

2.1.4 优化基础模型

2.1.4.1 训练误差和测试误差

训练误差就是模型在训练集上的误差平均值，度量了模型对训练集拟合的情况。训练误差大说明对训练集特性学习得不够，训练误差太小说明过度学习了训练集特性，容易发生过拟合。

测试误差是模型在测试集上的误差平均值，度量了模型的泛化能力。在实践中，希望测试误差越小越好。

随着训练轮次增加，通常来说训练误差会逐渐降低，而测试误差会呈现U型，先降低再升高，体现了偏差和方差的博弈。

2.1.4.2 偏差-方差的均衡

我们可以将均方误差进行分解，得到：

$$E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0))]^2 + \text{Var}(\epsilon)$$

其证明如下：

$$\begin{aligned} E(f; D) &= \mathbb{E}_D [(f(\mathbf{x}; D) - y_D)^2] \\ &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}) + \bar{f}(\mathbf{x}) - y_D)^2] \\ &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y_D)^2] \\ &\quad + \mathbb{E}_D [2(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))(\bar{f}(\mathbf{x}) - y_D)] \\ &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y_D)^2] \\ &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y + y - y_D)^2] \\ &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + \mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)^2] + \mathbb{E}_D [(y - y_D)^2] \\ &\quad + 2\mathbb{E}_D [(\bar{f}(\mathbf{x}) - y)(y - y_D)] \\ &= \mathbb{E}_D [(f(\mathbf{x}; D) - \bar{f}(\mathbf{x}))^2] + (\bar{f}(\mathbf{x}) - y)^2 + \mathbb{E}_D [(y_D - y)^2], \end{aligned}$$

偏差是用所有可能的训练数据集训练出的所有模型的输出的平均值与真实模型的输出值之间的差异。

方差是不同的训练数据集训练出的模型输出值之间的差异。

噪声的存在是学习算法所无法解决的问题，数据的质量决定了学习的上限。假设在数据已经给定的情况下，此时上限已定，我们要做的就是尽可能的接近这个上限。

我们的目标是最小化偏差和方差的和，两者之间存在一个权衡。一般来说，模型复杂度越高，方差越大，越容易过拟合；模型复杂度越低，方差越小，越容易欠拟合。

2.1.4.3 防止过拟合的策略

特征提取

模型简单时容易欠拟合，而复杂时容易过拟合，我们可以通过特征提取模型选择合适的特征数量。

$C_p = \frac{1}{N}(RSS + 2d\hat{\sigma}^2)$ ，其中d为模型特征个数， $RSS = \sum_{i=1}^N (y_i - \hat{f}(x_i))^2$ ， $\hat{\sigma}^2$ 为模型预测误差的方差的估计值，即残差的方差。

- AIC赤池信息量准则： $AIC = \frac{1}{d\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$
- BIC贝叶斯信息量准则： $BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2)$

交叉验证

另一种减少模型过拟合的方法是交叉验证，有k折交叉验证、ovo、ovr等。

添加正则化项

我们可以对回归模型添加惩罚项，从而对系数进行约束，显著降低模型方差，提高模型的拟合效果。常用的正则化项有L1正则项和L2正则项。

L1正则化 (Lasso)

$$J(w) = \sum_{i=1}^N (y_i - w_0 - \sum_{j=1}^p w_j x_{ij})^2 + \lambda \sum_{j=1}^p |w_j|, \text{ 其中, } \lambda \geq 0$$

L2正则化 (岭回归)

$$J(w) = \sum_{i=1}^N (y_i - w_0 - \sum_{j=1}^p w_j x_{ij})^2 + \lambda \sum_{j=1}^p w_j^2, \text{ 其中, } \lambda \geq 0$$
$$\hat{w} = (X^T X + \lambda I)^{-1} X^T Y$$

降维

我们也可以通过降维来减少高维特征的冗余信息和噪音信息，从而提高识别精度。常见的算法有主成分分析(PCA)、t-sne等。

主成分分析(PCA)：通过**最大投影方差**将原始空间进行重构，即由特征相关重构为无关，即落在某个方向上的点(投影)的方差最大。在进行下一步推导之前，我们先把样本均值和样本协方差矩阵推广至矩阵形式：

样本均值Mean: $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i = \frac{1}{N} X^T \mathbf{1}_N$ ，其中 $\mathbf{1}_N = (1, 1, \dots, 1)_N^T$

样本协方差矩阵 $S^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{N} X^T H X$ ，其中 $H = I_N - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T$

最大投影方差的步骤：

- 中心化： $x_i - \bar{x}$
- 计算每个点 x_1, \dots, x_N 至 \vec{u}_1 方向上的投影： $(x_i - \bar{x})\vec{u}_1$ ， $\|\vec{u}_1\| = 1$
- 计算投影方差： $J = \frac{1}{N} \sum_{i=1}^N [(x_i - \bar{x})^T \vec{u}_1]^2$ ， $\|\vec{u}_1\| = 1$
- 最大化投影方差求 \vec{u}_1 ：

$$\begin{aligned}\bar{u}_1 = \operatorname{argmax}_{u_1} \quad & \frac{1}{N} \sum_{i=1}^N [(x_i - \bar{x})^T \vec{u}_1]^2 \\ \text{s.t. } \vec{u}_1^T \vec{u}_1 = 1 & (\vec{u}_1 \text{ 往后不带向量符号})\end{aligned}$$

得到：

$$\begin{aligned}J &= \frac{1}{N} \sum_{i=1}^N [(x_i - \bar{x})^T \vec{u}_1]^2 = \frac{1}{N} \sum_{i=1}^N [u_1^T (x_i - \bar{x})(x_i - \bar{x})^T u_1] \\ &= u_1^T \left[\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \right] u_1 = u_1^T S^2 u_1\end{aligned}$$

即：

$$\begin{aligned}\hat{u}_1 &= \operatorname{argmax}_{u_1} u_1^T S^2 u_1, \quad \text{s.t. } u_1^T u_1 = 1 \\ L(u_1, \lambda) &= u_1^T S^2 u_1 + \lambda(1 - u_1^T u_1) \\ \frac{\partial L}{\partial u_1} &= 2S^2 u_1 - 2\lambda u_1 = 0\end{aligned}$$

即： $S^2 u_1 = \lambda u_1$ 可以看到： λ 为 S^2 的特征值， u_1 为 S^2 的特征向量。因此我们只需要对中心化后的协方差矩阵进行特征值分解，得到的特征向量即为投影方向。如果需要进行降维，那么只需要取 p 的前 M 个特征向量即可。