第3章 特征工程与模型评估

业界广泛流传这样一句话：数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已。那么，什么是特征工程？这里引用维基百科对特征工程（feature engineering）的定义：特征工程是利用数据领域的相关知识创建能够使机器学习算法达到最佳性能的特征的过程。也就是说，特征工程尽可能从原始数据中获取更多信息，从而使得预测模型达到最佳。简而言之，特征工程是一个把原始数据变成特征的过程，这些特征可以很好地描述数据，并且利用这些特征建立的模型在未知数据上表现性能可以达到最优。

在建立数据模型后，只有选择与问题相匹配的评估方法，才能快速地发现模型选择或训练过程中出现的问题，迭代地对模型进行优化。针对分类、排序、回归、序列预测等不同类型的机器学习问题，评估指标的选择也有所不同。知道每种评估指标的精确定义，有针对性地选择合适的评估指标，根据评估指标的反馈进行模型调整，这些都是机器学习在模型评估阶段的关键问题。

本章的主要内容为特征工程和模型评估，前者有利于建立更精确的模型，而后者则有利于对模型进行不断的调整和优化，从而使得构建的模型能够针对样本以外的数据也能做到高精度分析或预测。

3.1 特征工程

特征工程包含数据预处理、特征构建、特征选择和特征提取等子问题。

3.1.1 数据预处理

现实世界中的数据大体上都是不完整、不一致的“脏”数据，无法直接进行数据分析，或者模型的训练结果差强人意。为了提高特征的质量，往往在构建特征之前，需要对数据进行预处理。常用的数据类型有两种：

（1）结构化数据。可以看作关系数据库的一张表。每一列都有清晰的定义，包含了数值型和类别型两种基本类型；每一行表示一个样本的信息。

（2）非结构化数据。主要是文本、图像、音频和视频数据，其包含的信息无法用一个简单的数值表示，也没有清晰的类别定义，并且各个数据的大小也互不相同。

数据预处理针对的主要是缺失值和异常值。

1. 缺失值

1）缺失值的产生及影响

数据的缺失主要包括记录的缺失和记录中某个字段信息的缺失，两者都会造成分析结果的不准确。

缺失值产生的原因如下：

（1）信息暂时无法获取，或者获取信息的代价太大。

（2）信息被遗漏，包括人为的输入遗漏或者数据采集设备的遗漏。

（3）属性不存在。在某些情况下，缺失值并不意味着数据有错误。对一些对象来说，某些属性值是不存在的，如未婚者的配偶姓名、儿童的固定收入等。

缺失值的影响如下：

（1）训练模型将丢失大量的有用信息。

（2）训练模型表现出的不确定性更加显著，模型中蕴含的规律更难把握。

（3）包含空值的数据会使建模过程陷入混乱，导致不可靠的输出。

2）缺失值的处理

缺失值的处理主要采用删除法和填补法。

删除法包括以下 3 种方法：

（1）删除样本（行删除）。将存在缺失数据的样本行删除，从而得到一个完备的数据集。

（2）删除变量（列删除）。当某个变量缺失值较多且对研究目标影响不大时，可以将缺失值所在列的数据全部删除。

（3）改变权重。当删除缺失数据会改变数据结构时，通过对数据赋予不同的权重，可以降低缺失数据带来的偏差。

填补法包括单一填补、随机填补和基于模型预测 3 种方法。

（1）单一填补。

① 均值填补。对于非数值型特征，采用众数填补；对于数值型特征，采用所有对象的均值填补。

② 热平台填补。在完整数据中找到一个与空值最相似的值，然后用这个相似的值进行填补。

③ 冷平台填补。从过去的调查数据中获取信息进行填补。

④ k 均值填补。利用辅助变量，定义样本间的距离函数，寻找与包含缺失值的样本距离最近的无缺失值的 k 个样本，利用这 k 个样本的加权平均值填补缺失值。

⑤ 期望最大化。在缺失类型为随机缺失的情况下，假设模型对于完整的样本是正确的，那么通过观测数据的边际分布可以对未知参数进行极大似然估计。

（2）随机填补。在均值填补的基础上加上随机项，通过增强缺失值的随机性改善缺失值分布过于集中的缺陷。

（3）基于模型预测。将缺失属性作为预测目标，使用其余属性作为自变量，利用缺

失属性的非缺失样本构建分类模型或回归模型，使用构建的模型预测缺失属性的缺失值。

2. 异常值

模型通常是对整体样本数据结构的一种表达方式，这种表达方式通常抓住的是整体样本的一般性质，而那些在一般性质上表现完全与整体样本不一致的点就称为异常值，也称离群点。通常异常值在预测问题中是不受开发者欢迎的，因为预测问题通常关注的是整体样本的性质，而异常值的生成机制与整体样本完全不一致。如果算法对异常值敏感，那么生成的模型并不能较好地表达整体样本，从而预测也会不准确。

1）异常值的检测方法

异常值的检测方法有以下 5 种。

（1）3σ 法。

使用该方法的前提是数据需要服从正态分布，在正态分布中 σ 代表标准差，μ 代表均值，$x=\mu$ 为分布图的对称轴。采用 3σ 法时，观测值与均值的差别如果超过 3 倍标准差，那么就可以将其视为异常值。观测值落在$[-3\sigma,3\sigma]$区间的概率是 99.7%，那么距离均值超过 3σ 的值出现的概率为 $P(|x-\mu|>3\sigma)=0.3\%$，属于小概率事件。如果数据不服从正态分布，也可以用远离均值的多少倍标准差描述，倍数的取值需要根据经验和实际情况决定。

（2）箱形图。

箱形图是用于显示一组数据分散情况的统计图，如图 3-1 所示。这种方法利用箱形图的四分位距（Inter-Quartile Range，IQR）对异常值进行检测，四分位距就是上四分位与下四分位的差值。箱形图法以 IQR 的 1.5 倍为标准，规定位于上四分位＋1.5 倍 IQR（上界）以上或者下四分位－1.5 倍 IQR（下界）以下的点为异常值。

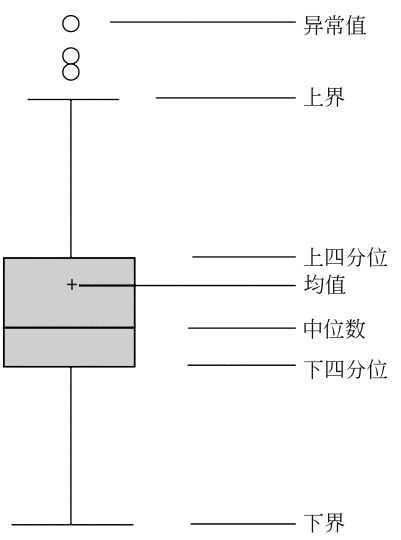

图 3-1　箱形图

（3）基于模型的检测。

构建一个概率分布模型，并计算对象符合该模型的概率，把具有低概率的对象视为异常值。如果模型是簇的集合，则异常值是不显著属于任何簇的对象；如果模型是回归模型，异常值是相对远离预测值的对象。

（4）基于密度的检测。

从基于密度的观点来说，异常值是在低密度区域中的对象。基于密度的异常值检测与基于邻近度的异常值检测密切相关，因为密度通常用邻近度定义。一种常用的密度定义是：密度为到 k 个最近邻的平均距离的倒数。如果该距离小，则密度高；否则密度低。另一种密度定义是 DBSCAN 聚类算法使用的密度定义，即一个对象周围的密度等于在该对象指定距离 d 内的对象个数。

(5) 基于聚类的检测。

从基于聚类的观点来说,如果一个对象不强属于任何簇,那么该对象属于异常值。

2) 异常值的处理方法

检测到异常值后,需要对其进行一定的处理。异常值的处理方法可大致分为以下几种:

(1) 删除含有异常值的记录。

(2) 将异常值视为缺失值,利用缺失值处理方法处理异常值。

(3) 用前后两个观测值的平均值修正异常值。

(4) 不处理异常值,直接在含有异常值的数据集上进行数据挖掘。

是否删除异常值可根据实际情况考虑。一些模型对异常值不敏感,即使有异常值也不影响模型效果;但是一些模型(例如逻辑回归算法)对异常值很敏感,如果不进行处理,可能会出现过拟合等非常差的效果。

3.1.2 特征构建

为了提高数据分析的质量,需要将原始数据转换为计算机可以理解的格式或符合数据分析要求的格式。通过转换函数将数据转换成更适合机器学习算法模型的特征数据的过程称为特征构建,特征构建包括数值型特征无量纲化、数值型特征分箱、统计变换和特征编码等方法。

1. 数值型特征无量纲化

数据一般是有单位的,数据样本不同属性具有不同量级时会对模型训练结果造成影响,量级的差异将导致量级较大的属性占据主导地位,使迭代收敛速度减慢,同时,依赖于样本距离的算法对数据的量级非常敏感,量级的差异可能造成训练结果不准确。因此,在进行模型训练之前需要将数据无量纲化。特征的无量纲化是使不同规格的数据转换到同一规格,常用的无量纲化方法为标准化方法。

数据的标准化(normalization)是将数据按比例缩放,使之落入一个小的特定区间。在对某些比较和评价的指标进行处理时,经常会去掉数据的单位,将其转换为无量纲的纯数值,便于不同单位或量级的指标能够进行比较和加权。其中最典型数据的标准化就是归一化处理,即将数据统一映射到[0,1]区间。常见的数据标准化方法有 min-max 标准化、对数函数转换、反正切函数转换、z-score 标准化等。

1) min-max 标准化

min-max 标准化也叫离差标准化,是对原始数据的线性变换,使结果落到[0,1]区间,转换函数如下:

$$y_i = \frac{x_i - \min\limits_{1\leqslant i\leqslant n}\{x_j\}}{\max\limits_{1\leqslant i\leqslant n}\{x_j\} - \min\limits_{1\leqslant i\leqslant n}\{x_j\}} \tag{3-1}$$

其中,max 为样本数据的最大值,min 为样本数据的最小值。

这种方法有一个缺陷,当有新数据加入时,可能导致 max 和 min 的变化,需要重新定义转换函数。

2）对数函数转换

通过以 10 为底的对数函数转换的方法可以实现归一化，把一系列数值转换到区间[0,1]，转换函数如下：

$$x^* = \frac{\log_{10} x}{\log_{10} \max} \tag{3-2}$$

3）反正切函数转换

用反正切函数也可以实现数据的归一化。使用这个方法时需要注意，如果映射的目标区间为[0,1]，则数据都应该大于或等于 0，小于 0 的数据将被映射到[−1,0]区间。

4）z-score 标准化

并非所有数据标准化的结果都映射到[0,1]区间。还有一种常见的标准化方法是 z-score 标准化；它也是 SPSS 中最常用的标准化方法，也叫标准差标准化。

z-score 标准化方法适用于属性的最大值和最小值未知或有超出取值范围的离群数据的情况。

z-score 标准化的步骤如下：

（1）求出各变量（指标）的算术平均值（数学期望）x_i 和标准差 s_i。

（2）进行标准化处理：

$$z_{ij} = (x_{ij} - x_i)/s_i \tag{3-3}$$

其中，z_{ij} 为标准化后的变量值，x_{ij} 为实际变量值。

（3）将逆指标前的正负号对调。正指标指的是越大越好的指标，例如产量、销售收入；逆指标指的是越小越好的指标，例如单位成本、原材料耗用率等。逆指标正负号对调的目的是使逆指标正向化，以便统一分析。

z-score 标准化后的变量值围绕 0 上下波动，大于 0 说明高于平均水平，小于 0 说明低于平均水平。

2. 数值型特征分箱

离散化是对数值型特征非常重要的一个处理，也就是将数值型数据转换成类别型数据。连续值的取值空间可能是无穷的，为了便于表示和在模型中处理，需要对连续值特征进行离散化处理，称为数值型特征分箱。常用的分箱法如下。

1）无监督分箱法

无监督分箱法包括以下 5 种方法。

（1）自定义分箱。根据业务经验或者常识等自行划分区间，然后将原始数据归类到各个区间中。

（2）等距分箱。按照相同宽度将数据分成几等份。

将最小值到最大值这一区间均分为 N 等份，如果 A 为最小值，B 为最大值，则每个区间的长度为 $W=(B-A)/N$，则区间边界值为 $A+W$，$A+2W$，…，$A+(N-1)W$。这里只考虑边界，每个区间里面的实例数量可能不等。

（3）等频分箱。

将数据分成几等份，每一份中的数据个数是一样的。边界值要经过选择，使得每一份包含大致相等的实例数量。例如，$N=10$，每一份应该包含大约 10%的实例。

(4) 基于 k 均值聚类分箱。

k 均值聚类法将观测值聚为 k 类。在聚类过程中需要保证类的有序性：第一类中的所有观测值都要小于第二类中的所有观测值，第二类中的所有观测值都要小于第三类中的所有观测值，以此类推。

(5) 二值化。

二值化可以将数值型特征进行阈值化，得到布尔型数据，这对于下游的概率估计来说可能很有用(例如数据分布为伯努利分布时)。

2) 有监督分箱法

有监督分箱法包括以下两种方法。

(1) 卡方分箱法。

卡方分箱法是自底向上(即基于合并)的数据离散化方法。它依赖于卡方检验：具有最小卡方值的相邻区间合并在一起，直到满足确定的停止准则。

对于精确的离散化，类分布在一个区间内应当完全一致。因此，如果两个相邻的区间具有非常类似的类分布，则这两个区间可以合并；否则，应当保持分开。而低卡方值表明这两个区间具有相似的类分布。

(2) 最小熵分箱法。

最小熵分箱法需要使总熵值达到最小，也就是使分箱能够最大限度地区分因变量的各类别。

熵是信息论中数据无序程度的度量标准，提出信息熵的基本目的是找出某种符号系统的信息量和冗余度之间的关系，以便能用最小的成本和消耗实现最高效率的数据存储、管理和传递。

数据集的熵越低，说明数据之间的差异越小，最小熵划分就是为了使每个分箱中的数据具有最大的相似性。给定分箱的个数，如果考虑所有可能的分箱情况，最小熵分箱法得到的分箱应该是具有最小熵的分箱。

3. 统计变换

数据分布倾斜有很多负面的影响。可以使用特征工程技巧，利用统计或数学变换减轻数据分布倾斜的影响，使原本密集的区间中的值尽可能分散，原本分散的区间中的值尽可能聚合。

统计变换中使用的变换函数都属于幂变换函数簇，通常用来创建单调的数据变换。其主要作用在于稳定方差，始终保持分布接近于正态分布，并使得数据与分布的平均值无关。

统计变换主要有以下两种方法：

(1) 对数变换。

(2) Box-Cox 变换。它有一个前提条件，即数值型数据必须先变换为正数(与对数变换的要求一样)。如果数值是负的，可以使用一个常数对数值进行偏移。

4. 分类特征编码

在统计学中，分类特征是可以采用有限且通常固定数量的可能值之一的变量，基于某

些定性属性将每个个体或其他观察单元分配给特定组或名义类别。

分类特征编码有以下 4 种方法：

(1) 标签编码。对不连续的数字或者文本进行编码，编码值介于 0 和 $n-1$ 之间的标签。其中 n 是标签中的类别数。

(2) 独热编码。用于将表示分类的数据扩维。对它最简单的理解就是它与位图类似，设置一个个数与类型数量相同的全 0 数组，每一位对应一个类型。如果该位为 1，表示属于该类型。需要注意的是，独热编码只能将数值型变量二值化，无法直接对字符串型的类别变量编码。

(3) 标签二值化。其功能与独热编码一样，但是独热编码只能将数值型变量二值化，无法直接对字符串型变量编码，而标签二值化可以直接将字符型变量二值化。

(4) 平均数编码。是针对高基数类别特征的有监督编码。当一个类别特征列包括了极多不同类别(如家庭地址，动辄上万)时，可以采用此编码方法。它是一种基于贝叶斯架构，利用要预测的因变量，有监督地确定最适合这个定性特征的编码方式。在 Kaggle 的数据竞赛中，平均数编码是一种常见的提高分数的手段。

3.1.3　特征选择

特征选择(feature selection)也称特征子集选择(Feature Subset Selection，FSS)或属性选择(attribute selection)，是指从已有的 M 个特征(feature)中选择 N 个特征，使得系统的特定指标最优化。特征选择是从原始特征中选择一些最有效的特征以降低数据集维度的过程，是提高机器学习算法性能的一个重要手段，也是模式识别中关键的数据预处理步骤。对于一个机器学习算法来说，好的学习样本是训练模型的关键。

当特征维度超过一定界限后，分类器的性能会随着特征维度的增加而下降，而且维度越高，训练模型的时间开销也会越大。导致分类器性能下降的原因往往是由于这些高维度特征中包含大量无关特征和冗余特征。

当特征数量较多时，需要判断哪些是相关特征，哪些是不相关特征。因而，特征选择的难点在于：其本质是一个复杂的组合优化问题。

当构建模型时，假设拥有 N 维特征，每个特征有两种可能的状态：保留和剔除。那么这组状态集合中的元素个数就是 2^N。如果使用穷举法，其时间复杂度为 $O(2^N)$。假设 N 为 10，如果穷举所有特征集合，需要进行 1024 次尝试，这显然需要耗费较长的时间和较多的计算资源。

在实际的工程应用中，需要选择有意义的特征输入机器学习算法和模型进行训练。通常从两方面考虑特征的选择：

(1) 特征是否发散。如果一个特征不发散，例如方差接近于 0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。

(2) 特征与目标的相关性。与目标相关性高的特征应当优选选择。常规特征选择算法包括基于相似度的方法、基于信息理论的方法、基于稀疏学习的方法、基于统计的方法。

根据特征选择的形式，特征选择方法大致可分为 3 类：过滤式(filter)、包裹式(wrapper)和嵌入式(embedding)。

1. 过滤式特征选择

过滤式方法先对数据集进行特征选择，然后再训练模型，特征选择过程与后续模型无关。通过对每一维特征打分，即给每一维特征赋予权重，就能以权重代表该维特征的重要性，然后依据权重排序，即按照特征的发散性或者相关性指标对各个特征进行评分，设定评分阈值或待选择阈值的个数，选择合适的特征。

过滤式特征选择包括以下 4 种方法：

(1) 方差选择法。通过判断其特征是否发散进行特征选择。方差大的特征，可以认为它是比较有用的；如果方差较小，例如小于 1，那么这个特征可能对算法作用没有那么大。最极端的情况是：如果某个特征方差为 0，即所有的样本该特征的取值都是一样的，那么它对模型训练没有任何作用，可以直接舍弃。

(2) 相关系数法。主要用于输出连续值的监督学习算法，分别计算所有训练集中各个特征与输出值之间的相关系数，设定一个阈值，选择相关系数较大的部分特征。

(3) 卡方检验。卡方检验最基本的思想就是通过观察实际值与理论值的偏差来确定理论的正确与否。具体实现方法是：通常先假设两个变量确实是相互独立的(原假设)，然后观察实际值(观察值)与理论值(指“如果两者确实相互独立”的情况下应该有的值)的偏差程度。如果偏差足够小，则认为偏差是很自然的样本误差，是测量不够精确导致的或者偶然产生的，两者确实是相互独立的，可以接受原假设；如果偏差大到一定程度，使得这样的偏差不太可能是测量不精确或者偶然产生所致，那么认为两者实际上是相关的，即否定原假设，而接受备选假设(即两个变量不是相互独立的)。

(4) 信息增益/互信息法。通常作为优先考虑的方法，即从信息熵的角度分析各特征和输出值之间的关系的评分，称之为互信息值。互信息值越大，说明该特征和输出值之间的相关性越大，越需要保留。

2. 包裹式特征选择

与过滤式特征选择不考虑后续模型不同，包裹式特征选择直接把最终将要使用的模型的性能作为特征集的评价准则。换言之，包裹式特征选择的目的就是为给定模型选择最有利于其性能的特征子集。

一般而言，由于包裹式特征选择方法直接针对给定模型进行优化，因此从最终模型性能来看，包裹式特征选择比过滤式特征选择更好。但另一方面，由于在特征选择过程中需多次训练模型，因此包裹式特征选择的计算开销通常比过滤式特征选择大得多。

包裹式特征选择包括以下两种方法：

(1) 递归特征消除。其主要思想是：反复构建模型，然后选出其中贡献最小的特征并将之剔除，然后在剩余特征上继续重复这个过程，直到所有特征都已遍历。这是一种后向搜索方法，采用了贪心法则，而特征被剔除的顺序即是它的重要性排序。为了增强稳定性，这里模型评估常采用交叉验证的方法。

(2) LVW 特征选择算法。LVW(Las Vegas Wrapper，Las Vegas 包裹)算法基于 Las Vegas 算法的框架，Las Vegas 算法是一个典型的随机化方法，即概率算法中的一种。它具有概率算法的特点，允许算法在执行的过程中随机选择下一步。许多情况下，当算法

在执行过程中面临一个选择时，随机性选择常比最优选择更省时，因此概率算法可在很大程度上降低算法的复杂度。LVW 算法在 Las Vegas 算法的基础上，每次从特征集中随机产生一个特征子集，然后使用交叉验证的方法估计学习器在特征子集上的误差，若该误差小于之前获得的最小误差，或者与之前的最小误差相当，但特征子集中包含的特征数更少，则将特征子集保留下来。在 LVW 算法中，以最终分类器的误差作为特征子集评价标准。

3. 嵌入式特征选择

在过滤式和包裹式特征选择方法中，特征选择过程与模型训练过程有明显的区别。而嵌入式特征选择是将特征选择过程与模型训练过程融为一体，两者在同一个优化过程中完成，即在模型训练过程中自动进行特征选择。首先使用某些机器学习算法和模型进行训练，得到各特征的权值系数，根据权值系数从大到小的顺序选择特征。嵌入式特征选择与过滤法类似，但嵌入式特征选择是通过机器学习训练确定特征的优劣，而不是直接从特征的统计学指标确定特征优劣。其主要思想是：在模型既定的情况下学习出对提高模型准确性最好的属性，即在确定模型的过程中挑选那些对模型的训练有重要意义的属性。嵌入式特征选择常用的方法包括基于惩罚项的特征选择法和基于树模型的特征选择法。

3.1.4　特征提取

特征提取的主要目的是降维。特征提取的主要思想是将原始样本投影到一个低维特征空间，得到最能反映样本本质或进行样本区分的低维样本特征，即特征提取的过程也是特征降维的过程。

1. 特征降维简介

1）特征降维的作用

特征降维的作用如下：

（1）降低时间复杂度和空间复杂度。

（2）节省提取不必要特征的开销。

（3）去掉数据集中夹杂的噪声。

（4）较简单的模型在小数据集上有更强的鲁棒性。

（5）当数据能用较少的特征进行解释时，可以更好地解释数据，便于提取知识。

（6）便于实现数据的可视化。

2）特征降维的目的

特征降维的主要目的是用来进行特征选择和特征提取。在特征提取中，要找到 k 个新的维度的集合，这些维度是原来 k 个维度的组合，这个方法可以是监督的，如线性判别式分析（Linear Discriminant Analysis，LDA）；也可以是非监督的，如主成分分析算法（Principal Components Analysis，PCA）。

3）子集选择

对于 n 个属性，有 2^n 个可能的子集。通过穷举搜索找出属性的最佳子集可能是不现实的，特别是当 n 和数据类的数目较大时。通常使用压缩搜索空间的启发式算法，这些

方法基本上是典型的贪心算法。在搜索特征空间时，总是做看上去是最佳的选择，其策略是局部最优选择，期望由此求出全局最优解。在实践中，这种贪心方法是有效的，并可以逼近最优解。

4）降维的本质

从一个维度空间映射到另一个维度空间，即学习一个映射函数 $f: \boldsymbol{x} \rightarrow \boldsymbol{y}$（$\boldsymbol{x}$ 是原始数据点的向量表达，$\boldsymbol{y}$ 是数据点映射后的低维向量表达。）f 可能是显式的或隐式的、线性的或非线性的。

2. 特征提取方法

特征提取方法主要有以下两种。

1）主成分分析

将样本投影到某一维上。其坐标的选择方式为：首先找到第一个坐标，使数据集在该坐标的方差最大（也就是在这个数据维度上能更好地区分不同类型的数据）；然后找到第二个坐标，使该坐标与原来的坐标正交。该过程一直重复，直到新坐标的个数和原来的特征个数相同，这时候就会发现数据的大部分方差都在前面几个坐标上表示出来了，这些新的维度就是主成分。

主成分分析（Principal Component Analysis，PCA）的基本思想是：寻找数据的主轴方向，由主轴构成一个新的坐标系，这里的维数可以比原维数低，然后数据由原坐标系向新坐标系投影，这个投影的过程就是降维的过程。

PCA 算法的过程如下：

（1）将原始数据中的每一个样本都用向量表示，把所有样本组合起来构成样本矩阵。通常对样本矩阵进行中心化处理，得到中心化样本矩阵。

（2）求中心化后的样本矩阵的协方差。

（3）求协方差矩阵的特征值和特征向量。

（4）将求出的特征值按从大到小的顺序排列，并将其对应的特征向量按照此顺序组合成一个映射矩阵，根据指定的 PCA 保留的特征个数取出映射矩阵的前 n 行或者前 n 列作为最终的映射矩阵。

（5）用映射矩阵对数据进行映射，达到数据降维的目的。

2）线性判别式分析

线性判别式分析（Linear discriminant analysis，LDA）也称为费希尔（Fisher）线性判别，是模式识别中的经典算法。它是一种监督学习的降维技术，其数据集的每个样本是有类别输出的。

该算法的思想是：投影后，类内距离最小，类间距离最大。

线性判别是将高维的模式样本投影到最佳鉴别向量空间，以达到抽取分类信息和压缩特征空间维数的效果，投影后保证模式样本在新的子空间有最大的类间距离和最小的类内距离。LDA 是一种有效的特征提取方法。使用此方法能够使投影后模式样本的类间散布矩阵最大，同时类内散布矩阵最小。

LDA 与 PCA 的共同点如下：

(1) 都属于线性方法。

(2) 在降维时都采用矩阵分解的方法。

(3) 都假设数据符合高斯分布。

两者的不同点如下：

(1) LDA 是有监督的。

(2) LDA 不能保证投影到的坐标是正交的(根据类别的标注，关注分类能力)。

(3) LDA 降维直接与类别的个数有关，与数据本身的维度无关(原始数据是 n 维的，有 c 个类别，降维后一般是 $c-1$ 维)。

(4) LDA 既可以用于降维，也可用于分类。

(5) LDA 可以选择分类性能最好的投影方向。

3.2 模型评估

3.2.1 模型误差

1. 训练误差和测试误差

机器学习的目的是使学习到的模型不仅对已知数据有很好的预测能力，而且对未知数据也有很好的预测能力。不同的学习方法会训练出不同的模型，不同的模型可能会对未知数据做出不同的预测。所以，如何评价模型好坏，并选择出好的模型是重点。通常会使用错误率、误差等对模型进行评价。

错误率指的是分类错误的样本占总体样本的比例。误差指的是样本的实际结果与预测输出的差别。误差根据训练的样本集可以分为 3 类：训练误差、测试误差和泛化误差。例如，对于分类学习算法，一般将样本集分为训练集和测试集，其中，训练集用于算法模型的学习或训练，而测试集通常用于评估训练好的模型对于数据的预测性能。它们的先后顺序是：先将算法在训练集上训练得到一个模型，然后在测试集上评估模型的性能。

由于测试学习算法是否成功的标准在于算法对于训练中未见过的数据的预测执行能力，因此一般将分类模型的误差分为训练误差(training error)和泛化误差(generalization error)。训练误差也叫经验误差，是指模型在训练集上的错分样本比率，即在训练集上训练完成后，在训练集上进行预测得到的错分率。同样，测试误差就是模型在测试集上的错分率。泛化误差是指模型在未知记录上的期望误差，也就是对在训练集中从未出现过的数据进行预测的错分样本比率。在划分样本集时，如果划分的训练集与测试集的数据没有交集，此时测试误差基本等同于泛化误差。实际在训练和测试模型的时候，往往不能事先拿到新样本的特征，也就是不能控制泛化误差，所以只能努力使训练误差最小化。

2. 过拟合与欠拟合

过拟合(overfitting)与欠拟合(underfitting)是统计学中的现象。过拟合是由于在统计模型中使用的参数过多而导致模型对观测数据(训练数据)过度拟合，以至于用该模型预测其他测试样本输出的时候与实际输出或者期望值相差很大的现象。欠拟合则刚好相

反，是由于统计模型使用的参数过少，以至于得到的模型难以拟合观测数据（训练数据）的现象。

从直观上理解，欠拟合就是模型拟合不够，在训练集上表现效果差，没有充分利用数据，预测的准确度低，具体体现在没有学习到数据的特征，还有待继续学习。而过拟合则是模型过度拟合，在训练集上表现好，但是在测试集上效果差。也就是说，在已知的数据集合中非常好，但学习进行得太彻底，以至于把数据的一些局部特征或者噪声带来的特征都学习了。但是，在添加一些新的数据进来后训练效果就会差很多，所以在进行测试时泛化误差也不佳。造成过拟合的原因是考虑的影响因素太多，大大超出自变量的维度。统计模型的 3 类评估结果如图 3-2 所示。

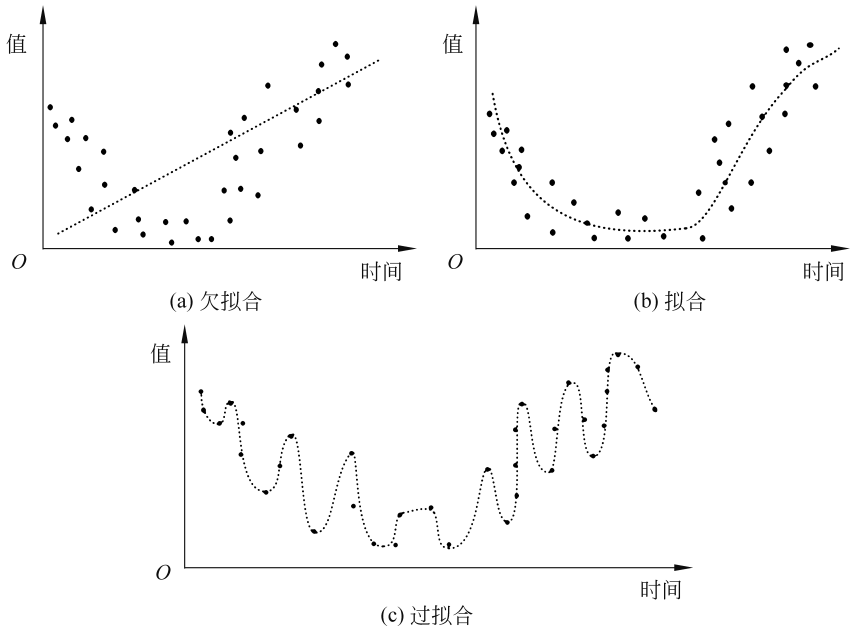

图 3-2　统计模型的 3 类评估结果

3. 偏差和方差

偏差（bias）是模型在样本上的输出与真实值之间的误差，它反映模型的精确度。

方差（variance）是模型每一次输出结果与模型输出期望之间的误差，它反映模型的稳定性。

如图 3-3 所示，偏差指的是模型的输出值与红色中心的距离，而方差指的是模型的每一个输出结果与期望的距离。

以射箭为例说明这两个概念。低偏差指的是瞄准的点与红色中心的距离很近，而高偏差指的是瞄准的点与红色中心的距离很远。低方差是指当瞄准一个点后，射出的箭击中靶子的位置与瞄准的点的距离比较近；高方差是指当瞄准一个点后，射出的箭击中靶子的位置与瞄准的点的距离比较远。

- 低偏差低方差是追求的效果，此时预测值非常接近靶心，且比较集中。

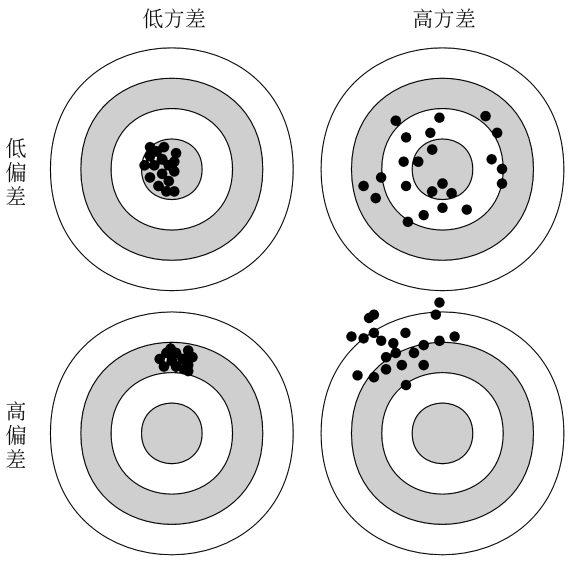

图 3-3　偏差和方差

- 低偏差高方差时，预测值基本落在真实值周围，但值很分散，方差较大。这说明模型的稳定性不够好。
- 高偏差低方差时，预测值与真实值有较大距离，但值很集中，方差小。这说明模型的稳定性较好，但预测准确率不高。
- 高偏差高方差是最不希望看到的结果，此时模型不仅预测不准确，而且不稳定，每次预测的值都差别比较大。

偏差和方差是统计学的概念，首先要明确，方差是多个模型间的比较，而非对一个模型而言的。对于单独的一个模型，偏差可以是单个数据集中的，也可以是多个数据集中的。方差和偏差的变化一般是和模型的复杂程度成正比的，当一味地追求模型精确匹配时，则可能会导致同一组数据训练出不同的模型，模型之间的差异非常大，称为方差，不过偏差就很小了。

图 3-4 中的圆点和三角点表示一个数据集中采样得到的不同的子数据集，有两个 N 次曲线拟合这两个点集，则可以得到两条曲线。两条曲线的差异很大，但是它们本是由同一个数据集生成的，这是由于模型复杂造成方差过大。模型越复杂，偏差就越小；而模型越简单，偏差就越大。方差和偏差是按图 3-5 所示的方式变化的。

方差和偏差两条变化曲线交叉的点对应的模型复杂度就是最佳的模型复杂度。用数学公式描述偏差和方差的关系如下：

$$E(L)=\int(y-h(x))^2p(x)\mathrm{d}x+\int(h(x)-t)^2p(x,t)\mathrm{d}x\,\mathrm{d}t \tag{3-4}$$

其中，$E(L)$是损失函数，$h(x)$表示真实值的平均，等号右边第一部分是与 y（模型的估计函数）有关的，这部分表示选择不同的估计函数（模型）带来的差异；而第二部分是与 y 无关的，这部分可以认为是模型的固有噪声。式(3-4)等号右边的第一部分可以化成下面的形式：

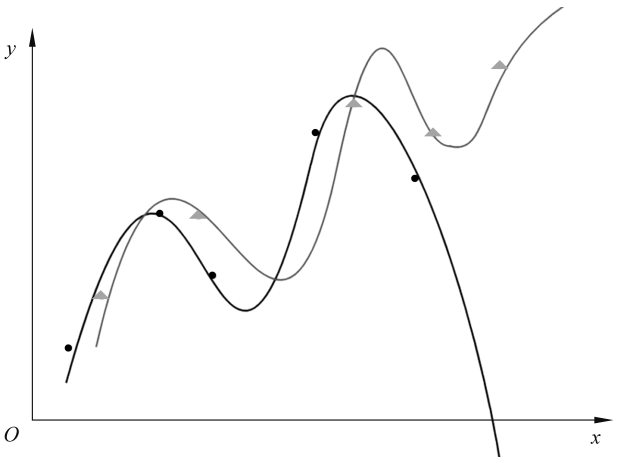

图 3-4　方差与偏差的关系

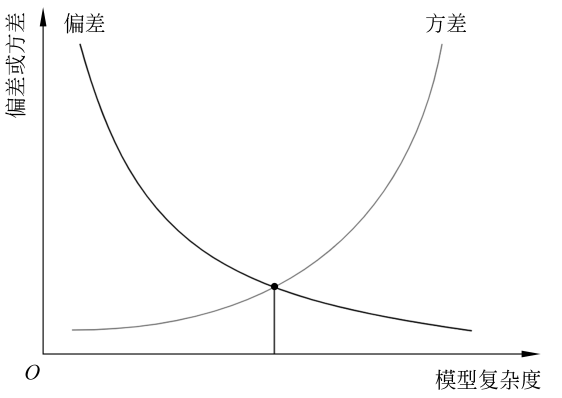

图 3-5　方差、偏差与模型复杂度

$$E_D(y(x;D)-h(x))=(E_D(y(x;D)-h(x)))^2+E_D((y(x;D)-E_D(y(x;D)))^2) \tag{3-5}$$

其中，D 是数据集，E_D 是在训练集 D 上对测试样本 x 的期望预测。在式(3-5)等号右边的部分中，加号前面的部分表示偏差，后面的部分表示方差。

3.2.2　模型评估方法

在建模过程中，存在偏差过大导致模型欠拟合以及方差过大导致模型过拟合这两种情况。为了解决这两个问题，需要一整套评估方法及评价指标，其中，评估方法用于评估模型的泛化能力，而评价指标则用于评价单个模型的性能。

在模型评估中，经常要将数据集划分为训练集和测试集。数据集划分通常要保证两个条件：

(1) 训练集和测试集的分布要与样本真实分布一致，即训练集和测试集都要保证是从样本真实分布中独立同分布采样而得的。

(2) 训练集和测试集要互斥,即两个子集没有交集。

基于划分方式的不同,评估方法可以分为留出法、交叉验证法及自助法。

(1) 留出法(hold-out)是直接将数据集划分为两个互斥的子集,其中一个子集作为训练集,另一个子集作为测试集。另外需要注意的是,在划分的时候要尽可能保证数据分布的一致性,避免因数据划分过程引入额外的偏差而对最终结果产生影响。而当数据明显分为有限类时,可以采用分层抽样方式选择测试数据,以保证数据分布比例的平衡。

(2) 交叉验证法(cross validation)先将数据集 D 划分为 k 个大小相似的互斥子集,可以分层划分,这样可以保证数据分布尽可能保持一致。每次用 $k-1$ 个子集作为训练集,剩下一个子集作为测试集,依次可以进行 k 次训练和测试,最后获得的是这 k 次的均值,又称为 k 折交叉验证。训练集中的样本数量必须足够多,一般应大于 50%。

(3) 自助法(bootstraping)是有放回的采样或重复采样的评估方法。在含有 m 个样本的数据集中,每次随机挑选一个样本,将其作为训练样本,再将此样本放回数据集中,这样有放回地抽样 m 次,生成一个与原数据集大小相同的新数据集,这个新数据集就是训练集。这样,有些样本可能在训练集中出现多次,每个样本被选中的概率是 $1/m$,因此未被选中的概率就是 $1-1/m$,一个样本在训练集中不出现的概率就是 m 次都未被选中的概率,即 $(1-1/m)^m$。当 m 趋于无穷大时,这一概率就趋近 $e^{-1}\approx 0.368$,所以留在训练集中的样本大概占原数据集的 63.2%。将未出现在训练集中的数据作为测试集。

3.2.3　模型评价指标

机器学习模型需要一些评价指标以判断它的好坏。了解各种评价指标,在实际应用中选择正确的评价指标,是十分重要的。以下对于常见的分类模型和聚类模型的评价指标进行介绍。

1. 分类模型的评价指标

1) 错误率与精度

最常用的两种评价指标既适用于分类模型,也适用于聚类模型。对于样例集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$,分类的错误率定义为

$$E(f;D)=1/m\sum_{i=1}^{m}//(f(x_i\neq y_i)) \tag{3-6}$$

精度的定义为

$$\mathrm{acc}(f;D)=1/m\sum_{i=1}^{m}//(f(x_i=y_i))=1-E(f;D) \tag{3-7}$$

在式(3-6)和式(3-7)中,"//(*)"是指示函数,括号中的内容"*"若符合则记为 1,否则记为 0。

更一般地,对于数据分布 D 和概率密度函数 $p(\cdot)$,错误率和精度可以分别描述为

$$E(f;D)=\int_{x\sim D}\|(f(x)!=y)\cdot p(x)\mathrm{d}x \tag{3-8}$$

$$\mathrm{acc}(f;D)=\int_{x\sim D}\|(f(x)=y)\cdot p(x)\mathrm{d}x=1-E(f;D) \tag{3-9}$$

在式(3-8)和式(3-9)中,“‖(*)”是指示函数,括号中的内容“*”若符合则记为1,否则记为0。

2) 混淆矩阵

混淆矩阵如表3-1所示。

表3-1 混淆矩阵

实　际	预　测	
	正	负
正	TP	FN
负	FP	TN

在表3-1中:

- TP(True Positives,真正):表示实际属于某一类,预测也属于该类的样本数。
- TN(True Negatives,真负):表示实际属于某一类,而预测不属于该类的样本数。
- FP(False Positives,假正):表示实际不属于某一类,而预测属于该类的样本数。
- FN(False Negatives,假负):表示实际不属于某一类,预测也不属于该类的样本数。

根据混淆矩阵,有以下7个评价指标:

(1) 正确率:

$$\text{Accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{3-10}$$

(2) 错误率:

$$\text{Error Rate}=1-\text{Accuracy} \tag{3-11}$$

(3) 查准率:

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{3-12}$$

(4) 召回率:

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3-13}$$

(5) F 值。查准率越高越好,召回率越高越好。在实际情况中,不会有分类器仅仅以查准率或者召回率作为单一的度量标准,而是使用这两者的调和平均,于是就有了 F 值(F-score)。

$$F_1=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{3-14}$$

(6) 敏感度:

$$\text{Sensitivity}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3-15}$$

(7) 准确度:

$$\text{Specificity}=\frac{\text{TN}}{\text{TN}+\text{FP}} \tag{3-16}$$

3) ROC 曲线与 AUC

ROC 全称是 Receiver Operating Characteristic(受试者工作特征)。ROC 曲线下的面积简称 AUC(Area Under the Curve)。AUC 用于衡量二分类问题机器学习算法的性能。分类阈值是判断样本为正例的阈值(threshold)。例如,当预测概率 $P(y=1|x)\geqslant$ threshold(常取 threshold=0.5) 或预测分数 Score≥threshold(常取 threshold=0)时,则判定样本为正例。

在 ROC 曲线和 AUC 中,使用以下两个评价指标:

(1) 真正例率(True Positive Rate,TPR):所有正例中,有多少被正确地判定为正。

(2) 假正例率(False Positive Rate,FPR):所有负例中,有多少被错误地判定为正。

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3-17}$$

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}} \tag{3-18}$$

在 ROC 曲线中,以 FPR 为横坐标,以 TPR 为纵坐标。ROC 曲线的绘制步骤如下:

(1) 假设已经得出一系列样本被划分为正类的概率,即 Score 值,将它们按照大小排序。

(2) 从高到低依次将每个 Score 值作为阈值,当测试样本属于正例的概率大于或等于这个阈值时,认为它为正例,否则为负例。举例来说,对于某个样本,其 Score 值为 0.6,那么 Score 值大于或等于 0.6 的样本都被认为是正例,而其他样本则都被认为是负例。

(3) 根据每次选取的不同的阈值,得到一组 FPR 和 TPR。以 FPR 值为横坐标,以 TPR 值为纵坐标,即确定 ROC 曲线上的一点。

(4) 根据(3)中的每个坐标点画出 ROC 曲线。

ROC 曲线如图 3-6 所示。

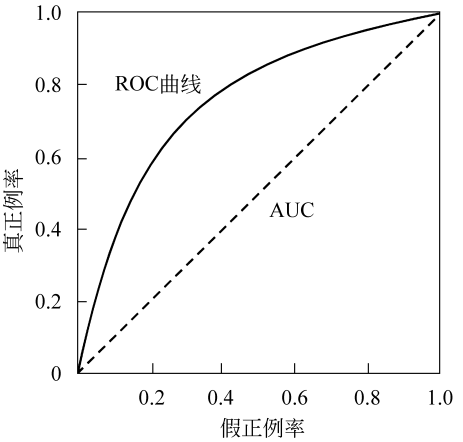

图 3-6　ROC 曲线图

利用 ROC 曲线很容易看出任意阈值对学习器的泛化性能的影响,有助于选择最佳的阈值。ROC 曲线越靠近左上角,模型的查全率就越高。最靠近左上角的 ROC 曲线上

的点是分类错误最少的最好的阈值，其假正例和假负例总数最少。可以比较不同的学习器的性能，将各个学习器的 ROC 曲线绘制到同一坐标中，直观地鉴别优劣，靠近左上角的 ROC 曲线代表的学习器准确性最高。ROC 曲线距离左上角越近，证明学习器效果越好。如果算法 1 的 ROC 曲线完全包含算法 2 的 ROC 曲线，则可以断定算法 1 的性能优于算法 2。

AUC 被定义为 ROC 曲线与横坐标轴之间的面积。如果两条 ROC 曲线没有相交，可以根据哪条 ROC 曲线最靠近左上角判定哪条 ROC 曲线代表的学习器性能最好。但是，在实际任务中，情况往往很复杂，如果两条 ROC 曲线发生了交叉，则很难简单地断言谁优谁劣。在很多实际应用中，往往希望把学习器性能分出高低。为此又引入了 AUC 的概念。AUC 是衡量二分类模型优劣的一种评价指标，表示预测的正例排在负例前面的概率。学习器的 AUC 值越高，其区分正例和负例的能力就越好。

2. 聚类模型的评价指标

聚类是一种无监督学习算法，训练样本的标记未知，按照某个标准或数据的内在性质及规律，将样本划分为若干不相交的子集。聚类性能度量也称有效性指标，分为两种：一是外部指标，即聚类完成后将聚类结果与某个参考模型进行比较时使用的指标；二是内部指标，即直接考察聚类结果而不利用任何参考模型时使用的指标。

1）外部指标

外部指标将聚类结果得到的类标签和已知类标签进行比较，此评价的前提是数据集样本的类标签已知。常用的评价指标如下。

（1）混淆矩阵。

如果能够得到类标签，那么聚类结果也可以像分类那样使用混淆矩阵的一些指标——查准率、回召率和 F 值作为评价指标。

（2）均一性、完整性以及 V-度量。

若一个簇中只包含一个类别的样本，则满足均一性(homogeneity)。其实也可以认为均一性就是正确率。它用每个簇中正确分类的样本数占该簇总样本数的比例之和表示：

$$p=\frac{1}{k}\sum_{i=1}^{k}\frac{N(C_i=K_i)}{N(K_i)} \tag{3-19}$$

其中，k 是簇的个数，C_i 是第 i 个簇中的样本的类别，K_i 是第 i 个簇的类别，N 表示总样本数，$N(K_i)$是第 i 个簇中的样本个数。

若同类别样本被归类到相同簇中，则满足完整性(completeness)。它用每个簇中正确分类的样本数占该类的总样本数比例之和表示：

$$r=\frac{1}{k}\sum_{i=1}^{k}\frac{N(C_i=K_i)}{N(C_i)} \tag{3-20}$$

单独考虑均一性或完整性都是片面的，因此引入两个指标的加权平均，称为 V-度量(V-measure)。

$$V_\beta=\frac{(1+\beta^2)pr}{\beta^2 p+r} \tag{3-21}$$

其中，β 为加权参数，p 和 r 分别为均一性和完整性。设置 $\beta>1$ 则更注重完整性，否则更

注重均一性。

(3) 杰卡德相似系数。

杰卡德相似系数(Jaccard similarity coefficient)为集合的交集与并集的比值,取值为[0,1],值越大,相似度越高。两个集合 A 和 B 的交集中的元素在 A、B 的并集中所占的比例称为这两个集合的杰卡德相似系数,用符号 $J(A,B)$ 表示。

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|} \tag{3-22}$$

与杰卡德相似系数相反的概念是杰卡德距离(Jaccard distance)。杰卡德距离等于两个集合中的不同元素占所有元素的比例,用于衡量两个集合的区分度,可以用式(3-23)表示:

$$J_\delta=1-J(A,B)=\frac{J(A\cup B)-|A\cap B|}{|A\cup B|} \tag{3-23}$$

杰卡德距离用于描述集合之间的不相似度,距离越大,相似度越低。

(4) 兰德指数与调整兰德指数。

兰德指数(Rand Index,RI)计算样本预测值与真实值之间的相似度,RI 取值范围是[0,1],值越大,意味着聚类结果与真实情况越吻合。

$$\mathrm{RI}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}=\frac{a+b}{C_m^2} \tag{3-24}$$

其中,C 表示实际类别信息,K 表示聚类结果,a 表示在 C 与 K 中是同一类别的元素对数,b 表示在 C 与 K 中是不同类别的元素对数,由于每个样本对仅能出现在一个集合中,因此 TP+FP+TN+FN=$C_m^2=m(m-1)/2$ 表示数据集中可以组成的样本对数。对于随机结果,RI 并不能保证接近 0,因此具有更高区分度的调整兰德指数(Adjusted Rand Index,ARI)被提出,其取值范围是[−1,1],值越大,表示聚类结果和真实情况越吻合。

$$\mathrm{ARI}=\frac{\mathrm{RI}-E(\mathrm{RI})}{\max(\mathrm{RI})-E(\mathrm{RI})} \tag{3-25}$$

其中,$E(\mathrm{RI})$ 表示期望值。

2) 内部指标

内部指标不需要数据集样本的类标签,直接根据聚类结果中簇内的相似度和簇间的分离度进行聚类质量的评估。常用的内部指标有以下 3 个。

(1) Dunn 指数。

Dunn 指数(Dunn Index,DI)是两个簇的簇间最短距离除以任意簇中的最大距离,DI 越大,说明聚类效果越好。DI 对环状分布的数据评价效果不好,而对离散点的聚类评价效果很好。C_i、C_j 表示两个簇,x_i、x_j 表示对应簇中的两个样本,$\mathrm{dist}(x_i,x_j)$ 表示这两个样本之间的距离,k 表示类别数。

$$\begin{aligned}
&d_{\min}(C_i,C_j)=\min_{x_i\in C_j,x_j\in C_j}\mathrm{dist}(x_i,x_j)\\
&\mathrm{diam}(C)=\max_{1\leqslant i<j\leqslant|C|}\mathrm{dist}(x_i,x_j)\\
&\mathrm{DI}=\min_{1\leqslant i\leqslant k}\left(\min_{i\neq j}\left(\frac{d_{\min}(C_i,C_j)}{\max\limits_{1\leqslant l\leqslant k}\mathrm{diam}(C_l)}\right)\right)
\end{aligned} \tag{3-26}$$

(2) 距离。

两个向量的相似程度可以用它们之间的距离表示。距离近,则相似程度高;距离远,则相似程度低。用距离度量数据时,需先做均一化处理。常用的距离有如下 3 种,其中,x_i、y_i 表示两个样本点,n 表示样本数,p 表示维数。

① 闵可夫斯基(Minkowski)距离:

$$\mathrm{dist}(X,Y)=\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{\frac{1}{p}} \tag{3-27}$$

② 当 $p=1$ 时为曼哈顿(Manhattan)距离:

$$\mathrm{M\_dist}(X,Y)=\sum_{i=1}^{n}|x_i-y_i| \tag{3-28}$$

③ 当 $p=2$ 时为欧几里得(Euclidean)距离:

$$\mathrm{E\_dist}(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \tag{3-29}$$

(3) 轮廓系数。

将簇 C 中的样本 i 到同簇中的其他样本的平均距离记为 a_i。a_i 越小,表示样本越应该被聚类到该簇。簇 C 中的所有样本的 a_i 的均值被称为簇 C 的簇内不相似度(compactness)。

将簇 C 中的样本 i 到另一个簇 C_j 中的所有样本的平均距离记为 b_{ij}。b_{ij} 越大,表示样本 i 越不属于簇 C_j。簇 C 中的所有样本的 b_{ij} 的均值被称为簇 C 与簇 C_j 的簇间不相似度(separation)。

轮廓系数(silhouette coefficient):s_i 值为[−1,1],越接近 1,表示样本 i 聚类越合理;越接近−1,表示样本 i 越应该被分类到其他簇中;越接近 0,表示样本 i 越应该在簇的边界上。所有样本的 s_i 的均值被称为聚类结果的轮廓系数。

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{3-30}$$

其中,$a(i)$是簇内不相似度,即样本 i 到同簇内其他点不相似程度的平均值,它体现了凝聚度;$b(i)$是簇间不相似度,即样本 i 到其他簇的平均不相似程度的最小值,它体现了分离度。

式(3-30)也可以写为

$$s(i)=\begin{cases}1-\dfrac{a(i)}{b(i)}, & a(i)<b(i)\\ 0, & a(i)=b(i)\\ \dfrac{b(i)}{a(i)}-1, & a(i)>b(i)\end{cases} \tag{3-31}$$

3. 回归模型的评价

在回归学习任务中,对于给定样例集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$,也有一些评估指标,下面分别进行介绍。