有等式约束的优化问题的求解方法

log barrier

首先, 介绍一下 log barrier 方法:

$$
\begin{array}{lll}
\min & f_0(x) & \\
\text{s.t.} & f_i(x) \leq 0 & i=1 \cdots m \\
& h_i(x)=0 & i=1 \cdots p \\
\Leftrightarrow & & \\
\min & f_0(x)-\sum_{i=1}^m u_i \log \left(-f_i(x)\right) & \\
\text{s.t.} & h_i(x)=0 & i=1 \cdots p
\end{array}
$$

这是一个惩罚的思想。具体来说就是 $f_i(x) \to 0$ 时, $p^* \to +\infty$, 会让目标函数根本无法得到最优解, 就好像有一个“栅栏”一样, 阻止 $f_i(x)$ 趋近于 0, 让它只能小于 0。
基于 log barrier 方法, 我们之后所有的约束都只讨论等式约束, 不等式约束用 log barrier 去掉。 即所有的问题都形如:

$$
\begin{array}{ll}
\min & f_0(x) \\
\text{s.t.} & h_i(x)=0 \quad i=1 \cdots p
\end{array}
$$

问题定义

$$
\begin{array}{ll}
\min & f(x) \\
\text{s.t.} & Ax=b
\end{array}
\tag{eq0}
$$

计算它的KKT条件:

$$
\begin{array}{c}
x^* \text{是最优解}, v^* \text{是拉格朗日乘子} \\
\left\{\begin{array}{ll}
Ax^*=b & \text{primal feasibility} \\
\nabla f\left(x^*\right)+A^T v^*=0 & \text{stationarity}
\end{array}\right.
\end{array}
$$

解约束优化问题就不得不用到KKT条件了， 解KKT条件(如果可解的话), 那么相当于一次性 把原问题和对偶问题的最优解都解出来了。目前有一个从KKT条件出发得出的算法, 就叫做拉格朗日乘子法]

当目标函数 $f(x)$ 是二次函数时,KKT条件是线性方程组

当目标函数 $f(x)$ 是二次函数时, KKT条件是线性方程组.例如

$$
\begin{array}{ll}
\min & \frac{1}{2} x^T P x+q^T x+r \\
\text{s.t.} & Ax=b
\end{array}
$$

其中 $P \in S_{+}^{n}$, A 一般是不可逆的(这样 x 才会是有一个解集的), 否则可逆矩阵的方程组 有唯一解就谈不上要优化了。
计算这个问题的KKT条件为:

$$\left\{\begin{array}{l} Ax^{*}=b \\ Px^{*}+q+A^{T}v^{*}=0 \end{array}\right. \Leftrightarrow \left[\begin{array}{cc} P & A^{T} \\ A & 0 \end{array}\right] \cdot \left[\begin{array}{c} x^{*} \\ v^{*} \end{array}\right]=\left[\begin{array}{c} -q \\ b \end{array}\right]$$

我们会发现，这时候KKT条件也是一个线性的方程组, 主要原因就是 $\nabla f\left(x^{*}\right)$ 是线性的, 这 个时候就可以用Gauss-Seidel等解线性方程组的方法求解最优值了。这样的情况不需要什 么下降算法, 本质就是求解线性方程组. 当然, 你也可以转化为求无约束优化的对偶问题。

当目标函数 $f(x)$ 是高次函数时,KKT条件不是线性的方程组

当 $\nabla f\left(x^{*}\right)$ 不是一个线性函数时,则有

$$\left\{\begin{array}{l} Ax^{*}=b \\ \nabla f\left(x^{*}\right)+A^{T}v^{*}=0 \end{array}\right.$$

如果 $\nabla f\left(x^{*}\right)$ 不是一个关于 x^{*} 的线性函数时, 自然会想到要找一个办法把它变成线性函数。例如使用近似的思想,如果函数二次可微的话, 可以对函数进行二阶展开。
假设现在有一个可行点 x^{k}, x^{k} 可以使上式 $(eq0)$ 成立,注意, x^{k} 只是可行解,而不一定是最优解.
现在我们构建一个问题 $(eq1)$ $(eq1)$: 在x^{k} 的邻域附近寻找一个新的点$x^{k}+d$,使得$f\left(x^{k}+d\right)$ 尽可能地小.
问题 $(eq1)$ $(eq1)$ 一定有解, 最坏情况下, $d=0$, $(eq1)$ 的解与 $(eq0)$ 的解相同. 令 $x^{k+1}=x^{k}+d$, 然后我们再将 x^{k+1} 当做 x^{k}, 再在 x^{k+1} 邻域内寻找问题 $(eq1)$ $(eq1)$ 的解, 如此往复, 则我们就可以得到原问题 $(eq0)$ 的解(或者近似解).

$$\begin{array}{ll} \min\limits_{d} & f\left(x^{k}+d\right) \\ \text{s.t.} & A\left(x^{k}+d\right)=b \\ & \text{因为} Ax^{k}=b, A\left(x^{k}+d\right)=b \text{所以} Ad=0 \end{array} \Leftrightarrow \begin{array}{ll} \min\limits_{d} & f\left(x^{k}+d\right) \\ \text{s.t.} & Ad=0 \end{array} \tag{eq1}$$

则我们现在把原问题(求解全局最优解)等价为新的优化问题(在x^{k}的邻域附近寻找一个新的点$x^{k}+d$,使得 $f\left(x^{k}+d\right)$ 尽可能地小).
这是一个关于 d 的新的优化问题。为了让原问题KKT线性, 先对问题 $(eq1)$目标函数做二阶泰勒展开

$$f\left(x^{k}+d\right)=f\left(x^{k}\right)+\nabla f\left(x^{k}\right)^{T} d+\frac{1}{2} d^{T}\left(\nabla^{2} f\left(x^{k}\right)\right) d$$

则问题 $(eq1)$又被近似等价为问题 $(eq2)$

$$\begin{array}{ll} \min\limits_{d} & f\left(x^{k}\right)+\nabla f\left(x^{k}\right)^{T} d+\frac{1}{2} d^{T} \nabla^{2} f\left(x^{k}\right)^{T} d \\ \text{s.t.} & Ad=0 \end{array}$$

求解这个问题 $(eq2)$的KKT条件为:

$$\left[\begin{array}{cc} \nabla^2 f\left(x^k\right) & A^T \\ A & 0 \end{array}\right] \cdot\left[\begin{array}{c} d^* \\ v^* \end{array}\right]=\left[\begin{array}{c} -\nabla f\left(x^k\right) \\ 0 \end{array}\right] \tag{eq2}$$

便又回到解线性方程组了。但是这里要注意的是, 我们这样近似解出来的是 d^k, 也就是用上一次的迭代点沿着 d 方向展开了得到的 d^k, 这一个过程我们进行了二阶展开这样的近似操作, 并不是求解出了原问题的最优解 但是我们可以发现, 解出来的 d^k 满足 $Ad^k=0$, 也就是说 x^k+d^k 起码是可行点,满足 $A\left(x^k+d^k\right)=b$. 到这里我们就会思考，可不可以用这个方向去下降呢? 答案是最好先别。因为 x^k+d^k 只 是可行点, 不一定相对于 x^k 是下降点。所以我们可以给 d^k 乘上一个步长, 对步长做一次线搜索，再来当作下降方向来用。 由此我们得到带等式约束的牛顿法:

$$\begin{cases} \left[\begin{array}{cc} \nabla^2 f\left(x^k\right) & A^T \\ A & 0 \end{array}\right] \cdot\left[\begin{array}{c} d^* \\ v^* \end{array}\right]=\left[\begin{array}{c} -\nabla f\left(x^k\right) \\ 0 \end{array}\right] \\ d^k=d^* \\ \alpha^k=\arg \min_{\alpha \geq 0} f\left(x^k+\alpha d^k\right) \\ x^{k+1}=x^k+\alpha^k d^k, \ x^{k+1}\text{始终满足}Ax^{k+1}=b \end{cases}$$

所以，对于带约束优化问题, KKT条件还是蛮重要的

拉格朗日乘子法Method of Lagrangian Multiplier, 拉格朗日法 Lagrangian Method

$$\begin{cases} x^{k+1}=x^k-\alpha^k\left(\nabla f\left(x^k\right)+A^T v^k\right) \\ v^{k+1}=v^k+\alpha^k\left(Ax^k-b\right) \end{cases}$$

$$\Leftrightarrow$$

$$\left[\begin{array}{c} x^{k+1} \\ v^{k+1} \end{array}\right]=\left[\begin{array}{c} x^k \\ v^k \end{array}\right]+\alpha^k\left[\begin{array}{c} -\left(\nabla f(x^k)+A^T v^k\right) \\ Ax^k-b \end{array}\right]$$

第二个式子乘子的迭代部分, x^k, v^k 肯定是可以换成上一步已经计算好的 x^{k+1}, v^{k+1}. 很明显,拉格朗日寻找下一个更新点的方法如上式,找到一个方向 $\left[\begin{array}{c} -\left(\nabla f(x^k)+A^T v^k\right) \\ Ax^k-b \end{array}\right]$, 再寻找合适的步长 α^k, 然后进行更新.

从鞍点角度理解拉格朗日法

因为

$$L(x, v)=f(x)+v^T(Ax-b)$$

所以鞍点

$$\begin{cases} (x^*, v^*) &= \arg\max\limits_{v} \min\limits_{x} L(x, v) \\ (x^*, v^*) &= \arg\min\limits_{x} \max\limits_{v} L(x, v) \end{cases}$$

我们知道，如果强对偶性成立，那么鞍点就是最优点。此时

$$x^* = \arg\min_x L\left(x, v^*\right) \text{(已知}v^*\text{求 } x^*\text{)}$$
$$v^* = \arg\max_v L\left(x^*, v\right) \text{(已知}x^*\text{求 } v^*\text{)}$$

上述求x^*, v^*的过程是无约束优化问题求极值的方法, 可以使用梯度下降或者梯度上升的算法.

如果已知某一个分量的最优点, 求另一个分量的最优点本质上就是求拉格朗日函数对于单个变量的最优解。接下来分类讨论下:

1. 当已知 v^* 求 x^* 时:
   用梯度下降法迭代求解 $L\left(x, v^*\right)$ 。因为 $-\nabla L\left(x, v^*\right) = -\nabla f(x) - A^T v^*$, 所以我们的迭代算法为 $x^{k+1} = x^k + \alpha^k \left(-\nabla f\left(x^k\right) - A^T v^*\right)$ 但是我们不可能真的已知 v^* 是什么, 所以我们用 v^k 近似。这便是拉格朗日法的关于 x 的 迭代部分： $x^{k+1} = x^k - \alpha^k \left(\nabla f\left(x^k\right) + A^T v^k\right)$
2. 当已知 x^* 求 v^* 时:
   因为我们求极大, 所以上面的迭代方向就是梯度方向,同样地 x^* 用 x^k 替代. $v^{k+1} = v^k + \alpha^k \left(Ax^k - b\right)$
   拉格朗日法实际上是在求 $L(x, v)$ 的鞍点, 来得出原问题的最优值.

从罚函数的角度理解拉格朗日法

我们把带等式约束的原问题的KKT条件:

$$\begin{cases} Ax^* = b \\ \nabla f\left(x^*\right) + A^T v^* = 0 \end{cases}$$

改写成一个罚函数的样子, 去当作一个新的优化问题来解:

$$\min P(x, v) = \frac{1}{2}\|Ax - b\|_2^2 + \frac{1}{2}\left\|\nabla f(x) + A^T v\right\|_2^2 \quad \text{(一般为非凸问题)} \tag{eq3}$$

特点: 这个新的无约束优化问题的解就是满足KKT条件的点。某种意义上说，相当于把原问题转化成了无约束优化问题, 我们可以去求这个新问题的目标函数的负梯度方向，然后梯度下降求最优解:

$$\begin{aligned} & -\nabla P\left(x^k, v^k\right) \\ & = \left[\ -\nabla_x P\left(x^k, v^k\right), -\nabla_v P\left(x^k, v^k\right)\ \right] \\ & = -\begin{bmatrix} A^T\left(Ax^k - b\right) + \nabla^2 f\left(x^k\right)\left(\nabla f\left(x^k\right) + A^T v^k\right) \\ A\left(\nabla f\left(x^k\right) + A^T v^k\right) \end{bmatrix} \end{aligned}$$

拉格朗日方法中迭代方向的正确性证明

我们选择 $-\nabla P\left(x^k, v^k\right)$ 为梯度方向,如果拉格朗日法所确定的迭代方向 d^k 与 $-\nabla P\left(x^k, v^k\right)$ 夹角小于 $\frac{\pi}{2}$,那么我们认为迭代方向 d^k 能够使函数值下降.

$$d^k=\left[\begin{array}{c}-\left(\nabla f(x^k)+A^T v^k\right) \\ A x^k-b\end{array}\right]$$

$$\begin{aligned}
&(d^k)^T\left(-\nabla P(x^k, v^k)\right) \\
&=\left(\nabla f\left(x^k\right)+A^T v^k\right)^T A^T\left(A x^k-b\right) \\
&\quad+\left(\nabla f\left(x^k\right)+A^T v^k\right)^T \nabla^2 f\left(x^k\right)\left(\nabla f\left(x^k\right)+A^T v^k\right) \\
&\quad-(A^T x^k-b)^T A(\nabla f\left(x^k\right)+A^T v^k) \\
&=\left(\nabla f\left(x^k\right)+A^T v^k\right)^T \nabla^2 f\left(x^k\right)\left(\nabla f\left(x^k\right)+A^T v^k\right)
\end{aligned}$$

上边这个式子(二次型), 当 $\nabla^2 f\left(x^k\right) \succ 0$ 且 $\left(\nabla f\left(x^k\right)+A^T v^k\right)$ 不全为 0 时, 会大于等于0 。即:
$\left\{\begin{array}{l}\nabla f\left(x^k\right)+A^T v^k \neq 0 \\ \nabla^2 f\left(x^k\right) \succ 0\end{array}\right.$ 时, d^k **是 $P\left(u^k, v^k\right)$ 的一个下降方向(和负梯度方向同向).** 而上面这个条件是成立的。因为原问题凸, 所以Hessian矩阵半正定。而当随着迭代 $x^k \rightarrow x^*$ 的时候, $\nabla^2 f\left(x^*\right)=0$, 那么大部分情况下 $\nabla^2 f\left(x^k\right)$ 都是正定的。另外, 只要 v^k, x^k 还没有到达最优解, $\nabla f\left(x^k\right)+A^T v^k \neq 0$ 就一直满足, 当 $v^* \rightarrow v^*, x^k \rightarrow x^*$ 时, $\nabla f\left(x^k\right)+A^T v^k=0$, 此时 $x^{k+1}=x^k, v^{k+1}=v^k$,算法也就停止更新了。 对拉格朗日法进行理论分析很有意义, 但是这个算法实际运行起来速度很慢, 所以为了提高一下效率, 诞生了增广拉格朗日法。

增广拉格朗日法

增广拉格朗日法相比与拉格朗日法,会构建一个新的函数,称为增广拉格朗日函数.增广拉格朗日函数是由拉格朗日函数加上一个罚项得到的, 记为:

$$L_c(x, v)=f(x)+v^T(A x-b)+\frac{c}{2}\|A x-b\|_2^2$$

它实际上是下面这个优化问题 $(eq4)$ 的拉格朗日函数:

$$\begin{array}{ll}\min & f(x)+\frac{c}{2}\|A x-b\|_2^2 \\ \text{s.t.} & A x=b\end{array} \tag{eq4}$$

也就是把原问题的等式约束变为一个惩罚项再放到目标函数里面去, 得到的优化问题的拉格朗日函数, 被称为原问题拉格朗日函数的增广拉格朗日函数。
接下来很容易可以验证, 上面 $(eq4)$ 这个问题和开头的标准问题 $(eq0)$ 的最优解相同, 两个问题是等价的。
对于问题 $(eq0)$, 其拉格朗日函数为: $L(x, v)=f(x)+v^T(A x-b)$, 假设拉格朗日函数的最优解为 (x^*, v^*), 则 $L\left(x^*, v^*\right)=0$, 即:$\nabla_x\left\{f\left(x^*\right)+v^{*T}\left(A x^*-b\right)\right\}=0$
对于问题 $(eq4)$, 其拉格朗日函数为: $L_c(x, v)=f(x)+v^T(A x-b)+\frac{c}{2}\|A x-b\|_2^2$
假设拉格朗日函数的最优解为 $\left(x_1^*, v_1^*\right)$, 则 $L_c\left(x_1^*, v_1^*\right)=0$, 即:

$\nabla_x \left\{ f\left(x_1^*\right) + v_1^{*T}\left(Ax_1^* - b\right)\right\} + cA^T\left(Ax_1^* - b\right) = 0$
又因为 $Ax_1^* - b = 0$, 所以实际上上式为 $\nabla_x \left\{ f\left(x_1^*\right) + v_1^{*T}\left(Ax_1^* - b\right)\right\} = 0$
这和 $(eq0)$ 的结论是一样的。所以说 $(x^*, v^*) = (x_1^*, v_1^*)$, 即 $(eq0) \Leftrightarrow (eq4)$.
既然两个问题是等价的，也就是它们各自的拉格朗日函数的最优值是等价的。那么, 自然可 以想到用 $\nabla L_c(x, v)$ 去替换掉 $\nabla L(x, v)$.
使用拉格朗日法来解决问题 $(eq4)$(**问题 $(eq0)$的增广拉格朗日函数就是问题 $(eq4)$的拉格朗日函数**)
$$\begin{cases} x^{k+1} = x^k - \alpha^k \nabla_x L_c\left(x^k, v^k\right) = x^k - \alpha^k\left(\nabla f\left(x^k\right) + A^T v^k + cA^T\left(Ax^k - b\right)\right) \\ v^{k+1} = v^k + \alpha^k \nabla_v L_c\left(x^k, v^k\right) = v^k + \alpha^k\left(Ax^k - b\right) \end{cases}$$
但是上面这组迭代式子, 有个东西没有利用起来, 就是第一个式子会更新好的 x^{k+1} 。在第二式子计算 v^{k+1} 时用的还是 x^k 的信息, 这是把它替换成新的 x^{k+1}, 那么整个算法变为:
$$\begin{cases} x^{k+1} = x^k - \alpha^k\left(\nabla f\left(x^k\right) + A^T v^k + cA^T\left(Ax^k - b\right)\right) \\ v^{k+1} = v^k + \alpha^k\left(Ax^{k+1} - b\right) \end{cases}$$
如果增广拉格朗日函数 $L_c(x, v)$ 性质比较理想或者比较容易被优化的话, 不一定非要用梯度下降法, 比如二阶可微也可以用牛顿法, 等等。所以统一改成用 $\arg\min$ 的形式来表达上面这 个算法为:
$$\begin{cases} x^{k+1} = \arg\min\limits_x L_c\left(x, v^k\right) \\ v^{k+1} = v^k + \alpha^k\left(Ax^{k+1} - b\right) \end{cases}$$
我们还可以把第二个式子的步长替换为罚参数, 得到最终的增广拉格朗日法:
$$\begin{cases} x^{k+1} = \arg\min\limits_x L_c\left(x, v^k\right) \\ v^{k+1} = v^k + c\left(Ax^{k+1} - b\right) \end{cases}$$

因此最终的增广拉格朗日法 $\mathrm{Augmented\ Lagrangian\ Method}$为

$$\begin{aligned} x^{k+1} &= \arg\min_x L_c\left(x, v^k\right) \\ v^{k+1} &= v^k + c\left(Ax^{k+1} - b\right) \end{aligned}$$

步长 α^k 都是要自己找的, 比如精确搜索或非精确搜索.

注：这里 c 要自己选定, 但是选定 c 肯定比去线搜索 α 简单多了。特别的，如果选择 $c = 1$, 基本上可以保证增广拉格朗日算法收敛。而且, 从收敛性分析可以知道, c 也可以 是递增序列，不用担心不收敛。

算法性质:

(1). 若 $v = v^*$,则 $\forall c > 0, x^* = \arg\min\limits_x L_c\left(x, v^*\right)$, 只要 $v \to v^*, c$ 的值只要大于0就可以得到最优解 x^*
(2). 若 $c \to +\infty$,则 $\forall v, \quad x^* = \arg\min\limits_x L_c(x, v)$, 步长 c 可以选择递增序列 $\left\{c^k\right\}$, 此时一定收敛.假设 $c = +\infty$