**ADMM算法**

考虑如下问题

**对偶分解性**

其实就是对偶上升法所利用性质，通过梯度上升法求取对偶问题的最优解，将其代入原问题中，再求取原问题的最优解。这样，便可以将问题分成两步求解。适用于原问题特别复杂，难以求解的情况。设目标函数是可分的

式中，，将A进行拆分

所以，约束条件可拆解为

则拉格朗日函数为

那么，利用对偶上升法对对偶问题和原问题进行更新的迭代公式为

要注意的是，在这里我们即把目标函数分解了，也把约束条件分解了。也就是说，我们把原问题分解成一个个子问题后，要分别取求这些子问题的最优解。相当于对每一个子问题都使用对偶上升法。

**增广拉格朗日乘数法**

增广拉格朗日乘数法其实就是引入惩罚项的拉格朗日乘数法，目的是使收敛速度更快，防止过拟合。形式如下：

其对应的问题就变为

对偶上升法迭代公式为

与常规的对偶上升法相比，最主要的改变在于使用了超参数代替来确定步长，也就是上述的迭代公式的差异。其次，在最小化拉格朗日函数时，带有惩罚项，这就使得原问题失去了可分解性。

**ADMM算法**

ADMM算法结合了对偶分解性与增广拉格朗日乘数法的优点，既能通过引入惩罚项使算法快速收敛，防止过拟合，又能对问题进行分解，对问题进行简化。

考虑以下优化问题

为n维向量，为m维向量，其实就是把前面的X拆成两部分。类似地，为矩阵，为矩阵，也是将原矩阵拆成两部分。为p维向量。

其对应的拉格朗日函数为

对应的迭代公式为