第11讲 模型评估与优化

主讲教师：欧新宇

February 21, 2020

Outlines

- 使用交叉验证对模型进行评估
  - 留出法 Holdout
  - K折交叉验证 K-Fold
  - 留一法 Leave One Out
  - 随机采样 ShuffleSplit
  - 分层采样法 Stratification
- 使用网格搜索(GridSearchCV)寻找模型的最优参数
- 对分类模型的可行度进行评估

模型评估与优化

回顾我们之前对一个模型进行性能评估的过程：

1. 载入数据集
2. 使用train_test_split类将数据集拆分成训练集(train set)和测试集（test set）
3. 使用训练集（train set）训练模型
4. 使用训练好的模型在测试集上进行测试并输出评分
5. 反复调整**超参数**并迭代地训练模型以获得最优模型

- 科学的模型评估算法：交叉验证法
- 获得最优模型的方法：网格搜索法

1. 使用交叉验证进行模型评估

交叉验证（Cross Validation），也称作**循环估计（Rotation Estimation）**，是一种统计学上将数据样本切割成较小子集的实用方法，该理论由Seymour Geisser提出的。主要用于建模应用中，在给定的建模样本中，拿出**大部分**样本进行建模型，留**小部分**样本用刚建立的模型进行预测，并求这小部分样本的预测误差，记录它们的平方加和PRESS(predicted Error Sum of Squares)。这个过程一直进行，直到所有的样本都被预报了一次而且仅被预报一次。

1. 使用交叉验证进行模型评估

在使用训练集对参数进行训练的时候，通常会将数据集为三个部分：训练集（train set），验证集（validation set），测试集（test set）。这样的划分是为了保证训练效果而特意设置的。

- **训练集**：用于训练模型的数据样本。
- **验证集**：在模型训练过程中，单独留出的样本集，用于调整模型的超参数和用于对模型的能力进行初步评估。
- **测试集**：用来评估模型最终的泛化能力，更多的时候用来对比不同算法的性能。但**不能**作为调参、选择特征等算法相关的选择的依据。

1. 使用交叉验证进行模型评估

在引入了验证集之后，模型评估过程可以改进为：

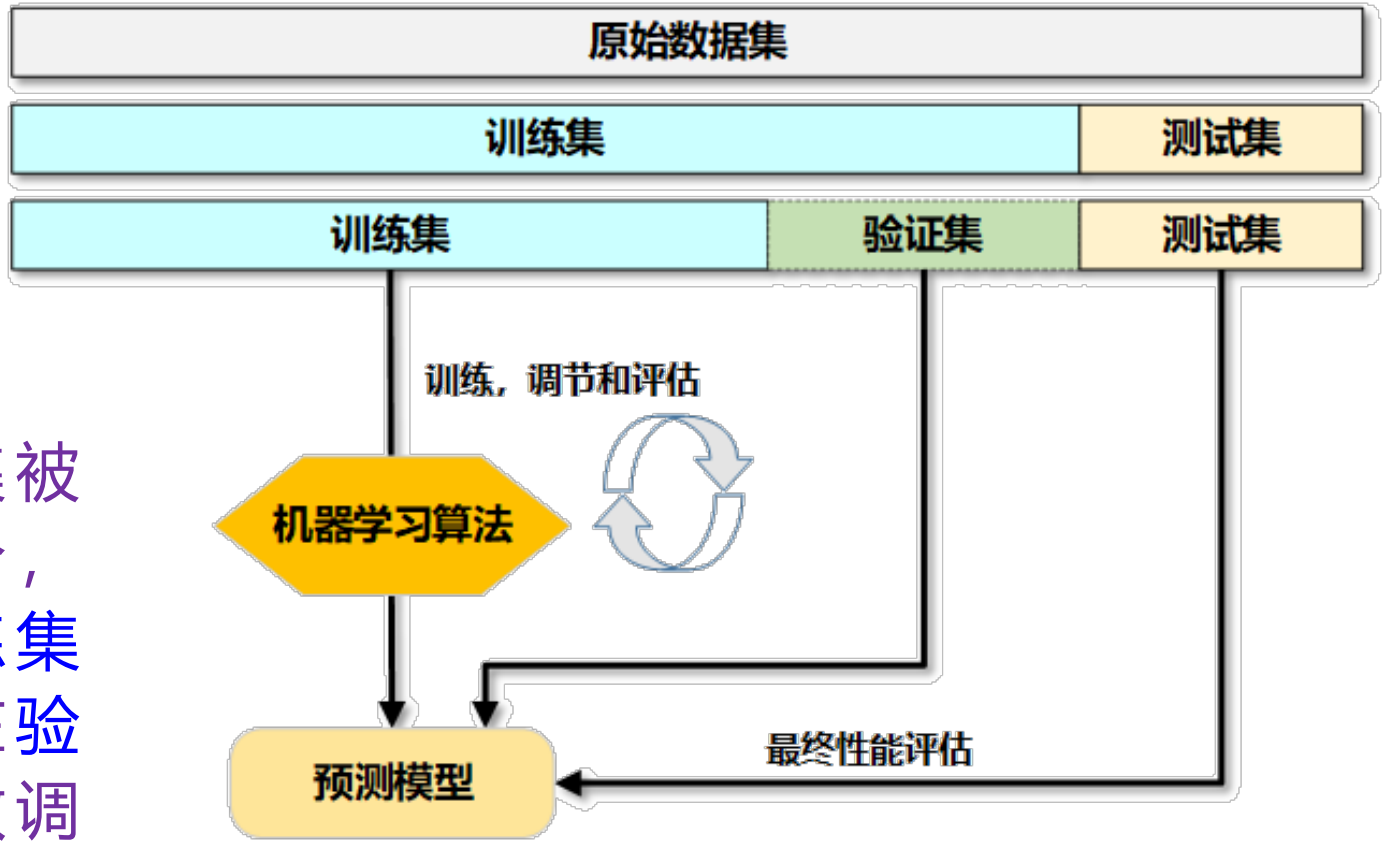
1. 载入数据集(dataset)
2. 将数据集拆分成训练集(train set)、测试集(test set)和验证集(validation set)
3. 使用训练集训练模型(model)
4. 在训练过程中**反复使用**验证集获取模型的评分(score)、误差(error)和损失(loss)（损失函数的值）等信息
5. 根据从验证集中获得的结论**调整超参数**(hyperparameter)
6. **反复迭代训练**，并**调整超参数**直到**收敛**(convergence)
7. 使用训练好(已收敛)的模型所获得的**超参数不变**，在 训练集+验证集 上再次进行训练，直到收敛，获得最终的模型**final_model**
8. 使用最终的模型**final_model**在测试集上输出评分

1. 使用交叉验证进行模型评估

常见的**交叉验证算法**有以下几种：

- 留出法（holdout cross validation）
- k折交叉验证（k-fold cross validation）
- 留一法（leave one out cross validation）
- 随机采样（ShuffleSplit）
- 分层采样法（Stratification）

1.1 留出法（Holdout）



原始的数据集被划分为三部分，我们会在训练集上做训练，在验证集上做参数调整，并最终在测试集上输出最终的性能评估。

1.1留出法（Holdout)

- **优点：**
  - 处理简单，只需随机把原始数据分为三组即可。
- **缺点：**
  - 只做一次分割，没有达到交叉的思想，由于是随机的将原始数据分组，所以最后验证集分类准确率的高低与原始数据的分组有很大的关系，得到的结果并不具有说服性。换句话说，划分出来作为验证集的样本，可能会存在类别不均衡的问题。
  - 数据集被分成三个集合后，用于训练的数据更少了。

1.2 k 折交叉验证（K-Fold Cross Validation）

k 折交叉验证法的基本思想是对 k 个不同分组训练的结果进行平均来减少方差，因此模型的性能对数据的划分并敏感。

1.2 k 折交叉验证（K-Fold Cross Validation）

- **第一步**：不重复抽样将原始数据随机分为 k 份（通常是平均划分）。
- **第二步**：每一次挑选其中 1 份作为测试集，剩余 k-1 份作为训练集用于模型训练。
- **第三步**：重复第二步 k 次，这样每个子集都有一次机会作为测试集，其余机会作为训练集。
  - 在每个训练集上训练后得到一个模型
  - 用模型在相应的测试集上测试，计算并保存模型的评估结果
- **第四步**：计算 k 组测试结果的**平均值**作为模型精度的估计，并作为当前 k 折交叉验证下模型的**性能指标**。

1.2 k 折交叉验证（K-Fold Cross Validation）

在 k 折交叉验证中，k 的一个典型设置时 = 10，称之为 十**折交叉验证。**

- 当数据量**较小**时，k 可以设置大一些，这样训练集占整体比例就比较大，不过同时训练得到的模型个数也会增多，需要更多的训练时间
- 当数据量**较大**时，k 可以设置稍小一些，因为大数据常常已经足够均衡样本类别的不平衡

1.2 k 折交叉验证（K-Fold Cross Validation）

- **三种实现Kfold的方法：**
  - 直接使用KFold实现
  - 配合cross_val_score的默认参数cv=5实现KFold
  - 配合cross_val_score的自定义参数cv=kfold实现KFold（该方法同样适用于其他交叉验证法）

1.3 留一法（Leave One Out）

留一法是一种极端的**折交叉验证法**，在每次训练中，它只保留一个样本用于验证，其他样本均参与训练。整个过程中，需要将所有的样本都依次迭代一遍，也就是说所有的样本都会被单独作为验证数据去参与训练。

这种方法可以让模型最终的评估结果更可靠，但是也会增加训练的复杂程度，因为构建的模型的数量与原始样本相同。通常情况下，留一法只在数据非常少，即缺乏数据的时候使用。

我们也可以将一换成 P 生成**留P法**，即每次保留P个样本作为验证数据，其他作为训练数据。

1.3 留一法（Leave One Out）

- **留一法的优点：**
  - 每一回合中几乎所有的样本皆用于训练模型，因此最接近原始样本的分布，这样评估所得的结果比较可靠。
  - 实验过程中没有随机因素会影响实验数据，确保实验过程是可以被复制的。
- **留一法的缺点：**
  - 计算成本高
  - 需要建立的模型数量与原始数据样本数量相同。
  - 当数据集较大时几乎不能使用。

1.4 随机采样（ShuffleSplit）

随机采样（ShuffleSplit）算法和最基本的数据集分割方法train_test_split基本一样，都是从原始数据集中随机选出一部分数据作为训练集，另外一部分作为测试集。它们具有以下特性：

- 都具有随机打乱数据的优点。
- 在数据集较大而系统性能不足，或者模型简单时，可以通过设置train_size + test_size < 1，来实现部分采样。
- 通过random_state种子参数，可以用来控制每次采样都是随机或者是伪随机

1.5 分层采样法（Stratification）

前面我们提到，在使用留出法的时候有一个弊端。那就是分出来的验证集的分类可能会存在严重的类别不平衡现象。这是因为随机选择出来的验证集中的样本类别本身就是不可控的。这个问题在 *k* 折交叉验证中同样存在。因此，我们需要一种办法来实现划分后的验证集的类别的平衡。换句话说，需要实现训练集和验证集具有相同的类别分布。

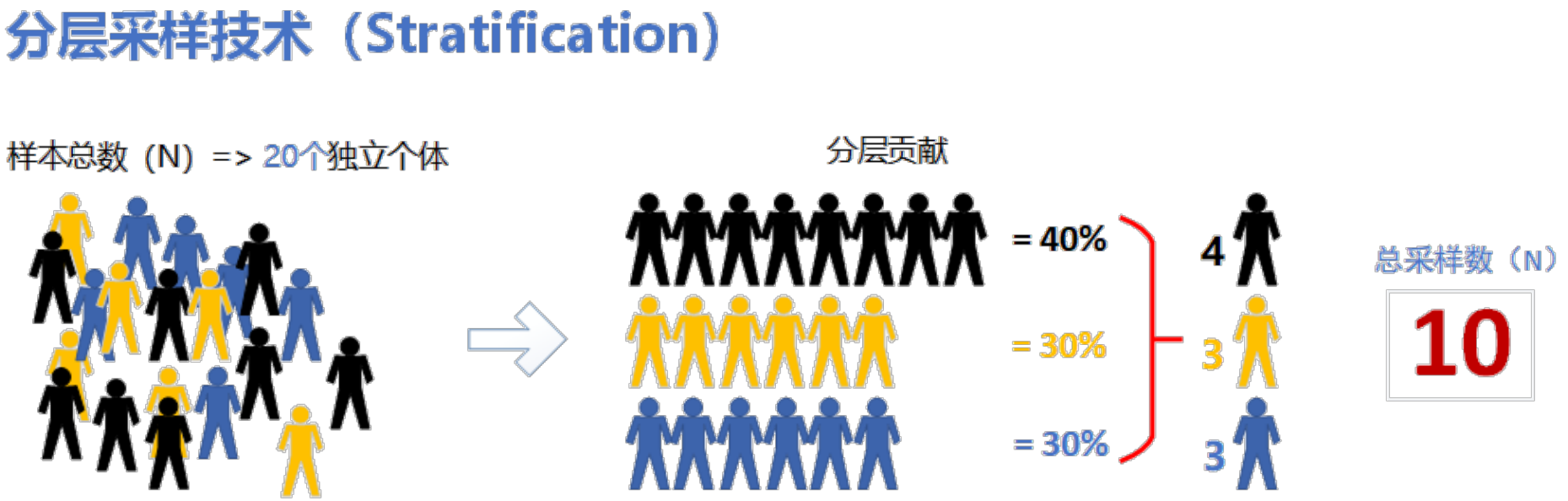
> **此处，为什么不说测试集也应该具有相同的类别分布呢?**

> **因为，测试集对于我们来说，是不可知的。**

1.5 分层采样法（Stratification）

分层采样法不是一种分类算法，而是一种思想，它可以被应用到其他交叉验证算法中，形成具有**分层功能**的方法，例如：StratifiedKFold, StratifiedShuffleSplit, train_test_split(with stratify).

通过分层法，当我们在分割数据时，我们可以在不同的划分区域中获得相似的目标分布，如下图所示。



1.5 分层采样法（Stratification）

分层采样法在多分类问题中比较有效，特别是相对较小，而且数据分类**不平衡**的数据集。在**大数据集**中效果相对不那么明显，因为在大数据环境下随机采样通常都能获得较平均的结果。所以，对于一个类别平衡的大型数据集，分层划分法和简单的随机划分基本一样。

- **分层采样法的优点：**

  在样本不平衡的数据集中，能有效解决不平衡问题

- **分层采样法的缺点：**

  对于平衡数据集，效果不明显，但也没有明显的缺点（但在某些情况下，可能会影响样本较多的分类的效果）

1.6 如何选择交叉验证法?

如何选择?

- 对于大规模数据集，优先考虑**留出法**。因为在大规模数据集中，一方面大量的数据避免了欠拟合的问题；另一方面，样本的随机划分通常都能相对均衡，这就避免了抽样不均衡带来的模型训练偏差，即减少了模型的方差。
- 对于中小型数据集，优先 **k 折交叉法**，通过平均误差的计算，确保了样本因为抽样不平衡带来的模型方差较大的问题。
- 对于样本规模非常小的数据集，可以考虑**留一法**，这种方法优先解决欠拟合问题。
- 对于样本不平衡的数据集，无论是大规模数据集还是中小规模数据集，**分层采样法**都是较好的选择。

2. 使用网格搜索优化模型参数

几乎所有的机器学习/深度学习算法都有超参数，超参数的设置对于获得**最优模型**具有决定性作用。超参数的选择方法：

- 手动测试法
- 经验值法
- 启发式搜索
- **网格搜索**
- 试错法
- 随机搜索
- 遗传算法

网格搜索法是指定参数值的一种穷举搜索方法，通过手动的给出一个模型中你想要改动的所有超参数，并将估计函数的参数通过交叉验证的方法进行优化来得到**最优**的学习算法。整个过程由程序自动的使用穷举法来将所有的参数都运行一遍。

3. 分类模型的可行度评估

在机器学习的任务中，并不是所有任务都像抛硬币一样能给出非常清晰的0，1分界面，很多时候可能面临的是**模棱两可**的状态。

- 例如，看到天阴，或者狂风大作，但是并不代表就一定会下雨，只能说这种天气下极有可能会下雨。没错，“极有可能”在数学中，我们称之为有**较大概率**。
- 再例如，小张准备去买一辆汽车，他买大众的概率是0.3，买别克的概率是0.4，买奔驰的概率是0.1，买法拉利的概率是0.2。

可见，对于多分类任务，通常也可以采用概率来评价。

通常**概率值较高**的类别通常会被判为最终的分类。例如，二分类任务中的"下雨"，多分类任务中的"买别克"。

本章小结

至此，我们一致都在采用score来对模型进行评价：

- 在**分类**任务中，它代表的是 Accuracy准确度这个评价指标；
- 在**回归**任务中代表的是R^2 可决系数，即回归平方和 与 总变差 之间的商。

在实际任务中还有许多很重要的评价指标，如精确度(Precision），召回率（Recall），F1分数（F1-Score），ROC曲线（Receiver Operation Characteristic Curve），PR曲线（精度和召回率的相关曲线），AUC（Area Under Curve）等。

在使用网格搜索的时候，我们也可以使用这些评价指标。

本章小结

值得注意的是，无论是模型评估还是参数调节都是数据科学家必备的知识之一。

- 对于**模型评估**，不同的指标适合不同的任务，选择什么样的指标需要根据任务的目录来决定。
- 对于**（超）参数调节**，几乎是所有模型都需要的，如何获得最优的超参数，直接关系到模型的性能与最终目标的成败。

欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Tel：18687840023