

第七章 复合优化算法

修贤超

<https://xianchaoxiu.github.io>

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

■ 考虑如下复合优化问题

$$\min_{x \in \mathbb{R}^n} \quad \psi(x) = f(x) + h(x)$$

□ $f(x)$ 为可微函数 (可能非凸)

□ $h(x)$ 可能为不可微函数

■ 定义 7.1 对于一个凸函数 h , 定义邻近算子为

$$\text{prox}_h(x) = \arg \min_u \left\{ h(u) + \frac{1}{2} \|u - x\|_2^2 \right\}$$

■ 定理 7.1 如果 h 为闭凸函数, 则对任意 x 有 $\text{prox}_h(x)$ 存在且唯一

■ **定理 7.2** 若 h 是适当的闭凸函数, 则

$$u = \text{prox}_h(x) \quad \Leftrightarrow \quad x - u \in \partial h(u)$$

证明 若 $u = \text{prox}_h(x)$, 则由最优性条件得 $0 \in \partial h(u) + (u - x)$, 因此 $x - u \in \partial h(u)$. 反之, 若 $x - u \in \partial h(u)$ 则由**次梯度**的定义可得到

$$h(v) \geq h(u) + (x - u)^\top (v - u), \quad \forall v \in \text{dom } h$$

两边同时加 $\frac{1}{2}\|v - x\|^2$, 即有

$$\begin{aligned} h(v) + \frac{1}{2}\|v - x\|^2 &\geq h(u) + (x - u)^\top (v - u) + \frac{1}{2}\|(v - u) - (x - u)\|^2 \\ &\geq h(u) + \frac{1}{2}\|u - x\|^2, \quad \forall v \in \text{dom } h \end{aligned}$$

根据定义可得 $u = \text{prox}_h(x)$

例 7.1

■ 给定 ℓ_1 范数 $h(x) = t\|x\|_1$, 则 $\text{prox}_{th}(x) = \text{sign}(x) \max\{|x| - t, 0\}$

证明 邻近算子 $u = \text{prox}_{th}(x)$ 的最优性条件为

$$x - u \in t\partial\|u\|_1 = \begin{cases} \{t\}, & u > 0 \\ [-t, t], & u = 0 \\ \{-t\}, & u < 0 \end{cases}$$

\Downarrow

$$u = \begin{cases} x - t, & x > t \\ x + t, & x < -t \\ 0, & x \in [-t, t] \end{cases}$$

例 7.1

■ 给定 ℓ_2 范数 $h(x) = t\|x\|_2$, 则 $\text{prox}_{th}(x) = \begin{cases} (1 - \frac{t}{\|x\|_2})x, & \|x\|_2 \geq t \\ 0, & \text{其他} \end{cases}$

证明 邻近算子 $u = \text{prox}_{th}(x)$ 的最优性条件为

$$x - u \in t\partial\|u\|_2 = \begin{cases} \{\frac{tu}{\|u\|_2}\}, & u \neq 0 \\ \{w : \|w\|_2 \leq t\}, & u = 0 \end{cases}$$

\Downarrow

$$u = \begin{cases} x - \frac{tx}{\|x\|_2}, & \|x\|_2 > t \\ 0, & \|x\|_2 \leq t \end{cases}$$

■ 邻近算子的计算规则

- ▣ 变量的常数倍放缩以及平移 ($\lambda \neq 0$)

$$h(x) = g(\lambda x + a), \quad \text{prox}_h(x) = \frac{1}{\lambda} \left(\text{prox}_{\lambda^2 g}(\lambda x + a) - a \right)$$

- ▣ 函数（及变量）的常数倍放缩 ($\lambda > 0$)

$$h(x) = \lambda g\left(\frac{x}{\lambda}\right), \quad \text{prox}_h(x) = \lambda \text{prox}_{\lambda^{-1}g}\left(\frac{x}{\lambda}\right)$$

- ▣ 加上线性函数

$$h(x) = g(x) + a^\top x, \quad \text{prox}_h(x) = \text{prox}_g(x - a)$$

例 7.2

- 加上二次项 ($u > 0$)

$$h(x) = g(x) + \frac{u}{2} \|x - a\|_2^2, \quad \text{prox}_h(x) = \text{prox}_{\theta g}(\theta x + (1 - \theta)a)$$

其中 $\theta = \frac{1}{1+u}$

- 向量函数

$$h\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \varphi_1(x) + \varphi_2(y), \quad \text{prox}_h\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} \text{prox}_{\varphi_1}(x) \\ \text{prox}_{\varphi_2}(y) \end{bmatrix}$$

例 7.3

- 设 C 为闭凸集, 则示性函数 I_C 的邻近算子为点 x 到 C 的投影 $\mathcal{P}_C(x)$

$$\begin{aligned}\operatorname{prox}_{I_C}(x) &= \arg \min_u \left\{ I_C(u) + \frac{1}{2} \|u - x\|^2 \right\} \\ &= \arg \min_{u \in C} \|u - x\|^2 \\ &= \mathcal{P}_C(x)\end{aligned}$$

- 几何意义

$$u = \mathcal{P}_C(x) \quad \Leftrightarrow \quad (x - u)^\top (z - u) \leq 0, \quad \forall z \in C$$

近似点梯度法

■ 考虑复合优化问题

$$\min_{x \in \mathbb{R}^n} \quad \psi(x) = f(x) + h(x)$$

■ 对于光滑部分 f 做梯度下降, 对于非光滑部分 h 使用邻近算子

=====

算法 7.1 近似点梯度法

- 1 给定函数 $f(x), h(x)$, 初始点 x^0
- 2 **while** 未达到收敛准则 **do**
- 3 $x^{k+1} = \text{prox}_{t_k h}(x^k - t_k \nabla f(x^k))$
- 4 **end while**

对近似点梯度法的理解

■ 把迭代公式展开

$$x^{k+1} = \text{prox}_{t_k h}(x^k - t_k \nabla f(x^k))$$

\Downarrow

$$\begin{aligned} x^{k+1} &= \arg \min_u \left\{ h(u) + \frac{1}{2t_k} \|u - x^k + t_k \nabla f(x^k)\|^2 \right\} \\ &= \arg \min_u \left\{ h(u) + f(x^k) + \nabla f(x^k)^\top (u - x^k) + \frac{1}{2t_k} \|u - x^k\|^2 \right\} \end{aligned}$$

■ 根据邻近算子与次梯度的关系, 可改写为

$$x^{k+1} = x^k - t_k \nabla f(x^k) - t_k g^k, \quad g^k \in \partial h(x^{k+1})$$

■ 对光滑部分做显式的梯度下降, 对非光滑部分做隐式的梯度下降

步长选取

- 当 f 为梯度 L -利普希茨连续函数时, 可取固定步长 $t_k = t \leq \frac{1}{L}$
- 当 L 未知时可使用线搜索准则

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^\top (x^{k+1} - x^k) + \frac{1}{2t_k} \|x^{k+1} - x^k\|^2$$

- BB 步长
- 可构造如下适用于近似点梯度法的非单调线搜索准则

$$\psi(x^{k+1}) \leq C^k - \frac{c_1}{2t_k} \|x^{k+1} - x^k\|^2$$

应用举例: LASSO 问题

- 考虑用近似点梯度法求解 LASSO 问题

$$\min_x \quad \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|^2$$

- 令 $f(x) = \frac{1}{2} \|Ax - b\|^2$, $h(x) = \mu \|x\|_1$, 则

$$\begin{aligned}\nabla f(x) &= A^\top (Ax - b) \\ \text{prox}_{t_k h}(x) &= \text{sign}(x) \max\{|x| - t_k \mu, 0\}\end{aligned}$$

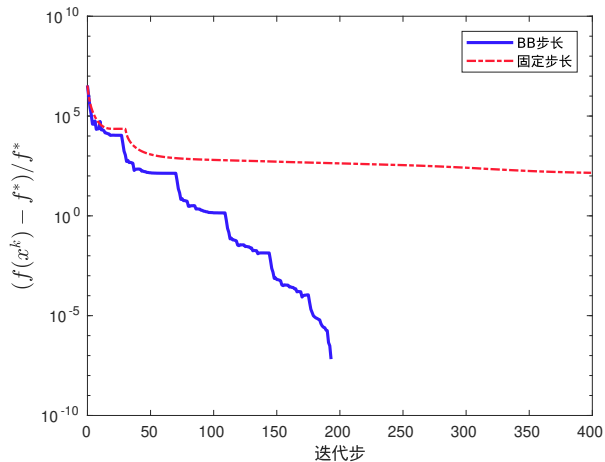
- 相应的迭代格式为

$$\begin{aligned}y^k &= x^k - t_k A^\top (Ax^k - b) \\ x^{k+1} &= \text{sign}(y^k) \max\{|y^k| - t_k \mu, 0\}\end{aligned}$$

即第一步做梯度下降, 第二步做收缩

应用举例: LASSO 问题

■ 使用 BB 步长加速收敛



应用举例：低秩矩阵恢复

■ 考虑低秩矩阵恢复模型

$$\min_{X \in \mathbb{R}^{m \times n}} \mu \|X\|_* + \frac{1}{2} \sum_{(i,j) \in \Omega} (X_{ij} - M_{ij})^2$$

■ 令

$$f(X) = \frac{1}{2} \sum_{(i,j) \in \Omega} (X_{ij} - M_{ij})^2, \quad h(X) = \mu \|X\|_*$$

■ 定义矩阵

$$P_{ij} = \begin{cases} 1, & (i,j) \in \Omega \\ 0, & \text{其他} \end{cases}$$

则

$$f(X) = \frac{1}{2} \|P \odot (X - M)\|_F^2$$

应用举例：低秩矩阵恢复

- 进一步可以得到

$$\begin{aligned}\nabla f(X) &= P \odot (X - M) \\ \text{prox}_{t_k h}(X) &= U \text{Diag}(\max\{|d| - t_k \mu, 0\}) V^\top\end{aligned}$$

- 得到近似点梯度法的迭代格式

$$\begin{aligned}Y^k &= X^k - t_k P \odot (X^k - M) \\ X^{k+1} &= \text{prox}_{t_k h}(Y^k)\end{aligned}$$

收敛性分析

■ 假设 7.1 为了保证近似点梯度算法的收敛性

□ f 在 \mathbb{R}^n 上是凸的; ∇f 为 L -利普希茨连续, 即

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y$$

□ h 是适当的闭凸函数

□ 函数 $\psi(x) = f(x) + h(x)$ 的最小值 ψ^* 是有限的, 并且在点 x^* 处取到

■ 定理 7.3 在假设 7.1 下, 取定步长为 $t_k = t \in (0, \frac{1}{L}]$, 设 $\{x^k\}$ 为迭代产生序列, 则

$$\psi(x^k) - \psi^* \leq \frac{1}{2kt} \|x^0 - x^*\|^2$$

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

典型问题形式

■ 考虑如下复合优化问题

$$\min_{x \in \mathbb{R}^n} \quad \psi(x) = f(x) + h(x)$$

□ $f(x)$ 是连续可微的凸函数, 且梯度是利普西茨连续的

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

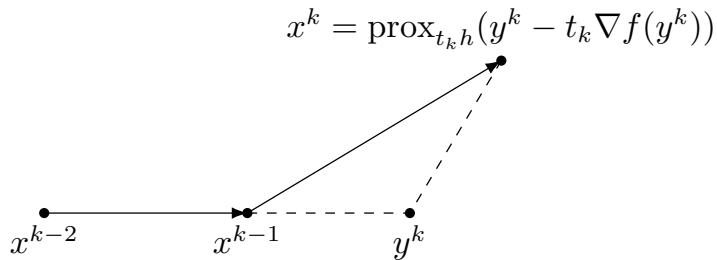
□ $h(x)$ 是适当的闭凸函数, 且邻近算子

$$\text{prox}_h(x) = \arg \min_{u \in \text{dom} h} \left\{ h(u) + \frac{1}{2} \|x - u\|^2 \right\}$$

■ 步长取常数 $t_k = 1/L$ 时, 近似点梯度法的收敛速度为 $\mathcal{O}(1/k)$

Nesterov 加速算法简史

- Nesterov 在 1983、1988、2005 提出了三种改进的一阶算法，收敛速度 $\mathcal{O}\left(\frac{1}{k^2}\right)$
- Beck 和 Teboulle 在 2008 年提出了 FISTA 算法，第一步沿着前两步的计算方向计算一个新点，第二步在该新点处做一步近似点梯度迭代



[引用] A method for solving the convex programming problem with convergence rate $O(1/k^2)$

[Y Nesterov](#) - Dokl akad nauk Sssr, 1983 - [cir.nii.ac.jp](#)

A **method** for **solving** the **convex programming problem** with convergence rate $o(1/k^2)$ | CiNii Research ... A **method** for **solving** the **convex programming problem** with convergence rate $o(1/k^2)$...

☆ 保存 ㊄ 引用 被引用次数: 5901 相关文章 所有 5 个版本 ㊄

A fast iterative shrinkage-thresholding algorithm for linear inverse problems

[A Beck](#), [M Teboulle](#) - SIAM journal on imaging sciences, 2009 - SIAM

... algorithm **FISTA** and ... that **FISTA** can be even faster than the proven theoretical rate and can outperform ISTA by several orders of magnitude, thus showing the potential promise of **FISTA**...

☆ 保存 ㊄ 引用 被引用次数: 13863 相关文章 所有 27 个版本

算法 7.1 近似点梯度法

```
1 给定函数  $f(x), h(x)$ , 初始点  $x^0$   
2 while 未达到收敛准则 do  
3    $x^{k+1} = \text{prox}_{t_k h}(x^k - t_k \nabla f(x^k))$   
4 end while
```

=====

算法 7.2 FISTA 算法

```
1 输入  $x^0 = x^{-1} \in \mathbb{R}^n, k \leftarrow 1$   
2 while 未达到收敛准则 do  
3 计算  $y^k = x^{k-1} + \frac{k-2}{k+1}(x^{k-1} - x^{k-2})$   
4 选取  $t_k = t \in (0, 1/L]$ , 计算  $x^k = \text{prox}_{t_k h}(y^k - t_k \nabla f(y^k))$   
5  $k \leftarrow k + 1$   
6 end while
```

算法 7.3 FISTA 算法的等价变形

- 1 输入 $v^0 = x^0 \in \mathbb{R}^n, k \leftarrow 1$
- 2 **while** 未达到收敛准则 **do**
- 3 计算 $y^k = (1 - \gamma_k)x^{k-1} + \gamma_k v^{k-1}$
- 4 选取 t_k , 计算 $x^k = \text{prox}_{t_k h}(y^k - t_k \nabla f(y^k))$
- 5 计算 $v^k = x^{k-1} + \frac{1}{\gamma_k}(x^k - x^{k-1})$
- 6 $k \leftarrow k + 1$
- 7 **end while**

第二类 Nesterov 加速算法

■ 第二类 Nesterov 加速算法

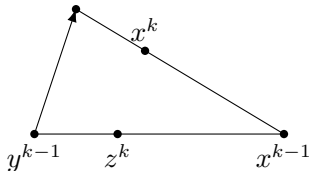
$$z^k = (1 - \gamma_k)x^{k-1} + \gamma_k y^{k-1}$$

$$y^k = \text{prox}_{(t_k/\gamma_k)h} \left(y^{k-1} - \frac{t_k}{\gamma_k} \nabla f(z^k) \right)$$

$$x^k = (1 - \gamma_k)x^{k-1} + \gamma_k y^k$$

■ 三个序列 $\{x^k\}$, $\{y^k\}$ 和 $\{z^k\}$ 都可以保证在定义域内

$$y^k = \text{prox}_{(t_k/\gamma_k)h} (y^{k-1} - (t_k/\gamma_k) \nabla f(z^k))$$



第三类 Nesterov 加速算法

■ 第三类 Nesterov 加速算法

$$z^k = (1 - \gamma_k)x^{k-1} + \gamma_k y^{k-1}$$

$$y^k = \text{prox}_{(t_k \sum_{i=1}^k 1/\gamma_i)h} \left(-t_k \sum_{i=1}^k \frac{1}{\gamma_i} \nabla f(z^i) \right)$$

$$x^k = (1 - \gamma_k)x^{k-1} + \gamma_k y^k$$

■ 计算 y^k 时需要利用全部已有的 $\{\nabla f(z^i)\}, i = 1, 2, \dots, k$

■ 取 $\gamma_k = \frac{2}{k+1}$, $t_k = \frac{1}{L}$ 时, 也有 $\mathcal{O}\left(\frac{1}{k^2}\right)$ 的收敛速度

针对非凸问题的 Nesterov 加速算法

- 考虑 $f(x)$ 是非凸函数，但可微且梯度是利普希茨连续
- 非凸复合优化问题的加速梯度法框架

$$\begin{aligned}z^k &= \gamma_k y^{k-1} + (1 - \gamma_k) x^{k-1} \\y^k &= \text{prox}_{\lambda_k h}(y^{k-1} - \lambda_k \nabla f(z^k)) \\x^k &= \text{prox}_{t_k h}(z^k - t_k \nabla f(z^k))\end{aligned}$$

- 当 λ_k 和 t_k 取特定值时，它等价于第二类 Nesterov 加速算法
- 当 f 为凸函数，收敛速度为 $\mathcal{O}\left(\frac{1}{k^2}\right)$
- 当 f 为非凸函数，收敛速度为 $\mathcal{O}\left(\frac{1}{k}\right)$

应用举例: LASSO 问题求解

- 考虑 LASSO 问题

$$\min_x \quad \frac{1}{2} \|Ax - b\|_2^2 + \mu \|x\|_1$$

- FISTA 算法可以由下面的迭代格式给出

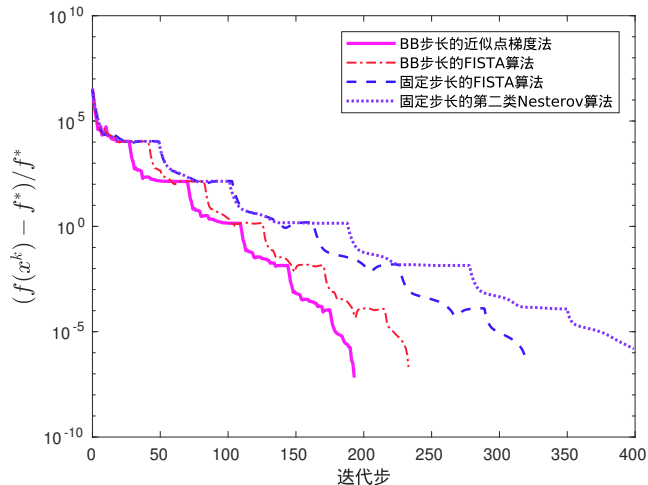
$$y^k = x^{k-1} + \frac{k-2}{k+1}(x^{k-1} - x^{k-2})$$

$$w^k = y^k - t_k A^\top (Ay^k - b)$$

$$x^k = \text{sign}(w^k) \max\{|w^k| - t_k \mu, 0\}$$

应用举例: LASSO 问题求解

- 取 $\mu = 10^{-3}$, 步长 $t = \frac{1}{L}$, 其中 $L = \lambda_{\max}(A^T A)$



收敛性分析

- **定理 7.5** 在假设 7.1 下, 当用 FISTA 算法求解凸复合优化问题时, 若取固定步长 $t_k = 1/L$, 则

$$\psi(x^k) - \psi(x^*) \leq \frac{2L}{(k+1)^2} \|x^0 - x^*\|^2$$

- **推论 7.1** 在假设 7.1 下, 当用 FISTA 算法求解凸复合优化问题时, 若迭代点 x^k, y^k , 步长 t_k 以及组合系数 γ_k 满足一定条件, 则

$$\psi(x^k) - \psi(x^*) \leq \frac{C}{k^2}$$

其中 C 仅与函数 f 和初始点 x^0 的选取有关

- 采用线搜索的 FISTA 算法具有 $\mathcal{O}\left(\frac{1}{k^2}\right)$ 的收敛速度

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

近似点算法

- 考虑一般形式的优化问题

$$\min_x \psi(x)$$

- ψ 是一个适当的闭凸函数，**并不要求连续或可微**
- 次梯度法求解收敛较慢，且收敛条件苛刻
- **近似点梯度法**做隐性的梯度下降

$$\begin{aligned} x^{k+1} &= \text{prox}_{t_k \psi}(x^k) \\ &= \arg \min_u \left\{ \psi(u) + \frac{1}{2t_k} \|u - x^k\|_2^2 \right\} \end{aligned}$$

- $\psi(x)$ 的邻近算子一般需要通过迭代求解
- 目标函数强凸，相比原问题更利于迭代法的求解

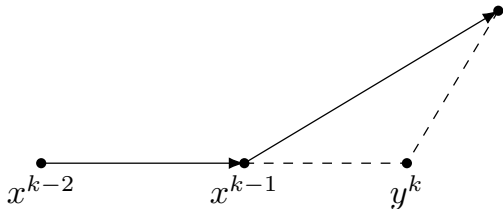
- 用 FISTA 算法对近似点算法进行加速，其迭代格式为

$$x^{k+1} = \text{prox}_{t_k \psi}(x^k)$$

\Downarrow

$$x^k = \text{prox}_{t_k \psi} \left(x^{k-1} + \gamma_k \frac{1 - \gamma_{k-1}}{\gamma_{k-1}} (x^{k-1} - x^{k-2}) \right)$$

$$x^k = \text{prox}_{t_k h}(y^k - t_k \nabla f(y^k))$$



- 第二类 Nesterov 加速算法的迭代格式可以写成

$$v^k = \text{prox}_{(t_k/\gamma_k)\psi}(v^{k-1}), \quad x^k = (1 - \gamma_k) x^{k-1} + \gamma_k v^k$$

- 关于算法参数的选择有两种策略

- 固定步长 $t_k = t$ 以及 $\gamma_k = \frac{2}{k+1}$

- 可变步长 t_k , 当 $k = 1$ 时取 $\gamma_1 = 1$; 当 $k > 1$ 时, γ_k 来自

$$\frac{(1 - \gamma_k) t_k}{\gamma_k^2} = \frac{t_{k-1}}{\gamma_{k-1}^2}$$

与增广拉格朗日函数法的关系

- 考虑具有如下形式的优化问题

$$\min_{x \in \mathbb{R}^n} f(x) + h(Ax)$$

- 例 7.4 一些常见例子

- 当 h 是单点集 $\{b\}$ 的示性函数时, 等价于线性等式约束优化问题

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad Ax = b$$

- 当 h 是凸集 C 上的示性函数时, 等价于凸集约束问题

$$\min f(x) \quad \text{s.t.} \quad Ax \in C$$

- 当 $h(y) = \|y - b\|$ 时, 等价于正则优化问题

$$\min f(x) + \|Ax - b\|$$

对偶问题

■ 原问题的增广拉格朗日函数法

$$(x^{k+1}, y^{k+1}) = \operatorname{argmin}_{x, y} \left\{ f(x) + h(y) + \frac{t_k}{2} \|Ax - y + z^k / t_k\|_2^2 \right\}$$
$$z^{k+1} = z^k + t_k (Ax^{k+1} - y^{k+1})$$

■ 对偶问题

$$\max \quad \psi(z) = \inf_{x, y} L(x, y, z) = -f^*(-A^\top z) - h^*(z)$$

近似点算法

$$z^{k+1} = \operatorname{prox}_{t\psi}(z^k) = \operatorname{argmin}_z \left\{ f^*(-A^\top z) + h^*(z) + \frac{1}{2t_k} \|z - z^k\|_2^2 \right\}$$

■ 原问题用增广拉格朗日函数法 \Leftrightarrow 对偶问题用近似点算法

应用举例: LASSO 问题

■ 考虑 LASSO 问题

$$\min_{x \in \mathbb{R}^n} \psi(x) = \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2$$

■ 引入变量 $y = Ax - b$, 等价地转化为

$$\min_{x, y} f(x, y) = \mu \|x\|_1 + \frac{1}{2} \|y\|_2^2 \quad \text{s.t.} \quad Ax - y - b = 0$$

■ 采用近似点算法进行求解, 第 k 步迭代为

$$(x^{k+1}, y^{k+1}) \approx \arg \min_{(x, y) \in \mathbb{D}} \left\{ f(x, y) + \frac{1}{2t_k} (\|x - x^k\|_2^2 + \|y - y^k\|_2^2) \right\}$$

其中 $\mathbb{D} = \{(x, y) \mid Ax - y = b\}$ 为可行域, t_k 为步长

应用举例: LASSO 问题

- 除了直接求解, 还可以通过偶问题求解
- 引入拉格朗日乘子 z , 对偶函数为

$$\begin{aligned}\Phi_k(z) &= \inf_x \left\{ \mu \|x\|_1 + z^\top A x + \frac{1}{2t_k} \|x - x^k\|_2^2 \right\} \\ &\quad + \inf_y \left\{ \frac{1}{2} \|y\|_2^2 - z^\top y + \frac{1}{2t_k} \|y - y^k\|_2^2 \right\} - b^\top z \\ &= \mu \Gamma_{\mu t_k}(x^k - t_k A^\top z) - \frac{1}{2t_k} \left(\|x_k - t_k A^\top z\|_2^2 - \|x_k\|_2^2 \right) \\ &\quad - \frac{1}{2(t_k + 1)} (\|z\|_2^2 + 2(y^k)^\top z - \|y^k\|_2^2) - b^\top z\end{aligned}$$

其中

$$\Gamma_{\mu t_k}(u) = \inf_x \left\{ \|x\|_1 + \frac{1}{2\mu t_k} \|x - u\|_2^2 \right\}$$

应用举例: LASSO 问题

- 记函数 $q_{\mu t_k} : \mathbb{R} \rightarrow \mathbb{R}$ 为

$$q_{\mu t_k}(v) = \begin{cases} \frac{v^2}{2\mu t_k}, & |v| \leq t \\ |v| - \frac{\mu t_k}{2}, & |v| > t \end{cases}$$

- 易知 $\Gamma_{\mu t_k}(u) = \sum_{i=1}^m q_{\mu t_k}(u_i)$ 是关于 u 的连续可微函数且导数为

$$\nabla_u \Gamma_{\mu t_k}(u) = u - \text{prox}_{\mu t_k \|x\|_1}(u)$$

- 对偶问题为

$$\min_z \Phi_k(z)$$

应用举例: LASSO 问题

- 设对偶问题的逼近最优解为 z^{k+1} , 根据最优性条件有

$$\begin{cases} x^{k+1} = \text{prox}_{\mu t_k \|x\|_1}(x^k - t_k A^T z^{k+1}) \\ y^{k+1} = \frac{1}{t_k + 1}(y^k + t_k z^{k+1}) \end{cases}$$

- 在第 k 步迭代, LASSO 问题的近似点算法的迭代格式

$$\begin{cases} z^{k+1} \approx \arg \max_z \Phi_k(z) \\ x^{k+1} = \text{prox}_{\mu t_k \|x\|_1}(x^k - t_k A^\top z^{k+1}) \\ y^{k+1} = \frac{1}{t_k + 1}(y^k + t_k z^{k+1}) \end{cases}$$

- 根据 $\Phi_k(z)$ 的连续可微性, 可以调用梯度法进行求解

收敛性分析

- **定理 7.6** 设 ψ 是闭凸函数, 最优值 ψ^* 有限且在 x^* 取到, 则对近似点算法有

$$\psi(x^k) - \psi^* \leq \frac{\|x^{(0)} - x^*\|_2^2}{2 \sum_{i=1}^k t_i}, \quad \forall k \geq 1$$

- 若 $\sum_{i=1}^k t_i \rightarrow \infty$, 则算法收敛
- 若 t_i 固定或在一个正下界以上变化, 则收敛速率为 $\mathcal{O}(\frac{1}{k})$
- 虽然 t_i 可以任意选取, 邻近算子的计算代价依赖于 t_i

加速版本的收敛性分析

- **定理 7.7** 设 ψ 是闭凸函数, 最优值 ψ^* 有限且在 x^* 处取到. 假设参数 t_k, γ_k 按照 Nesterov 加速策略选取, 那么

$$\psi(x^k) - \psi^* \leq \frac{2\|x^{(0)} - x^*\|_2^2}{(2\sqrt{t_1} + \sum_{i=2}^k \sqrt{t_i})^2}, \quad k \geq 1$$

- 若 $\sum_{i=2}^k \sqrt{t_i} \rightarrow \infty$, 则保证收敛
- 步长 t_i 取固定值或有正下界时, 其收敛速度可达到 $\mathcal{O}\left(\frac{1}{k^2}\right)$

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

问题形式

■ 考虑具有如下形式的问题

$$\min_{x \in \mathcal{X}} \quad F(x_1, x_2, \dots, x_s) = f(x_1, x_2, \dots, x_s) + \sum_{i=1}^s r_i(x_i)$$

- f 是关于 x 的可微函数，但不一定凸
- $r_i(x_i)$ 关于 x_i 是适当的闭凸函数，但不一定可微

■ 挑战和难点

- 在非凸问题上，很多针对凸问题设计的算法通常会失效
- 目标函数的整体结构十分复杂，变量的更新需要很大计算量

- 例 7.5 设参数 $x = (x_1, x_2, \dots, x_G) \in \mathbb{R}^p$, 分组 LASSO 模型

$$\min_x \quad \frac{1}{2n} \|b - Ax\|_2^2 + \lambda \sum_{i=1}^G \sqrt{p_i} \|x_i\|_2$$

- 例 7.6 设 $b \in \mathbb{R}^m$ 是已知的观测向量, 低秩矩阵恢复模型

$$\min_{X,Y} \quad \frac{1}{2} \|\mathcal{A}(XY) - b\|_2^2 + \alpha \|X\|_F^2 + \beta \|Y\|_F^2$$

- 例 7.7 设 $M \in \mathbb{R}^{m \times n}$ 是已知的矩阵, 非负矩阵分解模型

$$\min_{X,Y \geq 0} \quad \frac{1}{2} \|XY - M\|_F^2 + \alpha r_1(X) + \beta r_2(Y)$$

变量更新方式

- 按照 x_1, x_2, \dots, x_s 的次序依次固定其他 $(s-1)$ 块变量极小化 F

- 辅助函数

$$f_i^k(\mathbf{x}_i) = f(x_1^k, \dots, x_{i-1}^k, \mathbf{x}_i, x_{i+1}^{k-1}, \dots, x_s^{k-1})$$

- 在每一步更新中，通常使用以下三种更新格式之一

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ f_i^k(x_i) + r_i(x_i) \right\} \quad (1)$$

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ f_i^k(x_i) + \frac{L_i^{k-1}}{2} \|x_i - x_i^{k-1}\|_2^2 + r_i(x_i) \right\} \quad (2)$$

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ \langle \hat{g}_i^k, x_i - \hat{x}_i^{k-1} \rangle + \frac{L_i^{k-1}}{2} \|x_i - \hat{x}_i^{k-1}\|_2^2 + r_i(x_i) \right\} \quad (3)$$

算法 7.9 分块坐标下降法

```
1 选择两组初始点  $(x_1^{-1}, x_2^{-1}, \dots, x_s^{-1}) = (x_1^0, x_2^0, \dots, x_s^0)$   
2 for  $k = 1, 2, \dots$  do  
3   for  $i = 1, 2, \dots$  do  
4     使用格式 (1)、(2)、(3) 更新  $x_i^k$   
5   end for  
6   if 满足停机条件 then  
7     返回  $(x_1^k, x_2^k, \dots, x_s^k)$ , 算法终止  
8   end if  
9 end for
```

算法格式

- BCD 算法的子问题可采用三种不同的更新格式，这三种格式可能会产生不同的迭代序列，可能会收敛到不同的解，数值表现也不相同
- 格式 (1) 是最直接的更新方式，保证整个迭代过程的目标函数值是下降的。然而由于 f 的形式复杂，子问题求解难度较大
- 在收敛性方面，格式 (1) 在强凸问题上可保证目标函数收敛到极小值，但在非凸问题上不一定收敛

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ f_i^k(x_i) + r_i(x_i) \right\}$$

算法格式

- 格式 (2) (3) 是对格式 (1) 的修正, 不保证迭代过程目标函数的单调性, 但可以改善收敛性结果.
- 格式 (3) 实质上为目标函数的一阶泰勒展开近似, 在一些测试问题上有更好的表现, 可能的原因是使用一阶近似可以避开一些局部极小值点
- 格式 (3) 的计算量很小, 比较容易实现

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ f_i^k(x_i) + r_i(x_i) \right\}$$

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ f_i^k(x_i) + \frac{L_i^{k-1}}{2} \|x_i - x_i^{k-1}\|_2^2 + r_i(x_i) \right\}$$

$$x_i^k = \arg \min_{x_i \in \mathcal{X}_i^k} \left\{ \langle \hat{g}_i^k, x_i - \hat{x}_i^{k-1} \rangle + \frac{L_i^{k-1}}{2} \|x_i - \hat{x}_i^{k-1}\|_2^2 + r_i(x_i) \right\}$$

例 7.8

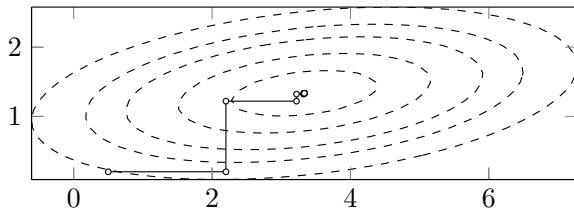
- 考虑二元二次函数的优化问题

$$\min f(x, y) = x^2 - 2xy + 10y^2 - 4x - 20y$$

- 采用格式 (1) 的分块坐标下降法

$$x^{k+1} = 2 + y^k \quad y^{k+1} = 1 + \frac{x^{k+1}}{10}$$

- 初始点为 $(x, y) = (0.5, 0.2)$ 时的迭代点轨迹



不收敛反例

■ 考虑

$$F(x_1, x_2, x_3) = -x_1x_2 - x_2x_3 - x_3x_1 + \sum_{i=1}^3 [(x_i - 1)_+^2 + (-x_i - 1)_+^2]$$

■ 设 $\varepsilon > 0$, 初始点取为

$$x^0 = \left(-1 - \varepsilon, 1 + \frac{\varepsilon}{2}, -1 - \frac{\varepsilon}{4}\right)$$

容易验证迭代序列满足

$$x^k = (-1)^k \cdot (-1, 1, -1) + \left(-\frac{1}{8}\right)^k \cdot \left(-\varepsilon, \frac{\varepsilon}{2}, -\frac{\varepsilon}{4}\right)$$

■ 迭代序列有两个聚点 $(-1, 1, -1)$ 与 $(1, -1, 1)$, 但都不是 F 的稳定点

应用举例: LASSO 问题求解

- 使用分块坐标下降法来求解 LASSO 问题

$$\min_x \quad \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|^2$$

- 将自变量 x 记为 $x = [x_i \quad \bar{x}_i^\top]^\top$, 矩阵 A 记为 $A = [a_i \quad \bar{A}_i]$

- 应用格式 (1), 替换 $c_i = b - \bar{A}_i \bar{x}_i$, 原问题等价于

$$\min_{x_i} \quad f_i(x_i) = \mu |x_i| + \frac{1}{2} \|a_i\|^2 x_i^2 - a_i^\top c_i x_i$$

- 可直接写出最小值点

$$x_i^k = \arg \min_{x_i} f_i(x_i) = \begin{cases} \frac{a_i^\top c_i - \mu_i}{\|a_i\|^2}, & a_i^\top c_i > \mu \\ \frac{a_i^\top c_i + \mu_i}{\|a_i\|^2}, & a_i^\top c_i < -\mu \\ 0, & \text{其他} \end{cases}$$

应用举例：非负矩阵分解

- 考虑最基本的非负矩阵分解问题

$$\min_{X, Y \geq 0} f(X, Y) = \frac{1}{2} \|XY - M\|_F^2$$

- 计算梯度

$$\frac{\partial f}{\partial X} = (XY - M)Y^\top, \quad \frac{\partial f}{\partial Y} = X^\top(XY - M)$$

- 应用格式 (3), 当 $r_i(X)$ 为凸集示性函数时, 得到

$$\begin{aligned} X^{k+1} &= \max\{X^k - t_k^x(X^k Y^k - M)(Y^k)^\top, 0\} \\ Y^{k+1} &= \max\{Y^k - t_k^y(X^k)^\top(X^k Y^k - M), 0\} \end{aligned}$$

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

对偶问题

- 设 f, h 是闭凸函数, 考虑复合优化问题

$$(P) \quad \min_{x \in \mathbb{R}^n} \quad \psi(x) = f(x) + h(Ax)$$

- 引入新变量 $y = Ax$, 考虑问题

$$\min_{x \in \mathbb{R}^n} \quad \psi(x) = f(x) + h(y) \quad \text{s.t.} \quad Ax = y$$

- 拉格朗日函数为

$$L(x, y, z) = f(x) + g(y) + z^\top (Ax - y)$$

- 对偶问题

$$(D) \quad \max_z \quad \phi(z) = -f^*(-A^\top z) - h^*(z)$$

强凸函数共轭函数的性质

- **引理 7.1** 设 $f(x)$ 是适当且闭的强凸函数, 强凸参数为 $\mu > 0$, 则 $f^*(y)$ 在全空间 \mathbb{R}^n 上有定义, $f^*(y)$ 是梯度 $\frac{1}{\mu}$ -利普希茨连续的可微函数
- 考虑在对偶问题上应用近似点梯度算法, 每次迭代更新如下

$$z^{k+1} = \text{prox}_{th^*}(z^k + tA\nabla f^*(-A^\top z^k))$$

- 引入变量 $x^{k+1} = \nabla f^*(-A^\top z^k)$, 利用共轭函数性质知 $-A^\top z^k \in \partial f(x^{k+1})$, 则迭代格式等价于

$$x^{k+1} = \arg \min_x \{f(x) + (A^\top z^k)^\top x\}$$

$$z^{k+1} = \text{prox}_{th^*}(z^k + tAx^{k+1})$$

- **引理 7.2** 设 f 是定义在 \mathbb{R}^n 上的适当的闭凸函数, 则对任意的 $x \in \mathbb{R}^n$ 有

$$x = \text{prox}_f(x) + \text{prox}_{f^*}(x)$$

$$\Downarrow$$

$$x = \text{prox}_{\lambda f}(x) + \lambda \text{prox}_{\lambda^{-1}f^*}\left(\frac{x}{\lambda}\right)$$

- 对任意的闭凸函数 f , 空间 \mathbb{R}^n 上的恒等映射总可以分解成两个函数 f 与 f^* 邻近算子的和

交替极小的解释

- 取 $\lambda = t$, $f = h^*$, 并注意到 $h^{**} = h$, 有

$$\begin{aligned} z^k + tAx^{k+1} &= \text{prox}_{th^*}(z^k + tAx^{k+1}) + t\text{prox}_{t^{-1}h}\left(\frac{z^k}{t} + Ax^{k+1}\right) \\ &= z^{k+1} + t\text{prox}_{t^{-1}h}\left(\frac{z^k}{t} + Ax^{k+1}\right) \end{aligned}$$

- 对偶近似点梯度法等价的针对原始问题的更新格式

$$\begin{aligned} x^{k+1} &= \arg \min_x \left\{ f(x) + (z^k)^\top Ax \right\} \\ y^{k+1} &= \text{prox}_{t^{-1}h}\left(\frac{z^k}{t} + Ax^{k+1}\right) \\ &= \arg \min_y \left\{ h(y) - (z^k)^\top (y - Ax^{k+1}) + \frac{t}{2} \|Ax^{k+1} - y\|_2^2 \right\} \\ z^{k+1} &= z^k + t(Ax^{k+1} - y^{k+1}) \end{aligned}$$

交替极小方法

■ 考虑等价问题

$$\min_{x,y} f(x) + h(y) \quad \text{s.t.} \quad y = Ax$$

■ 定义拉格朗日函数和增广拉格朗日函数

$$L(x, y, z) = f(x) + h(y) - z^\top (y - Ax)$$

$$L_t(x, y, z) = f(x) + h(y) - z^\top (y - Ax) + \frac{t}{2} \|y - Ax\|^2$$

■ 等价的交替极小格式是

$$x^{k+1} = \arg \min_x L(x, y^k, z^k)$$

$$y^{k+1} = \arg \min_y L_t(x^{k+1}, y, z^k)$$

$$z^{k+1} = z^k + t(Ax^{k+1} - y^{k+1})$$

■ 对偶近似点梯度法等价于对原始约束问题使用交替极小化方法

例 7.9

- 假设 f 是强凸函数, $\|\cdot\|$ 是任意一种范数, 考虑

$$\min_x f(x) + \|Ax - b\|$$

- 引入约束 $y = Ax$, 对应原始问题有 $h(y) = \|y - b\|$, 共轭函数为

$$h^*(z) = \begin{cases} b^\top z & \|z\|_* \leq 1 \\ +\infty & \text{其他} \end{cases} \quad \text{prox}_{th^*}(x) = \mathcal{P}_{\|z\|_* \leq 1}(x - tb)$$

- 从而对偶问题为

$$\max_{\|z\|_* \leq 1} -f^*(-A^\top z) - b^\top z$$

应用对偶近似点梯度法, 更新如下

$$x^{k+1} = \arg \min_x \{f(x) + (A^\top z^k)^\top x\}$$

$$z^{k+1} = \mathcal{P}_{\|z\|_* \leq 1}(z^k + t(Ax^{k+1} - b))$$

例 7.9

■ 考虑等价问题

$$\min_{x,y} f(x) + \|y\| \quad \text{s.t.} \quad Ax - b = y$$

■ 交替极小化格式

$$x^{k+1} = \arg \min_x f(x) + \|y^k\| + (z^k)^\top (Ax - b - y^k)$$

$$y^{k+1} = \arg \min_y f(x^{k+1}) + \|y\| + (z^k)^\top (Ax^{k+1} - b - y) + \frac{t}{2} \|Ax^{k+1} - b - y\|_2^2$$

$$z^{k+1} = z^k + t(Ax^{k+1} - b - y^{k+1})$$

例 7.10

- 假设 f 是强凸函数，考虑

$$\min_x f(x) + \sum_{i=1}^p \|B_i x\|_2$$

- 根据 $\|\cdot\|_2$ 的共轭函数定义，对偶问题

$$\max_{\|z_i\|_2 \leq 1} -f^* \left(-\sum_{i=1}^p B_i^\top z_i \right)$$

- 记 C_i 是 \mathbb{R}^{m_i} 中的单位欧几里得球，对偶近似点梯度法更新如下

$$\begin{aligned} x^{k+1} &= \arg \min_x \left\{ f(x) + \left(\sum_{i=1}^p B_i^\top z_i \right)^\top x \right\} \\ z_i^{k+1} &= \mathcal{P}_{C_i}(z_i^k + t B_i x^{k+1}), i = 1, 2, \dots, p \end{aligned}$$

例 7.11

- 假设 f 是强凸函数, 集合 C_i 为闭凸集, 且易于计算投影, 考虑

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & x \in C_1 \cap C_2 \cap \cdots \cap C_m\end{array}$$

- 令 $h(y_1, y_2, \cdots, y_m) = \sum_{i=1}^m I_{C_i}(y_i)$, $A = [I \ I \ \cdots \ I]^\top$

- 对偶问题

$$\max_{z_i \in C_i} -f^* \left(-\sum_{i=1}^m z_i \right) - \sum_{i=1}^m I_{C_i}^*(z_i),$$

$I_{C_i}^*(z_i)$ 是集合 C_i 的支撑函数, 其显式表达式不易求出

- 利用 Moreau 分解将迭代格式写成交替极小化方法的形式

$$x^{k+1} = \arg \min_x \left\{ f(x) + \left(\sum_{i=1}^m z_i \right)^\top x \right\}$$

$$y_i^{k+1} = \mathcal{P}_{C_i} \left(\frac{z_i^k}{t} + x^{k+1} \right), \quad i = 1, 2, \dots, m$$

$$z_i^{k+1} = z_i^k + t(x^{k+1} - y_i^{k+1}), \quad i = 1, 2, \dots, m$$

例 7.12

- 假设 f_i 是强凸函数, h_i^* 有易于计算的邻近算子. 考虑

$$\min \sum_{j=1}^n f_j(x_j) + \sum_{i=1}^m h_i(A_{i1}x_1 + A_{i2}x_2 + \cdots + A_{in}x_N)$$

- 对偶问题

$$\max - \sum_{i=1}^m h_i^*(z_i) - \sum_{j=1}^n f_j^*(-A_{1j}^\top z_1 - A_{2j}^\top z_2 - \cdots - A_{mj}^\top z_m)$$

- 对偶近似点梯度法更新如下

$$x_j^{k+1} = \arg \min_{x_j} \left\{ f_j(x_j) + \left(\sum_{i=1}^m A_{ij} z_i^k \right)^\top x_j \right\}, \quad j = 1, 2, \cdots, n$$

$$z_i^{k+1} = \text{prox}_{th_i^*} \left(z_i + t \sum_{j=1}^n A_{ij} x_j^{k+1} \right), \quad i = 1, 2, \cdots, m$$

- 令 f, h 是适当的闭凸函数. 考虑原始问题

$$\min_x f(x) + h(Ax)$$

- 由于 h 有自共轭性, 将问题变形为

$$(\text{LPD}) \quad \min_x \max_z \psi_{PD}(x, z) = f(x) - h^*(z) + z^\top Ax$$

- 另一种常用的鞍点问题定义方式构造拉格朗日函数

$$\min_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} f(x) + h(y) \quad \text{s.t.} \quad y = Ax$$

- 相应的鞍点问题形式如下

$$(\text{LP}) \quad \min_{x, y} \max_z f(x) + h(y) + z^\top (Ax - y)$$

- PDHG 算法的思想就是分别对两类变量应用近似点梯度算法
- 交替更新原始变量以及对偶变量，迭代格式如下

$$\begin{aligned} z^{k+1} &= \arg \max_z \left\{ -h^*(z) + \langle Ax^k, z - z^k \rangle - \frac{1}{2\delta_k} \|z - z^k\|_2^2 \right\} \\ &= \text{prox}_{\delta_k h^*}(z^k + \delta_k Ax^k) \\ x^{k+1} &= \arg \min_x \left\{ f(x) + (z^{k+1})^\top A(x - x^k) + \frac{1}{2\alpha_k} \|x - x^k\|_2^2 \right\} \\ &= \text{prox}_{\alpha_k f}(x^k - \alpha_k A^\top z^{k+1}) \end{aligned}$$

- 原始变量和对偶变量的更新顺序是无关紧要的

Chambolle-Pock 算法

- PDHG 算法的收敛性需要比较强的条件，有些情形下未必收敛
- Chambolle-Pock 算法与 PDHG 算法的区别在于多了一个外推步
- 具体的迭代格式如下

$$\begin{aligned}z^{k+1} &= \text{prox}_{\delta_k h^*}(z^k + \delta_k A y^k) \\x^{k+1} &= \text{prox}_{\alpha_k f}(x^k - \alpha_k A^\top z^{k+1}) \\y^{k+1} &= 2x^{k+1} - x^k\end{aligned}$$

- 当取常数步长 $\alpha_k = t, \delta_k = s$ 时，收敛性在 $\sqrt{st} < \frac{1}{\|A\|_2}$ 的条件下成立

应用举例: LASSO 问题求解

■ 考虑 LASSO 问题

$$\min_{x \in \mathbb{R}^n} \psi(x) = \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2$$

■ 取 $f(x) = \mu \|x\|_1$ 和 $h(x) = \frac{1}{2} \|x - b\|_2^2$, 相应的鞍点问题

$$\min_{x \in \mathbb{R}^n} \max_{z \in \mathbb{R}^m} f(x) - h^*(z) + z^\top Ax$$

■ 根据共轭函数的定义

$$h^*(z) = \sup_{y \in \mathbb{R}^m} \left\{ y^\top z - \frac{1}{2} \|y - b\|_2^2 \right\} = \frac{1}{2} \|z\|_2^2 + b^\top z$$

■ 应用 PDHG 算法, x^{k+1} 和 z^{k+1} 的更新格式分别为

$$z^{k+1} = \text{prox}_{\delta_k h^*}(z^k + \delta_k Ax^k) = \frac{1}{\delta_k + 1} (z^k + \delta_k Ax^k - \delta_k b)$$

$$x^{k+1} = \text{prox}_{\alpha_k \mu \|\cdot\|_1}(x^k - \alpha_k A^\top z^{k+1})$$

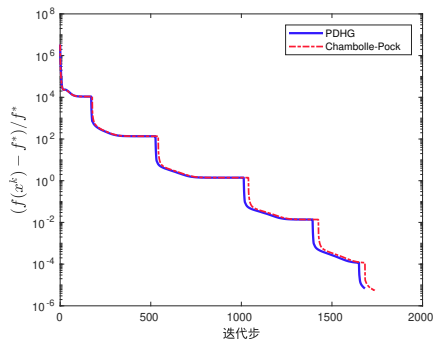
LASSO 问题求解

■ Chambolle-Pock 算法格式为

$$z^{k+1} = \frac{1}{\delta_k + 1} (z^k + \delta_k A y^k - \delta_k b)$$

$$x^{k+1} = \text{prox}_{\alpha_k \mu \|\cdot\|_1} (x^k - \alpha_k A^\top z^{k+1})$$

$$y^{k+1} = 2x^{k+1} - x^k$$



- 考虑去噪情形下的 TV- L^1 模型

$$\min_{U \in \mathbb{R}^{n \times n}} \|U\|_{TV} + \lambda \|U - B\|_1$$

- 对任意的 $W, V \in \mathbb{R}^{n \times n \times 2}$, 记

$$\|W\| = \sum_{1 \leq i, j \leq n} \|w_{ij}\|_2, \quad \langle W, V \rangle = \sum_{1 \leq i, j \leq n, 1 \leq k \leq 2} w_{i,j,k} v_{i,j,k}$$

- 利用 $\|\cdot\|$ 的定义, 有

$$\|U\|_{TV} = \|DU\|$$

- 取 D 为相应的线性算子, 并取

$$f(U) = \lambda \|U - B\|_1, \quad U \in \mathbb{R}^{n \times n}, \quad h(W) = \|W\|, \quad W \in \mathbb{R}^{n \times n \times 2}$$

- 相应的鞍点问题如下

$$(\text{LPD}) \quad \min_{U \in \mathbb{R}^{n \times n}} \max_{V \in \mathbb{R}^{n \times n \times 2}} f(U) - h^*(V) + \langle V, DU \rangle$$

- 根据共轭函数的定义

$$h^*(V) = \sup_{U \in \mathbb{R}^{n \times n \times 2}} \{\langle U, V \rangle - \|U\|\} = \begin{cases} 0, & \max_{i,j} \|v_{ij}\|_2 \leq 1 \\ +\infty, & \text{其他} \end{cases}$$

- 记 $\mathcal{V} = \{V \in \mathbb{R}^{n \times n \times 2} \mid \max_{ij} \|v_{ij}\|_2 \leq 1\}$, 其示性函数记为 $I_{\mathcal{V}}(V)$, 则问题 (LPD) 可以整理为

$$\min_U \max_V f(U) + \langle V, DU \rangle - I_{\mathcal{V}}(V)$$

- 应用 PDHG 算法, 则 V^{k+1} 的更新为

$$V^{k+1} = \text{prox}_{sI_V}(V^k + sDU^k) = \mathcal{P}_V(V^k + sDU^k)$$

- U^{k+1} 的更新如下

$$\begin{aligned} U^{k+1} &= \text{prox}_{tf}(U^k + tGV^{k+1}) \\ &= \arg \min_U \left\{ \lambda \|U - B\|_1 + \langle V^{k+1}, DU \rangle + \frac{1}{2t} \|U - U^k\|_F^2 \right\} \end{aligned}$$

其中 $G: \mathbb{R}^{n \times n \times 2} \rightarrow \mathbb{R}^{n \times n}$ 为离散的散度算子, 其满足

$$\langle V, DU \rangle = -\langle GV, U \rangle, \quad \forall U \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{n \times n \times 2}$$

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

典型问题形式

- 考虑如下凸问题

$$\begin{aligned} \min_{x_1, x_2} \quad & f_1(x_1) + f_2(x_2) \\ \text{s.t.} \quad & A_1 x_1 + A_2 x_2 = b \end{aligned} \tag{4}$$

- f_1, f_2 是适当的闭凸函数，但不要求是光滑的
- 目标函数可以分成彼此分离的两块，但是变量被线性约束结合在一起

■ 例 7.13 可以分成两块人无约束优化问题

$$\min_x f_1(x) + f_2(x)$$

引入一个新的变量 z 并令 $x = z$, 将问题转化为

$$\begin{aligned} \min_{x,z} \quad & f_1(x) + f_2(z) \\ \text{s.t.} \quad & x - z = 0 \end{aligned}$$

■ 例 7.14 带线性变换的无约束优化问题

$$\min_x f_1(x) + f_2(Ax)$$

引入一个新的变量 z , 令 $z = Ax$, 则问题变为

$$\begin{aligned} \min_{x,z} \quad & f_1(x) + f_2(z) \\ \text{s.t.} \quad & Ax - z = 0 \end{aligned}$$

- 例 7.15 凸集 $C \subset \mathbb{R}^n$ 上的约束优化问题

$$\begin{array}{ll}\min_x & f(x) \\ \text{s.t.} & Ax \in C\end{array}$$

引入约束 $z = Ax$, 那么问题转化为

$$\begin{array}{ll}\min_{x,z} & f(x) + I_C(z) \\ \text{s.t.} & Ax - z = 0\end{array}$$

■ 例 7.16 全局一致性问题

$$\min_x \sum_{i=1}^N \phi_i(x)$$

令 $x = z$, 并将 x 复制 N 份, 分别为 x_i , 那么问题转化为

$$\begin{aligned} \min_{x_i, z} \quad & \sum_{i=1}^N \phi_i(x_i) \\ \text{s.t.} \quad & x_i - z = 0, \quad i = 1, 2, \dots, N \end{aligned}$$

增广拉格朗日函数法

- 首先写出问题 (4) 的增广拉格朗日函数

$$L_{\rho}(x_1, x_2, y) = f_1(x_1) + f_2(x_2) + y^{\top}(A_1x_1 + A_2x_2 - b) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - b\|_2^2$$

- 增广拉格朗日函数法为如下更新

$$(x_1^{k+1}, x_2^{k+1}) = \arg \min_{x_1, x_2} L_{\rho}(x_1, x_2, y^k) \\ y^{k+1} = y^k + \tau \rho (A_1x_1^{k+1} + A_2x_2^{k+1} - b)$$

交替方向乘子法

- Alternating direction method of multipliers, ADMM
- 迭代格式如下

$$\begin{aligned}(x_1^{k+1}, x_2^{k+1}) &= \arg \min_{x_1, x_2} L_\rho(x_1, x_2, y^k) \\ y^{k+1} &= y^k + \tau \rho(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b) \\ &\Downarrow\end{aligned}$$

$$\begin{aligned}x_1^{k+1} &= \arg \min_{x_1} L_\rho(x_1, x_2^k, y^k) \\ x_2^{k+1} &= \arg \min_{x_2} L_\rho(x_1^{k+1}, x_2, y^k) \\ y^{k+1} &= y^k + \tau \rho(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b)\end{aligned}$$

原问题最优性条件

- 问题 (4) 的拉格朗日函数为

$$L(x_1, x_2, y) = f_1(x_1) + f_2(x_2) + y^\top (A_1 x_1 + A_2 x_2 - b)$$

- 若 x_1^*, x_2^* 为问题 (4) 的最优解, y^* 为对应的拉格朗日乘子, 则满足

$$0 \in \partial_{x_1} L(x_1^*, x_2^*, y^*) = \partial f_1(x_1^*) + A_1^\top y^* \quad (5a)$$

$$0 \in \partial_{x_2} L(x_1^*, x_2^*, y^*) = \partial f_2(x_2^*) + A_2^\top y^* \quad (5b)$$

$$A_1 x_1^* + A_2 x_2^* = b \quad (5c)$$

- 条件 (5c) 称为原始可行性条件
- 条件 (5a) 和条件 (5b) 称为对偶可行性条件

- 关于 x_2 的更新步骤

$$x_2^k = \arg \min_x \left\{ f_2(x) + \frac{\rho}{2} \|A_1 x_1^k + A_2 x - b + \frac{y^{k-1}}{\rho}\|^2 \right\}$$

- 根据最优性条件推出

$$0 \in \partial f_2(x_2^k) + A_2^\top [y^{k-1} + \rho(A_1 x_1^k + A_2 x_2^k - b)]$$

- 当 $\tau = 1$ 时知

$$0 \in \partial f_2(x_2^k) + A_2^\top y^k$$

■ 关于 x_1 的更新公式

$$x_1^k = \arg \min_x \left\{ f_1(x) + \frac{\rho}{2} \|A_1 x + A_2 x_2^{k-1} - b + \frac{y^{k-1}}{\rho}\|^2 \right\}$$

■ 假设子问题能精确求解，根据最优性条件

$$0 \in \partial f_1(x_1^k) + A_1^\top [\rho(A_1 x_1^k + A_2 x_2^{k-1} - b) + y^{k-1}]$$

■ 当 $\tau = 1$ 时知

$$0 \in \partial f_1(x_1^k) + A_1^\top (y^k + \rho A_2 (x_2^{k-1} - x_2^k))$$

■ 对比条件 (5a)

$$0 \in \partial f_1(x_1^*) + A_1^\top y^*$$

$$\Downarrow$$

$$0 \in \partial f_1(x_1^k) + A_1^\top (y^k + \rho A_2(x_2^{k-1} - x_2^k))$$

■ 当 x_2 更新取到精确解且 $\tau = 1$ 时, 判断 ADMM 是否收敛只需要检测

□ 原始可行性

$$0 \approx \|r^k\| = \|A_1 x_1^k + A_2 x_2^k - b\|$$

□ 对偶可行性

$$0 \approx \|s^k\| = \|A_1^\top A_2(x_2^{k-1} - x_2^k)\|$$

- 考虑第一个子问题

$$\min_{x_1} f_1(x_1) + \frac{\rho}{2} \|A_1 x_1 - v^k\|^2$$

- 当子问题目标函数可微时，线性化为

$$x_1^{k+1} = \arg \min_{x_1} \left\{ (\nabla f_1(x_1^k) + \rho A_1^\top (A_1 x_1^k - v^k))^\top x_1 + \frac{1}{2\eta_k} \|x_1 - x^k\|_2^2 \right\}$$

这等价于做一步**梯度下降**

- 当目标函数不可微时，可以考虑只将二次项线性化

$$x_1^{k+1} = \arg \min_{x_1} \left\{ f_1(x_1) + \rho (A_1^\top (A_1 x_1^k - v^k))^\top x_1 + \frac{1}{2\eta_k} \|x_1 - x^k\|_2^2 \right\}$$

这等价于做一步**近似点梯度步**

缓存分解

- 考虑目标函数中含二次函数

$$f_1(x_1) = \frac{1}{2} \|Cx_1 - d\|_2^2$$

\Downarrow

$$(C^\top C + \rho A_1^\top A_1)x_1 = C^\top d + \rho A_1^\top v^k$$

- 首先对 $C^\top C + \rho A_1^\top A_1$ 进行 Cholesky 分解并缓存分解的结果，在每步迭代中只需求解简单的三角形方程组
- 当 $C^\top C + \rho A_1^\top A_1$ 一部分容易求逆，另一部分是低秩的情形时，可以用 **SMW 公式** 来求逆

优化转移

- 为了方便求解子问题，可以用一个性质好的矩阵 D 近似二次项 $A_1^\top A_1$ ，即

$$x_1^{k+1} = \arg \min_{x_1} \left\{ f_1(x_1) + \frac{\rho}{2} \|A_1 x_1 - v^k\|_2^2 \right\}$$

\Downarrow

$$x_1^{k+1} = \arg \min_{x_1} \left\{ f_1(x_1) + \frac{\rho}{2} \|A_1 x_1 - v^k\|_2^2 + \frac{\rho}{2} (x_1 - x^k)^\top (D - A_1^\top A_1) (x_1 - x^k) \right\}$$

- 通过选取合适的 D ，优化转移简化子问题更容易计算
- 当 $D = \frac{\eta_k}{\rho} I$ 时，优化转移等价于做单步的近似点梯度步

二次罚项系数的动态调节

- 二次罚项系数 ρ 太大会导致原始可行性 $\|r^k\|$ 下降很快, 但是对偶可行性 $\|s^k\|$ 下降很慢. 二次罚项系数 ρ 太小, 则会有相反的效果
- 动态调节惩罚系数 ρ 的大小, 使得原始可行性和对偶可行性能够以比较一致的速度下降到零

$$\rho^{k+1} = \begin{cases} \gamma_p \rho^k, & \|r^k\| > \mu \|s^k\| \\ \rho^k / \gamma_d & \|s^k\| > \mu \|r^k\| \\ \rho^k, & \text{其他} \end{cases}$$

- 常见的选择为 $\mu = 10, \gamma_p = \gamma_d = 2$

多块问题的 ADMM

■ 考虑有多块变量的情形

$$\begin{aligned} \min_{x_1, x_2, \dots, x_N} \quad & f_1(x_1) + f_2(x_2) + \dots + f_N(x_N) \\ \text{s.t.} \quad & A_1 x_1 + A_2 x_2 + \dots + A_N x_N = b \end{aligned}$$

■ 多块 ADMM 迭代格式为

$$\begin{aligned} x_1^{k+1} &= \arg \min_x L_\rho(x, x_2^k, \dots, x_N^k, y^k) \\ x_2^{k+1} &= \arg \min_x L_\rho(x_1^{k+1}, x, \dots, x_N^k, y^k) \\ &\dots\dots\dots \\ x_N^{k+1} &= \arg \min_x L_\rho(x_1^{k+1}, x_2^{k+1}, \dots, x, y^k) \\ y^{k+1} &= y^k + \tau \rho(A_1 x_1^{k+1} + A_2 x_2^{k+1} + \dots + A_N x_N^{k+1} - b) \end{aligned}$$

其中 $\tau \in (0, (\sqrt{5} + 1)/2)$ 为步长参数

应用举例: LASSO 问题

- 考虑 LASSO 问题

$$\min \quad \mu \|x\|_1 + \frac{1}{2} \|Ax - b\|^2$$

- 转换为标准问题形式

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

- 交替方向乘子法迭代格式为

$$\begin{aligned} x^{k+1} &= \arg \min_x \left\{ \frac{1}{2} \|Ax - b\|^2 + \frac{\rho}{2} \|x - z^k + y^k / \rho\|_2^2 \right\} \\ &= (A^\top A + \rho I)^{-1} (A^\top b + \rho z^k - y^k) \end{aligned}$$

- 交替方向乘子法迭代格式为

$$\begin{aligned} z^{k+1} &= \arg \min_z \left\{ \mu \|z\|_1 + \frac{\rho}{2} \|x^{k+1} - z + y^k / \rho\|^2 \right\} \\ &= \text{prox}_{(\mu/\rho)\|\cdot\|_1}(x^{k+1} + y^k / \rho) \\ y^{k+1} &= y^k + \tau \rho (x^{k+1} - z^{k+1}) \end{aligned}$$

- 在求解 x 迭代时, 可以使用固定的罚因子 ρ , 缓存矩阵 $A^\top A + \rho I$ 的初始分解
- 主要运算量来自更新 x 变量时求解线性方程组, 复杂度为 $O(n^3)$

应用举例: LASSO 问题

- 考虑 LASSO 问题的对偶问题

$$\begin{array}{ll}\min & b^\top y + \frac{1}{2}\|y\|^2 \\ \text{s.t.} & \|A^\top y\|_\infty \leq \mu\end{array}$$

- 引入约束 $A^\top y + z = 0$, 可以得到如下等价问题

$$\begin{array}{ll}\min & \underbrace{b^\top y + \frac{1}{2}\|y\|^2}_{f(y)} + \underbrace{I_{\|z\|_\infty \leq \mu}(z)}_{h(z)} \\ \text{s.t.} & A^\top y + z = 0\end{array}$$

- 对约束 $A^\top y + z = 0$ 引入乘子 x , 对偶问题的增广拉格朗日函数

$$L_\rho(y, z, x) = b^\top y + \frac{1}{2}\|y\|^2 + I_{\|z\|_\infty \leq \mu}(z) - x^\top (A^\top y + z) + \frac{\rho}{2}\|A^\top y + z\|^2$$

应用举例: LASSO 问题

- 当固定 y, x 时, 对 z 的更新即向无穷范数球 $\{z \mid \|z\|_\infty \leq \mu\}$ 做欧几里得投影, 即将每个分量截断在区间 $[-\mu, \mu]$

- 当固定 z, x 时, 对 y 的更新即求解线性方程组

$$(I + \rho AA^\top)y = A(x^k - \rho z^{k+1}) - b$$

- ADMM 迭代格式为

$$\begin{aligned} z^{k+1} &= \mathcal{P}_{\|z\|_\infty \leq \mu}(x^k / \rho - A^\top y^k) \\ y^{k+1} &= (I + \rho AA^\top)^{-1}(A(x^k - \rho z^{k+1}) - b) \\ x^{k+1} &= x^k - \tau \rho (A^\top y^{k+1} + z^{k+1}) \end{aligned}$$

- 由于 $m \ll n$, 求解 y 更新的线性方程组需要的计算量是 $O(m^3)$

应用举例：矩阵分离问题

■ 考虑矩阵分离问题

$$\begin{aligned} \min_{X,S} \quad & \|X\|_* + \mu\|S\|_1 \\ \text{s.t.} \quad & X + S = M \end{aligned}$$

■ 引入乘子 Y 得到增广拉格朗日函数

$$L_\rho(X, S, Y) = \|X\|_* + \mu\|S\|_1 + \langle Y, X + S - M \rangle + \frac{\rho}{2}\|X + S - M\|_F^2$$

■ 对于 X 子问题

$$\begin{aligned} X^{k+1} &= \arg \min_X L_\rho(X, S^k, Y^k) \\ &= \arg \min_X \left\{ \|X\|_* + \frac{\rho}{2} \left\| X + S^k - M + \frac{Y^k}{\rho} \right\|_F^2 \right\} \\ &= \arg \min_X \left\{ \frac{1}{\rho} \|X\|_* + \frac{1}{2} \left\| X + S^k - M + \frac{Y^k}{\rho} \right\|_F^2 \right\} \\ &= U \text{Diag}(\text{prox}_{(1/\rho)\|\cdot\|_1}(\sigma(A))) V^\top \end{aligned}$$

其中 $A = M - S^k - \frac{Y^k}{\rho}$, $\sigma(A)$ 为 A 的所有非零奇异值构成的向量并且 $U \text{Diag}(\sigma(A)) V^\top$ 为 A 的约化奇异值分解

应用举例：矩阵分离问题

■ 对于 S 子问题

$$\begin{aligned} S^{k+1} &= \arg \min_S L_\rho(X^{k+1}, S, Y^k) \\ &= \arg \min_S \left\{ \mu \|S\|_1 + \frac{\rho}{2} \|X^{k+1} + S - M + \frac{Y^k}{\rho}\|_F^2 \right\} \\ &= \text{prox}_{(\mu/\rho)\|\cdot\|_1} \left(M - X^{k+1} - \frac{Y^k}{\rho} \right) \end{aligned}$$

■ 交替方向乘子法的迭代格式为

$$\begin{aligned} X^{k+1} &= U \text{Diag}(\text{prox}_{(1/\rho)\|\cdot\|_1}(\sigma(A))) V^\top \\ S^{k+1} &= \text{prox}_{(\mu/\rho)\|\cdot\|_1} \left(M - L^{k+1} - \frac{Y^k}{\rho} \right) \\ Y^{k+1} &= Y^k + \tau \rho (X^{k+1} + S^{k+1} - M) \end{aligned}$$

应用举例：全局一致性优化问题

■ 考虑全局一致性优化问题

$$\begin{aligned} \min_{x_i, z} \quad & \sum_{i=1}^N \phi_i(x_i) \\ \text{s.t.} \quad & x_i - z = 0, \quad i = 1, 2, \dots, N \end{aligned}$$

■ 增广拉格朗日函数为

$$L_\rho(x_1, \dots, x_N, z, y_1, \dots, y_N) = \sum_{i=1}^N \phi_i(x_i) + \sum_{i=1}^N y_i^\top (x_i - z) + \frac{\rho}{2} \sum_{i=1}^N \|x_i - z\|^2$$

■ 固定 z^k, y_i^k , 更新 x_i 的公式为

$$x_i^{k+1} = \arg \min_x \left\{ \phi_i(x) + \frac{\rho}{2} \|x - z^k + y_i^k / \rho\|^2 \right\}$$

应用举例：全局一致性优化问题

- 在一般情况下更新 x_i 的表达式为

$$x_i^{k+1} = \text{prox}_{\phi_i/\rho}(z^k - y_i^k/\rho)$$

- 固定 x_i^{k+1}, y_i^k , 关于 z 可以直接写出显式解

$$z^{k+1} = \frac{1}{N} \sum_{i=1}^N (x_i^{k+1} + y_i^k/\rho)$$

- 交替方向乘子法迭代格式为

$$x_i^{k+1} = \text{prox}_{\phi_i/\rho}(z^k - y_i^k/\rho), \quad i = 1, 2, \dots, N$$

$$z^{k+1} = \frac{1}{N} \sum_{i=1}^N (x_i^{k+1} + y_i^k/\rho)$$

$$y_i^{k+1} = y_i^k + \tau\rho(x_i^{k+1} - z^{k+1}), \quad i = 1, 2, \dots, N$$

- 7.1 近似点梯度法
- 7.2 Nesterov 加速算法
- 7.3 近似点算法
- 7.4 分块坐标下降法
- 7.5 对偶算法
- 7.6 交替方向乘子法
- 7.7 随机优化算法

- 假定 (a, b) 服从概率分布 P , 其中 a 为输入, b 为标签
 - 在自动邮件分类任务中, a 表示邮件内容, b 表示正常邮件或垃圾邮件
 - 在人脸识别任务中, a 表示人脸的图像信息, b 表示该人脸属于何人
- 实际问题中我们不知道真实的概率分布 P , 而是随机采样得到一个数据集

$$\mathcal{D} = \{(a_1, b_1), (a_2, b_2), \dots, (a_N, b_N)\}$$

数据集 \mathcal{D} 对应经验分布

$$\hat{P} = \frac{1}{N} \sum_{n=1}^N \delta_{a_i, b_i}$$

- 监督学习的任务是要给定输入 a 预测标签 b ，即决定一个最优的函数 ϕ 使得期望风险最小

$$\mathbb{E}[L(\phi(a), b)]$$

- 常用的 ℓ_2 损失函数

$$L(x, y) = \frac{1}{2} \|x - y\|_2^2$$

- 若 $x, y \in \mathbb{R}^d$ 为概率分布，则可定义互熵损失函数

$$L(x, y) = \sum_{i=1}^d x_i \log \frac{x_i}{y_i}$$

- 为了缩小目标函数的范围，需要将 $\phi(\cdot)$ 参数化为 $\phi(\cdot; x)$

- 用经验风险来近似期望风险，即要求解下面的极小化问题

$$\min_x \quad \frac{1}{N} \sum_{i=1}^N L(\phi(a_i; x), b_i) = \mathbb{E}_{(a,b) \sim \hat{P}} [L(\phi(a; x), b)]$$

- 记 $f_i(x) = L(\phi(a_i; x), b_i)$ ，则只需考虑如下随机优化问题

$$\min_{x \in \mathbb{R}^n} \quad f(x) = \frac{1}{N} \sum_{i=1}^N f_i(x)$$

- 由于数据规模巨大，通过采样的方式只计算部分样本的梯度来进行梯度下降

随机梯度下降算法 (SGD)

- SGD 的基本迭代格式为

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k), \quad \nabla f(x^k) = \frac{1}{N} \sum_{i=1}^N \nabla f_i(x^k)$$

\Downarrow

$$x^{k+1} = x^k - \alpha_k \nabla f_{s_k}(x^k)$$

- s_k 是从 $\{1, 2, \dots, N\}$ 中随机等可能地抽取的一个样本
- α_k 称为步长, 又称学习率
- 要保证随机梯度的条件期望恰好是全梯度, 即

$$\mathcal{E}_{s_k}[\nabla f_{s_k}(x^k) | x^k] = \nabla f(x^k)$$

随机梯度法

- **小批量 (mini-batch) 随机梯度法** 每次迭代中, 随机选择一个元素个数很少的集合 $I_k \subset \{1, 2, \dots, N\}$, 然后执行迭代格式

$$x^{k+1} = x^k - \alpha_k \nabla f_{s_k}(x^k)$$

\Downarrow

$$x^{k+1} = x^k - \frac{\alpha_k}{|\mathcal{I}_k|} \sum_{s \in \mathcal{I}_k} \nabla f_s(x^k)$$

- **随机次梯度法** 当 $f_i(x)$ 是凸函数但不一定可微时, 可以用 $f_i(x)$ 的次梯度代替梯度进行迭代

$$x^{k+1} = x^k - \alpha_k g^k$$

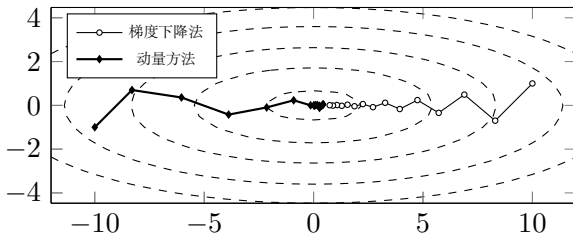
动量方法

- 动量方法的具体迭代格式如下

$$v^{k+1} = \mu_k v^k - \alpha_k \nabla f_{s_k}(x^k)$$

$$x^{k+1} = x^k + v^{k+1}$$

- 参数 μ_k 的范围是 $[0, 1)$ ，通常取 $\mu_k \geq 0.5$ ，其含义为迭代点带有较大惯性，每次迭代会在原始迭代方向的基础上做一个小的修正。当 $\mu_k = 0$ 时退化成随机梯度下降法



- 假设 $f(x)$ 为光滑的凸函数, 则 Nesterov 加速算法为

$$\begin{aligned}y^{k+1} &= x^k + \mu_k(x^k - x^{k-1}) \\x^{k+1} &= y^k - \alpha_k \nabla f(y^k)\end{aligned}$$

- 针对光滑问题的 Nesterov 加速算法迭代的随机版本为

$$\begin{aligned}y^{k+1} &= x^k + \mu_k(x^k - x^{k-1}) \\x^{k+1} &= y^{k+1} - \alpha_k \nabla f_{s_k}(y^{k+1})\end{aligned}$$

其中 $\mu_k = \frac{k-1}{k+2}$, 步长 α_k 是一个固定值或者由线搜索确定

- 二者的唯一区别为随机版本将全梯度 $\nabla f(y^k)$ 替换为随机梯度 $\nabla f_{s_k}(y^{k+1})$

Nesterov 加速算法与动量方法的联系

- 引入速度变量 $v^k = x^k - x^{k-1}$, 结合原始 Nesterov 加速算法的两步迭代得到

$$x^{k+1} = x^k + \mu_k(x^k - x^{k-1}) - \alpha_k \nabla f_k(x^k + \mu_k(x^k - x^{k-1}))$$

- 定义 $v^{k+1} = \mu_k v^k - \alpha_k \nabla f_k(x^k + \mu_k v^k)$, 于是关于 x^k 和 v^k 的等价迭代式

$$v^{k+1} = \mu_k v^k - \alpha_k \nabla f_{s_k}(x^k + \mu_k v^k)$$

$$x^{k+1} = x^k + v^{k+1}$$

- 与动量方法相比

$$v^{k+1} = \mu_k v^k - \alpha_k \nabla f_{s_k}(x^k)$$

$$x^{k+1} = x^k + v^{k+1}$$

Nesterov 加速算法先对点施加速度的作用再求梯度, 即对动量方法做了校正

- 令 $g^k = \nabla f_{s_k}(x^k)$, 引入

$$G^k = \sum_{i=1}^k g^i \odot g^i$$

- 当 G^k 的某分量较大时, 该分量变化比较剧烈, 应采用小步长, 反之亦然
- AdaGrad 迭代格式

$$\begin{aligned}x^{k+1} &= x^k - \frac{\alpha}{\sqrt{G^k + \varepsilon 1_n}} \odot g^k \\G^{k+1} &= G^k + g^{k+1} \odot g^{k+1}\end{aligned}$$

AdaGrad 的收敛阶

- 如果在 AdaGrad 中使用真实梯度 $\nabla f(x^k)$, 那么 AdaGrad 也可以看成是一种介于一阶和二阶的优化算法

- 考虑 $f(x)$ 在点 x^k 处的二阶泰勒展开

$$f(x) \approx f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top B^k (x - x^k)$$

- 选取不同的 B^k 可以导出不同的优化算法, 例如 AdaGrad 选择

$$B^k = \frac{1}{\alpha} \text{Diag}(\sqrt{G^k + \varepsilon 1_n})$$

- AdaGrad 会累加之前所有的梯度分量平方, 导致步长是单调递减的, 因此在训练后期步长会非常小, 计算的开销较大

- RMSProp (root mean square propagation) 是对 AdaGrad 的一个改进, 在非凸问题上可能表现更好
- RMSProp 只需使用离当前迭代点比较近的项, 同时引入衰减参数 ρ . 具体地, 令

$$M^{k+1} = \rho M^k + (1 - \rho) g^{k+1} \odot g^{k+1}$$

再对其每个分量分别求根, 就得到均方根 (root mean square)

$$R^k = \sqrt{M^k + \varepsilon 1_n}$$

最后将均方根的倒数作为每个分量步长的修正

■ RMSProp 迭代格式

$$\begin{aligned}x^{k+1} &= x^k - \frac{\alpha}{\sqrt{G^k + \varepsilon 1_n}} \odot g^k \\G^{k+1} &= G^k + g^{k+1} \odot g^{k+1}\end{aligned}$$

\Downarrow

$$\begin{aligned}x^{k+1} &= x^k - \frac{\alpha}{R^k} \odot g^k \\M^{k+1} &= \rho M^k + (1 - \rho) g^{k+1} \odot g^{k+1}\end{aligned}$$

■ RMSProp 和 AdaGrad 的唯一区别是将 G^k 替换成了 M^k

■ 一般取 $\rho = 0.9$, $\alpha = 0.001$

- Adam 选择了一个动量项进行更新

$$S^k = \rho_1 S^{k-1} + (1 - \rho_1) g^k$$

- 类似 RMSProp, Adam 也会记录梯度的二阶矩

$$M^k = \rho_2 M^{k-1} + (1 - \rho_2) g^k \odot g^k$$

- 与原始动量方法和 RMSProp 的区别是, 由于 S^k 和 M^k 本身带有偏差, Adam 在更新前先对其进行修正

$$\hat{S}^k = \frac{S^k}{1 - \rho_1^k}, \quad \hat{M}^k = \frac{M^k}{1 - \rho_2^k}$$

- Adam 最终使用修正后的一阶矩和二阶矩进行迭代点的更新

$$x^{k+1} = x^k - \frac{\alpha}{\sqrt{\hat{M}^k + \varepsilon 1_n}} \odot \hat{S}^k$$

Q&A

Thank you!

感谢您的聆听和反馈