# 第十二讲 交替方向乘子法（ADMM）

交替方向乘子法(Alternating Direction Method of Multipliers)，简称ADMM算法，最早分别由 Glowinski & Marrocco 及 Gabay & Mercier 于 1975 年和 1976 年提出，并被 Boyd 等人于 2011 年重新综述并证明其适用于大规模分布式凸优化问题，特别是在统计、机器学习和相关领域中出现的大规模问题。由于 ADMM 的提出早于大规模分布式计算系统和大规模优化问题的出现，所以在 2011 年以前，这种方法并不广为人知。

ADMM是ALM算法的一种延伸，只不过将无约束优化的部分用块坐标下降法（block coordinate descent，或叫做 alternating minimization）来分别优化。ADMM 通过分解协调（Decomposition-Coordination）过程，将大的全局问题分解为多个较小、较容易求解的局部子问题，并通过协调子问题的解而得到大的全局问题的解。

引进ADMM主要是为了弥补二次惩罚的缺点。在一些问题当中，用二次惩罚来近似约束问题在最优点附近需要惩罚项的系数趋近于无穷，而这种要求会使得海森矩阵很大，因此近似的目标函数很不稳定。为了解决这个问题，引入了线性逼近的部分，通过线性项系数不断的接近最优解（对偶上升），使得在二次惩罚项的系数很小的情况下，也能得到满足要求精度的解。

## 一、对偶上升法(Dual Ascent)

首先，我们考虑下面的具有等式约束的凸优化问题

 （1）

其中，是闭正常凸函数。是线性算子，。

问题（1）的拉格朗日函数和对偶函数分别为



其中是拉格朗日乘子，*f \**是*f*的共轭函数。

因此，问题（1）的对偶问题为



假设*f*是严格凸函数且强对偶定理成立，则原问题和对偶问题的最优值是相同的。从而，由对偶最优点*z\**可以得到原始最优点*x\**



此时，可以采用梯度上升法求解对偶问题，进而得到原问题（1）的最优解，这种方法称为**对偶上升法**。对偶上升法的迭代方式如下：



其中是步长，*k*是迭代步数。第一步(2)式是更新*x*，第二步(3)式是更新对偶变量。对偶变量*z*可以解释为价格向量，*z*更新则称为价格更新或价格调整步骤。通过选择合适的，每一步都能使对偶函数值增加, 因此，这个算法被称为对偶上升法。

对于对偶上升法，如果合适地选择并且其他几个假设成立，则收敛到原问题的最优值点，同时收敛到对偶问题的最优值点。然而，这些假设在许多应用中并不成立，因此通常不能使用对偶上升法。

例如，如果*f*是*x*任意分量的非零仿射函数，那么*x*将无法使用公式(2)进行更新，因为对于大多数z, 拉格朗日函数在*x*处是无下界的。

## 二、ALM

对于约束优化问题，增广拉格朗日函数（Augmented Lagrange function）及乘子法最早是在20世纪60年代末由Hestenes和Powell提出。

对于具有等式约束的凸优化问题(1)，其增广拉格朗日函数为

 （4）

其中是拉格朗日乘子，为称为惩罚参数。

增广拉格朗日函数可以看作如下与原问题相关函数的(未增广的)拉格朗日函数



这个问题显然等价于原问题(1)，因为对于任何可行的*x*，添加到目标中的项都是零。相应的对偶函数为。

加入惩罚项的好处：在原问题更加弱的假设下，*gρ*仍然是可微的。从而我们可以得到原问题修正后的对偶上升法，其迭代公式如下：



上述迭代方式称为问题（1）的**增广拉格朗日乘子法（ALM）**。该方法除了在更新*x*时使用增广拉格朗日函数外，其他的与对偶增长法相同，并且使用惩罚参数*λ*作为步长取。

在更新对偶变量(6)时，很容易想到选择特殊步长*λ*。为了简单起见，我们在这里假设*f*是可微的，尽管这不是算法所必需的。原始问题（1）和其对偶问题的最优性条件分别是



根据定义，是的最小值，所以，



从上式中可以看到，使用*λ*作为对偶变量更新的步长。随着ALM法迭代的进行，原始问题的残差收敛于零，从而得到最优性。

## 三、ADMM

ADMM是一种将对偶上升法的可分解性与ALM法的优越收敛性相结合的算法。

首先，我们考虑下面的具有可分结构的凸优化问题

 （7）

其中，是闭正常凸函数。是线性算子，。

与一般线性等式约束问题(1)的唯一区别是，变量*x*在这里被分割成两个部分，在这里被分割成*x*和*y*，目标函数在这个分割过程中是可分离的。

问题(7)对的增广拉格朗日函数为



其中是拉格朗日乘子，为称为惩罚参数。

对于原问题（7），ADMM算法的更新步骤如下：



ADMM算法非常类似于对偶上升法和ALM法：它包括一个x最小化更新步骤(8)、一个y最小化更新步骤(9)和一个乘子更新步骤(10)。与ALM法中相同，在乘子更新时使用惩罚参数*λ*作为步长。

在ADMM中，*x*和*z*以交替或顺序的方式更新，这就解释了交替方向这个术语。

下面，我们给出ADMM算法的基本收敛结果。

**定理一：**假设增广实值函数和是闭真凸函数，且拉格朗赴日函数存在有鞍点。则ADMM满足以下结论：

(1)当时，，即迭代方法是可行的；

(2)当时，，即迭代方法的目标函数收敛到最优值；

(3)当时，，其中是对偶问题最优值点。

## 四、邻近ADMM

问题(7)对的增广拉格朗日函数为



其中是拉格朗日乘子，为称为惩罚参数。

下面给出邻近ADMM算法求解问题（7）。

|  |
| --- |
| Proximal ADMM  第一步：给定初始点。  第二步：计算  计算    其中是惩罚因子，是步长，是两个自伴随的半正定算子。  第三步：如果停止条件不满足，则转第二步。 |

首先，我们给出一个约束规范（CQ）：存在，其中是约束集合。

在CQ约束规范下，是问题（1）的最优解当且仅当存在一个拉格朗日乘子使得



其中是的次微分。另一方面，因为次微分函数是闭正常凸函数，所以存在两个自伴随半正定算子使得对于所有的，有



对于所有的，有



为了方便，我们定义





下面，我们给出Proximal ADMM算法的收敛性定理。

**定理二：**假设问题（1）的解集是非空的并且满足CQ约束规范。假设是正定的。是由ADMM算法产生的点列，对于所有的，定义



则有下面结果：

（1）如果，对于所有的，我们有



（2）如果，对于所有的，我们有



（3）如果，则序列收敛到问题（1）的一个最优解，收敛到问题（1）的对偶问题的一个最优解。

（4）如果A没有并且，则对于，我们有，对于，有下面的成立



并且序列收敛到问题（1）的一个最优解，收敛到问题（1）的对偶问题的一个最优解。

**证明**：显然，序列在定理的假设下是有定义的。我们知道算法中第二步可以写成



可以推出



为方便起见，我们定义，并且



结合（3）（4）和（2）（10），我们有



将上面两个不等式相加然后根据的定义，并且根据下面的关系



可以得到



下面，我们估计（11）中的，我们注意到（10）中



我们通过的极大单调性，有



取，有



结合（11），有



通过和下面的关系，我们有



下面我们考虑步长的两种情况

**情况1.**，在这种情况下，利用下面的不等式



并且根据定义，经过简单的变形，我们可以通过（12）知（7）成立。

**情况2.，**同样的，利用不等式



同样的，根据（12）知（8）成立。

假设，从（6）-（8），我们可以得到是有界的并且



通过（5）中和（6）中的定义，我们可以知道序列和是有界的。因为是正定的，所以也是有界的。在通过下面的关系式



我们可以知道是有界的，因此是有界的，又因为是正定的，所以是有界的。因此序列是有界的。

所以存在一个收敛子列收敛到。接下来我们证明是问题（1）的一个最优解，是相应的拉格朗日乘子。

通过（13）知



并且利用下面的不等式



我们可以得到



对（10）两边取极限并且根据图的闭性，我们可以得到



即满足（2）。所以是问题（1）的一个最优解，是相应的拉格朗日乘子。下面我们证明是序列的唯一收敛点。

我们知道满足（2），所以我们可以用来替代。因为当时，子列收敛到0当时，当时，子列收敛到0当。又因为两个子列都是单调非增序列并且收敛到0，所以



，所以得到。通过（17）和（13），可以得到



所以，我们有因为是正定的。结合（14）推出，所以



又因为是正定的，所以。所以收敛到如果。

最后，我们可以看到（11）左边的第三项和第四项相互抵消如果没有A，，所以我们很容易从（11）得到（9）。并且的收敛性和第三部分的证明类似。

## 五、三块ADMM

在本章节中，我们主要给出三块ADMM算法。考虑如下凸优化问题



考虑第三项是线性算子的情况，即



对于一个给定的，（2.1）的增广拉格朗日函数为



**假设：**同样的，我们假设存在 使得

在假设的情况下，是问题（2.1）的最优解的充要条件是存在一个拉格朗日乘子使得



下面我们给出sPADMM来解决问题（2.1）

|  |
| --- |
| sPADMM3c: 3-block Proximal ADMM  取是惩罚因子，是步长，给定初始点。选择满足，令，第k步  第一步：计算  第二步： 计算  计算  第三步：计算    第四步： |

对于问题（2.1），我们知道等式约束，我们有



讲（2.3）带入问题（2.1），我们有



其中。是单位映射，我们可以知道算子具有如下性质：



给定，（2.4）的增广拉格朗日函数为



现在我们可以利用一般的两块的sPADMM来求解

|  |
| --- |
| sPADMM2s: A SPECIFIC 2-block Proximal ADMM  取是惩罚因子，是步长，给定初始点。，第k步执行  第一步：计算  第二步：计算  第三步： |

**定理三：**取是惩罚因子，是给定的参数，给定初始点。，取，那么对于所有的，我们有如下结果：

（1）由算法sPADMM2s产生的求解问题（2.4）的点等价于由算法sPADMM3c产生的求解问题（2.1）的点。

（2）满足下面的关系



**证明：**我们通过归纳法证明此定理。我们注意到和，我们有。

又因为，通过直接计算有。假设对于所有的都成立。通过（2.5）和属于的值域以及，我们可以很容易的知道对于，有



同样的，对于，我们有



其中并且



其中。综上所述，结论得以证明。对于两块sPADMM2s算法的收敛性证明我们在上一章节中已经给出。

## 参考文献

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

[2]MARYAM FAZEL, TING KEI PONG, DEFENG SUN, PAUL TSENG. Hankel matrix rank minimization with applications to system identification and realization[J]. SIAM J. MATRIX ANAL. APPL, 2013, Vol. 34, No. 3: 946–977