逐步二次规划 SQP 算法

姓名：陈泽豪　学号：SA22001009

2023 年 4 月 29 日

摘要

本报告将会针对逐步二次规划 SQP 算法，给出其具体实现的理论描述以及实验结果展示，最后给出本次 SQP 算法实验的收获以及总结。

一、SQP 算法介绍

1.1 对 SQP 算法的简要流程描述

SQP（sequential quadratic programming）算法，即逐步二次规划算法，是一类典型的用来处理非线性约束最优化问题的方法。我们可以从等式约束出发推广到不等式约束，对于如下的非线性等式约束最优化问题：

$$\begin{aligned} &min \quad f(x) \\ &s.t. \quad c(x) = 0 \end{aligned} \tag{1}$$

其中 $c(x) = (c_1(x), c_2(x), ..., c_m(x))^T$ 为约束函数向量。那么写出它的 (K-T) 条件如下所示：

$$G(x) = \begin{cases} L(x,\lambda) = \nabla f(x) - A(x)^T\lambda = 0 \\ c(x) = 0 \\ A(x) = (\nabla c(x)) = (\nabla c_1(x), ..., \nabla c_m(x))^T \in \mathbf{R}^{m\times n} \end{cases} \tag{2}$$

其中 $\lambda \in \mathbf{R}^m$. 那么利用 $Newton - Raphson$ 迭代求解 (2) 对应的等式组零点便有:

$$\begin{aligned} J(G(x)) \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} &= -G(x) \\ \begin{pmatrix} W(x,\lambda) & -A(x)^T \\ -A(x) & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} &= - \begin{pmatrix} \nabla f(x) - A(x)^T\lambda \\ -c(x) \end{pmatrix} \end{aligned} \tag{3}$$

其中有 $W(x,\lambda) = \nabla^2 L(x,\lambda) = \nabla^2 f(x) - \sum_{i=1}^m \lambda_i \nabla^2 c_i(x)$。在求解出 δ_x, δ_λ 之后更新给的初值 x, λ，并重新进行上述方程组的计算，最终利用价值函数 $\psi = \|\nabla f(x) - A(x)^T\lambda\|^2 + \|c(x)\|^2$ 作为终止判定条件。另一种思路：我们将 (3) 中的矩阵等式进行改写，可以变成如下的形式：

$$\begin{cases} W(x,\lambda)\delta_x + \nabla f(x) = A(x)^T(\lambda + \delta_\lambda) \\ c(x) + A(x)^T\delta_x = 0 \end{cases} \tag{4}$$

而我们可以进一步发现有此时的 δ_{x_k} 就是如下二次规划问题的 (K-T) 点：

$$\begin{aligned} \min_d \quad & \tfrac{1}{2} d^T W(x^{(k)}, \lambda^{(k)}) d + \nabla f(x^{(k)})^T d \\ s.t. \quad & c(x^{(k)}) + A(x^{(k)})^T d = 0 \end{aligned} \tag{5}$$

求解这个 QP 问题 dual problem，便有解 $d^{(k)}$ 满足 $d^{(k)} = \delta_{x^{(k)}}$，求解出来的 lagrange multipliers $\overline{\lambda}^{(k)}$ 满足 $\overline{\lambda}^{(k)} = \lambda^{(k)} + \delta_{\lambda^{(k)}}$。因此与上面类似的流程，每次都去更新 x, λ，并用一个价值函数进行终止判定以及步长调整。

在有了上面的等式约束 SQP 问题的铺垫之后，对于下面的不等式约束优化问题：

$$\begin{aligned} &min \quad f(x) \\ &s.t. \quad c_i(x) = 0, i \in \epsilon = \{1, ..., m_e\}, \\ &\qquad c_i(x) \geq 0, i \in I = \{m_e + 1, ..., m\}. \end{aligned} \tag{6}$$

便也是同样的处理思路，它的 lagrange 函数如下：

$$L(x,\lambda)=f(x)-\Big(\lambda_1,...,\lambda_{m_e},\lambda_{m_e+1},...,\lambda_m\Big)\begin{pmatrix}c_1(x)\\...\\c_{m_e}(x)\\c_{m_e+1}(x)\\...\\c_m(x)\end{pmatrix}=f(x)-\lambda^Tc(x) \tag{7}$$

此时矩阵等式 (4) 由于不等式约束的加入会发生变化，如下所示：

$$\begin{cases}W(x^{(k)},\lambda^{(k)})d^{(k)}+\nabla f(x^{(k)})=A(x^{(k)})^T\overline{\lambda}^{(k)}.\\ \overline{\lambda}_i^{(k)}\geq 0,i\in I.\\ c(x^{(k)})+A(x^{(k)})^Td^{(k)}=0.\\ d^{(k)}=\delta_{x^{(k)}}.\\ \overline{\lambda}^{(k)}=\lambda^{(k)}+\delta_{\lambda^{(k)}}.\end{cases} \tag{8}$$

而它就相当于求解如下的二次规划问题：

$$\begin{aligned}\min_d\quad & \tfrac{1}{2}d^TW(x^{(k)},\lambda^{(k)})d+\nabla f(x^{(k)})^Td\\ s.t.\quad & c_i(x^{(k)})+a_i(x^{(k)})d=0,i\in\epsilon\\ & c_i(x^{(k)})+a_i(x^{(k)})d\geq 0,i\in I\end{aligned} \tag{9}$$

其中有 $A(x)=(\nabla c_1(x),...,\nabla c_m(x))^T=(a_1(x),...,a_m(x))^T$，同样的，(9) 中解出来的解 $d^{(k)}$ 满足 $d^{(k)}=\delta_{x^{(k)}}$，求解出来的 lagrange multipliers $\overline{\lambda}^{(k)}$ 满足 $\overline{\lambda}^{(k)}=\lambda^{(k)}+\delta_{\lambda^{(k)}}$。

另一个与等式约束 SQP 不同的地方是，此时我们使用罚函数 $P(x,\sigma)$ 进行步长的调整，对于不等式约束问题，我们不希望下一步的 x^{k+1} 落在可行域外，因此需要使用罚项，当落在可行域外时使罚项变得很大，以此迫使 $\underset{\alpha}{argmin}P(x^{(k)}+\alpha d^{(k)},\sigma)$ 解出来的 α 满足 $x^{(k+1)}=x^{(k)}+\alpha d^{(k)}$ 落在可行域内。$P(x,\sigma)$ 如下所示定义：

$$\begin{aligned}&P(x,\sigma)=f(x)+\sigma\textstyle\sum_{i=1}^m|c_i(x)\_|\\ &c_i(x)\_=\begin{cases}c_i(x),i\in\epsilon\\ min(0,c_i(x)),i\in I\end{cases}\end{aligned} \tag{10}$$

整个算法流程如下所示：

1. 给定 $x^{(0)},\lambda^{(0)},\sigma>0,\rho\in(0,1),\epsilon\geq 0$，令 $k=0$。
2. 利用 $x^{(k)},\lambda^{(k)}$ 求出 $W(x^{(k)},\lambda^{(k)}),\nabla f(x^{(k)}),c(x^{(k)}),A(x^{(k)})$。求解子问题 (9) 得到解 $d^{(k)}$ 以及 lagrange multipliers $\overline{\lambda}^{(k)}$，如果此时 $\|d^{(k)}\|\leq\epsilon$，则停止整个过程，否则进行一维搜索:

$$\alpha_k=\underset{\alpha}{argmin}P(x^{(k)}+\alpha d^{(k)},\sigma) \tag{11}$$

3. 令 $x^{(k+1)}=x^{(k)}+\alpha_kd^{(k)},\lambda^{(k+1)}=\lambda^{(k)}+\alpha_k(\overline{\lambda}^{(k)}-\lambda^{(k)}),k=k+1$，转到 2 继续进行算法。

二、实验配置

三、对数据集的处理

四、实验过程

五、超参数测试

六、实验结论