DCS440 最优化理论

第四章：约束优化算法

杨 磊

yanglei39@mail.sysu.edu.cn

计算机学院，2024 秋

- 罚函数法
- 增广拉格朗日乘子法
- 交替方向乘子法
- 分布式优化简介
- 扩展：线性规划内点法简介
- 扩展：对偶算法在最优传输上的应用

约束优化问题

在本章，我们将学习求解约束优化问题的相关算法。

一般约束优化问题

$$\min_{x} \quad f_0(x) \quad \text{s.t.} \quad x \in \mathcal{X},$$

其中 $\mathcal{X} \in \mathbb{R}^n$ 为问题的可行域。

难点：由于约束的存在，前面学习过的无约束优化算法无法直接应用于求解约束优化问题。

以**梯度下降法**为例，对于约束优化问题而言：

- 沿着负梯度方向更新所得的点不一定可行，即：

$$x^{k+1} = x^k - \alpha_k \nabla f_0(x^k), \quad \text{但是 } x^{k+1} \notin \mathcal{X}.$$

- 目标函数的梯度在最优解处也未必是零向量，即：

$$x^* \in \arg\min_{x \in \mathcal{X}} f_0(x), \quad \text{但是 } \nabla f_0(x^*) \neq 0.$$

核心问题：如何处理约束？

等式约束优化问题

考虑等式约束的优化问题（不要求凸）

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \cdots, p, \end{aligned} \tag{P}$$

其中 $f_0(x)$，$h_i(x)$ 均为连续可微函数。

思考：能否巧妙地处理约束使得 约束优化问题(**P**) $\rightarrow$ 无约束优化问题？

思路1：对于一些特殊约束，可求解方程组 $h_i(x) = 0$，消去部分变量，代入目标函数，将问题转化为无约束优化问题。但是，一般而言，求解非线性方程组是十分困难的！

思路2：在目标函数中加入与约束函数相关的惩罚项，通过对可行域外的点施加“惩罚”来替代原可行域约束，进而将原约束问题转化为无约束问题。

★ 该方法称作**罚函数法**！

二次罚函数法

定义 1 (等式约束问题的二次罚函数)

对于上述的等式约束优化问题 (**P**)，它的**二次罚函数**定义为：

$$P_E(x,\sigma) = f_0(x) + \frac{\sigma}{2}\sum_{i=1}^{p}\left|h_i(x)\right|^2,$$

其中常数 $\sigma > 0$ 称为罚因子或罚参数。

罚函数的形式有很多种，二次罚函数是最简单且最常用的一种。

当我们极小化二次罚函数 $P_E(x,\sigma)$ 时，可以看到

- 对于可行域外的点，惩罚项为正，即对该点进行惩罚；
- 对于可行域内的点，惩罚项为 0，即对该点没有任何惩罚；
- 罚因子 σ 控制着惩罚项的权重，当 σ 变大时，逐步迫使 $P_E(x,\sigma)$ 的极小值点向原问题 (**P**) 的可行域靠近。

二次罚函数法

Algorithm 1: 求解问题 (**P**) 的二次罚函数法

给定初始点 x_0，初始罚参数 $\sigma_0 > 0$ 和罚因子增长系数 $\rho > 1$。令 $k = 0$。

while 未达到收敛准则 **do**

  以 x^k 为初始点，调用无约束优化算法求解

$$x^{k+1} \in \arg\min_x \ P_E(x, \sigma_k).$$

  增大罚参数：$\sigma_{k+1} = \rho\,\sigma_k$

  $k \leftarrow k + 1$;

end

定理 1 (二次罚函数法的收敛性)

假设对 $\forall \sigma > 0$，罚函数 $P_E(x, \sigma)$ 的全局最优解都存在。设罚因子 σ_k 单调上升趋于无穷，x^{k+1} 是 $P_E(x, \sigma_k)$ 的一个全局最优解，则序列 $\{x^k\}$ 的每一个聚点 x^* 都是原问题 (**P**) 的全局最优解。

二次罚函数法的局限

设 x^* 是原约束问题 (**P**) 的一个最优解，且 $\nabla f_0(x^*) \neq 0$。

注意到这样的最优解是存在的! 否则，约束问题 (**P**) 将“等价于” $\min\limits_x\{f_0(x)\}$，约束 $h_i(x) = 0\ (i = 1, \cdots, p)$ 不再需要，故可用无约束优化算法直接求解。

于是，由 KKT 条件可知，存在 Lagrange 乘子 $\lambda^* \neq 0$ 使得

$$\begin{aligned} \nabla f_0(x^*) + \sum_{i=1}^{p} \lambda_i^* \nabla h_i(x^*) = 0, \\ h_i(x^*) = 0, \quad \forall i = 1, \cdots, p. \end{aligned} \tag{KKT-P}$$

同时考虑第 k 步迭代时，二次罚函数法子问题的 KKT 条件：

$$\nabla f_0(x^{k+1}) + \sum_{i=1}^{p} \sigma_k h_i(x^{k+1}) \nabla h_i(x^{k+1}) = 0. \tag{KKT-E}$$

二次罚函数法的局限

对比两个 KKT 系统，若想通过逐步极小化二次罚函数求解原约束问题 (**P**)，则 x^k 在 $k \to \infty$ 时应满足如下条件：

$$\begin{cases} \sigma_k h_i(x^{k+1}) \ \to \ \lambda_i^*, \\ h_i(x^{k+1}) \ \to \ 0, \end{cases} \qquad \forall\, i = 1, \cdots, p.$$

由于 $\lambda^* \neq 0$，即存在 $\lambda_i \neq 0$，这样为保证上述条件成立，则需 $\sigma_k \to +\infty$。

然而，罚因子 $\sigma_k \to +\infty$ 将使得二次罚函数问题的求解出现数值问题！

思考：如何改进罚方法，减弱对罚因子 $\sigma_k \to +\infty$ 的要求？

- 罚函数法
- 增广拉格朗日乘子法
- 交替方向乘子法
- 分布式优化简介
- 扩展：线性规划内点法简介
- 扩展：对偶算法在最优传输上的应用

等式约束问题的 ALM

★ 采用**增广拉格朗日函数法 (Augmented Lagrangian Method, ALM)**

回顾**等式约束优化问题**（不要求凸）：

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \cdots, p. \end{aligned} \tag{P}$$

问题 (P) 的**增广拉格朗日函数**定义如下：

$$L_\sigma(x, \lambda) = f_0(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \frac{\sigma}{2} \sum_{i=1}^{p} h_i^2(x).$$

它可以看作是 Lagrange 函数 和 二次罚函数 的组合。

于是，我们可以给出**增广拉格朗日函数法**的迭代框架：令 $\rho > 1$，

$$\text{(ALM)} \quad \begin{cases} x^{k+1} \in \arg\min\limits_{x} \ L_{\sigma_k}(x, \lambda^k), \\ \lambda_i^{k+1} = \lambda_i^k + \sigma_k h_i(x^{k+1}), \quad i = 1, \cdots, p, \\ \sigma_{k+1} = \rho \sigma_k, \end{cases}$$

ALM 的构造思想

假设 (x^*, λ^*) 是 (**P**) 的原对偶最优解。考虑 (**P**) 的 KKT 条件：

$$\begin{aligned} &\nabla f_0(x^*) + \sum_{i=1}^{p} \lambda_i^* \nabla h_i(x^*) = 0, \\ &h_i(x^*) = 0, \quad \forall\, i = 1, \cdots, p. \end{aligned} \tag{**KKT-P**}$$

同时考虑第 k 步迭代时，增广拉格朗日函数法子问题的 KKT 条件：

$$x^{k+1} \in \arg\min_x \; L_{\sigma_k}(x, \lambda^k) \implies \nabla_x L_{\sigma_k}(x^{k+1}, \lambda^k) = 0,$$

$$\implies \quad \nabla f_0(x^{k+1}) + \sum_{i=1}^{p} \left(\lambda_i^k + \sigma_k h_i(x^{k+1})\right) \nabla h_i(x^{k+1}) = 0.$$

对比两个 KKT 系统，若想通过逐步极小化增广拉格朗日函数求解原约束问题 (**P**)，则 x^k 在 $k \to \infty$ 时应满足如下条件：

$$\begin{cases} \lambda_i^k + \sigma_k h_i(x^{k+1}) \;\to\; \lambda_i^*, \\ h_i(x^{k+1}) \;\to\; 0, \end{cases} \quad \forall\, i = 1, \cdots, p.$$

ALM 的构造思想

可以观察到：相较**二次罚函数法**，为保证上述条件成立，**增广拉格朗日函数法**的迭代格式对罚因子 σ_k 的要求可以减弱，可以不要求 $\sigma_k \to \infty$!

总结:

- $L_\sigma(x, \lambda)$ 在 Lagrange 函数的基础上，添加了等式约束的二次罚函数；
- $L_\sigma(x, \lambda)$ 在二次罚函数的基础上，引入了 Lagrange 乘子，添加了一次（线性）项，包含了对偶信息；
- 因此，$L_\sigma(x, \lambda)$ 是 Lagrange 函数 和 二次罚函数 的组合。
- 通过合理地更新乘子 λ^k 可以辅助我们控制约束违反度，同时减弱二次罚函数法对 $\sigma \to +\infty$ 的要求。
- 事实上，受 $\lambda_i^k + \sigma_k h_i(x^{k+1}) \to \lambda_i^*$ 启发，考虑如下方式更新乘子：

$$\lambda_i^{k+1} = \lambda_i^k + \sigma_k h_i(x^{k+1}).$$

罚因子 σ_k 的更新

更新罚因子 σ_k 时，应考虑如下的问题：

- σ_k 不应增长过快：
  - ⋆ 与二次罚函数法类似，过大的罚因子 σ_k 会导致增广拉格朗日函数法子问题 $\min\limits_x L_{\sigma_k}(x, \lambda^k)$ 变的病态 (ill-conditioned)，求解更加困难；
  - ⋆ 热启动 (warm-start)：在第 k 步迭代，可利用第 $k-1$ 步得到的解 x^{k-1} 作为求解第 k 步子问题的初始点，加快实际计算效率；

- σ_k 也不应增长过慢：过慢的增长可能导致“惩罚”不足，进而降低算法整体的收敛速度。

在实际应用增广拉格朗日方法时，应该控制 σ_k 的增长维持在一个合理的速度区间内。一个经验的方法是取增长因子 $\rho \in [2, 10]$。同时，也可以根据 KKT 残差信息自适应的调节 σ_k 的大小。

ALM 的收敛性 I

通过假设 Lagrange 乘子序列的有界性以及收敛点处的约束规范条件，可以证明增广拉格朗日函数法迭代产成的点列 $\{x^k\}$ 若有收敛子列，该子列将收敛到问题(**P**)的 KKT 点[1]。

定理 2 (增广拉格朗日函数法的收敛性 I（不要求凸）)

假设 Lagrange 乘子序列 $\{\lambda^k\}$ 是有界的，罚因子 $\sigma_k \to +\infty$，增广拉格朗日函数法产成的点列 $\{x^k\}$ 有一个收敛子列 $\{x^{k_j+1}\}$ 收敛到 x^*，并且在点 x^* 处线性独立约束规范条件 LICQ[1] 成立。那么，存在 λ^*，满足

$$\lambda^{k_j+1} \to \lambda^*, \quad j \to \infty,$$

$$\nabla f(x^*) + \sum_{i=1}^{p} \lambda_i^* \nabla h_i(x^*) = 0,$$

$$h_i(x^*) = 0, \quad i = 1, \cdots, p.$$

[1]关于增广拉格朗日函数法收敛性的详细分析，可参考教材“最优化：建模、算法与理论, 刘浩洋, 户将, 李勇锋, 文再文, 高教出版社, 2021” 的第七章第二节

[1]LICQ 指 $\{\nabla h_1(x^*), \nabla h_2(x^*), \cdots, \nabla h_p(x^*)\}$ 线性无关

ALM 的收敛性 II

定理 3 (增广拉格朗日函数法的收敛性 II（不要求凸）)

假设 x^*, λ^* 分别是问题 (**P**) 的严格局部极小解和相应的乘子，并且存在充分大的常数 $\bar{\sigma} > 0$ 和充分小的常数 $\delta > 0$，使得对某个 k 有

$$\sigma_k \geqslant \bar{\sigma}, \quad \frac{1}{\sigma_k}\|\lambda^k - \lambda^*\| < \delta,$$

则

$$\lambda^k \to \lambda^*, \quad x^k \to x^*.$$

同时，有以下结论成立：

- 如果 $\limsup_{k\to+\infty} \sigma_k < +\infty$ 且 $\lambda^k \neq \lambda^*, \forall k$，则乘子序列 $\{\lambda^k\}$ 收敛到 λ^* 的速度是 Q-线性的；
- 如果 $\limsup_{k\to+\infty} \sigma_k = +\infty$ 且 $\lambda^k \neq \lambda^*, \forall k$，则乘子序列 $\{\lambda^k\}$ 收敛到 λ^* 的速度是 Q-超线性的。

★ **增广拉格朗日函数法**不需要 $\sigma_k \to +\infty$ 即可保证收敛性！

★ 对于凸问题，增广拉格朗日函数法可以在更弱的条件下得到更好的收敛性结果；参见 “R.T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Mathematics of Operations Research, 1(2):97–116, 1976.”

案例：基追踪问题

基追踪问题来源于压缩感知领域，它的数学形式可以写为

$$\min_{x\in\mathbb{R}^n} \quad \|x\|_1 \qquad \text{s.t.} \qquad Ax=b, \tag{P}$$

其中 $A\in\mathbb{R}^{m\times n}$ $(m\leqslant n)$, $b\in\mathbb{R}^m$。

引入乘子 λ 和罚因子 σ，可以写出其增广拉格朗日函数：

$$L_\sigma(x,\,\lambda)=\|x\|_1+\lambda^\top(Ax-b)+\frac{\sigma}{2}\|Ax-b\|_2^2.$$

这样，求解问题 (**P**) 的 ALM 为（令 $\rho>1$ 和 $\bar{\sigma}\leq+\infty$）：

$$\begin{cases} x^{k+1}=\arg\min\limits_x\left\{L_{\sigma_k}(x,\,\lambda^k)\right\}=\arg\min\limits_x\left\{\|x\|_1+\dfrac{\sigma_k}{2}\left\|Ax-b+\sigma_k^{-1}\lambda^k\right\|_2^2\right\},\\ \lambda^{k+1}=\lambda^k+\sigma_k(Ax^{k+1}-b),\\ \sigma_{k+1}=\min\left\{\rho\sigma_k,\,\bar{\sigma}\right\}. \end{cases}$$

注意：ALM 的子问题一般无显式解，需调用其它有效算法迭代求解，因此一般只能得到一个近似解！

于是，需要构建非精确 ALM，以符合实际情形！本课程将不再介绍。

*一般约束问题的 ALM

考虑带有不等式约束的优化问题（不要求凸）：

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \cdots, p, \\ & g_j(x) \leqslant 0, \quad j = 1, \cdots, m, \end{aligned}$$

其中 $f_0(x), h_i(x), g_j(x)$ 均为连续可微函数。

对不等式约束 $g_j(x) \leqslant 0$ 引入松弛变量 s_j，得到如下等价形式：

$$\begin{aligned} \min_{x,s} \quad & f_0(x) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \cdots, p, \\ & g_j(x) + s_j = 0, \quad j = 1, \cdots, m, \\ & s_j \geq 0, \quad j = 1, \cdots, m. \end{aligned} \tag{▲}$$

*一般约束问题的 ALM

然后，将非负约束改写成示性函数加到目标函数中，得到问题的等价形式：

$$\begin{aligned} \min_{x,s} \quad & f_0(x) + I_{\geq 0}(s) \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \cdots, p, \\ & g_j(x) + s_j = 0, \quad j = 1, \cdots, m, \end{aligned}$$

其中

$$I_{\geq 0}(s) = \begin{cases} 0, & s_j \geq 0,\ j = 1, \cdots, m, \\ +\infty, & \text{otherwise.} \end{cases}$$

写出上述等价问题的增广拉格朗日函数：

$$\begin{aligned} \widetilde{L}_\sigma(x,\, s,\, \lambda,\, \mu) = f_0(x) + I_{\geq 0}(s) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{m} \mu_j (g_j(x) + s_j) \\ + \frac{\sigma}{2} \left(\sum_{i=1}^{p} h_i^2(x) + \sum_{j=1}^{m} \left(g_j(x) + s_j\right)^2 \right). \end{aligned}$$

*一般约束问题的 ALM

于是，可以得到求解不等式约束优化问题的 ALM：

$$\begin{cases} (x^{k+1},\, s^{k+1}) = \arg\min\limits_{x,\, s} \ \widetilde{L}_{\sigma_k}(x,\, s,\, \lambda^k,\, \mu^k), \\ \lambda_i^{k+1} = \lambda_i^k + \sigma_k h_i(x^{k+1}), \quad i = 1, \cdots, p, \\ \mu_j^{k+1} = \mu_j^k + \sigma_k (g_j(x^{k+1}) + s_j^{k+1}), \quad j = 1, \cdots, m, \\ \sigma_{k+1} = \rho \sigma_k. \end{cases}$$

可以观察到，上述 ALM 是借助引入松弛变量 s 转化为等式约束问题得到的！

思考：是否可以进一步消去松弛变量 s 使迭代格式只与 x 和乘子 λ, μ 有关？

答：可以的！

*一般约束问题的 ALM

观察上述 ALM 中的子问题：

$$\min_{x,\,s}\ \widetilde{L}_{\sigma_k}(x,\,s,\,\lambda^k,\,\mu^k)=\min_{x}\left\{\min_{s}\ \widetilde{L}_{\sigma_k}(x,\,s,\,\lambda^k,\,\mu^k)\right\}. \qquad (\star)$$

于是，对任意 x，关于 s 的子问题化为

$$\min_{s}\ I_{\geq 0}(s)+\sum_{j=1}^{m}\mu_j^k\big(g_j(x)+s_j\big)+\frac{\sigma_k}{2}\sum_{j=1}^{m}\big(g_j(x)+s_j\big)^2.$$

显然，这是一个关于 s 的二次优化问题，易分析可得最优解为：

$$s_j^{*,x}=\max\left\{-\frac{\mu_j^k}{\sigma_k}-g_j(x),\,0\right\},\quad j=1,\cdots,m.$$

这样，最优解 $s^{*,x}$ 可以用 x 显示表达出来！再将该表达式代入问题$(\star)$即可得到只关于 x 的子问题。

*一般约束问题的 ALM

具体来说，将上述 $s^{*,x}$ 的表达式代入问题($\star$)，可得

$$\begin{aligned}
&\min_{x,\,s}\ \widetilde{L}_{\sigma_k}(x,\,s,\,\lambda^k,\,\mu^k)\\
&= \min_{x} \underbrace{\left\{ \begin{aligned} &f_0(x)+\sum_{i=1}^{p}\lambda_i h_i(x)+\frac{\sigma_k}{2}\sum_{i=1}^{p}h_i^2(x)\\ &\qquad +\frac{\sigma_k}{2}\sum_{j=1}^{m}\left(\max\left\{\frac{\mu_j^k}{\sigma_k}+g_j(x),\,0\right\}^2-\frac{{\mu_j^k}^2}{\sigma_k^2}\right) \end{aligned} \right\}}_{=:\,L_{\sigma_k}(x,\lambda^k,\mu^k)}
\end{aligned}$$

这样，我们便消去了松弛变量 s，得到只关于 x 的极小化子问题。于是，有

$$\begin{cases} x^{k+1}=\arg\min\limits_{x}\ L_{\sigma_k}(x,\lambda^k,\mu^k), \\ s_j^{k+1}=\max\left\{-\dfrac{\mu_j^k}{\sigma_k}-g_j(x^{k+1}),\,0\right\}, \quad j=1,\cdots,m. \end{cases}$$

*一般约束问题的 ALM

进一步，再将 s_j^{k+1} 的表达式直接代入 μ_j^{k+1} 中，得到

$$\mu_j^{k+1} = \max\left\{\mu_j^k + \sigma_k g_j(x^{k+1}),\, 0\right\}, \quad j = 1, 2, \cdots, m.$$

综上，对于一般约束的优化问题，其增广拉格朗日函数法的迭代框架如下：

$$\begin{cases} x^{k+1} = \arg\min\limits_x \; L_{\sigma_k}(x, \lambda^k, \mu^k), \\ \lambda_i^{k+1} = \lambda_i^k + \sigma_k h_i(x^{k+1}), \quad i = 1, 2, \cdots, p, \\ \mu_j^{k+1} = \max\left\{\mu_j^k + \sigma_k g_j(x^{k+1}),\, 0\right\}, \quad j = 1, 2, \cdots, m, \\ \sigma_{k+1} = \rho\sigma_k. \end{cases}$$

这里，$L_{\sigma_k}(x, \lambda^k, \mu^k)$ 称作是一般约束优化问题的增广拉格朗日函数！

*与邻近点算法的联系

考虑有线性等式约束的凸优化问题：

$$\min_x \quad f_0(x) \qquad \text{s.t.} \qquad Ax = b,$$

其中 f_0 是适当闭凸函数。求解该问题的增广拉格朗日函数法如下：

$$x^{k+1} \in \arg\min_x \left\{ f_0(x) + \langle \lambda^k,\, Ax - b \rangle + \frac{\sigma_k}{2} \|Ax - b\|^2 \right\}$$

$$\lambda^{k+1} = \lambda^k + \sigma_k (Ax^{k+1} - b)$$

下面，我们将分析：

用增广拉格朗日函数法求解原问题 $\Longleftrightarrow$ 用邻近点算法求解对偶问题

首先，利用共轭函数写出原问题的 Lagrange 对偶函数：

$$\begin{aligned} g(\lambda) &:= \inf_x \{L(x,\, \lambda)\} = \inf_x \left\{ f_0(x) + \langle A^\top \lambda,\, x \rangle \right\} - b^\top \lambda \\ &= -\, f_0^*(-A^\top \lambda) - b^\top \lambda \end{aligned}$$

*与邻近点算法的联系

于是，得到如下对偶问题：

$$\max_{\lambda} \ g(\lambda) := -f_0^*(-A^\top \lambda) - b^\top \lambda$$

$$\Updownarrow$$

$$\min_{\lambda} \ \tilde{g}(\lambda) := f_0^*(-A^\top \lambda) + b^\top \lambda$$

这样，求解对偶问题的**邻近点算法**如下：

$$\lambda^{k+1} = \arg\min_{\lambda} \left\{ \tilde{g}(\lambda) + \frac{1}{2\sigma_k} \|\lambda - \lambda^k\|^2 \right\}$$

$$\Updownarrow \quad \text{根据最优性条件}$$

$$\begin{aligned} \lambda^{k+1} &\in \lambda^k - \sigma_k \partial \tilde{g}(\lambda^{k+1}) \\ &= \lambda^k - \sigma_k \left(-A \partial f_0^*(-A\lambda^{k+1}) + b \right) \\ &= \lambda^k + \sigma_k \left(A \partial f_0^*(-A\lambda^{k+1}) - b \right) \end{aligned}$$

*与邻近点算法的联系

由上述关系可知，求解对偶问题的**邻近点算法**等价于在每一迭代步，寻找 λ^{k+1} 以及 $x^{k+1} \in \partial f_0^*(-A\lambda^{k+1})$ 满足以下关系：

$$\begin{cases} x^{k+1} \in \partial f_0^*(-A\lambda^{k+1}) \\ \lambda^{k+1} = \lambda^k + \sigma_k(Ax^{k+1} - b) \end{cases}$$

$$\Updownarrow \quad \text{根据次微分的性质}$$

$$\begin{cases} -A\lambda^{k+1} \in \partial f_0(x^{k+1}) \\ \lambda^{k+1} = \lambda^k + \sigma_k(Ax^{k+1} - b) \end{cases}$$

$$\Updownarrow \quad \text{将 } \lambda^{k+1} \text{ 代入第一式}$$

$$\begin{cases} -A\left(\lambda^k + \sigma_k(Ax^{k+1} - b)\right) \in \partial f_0(x^{k+1}) \\ \lambda^{k+1} = \lambda^k + \sigma_k(Ax^{k+1} - b) \end{cases}$$

*与邻近点算法的联系

进一步，整理关系式 $-A\left(\lambda^k+\sigma_k(Ax^{k+1}-b)\right)\in\partial f_0(x^{k+1})$，有

$$0\in\partial f_0(x^{k+1})+A^\top\lambda^k+\sigma_kA^\top(Ax^{k+1}-b)$$

$$\Updownarrow \quad \text{根据最优性条件}$$

$$x^{k+1}\in\arg\min_x\left\{f_0(x)+\langle\lambda^k,\,Ax-b\rangle+\frac{\sigma_k}{2}\|Ax-b\|^2\right\}$$

综上，可以看到求解对偶问题的**邻近点算法**等价于在每一迭代步，寻找 λ^{k+1} 以及 x^{k+1} 满足以下关系：

$$\begin{cases}x^{k+1}\in\arg\min_x\left\{f_0(x)+\langle\lambda^k,\,Ax-b\rangle+\frac{\sigma_k}{2}\|Ax-b\|^2\right\},\\ \lambda^{k+1}=\lambda^k+\sigma_k(Ax^{k+1}-b).\end{cases}$$

这便得到了求解原问题的**增广拉格朗日函数法**。

- 罚函数法
- 增广拉格朗日乘子法
- 交替方向乘子法
- 分布式优化简介
- 扩展：线性规划内点法简介
- 扩展：对偶算法在最优传输上的应用

问题背景

考虑具有两块可分结构的线性约束凸优化问题：

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, y\in\mathbb{R}^m} \quad & f(x)+g(y) \\ \text{s.t.} \quad & Ax+By=b, \end{aligned} \tag{P}$$

其中 $f:\mathbb{R}^n\to\mathbb{R}$ 和 $g:\mathbb{R}^m\to\mathbb{R}$ 均是适当闭凸函数（均不要求光滑性），$A\in\mathbb{R}^{p\times n}$，$B\in\mathbb{R}^{p\times m}$ 和 $b\in\mathbb{R}^p$ 给定。

问题特点：

- 具有两块变量 x,y 和可分离的两个目标函数 f,g；
- 但是变量 x,y 被线性约束 $Ax+By=b$ 耦合在一起；
- 另外 f 和 g 均不要求光滑性。

★ 常见的无约束和有约束优化问题都可以通过简单变换表示为该问题形式。

问题背景

- 复合型无约束优化问题：（f, g 均不要求光滑性）

$$\min_{x} \quad f(x)+g(x).$$

引入新变量 y，令 $y=x$，将问题转化如下：

$$\Longrightarrow \quad \min_{x,\,y} \quad f(x)+g(y) \qquad \text{s.t.} \qquad x-y=0.$$

- 带线性变换的无约束优化问题：（f, g 均不要求光滑性）

$$\min_{x} \quad f(x)+g(Ax).$$

引入新变量 y，令 $y=Ax$，将问题转化如下：

$$\Longrightarrow \quad \min_{x,\,y} \quad f(x)+g(y) \qquad \text{s.t.} \qquad Ax-y=0.$$

- 集合 $\mathcal{C}\subset\mathbb{R}^n$ 上的约束优化问题：

$$\min_{x} \quad f(x) \qquad \text{s.t.} \qquad x\in\mathcal{C}.$$

引入新变量 y 和集合 $\mathcal{C}$ 的示性函数 $\delta_{\mathcal{C}}(y)$，令 $y=x$，将问题转化如下

$$\Longrightarrow \quad \min_{x,\,y} \quad f(x)+\delta_{\mathcal{C}}(y) \qquad \text{s.t.} \qquad x-y=0.$$

问题的求解

下面，我们讨论如何求解问题 (**P**)？

回顾上一节，可以考虑应用**增广拉格朗日函数法 (ALM)**！

首先，写出问题 (**P**) 的增广拉格朗日函数：

$$L_\sigma(x,y,\lambda)=f(x)+g(y)+\lambda^\top(Ax+By-b)+\frac{\sigma}{2}\|Ax+By-b\|_2^2.$$

然后，**ALM** 的迭代格式如下：（令 $\rho>1$ 和 $\bar{\sigma}<+\infty$）

$$\begin{cases}(x^{k+1},\,y^{k+1})=\arg\min\limits_{x,\,y}\ L_{\sigma_k}(x,\,y,\,\lambda^k),\\ \lambda^{k+1}=\lambda^k+\sigma_k(Ax^{k+1}+By^{k+1}-b),\\ \sigma_{k+1}=\min\left\{\rho\sigma_k,\,\bar{\sigma}\right\}.\end{cases}$$

可以观察到，**ALM** 的子问题需要对变量 x 和 y 同时作极小化！

思考：如果对变量 x 和 y 同时作极小化的难度过大该怎么办？

问题的求解

一种简单策略：对变量 x 和 y 进行逐个击破，即对 x 和 y 依次作极小化！

具体来说，考虑：

$$(x^{k+1},\, y^{k+1}) = \arg\min_{x,\,y}\ L_{\sigma_k}(x,\, y,\, \lambda^k)$$

$$\Downarrow \quad \text{对 } x \text{ 和 } y \text{ 依次作极小化}$$

$$\begin{cases} x^{k+1} = \arg\min\limits_{x}\ L_{\sigma_k}(x,\, y^k,\, \lambda^k) \\ \qquad\ = \arg\min\limits_{x}\left\{f(x) + (\lambda^k)^\top(Ax + By^k - b) + \dfrac{\sigma_k}{2}\|Ax + By^k - b\|_2^2\right\} \\ y^{k+1} = \arg\min\limits_{y}\ L_{\sigma_k}(x^{k+1},\, y,\, \lambda^k) \\ \qquad\ = \arg\min\limits_{y}\left\{g(y) + (\lambda^k)^\top(Ax^{k+1} + By - b) + \dfrac{\sigma_k}{2}\|Ax^{k+1} + By - b\|_2^2\right\} \end{cases}$$

相较 (x, y)-子问题，上述 x-子问题和 y-子问题的规模更小，求解一般要更加容易，特别是当 f 和 g 的邻近算子分别容易计算时。

交替方向乘子法 (ADMM)

回顾 ADMM 所求解的问题类型：

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\, y\in\mathbb{R}^m} \quad & f(x)+g(y) \\ \text{s.t.} \quad & Ax+By=b. \end{aligned}$$

基于上述“逐个击破”的思想，我们可以给出求解问题 (**P**) 的**交替方向乘子法 (Alternating Direction Method of Multipliers, ADMM)**：

$$\begin{cases} x^{k+1} = \arg\min\limits_{x}\ L_\sigma(x,\, y^k,\, \lambda^k), \\ y^{k+1} = \arg\min\limits_{y}\ L_\sigma(x^{k+1},\, y,\, \lambda^k), \\ \lambda^{k+1} = \lambda^k + \tau\sigma\left(Ax^{k+1}+By^{k+1}-b\right), \end{cases}$$

其中 $\sigma>0$ 为罚因子，$\tau\in\left(0,\, \frac{1+\sqrt{5}}{2}\right)$ 为对偶步长。

ADMM 的收敛性

定理 4 (ADMM 的收敛性)

假设：(1) $f(x)$ 和 $g(y)$ 均为适当闭凸函数，问题 (**P**) 的 KKT 系统解集非空；(2) ADMM 迭代的每个子问题强凸。那么，对任意 $\sigma > 0$ 和 $\tau \in \left(0, \frac{1+\sqrt{5}}{2}\right)$，由 ADMM 迭代得到的序列 $\left\{\left(x^k, y^k, \lambda^k\right)\right\}$ 收敛到问题 (**P**) 的一个 KKT 点。

扩展思考：ADMM 是否可以进一步推广，用于求解如下三块（或更多块）可分结构的问题？

$$\begin{aligned} \min_{x,y,z} \quad & f(x) + g(y) + h(z) \\ \text{s.t.} \quad & Ax + By + Cz = d. \end{aligned}$$

答：形式上可以，但不一定保证收敛！ 事实上，已有文献[1]给出了三块 ADMM 不收敛的反例！于是，很多 ADMM 的变种算法被相继提出，基本思路都是对 ADMM 作适当的修正。

[1]参见 “C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Mathematical Programming, 155(1): 57–79, 2016.”

案例：LASSO 问题

下面，我们讨论如何应用 ADMM 求解 LASSO 问题：

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1,$$

其中 $A\in\mathbb{R}^{p\times n}$，$b\in\mathbb{R}^p$ 以及 $\lambda>0$。

思路一：从原问题入手求解。

- 需将原问题等价变换为 ADMM 可求解的问题形式

思路二：从对偶问题入手求解。

- 需先求出对偶问题，再将其等价变换为 ADMM 可求解的问题形式

求解 LASSO 原问题

思路一：从原问题入手求解。

方法 I：引入辅助变量 $y = Ax - b$，将原问题等价变换为：

$$\begin{aligned} \min_{x,\,y} \quad & \underbrace{\lambda\|x\|_1}_{f(x)} + \underbrace{\frac{1}{2}\|y\|^2}_{g(y)}, \\ \text{s.t.} \quad & Ax - b = y. \end{aligned}$$

为约束 $Ax - b = y$ 引入拉格朗日乘子 z，并写出增广拉格朗日函数：

$$L_\sigma(x, y, z) = \lambda\|x\|_1 + \frac{1}{2}\|y\|^2 + \langle z,\, Ax - y - b\rangle + \frac{\sigma}{2}\|Ax - y - b\|^2.$$

求解 LASSO 原问题

这样，可以写出 ADMM 的迭代格式如下：

$$
\begin{cases}
x^{k+1} = \arg\min\limits_{x} \ L_\sigma(x,\, y^k,\, z^k) \\
\qquad\quad = \arg\min\limits_{x} \left\{ \lambda\|x\|_1 + \dfrac{\sigma}{2}\left\| Ax - y^k - b + \sigma^{-1} z^k \right\|^2 \right\} \\
y^{k+1} = \arg\min\limits_{y} \ L_\sigma(x^{k+1},\, y,\, z^k) \\
\qquad\quad = \arg\min\limits_{y} \left\{ \dfrac{1}{2}\|y\|^2 + \dfrac{\sigma}{2}\|Ax^{k+1} - y - b + \sigma^{-1} z^k\|^2 \right\} \\
z^{k+1} = z^k + \tau\sigma\left(Ax^{k+1} - y^{k+1} - b\right)
\end{cases}
$$

可以观察到，基于该变换方法得到的 ADMM，x-子问题本质上还是一个 LASSO 问题，这样将问题求解又绕回了起点，因此该方法显然无法适用！

思考：是否还能应用 ADMM 求解 LASSO 原问题？
回答：还是可以的，需要考虑其他“等价变换形式”！

求解 LASSO 原问题

方法 II：引入辅助变量 $y = x$，将原问题等价变换为：

$$\begin{aligned} \min_{x,\,y} \quad & \underbrace{\frac{1}{2}\|Ax-b\|^2}_{f(x)} + \underbrace{\lambda\|y\|_1}_{g(y)} \\ \text{s.t.} \quad & x = y. \end{aligned}$$

为约束 $x = y$ 引入拉格朗日乘子 z，并写出增广拉格朗日函数：

$$L_\sigma(x, y, z) = \frac{1}{2}\|Ax-b\|^2 + \lambda\|y\|_1 + \langle z,\, x-y\rangle + \frac{\sigma}{2}\|x-y\|^2.$$

求解 LASSO 原问题

这样，可以写出 ADMM 的迭代格式如下：

$$
\begin{cases}
x^{k+1} = \arg\min\limits_{x} \; L_\sigma(x,\, y^k,\, z^k) \\
\qquad\quad = \arg\min\limits_{x} \left\{ \dfrac{1}{2}\|Ax-b\|^2 + \dfrac{\sigma}{2}\left\| x - y^k + \sigma^{-1} z^k \right\|^2 \right\} \\
\qquad\quad = \left(A^\top A + \sigma I\right)^{-1} \left(A^\top b + \sigma y^k - z^k\right) \qquad \leftarrow \text{ 具有显式解!} \\
y^{k+1} = \arg\min\limits_{y} \; L_\sigma(x^{k+1},\, y,\, z^k) \\
\qquad\quad = \arg\min\limits_{y} \left\{ \lambda\|y\|_1 + \dfrac{\sigma}{2}\left\| x^{k+1} - y + \sigma^{-1} z^k \right\|^2 \right\} \\
\qquad\quad = \mathrm{prox}_{\sigma^{-1}\lambda\|\cdot\|_1}\left(x^{k+1} + \sigma^{-1} z^k\right) \qquad \leftarrow \text{ 具有显式解!} \\
z^{k+1} = z^k + \tau\sigma\left(x^{k+1} - y^{k+1}\right)
\end{cases}
$$

可以观察到，基于该变换方法得到的 ADMM，所有子问题都可以有效求解，因此该方法可以在实际中使用！ 数值技巧：在更新 x 时，可先缓存矩阵 $A^\top A + \sigma I$ 的分解以及 $A^\top b$，以减小后续迭代的计算量。

求解 LASSO 对偶问题

思路二：从对偶问题入手求解。

为此，需先导出 LASSO 的对偶问题。引入辅助变量 $y = Ax - b$，将原问题等价变换为：

$$\min_{x,\,y} \quad \lambda\|x\|_1 + \frac{1}{2}\|y\|^2 \qquad \text{s.t.} \qquad Ax - b = y.$$

下面，推导该问题的对偶：

Step 1. 为约束 $Ax - b = y$ 引入 Lagrange 乘子 z，并写出 Lagrange 函数：

$$L(x, y, z) = \lambda\|x\|_1 + \frac{1}{2}\|y\|^2 + \langle z,\, Ax - y - b\rangle.$$

求解 LASSO 对偶问题

Step 2. 推导 Lagrange 对偶函数：

$$\theta(z) = \inf_{x,\,y} L(x, y, z) = \inf_{x,\,y} \left\{ \lambda\|x\|_1 + \frac{1}{2}\|y\|^2 + \langle z,\, Ax - y - b\rangle \right\}$$

$$= \inf_{x} \left\{ \lambda\|x\|_1 + \langle A^\top z,\, x\rangle \right\} + \inf_{y} \left\{ \frac{1}{2}\|y\|^2 - \langle z,\, y\rangle \right\} - \langle z,\, b\rangle.$$

容易分析，得到

$$\inf_{x} \left\{ \lambda\|x\|_1 + \langle A^\top z,\, x\rangle \right\} = \begin{cases} 0, & \|A^\top z\|_\infty \leqslant \lambda, \\ -\infty, & \text{otherwise.} \end{cases}$$

$$\inf_{y} \left\{ \frac{1}{2}\|y\|^2 - \langle z,\, y\rangle \right\} = -\frac{1}{2}\|z\|^2.$$

综上，可以得到 Lagrange 对偶函数：

$$\theta(z) = \begin{cases} -\frac{1}{2}\|z\|^2 - \langle z,\, b\rangle, & \|A^\top z\|_\infty \leq \lambda \\ -\infty, & \text{otherwise.} \end{cases}$$

求解 LASSO 对偶问题

Step 3. 写出 LASSO 的对偶问题：

$$\begin{aligned} \max_{z} \quad & -\frac{1}{2}\|z\|^2 - \langle z,\, b\rangle \\ \text{s.t.} \quad & \|A^\top z\|_\infty \le \lambda \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \min_{z} \quad & \frac{1}{2}\|z\|^2 + \langle z,\, b\rangle \\ \text{s.t.} \quad & \|A^\top z\|_\infty \le \lambda \end{aligned}$$

接下来，为应用 ADMM，将约束 $\|A^\top z\|_\infty \leqslant \lambda$ 用示性函数加入目标函数中，并引入辅助变量 $s = A^\top z$。于是，可将上述对偶问题等价变换为：

$$\begin{aligned} \min_{z,\, s} \quad & \underbrace{\frac{1}{2}\|z\|^2 + \langle z,\, b\rangle}_{f(z)} + \underbrace{\delta_{\|\cdot\|_\infty \le \lambda}(s)}_{g(s)} \\ \text{s.t.} \quad & A^\top z - s = 0. \end{aligned}$$

为约束 $A^\top z - s = 0$ 引入拉格朗日乘子 λ，并写出增广拉格朗日函数：

$$L_\sigma(z, s, \lambda) = \frac{1}{2}\|z\|^2 + \langle z,\, b\rangle + \delta_{\|\cdot\|_\infty \le \lambda}(s) + \langle \lambda,\, A^\top z - s\rangle + \frac{\sigma}{2}\|A^\top z - s\|^2.$$

求解 LASSO 对偶问题

这样，可以写出 ADMM 的迭代格式如下：

$$
\begin{cases}
z^{k+1} = \arg\min\limits_{z} \; L_\sigma(z,\, s^k,\, \lambda^k) \\
\qquad = \arg\min\limits_{z} \left\{ \dfrac{1}{2}\|z\|^2 + \langle z,\, b\rangle + \dfrac{\sigma}{2}\left\| A^\top z - s^k + \sigma^{-1}\lambda^k \right\|^2 \right\} \\
\qquad = \left(I + \sigma A A^\top\right)^{-1} \left(A(\sigma s^k - \lambda^k) - b\right) \qquad \leftarrow \text{ 具有显式解！} \\
s^{k+1} = \arg\min\limits_{s} \; L_\sigma(z^{k+1},\, s,\, \lambda^k) \\
\qquad = \arg\min\limits_{s} \left\{ \delta_{\|\cdot\|_\infty \le \lambda}(s) + \dfrac{\sigma}{2}\left\| A^\top z^{k+1} - s + \sigma^{-1}\lambda^k \right\|^2 \right\} \\
\qquad = \operatorname{prox}_{\|\cdot\|_\infty \le \lambda}\left(A^\top z^{k+1} + \sigma^{-1}\lambda^k\right) \qquad \leftarrow \text{ 具有显式解！} \\
\lambda^{k+1} = \lambda^k + \tau\sigma\left(A^\top z^{k+1} - s^{k+1}\right)
\end{cases}
$$

可以观察到，基于该变换方法得到的 ADMM，所有子问题也都可以有效求解，因此该方法也可以在实际中使用！ **数值技巧**：在更新 z 时，可先缓存矩阵 $I + \sigma A A^\top$ 的分解，以减小后续迭代的计算量。

原始 LASSO vs 对偶 LASSO

LASSO 原问题（$A \in \mathbb{R}^{p\times n}$）

$$\begin{aligned}\min_{x} \quad & \frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1 \\ & \Updownarrow \\ \min_{x,\,y} \quad & \frac{1}{2}\|Ax-b\|^2+\lambda\|y\|_1 \\ \text{s.t.} \quad & x=y.\end{aligned}$$

ADMM for Primal

$$x^{k+1}=\left(A^\top A+\sigma I\right)^{-1}\left(A^\top b+\sigma y^k-z^k\right)$$

$$y^{k+1}=\operatorname{prox}_{\sigma^{-1}\lambda\|\cdot\|_1}\left(x^{k+1}+\sigma^{-1}z^k\right)$$

$$z^{k+1}=z^k+\tau\sigma\left(x^{k+1}-y^{k+1}\right)$$

当 $p \gg n$ 时，应用原始 ADMM

LASSO 对偶问题（$A \in \mathbb{R}^{p\times n}$）

$$\begin{aligned}\min_{z} \quad & \frac{1}{2}\|z\|^2+\langle z,\,b\rangle \ \text{ s.t. } \ \|A^\top z\|_\infty\le\lambda \\ & \Updownarrow \\ \min_{z,\,s} \quad & \frac{1}{2}\|z\|^2+\langle z,\,b\rangle+\delta_{\|\cdot\|_\infty\le\lambda}(s) \\ \text{s.t.} \quad & A^\top z-s=0.\end{aligned}$$

ADMM for Dual

$$z^{k+1}=\left(I+\sigma AA^\top\right)^{-1}\left(A(\sigma s^k-\lambda^k)-b\right)$$

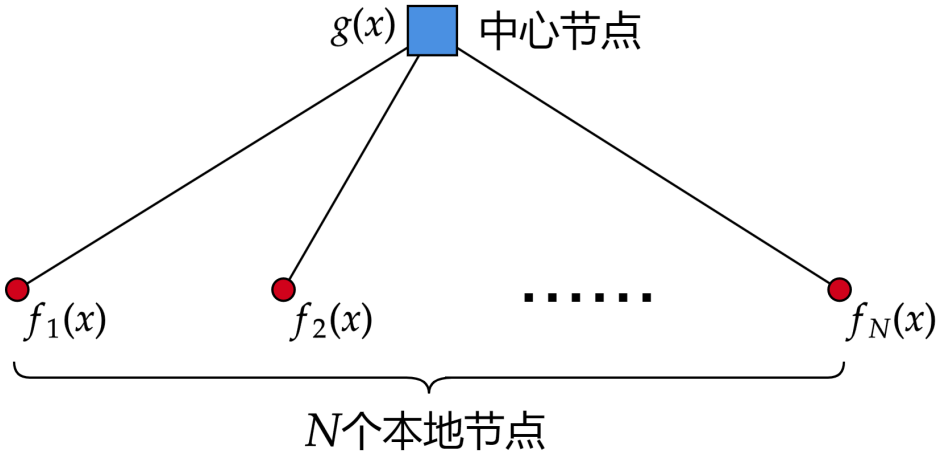
$$s^{k+1}=\operatorname{prox}_{\|\cdot\|_\infty\le\lambda}\left(A^\top z^{k+1}+\sigma^{-1}\lambda^k\right)$$

$$\lambda^{k+1}=\lambda^k+\tau\sigma\left(A^\top z^{k+1}-s^{k+1}\right)$$

当 $p \ll n$ 时，应用对偶 ADMM

- 罚函数法
- 增广拉格朗日乘子法
- 交替方向乘子法
- 分布式优化简介
- 扩展：线性规划内点法简介
- 扩展：对偶算法在最优传输上的应用

分布式优化简介



考虑一个有中心分布式网络上的凸优化问题：

$$\min_x \quad f_0(x) := \sum_{i=1}^{N} f_i(x) + g(x),$$

- $g(x)$ 表示中心服务器持有的正则函数；
- $f_i(x)$ 表示第 i 个本地节点或客户端持有的损失函数。

难点：目标函数的各部分分散在不同节点，但是被变量 x 耦合在一起；

思考：如何利用问题和网络的结构设计有效求解算法？（不考虑异步、时延等通信因素）

分布式优化简介

梯度下降法：（若 $f_i(x)$，$g(x)$ 均连续可微、凸）

$$x^{k+1} = x^k - \alpha \left(\sum_{i=1}^{N} \nabla f_i(x^k) + \nabla g(x^k) \right)$$

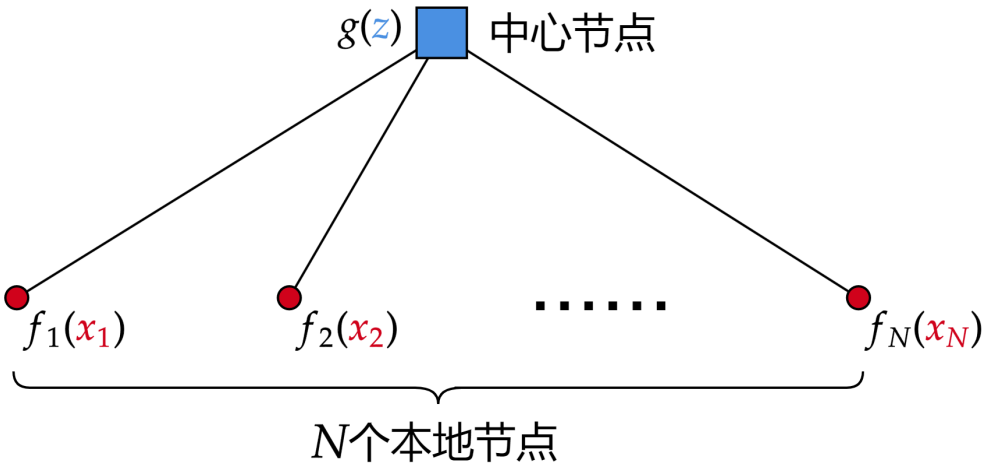
1. 中心将 x^k 发送给本地节点；
2. 本地节点计算 $\nabla f_i(x^k)$，并返回给中心；
3. 中心计算 $\sum_{i=1}^{N} \nabla f_i(x^k)$ 和 $\nabla g(x)$，再执行梯度下降步；

邻近梯度法：（若 $f_i(x)$ 凸且连续可微，$g(x)$ 凸且邻近算子易计算）

$$x^{k+1} = \mathrm{prox}_{\alpha g} \left(x^k - \alpha \sum_{i=1}^{N} \nabla f_i(x^k) \right)$$

1. 中心将 x^k 发送给本地节点；
2. 本地节点计算 $\nabla f_i(x^k)$，并返回给中心；
3. 中心计算 $x^k - \alpha \sum_{i=1}^{N} \nabla f_i\left(x^k\right)$，再执行邻近梯度步；

分布式优化简介



引入“本地复制” x_i , $i = 1, 2, \ldots, N$.

通过辅助变量 z 和约束

$$x_i = z \text{ , } i = 1, 2, \ldots, N.$$

实现一致性约束!

引入 x 的“本地复制” x_i 和“辅助中心变量” z，使目标函数实现分离：

$$\begin{aligned} \min_{x_i,\, z} \quad & \sum_{i=1}^{N} f_i(x_i) + g(z), \\ \text{s.t.} \quad & x_i = z, \quad i = 1, 2, \cdots, N. \end{aligned}$$

分布式优化简介

为约束 $x_i = z$ 引入 Lagrange 乘子 λ_i，并写出增广拉格朗日函数：

$$
\begin{aligned}
&L_\sigma\left(x_1, x_2, \cdots, x_N, z, \lambda_1, \lambda_2, \cdots, \lambda_N\right) \\
&= \sum_{i=1}^{N} f_i(x_i) + g(z) + \sum_{i=1}^{N} \langle \lambda_i,\, x_i - z \rangle + \sum_{i=1}^{N} \frac{\sigma}{2} \|x_i - z\|^2.
\end{aligned}
$$

考虑应用**增广拉格朗日函数法**：

$$
\begin{aligned}
&\left(x_1^{k+1}, x_2^{k+1}, \cdots, x_N^{k+1}, z^{k+1}\right) \\
&\qquad = \underset{\{x_i\}_{i=1}^N,\, z}{\arg\min}\ L_{\sigma_k}\left(x_1, x_2, \cdots, x_N, z, \lambda_1^k, \lambda_2^k, \cdots, \lambda_N^k\right)
\end{aligned}
$$

$$
\lambda_i^{k+1} = \lambda_i^k + \sigma_k(x_i^{k+1} - z^{k+1}), \quad i = 1, 2, \cdots, N.
$$

★ 子问题求解难度较大

考虑应用交替方向乘子法 (ADMM)：

$$
\begin{aligned}
x_i^{k+1} &= \arg\min_{x_i} \; L_\sigma\left(x_i, z^k, \lambda_i^k\right) \quad \leftarrow \text{子问题只与 } x_i \text{ 有关!} \\
&= \arg\min_{x_i} \left\{ f_i(x_i) + \frac{\sigma}{2} \left\| x_i - z^k + \sigma^{-1}\lambda_i^k \right\|^2 \right\}, \quad i = 1, 2, \cdots, N, \\
z^{k+1} &= \arg\min_{z} \; L_\sigma\left(x_1^{k+1}, \cdots, x_N^{k+1}, z, \lambda_1^k, \cdots, \lambda_N^k\right) \\
&= \arg\min_{z} \left\{ g(z) + \frac{\sigma}{2} \sum_{i=1}^{N} \left\| z - x_i^{k+1} - \sigma^{-1}\lambda_i^k \right\|^2 \right\}, \\
\lambda_i^{k+1} &= \lambda_i^k + \tau\sigma(x_i^{k+1} - z^{k+1}), \quad i = 1, 2, \cdots, N.
\end{aligned}
$$

ADMM 在分布式网络中的不同实现方式

1. 本地节点计算 x_i^{k+1} 返回给中心
2. 中心计算 z^{k+1} 发送给本地节点
3. 本地节点计算 λ_i^{k+1} 返回给中心

1. 本地节点计算 x_i^{k+1} 返回给中心
2. 中心计算 z^{k+1} 发送给本地节点
3. 中心计算 λ_i^{k+1} 发送给本地节点

- 罚函数法
- 增广拉格朗日乘子法
- 交替方向乘子法
- 分布式优化简介
- 扩展：线性规划内点法简介
- 扩展：对偶算法在最优传输上的应用

*问题背景

考虑线性规划 (Linear Programming, LP) 问题

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c^\top x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{P}$$

其中 $A \in \mathbb{R}^{m\times n}$，$b \in \mathbb{R}^m$，$c \in \mathbb{R}^n$ 是给定的矩阵和向量。

- 目标函数和约束函数都是线性函数，形式简单；
- 由于 LP 问题的特殊结构，它的解必然在可行域的一个顶点（或某一边界处）取到；
- 在现实中有非常多的应用，求解 LP 问题的算法也非常多，经典算法有单纯形法（Dantzig，1947）和内点法（Karmarkar，1984）

*单纯形法与内点法

单纯形法

- 通过某种方式不断找出可行域的**顶点**，判断其是否为最优解（在可行域边界移动）
- LP 问题可行域的顶点可能多达 $\mathcal{O}(2^n)$ 个（n 为自变量的维数），因此单纯形法在最坏的情况下具有指数级迭代复杂度
- 对于某些大规模问题和病态问题，单纯形法效果可能很差

内点法

- 在可行域**内部**寻找一条路径最终抵达其边界（与单纯形法思想不同）
- 内点法每个迭代步计算代价都远高于仅仅在可行域边界移动的单纯形法
- 但每一步迭代对问题解的改善是显著的，可以证明内点法实际上具有多项式级迭代复杂度
- **原始-对偶算法**是一种比较实用的内点法，在 LP 问题的求解上是单纯形法的有力竞争者

*原始-对偶算法：思想

考虑线性规划的原始问题和对偶问题

$$\textbf{(P)}\quad \begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b, \\ & x \geqslant 0. \end{aligned} \qquad \textbf{(D)}\quad \begin{aligned} \max \quad & b^\top y \\ \text{s.t.} \quad & A^\top y + s = c, \\ & s \geqslant 0. \end{aligned}$$

原始问题与对偶问题的 **KKT** 条件：

KKT 条件

$$Ax = b, \tag{1}$$

$$A^\top y + s = c, \tag{2}$$

$$x_i s_i = 0, \quad i = 1, 2, \cdots, n, \tag{3}$$

$$x \geqslant 0, \quad s \geq 0. \tag{4}$$

*原始-对偶算法：思想

KKT条件

$$Ax = b, \tag{1}$$

$$A^\top y + s = c, \tag{2}$$

$$x_i s_i = 0, \quad i = 1, 2, \cdots, n, \tag{3}$$

$$x \geqslant 0, \quad s \geqslant 0. \tag{4}$$

原始-对偶算法的思想

- 通过近似求解 KKT 系统不断地在可行域的相对内部产生迭代点列
- 每步迭代产生的解满足（1）、（2）、（4）式，而只能近似满足（3）式
- 当（4）式满足且（3）式对任意的 i 不满足时，我们有 $x_i s_i > 0$，$\forall i$，这意味着 (x, s) 为可行域的相对内点 ← 内点法得名的原因

*原始-对偶算法：思想

基于以上思想，互补松弛条件 $x_i s_i = 0$ 在算法迭代过程中不能严格满足。然而，若想要算法最终收敛到 LP 问题的解，我们仍然希望 $x_i s_i \to 0,\ \forall i$。因此，这个条件可以作为内点法的终止条件。

思考： 在具体计算中，该如何刻画这样的终止条件？

我们可以定义该条件的违反度，如下：

定义 2 (对偶间隙)

$$\mu := \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{x^\top s}{n},\quad \forall\, x > 0,\ s > 0.$$

- 当 $\mu \to 0$ 时，有 $x_i s_i \to 0,\ \forall i$。此时，$(x, s)$ 将越来越接近可行域的边界
- 算法构造的解将不断接近满足 KKT 条件，从而逼近最优解

*原始-对偶算法：推导

目标：给定当前可行点 (x, y, s)，寻找下一个点

$$(\tilde{x}, \tilde{y}, \tilde{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s),$$

使得如下条件成立：

$$\begin{cases} A^\top \tilde{y} + \tilde{s} = c, & \tilde{s} > 0, \\ A\tilde{x} = b, & \tilde{x} > 0, \\ \tilde{x}_i \tilde{s}_i = \sigma\mu, & i = 1, 2, \ldots, n, \end{cases}$$

其中 $0 < \sigma < 1$ 是取定的一个常数，μ 是当前点 (x, y, s) 的对偶间隙。该条件也被称为**扰动 KKT 条件**。

最后一个条件：

- 可进一步使用分量乘积简化为 $\tilde{x} \odot \tilde{s} = \sigma\mu\mathbf{1}$
- 可以理解为迭代下一步时对偶间隙将会缩小一个比例 σ

*原始-对偶算法：推导

通过如下方法近似求解上述方程组：

首先，不考虑非负约束，展开方程组可以得到

$$\begin{cases} A(x+\Delta x) = b, \\ A^\top (y+\Delta y) + (s+\Delta s) = c, \\ (s+\Delta s) \odot (x+\Delta x) = \sigma\mu \mathbf{1}. \end{cases}$$

然后，去除高阶非线性项 $\Delta x \odot \Delta s$ 后得到新的近似线性方程组：

$$\begin{cases} A\Delta x = r_p \overset{\text{def}}{=} b - Ax, \\ A^\top \Delta y + \Delta s = r_d \overset{\text{def}}{=} c - s - A^\top y, \\ x \odot \Delta s + s \odot \Delta x = r_c \overset{\text{def}}{=} \sigma\mu \mathbf{1} - x \odot s, \end{cases}$$

其中 $r = (r_p, r_d, r_c)^\top$ 刻画了 KKT 条件的残量。

*原始-对偶算法：推导

记 $L_x = \mathrm{Diag}(x)$，$L_s = \mathrm{Diag}(s)$，我们可将方程组化为矩阵形式

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^\top & I \\ L_s & 0 & L_x \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix},$$

利用矩阵分块消元，可以直接求解上述方程，得到

$$\begin{cases} \Delta y = \left(AL_s^{-1}L_xA^\top\right)^{-1}\left(r_p + AL_s^{-1}\left(L_xr_d - r_c\right)\right), \\ \Delta s = r_d - A^\top \Delta y, \\ \Delta x = -L_s^{-1}\left(L_x\Delta s - r_c\right), \end{cases}$$

其中 $AL_s^{-1}L_xA^\top$ 是对称矩阵，当 A 满秩时，$AL_s^{-1}L_xA^\top$ 正定。

注意：即使当前点 (x, y, z) 是可行的，求解线性方程组产生的新迭代点 $(\tilde{x}, \tilde{y}, \tilde{s})$ 也不一定是可行点！ $\rightarrow$ $\tilde{x} > 0$，$\tilde{s} > 0$ 不能保证成立！

*原始-对偶算法：总结

因此，采取线搜索中的回溯法来确定一个合适的更新：

$$(\tilde{x}, \tilde{y}, \tilde{s}) = (x, y, s) + \alpha_t(\Delta x, \Delta y, \Delta s),$$

其中 $\alpha_t = \alpha_0 \rho^t$，并选取最小的整数 t 使得 $\tilde{x} > 0$，$\tilde{s} > 0$，这里 $0 < \rho < 1$ 和 α_0 是给定常数。

适用于求解线性规划的原始对偶算法可总结为如下过程：

Step 1. 选定初始可行点 $\left(x^0, y^0, s^0\right)$，令 $k = 0$

Step 2. 根据原始对偶算法构造方程组，求解搜索方向 $(\Delta x, \Delta y, \Delta s)$

Step 3. 利用回溯法选取合适的步长 α_k，更新下一个迭代点

$$\left(x^{k+1}, y^{k+1}, s^{k+1}\right) = \left(x^k, y^k, s^k\right) + \alpha_k(\Delta x, \Delta y, \Delta s)$$

Step 4. 若满足停机条件，终止；否则令 $k = k + 1$，转 **Step 2.**

- 罚函数法
- 增广拉格朗日乘子法
- 交替方向乘子法
- 分布式优化简介
- 扩展：线性规划内点法简介
- 扩展：对偶算法在最优传输上的应用

*最优传输问题

考虑最优传输 (Optimal Transport, OT) 问题

$$\min_{\Pi\in\mathbb{R}^{m\times n}} \quad \langle C, \Pi\rangle \quad \text{s.t.} \quad \Pi \mathbf{1}_n = \boldsymbol{a}, \quad \Pi^\top \mathbf{1}_m = \boldsymbol{b}, \quad \Pi \geq 0. \qquad \textbf{(OT)}$$

其中 $C = [c_{ij}] \in \mathbb{R}^{m\times n}$ 是给定的系数矩阵，$\boldsymbol{a} := (a_1, \cdots, a_m)^\top \in \mathbb{R}^m_+$ 和 $\boldsymbol{b} := (b_1, \cdots, b_n)^\top \in \mathbb{R}^n_+$ 是给定的概率向量满足 $\sum_{i=1}^m a_i = 1$ 和 $\sum_{j=1}^n b_j = 1$，$\mathbf{1}_m$ 和 $\mathbf{1}_n$ 表示维度分别为 m 和 n 的元素全为 1 的列向量。

- 该问题本质上是线性规划问题，可用求解线性规划的经典算法来求解：单纯形法；对偶单纯形法；内点法等。
- 困难：OT 问题的决策变量为矩阵 $\Pi \in \mathbb{R}^{m\times n}$，当矩阵维数 m 和 n 增大时，问题的规模将急速变大，传统优化算法将难以应对！

  例： 当 $m = n = 10^4$ 时，OT 问题的变量个数：$mn = 10^8$。

*熵正则化最优传输问题

为了快速求解大规模 OT 问题，熵正则化方法是近年来的主流方法，其基本思路是求解如下的**熵正则化最优传输问题 (Entropic Regularized Optimal Transport, EROT) (Cuturi, NeurIPS 2013)**：

$$\begin{aligned} \min_{\Pi \in \mathbb{R}^{m\times n}} \quad & \langle C, \Pi \rangle + \varepsilon \sum_{i=1}^{m}\sum_{j=1}^{n} \pi_{ij}\left(\log(\pi_{ij}) - 1\right) \\ \text{s.t.} \quad & \Pi \mathbf{1}_n = \boldsymbol{a}, \quad \Pi^\top \mathbf{1}_m = \boldsymbol{b}, \quad \cancel{\Pi \geqslant 0}, \end{aligned} \tag{EROT}$$

其中 $\varepsilon \sum_{i=1}^{m}\sum_{j=1}^{n} \pi_{ij}\left(\log(\pi_{ij}) - 1\right)$ 是熵正则项，ε 是正则系数。

显然，(**EROT**) 可以看作是 (**OT**) 的一个近似问题。 正则系数 ε 越小，近似效果越好！

熵正则化目的：快速计算大规模 OT 问题的一个近似解，满足实际应用需求

思考：为何熵正则化 OT 问题相较原 OT 问题更容易求解？

回答：得益于熵正则项带来的良好性质。

*求解熵正则化 OT 问题

思路一：从 KKT 条件入手求解。为等式约束分别引入乘子 $\boldsymbol{f}$ 和 $\boldsymbol{g}$，于是由 KKT 条件可知，熵正则化 OT 问题的最优解及对应的乘子须满足以下方程：

$$
\begin{aligned}
C+\varepsilon\log(\Pi)-\boldsymbol{f}\mathbf{1}_n^\top-\mathbf{1}_m\boldsymbol{g}^\top=0 \qquad & \Pi=\mathrm{Diag}(e^{\boldsymbol{f}/\varepsilon})\,e^{-C/\varepsilon}\,\mathrm{Diag}(e^{\boldsymbol{g}/\varepsilon}) \\
\Pi\mathbf{1}_n-\boldsymbol{a}=0 \quad \Longleftrightarrow \quad & \Pi\mathbf{1}_n=\boldsymbol{a} \\
\Pi^\top\mathbf{1}_m-\boldsymbol{b}=0 \qquad & \Pi^\top\mathbf{1}_m=\boldsymbol{b}
\end{aligned}
$$

上述式中的 log 和 e 都是作用于矩阵/向量的每个元素，$\mathrm{Diag}(e^{\boldsymbol{f}/\varepsilon})$ 和 $\mathrm{Diag}(e^{\boldsymbol{g}/\varepsilon})$ 表示分别以向量 $e^{\boldsymbol{f}/\varepsilon}$ 和 $e^{\boldsymbol{g}/\varepsilon}$ 为对角线的对角矩阵。

令 $\boldsymbol{u}:=e^{\boldsymbol{f}/\varepsilon}$，$\boldsymbol{v}:=e^{\boldsymbol{g}/\varepsilon}$ 和 $K:=e^{-C/\varepsilon}$，以上 KKT 条件可等价地写为

$$
\begin{aligned}
\big(\mathrm{Diag}(\boldsymbol{u})\,K\,\mathrm{Diag}(\boldsymbol{v})\big)\mathbf{1}_n=\boldsymbol{a} \\
\big(\mathrm{Diag}(\boldsymbol{v})\,K^\top\,\mathrm{Diag}(\boldsymbol{u})\big)\mathbf{1}_m=\boldsymbol{b}
\end{aligned}
\quad\Longleftrightarrow\quad
\begin{aligned}
\boldsymbol{u}\odot K\boldsymbol{v}=\boldsymbol{a} \\
\boldsymbol{v}\odot K^\top\boldsymbol{u}=\boldsymbol{b}
\end{aligned}
$$

其中 $\odot$ 表示向量间对应元素的乘法。至此，(**EROT**) 的求解转化为寻找满足 KKT 条件的向量对 $(\boldsymbol{u},\boldsymbol{v})$。

*求解熵正则化 OT 问题

一个简单的想法：交替地求解上述 KKT 条件中的两个方程！

于是，可以得到一个十分简单的迭代算法：给定一个初始向量 $\boldsymbol{v}^0$（比如 $\boldsymbol{v}^0 = \mathbf{1}_n$），在 k 步，依次计算

$$\boldsymbol{u}^{k+1} = \boldsymbol{a}./K\boldsymbol{v}^k,$$

$$\boldsymbol{v}^{k+1} = \boldsymbol{b}./K^\top \boldsymbol{u}^{k+1},$$

其中 $./$ 表示向量间对应元素的除法。当 $(\boldsymbol{u}^{k+1}, \boldsymbol{v}^{k+1})$ 满足某种终止准则后，再根据 KKT 条件还原得到原问题的一个近似最优解：

$$\Pi^{k+1} = \mathrm{Diag}(\boldsymbol{u}^{k+1})\, K\, \mathrm{Diag}(\boldsymbol{v}^{k+1}).$$

上述算法便是著名的 **Sinkhorn's Algorithm** [Sinkhorn, 1967]，由 Marco Cuturi 在 2013 年首次应用于求解熵正则化 OT 问题 [Cuturi, NeurIPS '13]，取得了巨大成功，引发了 OT 研究的新一轮浪潮。

*求解熵正则化 OT 问题

思路二：从对偶问题入手求解。

首先，分别为约束 $\Pi \mathbf{1}_n = \boldsymbol{a}$ 和 $\Pi^\top \mathbf{1}_m = \boldsymbol{b}$ 引入 Lagrange 乘子 $\boldsymbol{f}$ 和 $\boldsymbol{g}$，写出 Lagrange 函数：

$$L(\Pi, \boldsymbol{f}, \boldsymbol{g}) = \langle C, \Pi \rangle + \varepsilon \sum_{i=1}^{m} \sum_{j=1}^{n} \pi_{ij} \left(\log(\pi_{ij}) - 1 \right) + \langle \boldsymbol{f}, \boldsymbol{a} - \Pi \mathbf{1}_n \rangle + \langle \boldsymbol{g}, \boldsymbol{b} - \Pi^\top \mathbf{1}_m \rangle.$$

进一步，可以写出对偶问题：

$$\min_{\boldsymbol{f} \in \mathbb{R}^m, \boldsymbol{g} \in \mathbb{R}^n} \quad Q(\boldsymbol{f}, \boldsymbol{g}) := \varepsilon \langle e^{\boldsymbol{f}/\varepsilon}, K e^{\boldsymbol{g}/\varepsilon} \rangle - \langle \boldsymbol{f}, \boldsymbol{a} \rangle - \langle \boldsymbol{g}, \boldsymbol{b} \rangle.$$

可以观察到，虽然熵正则化 OT 问题自身是约束优化问题，但其对偶问题是可微无约束优化问题，故可利用上一章学习的无约束优化算法进行求解！

*求解熵正则化 OT 问题

特别地，基于对偶目标函数 $Q(\boldsymbol{f}, \boldsymbol{g})$ 具有两块变量这一结构，一个简单的想法就是交替地极小化 $Q(\boldsymbol{f}, \boldsymbol{g})$，得到求解对偶问题的**分块坐标下降法：**

$$\begin{cases} \boldsymbol{f}^{k+1} = \arg\min\limits_{\boldsymbol{f}} Q(\boldsymbol{f}, \boldsymbol{g}^k), \\ \boldsymbol{g}^{k+1} = \arg\min\limits_{\boldsymbol{g}} Q(\boldsymbol{f}^{k+1}, \boldsymbol{g}). \end{cases}$$

根据子问题的一阶最优性条件，有

$$\nabla_{\boldsymbol{f}} Q(\boldsymbol{f}^{k+1}, \boldsymbol{g}^k) = e^{\boldsymbol{f}^{k+1}/\varepsilon} \odot K e^{\boldsymbol{g}^k/\varepsilon} - \boldsymbol{a} = 0$$
$$\nabla_{\boldsymbol{g}} Q(\boldsymbol{f}^{k+1}, \boldsymbol{g}^{k+1}) = e^{\boldsymbol{g}^{k+1}/\varepsilon} \odot K^\top e^{\boldsymbol{f}^{k+1}/\varepsilon} - \boldsymbol{b} = 0$$

$$\Longleftrightarrow \quad \begin{aligned} e^{\boldsymbol{f}^{k+1}/\varepsilon} &= \boldsymbol{a}./\left(K e^{\boldsymbol{g}^k/\varepsilon}\right) \\ e^{\boldsymbol{g}^{k+1}/\varepsilon} &= \boldsymbol{b}./\left(K^\top e^{\boldsymbol{f}^{k+1}/\varepsilon}\right) \end{aligned} \quad \underset{\boldsymbol{u}^k := e^{\boldsymbol{f}^k/\varepsilon}, \boldsymbol{v}^k := e^{\boldsymbol{g}^k/\varepsilon}}{\Longleftrightarrow} \quad \begin{aligned} \boldsymbol{u}^{k+1} &= \boldsymbol{a}./K\boldsymbol{v}^k \\ \boldsymbol{v}^{k+1} &= \boldsymbol{b}./K^\top \boldsymbol{u}^{k+1} \end{aligned}$$

其中 $K := e^{-C/\varepsilon}$，符号 $\odot$ 表示向量间对应元素的乘法，符号 $./$ 表示向量间对应元素的除法。

*求解熵正则化 OT 问题的启示

不难看出，求解对偶问题的块坐标下降法等价于 **Sinkhorn's Algorithm**

从上述求解熵正则化 OT 问题的两个思路可以看出，对于一些具有特殊结构的约束优化问题，我们可以考虑

- 从 KKT 条件入手：求解原约束问题 $\longrightarrow$ 求解方程
- 从对偶问题入手：求解原约束问题 $\longrightarrow$ 求解无约束问题

但同时也需要注意的是，这些方法实施的重要前提是待求解的问题是**凸优化问题**！这也进一步说明凸优化的重要性！