1 模型选择与评估

1.1 模型的训练, 选择与交叉检验

1.1.1 训练集, 验证集, 测试集

对于同一问题, 我们可以采取不同的模型来进行训练. 即使模型相同, 其中的超参数也可以不同. 如何选择最优的模型就是本节需要解决的问题.

我们在前面主要通过模型在测试集上的表现来评价模型. 然而, 如果用模型在测试集的误差来选择模型, 相当于把测试集的数据也作为训练过程的一部分, 这就导致了测试集的**数据泄露 (data leakage)**. 这会导致模型泛化能力的下降.

因此, 较好的做法如下:

定义 1.1 训练集, 验证集, 测试集 为了选择更优的模型的同时防止数据泄露, 可以将数据集划分为三部分: **训练集 (Training Set)**, **验证集 (Validation Set)** 和**测试集 (Test Set)**. 其中训练集用于训练模型, 验证集用于选择模型, 测试集用于评估模型的最终性能.

以监督学习为例, 记已知数据集 $\mathcal{D}:\{\boldsymbol{x}_n,t_n\}$. 假定模型给出的预测函数为 $\hat{f}(\boldsymbol{x};\boldsymbol{w})$, 单个样本的损失函数为 $L(t,\hat{f}(\boldsymbol{x};\boldsymbol{w}))$, 则任意数据集 $\mathcal{D}$ 上的误差可以表示为:

$$E(\boldsymbol{w},\mathcal{D})=\sum_{(\boldsymbol{x},t)\in\mathcal{D}}L(t,\hat{f}(\boldsymbol{x};\boldsymbol{w}))$$

在使用训练集 $\mathcal{D}_{\text{train}}$ 训练时, 我们一般还考虑正则化项, 因此训练的目标为:

$$\boldsymbol{w}(\lambda)=\arg\min_{\boldsymbol{w}}\tilde{E}\left(\boldsymbol{w},\mathcal{D}_{\text{train}}\right),\quad \tilde{E}\left(\boldsymbol{w},\mathcal{D}_{\text{train}}\right)=E\left(\boldsymbol{w},\mathcal{D}_{\text{train}}\right)+\frac{\lambda}{2}||\boldsymbol{w}||^2$$

此时模型在验证集 $\mathcal{D}_{\text{val}}$ 上的误差称作验证误差, 即 (注意这里不包含正则化项)

$$E_{\text{val}}(\lambda)=E\left(\boldsymbol{w},\mathcal{D}_{\text{val}}\right)$$

选择模型, 也就是选择使得验证误差最小的超参数 λ:

$$\lambda^*=\arg\min_{\lambda}E_{\text{val}}(\lambda)$$

最后, 用选择出的超参数 λ 和对应的模型参数 $\boldsymbol{w}$ 在测试集 $\mathcal{D}_{\text{test}}$ 上评估其性能, 即计算测试误差:

$$E_{\text{test}}=E\left(\boldsymbol{w}(\lambda^*),\mathcal{D}_{\text{test}}\right)$$

这就可以衡量模型的最终效果, 即泛化能力.

关于如何划分训练集, 验证集和测试集, 一般的做法是将数据集随机划分为 50% 的训练集,25% 的验证集和 25% 的测试集 (也有人建议为 60%,20%,20%). 如果数据量较少, 也可以考虑使用交叉检验的方法来更有效地利用数据.

1.1.2 K 折交叉检验

在数据不足的情况下, 如果还要划分出前述三个数据集, 每个数据集的样本量都会较少, 从而影响模型的训练和评估. 同时, 数据集的划分本身也是随机的, 这也可能导致结果的不准确.

为了减少数据浪费和数据拆分的随机性, 这时可以考虑使用 **K 折交叉检验 (K-Fold Cross Validation)** 的方法来更有效地利用数据.

定义 1.2 K 折交叉检验 **K 折交叉检验**将数据集平均划分为 K 个子集, 每次用其中的 $K-1$ 个子集作为训练集, 剩下的一个子集作为验证集, 重复 K 次, 每个子集都作为一次验证集. 最后将 K 次的验证误差取平均作为最终的验证误差.

在最极端的情形下, 还可以采用留一法.

定义 1.3 留一法 **留一法 (Leave-One-Out, LOO)** 是 K 折交叉检验的特例, 即 $K=N$, 每次用 $N-1$ 个样本作为训练集, 剩下的一个样本作为验证集, 重复 N 次, 每个样本都作为一次验证集. 最后将 N 次的验证误差取平均作为最终的验证误差.

交叉检验虽然计算代价很高, 但它对数据的使用更充分, 在数据较少的情况下具有不错的效果.

1.1.3 误差的偏差-方差分解

我们在前面讲到模型过于简单和过于复杂分别会导致欠拟合和过拟合, 都会使得模型在测试集上的误差较大. 下面我们通过对**误差的偏差-方差分解**来解释之.

设观察值 t_n 可以表示成下面的形式:

$$t_n = f(\boldsymbol{x}_n) + \varepsilon$$

其中 f 是内禀的真实函数, ε 是均值为 0 的噪声. 基于训练集 $\mathcal{D}_{\text{train}}$ 和验证集 $\mathcal{D}_{\text{val}}$(将两者合并记作 $\mathcal{D}$) 训练并选择得到的机器学习函数为 $\hat{f}(\boldsymbol{x};\boldsymbol{w}|\mathcal{D})$, 于是在测试集上某个样本 $(\boldsymbol{x}_m, t_m)$ 上的误差 (这里采取二乘误差) 为:

$$E_{\text{test}} = \left(f(\boldsymbol{x}_m) + \varepsilon - \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right)^2$$

由于数据选择的随机性, 以及一些模型 (比如随机梯度下降) 本身的随机性, 我们可以把 $\hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})$ 视作一个随机变量, 其期望记作 $\left\langle \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right\rangle$. 将上式稍作整理可得

$$E_{\text{test}} = \left[\left(f(\boldsymbol{x}_m) - \left\langle \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right\rangle\right) - \left(\hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}) - \left\langle \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right\rangle\right) + \varepsilon\right]^2$$

上述三项中, 第一项是无随机性的定值, 于是 E_{test} 的期望为:

$$\begin{aligned}\left\langle E_{\text{test}}\right\rangle = &\left[f(\boldsymbol{x}_m) - \left\langle \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right\rangle\right]^2 + \left\langle\left[\hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}) - \left\langle \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right\rangle\right]^2\right\rangle \\ &- 2\left\langle \varepsilon\left(\hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}) - \left\langle \hat{f}(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D})\right\rangle\right)\right\rangle + \left\langle \varepsilon^2\right\rangle\end{aligned}$$

由于交叉项 $2\left\langle \varepsilon \left(\hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right) - \left\langle \hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right)\right\rangle \right)\right\rangle$ 中的两个随机变量 ε 和 $\hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right) - \left\langle \hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right)\right\rangle$ 分别来源于训练集和测试集, 因此它们是独立的, 所以该项为零. 于是

$$\left\langle E_{\text{test}}\right\rangle = \underbrace{\left[f\left(\boldsymbol{x}_m\right) - \left\langle \hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right)\right\rangle\right]^2}_{\text{偏差}^2} + \underbrace{\left\langle \left[\hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right) - \left\langle \hat{f}\left(\boldsymbol{x}_m;\boldsymbol{w}|\mathcal{D}\right)\right\rangle\right]^2\right\rangle}_{\text{方差}} + \underbrace{\left\langle \varepsilon^2\right\rangle}_{\text{噪声}}$$

上式即为误差的偏差-方差分解, 亦可以简记为

$$\left\langle E_{\text{test}}\right\rangle = \left\langle \varepsilon^2\right\rangle + B^2 + V$$

其中, **偏差 (Bias)** 衡量了模型预测值的期望与真实值之间的差距, 反映了模型的拟合能力; **方差 (Variance)** 衡量了模型预测值的波动性, 反映了模型对训练数据的敏感程度; **噪声 (Noise)** 是数据本身固有的不可避免的误差.

如果选取过于简单的模型, 那么 $\hat{f}(\boldsymbol{x};\boldsymbol{w})$ 不太受到训练集的影响, 其方差较小, 但由于模型能力有限, 其偏差较大, 从而导致较大的误差. 这就是**欠拟合**.

如果选取过于复杂的模型, 那么 $\hat{f}(\boldsymbol{x};\boldsymbol{w})$ 会过度拟合训练集, 即使由于模型能力强, 其偏差较小, 也会因为对训练数据的敏感性较高而导致较大的方差, 从而导致较大的误差. 这就是**过拟合**.

1.1.4 学习曲线

为了更直观地理解偏差-方差分解, 我们可以通过绘制**学习曲线 (Learning Curve)** 来观察模型在训练集和验证集上的误差随训练样本数量的变化情况.

一般来说, 随着训练样本数量的增加, 模型在训练集上的误差会逐渐增大, 而在验证集上的误差会逐渐减小, 最终两者趋于一致.

对于欠拟合的情况, 两者很早就达到饱和并且相差较小. 这表明模型能力有限, 无法很好地拟合数据, 也就对应高偏差的情况.

对于过拟合的情况, 训练集上的误差很低, 而验证集上的误差较高, 并且两者相差较大. 这表明模型过度拟合了训练数据, 也就对应高方差的情况.

1.2 数据的预处理

1.2.1 数据标准化

数据的输入量 $\boldsymbol{x}$ 的各个分量可能具有不同的变化范围, 甚至可能具有不同的量纲. 例如, 在预测房价时, 房间楼层是一个整数, 但房屋面积是一个带面积单位的实数, 甚至其取值也会随着单位的变化而变化. 这样的情况会导致某些分量对模型的影响过大, 从而影响模型的训练效果.

为了避免这种情况, 我们通常会对数据进行标准化处理, 使得各个分量具有相似的变化范围和量纲. 常见的标准化方法如下.

定义 1.4 最小-最大缩放 最小-最大缩放 (Min-Max Scaling) 将数据对应到 $[0,1]$ 内, 具体做法是对每个分量 x_i 进行如下变换:

$$x_i \leftarrow \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

定义 1.5 Z-Score 标准化 Z-Score 标准化 (Z-Score Normalization) 将数据变换为均值为 0, 标准差为 1 的分布, 具体做法是对每个分量 x_i 进行如下变换:

$$x_i \leftarrow \frac{x_i - \mu_i}{\sigma_i}$$

其中 μ_i 和 σ_i 分别是 $\{x_i\}$ 的均值和标准差.

Z-Score 标准化对于大多数机器学习算法来说是较好的选择, 因为它能够更好地处理异常值和不同分布的数据.

概率模型一般不需要进行数据标准化, 因为它主要关心变量的分布和变量之间的条件概率, 决策树方法就属于此类. 除此之外, 线性回归, 逻辑回归, 神经网络, 支持向量机等模型一般都要进行数据标准化.

1.2.2 异常值检测

1.3 数据不平衡及分类模型的评价指标

在分类问题中, 有时某些类别的样本数量远少于其他类别, 这就导致了**数据不平衡 (Data Imbalance)** 的问题. 例如, 在样品合格检测中, 不合格的样本数量通常远少于合格的样本数量. 如果直接使用准确率作为评价指标, 可能会导致模型偏向于将所有样本都预测为合格, 从而不能达到预期.

在分类问题中, 可以用**混淆矩阵 (Confusion Matrix)** 来表示模型的预测结果.

定义 1.6 混淆矩阵 对于一个 n 个类别的分类问题, 模型的**混淆矩阵**是一个 $n \times n$ 的矩阵 C, 其中 C_{ij} 表示真实类别为 i 并且被模型预测为 j 的样本数量.

对于常见的二分类问题, 混淆矩阵可以表示为:

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

其中 TP 表示真阳性 (True Positive), 即真实类别为阳性且被模型预测为阳性的样本数量;TN 表示真阴性 (True Negative), 即真实类别为阴性且被模型预测为阴性的样本数量;FP 表示假阳性 (False Positive), 即真实类别为阴性但被模型预测为阳性的样本数量;FN 表示假阴性 (False Negative), 即真实类别为阳性但被模型预测为阴性的样本数量.

基于混淆矩阵, 我们可以定义一些常用的评价指标.

定义 1.7 基于混淆矩阵的评价指标

1. **准确率 (Accuracy)**: 表示模型预测正确的样本数量占总样本数量的比例, 定义为:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

准确率表征了一个模型的整体性能, 但这里的性能没有偏向于某一种类别 (更多时候, 它被占多数的类别所主导), 因此在数据不平衡的情况下, 准确率可能会误导我们对模型性能的判断.

2. **精度 (Precision)**(又称为**查准率**): 表示被模型预测为阳性的样本中真实为阳性的比例, 定义为:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Precision} = \frac{TN}{TN + FN}$$

精度反映了模型预测一个点属于某一类时, 该点真正属于此类的概率. 对某一类别的精度越高, 说明模型在确定某一点数以该类时越可靠.

在推荐系统中, 精度是一个重要的指标. 模型主要关心推荐的是否是用户真正感兴趣的内容, 而不太关心没有推荐的内容中是否有用户感兴趣的.

3. **召回率 (Recall)**(又称为**查全率**): 表示真实为某一类的样本中被模型预测为该类的比例, 定义为:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad \text{Recall} = \frac{TN}{TN + FP}$$

召回率反映了模型对某一类的覆盖能力. 对某一类别的召回率越高, 说明模型能够识别出更多属于该类的样本.

在疾病检测中, 召回率是一个重要的指标. 由于患病而漏诊的代价十分严重, 因此模型主要关心尽可能多地识别出患病的患者, 即使会误诊一些健康的患者.

4. F_1 **分数 (F_1 Score)**: 是精度和召回率的调和平均数, 定义为:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

F_1 分数综合考虑了精度和召回率, 在数据不平衡的情况下, 比单独使用精度或召回率更能反映模型的性能.