + c \\ [1 \quad -2] \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} - (-2) \end{bmatrix} \quad (4''''''')$$

to obtain $p^4 = (0.4, 0.2)^T$ (and $\hat{\lambda}_1 = -0.8$), and the new step length is 1. Therefore, $x^5 = (1.4, 1.7)^T$.

§16.5 Active-Set Methods for Convex QPs

Remark: 在上例中我們注意到每次得到 $\alpha_k = 1$ 後下一次迭代的 p^{k+1} 必為零向量，這是因為在 $\alpha_k = 1$ 時 Algorithm 16.3 並不調整第 $k+1$ 次迭代的工作集，亦即 $\mathcal{W}_{k+1} = \mathcal{W}_k$ ，因而 $A_{k+1} = A_k$ 而導致求解的第 k 次迭代的系統

$$\begin{bmatrix} G & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} -p^k \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} Gx^k + c \\ A_k x^k - b_k \end{bmatrix}. \quad (4')$$

在第 $k+1$ 次迭代變成了

$$\begin{bmatrix} G & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} -p^{k+1} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} G(x^k + p^k) + c \\ A_k(x^k + p^k) - b_k \end{bmatrix}. \quad (4'')$$

此系統解 p^{k+1} 必為零向量（且 $\hat{\lambda}$ 與上一步計算出的乘子相同）。因此，若遇到 $\alpha_k = 1$ 的情況，可以直接考慮刪除最負的 Lagrange 乘子所對應的 constraint 以減少迭代步驟。

§16.5 Active-Set Methods for Convex QPs

• Further remarks on the active-set method

我們在上面指出了在初始工作集的選擇上存在彈性，而每個初始選擇都會導致不同的迭代序列。當初始 active constraint 具有獨立的梯度時，我們可以將它們全部包含在 \mathcal{W}_0 中。或者，我們可以選擇其中的一個子集。例如，在上面的示例中，如果我們選擇了 $\mathcal{W}_0 = \{3\}$ ，那麼第一次迭代將產生 $p^0 = (0.2, 0.1)^T$ 以及新的迭代 $x^1 = (2.2, 0.1)^T$ 。如果我們選擇了 $\mathcal{W}_0 = \{5\}$ ，那麼我們將立即移動到新的迭代 $x^1 = (1, 0)^T$ ，而不必像示例中那樣先執行刪除 index 3 的操作。如果我們選擇了 $\mathcal{W}_0 = \emptyset$ ，我們將獲得 $p^1 = (-1, 2.5)^T$ ， $\alpha_1 = \frac{2}{3}$ 與 $\alpha_1 = \frac{2}{3}$ ，新的迭代 $x^1 = \left(\frac{4}{3}, \frac{5}{3}\right)^T$ ，以及新的工作集 $\mathcal{W}_1 = \{1\}$ 。解 x_* 將在下一次迭代中被找到。

§16.5 Active-Set Methods for Convex QPs

即使初始工作集 \mathcal{W}_0 與初始活動集合 $\mathcal{A}(x_0)$ 重疊，隨後的迭代中集合 \mathcal{W}_k 和 $\mathcal{A}(x_k)$ 可能會有所不同。例如，當特定步驟遇到多個 blocking constraints 時，只會將其中一個添加到工作集中，因此 \mathcal{W}_k 和 $\mathcal{A}(x_k)$ 之間的對應關係就會被打破。此外，根據所做的選擇，後續的迭代通常會有所不同。

我們要求 \mathcal{W}_0 中的限制式梯度是線性獨立的，而我們修改工作集的策略也確保了所有後續的工作集 \mathcal{W}_k 中的限制式梯度也線性獨立：當我們在特定步驟遇到一個 blocking constraint 時，該限制的梯度向量不會是當前工作集中法向量 a_i 的線性組合。因此，在將 blocking constraint 添加到工作集後，線性獨立性得以保持。另一方面，從工作集中刪除一個 index 不會引入線性相依性。

§16.5 Active-Set Methods for Convex QPs

即使初始工作集 \mathcal{W}_0 與初始活動集合 $\mathcal{A}(x_0)$ 重疊，隨後的迭代中集合 \mathcal{W}_k 和 $\mathcal{A}(x_k)$ 可能會有所不同。例如，當特定步驟遇到多個 blocking constraints 時，只會將其中一個添加到工作集中，因此 \mathcal{W}_k 和 $\mathcal{A}(x_k)$ 之間的對應關係就會被打破。此外，根據所做的選擇，後續的迭代通常會有所不同。

我們要求 \mathcal{W}_0 中的限制式梯度是線性獨立的，而我們修改工作集的策略也確保了所有後續的工作集 \mathcal{W}_k 中的限制式梯度也線性獨立：當我們在特定步驟遇到一個 blocking constraint 時，該限制的梯度向量不會是當前工作集中法向量 a_i 的線性組合。因此，在將 blocking constraint 添加到工作集後，線性獨立性得以保持。另一方面，從工作集中刪除一個 index 不會引入線性相依性。

§16.5 Active-Set Methods for Convex QPs

將最負的 Lagrange 乘子對應的限制刪除的策略在實踐中通常效果良好，但其缺點是容易受到限制式縮放的影響（通過將限制 i 乘以一正數 β 不會改變優化問題的結構，但會引入相應乘子 λ_i 的縮放 $1/\beta$ 倍）。選擇最負的乘子類似於線性規劃中 simplex 法的 Dantzig 原始樞紐規則（參見第 13 章），正如我們在那裡所指出的，對縮放不太敏感的策略通常會得到更好的結果。

我們將不再進一步討論這個進階主題。要注意的是，在 Algorithm 16.3 的每次迭代中添加或刪除至多一個限制，對到達最佳解所需的迭代次數就設置了自然下界。例如一個最佳化問題在解 x_* 上有 m 個 active 不等式限制，若從一個嚴格滿足所有不等式限制的點 x^0 開始進行迭代，算法至少需要 m 次迭代才能從 x^0 移到 x_* 。如果算法在某次迭代中添加了限制 j 到工作集中然後在稍後的步驟中又將其刪除，則需要進行更多次迭代。

§16.5 Active-Set Methods for Convex QPs

將最負的 Lagrange 乘子對應的限制刪除的策略在實踐中通常效果良好，但其缺點是容易受到限制式縮放的影響（通過將限制 i 乘以一正數 β 不會改變優化問題的結構，但會引入相應乘子 λ_i 的縮放 $1/\beta$ 倍）。選擇最負的乘子類似於線性規劃中 simplex 法的 Dantzig 原始樞紐規則（參見第 13 章），正如我們在那裡所指出的，對縮放不太敏感的策略通常會得到更好的結果。

我們將不再進一步討論這個進階主題。要注意的是，在 Algorithm 16.3 的每次迭代中添加或刪除**至多一個**限制，對到達最佳解所需的迭代次數就設置了自然下界。例如一個最佳化問題在解 x_* 上有 m 個 active 不等式限制，若從一個嚴格滿足所有不等式限制的點 x^0 開始進行迭代，算法至少需要 m 次迭代才能從 x^0 移到 x_* 。如果算法在某次迭代中添加了限制 j 到工作集中然後在稍後的步驟中又將其刪除，則需要進行更多次迭代。

§16.5 Active-Set Methods for Convex QPs

將最負的 Lagrange 乘子對應的限制刪除的策略在實踐中通常效果良好，但其缺點是容易受到限制式縮放的影響（通過將限制 i 乘以一正數 β 不會改變優化問題的結構，但會引入相應乘子 λ_i 的縮放 $1/\beta$ 倍）。選擇最負的乘子類似於線性規劃中 simplex 法的 Dantzig 原始樞紐規則（參見第 13 章），正如我們在那裡所指出的，對縮放不太敏感的策略通常會得到更好的結果。

我們將不再進一步討論這個進階主題。要注意的是，在 Algorithm 16.3 的每次迭代中添加或刪除**至多一個**限制，對到達最佳解所需的迭代次數就設置了自然下界。例如一個最佳化問題在解 x_* 上有 m 個 active 不等式限制，若從一個嚴格滿足所有不等式限制的點 x^0 開始進行迭代，算法至少需要 m 次迭代才能從 x^0 移到 x_* 。如果算法在某次迭代中添加了限制 j 到工作集中然後在稍後的步驟中又將其刪除，則需要進行更多次迭代。

§16.5 Active-Set Methods for Convex QPs

• Finite termination of active-set algorithm on strictly convex QPs

可以證明對於嚴格凸 QP，在某些假設下 Algorithm 16.3 在有限次迭代後會收斂；意即演算法會在有限次迭代後識別出解 x_* 。如果我們假設該方法在從 (31) 計算的方向 p^k 非零時總是採用非零的步長 α_k ，則這個說法顯然是正確的。我們的論證如下：

- 如果 (31) 的解是 $p^k = 0$ ，則當前點 x_k 是對於工作集合 \mathcal{W}_k 的 q 的唯一 global minimizer。如果它不是原問題 (1) 的解（即，至少一個 Lagrange 乘子為負），前述定理表明在刪除 blocking constraint 後計算的步進量 p^{k+1} 將是 q 的嚴格下降方向。因此，由於我們的假設 $\alpha_k > 0$ ，我們得出結論，在所有後續迭代中， q 的值都低於 $q(x_k)$ 。由於後續迭代的 q 值低於該工作集合的全域最小值，因此算法永遠不會返回工作集合 \mathcal{W}_k 。

§16.5 Active-Set Methods for Convex QPs

- **Finite termination of active-set algorithm on strictly convex QPs**
 可以證明對於嚴格凸 QP，在某些假設下 Algorithm 16.3 在有限次迭代後會收斂；意即演算法會在有限次迭代後識別出解 x_* 。如果我們假設該方法在從 (31) 計算的方向 p^k 非零時總是採用非零的步長 α_k ，則這個說法顯然是正確的。我們的論證如下：
 - 如果 (31) 的解是 $p^k = 0$ ，則當前點 x_k 是對於工作集合 \mathcal{W}_k 的 q 的唯一 global minimizer。如果它不是原問題 (1) 的解（即，至少一個 Lagrange 乘子為負），前述定理表明在刪除 blocking constraint 後計算的步進量 p^{k+1} 將是 q 的嚴格下降方向。因此，由於我們的假設 $\alpha_k > 0$ ，我們得出結論，在所有後續迭代中， q 的值都低於 $q(x_k)$ 。由於後續迭代的 q 值低於該工作集合的全域最小值，因此算法永遠不會返回工作集合 \mathcal{W}_k 。

§16.5 Active-Set Methods for Convex QPs

- 該演算法在連續 n 次迭代中都會遇到一個迭代 k 使得 $p^k = 0$ 是 (31) 的解。為了證明這一點，我們注意到對任何 $p^k \neq 0$ 的 k ，要麼我們有 $\alpha_k = 1$ （如前面 Remark 所述，在這種情況下下一個迭代將產生 $p^{k+1} = 0$ ），要麼是一個限制被添加到工作集 \mathcal{W}_k 中。如果後一種情況持續發生，那麼在最多 n 次迭代後，工作集將包含 n 個指標，這些指標對應於 n 個線性獨立的向量因而使 (31) 的解將為 $p^k = 0$ 。
- 綜合以上兩個說法，可以得出以下結論：該演算法定期（每 n 次迭代至少會有一次）在當前工作集上找到 q 的全域最小值，且一旦如此就不會再訪問這個特定的工作集。由於可能的工作集只有有限個，因此該算法不可能永遠迭代下去。最終，它會遇到一個滿足 (1) 最佳性條件的當前工作集的最小值因而給出問題的解。

§16.5 Active-Set Methods for Convex QPs

- 該演算法在連續 n 次迭代中都會遇到一個迭代 k 使得 $p^k = 0$ 是 (31) 的解。為了證明這一點，我們注意到對任何 $p^k \neq 0$ 的 k ，要麼我們有 $\alpha_k = 1$ （如前面 Remark 所述，在這種情況下下一個迭代將產生 $p^{k+1} = 0$ ），要麼是一個限制被添加到工作集 \mathcal{W}_k 中。如果後一種情況持續發生，那麼在最多 n 次迭代後，工作集將包含 n 個指標，這些指標對應於 n 個線性獨立的向量因而使 (31) 的解將為 $p^k = 0$ 。
- 綜合以上兩個說法，可以得出以下結論：該演算法定期（每 n 次迭代至少會有一次）在當前工作集上找到 q 的全域最小值，且一旦如此就不會再訪問這個特定的工作集。由於可能的工作集只有有限個，因此該算法不可能永遠迭代下去。最終，它會遇到一個滿足 (1) 最佳性條件的當前工作集的最小值因而給出問題的解。

§16.5 Active-Set Methods for Convex QPs

「我們可以總是沿著由 (31) 計算的非零下降方向 p^k 採取非零步長的假設，保證了該演算法不會進行循環」這句話中的循環指的是經過連續的迭代序列導致迭代 x 回到原處而工作集 \mathcal{W}_k 在刪除並添加 index 後也最終重複的情況，意即存在某些整數 k 和 $\ell \geq 1$ 使得 $x^k = x^{k+\ell}$ 和 $\mathcal{W}_k = \mathcal{W}_{k+\ell}$ 發生。在循環的每個迭代中，會捨棄一個限制（因有 Lagrange 乘子為負），但立即遇到一個不在 \mathcal{W}_k 中的新的限制 i ，而不會沿著計算得到的方向 p 移動。在處理二次規劃中的 degeneracy 和循環的步驟與第 13 章中討論的線性規劃類似；我們這裡不討論它們。大多數二次規劃的實現通常忽略了循環的可能性。

§16.5 Active-Set Methods for Convex QPs

• Updating factorizations

我們已經看到，Algorithm 16.3 中給出的 active-set 方法中的步驟計算需要解決等式限制子問題

$$\min_p \frac{1}{2} p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

正如本章開頭所提到的，這個計算相當於解決 KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

由於工作集在每次迭代中最多只會變動一個 index，所以 KKT 矩陣最多與上一次迭代的 KKT 矩陣相比只會有一行和一列的區別。實際上， G 保持不變，而與當前工作集相對應的 constraint gradient 矩陣 A 可能會通過添加和/或刪除單行而發生變化。

§16.5 Active-Set Methods for Convex QPs

根據此觀察，若我們可以通過更新在上一次迭代中計算的矩陣分解來計算在當前迭代中解決 (31) 所需的矩陣分解而不是從頭開始重新計算它們，這樣的更新技術對於 active-set 方法的效率至關重要。

以下我們將只討論使用 null-space 方法來計算步進量時的情況。假設 A 有 m 個線性獨立的行，並假設基底矩陣 Y 和 Z 是通過對 A^T 做 QR 分解定義的（詳見 §15.3）：即存在一個置換矩陣 Π 使得

$$A^T \Pi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad (40)$$

其中 R 是可逆的上三角方陣； $Q = [Q_1 : Q_2]$ 是 $n \times n$ 正交矩陣， Q_1 和 R 都有 m 行，而 Q_2 有 $n - m$ 行，然後定義

$$Y = Q_1, \quad Z = Q_2.$$

§16.5 Active-Set Methods for Convex QPs

根據此觀察，若我們可以通過更新在上一次迭代中計算的矩陣分解來計算在當前迭代中解決 (31) 所需的矩陣分解而不是從頭開始重新計算它們，這樣的更新技術對於 active-set 方法的效率至關重要。

以下我們將只討論使用 null-space 方法來計算步進量時的情況。假設 A 有 m 個線性獨立的行，並假設基底矩陣 Y 和 Z 是通過對 A^T 做 QR 分解定義的（詳見 §15.3）：即存在一個置換矩陣 Π 使得

$$A^T \Pi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad (40)$$

其中 R 是可逆的上三角方陣； $Q = [Q_1 : Q_2]$ 是 $n \times n$ 正交矩陣， Q_1 和 R 都有 m 行，而 Q_2 有 $n - m$ 行，然後定義

$$Y = Q_1, \quad Z = Q_2.$$

§16.5 Active-Set Methods for Convex QPs

以 null-space 方法來計算步進量時，我們必須依序解以下系統：

$$(AY)p_Y = -h, \quad (15)$$

$$(Z^T G Z)p_Z = -Z^T G Y p_Y - Z^T g, \quad (16)$$

$$(AY)^T \lambda_* = Y^T (g + Gp), \quad (17)$$

其中 (17) 式等號右邊的 p 來自於

$$p = Y p_Y + Z p_Z. \quad (14)$$

§16.5 Active-Set Methods for Convex QPs

假設在下一次迭代中將一個限制條件添加到工作集中，因此新的限制條件矩陣為 $\bar{A}^T = [A^T : a]$ ，其中 a 是長度為 n 的行向量且 \bar{A}^T 有滿秩。由於 $Q_1 Q_1^T + Q_2 Q_2^T = I$ ，我們有

$$\bar{A}^T \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix} = [A^T \Pi : a] = Q \begin{bmatrix} R & Q_1^T a \\ 0 & Q_2^T a \end{bmatrix}. \quad (41)$$

假設 \hat{Q} 是個可將向量 $Q_2^T a$ 轉到除第一個元素外所有元素都是零的向量之正交矩陣；亦即 \hat{Q} 滿足

$$\hat{Q} (Q_2^T a) = \begin{bmatrix} \gamma \\ 0 \end{bmatrix},$$

其中 $\gamma = \|Q_2^T a\|$ 是一個純量。從 (41) 式我們有

$$\bar{A}^T \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix} = Q \begin{bmatrix} R & Q_1^T a \\ 0 & \hat{Q}^T \begin{bmatrix} \gamma \\ 0 \end{bmatrix} \end{bmatrix} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \\ 0 & 0 \end{bmatrix}.$$

§16.5 Active-Set Methods for Convex QPs

假設在下一次迭代中將一個限制條件添加到工作集中，因此新的限制條件矩陣為 $\bar{A}^T = [A^T : a]$ ，其中 a 是長度為 n 的行向量且 \bar{A}^T 有滿秩。由於 $Q_1 Q_1^T + Q_2 Q_2^T = I$ ，我們有

$$\bar{A}^T \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix} = [A^T \Pi : a] = Q \begin{bmatrix} R & Q_1^T a \\ 0 & Q_2^T a \end{bmatrix}. \quad (41)$$

假設 \hat{Q} 是個可將向量 $Q_2^T a$ 轉到除第一個元素外所有元素都是零的向量之正交矩陣；亦即 \hat{Q} 滿足

$$\hat{Q} (Q_2^T a) = \begin{bmatrix} \gamma \\ 0 \end{bmatrix},$$

其中 $\gamma = \|Q_2^T a\|$ 是一個純量。從 (41) 式我們有

$$\bar{A}^T \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix} = Q \begin{bmatrix} R & Q_1^T a \\ 0 & \hat{Q}^T \begin{bmatrix} \gamma \\ 0 \end{bmatrix} \end{bmatrix} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \\ 0 & 0 \end{bmatrix}.$$

§16.5 Active-Set Methods for Convex QPs

假設在下一次迭代中將一個限制條件添加到工作集中，因此新的限制條件矩陣為 $\bar{A}^T = [A^T : a]$ ，其中 a 是長度為 n 的行向量且 \bar{A}^T 有滿秩。由於 $Q_1 Q_1^T + Q_2 Q_2^T = I$ ，我們有

$$\bar{A}^T \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix} = [A^T \Pi : a] = Q \begin{bmatrix} R & Q_1^T a \\ 0 & Q_2^T a \end{bmatrix}. \quad (41)$$

假設 \hat{Q} 是個可將向量 $Q_2^T a$ 轉到除第一個元素外所有元素都是零的向量之正交矩陣；亦即 \hat{Q} 滿足

$$\hat{Q} (Q_2^T a) = \begin{bmatrix} \gamma \\ 0 \end{bmatrix},$$

其中 $\gamma = \|Q_2^T a\|$ 是一個純量。從 (41) 式我們有

$$\bar{A}^T \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix} = Q \begin{bmatrix} R & Q_1^T a \\ 0 & \hat{Q}^T \begin{bmatrix} \gamma \\ 0 \end{bmatrix} \end{bmatrix} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \\ 0 & 0 \end{bmatrix}.$$

§16.5 Active-Set Methods for Convex QPs

上頁的矩陣分解有以下形式

$$\bar{A}^T \bar{\Pi} = \bar{Q} \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix},$$

其中

$$\bar{\Pi} = \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix}, \quad \bar{Q} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} = [Q_1 : Q_2 \hat{Q}^T], \quad \bar{R} = \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \end{bmatrix}.$$

如果我們知道 $Z (= Q_2)$ 的確切形式並且需要 \bar{Z} 的確切形式時，我們需要考慮獲得 \hat{Q} 的計算量以及形成乘積 $Q_2 \hat{Q}^T = Z \hat{Q}^T$ 的計算量。由於 \hat{Q} 的特殊結構，這個計算量是 $\mathcal{O}(n(n-m))$ ，而從頭開始計算 \bar{A}^T 的 QR 分解的計算量是 $\mathcal{O}(n^2 m)$ 。相比之下更新策略計算量較低，特別是當 null-space 較小時（即，當 $n-m \ll n$ 時）。

§16.5 Active-Set Methods for Convex QPs

上頁的矩陣分解有以下形式

$$\bar{A}^T \bar{\Pi} = \bar{Q} \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix},$$

其中

$$\bar{\Pi} = \begin{bmatrix} \Pi & 0 \\ 0 & 1 \end{bmatrix}, \quad \bar{Q} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} = [Q_1 : Q_2 \hat{Q}^T], \quad \bar{R} = \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \end{bmatrix}.$$

如果我們知道 $Z (= Q_2)$ 的確切形式並且需要 \bar{Z} 的確切形式時，我們需要考慮獲得 \hat{Q} 的計算量以及形成乘積 $Q_2 \hat{Q}^T = Z \hat{Q}^T$ 的計算量。由於 \hat{Q} 的特殊結構，這個計算量是 $\mathcal{O}(n(n-m))$ ，而從頭開始計算 \bar{A}^T 的 QR 分解的計算量是 $\mathcal{O}(n^2 m)$ 。相比之下更新策略計算量較低，特別是當 null-space 較小時（即，當 $n-m \ll n$ 時）。

§16.5 Active-Set Methods for Convex QPs

對於從 A 中刪除一列的情況，也可以設計一種更新技術。這個操作會從 (40) 中刪除 R 中的一行，因而可能在該矩陣的主對角線下方立即引入一些非零數字而擾亂了這個矩陣的上三角性質。可以通過應用一系列平面旋轉來恢復上三角性。這些旋轉將在 Q 的前 m 列中引入一些計算量低的轉換，並且在轉換完成後，可以從這個矩陣中選擇最後 $n - m + 1$ 列來獲得更新後的 null-space 基底矩陣。在這種情況下，新的 null-space 基底矩陣具有如下形式：

$$\bar{Z} = [\bar{z} : Z]; \quad (42)$$

也就是說，當前的矩陣 Z 被增加了一列。這個操作的總成本隨著從 A 中刪除的列的位置而變化，但通常比從頭開始重新計算 QR 分解的計算量少。有關這些程序的詳細信息，請參見 Gill 等人的文章 [124, 第 5 節]。

§16.5 Active-Set Methods for Convex QPs

現在我們考慮 reduced Hessian 矩陣。由於問題

$$\min_p \frac{1}{2} p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

的特殊形式，我們在其對應的 KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

中有 $h = 0$ ，因而根據 $p = Yp_Y + Zp_Z$ 所給出的步進量 $p_Y \equiv (AY)^{-1}h$ 為零。因此，由 (16) 式我們得到 null-space 分量 p_Z 是方程式

$$(Z^T G Z)p_Z = -Z^T g \quad (43)$$

的解。

§16.5 Active-Set Methods for Convex QPs

現在我們考慮 reduced Hessian 矩陣。由於問題

$$\min_p \frac{1}{2} p^T G p + g_k^T p \quad (31a)$$

subject to

$$a_i^T p = 0, i \in \mathcal{W}_k. \quad (31b)$$

的特殊形式，我們在其對應的 KKT 系統

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda_* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}, \quad (4)$$

中有 $h = 0$ ，因而根據 $p = Yp_Y + Zp_Z$ 所給出的步進量 $p_Y \equiv (AY)^{-1}h$ 為零。因此，由 (16) 式我們得到 null-space 分量 p_Z 是方程式

$$(Z^T G Z)p_Z = -Z^T g \quad (43)$$

的解。

§16.5 Active-Set Methods for Convex QPs

我們有時可以找到當 Z 發生變化時更新 reduced Hessian 矩陣 $Z^T G Z$ 的分解方式。假設我們有當前 reduced Hessian 矩陣的 Cholesky 分解，寫為 $Z^T G Z = L L^T$ ，並且在下一步中 Z 如

$$\bar{Z} = [\bar{z} : Z]; \quad (42)$$

中所示發生變化，即（因為有 blocking constraint）刪除一個限制後新的 A 的 null space 增加了一維（導致其 null space basis matrix Z 增加一行）。一系列計算量低的基本操作可以用來將 Cholesky 分解因子 L 轉換為新的因子 \bar{L} ，用於新的簡化的 Hessian 矩陣 $\bar{Z}^T G \bar{Z}$ 。

還有許多其他簡化的可能性。例如，正如在 §16.7 討論的那樣，我們可以在更新 Z 到 \bar{Z} 的同時更新簡化梯度 $Z^T g$ 。

§16.7 The Gradient Projection Method

在第 16.5 節描述的 active-set 方法中，active set 和工作集通常每次迭代只會以單一 index 緩慢變化。因此，這種方法遇到規模大的問題可能需要多次迭代才能收斂。例如，如果起始點 x_0 沒有 active constraint，而在（非退化的）解中有 200 個限制是 active，那麼至少需要 200 次 active-set 方法的迭代才能得到解。

梯度投影 (gradient projection) 法允許 active set 在迭代之間快速變化。當限制式的形式簡單（尤其是限制式為各變量有上下界）時，它的效率最高。因此，我們將注意力限制在以下 bound-constrained 問題上：

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \quad (44a)$$

subject to

$$l \leq x \leq u, \quad (44b)$$

其中 G 是對稱矩陣， l 和 u 分別是 x 分量的下界和上界向量。

§16.7 The Gradient Projection Method

在第 16.5 節描述的 active-set 方法中，active set 和工作集通常每次迭代只會以單一 index 緩慢變化。因此，這種方法遇到規模大的問題可能需要多次迭代才能收斂。例如，如果起始點 x_0 沒有 active constraint，而在（非退化的）解中有 200 個限制是 active，那麼至少需要 200 次 active-set 方法的迭代才能得到解。

梯度投影 (gradient projection) 法允許 active set 在迭代之間快速變化。當限制式的形式簡單（尤其是限制式為各變量有上下界）時，它的效率最高。因此，我們將注意力限制在以下 bound-constrained 問題上：

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \quad (44a)$$

subject to

$$l \leq x \leq u, \quad (44b)$$

其中 G 是對稱矩陣， l 和 u 分別是 x 分量的下界和上界向量。

§16.7 The Gradient Projection Method

在本節中，我們不對 G 進行任何正定性假設，因為梯度投影法可以應用於凸和非凸問題。因為由 (44b) 所定義的可行區域呈矩形形狀，它有時被稱為“box”。 x 的某些分量可能缺乏上界或下界；這樣的情況我們通過將 l 和 u 的適當分量分別設置為 $-\infty$ 和 $+\infty$ 來進行處理。另外，不失一般性我們假設 $l < u$ 。

梯度投影算法的每一次迭代包括兩個階段。在第一階段中，我們沿著從當前點 x 出發的最速下降方向 $-g$ 搜索，其中 $g = Gx + c$ 。每當遇到一個界限時，搜索方向就會被“彎曲”以確保保持可行性。我們沿著由此產生的分段線性路徑搜索，並找到函數 q 的第一個局部 minimizer，我們將其記為 x^c ，並類比於我們在第 4 章中的術語稱之為 Cauchy point。在梯度投影法中，工作集被定義為在 Cauchy point 處 active 的限制並表示為 $\mathcal{A}(x^c)$ 。

§16.7 The Gradient Projection Method

在本節中，我們不對 G 進行任何正定性假設，因為梯度投影法可以應用於凸和非凸問題。因為由 (44b) 所定義的可行區域呈矩形形狀，它有時被稱為“box”。 x 的某些分量可能缺乏上界或下界；這樣的情況我們通過將 l 和 u 的適當分量分別設置為 $-\infty$ 和 $+\infty$ 來進行處理。另外，不失一般性我們假設 $l < u$ 。

梯度投影算法的每一次迭代包括兩個階段。在第一階段中，我們沿著從當前點 x 出發的最速下降方向 $-g$ 搜索，其中 $g = Gx + c$ 。每當遇到一個界限時，搜索方向就會被“彎曲”以確保保持可行性。我們沿著由此產生的分段線性路徑搜索，並找到函數 q 的第一個局部 minimizer，我們將其記為 x^c ，並類比於我們在第 4 章中的術語稱之為 Cauchy point。在梯度投影法中，工作集被定義為在 Cauchy point 處 active 的限制並表示為 $\mathcal{A}(x^c)$ 。

§16.7 The Gradient Projection Method

在每個梯度投影迭代的第二階段，我們通過解決一個子問題來探索 Cauchy point 所在的可行區域（即 box）的邊界（面），其中對於屬於 $\mathcal{A}(x^c)$ 的 i 其所對應之分量 x_i 被固定為值 x_i^c 。

我們在本節的其餘部分詳細描述梯度投影方法。在本節中，我們的慣例是用上標表示迭代次數（即以 x^k 表示第 k 次迭代後的 x 值），並使用下標表示向量的分量。

§16.7 The Gradient Projection Method

• Cauchy point computation

接下來我們推導將最速下降方向投影到可行區域上所得到的分段線性路徑的明確表達式，並概述沿該路徑的第一個局部最小值 q 的搜索過程。

將點 x “投影” 到可行區域 (44b) 之投影點（在此投影點是指可行區域中離 x 最近的點）的第 i 個分量為

$$P(x, \ell, u)_i = \begin{cases} \ell_i & \text{if } x_i < \ell_i, \\ x_i & \text{if } x_i \in [\ell_i, u_i], \\ u_i & \text{if } x_i > u_i. \end{cases} \quad (45)$$

由 x 點出發沿著在 x 點的最速下降方向 $-g$ 延伸的射線投影到可行區域 (44b) 上得到的分段線性路徑 $x(t)$ 可以表示為：

$$x(t) = P(x - tg, \ell, u) \quad t \geq 0, \quad (46)$$

其中 $g = Gx + c$ 。參見圖 4。

§16.7 The Gradient Projection Method

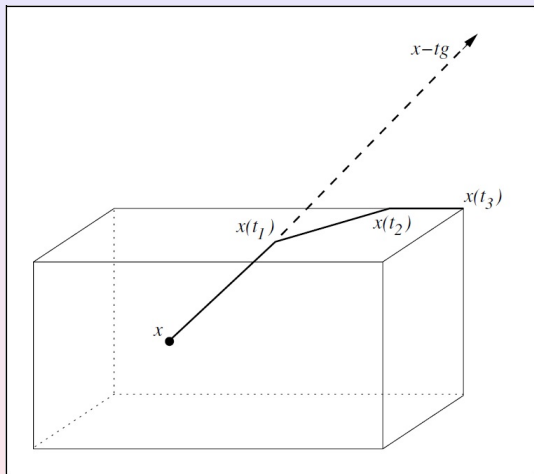


Figure 4: The piecewise-linear path $x(t)$, for an example in \mathbb{R}^3 .

§16.7 The Gradient Projection Method

Cauchy point x^c 定義為單變數分段二次函數 $q(x(t))$ 在 t 從零開始遞增後所遭遇的第一個局部極小值。這個極小值可以通過檢查構成 $x(t)$ 的每個線段來獲得。為了進行搜尋這個極小值，我們需要先確定在 $x(t)$ 中出現的拐點或斷點的 t 值。我們首先確定每個分量沿所選方向 $-g$ 。這些值 \bar{t}_i 由以下的公式給出：

$$\bar{t}_i = \begin{cases} (x_i - u_i)/g_i & \text{if } g_i < 0 \text{ and } u_i < +\infty, \\ (l_i - x_i)/g_i & \text{if } g_i > 0 \text{ and } l_i > -\infty, \\ \infty & \text{otherwise.} \end{cases} \quad (47)$$

因此，對於任何 t ， $x(t)$ 的分量如下所示：

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

§16.7 The Gradient Projection Method

為了找到函數 q 在 $x(t) = P(x - tg, \ell, u)$ 之函數值在 t 從零開始遞增所遭遇的第一個局部最小值，我們先消除集合 $\{\bar{t}_1, \bar{t}_2, \dots, \bar{t}_n\}$ 中的重複值和零值，以獲得重新排序（及編號）後的簡化斷點集 $\{t_1, t_2, \dots, t_\ell\}$ ，其中 $0 < t_1 < t_2 < \dots < t_\ell$ 。接下來我們依次檢查區間 $[0, t_1]$ 、 $[t_1, t_2]$ 、 $[t_2, t_3]$ 、 \dots 。假設我們已經檢查了 t_{j-1} 並且還沒有找到局部最小值。對於區間 $[t_{j-1}, t_j]$ ，我們有：

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

在此

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

且

$$p_i^{j-1} = \begin{cases} -g_i & \text{if } t_{j-1} < \bar{t}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (48)$$

§16.7 The Gradient Projection Method

然後，我們可以將在線段 $[x(t_{j-1}), x(t_j)]$ 上的二次函數 (44a) 寫成如下形式：

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

展開上式並將係數依 order 為 1、 Δt 和 $(\Delta t)^2$ 分組後，我們發現

$$q(x(t)) = f_{j-1} + f'_{j-1}\Delta t + \frac{1}{2}f''_{j-1}(\Delta t)^2, \quad \Delta t \in [0, t_j - t_{j-1}], \quad (49)$$

其中係數的定義為

$$f_{j-1} \equiv c^T x(t_{j-1}) + \frac{1}{2} x(t_{j-1})^T G x(t_{j-1}),$$

$$f'_{j-1} \equiv c^T p^{j-1} + x(t_{j-1})^T G p^{j-1},$$

$$f''_{j-1} \equiv (p^{j-1})^T G p^{j-1}.$$

§16.7 The Gradient Projection Method

將 (49) 對 Δt 微分並設為零，我們得到 $q \circ x$ 在 $\Delta t^* = -f'_{j-1}/f''_{j-1}$ 處導數為零。可能有以下幾種情況：

- 1 如果 $f'_{j-1} > 0$ ，則 $q \circ x$ 在 t_{j-1} 處導數為正，因而在 $t = t_{j-1}$ 處存在 $q(x(t))$ 的局部最小值；否則
- 2 如果 $f'_{j-1} \leq 0$ 且 $\Delta t^* \in [0, t_j - t_{j-1})$ ，則必有 $f''_{j-1} > 0$ 因而 f 在 $t = t_{j-1} + \Delta t^*$ 處取到局部最小值；
- 3 在其他所有情況下，我們移動到下一個區間 $[t_j, t_{j+1}]$ 並繼續搜索。

對於下一個搜索區間，我們需要根據 (48) 計算新方向 p^j ，並使用這個新值計算 f_j 、 f'_j 以及 f''_j 。由於 p^j 通常只在一個分量上與 p^{j-1} 有所不同，因此可以通過更新這些係數而不是從頭計算來節省計算量。

§16.7 The Gradient Projection Method

• Subspace minimization

在計算了 Cauchy point x^c 之後，其分量與該分量之上界或下界是否相等定義了 active set

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

在梯度投影迭代的第二階段，我們將對於 $i \in \mathcal{A}(x^c)$ 的 x_i 固定為值 x_i^c 並解以下的 QP 子問題：

$$\min_x q(x) = \frac{1}{2}x^T Gx + x^T c \quad (50a)$$

subject to

$$x_i = x_i^c, i \in \mathcal{A}(x^c), \quad (50b)$$

$$l_i \leq x_i \leq u_i, i \notin \mathcal{A}(x^c). \quad (50c)$$

注意到精確解決以上子問題是不可取的，尤其在問題規模大的情況下此子問題可能幾乎與原問題 (44) 一樣困難。

§16.7 The Gradient Projection Method

事實上，為了獲得梯度投影過程的全域收斂，我們只需要 (50) 的近似解 x^+ 是可行的且在該點的目標函數值不比在 x^c 的值更差（也就是 $q(x^+) \leq q(x^c)$ ）即可。一種介於選擇 $x^+ = x^c$ 作為近似解和精確解決之間的策略是通過使用 Algorithm 16.1 或 Algorithm 16.2 中描述的 CG 迭代計算 (50) 的近似解。

注意，對於等式限制 (50b)，Jacobian A 和 null-space 基底矩陣 Z 有特別簡單的形式。因此，我們可以將 CG 應用於問題 (50a)、(50b)，並在遇到界限 $\ell \leq x \leq u$ 時立即終止。

或者，我們可以繼續迭代，暫時忽略邊界，並將解投影回 box constraint 上。負曲率的情況可以像 Algorithm 7.2 中一樣處理，該算法是用於近似解可能是不定的 trust-region 子問題的無限制優化方法。

§16.7 The Gradient Projection Method

我們對二次規劃的梯度投影算法進行總結如下：

Algorithm 16.5 (Gradient Projection Method for QP).

Compute a feasible starting point x_0 ;

for $k = 0, 1, 2, \dots$

if x^k satisfies the KKT conditions for (44)

stop with solution $x_* = x^k$;

 Set $x = x^k$ and find the Cauchy point x^c ;

 Find an approximate solution x^+ of (50) such that $q(x^+) \leq q(x^c)$ and x^+ is feasible;

$x^{k+1} \leftarrow x^+$;

end (for)

§16.7 The Gradient Projection Method

如果由演算法得到近似解 x_* 且在該點上所有活動邊界相關的拉格朗日乘子都不為零（也就是，嚴格互補性成立），那麼對於所有充分大的 k 來說梯度投影法生成的 active set $\mathcal{A}(x^c)$ 將等於最佳的 active set。也就是說，在連續的迭代中，限制式的 index 不會重複進入和離開 active set。當問題是 degenerate 的時候，active set 可能無法達到其最佳值。有很多種方法已被提出來防止這種不良行為發生。

§16.7 The Gradient Projection Method

儘管梯度投影法原則上可以應用於具有一般線性限制式的問題，但在這些情況下，可能需要進行大量計算來將投影進行到可行集合上。例如，如果限制集被定義為 $a_i^T x \geq b_i$, $i \in \mathcal{I}$ ，我們必須解決以下凸二次規劃問題來計算給定點 \bar{x} 在這個集合上的投影：

$$\max_x \|x - \bar{x}\|^2 \quad \text{subject to} \quad a_i^T x \geq b_i \quad \forall i \in \mathcal{I}.$$

解決這個“投影子問題”的成本可能接近解決原始二次規劃的成本，因此通常不會將梯度投影應用於這種情況。

§16.7 The Gradient Projection Method

當我們使用對偶性來用對偶替換一個嚴格凸二次規畫時（請參見第 12 章的最後一個例子），梯度投影法可能在解決 bound constrained 的對偶問題時很有用，該問題以 Lagrange 乘子 λ 表述如下：

$$\begin{aligned} \max_{\lambda} \tilde{q}(\lambda) &= -\frac{1}{2}(A^T\lambda - c)^T G^{-1}(A^T\lambda - c) + b^T\lambda \\ &\text{subject to} \quad \lambda \geq 0. \end{aligned}$$

（注意，對偶問題通常被寫作一個最大化問題；我們可以等價地最小化 $-\tilde{q}(\lambda)$ ，並注意到這個轉換後的問題也是凸 QP。）當 G 具有如對角矩陣或分塊對角矩陣的簡單形式時，改用對偶替換的方法是最有用的。

§16.8 Perspectives and Software

在 [149] 中報告的凸 QP 的 active-set 方法和內點法的數值比較表明，內點法通常在大型問題上要快得多。然而，如果需要 warm start，則 active-set 方法可能通常更可取。儘管已經有相當多的研究集中在改善內點方法的 warm start 能力上，但這些技術的全部潛力現在（成書時間 2006）還不清楚。

在本章中到目前為止的討論，我們都假設所有具等式限制的二次規劃都有線性獨立的限制，亦即 $m \times n$ constraint Jacobian 矩陣 A 有滿秩 m 。對於未知是否有冗餘限制的情況，可以通過 A^T 的奇異值分解 (SVD) 或 rank-revealing QR 分解來檢測，然後從問題中刪除它們。當 A 較大時，可以將稀疏高斯消去技術應用於 A^T ，但它們的可靠性較低。

§16.8 Perspectives and Software

在 [149] 中報告的凸 QP 的 active-set 方法和內點法的數值比較表明，內點法通常在大型問題上要快得多。然而，如果需要 warm start，則 active-set 方法可能通常更可取。儘管已經有相當多的研究集中在改善內點方法的 warm start 能力上，但這些技術的全部潛力現在（成書時間 2006）還不清楚。

在本章中到目前為止的討論，我們都假設所有具等式限制的二次規劃都有線性獨立的限制，亦即 $m \times n$ constraint Jacobian 矩陣 A 有滿秩 m 。對於未知是否有冗餘限制的情況，可以通過 A^T 的奇異值分解 (SVD) 或 rank-revealing QR 分解來檢測，然後從問題中刪除它們。當 A 較大時，可以將稀疏高斯消去技術應用於 A^T ，但它們的可靠性較低。

§16.8 Perspectives and Software

我們也尚未考慮在 Hessian 矩陣 G 是不定的情況下的 active-set 方法，因為這些方法很難描述，且如何適應維度大的情況尚不清楚。在這裡，我們對主要技術進行一些評論。

對於凸 QP 的 active-set 方法，Algorithm 16.3 可以通過在某些情況下修改搜索方向和步長的計算來適應 G 是不定的情況。為了解釋修改的必要性，我們考慮通過 null-space 方法計算步長；也就是說， $p = Zp_z$ ，其中 p_z 由 (43) 式給出。如果 reduced Hessian $Z^T G Z$ 是正定的，那麼步進量 p 指向子問題 (31) 的最小值，迭代的邏輯不需要改變。然而，如果 $Z^T G Z$ 有負特徵值，那麼 p 只指向 (31) 的一個鞍點，因此不總是適合作為一步。相反，我們尋找一個替代方向 s_z ，它是 $Z^T G Z$ 的負曲率方向。如此一來我們有

$$q(x + \alpha Z s_z) \rightarrow -\infty \quad \text{as} \quad \alpha \rightarrow \infty. \quad (51)$$

§16.8 Perspectives and Software

此外，如果必要的話，我們可以改變 s_z 的正負號以確保 Zs_z 是在當前點 x 對函數 q 的非上升方向；也就是說， $\nabla q(x)^T Zs_z \leq 0$ 。通過沿著方向 Zs_z 移動，我們將找到一個可以添加到下一次迭代的工作集的限制（如果我們找不到這樣的限制式，則問題是無界的）。如果新工作集的 reduced Hessian 不是正定的，我們將重複此過程，直到添加了足夠多的限制，使 reduced Hessian 變為正定。然而，這種通用方法的一個困難之處在於，如果允許 reduced Hessian 具有多個負特徵值，那麼當從一個工作集變為下一個工作集而導致 reduced Hessian 有所變化的情況，會使得這些方法很難高效。