机器学习理论基础

黎雷蕾

2018 年 1 月 23 日

摘要

本文所写的均为经典常用机器学习算法的公式推理，作为将来面试时的参考资料。

想要取得较好的面试，除了理论外还需要有实际的经验，这样才能避免很多坑；当然，leetcode 经典的那 100 多题也是必不可少的。

目录

Chapter 1

代数基础

1.1 损失函数

1.1.1 0-1 损失 (Zero-one Loss)

最简单的损失函数，如果预测值 $\hat{y}_i$ 与目标值 y_i 不相等，那么为 1，否则为 0.

$$\ell(y_i, \hat{y}_i) = \begin{cases} 1, & y_i \neq \hat{y}_i; \\ 0, & y_i = \hat{y}_i. \end{cases} \tag{1.1}$$

1.1.2 感知损失 (Perceptron Loss)

用来改进 0-1 损失中判定较为严格的问题:

$$\ell(y_i, \hat{y}_i) = \begin{cases} 1, & |y_i - \hat{y}_i| > t \\ 0, & |y_i - \hat{y}_i| \leq t. \end{cases} \tag{1.2}$$

1.1.3 Hinge Loss

Hinge Loss 可以用来解决 SVM 只能够的间隔最大化问题。

$$\begin{aligned} \ell(y_i, \hat{y}_i) = \max\{0, 1 - y_i \cdot \hat{y}_i\}, \\ y_i \in \{-1, +1\}, \quad \hat{y}_i \in [-1, +1]. \end{aligned} \tag{1.3}$$

1.1.4 绝对值误差

常用回归中:

$$\ell(y_i, \hat{y}_i) = |y_i - \hat{y}_i| \tag{1.4}$$

1.1.5 均方误差

常用于回归中:

$$\ell(y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2, \tag{1.5}$$

1.1.6 交叉熵

神经网络、逻辑回归中常用的损失函数，二分类问题可写作:

$$\begin{aligned} \ell(y_i, \hat{y}_i) &= y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i) \\ y_i &\in \{0, 1\} \end{aligned} \tag{1.6}$$

多分类问题可写作:

$$\ell(y_i, \hat{y}_i) = -\sum_{i=0}^{n} y_i \log(\hat{y}_i) \tag{1.7}$$

1.1.7 指数误差 (Exponential)

常用于 boosting 算法:

$$\ell(y_i, \hat{y}_i) = \exp(-y_i \cdot \hat{y}_i) \tag{1.8}$$

1.2 数值优化算法

1.2.1 牛顿法 (Newton's method)

牛顿法是是一种在实数域和复数域上近似求解方程的方法。方法使用函数 $f(x)$ 的泰勒级数的前面几项来寻找方程 $f(x) = 0$ 的根。也被称为切线法。

将 $f(x)=0$ 在 x_0 处展开成泰勒级数：

$$f(x_0) \to \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n \tag{1.9}$$

我们只取其线性部分，作为非线性方程的近似方程：

$$f(x_0)+(x-x_0)f'(x_0)=0 \tag{1.10}$$

设 $f'(x_0)\neq 0$，则其解为：

$$x=x_1=x_0-\frac{f(x_0)}{f'(x_0)} \tag{1.11}$$

这个公式说明 $f(x_1)$ 的值将会比 $f(x_0)$ 更加接近 $f(x)=0$，我们就可以用迭代法逼近：

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)} \tag{1.12}$$

通过迭代，上式必然会在 $f(x)=0$ 处收敛。

1.2.2 拟牛顿法 (Quasi-Newton Methods)

拟牛顿法 (Quasi-Newton Methods) 是求解非线性优化问题最有效的方法之一。

海森矩阵 (Hessian Matrix)

海森矩阵是一个多元函数的二阶偏导数构成的方阵，描述了函数的局部曲率。假设二元函数 $f(x_1,x_2)$ 在 $\mathbf{X}^{(0)}(x_1^{(0)},x_2^{(0)})$ 点处的泰勒展开式为：

$$\begin{aligned}&f(x_1,x_2)=\\&f(x_1^{(0)},x_2^{(0)})+\left.\frac{\partial f}{\partial x_1}\right|_{\mathbf{X}^{(0)}}\Delta x_1+\left.\frac{\partial f}{\partial x_2}\right|_{\mathbf{X}^{(0)}}\Delta x_2+\\&\frac{1}{2}\left[\left.\frac{\partial^2 f}{\partial x_1^2}\right|_{\mathbf{X}^{(0)}}\Delta x_1^2+2\left.\frac{\partial^2 f}{\partial x_1\partial x_2}\right|_{\mathbf{X}^{(0)}}\Delta x_1\Delta x_2+\left.\frac{\partial^2 f}{\partial x_2^2}\right|_{\mathbf{X}^{(0)}}\Delta x_2^2\right]+\cdots\end{aligned} \tag{1.13}$$

其中，$\Delta x_1=x_1-x_1^{(0)},\Delta x_2=x_2-x_2^{(0)}$.

将上式写成矩阵形式：

$$\begin{aligned}
&f(\mathbf{X}) = \\
&f(\mathbf{X}^{(0)}) + \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)_{\mathbf{X}^{(0)}} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \\
&\frac{1}{2}(\Delta x_1, \Delta x_2) \left.\begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix}\right|_{\mathbf{X}^{(0)}} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \cdots
\end{aligned} \tag{1.14}$$

即:

$$f(\mathbf{X}) = f(\mathbf{X}^{(0)}) + \nabla f(\mathbf{X}^{(0)})^T \Delta\mathbf{X} + \frac{1}{2}\Delta\mathbf{X}^T H(\mathbf{X}^{(0)})\Delta\mathbf{X} + \cdots \tag{1.15}$$

其中:

$$H(\mathbf{X}^{(0)}) = \left.\begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix}\right|_{\mathbf{X}^{(0)}}, \ \Delta\mathbf{X} = \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} \tag{1.16}$$

$H(\mathbf{X}^{(0)})$ 是 $f(x_1, x_2)$ 在 $\mathbf{X}^{(0)}$ 处的海森矩阵，由 $f(x_1, x_2)$ 在 $\mathbf{X}^{(0)}$ 处的二阶偏导数组成。

将海森矩阵扩展到 n 元函数，对应的梯度 $\nabla f(\mathbf{X}^{(0)})$ 和海森矩阵可以写作 $H(\mathbf{X}^{(0)})$:

$$\nabla f(\mathbf{X}^{(0)}) = \left.\left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right]\right|_{\mathbf{X}^{(0)}}^T \tag{1.17}$$

$$H(\mathbf{X}^{(0)}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}_{\mathbf{X}^{(0)}} \tag{1.18}$$

拟牛顿法思想

在牛顿法的迭代中，需要计算海森矩阵的逆矩阵 H^{-1}，这个计算十分复杂，所以考虑到用一个 n 阶矩阵 $G_k = G(x^{(k)})$ 来近似代替 $H_k^{-1} = H^{-1}(x^{(k)})$，这就是拟牛顿法的基本思想了。

假设 $g_k = g(x^{(k)}) = \nabla f(x^k)$ 是 $f(x)$ 的梯度向量在点 $x^{(k)}$ 的值。那么牛顿法的更新公式就可以写作:

$$x^{(k+1)} = x^{(k)} - H_k^{-1} g_k \tag{1.19}$$

拟牛顿法的公式可以写作:

$$x^{(k+1)} = x^{(k)} - G_k g_k \tag{1.20}$$

Davidon Fletcher Powell, DFP 算法

1.2.3 梯度下降 (Gradient descent)

梯度

所谓梯度，就是指函数变化最快的地方。对于一个函数 $f(x, y)$，分别对于 x, y 求偏导，获得的梯度向量为 $(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})^T$，简称为 $grad\ f(x, y)$ 或者 $\nabla f(x, y)$。梯度方向指的是沿着梯度向量 $\nabla f(x, y)$ 的方向。

梯度下降和梯度上升

两者的实质是一样的，梯度下降取相反数就是梯度上升了。

梯度下降的代数方法表示

假设一个线性回归函数的表示公式为:

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i \tag{1.21}$$

损失函数取均方误差:

$$J(\theta) = \frac{1}{n} \sum_{i=0}^{n} (h_\theta(x) - y_i)^2 \tag{1.22}$$

求出 $J(\theta)$ 的梯度:

$$\frac{\partial J(\theta)}{\partial \theta_i} = \frac{2}{n} \left(\sum_{i=0}^{n} \frac{\partial h_\theta(x)}{\partial \theta_i} - y_i \right) x_i^{(j)} \tag{1.23}$$

那么更新的梯度表达式可以写作：

$$\theta_i' = \theta_i - \alpha \frac{\partial J(\theta)}{\partial \theta_i} \tag{1.24}$$

其中 $\alpha \in [0,1]$ 称为学习步长，取值过大会造成震荡，太小会造成收敛过慢。

梯度下降的矩阵方法表示

假设一个线性回归函数的矩阵表示公式为：

$$h_\theta(\mathbf{x}) = \mathbf{X}\theta \tag{1.25}$$

那么损失函数可以表示为:

$$J(\theta) = \frac{1}{2}(\mathbf{X}\theta - \mathbf{Y})^T(\mathbf{X}\theta - \mathbf{Y}) \tag{1.26}$$

求出 $J(\theta)$ 的梯度：

$$\begin{gathered}\frac{\partial}{\partial \mathbf{X}}(\mathbf{X}\mathbf{X}^T) = 2\mathbf{X} \\ \frac{\partial}{\partial \theta}(\mathbf{X}\theta) = \mathbf{X}^T \\ \frac{\partial}{\partial \theta}J(\theta) = \mathbf{X}^T(\mathbf{X}\theta - \mathbf{Y})\end{gathered} \tag{1.27}$$

那么更新的梯度表达式可以写作：

$$\theta_i' = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta) \tag{1.28}$$

梯度下降参数调优

1. 步长 α：太大会造成震荡从而错过最优解，太小会造成迭代速度太慢。
2. 参数初始值选择，梯度下降求出的是局部最优解，所以参数初始值不同求出的解也会不同，最好是正态分布随机生成。
3. 归一化，可以加快迭代速度。

与其它优化算法比较

梯度下降法和最小二乘法相比，梯度下降法需要选择步长，而最小二乘法不需要。梯度下降法是迭代求解，最小二乘法是计算解析解。如果样本量不算很大，且存在解析解，最小二乘法比起梯度下降法要有优势，计算速度很快。但是如果样本量很大，用最小二乘法由于需要求一个超级大的逆矩阵，这时就很难或者很慢才能求解解析解了，使用迭代的梯度下降法比较有优势。

梯度下降法和牛顿法/拟牛顿法相比，两者都是迭代求解，不过梯度下降法是梯度求解，而牛顿法/拟牛顿法是用二阶的海森矩阵的逆矩阵或伪逆矩阵求解。相对而言，使用牛顿法/拟牛顿法收敛更快。但是每次迭代的时间比梯度下降法长。

1.2.4 Momentum

Momentum 借鉴了物理学中动量的思想，通过积累之前的动量 m_{t-1} 来加速当前的梯度。设 μ 是动量因子，通常设为 0.9 或其近似值：

$$\begin{aligned} m_t &= \mu \cdot m_{t-1} + \alpha \nabla J(\theta) \\ \theta'_t &= \theta_t - m_t \end{aligned} \tag{1.29}$$

特点：

-

Chapter 2

线性模型

2.1 基本形式

设 x 为输入，$f(x)$ 为输出，w 为权重 (weight)，b 为偏差 (bias)，那么一般的向量线性模型可以写为:

$$f(x) = w^T x + b \tag{2.1}$$

2.2 线性回归 (linear regression)

线性回归试图学习一个线性模型来尽可能准确地预测实值输出标记，假设 y 为标签 (label)，即:

$$f(x_i) = w^T x_i + b, \text{使得} f(x_i) \simeq y_i \tag{2.2}$$

这里涉及到损失函数的概念，一般用得多的是均方误差和交叉熵两种，西瓜书上的推导是均方误差，所以这里也是写均方误差了。

均方误差 (mean-square error, MSE) 对应常用的欧式距离 (Euclidean distance),基于均方误差的求解方法也被称为最小二乘法 (least square method)，在线性回归中，最小二乘法试图找到一条直线，使得所有样本到该直线上的

欧式距离之和最小。

$$
\begin{aligned}
(w^*, b^*) &= \mathop{\arg\min}_{(w,b)} \sum_{i=1}^{m} (f(x_i) - y_i)^2 \\
&= \mathop{\arg\min}_{(w,b)} \sum_{i=1}^{m} (y_i - w^T x_i - b)^2 \\
&= E_{(w,b)}
\end{aligned}
\tag{2.3}
$$