

运筹学 & 最优化 重制版讲义

作者：原生生物

时间：February 20, 2025

版本：3.0

相关课程：1-5 章为运筹学内容，3-7 章为最优化算法内容。

目录

第 1 章 最优化绪论	1
1.1 最优化概述	1
1.2 最优性条件	3
1.3 下降算法	7
第 2 章 线性规划	10
2.1 基本模型	10
2.2 单纯形法	16
2.3 对偶理论	22
第 3 章 网络流与动态规划	26
3.1 网络最优化	26
3.2 流的相关问题	34
3.3 动态规划	39
第 4 章 无约束最优化	43
4.1 一维搜索	43
4.2 梯度方法	47
4.3 牛顿方法	51
4.4 信赖域法	55
第 5 章 有约束最优化	58
5.1 二次规划	58
5.2 逐步二次规划	64
5.3 罚函数	68
5.4 内点法	71
第 6 章 凸优化	75
6.1 凸函数	75
6.2 对偶性质	81
6.3 数值解法	86
第 7 章 大数据中的优化	93
7.1 稀疏优化	93
7.2 随机梯度法	97
7.3 批次梯度法	104
7.4 随机牛顿法	109
7.5 更多常用算法	114

第 8 章 总结	121
附录 A 数学基础	122
A.1 分析	122
A.2 代数	123
附录 B 数学进阶	126
附录 C 参考资料	129

第 1 章 最优化绪论

内容提要

- 运筹学简介
- 最优化的数学表达
- 无约束问题最优条件
- 约束问题最优条件
- 算法收敛性刻画
- 算法评价方式
- 迭代下降基本算法

1.1 最优化概述

1.1.1 运筹学简介

运筹学 (Operations Research, OR) 是从二十世纪三四十年代 (即二战期间) 发展起来的一门应用性很强的学科。它的研究对象是人类对各种资源的运用及筹划活动, 如战争、后勤、生产规划、经营管理、资本运作、工程设计等。研究内容就是资源筹划活动中各种问题的模型化及其数学方法, 学科分支包括:

- 线性规划 [Leontief, 1932]
- 非线性规划 [Kuhn Tucker, 1951]
- 整数规划 [GoMory, 1958]
- 目标规划 [Pareto, 1896]
- 动态规划 [Bellman, 1957]
- 图论与网络分析 [Euler, 1736]
- 网络计划 [Gantt, 1917]
- 存储论 [Harris, 1915]
- 排队论 [Erlang, 1909]
- 博弈论 [Neumann, 1928]
- 决策论 [Bernoulli, 1738]
- 启发式算法 [1940s]
-

运筹学的主要分析过程包括**定性分析**与**定量分析**。定性分析往往是解决问题的第一步, 具体来说即主要决策的内容、不同方案有效性的度量与比较各方案时度量间可能的权衡; 与之相对, 管理决策过程时往往需要的是定量分析, 标准步骤是:

1. 表达问题 [Definition of the problem]

列出问题的要素, 其中包含可控的变量 [决策变量]、不可控的变量 [参数]、各变量的约束条件与确定最佳方案采用的目标度量。

2. 建立模型 [Construction of the model]

将上方四个要素的关系用一定的数学模型进行刻画。

3. 分析求解 [Solution of the model]

分析模型并用各种数学方法和手段来求解模型，进而确定解对模型的技术条件的灵敏度。

4. 检验与改进 [Validation of the model]

将模型的解应用到实际问题中，检验解的合理性和有效性，可能产生的问题和修改模型。

5. 解的实施 [Implementation of the solution]

将模型的解应用于实际问题，并最终解决实际问题。

其中，**建立模型**的过程非常重要。它将实际问题简化、抽象为了反映实际问题的数学模型，而通过求解数学模型得到的结果可以返回到实际问题中检验。各种优化模型与求解它们的数学方法构成了运筹学的大部分内容，微分方程模型、统计模型、最优化模型等都是可供选择的代表性模型，但有限的模型并不能完全适用于所有实际问题。



一般的运筹学参考书不会着重叙述建立模型的过程，这并不是说建模不重要；相反，建模在任何时候任何场合都是极其重要的。

1.1.2 最优化问题

在很多实际应用问题中，从数学上看都是非适定 [ill-posed] 的，即解不唯一。对于这样的实际问题，人们往往通过制定相应的目标准则，然后从众多的解中选出在**一定条件下最好的解**。而这，正是最优化理论与方法所研究的内容。本节对最优化的基本概念作一些简要的介绍，并给出最优化建模方法的相关知识。

最优化讨论的是为找出实际问题的最优解决策略而采取的模型化及方法，其过程是：先把待解决的问题用最优化形式描述为在给定的约束条件下找出使某个**目标函数达到最大/小值**的解，再采用数学上严密的算法来求解。



最优化方法作为数学工具已在现实中被广泛应用，大多数代表性的算法也都有程序库与软件包。但是，有效利用这些成果的前提是待解决问题已化为最优化问题的形式，而转化的过程并不简单。

最优化方法在生产规划、经济管理、工业工程、交通运输、国防等重要领域中都有着广泛的应用，并已受到政府部门、科研机构和产业界的高度重视。例如，运筹学和计算几何的交叉分支“选址问题”，就关联到许多现实的选址模型。

此外，很多机器学习任务本质上都是最优化问题，如监督学习中的回归、分类，无监督学习中的聚类等（“Optimization lies at the heart of machine learning”）。

在数学上，最优化一般的描述是：在给定的**约束条件** [constraint] 下，找出**决策变量** [decision variable] 的一个值，使得被称为**目标函数** [objective function] 的表达愿望尺度的函数值达到**最值**。由于决策变量一般有多多个，可以用向量 $x = (x_1, \dots, x_n)^T$ 来表示，于是问题写为：

$$\min f(x) \quad \text{s.t. } x \in S \subset \mathbb{R}^n \quad (1.1)$$

目标函数 f 是定义在包含 S 的适当集合中的实值函数。



若所需的是最大值，利用 $\max g(x) = -\min(-g(x))$ 可以转化为上方的形式。

定义 1.1 (可行域)

在式 (1.1) 中， S 是问题变量 x 的可取值之集合，称为问题的可行域。

若 $S = \mathbb{R}^n$ ，则问题为 $\min_{x \in \mathbb{R}^n} f(x)$ ，称为无约束最优化问题，否则称为带约束最优化问题。



从属性上，最优化问题可以分为两大类，一类是具有离散变量（也即可行域是有限个点）的问题，又

称为组合优化问题；另一类则是我们讨论的重点，具有连续变量的优化问题。这样的问题通常可写为

$$\begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & g_i(x) = 0 \quad i \in \mathcal{E} = \{1, \dots, m\} \\ & c_i(x) \geq 0 \quad i \in \mathcal{I} \end{aligned} \quad (1.2)$$

其中 $c_i(x)$ 为约束函数， \mathcal{E}, \mathcal{I} 分别为等式约束与不等式约束的指标集合。

定义 1.2 (线性规划)

当目标函数与约束函数都是线性函数时，最优化问题称为线性规划 [Linear Programming]，否则称为非线性规划 [Nonlinear Programming]。



根据决策变量、目标函数和要求的不同，最优化还被分为整数规划、动态规划、网络规划、非光滑规划、随机规划、多目标规划等。

对一般的非线性规划问题，式 (1.2) 的 f, c_i 都是 n 变量的确定的实值函数，且 c_i 一般个数有限。 f 与 c_i 中至少有一个是非线性的。

定义 1.3 (可行解、最优解)

式 (1.1) 中满足约束条件 $x \in S$ 的 x 称为问题的可行解 [feasible solution]。

若可行解 x^* 进一步满足 $f(x^*) \leq f(x), \forall x \in S$ ，则 x^* 称为问题的全局最优解 [global optimal solution]。

此外，若可行解 x^* 满足 $f(x^*) \leq f(x), \forall x \in S \cap U(x^*)$ ，其中 $U(x^*)$ 代表包含 x^* 的某邻域，则 x^* 称为问题的局部最优解 [local optimal solution]。



不少问题的目标函数或约束条件可能很复杂，导致找出全局最优解非常困难。这时，目标往往是求出局部的最优解。而具体的相关研究分为两个方面：一是研究最优解的性质，二是设计有效算法来获得问题的解。

1.2 最优性条件

1.2.1 无约束问题的最优条件

问题的最优解所满足的必要或者充分条件称为最优性条件，它为最优化问题求解算法的设计、分析提供必不可少的理论基础。对于无约束最优化问题，最优条件的形式较为简洁的：

定理 1.4 (无约束问题的极值条件)

一阶必要条件： $f(x)$ 在点 \bar{x} 处可微，若其为极小值，则 $\nabla f(\bar{x}) = 0$ 。

二阶必要条件： $f(x)$ 在点 \bar{x} 处二阶可微，若其为极小值，则 $\nabla f(\bar{x}) = 0$ ，且 $\nabla^2 f(\bar{x})$ 半正定。

二阶充分条件： $f(x)$ 在点 \bar{x} 处二阶可微，若 $\nabla f(\bar{x}) = 0$ ，且 $\nabla^2 f(\bar{x})$ 正定，则其为极小值。



证明 一阶必要条件：若 f 可微，其可在 \bar{x} 邻域展开为

$$f(\bar{x} + \alpha t) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T t + O(\alpha^2)$$

于是若极小值点 $\nabla f(\bar{x}) \neq 0$ ，取 $t = -\nabla f(\bar{x})$ ，在 α 充分小时即有 $f(\bar{x}) > f(\bar{x} + \alpha t)$ ，矛盾。

二阶必要条件：若 f 二阶可微，由一阶必要条件有

$$f(\bar{x} + \alpha t) = f(\bar{x}) + \frac{1}{2} \alpha^2 t^T \nabla^2 f(x) t + O(\alpha^3)$$

由若 $\nabla^2 f(x)$ 非半正定，由定义存在 t 使得 $t^T \nabla^2 f(x) t < 0$ ，从而在 α 充分小时即有 $f(\bar{x}) > f(\bar{x} + \alpha t)$ ，矛盾。

二阶充分条件：由正定定义，对任何 t ， $\frac{1}{2} \alpha^2 t^T \nabla^2 f(x) t > 0$ ，从而在 α 充分小时一定有 $f(\bar{x}) < f(\bar{x} + \alpha t)$ 。对每个单位向量 t 考虑最优的 $\alpha(t) > 0$ ，由于所有可能 t 构成的集合（高维球面）是紧集，且可验证 $\alpha(t)$ 连续性，存在 $\alpha_0 = \min_t \alpha(t) > 0$ ，于是 $\{x \mid \|x - \bar{x}\| < \alpha_0\}$ 内 \bar{x} 是最小点，从而是极小值。

不过，这里只有必要条件与充分条件，并不能完整进行刻画。好在，当函数 f 有一定性质的时候，必要条件可以提升为充要条件，这个重要性质就是凸性。

定义 1.5 (凸函数)

\mathbb{R}^n 上的函数 f 为凸函数，当其满足对 $\forall x, y \in \mathbb{R}^n, \lambda \in [0, 1]$ 有

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$



对凸函数还有两个重要的等价定义，这里进行列举，后文凸优化中详细证明：

命题 1.6 (凸函数等价定义)

\mathbb{R}^n 上的可微函数 f 为凸函数，当且仅当其满足对 $\forall x, y \in \mathbb{R}^n$ 有

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

\mathbb{R}^n 上的二阶可微函数 f 为凸函数，当且仅当其满足对 $\forall x \in \mathbb{R}^n$ 有 $\nabla^2 f(x)$ 半正定。



事实上凸函数不必定义在 \mathbb{R}^n 上，只需要定义在凸集上就可以谈论，此处由于问题是无约束的，考虑全空间即可。

有了等价定义后，就可以有无约束优化的最优条件：

定理 1.7 (凸函数下的充要条件)

$f(x)$ 是 \mathbb{R}^n 上的可微凸函数，则 \bar{x} 是全局最优解的充要条件是 $\nabla f(\bar{x}) = 0$ 。



证明 必要性根据之前定理得到。对于充分性，由一阶等价条件直接得到 $f(y) \geq f(\bar{x})$ 。

1.2.2 约束问题的最优条件

在有约束的情况下，想判断一个点是否是局部最优解就要困难很多。约束导致可行域的复杂，也导致了最优判定必须考虑边界的情况。对此，有如下的直观定义与性质：

定义 1.8 (可行方向、下降方向)

\mathbb{R}^n 中，对 $x \in S$ 与非零向量 d ，若存在 $\delta > 0$ 使得 $x + \lambda d \in S, \forall \lambda \in (0, \delta)$ ，则称 d 是 S 在 x 处的可行方向，并记所有 S 在 x 处的可行方向集合为 $F(x, S)$ 。

对 \mathbb{R}^n 上实函数 f 与向量 x ，非零向量 d ，若存在 $\delta > 0$ 使得 $f(x + \lambda d) < f(x), \forall \lambda \in (0, \delta)$ ，则称 d 为 f 在 x 处的下降方向，并记所有 S 在 x 处的下降方向集合为 $D_0(x, S)$ 。

若 f 可微，记所有满足 $\nabla f(x)^T d < 0$ 的向量 d 构成 $D(x, f)$ 。



可行方向也即“这个方向还没有碰到边界”，下降方向也即“ f 沿此方向局部下降”。

定理 1.9 (几何必要条件)

对约束最优化问题 $\min_{x \in S} f(x)$, x^* 是问题的局部最优解, 则 $F(x^*, S) \cap D_0(x^*, S) = \emptyset$ 。

当 f 可微时, $D(x^*, f) \subset D_0(x^*, f)$, 因此 x^* 是问题的局部最优解可推出 $F(x^*, S) \cap D(x^*, S) = \emptyset$ 。♡

证明 根据定义, 若 $F(x^*, S) \cap D_0(x^*, S) \neq \emptyset$, 假设其中有方向 d , 取可行方向的 δ 与下降方向的 δ 的最小值为 δ_0 , 则 $f(x^* + \lambda d)$, $\lambda \in (0, \delta)$ 中的每个点都是比 $f(x^*)$ 更小的可行点, 矛盾。利用一阶泰勒展开与前文无约束最优化时相同可证 $D(x^*, f) \subset D_0(x^*, f)$, 于是即得可微时结论。

下面, 我们针对具体的可行域进行更细节的讨论。式 (1.2) 一般可以更进一步写为如下形式:

$$\begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & g_i(x) \geq 0 \quad i = 1, \dots, m \\ & h_j(x) = 0 \quad j = 1, \dots, l \end{aligned} \quad (1.3)$$

比起式 (1.2), 式 (1.3) 中的指标集合成为了有限集合, 之后讨论的也基本是这种情况。此时, 上方的必要条件可以进一步细化:

命题 1.10 (几何必要条件-函数约束)

记 $\mathcal{I}(x) = \{i \mid g_i(x) = 0\}$, 也即对不等式约束取等的指标集合。在式 (1.3) 中定义

$$\begin{aligned} D_f(x) &= \{d \mid \nabla f(x)^T d < 0\} \\ F_g(x) &= \{d \mid \nabla g_i(x)^T d > 0, \forall i \in \mathcal{I}(x)\} \\ F_h(x) &= \{d \mid \nabla h_j(x)^T d = 0, \forall j\} \end{aligned}$$

若 x^* 为局部最优解, $f, g_i, i \in \mathcal{I}(x^*)$ 与所有 h_j 在 x^* 可微, $g_i, i \notin \mathcal{I}(x^*)$ 在 x^* 连续, 且所有 $\nabla h_j(x^*), j = 1, \dots, l$ 线性无关, 则有

$$D_f(x^*) \cap F_g(x^*) \cap F_h(x^*) = \emptyset$$

这个结论的证明需要一些精细而复杂的微分几何操作, 此处略过。从这个定理出发, 即可以得到重要的 Fritz-John 条件:

定理 1.11 (Fritz-John 必要条件)

若 x^* 为式 (1.3) 中的可行点, 记 $\mathcal{I}(x) = \{i \mid g_i(x) = 0\}$ 。

连续性要求: $f, g_i, i \in \mathcal{I}(x^*)$ 与所有 h_j 在 x^* 可微, $g_i, i \notin \mathcal{I}(x^*)$ 在 x^* 连续。

在满足上述要求时, 若 x^* 是局部最优解, 则存在不全为 0 的 $\lambda_0, \lambda_i, i \in \mathcal{I}(x^*), \mu_j, j = 1, \dots, l$ 使得

$$\lambda_0 \nabla f(x^*) - \sum_{i \in \mathcal{I}(x^*)} \lambda_i \nabla g_i(x^*) - \sum_{j=1}^l \mu_j \nabla h_j(x^*) = 0$$

且 $\lambda_0 \geq 0, \lambda_i \geq 0, i \in \mathcal{I}(x^*)$ 。♡

证明 若 $\nabla h_j(x^*), j = 1, \dots, l$ 线性相关, 可直接取合适的 $\mu_j, \lambda_0, \lambda_i$ 均为 0 即得结论。

否则, 根据几何必要条件, 不等式组

$$\begin{cases} \nabla f(x^*)^T d < 0 \\ \nabla g_i(x^*)^T d > 0 & i \in \mathcal{I}(x^*) \\ \nabla h_j(x^*)^T d = 0 & j = 1, \dots, l \end{cases}$$

无解。记 $\mathcal{I}(x^*) = i_1, \dots, i_k$, 并记

$$A = (\nabla f(x^*), -\nabla g_{i_1}(x^*), \dots, -\nabla g_{i_k}(x^*)), B = (-\nabla h_1(x^*), \dots, -\nabla h_l(x^*))$$

则条件即 $A^T d < 0, B^T d = 0$ 无解。下证 $A\lambda + B\mu = 0, \lambda \geq 0$ 有解, 这样记 $\lambda = (\lambda_0, \dots, \lambda_k), \mu = (\mu_1, \dots, \mu_m)$ 即得结论。

[下方证明需含凸集分离定理 (详见参考资料) 在内的分析与线代知识, 可跳过。]

记

$$S_1 = \left\{ \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \mid \exists d, \lambda = A^T d, \mu = B^T d \right\}, S_2 = \left\{ \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \mid \lambda < 0, \mu = 0 \right\}$$

则条件为 $S_1 \cap S_2 = \emptyset$ 。可验证二者均为凸集, 因此根据凸集分离定理, 存在超平面 $\begin{pmatrix} p_1 \\ p_2 \end{pmatrix}^T x + \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = 0$ 分离两凸集, 也即

$$\forall s_1 \in S_1, s_2 \in S_2, \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}^T s_1 + \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \geq 0 \geq \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}^T s_2 + \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \Rightarrow \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}^T s_1 \geq \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}^T s_2$$

由连续性, 对任何 S_1 闭包与 S_2 闭包中的 s_1, s_2 , 大于等于号仍成立。取 $s_1 = 0$ (即 $d = 0$), 利用 s_2 范围可知必须 $p_1 \geq 0$, 而取 S_2 闭包中的 $s_2 = 0$, S_1 中 $d = -(Ap_1 + Bp_2)$ 可知 $-\|Ap_1 + Bp_2\|^2 \geq 0$, 所以只能 $Ap_1 + Bp_2 = 0$, 这里 p_1, p_2 即为所需的解 λ, μ 。

现实中更常用的是其特殊情况:

定理 1.12 (Kuhn-Tucker 必要条件)

若 x^* 为式 (1.3) 中的可行点, 记 $\mathcal{I}(x) = \{i \mid g_i(x) = 0\}$ 。

连续性要求: $f, g_i, i \in \mathcal{I}(x^*)$ 与所有 h_j 在 x^* 可微, $g_i, i \notin \mathcal{I}(x)$ 在 x^* 连续, 且向量集 $\{\nabla g_i(x^*), i \in \mathcal{I}(x^*), \nabla h_j(x^*), j = 1, \dots, l\}$ 线性无关。

在满足上述要求时, 若 x^* 是局部最优解, 则存在 $\lambda_i \geq 0, i \in \mathcal{I}(x)$ 与 $\mu_j, j = 1, \dots, l$ 使得

$$\nabla f(x^*) - \sum_{i \in \mathcal{I}(x^*)} \lambda_i \nabla g_i(x^*) - \sum_{j=1}^l \mu_j \nabla h_j(x^*) = 0$$



证明 根据 Fritz-John 条件, 由于 $\lambda_0, \lambda_i, \mu_j$ 不全为 0, 假设 $\lambda_0 = 0$, 则存在不全为 0 的 λ_i, μ_j 使得 $\sum_{i \in \mathcal{I}(x^*)} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^l \mu_j \nabla h_j(x^*) = 0$, 与线性无关矛盾, 因此 $\lambda_0 > 0$, 同除以 λ_0 即得到 Kuhn-Tucker 必要条件。

定义 Lagrange 函数 $L(x, \lambda, \mu) = f(x) - \sum_{i=1}^m \lambda_i g_i(x) - \sum_{j=1}^l \mu_j h_j(x)$, 则 K-T 条件可以表达为对

x 存在 λ, μ 使得 (将条件中 $i \notin \mathcal{I}(x^*)$ 的 λ_i 记为 0):

$$\begin{cases} \nabla_x L(x, \lambda, \mu) = 0 \\ g_i(x) \geq 0 & i = 1, \dots, m \\ h_j(x) = 0 & j = 1, \dots, l \\ \lambda_i \geq 0 & i = 1, \dots, m \\ \lambda_i g_i(x) = 0 & i = 1, \dots, m \end{cases} \quad (\text{K-T})$$

这也是 K-T 条件的经典表达形式。



有时 K-T 条件也叫 KKT 条件, 取自 Karush-Kuhn-Tucker 三个人的首字母。

虽然这些条件仍然是必要条件, 当可行域与优化目标具有某些性质时, 类似无约束最优化问题, 这些条件也能“升级”成充分必要条件, 之后所说的线性规划问题就是一个例子。

1.3 下降算法

1.3.1 基本定义

之前讨论的最优性条件是“最优解的判断”, 但是并没有说明得到最优解的方式。现实中, 在求解最优化问题时最常用的计算方法是迭代下降算法。以下的定义给出了算法的抽象表述:

定义 1.13 (算法映射)

空间 X 上的一个算法 \mathcal{A} 定义为 X 上的点到 X 上的集合的一个映射, 也即 $x \rightarrow \mathcal{A}(x) \subset X$ 。



直观的理解是, 迭代算法的目的是从初始点 x_0 开始, 得到一系列点 x_k , 使得它们可以趋向最优解。而将算法定义为点到集合的映射, 代表着 x_k 的下一个点 x_{k+1} 是在 $\mathcal{A}(x_k)$ 中任意选取的。



之所以不直接定义为点到点的映射, 是因为很多时候 (如非精确一维搜索时) 无法知道下一个点的具体取值, 只能确定在某个范围内。

仿照映射连续性的定义, 可以给出算法某种意义上连续的刻画:

定义 1.14 (算法闭性)

定义集合列 A_k 的“下闭极限” $\overline{\liminf}_{k \rightarrow \infty} A_k$ 为 $\{y \mid \exists y_k \in A_k, \lim_{k \rightarrow \infty} y_k = y\}$, 也即所有能成为集合列中逐个取点的极限点的点^a。

设 X, Y 是 $\mathbb{R}^p, \mathbb{R}^q$ 中的闭集, \mathcal{A} 为 X 上的点到 Y 的子集的函数, 若对任何满足 $\lim_{k \rightarrow \infty} x_k = x_0$ 的 x_k 有 $\overline{\liminf}_{k \rightarrow \infty} \mathcal{A}(x_k) \subset \mathcal{A}(x_0)$, 则称 \mathcal{A} 在 x_0 处是闭的。若对每个 $z \in Z$ 都有 \mathcal{A} 在 z 处是闭的, 则称 \mathcal{A} 在 Z 上是闭的。

^a此定义与记号为方便书写用, 并非专业定义/记号。



为了描述算法的性质, 我们还需要刻画迭代目标、迭代过程与迭代结果。迭代目标可以用一个集合 Ω 表示, 称为解集合, 如无约束优化中取所有梯度为 0 的点, 有约束优化中取所有 K-T 点。迭代过程与迭代结果则可以通过下降函数与收敛性表示:

定义 1.15 (下降函数、收敛性)

对解集合 $\Omega \in X$ 与 X 上的算法 \mathcal{A} , X 上的连续实函数 $\psi: X \rightarrow \mathbb{R}$ 称为关于 Ω 与 \mathcal{A} 的下降函数当且仅当:

$$\begin{cases} \psi(y) < \psi(x) & x \notin \Omega, y \in \mathcal{A}(x) \\ \psi(y) \leq \psi(x) & x \in \Omega, y \in \mathcal{A}(x) \end{cases}$$

对解集合 $\Omega \in X$ 与 X 上的算法 \mathcal{A} , 若从 $x_0 \in X$ 出发, 无论每次的 x_{k+1} 从 $\mathcal{A}(x_k)$ 中如何选取, 得到序列 $\{x_n\}$ 的每个收敛子列极限都在 Ω 中, 则称 \mathcal{A} 在 x_0 处收敛于 Ω 。若对每个 $y \in Y$ 都有 \mathcal{A} 在 y 处收敛于 Ω , 则称 \mathcal{A} 在 Y 上收敛于 Ω 。



作如上的抽象后, 就可以从数学角度找到算法收敛的必要条件, 也即:

定理 1.16 (收敛性条件)

对解集合 $\Omega \in X$ 与 X 上的算法 \mathcal{A} , 从某个初始点 x_0 出发, 每次 x_{k+1} 从 $\mathcal{A}(x_k)$ 中任意选取, 得到序列 $\{x_n\}$ 。在下列条件均满足时, $\{x_n\}$ 的每个收敛子列极限都在 Ω 中:

- $\{x_n\}$ 包含于 X 的某个紧子集中^a。
- 存在关于 Ω 与 \mathcal{A} 的下降函数 ψ 。
- \mathcal{A} 在 $X \setminus \Omega$ 上是闭的。

^a在 \mathbb{R}^n 上, 包含在紧集中即为有界。



证明 先证 $\psi(x_k)$ 的极限存在。由 x_k 包含在紧子集中, 其存在收敛子列 x_{k_i} , 设极限为 x 。由连续性可知 $\psi(x_{k_i}) \rightarrow \psi(x)$ 。另一方面, 根据条件 $\psi(x_k)$ 是单调下降的, 由数列极限知识即得 $\psi(x_k) \rightarrow \psi(x)$ 。

若 $x \notin \Omega$, 考虑子列 x_{k_i+1} , 其必然也有收敛子列, 假设收敛至 \bar{x} , 根据已证, 有 $\psi(\bar{x}) = \psi(x)$ 。然而, 由于 \mathcal{A} 在 Ω 补集上闭, $x_{k_i+1} \in \mathcal{A}(x_{k_i})$ 可推出 $\bar{x} \in \mathcal{A}(x)$, 根据下降函数定义与 $x \notin \Omega$ 即有 $\psi(\bar{x}) < \psi(x)$, 矛盾。

1.3.2 迭代方法概述

虽然我们从理论中得到了收敛性的结果, 对现实的算法, 迭代不可能无穷下去, 需要规定一些实用的终止迭代过程的准则, 一般称为**收敛准则**或**停机准则**, 例如 (ε 为充分小正数):

- $\|x_{k+1} - x_k\| < \varepsilon$ 或 $\frac{\|x_{k+1} - x_k\|}{\|x_k\|} < \varepsilon$
- $|f(x_{k+1}) - f(x_k)| < \varepsilon$ 或 $\frac{|f(x_{k+1}) - f(x_k)|}{|f(x_k)|} < \varepsilon$
- $\|\nabla f(x_k)\| < \varepsilon$

而评价算法优劣的一个重要标准就是收敛的快慢, 在数学上可以定义为:

定义 1.17 (收敛阶、线性收敛)

假设 $\lim_{k \rightarrow \infty} x_k = x^*$, 且满足 (假设所有 x_k 均非 x^*)

$$0 \leq \limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = \beta < \infty$$

则称 $\{x_n\}$ 以 p 阶收敛。

设所有这样的 p 构成集合 P , 称其上确界 $p_0 = \sup P$ 为 $\{x_n\}$ 的收敛阶。

若 $p_0 = 1$, 且存在 $0 < \beta < 1$ 使 p 取 1 时定义式成立, 则称 $\{x_n\}$ 是以收敛比 β 线性收敛的。若

$p_0 > 1$ 或 $p_0 = 1$ 且 p 取 1 时定义式中 $\beta = 0$ ，则称 $\{x_n\}$ 是超线性收敛的。



虽然名为线性收敛，实际上的逼近速度其实是指数量级，例如假设每个 $\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} = \frac{1}{2}$ ，则 $\|x_k - x^*\| \leq \frac{1}{2^k} \|x_0 - x^*\|$ 。

评价算法优劣的另一个重要标准是计算复杂性。虽然最优化可以追溯到十分古老的极值问题，但它成为一门独立的学科是在二十世纪四十年代末，而这个年代也恰恰是计算机从理论成为现实的年代。从 Dantzig 提出了求解一般线性规划问题的单纯形法起，各种最优化问题的理论及应用研究得到迅速发展，特别是线性规划由于其模型的普遍性和实用性，相关算法的进展引起广泛的重视。随着计算机技术的发展，实际问题的规模越来越大，计算复杂性成为了研究线性规划算法的重要标准，对非线性规划的算法也是如此。本讲义以基础理论的介绍为主，不将复杂度的分析与优化作为重点，更多相关理论可以参考附录文献。

绪论的最后，我们介绍两种基本的迭代算法结构。迭代算法的基本思想是：给定一个初始点，按照某一迭代规则产生一个点列，使得当其是有穷点列时，其最后一个点是最优化模型问题的最优解，否则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 x_k 能稳定地接近局部极小点 x^* 的小邻域，然后迅速收敛于 x 。一般地，对于某种算法，我们需要证明其迭代点列 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。

算法 1.18 (迭代算法-两步法)

1. 计算初始点 x_0 。
2. 构造价值函数 ψ ，取某个满足 $\nabla\psi(x_k)^T d < 0$ 的方向 d 为 d_k 。
3. 确定步长因子 α_k ，使得该价值函数值有某种程度的下降。
4. 计算 $x_{k+1} = x_k + \alpha_k d_k$ 。
5. 若满足事先给定的迭代终止条件则算法终止，否则回到第二步。



这是一种经典的算法结构，每次迭代分为确定方向与确定步长两个基本操作。价值函数 [merit function] ψ 可以直接取为目标函数 f ，也可以是某个与目标函数相关的可微函数。此外，也可以通过邻域的限制，每次找范围内的近似极小点，一步到位：

算法 1.19 (迭代算法-一步法)

1. 计算初始点 x_0 。
2. 构造价值函数 ψ 在 x_k 附近 (如一定半径内) 的二次近似函数 ψ_k 。
3. 计算 ψ_k 在范围内的最小点，得到 s_k 作为更新位移向量。
4. 计算 $x_{k+1} = x_k + s_k$ 。
5. 若满足事先给定的迭代终止条件则算法终止，否则回到第二步。



两种方法各有优劣，也有各自不同的适应场景，在之后的章节中，我们会进一步对这两种结构进行细化。

第2章 线性规划

内容提要

- 线性规划标准型
- 线性规划最优条件
- 单纯形法、单纯形表
- 对偶原理
- 灵敏度分析
- 更多线性规划方法

2.1 基本模型

2.1.1 标准型

由于问题与约束都是线性函数，线性规划问题总可以写成

$$\begin{aligned} \min \quad & c^T x \\ \text{s. t.} \quad & A_e x = b_e \\ & Ax \leq b \end{aligned} \quad (2.1)$$

其中 x, c 为 n 维向量， A_e 为 $m_1 \times n$ 阶矩阵， b_e 为 m_1 维向量， A 为 $m_2 \times n$ 阶矩阵， b 为 m_2 维向量。此处考虑约束时只考虑小于等于，不考虑小于，具体原因会在之后叙述。

写为上述形式的做法是，目标函数中的常数项不影响对 x 的求解，可以舍去，等式约束与不等式约束则均可移项写成对应的矩阵形式。为了进一步规范，在引入新的变量后，可以定义：

定理 2.1 (线性规划的标准型)

线性规划问题总可以等价为如下的标准型：

$$\begin{aligned} \min \quad & c^T x \\ \text{s. t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \quad (\text{LP-N})$$

其中 c, x 为 n 维向量， A 为 $m \times n$ 阶矩阵， b 为 m 维向量。为之后单纯形法计算需要，可进一步假设 $b \geq 0$ 。

“等价”指原问题全局最优解的集合和标准型全局最优解的集合在去除引入的新变量后完全相同。

证明 首先，若目标函数为 \max ，可添加负号转化为 \min 。对每个不等式约束 $a_i^T x \geq b_i$ ，可令 $x_t = b_i - a_i^T x$ ，即拆分为等式约束。对无约束的自由变量 x ，可拆分为约束 $x = x' - x''$ ， $x' \geq 0, x'' \geq 0$ 。最后，拆分后若有某个式子的 $b_i < 0$ ，直接对整个式子取相反数即可。

练习 2.1 将线性规划问题：

$$\begin{aligned} \max \quad & -4x_1 + x_2 + 3 \\ \text{s. t.} \quad & -x_1 + 2x_2 \leq 4 \\ & 2x_1 + 3x_2 \leq 12 \\ & x_1 - x_2 \geq 3 \\ & x_1 \geq 0 \end{aligned}$$

转化为标准型。

解 先将最优化目标转化为 $\min 4x_1 - x_2$ ，接着对约束条件进行处理。记 $\begin{cases} x_3 = 4 + x_1 - 2x_2 \\ x_4 = 12 - 2x_1 - 3x_2 \\ x_5 = x_1 - x_2 - 3 \end{cases}$ ，则

有 $x_3, x_4, x_5 \geq 0$ 。由 x_2 无约束，记 $\begin{cases} x_2 = x_6 - x_7 \\ x_6, x_7 \geq 0 \end{cases}$ ，于是可以排列出方程组 (这里已经调整了 b 的符号)：

$$\begin{pmatrix} -1 & 1 & 0 & 0 & 2 & -2 \\ 2 & 0 & 1 & 0 & 3 & -3 \\ 1 & 0 & 0 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} 4 \\ 12 \\ 3 \end{pmatrix}$$

再结合 $x = (x_1, x_3, \dots, x_7)^T \geq 0$ 即得到约束条件。

2.1.2 凸集基础知识

在给出了标准型以后，我们希望能够确定线性规划问题解的情况，以及解的可能性质。为了解决这些问题，我们需要先引入凸集的概念：

定义 2.2 (凸集)

若一个集合 $S \subset \mathbb{R}^n$ 满足 $\forall x, y \in S, \lambda \in [0, 1], \lambda x + (1 - \lambda)y \in S$ ，则称集合 S 是凸的。



直观来看，这也就是集合中的任意两点所连线段都属于此集合。

线性规划的每个约束条件，都相当于用若干维的平面去分割空间，而利用凸集的直观理解，可以想像，若被分割的区域是凸集，无论取分割出的哪一部分 (不等式约束) 还是相交的部分 (等式约束)，取出的部分仍然是凸集。不仅如此，每次都使用大于等于号分割，闭集的性质亦得到保持，重复此过程可以得到：

定理 2.3 (线性规划可行域性质)

线性规划问题的可行域是闭凸集。

证明 假设变量个数为 n ，由于 \mathbb{R}^n 为闭凸集，利用归纳法，从式 (2.1) 的形式中只需说明，定义集合 $A = \{x \in \mathbb{R}^n \mid a^T x = b\}$ 与集合 $B = \{x \in \mathbb{R}^n \mid a^T x \geq b\}$ ，当 T 是闭凸集时， $A \cap T$ 与 $B \cap T$ 都是闭凸集，这样每次新增条件后都是闭凸集。

利用极限保序性可以验证 B 是闭集，类似得 A 是闭集。由于闭集之交是闭集， $A \cap T$ 与 $B \cap T$ 是闭集。由定义，若 $a^T x_1 = a^T x_2 = b$ ，则 $a^T (\lambda x_1 + (1 - \lambda)x_2) = \lambda b + (1 - \lambda)b = b$ ，从而 A 是凸集，将前方等号换成大于等于号即知 B 是凸集。于是，只需要证明凸集之交是凸集，由定义知成立。综上即得证。

从而，为了研究线性规划的可行域性质，只需要研究闭凸集的性质。更准确来说，由于每步都是用超平面划分，最后划分出的一定是一个“多面体”的样子。从直观上看，如果可行域有界，多面体的

顶点连线可以得到棱，棱连线可以得到表面，表面再连线就能得到整体，也就是说，从几个特殊“顶点”可以“连接”出整个集合。这就引出了如下的定义：

定义 2.4 (极点、凸组合)

对非空凸集 S ，若 $x \in S$ 满足，只要存在 $x_1, x_2 \in S$ 使得 $x = \lambda x_1 + (1 - \lambda)x_2, \lambda \in (0, 1)$ ，则必须 $x_1 = x_2 = x$ ，那么称 x 为 S 的一个极点。

对于 \mathbb{R}^n 中的一些点 a_1, \dots, a_n ，定义它们的一个凸组合为 $\sum_{i=1}^n \lambda_i a_i$ ，其中 $\lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1$ 。



直观上，极点也就是“不处在某其他两点连接的线段中的点”，与上方多面体的顶点一致。凸组合则看成若干个点通过不断连接线段能连出的所有点。

定理 2.5 (凸集的等价定义)

集合 S 是凸的等价于其中任意有限个点的凸组合仍在其中。



证明 右推左：根据凸集的定义，两个点的凸组合就是 $\lambda x + (1 - \lambda)y, \lambda \in [0, 1]$ ，于是只要任意两个点的凸组合在其中，集合即为凸集，任意有限个点自然也正确。

左推右：归纳。任意两个点由凸集定义可知凸组合在其中。若任意 $n-1$ 个点的凸组合在其中，任意 n 个点时，若 $\lambda_1 = 1$ ，则其他 λ_i 必须为 0，实际上是一个点，否则 $\sum_{i=1}^n \lambda_i a_i = \lambda_1 a_1 + (1 - \lambda_1) \sum_{i=2}^n \frac{\lambda_i}{1 - \lambda_1} a_i$ 。由凸组合定义可知 $\frac{\lambda_i}{1 - \lambda_1}$ 均非负，且计算得和为 1，从而 $\sum_{i=2}^n \frac{\lambda_i}{1 - \lambda_1} a_i$ 是 a_2, \dots, a_n 的凸组合，记为 a_0 ，再由定义知 $\lambda_1 a_1 + (1 - \lambda_1) a_0$ 在凸集中，从而成立。

关于有界闭凸集，类似上方不断连线的过程可以得到如下性质：

命题 2.6 (有界闭凸集的性质)

有界闭凸集 = 其极点的凸组合得到的集合



但事实上，在凸集不是平面分割出时(如圆)，由于会出现无限个极点的情况，证明是有些复杂的，也不在线性规划的考虑范畴内，因此这里不予证明。

遗憾的是，对于无界集合，并不能简单通过极点确定。例如， \mathbb{R}^n 是凸集，但其不存在任何极点。考察几何可以发现，这时的凸集一定能向某些方向延伸，于是有定义：

定义 2.7 (方向、极方向)

设 $S \subset \mathbb{R}^n$ 为闭凸集， d 为一个非零向量。若 $\forall x \in S, \lambda \geq 0$ 有 $x + \lambda d \in S$ ，则称向量 d 为 S 的方向。

设 d_1, d_2 为 S 的两个方向，若对任何 $\lambda > 0$ 有 $d_1 \neq \lambda d_2$ ，则称它们不同。

对 S 的某方向 d ，若满足，只要存在 d_1, d_2 为 S 的方向使得 $d = \lambda_1 d_1 + \lambda_2 d_2, \lambda_1, \lambda_2 \geq 0$ ，则必须 $d_1 = d_2 = d$ ，那么称 d 为 S 的一个极方向。



值得注意的是，这里的方向对应的是射线的方向，而不是直线，例如 \mathbb{R} 有两个方向，1, -1，且两个方向都是极方向。

练习 2.2 给出以下集合的全部极点与极方向：

- $S := \{x \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \geq 1\}$
- $T := \{x \in \mathbb{R}^2 \mid x_1 + 2x_2 \geq 0, x_2 - x_1 \geq 0, x_2 \leq 1\}$

解 如图 2.1，由几何关系可以看出极点即为顶点，也即不等式约束能尽量多取等的点。因此分别考虑三

个不等式中的两个取等 (注意 S 前两个不等式取等不满足第三个不等式, 舍去) 可知 S 的极点为 $(0, 1)$ 与 $(1, 0)$, T 的极点为 $(0, 0), (-2, 1), (1, 1)$ 。 T 是有界集合, 不存在极方向, 而 S 的极方向由定义可以理解为方向中的“边界”, 于是为 $(0, 1)$ 与 $(1, 0)$ 。 S 有两个极点, 两个极方向; T 有三个极点, 没有极方向。

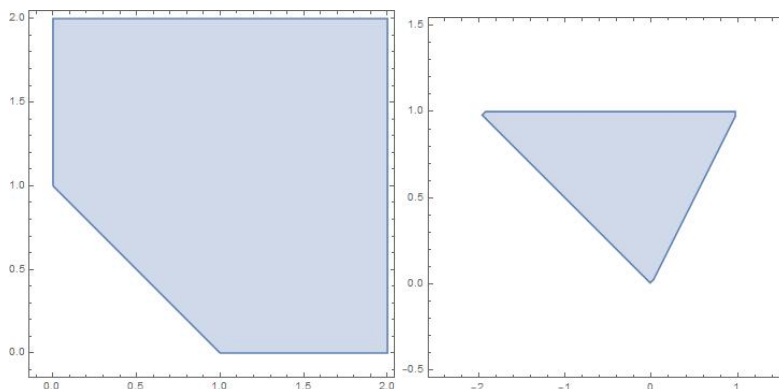


图 2.1: S 、 T 区域示意图

在有了极点与极方向后, 可以进一步刻画线性规划可行域的性质:

定理 2.8 (凸集表示定理)

设集合 $S = \{x \mid Ax = b, x \geq 0\}$ 非空, 则其满足:

1. 极点集非空有限;
2. 极方向集合为空等价于 S 有界;
3. S 无界时, 存在有限个极方向 $d^{(1)}, \dots, d^{(l)}$;
4. $x \in S \Leftrightarrow x = x_0 + \sum_{j=1}^l \mu_j d^{(j)}$, 其中 x_0 为极点的凸组合, 所有 $\mu_j \geq 0$ 。



证明

1. 为说明极点集合非空, 取 S 中为零的分量最多的点 (之一), 记为 y , 下面说明 y 是极点。若否, 有 $x^{(1)} \neq x^{(2)} \in S$ 满足 $\lambda x^{(1)} + (1 - \lambda)x^{(2)} = y, \lambda \in (0, 1)$ 。对 y 为 0 的分量, 由于 $x^{(1)}, x^{(2)} \geq 0$, 必然 $x^{(1)}, x^{(2)}$ 的对应分量都为 0, 于是由 y 是零分量最多的点可知 $x^{(1)}, x^{(2)}$ 的其他分量均不为 0。记 $\alpha = x^{(1)} - x^{(2)}$, 计算可知 $A\alpha = 0$, 于是 $y^{(t)} = y + t\alpha, t \in \mathbb{R}$ 都满足 $Ay^{(t)} = b$ 。记 r 为 $|\frac{y_i}{\alpha_i}|, \alpha_i \neq 0$ 中取到最小值的下标, 考虑 $y - \frac{y_r}{\alpha_r}\alpha$, 可验证其 $\in S$ 且比 y 多一个为 0 的分量 (即第 r 个分量), 矛盾。

为说明极点个数有限, 假设极点 y 的分量 y_{B_1}, \dots, y_{B_k} 非零, 我们证明 A 的第 B_1, \dots, B_k 列线性无关。这样一来, 假设这些列拼成的矩阵为 B_+ , 由于秩为 k , 可以从中选出可逆 k 阶方阵 B_- , 而 y, b 的对应分量为 y_-, b_- , 则有 $y_- = B_-^{-1}b_-$ 。也就是说, 对每一种非零分量选取, 得到的极点是至多唯一的, 因此极点总个数不超过选取数 2^n 。

记这些列为 a_{B_1}, \dots, a_{B_k} , 若线性相关, 则存在不全为 0 的 λ_i 使得 $\sum_i \lambda_i a_{B_i} = 0$ 。将 B_i 分量为 λ_i , 其他为 0 的向量记作 λ , 由定义可知对充分小的 δ 有 $y + \delta\lambda, y - \delta\lambda$ 均在 S 中, 而 y 为两者平均, 因此不是极点, 矛盾。

2. 当 S 有界时, 根据方向定义可知 S 方向集合为空, 极方向集合因而为空。否则, 我们先证明方向集合非空。假设已知 $Ax = b$ 的一个解 x_0 。由 S 无界, 对任何 n 存在满足 $\|x_n - x_0\|_1 > n$ 的

$x_n \in S$, 这里下标 1 表示 1 范数, 即各分量绝对值之和。记

$$y_n = \frac{x_n - x_0}{\|x_n - x_0\|_1}$$

则有 $Ay_n = 0, \|y_n\|_1 = 1$ 。由于 $\{y \mid \|y\|_1 = 1\}$ 为紧集, y_n 一定存在收敛子列 $\{y_{k_i}\}$, 记极限为 y_0 , 由连续仍有 $Ay_0 = 0$, 下面说明 y_0 是 S 的一个方向。

先说明 $y_0 \geq 0$ 。若 y_0 的某个分量 $y_{0t} < 0$, 根据极限定义可知对充分大的 i 有 $y_{k_i t} < -\epsilon = \frac{y_{0t}}{2}$ 。然而, 根据定义

$$x_{k_i t} = x_{0t} + \|x_{k_i} - x_0\|_1 y_{k_i t} < x_{0t} - k_i \epsilon$$

当 k_i 充分大时 $x_{k_i t} < 0$, 与 $x_{k_i} \in S$ 矛盾。

进一步地, 由 $Ay_0 = 0$ 与 $y_0 \geq 0$, 可计算知对任何 $x \in S$ 有 $x + \lambda y_0 \in S, \lambda \geq 0$, 从而 y_0 是 S 的方向。

下面说明方向集合非空时极方向集合非空且有限。

3. 由于 S 中要求 $x \geq 0$, 由定义可知 S 的方向 d 必须满足 $d \geq 0$ 。根据上一部分证明, 可记标准化方向集合 (由条件, 其中的 d 均为不同的方向)

$$D = \{d \mid \sum_i d_i = 1, Ad = 0, d \geq 0\}$$

由于这个集合也能写成 $\{x \mid A_0 x = b_0, x \geq 0\}$ 的形式, 且无界时由上一部分证明非空, 其极点必然存在有限。从而只需证明 D 的极点对应 S 的不同极方向。

若 d 不是 S 的极方向, 存在 d_1 不同于 d_2 , $\lambda_1, \lambda_2 \geq 0$ 使得 $d = \lambda_1 d_1 + \lambda_2 d_2$ 。记

$$d'_1 = \frac{d_1}{\|d_1\|_1}, d'_2 = \frac{d_2}{\|d_2\|_1}$$

则有

$$d'_1 \in D, d'_2 \in D, d = \lambda_1 \|d_1\|_1 d'_1 + \lambda_2 \|d_2\|_1 d'_2$$

而由于 d_1, d_2 所有分量非负, λ_1, λ_2 非负, 有

$$1 = \|d\|_1 = \|\lambda_1 d_1 + \lambda_2 d_2\|_1 = \lambda_1 \|d_1\|_1 + \lambda_2 \|d_2\|_1$$

从而取 $\lambda = \lambda_1 \|d_1\|_1$ 得

$$d = \lambda d'_1 + (1 - \lambda) d'_2$$

从而 d 不是 D 的极点。

反过来, 若 d 不是 D 的极点, 存在 $d_1 \neq d_2, \lambda \in (0, 1)$ 使得 $d = \lambda d_1 + (1 - \lambda) d_2$, 根据 D 中不同元素是 S 的不同方向即可知 d 不是 S 的极方向。

4. 右推左由方向定义与 S 是凸集可得证, 下面考虑左推右。我们进行归纳证明: 当 x 为一维向量时, 或 S 为空, 或 S 为单点集, 或 $S = \{x \mid x \geq 0\}$, 满足要求。下面假设 x 是 $n-1$ 维向量时满足, 考虑 n 维时。

根据上一问, 由于 D 有界 ($\sum_i d_i = 1$ 且 $d \geq 0$ 必然有界) 且凸, 其任何点可以写成极点的凸组合, 也即任何方向都可以写成极方向的凸组合。于是, 表示中的第二项即代表任何一个方向。假设 S 所有极点的凸组合构成 S_0 , 我们只需证明, 对任何 $x \notin S_0$ 存在 $x_0 \in S_0$, 有 $d_0 = \frac{x - x_0}{\|x - x_0\|_1} \in D$, 这样就有 $x = x_0 + \|x - x_0\|_1 d_0$ 。

若否, 对任何 x_0 , 由于 d_0 已满足 $\|d_0\|_1 = 1$ 与 $Ad_0 = 0$, 其若不是方向, 只能有小于 0 的分量。

记

$$\lambda_0 = \max_{\lambda} \{x_0 + \lambda d_0 \in S\}$$

由 $x \in S$, $\lambda_0 \geq \|x - x_0\|_1$, 而由于 d_0 有小于 0 分量, λ_0 必然存在。并记 $x_1 = x_0 + \lambda_0 d_0$, 并假设其第 p_1 个分量为 0 (根据定义必然有分量为 0)。注意到, x 能在 x_0 与 x_1 连线段上, 所以 x_1 亦不能写成 $x_0^{(1)} + \lambda^{(1)} d^{(1)}$ 的形式, 否则可直接表示出 x 。

记 $S_1 = S \cap \{x \mid x_{p_1} = 0\}$ 。由 x_1 存在可知非空, 且满足 S 的条件, 下面证明 S_1 的极点集与方向集恰好是 S 的极点集与方向集与 $\{x \mid x_{p_1} = 0\}, \{d \mid d_{p_1} = 0\}$ 的交集。根据定义, S_1 的方向是 S 的方向的子集, 从而 S 的极点、极方向一定是 S_1 的极点、极方向 (更大的集合中无法表示, 更小的集合中必然无法表示)。反过来, 若 S_1 有极点 s , 若其能写成 $\lambda a + (1 - \lambda)b, a, b \in S, \lambda \in (0, 1)$, 根据 $a_p \geq 0, b_p \geq 0, s_p = 0$ 可知 $a_p = b_p = 0, a, b \in S_1$, 从而只有 $a = b = s$ 。对极方向证明类似。由于 x_1 无法在 S 中被表示成 $x_0^{(1)} + \lambda^{(1)} d^{(1)}$, 其无法在 S_1 中被表示成 $x_0^{(1)} + \lambda^{(1)} d^{(1)}$ 。然而, 考虑

$$S_1^- = \{x^- \mid (a_1, \dots, a_{p-1}, a_{p+1}, \dots, a_n)x^- = b, x^- \geq 0\}$$

其极点、极方向与 S_1 去除第 p 个分量后完全对应, 于是 x_1 去除第 p 个分量后无法在 S_1^- 中表示。但是, x^- 是 $n - 1$ 维向量, 这与归纳假设矛盾。

2.1.3 解的存在性

确定可行域的性质之后, 就可以考察线性规划问题解的情况。假设可行域极点为 $x^{(1)}$ 到 $x^{(k)}$, 极方向为 $d^{(1)}$ 到 $d^{(l)}$, 则任何可行点可以写为 $\sum_i \lambda_i x^{(i)} + \sum_j \mu_j d^{(j)}, \sum_i \lambda_i = 1, \lambda_i, \mu_j \geq 0$ 。将其代入标准型中, 得到需要最小化的 $c^T x$ 成为

$$\sum_i \lambda_i (c^T x^{(i)}) + \sum_j \mu_j (c^T d^{(j)}) \quad (2.2)$$

在可行域非空时, 存在唯一解或无穷多解称为模型存在最优解, 而可行域为空或目标函数可达 $-\infty$ 称为最优解不存在。

定理 2.9 (线性规划问题解的性质)

在式 (2.2) 中, 标准形式线性规划模型最优解存在当且仅当所有 $c^T d^{(j)} \geq 0$ 。进一步地, 在最优解存在时, 一定在某个极点处能取到最优。



证明 若某个 $c^T d^{(j)} < 0$, 当 μ_j 增大时 $c^T x$ 可任意减小, 因此解不存在。根据式子形式可知

$$\sum_i \lambda_i (c^T x^{(i)}) + \sum_j \mu_j (c^T d^{(j)}) \geq \sum_i \lambda_i (c^T x^{(i)}) \geq \min_i c^T x^{(i)}$$

当且仅当 μ_j 全为 0, λ 在 $c^T x^{(i)}$ 最小的 i 处为 1, 其余为 0 时可取到最优解。



从几何来看, 线性规划问题相当于用某平面进行平移, 寻找与可行域有交时截距的最小值。于是, 存在最优解等价于不能在变小方向无限平移, 而如果能存在, 一定会在某顶点取到。从这个几何看法可以发现, 即使并未化为标准形式, 如果能直观看到极点与极方向, 仍然会满足上述性质。

练习 2.3 对可行域

$$1. S := \{x \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \geq 1\}$$

$$2. T := \{x \in \mathbb{R}^2 \mid x_1 + 2x_2 \geq 0, x_2 - x_1 \geq 0, x_2 \leq 1\}$$

分别求解线性规划问题 $\min -2x_1 - x_2$ 。

解

1. 作图可发现平面上 S 的极点为 $(0, 1), (1, 0)$, 极方向为 $(0, 1), (1, 0)$, 因此计算得解在 $(1, 0)$, 最优值 -2 。
2. 作图可发现平面上 T 的极点为 $(0, 0), (-2, 1), (1, 1)$, 因此计算得解在 $(1, 1)$, 最优值 -3 。

从解的性质证明过程中可以得到, 线性规划问题只有三种可能结果: 可行域为空, 无界 (即对任何 $M \in \mathbb{R}$, 存在 x 使得 $c^T x < M$), 或最优解存在。


2.2 单纯形法

2.2.1 可行基解

根据上方的推导, 想求解线性规划问题, 只需要在极点处进行即可, 于是, 我们需要寻找极点的更多性质。极点的定义可以说是从几何上得到的, 因此在这节进行一些代数上的推导。


首先, 对于标准型中的等式约束 $Ax = b$, 可以进行化简: 假设 A 有某些行向量线性相关, 不妨设有 $a_1^T = \sum_{i>1} \lambda_i a_i^T$, 这时若 $b_1 = \sum_{i>1} \lambda_i b_i$, 则第一个约束可以被其他约束推出, 从而可以删去; 否则产生了两个互相矛盾的等式约束, 可行域为空, 无意义。综合可不妨假设 $A_{m \times n}$ 是行满秩的, 由此其存在阶数为 m 的可逆子方阵。


假设第 B_1, B_2, \dots, B_m 列构成的子方阵可逆, 可形式上将 A 分块成 $\begin{pmatrix} B & N \end{pmatrix}$, 其中 B 包含了上方 B_i 的列, 同时将 x 对应分块为 $\begin{pmatrix} x_B \\ x_N \end{pmatrix}$ 。


 由于实际上 B 的行并不连续排列, 这是一个形式上的分块。

直接将等式约束代入, 可得 $Bx_B + Nx_N = b$, 于是 $x_B = B^{-1}b - B^{-1}Nx_N$ 。这也就意味着, 如此分块后, 等式约束中事实上只有 x_N 的部分是自由的, 剩下的 x_B 中分量可以由 x_N 解出来。

定义 2.10 (基解、可行基解)

对于任何可能的 B 的选取, 令 $x_N = 0$, 则 $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$, 这样的 x 称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 称为非基变量。
若这时的 x 又满足 $x \geq 0$, 即 $B^{-1}b \geq 0$, 则称为可行基解, 对应有可行基矩阵与可行基变量。 


 (可行) 基解的定义只与方程 $Ax = b$ 有关, 仍然与目标函数无关。

 **练习 2.4** 将 $S := \{x \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \geq 1\}$ 化为标准型, 并计算全部可行基解。与之前计算出的极点集合比较。

解 令 $x_3 = x_1 + x_2 - 1$, 则标准型为 $x \geq 0, x_1 + x_2 - x_3 = 1$, 基解有 $(1, 0, 0), (0, 1, 0), (0, 0, -1)$, 前两个为可行基解, 与之前所算出的极点一一对应。

接下来我们将看到, 对于化为标准形式的线性规划问题, 上面从代数角度计算出的可行基解与几何角度得到的极点集合是一致的:

定理 2.11 (可行基解与极点等价性)

对于标准形式线性规划问题的可行域 $S = \{x \mid Ax = b, x \geq 0\}$, 假设 A 是行满秩的, 则 S 的极点集合与可行基解集合相同。 

证明 假设 y 是极点, 根据凸集表示定理中的证明, 若 y_{B_1}, \dots, y_{B_k} 是 y 的非零分量, 则 A 的第 B_1, \dots, B_k 列线性无关。由于 A 至多 m 列线性无关, 必然有 $k \leq m$, 这就说明了 y 是基解, 而极点是可行点, 于是 y 是可行基解。

假设 y 是可行基解, 若存在 $y_1, y_2, \lambda \in (0, 1)$ 使 $y = \lambda y_1 + (1 - \lambda)y_2$, 设 $y_1 = \begin{pmatrix} y_{1B} \\ y_{1N} \end{pmatrix}, y_2 = \begin{pmatrix} y_{2B} \\ y_{2N} \end{pmatrix}$ (这里 B, N 分块与 y 一致)。由于

$$\lambda y_{1N} + (1 - \lambda)y_{2N} = 0, \lambda \in (0, 1), y_1 \geq 0, y_2 \geq 0$$

可得必须 $y_{1N} = y_{2N} = 0$, 而根据 $Ay_1 = Ay_2 = b$, 计算又可以证明 $y_{1B} = y_{2B} = B^{-1}b$, 从而 $y_1 = y_2 = y$, 得证。

结合这个定理与凸集表示定理, 可以得到很多有用的结论。例如, 通过标准形式可行域非空必存极点可以推出:

命题 2.12 (可行基解存在性定理)

若线性规划问题有可行解, 则必有可行基解。

而根据最优解存在时必能在极点取到即知:

命题 2.13 (线性规划问题解的性质 II)

在线性规划问题最优解存在时, 一定在某个可行基解能取到最优。

2.2.2 主要过程

上一节的讨论后, 我们已经说明, 只需要在可行基解处寻找最优解。因此, 单纯形算法的基本思想就是通过可行基解间不断迭代来找到最优的可行基解。这一节, 我们先考虑最优判定与转轴运算, 也即找到下一个可行基解的过程。以下假设 $A_{m \times n}$, 即 B 为 $m \times m$ 阶方阵, N 为 $m \times n - m$ 阶。

根据 $x_B = B^{-1}b - B^{-1}Nx_N$, 将 c 也对应划分为 c_B 与 c_N 后, 可直接计算出目标函数 $z = c_B^T x_B + c_N^T x_N = c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}N)x_N$ 。在 $c_N^T - c_B^T B^{-1}N$ 的每个分量都 ≥ 0 时, 由于可行解要求 $x_N \geq 0$, 在 $x_N = 0$ 时即取到了最小值 $c_B^T B^{-1}b$ 。记 $z_0 = c_B^T B^{-1}b, z_j = c_B^T B^{-1}a_j$, 即有:

定理 2.14 (最优解判定)

取定某个可行基解 $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ 后, 若对应的 B, N 满足 $c_j - z_j \geq 0, \forall j \in N$, 则 x 即为问题最优解。

证明 直接计算 $z = z_0 + (c_N^T - c_B^T B^{-1}N)x_N = z_0 + \sum_{j \in N} (c_j - z_j)x_j \geq z_0$ 得结论。

反之, 若最优解性质不成立, 一定存在某个 $p \in N$ 使 $c_p - z_p < 0$ 。这时, 适当增加 x_p 会使 z 更小 (注意增加 x_p 也意味着对应改变 x_B 的分量)。由于 $z = z_0 + \sum_{j \in N} (c_j - z_j)x_j$, 若 x_p 可以无限制增大, z 会趋于负无穷, 于是在解存在时, x_p 增大必然引起 x_B 的某个分量减小, 而临界值即为 x_B 的某个分量恰好减小到 0。这时, x_N 中的 x_p 分量增加到正, 而 x_B 中原本为正的某个分量减小到了 0, 非零分量仍然最多 m 个, 因此得到的仍然是可行基解, 这个过程就称为一次转轴运算。

算法 2.15 (单纯形法)**1. 初始化**

确定初始可行基划分 B, N ，并计算 $x_B = B^{-1}b$ 。

2. 最优判定

计算向量 $w = (B^T)^{-1}c_B$ ，对所有 $j \in N$ 计算 $z_j = w^T a_j$ 。若 $c_j \geq z_j$ 对一切 $j \in N$ 成立，则当前可行基解已是最优解，最优值为 $w^T b$ ，算法终止。否则，选择一个 $c_p - z_p < 0$ 的 $p \in N$ 进入下一步。

3. 转轴运算

计算向量 $y_p = B^{-1}a_p$ 。若 $y_p \leq 0$ ，则问题无有界解，算法终止。

否则，找出 r 使得 $\Delta = \frac{x_{B_r}}{y_{pr}}$ 是所有 $\frac{x_{B_i}}{y_{pi}}, y_{pi} > 0$ 中的最小值。

更新基变量：令 $x_p = \Delta$ ， $x_B = x_B - y_p \Delta$ ，这里 B 为原来的分量，更新后 x_{B_r} 由 Δ 定义成为 0。

更新 B, N ：将 B 原本的第 r 列 a_{B_r} 用 a_p 替换，并将 p 从 N 中移入 B 中下标， B_r 从 B 中移入 N 中下标。



算法中细化了转轴运算的过程，下面说明其正确性：

定理 2.16 (转轴运算正确性)

当上方算法中 $y_p \leq 0$ 时，问题不存在有界解；否则，转轴运算进行一步更新后，得到的仍然是可行基解，且目标函数值减少了 $(z_p - c_p)\Delta$ 。



证明 当 x_p 增加 δ ， x_N 其他分量保持为 0 时，由于要满足 $x_B = B^{-1}b - B^{-1}N x_N$ ，可得 $x_B = B^{-1}b - B^{-1}a_p \delta$ ，而函数值 $z = z_0 - (z_p - c_p)\delta$ 。

若 $y_p = B^{-1}a_p \leq 0$ ，则 δ 无论如何增加， x_B 都增大， $x_B \geq 0$ 仍满足，于是函数值可无限减小，无有界解。

否则，转轴运算取出的 Δ 满足 $\Delta \leq \frac{(B^{-1}b)_i}{y_{pi}}, \forall y_{pi} > 0$ ，这时化简可知 $(B^{-1}b)_i - (B^{-1}a_p)_i \Delta \geq 0$ 。另一方面， $y_{pi} \leq 0$ 时 $(B^{-1}b)_i - (B^{-1}a_p)_i \Delta \geq (B^{-1}b)_i \geq 0$ (由于 $x_B = B^{-1}b$ 是可行基解的一部分)，综合可知 $B^{-1}b - B^{-1}a_p \Delta \geq 0$ ，由更新过程，更新后 x_B 每个分量仍然 ≥ 0 。

于是，我们证明了更新后的 x 满足 B_r 分量变成 0， p 分量增加为正数，且所有分量仍然 ≥ 0 ， $Ax = b$ 仍成立，因此仍然是可行基解。另一方面，函数值减小了 $z_0 - ((z_p - c_p)\Delta) = (z_p - c_p)\Delta$ 。

注意到， $\Delta = \frac{x_{B_r}}{y_{pr}}$ ，由选取过程可知 $y_{pr} > 0$ ，而为了确认函数值是否真的减少，还需要考虑 $B^{-1}b$ 的零分量。

定义 2.17 (非退化可行基解)

当可行基解满足 $x_B = B^{-1}b > 0$ 时，其称为非退化的，否则其有分量为 0，称为退化的。



若迭代过程保持非退化，单纯形法具有良好的收敛性：

定理 2.18 (单纯形法有限收敛性)

若线性规划问题可以求出初始可行基解，且转轴运算中每步的可行基解都非退化，则有限次迭代后要么找到最优解，要么识别出问题无有界解，从而算法终止。



证明 当迭代过程非退化时，每一步 z 的减少量 $\frac{(z_p - c_p)x_{Br}}{y_{pr}} > 0$ ，因此每步找到的可行基解不同，又由于可行基解个数有限，必然会在找到某个后终止（可能找到最优或发现无有界解）。

2.2.3 初始化过程

上一部分讨论了单纯形算法的主要过程，不过有一个问题尚未解决，也即如何找到初始可行基解。两阶段法的思路是，先找到某辅助问题的最优解，再作为原问题初始可行基解进行计算。

算法 2.19 (两阶段法)

对问题 $\{\min c^T x \mid Ax = b, x \geq 0\}$ ，考虑如下问题：

$$\begin{aligned} \min \quad & \mathbf{1}^T y \\ \text{s. t.} \quad & Ax + y = b \\ & x \geq 0, y \geq 0 \end{aligned}$$

其中 $\mathbf{1}$ 代表每个分量都是 1 的向量， y 为 m 阶向量。

以 $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 进行单纯形法迭代（注意化为标准型的过程已经保证了 $b \geq 0$ ），若迭代到的最优解满足 $y = 0$ ，则此时的 x 为原问题可行基解，否则原问题不存在可行解。

证明 $\mathbf{1}^T y$ 即对 y 各个分量求和，而由于 $y \geq 0$ ，当且仅当 $y = 0$ 时其能取到最小值，此时 $Ax = b, x \geq 0$ ，因此 x 即为可行解。反之，最小值时若 $y \neq 0$ ，则说明可行解 x 不存在。

下面说明存在解时的 x 为可行基解。扩充 y 后，矩阵 A 事实上成为了 $(A \ I)$ ，而单纯形法找到的可行基矩阵是从 $(A \ I)$ 中选出了可逆的 m 列，又由于此时 $y = 0$ ，选出的 m 列必然在前 n 列之中，因此一定对应 A 的可行基划分（或从分量角度，找到的最优解一定至多 m 个分量非零，且全部为 x 的分量，因此是原问题可行基解）。

两阶段法的弊端在于，相当于要求解两个独立的线性规划问题，且对前一问题求解的信息只用来选取可行基解，没有进一步使用。**大 M 法** 对此作出了改进：

算法 2.20 (大 M 法)

对问题 $\{\min c^T x \mid Ax = b, x \geq 0\}$ ，考虑如下问题：

$$\begin{aligned} \min \quad & c^T x + M(\mathbf{1}^T y) \\ \text{s. t.} \quad & Ax + y = b \\ & x \geq 0, y \geq 0 \end{aligned}$$

其中 $\mathbf{1}$ 代表每个分量都是 1 的向量， y 为 m 阶向量， M 为取定的某充分大的数。

以 $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 进行单纯形法迭代，在 M 充分大时，若原问题最优解存在，则迭代到的结果中 $y = 0$ ，且 x 为原问题最优解。

直观上看，对于足够大的 M ， y 的“权重”是很大的，因此只要存在任何非零分量，都会导致 $c^T x + M(\mathbf{1}^T y)$ 无法取到最小。而当 $y = 0$ 时，取到最小即等价于 x 取到原问题的最优解。

对于 M 的具体取值, 可以这么来看: 假设在 $Ax = b$ 变为 $Ax = b - y$ 时, $c^T x$ 的最小值为 $f(y)$, 则只需要让 $M > \max_{y \neq 0} \frac{|f(y) - f(0)|}{1^T y}$, 就可以保证满足上述的条件。于是, 我们只需要估算 b 进行一定“扰动”时最小值的变化量, 而这会在后文灵敏度分析的部分解决。

2.2.4 单纯形表

单纯形表是单纯形方法的一种直观算法, 目的是将单纯形法的过程用矩阵消元运算进行直观表现。具体来说, 对于某个取定的可行基划分 B, N , 单纯形表是一个 $(m+1) \times (n+1)$ 维矩阵 (最上方一行为表头):

$$\begin{pmatrix} x_B & x_N & RHS \\ I_m & B^{-1}N & B^{-1}b \\ 0 & z_j - c_j & z_0 \end{pmatrix}$$

若最下方一行前 n 个分量均 ≤ 0 , 则最后一个分量即为最小值。更进一步地, 由于 $B^{-1}B = I_m$, $j \in B$ 时 $z_j - c_j$ 组合为 $c_B^T B^{-1}B - c_B^T = 0$, 单纯形表事实上的形式为:

$$\begin{pmatrix} x & RHS \\ B^{-1}A & B^{-1}b \\ c_B^T B^{-1}A - c^T & c_B^T B^{-1}b \end{pmatrix}$$

而转轴运算即为通过行变换, 将最末行与上方某行的倍数相加, 使某大于 0 的元素变为 0, 为 0 的元素变为非零。

由 $y_p = B^{-1}a_p$, 单纯形表的上方部分 $B^{-1}A$ 即为每个 i 对应的 y_i 。由此得到单纯形表的计算过程:

算法 2.21 (单纯形表)

1. 初始化

从某个初始的 $\begin{pmatrix} B^{-1}A & B^{-1}b \\ c_B^T B^{-1}A - c^T & c_B^T B^{-1}b \end{pmatrix}$ 开始。

2. 最优判定

若 $c_B^T B^{-1}A - c^T \leq 0$, 则右下角元素为最优解, 最优解时的 x 可观察左侧, 若最后一列外的某列只有一个元素为 1, 其他为 0, 则这列的列数即为 x_B 对应分量, 1 所在的这行的最右元素为其值。不在 x_B 分量中的元素值全为 0。否则, 在 $c_B^T B^{-1}A - c^T$ 找一个大于 0 的元素, 设列数为 p , 进入转轴运算。

3. 转轴运算

若第 p 列除最下方元素外均 ≤ 0 , 则问题无有界解, 算法终止。

否则, 找出 r 使得 $\Delta = \frac{x_{Br}}{y_{pr}}$ 是所有 $\frac{x_{Bi}}{y_{pi}}, y_{pi} > 0$ 中的最小值。这里 $\frac{x_{Bi}}{y_{pi}}$ 即为除最后一行外每行最后一列的元素与第 p 列元素之比。

更新单纯形表: 将第 r 行同除以 y_{pr} , 再将每行 (含最后一行) 减去第 r 行的倍数, 使得第 p 列只有第 r 行为 1, 其他均为 0。

证明 为了证明这个算法与之前的单纯形法等价, 需要观察转轴运算部分的减去倍数的过程。

对最后一行, 实际上减去的是第 r 行的 $\frac{z_p - c_p}{y_{pr}}$ 倍, 因此需要证明的是

$$\begin{aligned}(z_j - c_j)_{new} &= (z_j - c_j) - (B^{-1}A)_{rj} \frac{z_p - c_p}{y_{pr}} \\ (c_B^T B^{-1}b)_{new} &= c_B^T B^{-1}b - (B^{-1}b)_r \frac{z_p - c_p}{y_{pr}}\end{aligned}$$

根据定义, $B_{new} = (a_{B_1}, \dots, a_{B_{r-1}}, a_p, a_{B_{r+1}}, \dots, a_{B_m})$, 于是直接计算可知 $B^{-1}B_{new}$ 为 I 的第 r 列替换为 y_p 的结果。由此进一步计算得 (注意 $B_{new}^{-1} - B^{-1} = (I - B^{-1}B_{new})B_{new}^{-1}$)


$$\begin{aligned}(z_j - c_j)_{new} - (z_j - c_j) &= ((c_B^{new})^T - c_B^T)B_{new}^{-1}a_j + c_B^T(B_{new}^{-1} - B^{-1})a_j \\ &= (c_p - c_{B_r})e_r^T B_{new}^{-1}a_j + (c_{B_r} - c_B^T y_p)e_r^T B_{new}^{-1}a_j \\ &= (c_p - c_B^T y_p)e_r^T B_{new}^{-1}a_j\end{aligned}$$

由伴随矩阵, B_{new}^{-1} 的第 r 行与 B^{-1} 的第 r 行事实上只相差 $\det B / \det B_{new} = (\det B^{-1}B_{new})^{-1} = y_{pr}^{-1}$ 倍, 而 $e_r^T B^{-1}a_j = (B^{-1}A)_{rj}$, 第一个式子得证。对第二个式子, 将 a_j 换为 b 后完全相同展开得结果。

对其余行, 由于只进行了行变换, 也即 $B^{-1} \begin{pmatrix} A & b \end{pmatrix}$ 成为了 $PB^{-1} \begin{pmatrix} A & b \end{pmatrix}$, 相当于只需要验证 $PB^{-1} = B_{new}^{-1}$ 。在行变换后, x_B 除了第 r 个分量外所在的列均无变化 (它们的第 r 行为 0, 减 0 的倍数不会引起变化), 而第 p 列变为了只有第 r 行 1, 其他为 0, 因此与原来的第 $B_1, \dots, B_{r-1}, B_{r+1}, \dots, B_m$ 列可拼出 I_m , 这即说明了 $PB^{-1}B_{new} = I$, 于是 $PB^{-1} = B_{new}^{-1}$ 。

实际操作中, 大 M 法常与单纯形表结合使用。由于其相当于将 A 增广为了 $\begin{pmatrix} A & I \end{pmatrix}$, 初始可行基划分选取 I 的部分, $c_B = M \cdot \mathbf{1}^T$, 于是起始的单纯形表即为:

$$\begin{pmatrix} x & y & RHS \\ A & I & b \\ M(\mathbf{1}^T A) - c & 0 & M(\mathbf{1}^T b) \end{pmatrix}$$

 **练习 2.5** 用大 M 法构造单纯形表求解以下线性规划问题:

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

解 记 $x_4 = 11 - x_1 + 2x_2 - x_3, x_5 = 2x_1 + x_2 - 4x_3 - 3$, 则其标准形式的等式约束为

$$\begin{pmatrix} 1 & -2 & 1 & 1 & 0 \\ 2 & 1 & -4 & 0 & -1 \\ 1 & 0 & -2 & 0 & 0 \end{pmatrix} x = \begin{pmatrix} 11 \\ 3 \\ 1 \end{pmatrix}$$

于是构造的单纯形表为

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & y_1 & y_2 & y_3 & RHS \\ 1 & -2 & 1 & 1 & & 1 & & & 11 \\ 2 & 1 & -4 & & -1 & & 1 & & 3 \\ 1 & & -2 & & & & & 1 & 1 \\ 4M-1 & -M-1 & -5M+3 & M & -M & & & & 15M \end{pmatrix}$$

变形计算可最终得解为 $x_1 = 9, x_2 = 1, x_3 = 4$, 最优值为 -2 。



从最坏情况来看, 单纯形法并不是多项式时间算法, 而是指数时间算法, 因为其可能遍历所有可行基解。而针对线性规划问题的算法已有不少多项式时间的结果, 例如:

- 最先在理论上证明线性规划问题恒在多项式时间内可解的椭球算法。
- 可替代单纯形法的有效方法内点法。它亦可被推广并应用到不限于线性规划的更一般问题, 会在之后详述。

2.3 对偶理论

2.3.1 对偶问题定义

对于线性规划问题, 可以定义它的**对偶**, 而通过研究线性规划与其对偶的性质, 能得到更多的理论结果。为了进行对偶问题的定义, 需要另一种“标准形式”:

定理 2.22 (线性规划的对偶基本形式)

线性规划问题总可以等价于如下的形式:

$$\begin{aligned} \min \quad & c^T x \\ \text{s. t.} \quad & Ax \geq b \\ & x \geq 0 \end{aligned} \quad (\text{LP})$$

其中 c, x 为 n 维向量, A 为 $m \times n$ 阶矩阵, b 为 m 维向量。



证明 先化为LP-N的形式, 而 $Ax = b$ 即为 $Ax \geq b, -Ax \geq -b$, 于是可以写成 $\begin{pmatrix} A \\ -A \end{pmatrix} x \geq \begin{pmatrix} b \\ -b \end{pmatrix}$ 。

定义 2.23 (线性规划问题的对偶问题)

对于线性规划问题 (LP), 定义其对偶为

$$\begin{aligned} \max \quad & b^T w \\ \text{s. t.} \quad & A^T w \leq c \\ & w \geq 0 \end{aligned} \quad (\text{DP})$$

其中 w 为 m 维向量。



从约束与目标函数的对应, 即可以看出对偶的关系。在讨论对偶问题与原问题的对应性质之前, 我们先关注形式上的一些结果:

练习 2.6 计算如下问题的对偶:

1. $\min -b^T w \quad \text{s. t.} -A^T w \geq -c, w \geq 0$
2. $\min c^T x \quad \text{s. t.} Ax \geq b, -Ax \geq -b, x \geq 0$

解

1. 注意到 $\min -b^T w$ 与 $\max b^T w$ 等效, 这个问题事实上是计算对偶问题的对偶。结果为

$$\max -c^T x \quad \text{s. t.} -Ax \leq -b, x \geq 0$$

即

$$\min c^T x \quad \text{s.t. } Ax \geq b, x \geq 0$$

于是对偶问题的对偶就是原问题。

2. 这个问题事实上就是标准型的对偶，直接计算可以得到

$$\max \{b^T w_1 - b^T w_2\} \quad \text{s.t. } A^T w_1 - A^T w_2 \leq c, w_1, w_2 \geq 0$$

其中 w_1, w_2 都是 m 维向量。由于 $w_1 \geq 0, w_2 \geq 0$ ，可以将 $w_1 - w_2$ 看作无约束的 w ，即

$$\max b^T w \quad \text{s.t. } A^T w \leq c$$

这就是标准型的对偶结果。

2.3.2 对偶定理

下面，记 x 与 w 分别为问题 (LP) 与 (DP) 的某一可行解， x^* 与 w^* 为对应问题的最优解，讨论对偶的性质。

定理 2.24 (弱对偶定理)

1. $c^T x \geq b^T w$
2. $c^T x \geq b^T w^*, b^T w \leq c^T x^*$
3. $c^T x = b^T w \Rightarrow c^T x = c^T x^*, b^T w = b^T w^*$
4. $\min c^T x = -\infty \Rightarrow \nexists w$
 $\max b^T w = +\infty \Rightarrow \nexists x$



证明 当 $a \geq 0$ 时，若 $s \geq t$ ，由于每个分量对应成立大于等于，作正组合仍有 $s^T a \geq t^T a$ 。由此， $c^T x \geq (A^T w)^T x = w^T Ax = (Ax)^T w \geq b^T w$ ，第一个式子得证。由于对任何可行解都成立，当 x 或 w 取到一边最优解时仍成立，第二个式子得证。于是，当 $c^T x = b^T w$ 时，假设 $b^T w$ 取到的不是最大值，则会有 $b^T w' > b^T w$ ，矛盾，类似可知 $c^T x$ 取到的一定是最小值，第三个式子得证。最后，若 $c^T x$ 最小值无界，则任何 w 都无法满足 $c^T x \geq b^T w$ 对任何 x 成立，只能不存在可行解 w ，第四个式子的另一边同理。

本质来说，弱对偶定理可以被第一条 $c^T x \geq b^T w$ 概括，而这也体现出了对偶问题的价值：只要找到两个问题具有某种对应关系 ($c^T x = b^T w$) 的解，就一定找到了互相的最优解。从此出发，还可以得到其他有用的结论：

定理 2.25 (最优解存在性定理)

若 (LP) 与 (DP) 都有可行解，则它们都有最优解。



证明 分别记为 x_0 与 w_0 ，由弱对偶定理，任何可行解 x 满足 $c^T x \geq b^T w_0$ ，于是函数 $c^T x$ 有下界。这就排除了 (LP) 问题可行域为空与无界的情况，从而必须有最优解，对 (DP) 同理。

结合存在性定理，可以证明完整的对偶定理：

定理 2.26 (对偶定理)

若 (LP) 与 (DP) 一方有最优解，则另一方亦有最优解，且最优值一致。



证明 记 $y = Ax - b$, 则可写出问题 (LP) 的标准形式, 其中等式约束为 $\begin{pmatrix} A & -I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = b$ 。记 a_j 为 $\begin{pmatrix} A & -I \end{pmatrix}$ 的第 j 列, c_j 在原有列后均应扩充 0。设原问题最优解 x^*, y^* , 对应可行基分解为 B, N , 根据之前推导有 $c_B^T B^{-1} a_j - c_j \leq 0$ 。

记 $w = c_B^T B^{-1}$, 则上方条件在 $1 \leq j \leq n$ 时为 $A^T w \leq c$, 在 $j > n$ 时为 $-w \leq 0$, 于是 w 是对偶问题可行解, 而计算得 $w^T b = c_B^T B^{-1} b$ 为原问题最优解, 因此由弱对偶定理可知 w 为对偶问题最优解, 最优值一致。

由于 (DP) 的对偶与 (LP) 等价, 若对偶问题有最优解, 则同样得到原问题也有最优解, 最优值一致。

从对偶定理出发, 可以得到一个重要的判别条件:

定理 2.27 (互补松弛定理)

若 (LP) 与 (DP) 有可行解 x^* 与 w^* , 则 x^*, w^* 都是最优解的充要条件为:

$$\begin{cases} x^{*T}(A^T w^* - c) = 0 \\ w^{*T}(Ax^* - b) = 0 \end{cases}$$



证明 充分性: 利用弱对偶定理第一条的证明过程, 条件成立时即有 $c^T x^* = b^T w^*$, 由弱对偶定理第三条可知两者都是最优解。必要性: 考虑逆否命题, 若这两个式子中有不成立的, 仍然利用弱对偶定理第一条证明过程, 必然有 $c^T x^* > b^T w^*$, 与最优值相等矛盾, 于是不可能二者都是最优解。

从而, 只要知道了其中一个问题的最优解, 就可以根据互补松弛定理理解出另一个最优解。一个例子是, 若原问题是标准型, 其最优解用可行基划分表示为 $\begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$, 则最优值为 $c_B B^{-1} b$ 。练习 2.6 中给出了标准型问题的对偶形式, 取 $w = B^{-T} c_B^T$ 时, 即满足 $b^T w = c^T x$, 验证可知满足约束条件 (计算发现 $A^T w \leq c$ 即为原问题的最优判定条件), 于是这就是对偶问题的最优解。

对偶问题的另一个有趣的性质是, 对 (LP) 考虑绪论中提到的 (K-T) 条件。构造乘子 w, w_0 , 则 $L(x, w, w_0) = c^T x - w^T (Ax - b) - w_0^T x$, 写出此时的 K-T 条件:

$$\begin{cases} c - A^T w - w_0 = 0 \\ Ax \geq b, x \geq 0 \\ w \geq 0, w_0 \geq 0 \\ w^T (Ax - b) = 0, w_0^T x = 0 \end{cases}$$

将 $w_0 = c - A^T w$ 代入, 即得到:

$$\begin{cases} Ax \geq b, x \geq 0 \\ w \geq 0, A^T w \leq c \\ w^T (Ax - b) = 0, x^T (A^T w - c) = 0 \end{cases}$$

而这恰恰是原问题、对偶问题的要求与互补松弛条件。这意味着, 对于线性规划问题, K-T 条件是一个充分必要条件。

2.3.3 灵敏度分析

最后，我们来给出对偶问题的一个实际含义。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。此时，势必会出现系数的扰动及引起的变化。所谓灵敏度分析，便是当系数有微小变化时最优解的反应。

仍然考虑标准形式，在 b 作 Δb 的扰动后，问题变为

$$\begin{aligned} \min \quad & c^T x \\ \text{s. t.} \quad & Ax = b + \Delta b \\ & x \geq 0 \end{aligned}$$

假设已经得到了原问题的最优解 $\begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ ，其满足可行性条件 $B^{-1}b \geq 0$ 与最优性条件 $c_N^T - c_B^T B^{-1}N \geq 0$ (这里相当于对每个 $c_j - z_j$ 合为整体向量)。

定理 2.28 (扰动问题的最优解)

假设扰动后的 $b + \Delta b$ 满足可行性条件 $B^{-1}(b + \Delta b) \geq 0$ ，则 $\begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}(b + \Delta b) \\ 0 \end{pmatrix}$ 是扰动后问题的最优解，新的最优值即为 $c_B^T B^{-1}(b + \Delta b)$ 。

证明 由于最优性条件与 b 无关，只要 $b + \Delta b$ 仍满足可行性条件， $\begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}(b + \Delta b) \\ 0 \end{pmatrix}$ 保持最优性条件，从而是扰动后的最优解。直接计算得扰动后的最优值。

记原问题目标函数最优值是 $z(b) = c_B^T B^{-1}b$ ，新问题最优值 $z(b + \Delta b) = c_B^T B^{-1}(b + \Delta b)$ ，此时 $z(b + \Delta b) - z(b) = c_B^T B^{-1}\Delta b$ 。当原问题最优解非退化 (回顾其定义是 $B^{-1}b$ 每个分量都大于 0) 时，只要 $\|\Delta b\|$ 足够小，总能满足 $B^{-1}(b + \Delta b) \geq 0$ 。于是得结论：

定理 2.29 (解的扰动)

当线性规划问题标准型的最优解 $\begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ 非退化时，有 $\nabla_b z(b) = B^{-T}c_B$ 。

证明 由定义， $\nabla_b z(b) = \lim_{\|\Delta b\| \rightarrow 0} \frac{z(b + \Delta b) - z(b)}{\Delta b} = B^{-T}c_B$ ，这里对向量的除法代表除以每个分量后拼成向量。

而根据之前的讨论， $B^{-T}c_B$ 即为标准型问题对偶问题的最优解 w^* 。

第3章 网络流与动态规划

内容提要

- 运输问题
- 最短路问题
- 最大流问题
- 最小成本(循环)流
- 动态规划基本思路
- 背包问题、设备更新问题

3.1 网络最优化

网络最优化问题包括运输问题、最短路问题、最大流问题等等,是线性规划的特例,也有着重大的实际意义。因为网络模型的特殊数学结构,可以设计出比一般线性规划效率更高的求解算法。本节讨论两个重要的例子,由于重在解法与思路,省略部分证明的细节。

3.1.1 运输模型

定义 3.1 (运输问题)

形如


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

的线性规划问题被称为运输问题。



有时会放宽限制,将条件变为 $\sum_{j=1}^n x_{ij} \leq s_i$ 与 $\sum_{i=1}^m x_{ij} \geq d_j$ 。

将 s_i 看作第 i 个工厂的货物量, d_j 看作第 j 个需求点的需求量, c_{ij} 看作第 i 个工厂到第 j 个需求点的运费,运输问题的实际意义便是找一个总运费最小的供货方案。在收支平衡的情况(也即不放宽条件的限制), $\sum_{i=1}^m \sum_{j=1}^n x_{ij} = \sum_{i=1}^m s_i = \sum_{j=1}^n d_j$, 于是必须 $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$ 才可能有解。

 **练习 3.1** 计算运输问题的对偶问题。

解 根据标准型对偶的结果,直接计算(注意矩阵 A 的形式)可得对偶问题为:

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

事实上,由于 $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$, 所有 u_i 同加 t 、 v_j 同减 t 后满足的约束不变,目标函数也不变,因此可以不妨取定某一个的值,如假设 $u_1 = 0$ 。

对运输问题的解法和单纯形法的想法一致，都包含确定初始可行基解、最优判定与转轴运算的过程。自然，由于运输问题的特殊性，可对算法进行一些调整。先引入回路的概念：

定义 3.2 (回路)

对于一些互不相同的下标 $(i_1, j_1), \dots, (i_{2k}, j_{2k})$ ，若它们满足 $i_{2p-1} = i_{2p}$ 与 $j_{2p} = j_{2p+1}$ ，其中 $p = 1, 2, \dots, k$ ，且 j_{2k+1} 定义为 j_1 ，则称这些下标构成一个回路。

若一族下标中能挑出一些组成回路，则称这族下标中存在回路。



直观来看，这就代表着这些下标构成的点在平面上顺次连接，可以形成一个封闭图形，且每次连线交替运用水平线与竖直线。最简单的例子是，一个长方形的四个角的坐标可以形成一个回路。

有了回路的定义，就可以考虑运输问题的可行基解：

定理 3.3 (运输问题的可行基解)

在运输问题中，一个可行解 x 是可行基解当且仅当其至多有 $m+n-1$ 个下标非零，且非零分量的下标中不存在回路。



证明 首先，虽然限制条件共有 $m+n$ 个方程，但由于 $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$ 的要求，最后一个方程可以被其他方程推出，因此事实上只有 $m+n-1$ 个独立方程，从而可行基分量个数为 $m+n-1$ 个。

将方程组(含最后一个)写成 $Ax = b$ 的形式，由于每个 x_{ij} 在恰好出现两次，也就对应 A 的每一列恰好有 i 与 $m+j$ 两个元素为 1，其他为 0。若下标出现回路，假设回路中的分量为 $(i_1, j_1), \dots, (i_{2k}, j_{2k})$ ，对应 A 的第 a_1, \dots, a_{2k} 列，考虑 $\sum_{i=1}^{2k} (-1)^i a_i$ ，由于相邻列前系数正负不同，对应相同的分量互相抵消，因此此式结果为 0，从而 A 的这些列线性相关，不可能成为可行基的一部分。

反之，若 A 的某些列线性相关，则假设 $\sum_k \lambda_k a_k = 0, \lambda_k \neq 0$ ，由于对每个 a_i 必须有和它有共同非零分量的列消去，通过排列可以得到其中存在回路。

综上即证明了，非零分量的下标不存在回路等价于它们对应的列线性无关，而利用线代知识这又等价于存在包含它们的可行基划分(这里由于 A 的各行不独立，可行基划分只需要找出互相线性无关的 $m+n-1$ 列，即可以去掉某行得到可逆矩阵 B)，于是它们构成可行基解。

根据这个要求，有一个简单的方法寻找可行基解：

算法 3.4 (运输问题初始化-西北角法)

注意：算法中对 s_i, d_j 的修改只是暂时的，在算法结束后还原。


1. 初始化下标 $i = 1, j = 1$ 。
2. 若当前 i, j 对应的 s_i 为 0，则 i 增加 1；否则若 d_j 为 0，则 j 增加 1(都为 0 时只有 i 增加)。当 $i > m$ 或 $j > n$ 时停止，否则进入下一步。
3. 令 $x_{ij} = \min\{s_i, d_j\}$ ，记录其下标为可行基(即使 $x_{ij} = 0$)，并将 s_i, d_j 都减去 x_{ij} ，回到上一步。



证明 这个算法相当于：先让第一家工厂给第一个需求点供货，如果工厂货物量用完就换工厂，需求点拿到足够的货物就换需求点，直到全部供完。以此思路容易证明可行性。

在这样的计算过程中，每次确定的 i, j 都只增不减，假设最后选出的下标为 (i_1, j_1) 到 (i_t, j_t) ，由过程可以发现每一个比起前一个 i 增加 1 或 j 增加 1，从 $(1, 1)$ 到 (m, n) 一共选出了 $m+n-1$ 个 x_{ij} 。此外，这样的增加方式也与存在回路的定义矛盾，因为如此形成的是必须一条折线，不会存在环。

综上即证明这个算法选出的是可行基解。

 **练习 3.2** 对以下的供求表利用西北角法计算初始可行基解 (这里供给、需求表示总供给量与需求量, S_i, D_j 交叉位置表示供给成本, 不过计算初始可行基解的过程事实上与供给成本无关)。

	D1	D2	D3	D4	供给
S1	10	2	20	11	15
S2	12	7	9	20	25
S3	4	14	6	18	10
需求	5	15	15	15	

解 从左上角开始, S1 的供给为 15, D1 的需求为 5, 可以完成供给, 进入第二家:

	D1	D2	D3	D4	剩余
S1	5	当前			10
S2					25
S3					10
剩余	0	15	15	15	

S1 剩下的部分不足以供应完 D2, 因此供应 10 单位后换下一家工厂:

	D1	D2	D3	D4	剩余
S1	5	10			0
S2		当前			25
S3					10
剩余	0	5	15	15	

重复此过程, 直到走到右下角:

	D1	D2	D3	D4	剩余
S1	5	10			0
S2		5	15	5	0
S3				当前	10
剩余	0	0	0	10	

完成初始解构造:

	D1	D2	D3	D4	剩余
S1	5	10			0
S2		5	15	5	0
S3				10	0
剩余	0	0	0	0	

值得注意的是, 上面构造出可行基解时同时选取了可行基划分。虽然可行基解中未必有 $m+n-1$ 个非零分量, 但可行基划分中得到的一定是 $m+n-1$ 个下标。有了可行基划分以后, 就需要判断其

是否是最优解。利用对偶问题与互补松弛定理，可以得到引理：

定理 3.5 (标准型的互补松弛条件)

对于线性规划问题的标准型， x 是最优解当且仅当存在 w 使得：

$$\begin{cases} Ax = b, x \geq 0 \\ A^T w \leq c \\ x^T (A^T w - c) = 0 \end{cases}$$



证明 根据互补松弛定理下方的推论，线性规划问题的 K-T 条件是充分必要条件。仍然构造乘子 w, w_0 满足 $L(x, w, w_0) = c^T x - w^T (Ax - b) - w_0^T x$ ，直接计算此时的 K-T 条件 (注意 $Ax - b$ 这时对应等式约束)，并代入 $w_0 = c - A^T w$ 得结果。

从而可以进行判定：

算法 3.6 (运输问题最优解判定-势函数法)

1. 构造 m 维向量 u 与 n 维向量 v ，并固定 $u_1 = 0$ 。
2. 对可行基划分中的每个 (i, j) ，构造方程 $u_i + v_j = c_{ij}$ 。通过这 $m + n - 1$ 个方程解出除了 u_1 外的 $m + n - 1$ 个分量。
3. 对所有 (i, j) ，计算检验数 $\sigma_{ij} = u_i + v_j - c_{ij}$ ，若全部 ≤ 0 ，则得到了最优解，否则选取某个使其为正的 i, j 进行转轴运算。



证明 证明算法成立需要两个部分：证明这 $m + n - 1$ 个方程的系数矩阵线性无关，并且这样可以判定最优解。

将方程组排列成 $T \begin{pmatrix} u \\ v \end{pmatrix} = c_B$ 的形式，根据定义可以发现， c_B 即为选取的可行基划分中 c 对应的分量 (即与第一章的 c_B) 一致，而 T 相当于 A 划分出的可行基矩阵对应的 $m + n - 1$ 列的转置，记为 B_+^T 。由于 A 的形式， B_+ 对每行有对称性，因此去掉任何一行都可逆 (证明细节需要利用置换方阵，略去)，于是固定 u_1 后的系数矩阵是 B_+^T 去掉一列后的可逆方阵，存在唯一解。

根据方程组的计算过程，这样构造的 u, v 满足对 x 所有非零的分量都有 $u_i + v_j - c_{ij} = 0$ ，即为对应的 $A^T w - c = 0$ ，而 x 为 0 的分量也满足，因此有 $x^T (A^T w - c) = 0$ ，于是满足互补松弛条件，判定最优解也就只需要 u_i, v_j 满足约束条件，即 $u_i + v_j \leq c_{ij}$ 。

练习 3.3 对上一个练习中构造的初始可行基解用势函数法计算检验数。

解 按照非零分量可以构造方程组 $\begin{cases} u_1 = 0, u_1 + v_1 = 10, u_1 + v_2 = 2 \\ u_2 + v_2 = 7, u_2 + v_3 = 9, u_2 + v_4 = 20 \\ u_3 + v_4 = 18 \end{cases}$ ，逐个代入可以解出 $u =$

$(0, 5, 3)^T, v = (10, 2, 4, 15)^T$ ，于是检验数为：

	D1	D2	D3	D4
S1	0	0	-16	4
S2	3	0	0	0
S3	9	-9	-9	0

因此，可以选取大于 0 的分量 (如 (3, 1) 进行转轴运算)。



从这个过程也可以看出, 由于势函数法方程组的特殊性, 对其求解并不需要显式计算矩阵逆, 只需要以合适的顺序代入即可, 这可以显著降低计算复杂度。

为了进行转轴运算, 又需要一个与回路相关的引理:

定理 3.7 (回路存在性)

在 $i = 1, \dots, m; j = 1, \dots, n$ 的范围内选取 $m + n$ 个 (i, j) , 必然存在回路。



证明 利用可行基解的证明过程, 不存在回路等价于 A 中选出的对应列线性无关, 而已经证明了 A 的秩是 $m + n - 1$, 选出的 $m + n$ 列必然有线性相关的, 即得证。

这样, 选中进行加入转轴运算的下标与原本的 $m + n - 1$ 个可行基结合后, 一定能找到一个回路, 且由于原本可行基不存在回路, 回路必须包含新下标, 由此可以进行操作:

算法 3.8 (运输问题转轴运算)

1. 已知可行基划分与准备新加入可行基的下标 (a, b) 。
2. 原有可行基加入新下标后存在回路, 记为 $(i_1, j_1) = (a, b), (i_2, j_2) \dots, (i_k, j_k)$ 。
3. 记 $\Delta = \min_{t=1}^k \{x_{i_{2t}j_{2t}}\}$, 所有 $x_{i_{2t-1}j_{2t-1}}$ 增加 Δ , 所有 $x_{i_{2t}j_{2t}}$ 减小 Δ , $t = 1, \dots, k$ 。



证明 直接计算可知这个过程保持了可行性, 且最终达到的非零分量个数不会超过 $m + n - 1$, 新加入的分量替换了原有的非零分量, 因此可以说明这事实上就是一次转轴运算的操作, 从而根据单纯形法转轴运算过程的理论推导可知算法正确性。

综合以上部分, 我们最终得到了运输问题求解的单纯形法 (由可行域有界, 不会出现无界情况):

算法 3.9 (运输问题-单纯形法求解)

1. 利用西北角法构造可行基解。
2. 利用势函数法计算当前可行基解对应的检验数, 若全部 ≤ 0 则算法结束, 否则挑选大于 0 分量进入下一步。
3. 进行转轴运算得到新的可行基解。



练习 3.4 用单纯形法解决前述练习中的运输问题:

	D1	D2	D3	D4	供给
S1	10	2	20	11	15
S2	12	7	9	20	25
S3	4	14	6	18	10
需求	5	15	15	15	

解 先构造初始可行基解, 并计算对应的检验数 (方括号中为检验数, 由于可行基解部分的检验数一定为 0, 可以省略不写):

	D1	D2	D3	D4
S1	5	10	[-16]	[4]
S2	[3]	5	15	5
S3	[9]	[-9]	[-9]	10

选取左下角元素作转轴运算 (加粗的为选取的回路):

	D1	D2	D3	D4
S1	5-5	10+5		
S2		5-5	15	5+5
S3	0+5			10-5

重新计算检验数 (这里出现了退化的可行基解, 可以任取一个上一步减去的 0 作为可行基):

	D1	D2	D3	D4
S1	[-9]	15	[-16]	[4]
S2	[-6]	0	15	10
S3	5	[-9]	[-9]	5

以右上角再进行一次转轴运算:

	D1	D2	D3	D4
S1		15-10		0+10
S2		0+10		10-10
S3	5			5

再次计算检验数, 此时可以发现达到了最优解。

	D1	D2	D3	D4
S1	[-13]	5	[-16]	10
S2	[-10]	10	15	[-4]
S3	5	[-5]	[-5]	5

3.1.2 最短路问题

对下面的网络问题或是流的相关问题, 需要先定义网络模型:

定义 3.10 (网络模型)

设 $G = (V, E)$ 为有向图, 其中 V, E 分别为顶点、边的集合。有时, 会在顶点中选取特殊对待的起点 s 与终点 t 。对每条边赋予成本 $c(e)$ 与容量 $u(e)$ (流量一般为非负实数, 成本为实数, 且都可能引入无穷), 概括这些元素的 $\mathcal{N} = (G, s, t, c, u)$ 称为网络。



本节讨论的最短路径问题中, 不考虑容量, 成本 $c(e)$ 解释为边的“长度” (但可能会有负长度), 考虑基本问题: 在网络 (G, s, c) 中求某源点 s 到所有其他点的最短路径及其长度 (也称为单源最短路径问题)。



求某点 s 与另一点 t 间的最短路径问题最差情况复杂度与求 s 到所有其他点相同, 因为最差情况需要确认所有点后才知到 t 的最短路径是什么。最短路径问题的另一个基本问题是求所有点对间的最短路径, 这可以通过求解 $|V|$ 次单源最短路径问题来求解, 但有复杂度更优化的写法。

定义 3.11 (路径、回路、路径长度)

对一系列点 v_1, v_2, \dots, v_n , 若 $(v_i, v_{i+1}) \in E, i = 1, \dots, n-1$, 则其称为 v_1 到 v_n 的路径。若 $v_1 = v_n$, 则称为回路。

最短路问题中, 一条路径的长度定义为 $\sum_{i=1}^{n-1} c(v_i, v_{i+1})$ 。



最短路问题有基本的最优子结构性质:

定理 3.12 (最短路问题-最优子结构)

若 $s, v_{i_1}, \dots, v_{i_n}, t$ 是 s 到 t 的一条最短路径, 则 $s, v_{i_1}, \dots, v_{i_k}$ 是 s 到 v_{i_k} 的一条最短路径。



证明 若否, 由长度定义可发现, 将 $s, v_{i_1}, \dots, v_{i_k}$ 替换为最短路径, 可以使总长度更短。

由此可以说明:

定理 3.13 (最短路问题-解的性质)

假设存在 s 到所有点的路径, 则存在 s 到所有点的最短路径当且仅当图中没有长度为负的回路, 且 s 到所有点的最短路径可以用以 s 一棵树来表示, 称为最短路树。



证明 存在最短路径 \Rightarrow 无负回路: 若有负回路, 沿其绕圈可使路径长度不断减少,

无负回路 \Rightarrow 存在最短路径: 这时, 假设 s 到 v_k 的某路径 s, v_1, \dots, v_k 中有 $v_i = v_j, i \neq j$, 则将 v_i 与 v_j 之间的部分 (v_i, v_j) 去掉其中一个, 相当于去掉这个回路) 去掉后总长度更小, 因此所有不含重复点的路径中最短的就是最短路径, 而不含重复点的路径是有限的, 因此最短路径存在。

最短路树存在性: 选取 s 到每点的一条最短路径, 使得不重复的边数最少。若这些路径的并不能构成一棵树, 必须 s 到某点 v 出现两条路径, 根据最优子结构性质, 这两条路径必须都是最短路径, 假设两条路径中 v 前的点分别为 v_a, v_b , 则去掉 (v_a, v) 后剩下的图中依然有到每点的最短路径, 与不重复边数最少矛盾。

用运筹学的语言, 最短路问题可以如下描述:

定义 3.14 (净流出量)

对一个网络 $G = (V, E)$ 与边集到实数的映射 $x: E \rightarrow \mathbb{R}$, 定义这个映射所对应的每点“净流出量” $f_x: V \rightarrow \mathbb{R}$ 为 $f_x(v) = \sum_{(v, v_i) \in E} x(v, v_i) - \sum_{(v_j, v) \in E} x(v_j, v)$ 。



这个定义事实上是将 x 看作了每条边从起点到终点的流量。此外, 有限网络中, 由于顶点可用正整数表示, 常将 v_i 与 i 等效, s, t 此时看作两个正整数, $x(i, j), u(i, j)$ 等写作 x_{ij}, u_{ij} 。

定义 3.15 (两点最短路问题)

形如

$$\begin{aligned} \min \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{s.t.} \quad & f_x(s) = -f_x(t) = 1 \\ & f_x(k) = 0 \quad k \in V \setminus \{s, t\} \\ & x_{ij} \geq 0 \quad (i, j) \in E \end{aligned}$$

的线性规划问题为两点间的最短路问题。



从数学定义中可以看出, 最短路问题是运输问题的一种特殊形式(对所有 $(i, j) \notin E, i \neq j$ 可记 $c_{ij} = \infty$, 且记 $c_{ii} = 0, \forall i$, 假设每个点的供给为 $\begin{cases} M+1 & i=s \\ M & i \neq s \end{cases}$, 需求为 $\begin{cases} M+1 & i=t \\ M & i \neq t \end{cases}$, 其中 M 为充分大实数), 此外, 之后会说明, 假设已知最短路径, 让最短路径上的边对应的 $x = 1$, 否则 $x = 0$, 则这就是原问题的一个最优解, 因此这个问题与两点最短路问题等价。

练习 3.5 计算两点最短路问题的对偶问题。

解 根据定义, 类似运输问题可计算得对偶问题为

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

从对偶问题形式中更容易看出其为运输问题的特殊情况。与运输问题相同, 可记 $y_t = 0$, 则这时取 y_i 是 i 到 t 的最短路径长度就构成一个最优解。

定义 3.16 (单源最短路问题)

形如

$$\begin{aligned} \min \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{s. t.} \quad & f_x(s) = |V| - 1 \\ & f_x(k) = -1 \quad k \in V \setminus \{s\} \\ & x_{ij} \geq 0 \quad (i, j) \in E \end{aligned}$$

的线性规划问题为单源最短路问题。



定理 3.17 (模型等价性)

考虑 s 到每点 i 的最短路径, 记 x_{ij} 为边 (i, j) 在这些最短路径中出现的次数, 则其构成单源最短路问题的一个最优解。



证明 由于共有 $|V| - 1$ 条路径, 且除 s 外每点都恰在一条路径的终点上, 其他路径经过时不影响净流出量, x_{ij} 符合约束条件。

下面先说明任何一个单源最短路问题都可以分解为 $|V| - 1$ 个两点最短路问题。对某个点 k_0 , 重复执行如下算法: 找一条 s 到 k 的通路(使路上所有 x_{ij} 上非零的路径), 将路径上的所有 x_e 减小它们的最小值, 直到 $f_x(k) = 0, f_x(s) = |V| - 2$ 。这个过程一定可以结束, 因为网络中的边是有限的, 每次减少一定将至少一条边上的流量减为了 0; 而由于要时刻保持为一个流, $f_x(k_0) < 0$ 时必然还有 s 到 k 的通路。根据算法过程, 网络中比起原来的总减小量满足 $x_{ij} \geq 0$, 且 $f_x(s) = 1, f_x(k_0) = -1$, 这就是一个两点最短路问题。

重复执行此过程, 即证明了单源最短路的任何一个可行解都是对每个点的两点最短路问题的可行解的叠加。从而, 只要证明了两点间的最短路径上每个 x_{ij} 是 1 是两点最短路问题的一个最优解, 即完成了证明。假设最短路径长度为 c_{\min} 。

而对 s 流入 t 流出的两点最短路问题, 仍然可以如上方每次找到通路并减小, 直到网络中 $x_{ij} = 0$ 。假设一共找到 m 条通路, 每条通路长度分别为 c_1, \dots, c_m , 减少的量分别为 x_1, \dots, x_m , 则权重即为 $\sum_i c_i x_i$ 。但由于 $c_i \geq c_{\min}, \sum_i x_i = 1$, 这个权重必然 $\geq c_{\min}$, 于是得证。

不过, 用单纯形的思路进行最短路问题的求解是相对低效的, 因为其并没有运用到最优子结构性

质。假设所有边的长度 c_{ij} 为正，最短路问题有如下的高效算法：

算法 3.18 (Dijkstra 算法)

1. 初始化：令 $d_i = \begin{cases} 0 & i = s \\ \infty & i \neq s \end{cases}$ ，集合 $P = \emptyset$ 。
2. 固定最短路：记 i_0 为 $V \setminus P$ 中使 d_i 最小的 i ， $P = P \cup \{i_0\}$ 。
3. 更新：对所有满足 $(i_0, t) \in E, t \in V \setminus P$ 中的 t 作更新 $d_t = \min\{d_t, d_{i_0} + c_{i_0 t}\}$ 。
4. 终止判定： $P = V$ 时终止，否则回到第二步 (也即二三两步重复 $|V|$ 次)。
5. 结果：终止时的 d 即为 s 到每点的最短路径长度。



证明 我们归纳证明每次迭代中 P 内的最短路径均已经确定。第一步中，到 s 最近的点的最短路径必然是直接到达，于是满足。此后，假设本次更新了 i_0 ，现在的 P 中的点的最短路径已经确定，可以发现第三步更新后 d_t 为路径上所有点都在 P 中的最短路径 (具体来说，更新即为路径上所有点都在 P 中但不经过 i_0 的最短路径与路径上所有点都在 P 中且经过 i_0 的最短路径比较)。

下面考虑现在不在 P 中且 d_t 最小的 t 。若其最短路径长度 $< d_t$ ，则必然经过不在 P 中的点，考虑其上第一个不在 P 中的点 t_0 。由已证， s 到 t_0 的最短路径已经大于等于 d_t ，矛盾。于是到 t 的最短路径长度就是 d_t ，可将 t 放入 P 中，得证。



实际操作时，由于已经证明了在 P 中的都已找到最短路，第三步可以去掉 $t \in V - P$ 的条件。若想知道具体的最短路径，还需要维护数组 π ，当第三步中 d_t 取 $d_{i_0} + c_{i_0 t}$ 时，令 $\pi_t = i_0$ ，类似上方证明过程即可说明，迭代结束后 π_i 代表 s 到 i 最短路径上 i 的前一个结点，不断向前寻找即可。

练习 3.6 举例说明 Dijkstra 算法当有边长度为负时求得的结果未必是最短路径。

解 如图 3.1， s 为源点，第二次迭代确定到 a 的最短路径长度为 0，但事实上为 $s \rightarrow b \rightarrow a$ ，长度 -1。

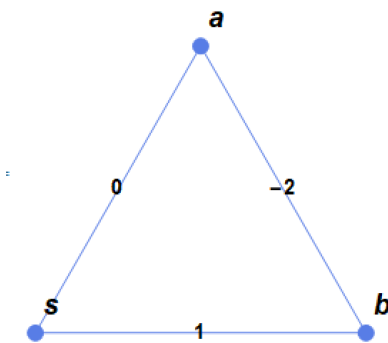


图 3.1: Dijkstra 算法反例

3.2 流的相关问题

3.2.1 最大流问题

网络相关的问题中，一个重要的情况是考虑其中的流：

定义 3.19 (流、流值、可行流)

网络 (G, s, t, u) 中边上的实值函数 $x: E \rightarrow \mathbb{R}$ 称为一个流, 当且仅当其满足 $f_x(v) = 0, v \in V \setminus \{s, t\}$ 。其流值 f_x^* 定义为 $f_x(s)$ 。

若一个流满足 $0 \leq x(e) \leq u(e), \forall e \in E$, 则称其为网络中的可行流。



由净流出量的定义可以有直接的性质: $\sum_{v \in V} f_x(v) = 0$, 这是因为每个 $x(i, j)$ 在 $f_x(i), f_x(j)$ 中分别以正负出现, 互相抵消。于是, 流中必然有 $f_x(s) = -f_x(t)$ 。

最大流问题, 也就是找出流值最大的可行流:

定义 3.20 (最大流问题)

形如

$$\begin{aligned} \max \quad & f_x(s) \\ \text{s. t.} \quad & f_x(k) = 0 \quad k \in V \setminus \{s, t\} \\ & 0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in E \end{aligned}$$

的线性规划问题为最大流问题。



就像 Dijkstra 算法中每步寻找到每个点的距离上界, 为解决最大流问题, 我们也希望有一个合适的方式刻画上界, 并且能通过最优的上界找到解。对最大流问题, 通过“分割”寻找这样的上界:

定义 3.21 (分割、割的容量)

将顶点集 V 划分为两个集合 S, T , 且 $s \in S, t \in T$, 则 (S, T) 称为网络的一个分割, 这个分割的容量定义为 $\sum_{(i,j) \in E, i \in S, j \in T} u_{ij}$ 。



注意 $i \in T, j \in S$ 的边 (i, j) 不参与计算。

直观来看, 由于任何 s 流向 t 的流都必须通过任何割, 割的容量一定大于等于最大流。

练习 3.7 在如图 3.2 所示的网络中计算最大流与每个割的容量。

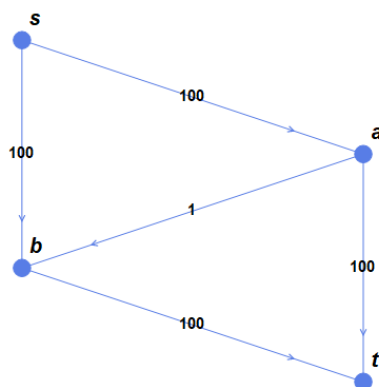


图 3.2: 简单网络

解 最大流为 sa, st, at, bt 各流 100, 流值为 200。割 $\{s, b\} \{a, t\}$ 、 $\{s\}, \{a, b, t\}$ 与 $\{s, a, b\}, \{t\}$ 容量为 200, $\{s, a\} \{b, t\}$ 容量为 201。

刚才的例子中似乎可以发现, 最大流流值与最小的割容量 (到达此容量的割称为**最小割**) 是相同的, 事实上这在一般情况下也成立:

定理 3.22 (最大流最小割定理)

在网络 (G, s, t, u) 中, 最大流值与最小割容量相同。



证明 [此定理证明需要下一节的余网络概念与最大流判定定理。]

先证明任何流的流量不超过任何割的容量。对任何流 x 与割 S, T 直接计算

$$f_x(s) = \sum_{p \in S} f_x(p) = \sum_{(i,j) \in E, i \in S, j \in T} x_{ij} - \sum_{(i,j) \in E, i \in T, j \in S} x_{ij} \leq \sum_{(i,j) \in E, i \in S, j \in T} x_{ij} \leq \sum_{(i,j) \in E, i \in S, j \in T} u_{ij}$$

另一方面, 由于最大流的余网络不存在流扩充路, 可以将余网络中所有存在 s 到其通路的点记为 S , 不存在通路的点记为 T , 则它们构成了一个分割。

而根据不存在通路, 余网络中 S 到 T 没有边, 根据第一种边可知所有 $(i, j) \in E, i \in S, j \in T$ 的 x_{ij} 为 u_{ij} , 根据第二种边可知所有 $(i, j) \in E, i \in S, j \in T$ 的 x_{ij} 为 0, 于是等号可以取到, 又由于小于等于恒成立, 这就说明了最大流与最小割相等。

“最大”与“最小”的对应与对偶问题的形式类似, 事实上有结论:

命题 3.23 (最大割问题的对偶)

最小割问题是最大流问题的对偶问题。



具体论证详见参考资料, 而这结合对偶理论也给出了最大流最小割定理的另一个证明。

3.2.2 余网络

虽然最小割给出了一种有限比较的方法, 但割的数量为 $2^{|V|-2}$ (除 s, t 外每一个可以选择在 S 或 T), 直接进行比较是不现实的。另一个想法是, 从某一个可行流出发 (如初始 $x(e) = 0, \forall e$), 通过有限次更新找到最大流。问题即转化为, 如何更新可以保证流值的上升。

最简单的想法是, 如果找到一条 s 到 t 的路径 p , 且路径上每条边都满足 $x(e) < u(e)$, 可以让路径上的边都增加 $\min_{e \in p} \{u(e) - x(e)\}$, 流值即增加。但是, 即使这样的路径不存在, 也还是有可能增加流值。在图3.2的网络中使 $x_{sa} = x_{bt} = 100, x_{sb} = x_{at} = 99, x_{ab} = 1$, 这时任何路径都至少有一条边流满, 可未达到最大流, 这是因为可以减少 (a, b) 边的容量。由这样的讨论可以得到定义:

定义 3.24 (余网络、流扩充路)

对网络 (G, s, t, u) , 假设其中有一个流 x , 定义其余网络 (G_x, s, t, \bar{u}) 满足:

- G_x 的顶点集 V 、源点 s 、汇点 t 与原网络相同。
- 对每条 $u(e) - x(e) > 0$ 的边 e , 定义所有 e 为 E_1 。
- 对每条 $x(e) > 0$ 的边 $e = (i, j)$, 定义所有 (j, i) 为 E_2 。
- 记 $E = E_1 \cup E_2$, 边的容量 \bar{u} 为
$$\begin{cases} u_{ij} - x_{ij} & (i, j) \in E_1 \\ x_{ij} & (j, i) \in E_2 \end{cases}$$

余网络中 s 到 t 的路径称为流扩充路。



如果将 E_1 中的边看作“增大空间”, E_2 中的边看作“减小空间”, 余网络即为改变空间的刻画 (若 $0 < x(e) < u(e)$, 则 $x(e)$ 既有增大空间又有减小空间, 于是 $e = (i, j)$ 与其逆 (j, i) 都在余网络中)。不过, 为使此定义良好, 需要 E_1, E_2 不交, 而这可以通过边 (i, j) 与 (j, i) 不同时属于 E 来保证, 这也是下文默认考虑的网络, 一般情况将在本节最后陈述。

定理 3.25 (最大流判定)

对某个流 x , 若 G_x 中存在流扩充路 p , 记 $\Delta = \min_{e \in p} \{\bar{u}(e)\}$, 则将 x 变更为 x^{new} , 满足 $x_{ij}^{new} =$

$$\begin{cases} x_{ij} + \Delta & (i, j) \in p \cap E_1 \\ x_{ij} - \Delta & (j, i) \in p \cap E_2, \text{ 有流值 } f_{x^{new}}^* = f_x^* + \Delta. \\ x_{ij} & \text{Otherwise.} \end{cases}$$

反之, 若 G_x 中不存在流扩充路, 则 x 是最大流。



证明 存在流扩充路时, 根据余网络的定义可知这样更新后必然仍有 $x_{ij} \geq 0$ 。对网络中的任何点 k , 分 E_1, E_2 讨论可发现, 余网络中进入 k 的边引起的更新均让 k 的净流出量减少了 Δ , 而余网络离开 k 的边的更新均让其净流出量增加 Δ , 因此更新结束后仍然为流, 流值增加 Δ 。


不存在流扩充路时, 与最大流最小割定理证明相同, 可找到此流与某个割容量一致, 而任何流的流量又小于等于任何割的容量, 于是此流必然为最大流。

利用此定理可得到算法:

算法 3.26 (Ford-Fulkerson 算法)

1. 初始化: 令所有 $x(e) = 0, e \in E, f_{max} = 0$ 。
2. 最优判定: 根据当前 x 构造余网络, 搜索流扩充路, 若无法找到则已达到最优, 输出当前的 x 与 f_{max} , 否则进入第三步。
3. 扩充流: 由上方定理通过流扩充路 p 得到 Δ 并扩充流, $f_{max} = f_{max} + \Delta$, 回到第二步。



 假设边的容量都是整数, 则每次扩充流值至少增加 1, 由于最大流存在, 算法必然会终止。

 **练习 3.8** 在图 3.2 所示的网络中, 利用 Ford-Fulkerson 算法, 最少需要几次扩充? 最多呢?

解 最少按照 sat 、 sbt 扩充, 两次。最多时可能选取的流扩充路是 $sabt$ 、 $sbat$ 不断循环, 扩充 200 次。

从这个例子可以看出, Ford-Fulkerson 算法的效率是与容量相关的, 不超过某个边数、顶点数、容量 (假设容量均为正整数) 构成的多项式, 这样的算法称为弱多项式复杂度。若是不超过某个边数、顶点数构成的多项式, 则称为强多项式复杂度。事实上, 只要每次选取含顶点数最少的流扩充路, 算法复杂度就能达到强多项式级别, 这就是 Edmonds-Karp 算法, 详见参考资料。

对一般情况, 可以不妨考虑 G 中任何两点都有连线的网络 E^+ , 不在 E 中的边 e^+ 有 $u(e^+) = 0$, 于是 $x(e^+)$ 必须为 0, 从而这样增加后与原网络最大流等价。这时, 定义余网络也任意两点都有连线, 且容量为 $\bar{u}_{ij} = u_{ij} - x_{ij} + x_{ji}$, 随后去掉容量为 0 的边。若能找到流扩充路 p , 不妨设其中无重复顶点 (否则去掉此环仍为流扩充路), 假设 $(i, j) \in e$, 并且改变量为 $\Delta = \min_{e \in p} \bar{u}(e)$, 可让 x_{ij} 增加、 x_{ji} 减小, 且使得 $(x_{ij}^{new} - x_{ij}) + (x_{ji} - x_{ji}^{new}) = \Delta$, 由于流扩充路无重复顶点, x_{ij}, x_{ji} 在这次扩充中不会再被调整, 类似前面的证明即说明这样构成流的扩充。

3.2.3 最小成本流

在网络中加入成本, 指定流值, 求最小成本的流, 就是最小成本流问题:

定义 3.27 (最小成本流问题)

形如

$$\begin{aligned}
& \min \quad \sum_{e \in E} c(e)x(e) \\
& \text{s. t.} \quad f_x(s) = -f_x(t) = f^* \\
& \quad \quad f_x(k) = 0 \quad k \in V \setminus \{s, t\} \\
& \quad \quad 0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in E
\end{aligned}$$

的线性规划问题为最小成本流问题。



有时也考虑多源最小成本流:

定义 3.28 (多源最小成本流问题)

形如

$$\begin{aligned}
& \min \quad \sum_{e \in E} c(e)x(e) \\
& \text{s. t.} \quad f_x(v) = b(v) \quad v \in V \\
& \quad \quad 0 \leq x_{ij} \leq u_{ij} \quad (i, j) \in E
\end{aligned}$$

的线性规划问题为多源最小成本流问题。



将 $b(v) > 0$ 的点看成源点, $b(v) < 0$ 的点看成汇点, 即为多源最小成本流名字的由来。此外, 根据之前讨论, 解存在至少需要 $\sum_{v \in V} b(v) = 0$ 。

或是额外增加下限, 考虑最小成本循环流:

定义 3.29 (最小成本循环流问题)

形如

$$\begin{aligned}
& \min \quad \sum_{e \in E} c(e)x(e) \\
& \text{s. t.} \quad f_x(v) = 0 \quad v \in V \\
& \quad \quad l_{ij} \leq x_{ij} \leq u_{ij} \quad (i, j) \in E
\end{aligned}$$

的线性规划问题为最小成本循环流问题。



前述三个问题, 若上界 u_{ij} 全部为无穷, 则称为无上界的对应问题。后两者由于对称性, 可以用来模型化更多的场景, 不过事实上有:

定理 3.30 (模型等价性)

最小成本流、多源最小成本流、无上界多源最小成本流、最小成本循环流可以互相进行等价转化。

**证明** 分别记四个问题为 1 到 4。 $2 \Rightarrow 1, 3$: 由定义即知 1、3 都为 2 的特殊情况。 $1 \Rightarrow 2$: 添加 s 与 t , s 到所有 $b(v) > 0$ 的 v 有边, 对应的 $u_{sv} = b(v), c_{sv} = 0$, 所有 $b(v) < 0$ 的 v

到 t 有边, 对应的 $u_{vt} = -b(v), c_{vt} = 0$, 且要求

$$f^* = \sum_{v|b(v)>0} b(v)$$

由于这样恰好保证了所有原网络中的点各自有了 $-b(v)$ 的净流出量, 且不影响总成本。为了平衡, 原网络中即各自需要 $b(v)$ 的流出量, 从而模型与多源最小成本流等价。

$3 \Rightarrow 2$: 对多源最小成本流的每条边 (i, j) , 假设上界 u , 成本 c 。去掉边 (i, j) 并添加一个点 v_{ij} , 连接 (i, v_{ij}) 与 (j, v_{ij}) 。要求 v_{ij} 净流入量为 u , (i, v_{ij}) 的成本为 c , (j, v_{ij}) 的成本为 0 , 并给 j 的净流出量增加 u 。

我们假设新的无上界多源最小成本流问题里 (i, v_{ij}) 的流量为 x_{ij} , 下面说明 x_{ij} 构成原网络中的可行流。首先, 由于 $x_{i,v_{ij}} \geq 0, x_{j,v_{ij}} \geq 0$, 可以得到 $x_{ij} \geq 0, u - x_{ij} \geq 0$, 从而 x_{ij} 满足界的要求。此外, 由于给 j 增加了净流出量 u , 而 j 在新网络中流出 $u - x_{ij}$, 事实上对应原网络恰好为 $-x_{ij}$ 的净流出量, 与一条 i 到 j 上流量 x_{ij} 的边完全等效, 从而得证。反过来, 若 x_{ij} 构成原网络可行流, 完全类似可说明这样构造后成为新网络可行流, 且由定义总成本相同, 从而得证。

$2 \Rightarrow 4$: 记 l 为最小成本循环流每条边流量 $x_{ij} = l_{ij}$ 的网络, 并记此时每个点的净流出量为 $l(v)$ 。则直接考虑每条边成本不变, 对应上界变为 $u_{ij} - l_{ij}$, 顶点净流出量为 $-l(v)$ 的多源最小成本流问题, 根据定义即可知其可行流加上 l 就变为原问题可行流, 且总成本恒相差 $\sum_e l(e)c(e)$ 。

$4 \Rightarrow 1$: 对最小成本流问题添加一条 t 到 s 的边, 要求 $l_{ts} = u_{ts} = f^*, c_{ts} = 0$, 其他边范围与成本均不变, 可看出此时其他边的情况与最小成本流完全相同, 总成本亦相同, 从而得证。

关于它们的模型化能力, 一个简单的例子是:

定理 3.31 (最小成本流的模型化)

最短路问题、最大流问题可以等价转化成最小成本流问题。



证明 最短路: 单源最短路问题完全符合无上界多源最小成本流问题的定义, 根据上个定理得证。

最大流: 假设最大流问题原网络上界不变, 成本均为 0 , 添加一条 s 到 t 的边 e_0 , 成本为 1 , 上界不限, 且要求 f^* 为充分大的 M , 如取网络中所有边的 u_{ij} 之和。这样, 原网络中的流越大, 经过 e_0 的流就越小, 从而成本越低, 即得证。此最小成本流与原问题等价。

从余网络中, 每次通过最短路径算法寻找成本最小的流扩充路扩充, 直到到达需要的 f^* , 就是最小成本流问题的 Ford-Fulkerson 算法。遗憾的是, 由于这次需要选取特定的路径, 无法简单调整成为强多项式级别算法, 于是关于最小成本流问题的强多项式算法一般较为复杂。

3.3 动态规划

3.3.1 基本要素

在本章的最后, 我们来看一个常见的主要用于离散优化的多项式级别算法思路, 动态规划。

在之前最短路径问题的讨论中, 我们强调了它具有最优子结构性质, 且由此结构可知到每点的最短路径长度 d 满足递推式 $d(v_0) = \min_{(v,v_0) \in E} \{d(v) + c(v, v_0)\}$, 于是通过一步步构造迭代, 最终得到了 Dijkstra 算法。此外, 通过记录 \min 中的选择, 我们不仅可以得到最短路径的长度, 还可以得到最短的路径。

从这里, 我们似乎找到了一个解决问题的模式:

1. 假设找到最优解，刻画问题的最优子结构性质。
2. 找到递推方程，通过递推方程定义最优解的值。
3. 从最简单的子问题开始，自底向上计算最优解的值。
4. 通过计算过程中的信息，(递归地) 构造出最优解。

而这就是动态规划的基本思路。

为了这个思路能够实现，问题必须满足一些性质，其中最基本的便是能找到最优子结构与递推方程，且递推方程中通过已确定的子问题能得到新的子问题。递推方程相关的这个性质被称为子问题覆盖，于是最优子结构与子问题覆盖即为动态规划问题需要满足的基本要素。

一般来说，动态规划问题所需要比较的方案个数都是问题规模的指数量级，而采用动态规划法后，实际解决只需要多项式量级的时间复杂度与空间复杂度(空间复杂度来源于保存子问题的解)，本质原因在于，每求解一个子问题，就否定了大量情况，类似在方案树中进行剪枝的操作。下面，我们来看两个具体的例子：

3.3.2 背包问题

0-1 背包问题的基本描述是，假设有 n 件物品，每件质量为 w_i ，价值为 r_i ，求将其中某些(物品不可拆分)装进最大质量为 W 的背包的最高价值方案。将其数学化即为：

定义 3.32 (背包问题)

形如

$$\begin{aligned} \min \quad & \sum_{i=1}^n r_i \chi_i \\ \text{s. t.} \quad & \sum_{i=1}^n w_i \chi_i \leq W \\ & \chi_i \in \{0, 1\} \quad i = 1, \dots, n \end{aligned}$$

的离散优化问题为背包问题，一般 $r_i > 0, w_i > 0, i = 1, \dots, n$ 。



这里 $\chi_i = 1$ 即代表放入背包， $\chi_i = 0$ 即代表不放。

下面用动态规划的四个步骤进行求解。

1. 最优子结构刻画

假设已经找到了最优的装载方案 χ^* ，考察此方案所具有的性质。注意到，假设 $\chi_i^* = 1$ ，也即最优方案装了第 i 件物品，那么 χ^* 剩下的部分必然是背包容量为 $W - w_i$ 时在去掉第 i 件物品情况下的最优装载方案。

2. 得到递推方程

从上方的最优子结构进一步进行研究。对 χ_n^* 进行分类讨论：若其为 1，则 $\chi_1^*, \dots, \chi_{n-1}^*$ 是容量为 $W - w_n$ 的背包装载前 $n-1$ 件物品的最优方案。否则， $\chi_1^*, \dots, \chi_{n-1}^*$ 是容量为 W 的背包装载前 $n-1$ 件物品的最优方案。于是，只需要比较两种选取方法的总价值，就能得到总的最优方案。

类似可得, 记 $f_i(w)$ 为容量 w 的背包装载前 i 件物品的最优方案, 即有

$$f_i(w) = \begin{cases} \max\{f_{i-1}(w), f_{i-1}(w - w_i) + c_i\} & w \geq w_i, i > 0 \\ f_{i-1}(w) & w < w_i, i > 0 \\ 0 & i = 0 \end{cases}$$

除了最上面是之前讨论过的, 下面两种都是边界情况: 无法选取第 i 件与不进行选取。

3. 自底向上计算

在递推方程中, 最后需要求解的是 $f_n(W)$, 且计算 $f_i(w)$ 只需要用到 $i_0 < i, w_0 \leq w$ 的 $f_{i_0}(w_0)$ 。因此, 构造 $(n+1) \times W$ 的数组 f , $i = 0, \dots, n$, $w = 1, \dots, W$, 初始化后按照先增加 w 后增加 i 的顺序 (即先计算所有 $f_0(w)$, 再计算所有 $f_1(w)$...) 即能计算出最优解的值。

4. 递归构造最优解

递归公式中的 \max 等于 $f_{i-1}(w - w_i) + c_i$ 即为选择第 i 件, 否则为不选, 因此只需要额外构造数组记录每次递推时的选择, 然后从 $i = n, w = W$ 开始, 选取了第 i 件就令 $\chi_i = 1$ 并将 W 减少 w_i 、 i 减少 1, 否则令 $\chi_i = 0$ 、 i 减少 1, 直到 i 为 0。根据递推过程, 这样得到的 χ 一定是最优解。

由于更新时是二选一, 常数量级, 这个方法的时空复杂度都是 $O(nW)$ 。0-1 背包问题的一个经典推广是多重背包:

定义 3.33 (多重背包问题)

形如

$$\begin{aligned} \min \quad & \sum_{i=1}^n r_i \chi_i \\ \text{s. t.} \quad & \sum_{i=1}^n w_i \chi_i \leq W \\ & \chi_i \in \{0, 1, \dots, m_i\} \quad i = 1, \dots, n \end{aligned}$$

的离散优化问题为多重背包问题, 一般 $r_i > 0, w_i > 0, i = 1, \dots, n$ 。



这时 n 相当于物品种数, 每种物品至多 m_i 件。

对多重背包问题, 可以将每件物品分开当作 0-1 背包, 但存在更优化的算法。类似最大流问题, 我们希望找到复杂度与物品件数无关的“强多项式时间复杂度算法”, 详见参考资料。

3.3.3 设备更新问题

设备更新问题的背景是, 假设现在有一台使用了 t 年的设备, 每年初需要决定设备是否更新, 令 r_i, c_i, s_i 表示已使用了 i 年的设备在这一年的年营业收入、年运营成本、年初残值 (更新设备时以残值的价格卖掉旧的设备), I_j 表示第 j 年初购买新设备的价格, 使用了 T 年的设备在下一年初必须卖出。假设第 n 年末卖出设备, 求使前 n 年总收入最高的第 1 到第 $n-1$ 年每年初是否卖出的方案。

设备更新问题事实上可以看作最短路问题: 在平面上, 点 (a, b) 表示在决定是否更换前第 a 年初持有一台已使用了 b 年的设备, 则考虑所有的 $(a, b), a = 1, \dots, n, b = 1, \dots, T$, 其中 $(i, j), j < T$ 到 $(i+1, j+1)$ 连有有向边, 边长是这一年继续使用的收入; (i, j) 到 $(i+1, 1)$ 连有有向边, 边长是这一年卖掉旧设备的收入 (由于是决定前的使用年龄, 决定后到下一年初时新设备已经使用了一年); 所有 (i, n) 到终点 end 连线, 边长是卖掉设备的残值 (因为第 n 年末不再购买运行)。如图 3.3, 希望找到起点

$(1, t)$ 到终点的最大价值方案，也就是每条边的边长取相反数 (也就是净支出) 后的最短路径。

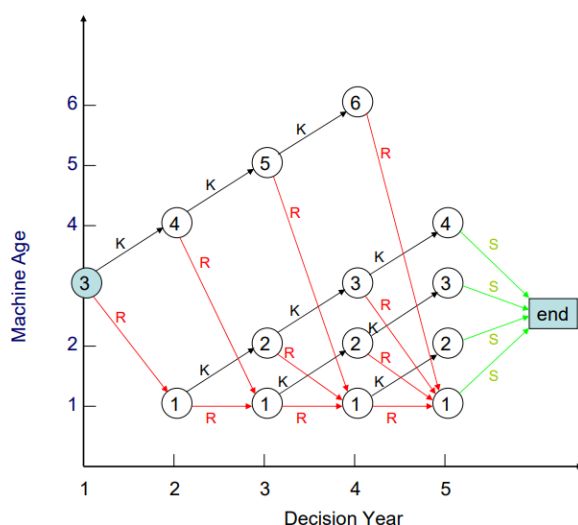


图 3.3: 设备更新问题的最短路径表示



一般情况下，最长路径问题不能直接通过取相反数化为最短路径问题，这里可以是由于结点按照 i 分组，每次一定从一组的某个点走向下一组的某个点。此类最短路径问题可以用 Viterbi 算法快速求解，这也是一个经典的动态规划算法。

不过，由于问题的特殊性，我们事实上没有必要利用最短路径的方法求解，不过，最短路径可以提供算法的思路。注意到，这里每个结点虽然可能有多条进入的边，但离开的边最多只有两条，因此可以考虑从右向左进行求解。接下来给出完整过程：

1. 最优子结构与递推方程

记 $f_i(a)$ 为第 i 年初有使用了 a 年的设备时此后的最大收入。假设最优方案这一年选择了保留，则 $f_{i+1}(a+1)$ 加上第 a 年的收入 $r_a - c_a$ 即为 $f_i(a)$ ；否则， $f_{i+1}(1)$ 加上第 a 年收入 $r_0 - c_0 + s_a - I_a$ 即为 $f_i(a)$ 。最后一年末直接卖出，因此可以认为 $f_{n+1}(a) = s_a$ ，从而得到递推方程：

$$f_i(a) = \begin{cases} \max\{f_{i+1}(a+1) + r_a - c_a, f_{i+1}(1) + r_0 - c_0 + s_a - I_i\} & i \leq n, a < T \\ f_{i+1}(1) + r_0 - c_0 + s_a - I_i & a = T \\ s_a & i = n + 1 \end{cases}$$

2. 自底向上计算

在递推方程中，最后需要求解的是 $f_1(t)$ ，且计算 $f_i(a)$ 只需要用到 $i_0 > i$ 的 $f_{i_0}(a_0)$ ，因此，构造 $(n+1) \times T$ 的数组 f ， $i = 1, \dots, n+1$ ， $a = 1, \dots, T$ ，初始化后按照先减少 a 后减少 i 的顺序即能计算出最优解的值。

3. 递归构造最优解

递归公式中的 \max 等于 $f_{i+1}(a+1) + r_a - c_a$ 即为不更新设备，否则为更新，因此只需要额外构造数组记录每次递推时的选择，然后从 $i = 0, a = t$ 开始，根据是否更新计算新的 i 与 a ，直到 $i = n + 1$ 。根据递推过程，这样一定可以得到最优方案。

类似背包问题，这个方法的时空复杂度都是 $O(nT)$ 。虽然动态规划本质是求解离散的问题，不过这样的思路对连续问题也有意义，例如，线性规划的连续问题事实上可以化为选取可行基的离散问题，于是在一些特殊情况也可以利用动态规划优化 (事实上最短路径问题就是这样的情况)。

第4章 无约束最优化

内容提要

- 梯度类方法基本模式
- 非精确一维搜索
- 一维搜索收敛性
- 梯度法
- 共轭梯度法
- 牛顿法
- 拟牛顿法
- 信赖域方法

4.1 一维搜索

4.1.1 梯度类方法

本章中，我们将介绍无约束最优化问题 $\min_{x \in \mathbb{R}^n} f(x)$ 的几种基本求解方法，并比较它们的效果。回顾绪论时提到的，优化问题的一个经典算法是迭代下降算法，每次迭代经历确定方向与确定步长两步。对无约束最优化问题，根据梯度的几何意义，下降速度最快的方向就是负梯度的方向，于是，两步法的迭代下降可以细化为：

算法 4.1 (梯度类方法)

1. 给定初始点 x_0 ，令 $k = 0$ 。
2. 取某正定对称阵 H_k ，并令 $d_k = -H_k \nabla f(x_k)$ 。
3. 解一维最优化问题 $\min_{\alpha \geq 0} f(x_k + \alpha d_k)$ ，得到合适的步长 α_k 。
4. 更新 $x_{k+1} = x_k + \alpha_k d_k$ ， $k = k + 1$ ，若满足终止判定 (如 $\|\nabla f(x_{k+1})\| < \varepsilon$) 则结束计算，否则回到第二步。

证明 这时 $\nabla f(x_k)^T d_k = -\nabla f(x_k)^T H_k \nabla f(x_k)$ ，由正定定义可知 $\nabla f(x_k)$ 非零时一定是下降方向，而若其为 0，第四步中已经判定收敛。由绪论中几何必要条件的证明，下降方向上只要 α 足够小，总能使 $f(x_k + \alpha d_k) < f(x_k)$ ，从而每次迭代函数值单调下降，趋于负无穷或收敛。

之后的部分中，我们会更多聚焦在方向的选取，而这一节的核心是一维最优化问题 $\min_{\alpha \geq 0} \varphi(\alpha) = f(x_k + \alpha d_k)$ ，求解这个问题的过程称为**一维搜索**。若是希望得到尽量精确的最优解，求解过程就称为**精确一维搜索** [Exact Line Search]，有黄金分割法、插值迭代法等方法。不过，即使说是精确一维搜索，通过有限次计算求出精确解一般也是不可能的，实际上是在以有足够精度的近似解作为步长。反之，若是找出满足某些适当条件的粗略近似解作为步长，就称为**非精确一维搜索** [Inexact Line Search]。由于非精确一维搜索的整体计算效率一般更高，我们主要介绍非精确一维搜索的经典方法。

4.1.2 Wolfe-Powell 准则

本节中，记 $\varphi(\alpha) = f(x + \alpha d)$ ，其中 x, d 是给定的起始点与搜索方向。在之前证明中也提到，由于梯度刻画的是局部性质，一般不具有整体意义，一维搜索 $\min_{\alpha \geq 0} \varphi(\alpha)$ 也常常会限定在可行范围内，如取 $\bar{\alpha}$ 为使得 $\varphi(\alpha) = \varphi(0)$ 的最小 α (假设存在)，然后在 $[0, \bar{\alpha}]$ 内寻找可接受的 α 。

下图即为初始区间 $[0, \bar{\alpha}]$ 的示意图, 由于 $\varphi'(t) = \nabla f(x + td)^T d$, 根据梯度法的要求可知 $\varphi'(0) < 0$, 于是 0 处一定下降。

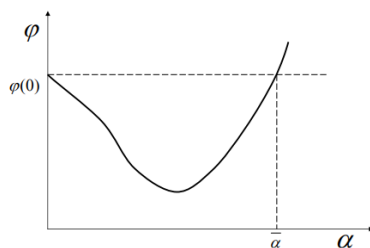


图 4.1: $[0, \bar{\alpha}]$ 区间示意

为了完成非精确一维搜索, 我们需要刻画怎样的 α 是可接受的, 下面给出两个例子:

定义 4.2 (Goldstein 准则)

Goldstein 条件定义符合要求的 α 为:

$$\begin{cases} \varphi(\alpha) \leq \varphi(0) + \rho\alpha\varphi'(0) \\ \varphi(\alpha) \geq \varphi(0) + (1 - \rho)\alpha\varphi'(0) \end{cases}$$

其中 $\rho \in (0, \frac{1}{2})$ 为事先给定的参数。



定义 4.3 (Wolfe-Powell 准则)

Wolfe-Powell 条件定义符合要求的 α 为:

$$\begin{cases} \varphi(\alpha) \leq \varphi(0) + \rho\alpha\varphi'(0) \\ \varphi'(\alpha) \geq \sigma\varphi'(0) \end{cases}$$

其中 $\rho \in (0, \frac{1}{2}), \sigma \in (\rho, 1)$ 为事先给定的参数。



在实际算法中, 第二个条件常被强化为 $|\varphi'(\alpha)| \leq -\sigma\varphi'(0)$ 。

这是在图4.1所示函数中两个准则找到的区间对比, 红蓝线之间为可接受区间:

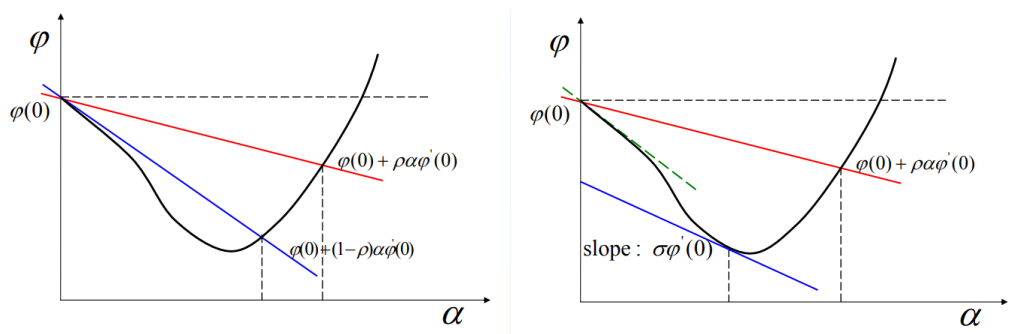


图 4.2: 可接受区间对比

由于 0 处的切线为 $\varphi(0) + t\varphi'(0)$, Goldstein 准则事实上以切线降低不同程度斜率作了两条割线, 并取割线之间的部分, 不过, 正如图中展示的, 它甚至不能保证最优解在可接受区间中。相比起来, Wolfe-Powell 准则的第一个条件与 Goldstein 相同, 限定了最远区域, 而第二个条件对斜率的限制 (注意由于 $\varphi'(0) < 0$, 事实上有 α 或在上升段, 或在下降段接近平缓的部分) 则保证了它至少能到达接近最优的位置。

有 Wolfe-Powell 准则后，我们需要一个能找到符合要求的点的算法。基本的思想是二分：每次从区间中取一个点，并根据结果确定其是否符合要求，若否更新其为左或右端点，继续选取。不过，比起直接选择区间中点，我们有更好的选取方式，也即以某种意义将其近似成二次函数，并寻找二次函数的最小值点。由此得到算法：

算法 4.4 (基于 W-P 准则的非精确一维搜索)

1. 给定参数 $\rho \in (0, \frac{1}{2})$, $\sigma \in (\rho, 1)$ ，初始左端点 $a_1 = 0$ 、初始右端点 $a_2 = \bar{\alpha}$ 与初始搜索点 (如区间中点) $\alpha \in [0, \bar{\alpha}]$ 。计算 $\varphi_0 = \varphi(0)$ 、 $\varphi'_0 = \varphi'(0)$ 、 $\varphi_1 = \varphi_0$ 、 $\varphi'_1 = \varphi'_0$ 。
2. 计算 $\varphi = \varphi(\alpha)$ ，若 $\varphi \leq \varphi_0 + \rho\alpha\varphi'_0$ ，直接进入第四步，否则代表 α 过大，进入第三步。
3. 构造二次函数 h 满足 $h(a_1) = \varphi_1$, $h'(a_1) = \varphi'_1$, $h(\alpha) = \varphi$ ，得到其最小值点

$$\hat{\alpha} = a_1 + \frac{1}{2} \frac{(a_1 - \alpha)^2 \varphi'_1}{(\varphi_1 - \varphi) - (a_1 - \alpha) \varphi'_1}$$

并更新 $a_2 = \alpha$, $\alpha = \hat{\alpha}$ ，回到第二步。

4. 计算 $\varphi' = \varphi'(\alpha)$ ，若 $\varphi' \geq \sigma\varphi'_0$ ，则算法终止， α 已符合要求，否则代表 α 过小，进入第五步。
5. 构造二次函数 g 满足 $g(\alpha) = \varphi$, $g'(\alpha) = \varphi'$, $g'(a_1) = \varphi'_1$ ，得到其最小值点

$$\hat{\alpha} = \alpha + \frac{(\alpha - a_1) \varphi'}{\varphi'_1 - \varphi'}$$

并更新 $a_1 = \alpha$, $\alpha = \hat{\alpha}$, $\varphi_1 = \varphi$, $\varphi'_1 = \varphi'$ ，回到第二步。



由于我们已经计算得到了 $\hat{\alpha}$ 的表达式，实际进行算法时无需显式构造 h, g ，这里写出只是为了表明表达式的来源。此外，知道 f 时也无需显式构造出 φ ，算法中的 $\varphi(\alpha)$ 用 $f(x + \alpha d)$ 计算，而 $\varphi'(\alpha)$ 实际上是 $\nabla f(x + \alpha d)^T d$ 。

由于算法的二分策略，一般情况下可以收敛到符合要求的点，即使该方向上 φ 不具有凸性。

4.1.3 全局收敛性

从之前的对比图中可以直观看出，非精确一维搜索的结果可能离精确解距离甚远，因此我们必须说明它的有效性，对下降算法而言，也就是不影响收敛。无约束最优化中的收敛性可以定义为：

定义 4.5 (聚点、驻点、全局收敛性、局部收敛性)

一个点列的聚点指所有有极限的子列的极限点，一个可微函数的驻点是指梯度为 0 的点。

从任意初始点出发，如果某迭代算法产生的点列的极限 (或聚点)，在适当假定下可保证恒为问题的最优解 (或驻点)，则称该迭代法具有全局收敛性 [global convergence]。

与此相对，如果仅在解的附近选取初始点时，才可以保证所生成的点列收敛于该解，则称这样的迭代法有局部收敛性 [local convergence]。



为了说明全局收敛性，我们还需要要求梯度类方法的搜索方向满足一个额外条件。类似二维与三维，在 n 维空间中，两个向量 a, b 夹角定义为 $\theta = \arccos \frac{a^T b}{\|a\| \|b\|}$ 。利用此， d_k 是下降方向也即要求 d_k 与 $-\nabla f(x_k)$ 的夹角 θ_k 小于 $\frac{\pi}{2}$ 。但是，当 θ_k 很接近 $\frac{\pi}{2}$ 时，这个搜索可能是无意义的，因此我们要求存在 $\mu \in (0, \frac{\pi}{2})$ 使得每次下降方向与负梯度方向夹角 $\theta_k \leq \frac{\pi}{2} - \mu, \forall k$ 。



记 $g_k = \nabla f(x_k)$, $s_k = x_{k+1} - x_k = \alpha_k d_k$ ，实际计算夹角常用 $\cos \theta_k = \frac{-g_k^T s_k}{\|g_k\| \|s_k\|}$ ，这样就有 $\cos \theta_k \geq$

$\cos(\frac{\pi}{2} - \mu) = \sin \mu$, 也即 $\cos \theta_k$ 不小于某正数 $\sin \mu$ 。

加入此条件后, 即可得到各种步长准则下的下降算法全局收敛性:

定理 4.6 (全局收敛性定理)

设 $\nabla f(x)$ 在 $\{x \mid f(x) \leq f(x_0)\}$ 上存在且一致连续。下降算法从 x_0 开始, 每步的搜索方向为 d_k , 其与 $-\nabla f(x_k)$ 的夹角 θ_k 满足 $\exists \mu \in (0, \frac{\pi}{2}), \forall k, \theta_k \leq \frac{\pi}{2} - \mu$, 且步长 α_k 由以下两者之一确定:

- Goldstein 准则
- Wolfe-Powell 准则

每次更新 $x_{k+1} = x_k + \alpha_k d_k$, 那么必出现以下三种情况之一:

- 对某个 k 有 $\nabla f(x_k) = 0$
- $\lim_{k \rightarrow \infty} \nabla f(x_k) = 0$
- $\lim_{k \rightarrow \infty} f(x_k) = -\infty$



证明 由于初始搜索区间是 $[0, \bar{\alpha}]$, 而 $\bar{\alpha}$ 是满足 $\varphi(0) = \varphi(\alpha)$ 的最小 α , 无论哪种搜索方式, $f(x_k)$ 一定单调减小。假设对所有 k , $\nabla f(x_k) \neq 0$, 且 $f(x_k)$ 有下界, 这时 $f(x_k)$ 必有极限, 下面对三种准则下的搜索证明 $\lim_{k \rightarrow \infty} \nabla f(x_k) = 0$ 。以下记 $g_k = \nabla f(x_k)$, $s_k = x_{k+1} - x_k = \alpha_k d_k$:

对两种搜索方式, $f(x_k)$ 都是严格下降的, 且由有下界, 必须有 $f(x_k) - f(x_{k+1}) \rightarrow 0$ 。由于它们的第一条要求是共同的, 根据 $f(x_k) - f(x_{k+1})$ 的收敛性可以算出 $g_k^T s_k \rightarrow 0$ 。若 $\lim_{k \rightarrow \infty} g_k = 0$ 不成立, 必有子列使得其模长恒大于 ε , 我们记此子列为 $\{g_i\}$ 。根据 θ_k 的范围必有

$$-g_k^T s_k = \|g_k\| \|s_k\| \cos \theta_k \geq \varepsilon \|s_k\| \sin \mu$$

于是即得 $\|s_k\| \rightarrow 0$ 。

对 Goldstein 准则的第二个条件简单变形 (注意由递推关系, α 为正, $\varphi'(0)$ 为负) 得到

$$\frac{\varphi(\alpha) - \varphi(0)}{\alpha} \geq (1 - \rho) \varphi'(0)$$

利用中值定理知 $\exists \xi \in [0, \alpha], \varphi'(\xi) \geq (1 - \rho) \varphi'(0)$ 。由于上方过程中得到了 $\|s^{(k)}\| \rightarrow 0$, 由梯度一致连续知当 $k \rightarrow \infty$ 时

$$\frac{\varphi'(\alpha)}{\varphi'(0)} = \frac{g_{k+1}^T s_k}{g_k^T s_k} \leq 1 - \rho$$

这与 Wolfe-Powell 条件第二个条件形式相同 (对应参数为 σ)。

另一方面, 由梯度一致连续性, 记 $h(t) = \nabla f(x + t)^T t$, 在 g_k 处展开可知 (令 $t = s_k$)

$$g_{k+1}^T s_k = g_k^T s_k + O(\|s_k\|^2)$$

于是 $\lim_{k \rightarrow \infty} \frac{g_{k+1}^T s_k}{g_k^T s_k} = 1$, 矛盾。



对精确一维搜索反而没有这样容易证明的收敛性, 这是由于精确一维搜索无法从 g_k 极限非零直接推出 $\|s_k\| \rightarrow 0$, 也即无法保证每步步长能被控制。若给精确一维搜索增添两准则的第一条要求, 由于它符合 Wolfe-Powell 准则的第二条要求, 就可直接得到结果。

由此, 我们已经确定了可行的一维搜索方式, 接下来注重的核心便是下降方向的选取。下面的部分以算法介绍为主, 一些跳过的收敛速率分析等详见参考资料。



对无约束最优化, 一个经典的收敛性、收敛速率分析的非凸测试函数是 Rosenbrock 函数, 也被称为

“香蕉函数”，表达式如下：

$$f(x) = \sum_{i=1}^{N-1} ((1-x_i)^2 + 100(x_i^2 - x_{i+1})^2)$$

它有唯一全局最小值点，也即各分量均取 1 时为 0，但并不容易找到。二维时，它的每个等高线大致呈抛物线形（香蕉形），因此得名。¹

4.2 梯度方法

4.2.1 最速下降法

由于 $f(x)$ 的一阶泰勒展开是 $f(x_0 + t) = f(x_0) + \nabla f(x_0)^T t + O(\|t\|^2)$ ，负梯度方向是下降最快的方向，因此最基本的下降方法就是直接选取负梯度方向进行下降，也称为最速下降法。标准的迭代格式是：

算法 4.7 (最速下降法)

1. 给定初始点 x_0 ，令 $k = 0$ ，给定终止误差 $\varepsilon > 0$ 。
2. 计算 $g_k = \nabla f(x_k)$ ，若 $\|g_k\| < \varepsilon$ ，则算法终止，输出 x_k ，否则进入下一步。
3. 解一维最优化问题 $\min_{\alpha \geq 0} f(x_k - \alpha g_k)$ ，得到合适的步长 α_k 。
4. 更新 $x_{k+1} = x_k + \alpha_k d_k$ ， $k = k + 1$ ，回到第二步。

最速下降法的性质如下 (回顾绪论对收敛性与收敛速率的定义)：

命题 4.8 (最速下降法-全局收敛性与收敛速率)

在 f 处处可微时，最速下降法利用精确或非精确一维搜索，无论初值如何选取产生点列的每一个聚点都是驻点。在收敛时，一般情况下最速下降法的收敛速率是线性的。

4.2.2 共轭方向

最速下降法的缺点很明显：由于可能重复选取方向，产生类似折线的下降路径，哪怕对很好的函数 (如有唯一最小值的二次函数)，收敛效率也是相对低的。为了规避这样的问题，我们希望在选取下降方向时尽量选取“相对无关”的方向，这就产生了共轭方向的概念：

定义 4.9 (共轭方向)

设 G 是 $n \times n$ 正定对称阵， \mathbb{R}^n 中两个非零向量 a, b 若满足 $a^T G b = 0$ ，则称它们 G -共轭。若一组非零向量两两 G -共轭，则称这组向量 G -共轭 (为方便，若这组中只有一个向量，也认为 G -共轭)。



共轭概念是正交的推广，当 $G = I$ 时，可发现共轭即为正交。

定理 4.10 (共轭方向性质)

一组 G -共轭的方向必然线性无关。

¹事实上，本讲义的封面即为 Rosenbrock 函数的改版 $\sqrt{(1-x)^2 + (y-x^2)^2}$ 。

证明 若否, 假设 $\sum_i \lambda_i \alpha_i = 0$, α_i 为 G -共轭, 则有

$$\left(\sum_i \lambda_i \alpha_i \right)^T G \left(\sum_i \lambda_i \alpha_i \right) = \sum_i \lambda_i^2 \alpha_i^T G \alpha_i + \sum_{i \neq j} \lambda_i \lambda_j \alpha_i^T G \alpha_j = 0$$

根据共轭性质, 后一项为 0, 而由正定性, 前一项每个 $\alpha_i^T G \alpha_i > 0$, 于是必须所有 $\lambda_i = 0$, 得证。

共轭方向类方法的思路是, 每次选取的方向 d_k 都相互共轭的下降方向时, 可以有较好的下降效果。当然, 这件事无法做到一直成立, 所以我们需要先讨论共轭方向法的产生来源: 二次函数的极小化问题, 再看它如何推广到非二次函数的极小化问题。

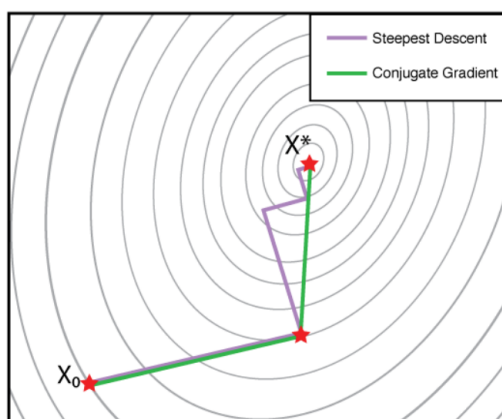


图 4.3: 二次函数的最速下降法、共轭梯度法对比

对于二次函数, 有如下基本定理:

定理 4.11 (共轭方向法基本定理)

对于二次函数 $f(x) = \frac{1}{2}x^T Gx + c^T x$, 若其严格凸 (即 G 正定对称), 假设搜索方向 d_0, \dots, d_k 是 x_0, \dots, x_k 处的下降方向, 且这些方向 G -共轭, 并且执行的是精确一维搜索, 即 $f(x_{i+1}) = \min_t f(x_i + td_i), i = 0, \dots, k$, 那么有

$$f(x_{k+1}) = \min_{x \in V} f(x), V = \left\{ x \mid x = x_0 + \sum_{j=0}^k \beta_j d_j, \forall \beta_i \in \mathbb{R} \right\}$$

并且是唯一极小点。



证明 首先, 根据一维精确搜索性质, 搜索后在当前方向的最小值点, 因此考察 $f(x + \alpha d_j)$ 对 α 的导数可知必然有 $g_{k+1}^T d_k = 0$ 对任何 j 成立。

又由于 d_j 相互共轭, 根据 $g_{k+1} - g_k = \alpha_k G d_k$, 对任何 $j < k$ 有

$$g_{k+1}^T d_j = g_{j+1}^T d_j + \sum_{i=j+1}^k \alpha_i d_i^T G d_j = 0$$

这也就说明每次的梯度与之前所有搜索方向垂直, 考察

$$f\left(x_0 + \sum_{j=0}^k \beta_j d_j\right)$$

将其看作对每个 β_j 的函数, 则问题变为无约束问题, 且可验证对 β 而言其为凸函数, 从而驻点是唯一极小点。

由 g_{k+1} 与 d_1, \dots, d_k 垂直可计算出 x_{k+1} 为驻点, 从而得证。



由于 $x^T G x = (x^T G x)^T = x^T G^T x$, G 不对称时可以取 $G' = \frac{G+G^T}{2}$, 不影响函数, 于是可不妨设 G 对称。此外, 常数项对优化无意义, 于是可不妨设无常数项。

4.2.3 共轭梯度法

从这个定理与共轭方向的性质可以直接推出, 对 \mathbb{R}^n 中的严格凸二次函数, 至多进行 n 次精确一维搜索就能找到全空间的最小值点。接下来的问题是, 如何找到这样一组相互共轭的下降方向。这就引申出了修改最速下降法得到的共轭梯度法。下面先给出对二次函数的算法, 再进行说明:

算法 4.12 (共轭梯度法-二次函数)

1. 给定严格凸二次函数 $f(x) = \frac{1}{2}x^T G x + c^T x$, 其梯度向量 $g(x) = Gx + c$, 给定初始点 x_0 , 记 $k = 0, g_0 = g(x_0), d_0 = -g_0$ 。
2. 计算 $x_{k+1} = x_k + \alpha_k d_k$, 其中 $\alpha_k = \frac{g_k^T g_k}{d_k^T G d_k}$ 。
3. 计算 $g_{k+1} = g(x_{k+1})$, 若为 0 则停止, 否则进入下一步。
4. 计算方向 $d_{k+1} = -g_{k+1} + \beta_k d_k$, 其中 $\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$, 回到第二步。

证明 为了能利用共轭方向法基本定理说明收敛, 事实上需要说明两件事: 这样的 α_k 满足精确一维搜索, 且这样得到的所有 d_k 共轭。对 α_k , 由于 $\varphi'_k(\alpha) = \nabla f(x_k + \alpha d_k)^T d_k = (G(x_k + \alpha d_k) + c)^T d_k$, 从而 $\varphi'_k(\alpha) = 0$ 也即 $d_k^T G x_k + \alpha d_k^T G d_k + d_k^T c = 0$ 。由于 $g_k = G x_k + c$, 可进一步写出 $\alpha_k = \frac{-d_k^T g_k}{d_k^T G d_k}$ 。利用下方的共轭梯度法性质定理可知 $-d_k^T g_k = g_k^T g_k$, 且 d_k 彼此 G -共轭, 即得证。

定理 4.13 (共轭梯度法性质定理)

在上方算法的迭代过程中, 经过 $m \leq n$ 步必然停止, 且如下性质对 $1 \leq k \leq m$ 成立:

1. $g_k^T d_j = 0, j = 0, \dots, k-1$
2. $d_k^T G d_j = 0, j = 0, \dots, k-1$
3. $g_k^T g_j = 0, j = 0, \dots, k-1$
4. $d_k^T g_k = -g_k^T g_k$
5. $\text{Span}\{g_0, \dots, g_k\} = \text{Span}\{d_0, \dots, d_k\} = \text{Span}\{g_0, G g_0, \dots, G^k g_0\}$

证明 归纳证明。首先, 根据递推式与 $g_0 = g(x_0) = G x_0 + c$ 可知 $g_{k+1} = g_k + \alpha_k G d_k$ (由 G 正定对称, 以下 G^T 直接换成 G , 不另行说明)。此外, 由于 $d_0 = -g_0$, 两个关于生成空间的式子可以统一为 $\text{Span}\{g_0, g_1, \dots, g_k\} = \text{Span}\{d_0, d_1, \dots, d_k\} = \text{Span}\{g_0, G g_0, \dots, G^k g_0\}$ 。

$k = 1$ 时, $g_1^T g_0 = g_0^T g_0 + \alpha_0 d_0^T G g_0$, 由 $d_0 = -g_0$ 知结果为 0, 亦得 $g_1^T d_0 = 0$ 。 $d_1^T G d_0 = -g_1^T G d_0 + \frac{g_1^T g_1}{g_0^T g_0} d_0^T G d_0$, 展开 g_1 计算得其为 0。 $d_1^T g_1 = -g_1^T g_1 + \beta_0 d_0^T g_1$, 而 $d_0^T g_1 = -g_1^T g_0 = 0$, 于是 $d_1^T g_1 = -g_1^T g_1$ 。 $\text{Span}\{g_0, g_1\} = \text{Span}\{g_0, G d_0\} = \text{Span}\{g_0, G g_0\}$, 同理 d_1 去除 g_0 的部分后也为此, 有 $\text{Span}\{g_0, G g_0\} = \text{Span}\{d_0, d_1\}$ 。

假设结论对小于等于 k 时成立, 下面说明对 $k+1$ 成立。

$g_{k+1}^T d_j = 0, j = 0, \dots, k$: 当 $j < k$ 时, $g_{k+1}^T d_j = g_k^T d_j + \alpha_k d_k^T G d_j$, 由归纳假设知为 0, $j = k$ 时, $g_k^T d_k + \alpha_k d_k^T G d_k$, 利用 $g_k^T d_k = -g_k^T g_k$ 由 α_k 定义知为 0。

$g_{k+1}^T g_j = 0, j = 0, \dots, k$: 由上 $g_{k+1} \perp \text{Span}\{d_0, \dots, d_k\}$, 又由归纳假设 $\text{Span}\{g_0, \dots, g_k\} = \text{Span}\{d_0, \dots, d_k\}$, 从而 $g_{k+1} \perp \text{Span}\{g_0, \dots, g_k\}$, 即得证。

$d_{k+1}^T G d_j = 0, j = 0, \dots, k$: 由已证与归纳假设计算得

$$\beta_k = -\frac{g_{k+1}^T(g_{k+1} - g_k)}{g_k^T(g_k - g_{k+1})} = \frac{g_{k+1}^T G d_k}{d_k^T G d_k}$$

于是, $d_{k+1}^T G d_j = \beta_k d_k^T G d_j - g_{k+1}^T G d_j$, 注意到 $G d_j = g_{j+1} - g_j$, 由归纳假设可说明 $j < k$ 时为 0, 由上方 β_k 的表达式可知 $j = k$ 时为 0。

$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1}$: 已证明 $d_k^T g_{k+1} = 0$, 左 = $\beta_k d_k^T g_{k+1} - g_{k+1}^T g_{k+1}$ = 右。

生成空间性质: 记 $\text{Span}\{g_0, G g_0, \dots, G^k g_0\}$ 为 U_k , 由归纳假设 $g_k, d_k \in U_k$, 于是 $g_{k+1} = g_k + \alpha_k G d_k \in U_{k+1}$, 且 $d_{k+1} = -g_{k+1} + \beta_k d_k \in U_{k+1}$ 。由于 $g_{k+1}^T g_j = d_{k+1}^T G d_j = 0, j = 0, \dots, k$, G 为正定阵, 可知非零向量 $g_{k+1} \perp U_k$, $d_{k+1} \perp U_k$ (非零理由见下一部分), 于是考虑维数必有 $\text{Span}\{g_0, \dots, g_{k+1}\} = \text{Span}\{d_0, \dots, d_{k+1}\} = \text{Span}\{g_0, \dots, G^{k+1} g_0\}$ 。

终止条件: 当 $g_{k+1} = 0$ 或 $d_{k+1} = 0$ 时, 直接计算可发现迭代不会继续下去, 从而终止。终止时从 $g_{k+1} = 0$ 可直接得出 $\nabla f(x_{k+1}) = 0$, 因此是最小值。而根据上方的证明, 非零时每个 g_{k+1} 都与之前的所有垂直, 这样的 g_k 至多有 n 个, 也即 n 步以内必定停止。

针对凸二次函数, 共轭梯度法能在有限次数找到最优解 (这称为二次终止性), 而对于一般函数, 就需要利用精确或非精确一维搜索确定 α_k 。此外, d_k 的更新公式一般仍按照 $d_{k+1} = -g_{k+1} + \beta_k d_k$ 的形式, 但选取的 β_k 未必能保证所有的 d_k 关于某个 G 严格共轭。 β_k 的选取方式可以是:

$$\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \frac{g_{k+1}^T(g_{k+1} - g_k)}{d_k^T(g_{k+1} - g_k)}, \frac{g_{k+1}^T(g_{k+1} - g_k)}{g_k^T g_k}, \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}, \frac{g_{k+1}^T g_{k+1}}{d_k^T(g_{k+1} - g_k)}$$



根据性质定理可以得出, 这些选取方式在二次函数时全部等价。

除了更新方式的差别, 根据共轭方向性质, 互相共轭的 d_k 最多能选取 n 个, 于是每过 n 次都需要重新选取更新方向为负梯度方向, 称为**重启**。综上所述可以得到一般的共轭梯度法:

算法 4.14 (共轭梯度法)

1. 给定 n 元可微函数 f 与初始点 x_0 , 令 $k = 0$ 。
2. 计算梯度 $g_k = \nabla f(x_k)$, 若满足终止条件则停止, 否则进入下一步。
3. 若 k 是 n 的倍数, 令 $d_k = -g_k$, 否则令 $d_k = -g_k + \beta_{k-1} d_{k-1}$, 其中 β_{k-1} 按上述方法之一选取。
4. 更新 $x_{k+1} = x_k + \alpha_k d_k$, 其中 α_k 由一维搜索得到, 回到第二步。



在大多数情况下, 为确保共轭梯度法的快速收敛, 预条件处理 [preconditioning] 是必要的, 而对一般共轭梯度法的收敛速率, 有如下结论:

命题 4.15 (共轭梯度法-收敛速率)

由于每 n 步迭代执行重启策略, 若记重新启动时得到的点列为 $\{z_k\}$, 则这些相隔 n 次的迭代点列超线性收敛。受实际计算误差的影响, 很多情形下共轭梯度法的收敛速率类似线性。



从收敛速率角度来看, 共轭梯度法的效果未必有接下来介绍的牛顿类方法好。但由于计算过程中仅用到一阶光滑性条件 (即梯度), 且只涉及向量运算, 当问题的规模大而且有稀疏结构时, 共轭梯度法更加高效。

4.3 牛顿方法

4.3.1 牛顿法

在函数有二阶光滑性时 (即二阶可微, 海森阵存在), 利用二阶微分的性质可以改进搜索方向, 这就是牛顿类方法的思想。对 f 作二阶泰勒展开 $f(x_k + t) = f(x_k) + g_k^T t + \frac{1}{2} t^T G_k t + O(\|t\|^3)$, 其中 $g_k = \nabla f(x_k)$, $G_k = \nabla^2 f(x_k)$ 。记二阶近似 $q_k(t) = g_k^T t + \frac{1}{2} t^T G_k t$, 由于其梯度为 $G_k t + g_k$, 假设 G_k 正定, 要使其取到最小点, 需 $t = -G_k^{-1} g_k$, 这个方向 $-G_k^{-1} g_k$ 就称为牛顿方向 [Newton direction]。

对严格凸二次函数 $\frac{1}{2} x^T G x + c^T x$, 其 $g_k = G x_k + c$, $G_k = G$, 无论 x_0 为何, 令 $x_1 = x_0 - G_0^{-1} g_0$, 则有

$$g_1 = G x_1 + c = G(x_0 - G^{-1}(G x_0 + c)) + c = G x_0 - G x_0 - c + c = 0$$

于是牛顿法无需一维搜索, 必然可以一步得到最小值点。

事实上, 实际应用牛顿法时, 常常不进行一维搜索 (即令步长因子 $\alpha_k = 1$), 直接取 $x_{k+1} = x_k - G_k^{-1} g_k$ 。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, 牛顿法的收敛速率一般会很快。事实上, 有结论:

定理 4.16 (牛顿法-局部收敛性与收敛速率)

设 f 二阶可微, 且海森阵 $G(x)$ 满足 Lipschitz 条件 ($\exists \beta > 0, \forall i, j, |G_{ij}(x) - G_{ij}(y)| \leq \beta \|x - y\|$, 其中 G_{ij} 代表 G 第 i 行第 j 列元素)。某局部最优值 x^* 满足 $\nabla f(x^*) = 0$, 且 $G(x^*)$ 正定。那么, 存在 ε 满足: 只要牛顿迭代法产生的序列中某个 x_k 有 $\|x_k - x^*\| < \varepsilon$, 那么该序列以二阶速率收敛到 x^* 。



证明 记 $g(x) = \nabla f(x)$, 由泰勒展开可知

$$g(x - h) = g(x) - G(x)h + O(\|h\|^2)$$

代入 $x = x_k$, $h_k = x_k - x^*$ 可得

$$g(x_k) - G(x_k)h_k + O(\|h_k\|^2) = 0$$

由于 $G(x)$ 满足 Lipschitz 条件, 在充分接近时行列式亦充分接近, 从而可逆性满足, 且根据伴随矩阵可知 $G(x)^{-1}$ 亦充分接近 $G(x^*)^{-1}$, 因此有界, 从而有

$$G(x_k)^{-1} g(x_k) - h_k + O(\|h_k\|^2) = 0$$

根据牛顿法迭代过程, 这就是 $-h_{k+1} + O(\|h_k\|^2) = 0$, 也即存在 c 使得 $\|h_{k+1}\| \leq c\|h_k\|^2$, 即得证二阶收敛速率。

从上方定理条件也可以看出, 当 x_0 不在 x^* 的附近, 直接以 $x_{k+1} = x_k - G_k^{-1} g_k$ 进行迭代未必能收敛于最优解。为保证算法的全局收敛性, 有必要对牛顿法作某些改进, 例如以一维搜索确定步长:

算法 4.17 (阻尼牛顿法)

1. 给定初始点 x_0 与终止误差 $\varepsilon > 0$, 令 $k = 0$ 。
2. 计算 $g_k = \nabla f(x_k)$, $G_k = \nabla^2 f(x_k)$, 若 $\|g_k\| < \varepsilon$, 则算法终止, 输出 x_k , 否则进入下一步。
3. 解一维最优化问题 $\min_{\alpha \geq 0} f(x_k - \alpha G_k^{-1} g_k)$, 得到合适的步长 α_k 。

4. 更新 $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$, 回到第二步。



此外, 当 G_k 不正定时, 近似为二次函数的极小点未必存在也是牛顿法的主要困难, 修正措施包括令 $d_k = \begin{cases} -G_k^{-1}g_k & G_k^{-1} \text{ exists, } \cos \theta_k > \eta \\ -g_k & \text{Otherwise.} \end{cases}$ (Goldstein, Price), 或 $d_k = -(G_k + \mu_k I)^{-1}g_k$ (Levenberg 等),

又或者非正定时 d_k 取满足 $d^T \nabla^2 f(x_k) d < 0$ 的方向 (称为负曲率方向)。



这里 θ_k 的定义是 $-g_k$ 与 $-G_k^{-1}g_k$ 夹角。

4.3.2 拟牛顿法

虽然牛顿法有着很快的局部收敛速率, 但运用牛顿法需要计算二阶导, 而目标函数的海森矩阵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 近似牛顿法中的海森矩阵的逆 G_k^{-1} 。由构造近似矩阵的方法不同, 拟牛顿法也有差异。

在函数 $f(x)$ 在 x_k 处的二阶近似 $f(x_k + t) \approx f(x_k) + g_k^T t + \frac{1}{2} t^T G_k t$ 两边对 t 计算梯度, 可得 $\nabla f(x_k + t) \approx G_k t + g_k$, 取 $t = x_{k-1} - x_k$ 即得 $g_{k-1} \approx G_k(x_{k-1} - x_k) + g_k$, 记 $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = g_k - g_{k-1}$, 则 $G_k s_{k-1} \approx y_{k-1}$, 于是构造出的海森矩阵逆的近似 H_k 应满足 $H_k y_{k-1} = s_{k-1}$, 这个条件称为正割条件, 或拟牛顿条件。

利用此条件可以得到拟牛顿法的一般迭代过程:

算法 4.18 (拟牛顿法)

1. 给定初始点 x_0 与终止误差 $\varepsilon > 0$, 令 $k = 0, H_0 = I$ 。
2. 计算 $g_k = \nabla f(x_k)$, 若 $\|g_k\| < \varepsilon$, 则算法终止, 输出 x_k , 否则进入下一步。
3. 计算搜索方向 $d_k = -H_k g_k$, 解一维最优化问题 $\min_{\alpha \geq 0} f(x_k + \alpha d_k)$, 得到合适的步长 α_k 。
4. 更新 $x_{k+1} = x_k + \alpha_k d_k$, 并根据 x_k 与 x_{k+1} 得到满足正割条件的 H_{k+1} 。 $k = k + 1$, 回到第二步。



下面, 我们来讨论如何得到海森矩阵逆的近似 H_{k+1} , 满足 $H_{k+1} y_k = s_k$ 。我们希望在其中保持已得到的 H_k 的性质, 因此以 $H_{k+1} = H_k + E_k$ 进行计算, 其中 E_k 是某低秩矩阵 (称为低秩校正)。

最简单的想法是所谓的对称秩一 [Symmetric Rank 1, SR1] 校正, 也即取 $H_{k+1} = H_k + a u u^T$, $a \in \mathbb{R}, u \in \mathbb{R}^n$ 。

定理 4.19 (SR1 校正)

对给定的 H, s, y , 令 $u = s - Hy$ 。当 $u^T y \neq 0$ 时, $L = H + \frac{u u^T}{u^T y}$ 为满足 $Ly = s$ 的唯一对 H 的 SR1 校正方式, 否则无解。当 H 正定时, 若 $u^T y > 0$, 则 L 正定。



证明 由 $(H + a u u^T) y = s$ 可得 $a u u^T y = u$, 这即代表 $(a u^T y) u = u$, 因此可取 $w = u$, 进一步推导即知只有在 $u^T y \neq 0$ 时 L 的形式可以满足。

若 H 正定且 $u^T y > 0$, 对任何非零 x 有

$$x^T L x = x^T H x + \frac{(x^T u)^2}{u^T y} > 0$$

从而得证。

由上方证明可知，只需如下更新 H_{k+1} 就符合要求：

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}$$

SR1 校正对二次函数具有良好的性质：

定理 4.20 (SR1 校正拟牛顿法性质)

当 f 是凸二次函数 $\frac{1}{2}x^T Gx + c^T x$ 时，对称秩 1 校正的拟牛顿法有如下性质：

- $H_k y_l = s_l, l = 0, 1, \dots, k-1$ (遗传性)
- 直接取 $\alpha_k = 1$ ，若在 $H \neq G^{-1}$ 时每次都能对 H 作出更新，则 $H_n = G^{-1}$ ，且 x_{n+1} 为最优解 (二次终止性)。



证明 直接计算发现，对二次函数， $\nabla^2 f(x)$ 为常矩阵，不受 x 的变化影响，于是记其逆为 H ，又利用 $\nabla f(x)$ 是 x 的一次函数，可计算出 $s_k = Hy_k$ 。由此，拟牛顿法满足的要求化为 $(H_{k+1} - H)y_k = 0$ 。记 $D_k = H_k - H$ ，计算得迭代过程中

$$D_{k+1} = D_k - \frac{D_k y_k y_k^T D_k^T}{y_k^T D_k y_k}$$

遗传性：

由于拟牛顿法性质已保证 $D_{k+1} y_k = 0$ ，只需说明若 $D_k y = 0$ ，则 $D_{k+1} y = 0$ ，即可归纳得到结果。

直接计算 $D_{k+1} y = D_k y - \frac{D_k y_k y_k^T (D_k^T y)}{y_k^T D_k y_k}$ ，注意到过程中从单位阵每次增加对称阵，对称性保持，于是 $D_k^T y = D_k y = 0$ ，从而上式为 0，原命题得证。

二次终止性：

由此每次都更新，即假定 $D_k \neq O$ 时 $D_k y_k \neq 0$ ，先说明 D_{k+1} 的秩比起 D_k 至少减少 1。

上方遗传性中已验证 $D_k y = 0$ 的解都是 $D_{k+1} y = 0$ 的解，而 $y^{(k)}$ 不是 $D_k y = 0$ 的解，但由迭代条件是 $D_{k+1} y = 0$ 的解，考虑解空间的基可发现 $D_{k+1} y = 0$ 的解空间维数比 $D_k y = 0$ 至少多 1，因此利用秩等于阶数减零化子空间维数 ($\text{null}(A)$ ，即 $Ax = 0$ 的解空间维数) 可知秩至少减少 1。由此，至多 n 步后， $\text{rank}(D_n) = 0$ ，从而 $D_n = O$ ，即 $H_n = H$ 。

下面说明，取步长因子为 1 每次更新 $x_{k+1} = x_k + d_k$ 后，至多 $n+1$ 次可以收敛到最小值。

最后一步， $x_{n+1} = x_n - H_n \nabla f(x_n)$ ，由 $H_n = H$ 有 $H_n (\nabla f(x_{n+1}) - \nabla f(x_n)) = x_{n+1} - x_n$ ，对比得 $H_n \nabla f(x_{n+1}) = 0$ 。由于 $H = \nabla^2 f^{-1}$ 可逆，这等价于 $\nabla f(x_{n+1}) = 0$ ，从而得证。

然而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。根据 SR1 校正的性质，当 $(s_k - H_k y_k)^T y_k > 0$ 时，正定性才能够保证，但即使成立，也可能因为它很小而导致数值上的困难。于是，SR1 校的拟牛顿法应用有较大局限性，我们必须寻求新的校正方式。

考虑对称秩二 [SR2] 校正，即取 $H_{k+1} = H_k + auu^T + bvv^T, a, b \in \mathbb{R}, u, v \in \mathbb{R}^n$ 。这时， auu^T 与 bvv^T 有不同的可能，不过一个相对自然的取法是 (当 $s_k^T y_k \neq 0$ 时可实现，注意对称校正下每步 H_k 都是对称阵，直接计算验证可得 $H_{k+1} y_k = s_k$)：

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$$

这个校正称为 DFP 校正，取自 Davidon-Fletcher-Powell 的首字母。它有大量重要性质：

命题 4.21 (DFP 校正性质)

对于凸二次函数，采用精确一维搜索时，它满足：

- $H_k y_l = s_l, l = 0, 1, \dots, k-1$ (遗传性)
- $H_n = G^{-1}$ (二次终止性)
- $H_0 = I$ 时所有 d_k 满足 G -共轭 (共轭性)。

对于一般非线性函数，它满足：

- H_k 必正定， d_k 一定为下降方向；
- 每次迭代乘法运算次数 $3n^2 + O(n)$ ；
- 具有超线性收敛速率。

另一个 SR2 校正思路是，通过海森矩阵的近似 B_k ，计算其逆 H_k 。由于 B_k 需满足 $B_k s_k = y_k$ ， H_k 的 DFP 校正中 s_k, y_k 互换可得到 B_k 的 DFP 校正公式，也即 $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$ 。为了计算它的逆，我们需要一个引理：

定理 4.22 (Sherman-Morrison 定理)

设 $A \in \mathbb{R}^{n \times n}$ 非奇异， $u, v \in \mathbb{R}^n$ ，若 $1 - v^T A^{-1} u \neq 0$ ，则

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1} uv^T A^{-1}}{1 - v^T A^{-1} u}$$

证明 若 $1 - v^T A^{-1} u \neq 0$ ，直接计算右侧乘 $(A + uv^T)$ 为

$$I + A^{-1} uv^T - \frac{1}{1 - v^T A^{-1} u} A^{-1} uv^T - \frac{1}{1 - v^T A^{-1} u} (A^{-1} uv^T)^2$$

将中间两项合并即可得到

$$\frac{1}{1 - v^T A^{-1} u} v^T A^{-1} u A^{-1} uv^T$$

而

$$A^{-1} uv^T A^{-1} uv^T = A^{-1} u (v^T A^{-1} u) v^T = v^T A^{-1} u A^{-1} uv^T$$

从而得证。

由此可以得到 BFGS [Broyden-Fletcher-Goldfarb-Shanno] 校正公式：

定理 4.23 (BFGS 校正)

当 $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$ ，且 $H_k = B_k^{-1}, H_{k+1} = B_{k+1}^{-1}$ 时，有

$$H_{k+1} = H_k + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k}\right) \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k s_k^T + s_k y_k^T H_k}{s_k^T y_k}$$

且 H_{k+1} 是 H_k 的 SR2 校正。

证明 为方便书写，记第 k 步的 H_k, B_k, y_k, s_k 为 H, B, y, s ，有 $B_{k+1} = B + \frac{yy^T}{y^T s} - \frac{Bs s^T B}{s^T B s}$ 。

$$T = \left(B + \frac{yy^T}{y^T s}\right)^{-1} = B^{-1} - \frac{B^{-1} y y^T B^{-1}}{y^T s + y^T B^{-1} y} = H - \frac{H y y^T H}{y^T s + y^T H y}$$

$$H_{k+1} = B_{k+1}^{-1} = T + \frac{T B s s^T B T}{s^T B s - s^T B T B s}$$

记 $M = \frac{yy^T}{y^T s + y^T H y}$, 则计算知 $T = H - HMH, BT = I - MH, TB = I - HM, B - BTB = M$, 于是

$$H_{k+1} = H - HMH + \frac{(I - HM)ss^T(I - MH)}{s^T M s}$$

直接计算发现 $M = \frac{Mss^T M}{s^T M s}$, 于是 HMH 与 $\frac{HMss^T MH}{s^T M s}$ 可消去, 上式化简成

$$H_{k+1} = H + \frac{ss^T - HMss^T - ss^T MH}{s^T M s}$$

利用 $s^T y = y^T s$ 展开得到此即为

$$H_{k+1} = H + \frac{(s^T y + y^T H y)ss^T}{(s^T y)^2} - \frac{Hys^T + sy^T H}{s^T y}$$

反过来, 将定理中的 B 与 H 、 s 与 y 互换, 可以得到 H 作 DFP 校正时 B 的更新公式。

4.4 信赖域法

4.4.1 整体思路

前面介绍的方法, 除了不进行一维搜索的牛顿法外, 都是基于绪论中给出的两步法的基本模式确定的, 也即先确定搜索方向, 再确定步长。现在, 我们讨论另一种全局收敛策略, 也就是标题中的信赖域方法 [Trust-Region Method]。

每一步中, 定义当前点 x_k 的一个邻域 $\Omega_k = \{x \mid \|x - x_k\| \leq e_k\}$ 为信赖域, e_k 称为信赖域半径。若在这个邻域里, 二次模型 $q_k(s) = f(x_k) + g_k^T s + \frac{1}{2}s^T G_k s$ 是目标函数 $f(x)$ 的一个合适的近似 (有时海森阵 G_k 会用其近似 B_k 代替), 则在信赖域中极小化二次模型, 得到近似极小点 s_k , 并更新 $x_{k+1} = x_k + s_k$ 。信赖域方法利用二次模型在信赖域内直接求得方向与步长, 使得目标函数的下降比一维搜索更有效, 事实上, 它不仅具有全局收敛性, 而且不要求目标函数的海森矩阵正定。



直接的牛顿迭代法不具有全局收敛性的主要原因是, 当一个点附近不能用二次模型良好近似时可能严重偏离, 而信赖域的引入避免了这点。

下面, 我们先跳过子问题 $\min_{\|s\| \leq e_k} q_k(s)$ 的求解, 来看如何进行整体的迭代。实际上需要解决的问题是, 如何选取信赖域半径 e_k , 而关键在于如何刻画二次模型对目标函数的近似程度。将信赖域方法中第 k 步迭代的 $\text{Act}_k = f(x_k) - f(x_k + s_k)$ 称为实际下降量, $\text{Pre}_k = q_k(0) - q_k(s_k)$ 为预测下降量, 则比值 $r_k = \frac{\text{Act}_k}{\text{Pre}_k}$ 可以代表二次模型的近似程度。若 r_k 接近 0 或取负值, 代表函数值几乎没有下降, 应当缩小信赖域半径; 若大于 0 但不接近 1, 代表函数下降但与预测量差别较大, 信赖域半径不变; 若接近 1, 则代表模型能较好进行局部近似, 应当扩大信赖域半径。从而可以构造算法:

算法 4.24 (信赖域方法)

1. 给定初始点 x_0 , 信赖域半径上界 \bar{e} , $\varepsilon > 0, 0 < \gamma_1 < \gamma_2 < 1, 0 < \eta_1 < 1 < \eta_2$ 。取初始 $e_0 \in (0, \bar{e})$, 令 $k = 0$ 。
2. 计算 $g_k = \nabla f(x_k)$, 若 $\|g_k\| < \varepsilon$, 则算法终止, 输出 x_k , 否则求解 $\min_{\|s\| \leq e_k} q_k(s)$ 得到 s_k , 进入下一步。
3. 计算 $r_k = \frac{f(x_k) - f(x_k + s_k)}{q_k(0) - q_k(s_k)}$, 当 $r_k > 0$ 时更新 $x_{k+1} = x_k + s_k$, 否则 $x_{k+1} = x_k$ 。

$$4. \text{ 计算 } e_{k+1} = \begin{cases} \eta_1 e_k & r_k < \gamma_1 \\ e_k & \gamma_1 \leq r_k < \gamma_2, \quad k = k+1, \text{ 回到第二步。} \\ \min(\eta_2 e_k, \bar{e}) & r_k \geq \gamma_2 \end{cases}$$

也可根据 r_k 的具体值自适应地调整 e_k ，而不是固定比例，不过一般固定比例已经够用。
它的全局收敛性定理如下：

命题 4.25 (信赖域方法-全局收敛性)

设在 $\{x \mid f(x) \leq f(x_0)\}$ 上， f 有界且二阶可微，则由信赖域算法产生的迭代序列存在聚点 x^* ，满足 $\nabla f(x^*) = 0$ 且 $\nabla^2 f(x^*)$ 半正定。

4.4.2 折线法

最后，我们给出一个近似求解子问题 $\min_{\|s\| \leq e_k} q_k(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s$ 的有效方法，即 Powell 提出的折线法。折线法的思路很简单：当二次函数的理论最小点，Newton 点 $x_k - B_k^{-1} g_k$ 无法取到时，连接 Cauchy 点（最速下降法产生的极小点）与 Newton 点，将其与信赖域边界的交点取为 x_{k+1} ，示意图如下：

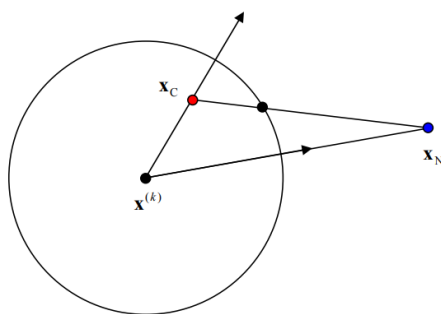


图 4.4: 折线法示意图

下面我们先计算 Cauchy 点的表达式。 $q_k(-\alpha g_k) = f(x_k) - g_k^T g_k \alpha + \frac{1}{2} g_k^T B_k g_k \alpha^2$ ，因此在最速下降方向， $\alpha = \frac{g_k^T g_k}{g_k^T B_k g_k}$ 时取到极小值，于是 Cauchy 点为 $x_k - \frac{g_k^T g_k}{g_k^T B_k g_k} g_k$ 。

为了说明此做法合理性，我们需要证明如下性质：

定理 4.26 (折线法性质定理)

记

$$s_k^C = -\frac{g_k^T g_k}{g_k^T B_k g_k} g_k, s_k^N = -B_k^{-1} g_k$$

则：

1. 沿着 s_k^C 到 s_k^N 的连线段， s 的模长单调增加。
2. 沿着 s_k^C 到 s_k^N 的连线段， $q_k(s)$ 单调减少。

证明 为方便书写，省略所有的下标的 k 。

距离单调增加：

到 x 的距离为 $\|s^C + \lambda(s^N - s^C)\|$, 需说明 $\lambda \in (0, 1)$ 时单调增加。距离平方对 λ 求导后即 $2(s^C + \lambda(s^N - s^C))^T(s^N - s^C)$, 为说明其大于等于 0, 即

$$\lambda \geq -\frac{s^{CT}(s^N - s^C)}{(s^N - s^C)^T(s^N - s^C)}$$

对 $(0, 1)$ 成立, 于是右侧非正, 进一步化简得 $s^{CT}(s^N - s^C) \geq 0$ 。直接代入得

$$\frac{g^T g}{(g^T B g)^2} (g^T B^{-1} g g^T B g - g^T g g^T g) \geq 0$$

原问题变为要证 $\frac{g^T B g}{g^T g} \frac{g^T B^{-1} g}{g^T g} \geq 1$ 。

由于 B 是对称阵, 可设 $B = Q^T D Q$ 是其正交相似对角化, 记 $h = Qg$ 可将原式化为 $\frac{h^T D h}{h^T h} \frac{h^T D^{-1} h}{h^T h} \geq 1$, 直接计算发现此即 $(\sum_i d_i h_i^2)(\sum_i d_i^{-1} h_i^2) \geq (\sum_i h_i^2)^2$, 由柯西不等式得证。

$q(s)$ 单调减少:

$$q(s) = f(x) + g^T(s^C + \lambda(s^N - s^C)) + \frac{1}{2}(s^C + \lambda(s^N - s^C))^T B(s^C + \lambda(s^N - s^C))$$

对 λ 求导得

$$g^T(s^N - s^C) + (s^C + \lambda(s^N - s^C))^T B(s^N - s^C)$$

由 $g^T = -s^{NT} B$ 即 $(\lambda - 1)(s^N - s^C)^T B(s^N - s^C)$ 。

计算知 $s^N - s^C = (\frac{g^T g}{g^T B g} I - B^{-1})g$, 于是

$$(s^N - s^C)^T B(s^N - s^C) = \frac{g^T g g^T g}{g^T B g} - 2 \frac{g^T g g^T g}{g^T B g} + g^T B^{-1} g = \frac{g^T B g g^T B^{-1} g - g^T g g^T g}{g^T B g}$$

与上方相同知分母非负, 而柯西步有意义要求 $g^T B g > 0$, 从而 $\lambda \in (0, 1)$ 时模型函数单调减。

由此, 我们连接 x_k 到 s_k^C , 再连接 s_k^C 到 s_k^N , 将折线与信赖域边界的交点取为 x_{k+1} , 具体来说, 折线法的选取是:

$$x_{k+1} = \begin{cases} x_k - \frac{e_k}{\|g_k\|} g_k & \|s_k^C\| \geq e_k \\ x_k - B_k^{-1} g_k & \|s_k^N\| \leq e_k \\ x_k + s_k^C + \lambda(s_k^N - s_k^C) & \|s_k^C\| < e_k, \|s_k^N\| > e_k \end{cases}$$

其中 λ 满足 $\|s_k^C + \lambda(s_k^N - s_k^C)\| = e_k$ 。

第5章 有约束最优化

内容提要

- | | |
|--|----------------------------------|
| <input type="checkbox"/> 消去法 | <input type="checkbox"/> 逐步二次规划法 |
| <input type="checkbox"/> 广义消去法 | <input type="checkbox"/> 罚函数 |
| <input type="checkbox"/> Lagrange 法 | <input type="checkbox"/> 乘子罚函数 |
| <input type="checkbox"/> 积极集法 | <input type="checkbox"/> 障碍函数 |
| <input type="checkbox"/> Lagrange-Newton 法 | <input type="checkbox"/> 内点法 |

5.1 二次规划

5.1.1 问题定义

最后一章，我们希望给出一般的约束最优化问题的解法。在这之前，我们先来看非线性规划最重要的特殊情况：二次规划 [Quadratic Programming] 问题。

定义 5.1 (二次规划)

形如

$$\begin{aligned} \min \quad & Q(x) \quad \left(= \frac{1}{2}x^T Gx + c^T x \right) \\ \text{s. t.} \quad & a_i^T x = b_i \quad i \in \mathcal{E} = \{1, \dots, m_e\} \\ & a_i^T x \geq b_i \quad i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \quad (\text{QP})$$

的最优化问题称为二次规划，其中 G 对称， $i \in \mathcal{E}$ 的 a_i 线性无关。



注意二次规划问题要求约束条件均为一次，根据线性规划时已证明的，此时可行域是凸集。由无约束最优化与线性规划中的推导，要求 G 对称、等式约束线性无关不会损失一般性。

若可行域为空，或不存在有限最小值，此时的二次规划问题无解。若 G 半正定，此时 $Q(x)$ 凸，称为凸二次规划问题，任何局部最优解也是整体最优解（由于可行域凸，结论证明与绪论中无约束时类似）；若 G 正定，此时称为正定二次规划问题，存在解即是唯一解；否则，问题是一般的二次规划问题，有可能出现非整体解的局部解。

为了接下来的讨论，我们先解决无约束时的二次规划问题。有如下结论：

定理 5.2 (二次函数的最小值)

对 n 元二次函数 $Q(x) = \frac{1}{2}x^T Gx + c^T x$ ，其中 G 对称。当 G 正定时，其存在唯一最小点 $x = -G^{-1}c$ ；当 G 半正定时，记 G 的 Moore-Penrose 广义逆^a为 G^+ ，则最小值存在当且仅当 $(I - GG^+)c = 0$ ，此时所有最小值点可表示为 $-G^+c + (I - GG^+)y, y \in \mathbb{R}^n$ ；否则，其最小值不存在。

^a定义为：满足 $AGA = G$ 、 $GAG = A$ 、 AG 与 GA 均对称的矩阵，可以证明对任何矩阵（未必方阵）存在唯一。



证明 设 G 的正交相似对角化为 $P^T D P$ ，其中 P 为正交阵， D 为对角阵，若其对角元为 d_i ，记 $y = Px$

可计算发现

$$Q = \sum_i \left(\frac{1}{2} d_i y_i^2 + (Pc)_i y_i \right)$$

若 G 正定, 所有 $d_i > 0$, 最小值存在唯一, 直接计算可解出 $x = -G^{-1}c$; 若 G 非半正定, 存在 d_i 为负, 最小值不存在。为考察 G 半正定时的情况, 我们在证明中直接利用 Moore-Penrose 广义逆的唯一性。


记 D^+ 为 D 的非零对角元取倒数, 零对角元不变的对角阵, 则可验证 $P^T D^+ P$ 即为 G 的 Moore-Penrose 广义逆, 由唯一性 $G^+ = P^T D^+ P$ 。于是 (不妨设 D 非零对角元在左上)

$$(I - GG^+)c = c - P^T \begin{pmatrix} I_r & O \\ O & O \end{pmatrix} Pc = P^T \begin{pmatrix} O & O \\ O & I_{n-r} \end{pmatrix} Pc$$

其为 0 等价于 Pc 的后 $n-r$ 个分量为 0, 这里 r 为 D 的非零对角元个数。根据上方的二次函数, 最小值存在当且仅当 $d_i = 0$ 时 $(Pc)_i = 0$, 而这与 $(I - GG^+)c = 0$ 一致。进一步计算可得最小点与最小值。

5.1.2 等式约束二次规划

接下来考虑等式约束与线性规划完全相同, 当只有等式约束时, 约束条件可以写为 $Ax = b$ 的形式, 其中 $A \in \mathbb{R}^{m \times n}$, 且 $\text{rank}(A) = m$ 。

 **练习 5.1** 计算等式约束二次规划问题的 K-T 条件。

解 假设乘子为 λ , 则 $\sum_i \lambda_i (a_i^T x - b_i)$ 可写为 $\lambda^T (Ax - b)$, 于是 $L(x, \lambda) = \frac{1}{2} x^T G x + (c^T - \lambda^T A)x + \lambda^T b$, 其梯度为 $Gx + (c - A^T \lambda)$, 因此 K-T 条件是:

$$\begin{cases} Gx + c = A^T \lambda \\ Ax = b \end{cases}$$

或者写成矩阵形式:

$$\begin{pmatrix} G & -A^T \\ -A & O \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = - \begin{pmatrix} c \\ b \end{pmatrix}$$

同样, 对 x 作分解 $\begin{pmatrix} x_B \\ x_N \end{pmatrix}$, 并将 A 对应分块为 $\begin{pmatrix} B & N \end{pmatrix}$, 等式约束可以写成 $x_B = B^{-1}(b - Nx_N)$ 。

将 G 同样对应分块为 $\begin{pmatrix} G_{BB} & G_{BN} \\ G_{NB} & G_{NN} \end{pmatrix}$, c 分块为 $\begin{pmatrix} c_B \\ c_N \end{pmatrix}$, 则计算可知

$$\frac{1}{2} x^T G x + c^T x = \frac{1}{2} x_N^T \hat{G}_N x_N + \hat{c}_N^T x_N$$

其中

$$\hat{G}_N = G_{NN} - G_{NB} B^{-1} N - N^T B^{-T} G_{BN} + N^T B^{-T} G_{BB} B^{-1} N$$

$$\hat{c}_N = c_N + G_{NB} B^{-1} b - N^T B^{-T} c_B + N^T B^{-T} G_{BB} B^{-1} b$$

于是, 根据之前对无约束二次函数的最小值结论, 可以直接分析解出最优的 x_N^* , 并进一步得到 x_B^* 。假设 x^* 对应的乘子向量为 λ^* , 代入 K-T 条件可得 $Gx^* + c = A^T \lambda^*$, 从而可分块计算解出

$$\lambda^* = B^{-1}(G_{BB} x_B^* + G_{BN} x_N^* + c_B)$$

这就是消去法的过程。

消去法的不足之处是, 当 B 接近奇异时, B^{-1} 可能导致数值计算的不稳定, 这就需要其他的方法。我们的思路仍然是选取一部分消去将其变成无约束问题, 但选择的方式可以改进。首先需要如下的性质定理:

定理 5.3 (解空间的性质)

记 $r = \text{rank}(A)$ 。对任何 $A \in \mathbb{R}^{m \times n}$, $\{x \in \mathbb{R}^n \mid Ax = 0\}$ 构成线性空间, 记作零化子空间 $\text{Ker}(A)$, 其维数是 $n - r$ 。

同时, $\{x \in \mathbb{R}^n \mid x^T a = 0, \forall a \in \text{Ker}(A)\}$ 也构成线性空间, 记作 $\text{Ker}(A)^\perp$, 其维数是 r 。假设其一组基 y_1, \dots, y_r 排成的矩阵是 Y , 则 $\text{rank}(AY) = r$ ($r = m$ 时即 AY 可逆)。

任何向量 $x \in \mathbb{R}^n$ 可以唯一表示成 $a + b, a \in \text{Ker}(A), b \in \text{Ker}(A)^\perp$ 。用矩阵的语言来说, 假设 $\text{Ker}(A)$ 的一组基 z_{r+1}, \dots, z_n 排成矩阵为 Z , $\text{Ker}(A)^\perp$ 的一组基 y_1, \dots, y_r 排成的矩阵为 Y , 则 $x \in \mathbb{R}^n$ 可以唯一表示成 $Yx_Y + Zx_Z, x_Y \in \mathbb{R}^{n-r}, x_Z \in \mathbb{R}^r$ 。



证明 定理中验证其为线性空间的部分容易得到, 我们先计算两个空间的维数。

由 $r = \text{rank}(A)$, 方程可以写为 (P, Q 可逆)

$$P \begin{pmatrix} I_r & O \\ O & O \end{pmatrix} Qx = 0 \Leftrightarrow \begin{pmatrix} I_r & O \\ O & O \end{pmatrix} Qx = 0$$

由于 Q 为可逆矩阵, $x \rightarrow Qx$ 为一一映射, 于是零化子空间维数与 $\begin{pmatrix} I_r & O \\ O & O \end{pmatrix} x = 0$ 维数一致, 可直接解得维数为 $n - r$ (前 r 个分量为 0, 其余随意变化)。

取 $\text{Ker}(A)$ 的一组基 z_{r+1}, \dots, z_n , 排成矩阵 Z , 则 $\text{Ker}(A)^\perp$ 可写为 $\{x \in \mathbb{R}^n \mid Zx = 0\}$, 而由基的性质 Z 是列满秩的, 于是 $\text{Ker}(A)^\perp$ 的维数与上方完全相同得为 r 。

若 $\text{Ker}(A)^\perp$ 的一组基为 y_1, \dots, y_r , 下面证明所有 y_i, z_j 构成全空间的一组基, 这样即有结论。若否, 必然存在不全为 0 的 λ_i, μ_j 使得

$$\sum_i \lambda_i y_i = \sum_j \mu_j z_j$$

但是, $z_0 = \sum_j \mu_j z_j \in \text{Ker}(A)^\perp, y_0 = \sum_i \lambda_i y_i \in \text{Ker}(A)$, 于是 $z_0^T y_0 = 0$, 由它们相等知 $y_0 = z_0 = 0$ 。而由于 y_i 与 z_j 各自均构成一组基, 必然所有 λ_i, μ_j 为 0, 矛盾。

根据此定理, $\text{rank}(A) = m$ 时, $Ax = b$ 可以写成 $A(Yx_Y + Zx_Z) = b$, 进一步分析可得到:

定理 5.4 (广义消去法)

对等式约束问题 $\min \frac{1}{2} x^T G x + c^T x \quad \text{s.t. } Ax = b$, 其中 G 对称, 且 A 行满秩, 假设 $\text{Ker}(A)$ 的一组基 z_{m+1}, \dots, z_n 排成矩阵为 Z , $\text{Ker}(A)^\perp$ 的一组基 y_1, \dots, y_m 排成的矩阵为 Y 。

当 $Z^T G Z$ 正定时, 问题有唯一最优解

$$x^* = Y(AY)^{-1} - Z(Z^T G Z)^{-1} Z^T (GY(AY)^{-1} b + c)$$

对应的乘子是

$$\lambda^* = AY^{-T} Y^T (Gx^* + c)$$



证明 将 $Ax = b$ 写成 $A(Yx_Y + Zx_Z) = b$, 由解空间性质 $AZ = O$, 于是约束变为 $AYx_Y = b$ 。下面说明 AY 可逆。由于 Ay_1, \dots, Ay_r 排成一行, 其不可逆当且仅当这些向量线性相关, 而这意味着存在

λ_i 不全为 0 使得

$$\sum_i \lambda_i A y_i = A \left(\sum_i \lambda_i y_i \right) = 0$$

这意味着 $\sum_i \lambda_i y_i \in \text{Ker}(A)$, 但这又是 $\text{Ker}(A)^T$ 的元素, 根据已证明的表示的唯一性知只能为 0, 从而 λ_i 均为 0, 矛盾。

于是, 约束可写为 $x_Y = (AY)^{-1}b$, 即 $x = Y(AY)^{-1}b + Zx_Z$, x_Z 可自由变化。将其代入后即成为无约束二次函数最小值问题, 类似消去法过程可得结论。

5.1.3 Lagrange 方法

在练习中, 我们已经求解出了 K-T 点对应的方程 $\begin{pmatrix} G & -A^T \\ -A & O \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = -\begin{pmatrix} c \\ b \end{pmatrix}$, Lagrange 方法其实就是直接求解得到 K-T 点。

由于对称阵的逆仍对称, 假设 $\begin{pmatrix} G & -A^T \\ -A & O \end{pmatrix}$ 可逆, 其逆一定可以写成 $\begin{pmatrix} U & W^T \\ W & V \end{pmatrix}$, $U \in \mathbb{R}^{n \times n}$, 其中 $W \in \mathbb{R}^{m \times n}$, $V \in \mathbb{R}^{m \times m}$, 且 U, V 对称, 这时, 直接计算可得 $\begin{cases} x^* = -Uc - W^T b \\ \lambda^* = -Wc - Vb \end{cases}$ 。

Lagrange 方法的好处是, $\begin{pmatrix} G & -A^T \\ -A & O \end{pmatrix}$ 并不依赖 G 可逆, 因此对半正定也可求解。计算分块矩阵 U, V, W 的不同方法可以得出不同的公式。

定理 5.5 (Lagrange 法-正定情况)

当 G 正定时, 由于 A 行满秩, $AG^{-1}A^T$ 可逆, 且


$$\begin{cases} U = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1} \\ V = -(AG^{-1}A^T)^{-1} \\ W = -(AG^{-1}A^T)^{-1}AG^{-1} \end{cases}$$

进一步可算出 x^*, λ^* 。



证明 我们证明 G 正定且 A 行满秩时 $AG^{-1}A^T$ 正定, 其余直接代入验算可得结论。

设 G 的正交相似对角化为 $P^T D P$, P 正交, D 为对角元均正的对角阵, 则记 $B = P A^T$, $D_0 = D^{-1}$, 原矩阵即 $B^T D_0 B$ 。由 A 行满秩, 乘可逆阵不影响秩, 可知 B 列满秩, 同时 D_0 对角元均正, 记 D_1 为 D_0 所有对角元对应作平方根, $C = D_1 B$, 即要证 C 列满秩时 $C^T C$ 正定。由 C 列满秩, 其列线性无关, 于是 x 非零时 Cx 非零, 这就直接说明了 $x^T C^T C x > 0$, 满足正定的定义。

 **练习 5.2** 举例: G 对称可逆, A 行满秩, $AG^{-1}A^T$ 不可逆。

解 取 $G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, 则 $G^{-1} = G$ 。再取 $A = \begin{pmatrix} 1 & 0 \end{pmatrix}$, 则 $AG^{-1}A^T = (0)$, 不可逆。

定理 5.6 (Lagrange 法-一般情况)

对行满秩的 A , 存在 $Y \in \mathbb{R}^{n \times m}$, $Z \in \mathbb{R}^{n \times (n-m)}$ 满足 $AY = I_{m \times m}$, $AZ = O$ 。这时若 $Z^T G Z$ 可

逆, 有

$$\begin{cases} U = Z(Z^T GZ)^{-1} Z^T \\ V = -Y^T G P^T Y \\ W = -Y^T P \end{cases}$$

其中 $P = I - GZ(Z^T GZ)^{-1} Z^T$, 进一步可算出 x^*, λ^* 。



证明 根据之前的假设, 已有 Y_0, Z_0 满足 AY_0 可逆, $AZ_0 = O$, 记 $Y = Y_0(AY_0)^{-1}, Z = Z_0$, 即符合要求。进一步计算可验证为逆。



也可用 QR 分解寻找 Y, Z , 过程与此处类似。

5.1.4 积极集法

对于一个一般的二次规划问题, 直观上, 不积极的不等式约束在解的附近不起作用, 可去掉不予考虑; 而积极的不等式约束在解处等号成立, 故我们可以用等式约束来代替这些积极的不等式约束。

练习 5.3 计算一般二次规划问题的 K-T 条件。

解 记等式约束 $A_E x = b$, 不等式约束 $A_I x \geq b$, 则

$$L(x, \lambda, \mu) = Q(x) - \lambda^T (A_I x - b) - \mu^T (A_E x - b)$$

即 K-T 条件为

$$\begin{cases} Gx + c = A_I^T \lambda + A_E^T \mu \\ A_E x = b_E \\ A_I x \geq b_I \\ \lambda \geq 0 \\ \lambda_i (A_I x - b_I)_i = 0, \forall i \end{cases}$$

对问题 (QP), 仍然记 $\mathcal{I}(x)$ 为 x 满足的不等式约束中取等的指标集, 则有如下结论:

定理 5.7 (积极集基本定理)

设 x^* 是问题 (QP) 的局部极小点, 则它必然是等式约束问题

$$\min Q(x) \quad \text{s. t. } a_i^T x = b_i, i \in \mathcal{E} \cup \mathcal{I}(x^*) \quad (\text{EP})$$

的局部极小点。

反之, 若 x^* 是问题 (QP) 的可行点, 且是 (EP) 的 K-T 点, 假设相应的 Lagrange 乘子 λ^* 满足 $\lambda_i^* \geq 0, i \in \mathcal{I}(x^*)$, 则 x^* 是 (QP) 的 K-T 点。



证明 原问题局部极小推 (EP) 局部极小: 假设其对式 122 成立局部极小性的邻域是 $U(x^*)$, 记 $V(x^*) = \bigcap_{i \in \mathcal{I}(x^*)} \{x \in \mathbb{R}^n \mid a_i^T x > b_i\}$, 其为包含 x^* 的开集, 因此 $W = U(x^*) \cap V(x^*)$ 亦为 x^* 的邻域, 且由 $W \subset U(x^*)$ 知其与原问题可行域交集 W_S 中 x^* 取到最小值。从 $W \subset V(x^*)$ 可推出 $W_S = W \cap \{x \mid a_i^T x = b_i, i \in \mathcal{E}\} \cap \{x \mid a_i^T x \geq b_i, i \in \mathcal{I}(x^*)\} \cap V(x^*)$ 必满足 $\mathcal{I}(x^*)$ 的约束, 于是 $W_{EQ} = W \cap \{x \mid a_i^T x = b_i, i \in \mathcal{E} \cup \mathcal{I}(x^*)\} \subset W_S$, W_{EQ} 即为 W 与 (EP) 可行域的交集, 从而得证。

推原问题 K-T 点: 记 (EP) 对应的 Lagrange 函数为 $L_0(x) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}(x^*)} \lambda_i^* a_i^T x$, 原问题为

$L(x) = f(x) - \sum_{i \in \mathcal{E}} \mu_i a_i^T x - \sum_{i \in \mathcal{I}} \lambda_i a_i^T x$, 取 $\mu_i = \lambda_i^*, \forall i \in \mathcal{E}, \lambda_i = \begin{cases} \lambda_i^* & i \in \mathcal{I}(x^*) \\ 0 & i \in \mathcal{I} \setminus \mathcal{I}(x^*) \end{cases}$, 下证这样取即是 x^* 满足的 K-T 条件。

这样取后直接有 $L(x) = L_0(x)$, 因此 $\nabla_x L(x) = \nabla_x L_0(x) = 0$; 可行性要求由条件已经满足; 由 $L_0(x)$ 满足 K-T 条件可知 $i \in \mathcal{I}(x^*)$ 时 $\lambda_i a_i^T x = 0$, 其他情况下 $\lambda_i = 0$, 因此仍有 $\lambda_i a_i^T x = 0$; 又由条件知 $\lambda_i \geq 0$, 从而成立。

一般地, 假设当前迭代点为 x_k , 有如下判定方式:

定理 5.8 (积极集更新)

记 $\mathcal{E}_k = \mathcal{E} \cup \mathcal{I}(x_k)$, 考虑如下等式约束问题:

$$\begin{aligned} \min \quad & Q_k(s) \quad \left(= \frac{1}{2} s^T G s + (G x_k + c)^T s \right) \\ \text{s.t.} \quad & a_i^T s = 0 \quad i \in \mathcal{E}_k \end{aligned} \quad (\text{EQ})$$

并记其解为 s_k , 相应的 Lagrange 乘子为 $\lambda_{ki}, i \in \mathcal{E}_k$, 则:

1. $s_k \neq 0$ 时, x_k 不是原问题的 K-T 点。
2. $s_k = 0$ 时, x_k 是问题 $\min Q(x) \quad \text{s.t.} \quad a_i^T x = b_i, i \in \mathcal{E}_k$ 的 K-T 点 (即 x_k 对应的 (EP) 问题)。
3. $s_k = 0$ 且 $\lambda_{ki} \geq 0, i \in \mathcal{I}(x_k)$ 时, x_k 是原问题 K-T 点。
4. $s_k = 0$ 且存在 $\lambda_{ki} < 0$ 时, 记 q 为使 λ_{ki} 取到最小值的 i , 记问题 $\min Q_k(s) \quad \text{s.t.} \quad a_i^T s = 0, i \in \mathcal{E}_k \setminus \{q\}$ 的最优解为 \hat{s} , 则 x_k 可以向 \hat{s} 方向更新, 即存在 $\varepsilon > 0$ 使得 $0 < \delta < \varepsilon$ 时 $x + \delta \hat{s}$ 在可行域中。



证明 将 \mathcal{E}_k 形成的矩阵记为 A_k , 对应 b 为 b_k , 其余记为 N_k , 根据 x_k 可行性与定义性, K-T 条件只关于 x 的部分直接满足。

由 K-T 条件最后一条, N_k 上乘子必须为 0, 于是原问题 K-T 条件可写为

$$\begin{cases} G x_k + c = A_k^T \lambda \\ \lambda_i \geq 0, i \in \mathcal{I}(x_k) \end{cases}$$

而 (EP) 的 K-T 条件即为

$$G x_k + c = A_k^T \lambda$$

(EQ) 的 K-T 条件为

$$\begin{cases} G s + G x_k + c = A_k^T \lambda \\ A_k s = 0 \end{cases}$$

当 $s \neq 0$ 时, 由 G 正定 $G s \neq 0, G x_k + c \neq A_k^T \lambda$, 于是不是原问题 K-T 点。 $s_k = 0$ 时, 其直接满足 (EP) 的 K-T 条件, 若还有乘子条件即可满足原问题条件。下面, 我们证明最关键的第四种情况。为方便书写, 省略所有的下标的 k 。

设 (EQ) 的 Lagrange 函数为

$$L(s) = \frac{1}{2} s^T G s + (G x + c)^T s - \sum_{i \in \mathcal{E}_k} \lambda_i a_i^T s$$

则第四种情况问题的 Lagrange 函数为 $L_0(s) = L(s) + \lambda_{iq} a_{iq}^T s$, 设解为 \hat{s} , 对 $L_0(s)$ 计算梯度后代入 (EP) K-T 条件得到 $\nabla_s L_0(s) = G s + \lambda_{iq} a_{iq}$, 于是 $G \hat{s} + \lambda_{iq} a_{iq} = 0$, 左乘 \hat{s}^T 得到 $\hat{s}^T G \hat{s} + \lambda_{iq} \hat{s}^T a_{iq} = 0$,

由 G 半正定与 $\lambda_{i_q} < 0$ 即得 $s^T a_{i_q} \geq 0$ 。另一方面, 根据 \hat{s} 所满足的 K-T 条件可知除了 $a_{i_q}^T \hat{s} \geq 0$ 外的更新方向要求均满足, 因此结合得到结论。

由此可以得到, 若 \mathcal{E}_k 称为每步的积极集, 当第四种情况发生时, 可以将下标 q 移出积极集并更新。从而得到算法:

算法 5.9 (积极集法)

1. 从可行点 x_0 开始, 令 $\mathcal{E}_0 = \mathcal{E} \cup \mathcal{I}(x_0)$, $k = 0$ 。
2. 求解问题 (EQ) 得到 s_k , 若其为零则直接进入第四步, 否则进入下一步。
3. 记

$$\alpha_k = \min\{1, \min_{i \notin \mathcal{E}_k, a_i^T s_k < 0} \frac{b_i - a_i^T x_k}{a_i^T s_k}\}$$

并使 $x_{k+1} = x_k + \alpha_k s_k$ 。若 $\alpha_k = 1$, 令 $\mathcal{E}_{k+1} = \mathcal{E}_k$; 否则可找到 $p \notin \mathcal{E}_k$ 使得 $a_p^T x_{k+1} = b_p$, 令 $\mathcal{E}_{k+1} = \mathcal{E}_k \cup \{p\}$ 。令 $k = k + 1$, 回到第二步。

4. 若此时 $\lambda_{ki} \geq 0, i \in \mathcal{I}(x_k)$, 迭代终止, 输出 x_k , 否则记 q 为使 λ_{ki} 取到最小值的 i , 令 $x_{k+1} = x_k, \mathcal{E}_{k+1} = \mathcal{E}_k \setminus \{q\}, k = k + 1$, 回到第二步。



证明 第四步对应的情况已在上个定理中证明, 我们主要分析第三步。当 $\alpha_k \neq 1$ 时, 意味着 x_{k+1} 在某个约束上碰到了边界, 这个约束就是对应的 $a_p^T x_{k+1} \geq b_p$, 因此将其放入积极集中。否则, 代表尚未碰到边界 (这里忽略了恰好在为 1 时碰到边界的极端情况), 积极集无需改变。

5.2 逐步二次规划

5.2.1 Lagrange-Newton 法

接下来的部分, 我们给出约束最优化的一些解法, 仍然以算法介绍为主, 跳过一些证明细节。对于一般问题, 可以借鉴二次规划的思路进行求解, 下面先考虑等式约束问题 $\min f(x) \quad \text{s.t. } c(x) = 0$, 其中 $c(x) = (c_1(x), \dots, c_m(x))^T$ 。

练习 5.4 计算等式约束最优化问题的 K-T 条件。

解 记 $A(x) = \mathcal{J}_c(x)$ (见附录 Jacobi 矩阵的定义), 则 $\nabla f(x) = \sum_i \lambda_i \nabla c_i(x)$ 可以写为 $\nabla f(x) = A(x)^T \lambda$,

于是 K-T 条件为
$$\begin{cases} \nabla f(x) - A(x)^T \lambda = 0 \\ c(x) = 0 \end{cases}。$$

与二次规划不同的是, 这个 K-T 条件的方程组无法直接求解, 但是可以通过牛顿迭代法逼近。类似无约束最优化问题, 解方程组的牛顿迭代法如下:

定理 5.10 (牛顿迭代法-方程组)

对于方程组 $F(x) = 0$, 若 F 是 $\mathbb{R}^n \rightarrow \mathbb{R}^n$ 的函数, 假设 $\mathcal{J}_F(x_k)$ 可逆, 则牛顿迭代法的迭代方式是 $x_{k+1} = x_k - \mathcal{J}_F^{-1}(x_k)F(x_k)$, 即增量是 $-\mathcal{J}_F^{-1}(x_k)F(x_k)$ 。



证明 记 $F(x) = (f_1(x), \dots, f_n(x))$, 则对每个方程作一阶泰勒展开得到

$$0 = f_i(x_k + t) = f_i(x_k) + \nabla f_i(x_k)^T t + O(t^2)$$

忽略二阶项后, 将这些方程拼接即得到

$$0 = F(x_k) + \mathcal{J}_F(x_k)t$$

于是更新量 $t = -\mathcal{J}_F^{-1}(x_k)F(x_k)$ 。

由此, 将 x, λ 都看成未知数, 可以得到 K-T 条件的牛顿迭代, 这就是 Lagrange-Newton 法, 最早由 Wilson 于 1963 年提出:

定理 5.11 (Lagrange-Newton 迭代)

$$\begin{cases} \nabla f(x) - A(x)^T \lambda = 0 \\ -c(x) = 0 \end{cases} \quad \text{的牛顿迭代是}$$

$$\begin{pmatrix} W(x, \lambda) & -A(x)^T \\ -A(x) & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - A(x)^T \lambda \\ -c(x) \end{pmatrix}$$

其中 δ_x, δ_λ 表示增量, $W(x, \lambda) = \nabla^2 f(x) - \sum_i \lambda_i \nabla^2 c_i(x)$ 。

记价值函数 $\psi(x, \lambda) = \|\nabla f(x) - A(x)^T \lambda\|^2 + \|c(x)\|^2$, 则其满足 $\nabla \psi(x, \lambda)^T \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = -2\psi(x, \lambda) \leq 0$, 也即它是此迭代法的下降函数。



证明 直接计算可得牛顿迭代的形式。为方便, 我们假设对 $\mathbb{R}^n \rightarrow \mathbb{R}^m$ 的映射 F , ∇F 即表示 \mathcal{J}_F 。

$$\psi(x, \lambda) = [\nabla f(x)]^T \nabla f(x) + \lambda^T A(x) A(x)^T \lambda - 2[\nabla f(x)]^T A(x)^T \lambda + c(x)^T c(x)$$

当 $f(x), g(x)$ 为同阶列向量时, 计算可知

$$\nabla[f(x)^T g(x)] = (\nabla f(x))g(x) + (\nabla g(x))f(x)$$

而 $\nabla A(x)^T \lambda = \nabla(\nabla c(x)\lambda) = [\nabla^2 c(x)]^T \lambda$, 其中 $S = [\nabla^2 c(x)]^T \lambda$ 意为 $\sum_i \lambda_i \nabla^2 c_i(x)$, 于是

$$\begin{aligned} \nabla_x \psi &= 2\nabla^2 f(x) \nabla f(x) + 2SA(x)^T \lambda - 2\nabla^2 f(x) A(x)^T \lambda - 2S \nabla f(x) + 2A(x)^T c(x) \\ &= 2W(x, \lambda)(\nabla f(x) - A(x)^T \lambda) + 2A(x)^T c(x) \end{aligned}$$

$$\nabla_\lambda \psi = -2A(x) \nabla f(x) + 2A(x) A(x)^T \lambda = -2A(x)(\nabla f(x) - A(x)^T \lambda)$$

注意到 W 对称, $(\nabla \psi)^T \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = 2(\nabla f(x) - A(x)^T \lambda)^T (W(x, \lambda) \delta_x - A(x)^T \delta_\lambda) + 2c(x)^T A(x) \delta_x$, 利用方程组

$$-2(\nabla f(x) - A(x)^T \lambda)^T (\nabla f(x) - A(x)^T \lambda) - 2c(x)^T c(x)$$

而这就是 -2ψ , 得证。

下面给出这个算法的实际操作流程:

算法 5.12 (Lagrange-Newton 法)

1. 给定初始 $x_0 \in \mathbb{R}^n, \lambda \in \mathbb{R}^m, \beta \in (0, 1), \varepsilon \geq 0$, 令 $k = 0$ 。
2. 计算价值函数 $\psi(x_k, \lambda_k)$, 若其 $\leq \varepsilon$ 就停止迭代并输出, 否则如上方迭代得到 $\delta_{x_k}, \delta_{\lambda_k}$, 并令 $\alpha_k = 1$ 。

3. 若

$$\psi(x_k + \alpha_k \delta_{x_k}, \lambda_k + \alpha_k \delta_{\lambda_k}) \leq (1 - \beta \alpha_k) \psi(x_k, \lambda_k)$$

则进入下一步, 否则令 $\alpha_k = \frac{1}{4} \alpha_k$, 重新执行这步。

4. 令 $x_{k+1} = x_k + \alpha_k \delta_{x_k}$, $\lambda_{k+1} = \lambda_k + \alpha_k \delta_{\lambda_k}$, $k = k + 1$, 回到第二步。



由于我们已经证明了此迭代得到的方向是价值函数的下降方向, 总存在足够小的 α 满足下降条件, 这就是为什么每次令 $\alpha_k = \frac{1}{4} \alpha_k$ 。对于这个方法的收敛性和收敛速率, 有结论:

命题 5.13 (Lagrange-Newton 法的收敛性)

若 $f(x)$ 与所有 $c_i(x)$ 均二阶可微, Lagrange-Newton 法产生的迭代点列 $\{(x_k, \lambda_k)\}$ 有界, 且存在 M 使得 $\begin{pmatrix} W(x, \lambda) & -A(x)^T \\ -A(x) & 0 \end{pmatrix}^{-1}$ 的每个元素绝对值小于 M 对任何 x, λ 成立, 那么 $\{(x_k, \lambda_k)\}$ 的任何聚点都是 $\psi(x, \lambda)$ 的根, 从而其中 $\{x_k\}$ 的任何聚点都是等式约束问题的 K-T 点



在一定条件下, 可以进一步证明 Lagrange-Newton 法具有二阶收敛速率。

5.2.2 逐步二次规划

Lagrange-Newton 法一大重要贡献是, 在其基础上发展出了逐步二次规划方法 [Sequential Quadratic Programming Methods], 而后者已成为求解一般非线性约束最优化问题的一类十分重要的方法。

刚才, 我们对方程组 $\begin{pmatrix} W(x, \lambda) & -A(x)^T \\ -A(x) & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - A(x)^T \lambda \\ -c(x) \end{pmatrix}$ 进行了直接求解。但是, 对比它与二次规划的 K-T 条件, 可以将其变为等式约束二次优化问题。

练习 5.5 从上述方程组构造一个等式约束二次优化问题, 使得 δ_x 是 K-T 点。此时的乘子向量是什么?

解 为使 δ_x 为 K-T 点, $-A(x)\delta_x = c(x)$ 决定了等式约束的形式为 $A(x)d + c(x) = 0$, 进一步可将 $A(x)^T \lambda$ 与 $A(x)^T \delta_\lambda$ 合并 (事实上构造并不唯一, 但此为较为自然的形式), 得到

$$\min_d \frac{1}{2} d^T W(x, \lambda) d + \nabla f(x)^T d \quad \text{s.t. } c(x) + A(x)d = 0$$

它对应的乘子向量就是 $\delta_\lambda + \lambda$ 。

于是, Lagrange-Newton 法可以理解为每次求解上述等式约束二次规划的方法。记求出的解为 d_k , 对应的乘子向量是 $\bar{\lambda}_k$, 则 x_k 与 λ_k 迭代更新的方式是 $\begin{cases} x_{k+1} = x_k + \alpha_k d_k \\ \lambda_{k+1} = \lambda_k + \alpha_k (\bar{\lambda}_k - \lambda_k) \end{cases}$ 。

回到一般的非线性约束最优化问题

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0 \quad i \in \mathcal{E} = \{1, \dots, m_e\} \\ & c_i(x) \geq 0 \quad i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \quad (\text{NLP})$$

类似上方可以给出迭代方式:

命题 5.14 (逐步二次规划-迭代)

假设已有当前的 x, λ , 求解子问题:

$$\begin{aligned} \min \quad & \frac{1}{2} d^T W(x, \lambda) d + \nabla f(x)^T d \\ \text{s. t.} \quad & c_i(x) + a_i(x)^T d = 0 \quad i \in \mathcal{E} \\ & c_i(x) + a_i(x)^T d \geq 0 \quad i \in \mathcal{I} \end{aligned}$$

这里 W 的定义与之前相同 (将所有等式、不等式约束作为 c), $a_i(x)$ 即为 $\nabla c_i(x)$, 以它们转置为行拼成的矩阵与之前的 $A(x)$ 相同。

将得到的 K-T 点作为搜索方向, 并进行迭代, 就是逐步二次规划法的迭代步骤。此外, 设

$$P_\sigma(x) = f(x) + \sigma \left(\sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} |c_i(x)_-| \right)$$

其中 $c_i(x)_- = \min(0, c_i(x))$, 则 P^α 是逐步二次规划迭代的下降函数, 也就是说, 任何 x 出发, 对迭代找到的 d^* 存在 $\varepsilon > 0$ 使得 $0 < \alpha < \varepsilon$ 时 $P_\sigma(x + \alpha d^*) < P_\sigma(x)$ 。

^a称为 L1 罚函数, L1 代表一范数, 罚函数定义见下文。



根据之前写出的一般二次规划问题 K-T 条件, 这时的 d 与对应的 $\bar{\lambda}$ 应满足 (下标 I, E 分别代表 \mathcal{I}, \mathcal{E} 中的部分):

$$\begin{cases} Wd + \nabla f(x) = A(x)^T \bar{\lambda} \\ c_E(x) + A_E(x)^T d = 0 \\ c_I(x) + A_I(x)^T d \geq 0 \\ \bar{\lambda}_I \geq 0 \\ \bar{\lambda}_I^T (c_I(x) + A_I(x)^T d) = 0 \end{cases}$$

从 K-T 条件可以证明下降函数的性质, 由于过程较为复杂, 此处省略。由此可以得到完整算法, 最早由 Han 于 1977 年提出:

算法 5.15 (逐步二次规划法)

1. 给定 $x_0 \in \mathbb{R}^n, W_0 \in \mathbb{R}^{n \times n}, \sigma > 0, \rho \in (0, 1), \varepsilon \geq 0$, 令 $k = 0$ 。
2. 求解逐步二次规划的迭代问题得到 d_k , 若 $\|d_k\| \leq \varepsilon$ 则停止迭代并输出, 否则求出 α_k 满足

$$P_\sigma(x_k + \alpha_k d_k) \leq \min_{0 \leq \alpha \leq \rho} P(x_k + \alpha d_k) + \varepsilon_k$$

其中 ε_k 可以理解为非精确一维搜索的容差。

3. 更新 $x_{k+1} = x_k + \alpha_k d_k$ 并计算 W_{k+1} 。令 $k = k + 1$, 回到第二步。



它具有较好的性质:

命题 5.16 (逐步二次规划法-收敛性)

若问题 (NLP) 中 f 和所有 c_i 连续可微, 且存在常数 $M_1, M_2 > 0$ 使得 $M_1 \|d\|^2 \leq d^T W_k d \leq M_2 \|d\|^2, \forall k, d$, 所有 λ_k 模最大分量的模不超过 σ , 那么逐步二次规划法算法产生的点列的任何聚点都是原问题的 K-T 点。



5.3 罚函数

5.3.1 基本定义

问题 (NLP) 中, 我们希望利用目标函数 $f(x)$ 和约束方程 $c(x)$ 所构造的具有“罚性质”的函数, 来刻画问题的不可行点违反可行条件的程度, 这就是罚函数的思想:

定义 5.17 (罚函数)

问题 $\min_{x \in S} f(x)$ 的罚函数 $P(x)$ 定义为满足 $P(x) = f(x), x \in S$ 与 $P(x) > f(x), x \notin S$ 的全空间函数。



为了刻画 (NLP) 中约束被破坏的程度, 我们记 $c_i(x)_- = \begin{cases} c_i(x) & x \in \mathcal{E} \\ \min(c_i(x), 0) & x \in \mathcal{I} \end{cases}$, 则 $|c_i(x)_-|$ 可以

描述第 i 个约束遭到破坏的程度 (对等式约束不为 0 即构成破坏, 对不等式约束需小于 0)。将它们合并为向量值函数 $c(x)_-$, $\|c(x)_-\|$ 就刻画了约束遭到破坏的程度, 只有为 0 时代表符合约束。

事实上, 罚函数 $P(x)$ 常取为 $f(x) + \phi(c(x)_-)$, 其中 $\phi(c(x)_-)$ 称为罚项。 ϕ 是一个定义在 \mathbb{R}^m 上的函数, 满足 $\phi(0) = 0, \phi_{\|c\| \in \infty} \phi(c) = +\infty$ 。下面考虑经典的 Courant 罚函数 $P_\sigma(x) = f(x) + \frac{\sigma}{2} \|c(x)_-\|^2$, 其中 $\sigma > 0$ 称为罚因子。

定理 5.18 (Courant 罚函数性质)

以下记 $\min_{x \in \mathbb{R}^n} P_\sigma(x)$ 的最优解为 $x(\sigma)$ 。

1. 对任何 $\sigma > 0$ 若 $x(\sigma)$ 同时是问题 (NLP) 的可行点, 那么它也是原问题的最优解。
2. 设 $\sigma_2 > \sigma_1 > 0$, 并记对应的 $P_{\sigma_2}, P_{\sigma_1}$ 为 P_2, P_1 , 最优解为 x_2, x_1 , 则

$$P_1(x_1) \leq P_2(x_2), \|c(x_1)_-\| \geq \|c(x_2)_-\|, f(x_1) \leq f(x_2)$$
3. 设问题 (NLP) 最优解为 x^* , 则对任何 $\sigma > 0$ 有 $f(x^*) \geq P_\sigma(x(\sigma)) \geq f(x(\sigma))$ 。
4. 令 $\delta = \|c(x(\sigma))_-\|$, 则 $x(\sigma)$ 也是约束问题 $\min_{\|c(x)_-\| \leq \delta} f(x)$ 的最优解。



证明

1. 由定义, 由于 $P_\sigma(x)$ 在可行域内即为 $f(x)$, $P_\sigma(x)$ 可行域内的最优解即为 $f(x)$ 最优解, 得证。
2. P : 若 $P_1(x_1) > P_2(x_2)$, 则

$$P_1(x_2) = P_2(x_2) - \frac{\sigma_2 - \sigma_1}{2} \|c(x_2)_-\|^2 < P_1(x_1)$$

与 x_1 最优性矛盾。

f : 由于 $P_1(x_1) \leq P_1(x_2), P_2(x_2) \leq P_2(x_1)$, 有

$$\sigma_2 P_1(x_1) + \sigma_1 P_2(x_2) \leq \sigma_1 P_2(x_1) + \sigma_2 P_1(x_2)$$

移项后 c 部分抵消, 得到 $(\sigma_2 - \sigma_1)(f(x_2) - f(x_1)) \geq 0$, 从而得证。

c : 若 $\|c(x_1)_-\| < \|c(x_2)_-\|$, 则

$$P_2(x_1) = f(x_1) + \frac{\sigma_2}{2} \|c(x_1)_-\|^2 < f(x_1) + \frac{\sigma_2}{2} \|c(x_2)_-\|^2$$

由已证 $f(x_1) \leq f(x_2)$, 于是 $P_2(x_1) < P_2(x_2)$, 与 x_2 最优性矛盾。

3. 由于

$$f(x^*) = \min_{x \in S} P_\sigma(x)$$

可知 $f(x^*) \geq P_\sigma(x(\sigma))$, 而根据罚函数定义又有 $P_\sigma(x) \geq f(x), \forall x$, 从而得证。

4. 若存在某个 t 使得 $f(t) < f(x(\sigma))$, $\|c(t)_-\| \leq \delta$, 则 $P_\sigma(t) = f(t) + \frac{\sigma}{2}\|c(t)_-\|^2 < P_\sigma(x(\sigma))$, 与 $x(\sigma)$ 最优性矛盾。

5.3.2 序贯无约束极小化

根据上方定理的第一条, 只要找到充分大的 σ , 通过求解无约束最优化问题即可找到相应约束最优化问题的最优解, 然而在实际计算中, 确定大小合适的 σ 往往比较困难, 故通常是选取一个单调增的罚因子序列 $\{\sigma_k\}$ 。这种通过求解一系列无约束问题来获得约束最优化问题的解的方式, 称为序贯无约束极小化技术 [SUMT]。

于是, 我们可以给出罚函数法的迭代步骤:

算法 5.19 (罚函数法)

1. 任选初始点 x_0 , 给定初始罚因子 $\sigma_0 > 1$ 与 $\beta > 1, \varepsilon > 0$, 令 $k = 0$ 。
2. 以 x_k 作为初始迭代点求解 $\min_{x \in \mathbb{R}^n} P_{\sigma_k}(x)$, 记结果为 x_{k+1} 。
3. 若 $\|c(x_{k+1})_-\| < \varepsilon$, 则停止迭代并输出 x_{k+1} , 否则令 $\sigma_{k+1} = \beta\sigma_k, k = k + 1$, 回到上一步。

利用 Courant 罚函数的性质可以得出它的收敛性:

定理 5.20 (罚函数法-收敛性)

记 $\bar{c} = \min_{x \in \mathbb{R}^n} \|c(x)_-\|$, 若 $\varepsilon > \bar{c}$, 则算法必然有限终止。

否则, 必有 $\lim_{k \rightarrow \infty} \|c(x_k)_-\| = \bar{c}$, 且迭代产生的任何聚点都是问题 $\min_{\|c(x)_-\|=\bar{c}} f(x)$ 的解。

证明 先证明第一种情况。由 $\sigma_0 > 0, \beta > 1$ 可知 σ_k 可任意大, 因此只需证明, 在 $\varepsilon > \min_{x \in \mathbb{R}^n} \|c(x)_-\|$ 时, 对足够大 σ , 有 $\|c(x(\sigma))_-\| < \varepsilon$ (由罚函数性质, 只要对某个 σ 成立, 对比它更大的都成立)。设取到上述最小值的 x 为 x^* , 对应最小值为 c_m 。

假设结论不成立, 对任何 σ 都有 $\|c(x(\sigma))_-\| \geq \varepsilon$, 而根据条件可知 $f(x(\sigma)) + \frac{\sigma}{2}\|c(x(\sigma))_-\|^2 \leq f(x^*) + \frac{\sigma}{2}c_m^2$, 于是 $f(x^*) + \frac{\sigma}{2}c_m^2 \geq f(x(\sigma)) + \frac{\sigma}{2}\varepsilon^2$, 整理得

$$\frac{f(x^*) - f(x(\sigma))}{\sigma} \geq \frac{\varepsilon^2 - c_m^2}{2}$$

利用罚函数性质, $f(x(\sigma)) \geq f(x(\sigma_0))$, 而若此式恒成立, $f(x(\sigma)) < f(x^*)$, 但对有界的 $f(x(\sigma))$, 可取充分大 σ 使得左侧趋于 0, 与右侧大于 0 矛盾。于是, 必存在 σ 使得 $\|c(x(\sigma))_-\| < \varepsilon$, 得证。


另一方面, 若 $\varepsilon < \bar{c}$, 算法无法终止, 而对任何 $\varepsilon_0 > \bar{c}$, 根据上方证明, 总会在某个 σ_k 后满足 $\|c(x(\sigma_k))_-\| < \varepsilon_0$, 又由 $\|c(x(\sigma_k))_-\| \geq \bar{c}$ 就证明了收敛性。

最后, 对迭代产生的任何聚点 (记子列为 x_{k_i}), 均有 $\lim_{i \rightarrow \infty} \|c(x_{k_i})_-\| = \bar{c}$, 而根据每一步解的性质与连续性, 即可知其为 $\|c(x)_-\| = \bar{c}$ 时 $f(x)$ 的最优值 (每步都会落在和最优值之差不超过 $\|c(x_k)_-\| - \bar{c}$ 的邻域内)。

5.3.3 乘子罚函数

罚函数法的问题在于, 在 $\min_{x \in \mathbb{R}^n} \|c(x)_-\| = 0$, 即可行域存在时, 它永远无法保证找到可行的最优解, 只能在一定容许度下可行。带乘子的罚函数解决了这个问题, 方便起见, 下面以等式约束问题 $\min_{c(x)=0} f(x)$ 为例介绍此方法。

根据 K-T 条件, 若 x^* 是此问题最优解且 λ^* 是对应的乘子, 则 x^* 必然是 $L(x, \lambda^*) = f(x) - \lambda^{*T} c(x)$ 的驻点, 但一般来说, 未必是极小点。

 **练习 5.6** 举例: 对等式约束问题 $\min_{c(x)=0} f(x)$ 、最优解 x^* 与对应乘子 λ^* , x^* 不是 $L(x, \lambda^*)$ 的极小点。

解 考虑 $\min_{x^3=1} x$, 其 $L(x, \lambda) = x - \lambda(x^3 - 1)$, 于是 K-T 条件有 $1 - 3x^2\lambda = 0$ 。取最优解 $x = 1$ 时, $\lambda^* = \frac{1}{3}$, $L(x, \lambda^*) = -\frac{1}{3}x^3 + x + \frac{1}{3}$, 计算二阶导数可发现 1 是极大点。

考虑乘子罚函数 (也叫增广 Lagrange) 函数 $P_\sigma(x, \lambda) = f(x) - \lambda^T c(x) + \frac{\sigma}{2} \|c(x)\|^2$ 。下面我们说明, 由于增广 Lagrange 函数的性态, 只要取足够大的罚因子 σ 而不必趋向无穷大, 就可通过极小化 $P_\sigma(x, \lambda)$ 求得原问题的最优解。为此, 先证明一个引理:

定理 5.21 (等式约束问题-二阶充分条件)

等式约束问题中, 记 $A(x) = \mathcal{J}_c(x)$ 。若对 x^*, λ^* 有 $\nabla_x L(x^*, \lambda^*) = 0, c(x^*) = 0$, 且对任何使得 $A(x^*)d = 0$ 的非零向量 d 都有 $d^T \nabla_x^2 L(x^*, \lambda^*)d > 0$, 则 x^* 是原问题的严格局部极小点^a。

^a此处严格指使其成为最小点的邻域中其他可行解函数值都比其严格大



证明 根据条件有

$$\nabla f(x^*) - A(x^*)^T \lambda = 0, c(x^*) = 0$$

而泰勒展开有

$$f(x^* + \alpha d) = f(x^*) + \alpha \nabla f(x^*)^T d + \frac{1}{2} \alpha^2 d^T \nabla^2 f(x^*) d + O(\alpha^3)$$

注意到, 若 $A(x^*)d \neq 0$, 对充分小的 α 有 $c(x^* + \alpha d) \neq c(x^*)$, 因此只有 $A(x^*)d = 0$ 的 d 可能可行。这时, 对第一个方程左乘 d^T 可知 $\nabla f(x^*)^T d = 0$, 从而有

$$f(x^* + \alpha d) = f(x^*) + \frac{1}{2} \alpha^2 d^T \nabla^2 f(x^*) d + O(\alpha^3)$$

欲说明其为严格极小点, 只须 $d^T \nabla^2 f(x^*) d > 0$ 。计算得到

$$d^T \nabla^2 f(x^*) = \nabla_x (d^T \nabla_x f(x^*)) = \nabla_x (d^T \nabla_x L(x^*, \lambda^*)) = d^T \nabla_x^2 L(x^*, \lambda^*)$$

同时右乘 d 即得 $d^T \nabla^2 f(x^*) d = d^T \nabla_x^2 L(x^*, \lambda^*) d$, 从而由条件得证。

由此, 我们证明极小化 $P_\sigma(x, \lambda)$ 的二阶充分条件和上述二阶充分条件可以相互转化:

定理 5.22 (乘子罚函数性质)

若存在 x^*, λ^* 满足等式约束问题的二阶充分条件, 则存在 σ_0 使得 $\sigma > \sigma_0$ 时 x^* 是 $P_\sigma(x, \lambda^*)$ 的严格局部极小点。

若 \bar{x} 是等式约束问题的可行解, 且存在 $\bar{\lambda}, \sigma$ 使得 $\nabla_x P_\sigma(\bar{x}, \bar{\lambda}) = 0$ 、 $\nabla_x^2 P_\sigma(\bar{x}, \bar{\lambda})$ 正定, 则 \bar{x} 是等式约束问题的严格局部极小点。



证明 原问题推罚函数: 由于 $c(x^*) = 0$ 直接计算可知

$$\nabla_x P_\sigma(x^*, \lambda^*) = \nabla_x L(x^*, \lambda^*) = 0$$

于是只需证明对充分大的 σ , 有 $\nabla_x^2 P_\sigma(x^*, \lambda^*)$ 正定。记 $L = \nabla_x^2 L(x^*, \lambda^*), C = A(x^*)^T$, 即已知对任何 $Cd = 0$ 的 d 有 $d^T L d > 0$, 且 L 对称, 求证存在充分大 σ 使得 $L + \sigma C^T C$ 正定。

由于 $x^T C^T C x = \|Cx\|^2$, 只要 $Cx \neq 0$, 必然有 $x^T C^T C x > 0$, 当 $Cx = 0$ 时, 必然 $x^T L x +$

$\sigma x^T C^T C x > 0$, 否则, 考虑 $\frac{|x^T L x|}{x^T C^T C x}$ 。由

$$\frac{|x^T L x|}{x^T C^T C x} = \frac{|x^T L x|}{x^T x} \frac{x^T x}{x^T C^T C x}$$

对两部分分别估算。设 L 的正交相似对角化为 $Q^T D Q$, 则

$$\frac{|x^T L x|}{x^T x} = \frac{|y^T D y|}{y^T y} \leq \max |D| = \rho(L)$$

其中 $y = Qx$, $\max |D|$ 代表其对角元模长最大值, 而这恰为 L 的谱半径 (特征值模长最大值)。另一方面, 由于 $C^T C$ 对称, 且 $Cx \neq 0$, 类似可知 $\frac{x^T C^T C x}{x^T x}$ 大于等于 $C^T C$ 相似对角化后的最小非零对角元 (事实上这即为 C 最小非零奇异值的平方), 记为 σ_m^2 。结合两者可知


$$\frac{|x^T L x|}{x^T C^T C x} \leq \frac{\rho(L)}{\sigma_m^2}$$

于是取 $\sigma > \frac{\rho(L)}{\sigma_m^2}$, 代入即得对任何 x 有 $x^T L x + \sigma x^T C^T C x > 0$, 满足正定性。

罚函数推原问题: 沿用上方记号, 即需要证明, 存在 σ 满足 $L + \sigma C^T C$ 正定对称时, 对任何 $Cd = 0$ 的非零 d 必有 $d^T L d > 0$ 。由正定定义知满足条件的 d 有

$$d^T L d + \sigma d^T C^T C d = d^T L d > 0$$

得证。

 $\nabla_x P_\sigma(\bar{x}, \bar{\lambda}) = 0$ 且 $\nabla_x^2 P_\sigma(\bar{x}, \bar{\lambda})$ 正定就是 \bar{x} 是 $P_\sigma(x, \bar{\lambda})$ 局部极小点的二阶充分条件。

但是, 乘子罚函数的弊端在于, 我们无法直接得知最优时的 λ , 因此必须估计 λ 才能得到有效的最优解。于是构建出算法:

算法 5.23 (乘子罚函数法)

1. 给定初始点 x_0 、初始估计 λ_0 、 $\sigma_0 > 0, \beta > 1, \gamma \in (0, 1), \varepsilon > 0$, 令 $k = 0, \sigma = \sigma_0$ 。
2. 以 x_k 作为初始迭代点求解 $\min_{x \in \mathbb{R}^n} P_{\sigma_k}(x, \lambda_k)$, 记结果为 x_{k+1} 。
3. 若 $\|c(x_{k+1})\| < \varepsilon$, 则停止迭代并输出 x_{k+1} , 否则令 $\lambda_{k+1} = \lambda_k - \sigma_k c(x_{k+1})$ 。
4. 若 $\frac{\|c(x_{k+1})\|}{\|c(x_k)\|} \geq \gamma$, 则置 $\sigma = \beta \sigma$, 否则 σ 不变, 令 $k = k + 1$, 回到第二步。



证明 我们主要说明乘子更新过程的合理性。计算 $\nabla_x P_\sigma(x, \lambda)$ 可知乘子应满足

$$\nabla f(x_{k+1}) - A(x_{k+1})^T \lambda_k + \sigma_k A(x_{k+1})^T c(x_{k+1}) = 0$$

于是更新后 λ_{k+1} 满足

$$\nabla f(x_{k+1}) - A(x_{k+1})^T \lambda_{k+1} = 0$$

这即是说 (x_{k+1}, λ_{k+1}) 构成 L 的驻点, 在此基础上可对 $c(x_{k+1})$ 进一步优化。



一般 σ_0 已经取得充分大, 于是算法主要在二三步反复。

5.4 内点法

5.4.1 障碍函数

对于等式约束最优化问题, 罚函数方法的搜索是在全空间中的, 并且只能保证结果趋于可行域。本节介绍的两种方法, 则都是保持在可行域内部进行搜索, 以严格满足约束条件, 因此相对适用于不等式约束的非线性最优化问题。

定义 5.24 (障碍函数)

考虑不等式约束最优化问题 $\min_{g(x) \geq 0} f(x)$, 其中 $g(x) = (g_1(x), \dots, g_m(x))^T$, 并记其可行域为 S . 其障碍函数定义为满足如下要求的 $S^\circ \rightarrow \mathbb{R}$ 连续函数 $B_\theta(x)$: 它能写成 $f(x) + \theta\psi(x)$ 的形式, 障碍因子 $\theta > 0$ 一般很小, $\psi(x)$ 是连续函数且在 ∂S 趋于正无穷^a.

^a定义中内部 S° 与边界 ∂S 的定义见附录



由于等式约束可以写为两个不等式约束, 这个表示方法也包含等式约束, 但存在等式约束时 S 的内部一般为空, 障碍函数没有意义。

两种经典的障碍形式是 $\psi(x) = \sum_{i=1}^m \frac{1}{g_i(x)}$ 或 $\psi(x) = -\sum_{i=1}^m \ln g_i(x)$, 当 x 趋向可行域边界时, 函数 $B_\theta(x) \rightarrow +\infty$, 否则, 由于 θ 很小, 函数 $B_\theta(x)$ 的取值近似于 $f(x)$. 于是, 我们可以通过求解带约束最优化问题 $\min_{x \in S^\circ} B_\theta(x)$ 得到原问题的近似解。



需要解释的是, 障碍问题表面上看起来仍是带约束的最优化问题, 且它的约束条件比原来的约束还要复杂。但是, 由于函数 $\psi(x)$ 的障碍阻挡作用是自动实现的, 从计算观点看, 它完全可当作无约束问题来处理, 迭代点一定不会落在可行域的外部。

类似罚函数, 障碍函数关于 θ 的解也具有某种单调性:

定理 5.25 (障碍函数性质)

若 $\theta_1 > \theta_2 > 0$, 并记对应的 $B_{\theta_1}, P_{\theta_1}$ 为 B_1, B_2 , 最优解为 x_1, x_2 , 则在 $\psi(x_1) \geq 0, \psi(x_2) \geq 0$ 时有

$$B_1(x_1) \geq B_2(x_2), \psi(x_1) \leq \psi(x_2), f(x_1) \geq f(x_2)$$



证明 B : 若 $B_1(x_1) < B_2(x_2)$, 则

$$B_2(x_1) = B_1(x_1) - (\theta_1 - \theta_2)\psi(x_1) < B_2(x_2)$$

与 x_2 最优性矛盾。

f : 由于 $B_1(x_1) \leq B_1(x_2), B_2(x_2) \leq B_2(x_1)$, 有

$$\theta_2 B_1(x_1) + \theta_1 B_2(x_2) \leq \theta_2 B_1(x_2) + \theta_1 B_2(x_1)$$

移项后 ψ 部分抵消, 得到 $(\theta_1 - \theta_2)(f(x_1) - f(x_2)) \geq 0$, 从而得证。

ψ : 若 $\psi(x_1) > \psi(x_2)$, 则

$$B_1(x_2) = f(x_2) + \theta_1 \psi(x_2) < f(x_2) + \theta_1 \psi(x_1)$$

由已证 $f(x_1) \geq f(x_2)$, 于是 $B_1(x_2) < B_1(x_1)$, 与 x_1 最优性矛盾。



据此, 取正的障碍函数时, f 随 θ 单调增加, 在 $\theta \rightarrow 0^+$ 时极限一定存在。但由于约束最优化问题的解很可能出现在边界上, 而障碍函数在边界处的情况无法保证, 这里无法简单得到 θ 充分小时收敛于最优。

于是可以构造迭代:

算法 5.26 (障碍函数法)


1. 给定初始可行点 $x_0 \in S^\circ$, 初始障碍因子 $\theta_0 > 0, \beta \in (0, 1), \varepsilon > 0$. 令 $k = 0$.
2. 以 x_k 作为初始迭代点求解 $\min_{x \in S^\circ} B_{\theta_k}(x)$, 记结果为 x_{k+1} .
3. 若 $\theta_k \psi(x_{k+1}) < \varepsilon$, 则停止迭代并输出 x_{k+1} , 否则令 $\theta_{k+1} = \beta \theta_k$, $k = k + 1$, 回到上一步。



5.4.2 原始-对偶方法

与类似罚函数构造的障碍函数不同, 原始-对偶方法类似 Lagrange-Newton 法, 给出了另一个内点迭代的思路。先将一般约束最优化问题 (NLP) 转化成如下形式:

$$\begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & c_E(x) = 0 \\ & c_I(x) - s = 0 \\ & s \geq 0 \end{aligned}$$

 **练习 5.7** 计算此问题的 K-T 条件。

解 注意此时 s 也是变元。记 $A_E(x) = \mathcal{J}_{c_E}(x)$, $A_I(x) = \mathcal{J}_{c_I}(x)$, 假设 c_E 的乘子为 y , c_I 的乘子为 z , s 的乘子为 l , 可得 $L(x, s, y, z, l) = f(x) - y^T c_E(x) - z^T (c_I(x) - s) - l^T s$, 对 x 求梯度即有 $\nabla f(x) - A_E(x)^T y - A_I(x)^T z = 0$, 而对 s 求梯度可得 $z = l$ 。约束条件必须满足, 因此有 $c_E(x) = 0$, $c_I(x) - s = 0$, $s \geq 0$, 又根据 K-T 条件中的不等式约束的情况知 $l \geq 0$, $l_i s_i = 0, \forall i$, 综合后由 $z = l$ 消去 l 可得

$$\begin{cases} \nabla f(x) - A_E(x)^T y - A_I(x)^T z = 0 \\ s_i z_i = 0, \forall i \\ c_E(x) = 0 \\ c_I(x) - s = 0 \\ s \geq 0, z \geq 0 \end{cases}$$

此时 Lagrange 函数为 $L(x, y, z) = f(x) - y^T c_E(x) - z^T c_I(x)$ 。

记 $\text{diag}(s)$ 为将 s 的元素放在对角元的对角阵, 则 $\text{diag}(s)z = 0$ 即代表 $s_i z_i = 0, \forall i$ 。将复杂的不等式约束通过互补松弛条件转化为 $s \geq 0, z \geq 0$ 的简单约束后, 若找到初始的可行 s, z , 就不难保持在可行域中迭代, 这就是原始-对偶法的思路。不过, 具体迭代求解的过程仍然需要使用 Newton 法:

 **练习 5.8** 计算利用 Newton 迭代从上方 K-T 条件根据当前的 x, s, y, z 得到的增量 $\delta_x, \delta_s, \delta_y, \delta_z$ 。

解 在不考虑不等式约束时, 可直接对应计算得到:

$$\begin{pmatrix} \nabla_x^2 L(x, y, z) & O & -A_E(x)^T & -A_I(x)^T \\ O & \text{diag}(z) & O & \text{diag}(s) \\ -A_E(x) & O & O & O \\ -A_I(x) & I & O & O \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_s \\ \delta_y \\ \delta_z \end{pmatrix} = - \begin{pmatrix} L(x, y, z) \\ \text{diag}(s)z \\ -c_E(x) \\ -c_I(s) + s \end{pmatrix}$$

根据包含 Lagrange 乘子的迭代, 可以给出完整的算法:

算法 5.27 (原始-对偶算法)

1. 给定初始可行的 x_0, s_0 与乘子初始估计 y_0, z_0 , 且满足 $s_0 > 0, z_0 > 0$, 给定 $\tau \in (0, 1)$, 令 $k = 0$ 。
2. 若 x_k, s_k, y_k, z_k 符合终止条件, 则停止迭代并输出。否则, 根据 Newton 迭代的方程解出 $\delta_x^{(k)}, \delta_s^{(k)}, \delta_y^{(k)}, \delta_z^{(k)}$ 。
3. 计算

$$\alpha_s^{(k)} = \max_{\alpha \in (0, 1]} \{s_k + \alpha \delta_s^{(k)} \geq (1 - \tau)s_k\}$$

$$\alpha_z^{(k)} = \max_{\alpha \in (0,1]} \{z_k + \alpha \delta_z^{(k)} \geq (1 - \tau)z_k\}$$

4. 更新

$$x_{k+1} = x_k + \alpha_s^{(k)} \delta_x^{(k)}, s_{k+1} = s_k + \alpha_s^{(k)} \delta_s^{(k)}$$

$$y_{k+1} = y_k + \alpha_z^{(k)} \delta_y^{(k)}, z_{k+1} = z_k + \alpha_z^{(k)} \delta_z^{(k)}$$

令 $k = k + 1$, 回到第二步。



τ 一般很接近 1, 如取 0.995。

第三步中的不等式条件称为边界比例原则 [fraction to the boundary rule], 确保了 s 与 z 不会过快趋向有分量为 0 的情况而无法继续迭代。

第6章 凸优化

内容提要

- 仿射集、凸集、锥
- 凸函数的定义与性质
- 凸优化例子
- 强弱对偶性
- 互补松弛条件
- 可行/不可行点起始的牛顿法
- 障碍函数法的实践
- 原始-对偶内点法

6.1 凸函数

6.1.1 仿射集与凸集

凸优化指的是在凸集上针对凸函数的优化问题，因此，需要先解决凸集与凸函数的定义问题。回顾第二章提到过的凸集定义，本节中，我们将从仿射集走向凸集。

定义 6.1 (仿射集)

若一个集合 $C \subset \mathbb{R}^n$ 满足 $\forall x, y \in C, \lambda \in \mathbb{R}, \lambda x + (1 - \lambda)y \in C$ ，则称集合 C 是仿射集 [affine set]。

也即对集合中任何两点，它们连成的直线也在集合中。对比定义可以发现，仿射集一定是凸集。

练习 6.1 对任何矩阵 $A \in \mathbb{R}^{m \times n}$ ，向量 $b \in \mathbb{R}^m$ ，证明 $C = \{x \mid Ax = b\}$ 是仿射集，并在 C 非空时找到线性空间 V 与向量 x_0 使得 $C = V + x_0$ 。

解 直接验证即可知为仿射集。只要 C 非空，取 $V = \{x \mid Ax = 0\}$ ，并任取 $x_0 \in C$ 即可验证满足条件。

由此直接推出超平面 $\{x \mid a^T x = b\}$ 是仿射集。

与凸集对应凸组合类似，仿射集也可对应仿射组合：

定义 6.2 (仿射组合)

对于 \mathbb{R}^n 中的一些点 a_1, \dots, a_n ，定义它们的一个仿射组合 [affine combination] 为 $\sum_{i=1}^n \lambda_i a_i$ ，其中 $\sum_{i=1}^n \lambda_i = 1$ 。

定理 6.3 (仿射集的等价定义)

- 集合 C 是仿射集的等价于其中任意有限个点的仿射组合仍在其中。
- 集合 C 是非空仿射集等价于其可以写成 $V + x_0$ ，其中 V 为某线性空间， x_0 为某向量。

证明

- 与第二章中凸集等价定义证明完全类似。
- 左推右：任取 $x_0 \in C$ ，定义 $V = C - x_0$ ，利用其包含 0，可验证对加法、数乘封闭性，从而成为线性空间。右推左：直接验证即可。

在代数中，有不少生成的概念，基本都是表示包含给定集合的符合要求的最小子集，例如生成子空间、生成子群。类似地，从任何集合出发也可以生成一个仿射集/凸集，这就称为仿射包/凸包：

定义 6.4 (仿射包、凸包)

对任何集合 A , 其所有点任意仿射组合形成的集合:

$$\text{aff}(A) = \left\{ \sum_i \lambda_i x_i \mid x_i \in A, \sum_i \lambda_i = 1 \right\}$$

称为其仿射包 [affine hull], 而所有点任意凸组合形成的集合:

$$\text{conv}(A) = \left\{ \sum_i \lambda_i x_i \mid x_i \in A, \lambda_i \geq 0, \sum_i \lambda_i = 1 \right\}$$

称为其凸包 [convex hull]。

**定理 6.5 (生成性)**

A 的仿射包/凸包是仿射集/凸集, 且任何包含 A 的仿射集/凸集都包含其仿射包/凸包。



证明 只证明凸集的情况, 对仿射集类似。对任何 $x, y \in \text{conv}(A)$, 假设由定义表示为 $x = \sum_{i=1}^m \lambda_i x_i$, $y = \sum_{i=1}^n \mu_i y_i$, 则对 $\lambda \in [0, 1]$ 有

$$\lambda x + (1 - \lambda)y = \sum_{i=1}^m (\lambda \lambda_i) x_i + \sum_{i=1}^n (\mu_i - \mu_i \lambda) y_i$$

可验证所有系数非负且和为 1, 从而 $\lambda x + (1 - \lambda)y \in \text{conv}(A)$, 符合凸集定义。

由凸集等价定义, 包含 A 的凸集必然包含 A 中所有点任意凸组合, 也就是 A 的凸包。

下面给出一些凸集的重要例子:

练习 6.2 证明下述集合是凸集:

1. 半空间 $\{x \mid a^T x \leq b\}$
2. 多面体集 $\{x \mid Ax \leq b, Cx = d\}$
3. $\mathbb{R}^{n \times n}$ 中的所有对称矩阵/正定矩阵/半正定矩阵
4. 任何范数下的球 $\{x \mid \|x\| \leq r, r > 0\}$
5. 任何范数下的锥 $\{(x, r) \mid \|x\| \leq r, r > 0\}$

解 前两问根据定义容易验证, 第三问利用正定/半正定矩阵满足的 $x^T A x$ 性质可验证, 后两问通过范数的三角不等式与数乘性质验证。

除了这些之外, 在第二章线性规划可行域性质处已经证明了凸集的交是凸集, 此性质也常用于判定凸集。

6.1.2 锥

某种意义上, 仿射集是凸集的要求中提取了“和为 1”的部分所定义的集合, 而如果提取要求中“系数非负”的部分, 可以得到另一个重要集合:

定义 6.6 (凸锥、锥)

若一个集合 $C \subset \mathbb{R}^n$ 满足 $\forall x, y \in C, \lambda, \mu \geq 0, \lambda x + \mu y \in C$, 则称集合 C 是凸锥 [convex cone]。

若一个集合 $C \subset \mathbb{R}^n$ 满足 $\forall x \in C, \lambda \geq 0, \lambda x \in C$, 则称集合 C 是锥。



由于 $(a\lambda, a(1-\lambda))$ 在 $a \geq 0, \lambda \in [0, 1]$ 时可以得到任何非负向量, 凸锥与是凸集的锥等价。后续叙述锥的更多相关定义时, “锥”一般都指凸锥, 这是因为凸锥才有与凸集、仿射集对应的性质。

完全类似定义锥组合与锥包:

定义 6.7 (锥组合、锥包)

对于 \mathbb{R}^n 中的一些点 a_1, \dots, a_n , 定义它们的一个锥组合 [conic combination] 为 $\sum_{i=1}^n \lambda_i a_i$, 其中 $\lambda_i \geq 0$ 。

对任何集合 A , 其所有点任意锥组合形成的集合:

$$\left\{ \sum_i \lambda_i x_i \mid x_i \in A, \lambda_i \geq 0 \right\}$$

称为其锥包 [conic hull]。

**定理 6.8 (锥包的生成性)**

A 的锥包是凸锥, 且任何包含 A 的凸锥都包含其锥包。



证明 与上方证明类似。

下图直观展示了凸包与锥包:

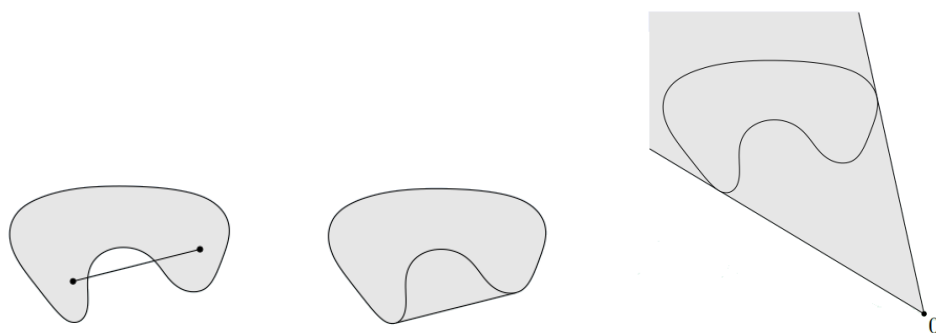


图 6.1: 凸包与锥包

可以看出, 凸包与原点是无关系的, 即原集合平移其凸包也会平移 (仿射包也满足此性质), 但锥包需要从原点出发, 因此与原点相关。

在锥中, 有一类特殊的锥非常重要, 即正常锥:

定义 6.9 (正常锥)

满足以下三个条件的凸锥 K 称为正常锥 [proper cone]:

1. K 是闭集;
2. K° 非空;
3. $x \in K, -x \in K \Leftrightarrow x = 0$ 。

**定理 6.10 (正常锥基本性质)**

正常锥中不包含任何直线。



证明 若 K 中有某条直线 $a + \lambda b, \lambda \in \mathbb{R}, b \neq 0$, 由其为锥可知 $\{xa + yb \mid x > 0, y \in \mathbb{R}\} \subset K$, 再由闭性知 $\{yb \mid y \in \mathbb{R}\} \subset K$, 与第三条性质矛盾。

正常锥的重要性在于, 它可以在空间中诱导一个性质良好的偏序关系, 如下方的例子。

定义 6.11 (正常锥诱导偏序)

给定正常锥 K , 定义关系 $x \preceq_K y \Leftrightarrow y - x \in K$, 称为此正常锥诱导的偏序关系。



证明 由于原点在 K 中, $x \preceq_K x$ 成立; 若 $x-y \in K, y-x \in K$, 由定义 $x=y$; 若 $x-y \in K, y-x \in K$, 根据凸锥定义 $x-z = (x-y) + (y-z) \in K$, 于是此关系满足偏序关系定义。

练习 6.3 证明对正常锥 K , 其诱导的偏序关系 \preceq 有如下性质:

1. $x \preceq y, u \preceq v \Rightarrow x+u \preceq y+v$
2. $x \preceq y, \alpha \geq 0 \Rightarrow \alpha x \preceq \alpha y$
3. $x_i \preceq y_i, \lim_{i \rightarrow \infty} x_i = x, \lim_{i \rightarrow \infty} y_i = y \Rightarrow x \preceq y$

解

1. $(y+v) - (x+u) = (y-x) + (v-u) \in K$
2. $\alpha y - \alpha x = \alpha(y-x) \in K$
3. $y-x = \lim_{i \rightarrow \infty} (y_i - x_i) \in K$ (利用正常锥闭性)

类似其他偏序关系, 我们可以讨论其中的极小、极大与最小、最大元素:

练习 6.4 验证 $K = \{(u, v) \mid u \geq 0, v \geq 0\}$ 是正常锥, 假设其诱导的偏序关系为 \preceq_K , 求下图 S_1, S_2 中所有的极小元与最小元:

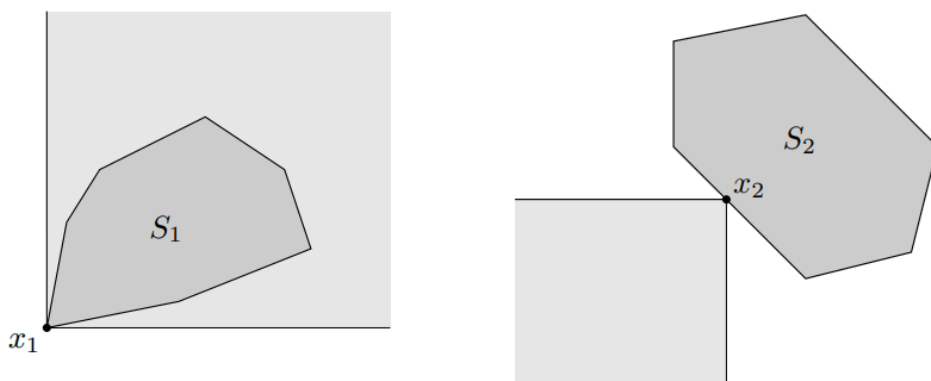


图 6.2: 偏序关系下的极小与最小

解 直接验证可得其为正常锥, 其对应的偏序关系即为 $(a, b) \preceq_K (c, d) \Leftrightarrow a \leq c, b \leq d$, 于是 S_1 中极小元与最小元均只有 x_1 , S_2 中 x_2 所在边上都为极小元, 没有最小元。

根据极小的定义, 集合 S 在锥 K 下的极小元 x 满足 $\forall s \in S, s \preceq_K x \Leftrightarrow x = s$, 也即 $s \in S, x-s \in K \Rightarrow x = s$, 从而等价于 $(x-K) \cap S = \{x\}$ 。而对最小元, 由于 $\forall s \in S, x \preceq_K s$, 即等价于 $S \subset x+K$ 。类似地, 极大元等价于 $(x+K) \cap S = \{x\}$, 最大元则等价于 $S \subset x-K$ 。

6.1.3 凸函数

在解决了凸集相关的定义后, 我们就可以给出凸函数的定义:

定义 6.12 ((严格) 凸函数)

对函数 $f: D \rightarrow \mathbb{R}$, 若满足 $D \in \mathbb{R}^n$ 为凸集, 且

$$\forall x, y \in D, \lambda \in [0, 1], f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

则称 f 为凸函数。若上方不等式的等号成立当且仅当 $\lambda = 0, \lambda = 1$ 或 $x = y$, 则称 f 为严格凸函数。



有时会定义 f 为 (严格) 凹函数当且仅当 $-f$ 为 (严格) 凸函数, 不过此定义使用较少。

与凸集等价定义类似, 凸函数的条件有等价的写法:

定理 6.13 (琴生不等式)

对函数 $f: D \rightarrow \mathbb{R}$, 若满足 $D \in \mathbb{R}^n$ 为凸集, 则 f 是凸函数等价于

$$\forall x_i \in D, \lambda_i \geq 0, \sum_i \lambda_i = 1, f\left(\sum_i \lambda_i x_i\right) \leq \sum_i \lambda_i f(x_i)$$



证明 与凸集等价定义类似归纳验证即可。

此外, 凸函数还有两个重要的性质:

定理 6.14 (下水平集凸性)

若 $f: D \rightarrow \mathbb{R}$ 是凸函数, 对任何 t , $\{x \mid f(x) \leq t\}$ (称为下水平集) 是凸集。



证明 若 $f(x_1) \leq t, f(x_2) \leq t$, 有 $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2) \leq t, \forall \lambda \in [0, 1]$, 从而得证。

反之, 若一个函数任何下水平集是凸集, 其称为拟凸函数, 但未必是凸函数, 如 $f(x) = x - |x|$ 。若一个函数在定义域大于 0 且其 \ln 是凸函数, 其称为对数凸函数。

定理 6.15 (局部最优解是全局最优解)

若 $f: D \rightarrow \mathbb{R}$ 是凸函数, 则其任何局部最小点一定也是全局最小点。



证明 对局部最小点 $x \in D$, 若存在 $y \in D$ 使得 $f(y) < f(x)$, 记 $g(\lambda) = \lambda y + (1-\lambda)x$ 。根据局部最小性, 存在 $\epsilon > 0$ 使得 $\lambda \in [0, \epsilon]$ 时 $f(g(\lambda)) \geq f(x)$, 但又有 $f(g(\lambda)) \leq \lambda f(y) + (1-\lambda)f(x) < f(x)$, 矛盾。

虽然上述定义已经足以推出很多良好的结果, 也给出了凸函数的一个直观的几何性质: 图像上任两点之间的函数图像在两点连线之下, 但由于大部分时候处理的函数具有一定光滑性, 更常用的是下面的一阶、二阶判据:

定理 6.16 (凸函数一阶条件)

对定义在凸集 D 上且一阶可微的 f , 其为凸函数当且仅当

$$\forall x, y \in D, f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

严格凸函数当且仅当对 $y \neq x$ 大于号成立。



证明 根据凸函数的定义, 我们只需要考虑任何两个 x, y 连线上的情况, 而这又意味着对函数 $g(t) = f(tx + (1-t)y)$ 考察性质, 因此可以不妨设 f 为一维。

一阶推零阶: 若满足一阶条件时存在 $x < y, \lambda \in (0, 1)$ 满足 $f(\lambda x + (1-\lambda)y) > \lambda f(x) + (1-\lambda)f(y)$, 记 $z = \lambda x + (1-\lambda)y$, 则一阶条件即

$$\frac{f(z) - f(x)}{z - x} \leq f'(z) \leq \frac{f(y) - f(z)}{y - z}$$

然而, 假设可化为 $\frac{f(z) - f(x)}{z - x} > \frac{f(y) - f(z)}{y - z}$, 矛盾。将上方证明中 \leq 变为 $<$, $>$ 变为 \geq , 即为严格凸的证明。

零阶推一阶：类似上方，对 $x < z < y$ ，零阶条件可化为

$$\frac{f(y) - f(z)}{y - z} \leq \frac{f(z) - f(x)}{z - x}$$

由不等式变形可得

$$\frac{f(x) - f(y)}{x - y} \leq \frac{f(x) - f(z)}{x - z}$$

分正负讨论可知 $\frac{f(x+\Delta x)-f(x)}{\Delta x}$ 随 Δx 减少而单调减，由此根据导数定义即得 $\Delta x > 0$ 时

$$\frac{f(x-\Delta x)-f(x)}{-\Delta x} \leq f'(x) \leq \frac{f(x+\Delta x)-f(x)}{\Delta x}$$

这即可变形为一阶条件。将上方证明中 \leq 变为 $<$ ，即为严格凸的证明。



一阶判据也有漂亮的几何解释，即函数图像在图像上任何点的切面上方。

定理 6.17 (凸函数二阶条件)

对定义在凸集 D 上且二阶可微的 f ，其为凸函数当且仅当 $\nabla^2 f(x)$ 处处半正定。

若 $\nabla^2 f(x)$ 处处半正定，且任何使其不正定的 x 都存在邻域 $B_\epsilon(x)$ ，其中除 x 外均正定，则其为严格凸函数。特别地，若至多有限点处不正定，则其为严格凸函数。



证明 类似上个定理，只需考虑一维的情况，注意到对凸函数，由上个定理证明过程， $x < z < y$ 时

$$f'(x) \leq \frac{f(z) - f(x)}{z - x} \leq \frac{f(y) - f(z)}{y - z} \leq f'(y)$$

，于是 $f'(x)$ 单调增加。反过来， $f'(x)$ 单调增加时根据中值定理可得 $x < z < y$ 时

$$\frac{f(z) - f(x)}{z - x} \leq \frac{f(y) - f(z)}{y - z}$$

因此凸函数等价于 $f'(x)$ 单调增加，严格凸函数对应严格单调增加。由此可直接得到二阶条件。

由于二阶条件的情况较为复杂，对严格凸性很难找到简单的等价条件。不过，大部分情况下“至多有限处不正定”已经足够使用了。

接下来仍然先给出一些凸函数的例子：

练习 6.5 证明以下函数为凸函数：

1. 定义在 \mathbb{R} 上， $f(x) = e^{ax}$, $a \in \mathbb{R}$;
2. 定义在正数上， $f(x) = x^a$, $a \geq 1$ or $a \leq 0$;
3. 定义在 \mathbb{R} 上， $f(x) = |x|^p$, $p \geq 1$;
4. 定义在正数上， $f(x) = -\ln x$;
5. 定义在非负实数上， $f(x) = x \ln x$ ，0 处用极限定义；
6. 定义在 \mathbb{R}^n 上，任何范数 $f(x) = \|x\|$;
7. 定义在 \mathbb{R}^n 上， $f(x) = \max\{x_1, \dots, x_n\}$;
8. 定义在 \mathbb{R}^n 上， $f(x) = \log \sum_i e^{x_i}$ (此函数可看作最大值的光滑估算)；
9. 定义在 \mathbb{R}^n 中所有分量均正的象限上， $f(x) = \sqrt[n]{\prod_i x_i}$;
10. 定义在正定矩阵上， $f(X) = -\ln \det X$ 。

证明 前九问直接通过零阶或二阶条件验证即可。对最后一问，由于对角阵行列式即为对角元乘积，由 $-\ln x$ 凸性可得到定义在对角阵上的凸性。利用正定矩阵可同时对角化，若 X, Y 正定，假设可逆阵 P

满足 $X' = P^T X P, Y' = P^T Y P$ 均为对角阵, 则

$$\begin{aligned}
 & \ln \det(\lambda X + (1 - \lambda)Y) \\
 &= \ln \det(P^{-T}(\lambda X' + (1 - \lambda)Y')P^{-1}) \\
 &= \ln \det(\lambda X' + (1 - \lambda)Y') - 2 \ln \det P \\
 &\geq \lambda \ln \det X' + (1 - \lambda) \ln \det Y' - 2(\lambda + 1 - \lambda) \ln \det P \\
 &= \lambda \ln \det(P^{-T} X' P^{-1}) + (1 - \lambda) \ln \det(P^{-T} Y' P^{-1}) \\
 &= \lambda \ln \det X + (1 - \lambda) \ln \det Y
 \end{aligned}$$

从而得证。

除了直接得到凸函数以外, 凸函数也可以组合出新的凸函数:

定理 6.18 (凸函数的组合)

以下假设组合成的函数定义域为一切有定义的范围 (如下方第一个的定义域为 $\bigcap_{i=1}^m D_{f_i}$):

1. 若 f_1, f_2, \dots, f_m 凸, $w_1, \dots, w_m \geq 0$, 则 $\sum_i w_i f_i$ 凸。
2. 若 $f(x, y)$ 给定任何 $y \in A$ 凸, $w(y) \geq 0, y \in A$, 则 $\int_A w(y) f(x, y)$ 在积分存在时凸。
3. 若 f 凸, 则 $f(Ax + b)$ 亦凸。
4. 若 f_1, f_2 凸, 则 $\max\{f_1, f_2\}$ 凸。
5. 若 $f(x, y)$ 给定任何 $y \in A$ 凸, $\sup_{y \in A} f(x, y)$ 在存在时凸。



证明 前三问直接由零阶条件放缩即可, 对 \max , 在 $x < y, \lambda \in (0, 1)$ 时, 由于 $\max\{f_1, f_2\}(x) \geq f_1(x), \max\{f_1, f_2\}(y) \geq f_1(y)$, 即有 $\lambda \max\{f_1, f_2\}(x) + (1 - \lambda) \max\{f_1, f_2\}(y) \geq f_1(\lambda x + (1 - \lambda)y)$ 同理其 $\geq f_2(\lambda x + (1 - \lambda)y)$, 因此得证, 对 \sup 证法类似。

一般来说, 需要考虑具体形式的凸函数均能通过一些常见凸函数组合而成。对凸优化来说, 问题的特殊性都是缘于凸函数的良好性质。

6.2 对偶性质

6.2.1 概念与例子

给出凸优化的具体定义:

定义 6.19 (凸优化)

形如

$$\begin{aligned}
 & \min \quad f_0(x) \\
 & \text{s. t.} \quad f_i(x) \leq 0 \quad i = 1, \dots, m \\
 & \quad \quad a_j^T x = b_j \quad j = 1, \dots, p
 \end{aligned} \tag{6.1}$$

的最优化问题被称为凸优化问题 [convex optimization], 其中 f_0, f_1, \dots, f_m 均为凸函数。



值得注意的是, 在之前讨论有约束最优化时, 我们一般把约束写成 $c_i(x) \geq 0$ 的形式, 而这里为了利用凸函数的性质, 约束均为 $f_i(x) \leq 0$ 。

定理 6.20 (凸优化的性质)

对凸优化问题，任何局部最优解都是全局最优解。



证明 由凸函数下水平集是凸集， $f_i(x) \leq 0$ 是凸集，而之前已证 $a_j^T x = b_j$ 也是凸集，因此可行域 S 是一些凸集的交，也是凸集。将 $f_0(x)$ 看作定义域 S 上的函数，问题变为凸函数的最小值点，而上一节已证明其局部最小值即为全局最小值。

根据此性质，对凸优化问题只需要找到局部最优点，就能保证其全局最优，因此之前介绍的对有约束最优化的各种处理方法都能用于寻找全局最优解。另一个等价条件是：

定理 6.21 (等式约束凸优化最优条件)

对只有等式约束的凸优化问题

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s. t.} \quad & Ax = b \end{aligned} \quad (6.2)$$

其最优解是 x^* 可推出对任何满足 $Au = 0$ 的 u 都有 $\nabla f(x^*)^T u \geq 0$ 。



证明 由于 $Ax = b$ 在某点的可行方向即为所有 $Au = 0$ 的方向，通过绪论中的几何必要条件知结论。

之前已经证明了线性规划的可行域为凸集，且 $c^T x$ 是凸函数，因此线性规划问题是凸优化问题的特殊情况。此外，当 G 半正定时， $\frac{1}{2}x^T Gx + c^T x$ 也是凸函数，这时对应的二次规划问题也是凸优化问题。注意到，线性规划问题可以看作凸二次规划问题的特殊情况，即 $G = O$ 。我们还可以进一步推广凸二次规划问题：

定义 6.22 (二次约束二次规划)

形如

$$\begin{aligned} \min \quad & \frac{1}{2}x^T P_0 x + q_0^T x + r_0 \\ \text{s. t.} \quad & \frac{1}{2}x^T P_i x + q_i^T x + r_i \leq 0 \quad i = 1, \dots, m \\ & Ax = b \end{aligned} \quad (\text{QCQP})$$

的最优化问题称为二次约束二次规划 [quadratically constrained quadratic program, QCQP]，其中 P_i 均半正定。

**定义 6.23 (二阶锥规划)**

形如

$$\begin{aligned} \min \quad & f^T x \\ \text{s. t.} \quad & \|L_i x + g_i\|_2 \leq c_i^T x + d_i \quad i = 1, \dots, m \\ & Ax = b \end{aligned} \quad (\text{SOCP})$$

的最优化问题称为二阶锥规划 [second order cone program, SOCP]。



这两个问题有关系：

定理 6.24 (QCQP 与 SOCP)

任何 QCQP 问题都可以等价于 SOCP 问题，于是 SOCP 是 QCQP 的推广。



证明 首先, 问题 (QCQP) 可以等价于

$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2}x^T P_0 x + q_0^T x + r_0 \\ \text{s. t.} \quad & \frac{1}{2}x^T P_0 x + q_0^T x - y + r_0 \leq 0 \\ & \frac{1}{2}x^T P_i x + q_i^T x + r_i \leq 0 \quad i = 1, \dots, m \\ & Ax = b \end{aligned}$$

这样就将目标函数转化为了线性函数。接着, 考虑 $\|Lx + g\|_2 \leq -q^T x - r + \frac{1}{2}$, 其中 $L = \begin{pmatrix} P^{1/2} \\ q^T \end{pmatrix}$, $g = \begin{pmatrix} 0 \\ r + \frac{1}{2} \end{pmatrix}$ 。计算其与其转置乘积即可化为 $\frac{1}{2}x^T P x + q^T x + r \leq 0$ 。从而每个约束都能写为二阶锥的形式。


6.2.2 乘子函数

为了对一般的凸优化问题提供算法, 我们仍然需要从乘子考虑。由于不等式约束是 $f_i(x) \leq 0$ 的形式, 其乘子的符号会与之前的有约束最优化问题有一定区别。考虑问题 (未必是凸优化)

$$\begin{aligned} \min \quad & f_0(x) \\ \text{s. t.} \quad & f_i(x) \leq 0 \quad i = 1, \dots, m \\ & h_j(x) = 0 \quad j = 1, \dots, p \end{aligned} \tag{6.3}$$

并假设可行域非空 (此后如无特殊说明, 默认可行域非空), 其乘子一般写成

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{j=1}^p \nu_j h_j(x)$$

 **练习 6.6** 写出问题 (6.3) 对应的 K-T 条件 (注意符号)。

解

$$\begin{cases} \nabla_x L(x, \lambda, \nu) = 0 \\ f_i(x) \leq 0 & i = 1, \dots, m \\ h_j(x) = 0 & j = 1, \dots, p \\ \lambda_i \geq 0 & i = 1, \dots, m \\ \lambda_i f_i(x) = 0 & i = 1, \dots, m \end{cases}$$

对偶方法的思路即为从 λ 与 ν 出发得到问题的最优解。具体来说, 可以如下定义对偶:

定义 6.25 (对偶函数、对偶问题)

记问题 (6.3) 中所有函数定义域交集为 D , 则其 Lagrange 对偶函数 (或对偶函数 [dual function]) 定义为

$$g(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu)$$

而其对偶问题定义为

$$\begin{aligned} \max \quad & g(\lambda, \nu) \\ \text{s. t.} \quad & \lambda \geq 0 \end{aligned} \quad (6.4)$$



下面首先说明, 前文中线性规划问题的对偶是此对偶的特殊情况:

练习 6.7 证明线性规划问题 (LP) 的对偶是问题 (DP)。

解 其乘子函数为 $L(x, \lambda) = c^T x + \lambda_{(1)}^T (b - Ax) + \lambda_{(2)}^T (-x) = (c^T - \lambda_{(1)}^T A - \lambda_{(2)}^T) x + \lambda_{(1)}^T b$, 从而

$$g(\lambda) = \begin{cases} \lambda_{(1)}^T b & c^T - \lambda_{(1)}^T A - \lambda_{(2)}^T = 0 \\ -\infty & c^T - \lambda_{(1)}^T A - \lambda_{(2)}^T \neq 0 \end{cases}$$

直接写出对偶问题为

$$\max g(\lambda) \quad \text{s. t. } \lambda_{(1)} \geq 0, \lambda_{(2)} \geq 0$$

由于其最大值只可能在 $c^T - \lambda_{(1)}^T A - \lambda_{(2)}^T = 0$ 时取到, 记 $w = \lambda_{(1)}$, 可化简为

$$\max b^T w \quad \text{s. t. } w \geq 0, c - A^T w \geq 0$$

这就是问题 (DP)。

从上方的推导已经可以发现, 对偶问题的可行域除了要求 $\lambda \geq 0$ 之外, 还应要求 $g(\lambda, \nu) > -\infty$, 这样才有进行优化的意义, 满足这两个条件的 λ, ν 称为对偶可行。

由于线性规划的对偶问题有着良好的性质, 我们希望这里一般情况的对偶也能满足此性质。事实上有:

定理 6.26 (弱对偶性质)

假设问题 (6.4) 最优值为 g^* , 问题 (6.3) 最优值为 v^* , 则有 $g^* \leq v^*$ 。这称为弱对偶性质 [weak duality], 而 $v^* - g^*$ 则称为最优对偶间隙 [optimal duality gap]。



证明 记原问题可行域为 S , 则有 $g(\lambda, \mu) \leq \inf_{x \in S} L(x, \lambda, \mu)$, 而 $x \in S$ 时 $L(x, \lambda, \mu) = f_0(x) + \sum_i \lambda_i f_i(x)$, 由于 $\lambda_i \geq 0, f_i(x) \leq 0$, 有 $L(x, \lambda, \mu) \leq f_0(x)$, 从而 $g(\lambda, \mu) \leq \inf_{x \in S} f_0(x) = v^*$, 得证。

更进一步地, 我们希望 $g^* = v^*$, 但这并不总是成立:

定义 6.27 (强对偶性质)

若问题 (6.4) 最优值 g^* 与问题 (6.3) 最优值 v^* 满足 $g^* = v^*$, 则称此问题有强对偶性质 [strong duality]。



线性规划问题在原问题对偶问题可行域都非空时具有强对偶性质, 此外, 凸优化问题在添加简单的条件后就能具有强对偶性质:

命题 6.28 (Slater 条件)

当凸优化问题 (6.1) 存在 $x \in \text{relint}(D)$ 使得 $f_i(x) < 0, Ax = b$ 时, 其满足强对偶性质。这样的 x 称为相对可行内点 [relative feasible interior point]。这里 relint 代表相对内部, 也即 D 在 $\text{aff}(D)$ 中的内部。



相对内部保证了内部不随空间改变而改变。例如, 平面上的圆的内部是去掉圆周, 而空间中圆的内部

是空集。但是，考虑相对内部后，空间中只考虑圆的仿射包(包含圆的平面)，于是圆的相对内部仍然为去掉圆周。

6.2.3 K-T 条件

在考虑对偶问题的利用方法前，我们用另一种方式去理解弱对偶性质：由于原最优化问题可以看作

$$\min_x \left\{ f_0(x) + \sum_i I_-(f_i(x)) + \sum_j I_0(h_j(x)) \right\}, I_-(u) = \begin{cases} 0 & u \leq 0 \\ +\infty & u > 0 \end{cases}, I_0(u) = \begin{cases} 0 & u = 0 \\ +\infty & u \neq 0 \end{cases}$$

而当 $\lambda_i \geq 0$ 时 $\lambda_i f_i(x) \leq I_-(f_i(x))$, $\nu_j h_j(x) \leq I_0(h_j(x))$ ，因此 $L(x, \lambda, \nu)$ 在最优 x 下的取值 $g(\lambda, \mu)$ 可以看作原问题的一个下界。

根据弱对偶性质与原问题、对偶问题的定义，只要有可行的 x 与对偶可行的 λ, μ ，必有

$$g(\lambda, \mu) \leq g^* \leq v^* \leq f_0(x)$$

于是，对满足强对偶性质的问题，只要通过优化使得 $f_0(x) - g(\lambda, \mu) \rightarrow 0$ ，得到的值必然是原问题与对偶问题的最优值，这个差称作**对偶间隙** [duality gap]。若可使对偶间隙为 0，还能得到更重要的性质：

定理 6.29 (互补松弛条件)

对满足强对偶性质的问题 (6.3)，设原问题最优解 x^* ，对偶问题最优解 λ^*, ν^* ，必有

$$\lambda_i^* f_i(x^*) = 0, i = 1, \dots, m$$

此条件称为**互补松弛条件** [complementary slackness condition]。



证明 由强对偶性知

$$g^* = \inf_x \{ f_0(x) + \sum_i \lambda_i^* f_i(x) + \sum_j \nu_j^* h_j(x) \} \leq f_0(x^*) + \sum_i \lambda_i^* f_i(x^*) + \sum_j \nu_j^* h_j(x^*) \leq f_0(x^*) = v^*$$

中的等号必须取到，因此有 $\sum_i \lambda_i^* f_i(x^*) + \sum_j \nu_j^* h_j(x^*) = 0$ ，而后半部分求和在可行点为 0，前半部分由可行性每项都 ≥ 0 ，于是只能全为 0。



优化问题中“松弛”指不取等的不等式约束，如 $f_i(x) < 0$ 或 $\lambda_i > 0$ ，而“互补松弛”即是说 $f_i(x)$ 与 λ_i 不能同时松弛。

容易发现，互补松弛条件与可行性约束 ($f_i(x^*) \leq 0, h_j(x^*) = 0, \lambda^* \geq 0$) 构成了 K-T 条件的后四条，而若原问题可微，根据 $g(\lambda, \nu)$ 的最小性可知使得 $L(x, \lambda, \mu) = g(\lambda, \nu)$ 的 x 一定有 $\nabla_x L(x, \lambda, \nu) = 0$ ，于是我们可以得到：

命题 6.30 (强对偶下的 K-T 条件必要性)

对于可微且满足强对偶性质的问题 (6.3)，原问题最优解与对偶问题最优解必须满足 K-T 条件。



6.3 数值解法

6.3.1 等式约束凸优化

由于凸优化问题不存在一般的解析方法，我们还是需要构造数值方法。回顾有约束最优化的讨论过程，我们仍然从如下的等式约束凸优化问题开始讨论：

$$\begin{aligned} \min \quad & f(x) \\ \text{s. t.} \quad & Ax = b \end{aligned} \quad (6.5)$$

其中 f 是 \mathbb{R}^n 上的二阶可微凸函数，此外与线性规划相同假设 A 行满秩，并假设最优解 x^* 存在，对应最优值 v^* 。

 **练习 6.8** 证明问题 (6.5) 满足 Slater 条件，从而有强对偶性质。

解 由已证， $S = \{x \mid Ax = b\}$ 是一个仿射集，因此其仿射包 $\text{aff}(S) = S$ 。而由于 S 在 S 中为开集 ($S = \mathbb{R}^n \cap S$, \mathbb{R}^n 是开集)， S 在 S 中的内部即为 S 自身，也即 $\text{relint}(S) = S$ ，而其中任何点都满足 $Ax = b$ ，于是存在相对可行内点，Slater 条件成立。

 **练习 6.9** 写出问题 (6.5) 对应的 K-T 条件。

解 类似线性规划时直接计算可得

$$\begin{cases} \nabla f(x) + A^T \nu = 0 \\ Ax = b \end{cases}$$

根据上节分析，这时的 K-T 条件是最优解的必要条件。与一般的等式约束问题一样，它可以通过 Lagrange-Newton 迭代求解：

 **练习 6.10** 计算问题 (6.5) 的 Lagrange-Newton 迭代步。

解 由逐步二次规划一节中的迭代步计算，代入 $c(x) = b - Ax$ ，则 $A(x) = -A$ ， $W(x, \nu) = \nabla^2 f(x)$ ，直接得到

$$\begin{pmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\nu \end{pmatrix} = \begin{pmatrix} -\nabla f(x) - A^T \nu \\ b - Ax \end{pmatrix}$$

若从可行点出发， $b - Ax = 0$ ，记 x 的更新为 $\delta_{x_{nt}}$ ，并令更新后的乘子 $w = \nu + \delta_\nu$ ，迭代步可以化简成

$$\begin{cases} \nabla^2 f(x) \delta_{x_{nt}} + A^T w = -\nabla f(x) \\ A \delta_{x_{nt}} = 0 \end{cases}$$

值得注意的是，由于 $\nabla f(x + \delta_{x_{nt}}) + A^T w \approx \nabla f(x) + \nabla^2 f(x) \delta_{x_{nt}} + A^T w$ ，牛顿法的更新步也可以看成 K-T 条件的一次泰勒展开近似。此迭代的下降速率可以表示为：

定理 6.31 (牛顿法-下降速率)

记 $\kappa(x) = \sqrt{\delta_{x_{nt}}^T \nabla^2 f(x) \delta_{x_{nt}}}$ ，则在 $\delta_{x_{nt}}$ 方向的函数下降速率

$$\left. \frac{d}{d\alpha} f(x + \alpha \delta_{x_{nt}}) \right|_{\alpha=0} = -\kappa(x)^2$$



证明 直接计算

$$\left. \frac{d}{d\alpha} f(x + \alpha \delta_{x_{nt}}) \right|_{\alpha=0} = \nabla f(x)^T \delta_{x_{nt}} = \nabla f(x)^T \delta_{x_{nt}} + w^T A \delta_{x_{nt}} = -\delta_{x_{nt}}^T \nabla^2 f(x) \delta_{x_{nt}}$$

最后两步运用了迭代步的两个等式。

因此, $\kappa(x)$ 可以表征该步牛顿迭代的下降速率, 由此作为中止条件, 得到可行点出发的牛顿迭代:

算法 6.32 (等式约束凸优化牛顿法-可行初始点)

1. 任选 $Ax = b$ 的解 x_0 , 给定 $\epsilon > 0$, 令 $k = 0$ 。
2. 求解牛顿更新 $\delta_{x_{nt}}^{(k)}$, 并计算对应的 $\kappa_k = \kappa(x_k)$ 。
3. 若 $\frac{\kappa_k^2}{2} < \epsilon$, 停止迭代并输出, 否则进入下一步。
4. 利用一维搜索获取步长因子 α_k , 并更新 $x_{k+1} = x_k + \alpha_k \delta_{x_{nt}}^{(k)}$, $k = k + 1$, 回到第二步。

若无法快速找到 $Ax = b$ 的一个根, 迭代就需要从不可行点开始, 这时迭代公式化为

$$\begin{cases} \nabla^2 f(x) \delta_{x_{nt}} + A^T w = -\nabla f(x) \\ A \delta_{x_{nt}} = b - Ax \end{cases}$$

从 K-T 条件的角度, 记余项 $r_{dual}(x, \nu) = \nabla f(x) + A^T \nu$, $r_{pri}(x) = Ax - b$, 将方程

$$\begin{cases} r_{dual}(x + \delta_{x_{nt}}, w) = 0 \\ r_{pri}(x + \delta_{x_{nt}}) = 0 \end{cases}$$

展开至一阶, 与上节相同可重新得到迭代公式。

注意到, Lagrange-Newton 法中的价值函数 $\psi(x, \nu)$ 恰为此处的 $\|r_{dual}(x, \nu)\|^2 + \|r_{pri}(x)\|^2$, 因此可以通过价值函数的大小作为收敛标准:

算法 6.33 (等式约束凸优化牛顿法-任意初始点)

1. 任选 x_0 , 给定 $\epsilon > 0, \tau \in (0, \frac{1}{2}), \gamma \in (0, 1)$, 令 $k = 0$ 。
2. 计算价值函数 $\psi(x_k, \nu_k)$, 若其小于 ϵ 就停止迭代并输出, 否则进入下一步。
3. 求解牛顿更新 $\delta_{x_{nt}}^{(k)}, \delta_{\nu_{nt}}^{(k)}$, 并令 $\alpha_k = 1$ 。
4. 若

$$\psi(x_k + \alpha \delta_{x_{nt}}^{(k)}, \nu_k + \alpha \delta_{\nu_{nt}}^{(k)}) > (1 - \tau \alpha) \psi(x_k, \nu_k)$$

则进入下一步, 否则令 $\alpha_k = \gamma \alpha_k$, 重新执行这步。

5. 更新 $x_{k+1} = x_k + \alpha_k \delta_{x_{nt}}^{(k)}, \nu_{k+1} = \nu_k + \alpha_k \delta_{\nu_{nt}}^{(k)}$, $k = k + 1$, 回到第二步。

本质来说, 这里的两种牛顿法都是 Lagrange-Newton 方法在等式约束凸优化问题的应用。不过, 由于具体的问题与初始条件的不同, 算法结构也存在差异。

6.3.2 障碍函数法

对一般的凸优化问题 (6.1), 为方便设计算法, 下面先作出几个假定:

- 所有 f_i 定义域为 \mathbb{R}^n 且二阶可微;
- 等式约束合成的 A 行满秩;
- 最优解 x^* 存在, 最优值为 v^* ;
- 存在 $Ax = b$ 且 $f_i(x) > 0$ 的 x (这称为此问题严格可行 [strictly feasible])。

由所有 f_i 定义域为 \mathbb{R}^n , 问题的定义域 $D = \mathbb{R}^n$, 于是再结合严格可行性知 Slater 条件成立, 即它是强对偶的。由于这是一个不等式约束问题, 有约束最优化章节中的两种内点法仍然可以使用。本段先介绍障碍函数法, 在下一段中介绍原始-对偶方法。

上一章中, 障碍函数法 [barrier method] 考虑的约束只有不等式约束, 而一般凸优化问题由于等式约束相对容易处理, 障碍函数仍然是作用在不等式约束上。具体来说, 原问题可以等价 (I_- 定义见上节)

$$\min_{Ax=b} f_0(x) + \sum_i I_-(f_i(x))$$

而障碍函数的想法就是用光滑函数近似 I_- , 一个较好的选择是取 $\hat{I}_-(u) = -\frac{1}{t} \ln(-u)$, 定义域为负实数。其示意图像如图 6.3 实线, t 越大近似程度越高。

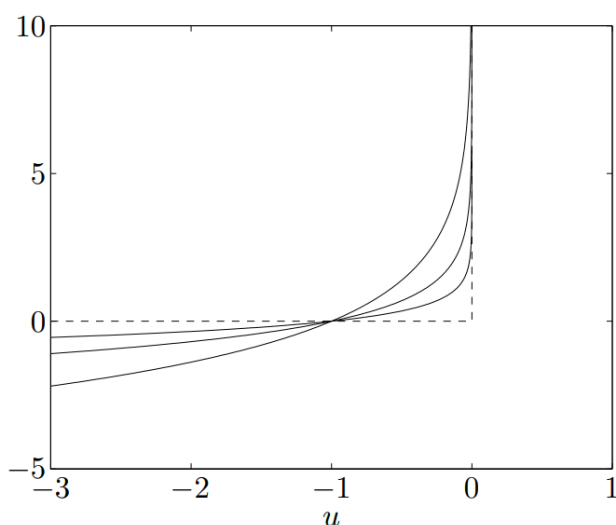



图 6.3: $t = 0.5, 1, 2$ 时的 $\hat{I}_-(u)$ 与 $I_-(u)$ 对比

采用此近似后, 障碍函数法对应的问题可以写为 (将优化目标乘 t 以方便书写)

$$\begin{aligned} \min \quad & f(x) + \frac{1}{t} \phi(x) \\ \text{s. t.} \quad & Ax = b \end{aligned} \quad (6.6)$$

这里 $\phi(x) = -\sum_{i=1}^m \ln(-f_i(x))$ 称为对数障碍 [logarithmic barrier]。

 **练习 6.11** 计算 $\nabla \phi(x)$, $\nabla^2 \phi(x)$, 并计算问题 (6.6) 的 K-T 条件。

解

$$\begin{aligned} \nabla \phi(x) &= -\sum_i \frac{1}{f_i(x)} \nabla f_i(x) \\ \nabla^2 \phi(x) &= \sum_i \frac{1}{f_i^2(x)} \nabla f_i(x) \nabla f_i(x)^T - \sum_i \frac{1}{f_i(x)} \nabla^2 f_i(x) \end{aligned}$$

于是进一步计算得 K-T 条件为

$$\begin{cases} \nabla f_0(x) - \sum_i \frac{1}{t f_i(x)} \nabla f_i(x) + A^T \nu = 0 \\ Ax = b \end{cases}$$

只要能找到一个满足所有 $f_i(x_0) < 0$ 的起点 x_0 , 对任何 $t > 0$, 问题 (6.6) 都可以采用牛顿法进行求解。

定义 6.34 (中心路径、中心点)

对给定 t 的问题 (6.6) 的最优解可以构成函数 $x^*(t)$, 称为中心路径 [central path], 中心路径上的点称为中心点 [centrol point]。

**定理 6.35 (对偶可行的构造)**

对任何 $t > 0$, 记 $\lambda_i^*(t) = -\frac{1}{tf_i(x^*(t))}$, 给定 t 时最优解对应的乘子为 $\nu^*(t)$, 则 $(\lambda^*(t), \nu^*(t))$ 对原问题对偶可行。



证明 由 ϕ 定义 $f_i(x^*(t)) < 0$, 于是 $\lambda^*(t) > 0$ 。此外, K-T 条件可化为

$$\nabla f_0(x^*(t)) + \sum_i \lambda_i^*(t) \nabla f_i(x^*(t)) + A^T \nu^*(t) = 0$$

由此计算得 $x^*(t)$ 是原问题的

$$L(x, \lambda^*(t), \nu^*(t)) = f_0(x) + \sum_i \lambda_i^*(t) f_i(x) + \nu^*(t)(Ax - b)$$

的驻点, 且 $\lambda^*(t) > 0$ 可推出 $L(x, \lambda^*(t), \nu^*(t))$ 是若干凸函数的正线性组合, 从而仍然是凸函数, 驻点必然为最小值。于是, $g(\lambda^*(t), \nu^*(t)) = L(x^*(t), \lambda^*(t), \nu^*(t)) > -\infty$, $(\lambda^*(t), \nu^*(t))$ 对偶可行。

这个定理展示了障碍函数法对凸优化的良好效果: 从任何 t , 都可以得到一组对偶可行的乘子, 更进一步地, 我们可以证明中心路径的收敛性:

定理 6.36 (障碍函数法-收敛性)

$$\lim_{t \rightarrow \infty} f_0(x^*(t)) = v^*$$



证明 利用 $Ax^*(t) - b = 0$ 可知

$$g(\lambda^*(t), \nu^*(t)) = f_0(x^*(t)) - \sum_{i=1}^m \frac{f_i(x^*(t))}{tf_i(x^*(t))} = f_0(x^*(t)) - \frac{m}{t}$$

从而根据 $g(\lambda^*(t), \nu^*(t)) \leq v^*$ 移项有

$$f_0(x^*(t)) - v^* \leq \frac{m}{t}$$

又由于 $f_0(x^*(t)) \geq v^*$ 即得证。

这意味着, 只要我们取充分大的 t , 就可以得到符合约束且任意接近最优的结果, 而前提只是找到一个内点 (严格可行点) 出发进行迭代, 足见障碍函数法对凸优化的效果。

事实上, 从 K-T 条件出发也可以观察到收敛性。计算发现 $x^*(t), \lambda^*(t), \nu^*(t)$ 满足

$$\begin{cases} \nabla_x L(x^*(t), \lambda^*(t), \nu^*(t)) = 0 \\ f_i(x^*(t)) < 0 & i = 1, \dots, m \\ Ax^*(t) = b \\ \lambda_i^*(t) > 0 & i = 1, \dots, m \\ \lambda_i^*(t) f_i(x^*(t)) = -\frac{1}{t} & i = 1, \dots, m \end{cases}$$

在 $t \rightarrow \infty$ 时, 最后一个式子均趋于 0, 因此趋于满足 K-T 条件。下面给出完整的算法:

算法 6.37 (凸优化-障碍函数法)

1. 给定严格可行点 x_0 , $t_0 > 0, \gamma > 1, \epsilon > 0$, 令 $k = 0$ 。
2. 由牛顿法优化 $\min_{Ax=b} \{f_0(x) + \frac{1}{t_k} \phi(x)\}$, 记最优解 x_{k+1} 。
3. 若 $\frac{m}{t} < \epsilon$ 则停止迭代并输出, 否则令 $t_{k+1} = \gamma t_k$, $k = k + 1$, 回到第二步。



算法的每次循环称为外循环 [outer iteration], 而牛顿法的循环称为内循环 [inner iteration]。

6.3.3 单相法

虽然此算法在理论上已经足够优美, 实践中仍然存在一些需要考虑的因素:

- $x^*(t)$ 并不需要精确计算, 因此内循环的迭代精度需要设置合适;
- t_0 若太大, 首个外循环可能需要太多次迭代, 但 t_0 太小则导致外循环次数增加;
- γ 越大, 所需外循环次数会更少, 而内循环的次数则会更多。

此外, 还有一个重要的问题: 初始的严格可行点未必容易找到。就像线性规划利用两阶段法寻找初始可行基解, 寻找初始严格可行点的算法称为**单相法** [phase I method]。在求解问题 (6.1) 前, 先考虑如下的问题:

$$\begin{aligned} \min_{x,s} \quad & s \\ \text{s.t.} \quad & f_i(x) \leq s \quad i = 1, \dots, m \\ & Ax = b \end{aligned} \quad (6.7)$$

由定义直接得出此问题的解与原问题可行/严格可行解的密切关系:

命题 6.38 (单相法的性质)

记问题 (6.7) 的最优值为 \bar{v}^* , 则:

- 若 $\bar{v}^* < 0$, 原问题存在严格可行解, 且任何 $s < 0$ 时的 x 均严格可行;
- 若 $\bar{v}^* = 0$ 且最优值可以取到, 原问题存在可行解但不存在严格可行解;
- 若 $\bar{v}^* > 0$ 或 $\bar{v}^* = 0$ 且最优值不可取到, 原问题不存在可行解。

且由于此问题的严格可行条件为 $f_i(x) < s, Ax = b$, 只要取充分大的 s 即可保证初始严格可行, 从而迭代后能判断原问题的解的情况。在得到当前 $s < 0$ 或由下界确定 $\bar{v}^* > 0$ 后, 迭代即可停止。

6.3.4 原始-对偶方法

在可行点初始的牛顿法中, 我们只需要对 x 进行优化, w 只是求解的中间变量; 而任意点初始的情况下, 考虑 ν 能更好刻画结果的可行性与最优性。那么, 既然一般的约束优化问题有两个乘子, 直接对两个乘子同时更新可能成为一个可靠的选择, 这就是原始-对偶方法的思路。作为例子, 我们先考虑用原始-对偶方法求解障碍函数法产生的中间问题 (6.6):

我们重新写出初始为可行点的牛顿迭代的具体方程 (这里 ν_{nt} 表示更新后的乘子):

$$\begin{pmatrix} \nabla^2 f_0(x) + \frac{1}{t} \nabla^2 \phi(x) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \nu_{nt} \end{pmatrix} = \begin{pmatrix} -\nabla f_0(x) - \frac{1}{t} \nabla \phi(x) \\ 0 \end{pmatrix}$$

计算仍可发现, 它相当于对修正的 **K-T 条件** [modified K-T equations]


$$\begin{cases} \nabla f_0(x) + \sum_i \lambda_i \nabla f_i(x) + A^T \nu = 0 \\ \lambda_i f_i(x) = -\frac{1}{t} \\ Ax = b \end{cases}$$

消去 λ_i 后进行一阶泰勒展开。

这意味着, 对问题 (6.6), 修正 **K-T 条件** 是一个本质的条件, 于是我们尝试由此出发构造原始-对偶算法。记 $f(x) = (f_1(x), \dots, f_m(x))^T$, 修正 **K-T 条件** 中的等式约束 (而不等式约束为 $f(x) < 0$ 与 $\lambda \geq 0$) 可重新写为

$$r_t(x, \lambda, \nu) = \begin{pmatrix} r_{dual}(x, \lambda, \nu) \\ r_{cent}(x, \lambda) \\ r_{pri}(x) \end{pmatrix} = \begin{pmatrix} \nabla f_0(x) + \mathcal{J}_f(x)^T \lambda + A^T \nu \\ -\text{diag}(\lambda)f(x) - \frac{1}{t} \mathbf{1} \\ Ax - b \end{pmatrix} = 0$$

这里 $\text{diag}(\lambda)$ 表示以 λ 为对角元的对角阵, 而 $\mathbf{1}$ 为所有元素均 1 的向量。从上到下的三个余项分别称为对偶余项 [dual residual]、原始余项 [primal residual] 与中心化余项 [centrality residual]。理论来说, 只要余项充分小, 且两个不等式约束满足, 我们就可以认为 x, λ, ν 充分逼近了最优解。

 **练习 6.12** 计算上述 $r_t(x, \lambda, \nu)$ 作为关于 x, λ, ν 的方程组的牛顿迭代更新步长。

解 求导计算得须满足

$$\begin{pmatrix} \nabla^2 f_0(x) + \sum_i \lambda_i \nabla^2 f_i(x) & \mathcal{J}_f(x)^T & A^T \\ -\text{diag}(\lambda)\mathcal{J}_f(x) & -\text{diag}(f(x)) & O \\ A & O & O \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \\ \delta_\nu \end{pmatrix} = -r_t(x, \lambda, \nu)$$

这样得到的 $\delta_{x_{pd}}, \delta_{\lambda_{pd}}, \delta_{\nu_{pd}}$ 称为原始-对偶搜索方向 [primal-dual search direction]。注意到, 直接使用此式进行搜索、更新已经忽略了显式的 ϕ 。

从搜索方向到完整的原始-对偶方法, 还有两个任务需要完成: 误差界的刻画与线性搜索。由于迭代过程中不能严格保证 x 可行, (λ, ν) 对偶可行, 原始-对偶方法对并没有 $\frac{m}{t}$ 这样简单的界限刻画。我们需要采取替代方案:

定义 6.39 (替代对偶间隙)

对满足 $f(x) < 0$ 与 $\lambda \geq 0$ 的 x 与 λ , 定义替代对偶间隙 [surrogate duality gap] 为 $\hat{\eta}(x, \lambda) = -f(x)^T \lambda$ 。



由 $\lambda^*(t)$ 定义可知 $\hat{\eta}(x^*(t), \lambda^*(t)) = \frac{m}{t}$, 于是替代对偶间隙确实可以刻画当前的对偶间隙。反过来说, 如果已知 $\hat{\eta}(x, \lambda)$, 可以估算当前的 t 为 $\frac{m}{\hat{\eta}}$, 这就给出了更新 t 的方式, 与障碍函数法思路并不相同。

此外, 线性搜索也不能采取无约束最优化中的方案, 因为需要保证不等式约束的满足。我们采取回溯线搜索 [backtracking line search] 的做法:

算法 6.40 (回溯线搜索)

1. 已知当前的 t, x, λ, ν 与更新方向 $\delta_x, \delta_\lambda, \delta_\nu$ 。
2. 记 $\alpha^{\max} = \sup\{\alpha \in (0, 1], \lambda + \alpha\delta_\lambda \geq 0\}$ 为 λ 的最大更新步长。
3. 记 $\alpha = 0.99\alpha^{\max}$, 取定 $\beta \in (0, 1), \tau \in [0.01, 0.1]$ 。

4. 若 $f(x + \alpha\delta_x) > 0$ 且

$$\|r_t(x + \alpha\delta_x, \lambda + \alpha\delta_\lambda, \nu + \alpha\delta_\nu)\| < (1 - \tau\alpha)\|r_t(x, \lambda, \nu)\|$$

则停止迭代并输出 α ，否则令 $\alpha = \beta\alpha$ ，重复执行本步骤。



由于可直接对每个分量计算后取最小值， α^{\max} 是容易计算的，而这里得 0.99 与 τ 的范围均为经验结果。

综上，我们得到了完整的原始-对偶算法：

算法 6.41 (凸优化-原始对偶法)

1. 初始 x_0, λ_0, ν_0 满足 $f(x_0) < 0, \lambda_0 > 0$ 。给定 $\gamma > 1, \epsilon_{feas} > 0, \epsilon > 0$ ，令 $k = 0$ 。
2. 计算

$$t_k = \gamma \frac{m}{\hat{\eta}(x_k, \mu_k)}$$

并以此计算出原始-对偶搜索方向 $\delta_x^{(k)}, \delta_\lambda^{(k)}, \delta_\nu^{(k)}$ 。

3. 利用回溯线搜索确定 $\alpha_k > 0$ ，更新 $x_{k+1} = x_k + \alpha_k \delta_x^{(k)}, \lambda_{k+1} = \lambda_k + \alpha_k \delta_\lambda^{(k)}, \nu_{k+1} = \nu_k + \alpha_k \delta_\nu^{(k)}$ ， $k = k + 1$ 。
4. 若 $\|r_{dual}(x_k, \lambda_k, \nu_k)\| \leq \epsilon_{feas}, \|r_{pri}(x_k)\| \leq \epsilon_{feas}, \hat{\eta}(x_k, \mu_k) \leq \epsilon$ 则停止迭代并输出，否则回到第二步。



这里 ϵ_{feas} 是 feasible 的缩写，代表可行性的约束。

比起障碍函数法，原始-对偶方法有更多的不确定性，数学上的收敛性也更加模糊。但是，在实际应用中，由于其对 t 自适应更新，且不用区分内外循环，可能收敛更加迅速。

第 7 章 大数据中的优化

内容提要

- 零范数与凸松弛
- 随机梯度类方法
- 紧邻线性算法
- 随机梯度法的收敛性
- ADMM 算法
- 降噪方法
- 机器学习的优化视角
- 二阶光滑方法

7.1 稀疏优化



本节中有较多范数相关定义，详见附录。

7.1.1 零范数优化问题

虽然我们已经介绍了不少一般的优化理论，在现实生活中，尤其是近年来大数据与机器学习技术不断发展的情况下，一些特殊优化问题的重要性逐渐显现。本节中主要介绍其中的一类，即**稀疏优化** [sparse optimization]。

数学上，稀疏一般指的是一个向量或矩阵大部分分量都是 0，或利用附录中的定义，其零范数很小。一个经典的稀疏优化问题即是

$$\min \|x\|_0 \quad \text{s. t. } Ax = b \quad (P_0)$$

或者在考虑容差时

$$\min \|x\|_0 \quad \text{s. t. } \|b - Ax\| \leq \epsilon \quad (P_0^\epsilon)$$

对此问题 (P_0^ϵ) 的一个经典的贪婪近似解法是**正交匹配追踪** [orthogonal matching pursuit] 算法：

算法 7.1 (正交匹配追踪)

1. 给定矩阵 A (记其每列为 a_i)，向量 b 与容差 ϵ 。初始 $x^{(0)} = 0, r^0 = b - Ax^{(0)} = b, S^{(0)} = \{i \mid x_i \neq 0\} = \emptyset$ ，令 $k = 0$ 。
2. 记 $z_j^{(k)} = \frac{a_j^T r^{(k)}}{\|a_j\|^2}$ ，并计算 $j_0^{(k)} = \arg\min_j \|z_j a_j - r^{(k)}\|$ 。
3. 更新 $S^{(k+1)} = S^{(k)} \cup \{j_0\}$ ， $x^{(k+1)} = x^{(k)} + z_{j_0}^{(k)} e_{j_0}$ 。
4. $k = k + 1$ ，计算 $r^{(k)} = b - Ax^{(k)}$ ，若 $\|r^{(k)}\| \leq \epsilon$ 则停止迭代并输出，否则回到第二步。

证明 下面证明每次选取的 j_0 满足¹

$$j_0 = \arg\min_j \min_t \|A(x^{(k)} + te_j) - b\|$$

也即 j_0 选取的是 x 只改变一个分量能减小最多误差的分量。此外，再说明 x 的改变量是此分量最优的改变量即可。

¹ $\arg\min_x f(x)$ 代表使 $f(x)$ 最小值取到的 x 值。

利用二次函数知识可算出 $z_j^{(k)} = \min_t \|ta_j - r^{(k)}\|$, 另一方面有

$$\|A(x^{(k)} + te_j) - b\| = \|tAe_j - r^{(k)}\| = \|ta_j - r^{(k)}\|$$

因此 $\operatorname{argmin}_j \min_t \|A(x^{(k)} + te_j) - b\| = \operatorname{argmin}_j \|z_j a_j - r^{(k)}\| = j_0$, 从而得证。

除了零范数作为优化目标外, 实践中也会涉及一些其作为约束的问题, 如机器学习中出现的:

定义 7.2 (字典学习)

形如

$$\begin{aligned} \min_{D, X} \quad & \|Y - DX\|_F \\ \text{s. t.} \quad & \|x_j\|_0 \leq k_0 \quad j = 1, \dots, N \end{aligned}$$

的最优化问题被称为字典学习 [dictionary learning], 其中 $Y \in \mathbb{R}^{n \times N}$, $D \in \mathbb{R}^{n \times m}$, $X \in \mathbb{R}^{m \times N}$, X 的每列记为 x_j 。



字典学习的经典算法有最优方向算法 [Method of Optimal Directions, MOD]、K-SVD 等。

零范数最优化问题并不是一个凸优化问题, 这是由于零范数并不是凸函数。我们希望能用凸优化问题近似原问题, 以获取更有效的算法, 这个过程称为凸松弛 [convex relaxation]。零范数优化常用的凸松弛为:

$$\min \|Wx\|_1 \quad \text{s. t. } Ax = b \quad (P_1)$$

$$\min \|Wx\|_1 \quad \text{s. t. } \|b - Ax\| \leq \epsilon \quad (P_1^\epsilon)$$

这里 W 是对角阵, 满足 AW^{-1} 的每列模长为 1。

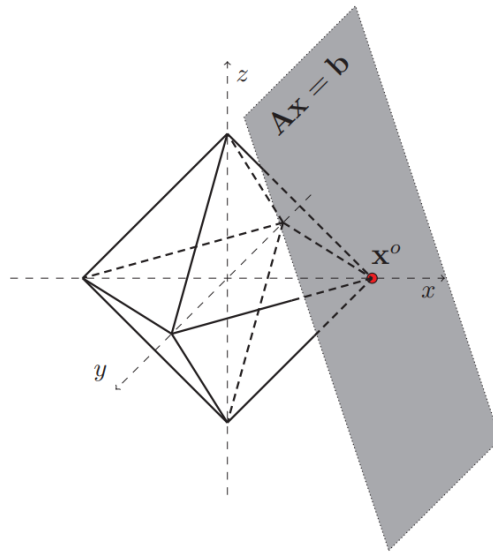


图 7.1: 一范数最优化示意图

值得注意的是, 凸松弛后的问题与原问题并不等价, 且需要一定的条件才能得到解的包含关系。尽管如此, 两个问题仍然有密切的关联, 我们从两个角度说明。首先, 图7.1展示了三维空间中 $Ax = b$ 的平面与一范数的等高线。可以看出, 由于一范数相等对应的区域是由超平面围成的, 在满足 $Ax = b$ 时的最优值很可能在其顶点/棱处, 而这些点即对应若干分量为 0 的点。其次, $\min \|x\|_1$ 可以看作分别要求每个分量正负的 2^n 个线性规划问题求解后取它们的最小值。根据线性规划知识, 这些中的每个解都

在极点处，而零范数最小的点一定是极点，于是最终的最优解很可能在零范数最小的点上。

松弛后问题的一个重要的情况是：

定义 7.3 (基追踪)

形如

$$\min \|x\|_1 \quad \text{s. t. } Ax = b \quad (BP)$$

的最优化问题称为基追踪 [basis pursuit, BP]，其中 A 每列模长为 1。



由于令 $w = Wx$ 后问题 (P_1) 即可化为基追踪，松弛后的零范数问题均可等价于对应的基追踪。

其具有不同的容差形式，如

$$\min \|Ax - b\|^2 \quad \text{s. t. } \|x\|_1 \leq \tau \quad (BP_\tau)$$

$$\min \|x\|_1 + \frac{\mu}{2} \|b - Ax\|^2 \quad (BP_\mu)$$

$$\min \|x\|_1 \quad \text{s. t. } \|b - Ax\| \leq \delta \quad (BP_\delta)$$

练习 7.1 证明以上松弛后的所有变种都是凸优化问题。

解 由于一范数、二范数是凸函数，且凸函数复合线性函数、正线性组合后仍然凸，可知以上均满足凸优化条件。

基追踪可以推广为任何一组基 Ψ 下的追踪问题

$$\min \|s\|_1 \quad \text{s. t. } A\Psi s = b$$

或是一般的

$$\min \|\mathcal{L}x\|_1 \quad \text{s. t. } Ax = b$$

这里 \mathcal{L} 是某个变换，如 DCT、小波变换等。又或者，令 $\mathcal{G}_1, \dots, \mathcal{G}_s$ 构成 $\{1, 2, \dots, n\}$ 的一个划分，有组稀疏 [group sparsity] (或称联合稀疏 [joint sparsity]) 优化问题

$$\min \sum_{i=1}^s w_s \sqrt{\sum_{j \in \mathcal{G}_s} x_j^2} \quad \text{s. t. } Ax = b$$

此外，有时还会采用一些不等式约束，如 $x \geq 0, l \leq x \leq u, Qx \leq c$ 等以控制结果。

7.1.2 收缩与紧邻线性算法

为了介绍一范数优化问题的算法，我们先引入一个重要的函数：

定义 7.4 (收缩函数)

对任何 $x \in \mathbb{R}, \tau > 0$ ，记

$$\text{shrink}(x, \tau) = \max(|x| - \tau, 0) \text{sign}(x) = \begin{cases} x - \tau & x > \tau \\ 0 & -\tau \leq x \leq \tau \\ x + \tau & x < -\tau \end{cases}$$

而对向量或矩阵作用则视为对每个分量/元素分别作用。



由于一范数问题不存在光滑性，直接求解往往是困难的。不过，在一些特殊情况下，还是可以得到解析的直接结果。考虑如下的 **Moreau-Yosida 正则化问题**：

$$\min_x r(x) + \frac{1}{2\tau} \|x - z\|^2$$

这里 $\tau > 0$, r 是非负函数，常见如一范数、二范数等。

 **练习 7.2** 试求：

1.

$$\operatorname{argmin}_x \|x\|_1 + \frac{1}{2\tau} \|x - z\|^2$$

2.

$$\operatorname{argmin} \|x\|_2 + \frac{1}{2\tau} \|x - z\|^2$$

3.

$$\operatorname{argmin}_X \|X\|_* + \frac{1}{2\tau} \|X - Z\|_F^2$$

解

1. 由于此问题中 x_i 互相独立，可以独立看待 $|x_i| + \frac{1}{2\tau}(x_i - z_i)^2$ ，得到结果即为 $\operatorname{shrink}(z, \tau)$ 。

2. 由光滑性可以直接计算梯度求解，当 $z = 0$ 时 $x = 0$ ，否则 $x = \frac{\operatorname{shrink}(\|z\|, \tau)}{\|z\|} z$ 。

3. 令 X 奇异值分解为 $U\Sigma V$ ，由于对正交阵 U 计算知 $\|UA\|_F = \|A\|_F$ ，可化为第一问的情况，得到结果为 $U \operatorname{shrink}(\Sigma, \tau) V^T$ 。



对酉不变 [unitary-invariant] 优化问题，即乘酉矩阵/正交矩阵不改变值或约束成立性的优化问题，一般都能通过奇异值分解转化为向量问题。

对于更一般的问题 $\min_x r(x) + f(x)$ ，这里 f 为数据保真项 [data fidelity term]， r 同上为正则化项 [regularization term]，我们希望能化为 Moreau-Yosida 的形式，这样无论对一范数还是二范数正则化都可以解析求解了。这样的做法就称为紧邻线性算法 [prox-linear algorithm]：

算法 7.5 (紧邻线性算法-迭代)

$$x_{k+1} = \operatorname{argmin}_x \left\{ r(x) + \frac{1}{2\delta_k} \|x - x_k + \delta_k \nabla f(x_k)\|^2 \right\}$$

其中 δ_k 为步长因子。



证明 计算得 x_{k+1} 可以写成 (argmin 内添加常数不影响)

$$\operatorname{argmin}_x \left\{ r(x) + f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2\delta_k} \|x - x_k\|^2 \right\}$$

其相当于 x_k 附近作一阶泰勒展开同时增加距离控制项，因此可以起到优化结果的作用。

7.1.3 ADMM 算法

最后，我们介绍另一个稀疏优化的重要算法，交替方向乘子法 [Alternating Direction Method of Multipliers, ADMM]。它在机器学习的稀疏对率回归、基追踪、支持向量机等诸多问题上都有应用。

ADMM 针对的是如下的一类问题：

$$\min_{X \in \mathbb{C}^{n \times T}} \mu \|X\|_\alpha + \|AX - B\|_\beta$$

的优化算法，这里第一项是正则化项 ($\mu \geq 0$)，第二项是数据保真项。矩阵范数常取为联合范数，如

$\alpha = (2, 1), \beta = (1, 1)$, 简化期间, 我们考虑 $T = 1$ 的情况, 问题变为简单的向量范数:

$$\min_x \mu \|x\|_p + \|Ax - b\|_q$$

进一步将问题转化为

$$\begin{aligned} \min \quad & \mu \|z\|_p + \|y\|_q \\ \text{s. t.} \quad & x - z = 0 \\ & Ax - y - b = 0 \end{aligned}$$

并对给定的控制参数 $\rho > 0$ 构造乘子

$$\begin{aligned} L_\rho(x, y, z, \lambda_y, \lambda_z) &= \mu \|z\|_p + \|y\|_q + \text{Re}(\lambda_z^T(x - z) + \lambda_y^T(Ax - y - b)) + \frac{\rho}{2}(\|x - z\|^2 + \|Ax - y - b\|^2) \\ &= \mu \|z\|_p + \|y\|_q + \frac{\rho}{2}(\|x - z + u^z\|^2 + \|Ax - y - b + u^y\|^2) + C \end{aligned}$$

这里 $u^y = \frac{\lambda_y}{\rho}, u^z = \frac{\lambda_z}{\rho}$, C 是某常数。此处的乘子与通常说的乘子并不相同, 不过可以看出, 当 $\rho \rightarrow \infty$ 时, 优化问题即为原问题, 因此最小化乘子是有意义的。而为对乘子进行最小化, 可以采用序贯最小化的思路, 即依次更新每个变量使得其为当前最小:

算法 7.6 (ADMM-迭代)

给定 $\rho > 0$ 与步长 $\gamma > 0$, 已知当前的 $x_k, y_k, z_k, u_k^y, u_k^z$, 则一次迭代为

$$\begin{aligned} x_{k+1} &= \operatorname{argmin}_x \left\{ \frac{\rho}{2}(\|x - z_k + u_k^z\|^2 + \|Ax - y_k - b + u_k^y\|^2) \right\} \\ y_{k+1} &= \operatorname{argmin}_y \left\{ \|y\|_q + \frac{\rho}{2}\|Ax_{k+1} - y - b + u_k^y\|^2 \right\} \\ z_{k+1} &= \operatorname{argmin}_z \left\{ \mu \|z\|_p + \frac{\rho}{2}\|x_{k+1} - z + u_k^z\|^2 \right\} \\ u_{k+1}^y &= u_k^y + \gamma(Ax_{k+1} - y_{k+1} - b) \\ u_{k+1}^z &= u_k^z + \gamma(x_{k+1} - z_{k+1}) \end{aligned}$$

注意到 x 的更新是可以精确求解的正定二次函数, 而 y, z 都化为了 Moreau-Yosida 正则化的形式, 一般可以求得解析解。

7.2 随机梯度法

7.2.1 有监督分类问题

最优化是机器学习的支柱之一, 而大规模机器学习的训练数据量和参数量都很大, 导致传统的非线性优化技术遇到困难。在引入本节与下一节的重点内容: 随机算法 [stochastic algorithms] 前, 我们需要简要了解机器学习中的问题背景。

我们考虑的重点在于如下的有监督分类问题:

定义 7.7 (有监督分类、泛化误差)

如下的最优化问题称为有监督分类 [supervised classification]:

给定样本空间 \mathcal{X} 、有限个标签组成的输出空间 \mathcal{Y} ，存在样本到标签的映射 $y: \mathcal{X} \rightarrow \mathcal{Y}$ ，求

$$\min_{h \in \mathcal{H}} R(h)$$

这里 \mathcal{H} 为一族 $\mathcal{X} \rightarrow \mathcal{Y}$ 的函数 (即可行的预测函数) 组成的空间， $R(h)$ 为期望风险 $E[I_{h(x)=y(x)}]$ ， E 为数学期望， I 为指标函数，当 $h(x) = y(x)$ 时为 1，否则为 0。

实际情况下，由于真实映射 $y(x)$ 未知，在给定 n 个样本点 (x_i, y_i) (称为训练集 [training set]) 时，需要用给定样本点上的经验风险 $R_n(h) = \frac{1}{n} \sum_i I_{h(x_i)=y_i}$ 近似 $R(h)$ 。即直接的优化问题为

$$\min_{h \in \mathcal{H}} \left\{ \frac{1}{n} \sum_i I_{h(x_i)=y_i} \right\}$$

期望风险 [expected risk] 与经验风险 [empirical risk] 的差距 $|R(h) - R_n(h)|$ 称为泛化误差 [Generalization Error]，一般绝对值内为正。



机器学习中损失 [loss] 与风险 [risk] 基本同义，讲义统一用风险一词。

之所以考虑这一类问题，是因为机器学习中的更多无监督或其他问题往往都能转化到有监督分类的优化模型，例如：

- 回归问题，即输出空间 \mathcal{Y} 是连续的 (如 \mathbb{R})，尽管解决方法会与分类有诸多差异，最终需要处理的优化模型是类似的，有监督的分类问题与回归问题统称为有监督学习 [supervised learning]。
- 深度强化学习，例如 DQN 网络是以同环境交互获得样本，再回归求解 Bellman 等式。
- 生成对抗网络 [Generative Adversarial Network, GAN] 由生成器与判别器组成，并交替训练，每部分训练均为有监督分类，标签即样本为真实数据还是生成数据。

回到对有监督分类的分析。容易看出，函数族 \mathcal{H} 的选取会显著影响优化的过程与结果，一般来说，需要符合三个要求：

1. 足够的容量： \mathcal{H} 中应存在损失较低的函数，避免欠拟合 [underfitting]。这可以通过选取足够大的函数族或利用先验知识实现。
2. 低泛化误差： \mathcal{H} 中的函数泛化误差应不能太大。样本点越多，泛化误差就越小，而由于过拟合 [overfitting]，函数族越大，可能的泛化误差就越大。
3. 训练有效性：寻找 \mathcal{H} 中最优函数的问题应该易于求解。一般来说，函数族与训练集越大，求解就越困难。

关于泛化误差，通过大数律与 Hoeffding 不等式有如下的结论：

命题 7.8 (特定函数泛化误差)

对任何函数 $h: \mathcal{X} \rightarrow \mathcal{Y}$ ，在随机取样下其泛化误差满足

$$|R(h) - R_n(h)| \leq \sqrt{\frac{1}{2n} \ln \frac{2}{\eta}}$$

的概率至少为 $1 - \eta$ 。



而为了获取泛化误差对 \mathcal{H} 的界，一般需要一致大数定律 [uniform laws of large numbers] 与对 VC 维度 [Vapnik-Chervonenkis dimension] 的分析。粗略来说，一个函数族的 VC 维度是指使该函数族中所有函数都失效的最小样本量，用于刻画函数族的容量。

🔥 **练习 7.3** 考虑 \mathbb{R}^2 上所有点的二分类, 已知其真实解是 $\text{label}(x,y) = \begin{cases} 1 & y \geq f(x) \\ -1 & y < f(x) \end{cases}$, 其中 $f(x)$ 为某多项式。求这个问题中取 f 为所有线性多项式形成的函数空间 \mathcal{H} 的 VC 维度。

解 假设已知三个样本, $(1,0), (-1,0)$ 的标签为 1, $(0,0)$ 的标签为 -1, 则不存在任何线性多项式能满足要求。而对任何两个样本, 都可以找一条线使得一个在上一个在下, 因此 VC 维度即为 3。

分析 VC 维度即有定理:

命题 7.9 (函数族泛化误差)

对一族函数 \mathcal{H} , 设其 VC 维度为 $d_{\mathcal{H}}$, 则存在 $c > 0$ 使得随机取样下其泛化误差满足

$$\sup_{h \in \mathcal{H}} |R(h) - R_n(h)| \leq c \sqrt{\frac{1}{2n} \ln \frac{2}{\eta} + \frac{d_{\mathcal{H}}}{n} \ln \frac{n}{d_{\mathcal{H}}}}$$

的概率至少为 $1 - \eta$ 。



7.2.2 函数族的选择

出于方便优化与估计泛化误差的要求, 直接选取一族函数往往是不合适的, 与之相对, 我们可以采取一种**结构**, 例如先选择一个偏好泛函 Ω , 将其每个函数映射到一个值, 再记 $\mathcal{H}_C = \{h \mid \Omega(h) \leq C\}$ 。当超参数 [hyperparameter] C 增大时, \mathcal{H}_C 中的函数增多, 优化后对应的估计误差一般会减小, 而泛化误差则会增大, 常出现的情况如图 7.2。

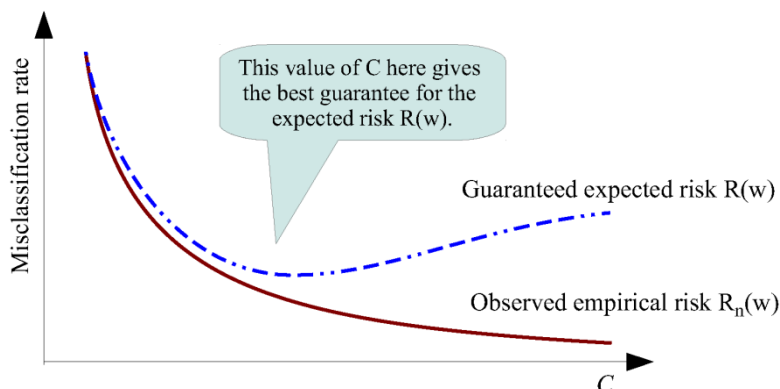


图 7.2: 分类错误率随 C 的变化

我们往往偏好更光滑的函数, 例如可以选取方差或二阶导信息等, 而若 h 由一个参数向量确定, 偏好泛函则可以设置为向量的一范数或二范数。因此, 偏好函数也称为正则化项。也可考虑正则化后的经验误差作为结构, 即 $\mathcal{H}_C = \{h \mid R_n(h) + \lambda \Omega(h) \leq C\}$, 这里 λ 也为超参数。

为了避免通过复杂的理论方法推导泛化误差, 有一个简单的估算泛化误差的方式: 不将已知标签的全部数据作为训练集, 而是分成三部分: 训练集、验证集 [validation set] 与测试集 [testing set]。具体来说, 先对不同的 C 产生的 \mathcal{H}_C 在训练集上优化出最小的经验风险 R_n , 产生一些候选函数。接着, 利用验证集估算这些函数的期望风险 (由于验证集并不在训练中, 其上的经验风险可以用于估算期望风险), 并取出候选中期望风险最小的函数。最后, 用测试集对选出的函数的期望风险进行最终测试。

为方便接下来的推导, 假设 h 有确定的形式, 且被一个参数向量 $w \in \mathbb{R}^d$ 所唯一确定, 即可写成 $h_w(x)$ (例如线性函数空间可写成 $h_{a,b}(x) = a^T x + b$), 并假设输入、输出空间都是连续向量, 维数分别为 d_x, d_y 。



输入与输出若为离散值，也可以直接看成连续值向量，例如若有三个标签，可以将输出看成三维空间中 $(1, 0, 0), (0, 1, 0), (0, 0, 1)$ 三个点，并将 h 得到的三维输出按照接近程度转化为三种标签的一个。

为检验结果，需要定义损失函数 $l: \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \rightarrow \mathbb{R}$ ，刻画输出与真实间的误差。对样本 $\xi = (x, y)$ ，其误差记为 $f_w(\xi) = l(h_w(x), y)$ 。这样，期望风险即为 $R(w) = E_\xi[f_w(\xi)]$ ，而对一系列样本 ξ_i 的经验风险即为 $R_n(w) = \frac{1}{n} \sum_i f_w(\xi_i)$ 。由于实际上样本给定而 w 可以优化，将 $f_w(\xi_i)$ 看作 w 的函数，记为 $f_i(w)$ 有时更加方便。

7.2.3 随机梯度法

首先回忆普通的梯度方法。在一次抓取一批 [batch] 样本 ξ_1, \dots, ξ_n 时，对整体进行一次梯度下降的方法是

$$w_{k+1} = w_k - \alpha_k \nabla R_n(w_k) = w_k - \frac{\alpha_k}{n} \sum_{i=1}^n \nabla f_i(w_k)$$

这里 α_k 为步长因子。

但是，对于复杂的模型，计算多次梯度的开销是很大的，所以，我们采用随机梯度法 [Stochastic Gradient method, SG]:

算法 7.10 (随机梯度法-基础迭代)

对每次抓取的样本 ξ_1, \dots, ξ_n ，先抽取一个下标 i_k ，再进行迭代

$$w_{k+1} = w_k - \alpha_k \nabla f_{i_k}(w_k)$$



虽然这个想法非常简单，但其正确性并不显然。总的来说，由于这样抽取后选择的方向的期望是下降方向 $\nabla R_n(w_k)$ ， w_k 期望于下降至 R_n 的最小点。后续我们会讨论利用降噪方法与二阶信息改进随机梯度法，而本节则会关注对方法本身的分析。除了基础迭代以外，自然也可以抽取多个下标，这也在我们分析的范畴内：

算法 7.11 (随机梯度法-多样本迭代)

对每次抓取的样本 ξ_1, \dots, ξ_n ，抽取 n_k 个下标，记为 $\{k, 1\}, \dots, \{k, n_k\}$ ，再进行迭代

$$w_{k+1} = w_k - \alpha_k \frac{1}{n_k} \sum_{i=1}^{n_k} \nabla f_{k,i}(w_k)$$

或

$$w_{k+1} = w_k - \alpha_k \frac{1}{n_k} H_k \sum_{i=1}^{n_k} \nabla f_{k,i}(w_k)$$

这里 H_k 为对称正定阵，可由牛顿法或拟牛顿法产生。



方便起见，将三种迭代步都记为 $w_{k+1} = w_k - \alpha_k g(w_k, \xi_k)$ ，后两种算法下 ξ_k 是一列样本。

为了证明 SG 方法的收敛，需要先保证具有一定的光滑性质，即：

定义 7.12 (Lipschitz 连续、梯度 Lipschitz 函数)

称一个映射 $\phi: D \rightarrow \mathbb{R}^n$ 具有界为 L 的 Lipschitz 连续性，若其满足

$$\|\phi(w) - \phi(\bar{w})\| \leq L \|w - \bar{w}\|, \forall w, \bar{w} \in D$$

而若一个函数 F 在定义域上可微且其梯度 Lipschitz 连续, 则称其为 (界为 L 的) 梯度 Lipschitz 函数。



将 $R_n(w)$ 记为 $F(w)$, 则有结论:

定理 7.13 (梯度 Lipschitz-SG 迭代下降性)

在 $F(w)$ 有界为 L 的梯度 Lipschitz 连续性时, 函数值下降有结论 (这里 E_{ξ_k} 表示 ξ_k 随机抽取下的期望)

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \nabla F(w_k)^T E_{\xi_k}[g(w_k, \xi_k)] + \frac{\alpha_k^2 L}{2} E_{\xi_k}[\|g(w_k, \xi_k)\|^2] + O(\alpha_k^3)$$



证明 对 $F(w_{k+1}) - F(w_k)$ 作二阶泰勒展开后代入计算对比原式, 发现只需证明 $x^T \nabla^2 F(w)x \leq L\|x\|^2$ 对任何 x, w 成立。若否, 假设对 w_0, x_0 不成立, 则考虑泰勒展开

$$\nabla F(w_0 + \alpha x_0) - \nabla F(w_0) = \alpha \nabla^2 F(w_0)x_0 + O(\alpha^2)$$

当 α 充分小时即有

$$\alpha x_0^T (\nabla F(w_0 + \alpha x_0) - \nabla F(w_0)) = \alpha^2 x_0^T \nabla^2 F(w_0)x_0 + O(\alpha^3) > L\|\alpha x_0\|^2$$

记 $y = \alpha x_0$, $z = \nabla F(w_0 + \alpha x_0) - \nabla F(w_0)$ 。由于 $y^T z > L\|y\|^2$, 又由柯西不等式 $y^T z < \|y\|\|z\|$ 即得 $\|z\| > L\|y\|$, 与 Lipschitz 条件矛盾。

直观来看, 等概率随机时的任何一种迭代步, 都有 $E_{\xi_k}[g(w_k, \xi_k)]$ 为 $\nabla F(w_k)$ 或 $H_k \nabla F(w_k)$, 于是非驻点处只要 α_k 充分小, 就能保证函数值期望的下降。由此过程对任何一次迭代都成立, $E_{\xi_1, \dots, \xi_{k-1}}[w_k]$ 一定是逐渐下降的, 这就说明了收敛性。

若作出进一步的假设, 还有更好的结论:

定理 7.14 (二阶矩条件-SG 迭代下降性)

在 $F(w)$ 有界为 L 的梯度 Lipschitz 连续性时, 进一步要求 (Var_{ξ_k} 为方差):

- 存在 $\mu_G \geq \mu > 0$, 使得对一切 k 有

$$\nabla F(w_k)^T E_{\xi_k}[g(w_k, \xi_k)] \geq \mu \|\nabla F(w_k)\|^2$$

$$E_{\xi_k}[\|g(w_k, \xi_k)\|] \leq \mu_G \|\nabla F(w_k)\|$$

- 存在 $M \geq 0$ 与 $M_V \geq 0$, 使得对一切 k 有

$$\text{Var}_{\xi_k}[\|g(w_k, \xi_k)\|] \leq M + M_V \|\nabla F(w_k)\|^2$$

记 $M_G = M_V + \mu_G^2$, 则函数值下降有结论

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\left(\mu\alpha_k - \frac{\alpha_k^2 L}{2} M_G\right) \|\nabla F(w_k)\|^2 + \frac{\alpha_k^2 L}{2} M + O(\alpha_k^3)$$



证明 根据方差定义计算知 $E_{\xi_k}[\|g(w_k, \xi_k)\|^2] \leq M + M_G \|\nabla F(w_k)\|^2$, 从而直接由上一定理代入不等式化简即可。

7.2.4 收敛性结论

根据以上两个关于下降性的定理，只要给函数补充适当的条件，就能得到收敛性的结论。首先，我们假设迭代序列 $\{w_k\}$ 始终在 F 有下界的某开集内，最优值存在，记为 F_* 。


定义 7.15 (强凸函数)


对函数 $F: \mathbb{R}^n \rightarrow \mathbb{R}$ ，若其可微，且存在 $c > 0$ 使得

$$\forall x, y \in \mathbb{R}^n, F(y) \geq F(x) + \nabla F(x)^T(y - x) + \frac{c}{2}\|y - x\|^2$$

则称 F 是参数为 c 的强凸函数。



 对比凸函数的一阶条件可知强凸函数一定是严格凸函数，从而其最优解若存在则唯一。

 **练习 7.4** 证明若函数 F 是参数为 c 的强凸函数，且在 w^* 取到最小值 F_* ，则有 $F(w) - F_* \leq \frac{1}{2c}\|\nabla F(w)\|^2$ 。

解 根据强凸函数的条件配方有

$$F(w) - F_* \leq -\nabla F(w)^T(w^* - w) - \frac{c}{2}\|w^* - w\|^2 = -\left\|\frac{1}{\sqrt{2c}}\nabla F(w) + \sqrt{\frac{c}{2}}(w^* - w)\right\|^2 + \frac{1}{2c}\|\nabla F(w)\|^2$$

从而得证。

定理 7.16 (强凸函数-固定步长 SG 算法收敛性)

若 F 是参数为 c 的强凸函数，且 SG 迭代满足二阶矩条件。假设步长 α_k 恒定为 $\bar{\alpha}$ ，满足

$$0 < \bar{\alpha} \leq \min\left\{\frac{\mu}{LM_G}, \frac{1}{c\mu}\right\}$$

则迭代过程中 (这里 E 代表每一步都随机抽取下的期望，对 w_k ，由于只有这之前的影响，因此为 $E_{\xi_1, \dots, \xi_{k-1}}$)

$$E[F(w_k) - F_*] \leq \frac{\bar{\alpha}LM}{2c\mu} + (1 - c\mu\bar{\alpha})^{k-1}\left(F(w_1) - F_* - \frac{\bar{\alpha}LM}{2c\mu}\right)$$



证明 根据二阶矩条件下的结论与上方练习即有 (根据范围可得 $\mu\bar{\alpha} - \frac{\bar{\alpha}^2L}{2}M_G > \frac{\mu\bar{\alpha}}{2}$)

$$\begin{aligned} E_{\xi_k}[F(w_{k+1})] - F(w_k) &\leq -\frac{\mu\bar{\alpha}}{2}\|\nabla F(w_k)\|^2 + \frac{\bar{\alpha}^2L}{2}M + O(\bar{\alpha}^3) \\ &\leq -c\mu\bar{\alpha}(F(w_k) - F_*) + \frac{\bar{\alpha}^2L}{2}M + O(\bar{\alpha}^3) \end{aligned}$$

同加 $F(w_k) - F_*$ 后取期望变形即

$$E[F(w_{k+1}) - F_*] \leq (1 - c\mu\bar{\alpha})E[F(w_k) - F_*] + \frac{\bar{\alpha}^2LM}{2} + O(\bar{\alpha}^3)$$

从而忽略高阶小量 $O(\bar{\alpha}^3)$ 展开计算知

$$\begin{aligned} E[F(w_{k+1}) - F_*] &\leq (1 - c\mu\bar{\alpha})^k E[F(w_1) - F_*] + \frac{\bar{\alpha}^2LM}{2} \sum_{n=0}^{k-1} (1 - c\mu\bar{\alpha})^n \\ &= (1 - c\mu\bar{\alpha})^k E[F(w_1) - F_*] + \frac{\bar{\alpha}LM}{2c\mu} (1 - (1 - c\mu\bar{\alpha})^k) \\ &= \frac{\bar{\alpha}LM}{2c\mu} + (1 - c\mu\bar{\alpha})^k \left(F(w_1) - F_* - \frac{\bar{\alpha}LM}{2c\mu}\right) \end{aligned}$$

这就是结论的形式。

理论上说，取使 $c\mu\bar{\alpha} < 1$ 的 $\bar{\alpha}$ ，最终误差即会趋于 $\frac{\bar{\alpha}LM}{2c\mu}$ 。根据 M 的定义， $g(w_k, \xi_k)$ 的方差能被

$\|\nabla F(w_k)\|^2$ 线性控制, 即有 $M = 0$, 可以趋于最优解, 否则, 必须选取较小的步长来保证结果在容差范围的收敛性。自然地, 实践中的梯度下降方法会在接近最优解时减少步长, 而关于此时会有结论:

定理 7.17 (强凸函数-变步长 SG 算法收敛性)

若 F 是参数为 c 的强凸函数, 且 SG 迭代满足二阶矩条件。假设步长 $\alpha_k = \frac{\beta}{\gamma+k}$ 满足 $\beta > \frac{1}{c\mu}, \gamma > 0, \alpha_1 \leq \frac{\mu}{LM_G}$, 则迭代过程中

$$E[F(w_k) - F_*] \leq \frac{\nu}{\gamma+k}, \nu = \max \left\{ \frac{\beta^2 LM}{2(\beta c \mu - 1)}, (\gamma+1)(F(w_1) - F_*) \right\}$$

证明 利用归纳法, 首项由于 $\nu \geq (\gamma+1)(F(w_1) - F_*)$ 知成立, 递推中与上一定理相同忽略 $O(\alpha_k^3)$ 有

$$E[F(w_{k+1}) - F_*] \leq (1 - \mu\alpha_k c)E[F(w_k) - F_*] + \frac{\alpha_k^2 LM}{2}$$

继续计算得右侧为 (第一个不等号利用 $\nu \geq \frac{\beta^2 LM}{2(\beta c \mu - 1)}$ 放缩, 参数范围限制了符号)

$$\frac{2(\gamma+k)\nu - 2\mu\beta c\nu + \beta^2 LM}{2(\gamma+k)^2} \leq \frac{2(k+\gamma)\nu - 2\nu}{2(\gamma+k)^2} = \frac{\gamma+k-1}{(\gamma+k)^2}\nu < \frac{\nu}{\gamma+k+1}$$

即得证。

遗憾的是, 一般的机器学习问题中, 遇到的函数往往是非凸的, 极小值与驻点会对算法产生很大的影响。不过, 依然可以从梯度角度刻画出 SG 算法的收敛性:

定理 7.18 (一般函数-固定步长 SG 算法收敛性)

若 SG 迭代满足二阶矩条件, 且步长 α_k 恒定为 $\bar{\alpha}$, 满足 $0 < \bar{\alpha} \leq \frac{\mu}{LM_G}$, 则迭代过程中

$$E \left[\sum_{k=1}^K \|\nabla F(w_k)\|^2 \right] \leq \frac{K\bar{\alpha}LM}{\mu} + \frac{2(F(w_1) - F_*)}{\mu\bar{\alpha}}$$

从而^a

$$\lim_{K \rightarrow \infty} E \left[\frac{1}{K} \sum_{k=1}^K \|\nabla F(w_k)\|^2 \right] \leq \frac{\bar{\alpha}LM}{\mu}$$

^a数列前 n 项平均值的极限称为数列的 Cesaro 平均。

证明 与强凸函数时相同忽略 $O(\bar{\alpha}^3)$ 得到

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\frac{\mu\bar{\alpha}}{2}\|\nabla F(w_k)\|^2 + \frac{\bar{\alpha}^2 L}{2}M$$

作期望后累加即有

$$F_* - F(w_1) \leq E[F(w_{k+1})] - F(w_1) \leq \frac{\mu\bar{\alpha}}{2} \sum_{k=1}^K \|\nabla F(w_k)\|^2 + \frac{K\bar{\alpha}^2 L}{2}M$$

移项变形得结论。

对变步长情况有结论:

定理 7.19 (一般函数-变步长 SG 算法收敛性)

若 SG 迭代满足二阶矩条件, 且步长 $\alpha_k > 0$ 满足

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

则迭代过程中

$$\sum_{k=1}^{\infty} \alpha_k E[\|\nabla F(w_k)\|^2] < \infty$$



证明 由已证的

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\left(\mu\alpha_k - \frac{\alpha_k^2 L}{2} M_G\right) \|\nabla F(w_k)\|^2 + \frac{\alpha_k^2 L}{2} M + O(\alpha_k^3)$$

忽略 $O(\alpha_k^3)$ 取梯度累加, 减去确定收敛的 $\sum_{k=1}^{\infty} \frac{\alpha_k^2 L}{2} M$ 得到下式收敛:

$$-\mu \sum_{k=1}^{\infty} \alpha_k E[\|\nabla F(w_k)\|^2] + \frac{LM_G}{2} \sum_{k=1}^{\infty} \alpha_k^2 E[\|\nabla F(w_k)\|^2] M$$

由于 $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, 必然有 $\lim_{k \rightarrow \infty} \alpha_k = 0$, 因此某项后 $\alpha_k < \frac{\mu}{LM_G}$, 由有限项不影响收敛性可不妨设 $\alpha_1 < \frac{\mu}{LM_G}$, 这时级数成为负项级数, 由收敛满足

$$-\mu \sum_{k=1}^{\infty} \alpha_k E[\|\nabla F(w_k)\|^2] + \frac{LM_G}{2} \sum_{k=1}^{\infty} \alpha_k^2 E[\|\nabla F(w_k)\|^2] M > -\infty$$

而根据刚才的假设

$$-\frac{\mu}{2} \sum_{k=1}^{\infty} \alpha_k E[\|\nabla F(w_k)\|^2] > -\mu \sum_{k=1}^{\infty} \alpha_k E[\|\nabla F(w_k)\|^2] + \frac{LM_G}{2} \sum_{k=1}^{\infty} \alpha_k^2 E[\|\nabla F(w_k)\|^2] M > -\infty$$

即证明了定理中的收敛性。

练习 7.5 证明此时有 (\liminf 表示下极限, 即所有有极限子列的极限下限)

$$\liminf_{k \rightarrow \infty} E[\|\nabla F(w_k)\|^2] = 0$$

解 记数列为 $\{a_n\}$ 。若否, 由于数列恒正, 其下极限 $c > 0$, 则取 $\epsilon = \frac{c}{2}$, 必然存在 N 使得 $n > N$ 时有 $a_n > \epsilon$ (否则能取出一个均 $< \epsilon$ 的子列, 其下极限 $\leq \epsilon < c$, 矛盾), 而由于 $\sum_{k=1}^{\infty} \alpha_k = \infty$, 这意味着

$$\sum_{k=1}^{\infty} \alpha_k a_k \geq \epsilon \sum_{k=1}^{\infty} \alpha_k - \sum_{k=1}^N \alpha_k a_k = \infty$$

矛盾。

除了上面练习中的结论外, 此结论还有更多的推论, 如:

命题 7.20 (一般函数-SG 算法梯度收敛性)

在 SG 算法与步长满足上述条件时, 有

- 记 X_K 为 1 到 K 中随机抽取一个下标的随机变量, 抽到 k 的概率正比于 α_k , 则 $\|\nabla F(w_{X_K})\|$ 在 $K \rightarrow \infty$ 时依概率收敛于 0。
- 进一步假设 F 二阶可微且 $\|\nabla F(w)\|^2$ 作为 w 的函数是梯度 Lipschitz 连续的, 则有

$$\lim_{k \rightarrow \infty} E[\|\nabla F(w_k)\|^2] = 0$$



7.3 批次梯度法

上一节中, 我们证明了原始的随机梯度法在各种情况下的下降与收敛性质。由于随机抽取样本进行的估算可能比起真实的梯度有很大误差 (也即估算的噪声 [noise]), 我们需要用不同的手段减小噪声

以获取更好的结果。增添噪声处理的随机梯度法称为**批次梯度法** [batch gradient method]。接下来主要介绍三种减小噪声的手段：

- **动态采样法** [dynamic sampling method] 通过逐步增加迭代过程中抽取的样本个数 n_k 来实现降噪；
- **梯度聚合法** [gradient aggregation method] 存储之前迭代中的梯度估计值，并在每次迭代中更新其中一部分，接着将搜索方向定义为这些估计值的加权平均值，从而提高搜索方向的质量；
- **迭代平均法** [iterate averaging methods] 通过维护一个优化过程中迭代的平均值以减小噪声。

7.3.1 动态采样法

回顾之前得到的

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \nabla F(w_k)^T E_{\xi_k}[g(w_k, \xi_k)] + \frac{\alpha_k^2 L}{2} E_{\xi_k}[\|g(w_k, \xi_k)\|^2] + O(\alpha_k^3)$$

如果 $E_{\xi_k}[\|g(w_k, \xi_k)\|^2]$ 可以快速下降，噪声就并不会影响收敛。事实上，只要方差 $\text{Var}_{\xi_k}[g(w_k, \xi_k)]$ 减小足够快，就能有良好的收敛性态：

定理 7.21 (强凸函数-方差衰减收敛性)

在 $F(w)$ 有界为 L 的梯度 Lipschitz 连续性与参数为 c 的强凸性时，进一步要求：

- 存在 $\mu_G \geq \mu > 0$ ，使得对一切 k 有

$$\nabla F(w_k)^T E_{\xi_k}[g(w_k, \xi_k)] \geq \mu \|\nabla F(w_k)\|^2$$

$$E_{\xi_k}[\|g(w_k, \xi_k)\|] \leq \mu_G \|\nabla F(w_k)\|$$

- 存在 $M \geq 0$ 与 $\zeta \in (0, 1)$ ，使得对一切 k 有

$$\text{Var}_{\xi_k}[\|g(w_k, \xi_k)\|] \leq M \zeta^{k-1}$$

步长 α_k 恒定为 $\bar{\alpha}$ ，满足

$$0 < \bar{\alpha} \leq \min \left\{ \frac{\mu}{L\mu_G^2}, \frac{1}{c\mu} \right\}$$

则迭代过程中 k 充分大时有

$$E[F(w_k) - F_*] \leq \omega \rho^{k-1}, \omega = \max \left\{ \frac{\bar{\alpha} L M}{c\mu}, F(w_1) - F_* \right\}, \rho = \max \left\{ 1 - \frac{c\mu\bar{\alpha}}{2}, \zeta \right\}$$

证明 由于 $E_{\xi_k}[\|g(w_k, \xi_k)\|^2] \leq M \zeta^{k-1} + \mu_G^2 \|\nabla F(w_k)\|^2$ ，代入可以类似之前得到

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\left(\mu\bar{\alpha} - \frac{\bar{\alpha}^2 L}{2} \mu_G^2\right) \|\nabla F(w_k)\|^2 + \frac{\bar{\alpha}^2 L}{2} M \zeta^{k-1} + O(\bar{\alpha}^3)$$

再根据 $\bar{\alpha} < \frac{\mu}{L\mu_G^2}$ 放缩并省略 $O(\bar{\alpha}^3)$ 有

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\frac{\mu\bar{\alpha}}{2} \|\nabla F(w_k)\|^2 + \frac{\bar{\alpha}^2 L}{2} M \zeta^{k-1}$$

再根据强凸函数性质得

$$E_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -c\mu\bar{\alpha}(F(w_k) - F_*) + \frac{\bar{\alpha}^2 L}{2} M \zeta^{k-1}$$

同加并取期望有

$$E[F(w_{k+1}) - F_*] \leq (1 - c\mu\bar{\alpha})E[F(w_k) - F_*] + \frac{\bar{\alpha}^2 L}{2} M \zeta^{k-1}$$

利用归纳法, 第一项满足要求, 若 $E[F(w_k) - F_*] \leq \omega \rho^{k-1}$, 则

$$E[F(w_{k+1}) - F_*] \leq (1 - c\mu\bar{\alpha})\omega\rho^{k-1} + \frac{\bar{\alpha}^2 L}{2} M \zeta^{k-1}$$

由于 $\zeta \leq \rho, \frac{\bar{\alpha}LM}{c\mu} \leq \omega$, 即有

$$E[F(w_{k+1}) - F_*] \leq (1 - c\mu\bar{\alpha})\omega\rho^{k-1} + \frac{c\mu\bar{\alpha}\omega}{2} M \rho^{k-1} = \left(1 - \frac{c\mu\bar{\alpha}}{2}\right)\omega\rho^{k-1} \leq \omega\rho^k$$



在实践中, 根据这些量通常的范围, 步长 $\bar{\alpha}$ 的取值一般是合理的。

为了说明这个定理与动态采样的关系, 我们先考虑如下的问题:

练习 7.6 已知 n 个数 x_1, \dots, x_n , 记随机变量 X_k 为从它们之中等概率随机抽取 k 个不同的数的均值, 求 $\text{Var}(X_k)$ 。

解 由于同平移不影响问题结论, 可不妨设这些数均值为 0, 则抽取 x_{i_1}, \dots, x_{i_k} 的概率为 $\frac{1}{C_n^k}$, 误差为 $\frac{\sum_{t=1}^k x_{i_t}}{k}$ 。于是方差为

$$\frac{1}{C_n^k} \sum_{i_1, \dots, i_k} \left(\frac{x_{i_1} + \dots + x_{i_k}}{k} \right)^2$$

由于乘积中只含有二次项与交叉项, 分析可得其为

$$\frac{1}{k^2 C_n^k} \left(C_{n-1}^{k-1} \sum_i x_i^2 + 2C_{n-2}^{k-2} \sum_{i < j} x_i x_j \right) = \frac{1}{nk} \left(\sum_i x_i^2 + \frac{2(k-1)}{n-1} \sum_{i < j} x_i x_j \right)$$

假定 $\sum_i x_i = 0$ 后, 平方得 $\sum_i x_i^2 + 2 \sum_{i < j} x_i x_j = 0$, 消去交叉项有其为

$$\frac{n-k}{k(n-1)} \frac{\sum_i x_i^2}{n} = \frac{n-k}{k(n-1)} \text{Var}(x)$$

此方差恒小于 $\frac{\text{Var}(x)}{k}$, 且在 n 远大于 k 时逼近 $\frac{\text{Var}(x)}{k}$ 。这说明, 只要我们每次取的样本个数指数增长, 就能使方差指数下降。于是即有:

算法 7.22 (动态采样法-迭代)

初始给定 $\tau > 1$, 对每次抓取的样本 ξ_1, \dots, ξ_n , 抽取 $n_k = \lceil \tau^k \rceil$ 个下标, 记为 $\{k, 1\}, \dots, \{k, n_k\}$, 再进行迭代

$$w_{k+1} = w_k - \alpha_k \frac{1}{n_k} \sum_{i=1}^{n_k} \nabla f_{k,i}(w_k)$$



根据上方分析, 它完全符合方差衰减的条件, 因此在 F 是强凸函数时能以指数速度逼近。但是, 如果每次迭代的运算时间也在以指数形式增长, 这样的结果还有意义吗? 为此, 我们再给出一个结论:

定理 7.23 (动态采样法-收敛速率)

假设动态采样的随机梯度法满足 F 的强凸性与二阶矩性质, 且 $1 < \tau \leq (1 - \frac{c\mu\bar{\alpha}}{2})^{-1}$, 则存在 C, D 使得对任何 ϵ , 达到

$$E[F(w_k) - F_*] \leq \epsilon$$

的误差需要的梯度计算次数不超过 $C \frac{1}{\epsilon} + D$ 次。



证明 根据动态采样法的定义, 更新至 w_{k+1} 需要的运算次数至多为 (最后一项由向上取整产生)

$$\tau + \tau^2 + \dots + \tau^k + k = \frac{\tau^{k+1} - \tau}{\tau - 1} + k$$

另一方面，注意到这里的 τ 与收敛性定理中 $\frac{1}{\zeta}$ 一致，根据范围要求可知 $\rho = \zeta = \frac{1}{\tau}$ ，于是有

$$E[F(w_k) - F_*] \leq \frac{\omega}{\tau^{k-1}}$$

考虑 ϵ 比 $\frac{\omega}{\tau^{k-1}}$ 略小，则必须计算 w_{k+1} 才能保证，代入消去 k 可得达到 ϵ 需要的计算次数至多为

$$\frac{\frac{\omega}{\epsilon} \tau^2 - \tau}{\tau - 1} + \log_{\tau} \frac{\omega}{\epsilon} + 1 = \frac{\tau^2 \omega}{\tau - 1} \frac{1}{\epsilon} + \log_{\tau} \omega - \frac{1}{\tau - 1} + \frac{\ln \frac{1}{\epsilon}}{\ln \tau}$$

由 $x > 0$ 时 $\ln x < x$ 可取 $C = \frac{\tau^2 \omega}{\tau - 1} + \frac{1}{\ln \tau}$, $D = \log_{\tau} \omega - \frac{1}{\tau - 1}$ 。

也即，事实上其可以保证与时间反比的收敛速率。

7.3.2 梯度聚合法

比起每次抓取新的更多样本，梯度聚合法选择利用已经计算出的估计值。我们先给出一个完整的例子，即随机方差衰减梯度 [stochastic variance reduced gradient, SVRG] 算法。

算法 7.24 (SVRG 算法)

1. 给定初始 w_1 ，步长 $\alpha > 0$ 与正整数 m ，令 $k = 1$ 。
2. 计算 $\nabla F(w_k)$ ，并记 $\tilde{w}_1 = w_k$ 。
3. 重复执行 m 次 (记当前次数为 j)：从 1 到 n 中抽取一个下标 i_j ，并更新

$$\tilde{w}_{j+1} = \tilde{w}_j - \alpha (\nabla f_{i_j}(\tilde{w}_j) - (\nabla f_{i_j}(w_k) - \nabla F(w_k)))$$

4. 选择一种方式计算 w_{k+1} ：

- $w_{k+1} = \tilde{w}_{m+1}$
- $w_{k+1} = \frac{1}{m} \sum_{j=1}^m \tilde{w}_{j+1}$
- 从 1 到 m 中抽取一个 j ，记 $w_{k+1} = \tilde{w}_{j+1}$

5. 判定是否终止，若否则 $k = k + 1$ ，回到第二步。



证明 我们先证明 $\tilde{g}_j = \nabla f_{i_j}(\tilde{w}_j) - (\nabla f_{i_j}(w_k) - \nabla F(w_k))$ 构成 $\nabla F(\tilde{w}_j)$ 的无偏估计。

$$E_{i_j}[\tilde{g}_j] = E_{i_j}[\nabla f_{i_j}(\tilde{w}_j)] - E_{i_j}[\nabla f_{i_j}(w_k)] + \nabla F(w_k)$$

于是 $E_{i_j}[\tilde{g}_j] = \nabla F(\tilde{w}_j) - \nabla F(w_k) + \nabla F(w_k)$ ，得证。

另一方面，由于第二项的稳定作用，在 \tilde{w}_j 与 w_k 距离不远时， $\nabla f_{i_j}(\tilde{w}_j) - \nabla f_{i_j}(w_k)$ 不会太大，可以说明 $\text{Var}_{i_j}[\tilde{g}_j]$ 比 $\text{Var}_{i_j}[\nabla f_{i_j}(\tilde{w}_j)]$ 更小，于是如此构造的 \tilde{g}_j 有更好的性质。

这个算法中，我们每次先计算完整的梯度，再以其作为估计对 w_k 进行多次更新，并集成结果。相较计算完整梯度后直接更新，它能在同样的更新范围中达到更精细的更新。其收敛速率有结论：

命题 7.25 (SVRG-收敛速率)

在 SVRG 算法中，若 F 是参数为 c 的强凸函数与界为 L 的梯度 Lipschitz 函数，设步长 α 与内循环迭代次数 m 满足

$$\rho = \frac{1}{1 - 2\alpha L} \left(\frac{1}{m c \alpha} + 2L\alpha \right) < 1$$

则对后两种方式更新的 w_{k+1} 有 $E[F(w_{k+1}) - F(w^*)] \leq \rho E[F(w_k) - F(w^*)]$ ，即 $\{w_k\}$ 满足线性收敛速率。



值得注意的是，SVRG 一次外循环需要计算 $2m + n$ 个梯度，因此外循环计算开销和直接进行梯度

下降同量级。若能把核心部分 $\nabla F(w)$ 的计算简化, 我们就能获得更高的效率。在计算机中, 一个常用的想法是用空间换时间, 即通过储存已经计算的结果来加快后续结果的计算速度。此处这样的方法称为随机平均梯度 [Stochastic Average Gradient, SAG] 算法, 而我们这里给出其加速版本, SAGA 算法²:

算法 7.26 (SAGA 算法)

1. 给定初始 w_1 , 计算向量 $\tilde{g}_i = \nabla f_i(w_1), i = 1, \dots, n$ 。
2. 步长 $\alpha > 0$, 令 $k = 1$ 。
3. 从 1 到 n 中抽取一个下标 j , 计算

$$g_k = \nabla f_j(w_k) - \tilde{g}_j + \frac{1}{n} \sum_{i=1}^n \tilde{g}_i$$

4. 更新 $\tilde{g}_j = g_k, w_{k+1} = w_k - \alpha g_k$ 。
5. 判定是否终止, 若否则 $k = k + 1$, 回到第二步。



算法中, \tilde{g}_j 记录的是 f_j 最近计算的一次梯度, 并以此作为真实梯度的估计。由于定义, 与上个算法相同可知 g_k 是无偏估计, 并一定程度减小了方差。对其收敛速率则有结论:

命题 7.27 (SAGA-收敛速率)

在 SAGA 算法中, 若 F 是参数为 c 的强凸函数与界为 L 的梯度 Lipschitz 函数, 设步长 $\alpha = \frac{1}{2(cn+L)}$, 则有

$$E[\|w_k - w^*\|^2] \leq \left(1 - \frac{c}{2(cn+L)}\right)^k \left(\|w_1 - w^*\|^2 + \frac{n(F(w_1) - F(w^*))}{cn+L}\right)$$



事实上, SAGA 算法也可以采取其他的初始化策略, 如先用普通随机梯度法进行一些迭代等。虽然此算法在高维时有很大的空间开销, 在一些特殊问题时仍可以简化, 如当 $f_i(w) = \hat{f}(\alpha_i^T w)$ 时, 有 $\nabla f_i(w) = \hat{f}'(\alpha_i^T w) \alpha_i$, 由此只要存储了每个 α_i , 只需要额外保存一个标量 $\hat{f}'(\alpha_i^T w)$ 即可, 这在一些回归模型中常会出现。

虽然上述梯度聚合方法在达到指定误差时的收敛比普通 SG 算法快, 但它们事实上并不明显优于 SG 算法。类似之前对计算时间的分析, 对参数为 c 的强凸函数与界为 L 的梯度 Lipschitz 函数 F , 记 $\kappa = \frac{L}{c}$, 当 n 维时欲达到 ϵ 误差, 直接 SG 算法的梯度计算次数正比于 $\frac{\kappa^2}{\epsilon}$, 而 SAGA 与 SVRG 则正比于 $-(n + \kappa) \ln \epsilon$ 。随着 n 的增大, 梯度聚合方法的计算次数会变得更加多。

7.3.3 迭代平均法

迭代平均缘于一个简单的想法: 由于普通 SG 方法在每次迭代中抽取一些, 若能将这些抽取的结果平均, 亦能减少噪声:

算法 7.28 (迭代平均法-迭代)

给定步长 $\alpha_k > 0$, 每次迭代中, 先计算 $w_{k+1} = w_k - \alpha_k g(w_k, \xi_k)$, 再令最终更新为

$$\tilde{w}_{k+1} = \frac{1}{k+1} \sum_{j=1}^{k+1} w_j$$



²作者原论文中并没有解释最后一个 A 是什么的缩写。

其计算过程与普通 SG 算法完全相同，只是没有选择最新计算出的点，而是选择了所有计算出点的平均。对它的收敛速率有结论：

命题 7.29 (迭代平均-收敛速率)

对符合衰减速率 $O(k^{-a})$, $a \in (0.5, 1)$ 的步长 α_k ，普通 SG 算法迭代平均后的收敛速率满足

$$E[\|w_k - w^*\|^2] = O(k^{-a}), E[\|\tilde{w}_k - w^*\|^2] = O(k^{-a})$$

也即迭代平均法改善了收敛效果。

7.4 随机牛顿法

另一种对 SG 算法的改进方式为利用上 F 的二阶信息。就像梯度类方法利用二阶信息后成为牛顿类方法，利用二阶信息的随机算法称为**随机牛顿法** [stochastic Newton method]。由于牛顿类方法往往在最优解附近有二阶收敛速率，随机牛顿法一般有更好的收敛效果。

我们主要介绍五种随机牛顿方法，值得注意的是，其中的**无海森牛顿法** [Hessian-free Newton] 与**自然梯度法** [natural gradient] 需要批次有一定规模时才有效，而剩下的**对角缩放** [diagonal scaling]、**拟牛顿** [quasi-Newton] 与**高斯-牛顿** [Gauss-Newton] 方法则无需此限制。

7.4.1 无海森牛顿法

一般的牛顿法迭代过程为 $w_{k+1} = w_k + \alpha_k s_k$ ，其中 $\nabla^2 F(w_k) s_k = -\nabla F(w_k)$ 。事实上，我们并不需要精确求解这个方程，只需要进行一些近似求解 (如利用**共轭梯度法**)，只要保证解的接近性，即可满足超线性的收敛速度。在共轭梯度法近似求解的过程中，不需要显式计算出海森阵，只会出现它与向量的乘积，因此称为无海森牛顿法。

另一方面，由于此方法不要求精确求解，对 ∇F 与 $\nabla^2 F$ 都可以抽取一些 f 并平均估算 (一般估算 $\nabla^2 F$ 取的样本个数更少)。当然，取样过多会引起计算速度降低，取样过少则导致估算并不准确，这其中需要一些经验性的选择。

算法 7.30 (采样无海森牛顿算法)

1. 给定初始 w_1 , $\rho \in (0, 1)$, $\eta \in (0, 1)$, $\gamma > 1$, 正整数 m , 令 $k = 1$ 。
2. 从 1 到 n 中抽取一族下标 S_k , 并从 S_k 中抽取一部分 $S_k^{(H)}$ 。计算 (这里 # 代表元素个数)

$$g_k = \nabla f_{S_k}(w_k) = \frac{1}{\#S_k} \sum_{i \in S_k} \nabla f(w_k, \xi_i)$$

$$H_k = \nabla^2 f_{S_k^{(H)}}(w_k) = \frac{1}{\#S_k^{(H)}} \sum_{i \in S_k^{(H)}} \nabla^2 f(w_k, \xi_i)$$

3. 用共轭梯度法求解 $H_k s = g_k$, 直到达到迭代次数上限 m 或误差满足 $\|H_k s + g_k\| \leq \rho \|g_k\|$ 。
4. 令 $\alpha_k = 1$, 若

$$f_{S_k}(w_k + \alpha_k s_k) \leq f_{S_k}(w_k) + \eta \alpha_k g_k^T s_k$$

则放大 $\alpha_k = \gamma \alpha_k$, 取使其能成立的最大可能 $\alpha_k = \gamma^{p_k}$ 作为更新步长。

5. 更新 $w_{k+1} = w_k + \alpha_k s_k$, 判定是否终止, 若否则 $k = k + 1$, 回到第二步。

当噪声较大时, $\#S_k^{(H)}$ 必须取得较大才能避免海森阵导致迭代出现问题, 因此 $\#S_k$ 较大时才可能采用此方法。此外, 虽然对精确计算 ∇F 与 $\nabla^2 F$ 的无海森牛顿法可以证明收敛, 加入采样后并没有办法保证超线性的收敛速率。

更多时候, 问题是非凸的, 这样的搜索很容易出现问题, 因此需要在上方第四步调整为: 用共轭梯度法求解 $H_k s = g_k$, 直到达到迭代次数上限 m 或误差满足 $\|H_k s + g_k\| \leq \rho \|g_k\|$ 或 s_k 是负曲率方向, 即 $s_k^T H_k s_k < 0$ 。当然, 比起处理不正定的海森阵, 我们更希望能维持一个正定或半正定的估计。

7.4.2 随机拟牛顿法

我们先从直接估计海森阵的拟牛顿法说起, 回顾无约束优化的拟牛顿法, 其更新形式满足

$$w_{k+1} = w_k - \alpha_k H_k \nabla F(w_k)$$

其中 H_k 为对 $\nabla^2 F(w_k)^{-1}$ 的估算。

其中常用的 BFGS 方法, 利用对 H_k 的动态更新, 在仅使用一阶信息时通过估计二阶信息在局部达到了超线性收敛速率。然而, 由于 H_k 不具有稀疏性 (即使真实的海森阵是稀疏的), 大规模计算中其存储开销非常大, 我们需要考虑有限记忆策略 [limited memory scheme], 例如不需要显式计算出 H_k 的 L-BFGS 方法。具体来说, 我们记录集合 P , 其中储存着若干对 s 与 y , 并每次根据其中的 s 与 y 计算出 $H_k g$:

算法 7.31 (L-BFGS 迭代)

给定包含 m 对 $(s_0, y_0), \dots, (s_{m-1}, y_{m-1})$ 的集合 P , 则根据 BFGS 迭代产生的矩阵为

$$H_0 = I$$

$$H_{k+1} = \left(I - \frac{y_k s_k^T}{s_k^T y_k} \right)^T H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k} \right) + \frac{s_k s_k^T}{s_k^T y_k}$$

由于每个 $I - \frac{y_k s_k^T}{s_k^T y_k}$ 或 $\frac{s_k s_k^T}{s_k^T y_k}$ 与向量的乘积均可只通过向量运算得到, 展开 $H_m g$ 可得到不含矩阵运算的计算方式。

即使这样, 引入随机后利用采样估算的梯度 $g(w_k, \xi_k)$ 进行的更新

$$w_{k+1} = w_k - \alpha_k H_k g(w_k, \xi_k)$$

也会遇到诸多问题:

1. 理论局限性: 引入随机后, 此方法无法达到线性收敛速率, 与一般的 SG 算法并无量级上的区别。不过, 由于常数上的改进, 随机采样下的拟牛顿法仍然存在意义。可以证明, 当 $H_k \rightarrow \nabla^2 F(w^*)^{-1}$ 时, 其常数较一般随机梯度法更好。
2. 较长的单次迭代时间: 设 m 为 L-BFGS 方法的记忆周期 (通常为 5), d 为计算 $g(w_k, \xi_k)$ 所需的操作次数, 则 $H_k g(w_k, \xi_k)$ 的计算需要 $4md$ 次计算, 也即直接计算 $g(w_k, \xi_k)$ 的 20 倍左右。为解决此问题, 一般估计 $g(w_k, \xi_k)$ 需要采用相对较小的批次, 如设定为 256。
3. 估计海森阵的训练过程: 估算 H_k 需要用到 $g(w_k, \xi_k)$ 与 $g(w_{k+1}, \xi_{k+1})$ 之间的差, 但二者都是估计而成, 可能造成很大的误差。
4. 误差的累计: 与普通拟牛顿法一样每步都对 H_k 进行更新可能没有必要, 或反而引起误差累积导致估计更加不准确。

对后两个问题有不同的解决方法, 例如采取更好的对 y_k 的估计方式。

算法 7.32 (随机拟牛顿法-估计 y_k)

已知 w_{k+1} 与 w_k 时, 有 $s_k = w_{k+1} - w_k$, 进一步假设已选出 S_k 并估计了梯度 $g(w_k, \xi_k) = \nabla f_{S_k}(w_k)$, 可由如下方法估计对应的 $y_k = \nabla F(w_{k+1}) - \nabla F(w_k)$:

1. 令 $y_k = \nabla f_{S_k}(w_{k+1}) - \nabla f_{S_k}(w_k)$;
2. 从 S_k 中抽取一部分 $S_k^{(H)}$, 令 $y_k = \nabla^2 f_{S_k^{(H)}}(w_k) s_k$ 。



前者通过相同采样的方式避免了过大的误差, 后者则根据 $y = Hs$ 的要求计算 y_k , 将梯度计算与海森阵更新解耦 [decouple]。综合估计 y_k 的方法, 我们可以得到完整的随机拟牛顿法:

算法 7.33 (随机拟牛顿法)

1. 给定初始 w_1 , 正整数 m , 恒正的步长序列 α_k 。令 $P = \emptyset$, $k = 1$ 。
2. 抽取 S_k , 并计算

$$g_k = \nabla f_{S_k}(w_k)$$

$$w_{k+1} = w_k - \alpha_k H_k g_k$$

这里 $H_k g_k$ 由 P 中的 (s, y) 通过 L-BFGS 方法得出。

3. 判定是否需要更新 H_k , 若是则继续, 否则令 $k = k + 1$, 回到第二步。
4. 任选一种方式估计 y_k , 并将 (s_k, y_k) 添加到 P 中, 若 P 中数量超过 m , 丢弃其中下标最小的 (s_i, y_i) 。
5. 令 $k = k + 1$, 回到第二步。

**7.4.3 高斯-牛顿法**

虽然随机拟牛顿法避免了对海森阵的复杂计算与存储, 仍然存在更新过程中未必正定导致迭代误差很大的问题。高斯-牛顿方法则对其作出了改进, 即使真实的海森阵非正定, 也能保证估计过程的半正定性。

回到基础定义, 对 $\xi = (\xi_x, \xi_y)$ 与参数 w 产生的预测函数 h_w , 考虑最小二乘误差

$$f_w(\xi) = l(h_w(\xi_x), \xi_y) = \frac{1}{2} \|h_w(\xi_x) - \xi_y\|^2$$

在给定 ξ 时记 h_w 对 w 的 Jacobi 阵为 $\mathcal{J}_{h_\xi}(w)$, 则有

$$h_w(\xi_x) = h_{w_k}(\xi_x) + \mathcal{J}_{h_\xi}(w_k)(w - w_k) + O(\|w - w_k\|^2)$$

于是计算可得 $f_w(\xi)$ 忽略二阶项近似为

$$\frac{1}{2} \|h_{w_k}(\xi_x) - \xi_y\|^2 + (h_{w_k}(\xi_x) - \xi_y)^T \mathcal{J}_{h_\xi}(w_k)(w - w_k) + \frac{1}{2} (w - w_k)^T \mathcal{J}_{h_\xi}(w_k)^T \mathcal{J}_{h_\xi}(w_k)(w - w_k)$$

若将其看作对 f 的展开, 海森阵即为 $\mathcal{J}_{h_\xi}(w_k)^T \mathcal{J}_{h_\xi}(w_k)$ 。由此思想, 我们可以作近似:

定义 7.34 (高斯-牛顿阵)


对一族下标 $S_k^{(H)}$, 其对最小二乘误差产生的高斯-牛顿阵为

$$G_{S_k^{(H)}}(w_k) = \frac{1}{\#S_k^{(H)}} \sum_{i \in S_k^{(H)}} \mathcal{J}_{h_{\xi_i}}(w_k)^T \mathcal{J}_{h_{\xi_i}}(w_k)$$

对一般的凸误差函数 $l(y_1, y_2)$, 高斯-牛顿阵则为

$$G_{S_k^{(H)}}(w_k) = \frac{1}{\#S_k^{(H)}} \sum_{i \in S_k^{(H)}} \mathcal{J}_{h_{\xi_i}}(w_k)^T \nabla_h^2 l(h, \cdot) \mathcal{J}_{h_{\xi_i}}(w_k)$$



 **练习 7.7** 对一般情况, 证明其半正定, 且对 $\lambda > 0$ 有 $\lambda I + G_{S_k^{(H)}}(w_k)$ 正定。

解 由凸函数二阶条件可知 $\nabla_h^2 l(h, \cdot)$ 半正定, 从而根据定义可知 $G_{S_k^{(H)}}(w_k)$ 半正定, 进一步得到 $\lambda I + G_{S_k^{(H)}}(w_k)$ 正定。

由此, 一般增加 λI 后得到正定的矩阵作为海森阵的近似, 从而进行迭代。此方法的计算成本取决于预测函数的维度。值得一提的是, 在机器学习中, 计算随机梯度向量 $\nabla f_{\xi}(w)$ 通常不需要明确计算 Jacobi 矩阵的所有行, 此外, 还有一些较新的方法能以较低成本解决高斯-牛顿迭代问题。

7.4.4 对角缩放法

接下来, 我们介绍以进一步降低每次迭代运算次数为目的的对角缩放法。在高斯-牛顿法中, 我们虽然能够近似海森阵, 却仍然需要近似其逆才能计算出迭代更新的方向。而对角缩放法的思路则是迭代更新高斯-牛顿阵, 并直接将其逆近似为对角元对应取逆的对角阵, 具体来说为:

算法 7.35 (对角缩放法-迭代)

若每一步取出样本 ξ_k , 对应梯度为 $g(w_k, \xi_k)$, 当前高斯-牛顿阵对角元构成的向量为 G_k , 则迭代过程先计算 w_{k+1} (下标 i 表示第 i 个分量, 对所有分量如此计算):

$$w_{k+1,i} = w_{k,i} - \frac{\alpha}{G_{k,i} + \mu} g(w_k, \xi_k)_i$$

再更新 G_k (下标 ii 即第 i 个对角元):

$$G_{k+1,i} = (1 - \lambda)G_{k,i} + \lambda(\mathcal{J}_{\xi_{k+1}}(w_{k+1})^T \mathcal{J}_{\xi_{k+1}}(w_{k+1}))_{ii}$$

这里 μ 为给定正数, 避免对角元太小, α 为步长因子, $\lambda \in (0, 1)$ 为更新权重。



此算法中, 对于高斯-牛顿阵利用权重衰减的方法进行近似, 且只保存与更新其对角元, 并完全用对角元近似 Hg_k 。比起二阶方法, 它其实更接近对 g_k 进行一定幅度调整的一阶方法。

除了这样直接通过 Jacobi 阵计算对角元与其倒数外, 也可以采用近似计算的方案。例如, 在拟牛顿法中, 可以考虑由当前的 s_k, y_k 迭代更新 H_k 的对角元 $H_{k,i}$:

$$H_{k+1,i} = (1 - \lambda)H_{k,i} + \lambda \text{Proj} \left(\frac{s_{k,i}}{y_{k,i}} \right)$$

这里 Proj 代表向某个正区间 $[a, b]$ 投影, 即将小于 a 的置为 a , 大于 b 的置为 b , 其余不变。然而, 由于 H_k 无法控制, 直接运用这样的方法是噪声较大且难以纠正的, 于是需要寻求更好的办法。

其中一个想法是, 迭代更新海森阵 G_k 而非其倒数 H_k , 再直接计算倒数, 更新方式为

$$G_{k+1,i} = G_{k,i} + \text{Proj} \left(\frac{y_{k,i}}{s_{k,i}} \right)$$

注意对角缩放法的迭代步骤可发现, Proj 的投影保证了实际更新的步长是以 $O(\frac{1}{k})$ 下降的。

7.4.5 自然梯度法

我们最后介绍的二阶方法是自然梯度法, 其想法为在空间 \mathcal{H} 中 (而非参数空间下) 进行梯度下降迭代, 因此得名。

假设参数空间中所有的函数都是密度函数, 即 $\int h_w(x)dx = 1$ 且 $h_w(x) \geq 0$, 由此可以定义 E_{h_w} 等, 并假设其充分正则 (由归一化条件可知此式为 0):

$$\forall t > 0, \int \frac{\partial^t}{\partial w^t} h_w(x) dx = \frac{\partial^t}{\partial w^t} \int h_w(x) dx$$

这里 $\frac{\partial^t}{\partial w^t}$ 指的是对任一种对不同分量求总计 t 阶导数的方式。


我们先定义 KL 散度:


定义 7.36 (KL 散度)

对两个非零的密度函数 h_1, h_2 , 可定义 h_2 对 h_1 的 KL 散度 [Kullback-Leibler divergence] 为

$$D_{KL}(h_1 \| h_2) = E_{h_1} \left[\ln \frac{h_1(x)}{h_2(x)} \right]$$



 注意 $D_{KL}(h_1 \| h_2)$ 未必与 $D_{KL}(h_2 \| h_1)$ 相同。

 练习 7.8 证明 $D_{KL}(h_1 \| h_2) \geq 0$, 且其取到 0 当且仅当 h_1 与 h_2 几乎处处相等。

解 由于 $-\ln x$ 是严格凸函数, 对和为 1 的正数 $\lambda_1, \dots, \lambda_n$ 与正数 x_1, \dots, x_n , 由琴生不等式有

$$\sum_i -\lambda_i \ln x_i \geq -\ln \left(\sum_i \lambda_i x_i \right)$$

且等号成立当且仅当 x_i 均相等。

将其连续化即得, 对满足 $\int f(x)dx = 1$ 的正函数 $f(x)$ 与正函数 $g(x)$, 有

$$\int -f(x) \ln g(x) dx \geq -\ln \int f(x)g(x) dx$$

且等号成立当且仅当 $g(x)$ 几乎处处恒定。


于是有

$$D_{KL}(h_1 \| h_2) = \int -\ln \frac{h_2(x)}{h_1(x)} h_1(x) dx \geq -\ln \int \frac{h_2(x)h_1(x)}{h_1(x)} dx = -\ln 1 = 0$$

由于 h_1, h_2 均为归一化的正函数, $\frac{h_1}{h_2}$ 几乎处处恒定可得只能恒定为 1, 从而得证。

将 $h_{w+\delta w}(x)$ 对 δw 展开到二阶, 可计算得

$$D_{KL}(h_w \| h_{w+\delta w}) = -\delta w^T E_{h_w} [\nabla_w \ln h_w(x)] - \frac{1}{2} \delta w^T E_{h_w} [\nabla_w^2 \ln h_w(x)] \delta w + O(\|\delta w\|^3)$$

 练习 7.9 证明 (省略期望下标 h_w)

$$E[\nabla_w \ln h_w(x)] = 0, -E[\nabla_w^2 \ln h_w(x)] = E[\nabla_w \ln h_w(x) \nabla_w \ln h_w(x)^T]$$

解 根据充分正则性, 写成期望形式, 对 $t=1, 2$ 可知

$$E \left[\frac{1}{h_w(x)} \nabla_w h_w(x) \right] = 0, E \left[\frac{1}{h_w(x)} \nabla_w^2 h_w(x) \right] = 0$$

由第一个式子内部可直接写成 $\nabla_w \ln h_w(x)$ 可知 $E[\nabla_w \ln h_w(x)] = 0$, 而进一步计算可知

$$E[\nabla_w^2 \ln h_w(x)] = E \left[\frac{1}{h_w(x)} \nabla_w^2 h_w(x) - \frac{1}{h_w^2(x)} \nabla_w h_w(x) \nabla_w h_w(x)^T \right]$$

由求和中第一项为 0 即得证。

从而, 记 $G(w) = E[\nabla_w \ln h_w(x) \nabla_w \ln h_w(x)^T]$, 即有

$$D_{KL}(h_w \| h_{w+\delta w}) = \frac{1}{2} \delta w^T G(w) \delta w + O(\|\delta w\|^3)$$

下面研究其如何应用于迭代。

我们考虑每次限制变化范围的优化

$$w_{k+1} = \operatorname{argmin}_w \left\{ F(w) \mid D_{KL}(h_{w_k} \| h_w) \leq \eta_k^2 \right\}$$

根据推导, 此约束可近似成 $\frac{1}{2}(w-w_k)^T G(w_k)(w-w_k) \leq \eta_k^2$ 。将硬约束化为软约束, 可看作优化目标增加项 $\frac{1}{2\alpha_k}(w-w_k)^T G(w_k)(w-w_k)$, α_k 越大则步长越大, 再将 $F(w)$ 近似为 $F(w_k) + \nabla F(w_k)^T(w-w_k)$, 最终变为

$$w_{k+1} = \operatorname{argmin}_w \left\{ \nabla F(w_k)^T(w-w_k) + \frac{1}{2\alpha_k}(w-w_k)^T G(w_k)(w-w_k) \right\}$$

求解可得到:

算法 7.37 (自然梯度法-迭代)

每次迭代中, 给定步长 $\alpha_k > 0$, 抽取一族下标 S_k , 估算 $G(w_k)$ 为 $(\xi_{i,x}$ 代表样本 ξ_i 中的 x)

$$\tilde{G}(w_k) = \frac{1}{\#S_k} \sum_{i \in S_k} \nabla_w \ln h_{w_k}(\xi_{i,x}) \nabla_w \ln h_{w_k}(\xi_{i,x})^T$$

并更新

$$w_{k+1} = w_k - \alpha_k \tilde{G}(w_k)^{-1} \nabla F(w_k)$$



与高斯-牛顿法类似可证 $\tilde{G}(w_k)$ 半正定, 可添加 λI 以保证正定性。

7.5 更多常用算法

7.5.1 动量法

动量法具有如下的迭代形式:

算法 7.38 (动量法-迭代)

给定 $\alpha_k > 0, \beta_k > 0$, 其迭代过程为

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k) + \beta_k (w_k - w_{k-1})$$

这里第二项称为**动量** [momentum] 项³, 当 α_k, β_k 固定为 α, β 时, 动量法迭代称为**重球方法** [heavy ball method], 对部分函数具有线性收敛性, 且收敛速率大于步长固定的最速下降法。注意到, 重球方法中迭代公式可以展开写成

$$w_{k+1} = w_k - \alpha \sum_{j=1}^k \beta^{k-j} \nabla F(w_j)$$

即次更新都是以往梯度的指数平均。

对于二次函数的动量法有性质:

³这是由于它代表了某个带摩擦的二阶常微分方程的离散。

定理 7.39 (动量法-二次终止性)

对严格凸二次函数 F ，若步长满足 (第一步取 $\beta = 0$)

$$(\alpha_k, \beta_k) = \operatorname{argmin}_{\alpha, \beta} F(w_k - \alpha \nabla F(w_k) + \beta(w_k - w_{k-1}))$$

则动量法迭代过程与共轭梯度法完全相同。



证明 记更新空间 $U_k = \{w_k - \alpha \nabla F(w_k) + \beta(w_k - w_{k-1}) \mid \alpha, \beta \in \mathbb{R}\}$ 。

利用归纳，第一步由定义可知一致，而此后，记 g_i, d_i 为共轭梯度法中对应结论，根据共轭方向法性质定理有

$$w_{k+1} = \operatorname{argmin}_w \left\{ F(w) \mid w = w_0 + \sum_{i=0}^k \lambda_i d_i, \lambda_i \in \mathbb{R} \right\}$$

将右侧集合记为 V_k ，而根据共轭梯度法的更新过程，存在 a_k, b_k 使得 $w_{k+1} = w_k + a_k d_k = w_k + a_k(-g_k + b_k d_{k-1})$ ，记共轭梯度法更新空间

$$W_k = \{w_k + a(-g_k + b_k d_{k-1}) \mid a, b \in \mathbb{R}\}$$

由于 $W_k \in V_k$ ，必有

$$(a_k, b_k) = \operatorname{argmin}_{a, b} F(w_k + a(-g_k + b_k d_{k-1}))$$

由于 $\nabla F(w_k) = g_k, w_k - w_{k-1} = a_{k-1} d_{k-1}$ ，迭代未终止时 $a_{k-1} \neq 0$ ，可知 $U_k = W_k$ ，而两者更新均为取空间中最小为 w_{k+1} ，从而得证。

另一个利用动量加速梯度法的思路来自 Nesterov 的迭代方式：

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k + \beta_k(w_k - w_{k-1})) + \beta_k(w_k - w_{k-1})$$

也即先将 w_k 加上 $\beta_k(w_k - w_{k-1})$ ，再进行一次最速下降。这时有结论：

命题 7.40 (Nesterov 迭代-收敛速率)

取 α_k 恒定为某 $\alpha > 0$ ，且 β_k 满足 $\beta_k < 1, \lim_{k \rightarrow \infty} \beta_k = 1$ 。若 F 凸且梯度 Lipschitz 连续，则 Nesterov 迭代的收敛速率为 $O(\frac{1}{k^2})$ 。



比起一般梯度方法的 $O(\frac{1}{k})$ 的收敛速率，此迭代的确改善了量级，但目前尚未有对原因的直观解释。

7.5.2 坐标下降法

坐标下降 [coordinate descent, CD] 法每次只更新一个坐标：

算法 7.41 (坐标下降法-迭代)

在 $w \in \mathbb{R}^d$ 时，给定步长 α_k 与下标 $i_k \in \{1, \dots, d\}$ ，坐标下降法迭代过程为

$$w_{k+1} = w_k - \alpha_k \frac{\partial F(w_k)}{\partial w_{k, i_k}} e_{i_k}$$



也即在一个方向上看作单元函数极值问题。

根据不同的 α_k 与 i_k 选取策略，坐标下降法也有不同形式，例如：

- 精确搜索 α_k ；

- 通过某些标准进行非精确一维搜索确定 α_k ;
- 将 α_k 选作 F 的某个二阶局部近似的最小值点 (这样的方法称为二阶坐标下降法);
- i_k 在 1 到 d 中循环;
- 将 1 到 d 排成随机序列, i_k 在其中选择, 每轮结束重排;
- i_k 在 1 到 d 中随机选择 (含随机性的 i_k 选择收敛效果一般好于循环选择)。

对一般的可微函数, 即使最速下降法可以收敛到驻点, CD 方法也无法保证收敛性。不过, 对一些特殊的 F , CD 方法还是能保证较好的效果:

命题 7.42 (坐标下降法-收敛性)

若 F 是参数为 c 的 d 维强凸函数, 且其梯度的每个分量都是 Lipschitz 连续的, 界分别为 L_1, \dots, L_d , 其中最大值为 \hat{L} , 取步长 α_k 恒定为 $\frac{1}{\hat{L}}$, i_k 从 1 到 d 中等概率随机选择一个, 则

$$E[F(w_{k+1}) - F_*] \leq \left(1 - \frac{c}{\hat{L}d}\right)^k (F(w_1) - F_*)$$

此结论可以通过直接将 F 展开后计算以说明。此时, 其收敛速率是线性的, 而若 d 次坐标下降与一次梯度下降的计算开销接近, CD 方法与最速下降法无论是理论还是实践中的收敛速率都是类似的。

有一类实践中常出现的函数适用 CD 方法, 也即

$$F(w) = \sum_{j=1}^n \tilde{F}_j(x_j^T w) + \sum_{i=1}^d \hat{F}_i(w_i) \quad (7.1)$$

这里第一项中 \tilde{F}_j 为可微函数, 而向量 x_j 一般是稀疏的, 第二项为正则化项, 未必可微。考虑如下的更简单的情况:

$$F(w) = \frac{1}{2} \|Aw - b\|^2 + \sum_{i=1}^d \hat{F}_i(w_i), A = (a_1, a_2, \dots, a_d)$$

这里 a_j 均为列向量。

而计算可得 (这里 ∂m 表示对第 m 个分量)

$$\frac{\partial F(w)}{\partial m} = \frac{1}{2} (Aw - b)^T a_m + \hat{F}'_m(w_m)$$

记 $r_k = Aw_k - b$, 验证可知若 $w_{k+1} = w_k + \beta_k e_{i_k}$, 则 $r_{k+1} = r_k + \beta_k a_{i_k}$, 这时方程可写为

$$0 = \frac{\partial F(w_{k+1})}{\partial i_k} = \frac{1}{2} r_{k+1}^T a_{i_k} + \hat{F}'_{i_k}(w_{k+1, i_k})$$


从而可以近似求解 β_k 得到更新。由于更新 r_k 的代价与 a_{i_k} 中非零分量的个数成正比, 进一步分析可得迭代过程亦与 a_{i_k} 中非零分量的个数成正比, 这样优化一轮后, 相比直接梯度法与 A 中非零分量正比的量级是一致的。

我们还可以从对偶问题出发对 (7.1) 进行优化。若 $\hat{F}_i(a) = \frac{\lambda}{2} a^2$, 则原问题写为

$$F(w) = \frac{1}{n} \sum_{j=1}^n \tilde{F}_j(x_j^T w) + \frac{\lambda}{2} \|w\|^2$$

记 $u_j = x_j^T w$, 以此作为等式约束, 可看作优化问题

$$F(u, w) = \frac{1}{n} \sum_{j=1}^n \tilde{F}_j(u_j) + \frac{\lambda}{2} \|w\|^2 \quad \text{s.t. } u = X^T w \quad (7.2)$$

 **练习 7.10** 计算问题 (7.2) 的对偶问题, 并给出最优乘子与原问题最优解的关系。

解 设 $u_j = x_j^T w$ 的乘子为 $\frac{1}{n}v_j$ (这里乘倍数是为了方便统一形式), 则

$$L(u, w, v) = \frac{1}{n} \sum_{j=1}^n \tilde{F}_j(u_j) + \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{j=1}^n v_j(u_j - x_j^T w)$$

由对偶问题定义, 我们需要找到 $g(v) = \min_{u, w} L(u, w, v)$, 并计算其最大值 (由于没有不等约束, $g(v)$ 是无约束最优化)。记 $\tilde{F}_j^*(v) = \min_u \{ \tilde{F}_j(u) + uv \}$, 则

$$\min_{u, w} L(u, w, v) = \frac{1}{n} \sum_{j=1}^n \tilde{F}_j^*(v_j) + \min_w \left\{ \frac{\lambda}{2} \|w\|^2 - \frac{1}{n} \sum_{j=1}^n v_j x_j^T w \right\}$$

对右侧进一步化简最终得到

$$g(v) = \frac{1}{n} \sum_{j=1}^n \tilde{F}_j^*(v_j) - \frac{1}{2\lambda n^2} \left\| \sum_{j=1}^n v_j x_j \right\|^2$$

根据 w 的最小条件, 从乘子可解出

$$w = \frac{1}{\lambda n} \sum_{j=1}^n v_j x_j$$

随机对偶坐标上升 [stochastic dual coordinate ascent, SDCA] 即对上述的 $-g(v)$ 采用坐标下降法迭代 (即对 $g(v)$ 依次选取坐标上升), 最后从 v 还原出 w 。



并行情况下, w 存储在不同的内存中, 各自的更新可能会有延迟, 不过在延迟有界时仍然可以证明收敛性, 于是 CD 算法在并行时仍有重要的意义。

7.5.3 含正则化的问题

出于减小泛化误差的要求, 正则化是必要的, 而根据稀疏优化的内容, 比起二范数正则化, 一范数更容易选取到稀疏解, 对机器学习而言意味着某种特征选择。一般的含正则化模型可以概括为

$$\min_w \Phi(w) = F(w) + \lambda \Omega(w) \quad (7.3)$$

这里 $F: \mathbb{R}^d \rightarrow \mathbb{R}$ 有较好光滑性, $\lambda > 0$ 为正则化超参数, $\Omega: \mathbb{R}^d \rightarrow \mathbb{R}$ 是凸的, 未必光滑的正则化函数。根据凸函数零阶条件推一阶条件的过程, 可说明全空间凸函数的任何方向导数都存在, 于是必须是连续的, 未必光滑也就成为了未必可微。我们着重考虑 1 范数的情况:

$$\min_w \phi(w) = F(w) + \lambda \|w\|_1 \quad (7.4)$$

我们首先给出一个**近端梯度下降** [proximal gradient iteration] 的迭代方法:

算法 7.43 (近端梯度下降-迭代)

对问题 (7.3), 给定步长 $\alpha_k > 0$, 迭代的搜索过程为

$$w_{k+1} = \operatorname{argmin}_w \left\{ F(w_k) + \nabla F(w_k)^T (w - w_k) + \frac{1}{2\alpha_k} \|w - w_k\|^2 + \lambda \Omega(w) \right\}$$



在自然梯度法的推导过程中, 我们所谓“限制范围的优化”事实上就是加入了近端项 $\frac{1}{2\alpha_k} \|w - w_k\|^2$ 的限制。而若最后一项不存在, 利用二次函数即知这个搜索的结果恰好为 $w_{k+1} = w_k - \alpha_k \nabla F(w_k)$ 。

对此迭代, 有收敛性定理:

命题 7.44 (近端梯度下降-收敛性)

在 $F(w)$ 可微、有界为 L 的梯度 Lipschitz 连续性与参数为 c 的强凸性时, 假设全局最优解存在为 w_* , 步长 α_k 恒定为 $\alpha \in (0, \frac{1}{L})$, 则迭代过程满足

$$\Phi(w_{k+1}) - \Phi(w_*) \leq (1 - \alpha c)^k (\Phi(w_1) - \Phi(w_*))$$



由于 $\nabla F(w_k)^T(w - w_k) + \frac{1}{2\alpha_k}\|w - w_k\|^2$ 可以重新配方为二次函数, 此迭代的计算核心步骤为对某个 \tilde{w} 寻找

$$\operatorname{argmin}_w \left\{ \lambda \Omega(w) + \frac{1}{2\alpha_k} \|w - \tilde{w}\|^2 \right\}$$

这又回到了 Moreau-Yosida 正则化问题, 当 $\Omega(w)$ 是一范数或可分离出 w 各个分量时是易于求解的。例如对一范数:

练习 7.11 求解

$$w_{k+1} = \operatorname{argmin}_w \left\{ F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2\alpha_k} \|w - w_k\|^2 + \lambda \|w\|_1 \right\}$$

解 配方得要优化的式子等价于

$$\frac{1}{2\alpha_k} \|w - (w_k - \alpha_k \nabla F(w_k))\|^2 + \lambda \|w\|_1$$

根据稀疏优化中完成的练习, 由此即得到

$$w_{k+1} = \operatorname{shrink}(\alpha_k \nabla F(w_k), \lambda \alpha_k)$$

上方练习中的迭代称为**迭代软阈值算法** [iterative soft-thresholding algorithm, ISTA], 从迭代过程中可以看出它的稀疏性。

继续考虑一范数正则化问题。对问题 (7.4), 可以做光滑化改造。记 $u = \max(w, 0), v = -\min(w, 0)$, 则有 $u, v \geq 0, w = u - v$, 问题变为

$$\min_{u,v} \left\{ F(u - v) + \lambda \sum_i (u_i + v_i) \right\} \quad \text{s. t. } u \geq 0, v \geq 0$$

对其可以考虑**梯度投影** [gradient projection] 方法, 也即作梯度下降后将小于 0 的分量置为 0。记 $P_+(x) = \max(x, 0)$, 计算可得有

算法 7.45 (梯度投影法-迭代)

给定步长 $\alpha_k > 0$, 迭代过程为:

$$\begin{pmatrix} u_{k+1} \\ v_{k+1} \end{pmatrix} = P_+ \begin{pmatrix} u_k - \alpha_k \nabla F(u_k - v_k) - \alpha_k \lambda \mathbf{1} \\ v_k + \alpha_k \nabla F(u_k - v_k) - \alpha_k \lambda \mathbf{1} \end{pmatrix}$$



在近端梯度下降的收敛性定理满足时, 若迭代中 u, v 可以还原出 w , 可以证明梯度投影法也具有全局收敛性。然而, 由于定义, 迭代中 u 与 v 的第 i 个分量不同时非零时, 才能保证还原出正确的 w 。

近端梯度下降与梯度投影法都有随机版本, 也即用 $g(w_k, \xi_k)$ 估计其中的 ∇F 部分。

7.5.4 正则化模型的二阶方法

虽然正则化项未必光滑, 在正则化后的模型中我们仍然可以引入二阶方法, 思路与引入一阶方法时类似, 即单独提出正则化项:

算法 7.46 (近端牛顿法-迭代)

在问题 (7.4) 中, 每步迭代先定义

$$q_k(w) = F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2}(w - w_k)^T H_k(w - w_k) + \lambda \|w\|_1$$

这里 H_k 是 $\nabla^2 F(w_k)$ 或其拟牛顿估计, 则 $q_k(w)$ 是原问题在 w 附近展开到二阶的结果。

记此问题最优解为 \tilde{w}_k , 采用一维搜索确定 α_k (保证更新后 $\phi(w)$ 下降) 并更新

$$w_{k+1} = w_k + \alpha_k(\tilde{w}_k - w_k)$$



我们假定 H_k 已经正定, 这样 $q_k(w)$ 最优解必然存在, 而我们必须找到方法求解, 一般有三种思路:

1. 进行理论精确的求解: 例如, 由于其前三项为二次函数, 对每个坐标是可以精确得到最小值的, 于是可以采用坐标下降法求得解析解。
2. 迭代出发的近似求解: 大部分情况下, 解析求解是没有必要的, 高效的近似解法是更好的选择。从任何迭代方法出发, 我们都需要找到一个认为足够近似而终止迭代的标准, 理论可以说明, 计算 ISTA 迭代的差距是一个有效的选择。

假设 $\mathcal{A}(w)$ 代表 w 进行一次 ISTA 迭代得到的点, 当前找到的解为 \tilde{w}_k , 取 $\eta \in [0, 1)$ 则终止条件可以为

$$\|\mathcal{A}(\tilde{w}_k) - \tilde{w}_k\| \leq \eta \|\mathcal{A}(w_k) - w_k\|, q_k(\tilde{w}_k) < q_k(w_k)$$

3. 稀疏出发的近似求解: 由于一范数正则化的性质, 可以考虑先确定一组 w 的积极 [active] 分量, 固定为 0, 再在剩余的自由 [free] 分量上进行求解。

另一个思路是注意到 $\|\cdot\|_1$ 在任何象限都是光滑的, 因此可以每次把搜索限制在固定的象限进行, 并不断更换象限, 这就是基于象限 [orthant-based] 的求解方案。先给出最小范数子梯度 [minimum norm subgradient] 的定义:

定义 7.47 (最小范数子梯度)

对分段可微的一元函数 $f(x)$, 记每点左右导数 (可微时相等) 分别为 $f'_-(x), f'_+(x)$, 则有其最小范数子梯度

$$g(x) = \begin{cases} \operatorname{argmin}_x \{|x| \mid x \in [f'_-(x), f'_+(x)]\} & f'_-(x) \leq f'_+(x) \\ \operatorname{argmin}_x \{|x| \mid x \in [f'_+(x), f'_-(x)]\} & f'_-(x) > f'_+(x) \end{cases}$$

对多元函数, 其最小范数子梯度的第 i 个分量为其看作第 i 个分量单元函数的最小范数子梯度。



对单元函数, 其也即左右导数构成区间内范数最小的点, 因此得名。



练习 7.12 计算问题 (7.4) 中 $\phi(w)$ 的最小范数子梯度 $\hat{g}(w)$ 。

解

$$\hat{g}(w)_i = \begin{cases} \nabla F(w)_i + \lambda & w_i > 0 \text{ or } w_i = 0, \nabla F(w)_i + \lambda < 0 \\ \nabla F(w)_i - \lambda & w_i < 0 \text{ or } w_i = 0, \nabla F(w)_i - \lambda > 0 \\ 0 & w_i = 0, |\nabla F(w)_i| \leq \lambda \end{cases}$$

当 w 各分量非零时, 其所在象限直接由定义确定, 否则通过 $\hat{g}_i(w)$ 的方向确定, 具体来说, 记 w_k

所在象限的符号为 ζ_k (每个分量为 0 或 ± 1)，则

$$\zeta_{k,i} = \begin{cases} \text{sign}(w_{k,i}) & w_{k,i} \neq 0 \\ \text{sign}(-\hat{g}(w)_i) & w_{k,i} = 0 \end{cases}$$

若 $\hat{g}(w)_i = 0$ ，即认为此分量已经固定在边界上，不进行优化。由此得到算法：

算法 7.48 (象限方法-迭代)

每次迭代中，先计算上述的 $\hat{g}(w_k)$ 与 ζ_k ，并记使 $\zeta_{k,i}$ 为 0 的 i 构成集合 \mathcal{A}_k 。求解

$$\min_d \left\{ \hat{g}(w_k)^T d + \frac{1}{2} d^T H_k d \right\} \quad \text{s.t. } d_i = 0, i \in \mathcal{A}_k$$

得到更新方向 d_k (H_k 为海森阵或其估计)，通过一维搜索确定步长 α_k ，并更新

$$w_{k+1} = P_k(w_k + \alpha_k d_k)$$

这里 P_k 代表到 ζ_k 所在象限的投影，即

$$P_{k,i}(x) = \begin{cases} \max\{x, 0\} & \zeta_{k,i} = 1 \\ \min\{x, 0\} & \zeta_{k,i} = -1 \\ 0 & \zeta_{k,i} = 0 \end{cases}$$



第 8 章 总结

在《运筹学》与《最优化算法》的课程结束之际，让我们再次回看学过的内容：

- 在**最优化绪论**中，我们给出了优化问题的一些基本定义与问题形式，并且对一般的优化问题得到进行分析，得到了始终贯穿的 **K-T 条件**。此外，我们还浅探了更加一般的拓扑空间中的下降。
- **线性规划**一章中，我们面对了第一个具体的问题，即诸多超平面分割下的优化模型。通过几何性质，我们将线性规划转化为了可行域有限的**组合优化问题**，得到了在可行基解间迭代的单纯形法，并简要观察了对偶理论。
- **网络流与动态规划**专注于一些更具体的组合优化场景，并根据不同问题的特性给出一些改进的算法。不过，这章中比起算法的实现与复杂度优化，我们更专注于理论的分析，这点与算法课形成了一些互补。
- 接下来的**无约束最优化**中，我们学到了优化算法中最常用的迭代优化，并且给出了一阶与二阶的几种不同的常用算法。这里所介绍的一维搜索技术同样是之后不断使用的。
- **有约束最优化**里，我们展示了解决约束的不同进路：局部看作二次函数的序贯优化，从乘子向量出发的牛顿求解，通过罚函数与障碍函数转化为无约束优化等等。这些进路在更特殊的问题中还有更多应用的细节。
- 对于**凸优化**这样性质良好的优化问题，我们在考察完定义与性质后即开始观察之前学习的各种方法如何使用在凸优化中，又如何得到更好的收敛性。
- 最后，在**大数据中的优化**里，我们既探索了一些现实中更复杂的优化，也着重分析了大规模科学计算下如何选取合适的算法以达到计算资源与收敛速率间的平衡。

优化，这样一个理论或应用的诸多方向都无法规避的问题，随着大数据时代的到来正凸显出越来越重要的作用。虽然学完这两门课程仅仅意味着对最优化了解的开始，但若是下次看到 Pytorch 代码中 `torch.optim.LBFGS` 能够会心一笑，大概也就不虚此行了吧。

附录 A 数学基础

本部分只罗列定义与结论，并不进行证明。

A.1 分析

定义 A.1 (差集)

集合 A 与 B 的差集定义为 $\{x \mid x \in A, x \notin B\}$ ，记作 $A \setminus B$ 。



定义 A.2 (内积、范数、夹角、正交)

对 \mathbb{R}^n 空间中的两个向量 a, b ，定义其内积为 $a^T b$ 。

对 \mathbb{R}^n 空间中的向量 a ，定义其范数为 $\sqrt{a^T a}$ ，记作 $\|a\|$ 。

对 \mathbb{R}^n 空间中的两个非零向量 a, b ，定义其夹角为 $\arccos \frac{a^T b}{\|a\| \|b\|}$ ，夹角为 $\frac{\pi}{2}$ 即为正交，当且仅当 $a^T b = 0$ 。



定义 A.3 (梯度、海森阵)

设 $f: U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ 为某可微函数，其梯度向量定义为 $\left(\frac{\partial f}{\partial x_1} \quad \dots \quad \frac{\partial f}{\partial x_n} \right)^T$ ，记作 ∇f 。

若其二阶可微，海森矩阵定义为第 i 行第 j 列是 $\frac{\partial^2 f}{\partial x_i \partial x_j}$ 的矩阵，记作 $\nabla^2 f$ 。



只需要某点处一次/二阶可微即可定义梯度/海森阵。由于偏导可交换，海森阵为对称阵。
二阶可微蕴含可微，可微蕴含连续。

定义 A.4 (Jacobi 阵)

对每个分量都可微的向量值函数 $F(x) = (f_1, \dots, f_n)^T$ ，定义其 Jacobi 阵为 $(\nabla f_1, \dots, \nabla f_n)^T$ ，假设 $F: \mathbb{R}^m \rightarrow \mathbb{R}^n$ ，那么它的 Jacobi 阵是 $n \times m$ 的，记为 \mathcal{J}_F 。



由于我们定义梯度为列向量，Jacobi 阵在 F 是一元函数时为梯度的转置。

定义 A.5 (开集、邻域、闭集)

\mathbb{R}^n 中，若对集合 U 中任何点 x 都存在 ε 使得 $B_\varepsilon(x) := \{y \in \mathbb{R}^n \mid \|y - x\| < \varepsilon\} \subset U$ ，则称 U 为开集。

若某集合 V 包含开集 U ，且 $x \in U$ ，则称 V 为 x 的一个邻域。

若一个集合的补集是开集，则其称为闭集。



命题 A.6 (\mathbb{R}^n 中开集、闭集的性质)

1. 既开又闭的集合只有空集和 \mathbb{R}^n 。
2. 任意多开集的并是开集，任意多闭集的交是闭集。
3. 有限个开集的交是开集，有限个闭集的并是闭集。
4. 闭集的等价定义：集合 B 中任何有极限点列 x_n 的极限在 B 中。



定义 A.7 (内部、闭包、边界)

对任何集合 A , A 包含的最大开集定义为 A 的内部, 记作 A° 或 $\text{int}(A)$; 包含 A 的最小闭集定义为 A 的闭包, 记作 \bar{A} ; 两者之差 $\bar{A} \setminus A^\circ$ 定义为 A 的边界, 记作 ∂A 。

**定义 A.8 (连续性)**

开集 $U \subset \mathbb{R}^n$ 中的函数 f 在一点 x_0 处连续, 当且仅当 $\lim_{x \rightarrow x_0} f(x) = f(x_0)$ 。若对定义域每点都满足此性质, 则称此函数为连续函数。

**命题 A.9 (连续的等价定义)**

1. 开集 $U \subset \mathbb{R}^n$ 中的函数 f 在一点 x_0 处连续, 当且仅当对任何满足 $x_k \rightarrow x_0$ 的点列 x_k 有 $\lim_{k \rightarrow \infty} f(x_k) = f(x_0)$ 。
2. 开集 $U \subset \mathbb{R}^n$ 中的函数 f 在一点 x_0 处连续, 当且仅当 $f(x_0)$ 的任何邻域原像是 x_0 的邻域。
3. 开集 $U \subset \mathbb{R}^n$ 中的函数 f 是连续函数当且仅当任何开集的原像是开集。
4. 开集 $U \subset \mathbb{R}^n$ 中的函数 f 是连续函数当且仅当任何闭集的原像是闭集。

**定义 A.10 (紧集)**

\mathbb{R}^n 中的紧集为有界闭集。

**命题 A.11 (紧集的等价定义)**

在 \mathbb{R}^n 中, 以下定义与 X 是紧集等价:

1. 若有一族开集 $U_i, i \in \mathcal{I}$ 覆盖 (即它们的并包含) X , 则存在有限个开集覆盖 X 。
2. X 上任何点列存在收敛子列。
3. X 上任何连续函数存在最大值、最小值。



A.2 代数

定义 A.12 (单位阵、全零阵)

只有主对角线上的元素为 1, 其他为 0 的矩阵称为单位阵, 记作 I , 或用下标 I_m 表示阶数 $m \times m$ 。所有元素全部为 0 的矩阵记作 O , 或用下标 $O_{m \times n}$ 表示阶数 $m \times n$ 。

**定义 A.13 (正定、半正定)**

若 n 阶实方阵 A 为对称阵, 且对一切 n 维实向量 x 有 $x^T A x \geq 0$, 则其称为半正定矩阵。进一步地, 若还有 $x^T A x = 0$ 当且仅当 $x = \mathbf{0}$, 则称 A 为正定矩阵。



这里的 $\mathbf{0}$ 表示零向量, 无歧义时也可以直接写作 0。对两个向量, 写 $x > y$ 的意思一般是表示 x 每个分量对应大于 y , 其他比较符号同理。



正定矩阵与半正定矩阵有时也在非对称阵上类似定义。部分教材将半正定记作 $A \geq 0$, 正定记作 $A > 0$, 但由于会与“按分量大于/大于等于”的含义产生歧义, 本讲义不引入此记号。

定义 A.14 (特征值、特征向量、特征多项式)

若存在非零向量 α 使得 $A\alpha = \lambda\alpha$, 称 λ 是 A 的一个特征值, α 是 A 的一个特征向量。

A 的特征多项式定义为 $p(\lambda) = \det(\lambda I - A)$, 其所有根 (含重根) 即 A 的所有特征值 (含重数)。

**命题 A.15 (秩、正定、半正定的特征值定义)**

秩等价于非零特征值的个数。

正定等价于对称且所有特征值大于 0。

半正定等价于对称且所有特征值大于等于 0。

**命题 A.16 (正定的其他等价定义)**

A 对称正定等价于存在可逆方阵 P 使得 $A = P^T P$ 。

A 对称半正定等价于存在方阵 P 使得 $A = P^T P$ 。

**定义 A.17 (秩、满秩)**

任何矩阵 A 可以写为 $P \begin{pmatrix} I_r & O \\ O & O \end{pmatrix} Q$ 的形式, 其中 P, Q 可逆, I_r 代表 r 阶单位阵, 则 r 称为矩阵 A 的秩, 记作 $\text{rank}(A)$ 。

若 A 的秩与行数相等, 则称其为行满秩, 类似可定义列满秩。



左乘可逆矩阵 P 相当于进行 (可逆的) 行变换, 右乘可逆矩阵 Q 相当于进行 (可逆的) 列变换。

命题 A.18 (秩的等价定义)

以下定义均与秩等价:

1. A 中最大可逆子方阵的阶数 (这里行列选取未必连续);
2. A 的行向量组的极大线性无关组元素个数;
3. A 的列向量组的极大线性无关组元素个数。

**命题 A.19 (满秩的性质)**

任何矩阵 $A_{m \times n}$ 可写为 $P_{m \times r} Q_{r \times n}$, 其中 P 是列满秩的, Q 是行满秩的。

对于方阵, 行满秩、列满秩、可逆三者等价。

**定义 A.20 (正交阵)**

对于 $n \times n$ 实方阵 P , 若其满足 $PP^T = P^T P = I$, 则称为正交阵。

**命题 A.21 (正交阵的等价定义)**

实方阵 P 是正交阵等价于所有行 (列) 构成的向量相互正交且模长均为 1。

**命题 A.22 (QR 分解)**

任何矩阵 $B \in \mathbb{R}^{n \times m}, n \geq m$ 可以写成 $Q \begin{pmatrix} R \\ O \end{pmatrix}$ 的形式, 其中 Q 为 $n \times n$ 正交阵, R 为 $m \times m$ 上三角矩阵, 且 $\text{rank}(R) = \text{rank}(B)$ 。



定义 A.23 (相似)

若存在可逆方阵 P 使得方阵 $A = P^{-1}BP$, 则称 A, B 相似。

**命题 A.24 (相似的性质)**

相似矩阵的秩、迹 (对角线元素和) 相同, 特征值对应相同。

**命题 A.25 (正交相似对角化)**

对任何实对称方阵 G , 其存在正交相似对角化 $G = Q^T D Q$, 其中 Q 是正交阵, D 是对角阵。由相似性质, D 的对角元即为 G 的特征值。



附录 B 数学进阶

本部分的知识仅在后两章用到。

定义 B.1 (向量与集合的线性运算)

对线性空间某子集 S 与向量 x_0 , 定义 $S+x_0$ 表示 $\{x \mid \exists s \in S, x = s+x_0\}$, 而 $S-x_0$ 由 $S+(-x_0)$ 类似定义。此外, 记 λS 表示 $\{x \mid \exists s \in S, x = \lambda s\}$, 则可定义 $x_0 - S$ 等。



定义 B.2 (范数)

在线性空间中, 满足

1. $\|x\| = 0 \Leftrightarrow x = 0$
2. $\|\lambda x\| = |\lambda| \|x\|$
3. $\|x+y\| \leq \|x\| + \|y\|$

的映射 $\|\cdot\|: V \rightarrow \mathbb{R}$ 称为一个范数。



定义 B.3 (p 范数)

在 \mathbb{C}^n 中, 定义函数

$$\|x\|_p = \begin{cases} x \text{ 中非零分量的个数} & p = 0 \\ (\sum_i |x_i|^p)^{1/p} & 0 < p < \infty \\ \max_i |x_i| & p = \infty \end{cases}$$

当且仅当 $p \geq 1$ 时此函数为范数, 称为 p 范数。若无下标, 默认为二范数。



当 $0 \leq p < 1$ 时有时仍称此函数为 p 范数。

定义 B.4 (关系)

集合 A 上, 任何一个映射 $R: A \times A \rightarrow \{0, 1\}$ 称为一个关系。 $R(x, y) = 1$ 记为 xRy , 否则记为 $x\bar{R}y$ 。



定义 B.5 (等价关系)

一个集合中, 满足

1. xRx (自反性)
2. $xRy \Rightarrow yRx$ (对称性)
3. $xRy, yRz \Rightarrow xRz$ (传递性)

的关系 R 称为一个等价关系。



定义 B.6 (偏序关系)

一个集合中, 满足

1. xRx (自反性)
2. $xRy, yRx \Rightarrow x = y$ (反对称性)

3. $xRy, yRz \Rightarrow xRz$ (传递性)

的关系 R 称为一个偏序关系。



常用的 \geq, \leq 都是偏序关系，但一般偏序关系与它们不同， xRy 与 yRx 可能都不成立。

定义 B.7 (极小、极大、最小、最大)

对定义了偏序关系 \preceq 的集合 S ， x 是极小元是指 $\forall y \in S, y \preceq x \Rightarrow y = x$ ， x 是极大元是指 $\forall y \in S, x \preceq y \Rightarrow y = x$ ； x 是最小元是指 $\forall y \in S, x \preceq y$ ； x 是最大元是指 $\forall y \in S, y \preceq x$ 。



通俗来说，极小是指“没有比它小的”，最小是指“其他都比它大”，考虑集合之间的包含关系 \subset 可以给出一些直观的例子。根据最小、最大的定义，由于反对称性可得最小元/最大元至多只有一个。

命题 B.8 (奇异值分解)

任何矩阵 $A \in \mathbb{R}^{n \times m}$ 可以写成 $U\Sigma V^T$ 的形式，其中 U 为 $n \times n$ 正交阵， V 为 $m \times m$ 正交阵， Σ 为对角元非负且从大到小排列的对角阵，其对角元称为奇异值，从大到小记为 $\sigma_1, \dots, \sigma_{\min(m,n)}$ 。



命题 B.9 (半正定阵的相合对角化)

对半正定矩阵 A, B ，存在可逆方阵 P 使得 $P^T A P$ 与 $P^T B P$ 均为对角阵。



命题 B.10 (半正定阵的次幂)

对半正定矩阵 A ，给定正整数 n ，存在唯一半正定矩阵 B 使得 $B^n = A$ ，记作 $B = A^{1/n}$ 。从而类似正整数次方推广到实数次方的过程，对半正定矩阵可以定义任何非负次方，对正定矩阵可以定义任何实数次方。



定义 B.11 (Hermite 阵、酉方阵)

将满足 $A^H = A$ 的复矩阵称为 Hermite 阵， $A^H A = I$ 则称为酉方阵，其性质与对称阵、正交阵类似。

复矩阵中，正定与半正定仍然等价于 Hermite 阵的全部特征值大于/大于等于 0，也仍然可以定义次幂。



定义 B.12 (矩阵范数)

对所有矩阵 $\mathbb{C}^{n \times n}$ 定义的函数 $\|\cdot\|$ ，若满足线性空间范数的三条要求，且满足 $\|AB\| \leq \|A\|\|B\|$ ，则称为一个矩阵范数。

矩阵 p 范数定义为

$$\|A\|_p = \inf_{\|x\|_p=1} \|Ax\|_p$$

Forbenius 范数为

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$$

而核范数为

$$\|A\|_* = \text{tr}((X^H X)^{1/2})$$

若无下标，亦默认为二范数。



命题 B.13 (矩阵范数与奇异值)

$$\|A\| = \sigma_1$$

$$\|A\|_F = \sqrt{\sum_i \sigma_i^2}$$

$$\|A\|_* = \sum_i \sigma_i$$



定义 B.14 (联合范数)

对矩阵 $A \in \mathbb{R}^{m \times n}$ ，设其列向量为 a_j ，则联合范数 $\|A\|_{p,q}$ 定义为

$$\|(\|a_1\|_p, \|a_2\|_p, \dots, \|a_m\|_p)\|_q$$

联合范数也即先计算每列的范数，再构成一个向量计算范数。于是 $\|A\|_F$ 也可以表示为 $\|A\|_{2,2}$ 。



定义 B.15 (子拓扑)

对 \mathbb{R}^n 的任何子集 S 与 S 的子集 A ，定义 A 在 S 中是开集/闭集当且仅当存在 \mathbb{R}^n 中的开集/闭集 B 满足 $A = S \cap B$ 。于是，我们可以进一步定义 A 在 S 中的内部 (A 包含的最大 S 中开集)、闭包 (包含 A 的最小 S 中闭集)、边界等。



附录 C 参考资料

- 《运筹学讲义》杨周旺
- 《最优化算法讲义》杨周旺
- github.com/hehaha68/USTC_2021Fall_Operations-Research [部分中间证明]
- faculty.bicmr.pku.edu.cn/~wenzw/optbook.html [推荐参考讲义]
- zhuanlan.zhihu.com/p/370120797 [运筹学简介]
- zhuanlan.zhihu.com/p/51127402 [凸函数等价定义]
- zhuanlan.zhihu.com/p/379262669 [几何必要条件]
- www.doc88.com/p-9156753467829.html [凸集分离定理]
- zhuanlan.zhihu.com/p/441064060 [紧凸集为极点闭凸组合 (\mathbb{R}^n 情况即凸组合)]
- 《算法导论 (第三版)》Thomas H.Cormen 等著，殷建平等译 [网络流与动态规划算法]
- www.bilibili.com/read/cv17908771 [最大流最小割对偶]
- personal.math.ubc.ca/~ansteemath523/StronglyPolynomialNetworkFlow.pdf [强多项式最小成本流]
- blog.csdn.net/weixin_46503238/article/details/115132242 [多重背包问题优化算法]
- blog.csdn.net/u013250861/article/details/122284630 [Viterbi 算法]
- 《数值最优化方法》高立 [收敛性与收敛速率]
- zhuanlan.zhihu.com/p/264515249 [内点法]
- blog.csdn.net/u010510549/article/details/100938214 [Slater 条件]
- blog.csdn.net/qianhen123/article/details/85227297 [字典学习]
- zhuanlan.zhihu.com/p/369874722 [各种风险与误差]
- faculty.bicmr.pku.edu.cn/~wenzw/optbook/lect/25-lect-sto-ch.pdf [随机优化算法]
- www.di.ens.fr/~fbach/Defazio_NIPS2014.pdf [SAGA 算法论文]