最小二乘问题

- 最小二乘问题的一般形式如下：

$$\min_{x\in\mathbb{R}^n} \quad \sum_{i=1}^{m} r_i^2(x), \tag{27}$$

其中$r_i : \mathbb{R}^n \to \mathbb{R}$为实值函数. 如果所有的$r_i$都是线性函数，则称问题(27) 为线性最小二乘问题，否则称其为非线性最小二乘问题. 最小二乘问题是线性回归和非线性回归的基础.

- 最小二乘问题也常用于线性（非线性）方程组问题当中. 当线性（非线性）观测带有噪声时，我们一般会基于该线性（非线性）系统建立最小二乘模型. 特别地，如果噪声服从高斯分布，最小二乘问题的解对应于原问题的最大似然解.

线性最小二乘问题

- 线性最小二乘问题是回归分析中的一个基本模型，它可以表示为

$$\min_{x\in\mathbb{R}^n} \quad \sum_{i=1}^{m}(a_i^{\mathrm{T}}x - b_i)^2,$$

即$r_i(x) = a_i^{\mathrm{T}}x - b_i,\ i = 1, 2, \cdots, m$.

- 记$A = [a_1, a_2, \cdots, a_m]^{\mathrm{T}}$，那么线性最小二乘问题可以等价地写成如下形式：

$$\min_{x\in\mathbb{R}^n} \quad f(x) \xlongequal{\text{def}} \frac{1}{2}\|Ax - b\|_2^2,$$

这是一个无约束二次目标函数的优化问题.

线性最小二乘问题

- 因为二次函数f 是凸的，故$x\in\mathbb{R}^n$ 为其全局极小解当且仅当x 满足方程

$$\nabla f(x)=A^{\mathrm{T}}(Ax-b)=0.$$

- 事实上，因为f 是二次的，则有

$$\begin{aligned} f(y) &= f(x)+\nabla f(x)^{\mathrm{T}}(y-x)+\frac{1}{2}(y-x)^{\mathrm{T}}\nabla^2 f(x)(y-x) \\ &= f(x)+\nabla f(x)^{\mathrm{T}}(y-x)+\frac{1}{2}(y-x)^{\mathrm{T}}A^{\mathrm{T}}A(y-x). \end{aligned}$$

- 因此，如果$\nabla f(x)=0$，根据$A^{\mathrm{T}}A$ 的半正定性，

$$f(y)\geq f(x),\quad \forall y\in\mathbb{R}^n,$$

即x 为$f(x)$ 的全局极小解.

- 反之，如果$\nabla f(x) \neq 0$，此时说明沿着负梯度方向目标函数将减小.
- 具体地，取$y = x - t\nabla f(x)$且$t = \frac{1}{\lambda_{\max}(A^\mathrm{T}A)}$，其中$\lambda_{\max}(A^\mathrm{T}A)$ 表示$A^\mathrm{T}A$ 的最大特征值，那么

$$
\begin{aligned}
f(x - t\nabla f(x)) &\leq f(x) - t\|\nabla f(x)\|_2^2 + \frac{1}{2}t^2\lambda_{\max}(A^\mathrm{T}A)\|\nabla f(x)\|_2^2 \\
&= f(x) + \left(-t + \frac{1}{2}t^2\lambda_{\max}(A^\mathrm{T}A)\right)\|\nabla f(x)\|_2^2 \\
&= f(x) - \frac{1}{2\lambda_{\max}(A^\mathrm{T}A)}\|\nabla f(x)\|_2^2 < f(x).
\end{aligned}
$$

因而在全局极小解x 处必有$\nabla f(x) = 0$.

- 给定数据集$\{a_i \in \mathbb{R}^p, b_i \in \mathbb{R}^q, i=1,2,\cdots,m\}$，插值是求一个映射$f$，使得
$$b_i = f(a_i), \quad i=1,2,\cdots,m.$$
- 在实际中，出于计算上的可行性，我们一般会限制在一个特定函数空间上来求f的一个逼近解．如果利用线性函数逼近，即$f(a)=Xa+y$，其中$X \in \mathbb{R}^{q\times p}$，$y \in \mathbb{R}^q$，则为了求解$X,y$，可以建立如下最小二乘问题：
$$\min_{X\in\mathbb{R}^{q\times p}} \quad \sum_{i=1}^{m} \|Xa_i + y - b_i\|^2.$$
- 一般地，假设$\{\phi_i(a)\}_{i=1}^n (n \le m)$ 为插值空间的一组基，数据插值问题可以写成
$$b_j = f(a_j) = \sum_{i=1}^{n} x_i \phi_i(a_j), \quad j=1,2,\cdots,m,$$
其中x_i 为待定系数．这是关于x 的线性方程组．

除了这种基函数的和的方式，深度学习也通过一些简单函数的复合来逼近原未知函数.

- 具体地，假设有一些简单的非线性向量函数$\phi_i(\theta):\mathbb{R}^q\to\mathbb{R}^q$，并构造如下复合函数：

$$f(\theta)=\phi_n\left(X_n\phi_{n-1}(X_{n-1}\cdots\phi_1(X_1\theta+y_1)\cdots+y_{n-1})+y_n\right).$$

- 在实际中常用的简单非线性函数有ReLU，即

$$\phi_i(\theta)=(\mathrm{ReLU}(\theta_1),\mathrm{ReLU}(\theta_2),\cdots,\mathrm{ReLU}(\theta_q))^{\mathrm{T}},\quad i=1,2,\cdots,n,$$

且

$$\mathrm{ReLU}(t)=\begin{cases}t, & t\geq 0,\\ 0, & \text{其他}.\end{cases}$$

- 这样的做法往往会带来更多未知的非线性，因而可能在更大的函数空间中得到未知函数的一个更好的逼近．将点a_i处的取值代入，则得到如下非线性方程组：

$$
\begin{aligned}
f(a_i) - b_i = &\ \phi_n\left(X_n\phi_{n-1}(X_{n-1}\cdots\phi_1(X_1a_i + y_1)\cdots + y_{n-1}) + y_n\right) - b_i \\
= &\ 0, \quad i = 1, 2, \cdots, n.
\end{aligned}
$$

- 需要求解的是关于$X_1 \in \mathbb{R}^{q\times p}, X_i \in \mathbb{R}^{q\times q}$, $y_i \in \mathbb{R}^q, i = 1, 2, \cdots, n$ 的非线性方程组．我们一般考虑替代的最小二乘问题

$$
\min_{\{X_i, y_i\}} \quad \sum_{i=1}^{m} \|f(a_i) - b_i\|^2.
$$

深度Q学习

- 在强化学习中，为了求解出最优策略及相应的期望奖励，往往需要考虑动作价值函数（action-value function）$Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$（注意，我们一般称$V$ 为价值函数，即value function），其表示从状态s 出发，采取动作a 可以获得的最大期望奖励.
- 根据最优性，其Bellman方程为

$$Q(s,a) = r(s,a) + \gamma \sum_{s'} P_a(s,s') \max_{a'} Q(s',a'). \tag{28}$$

对于求解方程(28)，一个常用的迭代算法的格式为：

$$Q_{k+1}(s,a) = r(s,a) + \gamma \sum_{s'} P_a(s,s') \max_{a'} Q_k(s',a'). \tag{29}$$

同样地，每一轮更新都需要对所有状态动作对(s,a)做一次迭代.

- 然而在实际问题中，我们通常没有模型信息，也就是说，上式涉及的奖励和状态转移概率都是未知的．为此，我们必须与环境进行交互，获取经验来估计这些项的大小．
- 在环境交互获得的经验中，可以得到很多形如(s_t, a_t, r_t, s_{t+1})的四元组，它记录了智能体在时刻t处于状态s_t时选择某个动作a_t，转移至状态s_{t+1}，同时获得奖励$r_t = r(s_t, a_t)$，算法可以根据这样的小段进行迭代更新：

$$Q_{k+1}(s_t, a_t) = (1-\alpha)Q_k(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q_k(s_{t+1}, a')),$$

其中$\alpha \in (0, 1)$．观察上式右端第二项，$r_t + \gamma \max_{a'} Q_k(s_{t+1}, a')$ 是对(29)式右端期望值的一个采样，为无偏估计．为了计算方便，我们不会等到所有数据生成后再计算平均值，而是利用凸组合的方式持续不断地将新的计算结果按一定权重加到原有数据上，这是强化学习中一种常用的均值计算技巧．

- 由于实际应用比较复杂，故利用$Q_\theta(s,a)$近似动作价值函数，它带有参数$\theta \in \mathbb{R}^d$. 当参数θ固定后，它就是一个关于状态s和动作a的二元函数. $Q_\theta(s,a)$有很多不同的形式，它可以是最简单的线性函数，也可以是复杂的神经网络（深度Q学习），其权重矩阵和偏差项作为这里的参数θ. 但$Q_\theta(s,a)$通常不会和真实动作价值函数完全相等.
- 严格来讲，我们希望最小化平方损失，即

$$\min_\theta \quad L(\theta) \stackrel{\text{def}}{=\!=} \mathbb{E}_{(s,a)\sim\rho(s,a)}[(y_\theta(s,a) - Q_\theta(s,a))^2], \tag{30}$$

其中$\rho(s,a)$是状态动作对(s,a)出现的概率分布，

$$y_\theta(s,a) = r(s,a) + \gamma \sum_{s'} P_a(s,s') \max_{a'} Q_\theta(s',a')$$

是近似函数希望逼近的目标.

- 问题(30) 实际上就是在极小化方程(28)的残差．但在实际应用中，由于(30) 比较复杂，深度Q 学习采用迭代方式求解．
- 在迭代的第i步近似求解如下优化问题：

$$\min_{\theta} \quad L_i(\theta) \stackrel{\text{def}}{=\!=} \mathbb{E}_{(s,a)\sim\rho(s,a)}[(y_i - Q_\theta(s,a))^2], \tag{31}$$

其中

$$y_i = r(s,a) + \gamma \sum_{s'} P_a(s,s') \max_{a'} \{Q_{\theta_{i-1}}(s',a')\},$$

参数θ_{i-1}来自上一步迭代的估计．和问题(30) 不同，y_i与待优化的变量θ无关，因此可以认为问题(31) 是随机版本的最小二乘问题．具体算法实现时还需进一步对它进行抽样处理．

带微分方程约束优化问题

- 当约束中含微分方程时，称相应的优化问题为带微分方程约束的优化问题．其在最优控制、形状优化等各种领域中有着广泛应用．求解瓦斯油催化裂解生成气体和其他副产物的反应系数的问题反应过程可以由如下非线性常微分方程组表示：

$$\begin{cases} \dot{y}_1 = -(\theta_1 + \theta_3) y_1^2, \\ \dot{y}_2 = \theta_1 y_1^2 - \theta_2 y_2, \end{cases} \tag{32}$$

其中系数$\theta_i \geq 0,\ i = 1,2,3$，且$y_1, y_2$ 的初值条件是已知的．

- 我们考虑的问题是

$$\begin{aligned} \min_{\theta \in \mathbb{R}^3} \quad & \sum_{j=1}^{n} \| y(\tau_j; \theta) - z_j \|^2, \\ \text{s.t.} \quad & y(\tau; \theta) \text{满足方程组(32)} \end{aligned}$$

这里z_j 是在时刻τ_j 的y 的测量值，n 为测量的时刻数量．