MATH 3011 运筹学 **秋季 2024**

Lecture 1: 课程介绍和基础

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 运筹与最优化

运筹学（Operations Research, 简称 OR），专注于使用高级分析方法来帮助人们在实际问题中做出更好的决策。它结合了统计学、计算机科学以及其他决策科学工具，目的是优化复杂系统的性能和效率。

优化问题是运筹学中的一个核心概念，它通常涉及在给定一系列约束的情况下，寻找某个函数的最大值或最小值。这个函数被称为目标函数，可以是成本、利润、效率或其他度量值。

数学表述

一个优化问题可以用以下形式表述：

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0,\ i = 1, \ldots, m, \\ & h_j(x) = 0,\ j = 1, \ldots, p, \end{aligned} \tag{1.1}$$

其中 $f(x)$ 是**目标函数**，x 是**决策变量**，$g_i(x)$ 和 $h_j(x)$ 分别表示不等式和等式约束, 所有满足约束条件的解构成的集合称为**可行域**。

记问题(1.1)中不等式和等式约束确定的可行域为 S, 即

$$S = \{x \mid g_i(x) \leq 0,\ i = 1, \ldots, m, \quad h_j(x) = 0,\ j = 1, \ldots, p\}.$$

满足约束条件 $\mathbf{x} \in S$ 的 $\mathbf{x}$ 称为问题的**可行解** (feasible solution), 如果可行解 $\mathbf{x}^* \in S$ 进一步满足

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}), \forall \mathbf{x} \in S. \tag{1.2}$$

则称 $\mathbf{x}^*$ 为问题(1.1)的**全局最优解** (global optimal solution). 另外，在包含可行解 $\mathbf{x}^* \in S$ 的适当邻域 $U(\mathbf{x}^*)$ 里，成立

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}), \forall \mathbf{x} \in S \cap U(\mathbf{x}^*). \tag{1.3}$$

此时称 $\mathbf{x}^*$ 为问题(1.1)的**局部最优解** (local optimal solution).

在实际问题中，寻找最优解是优化的主要目标。不少问题的目标函数或约束条件可能很复杂，要找出全局最优解非常困难，这时我们的目标就会是求出局部最优解。

运筹学与优化的应用

运筹学与优化在许多领域都有广泛的应用，以下是一些典型的应用领域：

- **供应链管理** 在供应链管理中，优化用于确定最佳的存货水平、分配策略和物流路线。例如，通过优化模型，企业可以最小化运输成本同时确保及时的产品交付。
- **金融工程** 在金融领域，优化技术用于资产组合管理、风险评估和定价策略。例如，优化模型可以帮助投资者在风险和收益之间找到最佳平衡。
- **工业工程** 工业工程中的优化应用包括生产规划、设施布局优化和质量控制。例如，通过优化生产计划，工厂可以提高效率，减少浪费。
- **人工智能** 在人工智能领域，许多模型的训练过程实际上是一个优化问题，旨在最小化预测误差或损失函数。例如，梯度下降法被广泛用于调整模型参数以减少误差。
- **公共政策** 运筹学和优化在制定公共政策时也发挥重要作用，如环境政策制定、城市规划和资源分配。例如，优化技术可以帮助政府在限制预算的情况下有效分配公共资源。

Example 1.1 (投资组合优化) (*Harry Max Markowitz,* 马科维茨*, 1927* 年 *8* 月 *24* 日—*2023* 年 *6* 月 *22* 日)

马科维茨在 *1952* 年发表了他的开创性论文《投资组合选择》(*"Portfolio Selection"*)，在该论文中，他提出了有效边界 (*efficient frontier*) 的概念。这一理论展示了在给定预期回报率下，如何构建风险最小的投资组合。他的理论强调了风险和回报之间的平衡，并介绍了分散投资的重要性。

马科维茨的理论以均值 (代表回报) 和方差 (代表风险) 作为投资组合表现的关键参数。他的模型指出，投资者应该不仅仅关注投资回报的期望值，还要考虑其风险 (即回报的波动性)。这是首次在金融理论中系统地将风险量化并纳入投资决策过程。

1990 年，马科维茨因为在投资组合选择和资本市场价格形成理论方面的工作而与米尔顿 · 米勒和威廉 · 夏普共同获得诺贝尔经济学奖.

- r_i, 随机变量, 股票的回报率 i
- x_i, 投资于股票的相对金额 i
- 回报: $r = r_1x_1 + r_2x_2 + \ldots + r_nx_n$
- 期望回报: $R = E(r) = \sum E(r_i)x_i = \sum \mu_i x_i$
- 风险: $V = Var(r) = \sum_{i,j} \sigma_{ij}x_ix_j = x^\top \Sigma x$

$$\begin{aligned} \min & \frac{1}{2} x^\top \Sigma x, \\ \text{s.t.} & \sum \mu_i x_i \geq r_0 \\ & \sum x_i = 1, \\ & x_i \geq 0 \end{aligned} \qquad \begin{aligned} \min & \quad \text{risk measure}, \\ \text{s.t.} & \sum \mu_i x_i \geq r_0 \\ & \sum x_i = 1, \\ & x_i \geq 0 \end{aligned}$$

通过后面的课程，我们将学会如何求解该**带约束的非线性规划问题**。

Example 1.2 (神经网络) 深度神经网络（*Deep Neural Networks, DNNs*）的发展历史可以追溯到上个世纪 *40-50* 年代，经历了多个阶段的发展，至今已成为人工智能和机器学习领域的核心技术之一。*1980* 年代，随着**反向传播算法** *(Back-propagation)* 的提出，多层神经网络开始受到关注，这个算法能有效地训练多层网络。*2006* 年，*Hinton* 等人提出了深度置信网络（*Deep Belief Networks*），这标志着深度学习时代的开始。*2012* 年，*AlexNet* 在 *ImageNet* 竞赛中取得了显著成绩，引起了广泛关注。此后，深度学习在语音识别、图像识别、自然语言处理等领域取得了突破性进展。深度神经网络通常最小化如下问题

$$\min_{W,b} L(y, \hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i),$$

其中，y_i 是给定的值。$\hat{y}_i$ 是神经网络最后一层输出向量 $\hat{y}$ 的分量，其表达式满足如下关系：

$$\hat{y} = h(\mathbf{z}^{[L]}),$$

h 是某种分类函数，例如 *Sigmoid* 函数。$\mathbf{z}^{[L]}$ 是由下述关系确定的。给定 x 为输入，记 $a^{[0]} = x$，对于 $l = 1, 2, ..., L,$

$$\mathbf{z}^{[l]} = \mathbf{W}^{[l]} \mathbf{a}^{[l-1]} + \mathbf{b}^{[l]},$$

$$\mathbf{a}^{[l]} = g^{[l]}(\mathbf{z}^{[l]}),$$

这里，$g^{[l]}$ 是激活函数，是非线性函数。

所以，神经网络的损失函数，是由 L 层非线性函数复合构成的。

通过后面的课程，我们将知道，反向传播算法即为随机梯度算法，并且也将了解更多的随机优化算法。

优化问题的分类

根据目标函数和约束条件的不同特性，优化问题可以分为不同的类型：

- **线性规划（LP）**：目标函数和约束条件均为线性。
- **非线性规划（NLP）**：目标函数或约束条件至少有一个是非线性的。
- **整数规划（IP）**：部分或全部决策变量被限制为整数。

- **动态规划（DP）**：涉及决策过程中多个时间步骤的问题。
- **随机规划**：至少有一个参数是随机变量的优化问题。

而根据目标函数和可行域是否为具有凸性，优化问题分为两类：

- **凸问题：**目标函数是凸函数，并且可行域是凸集。
- **非凸问题：**目标函数是非凸函数，或者可行域为非凸集合。

在本课程中，我们将重点学习线性规划、非线性规划以及基础的动态规划问题。课程内容将涵盖优化问题的基本性质、求解这些问题的关键算法，以及与这些算法在实际中相关的应用问题。课程涉及的主要算法有线性规划中的单纯形法、网络优化中动态规划的经典算法，以及求解非线性规划问题的梯度下降法和牛顿法。此外，我们还将学习求解带约束的非线性规划凸问题的算法，如罚函数法、近似点梯度法和交替方向乘子法（ADMM）。通过这些内容，我们将能够深入理解优化理论，并掌握实际问题求解的有效工具。

2 一些数学基础回顾

2.1 梯度和海瑟矩阵

Definition 1.1 (梯度) 给定函数 $f:\mathbb{R}^n\to\mathbb{R}$，且 f 在点 x 的一个邻域内有意义，若存在向量 $g\in\mathbb{R}^n$ 满足

$$\lim_{p\to 0}\frac{f(x+p)-f(x)-g^{\mathrm{T}}p}{\|p\|}=0,$$

其中 $\|\cdot\|$ 是任意的向量范数，就称 f 在点 x 处**可微**（或 Fréchet 可微）. 此时 g 称为 f 在点 x 处的**梯度**，记作 $\nabla f(x)$. 如果对区域 D 上的每一个点 x 都有 $\nabla f(x)$ 存在，则称 f 在 D 上可微.

若 f 在点 x 处的梯度存在，在定义式中令 $p=\varepsilon e_i$，e_i 是第 i 个分量为 1 的单位向量，可知 $\nabla f(x)$ 的第 i 个分量为 $\dfrac{\partial f(x)}{\partial x_i}$. 因此，

$$\nabla f(x)=\left[\frac{\partial f(x)}{\partial x_1},\frac{\partial f(x)}{\partial x_2},\cdots,\frac{\partial f(x)}{\partial x_n}\right]^{\mathrm{T}}.$$

Definition 1.2 (雅可比矩阵) 若 $f:\mathbb{R}^n\to\mathbb{R}^m$ 是多值函数，雅可比矩阵定义如下

$$J(f)(\mathbf{x})=\begin{bmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n}\\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n}\end{bmatrix}$$

Definition 1.3 (海瑟矩阵) 如果函数 $f(x):\mathbb{R}^n\to\mathbb{R}$ 在点 x 处的二阶偏导数 $\frac{\partial^2 f(x)}{\partial x_i\partial x_j}$ $i,j=1,2,\cdots,n$ 都存在，则 f 在点 x 处的海瑟矩阵为：

$$\nabla^2 f(x)=\begin{bmatrix}\frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1\partial x_2} & \frac{\partial^2 f(x)}{\partial x_1\partial x_3} & \cdots & \frac{\partial^2 f(x)}{\partial x_1\partial x_n}\\ \frac{\partial^2 f(x)}{\partial x_2\partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \frac{\partial^2 f(x)}{\partial x_2\partial x_3} & \cdots & \frac{\partial^2 f(x)}{\partial x_2\partial x_n}\\ \vdots & \vdots & \vdots & & \vdots\\ \frac{\partial^2 f(x)}{\partial x_n\partial x_1} & \frac{\partial^2 f(x)}{\partial x_n\partial x_2} & \frac{\partial^2 f(x)}{\partial x_n\partial x_3} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2}\end{bmatrix}$$

当 $\nabla^2 f(x)$ 在区域 D 上的每个点 x 处都存在时，称 f 在 D 上二阶可微. 若 $\nabla^2 f(x)$ 在 D 上还连续，则称 f 在 D 上二阶连续可微，可以证明此时海瑟矩阵是一个对称矩阵.

我们有 $\nabla^2 f(x)=J(\nabla f(x))$.

Example 1.3 若 $f(x)=\frac{1}{2}\|Ax-b\|^2$, $A\in\mathbb{R}^{m\times n}, x\in\mathbb{R}^n$. 我们有

$$\nabla f(x)=A^T(Ax-b).$$

$$\nabla^2 f(x)=A^TA.$$

2.2 线性相关和线性无关

设 V 是一个向量空间，$\{v_1,v_2,\ldots,v_n\}$ 是 V 中的一组向量。则这组向量是线性无关的当且仅当下述条件成立：

$$a_1v_1+a_2v_2+\cdots+a_nv_n=0$$

只有当所有系数 $a_1,a_2,\ldots,a_n$ 都等于零时才成立。

这意味着没有任何一个向量可以表示为其它向量的线性组合。若存在一组不全为零的系数使得上述等式成立，则称这组向量是线性相关的。

故，若 $A\in\mathbb{R}^{m\times n}$ 的所有列向量线性无关，即 A 的列满秩的，那么方程

$$Ax=0$$

仅有零解。

2.3 矩阵的特征根分解和奇异值分解

设 $A\in\mathbb{R}^{n\times n}$ 为对称矩阵，那么存在一个正交矩阵 Q 和一个对角矩阵 Λ，使得：

$$A=Q\Lambda Q^T$$

其中，Q 的列向量是 A 的特征向量，而 Λ 是一个对角矩阵，其对角线上的元素是 A 的特征值。

对于任意矩阵 $A \in \mathbb{R}^{m\times n}$，存在奇异值分解：

$$A = U\Sigma V^T$$

其中，U 是一个 $m \times m$ 的正交矩阵，V 是一个 $n \times n$ 的正交矩阵，而 Σ 是一个 $m \times n$ 的对角矩阵。对角线上的元素（奇异值）是 A 的奇异值。

Definition 1.4 矩阵 $A \in \mathbb{R}^{n\times n}$ 是半正定的，如果对于所有向量 $x \in \mathbb{R}^n$，都有

$$x^T Ax \geq 0.$$

通常，我们称矩阵半正定还同时假设 A 是对称矩阵。因此，半正定等价于 A 的所有特征根非负。

Definition 1.5 矩阵 $A \in \mathbb{R}^{n\times n}$ 是正定的，如果对于所有非零向量 $x \in \mathbb{R}^n$，都有

$$x^T Ax > 0$$

正定等价于 A 的所有特征根严格大于 0。

2.4 矩阵内积

对于两个相同大小的矩阵 $A, B \in \mathbb{R}^{n\times n}$，它们的内积可以定义为

$$\langle A, B\rangle = \mathrm{Tr}(A^T B)$$

其中 Tr 表示迹（trace），即矩阵对角线元素的总和。故，该定义是拓展了向量的 l_2 范数。

迹具有可轮换性质，这意味着对于任意两个大小相同的矩阵 A 和 B，有

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$$

由此，可以定义矩阵的 Frobenius 范数 (F-范数)

$$\|A\|_F^2 = \mathrm{Tr}(A^T A).$$

3 凸集和凸函数

3.1 凸集

Definition 1.6 (凸集) 一个集合 $S \subset \mathbb{R}^n$ 是一个凸集，如果 $\forall x, y$ 以及 $t \in [0, 1]$,

$$tx + (1 - t)y \in S.$$

也就是说，经过 S 中不同的两点 x, y 确定的线段仍在 S 中。

Example 1.4 圆盘 $\{x : \|x\|_2 \leq 1\}$ 是一个凸集。

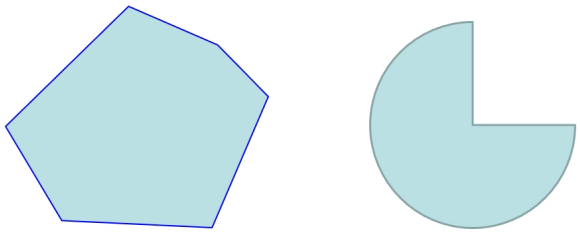


图 1.1: 凸集（左）和非凸集（右）。

Definition 1.7 (超平面，半空间) 给定 n 维非零向量 a 和实数 b,

1. 集合 $\{x \in \mathbb{R}^n \mid a^\top x = b\}$ 为超平面（*hyperplane).*
2. 集合 $\{x \in \mathbb{R}^n \mid a^\top x \geq b\}$ 为半空间（*halfspace).*

Example 1.5 $\mathbb{R}^2$ 中超平面为直线，例如直线 $x_1 + x_2 = 1$。

Definition 1.8 (多面体集) 多面体集是符合下述定义的集合

$$P = \{x \in \mathbb{R}^n \mid Ax \geq b\} = \{x \mid a_i^T x \geq b_i, i = 1, 2, \ldots, m\},$$

其中，矩阵 $A \in \mathrm{R}^{m \times n}$, 向量 $b \in \mathrm{R}^m$, a_i^T 为矩阵 A 的第 i 行，b_i 为向量 b 的第 i 的元素.

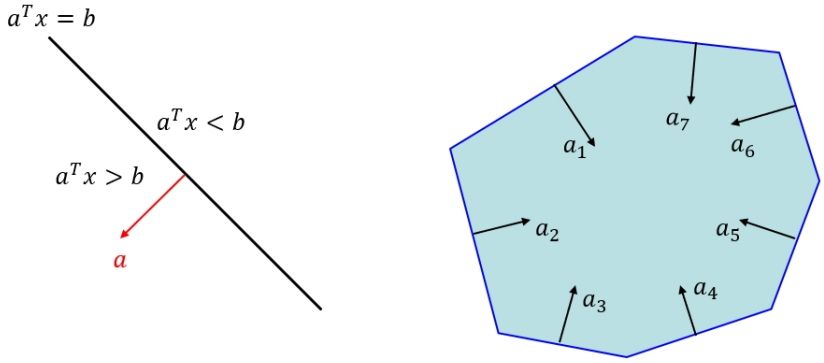


图 1.2: (a) 超平面和半空间。(b) 多面集

凸组合和凸包

从凸集中可以引出凸组合和凸包的概念.

Definition 1.9 (凸组合) 形如

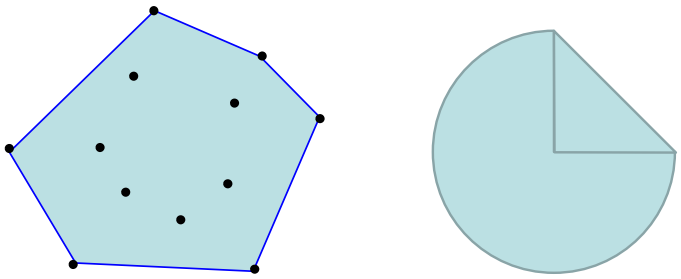
$$x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k,$$

$$\theta_1 + \cdots + \theta_k = 1, \theta_i \geqslant 0, i = 1, \cdots, k.$$

的点称为 $x_1, \cdots, x_k$ 的凸组合.

Definition 1.10 (凸包) 集合 $\mathcal{S}$ 的所有点的凸组合构成的点集为 $\mathcal{S}$ 的凸包, 记为 $\mathbf{conv}\mathcal{S}$.

例 在下图中我们列出了一些离散点集和连续点集的凸包. 其中, 左子图为离散点集的凸包, 右子图为扇形连续点集的凸包.



3.2 凸函数

Definition 1.11 (广义实值函数) 令 $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$ 为广义实数空间, 则映射 $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ 称为**广义实值函数**.

和数学分析一样, 我们规定

$$-\infty < a < +\infty, \quad \forall\, a \in \mathbb{R}$$
$$(+\infty) + (+\infty) = +\infty, \quad +\infty + a = +\infty, \forall\, a \in \mathbb{R}.$$

函数的定义域记为

$$\mathrm{dom} f = \{x \in \mathbb{R}^n \mid f(x) < \infty\}.$$

Definition 1.12 (适当函数) 给定广义实值函数 f 和非空集合 $\mathcal{X}$. 如果存在 $x \in \mathcal{X}$ 使得 $f(x) < +\infty$, 并且对任意的 $x \in \mathcal{X}$, 都有 $f(x) > -\infty$, 那么称函数 f 关于集合 $\mathcal{X}$ 是适当的.

概括来说，适当函数 f 的特点是 “至少有一处取值不为正无穷”，以及 “处处取值不为负无穷”. 表明定义域 $\mathrm{dom} f$ 非空。

Definition 1.13 (凸函数) $f : \mathbb{R}^n \to \mathbb{R}$ 为适当函数，如果 $\mathrm{dom}\ f$ 是凸集，且

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta) f(y)$$

对所有 $x, y \in \mathrm{dom}\ f$, $0 \le \theta \le 1$ 都成立，则称 f 是凸函数

注 1.1 若 $-f$ 是凸函数，则称 f 是凹函数。

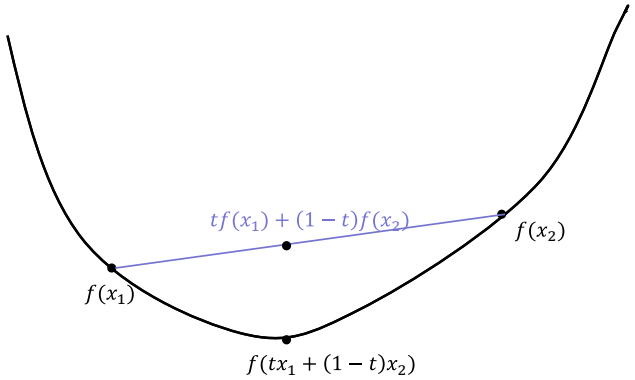


图 1.3: 凸函数图像. 根据定义可知，任意两点连线上的函数值小于等于两个函数值的凸组合。

凸函数有如下非常好的性质。

Theorem 1.1 *凸函数的局部最小值点为全局最小值点。*

Proof: 设 $\bar{x}$ 为其邻域 U 内的最小值点，假设 $\bar{x}$ 非全局最小值点，即存在 y s.t. $f(y) < f(x)$. 由于 $f(x)$ 的凸性，对于 $0 \le t \le 1$，我们有 $f((1-t)\bar{x} + ty) \le (1-t)f(\bar{x}) + tf(y) < f(\bar{x})$. 当 t 充分小时，$(1-t)\bar{x} + ty \in U$. 矛盾。 ■

然而对于非凸函数，可能存在非常多的局部极小值，使得问题相当困难。

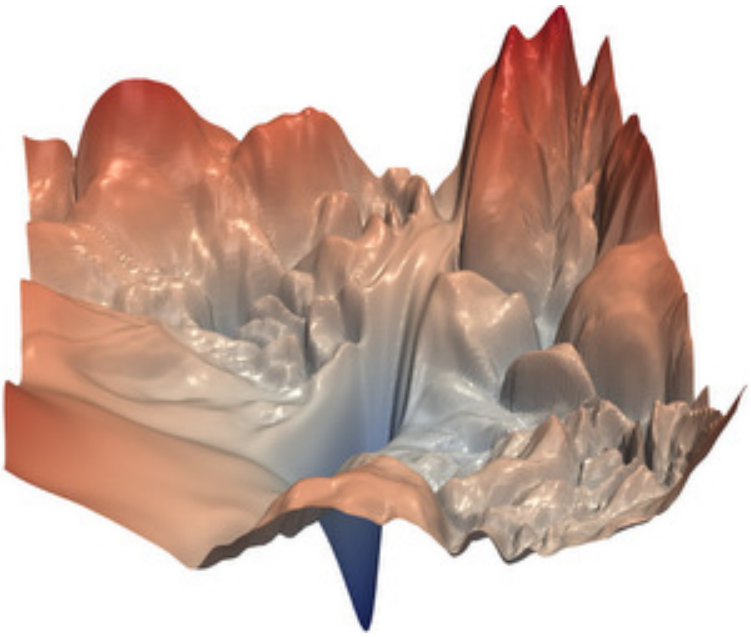


图 1.4: 神经网络 ResNet56 的损失函数局部图示。图片来源：https://github.com/tomgoldstein/loss-landscape

3.3 凸函数判定条件

如何判断函数是否为凸函数，除了使用定义，也可使用下面一阶和二阶判定条件。

Theorem 1.2 (判定凸函数的一阶条件) *f 是一个凸集上的可微函数。f 是凸函数，当且仅当其满足*

$$f(y) \ge f(x) + \nabla f(x)^T (y - x), \quad \forall x, y \in \mathrm{dom} f. \tag{1.4}$$

证明：

- (1.4)→ 凸函数：对任意的 $x, y \in \mathrm{dom} f$ 以及任意的 $t \in (0,1)$，定义 $z = tx + (1-t)y$，应用两次一阶条件我们有

$$f(x) \ge f(z) + \nabla f(z)^T (x - z),$$

$$f(y) \geq f(z) + \nabla f(z)^T (y - z).$$

将上述第一个不等式两边同时乘 t，第二个不等式两边同时乘 $1-t$，相加得

$$tf(x) + (1-t)f(y) \geq f(z).$$

这正是凸函数的定义，因此充分性成立。

- 凸函数 $\rightarrow$(1.4)：设 f 是凸函数。对于任意可行域的 x, y, 以及 $t \in (0,1)$, 根据凸函数定义，可得

$$tf(y) + (1-t)f(x) \geq f(x + t(y-x)).$$

由上式，经过移项处理，两边同时除以 t 可得

$$f(y) - f(x) \geq \frac{f(x + t(y-x)) - f(x)}{t}.$$

令 $t \to 0$, 因为极限保号性，可得

$$f(y) - f(x) \geq \lim_{t \to 0} \frac{f(x + t(y-x)) - f(x)}{t} = \nabla f(x)^T (y - x).$$

证毕。

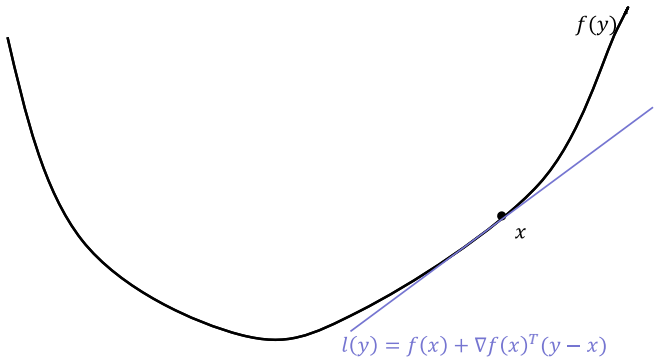


图 1.5: **定理 1.2的直观解释:** 任意点 x 的切线总是在函数图像下方。

Theorem 1.3 (判定凸函数二阶条件) *f 是一个凸集上的二阶连续可微。f 是凸函数，当且仅当其满足*

$$\nabla^2 f(x) \succcurlyeq 0.$$

证明.

- **必要性：** 反设 $f(x)$ 在点 x 处的 Hessian 矩阵 $\nabla^2 f(x) \not\succeq 0$，即存在非零向量 $v \in \mathbb{R}^n$ 使得 $v^T \nabla^2 f(x) v < 0$。根据二阶可微性，有泰勒展开，

  $$f(x+tv) = f(x) + t\nabla f(x)^T v + \frac{t^2}{2} v^T \nabla^2 f(x) v + o(t^2).$$

  移项后等式两边同时除以 t^2，

  $$\frac{f(x+tv) - f(x) - t\nabla f(x)^T v}{t^2} = \frac{1}{2} v^T \nabla^2 f(x) v + o(1).$$

  当 t 充分小时，

  $$\frac{f(x+tv) - f(x) - t\nabla f(x)^T v}{t^2} < 0,$$

  这显然和一阶条件矛盾，因此必有 $\nabla^2 f(x) \geq 0$ 成立。

- **充分性：** 若 f 满足二阶条件。对于可行域的任意 x, y, 根据二阶可微性，

  $$f(y) = f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x),$$

  这里 z 是 x, y 连线上的一个点。由于 $\nabla^2 f(z) \succcurlyeq 0$, 可知一阶条件成立。故 f 是凸函数。

Example 1.6 $f(x) = \frac{1}{2}\|Ax-b\|^2$. 则 $\nabla f(x) = A^T(Ax-b)$, $\nabla^2 f(x) = A^T A$ 是半正定的。故其为凸函数。

Example 1.7 log-sum-exp 函数 $f(x) = \log \sum_{k=1}^n \exp x_k$ 是凸函数

$$\nabla^2 f(x) = \frac{1}{\sum_k z_k} \mathrm{diag}(z) - \frac{1}{(\sum_k z_k)^2} z z^T$$

这里 $z_k = \exp x_k$, $\mathrm{diag}(z)$ 表示对角线为向量 z 的对角矩阵.

To prove $\nabla^2 f(x) \geq 0$, we only need to prove for any v, $v^T \nabla^2 f(x) v \geq 0$, i.e.,

$$v^T \nabla^2 f(x) v = \frac{\sum_k z_k v_k^2}{\sum_k z_k} - \frac{(\sum_k v_k z_k)^2}{(\sum_k z_k)^2} \geq 0$$

Using the Cauchy-Schwarz inequality, we have $(\sum_k v_k z_k)^2 \leq (\sum_k z_k v_k^2)(\sum_k z_k)$, thus f is a convex function.

3.3.1 判定凸函数的其他条件 *

非负数乘: 若 f 是凸函数，则 αf 是凸函数，其中 $\alpha \geq 0$.

求和: 若 f_1, f_2 是凸函数，则 $f_1 + f_2$ 是凸函数.

与仿射函数的复合: 若 f 是凸函数，则 $f(Ax+b)$ 是凸函数.

Example 1.8

- 线性不等式的对数障碍函数

$$f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x), \quad \text{dom } f = \{x | a_i^T x < b_i, i = 1, ..., m\}$$

- 仿射函数的（任意）范数：$f(x) = \|Ax + b\|$

逐点取最大值

若 $f_1, ..., f_m$ 是凸函数，则 $f(x) = \max\{f_1(x), ..., f_m(x)\}$ 是凸函数

Example 1.9

- 分段线性函数：$f(x) = \max_{i=1,...,m}(a_i^T x + b_i)$ 是凸函数
- $x \in \mathbb{R}^n$ 的前 r 个最大分量之和：

$$f(x) = x_{[1]} + x_{[2]} + \cdots + x_{[r]}$$

是凸函数 *($x_{[i]}$ 为 x 的从大到小排列的第 i 个分量)*
事实上，$f(x)$ 可以写成如下多个线性函数取最大值的形式：

$$f(x) = \max\{x_{i_1} + x_{i_2} + \cdots + x_{i_r} | 1 \le i_1 < i_2 < \cdots < i_r \le n\}$$

与标量函数的复合 给定函数 $g : \mathbb{R}^n \to \mathbb{R}$ 和 $h : \mathbb{R} \to \mathbb{R}$,

$$f(x) = h(g(x))$$

若 $\begin{matrix} g \text{ 是凸函数}, h \text{ 是凸函数, 且单调不减} \\ g \text{ 是凹函数}, h \text{ 是凸函数, 且单调不增} \end{matrix}$，那么 f 是凸函数。

- 对 $n = 1$，g, h 均可微的情形，我们给出简证

$$f''(x) = h''(g(x))g'(x)^2 + h'(g(x))g''(x)$$

推论

- 如果 g 是凸函数，则 $\exp g(x)$ 是凸函数
- 如果 g 是正值凹函数，则 $1/g(x)$ 是凸函数

取下确界

若 $f(x,y)$ 关于 (x,y) 整体是凸函数，C 是凸集，则

$$g(x) = \inf_{y \in C} f(x,y)$$

是凸函数.

Example 1.10

- 考虑函数 $f(x,y) = x^{\mathrm{T}}Ax + 2x^{\mathrm{T}}By + y^{\mathrm{T}}Cy$，假设海瑟矩阵满足

$$\begin{bmatrix} A & B \\ B^{\mathrm{T}} & C \end{bmatrix} \succeq 0, \quad C \succ 0,$$

  则 $f(x,y)$ 为凸函数. 对 y 求最小值得

$$g(x) = \inf_{y} f(x,y) = x^{\mathrm{T}}(A - BC^{-1}B^{\mathrm{T}})x,$$

  因此 g 是凸函数. 进一步地，这与结论 A 的 *Schur* 补 $A - BC^{-1}B^{\mathrm{T}} \succeq 0$ 等价。

- 点 x 到凸集 S 的距离 $\mathrm{dist}(x,S) = \inf_{y \in S} \|x - y\|$ 是凸函数。

作业 1.1 证明*:*

- $f(x) = -(\prod_{k=1}^{n} x_k)^{\frac{1}{n}}$ *(for* $x \in \mathbb{R}^n_+$*)* 是凸函数. 即几何平均是凹函数。
- $f(x,y) = x^2/y$ 是定义域 $\{(x,y) \mid y > 0\}$ 上的凸函数。即二次函数的分式变换是凸函数。

MATH 3011 运筹学 **秋季 2024**

Lecture 2: 线性规划介绍

Lecturer: 陈士祥 *Scribes:* 陈士祥、*ChatGPT*

1 线性规划简介

线性规划（Linear Programming, LP）是数学优化的一个重要分支，旨在最大化或最小化线性目标函数，同时受到线性不等式（称为约束条件）的限制。它是由前苏联数学家 Leonid Kantorovich 和美国数学家 George Dantzig 在 20 世纪 40 年代独立发展的。

发展历史：

- George Dantzig

  Dantzig 是线性规划方法的创始人之一。他在 1947 年发明了单纯形方法，这是一种广泛使用的线性规划解决算法，至今仍是解决实际线性规划问题的主要工具之一。他获得了许多奖项，包括国家科学奖章（1975 年）和约翰 · 冯 · 诺依曼理论奖（1974 年）。运筹学和管理学协会（INFORMS）设立了 “George B. Dantzig 奖”，以表彰在运筹学和管理科学中做出重要贡献的研究者。

- Leonid Kantorovich

  Kantorovich 是苏联数学家和经济学家，他在 1939 年独立发展了线性规划的数学理论，并引入了现代优化理论中的对偶性概念。由于当时未被领导重视, 直到 1960 年康托洛维奇再次发表了《The Best Uses of Economic Resources》一书后, 才受到国内外的一致重视。他因在线性规划、最优化、经济学等方面的工作而获得 1975 年的诺贝尔经济学奖。

- John von Neumann

  von Neumann 是线性规划理论的早期开拓者之一。他与 Oskar Morgenstern 合著的《Theory of Games and Economic Behavior》(1944 年) 是博弈论的奠基作, 这对线性规划和经济学的发展产生了深远影响。该书中还提出了线性规划中的对偶理论的重要概念。对 Dantzig 发明单纯形法影响重大。

  奖项：尽管 von Neumann 生前获得了许多荣誉，但他最著名的可能是他对计算机科学和数学理论的贡献。INFORMS 颁发约翰 · 冯 · 诺依曼理论奖，表彰在运筹学或管理科学的基础领域做出杰出和持久贡献的个人。

- Tjalling Koopmans 贡献：Koopmans 是经济学家和运筹学家，他的工作主要集中在资源分配和最优化理论上。线性规划提出后很快受到经济学家的重视, 如在第二次大战中从事运输模型研究的

库普曼斯很快看到了线性规划在经济中应用的意义，并呼吁年轻的经济学家要关注线性规划。其中阿罗、萨谬尔逊、西蒙、多夫曼和胡尔威茨等都获得了诺贝尔奖金，并在运筹学某些领域中发挥过重要作用。他与 Kantorovich 共同获得了 1975 年的诺贝尔经济学奖，以表彰他们在资源最优分配理论方面的贡献。以他名字命名的奖：美国经济协会设有 “Tjalling C. Koopmans 经济理论奖”，表彰在经济理论方面的杰出论文。

- 后续发展：随后几十年中，线性规划领域经历了迅速发展，包括理论的深化、算法的改进，以及与其他数学领域的交叉融合，如对偶理论和内点法等。

应用场景：线性规划广泛应用于各个领域，包括但不限于：

- 经济学：资源分配、成本最小化、利润最大化等。
- 军事：在第二次世界大战中用于军事物资的分配和后勤计划。
- 生产与制造：产品混合选择、生产计划、物流等。
- 服务行业：人力资源规划、运输路线设计、网络设计等。
- 金融：投资组合优化、风险管理等。

2 线性规划

2.1 例: 线性规划在饮食健身与营养均衡中的应用

以饮食健身和营养均衡为例，线性规划可以帮助制定一个既经济又满足所有营养需求的饮食计划。

1. **定义决策变量**：
   假设我们有 N 种食物，决策变量 x_i 表示第 i 种食物的摄入量。
2. **建立目标函数**：
   最小化总成本。例如，每种食物的单位成本为 c_i，我们想最小化 $\sum_{i=1}^{N} c_i x_i$。
3. **建立约束条件**：
   - 营养约束：每种食物含有不同的营养成分，如蛋白质、碳水化合物、脂肪、维生素和矿物质等。如果每日营养需求是已知的，我们可以设定约束条件来保证摄入量满足最低营养需求。
   - 能量约束：设定总热量的上限，以保证不超过个人维持体重或减肥的热量需求。
   - 食物量约束：考虑到实际摄入量的可能性，每种食物的摄入量应在合理的范围内。

线性规划模型示例

假设我们有三种食物：鸡肉、大米和蔬菜。我们需要确保蛋白质、碳水化合物和脂肪的最低摄入量分别为 P、C 和 F。而每日的能量最大摄入量为 K。

决策变量：

- x_1：鸡肉克数
- x_2：大米克数
- x_3：蔬菜克数

目标函数：

$$\text{Minimize} \quad c_1x_1 + c_2x_2 + c_3x_3$$

约束条件：

$$\begin{aligned} p_1x_1 + p_2x_2 + p_3x_3 &\geq P \\ a_1x_1 + a_2x_2 + a_3x_3 &\geq C \\ f_1x_1 + f_2x_2 + f_3x_3 &\geq F \\ k_1x_1 + k_2x_2 + k_3x_3 &\leq K \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

其中，c_i, p_i, a_i, f_i, k_i 分别是每种食物的单位成本、单位蛋白质、单位碳水化合物、单位脂肪和单位热量。P、C、F 和 K 是蛋白质、碳水化合物、脂肪和总热量的日推荐摄入量。

线性规划的求解：通过构建这个模型，我们可以使用线性规划算法（如单纯形法或内点法）来求解决策变量的最优值，这些值将告诉我们为了最小化成本，同时满足所有营养和热量需求，每种食物需要摄入多少克。

2.2 线性规划图解法

对于非常简单的线性规划问题，我们可以通过图解的方法得到最优解。例如，考虑 $\mathbb{R}^2$ 中的问题：

$$\begin{aligned} \min \; & -x_1 - x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 3 \\ & 2x_1 + x_2 \leq 3 \\ & x_1 \geq 0, x_2 \geq 0. \end{aligned} \tag{2.1}$$

可以做出可行域如下图

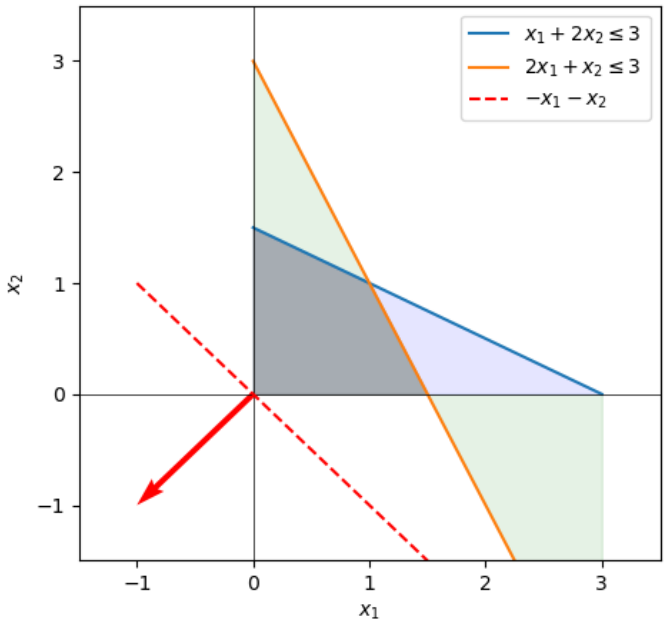


图 2.1: 问题(2.1)可行域图示，灰色区域即为可行域。红色箭头方向指向目标函数值的上升方向。故逆着红色箭头，移动红色虚线，达到可行域的某个顶点达到最小值。图中，$(1, 1)$ 即为最优点。

2.3 线性规划一般形式

我们针对实际问题，可以列出如下问题形式：

$$\begin{aligned} \min(\max) \quad & z = c_1 x_1 + \cdots + c_n x_n \\ \text{s.t.} \quad & a_{11} x_1 + \cdots + a_{1n} x_n \leq (=, \geq) b_1 \\ & \vdots \\ & a_{m1} x_1 + \cdots + a_{mn} x_n \leq (=, \geq) b_m \end{aligned} \tag{2.2}$$

我们可以把(2.2)转化为如下一般形式：

$$\begin{aligned} \min \quad & c_1 x_1 + \cdots + c_n x_n \\ \text{s.t.} \quad & a_{11} x_1 + \cdots + a_{1n} x_n \geq b_1 \\ & \vdots \\ & a_{m1} x_1 + \cdots + a_{mn} x_n \geq b_m \end{aligned} \tag{2.3}$$

若优化目标为最大化 (Maximize) 函数值，则可以在目标函数添加负号得到等价的最小化 (Minimize) 的问题。若约束中既有 $\leq$ 也有 $\geq$，也可以通过改变符号的方式统一为 $\geq$ 的约束。

故，我们有线性规划矩阵一般形式为：

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b. \end{aligned} \tag{2.4}$$

并且，线性规划的约束都是凸多面体。

2.4 线性规划的标准形式

线性规划的标准形式是理解和求解线性规划问题的一个基础和重要步骤。它为理论分析和算法实现提供了一个清晰、统一的框架。当我们研究线性规划理论时，一般形式更为方便。而在计算时，使用标准形式更为方便。

线性规划问题总可以写成如下标准形式：

$$\text{(LP)} \qquad \begin{aligned} \min \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j = b_i, i = 1, \cdots, m \\ & x_j \geq 0, j = 1, \cdots, n. \end{aligned} \tag{2.5}$$

或者用矩阵表示为：

$$\text{(LP)} \qquad \begin{aligned} \min \quad & \mathbf{c}^\top x \\ \text{s.t.} \quad & Ax = \mathbf{b} \\ & x \geq 0. \end{aligned} \tag{2.6}$$

其中矩阵 $A \in \mathbb{R}^{m \times n}$, $\mathbf{c}$ 是 n 维列向量，$\mathbf{b}$ 是 m 维列向量。$x \geq 0$ 表示所有分量 $x_i \geq 0$.

对于非标准形式，可能的标准化步骤有：

- 目标函数 $\max f(x) \longrightarrow \min -f(x)$
- 不等式约束的等式化（引入松弛变量或者剩余变量）
- 自由变量的非负化 $x_j = x_j^{'} - x_j^{''}, x_j^{'}, x_j^{''} \geq 0$

Proposition 2.1 线性规划一般形式和标准形式等价。*(两个优化问题等价的意义是指，对于一个优化问题的可行解，我们总可以找到另一个问题对应的可行解，使得它们的目标函数值相等。也有条件更弱的阐述：两个优化问题等价，如果给定一个问题的最优解，可以构造出另一个问题的最优解，并且最优值相同。)*

Proof: 对于一般形式(2.4)，

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b \end{aligned}$$

我们引入松弛变量 $s \in \mathbb{R}^m$, $s \geq 0$, 使得有等式 $Ax - s = b$ 成立. 另外, 可以将 x 记为 $x' - x''$, $x' \geq 0, x'' \geq 0$. 这是因为任意的实数总可以写成两个非负实数的差。

故问题(2.4)可以写为

$$\begin{aligned} \min \quad & c^\top x' - c^\top x'' + 0_m^\top s \\ \text{s.t.} \quad & Ax' - Ax'' - s = b \\ & x' \geq 0, x'' \geq 0, s \geq 0. \end{aligned}$$

这里 0_m 表示 m 维零向量。令 $\bar{c}^\top = (c^\top, -c^\top, 0_m^\top)$, $\bar{A} = [A, -A, -I_m]$, 这里 I_m 表示 $m \times m$ 的单位矩阵, $\bar{x}^\top = ((x')^\top, (x'')^\top, s^\top)$, 我们有

$$\begin{aligned} \min \quad & \bar{c}^\top \bar{x} \\ \text{s.t.} \quad & \bar{A}\bar{x} = b, \\ & \bar{x} \geq 0. \end{aligned}$$

此即为标准形式(2.6). 故, 一般形式 LP 总有标准形式 LP 与其对应。

反之, 对于标准形式, 我们可以将 $Ax = b$ 写为 $Ax \leq b$ 和 $Ax \geq b$. 这样也可以得到一般形式的 LP。■

作业 2.1 证明: 对于标准形式, 如果矩阵 $A \in \mathbb{R}^{m \times n}$ 的秩为 $k, k < m$, 其行向量为 $a_1^\top, a_2^\top, \ldots, a_m^\top$. 那么 A 的 k 个线性无关的行向量 $a_{i_1}^\top, a_{i_2}^\top, \ldots, a_{i_k}^\top$ 组成的子矩阵 $\tilde{A}$ 和对应的 $\tilde{b}$, 有 $P = Q$, 这里 $P = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$, $Q = \{x \mid \tilde{A}x = \tilde{b}, x \geq 0\}$.

因此, 不失一般性, 我们考虑标准形式可以假设 A 是行满秩的。

MATH 3011 运筹学 **秋季 2024**

Lecture 3: 线性规划基本理论

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 线性规划基本理论

对于线性规划基本理论，我们考虑一般形式，记可行域为 $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$.

结论 1：在线性规划中，约束条件均为线性等式及线性不等式，所以可行域 P 是凸集。

由线性规划图解中的例子，其最优解在某个顶点取得。一个自然的问题是，在一般情况下，线性规划的最优点是否仍是 P 的某个顶点?

这启发我们去研究如下几个定义。

1.1 顶点、极点、基解和可行基解

我们记 $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ 为一个凸多面集.
我们这里假设 P 中有一些为等式约束，记这些等式约束下标集为 $\mathcal{E}$, 即，$a_i^\top x = b_i, i \in \mathcal{E}$.
剩余的为不等式约束，下标集记为 $\mathcal{I}$, 即，$a_i^\top x \geq b_i, i \in \mathcal{I}$.

- **极点 (extreme point)：** $\hat{x} \in P$ 被称为 P 的极点，如果 $\hat{x}$ 不能被表示为其它两个可行点的凸组合。如图 Figure 3.1所示。

  具体地，$\hat{x} \in P$ 被称为极点, 如果我们找不到不同于 $\hat{x}$ 的两点 $x^{(1)}, x^{(2)} \in P$ 和 $\lambda \in [0,1]$, 使得 $\hat{x} = \lambda x^{(1)} + (1-\lambda)x^{(2)}$。

- **顶点 (vertex)：** $\hat{x}$ 被称作 P 的顶点，如果存在某个 $c \in \mathbb{R}^n$，使得 $c^\top \hat{x} < c^\top y$，$\forall y \in P, y \neq \hat{x}$.

  换言之，$\hat{x}$ 是 P 的顶点，当且仅当 P 在超平面 $\{y \mid c^\top y = c^\top \hat{x}\}$ 的一侧，并且 P 恰好与该超平面相交于 $\hat{x}$. 如图 Figure 3.2所示。

- **基解 (basic solution) 和可行基解 (basic feasible solution):**

  Definition 3.1 考虑约束 $P = \{x \mid Ax \geq b\} = \{x \mid a_i^T \geq b_i, i = 1, 2, \ldots, m\}$，我们称约束 $a_i^T x \geq b_i$ 对于点 $\bar{x}$ 是积极约束 (*active constraint*)，如果 $a_i^\top \bar{x} = b_i$。积极约束的下标集记为 $\mathcal{A} = \{i : a_i^\top \bar{x} = b_i, i = 1, 2, \ldots, m\}$，我们称 $\mathcal{A}$ 为积极集。

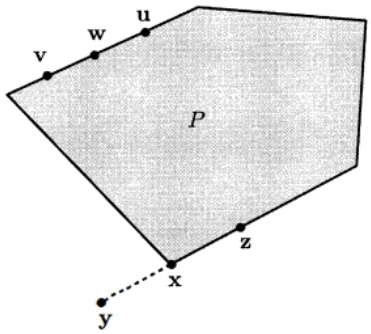


Figure 2.4: The vector **w** is not an extreme point because it is a convex combination of **v** and **u**. The vector **x** is an extreme point: if $\mathbf{x} = \lambda\mathbf{y} + (1-\lambda)\mathbf{z}$ and $\lambda \in [0,1]$, then either $\mathbf{y} \notin P$, or $\mathbf{z} \notin P$, or $\mathbf{x} = \mathbf{y}$, or $\mathbf{x} = \mathbf{z}$.

图 3.1: 图示来源《Introduction to Linear Optimization》：极点。

P 中的等式约束自然是积极约束；不等式约束在某些点取到等式时，成为积极约束。我们有，

$$\mathcal{A} = \mathcal{E} \cup \{i \in \mathcal{I} \mid a_i^\top \bar{x} = b_i\}.$$

Definition 3.2 (基解) *$\hat{x}$ 被称为一个基解，如果 (a). 所有 P 中的等式约束 (若有) 都成立；(b). 所有积极集 $\mathcal{A} = \{i \mid a_i^\top \hat{x} = b_i, i = 1, 2, ..., m\}$ 中，存在 n 个下标 i, 使得 a_i 线性无关。*

我们知道，n 个线性无关的等式确定唯一的点。故，上述基解体现了“唯一性”。这与顶点和极点的性质类似。然而，上述定义并未要求基解满足不等式约束，为了得到可行的基解，我们有如下定义。

Definition 3.3 (可行基解) *一个基解如果也是可行解，我们称其为一个* **可行基解** *(BFS)。*

注：根据以上定义，极点和顶点是几何层面的定义（有很强的几何直观），基解是代数层面的定义（由线性相关定义）。

Theorem 3.1 *如果 $P = \{x \mid Ax \geq b\}$ 是一个非空多面集，$x \in P$, 那么下述三种情况等价*

1. *x 是顶点；*
2. *x 是极点；*
3. *x 是可行基解。*

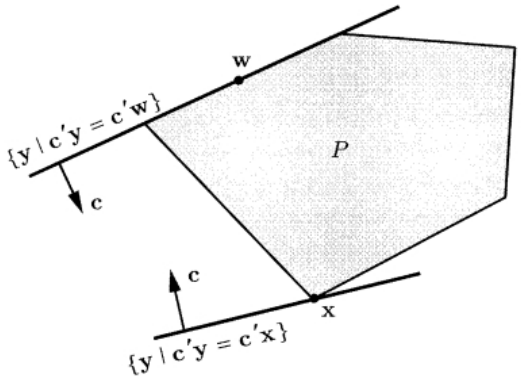


Figure 2.5: The line at the bottom touches P at a single point and $\mathbf{x}$ is a vertex. On the other hand, $\mathbf{w}$ is not a vertex because there is no hyperplane that meets P only at $\mathbf{w}$.

图 3.2: 图示来源《Introduction to Linear Optimization》: 顶点。

Proof: 顶点 → 极点:
给定 x 是一个顶点，根据定义，可以找到 c 使得 $c^\top x < c^\top y,\ y \in P, y \neq x$. 假设 x 不是极点，那么存在不同于 x 的点 $y, z \in P$, 使得 $x = ty + (1-t)z, t \in [0,1]$, 但是 $c^\top x < c^\top(ty + (1-t)z)$，得到矛盾。因此 x 不能被表示为其余两个可行点的凸组合。所以 x 是一个极点。

极点 → 可行基解:
我们证明：对于一个可行点 $x \in P$, 如果 x 不是可行基解，那么 x 也不是极点。
如果 $x \in P$ 非可行基解，记 $I = \{i : a_i^\top x = b_i\}$ 且 $a_i, i \in I$ 都线性无关, 那么 $|I| < n$. 所以 $a_i, i \in I$ 在 $\mathbb{R}^n$ 的一个严格子空间中。可以找到非零方向 $d \in \mathbb{R}^n$, 使得 $d^\top a_i = 0, i \in I$. 我们令 $y = x + \epsilon d, z = x - \epsilon d$, $\epsilon > 0$ 充分小。我们有 $a_i^\top y = b_i = a_i^\top z, i \in I$. 对于 $i \notin I$, 可以令 ϵ 充分小，使得 $a_i^\top y = a_i^\top x + \epsilon a_i^\top d > b_i$ 且 $a_i^\top z = a_i^\top x - \epsilon a_i^\top d > b_i$. 故 $y, z \in P$, $x = (y+z)/2$ 不是极点。 ■

作业 3.1 证明**可行基解 → 顶点:**
提示：构造 $c = \sum_{i \in I} a_i$. I 是积极集。

1.2 线性规划标准形式的可行基解

我们下面考虑线性规划标准形式的可行基解。

$$\text{(LP)} \qquad \begin{aligned} \min \quad & \mathbf{c}^\top x \\ \text{s.t.} \quad & Ax = \mathbf{b} \\ & x \geq 0. \end{aligned} \tag{3.1}$$

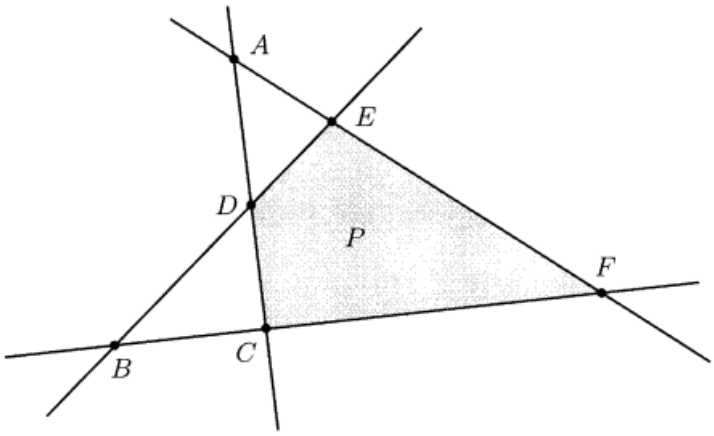


图 3.3: $\mathbb{R}^2$ 中，区域 P 是由四个半平面 $\{x : a_i^\top x \le b_i\}, i = 1, 2, 3, 4$ 围成的多面集，A, B, C, D, E, F 均为基解，C, D, E, F 为可行基解。

其中矩阵 $A \in \mathbb{R}^{m\times n}$, $\mathbf{c}$ 是 n 维列向量，$\mathbf{b}$ 是 m 维列向量。$x \ge 0$ 表示所有分量 $x_i \ge 0$.

Theorem 3.2 *考虑约束 $Ax = b$ 和 $x \ge 0$，并假设 $m \times n$ 矩阵 A 的所有行向量是线性无关的。向量 $x \in \mathbb{R}^n$ 是基解当且仅当我们有 $Ax = b$，并且存在下标 $B(1), \ldots, B(m)$ 使得:*

1. *矩阵 A 中的列向量 $A_{B(1)}, \ldots, A_{B(m)}$ 是线性无关的;*
2. *如果 $i \ne B(1), \ldots, B(m)$，那么 $x_i = 0$。*

不失一般性，假设 $A = (B, N)$, 其中 B 是 m 阶可逆矩阵。同时记 $x = (x_B^\top, x_N^\top)^\top$, 其中 x_B 的分量与 B 中的列对应，x_N 的分量与 N 中的列对应。这样 $Ax = \mathbf{b}$ 即可写成

$$(B, N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = \mathbf{b},$$

即 $Bx_B + Nx_N = \mathbf{b} \Longrightarrow x_B = B^{-1}\mathbf{b} - B^{-1}Nx_N$.

基解/基矩阵:

在上式中，x_N 的分量就是线性方程组 $Ax = \mathbf{b}$ 的自由变量。特别地令 $x_N = \mathbf{0}$, 则得到解

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{pmatrix}$$

为方程组的一个基解，对应的 B 称为基矩阵。

x_B 的各分量称为基变量，x_N 的各分量称为非基变量。若 $B^{-1}\mathbf{b} \geq \mathbf{0}$, 则 $x = \begin{pmatrix} B^{-1}\mathbf{b} \\ \mathbf{0} \end{pmatrix}$ 为 (LP) 的可行基解，相应的称 B 为可行基矩阵，$x_B = \begin{pmatrix} x_{B_1} \\ \vdots \\ x_{B_m} \end{pmatrix}$ 为一组可行基变量。

作业 3.2 给出下面线性规划问题的极点和基解：

$$\begin{aligned} \min \quad & -x_1 + 3x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 8 \\ & x_2 \leq 2 \\ & x_1, x_2 \geq 0. \end{aligned} \tag{3.2}$$

1.3 可行基解的存在性和最优性

Definition 3.4 *对于一个多面集 $P = \{x \mid Ax \geq b\} \subset \mathbb{R}^n$, 如果存在 $x \in P$ 和一个非零向量 $d \in \mathbb{R}^n$, 使得对任意实数 λ, 有 $x + \lambda d \in P$，那么称 P 包含一条直线.*

Theorem 3.3 (极点存在性定理) *假设 $P = \{x \mid Ax \geq b\} \subset \mathbb{R}^n$ 非空，下列 2 种情况等价：*

1. *P 中存在至少一个极点。*
2. *P 不包含直线。*

证明参考《Introduction to Linear optimization, Theorem 2.6》.

推论：对于非空有界的多面集，或者非空的标准形式多面集, 它们不包含直线，故必有可行基解。

Theorem 3.4 (可行基解最优性) *考虑线性规划问题，在多面集 $P = \{x \mid Ax \geq b\}$ 上，最小化 $c^\top x$。**假设 P 中存在至少一个极点**，并且存在最优解，要么必定有某个极点是最优解。*

Theorem 3.5 (可行基解最优性) *考虑线性规划问题，在多面集 $P = \{x \mid Ax \geq b\}$ 上，最小化 $c^\top x$。**假设 P 中存在至少一个极点**。那么，要么最优值是 $-\infty$，要么必定有某个极点是最优解。*

Theorem 3.5 和 3.5证明分别参考《Introduction to Linear optimization, Theorem 2.7 & 2.8》。值得注意的是，上面两个定义的假设有所区别，Theorem 3.4假设存在最优解，而定理 3.5分为最优值有限和无限两种情况。结合这两个定理，我们有如下推论。

推论：若线性规划问题可行域非空，则要么最优值为 $-\infty$，要么存在最优解。

该推论对于一般非线性规划问题不成立，例如：$\inf \frac{1}{x}$， s.t. $x \geq 0$. 显然最优值有界，但是最优解不存在。

由于标准形式至少存在一个极点，故有如下结论。

Theorem 3.6 (线性规划标准形式最优性结论) *考虑线性规划问题，假设标准形式多面集 $P = \{x \mid Ax = b, x \geq 0\}$ 非空。那么，要么最优值是 $-\infty$，要么必定有某个极点是最优解。*

由上述定理，我们有如下 **(标准形式) 最优解的分类约定:**

可行域为空 $\Longrightarrow$ 无解

可行域有界 $\Longrightarrow$ 唯一解 或者无穷多解

可行域无界 $\Longrightarrow$ (1) 唯一解 或者 (2) 无穷多解 或者 (3) 最优值为 $-\infty$

我们把“唯一解”和“无穷多解”称为模型存在最优解，

我们把“唯一解”和“无穷多解”称为模型存在最优解，而把“无界解 $-\infty$”归入不存在最优解的情形。

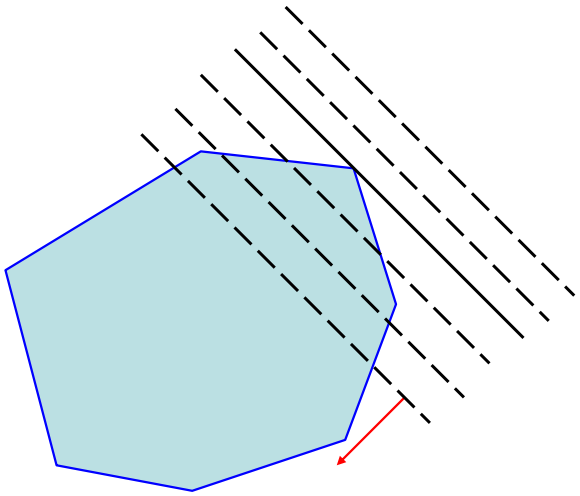


图 3.4: 线性规划的直观图解–有界区域唯一解：有界可行域，不包含直线，故存在可行基解，某个极点为最优解

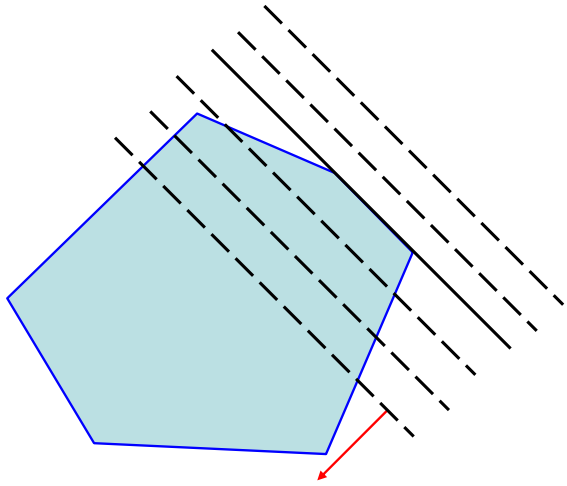
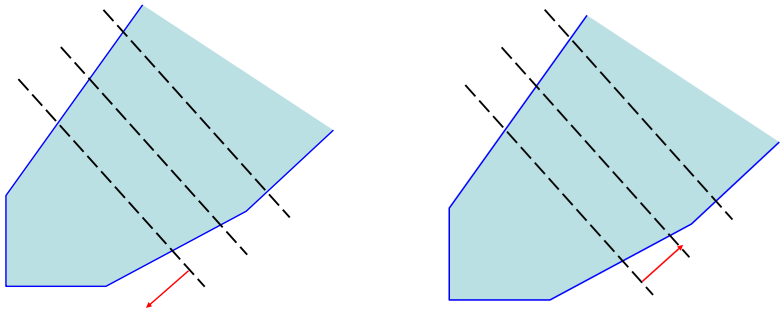


图 3.5: 线性规划的直观图解–有界区域不唯一解：有界可行域，不包含直线，故存在可行基解，某个极点为最优解，图中，最优解为边界线段上的所有点。



(a) 最优值 $-\infty$ (b) 最优解唯一

图 3.6: 无界可行域，最优解可能为 $-\infty$，也可能唯一存在。

2 总结

当线性规划标准形式存在最优解时，目标函数的最优值一定能在可行域的某个极点处达到，即 (LP) 存在最优解时，则一定存在一个可行基解是最优解。

这样，线性规划模型的求解（最优解）归结为求最优可行基解。这一思想正是单纯形方法的基本出发点。但可行基解的个数往往很多（上界为 $\frac{n!}{m!(n-m)!}$），不宜一一枚举。该采取何种策略？而这正是单纯形算法的最关键的步骤.

MATH 3011 运筹学 **秋季 2024**

Lecture 4: 单纯形法

Lecturer: 陈士祥 *Scribes:* 陈士祥、*ChatGPT*

1 单纯形方法简介

单纯形法（Simplex Method）是解决线性规划问题的一种算法，由美国数学家乔治 · 丹齐格（George Dantzig）于 1947 年发明。它在数学优化和运筹学领域中占有极其重要的地位。以下是单纯形法发展历史的简述：

二战期间，丹齐格在美国空军工作，负责计划和决策问题。他发现现有的线性规划方法无法高效解决这些问题。1947 年，丹齐格发明了单纯形法，这是第一个实用且高效的线性规划求解算法。该方法使用了几何上的单纯形概念，通过从一个顶点到另一个顶点的迭代过程，寻找最优解。

初期应用：单纯形法很快被用于军事和政府项目中，尤其是在军事物资分配和经济计划方面。

计算机时代：随着计算机的发展，单纯形法的计算变得更加快速和可行。在 20 世纪 50 年代和 60 年代，它成为了计算机上执行最频繁的算法之一。

单纯形法是线性规划领域的一个里程碑，对 20 世纪的科学和工程产生了深远的影响。尽管后续出现了一些新的算法，单纯形法仍然是解决实际线性规划问题的一种重要和常用方法。

2 单纯形方法

由于标准形式 LP 的最优解若存在，则必在某个顶点取到。单纯形的**基本思想** 如下：

从一个可行基解出发，求下一个使目标函数值有所改善的可行基解；通过不断迭代改进可行基解力图达到最优可行基解。

单纯形法的**主要步骤：**

- 最优判定 (optimality)
- 转轴运算 (pivoting)

2.1 最优判定

首先假定对线性规划标准形问题

$$\{\min c^\top x \mid Ax = b, x \geq 0\}$$

已得到一个可行基的划分 $A = (\mathbf{B}, N)$, $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, $c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$.

根据 Lecture 2, Theorem 3.2, 可行基解有如下形式：

$$x = \begin{pmatrix} x_B \\ 0 \end{pmatrix}, \quad x_B \geq 0.$$

从可行基解 $x = \begin{pmatrix} x_B \\ 0 \end{pmatrix}$ 开始，考虑沿着方向 $d = \begin{pmatrix} d_B \\ d_N \end{pmatrix}$ 移动到 $x + \theta d, \theta > 0$。d_B 被称为**基方向**，d_N 被称为**非基方向**。

这里的想法是，在非基方向中选取新的可行基解，使得目标函数值变小。不妨设，

$$\begin{cases} d_j = 1, & j \in N \\ d_i = 0, & i \in N, i \neq j, \end{cases}$$

d_B 待确定. 沿着 d 移动的步长为 θ：对于非基变量，

$$\tilde{x}_j = x_j + \theta, \theta > 0.$$

$$\tilde{x}_i = 0, i \neq j, i \in N.$$

移动后的点，首先需要保证 $A(x + \theta d) = b$。

那么由 $A(x + \theta d) = b$, $Ax = b$，我们有

$$0 = Ad = \mathbf{B}d_B + A_j,$$

所以

$$d_B = -\mathbf{B}^{-1} A_j.$$

A_j 表示矩阵 A 的第 j 列向量。在点 $x + \theta d$ 的目标函数值可表示为

$$\begin{aligned} z &= c_B^\top (x_B + \theta d_B) + \theta c_N^T d_N \\ &= c_B^\top (x_B - \theta \mathbf{B}^{-1} A_j) + \theta c_j \end{aligned} \tag{4.1}$$

所以，目标值相对于原来 $c^\top x$ 变化量为

$$z - c^\top x = \theta(c_j - c_B^\top \mathbf{B}^{-1} A_j).$$

我们将

$$\bar{c}_j = c_j - c_B^\top \mathbf{B}^{-1} A_j \tag{4.2}$$

称为非基方向 x_j 的减少量。

Theorem 4.1 (最优判定条件) 标准形式线性规划问题中，某个可行基解为 $x = \begin{pmatrix} \mathbf{B}^{-1}b \\ 0 \end{pmatrix}$，若由式(4.2)所决定的 $\bar{c}$ 满足

$$\bar{c}_j \geq 0, \forall j \in N, \tag{4.3}$$

那么 x 是一个最优解.

Proof: 记任意可行解 y（满足 $y_j \geq 0, j \in N$），令 $d = y - x$. 那么 $Ad = 0$, 可知 d 满足

$$d_B = -\mathbf{B}^{-1} N d_N.$$

进一步，计算

$$c^\top y - c^\top x = c^\top d = \sum_{j \in N} (c_j - c_B^\top \mathbf{B}^{-1} A_j) d_j = \sum_{j \in N} \bar{c}_j d_j.$$

由于 y 可行，故 $d_j \geq 0, j \in N$. 因此，$c^\top x$ 为最小值。 ■

2.2 转轴运算

上述变换过程 $x + \theta d$，并未考虑 $x + \theta d$ 的关于约束 $x \geq 0$ 的可行性。也就是说，这里需要考虑是否存在合适的 $\theta > 0$ 使得 $x + \theta d \geq 0$.

Definition 4.1 (可行方向) 令 $x \in P$ 是一个可行点，我们称 d 是 x 的可行方向，如果存在正数 $\theta > 0$，使得 $x + \theta d \in P$.

Definition 4.2 (退化基解) 基解 x 称为退化基解，如果 x 有大于 n 个积极约束。

对于标准形式的多面集，由于假设 A 的行向量线性无关，基解 x 至少有 $n - m$ 个坐标为 0。如果 x 有大于 n 个积极约束，那么有大于 $n - m$ 个积极约束是 $x_i = 0$, 也就是有大于 $n - m$ 个分量是 0。反之，如果 x 是一个基解，并且 x 有大于 $n - m$ 个分量是 0，那么 x 处有大于 n 个积极约束（因为等式约束有 m 个)。所以，对于标准形式，我们有如下定义：

Definition 4.3 (退化基解) 对于标准形式的多面集，基解 x 称为退化基解，如果 x 有大于 $n - m$ 个分量是 *0*。

假设 x 是一个可行基解，对 x 做上述移动：$x + \theta d$:

1. 如果 x 非退化，那么 $x_B > 0$, 存在充分小 $\theta > 0$，使得 $x_B + \theta d_B \geq 0$ 成立；

2. 如果 x 退化，因为有某个 $x_i = 0, i \in B$, 故方向 d 可能非可行方向。

Example 4.1 *考虑* $P = \{x = (x_1, \ldots, x_7) | Ax = b, x \geq 0\}$，*其中*

$$A = \begin{pmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 8 \\ 12 \\ 4 \\ 6 \end{pmatrix},$$

考虑由线性独立列 A_1, A_2, A_3, A_7 *组成的基。为了计算相应的基解，我们首先将非基变量* $x_4, x_5,$ *和* x_6 *设为零，然后解系统* $Ax = b$ *以获得其余变量，得到* $x = (4, 0, 2, 0, 0, 0, 6)$。*这是一个退化的基可行解，因为我们有四个变量为零，而* $n - m = 3$. *我们也可以选线性独立列* A_1, A_3, A_4, A_7 *组成基，同样得到* $x = (4, 0, 2, 0, 0, 0, 6)$.

我们令 $\mathbf{B} = (A_1, A_2, A_3, A_7)$. *对于非基变量* x_4, $d_B = -\mathbf{B}^{-1}A_4 = (0, -1.5, 0.25, 1.5)$, *所以* $d = (0, -1.5, 0.25, 1, 0, 0, 1.5)$, *不是可行方向。*
对于非基变量 x_5, *我们有* $d_B = -\mathbf{B}^{-1}A_5 = (0, 0.5, -0.25, -0.5)$, $d = (0, 0.5, -0.25, 0, 1, 0, -0.5)$。*只要* $\theta \leq 8$, $x + \theta d$ *可行。目标值的减少量为* $c_5 - c_B^\top \mathbf{B}^{-1}A_5 = c_5 + c_B^\top d_B$。*最后，* $x + 8d = (4, 4, 0, 0, 8, 0, 2)$ *也是一个可行基解。*

由上例可知，对于非退化可行基解，我们可以适当的选择非基变量 x_j 和 $\theta > 0$, 到达新的可行基解。单纯形法中的转轴运算便是这种想法。

我们先假设任意可行基解都非退化，后面我们会看到，对于退化解，有简单的手段处理。

回顾之前的做法，为了保证解的可行性，有必要进一步分析 θ 的取值及基变量 x_B 的变化情况。记 $\tilde{x} = x + \theta d$. 若取 $x_j = \theta$, $x_i = 0 (i, j \in N, i \neq j)$, 则由线性方程组 $Ax = b$ 得 $\tilde{x}_B$ 的值为

$$\tilde{x}_B = x_B - \theta \mathbf{B}^{-1} A_j.$$

记 $d_B = -\mathbf{B}^{-1}A_j$, 上式可写成

$$\begin{pmatrix} \tilde{x}_{B_1} \\ \tilde{x}_{B_2} \\ \vdots \\ \tilde{x}_{B_m} \end{pmatrix} = \begin{pmatrix} x_{B_1} \\ x_{B_2} \\ \vdots \\ x_{B_m} \end{pmatrix} + \theta \begin{pmatrix} d_{B_1} \\ d_{B_2} \\ \vdots \\ d_{B_m} \end{pmatrix}. \tag{4.4}$$

1. 如果 $d_B \geq 0$, $\theta = +\infty$. 这时若 $\bar{c}_j < 0$，线性规划的最优值为 $-\infty$

2. 如果存在 $i \in \{1, 2, \ldots, m\}$, $d_{B_i} < 0$. θ 必须满足 $\theta \leq -x_{B_i}/d_{B_i}$. 我们取最大可能的 θ, 记作

$$\theta^* = \min_{\{i|d_{B_i}<0\}} (-\frac{x_{B_i}}{d_{B_i}}). \tag{4.5}$$

得到新的点

$$\tilde{x} = x + \theta^* d$$

记 r 是取到(4.5)中最小值的下标 (可能在多个下标取到最小值，可以任取一个下标. 这种情况下，新得到的点是退化的。)，那么 $\tilde{x}_{B_r} = 0$.

注意到，$\tilde{x}_j = \theta^* > 0$, $\tilde{x}_i = 0, i \in N, i \neq j$.

至此，x_{B_r} 和 $x_i (i \in N, i \neq j)$ 合起来不少于 $n-m$ 个变量取值为 0; 取值为正的只能是 $\{x_{B_1}, \cdots, x_{B_{r-1}}, x_j, x_{B_{r+1}}, \cdots$ 把这 m 个变量看成新的基变量，另外 $n-m$ 个看成非基变量，就得到了新的一组基解和非基解。

简单来说，我们将 A_{B_r} 从 $\mathbf{B}$ 中踢出，让 A_j 进入，得到新的基矩阵 $\tilde{\mathbf{B}}$.

于是，我们获得了新的可行解

$$\tilde{x} = (x_{B_1}, \cdots, x_{B_{r-1}}, 0, x_{B_{r+1}}, \cdots, x_{B_m}, 0, \cdots, x_j, \cdots, 0)^\top.$$

Theorem 4.2 *上述新得到的 $\tilde{x}$ 是可行基解，即 $\tilde{\mathbf{B}} = (A_{B_1}, \cdots, A_{B_{r-1}}, A_j, A_{B_{r+1}}, \cdots, A_{B_m})$ 的秩为 m.*

Proof: 因为 $B^{-1}A_j = -d_B$, d_B 的第 r 个分量非零。并且 $B^{-1}A_{B_i}, i \neq r$ 是单位矩阵 I_m 的第 i 列。所以他们线性无关。故 $A_{B_1}, \cdots, A_{B_{r-1}}, A_j, A_{B_{r+1}}, \cdots, A_{B_m}$ 线性无关。 ■

以上操作是通过替换基变量 x_{B_r} 与非基变量 x_j 来确定新的可行基解，我们称之为**转轴运算** (pivoting).

另外，称这时的 x_{B_r} 为出基变量，x_j 为入基变量。这样得到的新可行基解，利用式(4.3)进行最优判定，若未达到最优解，则重复上述操作。

单纯形算法 (SIMPLEX)

1. 第一步（初始化）:

   确定一个可行基划分 $A = (B, N)$, 并计算 $x_B = \bar{b} := B^{-1}b$.

2. 第二步（最优判定）:

   计算向量 $w = (B^\top)^{-1}c_B$, 对所有的非基分量 $j \in N$ 求出 $z_j = w^\top A_j$.

   IF $\bar{c}_j = c_j - z_j \geq 0 (j \in N)$, 则当前的可行基解 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} \bar{b} \\ 0 \end{pmatrix}$ 已是最优解 [**stop!**]。

   ELSE 选取一个满足 $c_j - z_j < 0$ 的 $j \in N$ 进入下一步。(**注：** 选取 j 的方式有多种。如：下标从

小到大计算 $\bar{c}_j$，遇到小于 0 时直接选取。或者，计算出所有的情况，取 $\theta_j^*\bar{c}_j$ 中最小的那个。但是，后者在实际中并未提升求解效率。）

3. 第三步（转轴运算）：

   计算向量 $d = -B^{-1}A_j$.

   IF $d \geq 0$, 则问题无有界解 [**stop!!**]。

   ELSE

   找出 r 使得 $\theta^* = \min_{\{i|d_{B_i}<0\}}(-\frac{x_{B_i}}{d_{B_i}})$, 将基矩阵 B 的列向量 A_{B_r} 用 A_j 替换，得到新的基矩阵

   $$B = (A_{B_1}, \cdots, A_{B_{r-1}}, A_j, A_{B_{r+1}}, \cdots, A_{B_m}).$$

   进而，记新基变量的值为

   $$x_{B_i} = x_{B_i} + \theta^* d_i (i = 1, \cdots, m, i \neq r), x_j = \theta^*.$$

   令基变量的下标集合与非基变量的下标集合分别为

   $$B := (B \cup \{j\}) - \{B_r\}, \quad N := (N \cup \{B_r\}) - \{j\}.$$

   回到第二步。

2.3 单纯形法的收敛结论

如前所述，经过转轴运算，对应于 $\bar{c}_j < 0$ 的非基变量 x_j 入基得到新的可行基解时，目标函数值将减少 $\theta^*|\bar{c}_j|$.

对于非退化情况，每次转轴后，目标值严格减小，所以每个可行基解不会被重复到达 2 次及以上。另外，可行基解的个数是有限的，迭代不会无限重复下去，必在有限次迭代后结束计算。

Theorem 4.3（单纯形法的有限收敛性定理（非退化情形）） *若所给的线性规划问题可以求出初始可行基解，而且转轴运算过程中的所有可行基解都是非退化的，则利用单纯形法在有限次迭代后，要么找到最优解，要么识别出问题无有界解，从而结束计算。*

3 单纯形表

在上述算法中，如果我们每一步直接计算新的基矩阵的逆矩阵 $\mathbf{B}^{-1}$, 浮点数乘法复杂度为 $\mathcal{O}(m^3)$. 其余计算为矩阵向量乘法，例如 $\mathbf{B}^{-1}A_j$, $C_B^\top\mathbf{B}^{-1}$，复杂度为 $\mathcal{O}(mn + m^2)$. 故总体复杂度为 $\mathcal{O}(mn + m^3)$.

实际上，如果上一步的基矩阵的逆 $\mathbf{B}^{-1}$ 已经存在，那么计算新的基矩阵 $\tilde{\mathbf{B}}$ 的逆，仅需要 $\mathcal{O}(m^2)$. 这是因为，

$$\mathbf{B}^{-1}\tilde{\mathbf{B}} = [e_1, e_2, \ldots, e_{r-1}, u_r, e_{r+1}, \ldots, e_m],$$

这里的 e_i 是 $m \times m$ 单位矩阵 I_m 的第 i 列，$u_r = -d_B = \mathbf{B}^{-1}A_j$.

因为 $\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{B}} = I_m$, 我们只需要对 $\mathbf{B}^{-1}\tilde{\mathbf{B}}$ 作行变换，使得第 r 列变为 e_r 即可。这需要先将 $\mathbf{B}^{-1}\tilde{\mathbf{B}}$ 的第 r 行乘以 $-\frac{u_{ri}}{u_{rr}}$ 加到第 i 行，$i \neq r, i = 1, ..., m$. 最后让第 r 行除以 u_{rr}. 我们可以将更新 $\mathbf{B}^{-1}\tilde{\mathbf{B}}$ 写入下面的单纯形表格中，该表格共有 $m+1$ 行和 $n+1$ 列：

$-c_B^\top \mathbf{B}^{-1} b$	$c^\top - c_B^\top \mathbf{B}^{-1} A$
$\mathbf{B}^{-1} b$	$\mathbf{B}^{-1} A$

或者 （最上面这行叫做第 0 行）

$-c_B^\top x_B$	$\bar{c}_1$	$\ldots$	$\bar{c}_n$
$x_{B(1)}$	$\vert$		$\vert$
$\vdots$	$\mathbf{B}^{-1}A_1$	$\ldots$	$\mathbf{B}^{-1}A_n$
$x_{B(m)}$	$\vert$		$\vert$

单纯形表更新第 1 到 m 行的过程，就是更新逆矩阵 $\mathbf{B}^{-1}$ 的过程，或者说是对单纯形表作行变换。

- 注意，若 i 是可行基解的某个下标，那么它对应的最优判定值 $\bar{c}_i = c_i - c_B^\top \mathbf{B}^{-1} A_i = c_i - c_i = 0$. 所以，对于第 0 行，若 j 是新的入基坐标，我们用第 r 行乘以某个数加到第 0 行，使得新的表中 $\bar{c}_j = 0$.
  注：我们下面说明这样做是正确的。第 0 行可以写为：

  $$(0, c^\top) - g^\top (b, A),$$

  这里 $g^\top = c_B^\top \mathbf{B}^{-1}$. 记第 j 列为入基变量，第 r 行为出基变量。注意第 r 行的形式为

  $$h^\top (b, A),$$

  这里 $h^\top$ 是 $\mathbf{B}^{-1}$ 的第 r 行。所以，当对第 0 行做一次行变换后，其形式如下：

  $$(0, c^\top) - p^\top (b, A),$$

  p 是某个向量。行变换后我们有

  $$0 = \bar{c}_j = c_j - p^\top A_j.$$

  对于单纯形表中, 原基矩阵 $\mathbf{B}$ 中保留的列 $A_{B(1)}, A_{B(2)}, ..., A_{B(r-1)}, A_{B(r+1)}, ..., A_{B(m)}$, 其对应 $\mathbf{B}^{-1}A$ 的第 r 行为 0，所以行变换不改变其最优判定值，即

  $$\bar{c}_{B(i)} = c_{B(i)} = 0.$$

因此，$c_{\tilde{B}}^\top - p^\top \tilde{\mathbf{B}} = 0$. 故，$p^\top = c_{\tilde{B}}^\top \tilde{\mathbf{B}}^{-1}$. 从而，新表中的第 0 行为

$$(0, c^\top) - p^\top (b, A) = (0, c^\top) - c_{\tilde{B}}^\top \tilde{\mathbf{B}}^{-1}(b, A),$$

是我们想要的形式。

- 对于非出基坐标 r 对应的行，需要先将 $\mathbf{B}^{-1}\tilde{\mathbf{B}}$ 的第 r 行乘以 $-\frac{u_{ri}}{u_{rr}}$ 加到第 i 行，使其变为 0，$i \neq r, i = 1, ..., m$. 最后让第 r 行除以 u_{rr}，使其变为 1.

Example 4.2 考虑问题

$$\begin{aligned} \min \quad & -10x_1 - 12x_2 - 12x_3 \\ \text{s.t.} \quad & x_1 + 2x_2 + 2x_3 \leq 20 \\ & 2x_1 + x_2 + 2x_3 \leq 20 \\ & 2x_1 + 2x_2 + x_3 \leq 20 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

加入松弛变量，我们得到如下问题：

$$\begin{aligned} \min \quad & -10x_1 - 12x_2 - 12x_3 \\ \text{s.t.} \quad & x_1 + 2x_2 + 2x_3 + x_4 = 20 \\ & 2x_1 + x_2 + 2x_3 + x_5 = 20 \\ & 2x_1 + 2x_2 + x_3 + x_6 = 20 \\ & x_1, \ldots, x_6 \geq 0. \end{aligned}$$

$x = (0, 0, 0, 20, 20, 20)$ 是一个可行基解. 所以, $B(1) = 4, B(2) = 5$, *and* $B(3) = 6$. 基矩阵为单位阵 I. 我们有 $c_B = 0$, $c_B' x_B = 0$ 以及 $\bar{c} = c$.

		x_1	x_2	x_3	x_4	x_5	x_6
	0	-10	-12	-12	0	0	0
x_4=	20	1	2	2	1	0	0
x_5=	20	2 *	1	2	0	1	0
x_6=	20	2	2	1	0	0	1

表 4.1: 初始单纯形表。由于第 0 行中，-10 对应的 x_1 表示可以转轴到 x_1 减小函数目标值，第 2 列 -12 对应的 x_2 表示可以转轴到 x_2 减小函数目标值。第 3 列 -12 对应的表示转轴到 x_3 减小函数目标值. 我们任选其中一个，不妨就选 x_1 这列。此时，$d_B = -\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$.

因为 20/2 最小，故将 2 标 * 号。这表明：入基变量 x_1, 出基 x_5. 将第 2 行乘以 5 加到第 0 行，将第 2 行乘以 0.5 减到第一行，再用第 3 行减去第 2 行，最后将第二行除以 2. 得到下一张表。

		x_1	x_2	x_3	x_4	x_5	x_6
	100	0	-7	-2	0	5	0
x_4	10	0	1.5	1*	1	-0.5	0
x_1=	10	1	0.5	1	0	0.5	0
x_6=	0	0	1	-1	0	-1	1

表 4.2: 入基变量 x_3, 出基 x_4

		x_1	x_2	x_3	x_4	x_5	x_6
	120	0	-4	0	2	4	0
x_3=	10	0	1.5	1	1	-0.5	0
x_1=	0	1	-1	0	-1	1	0
x_6=	10	0	2.5*	0	1	-1.5	1

表 4.3: 入基变量 x_2, 出基 x_6

		x_1	x_2	x_3	x_4	x_5	x_6
	136	0	0	0	3.6	1.6	1.6
x_3=	4	0	0	1	0.4	0.4	-0.6
x_1=	4	1	0	0	-0.6	0.4	0.4
x_2=	4	0	1	0	0.4	-0.6	0.4

表 4.4: 第 0 行全为非负，最优解得到, 最小值为-136

3.1 单纯性表：退化基解的循环

当存在退化解时，下面这个例子说明单纯形法可能陷入循环。

Example 4.3 假设我们有如下初始单纯形表：

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	$-3/4$	20	$-1/2$	6	0	0	0
x_5 =	0	1/4*	-8	-1	9	1	0	0
x_6 =	0	1/2	-12	$-1/2$	3	0	1	0
x_7 =	1	0	0	1	0	0	0	1

采用如下规则选取出基入基下标：

1. 对于非基向量，选取下标 j 使得减少量 $\bar{c}_j$ 最小
2. 对于基向量，出基下标，选取所有满足条件中下标最小的 B_r。

图片 *4.1*中的 *6* 个表格，展示了循环的情况，目标值未改变。

如何处理存在退化可行基解情况 若遇到退化基解，可采取 Bland's rule[最小下标转轴规则] 避免循环:

1. 若非基向量中，存在多个 $\bar{c}_i < 0, i \in N$. 选取 j 为最小的下标。
2. 出基过程中，即在计算下标 r 使得 $\theta^* = \min_{\{i|d_{B_i}<0\}}(-\frac{x_{B_i}}{d_{B_i}})$ 时，选取最小的下标 B_r.

感兴趣，可参考：Bland, Robert G. (May 1977). "New finite pivoting rules for the simplex method". Mathematics of Operations Research. 2 (2): 103–107. doi:10.1287/moor.2.2.103. JSTOR 3689647. MR 0459599.

作业 4.1 用单纯形表求解如下问题：

$$\begin{aligned} \min \quad & -4x_1 - x_2 \\ \text{s.t.} \quad & -x_1 + 2x_2 \le 4 \\ & 2x_1 + 3x_2 \le 12 \\ & x_1 - x_2 \le 3 \\ & x_1, x_2 \ge 0. \end{aligned} \tag{4.6}$$

3.2 单纯形表初始化

初始可行基解： 前述单纯形法需要一个可行基解作为初始解。

一般问题中为给出初始可行基解，我们在这里介绍两种常用的初始化方法：

- 两阶段法
- 大 M 法

3.3 两阶段法

两阶段法我们只做简单的介绍。

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	−4	−7/2	33	3	0	0
$x_1 =$	0	1	−32	−4	36	4	0	0
$x_6 =$	0	0	4*	3/2	−15	−2	1	0
$x_7 =$	1	0	0	1	0	0	0	1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	0	0	−2	18	1	1	0
$x_1 =$	0	1	0	8*	−84	−12	8	0
$x_2 =$	0	0	1	3/8	−15/4	−1/2	1/4	0
$x_7 =$	1	0	0	1	0	0	0	1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	1/4	0	0	−3	−2	3	0
$x_3 =$	0	1/8	0	1	−21/2	−3/2	1	0
$x_2 =$	0	−3/64	1	0	3/16*	1/16	−1/8	0
$x_7 =$	1	−1/8	0	0	21/2	3/2	−1	1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	−1/2	16	0	0	−1	1	0
$x_3 =$	0	−5/2	56	1	0	2*	−6	0
$x_4 =$	0	−1/4	16/3	0	1	1/3	−2/3	0
$x_7 =$	1	5/2	−56	0	0	−2	6	1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	−7/4	44	1/2	0	0	−2	0
$x_5 =$	0	−5/4	28	1/2	0	1	−3	0
$x_4 =$	0	1/6	−4	−1/6	1	0	1/3*	0
$x_7 =$	1	0	0	1	0	0	0	1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	−3/4	20	−1/2	6	0	0	0
$x_5 =$	0	1/4	−8	−1	9	1	0	0
$x_6 =$	0	1/2	−12	−1/2	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

图 4.1: 表 1-6，回到了原点。目标值一直未改变。

两阶段法：第一阶段求一个辅助问题的最优解；第二阶段再以这个最优解，为原问题的初始可行基解。

在**两阶段法**的第一阶段里，针对线性规划问题

$$\{\min c^\top x \mid Ax = b, x \geq 0\},$$

考虑如下的辅助问题

$$\begin{array}{ll} \min & 1^\top y \\ \text{s.t.} & Ax + y = b \\ & x \geq 0, y \geq 0. \end{array} \qquad \text{(辅助问题)}$$

这里，y 为新引进的 m 维人工变量向量，1 是所有分量均为 1 的常数向量.
只需要对等式进行符号处理，我们不妨可以假设 $b \geq 0$.

在 (辅助问题)中，$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}$ 是可行基解。把该可行基解当作初始解，就可直接用单纯形法进行求解。

(辅助问题)的目标函数 $1^\top y$ 在可行域里取值是非负的。如果原问题存在可行解 x，那么 $\begin{pmatrix} x \\ 0 \end{pmatrix}$ 满足(辅助问题)的约束条件，且目标函数值为 0.

- 当辅助问题的最优解 $\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$ 成立 $1^\top \bar{y} > 0$ 时，原问题没有可行解。
- 反之，如果辅助问题的最优解为 $\begin{pmatrix} \bar{x} \\ 0 \end{pmatrix}$. 此时，$\bar{x}$ 显然是原问题的一个可行解。

1. 如果辅助问题的基矩阵 **B** 的列全部属于 A, $\bar{x}$ 是可行基解，done!
2. 否则，由于辅助问题的基矩阵 **B** 的某些列不在 A 中，由于 A 的秩为 m，$\begin{pmatrix} \bar{x} \\ 0 \end{pmatrix}$ 是辅助问题的退化解，$\bar{x}$ 中必定有大于 $n - m$ 个坐标为 0. 我们只需要在 A 的列中，找出一些与已有的基列线性无关的列即可。

3.4 大 M 法

先求出初始可行基解后再进行第二阶段最优解的计算合并成一步作业的方法，称为**大 M 法**。

大 M 法考虑如下线性规划问题

$$\begin{array}{ll} \min & c^\top x + M \cdot 1^\top y \\ \text{s.t.} & Ax + y = b \\ & x \geq 0, y \geq 0. \end{array} \qquad (4.7)$$

这里 $M > 0$ 为充分大的常数，我们也假设 $b \geq 0$, y 和常数向量 1 与两阶段法中的一样。故，我们直接得到了一个可行基解

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}.$$

在 大 M 法的问题(4.7)中，目标函数项 $M \cdot 1^\top y$ 占支配地位，所以单纯形法迭代将首先让人工变量 y_i 的值减小。其结果是，若所有人工变量都成为 0, 则目标函数 $c^\top x + M \cdot 1^\top y$ 和原问题的目标函数 $c^\top x$ 本质上一致，可认为在此之后单纯形法的迭代所进行的就是求原问题最优解的计算。

事实上，如果原问题可行且最优值有限，在 M 充分大时，得到大 M 法问题(4.7)的最优解 $\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$ 中 $\bar{y} = 0$，且 $\bar{x}$ 成为原问题的最优解。实际计算中，我们可以保留 M，默认其是一个大数。进行正常的单纯形表更新。下面的例子进一步说明大 M 单纯形法是如何工作的。

Example 4.4 *用大 M 法求解如下问题:*

$$\begin{aligned} \min \quad & x_1 + x_2 - 3x_3 \\ \text{s.t.} \quad & x_1 - 2x_2 + x_3 \leq 11 \\ & 2x_1 + x_2 - 4x_3 \geq 3 \\ & x_1 - 2x_3 = 1 \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{4.8}$$

标准化模型（大 M 法），先添加松弛变量 x_4 和 x_5 标准化，将不等式变为等式。由于 x_4 已经对应了形如 $(1,0,0)^\top$ 的基底，故第一个等式约束不用再添加大 M 变量，只对第 2, 3 个等式约束添加 x_6 和 x_7 引入大 M 变量:

$$\begin{aligned} \min \quad & x_1 + x_2 - 3x_3 + M(x_6 + x_7) \\ \text{s.t.} \quad & x_1 - 2x_2 + x_3 + x_4 = 11 \\ & 2x_1 + x_2 - 4x_3 - x_5 + x_6 = 3 \\ & x_1 - 2x_3 + x_7 = 1 \\ & x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0. \end{aligned} \tag{4.9}$$

表 4.5: 迭代单纯形表-1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	$-4M$	$1-3M$	$1-M$	$-3+6M$	0	M	0	0
x_4	11	1	-2	1	1	0	0	0
x_6	3	2	1	-4	0	-1	1	0
x_7	1	1*	0	-2	0	0	0	1

表 4.6: 迭代单纯形表-2

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	$-M-1$	0	$1-M$	-1	0	M	0	3M-1
x_4	10	0	-2	3	1	0	0	-1
x_6	1	0	1*	0	0	-1	1	-2
x_1	1	1	0	-2	0	0	0	1

表 4.7: 迭代单纯形表-3

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	-2	0	0	-1	0	1	0	M+1
x_4	12	0	0	3*	1	-2	2	-5
x_2	1	0	1	0	0	-1	1	-2
x_1	1	1	0	-2	0	0	0	1

表 4.8: 迭代单纯形表-4

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	2	0	0	0	1/3	1/3	2/3	M-2/3
x_3	4	0	0	1	1/3	-2/3	2/3	-5/3
x_2	1	0	1	0	0	-1	1	-2
x_1	9	1	0	0	2/3	-4/3	4/3	-7/3

4 总结

- 由多面集的几何性质，我们得到了线性规划标准形式的极点必存在。
- 线性规划的标准形式，如果最优解有限，必在某个可行基解处取到。
- 由大 M 法，我们可以通过单纯形表求解线性规划问题。并且，每个迭代的浮点数乘法复杂度为 $\mathcal{O}(m^2+mn)$.

补充：对于很多问题，通常只需要 $\mathcal{O}(m)$ 次转轴运算即可得到最优解。但是，最差的情况需要 2^n（可参考 Chapter 3.7. in Book: Bertsimas, Dimitris, and John N. Tsitsiklis. Introduction to linear optimization.

Vol. 6. Belmont, MA: Athena scientific, 1997.)。实际实现算法时，要考虑算法的稳定性（如舍入误差），以及结合数据的稀疏性（如稀疏矩阵分解求逆）进一步加速算法。可以参考：Chapter 3.3. in Book: Bertsimas, Dimitris, and John N. Tsitsiklis. Introduction to linear optimization.）

MATH 3011 运筹学 **秋季 2024**

Lecture 5: 线性规划的对偶理论

Lecturer: 陈士祥 *Scribes:* 陈士祥、*ChatGPT*

1 对偶理论

针对任意一个最优化问题，可以定义它的一个对偶问题 (Dual Problem) 。

对偶理论将揭示原问题与对偶问题之间的内在联系，为进一步深入研究线性规划的求解算法提供理论依据。

原问题：

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{LP}$$

定义 **Lagrange 函数**如下：

$$\begin{aligned} L(x, \lambda, \mu) = c^\top x + \lambda^\top (b - Ax) - \mu^\top x, \\ \mu \geq 0, \mu \in \mathbb{R}^n, \lambda \in \mathbb{R}^m. \end{aligned}$$

考虑关于 x 的无约束问题

$$\min_{x \in \mathbb{R}^n} L(x, \lambda, \mu),$$

其最优值和**对偶变量** λ, μ 有关。我们记 $g(\lambda, \mu) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu)$.

对于任意原问题的可行解 $\bar{x}$，我们有

$$g(\lambda, \mu) \leq c^\top \bar{x} + \lambda^\top (b - A\bar{x}) - \mu^\top x \leq c^\top \bar{x}, \quad \forall \mu \geq 0.$$

故，问题

$$\max_{\lambda, \mu \geq 0} g(\lambda, \mu)$$

可以看作寻找原问题的最紧下界。该问题叫做原问题的**对偶问题**。

我们可以算出 $g(\lambda, \mu)$ 的具体表达式，

$$\begin{aligned} \min_{x \in \mathbb{R}^n} L(x, \lambda, \mu) &= \min_x (c - A^\top \lambda - \mu)^\top x + \lambda^\top b \\ &= \begin{cases} b^\top \lambda & \text{if } A^\top \lambda + \mu = c, \mu \geq 0 \\ -\infty & \text{otherwise.} \end{cases} \end{aligned}$$

所以，线性规划的对偶问题如下：

$$\begin{aligned} \max \quad & \mathbf{b}^\top \lambda \\ \text{s.t.} \quad & A^\top \lambda \le \mathbf{c}. \end{aligned} \tag{DP}$$

对于一般优化问题:

$$\begin{aligned} & \min f(x) \\ \text{s.t.} \quad & c_i(x) = 0, i \in \mathcal{E}, \\ & c_i(x) \ge 0, i \in \mathcal{I}. \end{aligned}$$

一般优化问题的 Langrange 对偶：

$$\max_{\lambda,\mu \ge 0} \min_x L(x, \lambda, \mu) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i c_i(x) - \sum_{i \in \mathcal{I}} \mu_i c_i(x)$$

作业 5.1 推导出下面问题的对偶问题：

1. $$\min c^\top x, \quad s.t.\ Ax \ge b.$$

2. $$\min c^\top x, \quad s.t.\ Ax \le b.$$

3. 并说明：*(DP)* 的对偶等价于 *(LP)*. 即，对偶的对偶是原问题。

Theorem 5.1 (弱对偶定理) 任意线性规划问题 *(LP)* 及其对偶问题 *(DP)* 之间成立以下关系

$\mathbf{x}$ 是 *(LP)* 问题的可行解，λ 是 *(DP)* 问题的可行解，那么 $\mathbf{c}^\top \mathbf{x} \ge \mathbf{b}^\top \lambda$.

推论：

(a) $\mathbf{x}$ 是 (LP) 问题的可行解 $\Rightarrow \mathbf{c}^\top \mathbf{x} \ge$ 问题 (DP) 的最大值。
λ 是 (DP) 问题的可行解 $\Rightarrow \mathbf{b}^\top \lambda \le$ 问题 (LP) 的最小值。

(b) $\mathbf{x}$ 是 (LP) 问题的可行解，λ 是 (DP) 问题的可行解，且 $\mathbf{c}^\top \mathbf{x} = \mathbf{b}^\top \lambda \ \Rightarrow\ \mathbf{x}$ 是 (LP) 问题的最优解，
λ 是 (DP) 问题的最优解。

(c) 问题 (LP) 无界 $\Rightarrow$ 问题 (DP) 无可行解。
问题 (DP) 无界 $\Rightarrow$ 问题 (LP) 无可行解。

(d) **最优解存在性定理：**若 (LP) 与 (DP) 都有可行解，则它们均存在最优解。

问题：若 (LP) 与 (DP) 都有可行解，那么他们的最优值是否相等？强对偶定理告诉我们答案是肯定的！

Theorem 5.2 (强对偶定理) *原问题 (LP) 有最优解，则对偶问题 (DP) 也有最优解，且此时两方的最优值一致。*

Proof: 设 (LP) 的最优解为某个可行基解：$\bar{x} = (x_B, 0)^\top$, 那么最优值为 $c_B^\top x_B$. 由最优判定定理得：

$$c^\top - c_B^\top \mathbf{B}^{-1} A \geq 0^\top.$$

令 $\bar{\lambda} = \mathbf{B}^{-\top} c_B \in \mathbb{R}^m$, 那么，

$$A^\top \bar{\lambda} \leq c.$$

并且

$$\bar{\lambda}^\top b = c_B^\top B^{-1} b = c_B^\top x_B.$$

即, $\bar{\lambda}$ 是 (DP) 的最优解，且两者最优值一致。 ■

该定理的证明还告诉我们如下结论。
结论： 单纯形法中，$\mathbf{B}^{-\top} c_B$ 可以视为对偶变量。该结论在后面还会用到。

Theorem 5.3 (互补松弛定理) *设 $\bar{\mathbf{x}}$ 和 $\bar{\lambda}$ 分别是 (LP) 和 (DP) 的可行解，那么 $\bar{\mathbf{x}}$ 和 $\bar{\lambda}$ 是对应问题最优解的充要条件是：*

$$\begin{cases} \bar{\mathbf{x}}^\top (A^\top \bar{\lambda} - \mathbf{c}) = 0 \\ \bar{\lambda}^\top (A\bar{\mathbf{x}} - \mathbf{b}) = 0. \end{cases} \tag{5.1}$$

Proof: 由定理 5.2 证明可知，$\bar{\lambda}^\top b = c_B^\top B^{-1} b = c_B^\top x_B = \mathbf{c}^\top \mathbf{x}$, 即

$$\bar{\mathbf{x}}^\top (A^\top \bar{\lambda} - \mathbf{c}) = 0.$$

第二个等式因为 $A\bar{\mathbf{x}} - \mathbf{b} = 0$ 显然成立。 ■

Example 5.1 *利用互补松弛定理，当知道一个问题的最优解时，可求出其对偶问题的最优解。*

$$\begin{array}{ll} \min & 13x_1 + 10x_2 + 6x_3 \\ \text{s.t.} & 5x_1 + x_2 + 3x_3 = 8 \\ & 3x_1 + x_2 = 3 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{array} \quad \text{的最优解} \bar{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

对偶问题为：

$$\begin{array}{ll} \max & 8\lambda_1 + 3\lambda_2 \\ \text{s.t.} & 5\lambda_1 + 3\lambda_2 \leq 13 \\ & \lambda_1 + \lambda_2 \leq 10 \\ & 3\lambda_1 \leq 6. \end{array}$$

因为 $x_1 > 0, x_3 > 0$ 由互补松弛得，

$$5\lambda_1 + 3\lambda_2 = 13$$

$$3\lambda_1 = 6.$$

$$\text{所以对偶问题的最优解}\bar{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

2 对偶问题以及对偶变量的意义

灵敏度分析：实际问题中，很多时候是基于某些采集数据来决定模型的系数。具体来说，线性规划问题中的参数 $\mathbf{c}, A, \mathbf{b}$ 并非对真实问题的精确描述，并且他们可能是非确定的，存在随机性。在这种情况下，势必会出现参数的扰动及引起的变化。为简单起见，考虑问题

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{5.2}$$

约束条件右边常数向量作微小变化

$$\mathbf{b} + \Delta\mathbf{b} = (b_1 + \Delta b_1, \cdots, b_m + \Delta b_m)^\top,$$

得到新的线性规划问题

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} + \Delta\mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{5.3}$$

由强对偶理论，(5.3)的最优解 $\mathbf{c}^\top \mathbf{x} = \lambda^\top(\mathbf{b} + \Delta\mathbf{b})$, 故原问题的最优值关于约束资源 b 的变化率为

$$\partial \mathbf{c}^\top \mathbf{x} / \partial \mathbf{b} = \lambda.$$

由于直接解原问题，并不知道对偶问题的解，故对偶最优解也被称为 **“影子价格”**。

Example 5.2 假设一个公司 α 生产两种产品，产品 A 和产品 B。生产这两种产品需要两种资源：劳动力和原材料。每个产品的利润和每个资源的可用量是已知的，分别用 p_A, p_B 表示 A 和 B 的利润。

原问题 决策变量:

- x_A, x_B - 分别表示产品 A 和产品 B 的生产量。

目标函数:

$$\text{最大化总利润} \quad P = p_A x_A + p_B x_B$$

约束条件:

$$l_A x_A + l_B x_B \leq L \quad \textit{(劳动力)}$$

$$m_A x_A + m_B x_B \leq M \quad \textit{(原材料)}$$

l_A 和 l_B 分别表示每件产品 A, B 所需要的劳动力，m_A, m_B 分别表示每件产品 A, B 所需要的材料。

对偶问题 假设由另一个公司 β 向公司 α 购买所有的原材料和劳动力，希望总价最低。

决策变量:

- y_L, y_M - 公司 β 支付的单位劳动力和单位原材料金额。对于 α 公司来说，是所获得的单位资源收益。

目标函数:

$$\text{最小化资源总成本} \quad C = Ly_L + My_M$$

约束条件：α 公司愿意卖给 β 的条件是，获得的收益不得低于自己制造产品 A, B 的利润。

$$l_A y_L + m_A y_M \geq p_A$$
$$l_B y_L + m_B y_M \geq p_B$$

影子价格的意义 在对偶问题中，最优解的 y_L, y_M 是影子价格，它们有以下含义：

1. **资源价值**：y_L 和 y_M 表示每单位劳动力力和原材料对总利润的贡献。
2. **成本限制**：影子价格反映了在当前生产计划下，资源的稀缺程度和价值。
3. **经济决策**：如果增加资源的成本低于其影子价格，则增加该资源是有利的。也就是说，考虑原问题时，若 α 公司发现其支付的单位劳动力、资源成本小于影子价格，那么他会毫不犹豫增加招聘或者资源。

对偶问题与原问题的关系

- **强对偶性**：如果原问题和对偶问题都有可行解，则它们的最优解值相等。
- **经济解释**：对偶问题提供了资源分配和成本效益分析的另一个视角。强对偶理论表明了两种决策下的相同利润和成本，达到了平衡。

通过对偶问题和影子价格的理解，企业和组织可以更好地洞察资源分配的经济效益，从而作出更明智的生产和投资决策。

MATH 3011 运筹学 **秋季 2024**

Lecture 6: 运输问题概述

Lecturer: 陈士祥 *Scribes:* 陈士祥、*ChatGPT*

1 运输问题简介

网络中的运输问题是运筹学和优化理论中的一个基本问题，它涉及在一个网络上高效地分配和运输资源。在这个问题中，资源（如货物、信息或能源）需要从一个或多个供应地（**源点**）传输到一个或多个需求地（**汇点**），目标是最小化运输成本或最大化效率。

就数学模型而言，它们是线性规划的几个重要特例。针对线性规划模型已有高效算法，因为网络模型的特殊数学结构，利用其结构特性还可以设计出效率更高的求解算法。

运输问题的关键要素：

- 网络结构：通常表示为一个有向图，其中节点代表供应地、需求地或中转点，边代表运输路线，每条边可能有不同的运输成本和容量限制。
- 供应和需求：每个供应地都有一定量的可用资源，而每个需求地都有一定量的资源需求。
- 成本最小化：目标是找到一种运输计划，使得满足所有需求地的需求同时使得总运输成本最小。
- 容量限制：运输路线可能有容量限制，即每条路线上可以运输的最大资源量。

运输问题可以通过多种优化算法解决，其中包括：

- 线性规划：运输问题可以被公式化为线性规划问题，通过单纯形法或其他优化算法求解。
- 运输问题算法：特定的运输问题可以使用如西北角法、最小成本法或沃格尔近似法等专门算法进行初步解决，然后通过调整法优化。
- 网络流算法：一些运输问题可以转化为网络流问题，如最小费用流问题，并可以使用专门的网络流算法求解。

运输模型可扩展应用于其他领域，包括投资控制，工作调度，人员指派等。

2 一般运输模型

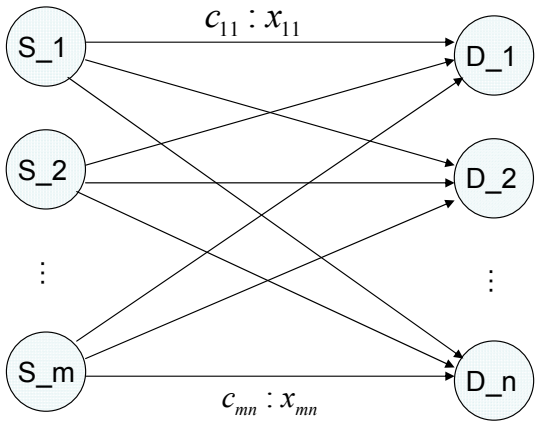
经典的运输问题：

工厂 i 生产的货物量 s_i, $i = 1, 2, \cdots, m$.

需求点 j 的需求量 d_j, $j = 1, 2, \cdots, n$.

从工厂 i 到需求点 j 的单位货运费用 c_{ij} 及其发货量 x_{ij}.

问题要求：选取一个能使运输总费用达到最小的路径规划。



如上图中的网络模型所示，每条边上，用 x_{ij} 表示从 i 到 j 的货物运输量, c_{ij} 表示单位货物运输成本。若要寻找满足要求的最低成本运输方式，有如下线性规划模型：

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} \le s_i, i = 1, \cdots, m \\ & \sum_{i=1}^{m} x_{ij} \ge d_j, j = 1, \cdots, n \\ & x_{ij} \ge 0, \quad i = 1, \cdots, m \quad j = 1, \cdots, n. \end{aligned} \tag{6.1}$$

上式中，约束 $\sum_{j=1}^{n} x_{ij} \le s_i$ 表示任意供货点 i 供货量不超过生产的总量 s_i; 约束 $\sum_{i=1}^{m} x_{ij} \ge d_j$ 表示需求点 j 接受到的总量不少于 d_j. 由于供货量实际物理意义上是非负的，故有非负约束 $x_{ij} \ge 0, \forall i, j$.

Example 6.1 假设有如下供货地:*Los Angeles*、*Detroit*、*New Orleans* 和需求地 *Denver* 和 *Miami*。

下表是城市之间的运输单价, 并且用 $i = 1, 2, 3$ 分别表示 *Los Angeles*、*Detroit*、*New Orleans*; 用 $j = 1, 2$ 表示 *Denver* 和 *Miami*。

表 6.1: Transportation Cost per Car

	Denver($j=1$)	Miami($j=2$)
Los Angeles($i=1$)	\$80	\$215
Detroit($i=2$)	\$100	\$108
New Orleans($i=3$)	\$102	\$68

下表中，我们用在最右列和最下面一行，列出供货量和需求量，并用 x_{ij} 表示 i 到 j 的供应量。该表格的构成形式，将用来设计特殊的单纯形方法。

表 6.2: MG Auto Transportation Model

	Denver	Miami	**Supply**
Los Angeles	80 x_{11}	215 x_{12}	**1000**
Detroit	100 x_{21}	108 x_{22}	**1500**
New Orleans	102 x_{31}	68 x_{32}	**1200**
Demand	**2300**	**1400**	

所以，我们列出如下线性规划问题：

$$
\begin{aligned}
\min \quad & z = 80x_{11} + 215x_{12} + 100x_{21} + 108x_{22} + 102x_{31} + 68x_{32} \\
\text{s.t.} \quad & x_{11} + x_{12} \le 1000 \quad \text{(LosAngeles)} \\
& x_{21} + x_{22} \le 1500 \quad \text{(Detroit)} \\
& x_{31} + x_{32} \le 1200 \quad \text{(NewOreleans)} \\
& x_{11} + x_{21} + x_{31} \ge 2300 \quad \text{(Denver)} \\
& x_{12} + x_{22} + x_{32} \ge 1400 \quad \text{(Miami)} \\
& x_{ij} \ge 0,\ i = 1,2,3,\ j = 1,2
\end{aligned}
\tag{6.2}
$$

我们下面将讨论如何改进单纯形法求解运输模型的线性规划问题。

2.1 运输模型供需平衡情形

考虑供需平衡的情形, 即 $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$。另外，我们只考虑恰好满足供需条件的情况，即约束均为等式的情况：

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = s_i, i = 1, \cdots, m \\ & \sum_{i=1}^{m} x_{ij} = d_j, j = 1, \cdots, n \\ & x_{ij} \geq 0, \quad i = 1, \cdots, m \quad j = 1, \cdots, n. \end{aligned} \tag{6.3}$$

注： 对于供大于求（或供低于求），我们可以通过添加冗余变量，变换为供需平衡。例如，若 $\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$, 可以假设存在额外的需求方 $n+1$, 令 $c_{i,n+1} = 0, i = 1, 2, \ldots, m$, 以及 $d_{n+1} = \sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j$，从而可以构造供需平衡运输模型。**具体参考 2.3 小节**。

对于运输问题，若将其改写成标准形式的线性规划问题，其矩阵 A 形如 (未填写部分为 0)：

x_{11}	x_{12}	…	x_{1n}	x_{21}	x_{22}	…	x_{2n}	…	x_{m1}	x_{m2}	…	x_{mn}
1	1	…	1									
				1	1	…	1					
								⋮				
									1	1	…	1
1				1					1			
	1				1			…		1		
		⋮				⋮					⋮	
			1				1					1

注意，这个矩阵并非行满秩，秩为 $n+m-1$, 故可行基解最多有 $n+m-1$ 的大于 0 的分量。

Theorem 6.1 运输问题有可行解的充分必要条件是供需平衡，即 $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$。

Proof: 必要性显然。充分性：令 $x_{ij} = \frac{s_i d_j}{T}, T = \sum_{i=1}^{m} s_i$, 可以验证此为可行解。 ■

直接应用第二章的单纯形法，当然可以求解运输规划问题。然而，变量维度是 mn，单纯性表太大，不易操作。我们这里针对运输模型的特点，保留运输模型的表格形式，设计新的单纯形法。

2.2 表格作业法

首先，我们有如下运输问题的对偶形式

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} s_i u_i + \sum_{j=1}^{n} d_j v_j \\ \text{s.t.} \quad & u_i + v_j \le c_{ij}, \quad i = 1, \cdots, m, \quad j = 1, \cdots, n \end{aligned} \tag{6.4}$$

该对偶问题可以看成运输公司的收费标准，

- u_i 为对供货地 i 处的单位物资收费，v_j 为对需求地 j 的收费单价.
- 约束条件：运输公司在路径 $i \to j$ 的收费，不能超过 c_{ij}，否则无竞争力。

下面我们讨论如何利用 A 的结构，设计更好的运输问题的单纯形法。

假设有一可行基解 (x_B, x_N) 以及对应的矩阵 A 的划分 (B, N).

根据对偶理论，若 x 为最优解，那么 $(u^\top, v^\top) = c_B^\top B^{-1}$ 为对偶问题最优解。在非最优点处，我们仍记 $(u^\top, v^\top) = c_B^\top B^{-1}$, 得到相应的对偶变量。

根据最优关系 $(u^\top, v^\top) = c_B^\top B^{-1}$：我们有减少量 $\bar{c}_{ij} = c_{ij} - c_B^\top B^{-1} A_{ij} = c_{ij} - (u_i + v_j)$

我们定义判定值 σ 如下：

$$\sigma_{ij} := u_i + v_j - c_{ij}.$$

对基变量 x_{ij} 而言，减少量 $\bar{c}_{ij} = c_{ij} - c_B^\top B^{-1} A_{ij} = 0$，故 $u_i + v_j = c_{ij}$, 即 $\sigma_{ij} = 0$.

对非基变量 x_{ij} 而言，若所有 $\sigma_{ij} = u_i + v_j - c_{ij} \le 0$, 已对偶可行（因此说明原问题最优）；若某个 $\sigma_{ij} = u_i + v_j - c_{ij} > 0$, 非对偶可行（因此原问题非最优），则引进基。

运输模型的单纯形法求解步骤：由于上面说到的，运输模型的矩阵 A 非行满秩，初始化可行基解一般采用下述的西北角算法（Northwest-Corner Starting Solution）。

1. 选取一组 $m + n - 1$ 个路径，用西北角算法选取初始可行基解，见表 6.4。
2. 检验当前解是否可改进 (表 6.5)，如果可改进，则找回路 (表 6.6) 引进一个非基变量进行步 3，否则停止。
3. 当把步 2 中挑选的变量引进时，确定哪个路径应当由基解中退出
4. 调整其他基本路径的流量（满足可行性），返回到步 2.

我们将以例子来说明运输模型的单纯形算法。

表 6.3: 某公司的运输表

	D(1)	D(2)	D(3)	D(4)	**Supply**
S(1)	10 x_{11}	2 x_{12}	20 x_{13}	11 x_{14}	**15**
S(2)	12 x_{21}	7 x_{22}	9 x_{23}	20 x_{24}	**25**
S(3)	4 x_{31}	14 x_{32}	16 x_{33}	18 x_{34}	**10**
Demand	**5**	**15**	**15**	**15**	

表 6.4: 算法迭代 1:（步骤 1:）初始可行基解 (Northwest-Corner Starting Solution) 从左上角出发，令 $x_{11} = \min\{d_1, s_1\}$, 优先向右移动，尽可能满足每一行的等式约束。若已满足该行 supply 约束，则向下移动，再尽可能满足 demand 等式约束。反复进行这个过程使得原问题达到所有等式成立，这样便得到一个初始可行基解（蓝色值为可行变量）。

	D(1)	D(2)	D(3)	D(4)	**Supply**
S(1)	$c_{11} = 10$ $x_{11} = \mathbf{5}$	2 **10**	20	11	**15**
S(2)	12	7 **5**	9 **15**	20 **5**	**25**
S(3)	4	14	16	18 **10**	**10**
Demand	**5**	**15**	**15**	**15**	

表 6.5: 算法迭代 1: （步骤 2: ）. 令 $u_1 = 0$. 然后根据可行基解的对偶关系，解出所有 $n + m - 1$ 组等式 $u_i + v_j = c_{ij}$, 这里 i, j 属于可行基解下标。这样得到所有的对偶变量值。对非基变量，计算 $(u_i + v_j) - c_{ij}$ (红色方框中的值)，大于 0 的变量可以作为转轴，最大值作为入基变量。本例子中选取 x_{31} 作为入基变量。

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	**Supply**
$u_1 \equiv 0$	$c_{11} = 10$ $x_{11} = 5$	2 10	20 [-16]	11 [4]	**15**
$u_2 = 5$	12 $u_2 + v_1 - c_{21} = [3]$	7 5	9 15	20 5	**25**
$u_3 = 3$	4 [9]	14 [-9]	16 [-9]	18 10	**10**
Demand	**5**	**15**	**15**	**15**	

表 6.6: 算法迭代 1:（步骤 2:）（找回路）从入基变量出发，寻找包含可行基解的回路。本例中，回路为: $x_{31}, x_{34}, x_{24}, x_{22}, x_{12}, x_{11}, x_{31}$. 在该回路中，令入基变量由 0 增加到值 θ, 并相应的更改回路中的其他值，使得原问题等式成立。

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	**Supply**
$u_1 \equiv 0$	10 $5 - \theta$	2 $10 + \theta$	20 [-16]	11 [4]	**15**
$u_2 = 5$	12 [3]	7 $5 - \theta$	9 15	20 $5 + \theta$	**25**
$u_3 = 3$	4 θ [9]	14 [-9]	16 [-9]	18 $10 - \theta$	**10**
Demand	**5**	**15**	**15**	**15**	

Definition 6.1 (回路) *我们将表中 $x_{i_1 j_1}, x_{i_1 j_2}, x_{i_2 j_2}, x_{i_2, j_3}, \ldots, x_{i_s, j_s}, x_{i_s, j_1}, x_{i_1 j_1}$, $(i_1, i_2, \ldots, i_s$ 互不相同，且 $1 \le i_k \le m$, $j_1, j_2, \ldots, j_s$ 互不相同，且 $1 \le j_k \le m$, $1 \le k \le s)$ 形成的集合成为一个回路。*

表 6.7: 算法迭代 1:（步骤 3:）确定出基变量，令 θ 为使得不等式 $x_{ij} \geq 0$ 的最大值。本例中，x_{11} 和 x_{22} 处，$\theta = 5$ 相等，对应退化解。我们根据 bland's rule 选择下标 ij 最小的，（或者随机选择一个），让 x_{11} 出基。接着，更新对偶变量。

	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	**Supply**
$u_1 \equiv 0$	10 [-9]	2 15	20 [-16]	11 [4]	**15**
$u_2 = 5$	12 [-6]	7 0	9 15	20 10	**25**
$u_3 = 3$	4 5	14 [-9]	16 [-9]	18 5	**10**
Demand	**5**	**15**	**15**	**15**	

表 6.8: 算法迭代 2:（步骤 2-3:）重复单纯形法第 2，3 步（确定入基变量，以及找回路）

	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	**Supply**
$u_1 \equiv 0$	10 [-9]	2 $15 - \theta$	20 [-16]	11 θ [4]	**15**
$u_2 = 5$	12 [-6]	7 $0 + \theta$	9 15	20 $10 - \theta$	**25**
$u_3 = 3$	4 5	14 [-9]	16 [-9]	18 5	**10**
Demand	**5**	**15**	**15**	**15**	

表 6.9: 算法迭代 3:（步骤 2-3）所有 $u_i + v_j - c_{ij}$ 为负数，得到最优解

	$v_1 = -3$	$v_2 = 2$	$v_3 = 4$	$v_4 = 11$	**Supply**
$u_1 \equiv 0$	10 [-13]	2 5	20 [-16]	11 10	**15**
$u_2 = 5$	12 [-10]	7 10	9 15	20 [-4]	**25**
$u_3 = 7$	4 5	14 [-5]	16 [-5]	18 5	**10**
Demand	**5**	**15**	**15**	**15**	

2.3 供需不平衡情况

前面讲的表上作业法，都是以供需平衡，即

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$$

为前提的，但是实际问题中供需往往是不平衡的。就需要把供需不平衡的问题化成供需平衡的问题。

当供大于需

$$\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$$

时，运输问题的数学模型可写成

$$\min\ Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

满足

$$\begin{cases} \sum_{j=1}^{n} x_{ij} \le s_i, & (i = 1, 2, \cdots, m) \\ \sum_{i=1}^{m} x_{ij} = d_j, & (j = 1, 2, \cdots, n) \\ x_{ij} \ge 0 \end{cases}$$

由于总的产量大于需求量，就要考虑多余的物资在哪一个产地就地储存的问题。设 $x_{i,n+1}$ 是产地 s_i 的储存量，于是有：

$$\sum_{j=1}^{n+1} x_{ij} = s_i, \quad (i = 1, \cdots, m)$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \quad (j = 1, \cdots, n)$$

$$\sum_{i=1}^{m} x_{i,n+1} = \sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j = d_{n+1}$$

令

$$c_{ij} = c_i, \quad \text{当} i = 1, \cdots, m, \quad j = 1, \cdots, n \text{ 时}$$

$$c_{ij} = 0, \quad \text{当} i = 1, \cdots, m, \quad j = n+1 \text{ 时}$$

将其分别代入，得到

$$\min\ Z' = \sum_{i=1}^{m}\sum_{j=1}^{n+1} c_{ij}x_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_i x_{ij} + \sum_{i=1}^{m} c_{i,n+1}x_{i,n+1} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_i x_{ij}$$

满足

$$\begin{cases} \sum_{j=1}^{n+1} x_{ij} = s_i, & (i = 1, \cdots, m) \\ \sum_{i=1}^{m} x_{ij} = d_j, & (j = 1, \cdots, n) \\ x_{ij} \ge 0 \end{cases}$$

由于这个模型中

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j + d_{n+1} = \sum_{j=1}^{n} d_j$$

所以这是一个供需平衡的运输问题。

总之，若当供大于需时，只要增加一个假想的需求地 $j = n+1$（实际上是储存），该需求地总需要量为

$$d_{n+1} = \sum_{i=1}^{m} s_i - \sum_{j=1}^{n} d_j$$

而在单位运价表中从各产地到假想需求地的单位运价为 $c_{i,n+1} = 0$，就转化成一个供需平衡的运输问题。类似地，当需大于供时，可以在供需平衡表中增加一个假想的产地 $i = m+1$，该地产量为

$$s_{m+1} = \sum_{j=1}^{n} d_j - \sum_{i=1}^{m} s_i$$

在单位运价表上令从该假想产地到各需求地的运价 $c_{m+1,j} = 0$，同样可以转化为一个供需平衡的运输问题。

MATH 3011 运筹学 **秋季 2024**

Lecture 7: 最短路和最大流问题

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 最短路径问题

1.1 最短路径问题介绍

最短路径问题是图论和计算机科学中的一个经典问题，其目标是在一个加权图中找到两点之间的最短路径。这个问题在多种实际应用中都非常重要，比如在网络路由、地图导航、社交网络分析等领域。

最短路径问题的关键要素：

- 加权图：这个问题通常在一个加权图中提出，其中图的每条边都有一个权重，代表两个顶点之间的“距离”或“成本”。
- 起点和终点：最短路径问题通常涉及找到从一个指定的起点到一个指定的终点的最短路径。
- 路径长度：路径的长度是路径上所有边的权重之和，最短路径问题旨在最小化这个总和。

解决方法：最短路径问题可以通过多种算法来解决，其中最著名的包括：

- 迪杰斯特拉算法（Dijkstra's Algorithm）：适用于非负权重的图，可以找到从单一源点到所有其他顶点的最短路径。
- 贝尔曼-福特算法（Bellman-Ford Algorithm）：适用于含有负权重边的图，可以处理图中的负权边，同时也能检测图中的负权环。
- 弗洛伊德算法（Floyd-Warshall Algorithm）：用于计算所有顶点对之间的最短路径，适用于任何加权图。

应用实例：

- 地理信息系统：在地图应用中寻找两地之间的最短行车或步行路线。
- 网络通信：在数据包传输中找到最有效的路径，以最小化延迟和带宽消耗。
- 社交网络分析：计算社交网络中两个人之间的“距离”或“联系程度”。

最短路径问题不仅在理论研究中重要，而且在我们日常生活中的许多方面都有广泛应用。通过各种算法的应用，我们能够高效地解决这些问题，无论是在现实世界中的物理网络，还是在虚拟世界中的信息网络。我们只探讨非负权重的图，因此只探讨最短路径的线性规划建模以及 Dijkstra 算法。

1.2 图、网络、成本、容量

设 $G = (V, E)$ 为有向图 (无向图也可)，其中 V 是节点的集合，E 是边的集合。有时把某个节点作为初始点 s, 另一个作为终点 t 而特殊对待。

各边 $e \in E$ 上赋有成本 $c(e)$ 以及容量 $u(e)$, 且都取实数值。每条边可以表示为 $e = (i, j)$，i, j 为顶点。

由它们组成的 $|E|$ 维相应向量分别记为

$$\mathbf{c} = (c(e)|e \in E), \quad \mathbf{u} = (u(e)|e \in E).$$

统括这些元素的 $\mathcal{N} = (G, s, t, \mathbf{c}, \mathbf{u})$ 称为网络。

1.3 最短路径问题

给定一个有向图 (V, E)，其中节点编号为 $1, 2, \ldots, N$。每条弧 $(i, j) \in E$ 具有与之相关的成本或 "长度" c_{ij}。路径 $(i_1, i_2, \ldots, i_k)$ 的长度，完全由向前的弧组成，等于其弧的长度之和

$$\sum_{n=1}^{k-1} c_{i_n i_{n+1}}.$$

如果路径的长度在所有具有相同起点和终点的路径中最小，则称该路径为最短路径。最短路径的长度也称为最短距离。约定上，节点到其自身的最短距离为零。最短路径问题是关于找到选定的节点对之间的最短距离的问题。

大部分主要的最短路径算法都基于以下简单的命题。

Proposition 7.1 设 $d = (d_1, d_2, \ldots, d_N)$ 是一个向量，满足

$$d_j \leq d_i + c_{ij}, \quad \forall (i, j) \in E. \tag{I.1}$$

且设 P 是从节点 i_1 开始并在节点 i_k 结束的路径。如果

$$d_j = d_i + c_{ij}, \quad \text{对于路径 } P \text{ 中的所有弧} (i, j) \text{成立}, \tag{I.2}$$

则 P 是从 i_1 到 i_k 的最短路径。

Proof: 通过对路径 P 的所有弧累加公式 (I.2)，我们可以看出 P 的长度为 $d_{i_k} - d_{i_1}$。通过对任何其他从 i_1 开始并在 i_k 结束的路径 P' 的所有弧累加公式(I.1)，我们可以看出 P' 的长度至少等于 $d_{i_k} - d_{i_1}$。因此，P 是最短路径。证毕。 ■

命题中 (I.1) 和 (I.2) 的条件称为最短路径问题的互补松弛条件。这将会在下面线性规划形式看到。

1.4 最短路径问题的线性规划建模

用 x_{ij} 表示顶点 i 到 j 的 “流量”，反之，用 x_{ji} 表示顶点 j 到 i 的 “流量”。$x_{ij} = 1$ 即为 i 到 j 的边在路径上，$x_{ij} = 0$ 即表示不在路径上。故首先我们可以建模为如下 0-1 整数线性规划模型：

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = 1 \\
& \sum_{(k,j)\in E} x_{kj} - \sum_{(i,k)\in E} x_{ik} = 0, \quad \forall k \in V - \{s,t\} \\
& \sum_{(t,i)\in E} x_{ti} - \sum_{(i,t)\in E} x_{it} = -1 \\
& x_{ij} \in \{0,1\}, \forall (i,j) \in E
\end{aligned} \tag{7.1}$$

上述约束中，

- $\sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = 1$ 表示从起点 s 流出的流量为 1. 事实上，可以直接写为 $\sum_{(s,j)\in E} x_{sj} = 1$. 因为若存在路径 $j \to s$，则有环，必然不是最短路。
- $\sum_{(k,j)\in E} x_{kj} - \sum_{(i,k)\in E} x_{ik} = 0, \quad \forall k \in V - \{s,t\}$ 表示除了起点 s 终点 t 的顶点，流入和流出相等。
- $\sum_{(t,i)\in E} x_{ti} - \sum_{(i,t)\in E} x_{it} = -1$ 表示流出终点的流量为 1.
- 上述所有约束构成了一条从 s 到 j 的路径。可以直接写为 $-\sum_{(i,t)\in E} x_{it} = -1$. 因为若存在路径 $t \to i$，则有环，必然不是最短路。

然而，带有形如 $x_{ij} \in \{0,1\}$ 的 01 约束问题难以求解。我们可以将其转化为**松弛问题：**

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} \\
\text{s.t.} \quad & \sum_{(s,j)\in E} x_{sj} - \sum_{(j,s)\in E} x_{js} = 1 \\
& \sum_{(k,j)\in E} x_{kj} - \sum_{(i,k)\in E} x_{ik} = 0, \quad \forall k \in V - \{s,t\} \\
& \sum_{(t,i)\in E} x_{ti} - \sum_{(i,t)\in E} x_{it} = -1 \\
& x_{ij} \geq 0, \forall (i,j) \in E
\end{aligned} \tag{7.2}$$

可以看出，从 s 到 t 的路径 P 是最短路径，当且仅当路径流 x 定义为：

$$x_{ij} = \begin{cases} 1, & \text{如果}(i,j)\text{属于}P, \\ 0, & \text{否则}. \end{cases} \tag{7.3}$$

是线性规划问题(7.1) 的一个最优解。下面我们通过线性规划的互补松弛定理和命题7.1说明问题(7.2)的最优解恰好对应整数线性规划问题(7.1)的最优解。

首先，问题(7.4)的对偶为 (我们略过推导过程)：

$$\begin{aligned} \max \quad & (y_s - y_t) \\ \text{s.t.} \quad & y_i - y_j \le c_{ij}, \quad \forall (i,j) \in E \end{aligned} \tag{7.4}$$

Proposition 7.2 *命题7.1中的互补松弛条件 (I.1)和(I.2) 实际上是线性规划问题(7.2)的对偶可行性和互补松弛条件，形式如下：*

$$y_i \le c_{ij} + y_j, \quad \forall (i,j) \in E, \tag{CS-1}$$

$$y_i = c_{ij} + y_j, \quad \text{对于所有弧 } (i,j) \text{ 满足} x_{ij} > 0. \tag{CS-2}$$

给定 x, y 分别满足原问题(7.2)和对偶问题(7.4)可行条件，并且 x, y 分别是原问题和对偶问题的最优解，则根据线性规划互补松弛定理可知有互补松弛条件(CS-2)成立。另外，由 x 的可行性条件，可知存在一条 “路径” P 由 s 到 t，且 $x_{ij} > 0, \forall (i,j) \in P$。

事实上，将 y_i 识别为命题7.1中的 $-d_i$，我们可以看到条件 (I.1) 和 (I.2) 与 CS 条件 (CS-1) 和 (CS-2) 是一致的。因此，命题7.1告诉我们 P 为最短路，故该路径 P 上存在对应的整数解满足(7.3)。

因此，一般网络的最短路径问题可以看成是一个线性规划模型（事实上是一个更特殊的运输模型），可依据对偶性构造其求解算法。

如果基于动态规划的思想，可给出最短路径问题的强多项式时间算法。以下仅说明具有代表性的算法之一的 **Dijkstra's algorithm**.

1.5 Dijkstra 算法

Dijkstra 算法描述如算法1。它的主要思想如下，我们显然有最短路径的性质：若 s 到 v 的一条最短路径经过某个顶点 c，即 $s \to c \to v$，那么这条路径上子路径 $s \to c$ 是 s 到 c 的最短路径。

Algorithm 1 Dijkstra 算法

1: [输入] 有向图或者无向图 $G=(V,E)$, 各边长度 $c: E \to \mathbb{R}_+$, 始点 $s \in V$.
2: [输出] 从始点到所有节点 $v \in V$ 的最短路径及其长度 $c^*(v)$.
3: 初始化：令 $d(s) := 0, d(v) := \infty (v \in V - \{s\})$, 以及 $X := \emptyset$
4: **while** $X \neq V$ **do**
5: 选取任一个满足 $d(v^*) = \min\{d(v) \mid v \in V - X\}$ 的顶点 $v^* \in V - X$
6: 更新：$X := X \cup \{v^*\}$.
7: 进一步对 $w \in V - X$ 的各边 $e = (v^*, w) \in E$ 作如下更新：

$$d(w) := \min\{d(w), d(v^*) + c(v^*, w)\}.$$

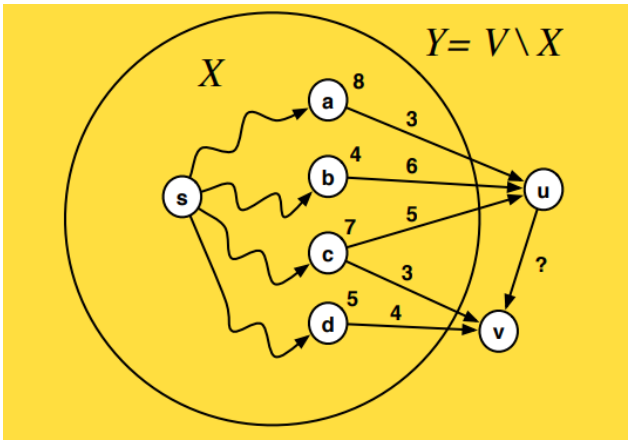
8: **end while**

算法1的各步解释如下

- 第 3 行：把图分为已访问节点 X 和未访问节点 $V - X$, 并记录 s 到所有点的（临时）最短路径为 $d(v)$. 对于已访问节点 $x \in X$，$d(x)$ 是 s 到 x 的最短路径；未访问节点 $y \in V - X$, $d(y)$ 叫做临时最短路径。

- 第 5 行详解：对于已访问节点集合 X，我们寻找 X 的所有邻居节点中，到 X 距离最小的顶点。即 $v^* \in V - X$ 是问题 $\min\limits_{x \in X,\ v \in V - X} (d(x) + c(x, v))$ 的解。

  Example 7.1 若我们有如下的图，已知 $\mathbf{s}$ 到所有 X 中所有顶点最短距离。对于 $\mathbf{u}, \mathbf{v} \in V - X$, 我们得到 $d(\mathbf{v}) = d(\mathbf{d}) + c(\mathbf{d}, \mathbf{v}) = 9$.



- 第 6-7 行：更新已访问集合 $X = X \cup \{v^*\}$, 更新临时距离 $d(w), w \in V - X$.

- 第 4 行：当所有节点均访问过，即 $X = V$ 时，结束算法；否则继续更新集合 X。

结合命题7.1，我们也可以这样看待 Dijkstra's 算法。首先，我们不知道从 s 出发到其他点的最短路径。故，我们设 d_j 都是 ∞，其对应某条“临时”路径. 若 $d_j > d_i + c_{ij}$，这说明，由 s 到 i，再从 i 到 j 的路径比“临时”路径更短，所以可以更新它为 $d_j = d_i + c_{ij}$。由第 5 行可知，算法每次新加入 X 的点 j，对于任意 $i \in X$ 都满足 $d_j \leq d_i + c_{ij}$. 直到 $d_j \leq d_i + c_{ij}$ 对所有点都成立，算法停止。故，Dijkstra 可以看为是不断满足对偶可行条件(CS-1)和互补松弛(CS-2)的算法。具体证明参考https://web.mit.edu/dimitrib/www/LNets_Full_Book.pdf中的命题 3.3.

Example 7.2 *如图7.1，给定网络和各边上的路径长度，我们用表格7.1的方式展示 Dijkstra 算法。*

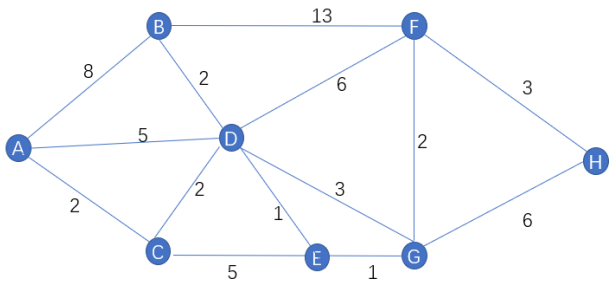


图 7.1: Dijkstra 算法示例

X	A	B	C	D	E	F	G	H
A	0	8_A	2_A	5_A	∞	∞	∞	∞
C		8_A	2_A	4_C	7_C	∞	∞	∞
D		6_D		4_C	5_D	10_D	7_D	∞
E		6_D			5_D	10_D	6_E	∞
B		6_D				10_D	6_E	∞
G						8_G	6_E	12_G
F						8_G		11_F
H								11_F

表 7.1: Dijkstra 算法表格。第 1 列，描述了每一步迭代过程中加入 X 的点。每一行是每一步迭代后计算出的由 A 到各个点的临时距离，右下角字母表示路径的上一个节点。红色方框标出的数字对应的列，表示即将加入 X 中的新节点。

2 最大流问题

2.1 最大流问题概述

最大流问题是图论和网络优化中的一个经典问题，其核心目标是确定在一个网络（通常是一个有向图）中从一个特定的源点（source）到一个特定的汇点（sink）能够传输的最大流量。这个问题在诸如交通系统、通信网络、管道网络等许多实际应用场景中非常重要。

2.2 最大流问题的关键要素

- **有向图**：问题通常在一个有向图中提出，图中的每条边有一个非负容量，代表这条边可以承载的最大流量。
- **源点和汇点**：指定一个源点和一个汇点。流量从源点出发，流向汇点。
- **边容量限制**：图中每条边的流量不能超过其容量。
- **流守恒约束**：除了源点和汇点外，图中每个顶点的流入量必须等于流出量。

2.2.1 解决方法

最大流问题可以通过以下算法来解决：

1. **Ford-Fulkerson 方法**：通过不断寻找从源点到汇点的增广路径来增加网络的流量，直到无法找到更多的增广路径为止。
2. **Edmonds-Karp 算法**：Ford-Fulkerson 方法的一个特定实现，使用广度优先搜索来寻找增广路径，确保算法的多项式时间复杂度。
3. **Dinic 算法**：通过构建层次图来实现更高效的搜索增广路径，通常比 Ford-Fulkerson 方法和 Edmonds-Karp 算法更快。

我们只介绍 Ford-Fulkerson 方法。

2.2.2 应用实例

- **交通系统**：计算道路或交通网络中的最大车流量。
- **通信网络**：确定数据在网络中从一个点到另一个点的最大传输速率。
- **管道网络**：在水管或石油管道网络中，确定最大的流体传输能力。

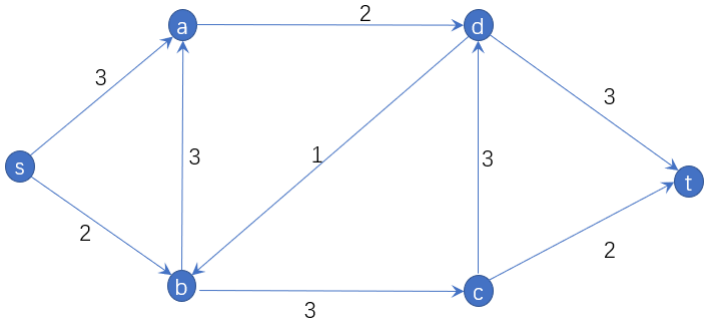


图 7.2: 有向网络图示，各边均指定了方向，每条边给定了最大流量。问题即为确定从 s 出发到达 t，网络中的最大流量。

2.3 最大流算法

首先，我们给出该问题的完整数学定义。

Definition 7.1 (Flow) *流量 (flow) 是有向图 $G = (V, E)$ 上的一个函数 $f : E \to R$, 满足如下条件:*

1. **容量约束:** $0 \le f(e) \le c(e)$.
2. **流量守恒:** *对于任意 $v \in V - \{s, t\}$, 有 $\sum_{e \in \mathrm{In}(v)} f(e) = \sum_{e \in \mathrm{Out}(v)} f(e)$, 这里 $\mathrm{Out}(v)$ 和 $\mathrm{In}(v)$ 分别表示 G 中流出和流入 v 节点的边集合。*

f 也可以写成顶点的函数，即 $f : V \times V \to R$. 相应地,

1. **容量约束:** 若 $(u, v) \in E$, 则 $0 \le f(u, v) \le c(u, v)$. (若 u, v 不直接相连, $f(u, v) = 0$.)
2. **流量守恒:** 对于任意 $u \in V - \{s, t\}$, 有 $\sum_{v \in V} f(u, v) = 0$.
3. **反对称:** $f(u, v) = -f(v, u)$.

Definition 7.2 *从 s 的净流出量，定义为流 f 的值, 记作 $|f|$,*

$$|f| \triangleq \sum_{v \in V} f(s, v) = f(s, V)$$

Lemma 7.1 *有如下结论*

- $f(X, X) = 0$,
- $f(X, Y) = -f(Y, X)$
- $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$, *if* $X \cap Y = \emptyset$.

- $f(u,V)=0,\ \forall u\in V-\{s,t\}$.

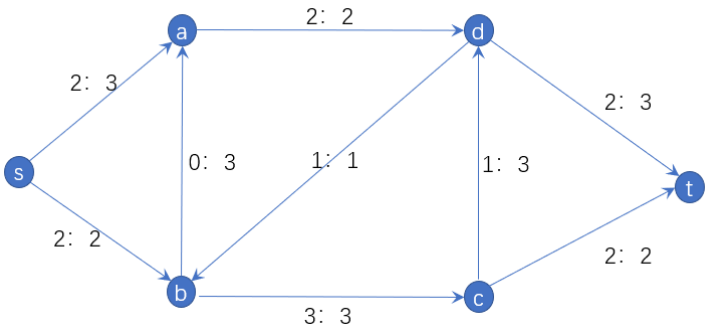
Theorem 7.1 流的值 $|f|$ 等于流出 t 的流的和：

$$|f|=\sum_{v\in V}f(v,t)=f(V,t).$$

证明：

$$\begin{aligned}|f|&=f(s,V)\\&=f(V,V)-f(V-s,V)\\&=f(V,V-s)\\&=f(V,t)+f(V,V-s-t)\\&=f(V,t).\end{aligned}\tag{7.5}$$

所谓最大流问题，就是求出使流值达最大的可行流问题，如下图，我们在图7.2中，给每条边添加满足条件的流量，得到 s 到 t 的流值为 4，等于从 s 的流出值和流入 t 的流值. 显然，若是改变某一边的流量，将会影响整个网络中的流量，故最大流问题是比较复杂的问题。



与最短路类似，我们也可也用线性规划可描述最大流问题：

$$\begin{aligned}\max\quad & f(s,V)=\textstyle\sum_{v:(s,v)\in E,v\in V}f(s,v)\\ \text{s.t.}\quad & \sum_{v\in V}f(v,u)=\textstyle\sum_{w\in V}f(u,w),\forall u\in V-s-t\\ & 0\le f(u,v)\le c(u,v),(u,v)\in E.\end{aligned}\tag{7.6}$$

求解上述线性规划问题可以得到最大流的最优解。但是很多情况下，这并不是一种有效率的方法。下面我们继续讨论该问题的最优解的性质，与之密切相关的问题叫做最小割问题。

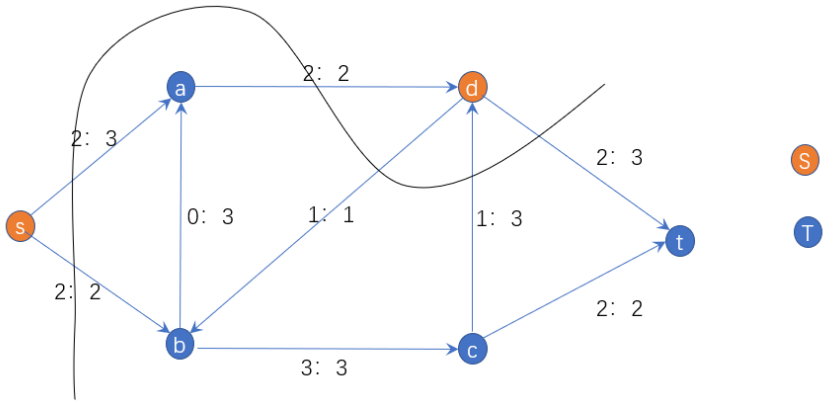
2.3.1 最小割问题 (Minimum Cut Problem)

Definition 7.3 流网络 $G=(V,E)$ 的一个割 (S,T) 是指，顶点集合 V 的一个分割，$S\cap T=\emptyset, S\cup T=V$，使得 $s\in S, t\in T$. 如果 f 是 G 的流量，那么一个割上的流量为 $f(S,T)$.

也就是说，割把一个流网络的顶点集划分成两个集合 S 和 T, 使得源点 $s \in S$ 且汇点 $t \in T$.

Definition 7.4 分割 (S,T) 的容量为

$$c(S,T) = \sum_{\substack{(u,v) \in E \\ u \in S,\ v \in T}} c(u,v).$$



如图，$S = \{s,d\}$, $T = \{a,b,c,t\}$. 分割的容量为 $c(S,T) = 3+2+1+3 = 9$. 割流量为 $f(S,T) = 2+2+(-2+1-1-2) = 4$.

所谓最小割问题，即找到一种割法，使割的容量最小。也就是说，找到通过能力最弱的断面。

Lemma 7.2 对于任意割，我们有 $|f| = f(S,T)$. 另外 $|f| \le c(S,T)$.

Proof:

$$\begin{aligned} f(S,T) &= f(S,V) - f(S,S) \\ &= f(S,V) \\ &= f(s,V) + f(S-s,V) \\ &= f(s,V) \\ &= |f|. \end{aligned} \tag{7.7}$$

$$\begin{aligned} |f| &= f(S,T) \\ &= \sum_{u \in S} \sum_{v \in T} f(u,v) \\ &\le \sum_{u \in S} \sum_{v \in T} c(u,v) \\ &= c(S,T). \end{aligned} \tag{7.8}$$

■

从直观上看，从源点 s 到汇点 t 的路，必然经过割集 $(S,T)=\{(u,v)\in E, u\in S, v\in T\}$，如果该路径的净流量已达到割容量，则无法继续增加流量。于是我们可以得到下面的定理。

Theorem 7.2 最大流最小割定理：任意一个流网络的最大流量等于该网络的最小割的容量。

该定理具体证明参考 https://web.mit.edu/dimitrib/www/LNets_Full_Book.pdf的 Proposition 2.3。另外，可以写出最大流线性规划问题(7.6)的对偶问题，并且其是最小割问题的对偶问题。 因此，由线性规划强对偶定理也可以证明上面的结论。

Ford-Fulkerson 算法的主要思想是对最大流原问题的目标值不断增大，主要方法是通过构造如下的余网络和流扩充路，直至路径流量达到最小割容量。

2.4 余网络（residual network）和流扩充路（augmenting path）

网络 $\mathcal{N}=(G,s,t,\mathbf{c})$ 中给定一个可行流 f，**余网络** 构造方式如下：

对于 $\mathcal{N}$ 的各边 $e=(u,v)\in E$，按照以下规则生成边 (u,v) 或 (v,u)，得到的有向边集合记为 E_f，同时确定其容量 c_f.

R-1 如果 $c(e)-f(e)>0$, 则生成 $(u,v)\in E_f$, 并令其容量 $c_f(u,v)=c(u,v)-f(u,v)$。这样，对应 E_f 中的边 (u,v) 可以继续增大流。

R-2 如果 $f(u,v)>0$, 则生成 $(v,u)\in E_f$, 并令其容量 $c_f(v,u)=f(u,v)$. 这样，对应 E_f 中的边 (v,u) 可以减少流。

所得到的有向网中所有容量 $\bar{c}>0$ 的边构成余网络 $\mathcal{N}_f=(G_f,s,t,c_f)$, 其中有向图 $G_f=(V,E_f)$, 容量 $c_f=\{c_f(e)>0\mid\forall e\in E_f\}$. 如图7.3展示了原网络和生成的余网络。余网络 $\mathcal{N}_f$ 中从 s 到 t 的路称为**流扩充路**。

Lemma 7.3 (流扩充路) 给定网络 $\mathcal{N}=(G,s,t,c)$, 其可行流 f 为最大流的充要条件是，余网络 $\mathcal{N}_f$ 中不存在有流扩充路。

如果存在有流扩充路 p，则 f 可修正为流值更大的流, 通过沿着 p 对原网络上的相应路径更改流 f, 更新量为 $c_f(p)=\min_{(u,v)\in p}\{c_f(u,v)\}$.

算法 Ford-Fulkerson Algorithm 描述如下

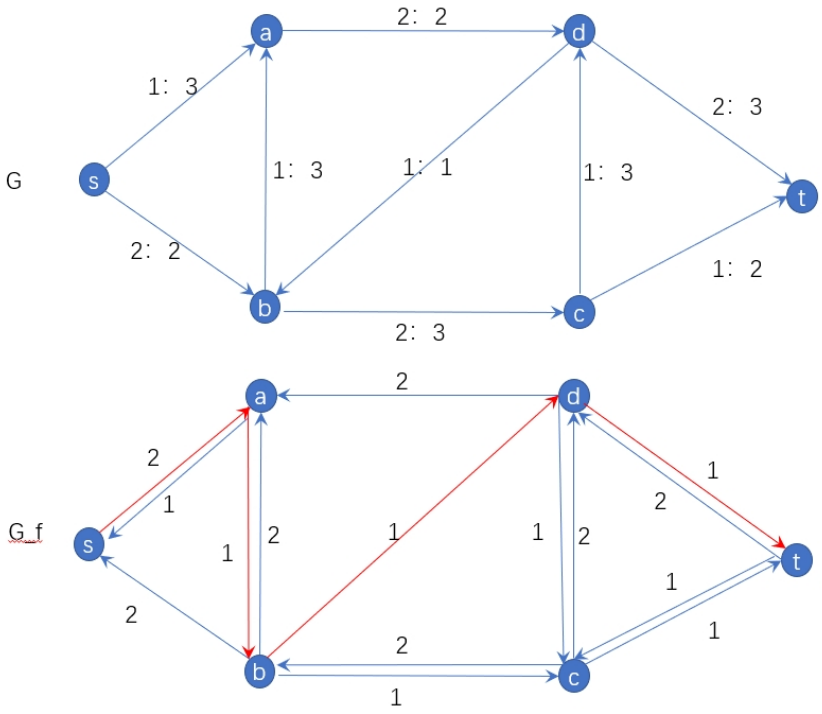


图 7.3: 原网络 (上) 和余网络 (下)。流扩充路为标红的边。

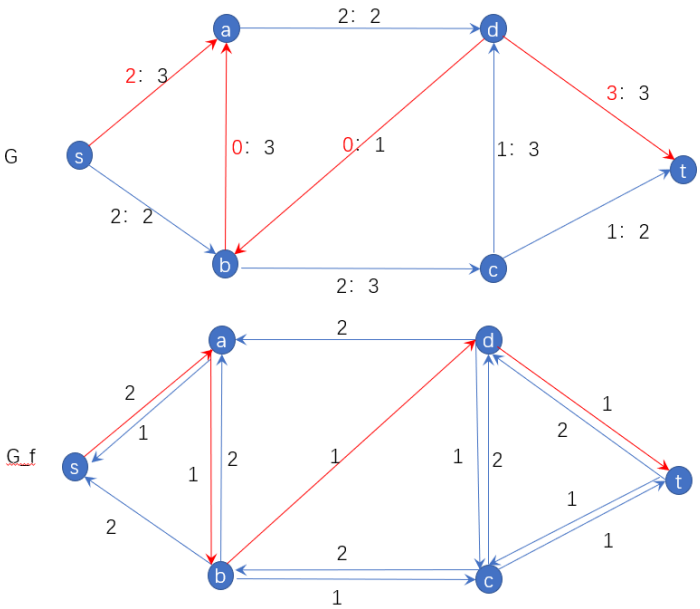


图 7.4: 根据流扩充路，更改原图中的流 (上图), 下图为原图的流扩充路。

输入 网络 $\mathcal{N}=(G,s,t,c)$, 其中有向图 $G=(V,E)$.

输出 从 s 到 t 的最大流值 $f_{\max}$.

步一 初始化：令 $f(e):=0(e\in E)$ 以及 $f_{\max}:=0$.

步二 余网络：构造当前可行流 f 的余网络 $\mathcal{N}_f=(G_f,s,t,c_f)$.

步三 流的扩充：如果 $\mathcal{N}_f$ 中没有流的扩充路则结束计算。相反，如果存在有流的扩充路，则选取其一 p, 通过沿着 p 对 f 增加 $c_f(p)=\min_{(u,v)\in p}\{c_f(u,v)\}$. (若 G_f 中的边与 G 中的有向边相反，则减去相应的值。) 令 $f_{\max}=f+c_f(p)$. 回到 步二。

3 最小成本流问题

考虑网络 $\mathcal{N}=(G,s,t,\mathbf{c},w)$ 上的流 f, 其中 $w(u,v)$ 是边 (u,v) 的费用，其中流值固定为

$$|f|=f^*.$$

在流值 $|f|=f^*$ 不超过网络的最大流的条件下，求出使成本达最小的流就是所谓最小成本流问题。

Example 7.3 有足够多辆卡车要将数量无限的某种物品从一个地点运输到另外一个地点，现在有有限条单向行驶道路直接或者间接地连接了这两地。但是每一条道路都有运输通过总数量的限制，称为容量，同时携带物品通过该路段时，都会按照携带物品数量多少被收取一定的费用。如何合理地安排每辆车的行驶路线，使得在完成一定运输量的情况下，交付的总费用尽可能少？

注意，在此问题中总费用仅包括携带物品通过路段时被收取的费用，车辆和路线安排上没有限制，但通过某一路段的物品数量总和不得超过它的容量，收取的费用与携带物品的多少成正比。

最小成本流问题的线性规划模型：

$$\begin{aligned}
\min\quad & \sum_{e\in E} w(e)f(e)\\
\text{s.t.}\quad & \sum_{e\in \mathrm{Out}(v)} f(e)-\sum_{e\in \mathrm{In}(v)} f(e)=0, v\in V-\{s,t\}\\
& \sum_{e\in \mathrm{Out}(s)} f(e)-\sum_{e\in \mathrm{In}(s)} f(e)=f^*\\
& \sum_{e\in \mathrm{Out}(t)} f(e)-\sum_{e\in \mathrm{In}(t)} f(e)=-f^*\\
& 0\le f(e)\le c(e), e\in E.
\end{aligned}\tag{7.9}$$

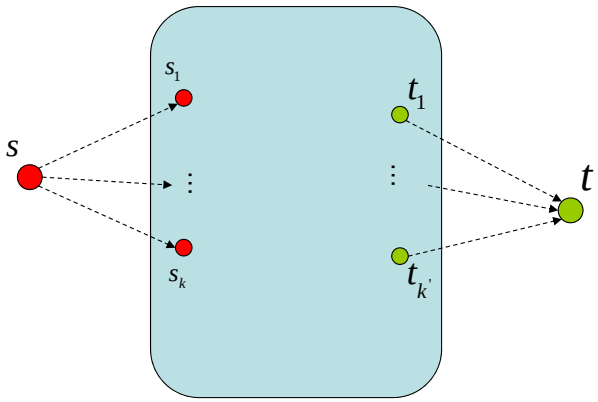


图 7.5: 多源点和多汇点图示，添加哑元源点和哑元汇点。

3.1 多个源点汇点的处理

在最小成本流问题中，可以有多个源点和汇点，运输问题便是一个例子。如果最小成本流问题具有 k 个源点 $s_i, i = 1, \cdots, k$ 以及各自的流出量 $f(s_i) = f^*(s_i)$, 还有 k' 个汇点 $t_j, j = 1, \cdots, k'$ 以及各自的流出量 $f(t_j) = f^*(t_j)$, 则可以引入哑元源点 s 和哑元汇点 t, 并添加 $(k + k')$ 条边

$$(s, s_i): \ w(s, s_i) = 0, c(s, s_i) = f^*(s_i), i = 1, \cdots, k;$$

$$(t_j, t): \ w(t_j, t) = 0, c(t_j, t) = f^*(t_j), j = 1, \cdots, k'.$$

然后我们令

$$f^* = \sum_{i=1}^{k} f^*(s_i),$$

那么对于 s，其约束为

$$\sum_{e \in \mathrm{Out}(s)} f(e) = f^*.$$

从而得到单汇点和单源点的问题。

由于 $c(s, s_i) = f^*(s_i)$, 必然有 $f(s, s_i) = c(s, s_i) = f^*(s_i)$, 所以，根据流量守恒可得 s_i 出的流出量为 $f^*(s_i)$. 在运输问题中，这对应由供应点 i 的供应量为 $f_i^*(s_i) = \sum_{e \in \mathrm{Out}(s_i)} f(e)$. 相应地，$t_i$ 为需求地，也满足供需平衡运输模型中的需求条件 $f_i^*(t_i) = \sum_{e \in \mathrm{In}(t_i)} f(e)$. 因此，运输模型是一个特殊的最小成本流问题。

3.2 最短路问题转化为最小成本流问题

最小成本流问题具有很好的模型化能力：

(一) 考虑如下最短路径问题：由 s 出发，到任意点 $v \in V \setminus \{s\}$ 的最短路。
通过引入哑元终点 t', 加入从 $v \in V$ 出发的边 (v, t') 且满足

$$w(v, t') = 0, \quad c(v, t') = 1.$$

那么，最短路径问题成为求解从 s 到 t' 的具有流值 $f^* = n$ 的最小成本流问题。这里，n 是图 G 的顶点个数, 并假定原网络各边的容量全为 ∞。

3.3 最大流问题转化为最小成本流问题

(二) 最大流问题：通过引入哑元始点 s', 并构造如下两条边：

$$(s', s); \quad w(s', s) = 0, \quad c(s', s) = \infty$$

$$(s', t); \quad w(s', t) = 1, \quad c(s', t) = \infty$$

那么，最大流问题成为求解从 s' 到 t 的最小成本流问题。这里，设定原网络里 $w(e) = 0 (e \in E)$, 并取最大流值的适当上界（比如 $\sum_{e \in E} c(e)$）为从 s' 出发的流值 f^*.

作业 7.1 通过构造说明最小成本流问题作为其特殊情况包含：最短路问题和最大流问题。

作业 7.2 考虑一个公司希望在员工与任务之间进行有效的分配。每个任务都需要特定的技能，并且每个员工都有一套技能。每个员工都有能处理的任务数的上限，每个任务需要一个员工去完成。

给定数据：

- 员工集合 $E = \{e_1, e_2, e_3\}$
- 任务集合 $T = \{t_1, t_2, t_3, t_4\}$
- 每个员工可以处理的任务数量为 $C = \{2, 1, 2\}$
- 任务分配矩阵 A 定义为：

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

其中 $A_{ij} = 1$ 表示员工 e_i 可以执行任务 t_j。

1. 描述上述数据的流网络图。

2. 使用 *Ford-Fulkerson* 算法或其他最大流算法，求出可以分配的最大任务数量。

3. 指出哪个员工应该分配哪个任务以达到最大流量。

4. 如果添加了一个新任务，它可以被 e_1 和 e_3 完成，应如何修改流网络？请指出修改后的最大流量分配。

MATH 3011 运筹学 **秋季 2024**

Lecture 8: 动态规划

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 动态规划介绍

动态规划是一种算法设计技巧，广泛应用于解决优化问题，尤其是那些可以分解为重叠子问题的复杂问题。动态规划主要思想：将问题分解为子问题，并重复使用已有的结论（即写出递归式）。

1.1 最短路问题

Example 8.1 最短路径问题的一个例子，回顾 *Dijkstra* 算法。

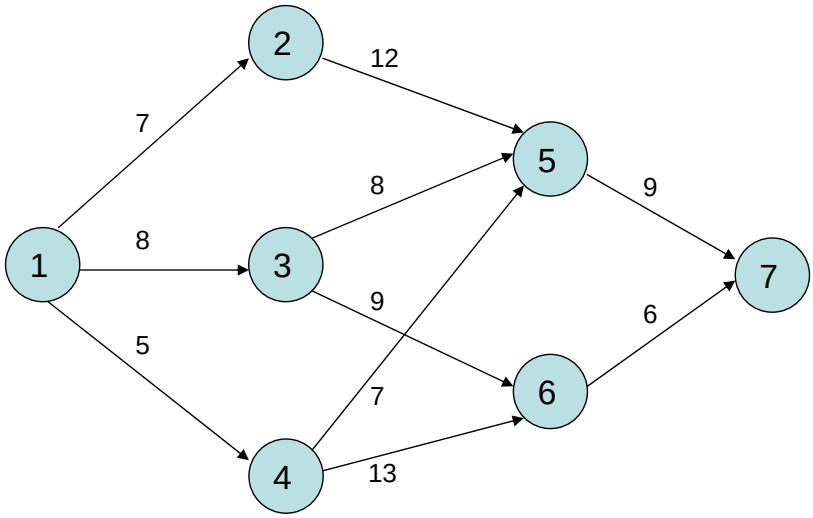


图 8.1: 最短路问题图

对问题进行分阶段考虑。与上节课的 *Dijkstra* 算法稍有区别，对于图 *8.1*中的有向图，我们可以分为如图 *8.2*中的 $\{0,1,2,3\}$ 共 *4* 个阶段，每个阶段包括图的不同顶点，分别为 $\{1\},\{2,3,4\},\{5,6\},\{7\}$。令 $f_i(x_i)$ 表示阶段 i 中的顶点 x_i 到起点的最短路长度。开始阶段，$f_0(x_0)=0$. 对于阶段 *1*，我们有

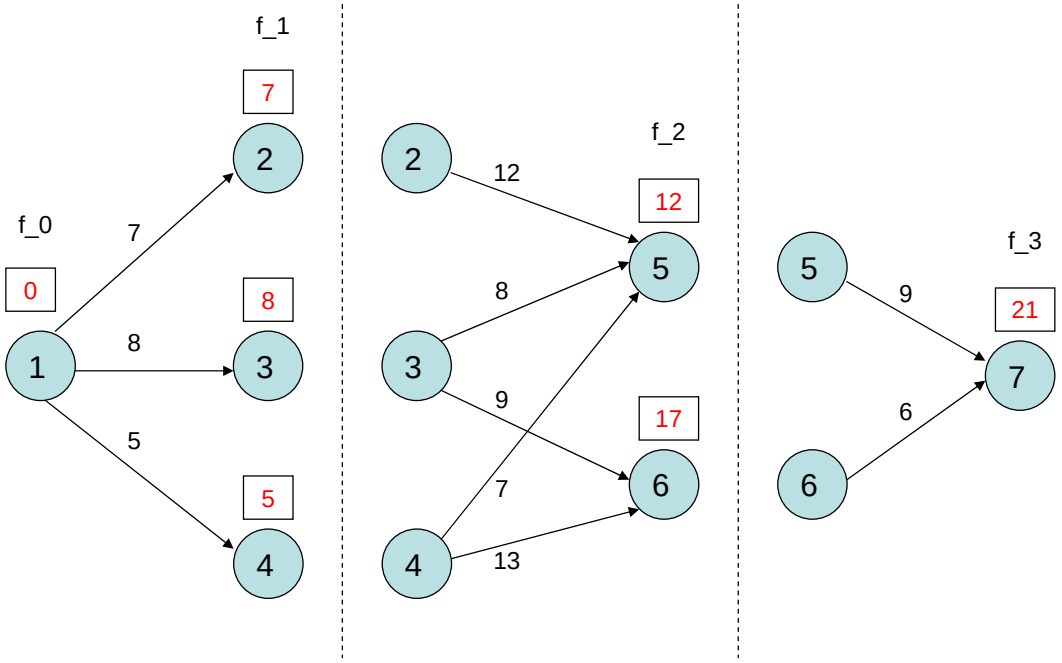


图 8.2: 有向图最短路分阶段

$f_1(x_1=2)=7, f_1(x_1=3)=8, f_1(x_1=4)=5$. 对于阶段 *2*，我们有

$$f_2(x_2)=\min_{(x_1,x_2)\in E}\{d(x_1,x_2)+f_1(x_1)\}$$

故，$f_2(x_2=5)=f_1(x_1=4)+d(4,5)=12, f_2(6)=f_1(x_1=3)+d(x_1=3,x_2=6)=17$.

我们可以写出递归方程 *(Recursive Equation)*

$$\begin{cases} f_i(x_i)=\min\limits_{(x_{i-1},x_i)\in E}\{d(x_{i-1},x_i)+f_{i-1}(x_{i-1})\}, & i=1,2,3, \\ f_0(x_0)=0. \end{cases}$$

1.2 适用场景

动态规划常常适用于有重叠子问题和最优子结构性质的问题，动态规划方法所耗时间往往远少于朴素解法。

- 最优化原理：假设问题的最优解所包括的**子问题的解也是最优的**，就称该问题具有最优子结构，即满足最优化原理。
- 无后效性：即某阶段状态一旦确定，就不受这个状态以后决策的影响。也就是说，**某状态以后的过程不会影响曾经的状态**。仅仅与当前状态有关。

- 有重叠子问题：即子问题之间是不独立的，**一个子问题在下一阶段决策中可能被多次使用到**（该性质并非动态规划适用的必要条件，可是假设没有这条性质。动态规划算法同其它算法相比就不具备优势）。

1.3 动态规划要素

动态规划简单来说就是，利用历史记录，来避免我们的重复计算。而这些历史记录，我们得需要一些变量来保存，一般是用一维数组或者二维数组来保存。下面我们先来讲下做动态规划题很重要的三大基本要素：

- **确定状态和保存状态变量：** 将问题发展到各个阶段时所处于的各种客观情况用不同的状态表示出来。最简单的就是用数组来保存当前的每一个状态，这个状态就是每个子问题的决策。
- **确定决策并写出状态转移方程：** 因为决策和状态转移有着天然的联系，状态转移就是根据上一阶段的状态和决策来导出本阶段的状态。所以如果确定了决策，状态转移方程也就可写出。但事实上常常是反过来做，根据相邻两个阶段的状态之间的关系来确定决策方法和状态转移方程。
- **确定边界条件：** 确定边界条件其实就是跟递归的终止条件是类似的。给出的状态转移方程是一个递推式，需要一个递推的终止条件或边界条件。

通过下面的背包问题，我们进一步说明这几个要素。

1.4 背包/货物装载 (Knapsack/Cargo-Loading) 问题

01 背包问题： 最基本的背包问题就是 01 背包问题（01 knapsack problem）：一共有 n 件物品，第 i（i 从 1 开始）件物品的重量为 w_i，价值为 r_i。在总重量不超过背包承载上限 W 的情况下，能够装入背包的最大价值是多少?
当然，我们可以使用枚举法，将所有的情况列举出来，每个物品有 2 种情况（放入背包或者不放入），所以最多有 2^n 种情况。枚举法的缺点是没有找到每种情况之间的关系，动态规划则可以通过利用历史信息避免重复枚举。

01 背包问题也可以写成如下的整数线性规划 (Integer Linear Programming)：

$$\begin{aligned} \max \quad & z = \sum_{i=1}^{n} r_i m_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i m_i \le W \\ & m_1, \cdots, m_n \in \{0, 1\} \end{aligned}$$

m_i 表示第 i 个物品是否装入背包。

完全背包问题：完全背包问题，基本设定与 01 背包相同，但是每个物品可以有无穷个。
完全背包问题 (n-种物品, W-总重背包) 可以用如下整数规划表示:

$$\begin{aligned} \max \quad & z = \sum_{i=1}^{n} r_i m_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i m_i \le W \\ & m_1, \cdots, m_n \in \mathbb{Z}_+ \cup \{0\} \end{aligned}$$

在最差情况下，求解整数线性规划需要指数时间。

对于 01 背包问题，我们有如下动态规划解法：

- 确定状态和保存状态的变量：先说状态，如何才能描述一个问题局面？只要给几个物品和一个背包的容量限制，就形成了一个背包问题。所以状态有两个，就是「背包的容量」和「可选择的物品」。在 0/1 背包问题中，每个阶段的状态通常包括以下信息：

  1. 当前考虑的物品（例如，物品 i）。
  2. 在该阶段背包的剩余容量（还可以添加多少重量到背包中）。

  因此，状态可以用一对变量 (i, w) 表示，其中 i 是当前物品的索引，w 是该阶段背包的剩余容量。

- 确定决策：决策指的是每个状态可以做出的选择或决策。在 0/1 背包问题中，每个阶段有两种选择：

  1. 选择当前物品并将其添加到背包中，前提是其重量不超过剩余容量。
  2. 拒绝当前物品，继续下一个物品，不将其添加到背包中。

- 递归方程：我们用 $f(i, w)$ 表示**前** i 个物品，放入最大承重 w 的背包最大价值。我们有边界条件 $f(i, 0) = 0, i = 0, 1, ..., n.$ 那么，根据上述分析，对于 $n \ge i > 1, W \ge w \ge 0$, 我们有如下递归方程：

  $$f(i, w) = \max\{f(i-1, w), f(i-1, w-w_i) + r_i, (w \ge w_i)\}$$

  其中，$f(i-1, w)$ 表示不放入物品 i 的价值；$f(i-1, w-w_i) + r_i$ 表示放入 i 的价值。

使用动态规划求解 01 背包问题的复杂度降至 $\mathcal{O}(nW)$.

注 8.1 *背包问题的判定形式（即是否能找到放入方式，使得不超过承重 W 的情况下达到价值 V?）是 NP-complete 问题。为什么是 NP 问题？首先，验证一个解是否满足判定条件，仅需要给定的放入方式 P, 进行求和判定 $\sum_{i\in P} w_i \le W, \sum_{i\in P} v_i \ge V$. 我们输入的是两个数组，分别表示每个物品的价值和重量，在计算机中表示他们的比特数和 n 线性相关。对于另外的参数，即总重量 W，需要 $m = \log W$ 的位数来表示. 因此，m 才是输入的规模。对于求和判定 $\sum_{i\in P} w_i \le W, \sum_{i\in P} v_i \ge V$, 需要的计算复杂度与 m, n 均线性，但是动态规划算法的复杂度 $O(n \cdot W) = O(n2^m)$, 并非真正的多项式时间。*

完全背包问题 我们用 $f(i,w)$ 表示前 i 种物品，放入最大承重 w 的背包最大价值。01 背包只有两种情况即取 0 件和取 1 件，而这里是取 0 件、1 件、2 件... 直到超过限重（$k > w/w_i$）

1. 状态 (i,w)，$i = 1,2,\cdots,n, w = 1,...,W$ 表示前 i 种物品放入承重 w 的背包的情况.
2. 第 i 种物品，放入承重 w 的背包中，决策的可能情况为 $k = 0,1,\cdots,\lfloor \frac{w}{w_i} \rfloor$.

我们有边界条件：$f(i,0) = 0$, $i = 0,1,...,n$, 递归方程如下

$$f(i,w) = \max_{k\,=\,0,\,1,\,\cdots,\,\lfloor \frac{w}{w_i} \rfloor} \{r_i k + f(i-1, w - w_i k)\}, \quad i = 1,\cdots,n, 1 \le w \le W.$$

Example 8.2 *计算容量为 7 的 01 或者完全背包问题的最大价值。*

Item i	1	2	3
w_i	2	3	1
r_i	31	47	15

1.5 设备更新模型 (Equipment Replacement Model)

假设我们考虑的是一个跨度为 n 年的设备更新问题。每年初我们需要决定是否保留当前设备再使用一年或者更换一个新的设备。令 $r(t)$, $c(t)$, $s(t)$ 分别表示一台 t-年龄设备的年营业收入，年运营成本及其残余价值。另外，在规划期内的任何一年购置一台新设备的成本是 I. 设某公司现有一台 3 年龄的设备，需制定一个未来 4 年 ($n = 5$) 的设备更新最优策略。该公司还规定 6 年龄的设备必须得以更换。一台新设备的成本是 \$100,000. 下表给出的是设备更新问题的相关数据，其中 t 是机器年龄，$r(t), c(t), s(t)$ 分别表示 t 年龄机器的年营业收入，年运营成本及残余价值。

表 8.1: 设备更新问题数据表

t	$r(t)$	$c(t)$	$s(t)$
0	20,000	200	–
1	19,000	600	80,000
2	18,500	1,200	60,000
3	17,200	1,500	50,000
4	15,500	1,700	30,000
5	14,000	1,800	10,000
6	12,200	2,200	5,000

- 该问题中，实际年份 i 和机器年龄 $t = 1,\cdots,6$，可以描述出机器的状态以及带来的收益. 所以用 (i,t) 表示状态

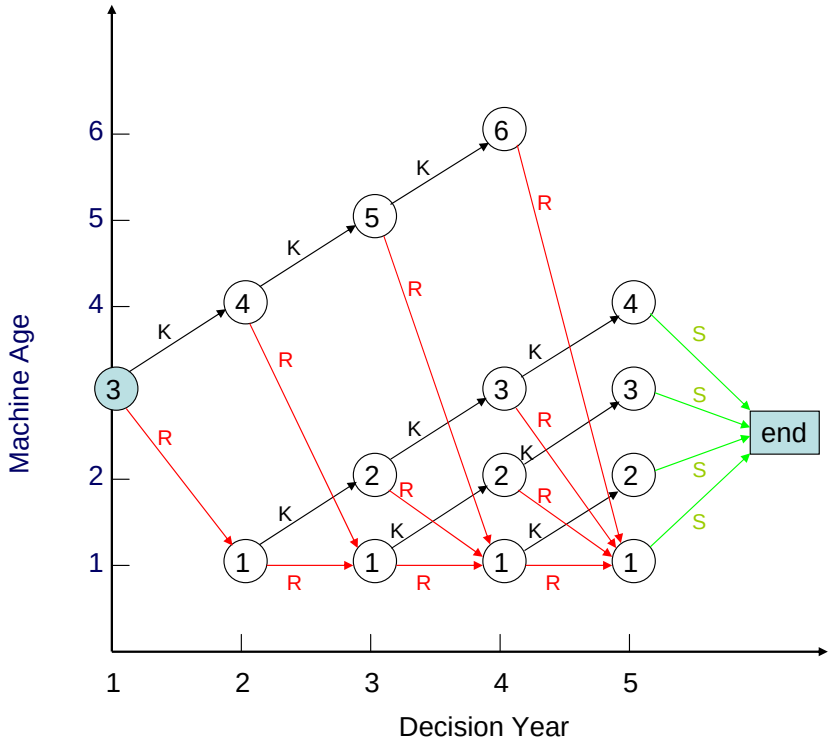


图 8.3: 设备更新模型图示

- 第 i 年的决策为在该年份开头，要么保留，要么更换机器.
- 递归方程: 用 $f_i(t)$ 表示年份 $i, i+1, \ldots, n$ 中，给定在第 i 年开始设备年龄为 t 的机器，所带来的最大收益。

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), t \le 6 & \text{if Keeping} \\ r(0) - c(0) + s(t) - I + f_{i+1}(1), & \text{if Replacing} \end{cases}$$
$$f_{n+1}(t) = s(t)$$

与之前的背包问题所不同的是，设备更新问题中，我们的递归边界条件为 $f_{n+1}(t) = s(t)$, 即从最后一项向前递归这通常称为**后向归纳法 (backward induction)**，而之前的背包问题为**前向递归 (forward induction)**.

作业 8.1 *设现有一台 2 年龄的设备，另规定 5 年龄的设备必须更换。在规划期购置新设备的成本分别是*

$$(p_1, p_2, p_3, p_4, p_5) = (100, 105, 110, 115, 120).$$

试构建表格 8.3中设备更新的动态规划模型并求其最优更新策略。

表 8.2: 五年期设备更新

设备年龄 t	残余价值 v_t	运行费用 c_t
0	-	30
1	50	40
2	25	50
3	10	75
4	5	90
5	2	-

表 8.3: 设备更新价值表格

作业 8.2 使用动态规划求解下面的问题。

1. - 问题 *1*: 如图 *8.4* 有一个移动机器人，可以在二维矩阵平面中移动，其移动的起点位于左上角，每次移动只能向右或者向下，其移动的终点是右下角，在这个矩阵形的平面中每个位置都有一个数值，代表机器人在经过这个区域时需要付出的代价。我们的目标就是，在这个平面中，找一条代价最小的路径，并且给出最小代价路径。

   - 问题 *2*: 若机器人从左上角出发，只能向右或者向下移动，起点坐标设为 $(0,\ 0)$，需要达到右下角，坐标设为 $(M,\ N)$。请问有多少种不同的路径（无需考虑每格代价）？

2	3	1
1	9	1
6	4	2

图 8.4: 机器人坐标代价表

2. 已知，维度为 $p \times q$ 和 $q \times r$ 的两个矩阵乘法的复杂度为 pqr（忽略常数）。若计算 *3* 个矩阵乘法 ABC，两种计算方法 $(AB)C$ 和 $A(BC)$ 的计算复杂度是不相同的。问题：给定 m 个矩阵 $A_1, A_2, \ldots, A_m$，其中矩阵 A_i 的维度为 $p_{i-1} \times p_i$。你的任务是计算这 m 个矩阵连乘的最优计算顺序，以最小化乘法运算的总次数，写出动态规划算法的递推式。

MATH 3011 运筹学 **秋季 2024**

Lecture 9: 非线性规划基础

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

本章开始，我们考虑更一般的非线性规划问题：

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in S \subset \mathbb{R}^n. \end{aligned} \tag{9.1}$$

在此节中，目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，S 是决策变量 $\mathbf{x}$ 的可取值之集合，称为问题的可行域 (feasible region).

最优化问题从属性上可以分为两大类：一类是具有连续变量的问题，另一类是离散变量的问题（即组合优化问题）。

非线性规划属于连续型最优化问题的范畴，通常可行域 S 可由一组方程来描述，即

$$S = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \geq 0, i = 1, \cdots, m; h_j(\mathbf{x}) = 0, j = 1, \cdots, \ell\}.$$

故，问题(9.1)可以写成如下形式：

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \geq 0, i = 1, \cdots, m \\ & h_j(\mathbf{x}) = 0, j = 1, \cdots, \ell \end{aligned} \tag{9.2}$$

这里，$f(\mathbf{x})$, $g_i(\mathbf{x})$, $h_j(\mathbf{x})$ 都是 $\mathbb{R}^n$ 上、实值、确定的函数，且至少有一个是非线性的。

- 如果求解在约束集合 S 上目标函数 $f(x)$ 的最大值, 则问题 (9.1) 的 “min” 应相应地替换为 “max”.
- 通常来说，我们不考虑严格不等式约束, 如 $g_i(\mathbf{x}) > 0$。因为此时可能没有最优解。
- 为了叙述简便，问题 (9.1) 中 x 为 $\mathbb{R}^n$ 空间中的向量. 实际上，根据具体应用和需求，x 还可以是矩阵、多维数组或张量等.

1.1 最优化问题的类型

- 最优解只有少量非零元素的问题称为稀疏优化；

- 最优解是低秩矩阵的问题称为低秩矩阵优化.
- 此外还有几何优化、二次锥规划、张量优化、鲁棒优化、全局优化、组合优化、网络规划、随机优化、动态规划、带微分方程约束优化、微分流形约束优化、分布式优化等.
- 就具体应用而言，问题 (9.1) 可涵盖统计学习、压缩感知、最优运输、信号处理、图像处理、机器学习、强化学习、模式识别、金融工程、电力系统等领域的优化模型.

Example 9.1 最小二乘法 *(Legendre, 1805)* 最小二乘法是一种标准的回归分析方法，用于在一组数据点中找到最佳拟合直线或曲线。其目标是最小化观测值与估计值之间差值的平方和。在统计学、工程学、经济学和数据科学等领域中，最小二乘法被广泛应用

基本原理: 给定数据点 $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$，最小二乘法寻求一个函数 $f(x)$（通常为线性），使得 ** 残差平方和 ** 最小。对于每个数据点的残差定义为：

$$\text{残差}_i = y_i - f(x_i)$$

** 目标函数 ** 为：

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

线性最小二乘法（简单线性回归）对于线性回归，我们假设模型为：

$$y = \beta_0 + \beta_1 x + \epsilon$$

其中，β_0 为截距，β_1 为斜率，ϵ 为误差项。

优化问题为：

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2$$

Example 9.2 投资组合优化（*Harry Max Markowitz* 马科维茨）

- r_i, 随机变量, 股票的回报率 i
- x_i, 投资于股票的相对金额 i
- 回报: $r = r_1 x_1 + r_2 x_2 + \ldots + r_n x_n$
- 期望回报: $R = E(r) = \sum E(r_i) x_i = \sum \mu_i x_i$
- 风险: $V = Var(r) = \sum_{i,j} \sigma_{ij} x_i x_j = x^\top \Sigma x$

$$\begin{aligned} \min & \frac{1}{2} x^\top \Sigma x, \\ \text{s.t.} & \sum \mu_i x_i \geq r_0 \\ & \sum x_i = 1, \\ & x_i \geq 0 \end{aligned} \qquad \begin{aligned} \min & \quad \text{risk measure}, \\ \text{s.t.} & \sum \mu_i x_i \geq r_0 \\ & \sum x_i = 1, \\ & x_i \geq 0 \end{aligned}$$

1.2 最优化问题的应用

数学建模很容易给出应用问题不同的模型，可以对应性质很不相同的问题，其求解难度和需要的算法也将差别很大. 在投资组合优化中，人们希望通过寻求最优的投资组合以降低风险、提高收益.

- 这时决策变量 x_i 表示在第 i 项资产上的投资额，向量 $x \in \mathbb{R}^n$ 表示整体的投资分配.
- 约束条件可能为总资金数、每项资产的最大（最小）投资额、最低收益等.
- 目标函数通常是某种风险度量.
- 如果是极小化收益的方差，则该问题是典型的二次规划.
- 如果极小化风险价值 (value at risk) 函数，则该问题是混合整数规划
- 如果极小化条件风险价值（conditional value at risk) 函数，则该问题是非光滑优化，也可以进一步化成线性规划.

1.3 最优解

满足约束条件 $\mathbf{x} \in S$ 的 $\mathbf{x}$ 称为问题的可行解 (feasible solution), 如果可行解 $\mathbf{x}^* \in S$ 进一步满足

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}), \quad \forall \mathbf{x} \in S. \tag{9.3}$$

则称 $\mathbf{x}^*$ 为问题(9.1)的**全局最优解** (global optimal solution), 对应的函数值叫做**全局最优值**。另外，在包含可行解 $\mathbf{x}^* \in S$ 的适当邻域 $U(\mathbf{x}^*)$ 里，成立

$$f(\mathbf{x}^*) \leqslant f(\mathbf{x}), \forall \mathbf{x} \in S \cap U(\mathbf{x}^*). \tag{9.4}$$

此时称 $\mathbf{x}^*$ 为问题(9.1)的**局部最优解** (local optimal solution)，对应的函数值叫做**局部最优值**。

解的存在唯一性

- 在数学分析课程中, 我们学习过 Weierstrass 定理, 即定义在紧集上的连续函数一定存在最大 (最小) 值点. 下面的推广的 Weierstrass 定理可以保证解存在性。

- 对于一般的凸函数, 其最优解可能不唯一, 比如 $f(x)=\max\{x,0\}$, 任意 $x\le 0$ 都是 $f(x)$ 的最优解.

Theorem 9.1 (推广的 Weierstrass 定理) *若函数 $f:\mathcal{X}\to(-\infty,+\infty]$ 适当[1]且闭, 且以下条件中任意一个成立:*

1. $\mathrm{dom} f=\{x\in S:f(x)<+\infty\}$ *是有界的;*
2. *存在一个常数 $\bar{\gamma}$ 使得下水平集*
$$C_{\bar{\gamma}}=\{x\in S:f(x)\le\bar{\gamma}\}$$
是非空且有界的;
3. *f 是强制的, 即对于任一满足 $\|x^k\|\to+\infty$ 的点列 $\{x^k\}\subset\mathcal{X}$, 都有*
$$\lim_{k\to\infty}f(x^k)=+\infty,$$

则函数 f 的最小值点集 $\{x\in\mathcal{X}\mid f(x)\le f(y),\ \forall y\in\mathcal{X}\}$ 非空且紧.

Lemma 9.1 *设 S 是 $\mathbb{R}^n$ 的一个非空, 紧且凸的子集, 如果 $f:S\to(-\infty,+\infty]$ 是强凸函数[2]且最小值存在, 那么存在唯一的 x^* 满足*
$$f(x^*)<f(x),\quad \forall x\in S\backslash\{x^*\}.$$

Definition 9.1 最优性条件：问题的最优解所满足的必要或者充分条件。

最优性条件将为各种求解算法的设计、分析提供必不可少的理论基础。

2 无约束最优条件

我们先从无约束优化问题开始讨论。

Definition 9.2 (下降方向 Descent direction) *设 $f(\mathbf{x})$ 是 $\mathbb{R}^n$ 上的实函数, $\bar{\mathbf{x}}\in\mathbb{R}^n$, $\mathbf{d}$ 是非零向量。若存在 $\delta>0$ 使得:*
$$f(\bar{\mathbf{x}}+\lambda\mathbf{d})<f(\bar{\mathbf{x}}),\forall\lambda\in(0,\delta) \tag{9.5}$$
则称 $\mathbf{d}$ 为函数 $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处的下降方向。记下降方向集合为 $\Omega(\bar{\mathbf{x}},f)$.

[1]适当函数：如果存在 $x\in\mathcal{X}$ 使得 $f(x)<+\infty$，并且对任意的 $x\in\mathcal{X}$，都有 $f(x)>-\infty$，那么称函数 f 关于集合 $\mathcal{X}$ 是适当的.

[2]强凸函数：若存在常数 $m>0$，使得
$$g(x)=f(x)-\frac{m}{2}\|x\|^2$$
为凸函数，则称 $f(x)$ 为**强凸函数**，其中 m 为**强凸参数**. 为了方便我们也称 $f(x)$ 为 m-强凸函数.

下降方向集的子集

如果 $f(\mathbf{x})$ 是可微函数，且 $\nabla f(\bar{\mathbf{x}})^T\mathbf{d} < 0$. 显然，此时 $\mathbf{d}$ 为 $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处的下降方向。记这样的方向集合为

$$D(\bar{\mathbf{x}}, f) = \{\mathbf{d} \mid \nabla f(\bar{\mathbf{x}})^T\mathbf{d} < 0\} \subset \{\text{下降方向}\}.$$

- 一阶最优性条件是利用梯度 (一阶) 信息来判断给定点的最优性.
- 由下降方向的定义, 在局部最优点处不能有下降方向.

Theorem 9.2 (一阶必要条件) 假设 f 在全空间 $\mathbb{R}^n$ 可微. 如果 x^* 是一个局部极小点, 那么

$$\nabla f(x^*) = 0.$$

Proof:

任取 $v \in \mathbb{R}^n$, 考虑 f 在点 $x = x^*$ 处的泰勒展开

$$f(x^* + tv) = f(x^*) + tv^{\mathrm{T}}\nabla f(x^*) + o(t),$$

整理得

$$\frac{f(x^* + tv) - f(x^*)}{t} = v^{\mathrm{T}}\nabla f(x^*) + o(1).$$

根据 x^* 的最优性, 在上式中分别对 t 取点 0 处的左, 右极限可知

$$\lim_{t\to 0^+} \frac{f(x^* + tv) - f(x^*)}{t} = v^{\mathrm{T}}\nabla f(x^*) \geq 0,$$
$$\lim_{t\to 0^-} \frac{f(x^* + tv) - f(x^*)}{t} = v^{\mathrm{T}}\nabla f(x^*) \leq 0,$$

即对任意的 v 有 $v^{\mathrm{T}}\nabla f(x^*) = 0$, 由 v 的任意性知 $\nabla f(x^*) = 0$. ■

- 在没有额外假设时, 如果一阶必要条件满足, 我们仍然不能确定当前点是否是一个局部极小点. 例如 $f(x) = x^3$, 在 $x = 0$ 处不是局部极小点。
- 假设 f 在点 x 的一个开邻域内是二阶连续可微的. 类似于一阶必要条件的推导, 可以借助当前点处的二阶泰勒展开来逼近该函数在该点附近的取值情况, 从而来判断最优性.
- 当一阶必要条件满足时, $\nabla f(x) = 0$, 那么考虑二阶展开为

  $$f(x + d) = f(x) + \frac{1}{2}d^{\mathrm{T}}\nabla^2 f(x)d + o(\|d\|^2).$$

  由此, 我们可以导出二阶最优性条件.

Theorem 9.3 (二阶最优性条件) *必要条件: 若 x^* 是 f 的一个局部极小点, 则 $\nabla f(x^*)=0, \nabla^2 f(x^*)\succeq 0$.*

充分条件: 若 $\nabla f(x^*)=0, \nabla^2 f(x^*)\succ 0$, 则 x^* 是 f 的一个局部极小点.

Proof:

- **必要性**: 若 $\nabla^2 f(x^*)$ 有负的特征值 $\lambda_- < 0$, 设 $\nabla^2 f(x^*)d = \lambda_- d$, 则
$$\frac{f(x^*+d)-f(x^*)}{\|d\|^2} = \frac{1}{2}\frac{d^{\mathrm{T}}}{\|d\|}\nabla^2 f(x^*)\frac{d}{\|d\|} + o(1) = \frac{1}{2}\lambda_- + o(1).$$
当 $\|d\|$ 充分小时, $f(x^*+d) < f(x^*)$, 这和点 x^* 的最优性矛盾.

- **充分性**: 由 $\nabla f(x^*)=0$ 时的二阶展开,
$$\frac{f(x^*+d)-f(x^*)}{\|d\|^2} = \frac{\frac{1}{2}d^{\mathrm{T}}\nabla^2 f(x^*)d + o(\|d\|^2)}{\|d\|^2} \geq \frac{1}{2}\lambda_{\min} + o(1).$$
这里 $\lambda_{\min}$ 是 $\nabla^2 f(x^*)$ 的最小特征根。当 $\|d\|$ 充分小时有 $f(x^*+d)\geq f(x^*)$, 即二阶充分条件成立.

■

注:

- 设点 $\bar{x}$ 满足一阶最优性条件 (即 $\nabla f(\bar{x})=0$), 且该点处的海瑟矩阵 $\nabla^2 f(\bar{x})$ 不是半正定的, 那么 $\bar{x}$ 不是一个局部极小点.
- 进一步地, 如果海瑟矩阵 $\nabla^2 f(\bar{x})$ 既有正特征值又有负特征值, 我们称稳定点 $\bar{x}$ 为**鞍点** (saddle point).
- 事实上, 记 d_1, d_2 为其正负特征值对应的特征向量, 那么对于任意充分小的 $t>0$, 我们都有 $f(\bar{x}+td_1) > f(\bar{x})$ 且 $f(\bar{x}+td_2) < f(\bar{x})$.
- 注意, 二阶最优性条件给出的仍然是关于局部最优性的判断. 对于给定点的全局最优性判断, 我们还需要借助实际问题的性质, 比如目标函数是凸的、非负目标函数值为 0 等.

Example 9.3 *$f(x)=x_1^3+x_2^2$, 我们有 $\nabla f(x)=(3x_1^2, 2x_2)^\top$, $\nabla^2 f(x)=\begin{pmatrix}6x_1 & 0\\ 0 & 2\end{pmatrix}$. $x=(0,0)^\top$ 满足二阶必要性条件, 但并非局部极小。这是因为 $f(-\epsilon,0)=-3\epsilon^3<0$.*

Theorem 9.4 (一阶充要条件) *假设 f 是凸函数, 且在全空间 $\mathbb{R}^n$ 可微. 如果 x^* 是一个全局极小点, 当且仅当*
$$\nabla f(x^*)=0.$$

Proof: 必要性：根据一阶必要条件，以及凸函数的任意局部极小为全局极小可得。
充分性：根据凸函数的一阶判定条件，即任意 x, y 有

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x).$$

令 $x = x^*$ 得证。 ■

Example 9.4 线性最小二乘*:*

$$\min_{x \in \mathbb{R}^n} \quad f(x) \overset{\text{def}}{=\!=} \frac{1}{2}\|b - Ax\|_2^2,$$

其中 $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$. 由 f 可微且凸知

$$x^* \text{ 为一个全局最优解} \Leftrightarrow \nabla f(x^*) = A^{\mathrm{T}}(Ax^* - b) = 0.$$

3 约束问题的最优性条件

3.1 可行方向，切锥，线性可行锥方向

若考虑约束问题，需要限制下降方向的选择，不能走出可行域，这里我们先定义可行方向。

Definition 9.3 (可行方向 Feasible direction) 设 $\bar{\mathbf{x}} \in S$, $\mathbf{d} \in \mathbb{R}^n$ 是非零向量。若存在 $\delta > 0$ 使得:

$$\bar{\mathbf{x}} + \lambda \mathbf{d} \in S, \forall \lambda \in (0, \delta) \tag{9.6}$$

则称 $\mathbf{d}$ 是 S 在 $\bar{\mathbf{x}}$ 处的可行方向。记 S 在 $\bar{\mathbf{x}}$ 处的所有可行方向的集合为 $F(\bar{\mathbf{x}}, S)$.

Theorem 9.5 对于问题 $\min\{f(\mathbf{x}) \mid \mathbf{x} \in S\}$, 设 $\bar{\mathbf{x}} \in S$, $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处可微。如果 $\bar{\mathbf{x}}$ 是问题的局部最优解，则可行方向集中无下降方向，即

$$F(\bar{\mathbf{x}}, S) \cap \Omega(\bar{\mathbf{x}}, f) = \emptyset. \tag{9.7}$$

Proof: 反证即可。若 $F(\bar{\mathbf{x}}, S) \cap \Omega(\bar{\mathbf{x}}, f)$ 非空，则 $\mathbf{x}$ 处存在可行下降方向，故不是局部最优。 ■

上面定义的下降方向，以及复杂约束 S 的可行方向，均比较难以刻画。**我们的目标是寻找易于用数学公式表述的定义**。首先，我们先将下降方向的子集刻画如下。

Definition 9.4 (下降方向集的子集) 如果 $f(\mathbf{x})$ 是可微函数，且 $\nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0$. 显然，此处的 $\mathbf{d}$ 为 $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处的下降方向。记这样的方向集合为

$$D(\bar{\mathbf{x}}, f) = \{\mathbf{d} \mid \nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0\}.$$

Proposition 9.1 对于问题(9.1), 设 $\bar{\mathbf{x}} \in S$, $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处可微。如果 $\bar{\mathbf{x}}$ 是问题的局部最优解，则下降方向子集 $D(\bar{\mathbf{x}}, f)$ 中无可行方向，即

$$F(\bar{\mathbf{x}}, S) \cap D(\bar{\mathbf{x}}, f) = \emptyset. \tag{9.8}$$

对于不等式约束，我们同样可以定义其可行方向的子集。若在 x 处，某个不等式约束 $g_j(x) < 0$, 此时，根据可微条件，无需考虑该不等式约束。所以我们仅考虑取到等号的不等式约束。回顾**积极集**的定义: $\mathcal{A}(x) = \{i = 1, 2, \ldots, m : g_i(x) = 0\} \cup \{j = 1, 2, ..., \ell\}$, 即所有等式下标，和不等式约束在 x 处取到等号条件成立时的下标集合。

Definition 9.5 (不等式可行方向的子集)

$$F_{\bar{\mathbf{x}},g} = \{\mathbf{d} \mid \nabla g_i(\bar{\mathbf{x}})^T \mathbf{d} > 0, i \in \mathcal{A}(\bar{\mathbf{x}})\}.$$

上述定理 9.1对于多面体集可以适用。但是，对于有些等式约束，例如二维圆环 $x_1^2 + x_2^2 = 1$ 上, 可行方向不存在（想一想，为什么?)。对于一般的非线性等式约束确定的可行域 C, 我们定义在点 $x \in C$ 处的**切锥**，为从在 C 中收敛到 x 的序列获得的极限方向的集合。

Definition 9.6 (切锥) 我们称方向 $d \in \mathbb{R}^n$ 属于可行点 $x \in S$ 处的切锥，如果存在序列 $\{x_i\} \subset S$, 和实数序列 $\tau_i \searrow 0$, 使得

$$\frac{x_i - x}{\tau_i} \to d, \ i \to \infty.$$

记其切锥集合为

$$T(x, S) := \{d : \exists \tau_i \searrow 0, \{x_i\} \subset S, x_i \to x, \quad \text{s.t.} \quad \frac{x_i - x}{\tau_i} \to d\}.$$

根据切锥的定义，显然对于任意可行方向 $d \in F(\bar{\mathbf{x}}, S)$, 必有 $d \in T(\bar{\mathbf{x}}, S)$. 故切锥相比于可行方向集合更大，

$$F(\bar{\mathbf{x}}, S) \subset T(\bar{\mathbf{x}}, S).$$

Theorem 9.6 对于问题(9.1), 设 $\bar{\mathbf{x}} \in S$, $f(\mathbf{x})$ 在 $\bar{\mathbf{x}}$ 处可微。如果 $\bar{\mathbf{x}}$ 是局部最优解，则

$$\nabla f(\bar{\mathbf{x}})^T d \geq 0, \quad \forall d \in T(\bar{\mathbf{x}}, S). \tag{9.9}$$

Proof: 若 $d \in T(\bar{\mathbf{x}}, S)$, 则存在序列 $\{x_k\} \subset S$, $\tau_k \searrow 0$, 且 $x_k \to \bar{\mathbf{x}}$ 使得 $d_k = \frac{x_k - \bar{\mathbf{x}}}{\tau_k} \to d$, 令 $x_k = \bar{\mathbf{x}} + \tau_k d_k$.

$$\nabla f(\bar{\mathbf{x}})^T d = \lim_{k\to\infty} \nabla f(\bar{\mathbf{x}})^T d_k = \lim_{k\to\infty} \frac{f(\bar{\mathbf{x}}) + \tau_k \nabla f(\bar{\mathbf{x}})^T d_k + o(\tau_k) - f(\bar{\mathbf{x}})}{\tau_k} = \lim_{k\to\infty} \frac{f(x_k) - f(\bar{\mathbf{x}})}{\tau_k} \geq 0.$$

■

Corollary 9.1 问题(9.1)的局部最优点 $\bar{\mathbf{x}}$ 满足

$$T(\bar{\mathbf{x}}, S) \cap D(\bar{\mathbf{x}}, f) = \emptyset. \tag{9.10}$$

上述极限定义的切锥仍然不易表达。针对等式和不等式定义的约束，我们分别有下面的结论。

- 对于等式约束 $\mathcal{E} := \{h_i(\bar{\mathbf{x}}) = 0, i = 1, ..., \ell\}.$，若 $d \in T(\bar{\mathbf{x}}, h)$, 这里 $T(\bar{\mathbf{x}}, h)$ 表示 $h_i(\bar{\mathbf{x}}) = 0, i = 1, ..., \ell$ 的切锥，则存在序列 $\{x_k\} \subset \mathcal{E}$, $\tau_k \searrow 0$, 且 $x_k \to \bar{\mathbf{x}}$ 使得 $\frac{x_k - \bar{\mathbf{x}}}{\tau_k} \to d$. 令 $d_k = \frac{x_k - \bar{\mathbf{x}}}{\tau_k}$, 则

$$\nabla h_i(\bar{\mathbf{x}})^T d = h_i(\bar{\mathbf{x}}; d) = \lim_{k \to \infty} \frac{h_i(\bar{\mathbf{x}} + \tau_k d_k) - h_i(\bar{\mathbf{x}})}{\tau_k} = 0. \tag{9.11}$$

- 同理，对于不等式积极约束集（active set），记 $\mathcal{A}(\bar{\mathbf{x}}) = \{i \mid g_i(\bar{\mathbf{x}}) = 0, i \in \{1, \cdots, m\}\}$,

$$\nabla g_i(\bar{\mathbf{x}})^T d = g_i(\bar{\mathbf{x}}; d) = \lim_{k \to \infty} \frac{g_i(\bar{\mathbf{x}} + \tau_k d_k) - g_i(\bar{\mathbf{x}})}{\tau_k} \geq 0, i \in \mathcal{A}(\bar{\mathbf{x}}). \tag{9.12}$$

另外，我们由(9.10)可知，

$$T(\bar{\mathbf{x}}, g) \cap T(\bar{\mathbf{x}}, h) \cap D(\bar{\mathbf{x}}, f) = \emptyset. \tag{9.13}$$

由 (9.11)和(9.12)，进一步定义如下**线性可行锥方向**。其优点是可以通过计算约束函数的梯度得到。

Definition 9.7 (等式的线性可行锥方向) $L(\bar{\mathbf{x}}, h) := \{\mathbf{d} \mid \nabla h_j(\bar{\mathbf{x}})^T \mathbf{d} = 0, j = 1, \cdots, \ell\}$.

Definition 9.8 (不等式的线性可行锥方向) $L(\bar{\mathbf{x}}, g) := \{\mathbf{d} \mid \nabla g_i(\bar{\mathbf{x}})^T \mathbf{d} \geq 0, i \in \mathcal{A}(\bar{\mathbf{x}})\}$.

根据线性可行锥的定义，我们有

$$F_{\bar{\mathbf{x}}, g} \subset F(\bar{\mathbf{x}}, g) \subset T(\bar{\mathbf{x}}, g) \subset L(\bar{\mathbf{x}}, g), \tag{9.14}$$

最后一个子集关系由(9.12)得到。

$$F(\bar{\mathbf{x}}, h) \subset T(\bar{\mathbf{x}}, h) \subset L(\bar{\mathbf{x}}, h), \tag{9.15}$$

最后一步由(9.11)得到。下图 9.1描述了几种定义.

假设切锥与线性可行锥方向相等，则我们可以得到易于计算的最优性条件：

$$L(\bar{\mathbf{x}}, g) \cap L(\bar{\mathbf{x}}, h) \cap D(\bar{\mathbf{x}}, f) = \emptyset.$$

然而，**线性可行锥方向与约束的表示形式有关**。同一个可行域，在不同的数学表述方式下，线性可行锥方向可能不同。

Example 9.5 令 $S = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1, (x_1 - 2)^2 + x_2^2 \leq 1\}$. 则 $S = \{(1, 0)^T\}$. 我们有 $T(x, S) = \{(0, 0)^T\}$, 然而 $L(x, S) = \{(0, t) \mid t \in \mathbb{R}\}$。但是，如果我们考虑约束的一种最直接表示：$S = \{(1, 0)^T\}$, 那么 $T(x, S) = L(x, S)$.

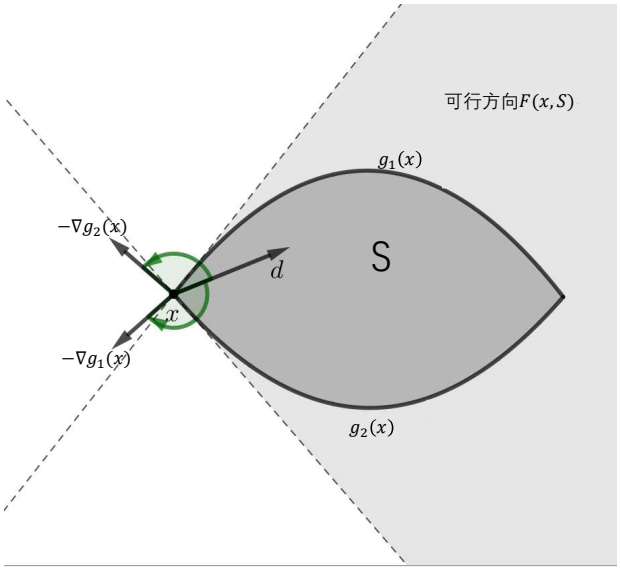


图 9.1: $\mathbb{R}^2$ 上的两个不等式约束集合，深灰色 S 是可行域，可行方向为浅灰色区域，不包括边界虚线。而线性可行锥方向是浅灰色区域包括虚线。

为此，我们引入以下**约束品性 (constraint qualification) 条件**: Mangasarian-Fromovitz Constraint Qualification (MFCQ) 条件, 以排除上面的例子中线性可行锥和切锥不相同的情况。

Definition 9.9 (MFCQ) *在问题(9.2)中, f 和 $g_i, i \in \mathcal{A}(\bar{\mathbf{x}})$ 在点 $\bar{\mathbf{x}}$ 可微, $g_i, i \notin \mathcal{A}(\bar{\mathbf{x}})$ 在点 $\bar{\mathbf{x}}$ 连续, 且 $\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$ 线性无关, $F_{\bar{\mathbf{x}},g} \cap L(\bar{\mathbf{x}}, h) = \{\mathbf{d} \neq 0 : \nabla g_i(\bar{\mathbf{x}})^\top \mathbf{d} > 0, \nabla h_j(\bar{\mathbf{x}})^\top \mathbf{d} = 0, i, j \in \mathcal{A}(\bar{\mathbf{x}})\}$ 非空, 则称该点处满足 MFCQ 约束品性条件。*

另一种更为常见的约束品性条件是线性无关约束品性。

Definition 9.10 (线性无关约束品性（LICQ）) *给定可行点 x 及相应的积极集 $\mathcal{A}(x)$. 如果积极集对应的约束函数的梯度, 即 $\nabla g_i(x)$, $i \in \mathcal{A}(x)$ 和 $\nabla h_j(x), j = 1, ..., \ell$ 所有向量是线性无关的, 则称**线性无关约束品性**（LICQ）在点 x 处成立.*

事实上，我们有 LICQ $\Longrightarrow$ MFCQ. 这个我们不去证明。

Lemma 9.2 (约束品性条件) *若在可行点 $\bar{\mathbf{x}} \in S$ 处满足 LICQ 或者 MFCQ, 则 $L(\bar{\mathbf{x}}, h) \cap L(\bar{\mathbf{x}}, g) = T(\bar{\mathbf{x}}, g) \cap T(\bar{\mathbf{x}}, h)$.*

该引理我们也略去证明。因此，在 LICQ 或者 MFCQ 条件下，根据(9.13)我们有

$$L(\bar{\mathbf{x}}, g) \cap L(\bar{\mathbf{x}}, h) \cap D(\bar{\mathbf{x}}, f) = \emptyset. \tag{9.16}$$

3.2 Fritz-John 条件

Fritz-John 条件仅是判别某点是否为最优点的必要条件,而不是充分条件。它是为后面我们证明 Karush-Kuhn-Tucker (KKT) 条件做准备。

根据上一小节的准备工作，我们有下面的结论。

Lemma 9.3 设 $\bar{\mathbf{x}}$ 为问题(9.2)的局部最优解，f 和 $g_i, i \in \mathcal{A}(\bar{\mathbf{x}})$ 在点 $\bar{\mathbf{x}}$ 可微，$g_i, i \notin \mathcal{A}(\bar{\mathbf{x}})$ 在点 $\bar{\mathbf{x}}$ 连续，h_j 在点 $\bar{\mathbf{x}}$ 连续可微，且 $\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$ 线性无关，则

$$D(\bar{\mathbf{x}}, f) \cap F_{\bar{\mathbf{x}}, g} \cap L(\bar{\mathbf{x}}, h) = \emptyset. \tag{9.17}$$

Proof: 该引理证明分为两种情况。若点 $\bar{\mathbf{x}}$ 处满足 MFCQ，则根据(9.16)和 $F_{\bar{\mathbf{x}}, g} \subset L(\bar{\mathbf{x}}, g)$ 得出结论。若点 $\bar{\mathbf{x}}$ 不满足 MFCQ，则 $F_{\bar{\mathbf{x}}, g} \cap L(\bar{\mathbf{x}}, h) = \emptyset$, 得证。 ■

为了证明 Fritz-John 条件，我们有如下凸集分离定理。

Theorem 9.7 (凸集分离定理) 假设 C_1 和 C_2 是两个不相交的非空凸集，那么存在一个非零向量 w 和一个实数 b 使得对于所有 $x_1 \in \mathrm{CL}(C_1)$ 和 $x_2 \in \mathrm{CL}(C_2)$ 有:

$$w^T x_1 \geq b \quad \text{和} \quad w^T x_2 \leq b,$$

这里 $\mathrm{CL}(C_1)$ 表示 C_1 的闭包。这意味着超平面 $\{x : w^T x = b\}$ 将 $\mathrm{CL}(C_1)$ 和 $\mathrm{CL}(C_2)$ 分开。

Theorem 9.8 (Fritz-John 条件) 在问题(9.2)中，设 $\bar{\mathbf{x}}$ 为可行点，f 和 $g_i, i \in \mathcal{A}(\bar{\mathbf{x}})$ 在点 $\bar{\mathbf{x}}$ 可微，$g_i, i \notin \mathcal{A}(\bar{\mathbf{x}})$ 在点 $\bar{\mathbf{x}}$ 连续，h_j 在点 $\bar{\mathbf{x}}$ 连续可微。如果 $\bar{\mathbf{x}}$ 是局部最优解，则存在不全为零的数 $\lambda_0, \lambda_i, i \in \mathcal{A}(\bar{\mathbf{x}})$ 和 $\mu_j, j = 1, \cdots, \ell$ 使得

$$\lambda_0 \nabla f(\bar{\mathbf{x}}) - \sum_{i \in \mathcal{A}(\bar{\mathbf{x}})} \lambda_i \nabla g_i(\bar{\mathbf{x}}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0. \tag{9.18}$$

其中 $\lambda_0 \geq 0, \lambda_i \geq 0, i \in \mathcal{A}(\bar{\mathbf{x}})$.

Proof:

证明:(1) 如果 $\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$ 线性相关，则存在不全为零的数 $\mu_j, j = 1, \cdots, \ell$ 使得

$$\sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0.$$

这时可令 $\lambda_0 = 0, \lambda_i = 0, i \in \mathcal{A}(\bar{\mathbf{x}})$, 结论成立。

(2) 如果 $\{\nabla h_j(\bar{\mathbf{x}})\}_{j=1}^{\ell}$ 线性无关。利用 Lemma 9.3知 $D(\bar{\mathbf{x}}, f) \cap F_{\bar{\mathbf{x}},g} \cap L(\bar{\mathbf{x}}, h) = \emptyset$.
即不等式组

$$\begin{cases} \nabla f(\bar{\mathbf{x}})^T \mathbf{d} < 0 \\ \nabla g_i(\bar{\mathbf{x}})^T \mathbf{d} > 0, i \in \mathcal{A}(\bar{\mathbf{x}}) \\ \nabla h_j(\bar{\mathbf{x}})^T \mathbf{d} = 0, j = 1, \cdots, \ell \end{cases} \tag{9.19}$$

无解。

令 A 是以 $\{\nabla f(\bar{\mathbf{x}}), -\nabla g_i(\bar{\mathbf{x}}), i \in \mathcal{A}(\bar{\mathbf{x}})\}$ 为列组成的矩阵，B 是以 $\{-\nabla h_j(\bar{\mathbf{x}}), j = 1, \cdots, \ell\}$ 为列组成的矩阵。
于是得

$$\begin{cases} A^T \mathbf{d} < 0 \\ B^T \mathbf{d} = 0 \end{cases} \tag{9.20}$$

无解。

下证

$$\begin{cases} A\mathbf{p}_1 + B\mathbf{p}_2 = 0 \\ \mathbf{p}_1 \geq 0 \end{cases} \tag{9.21}$$

有解。

现定义

$$S_1 = \{\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \mid \mathbf{y}_1 = A^T \mathbf{d}, \mathbf{y}_2 = B^T \mathbf{d}, \mathbf{d} \in \mathbb{R}^n\},$$

$$S_2 = \{\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \mid \mathbf{y}_1 < \mathbf{0}, \mathbf{y}_2 = \mathbf{0}\}.$$

显然 S_1 和 S_2 为非空凸集，且 $S_1 \cap S_2 = \emptyset$.

由凸集分离定理知，对 $\forall \mathbf{d} \in \mathbb{R}^n, \forall \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \in S_2$, 存在非零向量 $\begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}$ 使得 $\mathbf{p}_1^T A^T \mathbf{d} + \mathbf{p}_2^T B^T \mathbf{d} \geq \mathbf{p}_1^T \mathbf{y}_1 + \mathbf{p}_2^T \mathbf{y}_2$.
首先令 $\mathbf{y}_2 = \mathbf{0}$, 由 $\mathbf{d}$ 的任意性（取 $\mathbf{d} = \mathbf{0}$）及 $\mathbf{y}_1 < \mathbf{0}$, $\Longrightarrow \mathbf{p}_1 \geq \mathbf{0}$.

再令 $\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \in \mathrm{CL}(S_2)$, $\Longrightarrow \mathbf{p}_1^T A^T \mathbf{d} + \mathbf{p}_2^T B^T \mathbf{d} \geq 0$.

最后取 $\mathbf{d} = -(A\mathbf{p}_1 + B\mathbf{p}_2)$, $\Longrightarrow A\mathbf{p}_1 + B\mathbf{p}_2 = \mathbf{0}$.

综上所述，即得(9.21)有解。

把 $\mathbf{p}_1$ 的分量记作 λ_0 和 $\lambda_i, i \in \mathcal{A}(\bar{\mathbf{x}})$, $\mathbf{p}_2$ 的分量记作 $\mu_j, j = 1, \cdots, \ell$. 立即得到

$$\lambda_0 \nabla f(\bar{\mathbf{x}}) - \sum_{i \in I(\bar{\mathbf{x}})} \lambda_i \nabla g_i(\bar{\mathbf{x}}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{\mathbf{x}}) = 0. \tag{9.22}$$

■

3.3 Karush-Kuhn-Tucker(KKT) 条件

1951 年，Harold W. Kuhn 和 Albert W. Tucker 在非线性规划领域提出了一个重要的定理，即 Kuhn-Tucker 定理。这个定理对具有不等式约束的优化问题建立了必要条件，为其最优解提供了一种方法。实际上，这一成果是在 W. Karush 于 1939 年首次提出的基础上进行的扩展，其都是我们学过的拉格朗日乘子法的推广。因此，KKT 条件的全称为 Karush-Kuhn-Tucker 条件。

3.3.1 一阶必要条件

定义 Lagrange 函数 $\mathcal{L}(\mathbf{x},\lambda,\mu)=f(\mathbf{x})-\sum\limits_{i=1}^{m}\lambda_i g_i(\mathbf{x})-\sum\limits_{j=1}^{\ell}\mu_j h_j(\mathbf{x})$.

Theorem 9.9 *若 $\bar{\mathbf{x}}$ 为问题局部最优解，且 $\bar{\mathbf{x}}$ 处满足 LICQ 条件。此时，一阶必要条件可表达为*

$$(\text{KKT})\begin{cases}\textbf{稳定性条件} & \nabla_{\mathbf{x}}\mathcal{L}(\bar{\mathbf{x}},\lambda,\mu)=0\\ \textbf{原始可行性条件} & g_i(\bar{\mathbf{x}})\geq 0, i=1,\cdots,m; h_j(\bar{\mathbf{x}})=0, j=1,\cdots,\ell.\\ \textbf{互补松弛条件} & \lambda_i g_i(\bar{\mathbf{x}})=0, i=1,\cdots,m\\ \textbf{对偶可行性条件} & \lambda_i\geq 0, i=1,\cdots,m\end{cases} \tag{9.23}$$

实际上，任何可以保证切锥和线性可行锥相等的条件，均可推出上述的 KKT 一阶条件。我们这里给出了利用 LICQ 条件的证明。

证明：利用 Fritz-John 条件，若 LICQ 成立，则 $\lambda_0\neq 0$. 这是因为如果 $\lambda_0=0$, 利用线性无关，可得 $\lambda_i=0,\forall i=1,2...,m,\mu_j=0,\forall j=1,2,...,\ell$. 矛盾。

(9.22)两边同除以 λ_0 得

$$\nabla f(\bar{\mathbf{x}})-\sum_{i\in I(\bar{\mathbf{x}})}\lambda_i\nabla g_i(\bar{\mathbf{x}})-\sum_{j=1}^{\ell}\mu_j\nabla h_j(\bar{\mathbf{x}})=0. \tag{9.24}$$

若下标 $i\notin\mathcal{A}(\bar{x})$, 那么 $g_i(\bar{\mathbf{x}})<0$, 因此我们可以令相应的 $\lambda_i=0$. 于是有稳定性条件：$\nabla_{\mathbf{x}}L(\bar{\mathbf{x}},\lambda,\mu)=0$, 以及互补松弛性。原始可行性和对偶可行性源自于 Fritz-John 条件结论以及 Lagrange 函数的定义。

作业 9.1 *对于问题(9.2), 若 $\bar{x}$ 是局部最优解，并且 $\bar{x}$ 满足 MFCQ 条件。证明如下结论：*

1. *对于 $\forall d\in T(\bar{x},S)$, 我们有 $\nabla f(\bar{x})^T d\geq 0$。*

2. 考虑下面的线性规划问题：

$$\max(-\nabla f(\bar{x}))^T d \tag{9.25}$$

$$\text{s.t.} \quad \nabla g_j(\bar{x})^T d \geq 0, j \in \mathcal{I}(\bar{x}) \tag{9.26}$$

$$\nabla h_i(\bar{x})^T d = 0, i = 1, 2, \ldots, \ell. \tag{9.27}$$

写出其对偶问题，并利用强对偶定理证明，在 *MFCQ* 条件下的 *KKT* 条件成立。

注：

- 称满足 KKT 条件的变量对 $(\bar{\mathbf{x}}, \lambda, \mu)$ 为 KKT 对.
- 称 $\bar{\mathbf{x}}$ 为 KKT 点.
- 如果局部最优点 $\bar{\mathbf{x}}$ 处 $T_{\bar{\mathbf{x}},S} \neq L(\bar{\mathbf{x}}, S)$, 那么 x^* 不一定是 KKT 点.
- KKT 条件只是必要的, KKT 点不一定是局部最优点. 也就是说，可以通过计算 KKT 条件，求出最优点的备选集合。

Example 9.6 注意，约束品性条件保证了 *KKT* 条件成立，否则，即使是凸问题，也未必满足 *KKT* 条件。

$$\min x, \quad \text{s.t.} \quad x^2 \leq 0.$$

该问题是凸问题，但是最优点 $x = 0$ 不满足 *KKT* 条件。

3.3.2 一阶充分条件

我们这里给出特殊的凸优化问题的充分最优性条件。即问题都可以写为

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x), \\ \text{s.t.} \quad & c_i(x) \leqslant 0, \quad i = 1, 2, \cdots, m, \\ & Ax = b, \end{aligned} \tag{9.28}$$

其中 $f(x)$ 为适当的凸函数, $\forall i, c_i(x)$ 是凸函数且 $\text{dom} c_i = \mathbb{R}^n$.

Definition 9.11 (相对内点) 集合 $\mathcal{D}$ 的相对内点集定义为

$$\text{relint}\mathcal{D} = \{x \in \mathcal{D} \mid \exists\ r > 0,\ \text{使得}\ B(x, r) \cap \text{affine}\mathcal{D} \subseteq \mathcal{D}\}.$$

相对内点是内点的推广, 若 $\mathcal{D}$ 本身的 "维数" 较低, 则 $\mathcal{D}$ 不可能有内点, 但如果在它的仿射包 $\text{affine}\mathcal{D} := \{x \mid x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k,\ x_1, x_2, \cdots, x_k \in \mathcal{D},\ \sum_{i=1}^k \theta_i = 1\}$ 中考虑, 则 $\mathcal{D}$ 可能有相对内点.

Example 9.7 考虑 $\mathbb{R}^3$ 中的一条线段 l, l 在 $\mathbb{R}^3$ 中无内点，但是存在相对内点。

Definition 9.12 (Slater 条件) 若对凸优化问题

$$\min_{x\in\mathcal{D}} \quad f(x), \quad \text{s.t.} \quad c_i(x)\leqslant 0,\ i=1,2,\cdots,m, \quad Ax=b,$$

存在 $x\in \text{relint}\mathcal{D}$ 满足

$$c_i(x)<0, \quad i=1,2,\cdots,m, \quad Ax=b,$$

即存在**相对内点满足严格不等式**条件。则称对此问题 Slater 约束品性满足. 该约束品性也称为 ***Slater* 条件**.

注 9.1 若某些不等式约束 $c_i(x)$ 也为仿射函数时，那么这些不等式无需满足严格不等式，等式即可。例如，线性规划问题无需验证 *Slater* 条件。

对于凸优化问题, 当 Slater 条件满足时, KKT 条件则变为局部最优解的充要条件 (根据凸性, 局部最优解也是全局最优解).

Theorem 9.10 对于问题(9.28), 若 *Slater* 条件成立，x^* 是全局最优点当且仅当 x^* 是 *KKT* 点。

Example 9.8

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \quad & \frac{1}{2}\|x-y\|_2^2,\\ s.t. \quad & Ax=b,\end{aligned}$$

其中 $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$ 以及 $y\in\mathbb{R}^n$ 为给定的矩阵和向量且 A 满秩.

- 拉格朗日函数: $L(x,\lambda)=\frac{1}{2}\|x-y\|^2+\lambda^{\mathrm{T}}(Ax-b)$.
- 由于仅存在线性等式约束，*Slater* 条件成立。故，x^* 为一个全局最优解当且仅当存在 $\lambda^*\in\mathbb{R}^m$ 使得

$$\begin{cases} x^*-y+A^{\mathrm{T}}\lambda^*=0,\\ \qquad\qquad Ax^*=b.\end{cases}$$

- 由上述 *KKT* 条件第一式, 等号左右两边同时左乘 A 可得

$$Ax^*-Ay+AA^{\mathrm{T}}\lambda=0\Rightarrow\lambda^*=(AA^{\mathrm{T}})^{-1}(Ay-b).$$

- 将 λ^* 代回 *KKT* 条件第一式可知

$$x^*=y-A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}(Ay-b).$$

因此点 y 到集合 $\{x\mid Ax=b\}$ 的投影为 $y-A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}(Ay-b)$.

3.4 二阶最优条件

再次考虑约束问题(9.2)，即

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \geq 0, i = 1, \cdots, m \\ & h_j(\mathbf{x}) = 0, j = 1, \cdots, \ell \end{aligned}$$

假设 x^* 是满足 KKT 条件的点, 记 S 为问题的可行域, 并且切锥与线性可行锥方向相等, 即 $T(x^*, S) = L(x^*, S)$, 这里 $L(x^*, S)$ 是线性可行锥。则 $\forall d \in T(x^*, S)$

$$d^T \nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla g_i(x^*)^T d + \sum_{j=1}^{\ell} \mu_j^* \nabla h(x^*)^T d \geq 0.$$

- 若 $d^T \nabla f(x^*) = 0$, 此时一阶条件无法判断 x^* 是否是局部最优值点. 则需要利用二阶信息来进一步判断在其可行邻域内的目标函数值.
- 拉格朗日函数在这些方向上的曲率即可用来判断 x^* 的最优性.
- 这里引入临界锥来精确刻画这些方向.

Definition 9.13 (临界锥) *设 (x^*, λ^*, μ^*) 是满足 KKT 条件的 KKT 对, 定义临界锥为*

$$C(x^*, \lambda^*, \mu^*) = \{d \in L(x^*, S) | \nabla g_i(x^*)^T d = 0, \forall i \in \mathcal{A}(x^*) \text{且} \lambda_i^* > 0\}$$

其中 $L(x^*, S)$ 为点 x^* 处的线性化可行方向锥.

1. 临界锥是线性化可行锥 $L(x^*, S)$ 的子集.
2. 沿着临界锥中的方向进行优化, 所有等式约束和 $\lambda_i^* > 0$ 对应的不等式约束 (此时这些不等式均取等) 都会尽量保持不变.
3. 当 $d \in C(x^*, \lambda^*, \mu^*)$ 时, $\forall i = 1, ..., m, j = 1, ..., \ell$ 有 $\lambda_i^* \nabla g_i(x^*)^T d = 0$, $\mu_j^* \nabla h_j(x^*)^T d = 0$, 故

$$d^T \nabla f(x^*) = \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*)^T d + \sum_{j=1}^{\ell} \mu_j^* \nabla h(x^*)^T d = 0$$

4. 临界锥定义了依据一阶导数不能判断是否为下降或上升方向的线性化可行方向, 必须使用高阶导数信息加以判断.

Theorem 9.11 必要性: *假设 x^* 是问题的一个局部最优解, 并且 $T(x^*, S) = L(x^*, g) \cap \mathcal{L}(x^*, h)$ 成立. 令 (x^*, λ^*, μ^*) 满足 KKT 条件, 那么*

$$d^T \nabla_{xx}^2 L(x^*, \lambda^*, \mu^*) d \geq 0, \quad \forall d \in C(x^*, \lambda^*, \mu^*).$$

充分性*:* 假设在可行点 x^* 处, 存在一个拉格朗日乘子 λ^*, 使得 (x^*, λ^*, μ^*) 满足 KKT 条件. 如果

$$d^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*, \mu^*) d > 0, \quad \forall d \neq 0, d \in C(x^*, \lambda^*, \mu^*),$$

那么 x^* 为问题的一个严格局部极小解.

回顾无约束优化问题的二阶最优性条件:

$$\min_{x \in \mathbb{R}^n} f(x)$$

必要条件: 若 x^* 是 f 的一个局部极小点, 则 $\nabla f(x^*) = 0$, $\nabla^2 f(x^*) \succeq 0$
充分条件: 若 $\nabla f(x^*) = 0$, $\nabla^2 f(x^*) \succ 0$, 则 x^* 是 f 的一个局部极小点
约束优化问题的二阶最优性条件也要求某种 "正定性" , 但只需要考虑临界锥 $C(x^*, \lambda^*, \mu^*)$ 中的向量而无需考虑全空间的向量.

Example 9.9

$$\min x_1^2 + x_2^2$$

$$s.t. \ \frac{x_1^2}{4} + x_2^2 - 1 = 0$$

其拉格朗日函数为

$$\mathcal{L}(x, \lambda) = x_1^2 + x_2^2 + \lambda(\frac{x_1^2}{4} + x_2^2 - 1)$$

该问题可行域在任意一点 $x = (x_1, x_2)^T$ 处的线性化可行方向锥为

$$L(x, S) = \{(d_1, d_2) | \frac{x_1}{4} d_1 + x_2 d_2 = 0\}$$

因为只有一个等式约束且其对应函数的梯度非零, 故有 $LICQ$ 成立, 于是

$$L(x, S) = T(x, S)$$

若 (x, λ) 为 KKT 对, 由于无不等式约束, 故临界锥等于线性可行锥, 即

$$C(x, \lambda) = L(x, S)$$

可以计算出其 4 个 KKT 对

$$(x^T, \lambda) = (2, 0, -4), (-2, 0, -4), (0, 1, -1), (0, -1, -1)$$

考虑第一个 KKT 对 $(x^T, \lambda) = (2, 0, -4)^T$, 计算可得

$$\nabla^2_{xx}\mathcal{L}(x, \lambda) = \begin{bmatrix} 0 & 0 \\ 0 & -6 \end{bmatrix}$$

$$C(x, S) = \{(d_1, d_2) | d_1 = 0\}$$

取 $d = (0, 1)$, 则

$$d^T \nabla^2_{xx}\mathcal{L}(y, \lambda) d = -6 < 0$$

因此 y 不是局部最优点. 类似地, 对第三个 KKT 对 $(x^T, \lambda) = (0, 1, -1)$,

$$\nabla^2_{xx}\mathcal{L}(x, \lambda) = \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & 0 \end{bmatrix}$$

$$C(x, S) = \{(d_1, d_2) | d_2 = 0\}$$

对于任意的 $d = (d_1, 0)$ 且 $d_1 \neq 0$,

$$d^T \nabla^2_{xx}\mathcal{L}(z, \lambda) d = \frac{3}{2} d_1^2 > 0$$

因此, z 为一个严格局部最优点.

总结:

问题	一阶条件	二阶条件
可微问题	$\nabla f(x^*) = 0$ (必要)	$\nabla^2 f(x^*) \succeq 0$ (必要) $\nabla^2 f(x^*) \succ 0$ (充分)
凸问题	$\nabla f(x^*) = 0$ (充要)	—

问题	一阶条件	二阶条件	约束品性
一般问题	KKT 条件 (必要)	$d^{\mathrm{T}} \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) d \geq 0, \quad \forall d \in \mathcal{C}(x^*, \lambda^*, \mu^*)$ (必要) $d^{\mathrm{T}} \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) d > 0, \quad \forall d \in \mathcal{C}(x^*, \lambda^*, \mu^*),\ d \neq 0$, (充分)[3]	LICQ[4]
凸问题	KKT 条件 (充要)	—	Slater

1. 一般约束优化问题的二阶充分条件不需要 LICQ 作为前提.

2. 或其他可推出切锥等于线性可行锥的约束规范性条件.

MATH 3011 运筹学 **秋季 2024**

Lecture 10: 无约束优化 线搜索方法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{10.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$.

2 优化算法前置知识

2.1 收敛速率

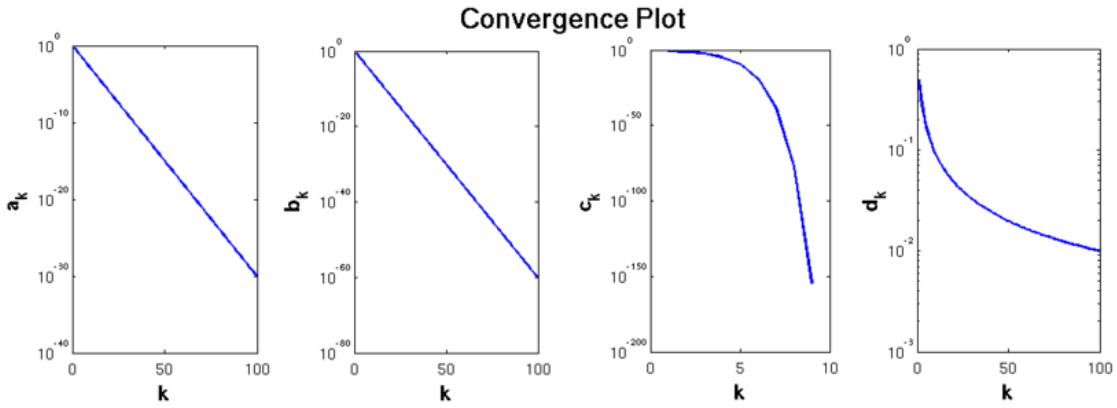
评价算法优劣的标准之一是收敛的快慢，通常称为收敛速率。

一般定义如下：设序列 $\{\mathbf{x}^{(k)}\}$ 收敛与 $\mathbf{x}^*$, 满足

$$0 \le \varlimsup_{k\to\infty} \frac{\|\mathbf{x}^{(k+1)}-\mathbf{x}^*\|}{\|\mathbf{x}^{(k)}-\mathbf{x}^*\|^p} = \beta < \infty \tag{10.2}$$

的非负数 p 的上确界称为序列 $\{\mathbf{x}^{(k)}\}$ 的收敛阶。

- 若在定义式(10.2)中，$p=1$ 且 $\beta<1$, 则称序列是（收敛比 β）**线性收敛**的。例如 $\mathbf{x}^{(k)}=\frac{1}{2^k}$
- 若在定义式(10.2)中，$p>1$, 或者 $\{p=1,\beta=0\}$, 则称序列是**超线性收敛**的。
- 若在定义式(10.2)中，$p=2$, 或则称序列是二**次收敛**的。例如 $\mathbf{x}^{(k)}=\frac{1}{2^{2^k}}$
- **次线性收敛**（sublinear）：一般形如 $\mathbf{x}^{(k)}=\mathcal{O}(1/k),\mathcal{O}(1/k^2)$ 的序列。



上述图像，分别表示线性收敛、超线性收敛、二次收敛和次线性收敛。图中的纵坐标是以 10 为基准取 log 放缩后的值。

2.2 优化算法基本格式

最优化方法通常采用迭代方法求问题的最优解，其基本思想是：

给定一个初始点 $\mathbf{x}^{(0)} \in \mathbb{R}^n$，按照某一迭代规则产生一个点列 $\{\mathbf{x}^{(k)}\}$，使得当 $\{\mathbf{x}^{(k)}\}$ 是有穷点列时，其最后一个点是最优化模型问题的最优解，当 $\{\mathbf{x}^{(k)}\}$ 是无穷点列时，它有极限点且其极限点是最优化模型问题的最优解。

一个好的迭代算法应具备的典型特征是：

迭代点 $\mathbf{x}^{(k)}$ 能稳定地接近局部极小点 $\mathbf{x}^*$ 的小邻域，然后迅速收敛于 $\mathbf{x}^*$. 一般地，对于某种算法我们需要证明其迭代点列 $\mathbf{x}^{(k)}$ 的聚点（即子列的极限点）为一局部极小点。在实际计算中，当指定的收敛准则满足时，迭代即终止。

(1) 最优化迭代算法的基本结构之一

(a) 给定初始点 $\mathbf{x}^{(0)}$

(b) 计算搜索方向 $\mathbf{d}^{(k)}$，即构造某价值函数 ψ 在 $\mathbf{x}^{(k)}$ 点处的下降方向作为搜索方向；

(c) 确定步长因子 α_k，使该价值函数值有某种程度的下降；

(d) 迭代更新，令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$.

(e) 判断停机准则，若 $\mathbf{x}^{(k+1)}$ 满足某种终止条件，则停止迭代，得到近似最优解 $\bar{\mathbf{x}} = \mathbf{x}^{(k+1)}$. 否则，返回 (b) 重复以上步骤。

(2) 最优化迭代算法的基本结构之二

(a) 给定初始点 $\mathbf{x}^{(0)}$

(b) 构造某价值函数 ψ 在 $\mathbf{x}^{(k)}$ 附近（如一定半径内）的二次近似模型；

(c) 求解该近似模型得到 $\mathbf{s}^{(k)}$ 作为更新位移向量；

(d) 迭代更新，令 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$.

(e) 判断停机准则，若 $\mathbf{x}^{(k+1)}$ 满足某种终止条件，则停止迭代，得到近似最优解 $\bar{\mathbf{x}} = \mathbf{x}^{(k+1)}$. 否则，返回 (b) 重复以上步骤。

3 梯度类算法

梯度向量 $\nabla f(x)$ 是函数 f 在点 x 处增加最快的方向，故它成为最优化时的重要工具。实际上针对无约束最优化问题，大部分求解算法属于下面的梯度方法类。

梯度类算法：

(0) 初始化：选取适当的初始点 $x^0 \in \mathbb{R}^n$, 令 $k := 0$.

(1) 计算搜索方向：利用适当的正定对称阵 H_k 计算搜索方向向量 $d^k := -H_k \nabla f(x^k)$.（如果 $\nabla f(x^k) = 0$, 则结束计算）

(2) 确定**步长**因子：解一维最优化问题 $\min\limits_{\alpha \geq 0} f(x^k + \alpha d^k)$, 求出**步长**$\alpha = \alpha_k$, 令 $x^{k+1} = x^k + \alpha_k d^k$, $k := k + 1$, 回到第 (1) 步。

注：在机器学习领域，步长通常被称为学习率（learning rate）。

例 10.1 若 $f(x)$ 二阶可导，我们有

$$f(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + O(\|x - x^k\|^2). \tag{10.3}$$

取负梯度方向

$$d^k = -\nabla f(x^k),$$

则当 α_k 足够小时，总能使

$$f(x^k + \alpha_k d^k) < f(x^k).$$

例 10.2 若 $f(x)$ 三阶可导，我们有

$$\begin{aligned} f(x) = \quad & f(x^k) + \nabla f(x^k)^T (x - x^k) \\ & + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k) + O(\|x - x^k\|^3) \end{aligned} \tag{10.4}$$

假设函数 f 在 x^k 点处的 *Hesse* 矩阵 $\nabla^2 f(x^k)$ 正定，取搜索方向

$$d^k = -G_k^{-1}\nabla f(x^k),$$

其中 $G_k = \nabla^2 f(x^k)$。这样的取法叫做牛顿方向，我们后面会进一步讨论。若 α_k 充分小，那么也可以得到

$$f(x^k + \alpha_k d^k) < f(x^k).$$

4 确定步长因子：一维搜索

在迭代格式中，沿着下降方向 d^k, 通过解一维最优化问题

$$\min_{\alpha \geq 0} \varphi(\alpha) = f(x^k + \alpha d^k) \tag{10.5}$$

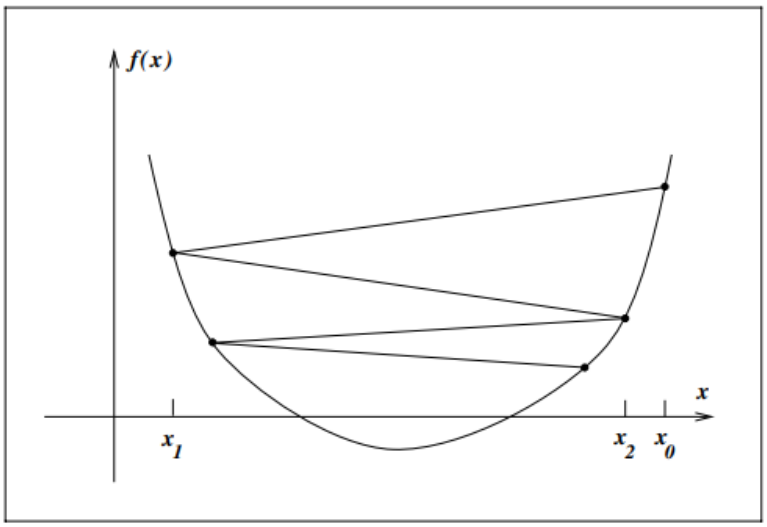
确定步长因子的方法称为**一维搜索** (Line Search).

若以问题(10.5)的最优解为步长，此时称为**精确一维搜索** (Exact Line Search).

经常用到的精确一维搜索有黄金分割法和插值迭代法。即使说是精确一维搜索，通过有限次计算求出问题(10.5)的严密解一般也是不可能的，实际上在得到有足够精度的近似解时，就采用它作为步长。

在实际计算中，往往不是求解一维最优化问题(10.5), 而是找出满足某些适当条件的粗略近似解作为步长，此时称为**非精确一维搜索** (Inexact Line Search). 与精确一维搜索相比，在很多情况下采用非精确一维搜索可以提高整体计算效率。

4.1 线搜索的重要性



上图中，由于步长 α 选择较大，迭代产生了左右震荡。反之，若是步长太小，那么算法收敛速度非常缓慢。线搜索的目标就是，使得沿着下降方向 d，每次的函数值满足充分下降（sufficient decrease）条件。

4.2 回溯线搜索法

最简单、常见的线搜索条件为回溯线搜索法 (Backtracking linesearch):

1. 选取 $\gamma \in (0,1)$ 和 $c \in (0,1)$
2. 选择最小的整数 $t \geq 0$, 使得

$$f(x^k + \gamma^t d^k) \leq f(x^k) + c\gamma^t \nabla f(x^k)^T d^k \tag{10.6}$$

3. 令 $\alpha_k = \gamma^t$, 更新 $x^{k+1} = x^k + \alpha_k d^k$

(10.6)被称为 Armijo-Goldstein 不等式, 它是函数值充分下降条件的一种形式。故 backtracking 也被称为 Armijo-backtracking linesearch. γ 通常选取 0.9 或者 0.5. c 通常选取 $10^{-2}, 10^{-3}$ 等较小的数。若 c 较大，则要求每次函数下降量足够大，但是需要更多的搜索步数。反之，则下降量较小，造成算法总体下降速度过慢。实际问题中，需要调试参数 γ, c 的选择对算法进行加速。

正整数 t 的存在性：

$$\lim_{\alpha \downarrow 0} \frac{f(x^k + \alpha d^k) - f(x^k)}{\alpha} = f'(x^k; d^k) < cf'(x^k; d^k) < 0.$$

故，存在 $\bar{\alpha} > 0$，使得

$$\frac{f(x^k + \alpha d^k) - f(x^k)}{\alpha} \leq cf'(x^k; d^k), \quad \forall \alpha \in (0, \bar{\alpha}) \tag{10.7}$$

回溯线搜索法和下面的 Armijo-Goldstein 线搜索非常相似。令

$$\varphi(\alpha) = f(x + \alpha d).$$

我们有 $\varphi'(\alpha) = \nabla f(x + \alpha d)^T d.$

Armijo-Goldstein 条件为:

$$\varphi(\alpha) \leq \varphi(0) + \rho\alpha\varphi^{'}(0) \tag{10.8}$$

$$\varphi(\alpha) \geq \varphi(0) + (1-\rho)\alpha\varphi^{'}(0) \tag{10.9}$$

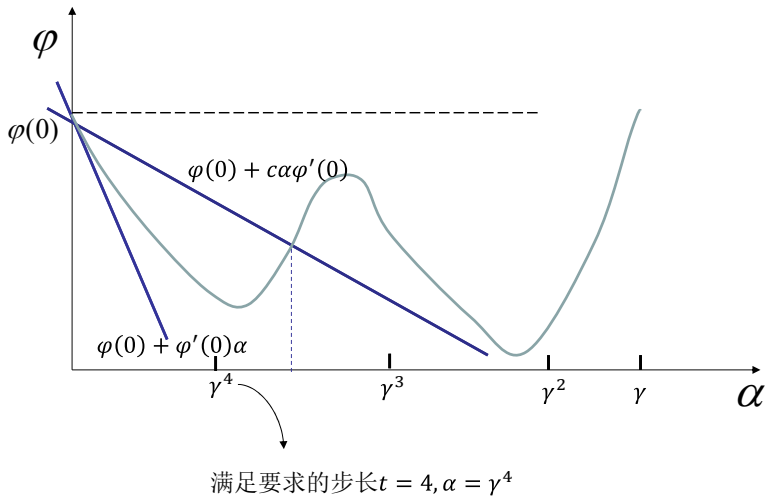


图 10.1: 满足回溯条件(10.6)的步长图例

其中 $\rho \in (0, 1/2)$ 是一个固定参数。(10.8)要求函数值满足充分下降条件，其对应于条件(10.6)。(10.9)要求函数值下降量不是很小，对应于我们在回溯法中取最小的 t 即最大的步长 $\alpha_k = \gamma^t$.

Armijo-Goldstein 条件(10.8)和(10.9)虽然可以使得函数值下降，但是如图 10.2, 其排除了局部最小值点。

4.3 Wolfe-Powell 条件

Wolfe(1968)-Powell(1976) 条件是另外一种常见的非精确线搜索方法。

Wolfe-Powell 条件如下:

$$\varphi(\alpha) \leq \varphi(0) + c_1 \alpha \varphi^{'}(0) \tag{10.10}$$

$$\varphi^{'}(\alpha) \geq c_2 \varphi^{'}(0) \tag{10.11}$$

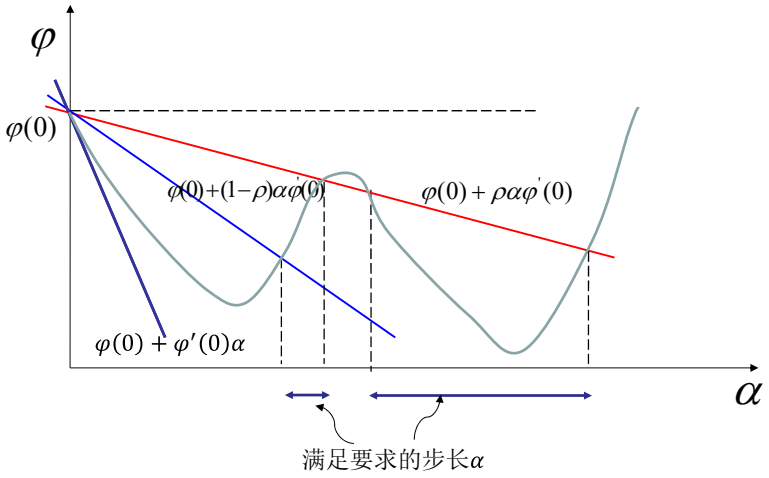


图 10.2: 满足 Armijo-Goldstein 条件(10.8)和(10.9)的步长图例

其中 $0 < c_1 < c_2 < 1$ 是固定参数。**通常来说 c_1 比较小，例如 $c_1 = 10^{-3}$. $c_2 = 0.9$.**
考虑问题

$$\min_{\alpha \in \mathbb{R}} \varphi(\alpha),$$

其最优条件为 $\varphi'(\alpha) = 0$. (10.11) 使得 $\varphi'(\alpha)$ 更接近 0，也被称为弱 Wolfe-Powell 条件。在很多实际算法中，式(10.11)常被强化的双边条件(10.12)所取代（也被称为强 Wolfe-Powell 条件）

$$|\varphi^{'}(\alpha)| \leq c_2 |\varphi^{'}(0)|, 0 < c_1 < c_2 < 1. \tag{10.12}$$

此条件排除了 $\varphi^{'}(\alpha)$ 为非常大的正数情况。
Wolfe-Powell 条件存在性：若问题(10.1)中 f 连续可微。令 d^k 是下降方向，并且假设 f 沿着射线方向 $\{x^k + \alpha d^k \mid \alpha > 0\}$ 有下界。那么一定存在 $0 < c_1 < c_2 < 1$ 使得 (10.10)、(10.11) 或(10.12)成立。

证明：因为 f 沿着射线方向 $\{x^k + \alpha d^k \mid \alpha > 0\}$ 有下界，$l(\alpha) := f(x^k) + \alpha c_1 \nabla f(x^k)^T d^k$ 单调减小至 $-\infty$, 可知 $\varphi(\alpha)$ 与 $l(\alpha)$ 至少有一个交点。设 $\bar{\alpha}$ 是最小的交点。有下述不等式成立

$$f(x^k + \alpha_1 d^k) \leq f(x^k) + c_1 \alpha_1 \nabla f(x^k)^T d^k, \quad \forall \alpha_1 \in (0, \bar{\alpha}).$$

由中值定理可知，存在 $\alpha_2 \in (0, \bar{\alpha})$ 使得 $f(x^k + \bar{\alpha} d^k) - f(x^k) = \bar{\alpha} \nabla f(x^k + \alpha_2 d^k)^T d^k$ 因为 $l(\bar{\alpha}) = \varphi(\bar{\alpha})$,

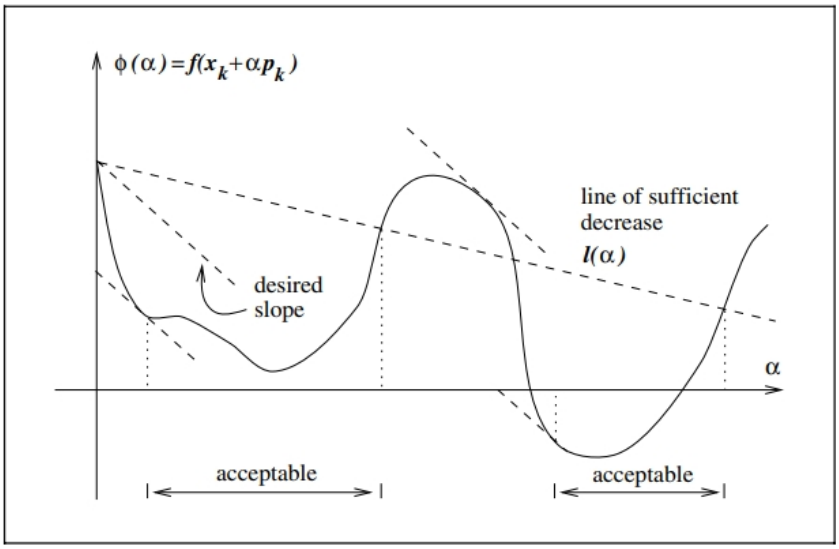


图 10.3: Wolfe-Powell 条件示例。图片来源：Numerical optimization. By Jorge Nocedal and Stephen J. Wright.

即 $f(x^k+\bar{\alpha}d^k)=f(x^k)+c_1\bar{\alpha}\nabla f(x^k)^Td^k$, 我们有

$$\nabla f(x^k+\alpha_2 d^k)^Td^k=c_1\nabla f(x^k)^Td^k>c_2\nabla f(x^k)^Td^k,\quad 1>c_2>c_1>0.$$

因此 $0<c_1<c_2<1,\alpha_1\in(0,\bar{\alpha}),\alpha_2\in(0,\bar{\alpha})$ 满足(10.10)、(10.11).
注意到 $\nabla f(x^k+\alpha_2 d^k)^Td^k<0$, 因此(10.12)也成立。

实际中，为了确定满足 Wolfe-Powell 条件的步长，一种方法是采用插值法。其步骤如下：

(1) 给定初始一维搜索区间 $[0,\alpha_0]$, 以及 $c_1\in(0,1/2)$, $c_2\in(c_1,1)$. 记 $a_1=0,a_2=\alpha_0$.
计算 $\varphi(0)=f(x^k)$, $\varphi'(0)=\nabla f(x^k)^Td^k$. 并令 $a_1=0,a_2=\alpha_0,\varphi_1=\varphi(0),\varphi_1'=\varphi'(0)$.
选取适当的 $\alpha\in(a_1,a_2)$.

(2) 计算 $\varphi=\varphi(\alpha)=f(x^k+\alpha d^k)$. 若 $\varphi(\alpha)\le\varphi(0)+c_1\alpha\varphi'(0)$, 则转到第 (3) 步。否则，由 $\varphi_1,\varphi_1',\varphi$ 构造两点二次插值 $p_1(\alpha)=A_1\alpha^2+B_1\alpha+C_1$, 逼近区间 $[a_1,\alpha]$ 上的 $\varphi(\alpha)$. 使得 $p_1(a_1)=\varpi_1,p_1'(a_1)=\varphi_1'$, $p_1(\alpha)=\varphi$. 并得 $p_1(a_1)$ 极小点

$$\hat{\alpha}=a_1+\frac{1}{2}\frac{(a_1-\alpha)^2\varphi_1'}{(\varphi_1-\varphi)-(a_1-\alpha)\varphi_1'}.$$

于是置 $a_2=\alpha,\alpha=\hat{\alpha}$, 重复第 (2) 步。

(3) 计算 $\varphi'=\varphi'(\alpha)=\nabla f(x^k+\alpha d^k)^Td^k$. 若 $\varphi'(\alpha)\ge c_2\varphi'(0)$, 则输出 $\alpha_k=\alpha$, 并停止搜索。否则，由 $\varphi,\varphi',\varphi_1'$ 构造两点二次插值多项式

$$p_2(\alpha)=A_2\alpha^2+B_2\alpha+C_2,$$

使得 $p_2'(a_1)=\varphi_1', p_2'(\alpha)=\varphi', p_2(\alpha)=\varphi$. 并得其极小点

$$\hat{\alpha}=\alpha-\frac{(a_1-\alpha)\varphi'}{\varphi_1'-\varphi'}.$$

于是置 $a_1=\alpha, \alpha=\hat{\alpha}, \varphi_1=\varphi, \varphi_1'=\varphi'$, 返回第 (3) 步。

4.4 线搜索的全局收敛性

从任意初始点出发，如果某迭代算法产生的点列的极限（聚点），在适当假定下可保证恒为问题的最优解（或者稳定点），则称该迭代法具有全局收敛性 (Global Convergence).

为了证明迭代法的下降性，我们应尽量避免搜索方向与负梯度方向几乎正交的情形，即要求 d^k 偏离 $g^k=\nabla f(x^k)$ 的正交方向远一些。否则，${g^k}^T d^k$ 接近于零，d^k 几乎不是下降方向。

为此，我们假设 d^k 与 $-g^k$ 的夹角 θ_k 满足

$$\theta_k \le \frac{\pi}{2}-\mu,\ \forall k \tag{10.13}$$

其中 $\mu>0$（与 k 无关）。
显然 $\theta_k\in[0,\pi/2)$, 其定义为

$$\cos\theta_k=\frac{-{g^k}^T d^k}{\|g^k\|\|d^k\|}=\frac{-{g^k}^T s^k}{\|g^k\|\|s^k\|} \tag{10.14}$$

这里 $s^k=\alpha_k d^k=x^{k+1}-x^k$.

下面给出各种步长准则下的下降算法的全局收敛性结论。

Theorem 10.1 *假设 $f(x)$ 有下界，即 $f(x)>-\infty, \forall x\in\mathbb{R}^n$. 设 $f(x)$ 在包含水平集 $L(x^0)=\{x\mid f(x)\le f(x^0)\}\subset\mathcal{N}$ 的开集 $\mathcal{N}$ 上连续可微。同时，梯度 $\nabla f(x)$ 在 $\mathcal{N}$ 上是李氏连续 (Lipschitz continuous) 的，即存在 $L>0$，使得*

$$\|\nabla f(x)-\nabla f(y)\|\le L\|x-y\|,\quad \forall x,y\in\mathcal{N}.$$

下降算法的搜索方向 d^k 与 $-\nabla f(x^k)$ 之间的夹角 θ_k 满足式(10.13)，其中步长 α_k 由 Wolfe-Powell (10.10),(10.11)确定。那么，$\nabla f(x^k)\to 0$ as $k\to\infty$.

Proof: 全局收敛性证明： 为了记号简洁，我们记所有的 k, $g_k=\nabla f(x^k)$，$f_k=f(x^k)$. 由梯度李氏连续性可知：

$$\|g_{k+1}-g_k\|\le L\|x_{k+1}-x_k\|=L\alpha_k\|d^k\|$$

由(10.11)可知，

$$(g_{k+1}-g_k)^T d^k\ge(c_2-1)g_k^T d^k.$$

结合上述两个不等式，我们有

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{g_k^T d^k}{\|d^k\|^2}. \qquad \text{(步长有下界)}$$

带入(10.10), 我们有

$$f_{k+1} \leq f_k - c_1 \frac{1 - c_2}{L} \frac{(g_k^T d^k)^2}{\|d^k\|^2}.$$

根据(10.14), 我们有

$$f_{k+1} \leq f_k - c_1 \frac{1 - c_2}{L} \cos^2 \theta_k \|g_k\|^2.$$

将上述不等式，对 $k = 0, ..., T$ 相加，可得

$$f_{T+1} \leq f_0 - c_1 \frac{1 - c_2}{L} \sum_{k=0}^{T} \cos^2 \theta_k \|g_k\|^2.$$

因为 f 有下界，故

$$\sum_{k=0}^{T} \cos^2 \theta_k \|g_k\|^2 < +\infty.$$

上式对任意 T 成立，故

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|g_k\|^2 < +\infty.$$

又因为(10.13), 存在 $\delta > 0$, 使得

$$\cos \theta_k \geq \delta.$$

所以

$$\lim_{k \to \infty} \|\nabla f(x^k)\| = 0.$$

■

MATH 3011 运筹学 **秋季 2024**

Lecture 11: 无约束优化 梯度下降法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{11.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$. f 是可微函数。

2 梯度下降方法

梯度下降法取负梯度作为迭代算法的搜索方向，其迭代格式为

$$x^{k+1} = x^k - \alpha_k \nabla f(k^k).$$

Algorithm 1 梯度下降算法 GD

Require: 选取初始点 x^0, 设置终止误差 $\varepsilon > 0$, 令 $k := 0$.

1: **while** $\|\nabla f(x^k)\| > \epsilon$ **do**

2: 令 $d^k = -\nabla f(x^k)$, 并由一维搜索确定步长因子 α_k 使得 $f(x^k+\alpha_k d^k)$ 满足 Backtracking linesearch 或者 Wolfe-Powell 条件

3: 迭代更新 $x^{k+1} = x^k + \alpha_k d^k$, 置 $k := k + 1$ 。

4: **end while**

3 梯度下降法全局收敛性定理

我们下面着重讲解，在非凸、凸函数、强凸三种情况下，梯度算法的收敛结果。

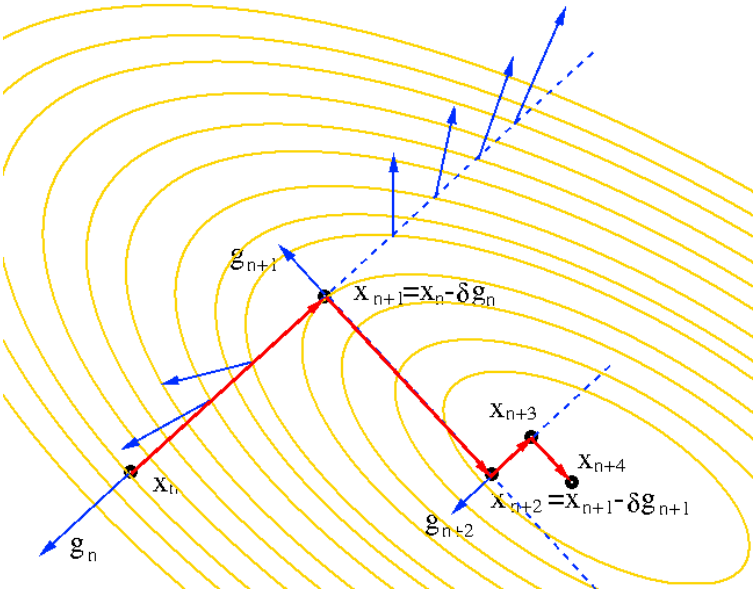


图 11.1: 梯度下降法求解二次问题的迭代示意图

3.1 非凸函数情况下的收敛

Definition 11.1 *若给定函数 f 是可微函数，并且对于任意定义域的点 x, y，梯度满足李氏连续性（Lipschitz continuous），即存在 $L > 0$，使得*

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|.$$

则称 f 是梯度李氏连续，或者李氏光滑（L-光滑）的。

Lemma 11.1 *若 f 是李氏光滑的 (即 ∇f 是李氏连续的)，则 f 有二次上界，即*

$$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}\|y - x\|^2.$$

Proof: 由 f 可微，可得

$$\begin{aligned} f(y) &= f(x) + \int_0^1 \nabla f(x + t(y - x))^T(y - x)dt \\ &= f(x) + \nabla f(x)^T(y - x) + \int_0^1 (\nabla f(x + t(y - x)) - \nabla f(x))^T(y - x)dt \end{aligned}$$

因此，

$$\begin{aligned} f(y) - f(x) - \nabla f(x)^T(y-x) &= \int_0^1 (\nabla f(x + t(y-x)) - \nabla f(x))^T (y-x) dt \\ &\le \int_0^1 \|\nabla f(x + t(y-x)) - \nabla f(x)\| \|y-x\| dt \\ &\le \int_0^1 Lt\|y-x\|^2 dt \\ &= \frac{L}{2}\|y-x\|^2. \end{aligned}$$

■

很多常用的函数满足李氏光滑性, 例如 $f(x) = \frac{1}{2}\|Ax-b\|^2$. 我们有

$$\|\nabla f(y) - \nabla f(x)\| \le \lambda_{\max}(A^TA)\|y-x\|,$$

这里 $\lambda_{\max}(A^TA)$ 是 A^TA 的最大特征根。因此，二次函数的 Lipschitz 常数是 $L = \lambda_{\max}(A^TA)$. 通常来说, $L = \max_x \lambda_{\max}(\nabla^2 f(x))$, 即定义域内的所有 Hessian 矩阵的最大特征根。

反例： $f(x) = e^x, f(x) = x^3$ 的二次导数没有界，它们不是 Lipshitz 光滑的。

作业 11.1 对于逻辑回归问题，$\min f(x) = \frac{1}{N}\sum_{i=1}^N \log(1 + e^{-y_i a_i^T x})$, 这里 $y_i \ge 0, a_i$ 是已知的。估计 ∇f 李氏常数 L.

注：梯度法的另一个理解 - 最大化最小化算法类

构造 $f(x)$ 的一个二次函数上界.

定义: $q_x(y)$ 是 f 的上界函数, 如果

- $q_x(y) = f(x)$
- $q_x(y) \ge f(y)$, for any y.

最大化-最小化方法:

$$x_{k+1} = \arg\min_y q_{x_k}(y)$$

我们有

$$f(x_{k+1}) \le q_{x_k}(x_{k+1}) \le q_{x_k}(x_k) = f(x_k)$$

Theorem 11.1 若 f 是 $L-$ 光滑函数并且 f 有最小值 f^*, 则选取步长 $\alpha_k = 1/L$, 我们有 $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ 并且并且对于任意正整数 T，有

$$\min_{k=0,1,\dots,T} \|\nabla f(x_k)\|^2 \le \frac{2L(f(x_0) - f^*)}{T}.$$

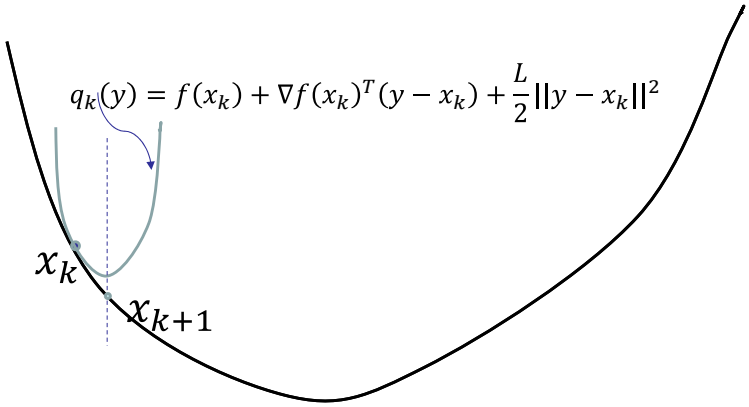


图 11.2: 最大化-最小化示例，在每个点 x_k 出，用二次上界 $q_k(y)$ 作为原函数的逼近。这样可以通过最小化上界函数得到函数值下降的迭代点 x_{k+1}.

证明： 由李氏光滑性，$q_{x_k}(y) = f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{1}{2\alpha}\|y - x_k\|^2$ 为一个上界函数。梯度法迭代满足

$$x_{k+1} = \arg\min_y q_{x_k}(y) = x_k - 1/L\nabla f(x_k).$$

所以

$$f(x_{k+1}) \le q_{x_k}(x_{k+1}) = q_{x_k}(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 = f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2. \tag{11.2}$$

因此

$$\sum_{k=0}^{\infty} \frac{1}{2L}\|\nabla f(x_k)\|^2 < \infty.$$

故

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

并且对于任意正整数 T，有 $\min_{k=0,1,\dots,T} \|\nabla f(x_k)\|^2 \le \frac{2L(f(x_0)-f^*)}{T}$.

3.2 凸函数情况下的收敛

我们需要凸函数的如下性质。

Lemma 11.2 *设函数 $f(x)$ 是 $\mathbb{R}^n$ 上的凸可微函数, 则以下结论等价:*

1. f 的梯度为 L -连续的;

2. $\nabla f(x)$ 有余强制性, 即对任意的 $x, y \in \mathbb{R}^n$, 有

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\|^2 \tag{11.3}$$

我们只证明: (1)⇒ (2) 定义函数 $\phi(y) = f(y) - \nabla f(x)^T y$. 函数 $\phi(y)$ 是凸函数, 并且也是 $L-$ 光滑的。因为 $\nabla \phi(x) = 0$, 故 x 是 ϕ 的最小值。根据 $L-$ 光滑,

$$\phi(x) \leq \phi(y - \frac{1}{L} \nabla \phi(y)) \leq \phi(y) - \frac{1}{2L} \|\nabla \phi(y)\|^2.$$

由 $\nabla \phi(y) = \nabla f(y) - \nabla f(x)$ 可得

$$\phi(y) - \phi(x) \geq \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|^2.$$

即

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|^2.$$

交换上面 x, y, 得到的不等式与上述不等式相加, 即可得到结论。

Theorem 11.2 *若 f 是 $L-$ 光滑的凸函数, 并且 f 有最小值 f^*, 则选取步长 $\alpha_k = 1/L$, 对于任意 $T \geq 1$,*

$$f(x_T) - f^* \leq \frac{L}{2T} \|x_0 - x^*\|^2.$$

证明: 由(11.2)和 f 是凸函数可得

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|^2 \\ &\leq f^* + \nabla f(x_k)^T (x_k - x^*) - \frac{1}{2L} \|\nabla f(x_k)\|^2 \\ &= f^* + \frac{L}{2} \left(\|x_k - x^*\|^2 - \|x_k - x^* - \frac{1}{L} \nabla f(x_k)\|^2 \right) \\ &= f^* + \frac{L}{2} \left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right) \end{aligned} \tag{11.4}$$

(11.4)表明梯度法中，函数值和最小值的差是严格减小的。对上式 $k = 0, 1, ..., T$ 相加可得

$$\begin{aligned}\sum_{k=1}^{T}(f(x_k) - f^*) &\leq \frac{L}{2}\sum_{k=1}^{T}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right) \\ &= \frac{L}{2}\left(\|x_0 - x^*\|^2 - \|x_{T+1} - x^*\|^2\right) \\ &\leq \frac{L}{2}\|x_0 - x^*\|^2.\end{aligned}$$

因为 $f(x_k)$ 单调递减，所以

$$f(x_T) - f^* \leq \frac{L}{2T}\|x_0 - x^*\|^2.$$

结论：$f(x_k) - f^*$ 收敛的速度是次线性的。收敛到 $f(x_k) - f^* \leq \epsilon$ 的速度是 $\mathcal{O}(1/k)$.

补充：一阶算法的复杂度下界。

Definition 11.2 一阶方法：任何选择 x_{k+1} 在集合中的迭代算法

$$x_0 + \text{span}\{\nabla f(x_0), \nabla f(x_1), \ldots, \nabla f(x_k)\}$$

问题类：满足 L 式光滑和凸假设的任何函数。
定理 (Nesterov)：对于每个整数 $k \leq \frac{n-1}{2}$ 和每个 x_0，存在在问题类中的函数，对于任何一阶方法

$$f(x_k) - f^* \geq \frac{3L\|x_0 - x^*\|^2}{32(k+1)^2}$$

- 表明梯度方法的 $\frac{1}{k}$ 速率不是最优的。
- Nesterov's 加速梯度方法有 $\frac{1}{k^2}$ 的收敛性。

该定理见 Yu. Nesterov, Lectures on Convex Optimization (2018), section 2.1. (Theorem 2.1.7 in the book.) Nesterov 加速梯度算法，感兴趣也参考此书 section 2.2.

3.3 强凸函数收敛性

Definition 11.3 可微函数是 $\mu-$ 强凸函数，如果

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2}\|y - x\|^2, \forall x, y \in \text{dom} f.$$

假设中增加强凸性后，我们可以得到更好的结果。强凸性意味着最小值点唯一。

Lemma 11.3 设函数 $f(x)$ 是 $\mathbb{R}^n$ 上的 $\mu-$ 强凸可微函数，则有如下不等式：

$$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \frac{\mu L}{L+\mu}\|x - y\|^2 + \frac{1}{L+\mu}\|\nabla f(x) - \nabla f(y)\|^2, \forall x, y \in \text{dom} f \qquad (11.5)$$

证明： 记 $\phi(x) = f(x) - \frac{\mu}{2}\|x\|^2$. 则 $\phi(x)$ 是凸函数, 并且是 $L-\mu$ 李氏光滑。由余强制性(11.3), 可得

$$(\nabla\phi(x) - \nabla\phi(y))^T(x-y) \geq \frac{1}{L-\mu}\,\|\nabla\phi(x) - \nabla\phi(y)\|^2, \forall x, y \in\in \mathrm{dom} f.$$

带入 $\nabla\phi(x) = \nabla f(x) - \mu x$，可得(11.5).

如果 $x^+ = x - \alpha\nabla f(x)$ 且 $0 < \alpha \leq \frac{2}{\mu+L}$:

$$\begin{aligned}
\left\|x^+ - x^*\right\|^2 &= \|x - \alpha\nabla f(x) - x^*\|^2 \\
&= \|x - x^*\|^2 - 2\alpha\nabla f(x)^T(x-x^*) + \alpha^2\|\nabla f(x)\|^2 \\
&\leq (1 - \alpha\frac{2\mu L}{\mu+L})\|x-x^*\|^2 + \alpha(\alpha - \frac{2}{\mu+L})\|\nabla f(x)\|^2 \\
&\leq (1 - \alpha\frac{2\mu L}{\mu+L})\|x-x^*\|^2
\end{aligned}$$

$$\|x_k - x^*\|^2 \leq c^k\|x_0 - x^*\|^2$$

其中 $c = 1 - \alpha\frac{2\mu L}{\mu+L}$。

- 这意味着 x_k 线性收敛至最优值 x^*。
- 对于 $\alpha = \frac{2}{\mu+L}$，我们得到 $c = \left(\frac{\kappa-1}{\kappa+1}\right)^2$ 其中 $\kappa = \frac{L}{\mu}$ 被称为条件数。例如，正定矩阵 A 的条件数是其最大特征根与最小特征根比值。矩阵条件数大，意味着问题是病态的。

$$f(x_k) - f^* \leq \frac{L}{2}\|x_k - x^*\|^2 \leq c^k\frac{L}{2}\|x_0 - x^*\|^2$$

结论：达到 $f(x_k) - f^* \leq \epsilon$ 所需的迭代次数是 $O(\log(1/\epsilon))$。

4 梯度下降法总结

问题类型	**收敛描述**	**迭代复杂度**
Nonconvex L-smooth	$\|\nabla f(x)\| \leq \epsilon$	$O\left(\frac{1}{\epsilon^2}\right)$
Convex L-smooth	$f(x_k) - f^* \leq \epsilon$	$O\left(\frac{1}{\epsilon}\right)$
Strongly convex μ-smooth	$\|x_k - x^*\|^2 < \epsilon$	$O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$

表 11.1: Convergence for gradient method under function properties

MATH 3011 运筹学 **秋季 2024**

Lecture 12: 无约束优化 随机梯度算法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题背景

我们以监督学习为例，作为随机梯度算法 (Stochastic gradient descent，简称 SGD) 的重要应用。

什么是监督学习？

监督学习是一种机器学习范式，其中模型在带标签的数据上进行训练。训练数据包含输入-输出对。给定这些对，算法学习输入和输出之间的映射，然后可以用于预测新的、以前未见过的输入的输出。

它是如何工作的？

设想一个老师和一个学生。老师有一本习题集。当学生尝试这些问题时，老师会指出正确的解答来纠正任何错误。随着时间的推移，学生变得擅长独立解决类似的问题。

在监督学习中，算法扮演学生的角色，数据充当问题集，标签是正确的解决方案。算法对训练数据进行预测，并在出错时根据真实标签调整其理解。

关键概念：

1. **训练数据**：由输入特征和正确的输出组成，这是监督学习的基础。

2. **模型**：这是用来根据输入特征预测输出的算法或算法集。

3. **损失函数**：监督学习的核心概念是损失函数。这是一个数学函数，用于测量我们模型的预测与真实值之间的差距。监督学习的目标是最小化这种损失。

损失函数的例子：最常用于回归任务（输出是连续值）的损失函数是**均方误差 (MSE)**。如果我们有 N 个数据点，对于每个点 a_i，模型 ϕ 预测的值是 $\hat{y_i}$，真实值（标签）是 y_i，那么 MSE 为：

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y_i} - y_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(\phi(a_i) - y_i)^2$$

简单来说，这计算了预测值和真实值之间的平均平方差。我们记模型 ϕ 的参数为 x，则优化的目标为

$$\min_x \frac{1}{N}\sum_{i=1}^{N}(\phi(x; a_i) - y_i)^2$$

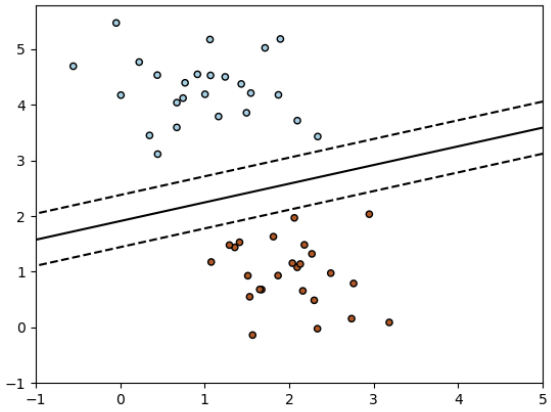


图 12.1: 分类任务示例。来源：https://scikit-learn.org/stable/modules/sgd.html

4. **优化算法**：这就是 SGD 等技术发挥作用的地方。它们的任务是调整模型的参数（如神经网络中的权重）以最小化损失。

深度神经网络 (DNN) 是一种前馈结构，通过堆叠多个神经元层来构建。每个神经元都会应用一个线性变换，然后再通过一个非线性激活函数进行激活。对于分类任务，最后一层通常后接一个 softmax 函数以产生类概率。然后，使用交叉熵损失来测量 DNN 的性能。
神经网络的函数如下：

- 第 l 层, 给定前一层的激活值 $a^{(l-1)}$ ：

$$z^{(l)} = \phi(W^{(l)}, b^{(l)}; a^{(l-1)}) = W^{(l)}a^{(l-1)} + b^{(l)} \tag{12.1}$$

- ReLU作为激活函数，得到第 l 层的激活值

$$a^{(l)} = \text{ReLU}(z^{(l)}) \tag{12.2}$$

- 对于最后一层，我们使用 softmax 函数得到类概率：

$$\text{softmax}(z_i^{(L)}) = \frac{e^{z_i^{(L)}}}{\sum_{j=1}^{C} e^{z_j^{(L)}}} \tag{12.3}$$

  其中，C 是类别的数量。

- 测量网络预测与实际标签之间差异的损失函数是交叉熵损失：

$$\text{CE}(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c}\log(\text{softmax}(z_{i,c}^{(L)})) \tag{12.4}$$

对于大规模神经网络训练，我们通常是求解下列形式的数学优化问题

$$\min_x f(x) = \frac{1}{N}\sum_{i=1}^{N} f(x, Y_i). \tag{12.5}$$

式子(12.5)中, 未知量 x 表示网络参数, $Y_i, i = 1, 2, \ldots, N$ 是训练数据, 它们组成总体训练集 $\{Y_1, \ldots, Y_N\}$。函数 $f(x)$ 表示的是模型的损失函数，例如在分类任务中，$f(x)$ 表示的是模型预测数据的分类与真实类别的误差。训练神经网络的目标即是寻找最优的参数 x 使得误差最小。一般来说, 求解优化问题(12.5)关注的是最小化训练误差，而机器学习关注的是降低模型的泛化误差。在本课程中，我们更关注的是优化算法求解训练模型的最优解。

神经网络模型具有以下两个特征：（1）参数量非常大；（2）训练样本数量非常大。表 12.1 和表 12.2 给出了一些经典的数据集和模型大小。模型规模和数据集大小制约了训练效率，大模型的训练有可能花费数天甚至数月时间。优化器的选择直接影响模型的训练效率。本节中，我们简单地探讨常用的模型训练优化器。下面，我们将会介绍常见的优化器的参数设定，如学习率，动量参数，批量大小等。

表 12.1: 大数据集数据量

数据集	数据集大小
Cifar10, Cifar100	60000 张图像
ImageNet	约 1400 万张图像
MS Coco	约 33 万张图像
GPT-3	45TB

表 12.2: 大模型参数量

模型名称	参数量
VGG16	1.4 亿
ResNet50	2500 万
DenseNet-201	2000 万
GPT-3	1750 亿
GPT-4	1.8 万亿

网络上爆料的 GPT-4 训练成本：一次的训练的成本为 6300 万美元，OpenAI 训练 GPT-4 的 FLOPS 约为 2.15e25，在大约 25000 个 A100 上训练了 90 到 100 天，利用率在 32% 到 36% 之间。

2 随机梯度算法 SGD

注意(12.5)的求和形式, 我们有

$$\nabla f(x) = \frac{1}{N}\sum_{i=1}^{N} \nabla f(x, Y_i).$$

因此当参数量非常大的时候，计算 $\nabla f(x, Y_i)$ 非常慢并且占用内存资源；当数据量 N 也非常大的时候，计算整体函数 $f(x)$ 的梯度是不可取的。所以，随机梯度算法应运而生。为了解决无法计算梯度的难点，SGD 的想法是随机抽取批量大小为 B 的子数据集 $\{Y_{i_1}, Y_{i_2}, \ldots, Y_{i_B}\}$，计算函数在子数据集上的梯度

$$\nabla f_B(x) = \frac{1}{B}\sum_{i=i_1,\ldots,i_B} \nabla f(x, Y_i).$$

该子数据集 $\{Y_{i_1}, Y_{i_2}, \ldots, Y_{i_B}\}$ 是以等概率 B/N 在每一轮迭代（epoch）中，SGD 有两种方式选取子数据集（i）无重复（无放回）地选取子数据集进行批量梯度方向更新；（ii）可重复（有放回）地选取

子数据集进行批量梯度方向更新。一般来说，实际中我们采用第（i）种方式，因为这种方式易于遍历所有数据集，使得训练得到的模型泛化能力更强。

Algorithm 1 随机梯度算法 SGD

Require: 算法迭代轮数 epochs，步长 (学习率)γ_t，批数据量 B

1: **for** t=1,2,..., epochs **do**
2: **for** i=1,2,...,n **do**
3: 随机选取大小为 B 的批数据集, 计算梯度 $\nabla f_B(x_{t,i})$
4: $x_{t,i+1} = x_{t,i} - \gamma_t \nabla f_B(x_{t,i})$
5: **end for**
6: $x_{t+1,1} = x_{t,n}$
7: **end for**

算法 1是 SGD 的迭代更新流程。其中，epochs 是总的更新轮数，t 是当前更新轮数。学习率 γ_t 是和 t 相关的参数。在每一轮更新中，我们采用无放回方式选取批数据，遍历全部数据集 $\{Y_1, Y_2, \ldots, Y_N\}$ 计算梯度进行更新。因此，共需要 $n = \lceil \frac{N}{B} \rceil$ 次内循环迭代（即算法 1中的第 2 到 5 行），每个内层循环更新即为批量梯度更新。

由于我们不知道整体梯度信息，批次梯度 $-\nabla f_B(x_{t,i})$ 并非下降方向。所以线性搜索在随机梯度法中无法使用。

2.1 参数选取

下面我们介绍 SGD 中常见的参数选取方式。

1. 迭代轮数设置。常见的 epochs 设定为 200 左右。如 Resnet 网络可在 100-300 次迭代得到很好的结果，Transformer 模型则需要更多的迭代，常用 300-500 次迭代。
2. 学习率设置。学习率 γ_t 和迭代轮数相关，随着 t 增大，γ_t 逐渐减小。一般有如下几种流行的方式。
   - γ_t 为一个恒定的常数。这种方式易于调参，但是最终训练误差会停在某个较大的值，所以不推荐此种方式。
   - γ_t 分段减小。例如，当 epochs 为 200 时，可以设定

$$\gamma_t = \begin{cases} 0.1, & t \leq 100 \\ 0.01, & 100 < t \leq 150 \\ 0.001, & 150 < t \leq 200 \end{cases} \tag{12.6}$$

   也就是说，在 $t = 100$ 和 $t = 150$ 时，我们缩小学习率 10 倍。这种分段缩小学习率的方法是最常见的设定方法。需要根据经验选择合适的初始学习率以及缩小阶段点。图 12.3展示的

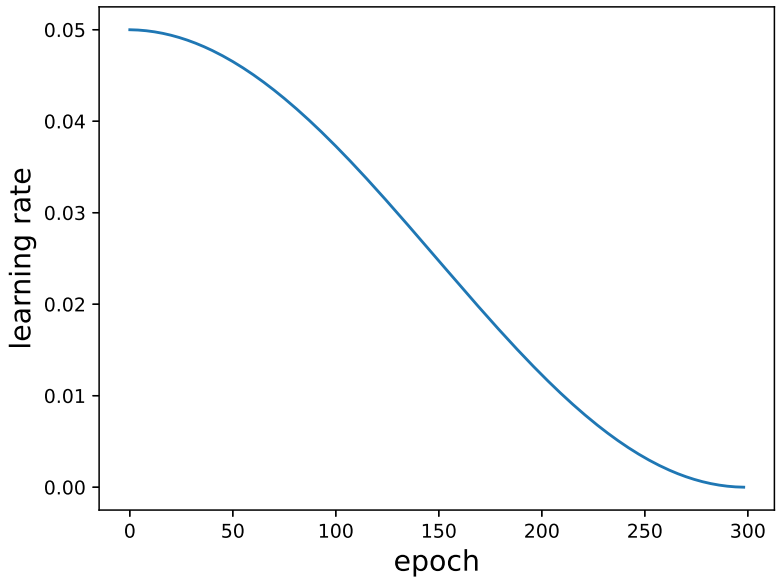


图 12.2: Cosine learning rate 示意图

是某神经网络，采用(12.6)学习率设定，函数值随着迭代轮数的变化情况。我们看到，在 100 和 150 次迭代时，由于学习率缩小 10 倍，函数值也有明显的快速下降。

3. 批量大小设置为了充分利用计算资源，我们尽可能选择最大的批量。因为大批量可以增加计算并行效率，提高训练速度。然而，有研究表明，批数据量大小直接影响泛化误差。因此，我们实际中也为设定一个上限，例如 $B = 128, 256$ 是最常见的选择。

4. 余弦函数学习率。由分段减小学习率函数变化启发，当迭代点接近最优点时，应当选取很小的学习率。训练初期，学习率应缓慢变小，起到热启动的作用。中期则快速下降，达到快速训练的目标。而训练末期又缓慢减小，逐步逼近最优点。利用余弦函数

$$\gamma_t = \gamma_{\min} + \frac{1}{2}(\gamma_{\max} - \gamma_{\min})(1 + \cos(\frac{t}{\text{epochs}}))$$

有效模拟了该想法。其中 $\gamma_{\min}$ 和 $\gamma_{\max}$ 分别是学习率的最小值和最大值。图 12.2所示为 cosine 学习率，以 $\gamma_{\min} = 10^{-5}$ 为最小，$\gamma_{\max} = 0.05$。此方法也是最常见的学习率设定方法。

3 动量随机梯度算法

(随机) 梯度法简单易用，但解决困难的问题（例如优化函数条件数非常大，或者函数是非常差的非凸函数）时，收敛非常缓慢。如图 4(a)所示的函数 $x_1^2 + 10x_2^2$ 等高线，由于函数在 x_2 方向变化更为陡峭，所

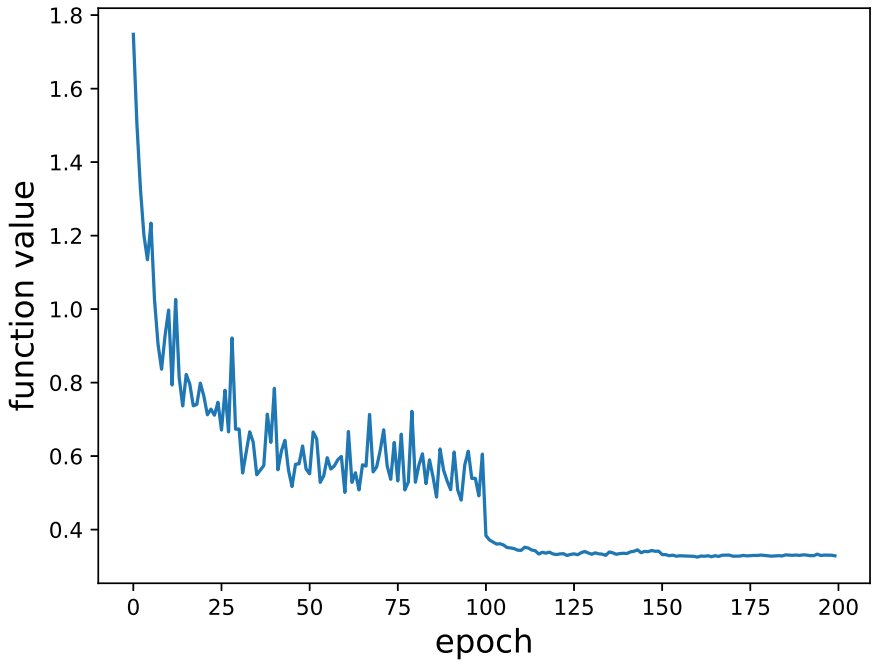


图 12.3: SGD 采用分段步长函数值变化示意图

以梯度更新轨迹在 x_2 方向过于“激进”，过于“信任”当前的梯度信息，而在 x_1 方向变化缓慢，造成了“之”字形轨迹，导致降低了算法收敛速度。动量随机梯度算法（SGD with momentum）可有效缓解这种情况，其描述见算法 2。动量随机梯度法更新方向为动量方向 $v_{t,i+1}$，该算法也多一个动量参数 η。易知，$\eta = 0$ 时，其等价于随机梯度法。实际中，η 常用的值为 0.9。动量迭代 $v_{t,i+1} = \eta v_{t,i} + \gamma_t \nabla f_B(x_{t,i})$ 可理解为利用了“历史”梯度信息，纠正当前过于“激进”的梯度方向。

Algorithm 2 动量随机梯度算法

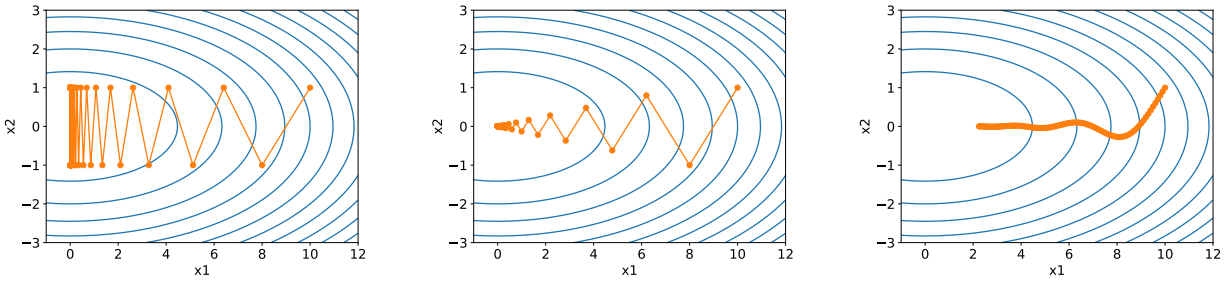
Require: 算法迭代轮数 epochs，学习率 γ_t，批数据量 B，初始动量 $v_{1,1} = 0$, 动量参数 $0 \le \eta < 1$。

for t=1,2,..., epochs **do**
  for i=1,2,...,n **do**
    随机选取大小为 B 的批数据集, 计算梯度 $\nabla f_B(x_{t,i})$
    $v_{t,i+1} = \eta v_{t,i} + \gamma_t \nabla f_B(x_{t,i})$
    $x_{t,i+1} = x_{t,i} - v_{t,i+1}$
  end for
  $x_{t+1,1} = x_{t,n}, v_{t+1,1} = v_{t,n+1}$
end for

具体地，我们可以把迭代 $v_{t,i+1} = \eta v_{t,i} + \gamma_t \nabla f_B(x_{t,i})$ 展开为

$$\begin{aligned} v_{t,i+1} &= \gamma_t \nabla f_B(x_{t,i}) + \eta v_{t,i} \\ &= \gamma_t \nabla f_B(x_{t,i}) + \eta(\gamma_t \nabla f_B(x_{t,i-1}) + \eta v_{t,i-1}) \\ &= \dots \\ &= \gamma_t(\nabla f_B(x_{t,i}) + \eta \nabla f_B(x_{t,i-1}) + \eta^2 \nabla f_B(x_{t,i-2}) + \dots + \eta^j \nabla f_B(x_{t,i-j}) + \dots) \end{aligned}$$

图 4(b)展示的是动量梯度法求解函数 $x_1^2 + 10x_2^2$ 最小值的迭代轨迹。与图 4(a)中的梯度法相对比，动量梯度法轨迹在 x_2 更加缓和，因为动量随机梯度法在历史相同的梯度方向累积起来形成动量，而梯度变化快的方向相互抵消，缓解了振荡现象。总体上来说，动量梯度法只需添加一个参数 η, 其余参数设定方式可与 SGD 相似。因此，动量随机梯度法是训练神经网络非常流行的选择之一。



(a) 梯度法迭代轨迹示意图　(b) 动量梯度法迭代轨迹示意图　(c) Adam 迭代轨迹示意图

图 12.4: 三种优化器在函数 $x_1^2 + 10x_2^2$ 迭代轨迹

4 RMSProp 算法

前面我们知道，学习率很大程度上影响梯度算法的收敛表现。特别是对于如函数 $x_1^2 + 10x_2^2$，自变量 x_1, x_2 梯度方向大小不一致的情况，梯度法出现图 4(a)中“之”字振荡情况。RMSProp(全称为 Root Mean Square Prop, 由 Hinton 等提出）算法对于自变量 x 的不同坐标采用不同学习率，其算法描述见算法 3。

Algorithm 3 RMSProp 算法

Require: 算法迭代轮数 epochs，学习率 α，批数据量 B，初始向量 $s_{1,1}=0$, 参数 $0\le\beta<1$。

for t=1,2,..., epochs **do**
  for i=1,2,...,n **do**
    随机选取大小为 B 的批数据集, 计算梯度 $\nabla f_B(x_{t,i})$
    $s_{t,i+1}=\beta s_{t,i}+(1-\beta)\nabla f_B(x_{t,i})\odot\nabla f_B(x_{t,i})$
    $x_{t,i+1}=x_{t,i}-\frac{\alpha}{\sqrt{s_{t,i+1}+\epsilon}}\odot\nabla f_B(x_{t,i})$
  end for
  $x_{t+1,1}=x_{t,n}, s_{t+1,1}=s_{t,n+1}$
end for

初始 $s_{1,1}$ 是一个和变量 x 维度相同的全 0 向量。迭代 $s_{t,i+1}=\beta s_{t,i}+(1-\beta)\nabla f_B(x_{t,i})\odot\nabla f_B(x_{t,i})$ 采用**指数平均**的方式更新。其中，符号 $\odot$ 表示向量之间按元素相乘。$s_{t,i+1}$ 追踪了函数梯度在不同坐标上的大小，指数平均的方式使得累加的梯度平方项变化更加平滑。具体地，我们考虑一般形式的指数平均迭代

$$y_t=\beta y_{t-1}+(1-\beta)b_t.$$

上式可以展开为

$$\begin{aligned}y_t&=\beta y_{t-1}+(1-\beta)b_t\\&=(1-\beta)b_t+(1-\beta)\beta b_{t-1}+\beta^2y_{t-2}\\&=(1-\beta)b_t+(1-\beta)\beta b_{t-1}+(1-\beta)\beta^2b_{t-2}+\beta^3y_{t-3}\\&=\dots\end{aligned}$$

参数 β 作为权重系数，一般选为 0.9。因此，指数平均赋予了距离当前时间 t 近的参数相对大的权重系数，而逐渐忽略很久之前的参数，从而达到平滑化 y_t 随着时间的变化率。

因此，RMSProp 有效追踪了梯度在不同坐标分量的情况，并且变化率比较平滑。若是累积梯度平方项 s 较大，通过 $\frac{\gamma_t}{\sqrt{s_{t,i+1}+\epsilon}}$ 处理后使用较小的学习率, 这里的 $\epsilon>0$ 是为了防止分母太小，保证数值稳定，例如可设置为 $\epsilon=10^{-6}$。反之，若是某项梯度较小，则使用更大的学习率平衡各个方向学习率。特别注意到，机器学习中部分参数是稀疏的。在这种情况下，使用 RMSProp 将放大稀疏部分的更新量，从而加速收敛。

5 Adam 算法

由指数平均启发，动量随机梯度法也可以写成以下指数平均的形式：

$$v_{t,i+1}=(1-\eta)\frac{\gamma_t}{1-\eta}\nabla f_B(x_{t,i})+\eta v_{t,i}.$$

结合 RMSProp 中的指数平均，可以组合动量估计 v_t 和梯度平方量估计 s_t 设计算法。因此，Adam(全称 adaptive moment estimation) 算法应运而生，其完整描述见算法 4, 它也是最为流行的训练神经网络的方法。β_1，β_2 分别是累积梯度的权重系数和动量权重系数，Adam 算法提出者建议设置为 $\beta_1 = 0.999, \beta_2 = 0.9$。算法中第 6 行为偏差修正：$\hat{v}_{t,i+1} = \frac{v_{t,i+1}}{1-\beta_1^t}$, $\hat{s}_{t,i+1} = \frac{s_{t,i+1}}{1-\beta_2^t}$, 第 7 行更新使用修正后的 $\hat{v}_{t,i+1}$ 和 $\hat{s}_{t,i+1}$ 进行更新，其中 $\epsilon > 0$ 为保证数值稳定的常数，一般设置为 $\epsilon = 10^{-8}$。

Algorithm 4 Adam 算法

Require: 算法迭代轮数 epochs，学习率 α，批数据量 B，初始向量 $s_{1,1} = 0$, 参数 $0 \le \beta_1 < 1$，初始动量 $v_{1,1} = 0$, 动量参数 $0 \le \beta_2 < 1$。

1: **for** t=1,2,..., epochs **do**
2: **for** i=1,2,...,n **do**
3: 随机选取大小为 B 的批数据集, 计算梯度 $\nabla f_B(x_{t,i})$
4: $s_{t,i+1} = \beta_1 s_{t,i} + (1-\beta_1)\nabla f_B(x_{t,i}) \odot \nabla f_B(x_{t,i})$
5: $v_{t,i+1} = \beta_2 v_{t,i} + (1-\beta_2)\nabla f_B(x_{t,i})$
6: 偏差修正：$\hat{v}_{t,i+1} = \frac{v_{t,i+1}}{1-\beta_1^t}$, $\hat{s}_{t,i+1} = \frac{s_{t,i+1}}{1-\beta_2^t}$
7: $x_{t,i+1} = x_{t,i} - \frac{\alpha}{\sqrt{\hat{s}_{t,i+1}}+\epsilon} \odot \hat{v}_{t,i+1}$
8: **end for**
9: $x_{t+1,1} = x_{t,n}, s_{t+1,1} = s_{t,n+1}, v_{t+1,1} = v_{t,n+1}$
10: **end for**

Adam 算法因结合了动量法和 RMSProp 算法，其优势在于学习率 α 更易于调节，算法在各种任务上表现更为稳定。实际中 α 对于不同任务可以采用大致相同量级的初始设定。另外，α 也可像 SGD 中的学习率 γ_t 一样随着时间改变，例如设为 cosine 学习率。图 4(c)也展示了 Adam 在求解 $x_1^2 + 10x_2^2$ 最小值的收敛轨迹。

6 其它随机优化算法

其他随机优化算法还有: 降方差的方法：SVRG、SAG、SAGA。自适应步长的算法：AdaGrad, AdaFactor 等。还有 Google 提出的用强化学习搜索得到的算法 lion。

7 随机梯度算法收敛性

我们只讨论随机梯度算法（SGD）的收敛性。

为了方便，我们记 $f(x, Y_i) = f_i(x)$. 每次以同等概率 $1/N$ 随机选取 $i_k \in \{1, 2, ..., N\}$, SGD 迭代为

$$x^{k+1} = x^k - \alpha_k \nabla f_{i_k}(x^k).$$

假设：我们将在以下假设下分析随机梯度率：

- f 有下界（不一定是凸的）。
- ∇f 是 L-Lipschitz 连续的。
- 对于某个常数 σ^2 和所有 x，有 $E[\|\nabla f_i(x)\|_2] \leq \sigma^2$（“方差” 有界）。

可以放宽噪声边界到更实际的假设 $E[\|\nabla f_i(x_k) - \nabla f(x_k)\|_2] \leq \sigma^2$。这只是在结果中引入了一些额外的项。

由’ 方差’ 有上界可得,

$$\begin{aligned}\mathbb{E}\left[f\left(x^{k+1}\right)\right] &\leq f\left(x^k\right) - \alpha_k\left\|\nabla f\left(x^k\right)\right\|^2 + \alpha_k^2\frac{L}{2}\mathbb{E}\left[\left\|\nabla f_{i_k}\left(x^k\right)\right\|^2\right] \\ &\leq f\left(x^k\right) - \alpha_k\left\|\nabla f\left(x^k\right)\right\|^2 + \alpha_k^2\frac{L\sigma^2}{2}\end{aligned}$$

- 整理可得

$$\alpha_k\left\|\nabla f\left(x^k\right)\right\|^2 \leq f\left(x^k\right) - \mathbb{E}\left[f\left(x^{k+1}\right)\right] + \alpha_k^2\frac{L\sigma^2}{2}.$$

- 对 $k = 1, ...t$ 求和得

$$\sum_{k=1}^{t}\alpha_{k-1}\mathbb{E}\left\|\nabla f\left(x^{k-1}\right)\right\|^2 \leq \sum_{k=1}^{t}\left[\mathbb{E}f\left(x^{k-1}\right) - \mathbb{E}f\left(x^k\right)\right] + \sum_{k=1}^{t}\alpha_{k-1}^2\frac{L\sigma^2}{2}$$

继续处理上述不等式:

$$\sum_{k=1}^{t}\alpha_{k-1}\mathbb{E}\underbrace{\left\|\nabla f\left(x^{k-1}\right)\right\|^2}_{\text{bound by min}} \leq \sum_{k=1}^{t}\underbrace{[\mathbb{E}f\left(x^{k-1}\right) - \mathbb{E}f\left(x^k\right)]}_{\text{telescope}} + \sum_{k=1}^{t}\alpha_{k-1}^2\underbrace{\frac{L\sigma^2}{2}}_{\text{no } k}.$$

- 化简得

$$\min_{k=0,1,\ldots,t-1}\left\{\mathbb{E}\left\|\nabla f\left(x^k\right)\right\|^2\right\}\sum_{k=0}^{t-1}\alpha_k \leq f\left(x^0\right) - \mathbb{E}f\left(x^t\right) + \frac{L\sigma^2}{2}\sum_{k=0}^{t-1}\alpha_k^2.$$

- 因为 $\mathbb{E}f\left(x^k\right) \geq f^*$ ，两边同时除以 $\sum_k \alpha_{k-1}$ 得

$$\min_{k=0,1,\ldots,t-1}\left\{\mathbb{E}\left\|\nabla f\left(x^k\right)\right\|^2\right\} \leq \frac{f\left(x^0\right) - f^*}{\sum_{k=0}^{t-1}\alpha_k} + \frac{L\sigma^2}{2}\frac{\sum_{k=0}^{t-1}\alpha_k^2}{\sum_{k=0}^{t-1}\alpha_k}$$

结果表明:

$$\min_{k=0,1,\ldots,t-1}\left\{\mathbb{E}\left\|\nabla f\left(x^k\right)\right\|^2\right\} \leq \frac{f\left(x^0\right) - f^*}{\sum_{k=0}^{t-1}\alpha_k} + \frac{L\sigma^2}{2}\frac{\sum_{k=0}^{t-1}\alpha_k^2}{\sum_{k=0}^{t-1}\alpha_k}.$$

- 若 $\sigma^2 = 0$, then we could use a constant step-size and would get a $O(1/t)$ rate.

- 由于随机性存在, 收敛速度取决于 $\sum_k \alpha_k^2 / \sum_k \alpha_k$.
- 单调下降步长: set $\alpha_k = \alpha/k$ for some α.
  - Gives $\sum_k \alpha_k = O(\log(t))$ and $\sum_k \alpha_k^2 = O(1)$, so error at t is $O(1/\log(t))$.
- 更大的下降步长: set $\alpha_k = \alpha/\sqrt{k}$ for some α.
  - Gives $\sum_k \alpha_k = O(\sqrt{k})$ and $\sum_k \alpha_k^2 = O(\log k)$, so error at t is $O(\log(t)/\sqrt{t})$.
- - 常数步长: set $\alpha_k = \alpha$ for some α.
  - Gives $\sum_k \alpha_k = k\alpha$ and $\sum_k \alpha_k^2 = k\alpha^2$, so error at t is $O(1/t) + O(\alpha)$

MATH 3011 运筹学 **秋季 2024**

Lecture 13: 无约束优化 牛顿法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{13.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$. f 是二次可微的。

2 牛顿法

设 $f(x)$ 是二次可微实函数，在 x^k 附近作二阶 Taylor 展开近似

$$f(x^k+s)\approx q^k(s)=f(x^k)+{g^k}^Ts+\frac{1}{2}s^TG_ks \tag{13.2}$$

其中 $g^k=\nabla f(x^k), G_k=\nabla^2 f(x^k)$.

将 $q^k(s)$ 极小化便得

$$s=-G_k^{-1}g^k. \tag{13.3}$$

上式给出的搜索方向 $-G_k^{-1}g^k$ 称为**牛顿方向 (Newton Direction).**

Example 13.1 在目标函数是正定二次函数

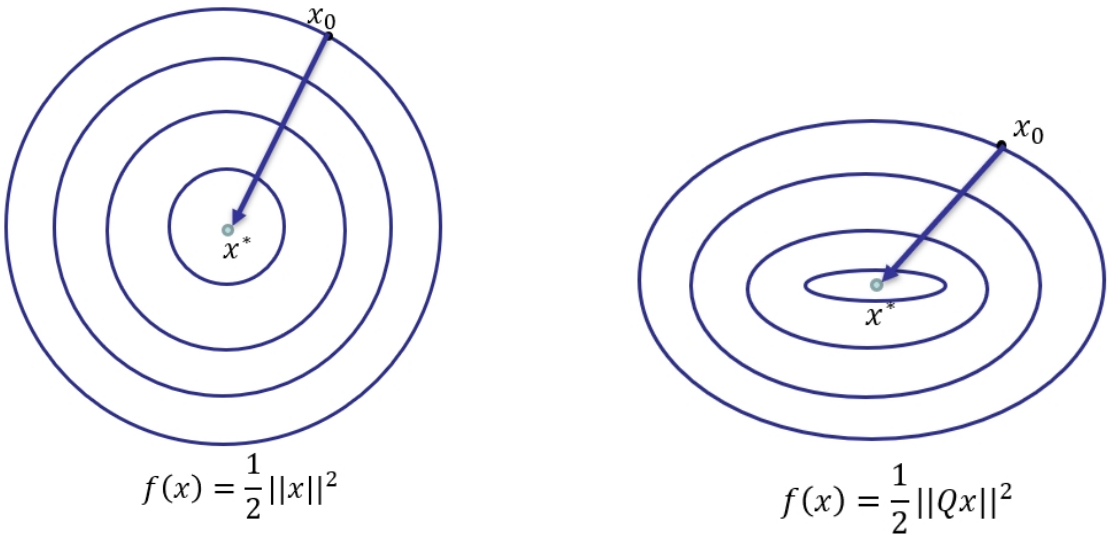
$$f(x)=\frac{1}{2}x^TGx-c^Tx$$

的情况下 *(G 为正定阵)*，对任意的 x 有 $\nabla^2 f(x)=G$.

在第一次迭代里令 $H_0=G^{-1}$，则有

$$d^0=-H_0\nabla f(x^0)=-G^{-1}(Gx^0-c)=-(x^0-x^*).$$

这里，$x^*=G^{-1}c$ 是问题的最优解。若 $x^0\neq x^*$，取步长 $\alpha_0=1$，于是得 $x^1=x^0+\alpha_0d^0=x^*$. 由此知道，不管初始点 x^0 如何取，在一次迭代后即可到达最优解 x^*.



(a) 牛顿法 1　　(b) 牛顿法 2

图 13.1: 牛顿法对于正定二次问题，可以一步得到最优解。

选取步长 $\alpha_k \equiv 1$ 的迭代公式为

$$x^{k+1} = x^k + d^k = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k). \tag{13.4}$$

这就是经典的牛顿迭代法。

2.1 Why is Newton's method good?

对于正定二次函数而言，牛顿法一步即可达到最优解。对于非二次函数，牛顿法并不能保证经有限次迭代求得最优解。但由于目标函数在极小点附近可用二次函数较好地近似，故当初始点靠近极小点时，牛顿法的收敛速度一般会很快。

仿射不变性 (affine-invariant): 令 $A \in \mathbb{R}^{n\times n}$ 为一个可逆矩阵。$f(x)$ 为 $\mathbb{R}^n$ 上的一个函数。考虑如下函数

$$\phi(y) = f(Ay).$$

即对于原来的函数 f，我们选择了 $\mathbb{R}^n$ 新的一组基底 A，得到新坐标下的函数 $\phi(y)$. 牛顿法的关键性质可由下面的结论说明。

Lemma 13.1 令 $\{x_k\}$ 是牛顿法对于 $f(x)$ 的序列，即

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k);$$

令 $\{y_k\}$ 是牛顿法对于 $\phi(y)$ 的序列，即

$$y_{k+1} = y_k - \nabla^2\phi(y_k)^{-1}\nabla\phi(y_k);$$

若 $y_0 = A^{-1}x_0$, 则对于任意 $k \geq 1$, $y_k = A^{-1}x_k$ 。

作业 13.1 证明：Lemma 13.1

该结论说明，牛顿法的迭代点不依赖于基底和度量的选择，因此只依赖于函数的拓扑性质。

2.2 牛顿法求解等式问题

牛顿法最初是为了求解一般等式问题。设 $F : \mathbb{R}^n \to \mathbb{R}^n$, 考虑如下问题：

$$F(x) = 0.$$

迭代为

$$x_{k+1} = x_k - JF(x_k)^{-1}F(x_k). \tag{13.5}$$

对于凸问题(13.1)来说，求解(13.1)的最小值等价于求解下面的等式：

$$\nabla f(x) = 0.$$

记 $F = \nabla f(x)$, 则(13.5)与(13.4)相同。

对于一维问题，即 $F : \mathbb{R} \to \mathbb{R}$，下面的例子展示牛顿法的迭代过程。

Example 13.2 用牛顿法求解 $F(x) = x^3 - 2x + 2 = 0$ 的根。在迭代点 x_k 处，作出函数图像的切线 $l(y) = F(x_k) + F'(x_k)(y - x_k)$，与 x 轴的交点得到下一个迭代点 x_{k+1}, 即 $x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}$. 从初始点 $x_0 = -3$ 和 $x_0 = 1$ 出发, 牛顿法迭代分别如图 *13.2*和 *13.3.* 从 $x_0 = 1$ 出发的点，由于离 $F(x) = 0$ 的根太远，牛顿法不收敛。

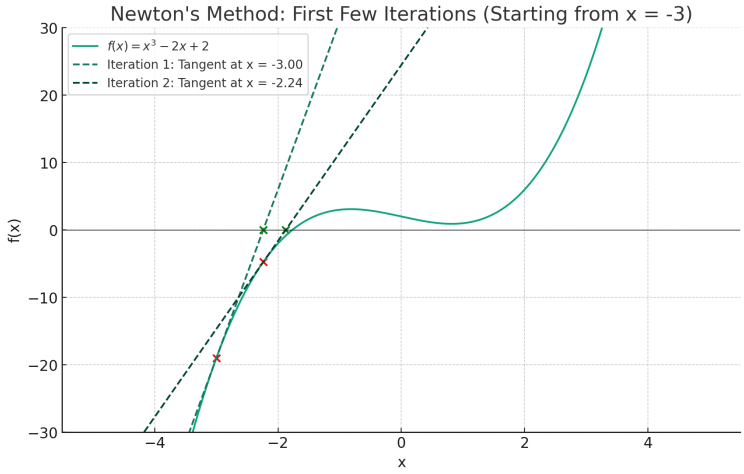


图 13.2: 从 $x_0 = -3$ 出发，收敛到零点

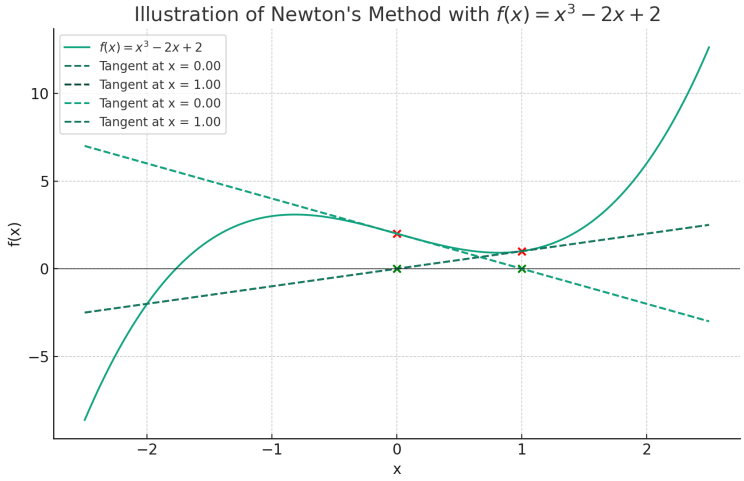


图 13.3: 从 $x_0 = 1$ 出发，牛顿法不收敛, 迭代点困于 0，1 两点。

上例我们知道，初始点若离最优点太远，牛顿法并不收敛。我们下面讨论牛顿法的局部收敛性质。

3 牛顿法的收敛性

Theorem 13.1 假设 f 二阶连续可微, 且存在 x^* 的一个邻域 $N_\delta(x^*)$ 及常数 $L > 0$ 使得

$$\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| \leqslant L\|x - y\|, \quad \forall x, y \in N_\delta(x^*)$$

如果 $f(x)$ 满足 $\nabla f(x^*) = 0, \nabla^2 f(x^*) \succ 0$, 则对于牛顿法有:

- 如果初始点离 x^* 足够近, 则迭代点列 $\left\{x^k\right\}$ 收敛到 x^*;
- $\left\{x^k\right\}$-二次收敛到 x^*;
- $\left\{\left\|\nabla f\left(x^k\right)\right\|\right\}$-二次收敛到 0.

Proof: 根据牛顿法定义以及 $\nabla f\left(x^*\right)=0$, 得

$$\begin{aligned} x^{k+1}-x^* &= x^k-\nabla^2 f\left(x^k\right)^{-1}\nabla f\left(x^k\right)-x^* \\ &= \nabla^2 f\left(x^k\right)^{-1}\left[\nabla^2 f\left(x^k\right)\left(x^k-x^*\right)-\left(\nabla f\left(x^k\right)-\nabla f\left(x^*\right)\right)\right], \end{aligned} \tag{13.6}$$

注意到

$$\nabla f\left(x^k\right)-\nabla f\left(x^*\right)=\int_0^1 \nabla^2 f\left(x^k+t\left(x^*-x^k\right)\right)\left(x^k-x^*\right)\mathrm{d}t,$$

由此

$$\begin{aligned} &\left\|\nabla^2 f\left(x^k\right)\left(x^k-x^*\right)-\left(\nabla f\left(x^k\right)-\nabla f\left(x^*\right)\right)\right\| \\ =&\left\|\int_0^1\left[\nabla^2 f\left(x^k+t\left(x^*-x^k\right)\right)-\nabla^2 f\left(x^k\right)\right]\left(x^k-x^*\right)\mathrm{d}t\right\| \\ \leqslant&\int_0^1\left\|\nabla^2 f\left(x^k+t\left(x^*-x^k\right)\right)-\nabla^2 f\left(x^k\right)\right\|\left\|x^k-x^*\right\|\mathrm{d}t \\ \leqslant&\left\|x^k-x^*\right\|^2\int_0^1 Lt\ \mathrm{d}t=\frac{L}{2}\left\|x^k-x^*\right\|^2. \end{aligned} \tag{13.7}$$

因为 $\nabla^2 f(x)\succ 0$, 由 Lipschitz 连续, 所以 $\exists r>0$, 当 $\|x-x^*\|\leqslant r$ 时有 $\|\nabla^2 f(x)^{-1}\|\leqslant 2\left\|\nabla^2 f\left(x^*\right)^{-1}\right\|$ 成立, 故结合 (13.6)和(13.7), 得到

$$\begin{aligned} &\left\|x^{k+1}-x^*\right\| \\ \leqslant&\left\|\nabla^2 f\left(x^k\right)^{-1}\right\|\left\|\nabla^2 f\left(x^k\right)\left(x^k-x^*\right)-\left(\nabla f\left(x^k\right)-\nabla f\left(x^*\right)\right)\right\| \\ \leqslant&\left\|\nabla^2 f\left(x^k\right)^{-1}\right\|\cdot\frac{L}{2}\left\|x^k-x^*\right\|^2 \\ \leqslant&L\left\|\nabla^2 f\left(x^*\right)^{-1}\right\|\left\|x^k-x^*\right\|^2. \end{aligned}$$

当初始点 x^0 满足 $\|x^0-x^*\|\leqslant\min\left\{\delta,r,\frac{1}{2L\left\|\nabla^2 f\left(x^*\right)^{-1}\right\|}\right\}$ 时, 我们 $\left\|x^{k+1}-x^*\right\|\leq 1/2\left\|x^k-x^*\right\|$. 因此, 迭代点列一直处于邻域 $N_{\hat{\delta}}\left(x^*\right)$ 中, 故 $\left\{x^k\right\}$ 二次收敛到 x^*.

另一方面, 由牛顿方程可知

$$\begin{aligned}\left\|\nabla f\left(x^{k+1}\right)\right\| &= \left\|\nabla f\left(x^{k+1}\right)-\nabla f\left(x^{k}\right)-\nabla^2 f\left(x^{k}\right)d^k\right\| \\ &= \left\|\int_0^1 \nabla^2 f\left(x^k+td^k\right)d^k dt-\nabla^2 f\left(x^k\right)d^k\right\| \\ &\leqslant \int_0^1 \left\|\nabla^2 f\left(x^k+td^k\right)-\nabla^2 f\left(x^k\right)\right\|\left\|d^k\right\|dt \\ &\leqslant \frac{L}{2}\left\|d^k\right\|^2 \leqslant \frac{1}{2}L\left\|\nabla^2 f\left(x^k\right)^{-1}\right\|^2\left\|\nabla f\left(x^k\right)\right\|^2 \\ &\leqslant 2L\left\|\nabla^2 f\left(x^k\right)^{-1}\right\|^2\left\|\nabla f\left(x^k\right)\right\|^2.\end{aligned}$$

这证明梯度的范数二次收敛到 0 .

■

4 修正牛顿法

在式(13.4)的牛顿迭代法里，如果选取的初始点 x^0 不在解 x^* 的附近，那么生成的点列 $\{x^k\}$ 未必收敛于最优解。为了保证算法的全局收敛性，有必要对牛顿法作某些改进。

线搜索牛顿法：

(0) 选取初始点 x^0, 设置终止误差 $\varepsilon>0$, 令 $k:=0$.

(1) 计算 $g^k=\nabla f(x^k)$. 若 $\|g^k\|<\varepsilon$, 停止迭代并输出 x^k.
否则进行第 (2) 步。

(2) 解线性方程组 $\nabla^2 f(x^k)d=-g^k$, 求出牛顿方向 d^k.

(3) 采用一维搜索确定步长因子 α_k, 令 $x^{k+1}=x^k+\alpha_k d^k$, 置 $k:=k+1$, 回到第 (1) 步。

牛顿法面临的另一个主要困难是 Hesse 矩阵 $G_k=\nabla^2 f(x^k)$ 不正定。这时二阶近似模型不一定有极小点，即二次函数 $q^k(s)$ 是无界的。另外，如果初始点离最优点较远，牛顿方向使用步长为 1 不一定能使得函数值减小。

为了克服这些困难，人们提出了很多修正措施。例如 Goldstein & Price (1967) 提出，

$$d^k=\begin{cases}-G_k^{-1}g^k, & \text{if } \cos\theta_k>\eta \\ \\ -g^k, & \text{otherwise}\end{cases} \tag{13.8}$$

上式中，θ_k 是 $-G_k^{-1}g^k$ 与 $-g^k$ 的夹角。即，如果牛顿方向与负梯度方向接近直角，则采用负梯度方向。

如果出现 G_k 非严格正定，或者为了保证牛顿法全局收敛，则有如下修正 Levenberg(1944), Marquardt(1963), Goldfeld et. al(1966)

$$(G_k + \mu_k I)d^k = -g^k \tag{13.9}$$

进一步的参考资料

- Nocedal J, Wright S. Numerical optimization[M]. Springer Science and Business Media, 2006.

MATH 3011 运筹学 **秋季 2024**

Lecture 14: 无约束优化 拟牛顿法

Lecturer: 陈士祥 *Scribes:* 陈士祥

致谢：本节感谢北京大学文再文老师提供的《最优化方法》参考讲义

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{14.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$. f 是一次可微的。

2 拟牛顿法

牛顿法的突出优点是局部收敛很快（具有二阶收敛速率），
但运用牛顿法需要计算二阶导，而且目标函数的 Hesse 矩阵 $\nabla^2 f(x^k)$ 可能非正定，甚至奇异。为了克服这些缺点，
人们提出了拟牛顿法。其基本思想是：用不含二阶导数的矩阵 H_k 近似牛顿法中的 Hesse 矩阵的逆 $\nabla^2 f(x^k)^{-1}$.

由构造近似矩阵的方法不同，有不同的拟牛顿法。

2.1 割线方程

首先，对于 Hesse 矩阵的近似，我们有如下的要求。设第 k 次迭代后得到 x^{k+1}, 将目标函数 $f(x)$ 在 x^{k+1} 处二阶 Taylor 展开：

$$f(x) \approx \quad f(x^{k+1}) + \nabla f(x^{k+1})^T(x - x^{k+1}) + \frac{1}{2}(x - x^{k+1})^T\nabla^2 f(x^{k+1})(x - x^{k+1}),$$

进一步有

$$\nabla f(x) \approx \nabla f(x^{k+1}) + \nabla^2 f(x^{k+1})(x - x^{k+1}),$$

于是令 $x = x^k$ 得

$$\nabla f(x^k) \approx \nabla f(x^{k+1}) + \nabla^2 f(x^{k+1})(x^k - x^{k+1}).$$

记 $s^k = x^{k+1} - x^k$, $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$, 则有

$$\nabla^2 f(x^{k+1}) s^k \approx y^k \quad \text{or} \quad \nabla^2 f(x^{k+1})^{-1} y^k \approx s^k.$$

这样，计算出 s^k 和 y^k 后，可依上式估计在 x^{k+1} 处的 Hessian 矩阵或者 Hessian 的逆矩阵。要求在迭代中构造出 Hesse 矩阵的近似 B_{k+1}, 使其满足

$$B_{k+1} s^k = y^k. \tag{14.2}$$

我们有理由要求在迭代中构造出 Hesse 矩阵逆的近似 H_{k+1}, 使其满足

$$H_{k+1} y^k = s^k. \tag{14.3}$$

通常把式(14.2)和(14.3)称作割线方程，也称为拟牛顿条件。

曲率条件 由于近似矩阵必须保证迭代收敛, 正如牛顿法要求 Hesse 矩阵正定, B^k 正定也是必须的, 即有必要条件

$$\left(s^k\right)^{\mathrm{T}} B^{k+1} s^k > 0 \Longrightarrow \left(s^k\right)^{\mathrm{T}} y^k > 0,$$

Definition 14.1 *曲率条件在迭代过程中满足* $\left(s^k\right)^{\mathrm{T}} y^k > 0, \forall k \in \mathbb{N}^+$.

如果线搜索使用 Powell-Wolfe 准则:

$$\nabla f\left(x^k + \alpha d^k\right)^{\mathrm{T}} d^k \geqslant c_2 \nabla f\left(x^k\right)^{\mathrm{T}} d^k,$$

其中 $c_2 \in (0,1)$. 上式即 $\nabla f\left(x^{k+1}\right)^{\mathrm{T}} s^k \geqslant c_2 \nabla f\left(x^k\right)^{\mathrm{T}} s^k$. 在不等式两边同时减去 $\nabla f\left(x^k\right)^{\mathrm{T}} s^k$, 由于 $c_2 - 1 < 0$ 且 s^k 是下降方向, 因此最终有

$$\left(y^k\right)^{\mathrm{T}} s^k \geqslant \left(c_2 - 1\right) \nabla f\left(x^k\right)^{\mathrm{T}} s^k > 0.$$

拟牛顿法的迭代格式如下:

Algorithm 1 拟牛顿算法 (Quasi-Newton method)

Require: 选取初始点 x^0, 令 $H_0 = I$ 或 $B_0 = I$, $k := 0$.

1: **while** 未满足终止条件: **do**
2: 计算搜索方向 $d^k = -H_k \nabla f(x^k)$ 或者 $d^k = -(B_k)^{-1} \nabla f(x^k)$.
3: 采用一维搜索确定步长因子 α_k, 令 $x^{k+1} = x^k + \alpha_k d^k$.
4: 基于 x^k 到 x^{k+1} 的梯度变化，更新 H_{k+1} 或者 B_{k+1}.
5: $k := k + 1$
6: **end while**

下面我们就来讨论怎样构造及确定满足拟牛顿条件的 Hesse 矩阵逆的近似 H_{k+1}.

2.2 SR1 公式

设 H_k 是第 k 次迭代的 Hesse 矩阵逆的近似，我们希望以 H_k 来产生 H_{k+1}, 即

$$H_{k+1} = H_k + E_k,$$

其中 E_k 是一个低秩的矩阵。

为此，可采用对称秩一 (SR1) 校正

$$H_{k+1} = H_k + auu^T, \ (a \in \mathbb{R}, u \in \mathbb{R}^n).$$

由拟牛顿条件(14.3)知

$$H_{k+1}y^k = H_k y^k + (au^T y^k)u = s^k$$

故 u 必与方向 $s^k - H_k y^k$ 一致，且假定 $s^k - H_k y^k \neq 0$.

不妨取 $u = s^k - H_k y^k$, 此时 $a = \dfrac{1}{u^T y^k}$, 从而得到

$$H_{k+1} = H_k + \frac{(s^k - H_k y^k)(s^k - H_k y^k)^T}{(s^k - H_k y^k)^T y^k}. \tag{14.4}$$

上式称为对称秩一校正。

同理，由 $B_{k+1}s_k = y_k$ 得

$$B_{k+1} = B_k + \frac{uu^T}{u^T s^k}, \quad u = y^k - B_k s^k.$$

SR1 的缺陷：

对称秩一校正的缺点是，不能保持迭代矩阵 H_{k+1} 的正定性。

仅当 H_k 正定以及 $(s^k - H_k y^k)^T y^k > 0$ 时，对称秩一校正才能保持正定性。

证明： 设 $0 \neq w \in \mathbb{R}^n$, 则

$$w^T H_{k+1} w = w^T H_k w + \frac{(u^T w)^2}{u^T y_k} > 0.$$

而这个条件往往很难保证，即使 $(s^k - H_k y^k)^T y^k > 0$ 满足，它也可能很小从而导致数值上的困难。这些都使得对称秩一校正的拟牛顿法应用有较大局限性。

2.3 DFP 公式

采用对称秩二 (SR2) 校正

$$H_{k+1} = H_k + auu^T + bvv^T,$$

并使得拟牛顿条件(14.3)成立，则有

$$H_{k+1}y^k = H_k y^k + (au^T y^k)u + (bv^T y^k)v = s^k.$$

这里 u, v 显然不是唯一确定的，但有一种明显的选择是：

$$\begin{cases} u = s^k, & au^T y^k = 1; \\ v = H_k y^k, & bv^T y^k = -1. \end{cases}$$

因此有

$$H_{k+1} = H_k + \frac{s^k {s^k}^T}{{s^k}^T y^k} - \frac{H_k y^k {y^k}^T H_k}{{y^k}^T H_k y^k}. \tag{14.5}$$

上式称为 DFP(Davidon-Fletcher-Powell) 校正公式，由 Davidon(1959) 提出，后经 Fletcher & Powell(1963) 修改而来。

2.4 BFGS 公式

BFGS 是由 Broyden、Fletcher、Goldfarb、Shanno 4 人从不同角度提供了一种新的拟牛顿公式。

根据割线方程, 将秩二更新的待定参量式代入, 得

$$B^{k+1}s^k = \left(B^k + auu^{\mathrm{T}} + bvv^{\mathrm{T}}\right)s^k = y^k,$$

整理可得

$$\left(a \cdot u^{\mathrm{T}}s^k\right)u + \left(b \cdot v^{\mathrm{T}}s^k\right)v = y^k - B^k s^k.$$

简单的取法是令 $\left(a \cdot u^{\mathrm{T}}s^k\right)u$ 对应 y^k 相等, $\left(b \cdot v^{\mathrm{T}}s^k\right)v$ 对应 $-B^k s^k$ 相等, 即有

$$a \cdot u^{\mathrm{T}}s^k = 1, \quad u = y^k, \quad b \cdot v^{\mathrm{T}}s^k = -1, \quad v = B^k s^k.$$

将上述参量代入割线方程, 即得 BFGS 更新公式

$$B^{k+1} = B^k + \frac{uu^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}u} - \frac{vv^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}v}.$$

利用 SMW 公式以及 $H^k = \left(B^k\right)^{-1}$, 可以推出关于 H^k 的 BFGS 公式.

Definition 14.2 *BFGS 公式 在拟牛顿类算法中, 基于 B^k 的 BFGS 公式为*

$$B^{k+1} = B^k + \frac{y^k\left(y^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k} - \frac{B^k s^k\left(B^k s^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}B^k s^k},$$

基于 H^k 的 BFGS 公式为

$$H^{k+1}=\left(I-\frac{y^k\left(s^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k}\right)^{\mathrm{T}}H^k\left(I-\frac{y^k\left(s^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k}\right)+\frac{s^k\left(s^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k}.$$

推导 H^k 的 BFGS 公式, 我们需要下面的 Sherman-Morrison-Woodbury（SMW）公式。

Sherman-Morrison-Woodbury 公式：设 $A\in\mathbb{R}^{n\times n}$ 是非奇异阵，$u,v\in\mathbb{R}^n$ 是任意向量。若 $1+v^TA^{-1}u\neq 0$, 则 A 的秩一校正 $A+uv^T$ 非奇异，且其逆可以表示为

$$(A+uv^T)^{-1}=A^{-1}-\frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}. \tag{14.6}$$

Sherman-Morrison-Woodbury 推广公式：设 $A\in\mathbb{R}^{n\times n}$ 是非奇异阵，$U\in\mathbb{R}^{n\times k},V\in\mathbb{R}^{n\times k}$ 是任意矩阵。若 $I_k+V^TA^{-1}U$ 可逆, 则 $A+UV^T$ 非奇异，且其逆可以表示为

$$(A+UV^T)^{-1}=A^{-1}-A^{-1}U(I_k+V^TA^{-1}U)^{-1}V^TA^{-1}. \tag{14.7}$$

推导 H^k 的 BFGS 公式之提示:
对于可逆矩阵 $B\in\mathbb{R}^{n\times n}$ 与矩阵 $U\in\mathbb{R}^{n\times m},V\in\mathbb{R}^{n\times m}$, 根据 SMW 公式(14.7)为:

$$\left(B+UV^{\mathrm{T}}\right)^{-1}=B^{-1}-B^{-1}U\left(I+V^{\mathrm{T}}B^{-1}U\right)^{-1}V^{\mathrm{T}}B^{-1}.$$

在 BFGS 的推导中, 关于 B^k 的更新公式为:

$$B_{k+1}=B_k+\frac{y_ky_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}-\frac{B_ks_k\left(B_ks_k\right)^{\mathrm{T}}}{s_k^{\mathrm{T}}B_ks_k}=B_k+\begin{pmatrix}-\frac{B_ks_k}{s_k^{\mathrm{T}}B_ks_k} & \frac{y_k}{s_k^{\mathrm{T}}y_k}\end{pmatrix}\begin{pmatrix}s_k^{\mathrm{T}}B_k\\ y_k^{\mathrm{T}}\end{pmatrix}.$$

对照 SMW 公式(14.7), 令式中 $B=B_k$, 且

$$U_k=\begin{pmatrix}-\frac{B_ks_k}{s_k^{\mathrm{T}}B_ks_k} & \frac{y_k}{s_k^{\mathrm{T}}y_k}\end{pmatrix},\quad V_k=\begin{pmatrix}B_ks_k & y_k\end{pmatrix},$$

此时公式的左端就等于 B_{k+1}^{-1}, 且右端只需计算一个 2 阶矩阵的逆. 假设 $B_k^{-1}=H_k$, 由 SMW 公式就得到

$$H_{k+1}=\left(B_k+U_kV_k^{\mathrm{T}}\right)^{-1}=\left(I-\frac{s_ky_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}\right)H_k\left(I-\frac{y_ks_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}\right)+\frac{s_ks_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}.$$

作业 14.1 利用 *SMW* 公式，由 $H_{k+1}^{(DFP)}$ 推导 $B_{k+1}^{(DFP)}$.

BFGS 公式的有效性

BFGS 公式产生的 B^{k+1} 或 H^{k+1} 是否正定呢?

Theorem 14.1 *BFGS 公式使拟牛顿矩阵正定的充分条件使用秩二更新公式从 B^k 或 H^k 更新 B^{k+1} 或 H^{k+1}, 拟牛顿矩阵正定的充分条件可以是:*
(1) B^k 或 H^k 正定;
(2) 满足曲率条件 $\left(s^k\right)^{\mathrm{T}} y^k > 0, \forall k \in \mathbb{N}^+$.

证明上述定理, 只需要从基于 H^k 的 BFGS 公式分析即可, 从而得到 H^{k+1} 和其逆 B^{k+1} 均正定.

因为在确定步长时使用某一 Wolfe 准则线搜索即可满足曲率条件, 因此 BFGS 公式产生的拟牛顿矩阵有望保持正定.

2.4.1 从优化意义理解 BFGS 公式

基于 H^k 的 BFGS 格式恰好是优化问题

$$\begin{aligned} \min_{H} & \left\|H - H^k\right\|_W, \\ \text{s.t.} \quad & H = H^{\mathrm{T}}, \\ & Hy^k = s^k. \end{aligned} \tag{14.8}$$

的解. 上式中 $\|\cdot\|_W$ 是加权范数, 定义为

$$\|H\|_W = \left\|W^{1/2} H W^{1/2}\right\|_F,$$

且 W 满足割线方程, 即 $Ws^k = y^k$. 使用 $\|\cdot\|_W$ 可以让得到的拟牛顿公式同样满足 仿射不变性（请回顾：“牛顿法为什么好”-牛顿法的仿射不变性质）。注意 $Hy^k = s^k$ 是割线方程, 因此优化问题的意义是在满足割线方程的对称矩阵中找到距离 H^k 最近的矩阵 H 作为 H^{k+1}. 因此我们可以进一步认知, BFGS 格式更新的拟牛顿矩阵是正定对称的, 且在满足割线方程的条件下采取的是最佳逼近策略.

2.4.2 从优化意义理解 DFP 公式

有了 BFGS 公式的优化意义做铺垫, 讨论 DFP 公式的优化意义显得十分简单. 利用对偶性质, 基于 B^k 的 DFP 格式将是优化问题

$$\begin{aligned} \min_{B} \quad & \left\|B - B^k\right\|_W, \\ \text{s.t.} \quad & B = B^{\mathrm{T}}, \\ & Bs^k = y^k. \end{aligned}$$

的解. 上式中 $\|\cdot\|_W$ 是加权范数, 定义为

$$\|B\|_W = \left\|W^{1/2} B W^{1/2}\right\|_F,$$

λ	0.1	0.01	10^{-4}	10^{-8}
10	5	6	8	10
100	7	8	10	12
10^4	12	13	15	17
10^6	17	18	20	22
10^9	24	25	27	29

表 14.1: BFGS 方法的迭代次数

λ	0.1	0.01	10^{-4}	10^{-8}
10	10	13	16	19
30	25	32	37	40
100	80	99	107	111
300	237	290	307	313
10^3	787	958	1006	1014

表 14.2: DFP 方法的迭代次数

且 W 满足另一割线方程, 即 $Wy^k = s^k$. 注意 $Bs^k = y^k$ 是另一割线方程, 因此优化问题的意义是在满足割线方程的对称矩阵中找到距离 B^k 最近的矩阵 B 作为 B^{k+1}.

DFP 公式的缺陷

尽管 DFP 格式与 BFGS 对偶, 但从实际效果而言, DFP 格式的求解效率整体上不如 BFGS 格式. M.J.D. Powell 曾求解问题

$$\min_{x\in\mathbb{R}^2} f(x) = \frac{1}{2}\|x\|_2^2.$$

设置初始值

$$B^0 = \begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}, \quad x_1 = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix},$$

其中 $\tan^2\psi = \lambda$. 当误差阈 $\epsilon = 10^{-4}$ 时, 分别取 λ 为不同的值, 使用 BFGS 算法与 DFP 算法所产生的迭代步数分别如表 14.1和 14.2所示. 由此看出, 在本问题中, BFGS 算法的求解效率要远高于 DFP 算法. (参考文献: Powell M J D. How bad are the BFGS and DFP methods when the objective function is quadratic?[J]. Mathematical Programming, 1986, 34(1): 34-47.)

2.4.3 BFGS 另一种解释

$X = B_{k+1}$ 是下面优化问题的最优解：

$$\begin{aligned} \min_{X} \quad & \mathrm{Tr}(B_k^{-1}X) - \log\det(B_k^{-1}X) - n \\ \text{s.t.} \quad & Xs_k = y_k, X^T = X. \end{aligned} \tag{14.9}$$

上述问题中的目标函数，是概率分布 $N(0, X)$ 和 $N(0, B_k)$ 的相对熵。

作业 14.2 证明以下结论：

1. 问题(14.10) 目标函数值是非负的。目标值为 0 仅当 $X = H_k$.
2. 证明 BFGS 的迭代公式 B_{k+1} 是该问题的最优解。

(提示：$-\log\det(X)$ 是关于 X 的凸函数，并且 $\frac{\partial \log\det X}{\partial X} = X^{-T}$, $\frac{\partial Tr(C^T X)}{\partial X} = C$.)

对于 DFP 公式： $X = H_{k+1}$ 是下面优化问题的最优解：

$$\begin{aligned} \min_{X} \quad & \mathrm{Tr}(H_k^{-1}X) - \log\det(H_k^{-1}X) - n \\ \text{s.t.} \quad & Xy_k = s_k, X^T = X. \end{aligned} \tag{14.10}$$

上述问题中的目标函数，设 $Y = X^{-1}$ 有

$$Tr(H_k^{-1}X) - \log\det(H_k^{-1}X) - n = Tr(B_k Y^{-1}) - \log\det(B_k Y^{-1}) - n$$

是概率分布 $N(0, B_k)$ 和 $N(0, Y)$ 的相对熵。

2.5 BFGS 算法收敛性

Theorem 14.2 *BFGS 全局收敛性：设初始矩阵 B^0 是对称正定矩阵，目标函数 $f(x)$ 是二阶连续可微函数，下水平集*

$$\mathcal{L} = \left\{x \in \mathbb{R}^n \mid f(x) \leqslant f\left(x^0\right)\right\}$$

凸，且存在 $m, M \in \mathbb{R}^+$ 使得对 $\forall z \in \mathbb{R}^n, x \in \mathcal{L}$ 满足

$$m\|z\|^2 \leqslant z^{\mathrm{T}}\nabla^2 f(x) z \leqslant M\|z\|^2$$

(即 $z^{\mathrm{T}}\nabla^2 f(x) z$ 被 $\|z\|$ 控制)，那么 BFGS 格式结合 Wolfe 线搜索的拟牛顿算法全局收敛到 $f(x)$ 的极小值点 x^*.

局部收敛速度：进一步假设 f 的 Hessian 在最优点邻域内是 Lipschitz 连续，那么 BFGS 的迭代点最终超线性收敛到最优点 x^*.

我们略过证明。

3 有限内存 BFGS 方法

基本思路: 标准的拟牛顿近似矩阵的更新公式可以记为

$$B^{k+1} = g\left(B^k, s^k, y^k\right), s^k = x^{k+1} - x^k, y^k = \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right).$$

若变量维度太大，那么存储 H_k 需要大量内存，并且更新的计算量为 $O(n^2)$. 如果只保存最近的 m 组数据, 那么迭代公式可以写成

$$B^{k+1} = g\left(g\left(\cdots g\left(B^{k-m+1}, s^{k-m+1}, y^{k-m+1}\right)\right)\right).$$

考虑 BFGS 方法:

$$d^k = -\left(B^k\right)^{-1}\nabla f\left(x^k\right) = -H^k\nabla f\left(x^k\right).$$

重写 BFGS 更新公式为

$$H^{k+1} = \left(V^k\right)^{\mathrm{T}} H^k V^k + \rho_k s^k \left(s^k\right)^{\mathrm{T}},$$

其中

$$\rho_k = \frac{1}{\left(y^k\right)^{\mathrm{T}} s^k}, \quad V^k = I_{n\times n} - \rho_k y^k \left(s^k\right)^{\mathrm{T}}.$$

将上式递归地展开 m 次, 即

$$\begin{aligned} H^k = &\left(\prod_{j=k-m}^{k-1} V^j\right)^{\mathrm{T}} H^{k-m} \left(\prod_{j=k-m}^{k-1} V^j\right) + \\ &\rho_{k-m}\left(\prod_{j=k-m+1}^{k-1} V^j\right)^{\mathrm{T}} s^{k-m}\left(s^{k-m}\right)^{\mathrm{T}}\left(\prod_{j=k-m+1}^{k-1} V^j\right) + \\ &\rho_{k-m+1}\left(\prod_{j=k-m+2}^{k-1} V^j\right)^{\mathrm{T}} s^{k-m+1}\left(s^{k-m+1}\right)^{\mathrm{T}}\left(\prod_{j=k-m+2}^{k-1} V^j\right) + \cdots + \\ &\rho_{k-1} s^{k-1}\left(s^{k-1}\right)^{\mathrm{T}}. \end{aligned}$$

为了节省内存, 我们只展开 m 次, 利用 H^{k-m} 进行计算, 即可求出 H^{k+1}. 下面介绍一种不计算 H^k, 只利用展开式计算 $d^k = -H^k\nabla f\left(x^k\right)$ 的巧妙算法: 双循环递归算法. 它利用迭代式的结构尽量节省计算 d^k 的开销.

由于我们只需要得到 $-d^k = H^k \nabla f(x_k)$，将等式两边同右乘 $\nabla f\left(x^k\right)$. 观察等式右侧需要计算

$$V^{k-1}\nabla f\left(x^k\right), \cdots, V^{k-m}\cdots V^{k-1}\nabla f\left(x^k\right).$$

这些计算可以递归地进行. 同时在计算 $V^{k-l}\cdots V^{k-1}\nabla f\left(x^k\right)$ 的过程中, 可以计算上一步的 $\rho_{k-l}\left(s^{k-l}\right)^{\mathrm{T}}\left[V^{k-l+1}\cdots V^{k-1}\nabla f\left(x^k\right)\right]$, 这是一个标量. 记

$$q = V^{k-m}\cdots V^{k-1}\nabla f\left(x^k\right),$$
$$\alpha_{k-l} = \rho_{k-l}\left(s^{k-l}\right)^{\mathrm{T}}\left[V^{k-l+1}\cdots V^{k-1}\nabla f\left(x^k\right)\right],$$

因此递归公式可化为如下的形式:

$$H^k\nabla f\left(x^k\right) = \left(\prod_{j=k-m}^{k-1} V^j\right)^{\mathrm{T}} H^{k-m}q + \left(\prod_{j=k-m+1}^{k-1} V^j\right)^{\mathrm{T}} s^{k-m}\alpha_{k-m} + \cdots + s^{k-1}\alpha_{k-1}$$

在双循环递归算法中, 除了上述第一个循环递归过程 (自下而上) 外, 还有以下第二个循环递归过程。我们需要在公式中自上而下合并每一项. 以前两项为例, 它们有公共的因子 $\left(V^{k-m+1}\cdots V^{k-1}\right)^{\mathrm{T}}$, 提取后可以将前两项写为 (注意将 V^{k-m} 的定义回代)

$$\begin{aligned}&\left(V^{k-m+1}\cdots V^{k-1}\right)^{\mathrm{T}}\left[\left(V^{k-m}\right)^{\mathrm{T}} r + \alpha_{k-m}s^{k-m}\right]\\ &= \left(V^{k-m+1}\cdots V^{k-1}\right)^{\mathrm{T}}\left(r + \left(\alpha_{k-m} - \beta\right)s^{k-m}\right),\end{aligned}$$

这正是第二个循环的迭代格式. 注意合并后原递归式的结构仍不变, 因此可以递归地计算下去. 最后, 变量 r 就是我们期望的结果 $H^k\nabla f\left(x^k\right)$.

Algorithm 2 算法 L-BFGS 双循环递归

Require: 初始化 $q \leftarrow \nabla f\left(x^k\right)$.

Ensure: r, 即 $H^k\nabla f\left(x^k\right)$.

1: **for** $i = k-1, \cdots, k-m$ **do**
2: 计算并保存 $\alpha_i \leftarrow \rho_i \left(s^i\right)^{\mathrm{T}} q$.
3: 更新 $q \leftarrow q - \alpha_i y^i$
4: **end for**
5: 初始化 $r \leftarrow \hat{H}^{k-m}q$, 其中 $\hat{H}^{k-m}$ 是 H^{k-m} 的近似矩阵.
6: **for** $i = k-m, \cdots, k-1$ **do**
7: $\beta \leftarrow \rho_i \left(y^i\right)^{\mathrm{T}} r$
8: 更新 $r \leftarrow r + \left(\alpha_i - \beta\right)s^i$
9: **end for**

L-BFGS 双循环递归算法约需要 $4mn$ 次乘法运算, $2mn$ 次加法运算; 若近似矩阵 $\hat{H}^{k-m}$ 是对角矩阵, 则额外需要 n 次乘法运算. 由于 m 不会很大, 因此算法的复杂度是 $\mathcal{O}(mn)$. 算法需要的额外存储为临时变量 α_i, 其大小是 $\mathcal{O}(m)$.

进一步的参考资料

- R. Fletcher, Practical Methods of Optimization (2nd Edition). John Wiley & Sons, 1987.
- D. C. Liu and J. Nocedal, On the Limited Memory Method for Large Scale Optimization. Mathematical Programming B, 45(3), pp. 503-528, 1999.
- Nocedal, Jorge, and Stephen J. Wright, eds. Numerical optimization. New York, NY: Springer New York, 1999.

MATH 3011 运筹学 **秋季 2024**

Lecture 15: 无约束优化 共轭梯度法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{15.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$. f 是一次可微的。

2 线性共轭梯度法

我们首先考虑二次问题:

$$\min f(x) = \frac{1}{2}x^TAx - b^Tx, \tag{15.2}$$

这里，$A \in \mathbb{R}^{n\times n}$。最简单的情形，如果 A 是对角矩阵，并且考虑下图中的二维情况下，我们可以每次沿着坐标轴对二次函数求最小值。这样，只需要两步即可得到最优解。一般地，若 A 的对角矩阵，对角元为 $\lambda_1, \lambda_2, \ldots, \lambda_n$, 那么

$$f(x) = \sum_{i=1}^{n}\left(\frac{1}{2}\lambda_i x_i^2 - b_i x_i\right).$$

因此，我们可以每次可以固定一个维度 k，对 $\frac{1}{2}\lambda_k x_k^2 - b_k x_k$ 极小化。只需 n 步即可得到最优值。

但是，若 A 不是对角矩阵，若是仍然沿着坐标轴最小化函数，如图 15.2，迭代点不会很快收敛。

2.1 共轭方向

定义： 设 A 是 $n\times n$ 正定阵。对于 $\mathbb{R}^n$ 中的任一组非零向量 $\{p_0, p_1, \cdots, p_k\}$, 如果 ${p_i}^TAp_j = 0 (i \neq j)$, 则称 $p_0, p_1, \cdots, p_k$ 是关于 A 共轭的。

共轭是正交概念的推广，当取 $A = I$ 时，共轭即为正交。

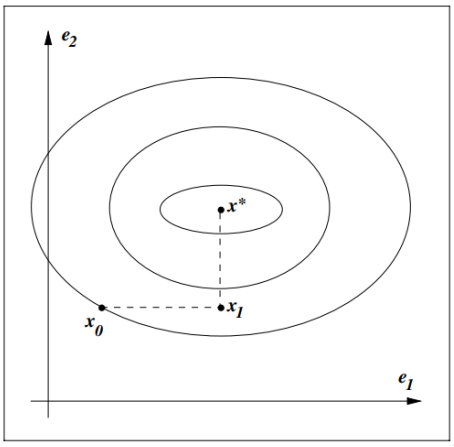


图 15.1: 2 维对角矩阵 A 示例。图片来源: Numerical optimization, J. Nocedal, S. J. Wright.

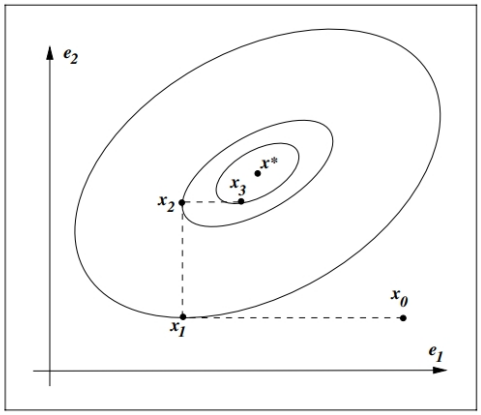


图 15.2: 2 维非对角矩阵 A 示例。图片来源: Numerical optimization, J. Nocedal, S. J. Wright.

假设 $\{p_0, p_1, \cdots, p_{n-1}\}$ 是给定的关于 A 的共轭方向。令 $S = [p_0, p_1, \cdots, p_{n-1}]$. 若我们对 $f(x)$ 做变换:

$$\hat{f}(\hat{x}) = f(S\hat{x}) = \frac{1}{2}\hat{x}^T(S^TAS)\hat{x} - (Sb)^T\hat{x}.$$

由共轭方向的定义，我们知道矩阵 S^TAS 是对角正定矩阵。因此，对于 $\hat{f}(\hat{x})$ 我们可以在 $\hat{x}$ 的各个坐标轴方向 $\{e_1, e_2, ...e_n\}$ 求解最小化问题 (对应于 x 的坐标在 $p_0, p_1, \cdots, p_{n-1}$)，最终得到最优解。

线性共轭梯度方法便是通过构造出 A 的共轭方向 $\{p_0, p_1, \cdots, p_{n-1}\}$，快速求解(15.2)。当然，矩阵 A 的特征根方向是共轭方向，但是若 A 的规模太大，特征根分解很慢。也可以通过改进 Gram-Schmidt 正交化得到共轭方向，但是其需要存储所有的 $p_0, p_1, ...p_{n-1}$。线性共轭梯度法的优点在于，在某个迭代 k 时，只需要 p_{k-1}, 即可构造出 p_k，并不需要所有的 $p_0, ..., p_{k-2}$.

线性共轭梯度方法 由于优化二次函数(15.2)等价于求解方程 $Ax = b$，该方法叫做“线性”共轭梯度法。

而名字中“梯度”时来源于第一个方向 p_0 取最速下降方向, 即 $p_0 = -\nabla f(x_0)$. 具体的来说，线性共轭梯度算法如下

Algorithm 1 线性共轭梯度方法 CG

Require: 初始点 x_0

$r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$

while $r_k \neq 0$ **do**

  $\alpha_k \leftarrow \frac{-r_k^T p_k}{p_k^T A p_k}$

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

  $r_{k+1} \leftarrow Ax_{k+1} - b$

  $\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$

  $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$

  $k \leftarrow k + 1$

end while

我们下面讲解为何如此选取 $\alpha_k, x_{k+1}, r_{k+1}, p_{k+1}$。

(0) 给定正定阵 A, 选取初始点 x_0, $p_0 = -\nabla f(x_0)$ 保证第一步为下降方向。

记

$$r_k = Ax_k - b = \nabla f(x_k) \tag{15.3}$$

(1) 由于我们想每步迭代在 p_k 方向最小化函数值,即求精确的一维搜索步长 α_k, 即 $\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha p_k)$. 由此可得

$$\alpha_k = \frac{-r_k^T p_k}{p_k^T A p_k}$$

(2) 更新迭代点 $x_{k+1} = x_k + \alpha_k p_k$, 并构造 p_{k+1} 是负梯度方向和前一个共轭方向 p_k 的线性组合，即

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k. \tag{15.4}$$

由于 $p_{k+1}^T A p_k = 0$, 可得

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

虽然构造 p_{k+1} 的方式, 仅保证了 p_k 与 p_{k+1} 互为共轭, 但是如下定理告诉我们, p_{k+1} 与所有的 $p_0, ..., p_k$ 均共轭。

Theorem 15.1 (线性共轭梯度法性质) *设线性共轭梯度法的第 k 步迭代的结果 x_k 不是问题 (15.2)的解，那么有以下结论成立*

1. $\text{span}(r_0, r_1, \cdots, r_k) = \text{span}\left(r_0, Ar_0, \cdots, A^k r_0\right)$

2. $\text{span}\,(p_0, p_1, \cdots, p_k) = \text{span}\left(r_0, Ar_0, \cdots, A^k r_0\right)$

3. $r_k^T p_i = 0, \forall i < k$

4. $p_k^T A p_i = 0, \forall i < k$

5. $r_k^T r_i = 0, \forall i < k$

作业 15.1 证明上述定理。

下面的定理阐述了线性共轭算法的另一个重要性质。

Theorem 15.2 *严格凸二次函数* $f(x) = \frac{1}{2}x^T Ax - b^T x$*, 共轭方向法执行精确一维搜索，则每步迭代点* x_{k+1} *是* $f(x)$ *在空间*

$$\mathcal{V}_k = \{x \mid x = x_0 + \sum_{j=0}^{k} \beta_j p_j, \forall \beta_j \in \mathbb{R}\}$$

中的唯一极小点。因此，最多需要 n *步，* x_k *收敛到最优点* $x^* = A^{-1}b$*.*

Proof: 由共轭方向的定义知，$\{p_0, p_1, \cdots, p_{n-1}\}$ 线性无关。若 x_n 是 $\mathcal{V}_{n-1}$ 上的最小值，那么是 $\mathbb{R}^n$ 上的最小值。

若要证明 x_{k+1} 是 $\mathcal{V}_k$ 上最小值，下面只要证：对所有 $k < n$ 成立

$${r_{k+1}}^T p_j = 0,\ j = 0, 1, \cdots, k.$$

即在点 x_{k+1} 处的函数梯度 $r_{k+1} = \nabla f(x_{k+1})$ 与子空间 $\text{span}\{p_0, p_1, \cdots, p_k\}$ 正交。

由线性共轭梯度梯度法性质定理知，对 $\forall j = 0, 1, ..., k$ 有如下关系成立

$$r_{k+1}^T p_j = 0.$$

因此，得证。 ■

进一步，由以下关系，可以得到更加实用的线性共轭梯度算法。该形式中，我们减少了 2 组向量内积(想一想，为什么)，因此更加有效率。

首先 $p_k^T r_k = (-r_k + \beta_k p_{k-1})^T r_k = -r_k^T r_k$. 又因为

$$r_{k+1} - r_k = A(x_{k+1} - x_k) = \alpha_k A p_k. \tag{15.5}$$

故(15.5)两边同时乘以 r_{k+1}^T 和 p_k 分别得 $\alpha_k r_{k+1}^T A p_k = r_{k+1}^T r_{k+1}, \quad \alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$

Algorithm 2 线性共轭梯度方法 CG (Practical form)

Require: 初始点 x_0

$r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$

while $r_k \neq 0$ **do**

  $\alpha_k \leftarrow \frac{-r_k^T p_k}{p_k^T A p_k} \Longleftrightarrow \alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}$;

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

  $r_{k+1} \leftarrow Ax_{k+1} - b$

  $\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k} \Longleftrightarrow \beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$

  $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$

  $k \leftarrow k + 1$

end while

2.2 线性共轭梯度法结论

结论 1: 若 A 有特征根 $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, 我们有

$$\|x_{k+1} - x^*\|_A^2 \leq (\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1})^2 \|x_0 - x^*\|_A^2.$$

该结论说明共轭梯度法的收敛速度和特征根的分布有关。若 A 有 m 个较大的特征根，剩余的 $n-m$ 的特征根都约等于 1。令 $\epsilon = \lambda_{n-m} - \lambda_1$. 那么，只有在 $m+1$ 步后，我们有

$$\|x_{m+1} - x^*\|_A \approx \epsilon \|x_0 - x^*\|_A.$$

$m+1$ 步后共轭梯度法收敛较快，而之前都比较慢。

此外，我们还有如下结论：

结论 2: 记 $\kappa = \|A\|_2 \|A^{-1}\|$ 为矩阵 A 的条件数，那么

$$\|x_k - x^*\|_A \leq 2(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1})^k \|x_0 - x^*\|_A.$$

相较于梯度法，系数 $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = 1 - \frac{2}{\sqrt{\kappa}}$ 比梯度法的 $\frac{\kappa-1}{\kappa+1} = 1 - \frac{2}{\kappa}$ 更小. 因此收敛速度更快。

2.3 The preconditioned conjugate gradient method 预条件法

预条件方法的主要思路是对问题做一个线性变换，使得新线性系统矩阵的条件数降低, 并且矩阵的特征根分布更加均匀。对 x 做变换 $\hat{x} = Cx$, 这里 C 是一个可逆矩阵。考虑最小化

$$g(\hat{x}) = f(C^{-1}\hat{x}) = \frac{1}{2}\hat{x}^T (C^{-T} A C^{-1})\hat{x} - (C^{-T} b)^T \hat{x}.$$

通过选取适当的 C, 可以使得新矩阵 $C^{-T}AC^{-1}$ 特征根分布更加均衡, 即理想情况下, 让 $C^{-T}AC^{-1} \approx I$。例如，PDE 数值求解中，可以通过取 $C^T C \approx A$, 即 $C^T C$ 为 A 的三对角部分。

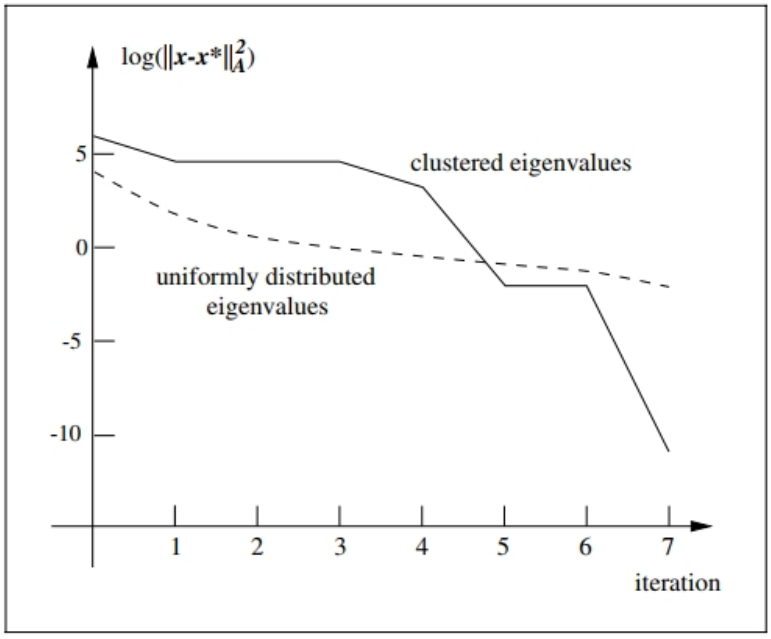


图 15.3: 共轭梯度法：共轭梯度法收敛图像：虚线表示矩阵 A 有较均匀特征根和集中的特征根，此时收敛较慢。实线对应特征根分布更为集中的情况，收敛更快。(图片来源：Numerical optimization)

3 非线性共轭梯度法

将共轭梯度法推广到非二次函数的极小化问题，其迭代为

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

步长 α_k 由精确或者非精确一维搜索决定，p_{k+1} 的构造如下：

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k.$$

同样，这里 $r_k = \nabla f(x_k)$.

有如下 4 种最为经典的 β_k 的选取方式

$$\beta_k := \frac{{r_{k+1}}^T r_{k+1}}{{r_k}^T r_k} \quad (\text{Fletcher} - \text{Reeves})$$

$$\beta_k := \frac{{r_{k+1}}^T (r_{k+1} - r_k)}{{p_k}^T (r_{k+1} - r_k)} \quad (\text{Hestenes} - \text{Stiefel})$$

$$\beta_k := \frac{{r_{k+1}}^T (r_{k+1} - r_k)}{{r_k}^T r_k} \quad (\text{Polak} - \text{Ribiere} - \text{Polyak})$$

$$\beta_k := \frac{{r_{k+1}}^T r_{k+1}}{{p_k}^T (r_{k+1} - r_k)} \quad (\text{Dai} - \text{Yuan})$$

Lemma 15.1 设 $\{x_k\}$ 为使用 *Fletcher-Reeves* 格式 *(*也即 $\beta_{k+1} = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)}$ *)* 非线性共轭梯度法得到的迭代点序列。α_k 为非精确线搜索强 *Wolfe* 条件得到的步长，*Wolfe* 条件的系数满足 $0 < c_1 < c_2 < 0.5$ ，那么搜索方向 p_k 满足

$$-\frac{1}{1 - c_2} \leq \frac{\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|^2} \leq \frac{2c_2 - 1}{1 - c_2}. \tag{15.6}$$

因此，p_k 为下降方向。

Proof:

我们用归纳法证明，首先，对于 $k = 0$, 因为 $p_0 = -\nabla f(x_0)$ 显然成立。如果对于 k 成立，对于 $k + 1$, 注意 $p_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} p_k$ ，有

$$\frac{\nabla f(x_{k+1})^T p_{k+1}}{\|\nabla f(x_{k+1})\|^2} = -1 + \beta_{k+1} \frac{\nabla f(x_{k+1})^T p_k}{\|\nabla f(x_{k+1})\|^2} = -1 + \frac{\nabla f(x_{k+1})^T p_k}{\|\nabla f(x_k)\|^2}$$

最后的一项是因为 $\beta_{k+1} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$ 。由强 Wolfe 条件中的第二个不等式，我们有

$$\left|\nabla f\left(x_{k+1}\right)^T p_k\right| \leq -c_2 \nabla f\left(x_k\right)^T p_k$$

因此

$$-1+c_2\frac{\nabla f\left(x_k\right)^T p_k}{\left\|\nabla f\left(x_k\right)\right\|^2} \leq -1+\frac{\nabla f\left(x_{k+1}\right)^T p_k}{\left\|\nabla f\left(x_k\right)\right\|^2} \leq -1-c_2\frac{\nabla f\left(x_k\right)^T p_k}{\left\|\nabla f\left(x_k\right)\right\|^2}$$

代入我们的归设即可得到结论。

■

Fletcher-Reeves 格式的问题：

定义 $\cos\theta_k = \frac{-\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|\|p_k\|}$. 若我们有 $\cos\theta_k \approx 0$, 即 p_k 是一个比较差的下降方向。在(15.6)两边同时乘 $\frac{\|\nabla f(x_k)\|}{\|p_k\|}$, 得到

$$\frac{1-2c_2}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|} \leq \cos\theta_k \leq \frac{1}{1-c_2}\frac{\|\nabla f_k\|}{\|p_k\|}, \quad \text{for all } k=0,1,\ldots$$

以上不等式告诉我们 $\cos\theta_k \approx 0$ 当且仅当

$$\|\nabla f(x_k)\| \ll \|p_k\|.$$

因为 p_k 几乎正交于梯度, 那么 x_k 到 x_{k+1} 的移动非常小，即, $x_{k+1} \approx x_k$. 那么, 我们有 $\nabla f_{k+1} \approx \nabla f_k$, 因此

$$\beta_{k+1}^{\mathrm{FR}} \approx 1$$

由上述结果以及 $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| \ll \|p_k\|$, 我们得到

$$p_{k+1} \approx p_k,$$

所以新的搜索方向几乎与之前的相同。从而，算法陷入停滞。

Polak–Ribiere (PR) 算法在上述 Fletcher-Reeves (FR) 遇到的问题中, 表现大不相同。假设 $\cos\theta_k \approx 0$, 那么 $\nabla f_{k+1} \approx \nabla f_k$ 得到

$$\beta_{k+1}^{\mathrm{PR}} = \frac{{r_{k+1}}^T(r_{k+1}-r_k)}{{r_k}^T r_k} \approx 0.$$

由

$$p_{k+1} = -r_{k+1} + \beta_k p_k.$$

可知新的 $p_{k+1} \approx \nabla f(x_{k+1})$. 也就是说，算法选择了负梯度方向重新开始。Hestenes-Stiefel（HS）同样重新开始。但是, PR 和 HS 算法并没有全局收敛性。主要因为所产生的搜索方向 $p_{k+1} = -r_{k+1} + \beta_k p_k$ 可能不再是下降方向。

因此，对于 FR 算法，我们应该周期性采用梯度下降方向作为搜索方向，例如，每 n 步重新选择负梯度方向作为搜索方向即令 $p_{(\ell n)} = -r_{(\ell n)}$, $\ell = 1,2,\ldots$

这种策略称为重启动策略，这样的共轭梯度法也称作重启动共轭梯度法。Fletcher-Reeves 方法在实现中必须使用重启动。实际中，也可以通过判定比值

$$|\frac{\nabla f(x_{k+1})^T \nabla f(x_k)}{\|\nabla f(x_{k+1})\|^2}|$$

确定是否需要重启动。如果比值很小，则说明梯度相差较大，是理想的情况；反之则需重启动。

Dai-Yuan 方法，是另一个可以保证全局收敛的算法，仅需要弱 Wolfe 线搜索条件。

以上 4 种方法，各有优势。实际中还需结合具体问题确定方法。

3.1 共轭梯度法和拟牛顿法

最后，对于一般非线性问题，我们简单比较一下共轭梯度法和拟牛顿法。从实际计算效率及稳定性来看，共轭梯度法未必比拟牛顿法好。但是，共轭梯度法中搜索方向的计算仅仅用到目标函数的梯度，而不必像拟牛顿法那样在每次迭代中更新 Hesse 矩阵（或其逆）的近似阵并记忆之。所以，当问题的规模大而且有稀疏结构时，共轭梯度法有高效执行计算的好处。

其实，共轭梯度法和拟牛顿法有如下的联系。在有限内存 BFGS 中，令 $m = 1, H_k^0 \equiv I_n$, BFGS 更新公式变为了

$$H_{k+1} = \left(I_n - \frac{s_k y_k^{\mathsf{T}}}{y_k^{\mathsf{T}} s_k}\right)\left(I_n - \frac{y_k s_k^{\mathsf{T}}}{y_k^{\mathsf{T}} s_k}\right) + \frac{s_k s_k^{\mathsf{T}}}{y_k^{\mathsf{T}} s_k}.$$

使用精确线搜索时我们总有

$$d_k^{\mathsf{T}} \nabla f(x_{k+1}) = 0, s_k^{\mathsf{T}} \nabla f(x_{k+1}) = 0,$$

于是，我们得到

$$d_{k+1} = -H_{k+1}\nabla f(x_{k+1}) = -\nabla f(x_{k+1}) + \frac{s_k y_k^{\mathsf{T}}}{y_k^{\mathsf{T}} s_k}\nabla f(x_{k+1}). \tag{15.7}$$

注意到

$$\frac{s_k y_k^{\top}}{y_k^{\top} s_k}\nabla f(x_{k+1}) = \frac{y_k^{\top}\nabla f(x_{k+1})}{y_k^{\top} s_k} s_k = \frac{(g_{k+1} - g_k)^{\top} g_{k+1}}{y_k^{\top} d_k} d_k = -\frac{(g_{k+1} - g_k)^{\top} g_{k+1}}{g_k^{\top} d_k} d_k.$$

我们这里使用了记号 $g_k = \nabla f(x_k)$. 对于分母，我们有

$$g_k^{\top} d_k = g_k^{\top}(-g_k + c d_{k-1}) = -g_k^{\top} g_k,$$

这里第一个等号根据(15.7), 得到

$$d_k = -g_k + \frac{y_{k-1}^T g_k}{g_{k-1}^T s_{k-1}} s_{k-1} = -g_k + c d_{k-1}.$$

$$d_{k+1} = -g_{k+1} + \frac{(g_{k+1} - g_k)^{\mathsf{T}} g_{k+1}}{g_k^{\mathsf{T}} g_k} d_k.$$

Polak-Ribiere 法的共轭梯度的更新为:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad \beta_k = \frac{(g_{k+1} - g_k)^{\mathsf{T}} g_{k+1}}{g_k^{\mathsf{T}} g_k}.$$

故两种方法在此时等价。

MATH 3011 运筹学 **秋季 2024**

Lecture 16: 无约束优化 信赖域方法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{16.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数, 决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$. f 是二次可微的。

2 信赖域方法

为了保证迭代法的全局收敛性, 之前我们采用了一维搜索策略。一维搜索策略先确定一个搜索方向 d^k, 然后沿着这个方向选择适当的步长因子 α_k, 新的迭代点 $x^{k+1} = x^k + \alpha_k d^k$.

现在, 我们讨论另一种全局收敛策略 - 信赖域方法 (Trust-Region Method).

信赖域方法的基本思路如下:

1. 在当前迭代点 x^k 建立局部模型

$$d^k = \arg\min_d (g^k)^\top d + \frac{1}{2} d^\top B d, \quad \text{s.t.} \quad \|d\|_2 \le \Delta_k \tag{16.2}$$

2. 求出局部模型的最优解

3. 更新模型信赖域的半径:

   - 模型足够好 ⇒ 增大半径
   - 模型比较差 ⇒ 缩小半径
   - 否则半径不变

4. 对模型进行评价:

   - 好 ⇒ 子问题的解即下一个迭代点

- 差 ⇒ 迭代点不改变

信赖域法的理解：

- 根据带拉格朗日余项的泰勒展开

$$f(x^k + d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k + td) d$$

  其中 $t \in (0, 1)$ 为和 d 有关的正数.

- 和牛顿法相同，利用 $f(x)$ 的二阶近似来刻画 $f(x)$ 在点 x^k 处的性质：

$$m_k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T B^k d$$

  其中 B^k 是对称矩阵，并且是海瑟矩阵的近似矩阵. B^k **可能不是正定矩阵**。

- 由于泰勒展开的局部性, 需对上述模型添加信赖域约束：

$$\Omega_k = \{x^k + d | \ ||d|| \leqslant \Delta_k\},$$

  其中 $\Delta_k > 0$ 是**信赖域半径**。

- 引入如下定义来衡量 $m_k(d)$ 近似程度的好坏：

$$\rho_k = \frac{f\left(x^k\right) - f\left(x^k + d^k\right)}{m_k\left(0\right) - m_k\left(d^k\right)} \tag{16.3}$$

  其中 d^k 为解信赖域子问题得到的迭代方向. 根据 ρ_k 的定义可知，其为函数值实际下降量与预估下降量（即二阶近似模型下降量）的比值.

- 如果 ρ_k 接近 1，说明 $m_k(d)$ 来近似 $f(x)$ 是比较成功的，则应该扩大 Δ_k；如果 ρ_k 非常小甚至为负，就说明我们过分地相信了二阶模型 $m_k(d)$，此时应该缩小 Δ_k. 使用这个机制可以动态调节 Δ_k，让二阶模型 $m_k(d)$ 的定义域处于一个合适的范围.

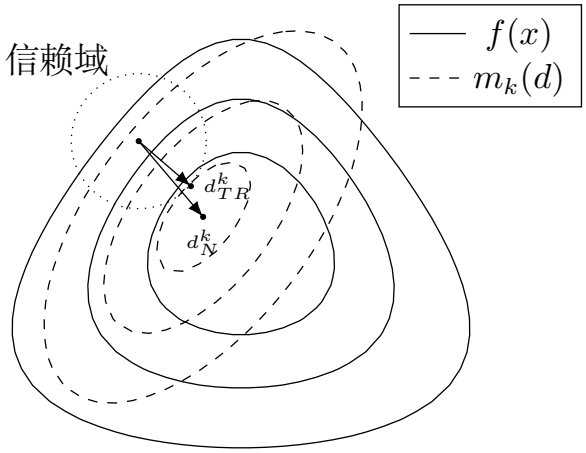


图 16.1: 图中实线表示 $f(x)$ 的等高线，虚线表示 $m_k(d)$ 的等高线，d_N^k 表示解无约束问题得到的下降方向，d_{TR}^k 表示解信赖域子问题得到的下降方向.

Algorithm 1 信赖域算法

1: 给定最大半径 $\Delta_{\max}$，初始半径 Δ_0，初始点 x^0，$k \leftarrow 0$.

2: 给定参数 $0 \leq \eta < \bar{\rho}_1 < \bar{\rho}_2 < 1$，$\gamma_1 < 1 < \gamma_2$.

3: **while** 未达到收敛准则 **do**

4: 计算子问题(16.2)得到迭代方向 d^k.

5: 根据(16.3)式计算下降率 ρ_k.

6: 更新信赖域半径：

$$\Delta_{k+1} = \begin{cases} \gamma_1 \Delta_k, & \rho_k < \bar{\rho}_1, \\ \min\{\gamma_2 \Delta_k, \Delta_{\max}\}, & \rho_k > \bar{\rho}_2 \text{ 以及 } \|d^k\| = \Delta_k, \\ \Delta_k, & \text{其他.} \end{cases}$$

7: 更新自变量：

$$x^{k+1} = \begin{cases} x^k + d^k, & \rho_k > \eta, \\ x^k, & \text{其他.} \end{cases} \qquad \texttt{/*} \text{ 只有下降比例足够大才更新 } \texttt{*/}$$

8: $k \leftarrow k + 1$.

9: **end while**

2.1 信赖域子问题

信赖域算法每一步都需要求解如下子问题:

$$\min_{d\in R^n} \quad m_k(d), \quad \text{s.t.} \quad \|d\| \le \Delta_k. \tag{16.4}$$

首先我们有如下最优性条件。

Theorem 16.1 (信赖域子问题最优性条件) *d^* 是信赖域子问题*

$$\min \quad m(d) = f + g^{\mathrm{T}}d + \frac{1}{2}d^{\mathrm{T}}Bd, \quad \text{s.t.} \quad \|d\| \le \Delta \tag{16.5}$$

的全局极小解当且仅当 d^* 是可行的且存在 $\lambda \ge 0$ 使得

$$(B + \lambda I)d^* = -g, \tag{16.6a}$$

$$\lambda(\Delta - \|d^*\|) = 0, \tag{16.6b}$$

$$B + \lambda I \succeq 0. \tag{16.6c}$$

Proof: 必要性：问题(16.5)的拉格朗日函数为

$$L(d, \lambda) = f + g^{\mathrm{T}}d + \frac{1}{2}d^{\mathrm{T}}Bd - \frac{\lambda}{2}(\Delta^2 - \|d\|^2),$$

其中乘子 $\lambda \ge 0$.

- 由 KKT 条件, d^* 是可行解, 且 $\nabla_d L(d^*, \lambda) = (B+\lambda I)d^* + g = 0$. 此外由互补条件 $\frac{\lambda}{2}(\Delta^2 - \|d^*\|^2) = 0$, 整理后就是(16.6a) 式和(16.6b) 式.
- 为了证明(16.6c)式，我们任取 d 满足 $\|d\| = \Delta$，根据最优性，有

  $$m(d) \ge m(d^*) = m(d^*) + \frac{\lambda}{2}(\|d^*\|^2 - \|d\|^2).$$

  利用(16.6a)式消去 g，代入上式整理有 $(d - d^*)^{\mathrm{T}}(B + \lambda I)(d - d^*) \ge 0$, 由 d 的任意性可知 $B + \lambda I$ 半正定.

再证明充分性. 定义辅助函数

$$\hat{m}(d) = f + g^{\mathrm{T}}d + \frac{1}{2}d^{\mathrm{T}}(B + \lambda I)d = m(d) + \frac{\lambda}{2}d^{\mathrm{T}}d,$$

由条件 (16.6c) 可知 $\hat{m}(d)$ 关于 d 是凸函数. 根据条件 (16.6a)，d^* 满足凸函数一阶最优性条件，可推出 d^* 是 $\hat{m}(d)$ 的全局极小值点，进而对任意可行解 d，我们有

$$m(d) \ge m(d^*) + \frac{\lambda}{2}(\|d^*\|^2 - \|d\|^2).$$

由互补条件 (16.6b) 可知 $\lambda(\Delta^2 - \|d^*\|^2) = 0$，代入上式消去 $\|d^*\|^2$ 得

$$m(d) \ge m(d^*) + \frac{\lambda}{2}(\Delta^2 - \|d\|^2) \ge m(d^*).$$

■

2.2 求解信赖域子问题迭代法

上述定理提供了一个寻找 d 的方法:

- – $\lambda = 0$，并且 $B \succeq 0$，此时 $m(d)$ 是凸函数。求出 d 使其满足 $Bd = -g$ 且 $\|d\| \le \Delta$ 满足.
  – 选择充分大的 $\lambda > 0$ 使得 $B + \lambda I \succeq 0$ 且 $\|d(\lambda)\| = \Delta$，并且满足

$$(B + \lambda I)d(\lambda) = -g. \tag{16.7}$$

  此时问题等价于：求解关于 λ 的方程 $\|d(\lambda)\| = \Delta$ 或者 $1/\|d(\lambda)\| = 1/\Delta$.

- 设 B 有特征值分解 $B = Q\Lambda Q^{\mathrm{T}}$，其中 $Q = [q_1, q_2, \cdots, q_n]$ 是正交矩阵，$\Lambda = \mathrm{Diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ 是对角矩阵，$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ 是 B 的特征值. 若 $\lambda > -\lambda_1$, 我们有

$$d(\lambda) = -Q(\Lambda + \lambda I)^{-1}Q^{\mathrm{T}}g = -\sum_{i=1}^{n} \frac{q_i^{\mathrm{T}}g}{\lambda_i + \lambda} q_i. \tag{16.8}$$

  这正是 $d(\lambda)$ 的正交分解，由正交性可容易求出

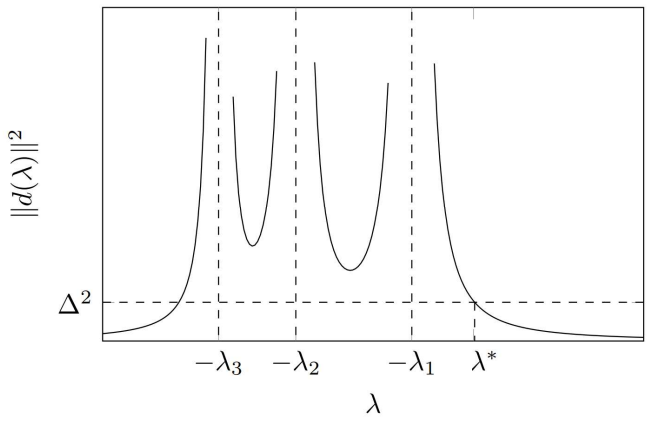
$$\|d(\lambda)\|^2 = \sum_{i=1}^{n} \frac{(q_i^{\mathrm{T}}g)^2}{(\lambda_i + \lambda)^2}. \tag{16.9}$$

- 由 (16.9) 式，当 $\lambda > -\lambda_1$ 且 $q_1^{\mathrm{T}}g \neq 0$ 时，$\|d(\lambda)\|^2$ 是关于 λ 的严格减函数，且有

$$\lim_{\lambda \to \infty} \|d(\lambda)\| = 0, \quad \lim_{\lambda \to -\lambda_1 +} \|d(\lambda)\| = +\infty.$$

- 由连续函数介值定理，$\|d(\lambda)\| = \Delta$ 的解必存在且唯一.

- 所以寻找 λ^* 已经转化为一个一元方程求根问题，可使用牛顿法求解.



习题 6.9： 当 $q_1^T g \neq 0$ 时，写出求解 $\phi(\lambda) = 1/\Delta - 1/\|d(\lambda)\| = 0$ 且 $B + \lambda I \succeq 0$. 的牛顿迭法代公式。

- 上面的分析中假定了 $q_1^{\mathrm{T}} g \neq 0$，在实际中这个条件未必满足. 当 $q_1^{\mathrm{T}} g = 0$ 时，(16.9)式将没有和 λ_1 相关的项. 此时未必存在 $\lambda^* > -\lambda_1$ 使得 $\|d(\lambda^*)\| = \Delta$ 成立. 记

$$M = \lim_{\lambda \to -\lambda_1 +} \|d(\lambda)\|$$

- 当 $M \geq \Delta$ 时，仍然可以根据介值定理得出 $\lambda^* (> -\lambda_1)$ 的存在性；

- 当 $M < \Delta$ 时，无法利用前面的分析求出 λ^* 和 d^*，此时信赖域子问题变得比较复杂.
  实际上，$q_1^{\mathrm{T}} g = 0$ 且 $M < \Delta$ 的情形被称为“困难情形 (hard case)”. 此情形发生时，区间 $(-\lambda_1, +\infty)$ 中的点无法使得 (16.6b) 成立，而定理 16.1 的结果说明 $\lambda^* \in [-\lambda_1, +\infty)$，因此必有 $\lambda^* = -\lambda_1$.

- 为了求出 d^*，可以利用（奇异）线性方程组 (16.6a) 解的结构，其通解可以写为

$$d(\alpha) = -\sum_{i=2}^{n} \frac{q_i^{\mathrm{T}} g}{\lambda_i - \lambda_1} q_i + \alpha q_1, \quad \alpha \in \mathbb{R}.$$

- 由正交性，

$$\|d(\alpha)\|^2 = \alpha^2 + \sum_{i=2}^{n} \frac{(q_i^{\mathrm{T}} g)^2}{(\lambda_i - \lambda_1)^2}.$$

  注意在困难情形中有 $M = \sqrt{\sum_{i=2}^{n} \frac{(q_i^{\mathrm{T}} g)^2}{(\lambda_i - \lambda_1)^2}} < \Delta$，因此必存在 α^* 使得 $\|d(\alpha^*)\| = \Delta$. 这就求出了 d^* 的表达式.

该方法需要对 B 进行特征根分解，故很多情况下效率较低。

2.3 截断共轭梯度法介绍

下面再介绍一种信赖域子问题的求解方法.

- 既然信赖域子问题的解不易求出，则求出其近似解，Steihaug 在 1983 年对共轭梯度法进行了改造，使其成为能求解子问题的算法. 此算法能够应用在大规模问题中，是一种非常有效的信赖域子问题的求解方法.

- 由于子问题和一般的二次极小化问题相差一个约束，如果先不考虑其中的约束 $\|d\| \leq \Delta$ 而直接使用共轭梯度法求解，在迭代过程中找到合适的迭代点作为信赖域子问题的近似解，检测到负曲率或者到达信赖域边界 $\|d\| = \Delta$ 即终止. 这就是截断共轭梯度法的基本思想.

- 回顾标准共轭梯度法求解二次极小化问题

$$\min_s \quad q(s) = g^{\mathrm{T}}s + \frac{1}{2}s^{\mathrm{T}}Bs,$$

- 给定初值 $s^0 = 0, r^0 = g, p^0 = -g$，迭代过程为

$$\begin{aligned}
\alpha_k &= \frac{\|r^k\|^2}{(p^k)^{\mathrm{T}}Bp^k},\\
s^{k+1} &= s^k + \alpha_k p^k,\\
r^{k+1} &= r^k + \alpha_k Bp^k,\\
\beta_k &= \frac{\|r^{k+1}\|^2}{\|r^k\|^2},\\
p^{k+1} &= -r^{k+1} + \beta_k p^k,
\end{aligned}$$

  其中迭代序列 $\{s^k\}$ 最终的输出即为二次极小化问题的解，算法的终止准则是判断 $\|r^k\|$ 是否足够小.

- 截断共轭梯度法则是标准共轭梯度法增加了两条终止准则，并对最后一步的迭代点 s^k 进行修正来得到信赖域子问题的解.

- 矩阵 B 不一定正定，在迭代过程中可能会产生如下三种情况：

  1. $(p^k)^{\mathrm{T}}Bp^k \le 0$，即 B 不是正定矩阵. 遇到这种情况立即终止算法. 但根据这个条件也找到了一个负曲率方向，注意到此时 ${r^k}^T p^k < 0$, 因为 $g^T(s^k + \tau p_k) + \frac{1}{2}((s^k + \tau p^k)^T B(s^k + \tau p^k)) = q(s^k) + \tau(r^k)^T p^k + \frac{\tau^2}{2}(p^k)^T B_k p^k < q(s^k)$. 此时只需要沿着这个方向走到信赖域边界即可.（具体讨论在后面的定理 16.2中。）

  2. $(p^k)^{\mathrm{T}}Bp^k > 0$ 但 $\|s^{k+1}\| \ge \Delta$，这表示若继续进行共轭梯度法迭代，则点 s^{k+1} 将处于信赖域之外或边界上，此时必须马上停止迭代，并在 s^k 和 s^{k+1} 之间找一个近似解.

  3. $(p^k)^{\mathrm{T}}Bp^k > 0$ 且 $\|r^{k+1}\|$ 充分小，这表示若共轭梯度法成功收敛到信赖域内. 子问题(16.4)和不带约束的二次极小化问题是等价的.

- 从上述终止条件来看截断共轭梯度法仅仅产生了共轭梯度法的部分迭代点，这也是该方法名字的由来.

$$\min_s \quad q(s) = g^{\mathrm{T}}s + \frac{1}{2}s^{\mathrm{T}}Bs, \quad \text{s.t.} \quad \|s\| \le \Delta.$$

Algorithm 2 截断共轭梯度法 1 (Steihaug-CG)

给定精度 $\varepsilon > 0$，初始化 $s^0 = 0, r^0 = g, p^0 = -g$，$k \leftarrow 0$.
if $\|p^0\| \leq \varepsilon$ **then**
　算法停止，输出 $s = 0$.
end if
LOOP
if $(p^k)^{\mathrm{T}} B p^k \leq 0$ **then**
　计算 $\tau > 0$ 使得 $\|s^k + \tau p^k\| = \Delta$.
　算法停止，输出 $s = s^k + \tau p^k$.
end if
计算 $\alpha_k = \frac{\|r^k\|^2}{(p^k)^{\mathrm{T}} B p^k}$，更新 $s^{k+1} = s^k + \alpha_k p^k$.
if $\|s^{k+1}\| \geq \Delta$ **then**
　计算 $\tau > 0$ 使得 $\|s^k + \tau p^k\| = \Delta$.
　算法停止，输出 $s = s^k + \tau p^k$.
end if
计算 $r^{k+1} = r^k + \alpha_k B p^k$.
if $\|r^{k+1}\| < \varepsilon \|r^0\|$ **then**
　算法停止，输出 $s = s^{k+1}$.
end if
计算 $\beta_k = \frac{\|r^{k+1}\|^2}{\|r^k\|^2}$，更新 $p^{k+1} = -r^{k+1} + \beta_k p^k$.
$k \leftarrow k + 1$.
ENDLOOP

截断共轭梯度法的迭代序列 $\{s^k\}$ 有非常好的性质，实际上我们可以证明如下定理:

Theorem 16.2 设 $q(s)$ 是任意外迭代步信赖域子问题的目标函数，令 $\{s^j\}$ 是由截断共轭梯度算法产生的迭代序列，则在算法终止前 $q(s^j)$ 是严格单调递减的，即

$$q(s^{j+1}) < q(s^j). \tag{16.10}$$

并且 $\|s^j\|$ 是严格单调递增的，即

$$0 = \|s^0\| < \|s^1\| < \cdots < \|s^{j+1}\| < \cdots \leq \Delta. \tag{16.11}$$

Proof:

设迭代在第 t 步终止. 根据算法 2，在终止前，若 $(p^j)^{\mathrm{T}} B p^j > 0$, $j < t$ 一直成立. 此时算法即共轭梯度法，容易证明(16.10)式和(16.11)式.
又 $q(s)$ 在点 s^j 处的梯度为 r^j，由共轭梯度法性质 $(r^j)^{\mathrm{T}} p^i = 0, i < j$，所以 $(r^j)^{\mathrm{T}} p^j = (r^j)^{\mathrm{T}}(-r^j +$

$\beta_{j-1}p^{j-1}) = -\|r^j\|^2 < 0$, 即 p^j 是下降方向. 而 α_j 的选取为精确步长, 因此有 $q(s^{j+1}) < q(s^j)$. 此外由 s^j 的定义,$s^j = \sum_{i=0}^{j-1} \alpha_i p^i, \alpha_i > 0$. 再根据共轭梯度法的性质:

$$(p^j)^{\mathrm{T}} s^j = \sum_{i=0}^{j-1} \alpha_i (p^j)^{\mathrm{T}} p^i = \sum_{i=0}^{j-1} \alpha_i \frac{\|r^j\|^2}{\|r^i\|^2} \|p^i\|^2 > 0.$$

结合以上表达式可得

$$\|s^{j+1}\|^2 = \|s^j + \alpha_j p^j\|^2 = \|s^j\|^2 + 2\alpha_j (p^j)^{\mathrm{T}} s^j + \alpha_j^2 \|p^j\|^2 > \|s^j\|^2.$$

■

实际上，我们还可进一步说明截断共轭梯度算法的输出 s 满足如下关系:

$$q(s) \le q(s^t), \quad \|s^t\| \le \|s\|,$$

其中 t 为算法终止时的迭代数. 这只需要分别讨论三种终止条件即可.

1. 若 $(p^t)^{\mathrm{T}} B p^t \le 0$, 则 p^t 是负曲率方向, 沿着负曲率方向显然有 $q(s) \le q(s^t)$. 注意到此时 $\|s\| = \Delta$, 因此有 $\|s^t\| \le \|s\| = \Delta$.
2. 若 $(p^t)^{\mathrm{T}} B p^t > 0$ 但 $\|s^{t+1}\| \ge \Delta$，根据最速下降法的性质，$q(s^t + \alpha p^t)$ 关于 $\alpha \in (0, \alpha_t]$ 单调下降，根据 τ 的取法显然有 $q(s) \le q(s^t)$. 此时依然有 $\|s\| = \Delta$，因此 $\|s^t\| \le \|s\| = \Delta$ 仍成立.
3. 若 $(p^t)^{\mathrm{T}} B p^t > 0$ 且 $\|r^{t+1}\| \le \varepsilon \|r^0\|$，此时算法就是共轭梯度法，结论自然成立.

2.4 柯西点

Definition 16.1 (柯西点) 设 $m_k(d)$ 是 $f(x)$ 在点 $x = x^k$ 处的二阶近似，τ_k 为如下优化问题的解:

$$\begin{aligned} \min \quad & m_k(-\tau \nabla f(x^k)), \\ \text{s.t.} \quad & \|\tau \nabla f(x^k)\| \le \Delta_k, \ \tau \ge 0. \end{aligned}$$

则称 $x_C^k := x^k + d_C^k$ 为**柯西点**，其中 $d_C^k = -\tau_k \nabla f(x^k)$.

给定 $m_k(d)$，柯西点可以显式计算出来. 为了方便我们用 g^k 表示 $\nabla f(x^k)$，根据 τ_k 的定义，容易计算出其表达式为

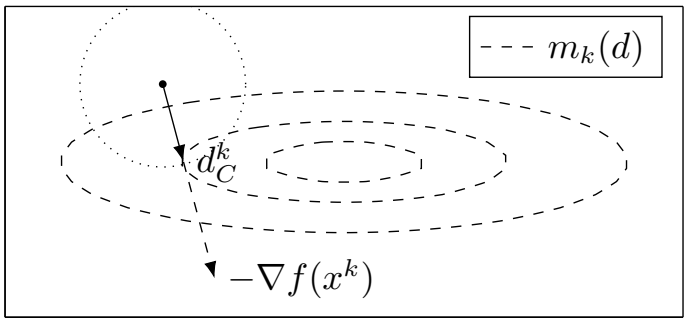
$$\tau_k = \begin{cases} \frac{\Delta_k}{\|g^k\|}, & (g^k)^{\mathrm{T}} B^k g^k \le 0, \\ \min\left\{ \frac{\|g^k\|^2}{(g^k)^{\mathrm{T}} B^k g^k}, \frac{\Delta_k}{\|g^k\|} \right\}, & \text{其他}. \end{cases}$$

Lemma 16.1 (柯西点的下降量) 设 d_C^k 为求解柯西点产生的下降方向，则

$$m_k(0) - m_k(d_C^k) \ge c_1 \|g^k\| \min\left\{ \Delta_k, \frac{\|g^k\|}{\|B^k\|_2} \right\}. \tag{16.12}$$

其中 $c_1 = \frac{1}{2}$

柯西点实际上是在约束下对 $m_k(d)$ 进行了一次精确线搜索的梯度法



2.5 Dogleg (折线法)

折线法的，是连接 Cauchy 点（由梯度下降法产生的极小点）和牛顿点（由牛顿法产生的极小点），其连线与信赖域边界的交点取为 x^{k+1}. 我们略过该方法。

3 信赖域法收敛结论

3.1 全局收敛性

回顾信赖域算法，我们引入了一个参数 η 来确定是否应该更新迭代点. 这分为两种情况：当 $\eta = 0$ 时，只要原目标函数有下降量就接受信赖域迭代步的更新；当 $\eta > 0$ 时，只有当改善量 ρ_k 达到一定程度时再进行更新. 在这两种情况下得到的收敛性结果是不同的，我们分别介绍这两种结果. 在 $\eta = 0$ 的条件下有如下收敛性定理：

Theorem 16.3 (全局收敛性 1) *设近似海瑟矩阵 B^k 有界，即 $\|B^k\|_2 \leq \beta, \forall\ k$，$f(x)$ 在下水平集 $\mathcal{L} = \{x \mid f(x) \leq f(x^0)\}$ 上有下界，且 $\nabla f(x)$ 在 $\mathcal{L}$ 的一个开邻域 $S(R_0)$ 内利普希茨连续. 若 d^k 为信赖域子问题的近似解且满足(16.12)式，信赖域算法选取参数 $\eta = 0$，则*

$$\liminf_{k\to\infty} \|\nabla f(x^k)\| = 0,$$

即 x^k 的聚点中包含稳定点.

定理 (全局收敛性 1) 表明若无条件接受信赖域子问题的更新，则信赖域算法仅仅有子序列的收敛性，迭代点序列本身不一定收敛. 根据下面的定理则说明选取 $\eta > 0$ 可以改善收敛性结果.

Theorem 16.4 (全局收敛性 2) *在定理 (全局收敛性 1) 的条件下，若信赖域算法选取参数 $\eta > 0$, 且信赖域子问题近似解 d^k 满足(16.12) 式，则*

$$\lim_{k\to\infty} \|\nabla f(x^k)\| = 0.$$

和牛顿类算法不同，信赖域算法具有全局收敛性，因此它对迭代初值选取的要求比较弱. 而牛顿法的收敛性极大地依赖初值的选取.

3.2 局部收敛性

- 在构造信赖域子问题时利用了 $f(x)$ 的二阶信息，它在最优点附近应该具有牛顿法的性质. 特别地，当近似矩阵 B^k 取为海瑟矩阵 $\nabla^2 f(x^k)$ 时，根据信赖域子问题的更新方式，二次模型 $m_k(d)$ 将会越来越逼近原函数 $f(x)$.

Theorem 16.5 *设 $f(x)$ 在最优点 $x = x^*$ 的一个邻域内二阶连续可微，且 $\nabla f(x)$ 利普希茨连续，在最优点 x^* 处二阶充分条件成立，即 $\nabla^2 f(x) \succ 0$.* **假设** *(1) 迭代点列 $\{x^k\}$ 收敛到 x^*，并且 (2) 在迭代中选取 B^k 为海瑟矩阵 $\nabla^2 f(x^k)$，(3) 子问题算法产生的迭代方向 d^k 均满足(16.12) (4) 当 $\|d_N^k\| \le \frac{\Delta_k}{2}$ 时，对充分大的 k,*

$$\|d^k - d_N^k\| = o(\|d_N^k\|). \tag{16.13}$$

那么当 k 足够大时，信赖域约束 Δ_k 将未被激活，信赖域算法产生的迭代序列 $\{x^k\}$ 具有 Q-超线性收敛速度.

- 容易看出，若在迭代后期 $d^k = d_N^k$ 能得到满足，则信赖域算法是 Q- 二次收敛的.

- 很多算法都会有这样的性质，例如前面提到的截断共轭梯度法和 dogleg 方法. 因此在实际应用中，截断共轭梯度法是最常用的信赖域子问题的求解方法，使用此方法能够同时兼顾全局收敛性和局部 Q-二次收敛性.

MATH 3011 运筹学 **秋季 2024**

Lecture 17: 次梯度和次梯度法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 次梯度

许多优化问题，目标函数都是不可微的，例如 Lecture 1 中我们见到的基追踪问题和矩阵补全问题，目标函数分别是最小化 ℓ_1 范数和矩阵变量的核范数。为了研究不可微时问题的最优条件，我们可以定义一般非光滑凸函数的次梯度。

非光滑优化在多个其他领域都有应用，例如：

- 机器学习：L1 正则化就是一个典型的非光滑优化问题。
- 信号处理：稀疏信号恢复中常用的 L1 范数最小化。
- 经济学：在某些经济模型中，效用函数或成本函数可能是非光滑的。
- 深度学习：深度学习中，损失函数其实是非光滑的。由于其复杂性太高，故实际中常常忽略了这点。

1.1 次梯度的定义

回顾可微凸函数 f 的一阶等价条件:

$$f(y) \geq f(x) + \nabla f(x)^{\mathrm{T}}(y - x).$$

这表明，f 在点 x 处的一阶近似是 f 的一个全局下界。我们这里的想法是，将上述不等式拓展到一般不可微的情形。我们先考虑简单的函数 $f(x) = |x|, x \in \mathbb{R}$. $f(x)$ 在 $x = 0$ 处不可导，因为其左右导数分别为

$$\lim_{t \to 0^-} \frac{|t|}{t} = -1, \quad \lim_{t \to 0^+} \frac{|t|}{t} = 1.$$

可以验证, 对于任意 $g \in [-1, 1]$, 下面的不等式成立

$$|y| \geq 0 + g \cdot y,$$

此即

$$f(y) \geq f(0) + g \cdot (y - 0).$$

上面的不等式的几何意义为，任意斜率为 $g \in [-1,1]$ 过原点的直线，均为函数 f 的一个下界。

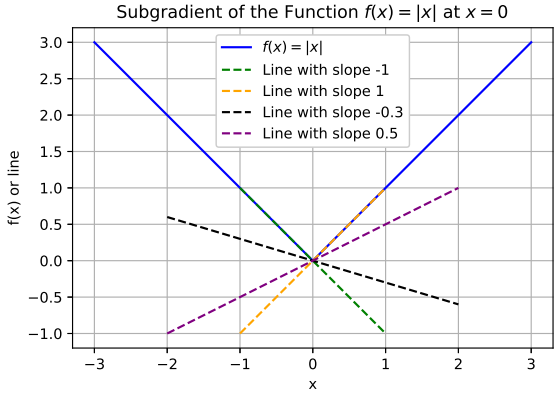


图 17.1: 函数 $f(x) = |x|$ 次梯度示意图。任意斜率为 $g \in [-1,1]$ 过原点的直线，均为函数 f 的一个下界。

我们引出如下定义。

Definition 17.1 (次梯度和次微分) *设 f 为适当凸函数，x 为定义域 $\mathbf{dom}\ f$ 中的一点. 若向量 $g \in \mathbb{R}^n$ 满足*

$$f(y) \geq f(x) + g^{\mathrm{T}}(y - x), \quad \forall y \in \mathbf{dom}\ f,$$

*则称 g 为函数 f 在点 x 处的一个**次梯度** (subgradient). 进一步地，称集合*

$$\partial f(x) = \{g \,|\, g \in \mathbb{R}^n, f(y) \geq f(x) + g^{\mathrm{T}}(y - x), \forall y \in \mathbf{dom}\ f\}$$

*为 f 在点 x 处的**次微分** (subdifferential).*

注 17.1

- 定义中的凸函数，值域可以为广义实数 $\mathbb{R} \cup \{+\infty\}$ 空间。适当函数是指，存在 x 使得 $f(x) < +\infty$.
- 由定义可知，次微分是一个集合，次梯度是某个次微分的元素。

Example 17.1 *$f(x) = \|x\|_2$ 为凸函数。若 $x \neq 0$, $f(x)$ 可微，故*

$$\partial f(x) = \frac{1}{\|x\|_2} x.$$

若 $x = 0$, 我们下面证明 $\partial f(x) = \{g | \|g\|_2 \leq 1\}$. 由定义可知，

$$\|y\|_2 \geq g^T y, \quad \forall y.$$

首先，若 $\|g\|_2 \leq 1$, 由 Cauchy 不等式得 $g^T y \leq \|g\|_2 \|y\|_2 \leq \|y\|_2$, 故

$$\{g | \|g\|_2 \leq 1\} \subset \partial f(0).$$

反之，若 $g \in \partial f(0)$, 故

$$\max_{\|y\|_2=1} g^T y = \|g\|_2 \leq \|y\|_2 = 1.$$

故

$$\partial f(0) \subset \{g | \|g\|_2 \leq 1\}.$$

1.2 次梯度存在性

为了说明定义 17.1中的次梯度对于一般凸函数存在，我们引入如下定义。

Definition 17.2 *设 $f(x)$ 为 $\mathbb{R}^n$ 上的实值函数，函数的上方图* **epi** f *定义为*

$$\textbf{epi } f := \{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{n+1} \mid z \geq f(x) \}.$$

Lemma 17.1 *函数 $f(x)$ 是凸函数，当且仅当其上方图是凸集。*

当 f 可微时，我们有

$$f(x) + \nabla f(x)^T (y - x) \leq f(y) \leq z.$$

即

$$\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}^T \left(\begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} x \\ f(x) \end{bmatrix} \right) \leq 0 \quad \forall\ (y, z) \in \textbf{epi } f$$

这表明，$\nabla f(x)$ 可以诱导出上方图 **epi** f 在点 $(x, f(x))$ 处的支撑超平面, 如下图所示。

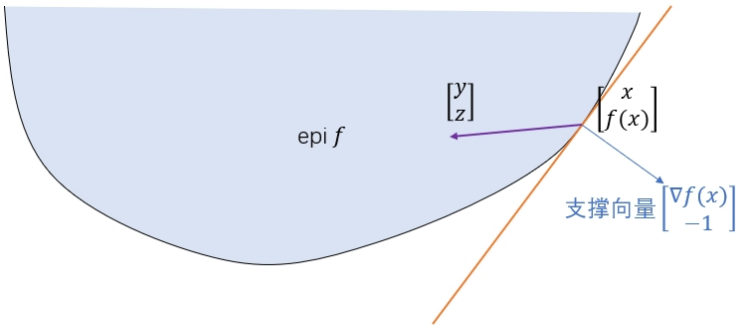


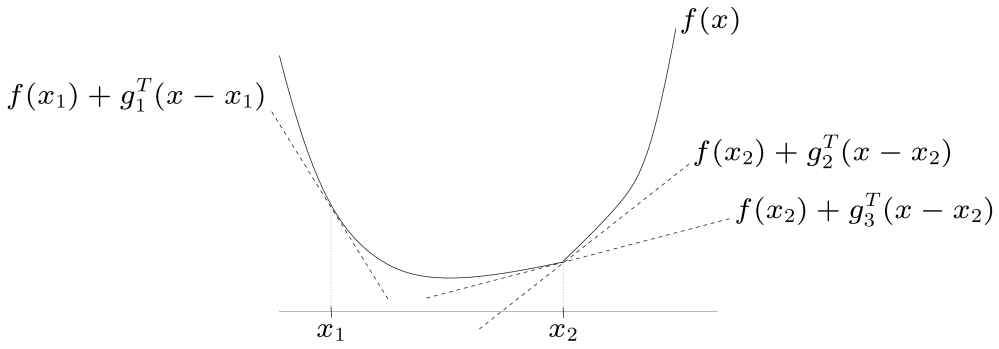
图 17.2: 对于凸函数 $f(x)$, 其上方图 **epi** f 是一个凸集。$\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 是 **epi** f 的支撑向量。

由次梯度的定义 17.1可知，

- $f(x)+g^{\mathrm{T}}(y-x)$ 是 $f(y)$ 的一个全局下界
- g 可以诱导出上方图 **epi** f 在点 $(x,f(x))$ 处的一个支撑超平面

$$\begin{bmatrix} g \\ -1 \end{bmatrix}^T \left(\begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} x \\ f(x) \end{bmatrix} \right) \leq 0 \quad \forall\ (y,z) \in \textbf{epi}\ f$$

- 如果 f 是可微凸函数, 那么 $\nabla f(x)$ 是 f 在点 x 处的一个次梯度
- 例: g_2, g_3 是点 x_2 处的次梯度; g_1 是点 x_1 处的次梯度



图片来源:《最优化计算方法》文再文等讲义。

次梯度的存在性主要依赖于凸集的下述性质:

Lemma 17.2 *任意凸集的边界点处都存在支撑超平面。*

Theorem 17.1 *设 f 为凸函数, **dom** $f=\{x: f(x)<\infty\}$ 为其定义域. 如果 $x \in$ **int dom** f, 则 $\partial f(x)$ 是非空的, 其中 **int dom** f 的含义是集合 **dom** f 的所有内点.*

Proof: $(x,f(x))$ 是 **epi** f 边界上的点. 因此凸集 **epi** f 在点 $(x,f(x))$ 处存在支撑超平面:

$$\exists\ (a,b) \neq 0, \quad \begin{bmatrix} a \\ b \end{bmatrix}^{\mathrm{T}} \left(\begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} x \\ f(x) \end{bmatrix} \right) \leq 0 \quad \forall\ (y,z) \in \textbf{epi}\ f$$

令 $z \to +\infty$, 可知 $b \leq 0$. 由于 $x \in$ **int dom** f, 取 $y = x + \epsilon a \in$ **dom** f, $\epsilon > 0$, 可知 $b \neq 0$.

因此 $b<0$, 令 $g = a/|b|$, 则 g 是 f 在点 x 处的次梯度。

■

Example 17.2 (反例) *如下函数在点 $x=0$ 处不是次可微的:*

- $f:\mathbf{R}\to\mathbf{R}, \mathbf{dom}\ f=\mathbf{R}_+=\{x\in\mathbf{R}\mid x\geq 0\}$

$$x=0 \text{ 时, } f(x)=1, x>0 \text{ 时, } f(x)=0$$

- $f:\mathbf{R}\to\mathbf{R}, \mathbf{dom}\ f=\mathbf{R}_+$

$$f(x)=-\sqrt{x}$$

epi f 在点 $(0,f(0))$ 处的唯一支撑超平面是垂直的

1.3 次梯度的计算法则

- **可微凸函数：** 若凸函数 f 在点 x 处可微，则 $\partial f(x)=\{\nabla f(x)\}$.
- **凸函数的非负线性组合：** 设凸函数 f_1,f_2 满足 **int dom** $f_1\cap$ **dom** $f_2\neq\emptyset$，而 $x\in$ **dom** $f_1\cap$ **dom** f_2. 若

$$f(x)=\alpha_1 f_1(x)+\alpha_2 f_2(x),\quad \alpha_1,\alpha_2\geq 0,$$

则 $f(x)$ 的次微分

$$\partial f(x)=\alpha_1\partial f_1(x)+\alpha_2\partial f_2(x).$$

- **线性变量替换：** 设 h 为适当凸函数，f 满足 $f(x)=h(Ax+b)$. 若存在 $x^\sharp\in\mathbb{R}^m$，使得 $Ax^\sharp+b\in$ **int dom** h，则

$$\partial f(x)=A^{\mathrm{T}}\partial h(Ax+b),\quad \forall\ x\in\textbf{int dom}\ f.$$

- **多个函数取上界：** 设 $f_1,f_2,\cdots,f_m:\mathbb{R}^n\to(-\infty,+\infty]$ 均为凸函数，令

$$f(x)=\max\{f_1(x),f_2(x),\cdots,f_m(x)\},\quad \forall x\in\mathbb{R}^n.$$

对 $x_0\in\bigcap_{i=1}^m$ **int dom** f_i，定义 $I(x_0)=\{i\mid f_i(x_0)=f(x_0)\}$，则

$$\partial f(x_0)=\textbf{conv}\bigcup_{i\in I(x_0)}\partial f_i(x_0).$$

  - $I(x_0)$ 表示点 x_0 处“有效”函数的指标
  - **conv**S 表示集合 S 的凸包, 即集合 S 的所有点的凸组合构成的点集.
  - $\partial f(x_0)$ 是点 x_0 处“有效”函数的次微分并集的凸包
  - 如果 f_i 可微, $\partial f(x_0)=\textbf{conv}\{\nabla f_i(x_0)\mid i\in I(x_0)\}$

- **固定分量的函数极小值**

$$f(x) = \inf_y h(x, y), \quad h \text{ 关于 } (x, y) \text{ 联合凸}$$

计算点 $\hat{x}$ 处的一个次梯度:

- 设 $\hat{y} \in \mathbb{R}^m$ 满足 $h(\hat{x}, \hat{y}) = f(\hat{x})$, 即 $\hat{y}$ 是固定 $x = \hat{x}$ 后 $h(\hat{x}, y)$ 的极小解。
- 存在 $g \in \mathbb{R}^n$ 使得 $(g, 0) \in \partial h(\hat{x}, \hat{y})$, 则 $g \in \partial f(\hat{x})$, 即若 $g \in \partial_x h(\hat{x}, \hat{y})$, 则 $g \in \partial f(\hat{x})$.

证明: 对任意 $x \in \mathbb{R}^n, y \in \mathbb{R}^m$

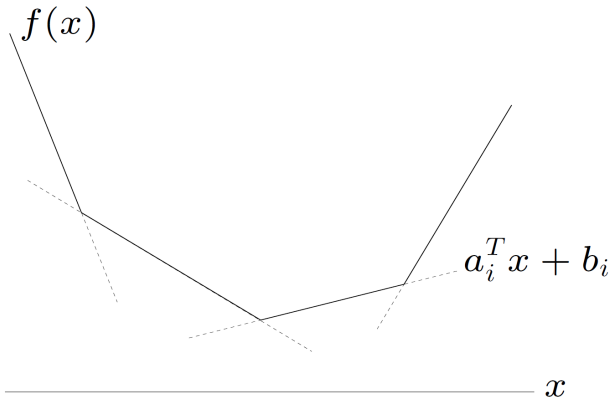
$$\begin{aligned} h(x, y) &\geq h(\hat{x}, \hat{y}) + g^{\mathrm{T}}(x - \hat{x}) + 0^{\mathrm{T}}(y - \hat{y}) \\ &= f(\hat{x}) + g^{\mathrm{T}}(x - \hat{x}). \end{aligned}$$

于是

$$f(x) = \inf_y h(x, y) \geq f(\hat{x}) + g^{\mathrm{T}}(x - \hat{x})$$

Example 17.3 分段线性函数

$$f(x) = \max_{i=1,2,\cdots,m} \{a_i^{\mathrm{T}} x + b_i\}$$



- 点 x 处的次微分是一个多面体

$$\partial f(x) = \mathbf{conv}\{a_i \mid i \in I(x)\}$$

其中 $I(x) = \{i \mid a_i^{\mathrm{T}} x + b_i = f(x)\}$

Example 17.4 鲁棒线性回归: 求函数 $f(x) = \|Ax - b\|_1$ 的次微分, 这里 $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$。

解： 首先考虑函数 $h(y) = \|y\|_1 = \max_{s\in\{-1,1\}^m} s^T y, y \in \mathbb{R}^m$. 故，

$$\partial h(y) = J_1 \times \cdots \times J_m, \quad J_k = \begin{cases} [-1,1], & y_k = 0 \\ \{1\}, & y_k > 0 \\ \{-1\}, & y_k < 0 \end{cases}$$

对于 $f(x)$,

$$\partial f(x) = A^T \left(\partial h(y)|_{y=Ax-b}\right).$$

Example 17.5 设 C 是 $\mathbb{R}^n$ 中一闭凸集，令

$$f(x) = \min_{y\in C} \|x - y\|_2$$

计算点 $\hat{x}$ 处的一个次梯度：

- 若 $f(\hat{x}) = 0$，则容易验证 $g = 0 \in \partial f(\hat{x})$;
- 若 $f(\hat{x}) > 0$，取 $\hat{y}$ 为 $\hat{x}$ 在 C 上的投影，即 $\hat{y} = \mathcal{P}_c(\hat{x})$，计算

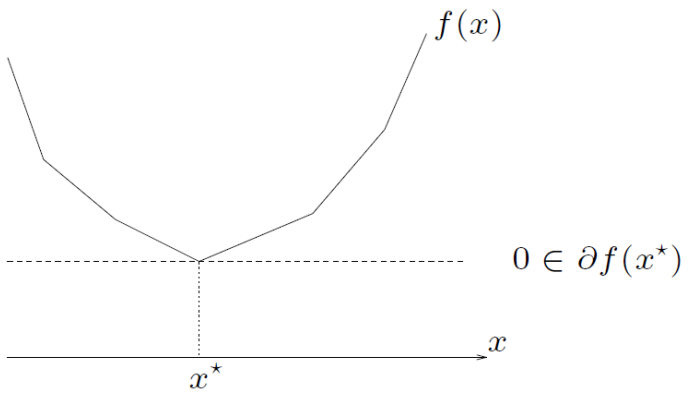
$$g = \frac{1}{\|\hat{x} - \hat{y}\|_2}(\hat{x} - \hat{y}) = \frac{1}{\|\hat{x} - \mathcal{P}_c(\hat{x})\|_2}(\hat{x} - \mathcal{P}_c(\hat{x})).$$

1.4 最优性条件

Theorem 17.2 对于凸函数 $f(x)$, x^* 是 $f(x)$ 的全局极小点当且仅当

$$0 \in \partial f(x^*).$$

Proof:



图片来源：《最优化计算方法》，文再文等讲义。

根据次梯度的定义以及最优性，我们有

$$f(y) \geq f(x^*),\ \forall y \quad \Longleftrightarrow f(y) \geq f(x^*) + 0^{\mathrm{T}}(y - x^*),\ \forall y \quad \Longleftrightarrow \quad 0 \in \partial f(x^*).$$

■

Example 17.6 (分段线性函数最优条件)

$$f(x) = \max_{i=1,\cdots,m} (a_i^{\mathrm{T}} x + b_i)$$

- **最优性条件**

$$0 \in \mathbf{conv}\{a_i \mid i \in I(x^*)\}, \quad \text{其中 } I(x) = \{i \mid a_i^{\mathrm{T}} x + b_i = f(x)\}$$

- 也就是说, x^* 是最优解当且仅当存在 λ 使得

$$\lambda \geq 0, \quad \mathbf{1}^{\mathrm{T}}\lambda = 1, \quad \sum_{i=1}^{m} \lambda_i a_i = 0, \quad \lambda_i = 0 \text{ for } i \notin I(x^*)$$

- 这是等价（思考：为何等价？）线性规划问题的最优性条件: $A = [a_1^\top; \ldots; a_m^\top]$

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & Ax + b \leq t\mathbf{1}. \end{aligned}$$

其对偶为

$$\begin{aligned} \max \quad & b^{\mathrm{T}}\lambda \\ \text{s.t.} \quad & A^{\mathrm{T}}\lambda = 0, \\ & \lambda \geq 0, \quad \mathbf{1}^{\mathrm{T}}\lambda = 1, \end{aligned}$$

强对偶定理为：原始可行：

$$Ax + b \leq t\mathbf{1} \Longleftrightarrow f(x) = \max_{i=1,\cdots,m} (a_i^{\mathrm{T}} x + b_i)$$

互补松弛

$$\lambda^T(Ax + b - t\mathbf{1}) = 0 \quad \Longleftrightarrow \lambda_i = 0 \text{ for } i \notin I(x^*),$$

对偶可行

$$A^{\mathrm{T}}\lambda = 0, \quad \lambda \geq 0, \quad \mathbf{1}^{\mathrm{T}}\lambda = 1.$$

作业 17.1 计算下面两个问题的一个次梯度

- $f(x) = \|Ax - b\|_2$.
- $f(x) = \min_y \|Ay - x\|_\infty$，这里 $\|x\|_\infty = \max_i |x_i|$ 表示无穷范数。假设存在 $\hat{y}$ 使得 $f(\hat{x}) = \min_y \|Ay - x\|_\infty$. 计算一个 $f(\hat{x})$ 的次梯度。

1.5 约束问题的次梯度和最优条件

给定约束 C 为 $\mathbb{R}^n$ 中的闭凸集，考虑问题

$$\begin{aligned} &\min f(x) \\ &\text{s.t.} \quad x \in C. \end{aligned} \tag{17.1}$$

可定义指示函数

$$\mathcal{I}_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{if } x \notin C. \end{cases}$$

则问题(17.1)等价于

$$\min h(x) := f(x) + \mathcal{I}_C(x). \tag{17.2}$$

对于问题(17.2), 最优条件为

$$0 \in \partial f(x) + \partial \mathcal{I}_C(x), x \in C.$$

这里，若 $g \in \partial \mathcal{I}_C(x)$, 则 $\mathcal{I}_C(y) \geq \mathcal{I}_C(x) + g^T(y-x), \forall y \in C$, 即

$$0 \geq g^T(y-x), \quad \forall y \in C.$$

这里说明次梯度在 x 处的法锥中，法锥的定义为 $N_C(x) = \{g \mid 0 \geq g^T(y-x), \forall y \in C\}$. 事实上，法锥与切锥不相交，故这表明次梯度不在切锥中，与前面的课程一致。

一般来说，非光滑约束问题也有 KKT 条件。由于课程的设置，我们不再学习它们。

2 次梯度算法

为了极小化一个不可微的凸函数 f，可类似梯度法构造如下次梯度算法的迭代格式：

$$x^{k+1} = x^k - \alpha_k g^k, \quad g^k \in \partial f(x^k),$$

其中 $\alpha_k > 0$ 为步长, g^k 为 x_k 处函数 f 任意的一个次梯度. 它通常有如下四种选择：

1. 固定步长 $\alpha_k = \alpha$；
2. 消失步长 $\alpha_k \to 0$ 且 $\sum_{k=0}^{\infty} \alpha_k = +\infty$；

下面我们讨论在不同步长取法下次梯度算法的收敛性质.

2.1 次梯度法的收敛结论

Theorem 17.3 假设 f 满足如下性质：

(1) f 为凸函数；

(2) f 至少存在一个有限的极小值点 x^*，且 $f(x^*) > -\infty$；

(3) f 为利普希茨连续的，即

$$|f(x) - f(y)| \leq G\|x - y\|, \quad \forall\ x, y \in \mathbb{R}^n,$$

其中 $G > 0$ 为利普希茨常数.

这等价于 $f(x)$ 的次梯度是有界的 *(*我们略过证明*)*，即

$$\|g\| \leq G, \quad \forall\ g \in \partial f(x), x \in \mathbb{R}^n.$$

取 α_i 为消失步长，即 $\alpha_i \to 0$ 且 $\sum_{i=0}^{\infty} \alpha_i = +\infty$，令

$$\hat{f}^k = \min_{0 \leq i \leq k} f(x_i),$$

则

$$\hat{f}^k - f^* \leq \frac{\|x^0 - x^*\|^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2 \sum_{i=0}^{k} \alpha_i};$$

进一步可得 $\hat{f}^k$ 收敛到 f^*.

Proof: 设 x^* 是 $f(x)$ 的一个全局极小值点，$f^* = f(x^*)$，根据迭代格式，

$$\begin{aligned} \left\|x^{i+1} - x^*\right\|^2 &= \left\|x^i - \alpha_i g^i - x^*\right\|^2 \\ &= \left\|x^i - x^*\right\|^2 - 2\alpha_i \left\langle g^i, x^i - x^* \right\rangle + \alpha_i^2 \left\|g^i\right\|^2 \\ &\leqslant \left\|x^i - x^*\right\|^2 - 2\alpha_i \left(f\left(x^i\right) - f^*\right) + \alpha_i^2 G^2 \end{aligned}$$

结合 $i = 0, \cdots, k$ 时相应的不等式，并定义 $\hat{f}^k = \min_{0 \leqslant i \leqslant k} f\left(x^i\right)$：

$$\begin{aligned} 2\left(\sum_{i=0}^{k} \alpha_i\right)\left(\hat{f}^k - f^*\right) &\leqslant \left\|x^0 - x^*\right\|^2 - \left\|x^{k+1} - x^*\right\|^2 + G^2 \sum_{i=0}^{k} \alpha_i^2 \\ &\leqslant \left\|x^0 - x^*\right\|^2 + G^2 \sum_{i=0}^{k} \alpha_i^2. \end{aligned}$$

■

注 17.2

- 若用常数步长 $\alpha_k = t$, 则

$$\hat{f}^k - f^* \leq \frac{\|x^0 - x^*\|^2}{2kt} + \frac{G^2 t}{2};$$

  - $\hat{f}^k$ 无法保证收敛性
  - 当 k 足够大时, $\hat{f}^k$ 近似为 $G^2t/2$-次优的

- 次梯度方法不是一个下降方法，即无法保证 $f(x^{k+1}) < f(x^k)$

3 应用：LASSO 问题

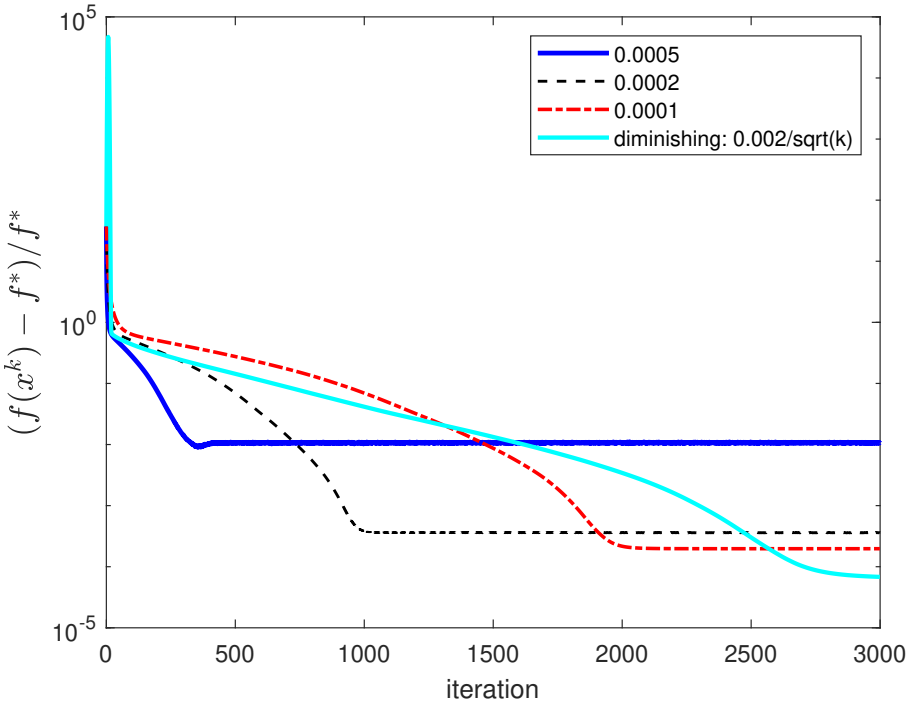
考虑 LASSO 问题

$$\min \quad f(x) = \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1,$$

$f(x)$ 的一个次梯度为 $g = A^{\mathrm{T}}(Ax - b) + \mu\mathrm{sign}(x)$, 其中 $\mathrm{sign}(x)$ 是关于 x 逐分量的符号函数. 因此的次梯度算法为

$$x^{k+1} = x^k - \alpha_k(A^{\mathrm{T}}(Ax^k - b) + \mu\mathrm{sign}(x^k)),$$

步长 α_k 可选为固定步长或消失步长. 图中，选取 $\alpha_k = 0.0001, 0.0002, 0.0005$ 等固定步长，到一定精度



无法继续下降。选取 $\alpha_k = \frac{0.002}{\sqrt{k}}$ 的 diminishing 步长求解精度更高。

参考资料：刘浩洋, 户将, 李勇锋, 文再文, 最优化：建模、算法与理论, 高教出版社, ISBN：9787040550351
H. Liu, J. Hu, Y. Li, Z. Wen, Optimization: Modeling, Algorithm and Theory (in Chinese)

Boyd, Stephen P., and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.

MATH 3011 运筹学 **秋季 2024**

Lecture 18: 约束优化问题 罚函数方法

Lecturer: 陈士祥 *Scribes:* 陈士祥

致谢：感谢北京大学文再文老师提供的《最优化方法》参考讲义

1 问题形式

考虑约束优化问题：

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X}. \end{aligned}$$

其中，$\mathcal{X}$ 为 x 的可行域。约束问题相比于无约束问题的困难：

- 约束优化问题中 x 不能随便取值，梯度下降法所得点不一定在可行域内
- 最优解处目标函数的梯度不一定为零向量

为了解决这些困难，考虑使用**罚函数法**将约束优化问题转化为无约束优化问题处理。

2 二次罚函数方法

2.1 等式问题的二次罚函数法

首先考虑简单情形：仅包含等式约束的约束优化问题

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \end{aligned} \tag{18.1}$$

其中 $x \in \mathbb{R}^n$，$\mathcal{E}$ 为等式约束的指标集，$c_i(x)$ 为连续函数。

定义该问题的二次罚函数为:

$$P_E(x, \sigma) = f(x) + \frac{1}{2}\sigma \sum_{i \in \mathcal{E}} c_i^2(x) \tag{18.2}$$

其中等式右端第二项称为**二次罚函数**，$\sigma > 0$ 称为**罚因子**。

- 由于这种罚函数对不满足约束的点进行惩罚，在迭代过程中点列一般处于可行域之外，因此它也被称为**外点罚函数**.

为了直观理解罚函数的作用，我们给出一个例子:

Example 18.1 *考虑优化问题*

$$\begin{aligned} \min \quad & x+\sqrt{3}y \\ s.t. \quad & x^2+y^2=1 \end{aligned}$$

容易求得最优解为 $\left(-\frac{1}{2},-\frac{\sqrt{3}}{2}\right)^{\mathrm{T}}$，*考虑二次罚函数*

$$P_E(x,y,\sigma)=x+\sqrt{3}y+\frac{\sigma}{2}\left(x^2+y^2-1\right)^2$$

并在下图中绘制出 $\sigma=1$ *和* $\sigma=10$ *对应的罚函数的等高线. 当* $\sigma=1$ *时，罚函数的最小值大概为* $x\approx-0.6625, y\approx-1.147$. *而当* $\sigma=10$ *时，出现了两个局部最优解。*

图 18.1: 取不同的值时二次罚函数 $P_E(x,y,\sigma)$ 的等高线

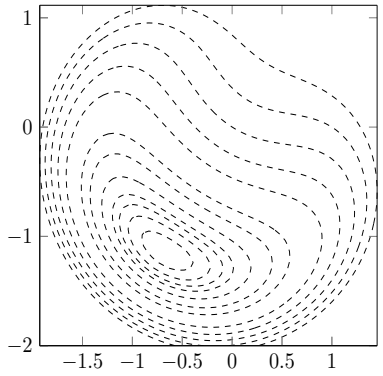


图 18.2: (a) $\sigma=1$

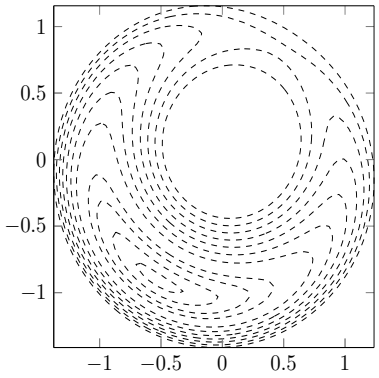


图 18.3: (b) $\sigma=10$

下面这个例子表明，当 σ 选取过小时罚函数可能无下界.

Example 18.2 *考虑优化问题*

$$\begin{aligned} \min \quad & -x^2+2y^2 \\ s.t. \quad & x=1 \end{aligned}$$

容易求得最优解为 $(1,0)^T$，*然而考虑罚函数*

$$P_E(x,y,\sigma)=-x^2+2y^2+\frac{\sigma}{2}(x-1)^2$$

对任意的 $\sigma\leq 2$，*该罚函数无下界。*

上述两个例子表明，罚因子 σ 的选取需要充分大，同时也改变了原问题的性质。

我们从 KKT 条件角度分析原问题和罚函数的性质。

- 原问题的 KKT 条件：

$$\nabla f(x^*) - \sum_{i\in\mathcal{E}} \lambda_i^* \nabla c_i(x^*) = 0$$
$$c_i(x^*) = 0, \quad \forall i \in \mathcal{E}$$

- 添加罚函数项问题的 KKT 条件：

$$\nabla f(x) + \sum_{i\in\mathcal{E}} \sigma c_i(x) \nabla c_i(x) = 0$$

假设两个问题收敛到同一点，对比 KKT 条件 (梯度式)，应有下式成立：

$$\sigma c_i(x) \approx -\lambda_i^*, \quad \forall i \in \mathcal{E}$$

最优点处乘子 λ^* 固定，为使约束 $c_i(x) = 0$ 成立，需要 $\sigma \to \infty$.

因此，我们有如下的算法。

Algorithm 1 二次罚函数法

1: 给定 $\sigma_1 > 0, x_0, k \leftarrow 1$. 罚因子增长系数 $\rho > 1$.
2: **while** 未达到收敛准则 **do**
3: 以 x^{k-1} 为初始点，求解 $x^k = \arg\min_x P_E(x, \sigma_k)$
4: 选取 $\sigma_{k+1} = \rho\sigma_k$.
5: $k \leftarrow k + 1$
6: **end while**

- 考虑罚函数 $P_E(x, \sigma)$ 的海瑟矩阵:

$$\nabla^2_{xx} P_E(x, \sigma) = \nabla^2 f(x) + \sum_{i\in\mathcal{E}} \sigma c_i(x) \nabla^2 c_i(x) + \sigma \sum_{i\in\mathcal{E}} \nabla c_i(x) \nabla c_i(x)^{\mathrm{T}}$$

- 等号右边的前两项可以使用拉格朗日函数 $L(x, \lambda^*)$ 来近似，即：

$$\nabla^2_{xx} P_E(x, \sigma) \approx \nabla^2_{xx} L(x, \lambda^*) + \sigma \sum_{i\in\mathcal{E}} \nabla c_i(x) \nabla c_i(x)^{\mathrm{T}}$$

- 右边为一个定值矩阵和一个最大特征值趋于正无穷的矩阵，这导致 $\nabla^2_{xx} P_E(x, \sigma)$ 条件数越来越大，求解子问题的难度也会相应地增加.

- 此时使用梯度类算法求解将会变得非常困难. 若使用牛顿法，则求解牛顿方程本身就是一个非常困难的问题. 因此在实际应用中，我们不可能令罚因子趋于正无穷.

注意事项:

- 选取合适的参数 ρ: 如果 σ_k 太大，则对应的罚函数条件数非常差，导致问题病态。 σ_k 增长过快会使子问题求解困难，σ_k 增长过慢则会增加迭代次数. 另外，也可以自适应地调整 ρ。
- 检测到迭代点发散就应该立即终止迭代并增大罚因子。
- 为保证收敛，子问题求解误差需要趋于零。

作业 18.1 证明如下 *3* 个结论:

结论 1: 设 $\sigma_{k+1} > \sigma_k > 0$, 则有 $P_E(x^k, \sigma^k) \le P_E(x^{k+1}, \sigma^{k+1})$,

$$\sum_{i\in\mathcal{E}} \|c_i(x^k)\|^2 \ge \sum_{i\in\mathcal{E}} \|c_i(x^{k+1})\|^2, \quad f(x^k) \le f(x^{k+1}).$$

结论 2: 设令 $\bar{\mathbf{x}}$ 是原问题(18.1)的最优解，则对任意的 $\sigma^k > 0$ 成立

$$f(\bar{\mathbf{x}}) \ge P_E(x^k, \sigma^k) \ge f(x^k).$$

结论 3: 令 $\delta = \sum_{i\in\mathcal{E}} \|c_i(x^k)\|^2$, 则 x^k 也是约束问题

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & \sum_{i\in\mathcal{E}} \|c_i(x)\|^2 \le \delta \end{array}$$

的最优解。

2.2 罚函数法的收敛性结论

下面的定理需要假设每个罚函数 $P_E(x, \sigma_k)$ 都有最小值，并且 $\{x^k\}$ 有极限点。

Theorem 18.1 (二次罚函数法的收敛性 1) 设 x^k 是 $P_E(x, \sigma_k)$ 的全局极小解，σ_k 单调上升趋于无穷，则 x^k 的每个极限点 x^* 都是原问题的全局极小解.

Proof: 设 $\bar{x}$ 为原问题的极小解. 由 x^k 为 $P_E(x, \sigma_k)$ 的极小解，得 $P_E\left(x^k, \sigma_k\right) \leqslant P_E\left(\bar{x}, \sigma_k\right)$，即

$$f\left(x^k\right) + \frac{\sigma_k}{2} \sum_{i\in\mathcal{E}} c_i^2\left(x^k\right) \leqslant f(\bar{x}) + \frac{\sigma_k}{2} \sum_{i\in\mathcal{E}} c_i^2(\bar{x}) = f(\bar{x}) \tag{18.3}$$

整理得:

$$\sum_{i\in\mathcal{E}} c_i^2\left(x^k\right) \leqslant \frac{2}{\sigma_k}\left(f(\bar{x}) - f\left(x^k\right)\right) \tag{18.4}$$

设 x^* 是 x^k 的一个极限点,不妨设 $\{x^k\}$ 的子列 $x^{k_n} \to x^*$。在(18.4)式中令 $k_n \to \infty$,得 $\sum_{i\in\mathcal{E}} c_i^2(x^*) = 0$. 由此易知, x^* 为原问题的可行解, 又由(18.3)式知 $f\left(x^k\right) \leqslant f(\bar{x})$, 取极限得 $f(x^*) \leqslant f(\bar{x})$, 故 x^* 为全局极小解。 ■

由于定理 1 需要每个罚函数 $P_E(x,\sigma_k)$ 解出全局最小值。这个要求比较高。下面的定理给出更弱的情况下的收敛结果, 即求解子问题时, 只需满足一阶最优条件下的收敛结论。

Theorem 18.2 (二次罚函数法的收敛性 2) *设 $f(x)$ 与 $c_i(x)$ ($i \in \mathcal{E}$) 连续可微, 正数序列 $\varepsilon_k \to 0$, $\sigma_k \to +\infty$。在算法 1中, 子问题的解 x^{k+1} 满足 $\left\|\nabla_x P_E\left(x^k,\sigma_k\right)\right\| \leq \varepsilon_k$, 而对 x^k 的任何极限点 x^*, 都有 $\{\nabla c_i(x^*), i \in \mathcal{E}\}$ 线性无关, 则 x^* 是等式约束最优化问题(18.1) 的 KKT 点, 且*

$$\lim_{k\to\infty}\left(-\sigma_k c_i\left(x^k\right)\right) = \lambda_i^*, \quad \forall i \in \mathcal{E}$$

其中 λ_i^* 是约束 $c_i(x^*) = 0$ 对应的拉格朗日乘子.

关于上述定理, 我们有如下说明。

- 不管 $\{\nabla c_i(x^*)\}$ 是否线性无关, 通过算法 1给出解 x^k 的聚点总是 $\phi(x) = \|c(x)\|^2$ 的一个稳定点. 这说明即便没有找到可行解, 我们也找到了使得约束 $c(x) = 0$ 违反度相对较小的一个解.
- 定理 18.2虽然不要求每一个子问题精确求解, 但要获得原问题的解, 子问题解的精度需要越来越高.

2.3 一般约束问题的二次罚函数

考虑不等式约束问题:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) \leqslant 0, i \in \mathcal{I} \end{aligned}$$

定义该问题的二次罚函数为:

$$P_I(x,\sigma) = f(x) + \frac{1}{2}\sigma\sum_{i\in\mathcal{I}}\tilde{c}_i^2(x)$$

其中 $\tilde{c}_i(x)$ 定义为:

$$\tilde{c}_i(x) = \max\{c_i(x), 0\}.$$

即, 我们只对违反不等式约束的部分进行惩罚。

注: $h(t) = (\min\{t,0\})^2$ 关于 t 可导, 故 $P_I(x,\sigma)$ 梯度存在, 所以可以使用梯度类算法求解。

现在考虑一般约束问题：

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, i \in \mathcal{E} \\ & c_i(x) \leqslant 0, i \in \mathcal{I}. \end{aligned} \tag{18.5}$$

定义该问题的二次罚函数为:

$$P_E(x, \sigma) = f(x) + \frac{1}{2}\sigma \left[\sum_{i \in \mathcal{E}} c_i^2(x) + \sum_{i \in \mathcal{I}} \tilde{c}_i^2(x) \right].$$

其中等式右端第二项称为惩罚项，$\tilde{c}_i(x)$ 的定义如(18.5)式，常数 $\sigma > 0$ 称为罚因子.

定理 18.1和 18.2对于一般约束问题同样成立。

3 应用举例

3.1 低秩矩阵恢复

某视频网站提供了约 48 万用户对 1 万 7 千多部电影的上亿条评级数据，希望对用户的电影评级进行预测，从而改进用户电影推荐系统，为每个用户更有针对性地推荐影片.

显然每一个用户不可能看过所有的电影，每一部电影也不可能收集到全部用户的评级. 电影评级由用户打分 1 星到 5 星表示，记为取值 1- 5 的整数. 我们将电影评级放在一个矩阵 M 中，矩阵 M 的每一行表示不同用户，每一列表示不同电影. 由于用户只对看过的电影给出自己的评价，矩阵 M 中很多元素是未知的。

	电影 1	电影 2	电影 3	电影 4	···	电影n
用户 1	4	?	?	3	···	?
用户 2	?	2	4	?	···	?
用户 3	3	?	?	?	···	?
用户 4	2	?	5	?	···	?
⋮	⋮	⋮	⋮	⋮		⋮
用户m	?	3	?	4	···	?

该问题在推荐系统、图像处理等方面有着广泛的应用。由于用户对电影的偏好可进行分类，按年龄可分为：年轻人，中年人，老年人；且电影也能分为不同的题材：战争片，悬疑片，言情片等。故这类问题隐含的假设为补全后的矩阵应为低秩的。即矩阵的行与列会有“合作”的特性，故该问题具有别名“collaborative filtering”。除此之外，由于低秩矩阵可分解为两个低秩矩阵的乘积，所以低秩限制下的矩阵补全问题是比较实用的，这样利于储存且有更好的诠释性。

由上述分析可以引出该问题：

- 令 Ω 是矩阵 M 中所有已知评级元素的下标的集合，则该问题可以初步描述为构造一个矩阵 X，使得在给定位置的元素等于已知评级元素，即满足 $X_{ij} = M_{ij},\ (i,j) \in \Omega$.

- 低秩矩阵恢复（low rank matrix completion）

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \operatorname{rank}(X), \\ \text{s.t.} \quad & X_{ij} = M_{ij},\ (i,j) \in \Omega. \end{aligned} \tag{18.6}$$

  $\operatorname{rank}(X)$ 正好是矩阵 X 所有非零奇异值的个数

- 矩阵 X 的核范数（nuclear norm）为矩阵所有奇异值的和，即：$\|X\|_* = \sum_i \sigma_i(X)$，最小化核范数可以近似的看成最小化矩阵的秩，因此我们有如下的优化问题：

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \|X\|_*, \\ \text{s.t.} \quad & X_{ij} = M_{ij},\ (i,j) \in \Omega. \end{aligned} \tag{18.7}$$

对于上述问题，我们引入等式约束的二次罚函数，

$$\min \quad \|X\|_* + \frac{\sigma}{2} \sum_{(i,j) \in \Omega} (X_{ij} - M_{ij})^2$$

令 $\sigma = \frac{1}{\mu}$，即有等价形式的优化问题：

$$\min \quad \mu \|X\|_* + \frac{1}{2} \sum_{(i,j) \in \Omega} (X_{ij} - M_{ij})^2 \tag{18.8}$$

当然，核范数是非光滑函数，我们可以用次梯度法求解上述问题，我们会在后面的课程介绍次梯度法。

Algorithm 2 矩阵补全问题求解的罚函数法

1: 给定初值 X^0，最终参数 μ，初始参数 μ_0，因子 $\gamma \in (0,1), k \leftarrow 1$
2: **while** $\mu_k \geq \mu$ **do**
3: 以 X^{k-1} 为初值，$\mu = \mu_k$ 为正则化参数求解问题(18.8)，得 X^k
4: **if** $\mu_k = \mu$ **then**
5: 停止迭代，输出 X^k
6: **else**
7: 更新罚因子 $\mu_{k+1} = \max\{\mu, \gamma\mu_k\}$
8: $k \leftarrow k + 1$
9: **end if**
10: **end while**

4 其他罚函数方法

4.1 精确罚函数方法

- 由于二次罚函数存在数值困难，并且与原问题的解存在误差，故考虑精确罚函数.
- **精确罚函数**，是一种问题求解时不需要令罚因子趋于正无穷（或零）的罚函数. 常用的精确罚函数是 ℓ_1 罚函数.
- 二次罚函数对应的问题是光滑的,ℓ_1 罚函数对应的问题是非光滑的。

定义一般约束优化问题的 ℓ_1 罚函数:

$$P(x,\sigma) = f(x) + \sigma\left[\sum_{i\in\mathcal{E}}|c_i(x)| + \sum_{i\in\mathcal{I}}\tilde{c}_i(x)\right]$$

这里用绝对值代替二次惩罚项，下面的定理揭示了 ℓ_1 罚函数的精确性。

Theorem 18.3 (精确罚函数法的收敛性) *设 x^* 是一般约束优化问题(18.5)的一个严格局部极小解，且满足 KKT 条件，其对应的拉格朗日乘子为 $\lambda_i^*, i\in\mathcal{E}\cup\mathcal{I}$，则当罚因子 $\sigma>\sigma^*$ 时，x^* 也为 $P(x,\sigma)$ 的一个局部极小解，其中*

$$\sigma^* = \|\lambda^*\|_\infty \overset{def}{=} \max_i|\lambda_i^*|.$$

另一方面，存在 $\hat{\sigma}>0$, 对于 $\sigma\geq\hat{\sigma}$，如果 $\hat{x}$ 是罚函数 $P(x,\sigma)$ 的稳定点. 那么，如果 $\hat{x}$ 是一般约束优化问题(18.5)的可行点，则 $\hat{x}$ 也满足(18.5)的 KKT 条件。

定理 18.3说明对于精确罚函数，罚因子充分大（不是正无穷），原问题的极小值点是 ℓ_1 罚函数的极小值点，这和定理 18.1是有区别的. 反之，罚函数的稳定点若是可行点，在较弱的假设下，则也是约束问题的 KKT 点。

我们有如下的传统精确罚函数方法。

Algorithm 3 精确罚函数法

1: 给定 $\sigma_1>0, x_0, k\leftarrow 0$. 罚因子增长系数 $\rho>1$.
2: **while** 未达到收敛准则 **do**
3: 以 x^{k-1} 为初始点，求解
$x^k = \arg\min_x\left\{f(x)+\sigma\left[\sum_{i\in\mathcal{E}}|c_i(x)|+\sum_{i\in\mathcal{I}}\tilde{c}_i(x)\right]\right\}$
4: 选取 $\sigma^{k+1}=\rho\sigma_k$.
5: $k\leftarrow k+1$
6: **end while**

- 取 ρ 为固定值是一种在实际中行之有效的方法，然而也可能出现：
  - 初始罚因子过小，迭代次数增加，且最优解可能远离原问题最优解
  - 罚因子过大时子问题求解困难，此时需要适当减小罚因子
- 子问题求解的初始点取法不唯一。一般取上一次子问题求解的最优值点作为下一次子问题求解的起点。

除了 ℓ_1 范数，可以用更一般的范数定义精确罚函数法：

$$P(x,\sigma) = f(x) + \mu \|c_{\mathcal{E}}(x)\| + \mu \left\| [c_{\mathcal{I}}(x)]^+ \right\|$$

其中，$\|\cdot\|$ 可以是任意的向量范数，$[c_{\mathcal{I}}(x)]^+$ 为向量各分量取 $\max\{0, x\}$。则我们可以推广定理 18.3，将 $\|\cdot\|_\infty$ 替换为 $\|\cdot\|_D$ (这里 $\|\cdot\|_D$ 表示 $\|\cdot\|$ 的对偶范数)。对偶范数的定义如下：

$$\|x\|_D = \max_{\|y\|=1} x^T y.$$

常见的对偶范数：

- $\|\cdot\|_1$ 和 $\|\cdot\|_\infty$ 互为对偶;
- ℓ_2 范数的对偶是它自身.

4.2 精确罚函数的非光滑性

下面说明，精确罚函数必然是非光滑的。
为简化讨论，假设仅有一条等式约束 $c_1(x) = 0$。设罚函数的形式为：

$$P(x,\sigma) = f(x) + \sigma h(c_1(x)).$$

其中，函数 $h : \mathbb{R} \to \mathbb{R}$ 满足 $h(y) \geq 0,\ \forall y \in \mathbb{R}$ 且 $h(0) = 0$.

若函数 h 连续可微，因为 $y = 0$ 是 $h(y)$ 最小值点，则有 $\nabla h(0) = 0$ 成立。故对于 $P(x,\sigma)$ 最优点 x^*，有

$$0 = \nabla P(x^*,\sigma) = \nabla f(x^*) + \sigma \nabla c_1(x^*) \nabla h(c_1(x^*)) = \nabla f(x^*).$$

然而，在约束优化问题中，f 取到最小值时，其梯度不一定为 0。这说明假设 h 连续可微是不正确的，即罚函数项必须是非光滑的。
另一方面，正是罚函数项的非光滑性，克服了原函数在最优点处的梯度，才能在充分大的罚因子下实现精确求解。

MATH 3011 运筹学 **秋季 2024**

Lecture 19: 邻近点梯度算法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

我们将考虑如下复合优化问题：

$$\min_{x\in\mathbb{R}^n} \quad \psi(x) = f(x) + h(x) \tag{19.1}$$

- 函数 f 为可微函数，其定义域 $\mathbf{dom}\ f = \mathbb{R}^n$
- 函数 h 为凸函数，可以是非光滑的，并且邻近算子容易计算
- LASSO 问题：$f(x) = \frac{1}{2}\|Ax-b\|^2$，$h(x) = \mu\|x\|_1$
- 次梯度法计算的复杂度：$\mathcal{O}(1/\sqrt{k})$

问题(19.1)可以用次梯度算法求解，但是次梯度方向并非下降方向，收敛速度是 $1/\sqrt{k}$。

是否可以设计复杂度为 $\mathcal{O}(1/k)$ 的算法？

2 邻近点算法

若没有非光滑函数项 $h(x)$，回顾梯度算法，我们每次用一个二次上界函数近似 $f(x)$, 即

$$x_{k+1} = \arg\min_y f(x_k) + \nabla f(x_k)^T(y-x_k) + \frac{1}{2\alpha}\|y-x_k\|^2 = x_k - \alpha\nabla f(x_k).$$

若存在 $h(x)$, 邻近点梯度算法的迭代如下：

$$\begin{aligned} x^{k+1} &= \arg\min_u \left\{ h(u) + f\left(x^k\right) + \nabla f\left(x^k\right)^{\mathrm{T}}\left(u-x^k\right) + \frac{1}{2t_k}\left\|u-x^k\right\|^2 \right\} \\ &= \arg\min_u \left\{ h(u) + \frac{1}{2t_k}\left\|u-x^k+t_k\nabla f\left(x^k\right)\right\|^2 \right\}. \end{aligned}$$

对于某些 $h(x)$，上述迭代子问题是容易求解的。该子问题和邻近算子相关。

2.1 邻近算子

定义**邻近算子**:

$$\mathrm{prox}_h(x) = \underset{u}{\mathrm{argmin}} \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

直观理解：求解一个距 x 不算太远的点 u，并使函数值 $h(u)$ 也相对较小。

Definition 19.1 一个函数被称为闭函数，如果它的上方图是闭集.

Lemma 19.1 f 是闭函数当且仅当 f 的所有 α-下水平集都是闭集。其中，α-下水平集是如下集合

$$\{x : f(x) \le \alpha\}.$$

Proposition 19.1（邻近算子是良定义的）如果 h 为闭凸函数，则对任意 x，$\mathrm{prox}_h(x)$ **存在且唯一**

Proof: 首先注意到 $h(u) + \frac{1}{2}\|u - x\|_2^2$ 是关于 u 的强凸函数，则

- 存在性:强凸函数的所有 α-下水平集有界,由 h 是闭函数可知 α-下水平集是闭集。故由 Weierstrass 定理知最小值存在
- 唯一性：强凸函数最小值唯一

■

Example 19.1 (反例) 若 C 是开凸集，那么指示函数 $\mathcal{I}_C(x)$ 的邻近算子为投影点，故邻近点不存在。

2.1.1 一些典型的例子

Example 19.2 给定 $t > 0$, $h(x) = \|x\|_1$, $\quad \mathrm{prox}_{th}(x) = \mathrm{sign}(x)\max\{|x| - t, 0\}$.

Proof: 邻近算子 $u = \mathrm{prox}_{th}(x)$ 的最优性条件为

$$x - u \in t\partial\|u\|_1 = \begin{cases} \{t\}, & u > 0 \\ [-t, t], & u = 0 \\ \{-t\}, & u < 0 \end{cases}$$

当 $x > t$ 时, $u = x - t$; 当 $x < -t$ 时, $u = x + t$; 当 $x \in [-t, t]$ 时, $u = 0$,
即有 $u = \mathrm{sign}(x)\max\{|x| - t, 0\}$, 该映射也叫做软阈值函数 *(Soft-thresholding mapping)*。 ■

Example 19.3 给定 $t>0$, $h(x)=\|x\|_2$, $\quad \mathrm{prox}_{th}(x)=\begin{cases}\left(1-\frac{t}{\|x\|_2}\right)x, & \|x\|_2\geqslant t,\\ 0, & \text{其他.}\end{cases}$

Proof: 邻近算子 $u=\mathrm{prox}_{th}(x)$ 的最优性条件为

$$x-u\in t\partial\|u\|_2=\begin{cases}\left\{\frac{tu}{\|u\|_2}\right\}, & u\neq 0,\\ \{w:\|w\|_2\leqslant t\}, & u=0,\end{cases}$$

因此, 当 $\|x\|_2>t$ 时, $u=x-\frac{tx}{\|x\|_2}$; 当 $\|x\|_2\leqslant t$ 时, $u=0$. ■

Example 19.4 二次函数 *(*其中 A 对称正定*)*

$$h(x)=\frac{1}{2}x^{\mathrm{T}}Ax+b^{\mathrm{T}}x+c,\quad \mathrm{prox}_{th}(x)=(I+tA)^{-1}(x-tb)$$

2.1.2 闭凸集上的投影

Example 19.5 设 C 为闭凸集, 则示性函数 I_C 的邻近算子为点 x 到 C 的投影 $\mathcal{P}_C(x)$:

$$\begin{aligned}\mathrm{prox}_{I_C}(x)&=\arg\min_u\left\{I_C(u)+\frac{1}{2}\|u-x\|^2\right\}\\&=\operatorname*{argmin}_{u\in C}\|u-x\|^2=\mathcal{P}_C(x)\end{aligned}$$

这个等式具有几何意义:

$$u=\mathcal{P}_C(x)\Leftrightarrow(x-u)^{\mathrm{T}}(z-u)\leqslant 0,\quad \forall z\in C$$

- **超平面** $C=\{x|a^Tx=b\}$ ($a\neq 0$)

$$P_C(x)=x+\frac{b-a^Tx}{\|a\|_2^2}a$$

- **仿射集** $C=\{x|Ax=b\}$ ($A\in\mathbb{R}^{p\times n}$, 且 $\mathrm{rank}(A)=p$)

$$P_C(x)=x+A^T(AA^T)^{-1}(b-Ax)$$

  当 $p\ll n$, 或$AA^T=I,\ldots$ 时, 计算成本较低.

- **半平面** $C=\{x|a^Tx\leq b\}$ ($a\neq 0$)

$$P_C(x)=x+\frac{b-a^Tx}{\|a\|_2^2}a\quad if\quad a^Tx>b,$$

$$P_C(x)=x\quad if\quad a^Tx\leq b$$

- **矩形**：$C = [l, u] = \{x \mid l \le x \le u\}$

$$P_C(x)_i = \begin{cases} l_i & x_i \le l_i \\ x_i & l_i \le x_i \le u_i \\ u_i & x_i \ge u_i \end{cases}$$

- **非负象限**：$C = \mathbf{R}^n_+$

$$P_C(x) = x_+ \qquad (x_+ \text{ 表示各分量取} \max\{0, x\})$$

- **概率单纯形**：$C = \{x | 1^T x = 1, x \ge 0\}$

$$P_C(x) = (x - \lambda 1)_+$$

其中，λ 是下面方程的解：

$$1^T (x - \lambda 1)_+ = \sum_{i=1}^{n} \max\{0, x_k - \lambda\} = 1.$$

- **（一般的）概率单纯形**：$C = \{x | a^T x = b, l \le x \le u\}$

$$P_c(x) = P_{[l,u]}(x - \lambda a)$$

其中, λ 是下面方程的解：

$$a^T P_{[l,u]}(x - \lambda a) = b$$

- **Euclidean 球**：$C = \{x | \|x\|_2 \le 1\}$

$$P_C(x) = \frac{1}{\|x\|_2} x \quad if \quad \|x\|_2 > 1,$$

$$P_C(x) = x \quad if \quad \|x\|_2 \le 1.$$

- ℓ_1 **范数球**：$C = \{x | \|x\|_1 \le 1\}$

$$P_c(x)_k = \begin{cases} x_k - \lambda & x_k > \lambda \\ 0 & -\lambda \le x_k \le \lambda \\ x_k + \lambda & x_k < -\lambda \end{cases}$$

若 $\|x\|_1 \le 1$，则 $\lambda = 0$; 其他情形，λ 是下面方程的解

$$\sum_{k=1}^{n} \max\{|x_k| - \lambda, 0\} = 1.$$

作业 19.1 证明 ℓ_1 范数球的投影算子如上，并给出求解 λ 的一个算法。

3 邻近点梯度算法

对于光滑部分 f 做梯度下降，对于非光滑部分 h 使用邻近算子，则邻近点梯度法的迭代格式为

$$x^{k+1} = \mathrm{prox}_{t_k h}\left(x^k - t_k \nabla f\left(x^k\right)\right) \tag{19.2}$$

其中 $t_k > 0$ 为每次迭代的步长，它可以是一个常数或者由线搜索得出.

Algorithm 1 邻近点梯度法

1: 输入：函数 $f(x), h(x)$, 初始点 x^0.
2: **while** 未达到收敛准则 **do**
3: $\quad x^{k+1} = \mathrm{prox}_{t_k h}\left(x^k - t_k \nabla f\left(x^k\right)\right)$.
4: **end while**

根据定义，(19.2)式等价于

$$\begin{aligned} x^{k+1} &= \arg\min_u \left\{ h(u) + \frac{1}{2t_k}\left\|u - x^k + t_k \nabla f\left(x^k\right)\right\|^2 \right\} \\ &= \arg\min_u \left\{ h(u) + f\left(x^k\right) + \nabla f\left(x^k\right)^{\mathrm{T}}\left(u - x^k\right) + \frac{1}{2t_k}\left\|u - x^k\right\|^2 \right\}. \end{aligned}$$

根据邻近算子与次梯度的关系, 又可以形式地写成

$$x^{k+1} = x^k - t_k \nabla f\left(x^k\right) - t_k g^k, \quad g^k \in \partial h\left(x^{k+1}\right).$$

即对光滑部分做显式的梯度下降，关于非光滑部分做隐式的梯度下降.

3.1 投影梯度法

考虑问题

$$\min_x f(x), \quad \text{s.t.} \quad x \in C.$$

集合 C 是给定的闭凸集，定义 $\mathcal{I}_C(x)$ 表示指示函数，若 $h(x) = \mathcal{I}_C(x)$。那么(19.2)可以写为

$$x^{k+1} = \mathcal{P}_C\left(x^k - t_k \nabla f\left(x^k\right)\right). \tag{19.3}$$

这便是投影梯度法，即每次先沿着负梯度方向更新以减少函数值，再投影回到可行域 C 上保证迭代点可行性。所以投影梯度法可以看成邻近点梯度法的一个特例。

3.2 收敛性分析

收敛性分析依赖于下面基本的假设。

假设 19.1

- f 在 $\mathbb{R}^n$ 上是凸的; ∇f 为 L-利普希茨连续，即

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|, \quad \forall x, y$$

- h 是适当的闭凸函数 (因此 prox_{th} 的定义是合理的);
- 函数 $\psi(x) = f(x) + h(x)$ 的最小值 ψ^* 是有限的，并且在点 x^* 处可取到 (并不要求唯一).

在基本假设的基础上，我们定义**梯度映射**:

$$G_t(x) = \frac{1}{t}\left(x - \mathrm{prox}_{th}(x - t\nabla f(x))\right) \quad (t > 0) \tag{19.4}$$

不难推出梯度映射具有以下的性质:

- "负搜索方向": $x^{k+1} = \mathrm{prox}_{th}\left(x^k - t\nabla f\left(x^k\right)\right) = x^k - tG_t\left(x^k\right)$
- 根据邻近算子和次梯度的关系，我们有

$$G_t(x) - \nabla f(x) \in \partial h\left(x - tG_t(x)\right) \tag{19.5}$$

- 与算法的收敛性的关系:

$$G_t(x) = 0 \iff x \text{ 为 } \psi(x) = f(x) + h(x) \text{ 的最小值点}$$

Theorem 19.1 (固定步长邻近点梯度法的收敛性) 取定步长为 $t_k = t \in \left(0, \frac{1}{L}\right]$，设 $\left\{x^k\right\}$ 由迭代格式(19.2)产生，则

$$\psi\left(x^k\right) - \psi^* \leqslant \frac{1}{2kt}\left\|x^0 - x^*\right\|^2.$$

Proof: 根据利普希茨连续"二次上界"的性质，得到

$$f(y) \leqslant f(x) + \nabla f(x)^{\mathrm{T}}(y - x) + \frac{L}{2}\|y - x\|^2, \quad \forall x, y \in \mathbb{R}^n.$$

令 $x^+ = x - tG_t(x)$, 有

$$\begin{aligned} f\left(x^+\right) &\leqslant f(x) - t\nabla f(x)^{\mathrm{T}} G_t(x) + \frac{t^2 L}{2}\left\|G_t(x)\right\|^2 \\ &\leqslant f(x) - t\nabla f(x)^{\mathrm{T}} G_t(x) + \frac{t}{2}\left\|G_t(x)\right\|^2. \end{aligned} \tag{19.6}$$

此外，由 $f(x), h(x)$ 为凸函数，结合(19.4)式我们有

$$h\left(x^+\right) \leqslant h(z) - \left(G_t(x) - \nabla f(x)\right)^{\mathrm{T}}\left(z - x^+\right) \tag{19.7}$$

$$f(x) \leqslant f(z) - \nabla f(x)^{\mathrm{T}}(z - x) \tag{19.8}$$

将(19.6)(19.7)(19.8)式相加可得对任意 $z \in \mathbf{dom}\ \psi$ 有

$$\psi\left(x^{+}\right) \leqslant \psi(z)+G_t(x)^{\mathrm{T}}(x-z)-\frac{t}{2}\left\|G_t(x)\right\|^2 \tag{19.9}$$

由 $x^i=x^{i-1}-tG_t(x^{i-1})$，在不等式 (19.9) 中，取 $z=x^*, x=x^{i-1}$ 得到

$$\begin{aligned}\psi(x^i)-\psi^* &\leqslant G_t(x^{i-1})^{\mathrm{T}}\left(x^{i-1}-x^*\right)-\frac{t}{2}\left\|G_t(x^{i-1})\right\|^2 \\ &=\frac{1}{2t}\left(\left\|x^{i-1}-x^*\right\|^2-\left\|x^{i-1}-x^*-tG_t(x^{i-1})\right\|^2\right) \\ &=\frac{1}{2t}\left(\left\|x^{i-1}-x^*\right\|^2-\left\|x^i-x^*\right\|^2\right)\end{aligned} \tag{19.10}$$

取 $i=1,2,\cdots,k$ 并累加，得

$$\begin{aligned}\sum_{i=1}^k\left(\psi\left(x^i\right)-\psi^*\right) &\leqslant \frac{1}{2t}\sum_{i=1}^k\left(\left\|x^{i-1}-x^*\right\|^2-\left\|x^i-x^*\right\|^2\right) \\ &=\frac{1}{2t}\left(\left\|x^0-x^*\right\|^2-\left\|x^k-x^*\right\|^2\right) \\ &\leqslant \frac{1}{2t}\left\|x^0-x^*\right\|^2.\end{aligned}$$

注意到在不等式 (19.9)中，取 $z=x^{i-1}$ 即得:

$$\psi(x^i) \leqslant \psi(x^{i-1})-\frac{t}{2}\left\|G_t(x^{i-1})\right\|^2.$$

即 $\psi\left(x^i\right)$ 不增，因此

$$\psi\left(x^k\right)-\psi^* \leqslant \frac{1}{k}\sum_{i=1}^k\left(\psi\left(x^i\right)-\psi^*\right) \leqslant \frac{1}{2kt}\left\|x^0-x^*\right\|^2.$$

■

4 应用

考虑低秩矩阵恢复模型:

$$\min_{X\in\mathbb{R}^{m\times n}} \quad \mu\|X\|_*+\frac{1}{2}\sum_{(i,j)\in\Omega}\left(X_{ij}-M_{ij}\right)^2,$$

其中 M 是想要恢复的低秩矩阵，但是只知道其在下标集 Ω 上的值. 令

$$f(X)=\frac{1}{2}\sum_{(i,j)\in\Omega}\left(X_{ij}-M_{ij}\right)^2, \quad h(X)=\mu\|X\|_*.$$

定义矩阵 $P\in\mathbb{R}^{m\times n}$:

$$P_{ij}=\begin{cases}1, & (i,j)\in\Omega, \\ 0, & \text{其他},\end{cases}$$

则

$$f(X) = \frac{1}{2}\|P \odot (X - M)\|_F^2$$

由于 $\|X\|_* = \max_{\|Y\|_2 \le 1} \langle X, Y\rangle$[1], 所以核范数 $\|\cdot\|_*$ 是凸函数。

已知有如下结论： 假设 $m \ge n$, 令 $X = USV^{\mathrm{T}}$ 为矩阵 X 的奇异值分解。记对角矩阵 $S = \mathrm{Diag}(d) \in \mathbb{R}^{n\times n}$, 对角部分 $d \in \mathbb{R}^n$ 中的分量由大到小排列 $d_1 \ge d_2 \ge \ldots \ge d_n$。若 $\mathrm{rank}(X) = r, r \le m$, 则

$$X = U_0 S_0 V_0^T, \quad U_0 \in \mathbb{R}^{m\times r}, S_0 \in \mathbb{R}^{r\times r}, V_0 \in \mathbb{R}^{n\times r}$$

这里 S_0 为保留 S 中非零的奇异值部分，即 $S_0 = \mathrm{Diag}(d_1, d_2, \ldots, d_r)$。我们有次微分集的如下表示：

$$\partial\|X\|_* = \{U_0 V_0^T + W : U_0^T W = 0, WV_0 = 0, \|W\|_2 \le 1, W \in \mathbb{R}^{m\times n}\}.$$

进一步可以验证如下结论：

$$\begin{aligned}\nabla f(X) &= P \odot (X - M),\\ \mathrm{prox}_{t_k h}(X) &= U\mathrm{Diag}\left(\max\{|d| - t_k\mu, 0\}\right)V^{\mathrm{T}},\end{aligned}$$

由此可以得到邻近点梯度法的迭代格式：

$$\begin{aligned}Y^k &= X^k - t_k P \odot \left(X^k - M\right)\\ X^{k+1} &= \mathrm{prox}_{t_k h}\left(Y^k\right)\end{aligned}$$

[1]这是因为，设 $X = U\Sigma V^T$ 为 svd 分解，则 $\|X\|_* = \sum_i \sigma_i = \langle X, UV^T\rangle$. 这表明 $\|X\|_* \le \max_{\|Y\|_2\le 1}\langle X, Y\rangle$。另一方面，$\forall Y$, 我们有 $\langle X, Y\rangle = \sum_i \sigma_i(u_i^T Y v_i) \le \sum_i \sigma_i = \|X\|_*$. 得证。

MATH 3011 运筹学 **秋季 2024**

Lecture 20: 约束优化 增广拉格朗日函数法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

考虑一般的约束优化问题，可以写成

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x), \\ \text{s.t.} \quad & c_i(x)=0, i\in\mathcal{E}, \\ & c_i(x)\leqslant 0, i\in\mathcal{I}. \end{aligned} \tag{20.1}$$

目标函数 $f(x)$ 和约束函数 $c_i(x), \forall i\in\mathcal{E},\mathcal{I}$ 都是可微的。

2 等式约束问题的增广拉格朗日函数法

2.1 二次罚函数法的数值困难

对于等式约束问题:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & c_i(x)=0, \quad i\in\mathcal{E} \end{aligned} \tag{20.2}$$

二次罚函数法需要求解最小化罚函数的子问题:

$$\min_x \quad P_E(x,\sigma)=f(x)+\frac{1}{2}\sigma\sum_{i\in\mathcal{E}}c_i^2(x).$$

由 $c_i\left(x^{k+1}\right)\approx-\frac{\lambda_i^*}{\sigma_k}$, 为了满足可行性条件, 必须使 σ_k 趋于 ∞, 这造成了子问题求解的数值困难.

我们接下来介绍的增广拉格朗日函数法可以利用有限的罚因子逼近最优解, 从而避免了上述困难.

2.2 等式约束问题的增广拉格朗日函数法

增广拉格朗日函数法的每步都需要构造增广拉格朗日函数. 根据不同的约束, 增广拉格朗日函数的形式也不同, 因此我们分别论述.

我们首先考虑 等式约束问题的增广拉格朗日函数。

Definition 20.1 对于等式约束问题(20.2), 定义**增广拉格朗日函数**为:

$$L_\sigma(x,\lambda)=f(x)+\sum_{i\in\mathcal{E}}\lambda_i c_i(x)+\frac{\sigma}{2}\sum_{i\in\mathcal{E}}c_i^2(x).$$

即在拉格朗日函数的基础上添加等式约束的二次罚函数.

由定义可得, 在第 k 步迭代, 给定罚因子 σ_k 和乘子 λ^k, $L_{\sigma_k}\left(x,\lambda^k\right)$ 的最小值点 x^{k+1} 应满足梯度条件

$$\nabla_x L_{\sigma_k}\left(x^{k+1},\lambda^k\right)=\nabla f\left(x^{k+1}\right)+\sum_{i\in\mathcal{E}}\left(\lambda_i^k+\sigma_k c_i\left(x^{k+1}\right)\right)\nabla c_i\left(x^{k+1}\right)=0. \tag{20.3}$$

我们将(20.3)式对比优化问题(20.2)满足的 KKT 条件 (对最优解 (x^*,λ^*) 的梯度条件)

$$\nabla f\left(x^*\right)+\sum_{i\in\mathcal{E}}\lambda_i^*\nabla c_i\left(x^*\right)=0, \tag{20.4}$$

为保证(20.3)和(20.4)式在最优解处的一致性, 对充分大的 k, 应满足:

$$\lambda_i^*\approx\lambda_i^k+\sigma_k c_i\left(x^{k+1}\right),\quad \forall i\in\mathcal{E}, \tag{20.5}$$

即等价于

$$c_i\left(x^{k+1}\right)\approx\frac{1}{\sigma_k}\left(\lambda_i^*-\lambda_i^k\right).$$

由此得出我们希望设计的增广拉格朗日算法具有如下的特性.

性质:

- 增广拉格朗日函数法通过合理更新乘子, 即通过控制 $\lambda_i^*-\lambda_i^k$ 降低约束违反度.
  因为根据约束违反度满足的公式, 当 λ_i^k 足够接近 λ_i^* 时, $c_i(x^{k+1})$ 将远小于 $1/\sigma_k$. 如此，避免了 σ_k 趋向无穷大，从而避免了数值困难。
- (20.5)启发我们这样更新 λ^k:

$$\lambda_i^{k+1}=\lambda_i^k+\sigma_k c_i\left(x^{k+1}\right),\quad \forall i\in\mathcal{E},$$

根据如上讨论, 并对 $c(x),\nabla c(x)$ 沿用罚函数法的定义, 我们将在下文写出等式约束问题增广拉格朗日函数法的具体算法.

在这之前我们先看一个数值例子，与二次罚函数对比，说明增广拉格朗日函数的优势。

Example 20.1 我们考虑优化问题

$$\begin{aligned} \min \quad & x+\sqrt{3}y, \\ s.t. \quad & x^2+y^2=1. \end{aligned}$$

容易求得最优解为 $x^*=\left(-\frac{1}{2},-\frac{\sqrt{3}}{2}\right)^{\mathrm{T}}$, 相应的拉格朗日乘子 $\lambda^*=1$.
根据增广拉格朗日函数的形式, 写出本问题的增广拉格朗日函数:

$$L_\sigma(x,y,\lambda)=x+\sqrt{3}y+\lambda\left(x^2+y^2-1\right)+\frac{\sigma}{2}\left(x^2+y^2-1\right)^2,$$

并在下图中绘制 $L_2(x,y,0.9)$ 的等高线.

下图中标 "$*$" 的点为原问题的最优解 x^*, 标 "$\circ$" 的点为罚函数或增广拉格朗日函数的最优解

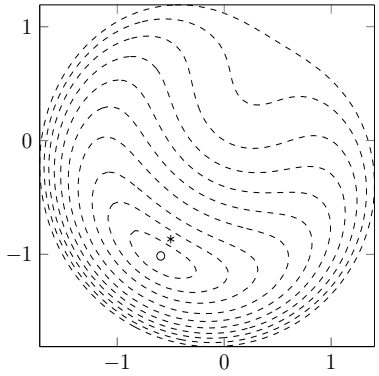


图 20.1: (a) 二次罚函数等高线

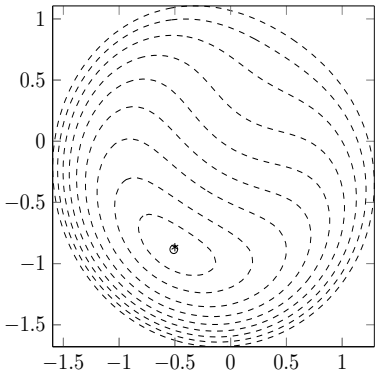


图 20.2: (b) 增广拉格朗日函数等高线

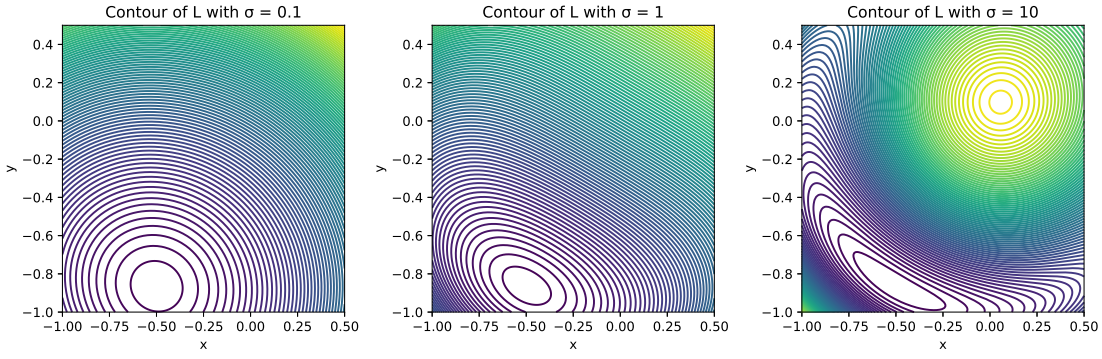


图 20.3: 增广拉格朗日函数 $L_\sigma(x,y,0.95)$ 等高线。紫色表示小的函数值, 绿色表示大的函数值。三种情况的局部最优: $\sigma=0.1:[-0.54924849-0.95132628]$; Critical point for $\sigma=1:[-0.52528898-0.9098272]$; Critical point for $\sigma=10:[-0.50453049-0.87387245]$.

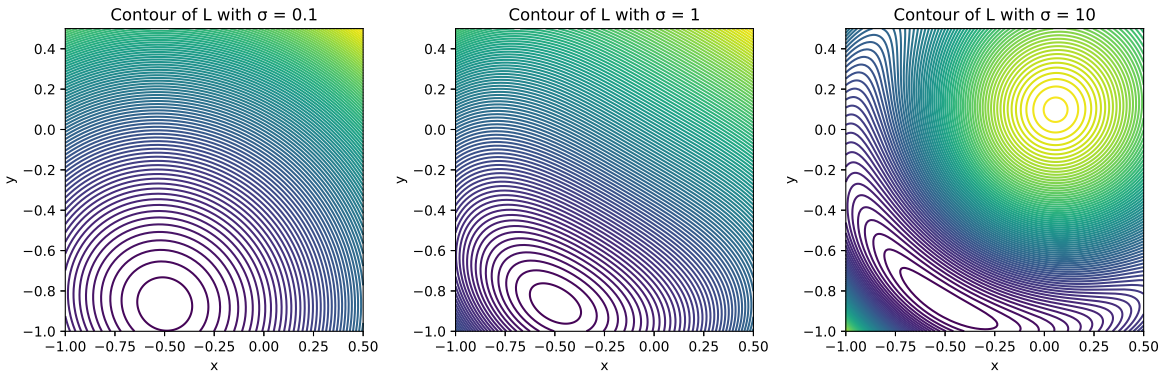


图 20.4: 增广拉格朗日函数 $L_\sigma(x, y, 1)$ 等高线。紫色表示小的函数值，绿色表示大的函数值。此时，$x = -1/2, y = -\sqrt{3}/2$ 是局部最小值点。

我们比较二次罚函数和增广拉格朗日函数在最优解探寻方面的有效性.

- 二次罚函数法求出的最优解为 $(-0.5957, -1.0319)$, 与最优解的欧氏距离约 0.1915, 约束违反度为 0.4197.

- 增广拉格朗日罚函数法求出的最优解为 $(-0.5100, -0.8833)$, 与最优解的欧氏距离约 0.02, 约束违反度为 0.0403.

- 对偶乘子 λ 更接近 KKT 对处的 λ^* 时，增广拉格朗日函数的局部极小则更接近原问题的局部极小。

由此可见, 成立如下的经验性结论.

结论：增广拉格朗日函数法可具有比二次罚函数法更精确的寻优能力, 且约束违反度一般更低.

基于上述讨论，我们给出如下等式约束的增广拉格朗日函数法。

Algorithm 1 等式约束的增广拉格朗日函数法 (ALM: Augmented Lagrangian Method)

Require: 初始坐标 $x^0 \in \mathbb{R}^n$, 乘子 λ^0, 罚因子 $\sigma_0 > 0$, 约束违反度常数 $\varepsilon > 0$, 精度 $\eta > 0$, 迭代步 $k = 0$.

Ensure: x^{k+1}, λ^k.

1: 检查初始元素.
2: **for** $k = 0, 1, 2, \cdots$ do **do**
3: 以 x^k 为初始点, 求解 $\min_x \quad L_{\sigma_k}\left(x, \lambda^k\right)$, 得到满足需求的精度条件 $\left\|\nabla_x L_{\sigma_k}\left(x, \lambda^k\right)\right\| \leqslant \eta_k$ 的解 x^{k+1}.
4: **if** $\left\|c\left(x^{k+1}\right)\right\| \leqslant \varepsilon$ 且 $\eta_k \leq \eta$ **then**
5: 返回近似解 (x^{k+1}, λ_k), 终止迭代.
6: **end if**
7: 更新对偶乘子: $\lambda^{k+1} = \lambda^k + \sigma_k c\left(x^{k+1}\right)$.
8: 更新罚因子: $\sigma_{k+1} = \rho\sigma_k$. 减小子问题精度 η_k.
9: **end for**

注 20.1

- *算法中第 3 行的意义为, 可以非精确求解关于 x 的子问题。*
- *算法第 4 行为终止条件。若找到一个可行点且为子问题的解, 则终止迭代。*
- σ_k *不应增长过快:*
  (1) 随着罚因子 σ_k 的增大, 可见 $L_{\sigma_k}(x, \lambda^k)$ 关于 x 的海瑟矩阵的条件数也将增大, 这将导致数值困难;
  (2)σ_k 与 σ_{k+1} 接近时, x^k 可以作为求解 x^{k+1} 的初始点, 以加快收敛.
- σ_k *不应增长过慢: 算法整体的收敛速度将变慢 (惩罚不足).*
- *因此在实际中, 我们应该控制 σ_k 的增长维持在一个合理的速度区间内. 一个简单的方法是维持 $\rho \in [2, 10]$, 不过近年来也有学者设计了更合理的自适应方法.*

2.3 收敛性结论

我们阐述由增广拉格朗日函数法导出的极小值点和原问题的极小值点有什么关系. 实际上, 增广拉格朗日函数在一定条件下将成为精确罚函数.

Theorem 20.1 (严格局部极小解定理) *设 x^*, λ^* 分别为问题 (20.2)的局部极小解和相应的乘子, 且点 x^* 处满足 LICQ 条件, 若 x^* 处的二阶充分条件成立. 则: 存在有限的常数 $\bar{\sigma}$, 对任意的 $\sigma \geqslant \bar{\sigma}$, x^* 都是 $L_\sigma\left(x, \lambda^*\right)$ 的严格局部极小解.*
反之, 若 x^* 为 $L_\sigma\left(x, \lambda^*\right)$ 的局部极小解且满足 $c_i\left(x^*\right) = 0, i \in \mathcal{E}$, 则 x^* 为问题 (20.2)的局部极小解.

Proof: 因为 x^* 为问题 (1) 的局部极小解且二阶充分条件成立, 所以

$$\begin{aligned} \nabla_x L\left(x^*, \lambda^*\right) &= \nabla f\left(x^*\right) + \sum_{i \in \mathcal{E}} \lambda_i^* \nabla c_i\left(x^*\right) = 0, \\ u^{\mathrm{T}} \nabla_{xx}^2 L\left(x^*, \lambda^*\right) u > 0, &\forall u \in \{u \mid \nabla c\left(x^*\right)^{\mathrm{T}} u = 0\}. \end{aligned} \tag{20.6}$$

这里

$$\nabla c\left(x^*\right) := \left(\nabla c_1(x^*), \nabla c_2(x^*), \ldots, \nabla c_{|\mathcal{E}|}(x^*)\right) \in \mathbb{R}^{n \times |\mathcal{E}|}.$$

对比 $L_\sigma\left(x^*, \lambda^*\right)$ 和 $L\left(x^*, \lambda^*\right)$ 的表达式, 由 $c_i\left(x^*\right) = 0, i \in \mathcal{E}$, 得

$$\begin{aligned} \nabla_x L_\sigma\left(x^*, \lambda^*\right) &= \nabla_x L\left(x^*, \lambda^*\right) = 0, \\ \nabla_{xx}^2 L_\sigma\left(x^*, \lambda^*\right) &= \nabla_{xx}^2 L\left(x^*, \lambda^*\right) + \sigma \sum_{i \in \mathcal{E}} \nabla c_i\left(x^*\right) \nabla c_i\left(x^*\right)^{\mathrm{T}}. \end{aligned} \tag{20.7}$$

为了证明 x^* 是 $L_\sigma\left(x^*, \lambda^*\right)$ 的严格局部极小解, 只需证对于充分大的 σ 成立

$$\nabla_{xx}^2 L_\sigma\left(x^*, \lambda^*\right) \succ 0.$$

假设该结论不成立, 则对任意 k 以及 $\sigma_k > 0$, 存在 u_k 满足 $\|u_k\| = 1$, 且满足:

$$u_k^{\mathrm{T}} \nabla_{xx}^2 L_{\sigma_k}\left(x^*, \lambda^*\right) u_k = u_k^{\mathrm{T}} \nabla_{xx}^2 L\left(x^*, \lambda^*\right) u_k + \sigma_k \left\| \nabla c\left(x^*\right)^{\mathrm{T}} u_k \right\|^2 \leqslant 0,$$

则

$$\left\| \nabla c\left(x^*\right)^{\mathrm{T}} u_k \right\|^2 \leqslant -\frac{1}{\sigma_k} u_k^{\mathrm{T}} \nabla_{xx}^2 L\left(x^*, \lambda^*\right) u_k \to 0, \quad \sigma_k \to \infty.$$

因为 $\{u_k\}$ 为有界序列, 必存在聚点, 设为 u. 那么

$$\nabla c\left(x^*\right)^{\mathrm{T}} u = 0, \quad u^{\mathrm{T}} \nabla_{xx}^2 L\left(x^*, \lambda^*\right) u \leqslant 0.$$

这与(20.6)式矛盾, 故结论成立.

反之, 若 x^* 满足 $c_i(x^*) = 0$ 且为 $L_\sigma(x, \lambda^*)$ 的局部极小解, 那么对于任意与 x^* 充分接近的可行点 x, 我们有

$$f\left(x^*\right) = L_\sigma\left(x^*, \lambda^*\right) \leqslant L_\sigma\left(x, \lambda^*\right) = f(x),$$

因此, x^* 为原问题(20.2)的一个局部极小解, 证毕. ■

关于算法的收敛性结果，我们这里只给出基于较强假设下的结论，并不讨论一般的收敛结果。

对于增广拉格朗日方法, 通过进一步假设乘子点列的有界性和收敛点处的约束正则条件, 算法迭代生成的序列 $\{x^k\}$ 会有子列收敛至问题(20.2)的一阶稳定点.

Theorem 20.2 (增广拉格朗日函数法的收敛性) 假设乘子列 $\left\{\lambda^k\right\}$ 是有界的, 罚因子 $\sigma_k \to +\infty$, $k \to \infty$, 上述增广拉格朗日方法中精度 $\eta_k \to 0$, 迭代点列 $\left\{x^k\right\}$ 的一个子序列 $\left\{x^{k_j+1}\right\}$ 收敛到 x^* , 并且在

点 x^* 处 $LICQ$ 成立.
那么存在 λ^*，满足:

$$\lambda^{k_j+1} \to \lambda^*, \quad j \to \infty,$$

$$\nabla f(x^*) + \nabla c(x^*)\lambda^* = 0, \quad c(x^*) = 0.$$

该定理中的关于乘子列 $\{\lambda^k\}$ 是有界假设比较强，需要对具体问题具体证明。我们不过多赘述。

3 一般约束问题的增广拉格朗日函数法

一般的约束优化问题可以写成

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & c_i(x) = 0, i \in \mathcal{E}, \\ & c_i(x) \leqslant 0, i \in \mathcal{I}. \end{aligned} \tag{20.8}$$

对于问题(20.8), 我们一般引入松弛变量 $s \in \mathbb{R}^{|\mathcal{I}|}$, 得到如下等价形式:

$$\begin{aligned} \min_{x,s} \quad & f(x), \\ \text{s.t.} \quad & c_i(x) = 0, i \in \mathcal{E}, \\ & c_i(x) + s_i = 0, i \in \mathcal{I}, \\ & s_i \geqslant 0, i \in \mathcal{I}. \end{aligned} \tag{20.9}$$

这样的做法我们已经用过多次了, 应熟练掌握.

保留关于 s 的非负约束, 可以构造拉格朗日函数

$$L(x, s, \lambda, \mu) = f(x) + \sum_{i\in\mathcal{E}} \lambda_i c_i(x) + \sum_{i\in\mathcal{I}} \mu_i \left(c_i(x) + s_i\right), s_i \geqslant 0, i \in \mathcal{I}.$$

记问题(20.9)中等式约束的二次罚函数为 $p(x, s)$, 即

$$p(x, s) = \sum_{i\in\mathcal{E}} c_i^2(x) + \sum_{i\in\mathcal{I}} \left(c_i(x) + s_i\right)^2,$$

那么可以同样构造增广拉格朗日函数如下:

$$\begin{aligned} L_\sigma(x, s, \lambda, \mu) =& f(x) + \sum_{i\in\mathcal{E}} \lambda_i c_i(x) + \sum_{i\in\mathcal{I}} \mu_i \left(c_i(x) + s_i\right) + \frac{\sigma}{2} p(x, s), \\ & s_i \geqslant 0, i \in \mathcal{I}. \end{aligned}$$

下面我们将看到，添加这种非负约束变量，子问题同样可以求解。

3.1 求解非负约束的子问题

在第 k 步迭代中, 给定乘子 λ^k, μ^k 和罚因子 σ_k, 需要求解如下问题:

$$\min_{x,s} \quad L_{\sigma_k}\left(x, s, \lambda^k, \mu^k\right), \quad \text{s.t.} \quad s \geqslant 0, \tag{20.10}$$

可以得到 x^{k+1}, s^{k+1}. 我们现在介绍一种基于消元的方法, 即考虑消去 s, 求解只关于 x 的优化问题.

1. 首先, 固定 x, 关于 s 的子问题化为

$$\min_{s \geqslant 0} \quad \sum_{i \in \mathcal{I}} \mu_i \left(c_i(x) + s_i\right) + \frac{\sigma_k}{2} \sum_{i \in \mathcal{I}} \left(c_i(x) + s_i\right)^2.$$

   容易直接解得使子问题最优且满足非负约束的 s_i 为

$$s_i = \max\left\{-\frac{\mu_i}{\sigma_k} - c_i(x), 0\right\}, \quad i \in \mathcal{I}. \tag{20.11}$$

2. 将 s_i 的表达式代入 L_{σ_k} 我们有

$$L_{\sigma_k}\left(x, \lambda^k, \mu^k\right) = f(x) + \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\sigma_k}{2} \sum_{i \in \mathcal{E}} c_i^2(x) + \frac{\sigma_k}{2} \sum_{i \in \mathcal{I}} \left(\max\left\{\frac{\mu_i}{\sigma_k} + c_i(x), 0\right\}^2 - \frac{\mu_i^2}{\sigma_k^2}\right).$$

   其为关于 x 的连续可微函数 (假设 $f(x), c_i(x), i \in \mathcal{I} \cup \mathcal{E}$ 连续可微). 因此, 问题(20.10)等价于

$$\min_{x \in \mathbb{R}^n} \quad L_{\sigma_k}\left(x, \lambda^k, \mu^k\right).$$

   并可以利用梯度法进行求解.

注意: 这里, 我们消去了变量 s, 因此可以只考虑关于 x 的优化问题.

3.2 更新对偶乘子

对于问题(20.9), 其最优解 x^*, s^* 和乘子 λ^*, μ^* 需满足 KKT 条件:

$$\begin{aligned} &0 = \nabla f\left(x^*\right) + \textstyle\sum_{i \in \mathcal{E}} \lambda_i^* \nabla c_i\left(x^*\right) + \sum_{i \in \mathcal{I}} \mu_i^* \nabla c_i\left(x^*\right), \\ &\mu_i^* \geqslant 0, \ s_i^* \geqslant 0, i \in \mathcal{I} \end{aligned}$$

问题(20.10)的最优解 x^{k+1}, s^{k+1} 满足

$$0 = \nabla f\left(x^{k+1}\right) + \sum_{i \in \mathcal{E}} \left(\lambda_i^k + \sigma_k c_i\left(x^{k+1}\right)\right) \nabla c_i\left(x^{k+1}\right) +$$

$$\sum_{i\in\mathcal{I}}\left(\mu_i^k+\sigma_k\left(c_i\left(x^{k+1}\right)+s_i^{k+1}\right)\right)\nabla c_i\left(x^{k+1}\right),$$

$$s_i^{k+1}=\max\left\{-\frac{\mu_i^k}{\sigma_k}-c_i\left(x^{k+1}\right),0\right\},\quad i\in\mathcal{I}.$$

对比问题(20.9)和问题(20.10)的 KKT 条件, 易知乘子的更新格式为

$$\begin{aligned}\lambda_i^{k+1}&=\lambda_i^k+\sigma_k c_i\left(x^{k+1}\right),\quad i\in\mathcal{E},\\ \mu_i^{k+1}&=\max\left\{\mu_i^k+\sigma_k c_i\left(x^{k+1}\right),0\right\},\quad i\in\mathcal{I}.\end{aligned}\tag{20.12}$$

3.3 约束违反度

对于等式约束, 我们定义约束违反度为

$$v_k\left(x^{k+1}\right)=\sqrt{\sum_{i\in\mathcal{E}}c_i^2\left(x^{k+1}\right)+\sum_{i\in\mathcal{I}}\left(c_i\left(x^{k+1}\right)+s_i^{k+1}\right)^2}.$$

根据 (20.11) 式消去 s, 得

$$v_k\left(x^{k+1}\right)=\sqrt{\sum_{i\in\mathcal{E}}c_i^2\left(x^{k+1}\right)+\sum_{i\in\mathcal{I}}\max\left\{c_i\left(x^{k+1}\right),-\frac{\mu_i^k}{\sigma_k}\right\}^2}.$$

在算法中, 需要根据约束违反度的大小判断参数的更新方式:

- 若 $v_k\left(x^{k+1}\right)$ 满足精度条件, 则进行乘子的更新, 并提高子问题求解精度, 罚因子不变;
- 若不满足, 则不进行乘子的更新, 并适当增大罚因子以便得到约束违反度更小的解.

综上，一般约束的增广拉格朗日函数算法描述见算法 2.

4 应用：基追踪问题

基追踪（Basis Pursuit）的应用发展历史与压缩感知（Compressed Sensing）和信号处理的历史紧密相关。下面是其应用发展的简要概述：

1. 20 世纪末的理论基础：基追踪的理论基础在 20 世纪末由数学和工程领域的研究者建立。最初，它是作为一种稀疏表示和信号恢复的技术而被探索。
2. 压缩感知的兴起（2000 年代初）：2006 年，Emmanuel Candes、Terence Tao 和 David Donoho 提出了压缩感知理论，这标志着基追踪应用的一个重要转折点。压缩感知理论显示，在某些条件下，可以从远少于 Nyquist 采样定理要求的样本中恢复稀疏信号，而基追踪成为实现这一目标的关键技术之一。

Algorithm 2 一般约束增广拉格朗日函数法 (ALM: Augmented Lagrangian Method)

Require: 选取初始点 x^0, 乘子 λ^0, μ^0, 罚因子 $\sigma_0 > 0$, 约束违反度常数 $\varepsilon > 0$, 精度常数 $\eta > 0$, 以及常数 $0 < \alpha \leqslant \beta \leqslant 1$ 和 $\rho > 1$. 令 $\eta_0 = \frac{1}{\sigma_0}, \varepsilon_0 = \frac{1}{\sigma_0^\alpha}$ 以及 $k = 0$.

Ensure: 输出 x^{k+1}, λ^k.

检查初始元素.

for $k = 0, 1, 2, \cdots$ do **do**

以 x^k 为初始点, 求解

$$\min_x \quad L_{\sigma_k}\left(x, \lambda^k, \mu^k\right),$$

得到满足精度条件

$$\left\|\nabla L_{\sigma_k}\left(x^{k+1}, \lambda^k, \mu^k\right)\right\|_2 \leqslant \eta_k$$

的解 x^{k+1}.

if $v_k\left(x^{k+1}\right) \leqslant \varepsilon_k$ **then**

if $v_k\left(x^{k+1}\right) \leqslant \varepsilon$ 且 $\left\|\nabla_x L_{\sigma_k}\left(x^{k+1}, \lambda^k, \mu^k\right)\right\|_2 \leqslant \eta$ **then**

得到逼近解 $x^{k+1}, \lambda^k, \mu^k$, 终止迭代

end if

更新乘子:

$$\lambda_i^{k+1} = \lambda_i^k + \sigma_k c_i\left(x^{k+1}\right), \quad i \in \mathcal{E},$$
$$\mu_i^{k+1} = \max\left\{\mu_i^k + \sigma_k c_i\left(x^{k+1}\right), 0\right\}, \quad i \in \mathcal{I}.$$

罚因子不变: $\sigma_{k+1} = \sigma_k$.

减小子问题求解误差和约束违反度:

$$\eta_{k+1} = \frac{\eta_k}{\sigma_{k+1}}, \quad \varepsilon_{k+1} = \frac{\varepsilon_k}{\sigma_{k+1}^\beta}.$$

else

乘子不变: $\lambda^{k+1} = \lambda^k$.

更新罚因子: $\sigma_{k+1} = \rho\sigma_k$.

调整子问题求解误差和约束违反度:

$$\eta_{k+1} = \frac{1}{\sigma_{k+1}}, \quad \varepsilon_{k+1} = \frac{1}{\sigma_{k+1}^\alpha}.$$

end if

end for

3. 信号处理和图像重构：随后，基追踪在信号处理和图像重构领域得到了广泛应用。它被用于从有限的观测数据中恢复图像和信号，特别是在 MRI（磁共振成像）和雷达成像等领域。
4. 机器学习和数据科学：随着机器学习和数据科学的发展，基追踪也被应用于这些领域，特别是在特征选择和稀疏建模方面。它帮助分析和处理高维数据集，提高了模型的解释性和效率。
5. 算法和计算方法的进步：为了应对基追踪中的计算挑战，研究者开发了多种算法，如正交匹配追踪（OMP）、迭代阈值算法等，提高了问题求解的效率和可行性。
6. 其他领域的应用：基追踪还被应用于无线通信、生物信息学、金融数据分析等领域，展示了其在处理复杂和高维数据问题上的广泛潜力。

总的来说，基追踪的发展和应用反映了现代科学和工程中对高效、稀疏数据表示和处理方法的持续需求。随着技术的不断进步，预计基追踪将在更多领域发挥重要作用。

基追踪是一个数学上的优化问题，它的目的是从一组过完备基（overcomplete basis）中选择出最少的基元素，以便这些基元素的线性组合可以精确地或近似地表示给定的信号。基追踪问题可以被表述为一种特殊的优化问题，其核心在于寻找最稀疏的解。

考虑一类简单的基追踪问题. 设 $A \in \mathbb{R}^{m\times n}(m \leqslant n)$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, 基追踪问题被描述为

$$\min_{x\in\mathbb{R}^n} \quad \|x\|_1, \qquad \text{s.t.} \qquad Ax = b. \tag{20.13}$$

这里, ℓ_1 范数用来使得 x 尽可能稀疏。

4.1 原问题的增广拉格朗日函数法

根据问题(20.13)的形式, 引入罚因子 σ 和乘子 λ, 其增广拉格朗日函数为

$$L_\sigma(x,\lambda) = \|x\|_1 + \lambda^{\mathrm{T}}(Ax-b) + \frac{\sigma}{2}\|Ax-b\|_2^2. \tag{20.14}$$

固定 σ, 第 k 步迭代更新格式为

$$\begin{cases} x^{k+1} = \arg\min\limits_{x\in\mathbb{R}^n} \left\{ \|x\|_1 + \frac{\sigma}{2}\left\|Ax - b + \frac{\lambda^k}{\sigma}\right\|_2^2 \right\}, \\ \lambda^{k+1} = \lambda^k + \sigma\left(Ax^{k+1} - b\right). \end{cases} \tag{20.15}$$

这里，关于 x 的子问题是一个类似 LASSO（least absolute shrinkage and selection operator）的非光滑问题，我们可以使用邻近点梯度法求解。

Theorem 20.3 简单基追踪问题的收敛性定理

假设问题 (20.13)的可行域非空, 迭代序列 $\{x^k\}, \{\lambda^k\}$ 是由迭代格式 (20.15) 从初始点 $x^0 = \lambda^0 = 0$ 产生的, 则存在正整数 K 使得任意的 $x^k, k \geqslant K$ 是问题(20.13)的解.

4.2 对偶问题的增广拉格朗日函数法

考虑其对偶问题:

$$\min_{y\in\mathbb{R}^m} \quad b^{\mathrm{T}}y, \quad \text{s.t.} \quad \left\|A^{\mathrm{T}}y\right\|_\infty \leqslant 1. \tag{20.16}$$

通过引入变量 s, 对偶问题可以等价地写成

$$\min_{y\in\mathbb{R}^m,\ s\in\mathbb{R}^n} b^{\mathrm{T}}y, \quad \text{s.t.} \quad A^{\mathrm{T}}y - s = 0,\ \|s\|_\infty \leqslant 1. \tag{20.17}$$

现在我们考虑另一重要的问题. 设对偶问题(20.17):

$$\min_{y\in\mathbb{R}^m, s\in\mathbb{R}^n} \quad b^{\mathrm{T}}y, \quad \text{s.t.} \quad A^{\mathrm{T}}y - s = 0, \quad \|s\|_\infty \leqslant 1.$$

引入拉格朗日乘子 λ 和罚因子 σ, 作增广拉格朗日函数

$$L_\sigma(y,s,\lambda) = b^{\mathrm{T}}y + \lambda^{\mathrm{T}}\left(A^{\mathrm{T}}y - s\right) + \frac{\sigma}{2}\left\|A^{\mathrm{T}}y - s\right\|_2^2, \quad \|s\|_\infty \leqslant 1.$$

增广拉格朗日函数法的迭代格式为 ($\rho > 1$ 和 $\bar{\sigma} < +\infty$ 为算法参数):

$$\begin{cases} \left(y^{k+1}, s^{k+1}\right) = \underset{y,\|s\|_\infty\leqslant 1}{\arg\min}\left\{b^{\mathrm{T}}y + \dfrac{\sigma_k}{2}\left\|A^{\mathrm{T}}y - s + \dfrac{\lambda}{\sigma_k}\right\|_2^2\right\}, \\ \lambda^{k+1} = \lambda^k + \sigma_k\left(A^{\mathrm{T}}y^{k+1} - s^{k+1}\right), \\ \sigma_{k+1} = \min\left\{\rho\sigma_k, \bar{\sigma}\right\}. \end{cases}$$

其中 $\left(y^{k+1}, s^{k+1}\right)$ 的显式表达式未知, 需要迭代求解.

除了利用投影梯度法求解关于 (y,s) 的联合最小化问题外, 还可以利用最优性条件将 s 用 y 来表示, 转而求解只关于 y 的最小化问题.

先只考虑关于 s 的极小化问题

$$\min_s \quad \frac{\sigma}{2}\left\|A^{\mathrm{T}}y - s + \frac{\lambda}{\sigma}\right\|_2^2, \qquad \textbf{s.t.} \quad \|s\|_\infty \leqslant 1.$$

这是一个关于 s 的二次型函数, 因此问题的解为

$$s = \mathcal{P}_{\|s\|_\infty\leqslant 1}(A^{\mathrm{T}}y + \frac{\lambda}{\sigma}),$$

其中 $\mathcal{P}_{\|s\|_\infty\leqslant 1}(z)$ 为集合 $\{s|\,\|s\|_\infty \leqslant 1\}$ 的**投影算子**, 即

$$\mathcal{P}_{\|s\|_\infty\leqslant 1}(z) = \max\left\{\min\left\{z,\ 1\right\},\ -1\right\}.$$

将上述 s 的表达式代入的增广拉格朗日函数法的迭代格式, 得

$$\begin{cases} y^{k+1} = \underset{y}{\arg\min}\left\{b^{\mathrm{T}}y + \frac{\sigma}{2}\left\|\psi\left(A^{\mathrm{T}}y + \frac{\lambda}{\sigma}\right)\right\|_2^2\right\}, \\ \lambda^{k+1} = \sigma_k\psi\left(A^{\mathrm{T}}y^{k+1} + \frac{\lambda^k}{\sigma_k}\right), \\ \sigma_{k+1} = \min\left\{\rho\sigma_k, \bar{\sigma}\right\}. \end{cases} \tag{20.18}$$

其中 $\psi(x) = \text{sign}(x)\max\{|x| - 1, 0\}$.

性质：

- 我们不能得到关于 y^{k+1} 的显式表达式. 但是由于 $L_{\sigma_k}\left(y, \lambda^k\right)$ 关于 y 连续可微, 故可以利用梯度法求解.
- 注意 y 的维度是 m, 原问题 x 的变量维度是 n. 若 $m < n$, 则求解对偶问题更有优势。

MATH 3011 运筹学 **秋季 2024**

Lecture 21: 交替方向乘子法

Lecturer: 陈士祥 *Scribes:* 陈士祥

1 问题形式

交替方向乘子法 (ADMM) 是一种常用的优化算法，适用于解决具有特定结构的优化问题，特别是可以分解为多个子问题的情况。主要可以处理如下可分的凸问题：

$$\begin{aligned} \min_{x_1,x_2} \quad & f_1(x_1) + f_2(x_2), \\ \text{s.t.} \quad & A_1x_1 + A_2x_2 = b, \end{aligned} \tag{21.1}$$

- f_1, f_2 是适当的闭凸函数，不需要是光滑的，$x_1 \in \mathbb{R}^n, x_2 \in \mathbb{R}^m$, $A_1 \in \mathbb{R}^{p\times n}, A_2 \in \mathbb{R}^{p\times m}$, $b \in \mathbb{R}^p$.
- 可分：目标函数可以分成两个变量独占的函数，但是变量被线性约束结合在一起. 常见的一些无约束和带约束的优化问题都可以表示成这一形式。

之前的课程中，增广拉格朗日函数法亦可以求解此类问题，但是其子问题难以求解。本节中，ADMM 可以看作是基于增广拉格朗日函数法的修改算法。

2 交替方向乘子法

Example 21.1 例

- 考虑如下问题

$$\min_{x} \quad f_1(x) + f_2(x).$$

若 f_1, f_2 都是非光滑函数，之前学过的算法，例如近似点梯度法，无法处理该问题。

引入一个新的变量 z 并令 $x = z$，将问题转化为 (21.1)的形式：

$$\begin{aligned} \min_{x,z} \quad & f_1(x) + f_2(z), \\ \text{s.t.} \quad & x - z = 0. \end{aligned}$$

- 带线性变换的无约束优化问题

$$\min_{x} \quad f_1(x) + f_2(Ax).$$

可以引入一个新的变量 z，令 $z = Ax$，则问题变为

$$\begin{aligned} \min_{x,z} \quad & f_1(x) + f_2(z), \\ \text{s.t.} \quad & Ax - z = 0. \end{aligned}$$

同样转化为 (21.1)的形式。

- 全局一致性问题

$$\min_{x} \quad \sum_{i=1}^{N} \phi_i(x).$$

令 $x = z$，并将 x 复制 N 份，分别为 x_i，那么问题转化为

$$\begin{aligned} \min_{x_i,z} \quad & \sum_{i=1}^{N} \phi_i(x_i), \\ \text{s.t.} \quad & x_i - z = 0,\ i = 1, 2, \cdots, N. \end{aligned}$$

2.1 回顾：增广拉格朗日函数法

- 首先写出问题(21.1)的增广拉格朗日函数

$$\begin{aligned} L_\rho(x_1, x_2, y) =& f_1(x_1) + f_2(x_2) + y^{\mathrm{T}}(A_1x_1 + A_2x_2 - b) \\ &+ \frac{\rho}{2}\|A_1x_1 + A_2x_2 - b\|_2^2, \end{aligned} \tag{21.2}$$

其中 $\rho > 0$ 是二次罚项的系数.

- 增广拉格朗日函数法 (ALM) 为如下更新：

$$(x_1^{k+1}, x_2^{k+1}) = \arg\min_{x_1,x_2} L_\rho(x_1, x_2, y^k), \tag{21.3}$$

$$y^{k+1} = y^k + \tau\rho(A_1x_1^{k+1} + A_2x_2^{k+1} - b), \tag{21.4}$$

其中 τ 为步长.

ALM 的缺点：子问题(21.3)不易求解。

2.2 交替乘子方向法

英文名：Alternating direction method of multipliers, 简称 ADMM

- 交替方向乘子法的基本思路: 第一步迭代(21.3)同时对 x_1 和 x_2 进行求解有时候比较困难，而固定一个变量求解关于另一个变量的极小问题可能比较简单，因此我们可以考虑对 x_1 和 x_2 交替求极小

- 其迭代格式可以总结如下:

$$x_1^{k+1} = \arg\min_{x_1} L_\rho(x_1, x_2^k, y^k), \tag{21.5}$$

$$x_2^{k+1} = \arg\min_{x_2} L_\rho(x_1^{k+1}, x_2, y^k), \tag{21.6}$$

$$y^{k+1} = y^k + \tau\rho(A_1x_1^{k+1} + A_2x_2^{k+1} - b), \tag{21.7}$$

其中 τ 为步长，通常取值于 $\left(0, \frac{1+\sqrt{5}}{2}\right]$

2.3 应用举例

Example 21.2 (基追踪问题) 对于基追踪问题. 设 $A \in \mathbb{R}^{m\times n}(m \leqslant n)$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, 基追踪问题被描述为

$$\min_{x\in\mathbb{R}^n} \quad \|x\|_1, \quad s.t. \quad Ax = b. \tag{21.8}$$

ALM 迭代更新格式为

$$\begin{cases} x^{k+1} = \arg\min\limits_{x\in\mathbb{R}^n} \left\{ \|x\|_1 + \frac{\sigma}{2} \left\| Ax - b + \frac{\lambda^k}{\sigma} \right\|_2^2 \right\}, \\ \lambda^{k+1} = \lambda^k + \sigma\left(Ax^{k+1} - b\right). \end{cases} \tag{21.9}$$

引入 $y = x$, 问题变为

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^n} \quad \|y\|_1, \quad s.t. \quad Ax = b, \quad x = y. \tag{21.10}$$

ADMM 迭代更新格式为

$$\begin{cases} x^{k+1} = \arg\min\limits_{x\in\mathbb{R}^n} \left\{ \langle\lambda_{1,k}, Ax - b\rangle + \frac{\sigma}{2}\|Ax - b\|_2^2 + \langle\lambda_{2,k}, y_k - x\rangle + \frac{\sigma}{2}\|y_k - x\|_2^2 \right\}, \\ y^{k+1} = \arg\min\limits_{y\in\mathbb{R}^n} \left\{ \|y\|_1 + \langle\lambda_{2,k}, y - x_{k+1}\rangle + \frac{\sigma}{2}\|y - x_{k+1}\|_2^2 \right\}, \\ \lambda_{1,k+1} = \lambda_{1,k} + \sigma\left(Ax^{k+1} - b\right), \\ \lambda_{2,k+1} = \lambda_{2,k} + \sigma\left(y_{k+1} - x_{k+1}\right) \end{cases} \tag{21.11}$$

Example 21.3 (LASSO 问题) *LASSO* 问题

$$\min \quad \mu\|x\|_1 + \frac{1}{2}\|Ax - b\|^2.$$

转换为标准问题形式：

$$\begin{aligned}\min_{x,z} \quad & \frac{1}{2}\|Ax-b\|^2+\mu\|z\|_1,\\ \text{s.t.} \quad & x=z.\end{aligned}$$

ADMM 迭代格式为

$$\begin{aligned}x^{k+1} &= \operatorname*{argmin}_x\left\{\frac{1}{2}\|Ax-b\|^2+\frac{\rho}{2}\|x-z^k+y^k/\rho\|_2^2\right\},\\ &= (A^{\mathrm{T}}A+\rho I)^{-1}(A^{\mathrm{T}}b+\rho z^k-y^k),\\ z^{k+1} &= \operatorname*{argmin}_z\left\{\mu\|z\|_1+\frac{\rho}{2}\|x^{k+1}-z+y^k/\rho\|^2\right\},\\ &= \operatorname{prox}_{(\mu/\rho)\|\cdot\|_1}\left(x^{k+1}+y^k/\rho\right),\\ y^{k+1} &= y^k+\tau\rho(x^{k+1}-z^{k+1}).\end{aligned}$$

对于 x_k 的子问题，有如下方式减少每个迭代步的计算量：

- 因为 $\rho>0$，所以 $A^{\mathrm{T}}A+\rho I$ 总是可逆的．若使用固定的罚因子 ρ，我们使用例如 *Cholesky* 分解得到 $A^{\mathrm{T}}A+\rho I$ 的初始分解，从而减小后续迭代中的计算量．

- 在 *LASSO* 问题中，矩阵 $A\in\mathbb{R}^{m\times n}$ 通常有较多的列（即 $m\ll n$），因此 $A^{\mathrm{T}}A\in\mathbb{R}^{n\times n}$ 是一个低秩矩阵，二次罚项的作用就是将 $A^{\mathrm{T}}A$ 增加了一个正定项．该 *ADMM* 主要运算量来自更新 x 变量时求解线性方程组，复杂度为 $O(n^3)$

- 可以利用 *SMW* 公式减少矩阵求逆计算量：

$$(A^{\mathrm{T}}A+\rho I_n)^{-1}=\rho^{-1}I-\rho^{-1}A^{\mathrm{T}}(\rho I_m+AA^{\mathrm{T}})^{-1}A$$

Example 21.4 (Fused LASSO 问题) 对许多问题 x 本身不稀疏，但在某种变换下是稀疏的*:*

$$\min_x \quad \mu\|Dx\|_1+\frac{1}{2}\|Ax-b\|^2. \tag{21.12}$$

一个重要的例子是当 $D\in\mathbb{R}^{(n-1)\times n}$ 是一阶差分矩阵

$$D_{ij}=\begin{cases}1, & j=i+1,\\ -1, & j=i,\\ 0, & \text{其他},\end{cases}$$

且 $A=I$ 时，广义 *LASSO* 问题为

$$\min_x \quad \frac{1}{2}\|x-b\|^2+\mu\sum_{i=1}^{n-1}|x_{i+1}-x_i|,$$

这个问题就是图像去噪问题模型.

通过引入约束 $Dx = z$:

$$\begin{aligned} \min_{x,z} \quad & \frac{1}{2}\|Ax - b\|^2 + \mu\|z\|_1, \\ \text{s.t.} \quad & Dx - z = 0, \end{aligned} \tag{21.13}$$

引入乘子 y，其增广拉格朗日函数为

$$L_\rho(x, z, y) = \frac{1}{2}\|Ax - b\|^2 + \mu\|z\|_1 + y^{\mathrm{T}}(Dx - z) + \frac{\rho}{2}\|Dx - z\|^2.$$

此问题的 x 迭代是求解方程组

$$(A^{\mathrm{T}}A + \rho D^{\mathrm{T}}D)x = A^{\mathrm{T}}b + \rho D^{\mathrm{T}}\left(z^k - \frac{y^k}{\rho}\right),$$

而 z 迭代依然通过 ℓ_1 范数的邻近算子.

- 因此交替方向乘子法所产生的迭代为

$$\begin{aligned} x^{k+1} &= (A^{\mathrm{T}}A + \rho D^{\mathrm{T}}D)^{-1}\left(A^{\mathrm{T}}b + \rho D^{\mathrm{T}}\left(z^k - \frac{y^k}{\rho}\right)\right), \\ z^{k+1} &= \mathrm{prox}_{(\mu/\rho)\|\cdot\|_1}\left(Dx^{k+1} + \frac{y^k}{\rho}\right), \\ y^{k+1} &= y^k + \tau\rho(Dx^{k+1} - z^{k+1}). \end{aligned}$$

- 对于全变差去噪问题，$A^{\mathrm{T}}A + \rho D^{\mathrm{T}}D$ 是三对角矩阵，所以此时 x 迭代可以在 $\mathcal{O}(n)$ 的时间复杂度内解决；对于图像去模糊问题，A 是卷积算子，则利用傅里叶变换可将求解方程组的复杂度降低至 $O(n \log n)$.

2.3.1 图像去噪模型

图像去噪是图像处理领域的一个重要任务，旨在从损坏的或嘈杂的图像中恢复出清晰的图像。去噪模型的目标是在尽可能保留图像细节和结构的同时，移除噪声。其数学模型如下：

$$b = Kx_t + w$$

- x_t 为未知图像
- b 为观察到的图像，模糊且有噪声; w 为噪声
- $N \times N$ 的像素点按列储存为长为 N^2 的向量

模糊矩阵 K

- 表示一个 2 维的卷积，是有空间不动点的扩散函数
- 满足周期边界条件，有循环块 (circulant blocks)
- 可对角化，即存在酉的 2 维离散傅立叶变换矩阵 W，使得

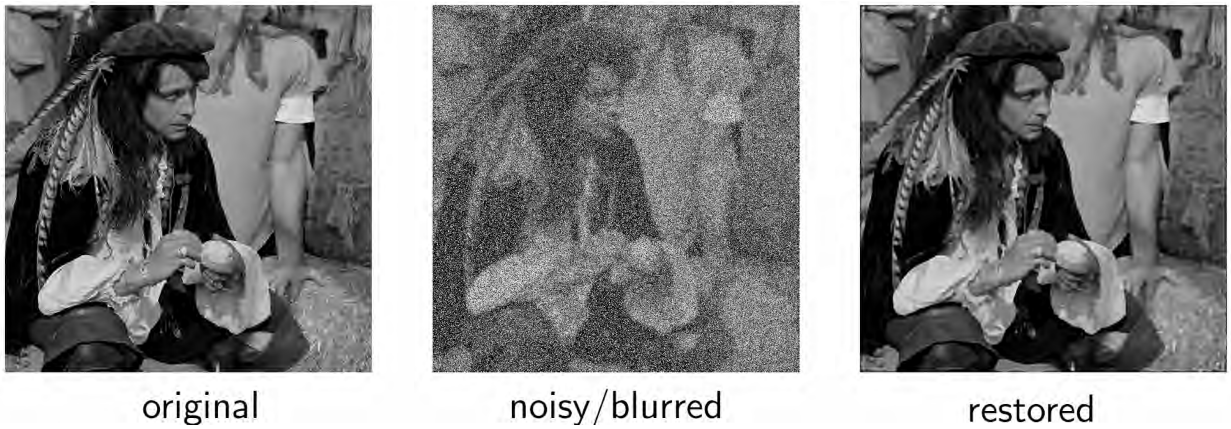
$$K = W^H \mathbf{diag}(\lambda) W.$$

系数矩阵为 $I + K^T K$ 的线性方程组可在 $O(N^2 \log N)$ 的时间内求解。

我们有如下图像添加噪声和去噪声例子：求解下面问题，以恢复出带噪声/模糊的图片，

$$\min_x \frac{1}{2}\|Kx - b\|^2 + \|Dx\|_1.$$

- b 为给定的带噪声的图像，如下图中间的图片，1024 × 1024 的图像，满足周期边界条件
- 高斯模糊算子 K
- 椒盐噪声 (salt-and-pepper noise) w：50% 的像素点被随机替换为 0/1



2.3.2 鲁棒主成分分析

在经典的 PCA 中，数据被分解为几个主成分，这些主成分捕获了数据中的主要变异性。然而，当数据中存在离群点或异常值时，PCA 的性能可能会大大下降，因为它试图捕获所有数据点的变异性，包括异常值。

RPCA 解决了这个问题。它将数据矩阵分解为两部分：一个低秩矩阵和一个稀疏矩阵。低秩矩阵捕获数据的主要结构，而稀疏矩阵则包含异常值或离群点。通过这种方式，RPCA 能够在保持数据主要结构的同时，有效地处理异常值。

其数学模型如下

$$\begin{aligned} \min_{X,S} \quad & \|X\|_* + \mu\|S\|_1, \\ & X + S = M, \end{aligned} \tag{21.14}$$

其中 $\|\cdot\|_1$ 与 $\|\cdot\|_*$ 分别表示矩阵 ℓ_1 范数与核范数.

引入乘子 Y 作用在约束 $X+S=M$ 上，我们可以得到此问题的增广拉格朗日函数

$$L_\rho(X,S,Y) = \|X\|_* + \mu\|S\|_1 + \langle Y, X+S-M\rangle + \frac{\rho}{2}\|X+S-M\|_F^2. \tag{21.15}$$

- 对于 X 子问题，

$$\begin{aligned} X^{k+1} &= \operatorname*{argmin}_X \ L_\rho(X, S^k, Y^k) \\ &= \operatorname*{argmin}_X \left\{ \|X\|_* + \frac{\rho}{2}\left\| X + S^k - M + \frac{Y^k}{\rho}\right\|_F^2 \right\}, \\ &= \operatorname*{argmin}_X \left\{ \frac{1}{\rho}\|X\|_* + \frac{1}{2}\left\| X + S^k - M + \frac{Y^k}{\rho}\right\|_F^2 \right\}, \\ &= U\mathrm{Diag}\Big(\mathrm{prox}_{(1/\rho)\|\cdot\|_1}(\sigma(A))\Big)V^{\mathrm{T}}, \end{aligned}$$

  其中 $A = M - S^k - \frac{Y^k}{\rho}$，$\sigma(A)$ 为 A 的所有非零奇异值构成的向量并且 $U\mathrm{Diag}(\sigma(A))V^{\mathrm{T}}$ 为 A 的约化奇异值分解.

- 对于 S 子问题，

$$\begin{aligned} S^{k+1} &= \operatorname*{argmin}_S \ L_\rho(X^{k+1}, S, Y^k) \\ &= \operatorname*{argmin}_S \left\{ \mu\|S\|_1 + \frac{\rho}{2}\left\| X^{k+1} + S - M + \frac{Y^k}{\rho}\right\|_F^2 \right\} \\ &= \mathrm{prox}_{(\mu/\rho)\|\cdot\|_1}\left(M - X^{k+1} - \frac{Y^k}{\rho}\right). \end{aligned}$$

- 那么交替方向乘子法的迭代格式为

$$\begin{aligned} X^{k+1} &= U\mathrm{Diag}\Big(\mathrm{prox}_{(1/\rho)\|\cdot\|_1}(\sigma(A))\Big)V^{\mathrm{T}}, \\ S^{k+1} &= \mathrm{prox}_{(\mu/\rho)\|\cdot\|_1}\left(M - L^{k+1} - \frac{Y^k}{\rho}\right), \\ Y^{k+1} &= Y^k + \tau\rho(X^{k+1} + S^{k+1} - M). \end{aligned}$$

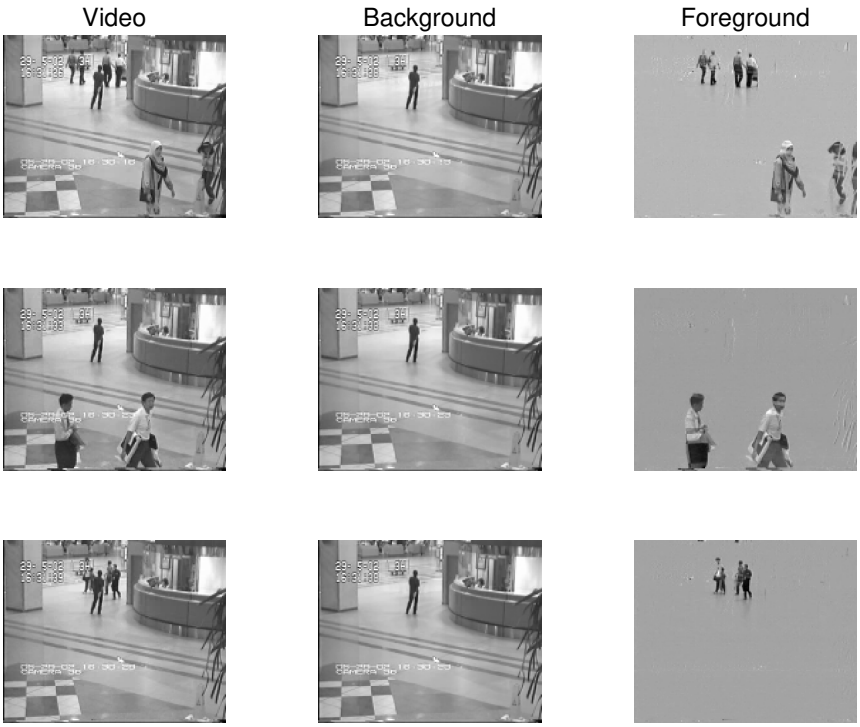


图 21.1: 通过 RPCA，将视频图片分为背景（低秩）和前景（稀疏）两个部分。

2.4 ADMM 收敛结果

我们先引入一些必要的假设.

- $f_1(x), f_2(x)$ 均为闭凸函数，且每个 ADMM 迭代子问题存在唯一解；
- 原始问题的解集非空，且 Slater 条件满足.

注：假设给出的条件是很基本的.

- f_1 和 f_2 的凸性保证了要求解的问题是凸问题，每个子问题存在唯一解是为了保证迭代的良定义
- 在 Slater 条件满足的情况下，原始问题的 KKT 对和最优解是对应的，因此可以很方便地使用 KKT 条件来讨论收敛性.

Theorem 21.1 在假设的条件下,进一步假定 A_1, A_2 列满秩.如果 $\tau \in \left(0, \frac{1+\sqrt{5}}{2}\right)$,则序列 $\{(x_1^k, x_2^k, y^k)\}$ 收敛到原始问题的一个 KKT 对.

2.5 多块问题的 ADMM

考虑有多块变量的情形

$$\begin{aligned} \min_{x_1,x_2,\cdots,x_N} \quad & f_1(x_1) + f_2(x_2) + \cdots + f_N(x_N), \\ \text{s.t.} \quad & A_1x_1 + A_2x_2 + \cdots + A_Nx_N = b. \end{aligned} \tag{21.16}$$

这里 $f_i(x_i)$ 是闭凸函数，$x_i \in \mathbb{R}^{n_i}, A_i \in \mathbb{R}^{m\times n_i}$. 同样写出增广拉格朗日函数 $L_\rho(x_1, x_2, \cdots, x_N, y)$，相应的多块 ADMM 迭代格式为

$$\begin{aligned} x_1^{k+1} &= \operatorname*{argmin}_x L_\rho(x, x_2^k, \cdots, x_N^k, y^k), \\ x_2^{k+1} &= \operatorname*{argmin}_x L_\rho(x_1^{k+1}, x, \cdots, x_N^k, y^k), \\ &\cdots\cdots\cdots\cdots \\ x_N^{k+1} &= \operatorname*{argmin}_x L_\rho(x_1^{k+1}, x_2^{k+1}, \cdots, x, y^k), \\ y^{k+1} &= y^k + \tau\rho(A_1x_1^{k+1} + A_2x_2^{k+1} + \cdots + A_Nx_N^{k+1} - b), \end{aligned}$$

其中 $\tau \in (0, (\sqrt{5}+1)/2)$ 为步长参数.

需要说明的是，多块 ADMM 有时候未必收敛。有如下例子。

Example 21.5 (多块 ADMM 收敛性反例) 考虑最优化问题

$$\begin{aligned} \min \quad & 0, \\ \text{s.t.} \quad & A_1x_1 + A_2x_2 + A_3x_3 = 0, \end{aligned} \tag{21.17}$$

其中 $A_i \in \mathbb{R}^3$, $i = 1, 2, 3$ 为三维空间中的非零向量，$x_i \in \mathbb{R}$, $i = 1, 2, 3$ 是自变量. 该问题实际上就是求解三维空间中的线性方程组，若 A_1, A_2, A_3 之间线性无关，则问题(21.17) 只有零解. 此时容易计算出最优解对应的乘子为 $y = (0, 0, 0)^{\mathrm{T}}$.

(21.17)的增广拉格朗日函数为

$$L_\rho(x, y) = 0 + y^{\mathrm{T}}(A_1x_1 + A_2x_2 + A_3x_3) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 + A_3x_3\|^2.$$

- 当固定 x_2, x_3, y 时，对 x_1 求最小可推出

  $$A_1^{\mathrm{T}}y + \rho A_1^{\mathrm{T}}(A_1x_1 + A_2x_2 + A_3x_3) = 0,$$

  整理可得

  $$x_1 = -\frac{1}{\|A_1\|^2}\left(A_1^{\mathrm{T}}\left(\frac{y}{\rho} + A_2x_2 + A_3x_3\right)\right).$$

  可类似地计算 x_2, x_3 的表达式

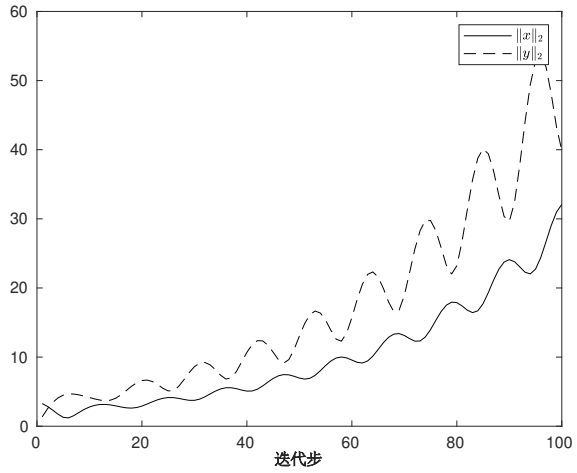
- 因此多块交替方向乘子法的迭代格式可以写为

$$\begin{aligned} x_1^{k+1} &= -\frac{1}{\|A_1\|^2} A_1^{\mathrm{T}} \left(\frac{y^k}{\rho} + A_2 x_2^k + A_3 x_3^k \right), \\ x_2^{k+1} &= -\frac{1}{\|A_2\|^2} A_2^{\mathrm{T}} \left(\frac{y^k}{\rho} + A_1 x_1^{k+1} + A_3 x_3^k \right), \\ x_3^{k+1} &= -\frac{1}{\|A_3\|^2} A_3^{\mathrm{T}} \left(\frac{y^k}{\rho} + A_1 x_1^{k+1} + A_2 x_2^{k+1} \right), \\ y^{k+1} &= y^k + \rho(A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1}). \end{aligned} \tag{21.18}$$

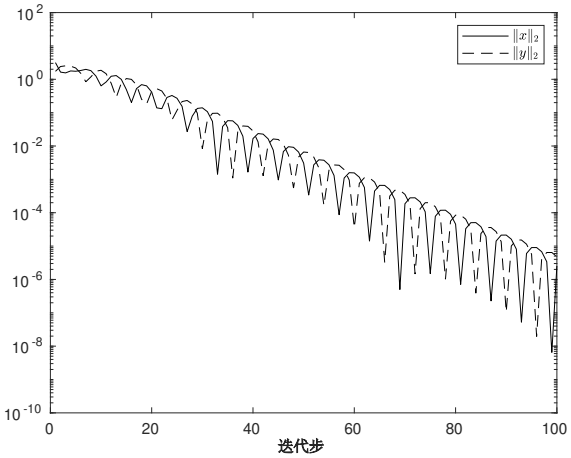
- 自变量初值初值选为 $(1,1,1)$，乘子选为 $(0,0,0)$. 选取 A 为

$$\widetilde{A} = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad 或 \quad \widehat{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{bmatrix}.$$

- 下图记录了在不同 A 下 x 和 y 的 ℓ_2 范数随迭代的变化过程.



(a) 系数矩阵为 $\widetilde{A}$，ADMM 不收敛。



(b) 系数矩阵为 $\widehat{A}$

图 21.2: 选取不同 A 时的数值结果

作业 21.1 *1.* 给出 *ADMM* 求解线性规划标准形式，以及其对偶问题的迭代形式，要求写出子问题的求解公式。

2. 考虑如下问题

$$\min \sum_{i=1}^{n} f_i(x_i), \quad \text{s.t.} \quad x_1 = x_2 = \ldots = x_n.$$

其中，f_i 均为闭凸函数，且其近似点映射有显式解。写出多块 *ADMM* 求解该问题的迭代形式，要求每个 x_i 的子问题均有显式解。

3. 考虑如下 *Sparse Inverse Covariance Selection* 问题：给定对称正定矩阵 $C \in \mathbb{R}^{n\times n}$，$\lambda > 0$ 为给定的实数，求解

$$\min_X \operatorname{Tr}(CX) - \log\det X + \lambda\|X\|_1,$$

这里，$\|X\|_1 = \sum_{i=1}^n \sum_{j=1}^n |X_{ij}|$。使用 *ADMM* 求解该问题，并给出子问题的显式解。(提示：该问题为凸问题，默认 X 的定义域为对称正定矩阵集合，即无需添加正定矩阵约束，直接考虑无约束问题，$\log\det X$ 的梯度为 X^{-1}) .

MATH 3011 运筹学 **秋季 2024**

Lecture 22: 习题集

Lecturer: 陈士祥 *Scribes:* 陈士祥

问题 1 (判断题) *1.* 线性规划标准形式中，设有划分 $A = (B, N)$, 这里 B 是可逆矩阵。那么 $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ 是可行基解。

2. 线性规划的标准形式中，可行基解 x 是退化的等价于 x 有 $n-m$ 的坐标为 *0.*

3. 考虑标准形式的线性规划 *(LP)* 和其对偶问题 *(DP)*，若 *(LP)* 无界，那么对偶问题可能有可行解。

4. 单纯形法是多项式时间算法。

5. 对于带约束的非线性规划问题，已知目标函数和约束函数均充分光滑。若 $\bar{x}$ 处满足正则性条件 *LICQ*，那么切锥和线性可行锥相等。

6. 给定 $f(x)$，$f(x)$ 可微，则无约束优化问题 $\min f(x)$ 的最优充分条件是其梯度为 *0.*

7. 多面体集 $\{x \mid Ax \leq b\}$ 是凸集。

8. $f(x) = -\prod_{i=1}^{n} x_i$ 是凸函数。

9. 求解凸的光滑问题，牛顿法（步长为 *1*）总是可以收敛。

10. DFP 和 *BFGS* 的区别仅在于使用的割线方程不同。

问题 2 (凸函数) *1.* 假设 $f : \mathbb{R} \to \mathbb{R}$ 是凸函数，给定 $a < b \in \mathrm{dom} f$. 证明：

(a)
$$f(x) \leq \frac{b-x}{b-a} f(a) + \frac{x-a}{b-a} f(b), \quad \forall x \in [a, b].$$

(b)
$$\frac{f(x)-f(a)}{x-a} \leq \frac{f(b)-f(a)}{b-a} \leq \frac{f(b)-f(x)}{b-x}.$$

2. 证明下面的函数是凸函数

(a) $f(x) = \frac{1}{2}\|Ax-b\|^2$. 则 $\nabla f(x) = A^T(Ax-b)$, $\nabla^2 f(x) = A^T A$ 是半正定的。故其为凸函数。

(b) $f(x) = \log \sum_{k=1}^{n} \exp x_k$ 是凸函数.

(c) $f(x, y) = x^2/y$ 是定义域 $\{(x, y) \mid y > 0\}$ 上的凸函数。即二次函数的分式变换是凸函数。

(d) (*) $f(x,u,v) = -\log(uv - x^Tx)$, 其定义域为 $\{(x,u,v) \mid uv > x^Tx, u > 0, v > 0\}$. 提示：复合函数 $f(x) = h(g(x))$ 是凸函数，如果 g 是凹函数，h 是凸函数且单调递减。

(e) (*) $f(x,t) = -\log(t^p - \|x\|_p^p), p > 1$, $\mathrm{dom} f = \{(x,t) \mid t > \|x\|_p\}$. 提示：可以使用如下两个结论：*(1)* x^Tx/u 是凸函数；*(2)*$\|x\|_p^p/u^{p-1}$ 是关于 (x,u), $u > 0$ 的凸函数.

(f) 对于 $x \in \mathbb{R}^n, n > 1$, 令 $x_{[k]}$ 表示 x 的第 k 大的分量。例如，$x_{[1]} = \max_{i=1,\ldots,n} x_i$, $x_{[n]} = \min_{i=1,\ldots,n} x_i$. 确定下面的函数是否为凸函数。

- $x_{[1]} - x_{[n]}$。
- $\mathrm{median}(x) = x_{[n+1]/2}$, 假设 n 为奇数。
- $(x_{[1]} + x_{[n]})/2$.
- $x_{[1]} + x_{[2]} + ... + x_{[k]}, k \le n$.

问题 3 (计算次梯度) 计算下面函数的一个次梯度：

1. $f(x) = \max_{i=1,\ldots,m}(a_i^Tx + b_i)$.
2. $f(x) = x_{[1]} + \ldots + x_{[k]}$, 这里 $x_{[i]}$ 表示 $x \in \mathbb{R}^n, n \ge k$ 的第 i 大的分量。
3. $f(x) = \|Ax - b\|_2 + \|x\|_2$.
4. $f(x) = \inf_y \|x - y\|^2$, s.t. $Ay \le b$.

问题 4 (对偶理论,KKT 条件, 最优条件) 1. 计算 $\|x\|_2$ 的近似点映射。

2. 假设 $m \ge n$, 令 $X = U\Sigma V^{\mathrm{T}}$ 为矩阵 X 的奇异值分解。记对角矩阵 $\Sigma = \mathrm{Diag}(d) \in \mathbb{R}^{n\times n}$，对角部分 $d \in \mathbb{R}^n$ 中的分量由大到小排列 $d_1 \ge d_2 \ge \ldots \ge d_n$。若 $\mathrm{rank}(X) = r, r \le m$, 则

$$X = U_0\Sigma_0V_0^T, \quad U_0 \in \mathbb{R}^{m\times r}, \Sigma_0 \in \mathbb{R}^{r\times r}, V_0 \in \mathbb{R}^{n\times r}$$

这里 Σ_0 为保留 Σ 中非零的奇异值部分，即 $\Sigma_0 = \mathrm{Diag}(d_1, d_2, \ldots, d_r)$。我们有次微分集的如下表示：

$$\partial\|X\|_* = \{U_0V_0^T + W : U_0^TW = 0, WV_0 = 0, \|W\|_2 \le 1, W \in \mathbb{R}^{m\times n}\}.$$

验证 $\mathrm{prox}_{\tau\|\cdot\|_*}(Y) = US_\tau(\Sigma)V^\top$. 其中 $S_\tau(\Sigma) = \mathrm{Diag}\left(\max\{|d| - \tau, 0\}\right)$.

3. 给定正定对称矩阵 $A \in \mathbb{R}^{n\times n}$, 考虑问题

$$\max_x x^TAx, \quad \text{s.t.} \quad \|x\|_2 = 1.$$

证明全局最优解是 A 的最大特征根方向。

4. $x, y \in \mathbb{R}^n_{++}$ 的相对熵记为：

$$\sum_{i=1}^{n} x_i \log(x_i/y_i).$$

已知，该函数关于 (x, y) 是凸函数。考虑如下关于 $x \in \mathbb{R}^n$ 的优化问题：

$$\begin{aligned} \min \sum_{i=1}^{n} x_i \log(x_i/y_i) \\ \text{s.t.} \quad Ax = b \\ \sum_{i=1}^{n} x_i = 1, \end{aligned}$$

其中，$y \in \mathbb{R}^n_{++}$, $A \in \mathbb{R}^{m\times n}$，以及 $b \in \mathbb{R}^m$ 是已知的。

推导其对偶问题为：

$$\max_z b^T z - \log \sum_{i=1}^{n} y_i \exp(a_i^T z),$$

其中 a_i 是矩阵 A 的第 i 列。

问题 5（优化问题的等价转化） 两个优化问题等价，如果他们的最优解可以互相转化，即解决其中一个便解决了另一个问题。或者，他们的可行点可以相互转化。

1. 考虑如下问题：

$$\min_{x\in\mathbb{R}^n} \max_{i=1,\ldots,m} (a_i^T x + b_i).$$

将其写成等价的线性规划形式。

2. 考虑如下问题：

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_\infty,$$

其中，$A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. 将其写成等价的线性规划形式。

3. 考虑如下问题：

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_1,$$

其中，$A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. 将其写成等价的线性规划形式。

4. (*) 考虑如下鲁棒线性规划，

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c^T x \\ \text{s.t.} \quad & \sup_{a\in P_i} a^T x \le b_i, i = 1, ..., m, \end{aligned}$$

这里，$P_i = \{a \mid C_i a \le d_i\}$, 其中 $C_i \in \mathbb{R}^{m_i \times n}, b_i \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ 都是给定的数据. 证明该问题等价与下面的线性规划：

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & d_i^T z_i \le b_i, i = 1, ..., m, \\ & C_i^T z_i = x, i = 1, ..., m, \\ & z_i \ge 0, i = 1, ..., m, \end{aligned}$$

变量为 $x \in \mathbb{R}^n$, $z_i \in \mathbb{R}_i^m, i = 1, ..., m$. *(***提示：** 考虑约束问题关于 a 的对偶问题。*)*

问题 6 (算法迭代) *1.* 使用牛顿法计算 $f(x) = x^2 - 2 = 0$ 的零点，写出迭代公式。

2. 使用 *ADMM* 求解

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_\infty$$