

# 第三次作业

宋晓宇

## 1.

### 梯度下降法

迭代思想：从初始点出发，沿着负梯度方向移动，搜索。

逼近思想：每次迭代中，随着变量值的不断更新，逐渐逼近至局部最小值，乃至全局最小值。

### 牛顿法

迭代思想：利用二阶导数（Hessian 矩阵  $H$ ）的信息，加速收敛，逐步迭代至极小点。

逼近思想：牛顿法结合二阶导数的信息，更精确地逼近函数的极小值点。

### 共轭梯度法

迭代思想：用于求解线性方程组或二次函数优化。每次迭代方向与前一方向共轭，避免“锯齿现象”。

逼近思想：通过构造共轭方向，确保在  $n$  步内（ $n$  为维度数）精确收敛到二次函数的最优解，逼近效率高于梯度下降。

### 拟牛顿法

迭代思想：用近似矩阵替代 Hessian 矩阵，避免直接计算二阶导数。

逼近思想：通过低秩更新（如秩-2 BFGS 公式）逼近 Hessian 的逆，兼具牛顿法的快速收敛和梯度下降的低计算成本。

2.

$$\nabla f(x_k) \cdot d \leq 0$$

$$\nabla h(x_k) \cdot d = 0$$

$$\nabla g(x_k) \cdot d \leq 0 \quad \text{or} \quad g(x_k) > 0$$

3.

对原函数求梯度,  $g(x) = [1 + 4x_1 + 2x_2, -1 + 2x_2 + 2x_1]$ , 取迭代步长为  $d = 0.1$ , 有

$$g(x_0) = (1, -1)^T$$

此时  $\phi(d) = d^2 + 2d$  当  $d = -1$  时取最小值, 此时  $x = (-1, 1)$

在  $x = (-1, 1)$  上,  $g(x) = (-1, -1)$ , 此时  $\phi(d) = 5d^2 + 2d - 1$ , 当  $d = -0.1$  时取最小值. 此时  $x = (-0.9, 1.1)$

4.

a)

梯度  $g(x) = (4x_1 - 2x_2 - 4, 4x_2 - 2x_1 - 6)$

hessian 矩阵  $H(x) = \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix}$

目标函数的临界点为  $(1, 2)^T$ , 经验证, 它不满足极值条件, 不可能为极值点

b)

各约束条件分别为

$$\begin{cases} x_1 + x_2 \leq 2 \\ x_1 + 5x_2 \leq 5 \\ -x_1 \leq 0 \\ -x_2 \leq 0 \end{cases}$$

其梯度分别为:

$$\begin{cases} g_1(x) = (1, 1)^T \\ g_2(x) = (1, 5)^T \\ g_3(x) = (-1, 0)^T \\ g_4(x) = (0, -1)^T \end{cases}$$

c)

在该点  $(0, 0)$  的梯度为  $g((0, 0)^T) = (-4, -6)^T$ , 此时  $g_3, g_4$  处于临界状态, 故可行方向的要求是:

$$\begin{cases} d \cdot (-1, 0)^T \leq 0 \\ d \cdot (0, -1)^T \leq 0 \\ \text{最小化 } d \cdot (-4, -6) \end{cases}$$

解得  $d = (2, 3)^T$ , 于是在该方向上  $\phi(d) = 14d^2 - 26d$ , 此时考虑约束

$$\begin{cases} 5d - 2 \leq 0 \\ 17d - 5 \leq 0 \\ -2d \leq 0 \\ -3d \leq 0 \end{cases}$$

解得  $0 \leq d \leq \frac{5}{17}$

$\phi(d)$  在  $d = \frac{13}{14}$  取最小值, 但是该最小值不在约束条件内, 最多可以取到  $d = \frac{5}{17}$ , 此时  $x = (\frac{10}{17}, \frac{15}{17})$

此时  $g_2 : x_1 + 5x_2 \leq 5$  为有效约束, 函数梯度为  $(-\frac{58}{17}, -\frac{62}{17})^T$  于是  $d$  的约束为

$$\begin{cases} d \cdot (1, 5)^T \leq 0 \\ \text{最小化 } d \cdot (-\frac{58}{17}, -\frac{62}{17})^T \leq 0 \end{cases}$$

解得合适的可行方向  $d = (5, -1)$ , 此时  $\phi(d) = 62d^2 - \frac{228d}{17} - \frac{1860}{289}$ ,

$\phi(d)$  在  $d = \frac{57}{527}$  处取最小值, 此时  $x = (\frac{35}{31}, \frac{24}{31})$ , 在该点, 函数没有更多的可行方向.

## 5.

外点法 (罚函数法):

$$\min f(x) + \mu \sum_{i=1}^4 [\max(0, g_i(x))]^2, \quad s.t. \quad \begin{cases} g_1(x) = x_1^2 - x_2 + 2 \\ g_2(x) = x_1 + x_2 - 6 \\ g_3(x) = x_1 \geq 0 \\ g_4(x) = x_2 \geq 0 \end{cases}$$

$\mu$  是一个很大的正数, 作为约束条件的惩罚系数.

内点法 (障碍函数法):

$$\min f(x) + \mu B(x) \quad s.t. \quad x \in \mathbb{R}^n$$

其中:

$$B(x) = -\frac{1}{x_1^2 - x_2 + 1} - \frac{1}{x_1 + x_2 - 6} + \frac{1}{x_1} + \frac{1}{x_2}$$

$\mu$  为约束条件的惩罚系数.

## 6.

a)

在拉格朗日方程  $L(x, u, v)$  中, 对于每一个固定的  $x$ ,  $L(x, u, v)$  对于  $(u, v)$  都是仿射的, 根据仿射函数的性质 (仿射函数既是凸的也是凹的), 可以得到  $L(x, u, v)$  对  $(u, v)$  是凹的.  $\theta(u, v)$  是对  $L(x, u, v)$  取 pointwise infimum 的结果, 所以  $\theta(u, v)$  是凹的.

b)

对偶问题在最优化中的作用: 通过 (1) 中的证明我们可以知道, 一个优化问题的拉格朗日对偶问题一定是一个凹问题, 这对于原问题的目标函数和不等式约束的凸性没有要求. 同时, 对偶的性质还保证的对偶问题的最优解一定是原问题的下界, 即有  $\theta(u, v) \leq p^*, \forall u \geq 0, \forall v$ , 其中  $p^*$  是原问题的最优解, 在弱对偶定理和强对偶定理的帮助下, 可以很容易的估计原问题的最优解的界, 甚至将原问题转换成对偶问题求解.

弱对偶定理:  $\inf f(x)|x \in S \geq \sup \theta(u, v)$ , 即对偶问题的上界是原问题的下界, 但是不等式的等号不一定能够取到.

强对偶定理: 存在  $\hat{x}$ , 使得  $\inf\{f(\hat{x})|\hat{x} \in S\} = \sup\{\theta(u, v)\}$ .

7.

a)

设该约束问题不等式约束的 lagrange 乘子是  $\lambda_i, i = 1, 2, 3, 4$ . 该问题的 KKT 条件为

$$\begin{cases} g_i(x) \leq 0 & , i = 1, 2, 3, 4 \\ \lambda_i \geq 0 & , i = 1, 2, 3, 4 \\ \nabla_x f(x) + \sum_{i=1}^4 \lambda_i * \nabla_x g_i(x) = 0 \\ \lambda_i * g_i(x) = 0 & , i = 1, 2, 3, 4 \end{cases}$$

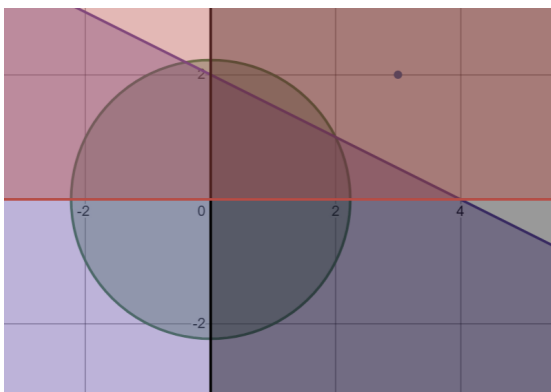
b)

$(0, 0)^T$  不满足 KKT 条件, 求得的  $\lambda = (0, 0, -6, -4)^T$  不满足非负要求.

c)

$(2, 1)^T$  满足 KKT 条件, 求得的  $\lambda = (1/3, 2/3, 0, 0)^T$  满足非负要求.

d)



如图所示, 在  $(0,0)^T$  处, 起作用集的梯度在目标函数下降方向上的投影和下降方向相反, 所以得到的乘子为负, 不符合要求. 在  $(2,1)^T$  处, 起作用集的梯度在目标函数下降方向的投影与下降方向相同, 所以得到的乘子为正, 梯度下降方向可以由起作用集梯度线性组合.

8.

a)

$$\partial f(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \\ [-1, 1] & x = 0 \end{cases}$$

b)

$$\partial f(x) = \begin{cases} x & x < -1 \\ [-1, 0] & x = -1 \\ 0 & -1 < x < 1 \\ [0, 1] & x = 1 \\ x & x > 1 \end{cases}$$

9.

启发式算法基本思想: 在可接受的计算代价下找到最好的解, 但不一定能保证解的最优性, 是一种近似算法. 通常是基于某种直观感觉或者经验构造的算法, 在解决大规模, 复杂的组合优化问题是有较好的表现. 与最优化算法不同, 启发式算法不需要提供一个最优解, 而是提供一个在可接受精度范围内的解, 是精度和速度的权衡.

下面尝试给出点集匹配问题的启发式算法, 首先形式化描述该问题, 设点集  $P, Q$  分别是模板点集和目标点集, 求映射  $f(p_i) = q_j, \forall p_i \in P$ , 目标函数通常定义为

$$g(P, Q) = \sum_{i,j \in P} d(p_i, p_j) - d(f(p_i) - f(p_j)) + \lambda(n_P - n_{P'})$$

其中  $f(p_i)$  是目标点集的一个点,  $p_i$  是模板点集的一个点,  $d(p_i, p_j)$  是  $P$  中两个点的距离,  $n_P, n_{P'}, \lambda$  分别是模板点集中的点总个数, 模板点集中匹配上的点个数和惩罚系数. 显然, 惩罚系数越大, 强迫找到一个该问题的最大匹配.

问题的难点在于如何得到映射  $f$ , 考虑实际问题, 可能是一个非线性的映射, 如何建模该映射, 以及如何找到一个最优的映射.

可以考虑用人工神经网络来建模该映射, 直观上的理解神经网络是一个通用的函数拟合模型, 对于神经网络参数的学习, 可以通过引入模拟退火算法的随机梯度下降来获得一个较优的参数解. 可能存在的问题是有监督学习可能需要大量的样本对, 对于图像的点集匹配问题, 可以收集一定数量的图片, 然后通过随机裁剪来获得样本对, 尝试训练模型, 得到问题的解.

## 10.

几种求解大规模问题的分布式优化算法: 可以简单的将分布式优化算法按照有无中心节点分成两类:

- 有中心节点算法基本思想是每个节点计算自身梯度, 将信息传递给中心节点, 中心节点汇聚所有节点信息后计算决策变量, 然后将决策变量回传给各个子节点.
  - 该算法的优点是收敛速度快, 适合大规模问题的求解.
  - 该算法的缺点是中心节点需要收集所有局部节点的信息有较大的延迟, 中心节点需要较大的带宽, 优化的过程也不够鲁棒.

常见的有中心节点的分布式优化算法有 ADMM, PSGD 等, 其优点是收敛速度快, 但是缺点是中心节点需要收集所有局部节点的信息有较大的延迟, 中心节点需要较大的带宽, 优化的过程也不够鲁棒.

- 无中心节点算法基本思想是每个节点计算自身梯度, 然后将信息传递给邻居, 每个节点通过自身和邻居节点的信息更新.

常见的有分布式梯度下降算法 DGD, 其优点是通信负载均衡到每一个节点, 鲁棒性和隐私性都好, 缺点是算法的设计需要单独考虑.

## 11.

罚因子方法: 将约束作为惩罚条件加入目标函数中, 得到一个原约束问题的无约束辅助问题, 利用无约束优化算法进行求解。主要有内点法和外点法两种形式。

- 外点法: 可以处理等式约束和不等式约束的情况, 通过添加惩罚项, 迫使迭代点向约束的可行域靠近。由于初始点的选取可以在约束可行域外随机选取, 称为外点法。
- 内点法: 只能处理不等式约束的情况, 将惩罚添加在约束集边界。当可行解在约束界内, 约束较小 (可忽略); 当解在约束界外, 惩罚趋于无穷大, 又称为罚函数法。需要注意的是, 内点的初始点和迭代点需要在约束可行域内。

罚函数法通过构造辅助问题, 将约束优化问题转换成无约束优化问题求解。但是, 辅助问题的最优解想要达到对原问题解的较好近似, 常常需要罚因子趋于无穷, 此时出现了  $0 \cdot \infty$  的情况, 可能会导致较大误差, 实际求解变得很不方便。

增广拉格朗日方法的基本思想是通过将约束条件作为惩罚项加入拉格朗日函数, 得到增广的拉格朗日函数。这种做法的好处在于利用了拉格朗日函数在 KKT 点导数为零的稳定性条件, 克服了罚函数法中因为  $\nabla f(x) = 0$  而出现的困难。同时, 相比于拉格朗日法, 增广后的方法也有明显的好处, 即不再需要交替求解原始问题和对偶问题的解。同时, 算法的速度和稳定性也有很大的提升, 不再依赖每次搜索的步长。实际上, 增广拉格朗日方法通过收敛性定理保证了对于大多数惩罚因子  $c$  都有良好的收敛性, 并且仅需关注求解对偶问题的解, 即可通过解优化问题得到原始问题的解。

## 12.

对于凸问题, 局部最优解就是全局最优解 (更准确的说法是严格凸函数), 同时, 非凸问题的困难在于在高维空间中, 存在许多鞍点 (梯度为零, 但是 Hessian 矩阵不定)。

回顾凸问题的定义, 需要满足两个条件 (1) 问题的可行域是一个凸集 (2) 目标函数是可行域上的一个凸函数。

在非凸问题转化成凸问题的过程中可以从这两个方向入手:



- 修改目标函数，使其成为一个凸函数，这样就可以满足条件（1）。
- 抛弃一些约束条件，或者对约束条件做松弛处理，是的新的可行域是凸集，同时包含原可行域的所有点