优化和搜索

高　阳, 李文斌

http://cs.nju.edu.cn/rl, 2021.4.1

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

最小二乘优化

搜索

利用和探索

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

最小二乘优化

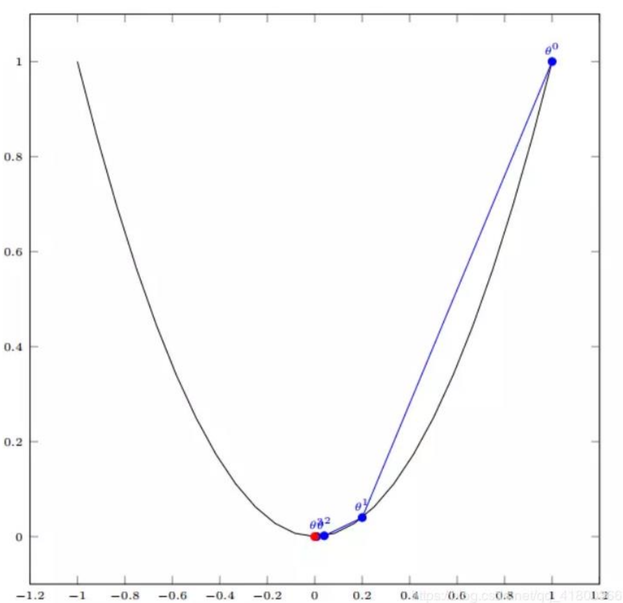
搜索

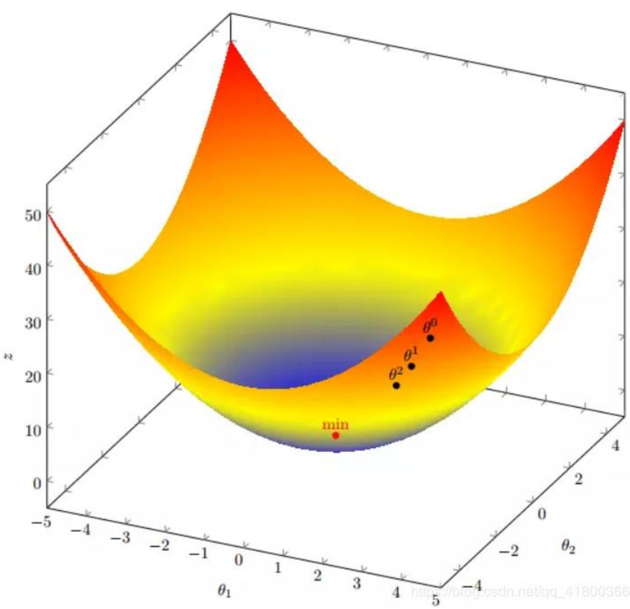
利用和探索

数学基础

✓ 导数和偏导数：一元函数和多元函数$\frac{\partial f}{\partial \mathbf{x}_i}$

✓ (一阶)梯度：$\nabla f(\mathbf{x}) = (\frac{\partial f}{\partial \mathbf{x}_1}, \frac{\partial f}{\partial \mathbf{x}_2}, \ldots, \frac{\partial f}{\partial \mathbf{x}_n})$，是一个向量(值，方向)





数学基础

✓ 泰勒展开(Taylor expansion)：

$$f(\mathbf{x}) \approx f(\mathbf{x_0}) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\mathbf{H}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \cdots$$

✓ 雅可比阵(Jacobian), 海森阵(Hessian)：

$$\mathbf{J}(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \left(\frac{\partial f_1(\mathbf{x})}{\partial x_1}, \frac{\partial f_1(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f_1(\mathbf{x})}{\partial x_n}\right)$$

$$\mathbf{H}(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}_i}\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_j} = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ & \cdots & & \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x})}{\partial x_n^2} \end{pmatrix}$$

数学基础

- ✓ 正定阵(positive definite matrix) $X^T A X > 0$
- ✓ 对称正定阵
- ✓ 半正定矩阵(positive semidefinite matrix)
- ✓ Hermite正定矩阵(复数共轭阵，实对称阵)
- ✓ 奇异值分解SVD

大纲

- 数学基础
- 梯度下降法
- 牛顿法
- 共轭梯度法
- 最小二乘优化
- 搜索
- 利用和探索

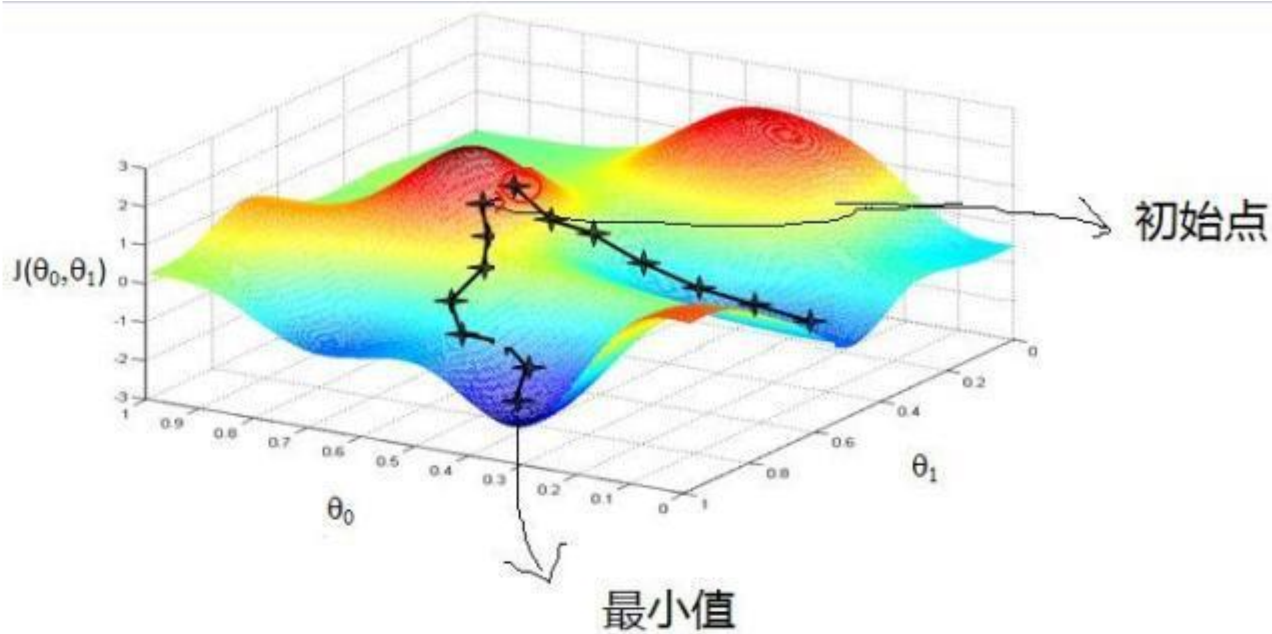
梯度下降法

- 原理(迭代法)
  - ✓ 解目标函数的最小值 $f(\mathbf{x}) = f(x_1, \cdots, x_n)$
  - ✓ 从初始点开始 $\mathbf{x}^{(0)} = (x_1^{(0)}, \cdots, x_n^{(0)})$
  - ✓ 基于学习率构建迭代过程 $\eta > 0$

$$\left.\begin{aligned} x_1^{(i+1)} &= x_1^{(i)} - \eta \cdot \frac{\partial f}{\partial x_1}(\mathbf{x}^{(i)}), \\ &\cdots \\ x_n^{(i+1)} &= x_n^{(i)} - \eta \cdot \frac{\partial f}{\partial x_n}(\mathbf{x}^{(i)}). \end{aligned}\right\} \quad \boldsymbol{X^{(i+1)} = X^{(i)} - \eta \nabla f(X)}$$

  - ✓ 直至迭代结束

梯度下降法



- 研究点
  - ✓ 目标函数定义、初始点选择、更新公式、结束机制

梯度下降法的发展

- 基于学习率的改进机制(learning rate schedulers)
  - ✓ 将学习率乘以一个常数或者时间因子 想象一下：当你骑自行车下陡坡时；
  - ✓ 使学习率能适应梯度。当梯度值很大时，希望更新小
  - ✓ 因此，将学习率除以梯度值。为保证为正，所以使用平方开根号 root mean square
- 基于二阶梯度的改进机制(gradient descent optimizer)
  - ✓ 将学习率乘以一个梯度的因子来进行调整
  - ✓ 利用到之前的梯度来更好的指导梯度下降
  - ✓ 更近的梯度值获取更高的权重，因而实现了exponential moving average

梯度下降法的发展

Optimiser	Year	Learning Rate	Gradient
Momentum	1964		✓
AdaGrad	2011	✓	
RMSprop	2012	✓	
Adadelta	2012	✓	
Nesterov	2013		✓
Adam	2014	✓	✓
AdaMax	2015	✓	✓
Nadam	2015	✓	✓
AMSGrad	2018	✓	✓

代表性梯度下降法

冲量法 (Momentum)

- ✓ 将当本次梯度下降方向与上次更新量的方向相同时，上次的更新量能够对本次的搜索起到一个正向加速的作用；反之减速。

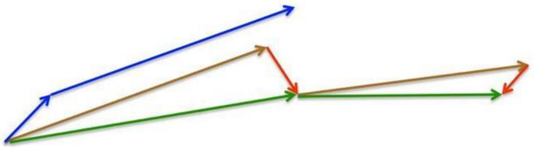
$$W_{t+1} = W_t - \alpha V_t$$

$$V_t = \beta V_{t-1} + (1-\beta)\frac{\partial L}{\partial W_t}$$

exponential moving average

NAG(Nesterov Accelerated Gradient)

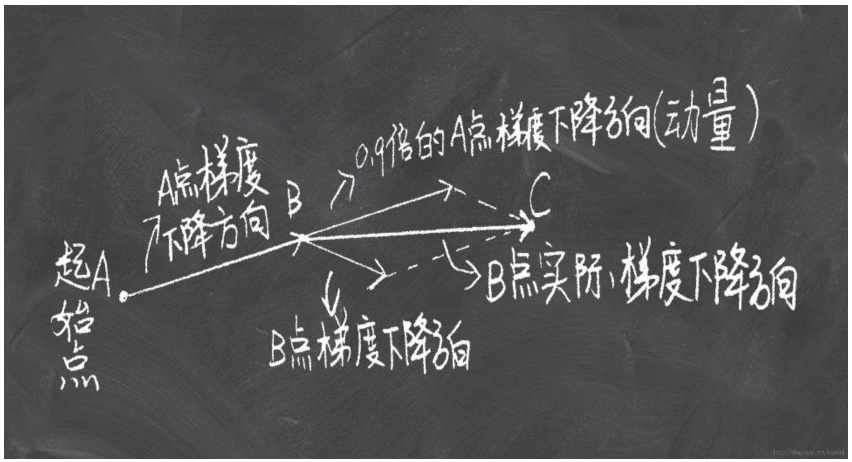
- ✓ 不仅仅把SGD梯度下降以前的方向考虑，还将Momentum梯度变化的幅度也考虑了进来。



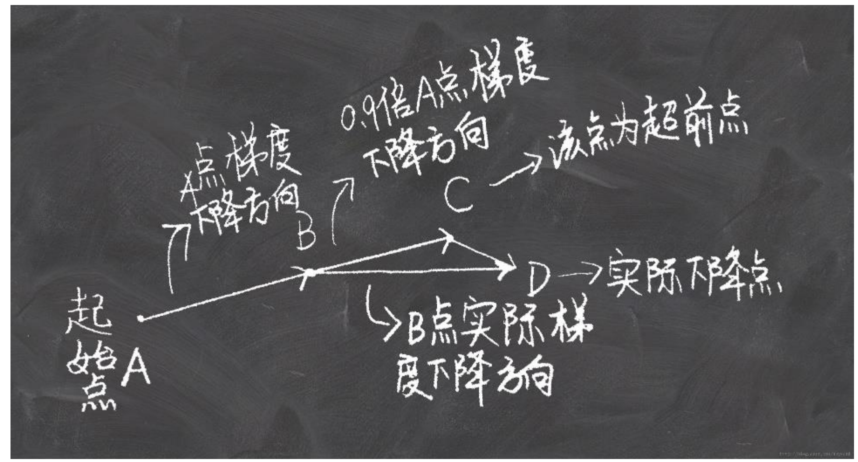
$$V_t = \beta V_{t-1} + (1-\beta)\frac{\partial L}{\partial W^*}$$

使用先前的V更新权重w，得到w*；使用w*进行前向传播；获取投影梯度 ∂L/∂w*；相应计算V和W。

代表性梯度下降法



0.9倍的A点梯度下降方向(动量)
A点梯度下降方向
B
C
起始点A
B点实际梯度下降方向
B点梯度下降方向



0.9倍A点梯度下降方向
该点为超前点
A点梯度下降方向
B
C
D
实际下降点
起始点A
B点实际梯度下降方向

代表性梯度下降法

◻ 自适应梯度法 (AdaGrad)

✓ 通过将learning rate除以S的平方根进行更新。S指的是历史和当前的梯度的平方的累加。

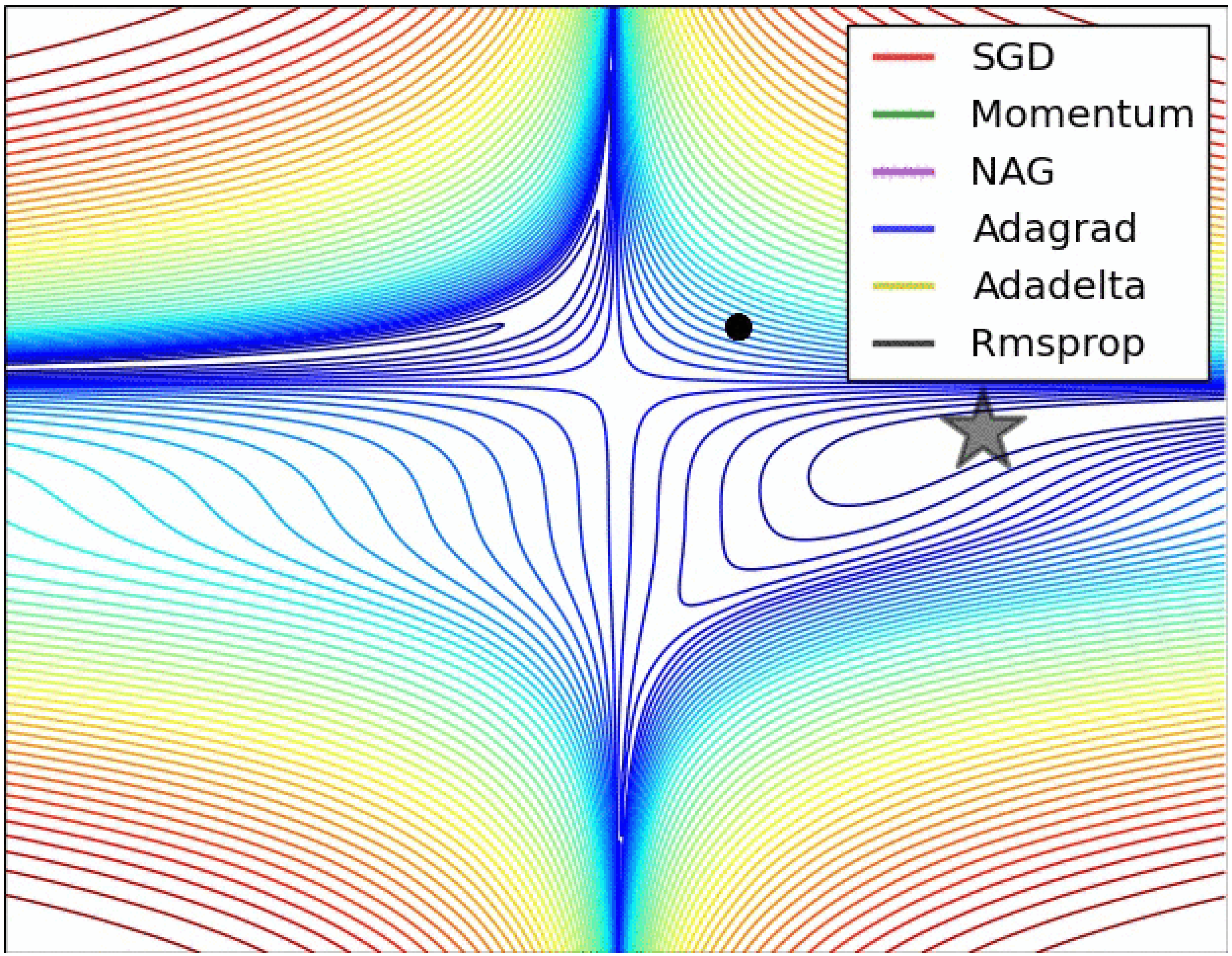
$$W_{t+1} = W_t - \frac{\alpha}{\sqrt{S_t + \varepsilon}} \frac{\partial L}{\partial W_t}$$

$$S_t = S_{t-1} + \left[\frac{\partial L}{\partial W_t}\right]^2$$

root mean square

学习率适应梯度

代表性梯度下降法的图示



梯度下降法

❑缺点

- ✓ 靠近极小值时收敛速度减慢；
- ✓ 直线搜索时可能会产生一些问题；
- ✓ 可能会“之字形”地下降。

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

最小二乘优化

搜索

利用和探索

泰勒展开

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x} - \mathbf{x}_0) + \cdots$$

忽略三阶导以上，直接对上式求导(解析法)

那么第k步的牛顿方向为

$$\mathbf{p}_k = (\mathbf{x} - \mathbf{x}_k) = -(\nabla f^2(\mathbf{x}_k))^{-1} \nabla f(\mathbf{x}_k) = -\frac{\nabla f(\mathbf{x}_k)}{\nabla f^2(\mathbf{x}_k)}$$

$$\mathbf{p}_k = (\mathbf{x} - \mathbf{x}_k) = \boldsymbol{H}^{-1}\boldsymbol{G}$$

需要计算海森矩阵的逆，计算复杂度为$O(N^3)$

牛顿法原理

- 牛顿法(Newton's method)
  - ✓ 迭代点处的一阶导数(Jacobian梯度)和二阶导数(Hessen矩阵)对目标函数进行二次函数近似;
  - ✓ 然后把二次模型的极小点作为新的迭代点，并不断重复这一过程;
  - ✓ 直至求得满足精度的近似极小值。

牛顿法算法流程

■牛顿法

- ✓ 给定初始值$\mathbf{x}_0$和精度阈值ε，并令$k = 0$
- ✓ 计算$\|g_k\|$和$\|H_k\|$
- ✓ 如果$\|g_k\| \le \varepsilon$，终止迭代
- ✓ 否则$p_k = -{H_k}^{-1} g_k$
- ✓ 得到新的迭代点$\mathbf{x}_{\mathrm{k+1}} = \mathbf{x}_{\mathrm{k}} + p_k$ ➡ 没有步长
- ✓ 令$k = k + 1$，转到第二步

牛顿法的改进

- 阻尼牛顿法(damped-newton method)
  - ✓ 对非二次型函数，牛顿方向不区分函数上升还是下降，导致无法使函数稳定地下降。
  - ✓ 通过控制步长调节。对当前步长再做极小化优化。
- 拟牛顿法(Quasi-Newton method)
  - ✓ 计算H阵以及求逆计算复杂度太大
  - ✓ 利用一阶导数(J阵)拟合H阵，构造一个替代阵

阻尼牛顿法

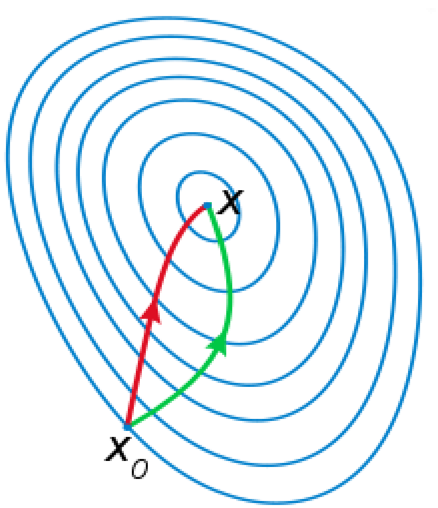
$$\boldsymbol{\lambda}_{\mathrm{k}} = argmin_{\boldsymbol{\lambda}} f(x_k + \boldsymbol{\lambda}_{\mathrm{k}} p_{\mathrm{k}})$$

■阻尼牛顿法

- ✓ 给定初始值$\mathbf{x}_0$和精度阈值ε，并令$k = 0$
- ✓ 计算$\|g_k\|$和$\|H_k\|$
- ✓ 如果$\|g_k\| \leq \varepsilon$，终止迭代; 否则$p_k = -{H_k}^{-1} g_k$
- ✓ 计算$\boldsymbol{\lambda}_{\mathrm{k}}$，得到新的迭代点$\mathbf{x}_{\mathrm{k+1}} = \mathbf{x}_{\mathrm{k}} + \boldsymbol{\lambda}_{\mathrm{k}} p_k$
- ✓ 令$k = k + 1$，转到第二步

牛顿法优点

- 牛顿法：用二次曲面去拟合当前点的局部曲面
- 梯度下降法：是一次平面去拟合当前点的局部曲面
- 通常情况下：二次曲面的拟合优于平面，所以牛顿法选择的下降路径会更符合真实的最优下降路径。



✓优点：二阶收敛，收敛速度快；

牛顿法缺点

- 计算复杂度
  - 每一步都需要求解目标函数的H矩阵的逆矩阵，计算比较复杂。
- 目标函数可导性
  - 必须一阶二阶可导，H阵须正定。

拟牛顿法

- 拟牛顿法(Quasi-Newton method)

  ✓ 改善牛顿法每次需要求解复杂的Hessian矩阵的逆矩阵的缺陷，它使用正定矩阵来近似Hessian矩阵的逆，从而简化了运算的复杂度。

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\mathbf{H}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \cdots$$

$$\Downarrow$$

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\mathbf{U}(\mathbf{x}-\mathbf{x}_0)$$

拟牛顿条件

$$\nabla f(x_{k+1}) - \nabla f(x_k) = H \cdot (x_{k+1} - x_k)$$

$$x_{k+1} - x_k = H^{-1}\left(\nabla f(x_{k+1}) - \nabla f(x_k)\right)$$

$$x_{k+1} - x_k = U_{k+1}\left(\nabla f(x_{k+1}) - \nabla f(x_k)\right)$$

$$\boxed{\Delta x_k = \boxed{U_{k+1}} \Delta g_k}$$

拟牛顿条件

不同的拟牛顿法(DFP,BFGS,Broyden)，采用不同的计算方式

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

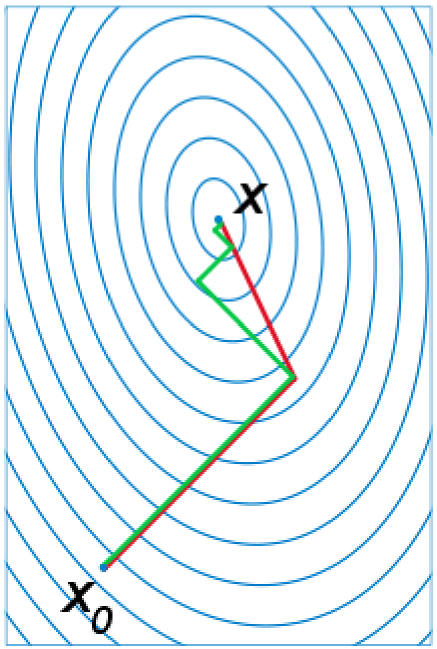
最小二乘优化

搜索

利用和探索

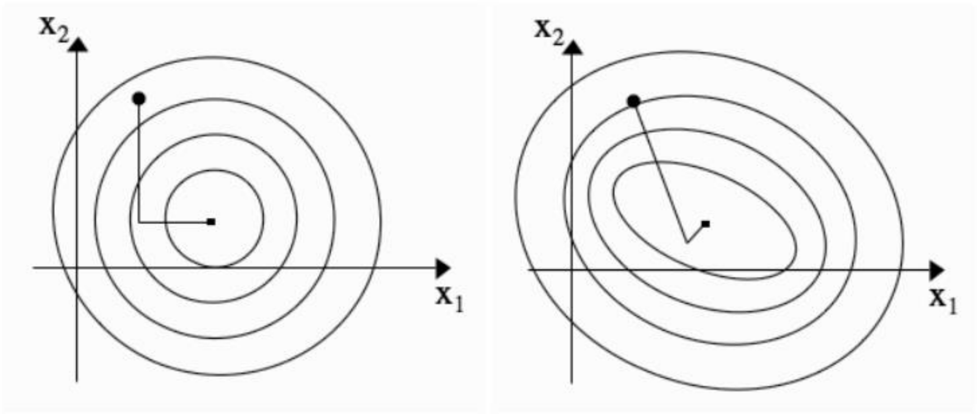
共轭梯度法

- 共轭梯度法(Conjugate Gradient)：
  - ✓ 介于最速下降法与牛顿法之间的一个方法
  - ✓ 仅需利用一阶导数信息
  - ✓ 克服了最速下降法收敛慢的缺点，又避免了牛顿法需要存储和计算Hesse矩阵并求逆的缺点。



共轭梯度的数学定义

- 共轭(conjugate)的定义
  - ✓ 两个向量$\mathbf{p}_i$, $\mathbf{p}_j$是共轭的, 则存在对半正定矩阵$\mathbf{A}$
  - ✓ $\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j = 0$
  - ✓ $\mathbf{p}_i$, $\mathbf{p}_j$并不直接正交(又称为**A**-正交)



共轭梯度法流程

- ✓ 计算一个初始的搜索方向$\mathbf{p}_0$(利用梯度下降法)
- ✓ 找一个使得目标函数$f(\mathrm{x}_i+\alpha_i\mathbf{p}_i)$最小的$\mathrm{\alpha_i}$
- ✓ 计算$\mathbf{x}_{i+1}=\mathbf{x}_i+\alpha_i\mathbf{p}_i$
- ✓ 确定下一个搜索方向$\mathbf{p}_{i+1}=-\nabla f(\mathbf{x}_{i+1})+\beta_{i+1}\mathbf{p}_i$
  - ✓ β_{i+1}由前一个迭代点梯度$\nabla f(\mathbf{x}_i)$，当前迭代点梯度$\nabla f(\mathbf{x}_{i+1})$共同决定
  - ✓ 迭代完所有共轭方向
- ✓ 如不满足终止条件，则回到第一步，继续确定新的第一个梯度方向

共轭梯度法要点

$$f(\mathbf{x}) \approx f(\mathbf{x_0}) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\mathbf{H}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x}-\mathbf{x}_0) + \cdots$$

$$\approx f(\mathbf{x_0}) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x}-\mathbf{x}_0)$$

- 某共轭方向上步长的确定
  - ✓ 假定在$\mathbf{x}_k$是局部线性的，则$\boldsymbol{Ax} = \boldsymbol{b}$

$$\alpha_i = \frac{\mathbf{p}_i^T\boldsymbol{b}}{\mathbf{p}_i^{\mathbf{T}}\mathbf{A}\mathbf{p}_i} = \frac{\mathbf{p}_i^T\big(-\nabla f(\mathbf{x}_{i-1})\big)}{\mathbf{p}_i^{\mathbf{T}}\mathbf{A}\mathbf{p}_i}$$

共轭梯度法要点

- 新共轭方向的确定

前一个迭代点梯度

$$\mathbf{p}_k = \mathbf{u}_k + \sum_{i=0}^{k-1} \beta_{k_i} \mathbf{p}_i$$

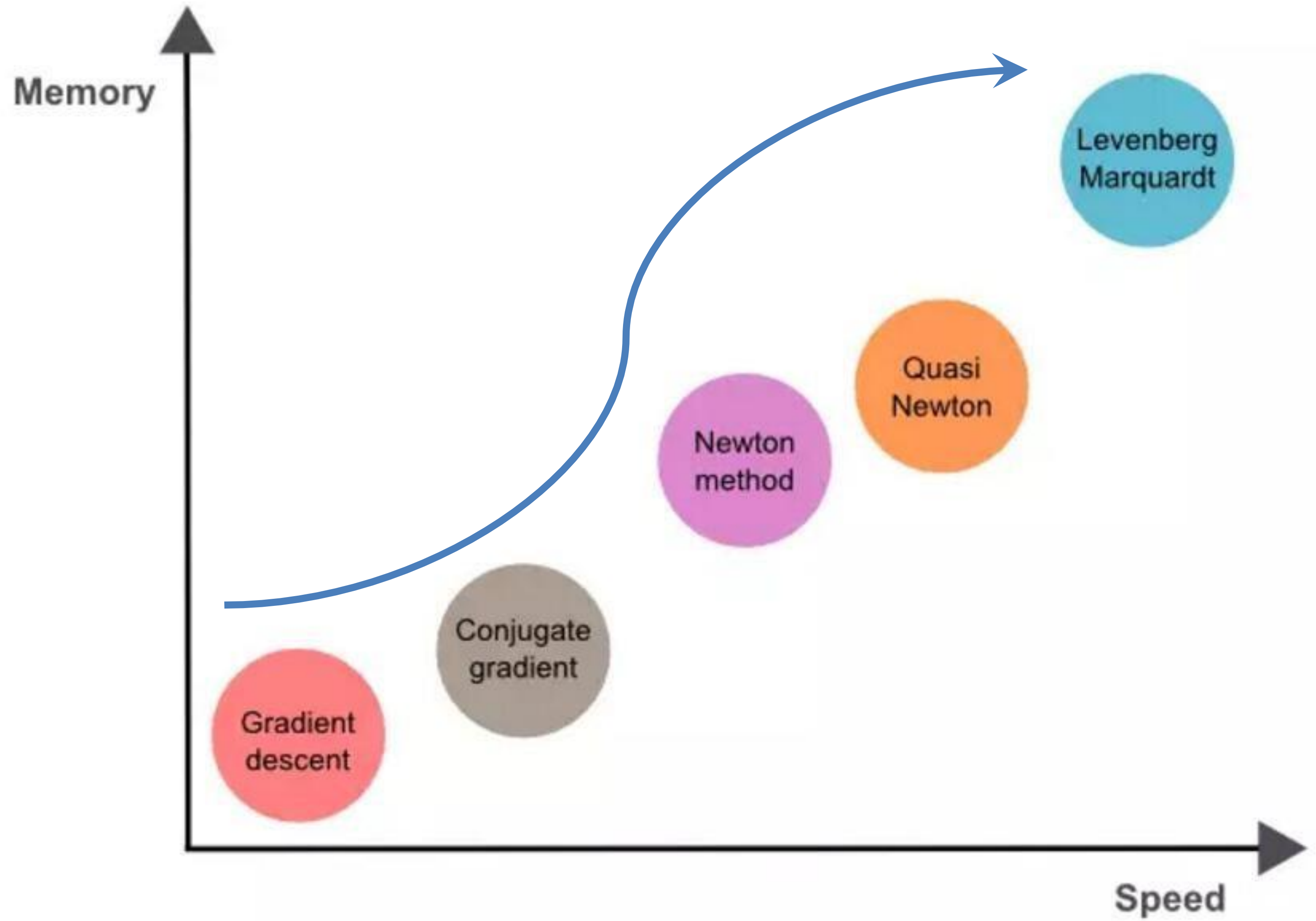
- Fletcher-Reeves公式

$$\beta_{i+1} = \frac{\nabla f(\mathbf{x}_{i+1})^T \nabla f(\mathbf{x}_{i+1})}{\nabla f(\mathbf{x}_i)^T \nabla f(\mathbf{x}_i)}$$

- Polak-Ribiere公式

$$\beta_{i+1} = \frac{\nabla f(\mathbf{x}_{i+1})^T \left(\nabla f(\mathbf{x}_{i+1}) - \nabla f(\mathbf{x}_i)\right)}{\nabla f(\mathbf{x}_i)^T \nabla f(\mathbf{x}_i)}$$

常见优化方法对比


Memory
Levenberg Marquardt
Quasi Newton
Newton method
Conjugate gradient
Gradient descent
Speed

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

最小二乘优化

搜索

利用和探索

最小二乘法

如损失函数中

的残差residual

目标函数

$$f(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(\mathbf{x}) = \frac{1}{2}\|\boxed{\mathbf{r}(\mathbf{x})}\|_2^2$$

r函数的J阵(非f的雅可比阵)

$$\mathbf{r}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \ldots, r_m(\mathbf{x}))^T$$

$$\mathbf{J}^T(\mathbf{x}) = \left\{\begin{matrix} \frac{\partial r_1}{\partial x_1} & \frac{\partial r_2}{\partial x_1} & \cdots & \frac{\partial r_m}{\partial x_1} \\ \frac{\partial r_1}{\partial x_2} & \frac{\partial r_2}{\partial x_2} & \cdots & \frac{\partial r_m}{\partial x_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial r_1}{\partial x_n} & \frac{\partial r_2}{\partial x_n} & \cdots & \frac{\partial r_m}{\partial x_n} \end{matrix}\right\} = \left[\frac{\partial r_j}{\partial x_i}\right]_{j=1,\ldots,m, i=1,\ldots,n}$$

$$\boxed{f(\mathbf{x}) \approx f(\mathbf{x}_0) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T\mathbf{H}\big(f(\mathbf{x})\big)|_{\mathrm{x}_0}(\mathbf{x}-\mathbf{x}_0) + \cdots}$$

最小二乘的梯度

$$\nabla f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T\mathbf{r}(\mathrm{x})$$

$$\nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x}) + \sum_{j=1}^{m} r_j(\mathbf{x})\nabla^2 r_j(\mathbf{x})$$

如果$\| \mathbf{r}(\mathbf{x}) \|$是线性的，即$f(\mathbf{x})$是二次，则$\mathbf{J}$阵为常数, 二阶导数为0

对$f(\mathbf{x})$按泰勒展开，求导并为0，则

$$\nabla f(\mathbf{x}) = \mathbf{J}^T(\mathbf{J}\mathbf{x} + \mathbf{r}) = 0$$

$$\mathbf{J}^T\mathbf{J}x = -\mathbf{J}^T\mathbf{r}(\mathbf{x}) \qquad\qquad \mathbf{x} = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathbf{r}$$

SVD分解法

SVD分解

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

用SVD解线性最小二乘

$$\mathbf{x} = [(\mathbf{U}\mathbf{S}\mathbf{V}^T)^T(\mathbf{U}\mathbf{S}\mathbf{V}^T)]^{-1}(\mathbf{U}\mathbf{S}\mathbf{V}^T)^T\mathbf{J}^T\mathbf{r}$$
$$= \mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{J}^T\mathbf{r}$$

对于线性最小二乘，用解析计算法直接求解

信赖域法

如果是非线性最小二乘，无法直接求解。如下例：

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

舍弃掉

回顾最小二乘的二阶梯度

$$\nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + \boxed{\sum_{j=1}^{m} r_j(\mathbf{x}) \nabla^2 r_j(\mathbf{x})}$$

每次迭代

$$\nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})$$

$$\boxed{f(\mathbf{x}) \approx f(\mathbf{x}_0) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \cdots}$$

目标式的物理和数学意义

信赖域法

修正非线性最小二乘每次迭代时的目标项

信赖域

$$\min_{\mathbf{p}} \frac{1}{2} \| \mathbf{J}_k \mathbf{p} + \mathrm{r_k} \|_2^2, \boxed{\| \mathbf{p} \| \leqslant \Delta_k}$$

- ✓ 从初始点开始，先假设一个可以信赖的最大位移Δ
- ✓ 以当前点为中心，以Δ为半径的区域内，通过寻找目标函数的一个近似函数(二次的)的最优点，来求解得到实际位移。
- ✓ 在得到实际位移后，计算目标函数值，如果目标函数值的下降满足特定条件，则说明该位移是可靠的，则继续按此规则迭代计算；
- ✓ 如果其不能使目标函数值的下降不满足特定条件，则应减小信赖域，再重新求解。

本质是假设非线性函数局部为二次函数

Levenberg-Marquardt算法

莱文贝格—马夸特方法Levenberg-Marquardt算法，最小化的$\mathbf{p}$满足

阻尼因子

$$(\mathbf{J}^T\mathbf{J} + \boxed{\nu\mathbf{I}})\mathbf{p} = -\mathbf{J}^T\mathbf{r}$$

✓ $\nu \geq 0$，则$(\mathbf{J}^T\mathbf{J} + \nu\mathbf{I})$正定，保证在梯度下降方向；

✓ ν取大值，退化成梯度下降法；

✓ ν取小值，接近于牛顿法。

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \mathbf{J}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T\mathbf{H}\big(f(\mathbf{x})\big)|_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0) + \cdots$$

Levenberg-Marquardt算法流程

□算法流程

- ✓ 用线性最小二乘法算出$(\mathbf{J}^T\mathbf{J} + \nu\mathbf{I})\mathbf{dx} = -\mathbf{J}^T\mathbf{r}$中的$\mathbf{dx}$
- ✓ 令$\mathbf{x}_{new} = \mathbf{x} + \mathbf{dx}$
- ✓ 实际减少=$\| f(\mathbf{x}) - f(\mathbf{x}_{new}) \|$
- ✓ 预测减少=$\nabla f^T(\mathbf{x}) \times \mathbf{x}_{new} - \mathbf{x}$
- ✓ $\rho =$ 实际减少/预测减少
- ✓ 如果$0 < \rho < 0.25$，接受$\mathbf{x}_{new}$
- ✓ 如果$\rho > 0.25$，接受$\mathbf{x}_{new}$，增加信赖域，减少ν
- ✓ 否则，拒绝$\mathbf{x}_{new}$。减少信赖域

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

最小二乘优化

搜索

利用和探索

旅行商问题

□旅行商问题(Traveling Salesman Problem, TSP)

- ✓ 任务：一个商品推销员要去若干个城市推销商品，该推销员从一个城市出发，需要经过所有城市后，回到出发地。应如何选择行进路线，以使总的行程最短。

- ✓ 在图论上，是在一个带权完全无向图中，找一个权值最小的Hamilton回路。
- ✓ 经典的组合优化难题，NP完全。

搜索方式

- 穷举法
  - ✓ 排列测试所有可能的路线并计算所有路线的距离
  - ✓ 计算的复杂度是$O(N!)$
  - ✓ 比指数函数上升还要快！组合爆炸！！！

搜索方式

☐ 贪婪搜索

- ✓ 在每一步都找局部最优解
- ✓ TSP：任意选择第一个城市，然后不断重复选择和当前所在城市最近并且没有访问过的城市，直到走完所有城市。
- ✓ 计算的复杂度是$O(NlogN)$
- ✓ 无法保证最优解！！

搜索方式

- 爬山法
  - ✓ 对当前解决方案的局部搜索，选择任一个选项来改善结果。
  - ✓ 本讲前面所有的迭代方法
- 三种失效的原因
  - ✓ 陷入局部极小
  - ✓ 各方同向的锅底
  - ✓ 平面区域

大纲

数学基础

梯度下降法

牛顿法

共轭梯度法

最小二乘优化

搜索

利用和探索

利用和探索

◘ 搜索的两种机制

- ✓ 探索(exploration)搜索空间，总是尝试新的方法，比如穷举法。
- ✓ 尝试局部变化利用(exploitation)当前已有的最好的方案，比如爬山法。

机器学习算法混合这两种实现机制

多臂老虎机

- 多臂老虎机(Multi-armed Bandit)
  - ✓ 任务：一个赌徒面前有N个老虎机,事先他不知道每台老虎机的真实盈利情况,他如何根据每次玩老虎机的结果来选择下次拉哪台或者是否停止赌博,来最大化自己的从头到尾的收益。

  - ✓ 特殊类型强化学习问题。

模拟退火算法

□ 原理(一种启发式算法)

✓ 利用的概率是玻尔兹曼分布(Softmax探索)

$$\exp((E_{before} - E_{after})/T)$$

✓ 余下概率采用探索/搜索(以跳出局部极小)

✓ 随时间变化，调节利用的概率

$$T(t+1) = cT(t) \quad 0 < c < 1$$

UCB算法

UCB算法

- ✓ 选择置信区间上界最大的臂
- ✓ 如果item置信区间很宽(被选次数很少，还不确定)，那么它会倾向于被多次选择，这个是算法冒风险的部分；
- ✓ 如果item置信区间很窄(备选次数很多，比较确定其好坏了)，那么均值大的倾向于被多次选择，这个是算法保守稳妥的部分；
- ✓ UCB是一种乐观的算法，选择置信区间上界排序，如果时悲观保守的做法，是选择置信区间下界排序。

置信度

☐ 置信度指数

- ✓ 依赖于收益
- ✓ 依赖于被选择的次数
- ✓ 选择指数最大的臂

$$I_i = \overline{U_i} + \sqrt{2\frac{logt}{n_i}}$$

思考和讨论

1. 梯度下降法的变体。
2. 牛顿法和梯度下降法的不同。
3. 最小二乘优化的特殊性。
4. 共轭梯度法原理。
5. 探索和利用。
6. MAB的机器学习算法。

谢 谢 ！