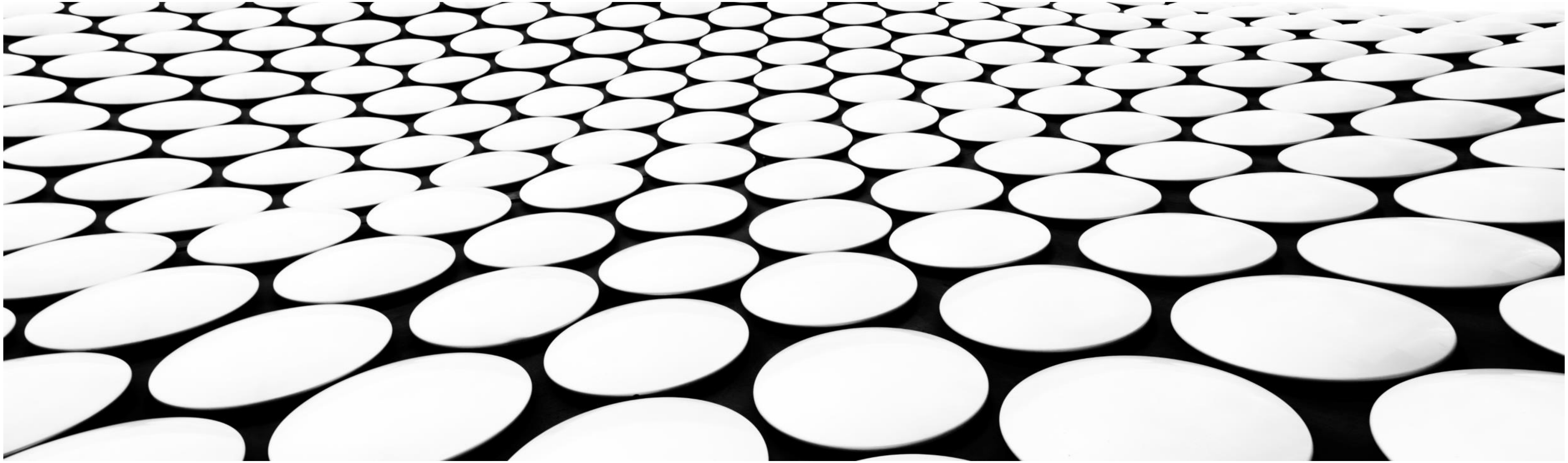
分布式计算

邱怡轩



今天的主题

- 典型机器学习模型的分布式算法
- ADMM 算法（一）

回顾

- Logistic 回归
- $Y|x \sim Bernoulli(\rho(\beta' x))$
- $\rho(x) = 1/(1 + e^{-x})$
- 给定数据 (y_i, x_i), $i = 1, \ldots, n$
- 估计 β

目标函数

- 利用极大似然准则

$$L(\beta) = -\sum_{i=1}^{n}\{y_i \log \rho_i + (1-y_i)\log(1-\rho_i)\}$$

- 其中 $\rho_i = \rho(x_i'\beta)$

- 找到一个 β 的取值，使得 $L(\beta)$ 最小

扩展

- 事实上，很多统计和机器学习模型都可以写成形如

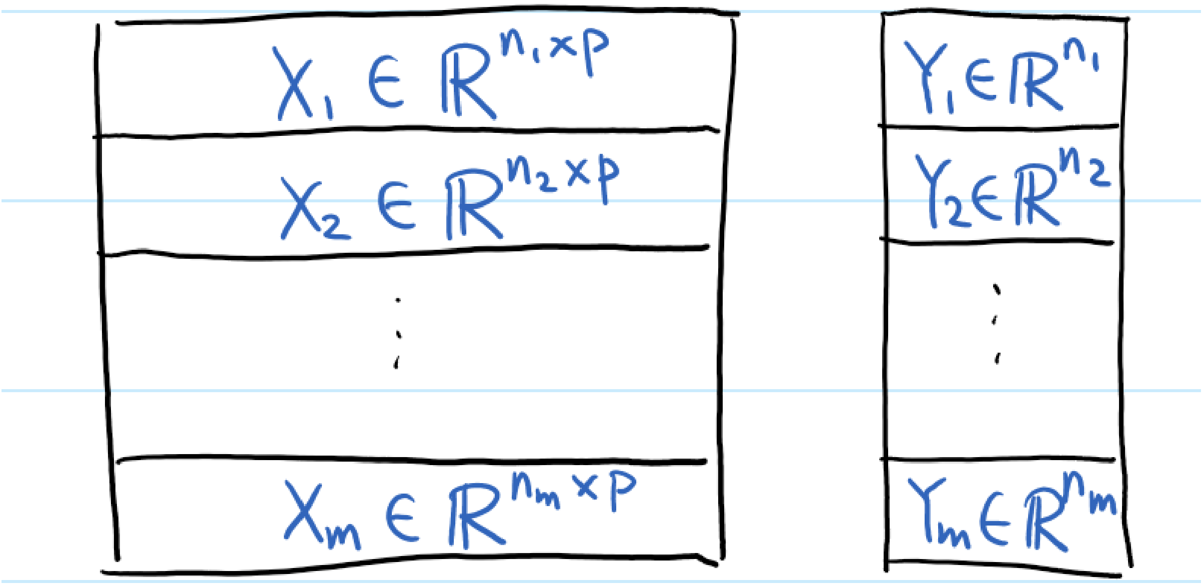
$$\min_{\beta} \ L(\beta) = \sum_{i=1}^{n} l_i(\beta)$$

的表达式

- 其中 $l_i(\beta)$ 代表第 i 个观测上的损失函数
- $l_i(\cdot)$ 依赖于数据 (y_i, x_i)
- $l_i(\cdot)$ 是光滑函数（二阶可导）

数据切分

- 数据按行切分
- 每个分块包含一部分观测
- 每个分块包含所有的变量



计算思路

- 分布式计算每个分块上的损失函数（及梯度）求和
- 再汇总所有分块的结果，计算总损失函数（及梯度）
- 利用梯度下降或 L-BFGS 等方法更新参数

$l_i(\beta) \triangleq l(\beta; x_i, y_i)$

$X_1 \in \mathbb{R}^{n_1 \times p}$	$Y_1 \in \mathbb{R}^{n_1}$
$X_2 \in \mathbb{R}^{n_2 \times p}$	$Y_2 \in \mathbb{R}^{n_2}$
$\vdots$	$\vdots$
$X_m \in \mathbb{R}^{n_m \times p}$	$Y_m \in \mathbb{R}^{n_m}$

$\to L_1(\beta) = \sum_{i \in S_1} l_i(\beta)$, $S_1 = \{1, \cdots, n_1\}$

$\to L_2(\beta) = \sum_{i \in S_2} l_i(\beta)$, $S_2 = \{n_1+1, \cdots, n_1+n_2\}$

$\to L_m(\beta) = \sum_{i \in S_m} l_i(\beta)$, $S_m = \{\cdots\}$

$\downarrow$

$$L(\beta) = \sum_{j=1}^{m} L_j(\beta)$$

例子

- Poisson 回归
- $Y_i|x_i \sim Poisson(\lambda_i)$
- $\lambda_i = e^{\beta' x_i}$
- 给定数据 (y_i, x_i)，$i = 1, \ldots, n$，估计 β
- $L(\beta) =?$，$\partial L(\beta)/\partial\beta =?$

挑战

- 然而，实际情况中有一些问题不符合如上的框架，例如：

  - 目标函数不光滑
  - 参数存在约束

例子

- Least absolute deviations：

$$\min_x \ \|Y - X\beta\|_1$$

- Lasso：

$$\min_x \frac{1}{2}\|Y - X\beta\|^2 + \lambda\|\beta\|_1$$

- SVM：

$$\begin{aligned} &\min_{w,b} \quad ||w||^2 \\ &s.t. \quad y_i(w'x_i + b) \geq 1 \end{aligned}$$

目标

- 我们希望能有一个足够通用的框架

- 支持不光滑的目标函数
- 处理参数的约束
- 实现分布式计算

ADMM

概览

- 一种解复杂优化问题的方法
- 对一类统计和机器学习模型提供通用的并行框架

ADMM

- Minimize $f(x) + g(z)$
- Subject to $Ax + Bz = c$

- x: n 维向量
- z: m 维向量
- $A\ [p \times n]$，$B\ [p \times m]$，$c\ [p \times 1]$：约束条件
- $f(\cdot)$，$g(\cdot)$：凸函数

凸函数

- $f(x), x \in \mathbf{R}^n$ 是凸函数:
- 对任意 $0 \le t \le 1$ 及 $x_1, x_2 \in \mathbf{R}^n$,

  $$f(tx_1 + (1-t)x_2) \le tf(x_1) + (1-t)f(x_2)$$

- 如果 $f(x)$ 有二阶导数，那么该二阶导非负
- 多变量时，二阶导（Hessian 矩阵）非负定

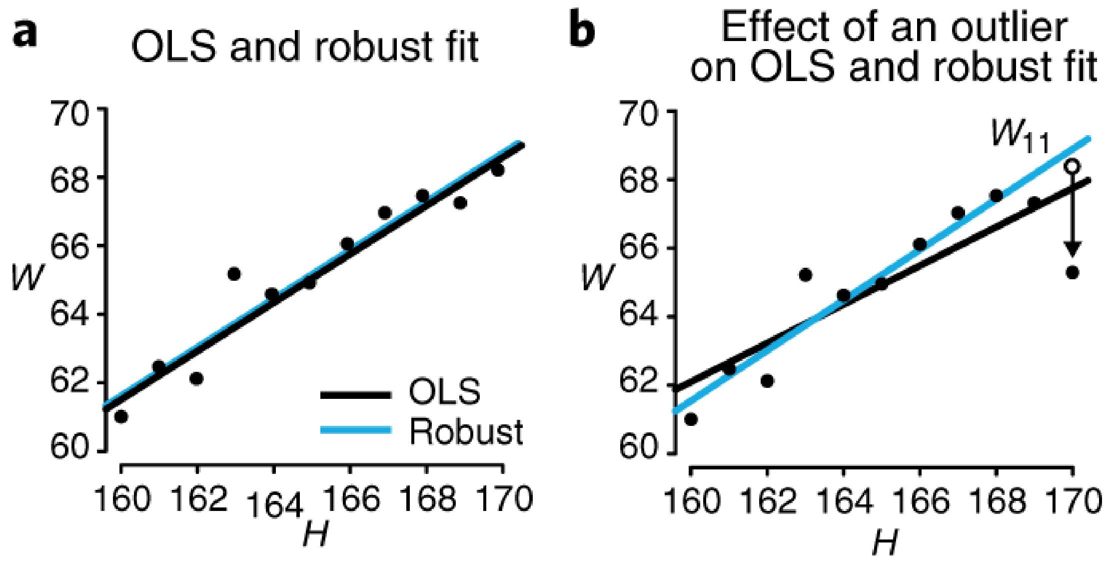
凸函数

- https://zhuanlan.zhihu.com/p/56876303
- lec11-convex.pdf

例子

LAD

- 回归： $\min_x \|Ax-b\|^2$
- Least absolute deviations： $\min_x \|Ax-b\|_1$
- 其中 $\|v\|_1 = |v_1| + \cdots + |v_p|$，若 $v = (v_1, \dots, v_p)'$
- 起到稳健回归的作用（中位数回归）



LAD

- 转换成 ADMM 形式
- 令 $f = 0$， $g = \|\cdot\|_1$

- Minimize $\|z\|_1$
- Subject to $Ax - z = b$

Lasso

- 回归： $\min_x \|Ax-b\|^2$
- Lasso： $\min_x \frac{1}{2}\|Ax-b\|^2+\lambda\|x\|_1$

- 起到变量选择的作用
- 选取适当的 λ， x 的一些元素会变成0

Lasso

- 转换成 ADMM 形式
- 令 $f(x) = \frac{1}{2}\|Ax - b\|^2$， $g(z) = \lambda\|z\|_1$

- Minimize $\frac{1}{2}\|Ax - b\|^2 + \lambda\|z\|_1$
- Subject to $x - z = 0$

ADMM 算法

增广函数

- $L_\rho(x,z,y) = f(x) + g(z) + y'(Ax + bz - c) + (\rho/2)\|Ax + Bz - c\|^2$
- y $[p \times 1]$ 为辅助变量
- ρ 为任意给定的正数

ADMM 算法

- $x^{k+1} = \operatorname{argmin}_x L_\rho\left(x, z^k, y^k\right)$
- $z^{k+1} = \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k)$
- $y^{k+1} = y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$

等价形式

- $x^{k+1} = \text{argmin}_x \ f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|^2$
- $z^{k+1} = \text{argmin}_z \ g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|^2$
- $u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c$

停止条件

- 定义两类残差
- 对偶问题残差

$$s^{k+1} = \rho A' B(z^{k+1} - z^k)$$

- 原问题残差

$$r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$$

- 当 $\|r^k\|$ 和 $\|s^k\|$ 小于某些阈值时停止算法
- 参见 lec11-admm1.pdf

例：LAD

- $x^{k+1} = (A'A)^{-1}A'(b + z^k - u^k)$
- $z^{k+1} = S_{1/\rho}\,(Ax^{k+1} - b + u^k)$
- $u^{k+1} = u^k + Ax^{k+1} - z^{k+1} - b$

- $S_\kappa(a)$ 称为 Soft-thresholding 运算符

$$S_\kappa(a) = \begin{cases} a - \kappa & a > \kappa \\ 0 & |a| \le \kappa \\ a + \kappa & a < -\kappa, \end{cases}$$

例：Lasso

- $x^{k+1} = (A'A + \rho I)^{-1}(A'b + \rho(z^k - u^k))$
- $z^{k+1} = S_{\lambda/\rho}\,(x^{k+1} + u^k)$
- $u^{k+1} = u^k + x^{k+1} - z^{k+1}$

适用范围

- ADMM 适用的问题通常有以下一些特征
  - 没有显式解
  - 目标函数不是处处可导，如带有 $\|\cdot\|_1$
  - 带有线性约束
  - 每个分步更新都有显式解，或可以较容易地计算
- 难点在于如何将某个优化问题转换成 ADMM 形式，同时让分步更新有显式解
- lec11-admm1.pdf 的第6章给出了若干例子

适用范围

- “通用”的分布式计算框架
  - 一致性优化（Consensus）
  - 共享优化（Sharing）