# ADMM

ADMM（Alternating Direction Method of Multipliers）交替方向乘子法

# 介绍

ADMM已发展许多年，由于在大数据处理上的优越性，ADMM越来越多的被应用在工程上。

ADMM由Dual Ascent对偶上升和Augmented Lagrangians and the Method of Multipliers增广拉格朗日乘子法发展而来。

# Dual Ascent

考虑一个等式约束的凸优化问题

$$\begin{aligned} &\min\ f(x) \\ &s.t.\ Ax=b \end{aligned} \tag{1}$$

利用拉格朗日乘子法

$$L(x,y)=f(x)+y^T(Ax-b) \tag{2}$$

求(1)的对偶函数

$$\begin{aligned} g(y)&=\inf_x(f(x)+y^T(Ax-b)) \\ &=\inf_x(f(x)+y^TAx)-y^Tb \\ &=-\sup_x(-f(x)-y^TAx)-y^Tb \\ &=-f^*(-A^Ty)-y^Tb \end{aligned} \tag{3}$$

其中$f^*$为$f$的共轭函数。

对原问题稍作推广，若该优化问题带有不等式约束条件，则有

$$\min\ f(x)\geq L(x,y)\geq g(y)$$

$g(y)$便是原问题最优解的下界，于是，考虑寻找最大的下界。于是，便导出拉格朗日对偶问题

$$\max\ g(y)$$

由于$f^*$为共轭函数，所以$f^*$一定为凹函数，因此该问题一定为凸优化问题。

令$\hat{y}=\arg\min_y g(y)$，$\hat{x}=\arg\min_x L(x,y)$

在**强对偶**条件下$\hat{y}=\hat{x}$

对(3)求导

$$\begin{aligned} \nabla g(y)&=-\sup_x(-Ax)-b \\ &=\inf_x(Ax)-b \end{aligned}$$

由此，便可以写出对偶上升法的算法了

$$x^{k+1} = \arg\min_x L(x, y^k)$$
$$y^{k+1} = y^k + \alpha(Ax^{k+1} - b)$$

---

在上述过程中，$g(y^{k+1}) > g(y^k)$，对偶函数一直在增大，这便是对偶上升这个名字的由来。随着$y^{k+1}$向着$\hat{y}$逼近，$x^{k+1}$也随着向$\hat{x}$逼近。

- 对偶上升法的缺点

  1对偶上升法对所优化的问题有一定的限制。2对$\alpha$的选择，也需要满足一定的假设。

# Augmented Lagrangians and the Method of Multipliers

ALM是在增强对偶上升法强度的目标下发展起来的。ALM增加了一个惩罚项，从而不再需要假定$f(x)$是严格凸的。

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

增广拉格朗日乘子法可以很轻松的被写成普通的拉格朗日乘子法问题。

$$\min_x \ f(x) + \frac{\lambda}{2}||Ax - b||_2^2$$
$$s.t. \ Ax = b$$

对该问题使用对偶上升法，便可得ALM算法：

---

$$x^{k+1} = \arg\min_x L(x, y^k)$$
$$y^{k+1} = y^k + \lambda(Ax^{k+1} - b)$$

---

ALM会比对偶上升法有更好的收敛性

- ALM的缺点

  当$f$是可分解的时候，$L(x, y)$便是不可分解的。

# ADMM

ADMM算法试图融合对偶上升法的可分解性与ALM的收敛性。ADMM所优化的问题，表述如下

$$\min_{x,z} \ f(x) + g(z)$$
$$s.t. \ Ax + Bz = c$$

应用ALM将该问题写成以下形式

$$L(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\lambda}{2}||AX + Bz - c||_2^2$$

ADMM算法表示为如下形式

---

$$x^{k+1} = \arg\min_x L(x, z^k, y^k)$$
$$z^{k+1} = \arg\min_z L(x^{k+1}, z, y^k)$$
$$y^{k+1} = y^k + \lambda(Ax^{k+1} + Bz^{k+1} - c)$$

而ALM算法表示为如下形式

$$(x^{k+1}, y^{k+1}) = \arg\min_{x,z} L(x, z, y^k)$$
$$y^{k+1} = y^k + \lambda(Ax^{k+1} + Bz^{k+1} - c)$$

可见ADMM把下x,z这两个主元分开求解了，而ALM是将x,z合在一起做优化。