组合优化与凸优化作业 2

胡冠宇

所有代码见附件或 GitHub 备份[1].

1

使用共轭梯度法求解

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2,$$

得到最优解 $\boldsymbol{x}^* = (-1, 1.5)^{\mathsf{T}}$, 最优值 $f(\boldsymbol{x}^*) = -1.25$.

2

分别使用 Golden Search, Fibonacci Search, Bisection Search 求解

$$\min_{\boldsymbol{x}} \ f(x) = 2x^2 - x - 1 \tag{1}$$

和

$$\min_{\boldsymbol{x}} \ f(x) = 3x^2 - x - 1, \tag{2}$$

结果见表 1 和表 2 .

表 1　不同方法求解公式 (1) 的结果

方法	最优解	最优值	迭代轮次
Golden Search	0.2574	-1.1249	8
Fibonacci Search	0.2500	-1.1250	30
Bisection Search	0.2500	-1.1250	2
ShubertPiyavskii Search ($l = 2$)	0.2500	-1.1250	0

[1] https://github.com/ikaroinory/convex-opt/tree/main/homework/homework2/src

表 2　不同方法求解公式 (2) 的结果

方法	最优解	最优值	迭代轮次
Golden Search	3.6086	−39.8798	12
Fibonacci Search	3.6000	−39.8800	36
Bisection Search	3.5889	−39.8796	9
ShubertPiyavskii Search ($l=5$)	0.0000	0.0000	0

3

分别使用 Goldstein 法, Armijo 法, Wolfe-Powell 法及改进 Wolfe-Powell 法求解

$$\min_{\lambda}\ f(\boldsymbol{x}+\lambda\boldsymbol{d}), \tag{3}$$

其中 $f(\boldsymbol{x})=100(x_2-x_1^2)^2+(1-x_1)^2$, 结果见表 3 所示.

表 3　不同方法求解公式 (3) 的结果

方法	最优解	迭代轮次
Goldstein	0.0042	52
Armijo	0.0039	8
WolfePowell	0.0001	13
Improved WolfePowell	0.0020	9

4

(1)

$$\begin{aligned}\partial f(\mathbf{0}) &= \left\{\boldsymbol{g}\in\mathbb{R}^3 \;\middle|\; f(\boldsymbol{x})\geq f(\mathbf{0})+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{x},\ f(\boldsymbol{x})=\max\{|\boldsymbol{x}|\}\right\}\\ &= \left\{\boldsymbol{g}\in\mathbb{R}^3 \;\middle|\; \max\{|\boldsymbol{x}|\}\geq \boldsymbol{g}^{\mathrm{T}}\boldsymbol{x}\right\}\\ &= \left\{\boldsymbol{g}\in\mathbb{R}^3 \;\middle|\; \|\boldsymbol{g}\|_1\leq (1,1,1)^{\mathrm{T}}\right\}.\end{aligned}$$

(2)

$$
\begin{aligned}
\partial f(0) &= \left\{g \in \mathbb{R} \mid f(x) \geq f(0) + gx,\ f(x) = \mathrm{e}^{|x|}\right\} \\
&= \left\{g \in \mathbb{R} \mid \mathrm{e}^{|x|} \geq 1 + gx\right\} \\
&= [-1, 1].
\end{aligned}
$$

(3)

$$
\partial f(\boldsymbol{x}_0) = \left\{\boldsymbol{g} \in \mathbb{R}^2 \;\middle|\; f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{x},\ f(\boldsymbol{x}) = \max\{f_1(\boldsymbol{x}), f_2(\boldsymbol{x})\}\right\},
$$

其中 $\boldsymbol{x}_0 = (1, 1)$, $f_1(\boldsymbol{x}) = x_1 + x_2 - 1, f_2(\boldsymbol{x}) = x_1 - x_2 + 1$. 由 Finite pointwise maximum 可知,

$$
\partial f(\boldsymbol{x}_0) = \operatorname{conv}\left\{\bigcup_{i \in \{i | f_i(\boldsymbol{x_0}) = f(\boldsymbol{x_0})\}} \partial f_i(\boldsymbol{x}_0)\right\} = \left\{(1, \lambda)^{\mathrm{T}} \mid \lambda \in [-1, 1]\right\}.
$$

5

使用 DFP 法求解

$$
\min_{x_1, x_2}\ 10x_1^2 + x_2^2, \tag{4}
$$

得到最优解 $\boldsymbol{x}^* = (0, 0)^{\mathrm{T}}$, 最优值 $f(\boldsymbol{x}^*) = 0$.

6

使用 BFGS 法求解

$$
\min_{x_1, x_2}\ x_1^2 + 4x_2^2 - 4x_1 - 8x_2,
$$

得到最优解 $\boldsymbol{x}^* = (2, 1)^{\mathrm{T}}$, 最优值 $f(\boldsymbol{x}^*) = -8$.

7

分别使用 DFP 法, BFGS 法和共轭梯度法 (FR) 求解

$$
\min_{x_1, x_2}\ x_1^2 + x_2^2 - x_1x_2 - 10x_1 - 4x_2 + 60, \tag{5}
$$

结果见表 4 .

表 4　不同方法求解公式 (5) 的结果

方法	最优解	最优值	迭代轮次
DFP	$(0,0)^{\mathrm{T}}$	60	0
BFGS	$(0,0)^{\mathrm{T}}$	60	0
Conjugate Gradient (FR)	$(-0.1014,-0.0405)^{\mathrm{T}}$	61.1836	1

8

1. 考虑凸函数 $f(\boldsymbol{x})$ 一阶连续可导, 假设其梯度 Lipschitz 连续, 即

$$\|\nabla f(\boldsymbol{y})-\nabla f(\boldsymbol{x})\| \leq L\|\boldsymbol{y}-\boldsymbol{x}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

设梯度下降法迭代形式为

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\lambda \nabla f(\boldsymbol{x}_k), \quad \lambda_k \in\left(0, \frac{2}{L}\right),$$

则有

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k)-\frac{\lambda_k}{2}\|\nabla f(\boldsymbol{x}_k)\|^2,$$

由此得出梯度下降法是单调下降的, 且目标函数值逐步趋于最小值. 进一步地, 如果 $f(\boldsymbol{x})$ 是强凸的, 则可得线性收敛率

$$\|\boldsymbol{x}_k-\boldsymbol{x}^*\| \leq C \rho^k, \quad \rho \in(0,1).$$

2. 设 $f(\boldsymbol{x})$ 在最优点附近二阶可导且 Hessian 矩阵正定. 由 Newton 法迭代公式

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\lambda_k\left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k)$$

得到, 在最优点 $\boldsymbol{x}^*$ 附近, 若满足

$$\begin{cases}\nabla f(\boldsymbol{x}^*)=\mathbf{0}, \\ \nabla^2 f(\boldsymbol{x}^*)>\mathbf{0}, \\ \text{Hessian 连续,}\end{cases}$$

则容易证明

$$\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\| \leq C\|\boldsymbol{x}_k-\boldsymbol{x}^*\|^2.$$

即牛顿法具有二次收敛速度.

3. 设目标函数为

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x},$$

   其中矩阵 $\boldsymbol{Q}$ 正定, 且梯度

$$\nabla f(\boldsymbol{x})=\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{c}.$$

   使用梯度下降法, 有

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k-\lambda_k(\boldsymbol{Q}\boldsymbol{x}_k+\boldsymbol{c}).$$

   设最优解为 $\boldsymbol{x}^*$, 且满足 $\boldsymbol{Q}\boldsymbol{x}^*+\boldsymbol{c}=\mathbf{0}$, 解得 $\boldsymbol{x}^*=-\boldsymbol{Q}^{-1}\boldsymbol{c}$. 定义误差 $\boldsymbol{e}_k=\boldsymbol{x}_k-\boldsymbol{x}^*$, 则有

$$\begin{aligned}\boldsymbol{e}_{k+1}&=\boldsymbol{x}_{k+1}-\boldsymbol{x}^*\\&=\boldsymbol{x}_k-\lambda_k(\boldsymbol{Q}\boldsymbol{x}_k+\boldsymbol{c})-\boldsymbol{x}^*\\&=\boldsymbol{e}_k-\lambda_k\boldsymbol{Q}\boldsymbol{e}_k\\&=(\boldsymbol{I}-\lambda_k\boldsymbol{Q})\boldsymbol{e}_k,\end{aligned}$$

   于是

$$\boldsymbol{e}_k=(\boldsymbol{I}-\lambda_k\boldsymbol{Q})^k\boldsymbol{e}_{\mathbf{0}}.$$

   因此, 只要选择合适的 λ_k, 则收敛因子 $\|\boldsymbol{I}-\lambda_k\boldsymbol{Q}\|<1$, 从而保证梯度下降线性收敛.

9

无约束优化问题形式为

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \ f(\boldsymbol{x}),$$

其中 $f(\boldsymbol{x})$ 为目标函数, 其定义在整个空间内, 且无其他约束条件.

无约束优化问题的求解通常有两类方法

1. **梯度类方法**.

   梯度类方法利用梯度或 Hessian 矩阵信息, 来选取合适的搜索方向, 即

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\lambda_k\nabla f\left(\boldsymbol{x}^{(k)}\right).$$

   常见的梯度类方法有最速下降法, 牛顿法, 共轭梯度法等.

2. **搜索方法**.

   搜索方法不使用任何梯度或 Hessian 矩阵信息, 直接根据约束在空间中挑选线性无关的方向搜索. 常见的搜索方法有模式搜索, Rosenbrock 法, 单纯形搜索, Powell 方

法等.

求解无约束优化问题通常有两个步骤:

1. 确定搜索方向.
2. 确定搜索步长.

非凸优化难以求解, 因此在理论和工程中常希望将其转化为凸问题. 常用方法有:

1. 松弛.

   将原问题的非凸部分用凸函数近似或放松.

2. 凸包方法.

   对非凸函数取其最紧凸上界, 构成凸优化问题.

3. 重参数化或替代变量.

   通过变量替换, 使目标函数或约束变凸.

4. 局部凸化.

   将非凸函数在某个邻域内近似为凸函数.

5. 正则化与惩罚项.

   加入凸正则项, 引导目标函数变得更凸.

10

最速下降法, Newton 法, 修正 Newton 法的计算公式可以统一描述为

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \lambda_k \boldsymbol{H}_k \nabla f(\boldsymbol{x}_k),$$

其中, λ_k 为迭代步长, $\boldsymbol{H}_k$ 为正定对称矩阵.

最速下降法, Newton 法, 修正 Newton 法的不同点在于:

- 最速下降法: $\boldsymbol{H}_k = \boldsymbol{I}$, 即

  $$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \lambda_k \nabla f(\boldsymbol{x}_k).$$

  最速下降法收敛速度慢, 易出现锯齿形路径, 尤其在等高线椭圆扁长时.

- Newton 法: $\boldsymbol{H}_k = \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1}$, 即

  $$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \lambda_k \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k).$$

Newton 法使用了二阶信息, 因此收敛速度快, 但 Hessian 矩阵计算复杂, 且可能不可逆或非正定.

- 修正 Newton 法: $\boldsymbol{B}_k \approx \nabla^2 f(\boldsymbol{x}_k)$, 即

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \lambda_k \boldsymbol{B}_k^{-1} \nabla f(\boldsymbol{x}_k).$$

修正 Newton 法避免直接计算 Hessian 矩阵, 且收敛速度接近牛顿法, 计算开销远小于牛顿法.

变尺度法的核心思想为: 不断调整目标函数的度量方式, 改变梯度方向的尺度或几何结构, 以更快地逼近最优解.

11

使用含有动量 (momentum=0.9) 的随机梯度优化器法求解公式 (4) , 得到最优解 $\boldsymbol{x}^* = (0,0)^{\mathrm{T}}$, 最优值 $f(\boldsymbol{x}^*) = 0$.

12

观察 GDN[1]2中 Embedding 层的参数分布 (如图 1 所示), 参数基本分布在一个区域, 因此满足低维特性.

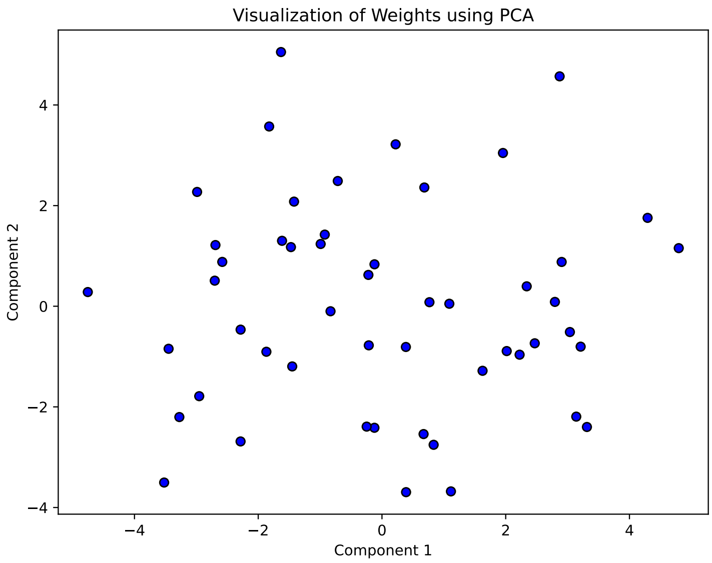

图 1　使用 PCA 对 GDN 的 Embedding 层降维后的结果

2本文使用复现的 GDN 模型, 代码见https://github.com/ikaroinory/GDN.

13

1. Crested Porcupine Optimizer (CPO)[2].

   该算法受到豪猪的防御行为的启发, 旨在尽可能多地解决这些缺点, 以准确解决一些优化问题, 特别是大规模优化问题. 现有的大多数元启发式算法都试图模拟动物的攻击行为, 但 CPO 首次尝试模拟豪猪的防御行为, 以提出一种具有不同特征的新型元启发式算法, 该算法可能在大量优化问题中取得优异的结果. 豪猪通过四种不同的方式自卫: 视觉, 听觉, 嗅觉和物理攻击. 在 CPO 中, 声音和视觉防御机制代表探索行为, 而其他机制代表利用行为. 此外, 一种名为循环种群减少技术的新颖策略已与所提出的优化器相结合, 以模拟假设: 只有部分豪猪在感知到威胁时才会激活防御机制. 该策略在保持种群多样性的同时, 加快了向近最优解的收敛速度.

2. Puma Optimizer (PO)[3].

   文章提出了一种新的优化算法——美洲狮优化器 (PO), 其灵感源自美洲狮的智慧和生命. 该算法在探索和开发的每个阶段都提出了独特而强大的机制, 从而提高了算法在各种优化问题中的性能. 此外, 文章还提出了一种新型智能机制, 即一种用于相变的超启发式算法. 利用这种机制, PO 算法可以在优化操作期间执行相变操作, 并平衡两个阶段. 每个阶段都会根据问题的性质自动调整.

14

Krylov 子空间方法是一类用于求解大型稀疏线性方程组的迭代算法, 其基本思想是利用矩阵和残差向量生成的 Krylov 子空间, 在该子空间内寻找近似解. 对于线性系统 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 从初始残差 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_0$ 出发, 通过不断地计算 $\boldsymbol{A}$ 与前一残差的乘积, 生成 Krylov 子空间

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)=\operatorname{span}\{\boldsymbol{r}_0,\boldsymbol{A}\boldsymbol{r}_0,\boldsymbol{A}^{\boldsymbol{2}}\boldsymbol{r}_0,\cdots,\boldsymbol{A}^{\boldsymbol{k-1}}\boldsymbol{r}_0\},$$

然后在该子空间中寻找使残差最小的近似解.

子空间投影方法在数值线性代数中有广泛应用, 典型的包括:

1. 求解线性方程组. 如共轭梯度法 (CG) 和广义最小残差法 (GMRES), 通过在 Krylov 子空间中投影, 寻找近似解.

2. 特征值问题. Lanczos 方法和 Arnoldi 方法通过在 Krylov 子空间中投影, 逼近矩阵的特征值和特征向量.

3. 预处理技术. 在迭代求解过程中, 通过子空间投影构造预处理矩阵, 加速收敛.

以 GMRES 方法为例, 考虑线性系统 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, 其中 $\boldsymbol{A}$ 为非对称矩阵. GMRES 通过构造 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{r}_0)$, 在其中寻找使残差范数最小的近似解. 具体步骤如下:

1. Arnoldi 正交化. 生成正交基 $(\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k)$, 将 $\boldsymbol{A}$ 在该基下投影为上 Hessenberg 矩阵 $\boldsymbol{H}_k$.

2. 最小化残差. 求解最小二乘问题

$$\min_{\boldsymbol{y}} |\boldsymbol{H}_k \boldsymbol{y} - \beta \boldsymbol{e}_1|,$$

得到 $\boldsymbol{y}_k$, 进而计算近似解 $\boldsymbol{x}_k = \boldsymbol{V}_k \boldsymbol{y}_k$, 其中 $\boldsymbol{V}_k = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k)$.

在实际应用中, GMRES 方法通常表现出良好的收敛性, 特别是对于非对称或非正定矩阵.

GMRES 方法与共轭梯度法的比较:

1. 适用范围: 共轭梯度法适用于对称正定矩阵, 而 GMRES 可处理任意矩阵, 特别是非对称或非正定矩阵.

2. 存储需求: GMRES 需要存储 Krylov 子空间的全部基向量, 存储需求随迭代次数增加, 可能导致内存问题. 共轭梯度法仅需存储少量向量, 存储需求固定.

3. 收敛性: 对于对称正定矩阵, CG 方法通常收敛速度快于 GMRES. 然而, 对于非对称或非正定矩阵, GMRES 可能是更合适的选择.

15

共轭函数定义如下:

定义 1. 对于函数 $f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\}$, 其共轭函数 f^* 定义为

$$f^*(\boldsymbol{y}) = \sup_{x \in \mathbb{R}^n} \{\langle \boldsymbol{y}, \boldsymbol{x} \rangle - f(\boldsymbol{x})\},$$

其中 $\langle \cdot, \cdot \rangle$ 为 Euclid 内积.

求解共轭函数一般有如下步骤:

1. **构造优化问题**. 由定义 1 可知, 求解共轭函数是一个关于 $\boldsymbol{x}$ 的优化问题.

2. **求导**. 若 $f(x)$ 可微, 找到使得梯度为零的 x, 即解方程

$$\nabla_x (\langle \boldsymbol{y}, \boldsymbol{x} \rangle - f(\boldsymbol{x})) = \boldsymbol{0}.$$

3. **凸性与 Legendre 变换**. 若 $f(\boldsymbol{x})$ 是严格凸函数, 求共轭函数的过程等价于 Legendre 变换, 即

$$f^*(\boldsymbol{y}) = \langle \boldsymbol{y}, \boldsymbol{x}^*(\boldsymbol{y}) \rangle - f(\boldsymbol{x}^*(\boldsymbol{y})),$$

其中 $\boldsymbol{x}^*(\boldsymbol{y})$ 是由 $\boldsymbol{y} = \nabla f(\boldsymbol{x})$ 反解得到的.

共轭函数在优化问题中扮演了关键角色, 尤其是在 Lagrange 对偶和 Fenchel 对偶中.

1. **Lagrange 对偶**.

   设优化问题为

$$\begin{cases} \min\limits_{\boldsymbol{x}} & f(\boldsymbol{x}) \\ \text{s.t.} & g_i(\boldsymbol{x}) \leq 0, \\ & h_j(\boldsymbol{x}) = 0, \end{cases}$$

   则其对偶问题可通过 Lagrange 函数

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_i \lambda_i g_i(\boldsymbol{x}) + \sum_j \mu_j h_j(\boldsymbol{x})$$

   用共轭函数表示目标函数, 从而推导对偶问题.

2. **Fenchel 对偶**.

   给定函数 $f(\boldsymbol{x})$ 和 $g(\boldsymbol{y})$, 考虑优化问题

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) + g(\boldsymbol{A}\boldsymbol{x}),$$

   其 Fenchel 对偶问题为

$$\max_{\boldsymbol{y}} f^*(-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}) - g^*(\boldsymbol{y}),$$

   其中 f^* 和 g^* 分别为 f 和 g 的共轭函数.

共轭函数在优化理论中起到了桥梁作用, 将原始问题映射到其对偶问题, 从而提供了一种有效的分析工具。利用共轭函数, 可以深入研究优化问题的对偶结构, 如强对偶性条件, 对偶间隙以及对偶问题的解构特性. 在凸优化领域, 共轭函数广泛应用于凸对偶理论, Fenchel 对偶以及拉格朗日对偶方法, 为求解复杂优化问题提供了重要的理论支撑.

参考文献

[1] DENG A, HOOI B. Graph neural network-based anomaly detection in multivariate time series[C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 35. 2021: 4027-4035.

[2] ABDEL-BASSET M, MOHAMED R, ABOUHAWWASH M. Crested porcupine optimizer: A new nature-inspired metaheuristic[J]. Knowledge-Based Systems, 2024, 284: 111257.

[3] LIAN J, HUI G, MA L, et al. Parrot optimizer: Algorithm and applications to medical problems[J]. Computers in Biology and Medicine, 2024, 172: 108064.