6.2 坐标下降法

SMaLL

[1] 中国石油大学（华东）
SMaLL 课题组：梁锡军
small.sem.upc.edu.cn
liangxijunsd@163.com

2023

3.

坐标下降法

1. 算法简介
2. 收敛性
3. 示例
4. 算法对比
5. 应用举例

算法描述

坐标下降法是一种非梯度优化算法，沿着坐标方向进行迭代搜索，在保持所有其他变量不变的情况下最小化目标函数，然后以类似的迭代方式更新其他变量的方法。

1. 在整个过程中循环使用不同的坐标方向。
2. 在每次迭代中，在当前点处沿一个坐标方向进行一维搜索以求得一个函数的局部极小值。
3. 为了加速收敛，可以采用一个适当的坐标系，例如通过主成分分析获得一个坐标间尽可能不相互关联的新坐标系。

算法描述

考虑使用坐标下降法求解无约束优化问题:

$$\min_x f(x)$$

其中 f:$R^d \to R$ 为连续可微函数

坐标下降法迭代格式如下:

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla_{i_k} f(w_k) e_{i_k}, \nabla_{i_k} f(w_k) := \frac{\partial f}{\partial w^{i_k}}(w_k)$$

其中, w^{i_k} 表示向量 w 的第 i_k 个元素, e^{i_k} 为第 i_k 个坐标向量, $i_k \in \{1, \cdots, d\}$

迭代向量 w_{k+1} 和 w_k 只在第 i_k 个元素存在差异。

算法描述

关于 i_k 的选择，至少有三种不同的方式:

- 遍历指标集 $\{1, \cdots, d\}$;
- 循环指标 $\{1, \cdots, d\}$ 的随机排序 (在每一组 d 步之后, 指标重新排序);
- 简单地在每次迭代中随机选择一个索引;
  随机化坐标下降算法 (后两种策略) 比循环方法 (第一种策略) 具有更好的理论特性，因为它们不太可能连续选择较差的坐标。然而，这种随机算法在应用中是否更有效仍然是未解决的问题。

坐标下降法

算法收敛性

Theorem [定理]　假设目标函数 f:$R^d \rightarrow R$ 是连续可微的强凸函数，常数 c>0，函数 f 的梯度沿坐标方向 lipschitz 连续，lipschitz 常数为 $L_1, ..., L_d$。另外，假设 $\alpha_\kappa = \frac{1}{\hat{L}}$，对所有的 $k \in \mathbf{N}$，i_k 从 1,⋯,d 中随机选取。对于所有 $k \in \mathbf{N}$ ，使用迭代公式得到

$$\mathbb{E}[f(w_{k+1})] - f_* \leqslant \left(1 - \frac{c}{d\hat{L}}\right)^k (f(w_1) - f_*)$$

算法收敛性

当目标函数 $f(x)$ 是光滑且凸时，必有 $f(x^0) \geq f(x^1) \geq f(x^2) \geq \ldots$

证明：

当 $k=0$ 时，对于的 $f(x)$ 的值为 $f(x^0)=f(x_1^0,x_2^0,\ldots,x_n^0)$

由于 $x_1^1=\arg\min f(x_1,x_2^0,\ldots,x_n^0)$

所以 $f(x_1^1,x_2^0,\ldots,x_n^0)\leq f(x_1^0,x_2^0,\ldots,x_n^0)=f(x^0)$

以此类推

$f(x_1^1,x_2^1,\ldots,x_n^0)\leq f(x_1^1,x_2^0,\ldots,x_n^0)\leq f(x_1^0,x_2^0,\ldots,x_n^0)=f(x^0)$

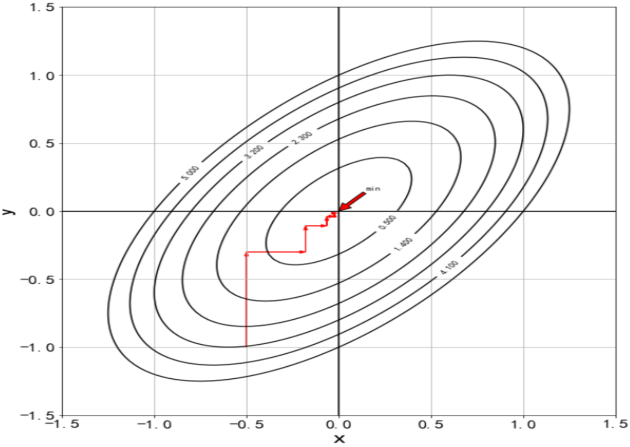
$f(x^1)=f(x_1^1,x_2^1,\ldots,x_n^1)\leq \ldots f(x_1^1,x_2^1,\ldots,x_n^0)\leq f(x_1^1,x_2^0,\ldots,x_n^0)\leq f(x_1^0,x_2^0,\ldots,x_n^0)=f(x^0)$

同理可得 $f(x^2)\leq f(x^1)\leq f(x^0)$, 命题得证

坐标下降法

示例

直观地了解一下坐标下降法。给定二元函数 $f(x, y) = 5x^2 - 6xy + 5y^2$



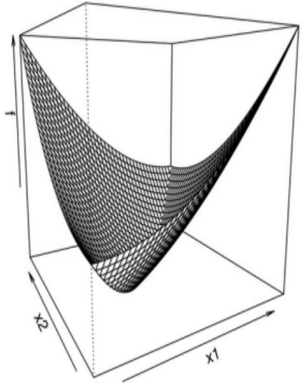
起始点 (-0.5,-1.0), 此时 f=3.25。现在我们固定 x，将 f 看成关于 y 的一元二次方程，并求当 f 最小时 y 的值：

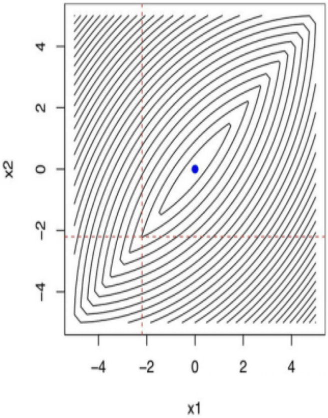
$$
\begin{aligned}
&f(x, y) = 5x^2 - 6xy + 5y^2 \\
&\ f(y \mid x = -0.5) = 5 * (-0.5)^2 - 6 * (-0.5) * y + 5 * y^2 + 1 \\
&\ = 5y^2 + 3y + 1.25 \\
&\ f'_{(y|x=-0.5)} = 10y + 3 = 0 \\
&y = -0.3
\end{aligned}
$$

所以现在自变量的值更新为 (-0.5,-0.3)，有 f=0.8。

思考

如果函数没有光滑性，但有凸性，收敛性是否还能成立？答案是否定的。通过图示，考虑二维的简单情况，我们就能看出这一点。





示例

难道光滑性是一个必要条件？事实上，对于一种特殊的情况

$$f(x) = g(x) + \sum_{i=1}^{n} h_i(x_i)$$

即 f 可以拆分成两个函数 g 和 h，其中 g 是凸函数且光滑，h 可以继续拆分成 $h_1, \ldots, h_n$(n 是 x 的维数)，并且每一个 h_i 都是凸函数。在这种情况下，$f(x)$ 仍然可以通过坐标下降法求解其最小值。

示例

根据条件，我们有

$$f(y)-f(x)\geq\nabla g(x)^T(y-x)+\sum_{i=1}^n\left[h_i\left(y_i\right)-h_i\left(x_i\right)\right]$$
$$=\sum_{i=1}^n\left[\nabla_i g(x)\left(y_i-x_i\right)+h_i\left(y_i\right)-h_i\left(x_i\right)\right]\geq 0$$

坐标下降法

计算复杂度

对于坐标下降法，如果数据的维度很大，可能需要很多次迭代。但如果运算都得当，实际是不需要的。我们不妨用线性回归的例子来对比坐标下降法和梯度下降法：

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2$$

计算复杂度

考虑坐标下降法。在每一个维度都取到极小值，要求 $\nabla_i f(\beta) = 0$，即

$$X_i^T(X\beta - y) = 0$$

由于需要知道 β_i 的更新公式，可以先把 β_i 拆出来，也就得到

$$X\beta = X_i\beta_i + X_{-i}\beta_{-i}$$

进一步得到 β_i 第 $k+1$ 次迭代的更新公式

$$\beta_i^{k+1} = \frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i} = \frac{X^T r}{\|X_i\|_2^2} + \beta_i^k$$

计算复杂度

此外，梯度下降法的更新公式是

$$\beta^{k+1} = \beta^k + tX^T(y - X\beta)$$

对于梯度下降法，我们可以先计算 $X\beta$，再计算 $y - X\beta$，最后计算 $X^T(y - X\beta)$，计算复杂度为 $O(np)$（算出来大概是 $4np$）。

对于坐标下降法，需要先更新 r，再计算 $X_i^T r$，是一个 $O(n)$ 的复杂度，而且差不多也是 $4n$ 的浮点数运算次数。对于 p 个维度，就是 $4np$ 次运算。可以看出它和梯度下降法的运算次数是一样的，

SMaLL Statistical Machine Learning Lyceum

坐标下降法

1. 算法简介

2. 收敛性

3. 示例

4. 算法对比

5. 应用举例

Lasso 回归

坐标下降法可用于求解 Lasso 回归。Lasso 相当于带有 L1 正则化项的线性回归。其目标函数如下：

$$RSS(w) + \lambda\|w\|_1 = \sum_{i=0}^{N}\left(y_i - \sum_{j=0}^{D} w_j h_j(x_i)\right)^2 + \lambda\sum_{j=0}^{D}|w_j|$$

利用坐标下降法求解可得

$$\hat{W}_j = \begin{cases} \left(p_j + \frac{\lambda}{2}\right)/z_j & , p_j < -\frac{\lambda}{2} \\ 0 & , p_j \in \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right] \\ \left(p_j - \frac{\lambda}{2}\right)/z_j & , p_j > \frac{\lambda}{2} \end{cases}$$

其中 $p_j = \sum_{i=1}^{N} h_j(x_i)\left(y_i - \sum_{k\neq j}^{D} w_k h_k(x_i)\right) \quad , \quad z_j = \sum_{i=1}^{N} h_j^2(x_i)$

Lasso 回归

利用坐标下降法求解分两步进行，先对 RSS 部分求偏导

$$\begin{aligned}
\frac{\partial RSS(w)}{\partial w_j} &= -2\sum_{i=1}^{N} h_j\left(x_i\right)\left(y_i - \sum_{j=0}^{D} w_j h_j\left(x_i\right)\right) \\
&= -2\sum_{i-1}^{N} h_j(x_i)\left(y_i - \sum_{k\neq j}^{D} w_k h_k\left(x_i\right) - W_i h_j\left(x_i\right)\right) \\
&= -2\sum_{i=1}^{N} h_j\left(x_i\right)\left(y_i - \sum_{k\neq j}^{D} w_k h_k\left(x_i\right)\right) + 2\,W_j\sum_{i=1}^{N} h_j^2\left(x_i\right)
\end{aligned}$$

记

$$p_j = \sum_{i=1}^{N} h_j\left(x_i\right)\left(y_i - \sum_{k\neq j}^{D} w_k h_k\left(x_i\right)\right) \qquad z_j = \sum_{i=1}^{N} h_j^2\left(x_i\right)$$

Lasso 回归

$$\therefore \frac{\partial RSS(w)}{\partial w_j} = -2p_j + 2w_j z_j$$

利用次梯度方法求解不可导部分有

$$\lambda \partial w_j |w_j| = \begin{cases} -\lambda & , w_j < 0 \\ [-\lambda, \lambda] & , w_j = 0 \\ \lambda & , w_j > 0 \end{cases}$$

Lasso 回归

整体偏导数为:

$$2z_jw_j - 2p_j + \left[\begin{array}{cc} -\lambda & ,\omega_j < 0 \\ [-\lambda, \lambda] & ,w_j = 0 \\ \lambda & ,w_j > 0 \end{array}\right.$$

令其等于 0，有

$$\hat{w}_j = \begin{cases} \left(p_j + \frac{\lambda}{2}\right)/z_j & ,p_j < -\frac{\lambda}{2} \\ 0 & ,p_j \in \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right] \\ \left(p_j - \frac{\lambda}{2}\right)/z_j & ,p_j > \frac{\lambda}{2} \end{cases}$$