简析交替乘子方向法（ADMM）

李振东

钱伟长学院
指导老师: 余长君

dmax13@shu.edu.cn

2021 年 11 月 28 日

Overview

问题背景

在机器学习问题中出现了很多结构复杂且可能非凸、光滑的优化问题，我们接下来介绍的交替方向乘子法很自然的提供了一个适用范围广，容易实现，可靠性不错的解决方案。交替方向乘子法是块坐标下降法的延伸，在 1975 年和 1976 年由 Glowinski 和 Marrocco 以及 Gabay 和 Mercier 分别提出。
2011 年 S.Boyd 等人又再次对交替方向乘子法进行综述并证明了其使用于大规模的分布式优化问题；我国学者何炳生等人也在此方面做了许多不错的研究工作，比如研究了其收敛速率等困难问题。之后 ADMM 被广泛应用于求解带可分离结构的凸优化问题上，比如 2013 年 S.Boyd 曾将 ADMM 应用在最优控制问题上，也得到了很不错的结果。

迭代格式

我们考虑这样一个凸优化问题：

$$\begin{aligned} &\min f_1(x_1) + f_2(x_2), \\ &\text{s.t.} \quad A_1x_1 + A_2x_2 = b \end{aligned} \tag{1}$$

可以写出他的增广 Lagrange 函数：

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

那么 ADMM 的迭代格式为

$$\begin{aligned} x_1^{k+1} &= \underset{x_1}{argmin} L_\rho\left(x_1, x_2^k, y^k\right) \\ x_2^{k+1} &= \underset{x_2}{argmin} L_\rho\left(x_1^{k+1}, x_2, y^k\right) \\ y^{k+1} &= y^k + \tau\rho\left(A_1x_1^{k+1} + A_2x_2^{k+1} - b\right) \end{aligned} \tag{3}$$

其中 τ 为步长，一般取 $(0, \frac{\sqrt{5}+1}{2})$

终止条件

终止条件为：

原始残量 $||r^k|| = ||A_1 x_1^k + A_2 x_2^k - b|| \leq \epsilon^{pri}$

对偶残量 $||s^k|| = ||\rho \cdot A_1^T A_2 \left(y^k - y^{k-1}\right)|| \leq \epsilon^{dual}$

Lasso 及其对偶问题

考虑如下优化问题：

$$\min_{x} \frac{1}{2}||Ax - b||_2^2 + \mu \ ||x||_1 \tag{4}$$

其中 $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, \mu > 0$ 是给定的。
Lasso 问题的对偶问题为

$$\max \quad b^T\lambda - \frac{1}{2}||\lambda||_2^2$$

s.t. $\quad ||A^T\lambda||_\infty \leq \mu$

Lasso 及其对偶问题

令 $y = Ax - b$ 将问题变成带约束的优化问题

$$\begin{array}{l} \min_{x,y} \frac{1}{2}||y||_2^2 + \mu||x||_1 \\ s.t. \quad y = Ax - b \end{array}$$

我们写出他的拉格朗日函数

$$L(x, y, \lambda) = \mu||x||_1 + \frac{1}{2}||y||_2^2 - \lambda^T (Ax - b - y)$$

$=\mu||x||_1 - \left(A^T\lambda\right)^T x + \frac{1}{2}||y||_2^2 + \lambda^T y + b^T\lambda$
分别对拉格朗日函数中的 x，y 求极小值。对于 x，因为 $\mu||x||_1 - \left(A^T\lambda\right)^T x = \Sigma_i \mu|x_i| - \left(A^T\lambda\right)_i x_i$，而当 $|\left(A^T\lambda\right)_i| \le \mu$ 时，在 x=0 处 $\mu|x_i| - \left(A^T\lambda\right)_i x_i$ 取极小值 0。
对于 y 这是个二次函数，在 $y = -\lambda$ 处 $\frac{1}{2}||y||_2^2 + \lambda^T y$ 取极小值 $-\frac{1}{2}||\lambda||_2^2$。

通过以下三个问题来探究各个方法的优劣性：

1. 通过增广拉格朗日方法 (ALM) 求解对偶问题
2. 通过交替乘子方向法 (ADMM) 求解对偶问题
3. 通过使用线搜索的交替乘子方向法 (ADMM) 求解原问题

ALM 求解对偶问题

ALM 方法需要等式约束才可以解，将上述问题转化为

$$\begin{array}{ll} \min & -b^T\lambda + \frac{1}{2}||\lambda||_2^2 \\ \text{s.t.} & A^T\lambda = s, ||s||_\infty \leq \mu \end{array}$$

他的增广拉格朗日函数为

$$L(x, y, \lambda) = -b^T\lambda + \frac{1}{2}||\lambda||_2^2 + x^T\left(A^T\lambda - s\right) + \frac{1}{2\tau}||A^T\lambda - s||^2$$

假设当前迭代点为 x^k, λ^k, s^k, 则 ALM 算法框架如下

1. 固定 x^k, 求解如下 DL 子问题

$$\min_{\lambda, s} L\left(\lambda, s, x^k\right), \qquad \text{s.t.} ||s||_\infty \leq \mu$$

2. 更新拉格朗日乘子

$$x^{k+1} = x^k + \frac{A^T\lambda^{k+1} - s^{k+1}}{\tau}$$

求解 DL 子问题

接下来我们求解 DL 子问题，这是一个约束优化问题，由于目标函数是可微的并且约束集合很特殊，因此我们可以考虑使用投影梯度法，迭代公式为

$$\lambda_{j+1} = \lambda_j - t\nabla_\lambda L\left(\lambda_j, s_j, x^k\right)$$

$$s_{j+1} = P\left(s_j - t\nabla_s L\left(\lambda_{j+1}, s_j, x_k\right)\right)$$

其中

$$\begin{array}{l} \nabla_\lambda L = \lambda - b + Ax + \frac{1}{\tau} A\left(A^T\lambda - s\right) \\ \nabla_s L = -x - \frac{1}{\tau}\left(A^T\lambda - s\right) \end{array}$$

$P(x)$ 为投影算子，将 x 的每个分量投影到区间 $[-\mu, \mu]$，初始迭代点为 $\lambda_0 = \lambda^k, s_0 = s^k$ 一直迭代到增广拉格朗日函数 $L(x, y, \lambda) = -b^T\lambda + \frac{1}{2}||\lambda||_2^2 + x^T\left(A^T\lambda - s\right) + \frac{1}{2\tau}||A^T\lambda - s||^2$ 没有明显下降趋势，即 λ, s 达到收敛，用收敛点表式 λ^{k+1}, s^{k+1}。

To be continued

也可以换一种想法，将两个变量分开求解，具体迭代公式为

$$\lambda_{j+1} = \left(AA^T + \tau I\right)^{-1} \left(\tau \left(b - Ax^k\right) + As_j\right)$$

$$s_{j+1} = P_{[-\mu,\mu]} \left(A^T\lambda_{j+1} + \mu x^k\right)$$

同样一直迭代到增广拉格朗日函数没有明显下降趋势，即 λ, s 达到收敛，用收敛点表式，这样的好处是变量有显式解且简单易求。

通过以上两种方法可以求解 DL 子问题，然后更新拉格朗日乘子，一直迭代直到达到收敛准则。

ADMM 求解对偶问题

将 λ, s 看成独立的变量分别求解极值问题，这时都是有显式解的，其实也就是问题一中求解 DL 子问题时的第二种方法, 只是更新一次 λ, s 就更新一次拉格朗日乘子, 没有对 λ, s 迭代。因此 ADMM 方法的更新步骤为

$$
\begin{aligned}
\lambda^{k+1} &= \left(AA^T + \tau I\right)^{-1}\left(\tau\left(b - Ax^k\right) + As^k\right) \\
s^{k+1} &= P_{[-\mu,\mu]}\left(A^T\lambda^{k+1} + \tau x^k\right) \\
x^{k+1} &= x^k + \frac{A^T\lambda^{k+1} - s^{k+1}}{\tau}
\end{aligned}
$$

线搜索 ADMM 求解 Lasso 问题

改写 Lasso 问题为

$$\begin{aligned}&\min_{x,y}\ \tfrac{1}{2}\,||Ax-b||_2^2+\mu\ \ ||y||_1\\&\text{s.t.}\quad x-y=0\end{aligned}$$

其增广拉格朗日函数为

$$L\left(x,y,z\right)=\frac{1}{2}||Ax-b||_2^2+\mu||y||_1+z^T\left(x-y\right)+\frac{\tau}{2}||x-y||_2^2$$

- 若 y,z 已知，增广拉格朗日函数为关于 x 的二次函数，因此有极小值 $x=\left(A^TA+\tau I\right)^{-1}\left(A^Tb+\tau y-z\right)$
- 若 x,z 已知，求增广拉格朗日函数关于 y 的极小值为

$$\begin{aligned}&\min\mu||y||_1+z^T\left(x-y\right)+\tfrac{\tau}{2}||x-y||_2^2\\&\Leftrightarrow\min\tfrac{\mu}{\tau}||y||_1+\tfrac{1}{\tau}z^T\left(x-y\right)+\tfrac{1}{2}||x-y||_2^2\\&\Leftrightarrow\min\mu\tau||y||_1+\tfrac{1}{2}||x-y+\tfrac{z}{\tau}||_2^2\end{aligned}$$

线搜索 ADMM 求解 Lasso 问题

这是一个关于 1-范数的 proximal 算子，因此可以得到 y 的值为

$$y = prox_{\mu/\tau\|\cdot\|_1}\left(x + z/\tau\right)$$

综合以上我们可以得到 ADMM 方法的迭代公式

$$\begin{aligned} x^{k+1} &= \left(A^T A + \tau I\right)^{-1}\left(A^T b + \tau y^k - z^k\right) \\ y^{k+1} &= prox_{\mu/\tau\|\cdot\|_1}\left(x^{k+1} + z^k/\tau\right) \\ z^{k+1} &= z^k + \tau\left(x^{k+1} - y^{k+1}\right) \end{aligned}$$

参数设置

我们需要设置这六种方法的参数包括初始值 μ、固定迭代步长 t、固定惩罚系数 τ、算法终止的精度 eps 和对应每个 μ 值的最大迭代次数 Max。当迭代次数达到设置的最大值或者两次迭代的函数值的差小于设置的精度 eps 时，即迭代已经不能使得函数值有明显下降时，将 μ 取向下一个值，直到 $\mu = 0.001$。

参数	PGD	SGD	Prox	ALM_dual	ADMM_dual	ADMM_pri
μ	100	100	10	1	1	1
t	4.5e-4	4e-4	7e-4	-	-	-
τ	-	-	-	100	100	100
eps	1e-9	1e-10	1e-9	1e-9	1e-9	1e-10
Max	200	300	250	150	150	200

表: 六种方法的参数设置

数值结果

Solver	Fval	Time	Errfun	Iter	Sparisity	recovery
cvx_mosek	8.0877e-2	1.27	-	-	80.8764	5.2e-04
PGD	8.0876e-2	0.21	1.53e-06	1136	80.8762	5.6e-04
SGD	8.0877e-2	0.29	1.27e-06	1662	80.8764	6.5e-04
ProxGD	8.0876e-2	0.15	1.83e-06	980	80.8763	5.4e-04
ALM_dual	8.0876e-2	0.40	2.86e-06	343	80.8763	5.0e-04
ADMM_dual	8.0876e-2	0.18	2.84e-06	344	80.8763	5.0e-04
ADMM_ldual	8.0876e-2	0.35	2.75e-06	479	80.8763	4.8e-04

表: 六种方法数值结果的比较

带线性 PDE 约束的最优控制方程求解

我们接下来讨论了以下带有线性 PDE 约束的最优控制问题
其中状态方程可以表示为

$$\begin{cases} \frac{\partial y}{\partial t} - \nabla^2 y + y = v\chi\mathcal{O}, & \text{in} \quad \Omega \times (0, T) \\ y = 0, & \text{on} \quad \Gamma \times (0, T) \\ y(0) = \sin 2\pi x_1 \sin 2\pi x_2 \end{cases}$$

其中 $\Omega = \left\{(x_1, x_2) \in \mathbb{R}^2 \mid 0 < x_1 < 1, 0 < x_2 < 1\right\}, T = 1$.
目标函数 y_d 为:

$$J(v) = \frac{1}{2} \iint_Q |y - y_d|^2 \, dxdt + \frac{\gamma}{2} \iint_{\mathcal{O}} |v|^2 dxdt$$

其中 $y_d(x_1, x_2, t) = e^t \sin 2\pi x_1 \sin 2\pi x_2, \quad (x_1, x_2) \in \Omega, \quad 0 < t < T$
可行集为 $\mathcal{C} = \left\{v \mid v \in L^2(\mathcal{O}), -2 \leq y(t; v) \leq 2 \text{ a.e. in } \Omega \times (0, T)\right\}$
设置初始点 $u^0 = \lambda^0 = 0$ [1]

[1] Joined work with 王海伶

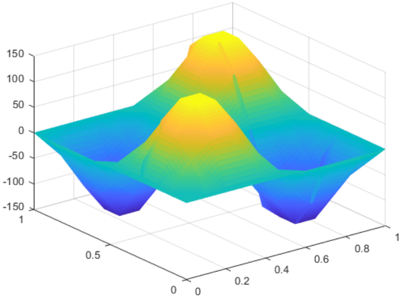

图: Control

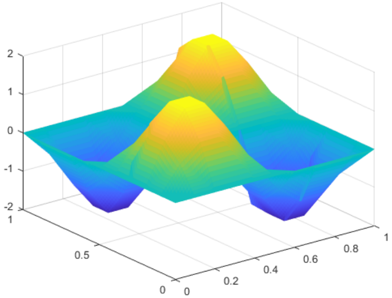

图: State

References

何炳生 (2018)
我和乘子交替方向法 20 年
[J] 运筹学学报

Boyd S , Parikh N , Chu E , et al (2010)
Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers
[J]Foundations & Trends in Machine Learning, 2010, 3(1):1-122.

刘浩洋, 户将, 李勇锋, 等 (2021)
最优化：建模、算法与理论
[M]. 高教出版社: 中国,2021:430

Long Chen(2009)
iFEM: an integrated finite element methods package in MATLAB
[CP]. available at `https://github.com/lyc102/ifem`,

谢谢大家