

挖掘建模

分类与预测

分类和预测是预测问题的两种主要类型，分类主要是预测分类标号（离散属性），而预测主要是建立连续值函数模型，预测给定自变量对应的因变量的值。

实现过程

● 分类是构造一个分类模型，输入样本的属性值，输出对应的类别，将每个样本映射到预先定义好的类别。

分类模型建立在已有类标记的数据集上，模型在已有样本上的准确率可以方便的计算，所以分类属于有监督的学习。

● 预测时建立在两种或两种以上变量间相互依赖的函数模型，然后进行预测与控制。

● 分类：第一步时学习步，通过归纳分析训练样本集来建立分类模型得到分类规则；第二步时分类步，先用已知的测试样本集评估分类规则的准确率，如果准确率时可以接受的，则使用该模型对未知类标号的待测试样本集进行预测。

预测第一步是通过训练集建立预测属性的函数模型，第二步在模型通过检验后进行预测和控制。

常用的分类与预测算法

算 法 名 称	算 法 描 述
回归分析	回归分析是确定预测属性（数值型）与其他变量间相互依赖的定量关系最常用的统计学方法。包括线性回归、非线性回归、Logistic 回归、岭回归、主成分回归、偏最小二乘回归等模型
决策树	决策树采用自顶向下的递归方式，在内部节点进行属性值的比较，并根据不同的属性值从该节点向下分支，最终得到的叶节点是学习划分的类
人工神经网络	人工神经网络是一种模仿大脑神经网络结构和功能而建立的信息处理系统，表示神经网络的输入与输出变量之间关系的模型
贝叶斯网络	贝叶斯网络又称信度网络，是 Bayes 方法的扩展，是目前不确定知识表达和推理领域最有效的理论模型之一
支持向量机	支持向量机是一种通过某种非线性映射，把低维的非线性可分转化为高维的线性可分，在高维空间进行线性分析的算法

回归分析

回归模型名称	适 用 条 件	算 法 描 述
线性回归	因变量与自变量是线性关系	对一个或多个自变量和因变量之间的线性关系进行建模，可用最小二乘法求解模型系数
非线性回归	因变量与自变量之间不都是线性关系	对一个或多个自变量和因变量之间的非线性关系进行建模。如果非线性关系可以通过简单的函数变换转化成线性关系，用线性回归的思想求解；如果不能转化，用非线性最小二乘方法求解
Logistic 回归	因变量一般有 1 和 0(是否)两种取值	是广义线性回归模型的特例，利用 Logistic 函数将因变量的取值范围控制在 0 和 1 之间，表示取值为 1 的概率

岭回归	参与建模的自变量之间具有多重共线性	是一种改进最小二乘估计的方法
主成分回归	参与建模的自变量之间具有多重共线性	主成分回归是根据主成分分析的思想提出来的，是对最小二乘法的一种改进，它是参数估计的一种有偏估计。可以消除自变量之间的多重共线性

决策树

决策树是一树状结构，它的每一个叶节点对应着一个分类，非叶节点对应着在某个属性上的划分，根据样本在该属性上的不同取值将其划分成若干个子集。对于非纯的叶节点，多数类的标号给出到达这个节点的样本所属的类。构造决策树的核心问题是在每一步如何选择适当的属性对样本做拆分。对一个分类问题，从已知类标记的训练样本中学习并构造出决策树是一个自上而下，分而治之的过程。

决策树算法	算法描述
ID3 算法	其核心是在决策树的各级节点上，使用信息增益方法作为属性的选择标准，来帮助确定生成每个节点时所应采用的合适属性
C4.5 算法	C4.5 决策树生成算法相对于 ID3 算法的重要改进是使用信息增益率来选择节点属性。C4.5 算法可以克服 ID3 算法存在的不足：ID3 算法只适用于离散的描述属性，而 C4.5 算法既能够处理离散的描述属性，也可以处理连续的描述属性
CART 算法	CART 决策树是一种十分有效的非参数分类和回归方法，通过构建树、修剪树、评估树来构建一个二叉树。当终结点是连续变量时，该树为回归树；当终结点是分类变量，该树为分类树

1. ID3 算法简介及基本原理

ID3 算法基于信息熵来选择最佳测试属性。它选择当前样本集中具有最大信息增益值的属性作为测试属性；样本集的划分则依据测试属性的取值进行，测试属性有多少不同取值就将样本集划分为多少子样本集，同时决策树上相应于该样本集的节点长出新的叶子节点。ID3 算法根据信息论理论，采用划分后样本集的不确定性作为衡量划分好坏的标准，用信息增益值度量不确定性：信息增益值越大，不确定性越小。因此，ID3 算法在每个非叶节点选择信息增益最大的属性作为测试属性，这样可以得到当前情况下最纯的拆分，从而得到较小的决策树。

设 S 是 s 个数据样本的集合。假定类别属性具有 m 个不同的值： $C_i (i = 1, 2, \dots, m)$ 。设 s_i 是类 C_i 中的样本数。对一个给定的样本，它总的信息熵为

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m P_i \log_2(P_i) \quad (5-5)$$

其中， P_i 是任意样本属于 C_i 的概率，一般可以用 $\frac{s_i}{s}$ 估计。

设一个属性 A 具有 k 个不同的值 $\{a_1, a_2, \dots, a_k\}$ ，利用属性 A 将集合 S 划分为个子集 $\{S_1, S_2, \dots, S_k\}$ ，其中 S_j 包含了集合 S 中属性 A 取 a_j 值的样本。若选择属性 A 为测试属性，则这些子集就是从集合 S 的节点生长出来的新的叶节点。设 s_{ij} 是子集 S_j 中类别为 C_i 的样本数，则

根据属性 A 划分样本的信息熵值为

$$E(A) = \sum_{j=1}^k \frac{s_{1j} + s_{2j} + \cdots + s_{mj}}{s} I(s_{1j}, s_{2j}, \cdots, s_{mj}) \quad (5-6)$$

其中, $I(s_{1j}, s_{2j}, \cdots, s_{mj}) = - \sum_{i=1}^m P_{ij} \log_2(P_{ij})$, $P_{ij} = \frac{s_{ij}}{s_{1j} + s_{2j} + \cdots + s_{mj}}$ 是子集 S_j 中类别为 C_i 的样本的概率。

最后, 用属性 A 划分样本集 S 后所得的信息增益 (Gain) 为

$$Gain(A) = I(s_1, s_2, \cdots, s_m) - E(A) \quad (5-7)$$

显然 $E(A)$ 越小, $Gain(A)$ 的值越大, 说明选择测试属性 A 对于分类提供的信息越大, 选择 A 之后对分类的不确定程度越小。属性 A 的 k 个不同的值对应样本集 S 的 k 个子集或分支, 通过递归调用上述过程 (不包括已经选择的属性), 生成其他属性作为节点的子节点和分支来生成整个决策树。ID3 决策树算法作为一个典型的决策树学习算法, 其核心是在决策树的各级节点上都用信息增益作为判断标准进行属性的选择, 使得在每个非叶节点上进行测试时, 都能获得最大的类别分类增益, 使分类后数据集的熵最小。这样的处理方法使得树的平均深度较小, 从而有效地提高了分类效率。

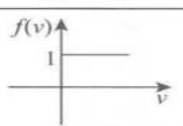
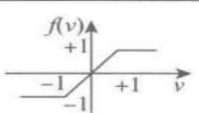
2. ID3 算法具体流程

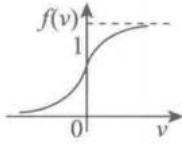
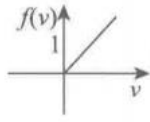
ID3 算法的具体详细实现步骤如下。

- 1) 对当前样本集合, 计算所有属性的信息增益;
- 2) 选择信息增益最大的属性作为测试属性, 把测试属性取值相同的样本划为同一个子样本集;
- 3) 若子样本集的类别属性只含有单个属性, 则分支为叶子节点, 判断其属性值并标上相应的符号, 然后返回调用处; 否则对子样本集递归调用本算法。

由于 ID3 决策树算法采用了信息增益作为选择测试属性的标准, 会偏向于选择取值较多的, 即所谓高度分支属性, 而这类属性并不一定是最优的属性。同时 ID3 决策树算法只能处理离散属性, 对于连续型的属性, 在分类前需要对其进行离散化。为了解决倾向于选择高度分支属性的问题, 人们采用信息增益率作为选择测试属性的标准, 这样便得到 C4.5 决策树算法。此外, 常用的决策树算法还有 CART 算法、SLIQ 算法、SPRINT 算法和 PUBLIC 算法等。

人工神经网络

激活函数	表达形式	图 形	解释说明
域值函数 (阶梯函数)	$f(v) = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases}$		当函数的自变量小于 0 时, 函数的输出为 0; 当函数的自变量大于或等于 0 时, 函数的输出为 1, 用该函数可以把输入分成两类
分段线性函数	$f(v) = \begin{cases} 1, & v \geq 1 \\ v, & -1 < v < 1 \\ -1, & v \leq -1 \end{cases}$		该函数在 (-1, +1) 线性区内的放大系数是一致的, 这种形式的激活函数可以看作是非线性放大器的近似

非线性转移函数	$f(v) = \frac{1}{1 + e^{-v}}$		单极性 S 型函数为实数域 \mathbf{R} 到 $[0, 1]$ 闭集的连续函数，代表了连续状态型神经元模型。其特点是函数本身及其导数都是连续的，能够体现数学计算上的优越性
Relu 函数	$f(v) = \begin{cases} v, & v \geq 0 \\ 0, & v < 0 \end{cases}$		这是近年来提出的激活函数，它具有计算简单、效果更佳的特点，目前已经有取代其他激活函数的趋势。本书的神经网络模型大量使用了该激活函数

人工神经网络的学习也称为训练，指的是神经网络在受到外部环境的刺激下调整神经网络的参数，使神经网络以一种新的方式对外部环境作出反应的一个过程。在分类与预测中，人工神经网络主要使用有指导的学习方式，即根据给定的训练样本，调整人工神经网络的参数以使网络输出接近于已知的样本类标记或其他形式的因变量。

在人工神经网络的发展过程中，提出了多种不同的学习规则，没有一种特定的学习算法适用于所有的网络结构和具体问题。在分类与预测中， δ 学习规则（误差校正学习算法）是使用最广泛的一种。误差校正学习算法根据神经网络的输出误差对神经元的连接强度进行修正，属于有指导学习。设神经网络中神经元 i 作为输入，神经元 j 为输出神经元，它们的连接权值为 w_{ij} ，则对权值的修正为 $\Delta w_{ij} = \eta \delta_j Y_i$ ，其中 η 为学习率， $\delta_j = T_j - Y_j$ 为 j 的偏差，即输出神经元 j 的实际输出和教师信号之差，示意图如图 5-7 所示。

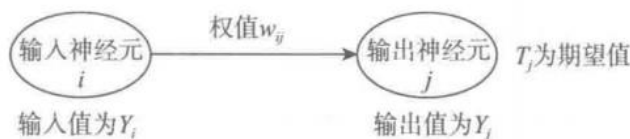


图 5-7 δ 学习规则示意图

神经网络训练是否完成常用误差函数（也称目标函数） E 来衡量。当误差函数小于某一个设定的值时即停止神经网络的训练。

误差函数为衡量实际输出向量 Y_k 与期望值向量 T_k 误差大小的函数，常采用二乘误差函数来

定义为 $E = \frac{1}{2} \sum_{k=1}^N [Y_k - T_k]^2$ （或 $E = \sum_{k=1}^N [Y_k - T_k]^2$ ） $k = 1, 2, \dots, N$ 为训练样本个数。

算法名称	算法描述
BP 神经网络	是一种按误差逆传播算法训练的多层前馈网络，学习算法是 δ 学习规则，是目前应用最广泛的神经网络模型之一
LM 神经网络	是基于梯度下降法和牛顿法结合的多层前馈网络，特点：迭代次数少，收敛速度快，精确度高
RBF 径向基神经网络	RBF 网络能够以任意精度逼近任意连续函数，从输入层到隐含层的变换是非线性的，而从隐含层到输出层的变换是线性的，特别适合于解决分类问题
FNN 模糊神经网络	FNN 模糊神经网络是具有模糊权系数或者输入信号是模糊量的神经网络，是模糊系统与神经网络相结合的产物，它汇聚了神经网络与模糊系统的优点，集联想、识别、自适应及模糊信息处理于一体
GMDH 神经网络	GMDH 网络也称为多项式网络，它是前馈神经网络中常用的一种用于预测的神经网络。它的特点是网络结构不固定，而且在训练过程中不断改变
ANFIS 自适应神经网络	神经网络镶嵌在一个全部模糊的结构之中，在不知不觉中向训练数据学习，自动产生、修正并高度概括出最佳的输入与输出变量的隶属函数以及模糊规则；另外，神经网络的各层结构与参数也都具有了明确的、易于理解的物理意义

BP神经网络的学习算法是 δ 学习规则，目标函数采用 $E = \sum_{k=1}^n [Y_k - T_k]^2$ ，下面详细介绍BP神经网络算法。

反向传播（Back Propagation, BP）算法的特征是利用输出后的误差来估计输出层的直接前导层的误差，再用这个误差估计更前一层的误差，如此一层一层的反向传播下去，就获得了所有其他各层的误差估计。这样就形成了将输出层表现出的误差沿着与输入传送相反的方向逐级向网络的输入层传递的过程。

分类与预测算法评价

分类与预测模型对训练集进行预测而得出的准确率并不能很好地反映预测模型未来的性能，为了有效判断一个预测模型的性能表现，需要一组没有参与预测模型建立的数据集，并在该数据集上评价预测模型的准确率，这组独立的数据集叫作测试集。模型预测效果评价，通常用相对/绝对误差、平均绝对误差、均方误差、均方根误差等指标来衡量。

（1）绝对误差与相对误差

设 Y 表示实际值， \hat{Y} 表示预测值，则称 E 为绝对误差（Absolute Error），计算公式如下。

$$E = Y - \hat{Y} \quad (5-8)$$

e 为相对误差（Relative Error），计算公式如下。

$$e = \frac{Y - \hat{Y}}{Y} \quad (5-9)$$

有时相对误差也用百分数表示。

$$e = \frac{Y - \hat{Y}}{Y} * 100\% \quad (5-10)$$

这是一种直观的误差表示方法。

（2）平均绝对误差

平均绝对误差（Mean Absolute Error, MAE）定义如下。

$$MAE = \frac{1}{n} \sum_{i=1}^n |E_i| = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (5-11)$$

式中各项的含义如下。

- MAE : 平均绝对误差。
- E_i : 第 i 个实际值与预测值的绝对误差。
- Y_i : 第 i 个实际值。
- \hat{Y}_i : 第 i 个预测值。

由于预测误差有正有负，为了避免正负相抵消，故取误差的绝对值进行综合并取其平均数，这是误差分析的综合指标法之一。

（3）均方误差

均方误差（Mean Squared Error, MSE）定义如下。

$$MSE = \frac{1}{n} \sum_{i=1}^n E_i^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (5-12)$$

在上式中， MSE 表示均方差，其他符号同前。

本方法用于还原平方失真程度。

均方误差是预测误差平方之和的平均数，它避免了正负误差不能相加的问题。由于对误差 E 进行了平方，加强了数值大的误差在指标中的作用，从而提高了这个指标的灵敏性，是

一大优点。均方误差是误差分析的综合指标法之一。

(4) 均方根误差

均方根误差 (Root Mean Squared Error, RMSE) 定义如下。

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n E_i^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (5-13)$$

上式中, $RMSE$ 表示均方根误差, 其他符号同前。

这是均方误差的平方根, 代表了预测值的离散程度, 也称为标准误差, 最佳拟合情况为 $RMSE = 0$ 。均方根误差也是误差分析的综合指标之一。

(5) 平均绝对百分误差

平均绝对百分误差 (Mean Absolute Percentage Error, MAPE) 定义如下。

$$MAPE = \frac{1}{n} \sum_{i=1}^n |E_i/Y_i| = \frac{1}{n} \sum_{i=1}^n |(Y_i - \hat{Y}_i)/Y_i| \quad (5-14)$$

上式中, $MAPE$ 表示平均绝对百分误差。一般认为 $MAPE$ 小于 10 时, 预测精度较高。

(6) Kappa 统计

Kappa 统计是比较两个或多个观测者对同一事物, 或观测者对同一事物的两次或多次观测结果是否一致, 以由于机遇造成的一致性和实际观测的一致性之间的差别大小作为评价基础的统计指标。Kappa 统计量和加权 Kappa 统计量不仅可以用于无序和有序分类变量资料的一致性、重现性检验, 而且能给出一个反映一致性大小的“量”值。

Kappa 取值在 $[-1, +1]$ 之间, 其值的大小均有不同意义。

- ☐ Kappa = +1 说明两次判断的结果完全一致。
- ☐ Kappa = -1 说明两次判断的结果完全不一致。
- ☐ Kappa = 0 说明两次判断的结果是机遇造成。
- ☐ Kappa < 0 说明一致程度比机遇造成的还差, 两次检查结果很不一致, 在实际应用中无意义。
- ☐ Kappa > 0 此时说明有意义, Kappa 越大, 说明一致性越好。
- ☐ Kappa ≥ 0.75 说明已经取得相当满意的一致程度。
- ☐ Kappa < 0.4 说明一致程度不够。

(7) 识别准确度

识别准确度 (Accuracy) 定义如下。

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \times 100\% \quad (5-15)$$

式中各项说明如下。

- ☐ TP (True Positives): 正确的肯定表示正确肯定的分类数。
- ☐ TN (True Negatives): 正确的否定表示正确否定的分类数。
- ☐ FP (False Positives): 错误的肯定表示错误肯定的分类数。
- ☐ FN (False Negatives): 错误的否定表示错误否定的分类数。

（8）识别精确率

识别精确率（Precision）定义如下。

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (5-16)$$

（9）反馈率

反馈率（Recall）定义如下。

$$Recall = \frac{TP}{TP + TN} \times 100\% \quad (5-17)$$

（10）ROC 曲线

受试者工作特性（Receiver Operating Characteristic, ROC）曲线是一种非常有效的模型评价方法，可为选定临界值给出定量提示。将灵敏度（Sensitivity）设在纵轴，1-特异性（1-Specificity）设在横轴，就可得出 ROC 曲线图。该曲线下的积分面积（Area）大小与每种方法优劣密切相关，反映分类器正确分类的统计概率，其值越接近 1 说明该算法效果越好。

Python 分类预测模型特点

模 型	模 型 特 点	位 于
逻辑回归	比较基础的线性分类模型，很多时候是简单有效的选择	sklearn.linear_model
SVM	强大的模型，可以用来回归、预测、分类等，而根据选取不同的核函数。模型可以是线性的 / 非线性的	sklearn.svm
决策树	基于“分类讨论、逐步细化”思想的分类模型，模型直观，易解释，如前面 5.1.4 节中可以直接给出决策图	sklearn.tree
随机森林	思想跟决策树类似，精度通常比决策树要高，缺点是由于其随机性，丧失了决策树的可解释性	sklearn.ensemble
朴素贝叶斯	基于概率思想的简单有效的分类模型，能够给出容易理解的概率解释	sklearn.naive_bayes
神经网络	具有强大的拟合能力，可以用于拟合、分类等，它有很多个增强版本，如递归神经网络、卷积神经网络、自编码器等，这些是深度学习的模型基础	Keras

fit() 为对模型进行训练，predict() 对输入样集进行预测，score() 对模型进行评估。

聚类分析

常用聚类分析算法

与分类不同，聚类分析是在没有给定划分类别的情况下，根据数据相似度进行样本分组的一种方法。与分类模型需要使用有类标记样本构成的训练数据不同，聚类模型可以建立在无类标记的数据上，是一种非监督的学习算法。聚类的输入是一组未被标记的样本，聚类根据数据自身的距离或相似度将其划分为若干组，划分的原则是组内距离最小化而组间距离最大化。

类 别	包括的主要算法
划分（分裂）方法	K-Means 算法（K-平均）、K-MEDOIDS 算法（K-中心点）、CLARANS 算法（基于选择的算法）
层次分析方法	BIRCH 算法（平衡迭代规约和聚类）、CURE 算法（代表点聚类）、CHAMELEON 算法（动态模型）

基于密度的方法	DBSCAN 算法（基于高密度连接区域）、DENCLUE 算法（密度分布函数）、OPTICS 算法（对象排序识别）
基于网格的方法	STING 算法（统计信息网络）、CLIQUE 算法（聚类高维空间）、WAVE-CLUSTER 算法（小波变换）
基于模型的方法	统计学方法、神经网络方法

算 法 名 称	算 法 描 述
K-Means	K- 均值聚类也称为快速聚类法，在最小化误差函数的基础上将数据划分为预定的类数 K。该算法原理简单并便于处理大量数据
K- 中心点	K- 均值算法对孤立点的敏感性，K- 中心点算法不采用簇中对象的平均值作为簇中心，而选用簇中离平均值最近的对象作为簇中心
系统聚类	系统聚类也称为多层次聚类，分类的单位由高到低呈树形结构，且所处的位置越低，其所包含的对象就越少，但这些对象间的共同特征越多。该聚类方法只适合在小数据量的时候使用，数据量大的时候速度会非常慢

K-Means 聚类算法

K-Means 算法^[11]是典型的基于距离的非层次聚类算法，在最小化误差函数的基础上将数据划分为预定的类数 K ，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。

1. 算法过程

- 1) 从 N 个样本数据中随机选取 K 个对象作为初始的聚类中心。
- 2) 分别计算每个样本到各个聚类中心的距离，将对象分配到距离最近的聚类中。
- 3) 所有对象分配完成后，重新计算 K 个聚类的中心。
- 4) 与前一次计算得到的 K 个聚类中心比较，如果聚类中心发生变化，转过程 2)，否则转过程 5)。
- 5) 当质心不发生变化时停止并输出聚类结果。

聚类的结果可能依赖于初始聚类中心的随机选择，可能使得结果严重偏离全局最优分类。实践中，为了得到较好的结果，通常选择不同的初始聚类中心，多次运行 K-Means 算法。在所有对象分配完成后，重新计算 K 个聚类的中心时，对于连续数据，聚类中心取该簇的均值，但是当样本的某些属性是分类变量时，均值可能无定义，可以使用 K- 众数方法。

2. 数据类型与相似性的度量

(1) 连续属性

对于连续属性，要先对各属性值进行零 - 均值规范，再进行距离的计算。在 K-Means 聚类算法中，一般需要度量样本之间的距离、样本与簇之间的距离以及簇与簇之间的距离。

度量样本之间的相似性最常用的是欧几里得距离、曼哈顿距离和闵可夫斯基距离；样本与簇之间的距离可以用样本到簇中心的距离 $d(e_i, x)$ ；簇与簇之间的距离可以用簇中心的距离 $d(e_i, e_j)$ 。

欧几里得距离：

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \quad (5-23)$$

曼哈顿距离:

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \quad (5-24)$$

闵可夫斯基距离:

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|)^q + (|x_{i2} - x_{j2}|)^q + \cdots + (|x_{ip} - x_{jp}|)^q} \quad (5-25)$$

q 为正整数, $q = 1$ 时即为曼哈顿距离; $q = 2$ 时即为欧几里得距离。

(2) 文档数据

对于文档数据使用余弦相似性度量, 先将文档数据整理成文档-词矩阵。

两个文档之间的相似度的计算公式为:

$$d(i, j) = \cos(i, j) = \frac{\vec{i} \cdot \vec{j}}{|\vec{i}| |\vec{j}|} \quad (5-26)$$

3. 目标函数

使用误差平方和 SSE 作为度量聚类质量的目标函数, 对于两种不同的聚类结果, 选择误差平方和较小的分类结果。

连续属性的 SSE 计算公式为:

$$SSE = \sum_{i=1}^K \sum_{x \in E_i} dist(e_i, x)^2 \quad (5-27)$$

文档数据的 SSE 计算公式为:

$$SSE = \sum_{i=1}^K \sum_{x \in E_i} \cos(e_i, x)^2 \quad (5-28)$$

簇 E_i 的聚类中心 e_i 计算公式为:

$$e_i = \frac{1}{n_i} \sum_{x \in E_i} x \quad (5-29)$$

聚类分析算法评价

聚类分析仅根据样本数据本身将样本分组。其目标是实现组内的对象相互之间是相似的(相关的), 而不同组中的对象是不同的(不相关的)。组内的相似性越大, 组间差别越大, 聚类效果就越好。

(1) purity 评价法

purity 方法是极为简单的一种聚类评价方法, 只需计算正确聚类数占总数的比例。

$$purity(X, Y) = \frac{1}{n} \sum_k \max_i |x_k \cap y_i| \quad (5-30)$$

其中, $x = (x_1, x_2, \cdots, x_k)$ 是聚类的集合。 x_k 表示第 k 个聚类的集合。 $y = (y_1, y_2, \cdots, y_k)$ 表示需要被聚类的集合, y_i 表示第 i 个聚类对象。 n 表示被聚类集合对象的总数。

(2) RI 评价法

实际上, 这是一种用排列组合原理来对聚类进行评价的手段, RI 评价公式如下。

$$RI = \frac{R + W}{R + M + D + W} \quad (5-31)$$

其中, R 是指被聚在一类的两个对象被正确分类了, W 是指不应该被聚在一类的两个对象被正确分开了, M 指不应该放在一类的对象被错误的放在了一类, D 指不应该分开的对象被错误的分开了。

(3) F 值评价法

这是基于上述 RI 方法衍生出的一个方法，F 评价公式如下。

$$F_{\alpha} = \frac{(1 + \alpha^2)pr}{\alpha^2 p + r} \quad (5-32)$$

其中， $p = \frac{R}{R + M}$ ， $r = \frac{R}{R + D}$ 。

实际上 RI 方法就是把准确率 p 和召回率 r 看得同等重要，事实上，有时候我们可能需要某一特性更多一点，这时候就适合使用 F 值方法。

Python 主要聚类分析算法

对象名	函数功能	所属工具箱
KMeans	K 均值聚类	sklearn.cluster
AffinityPropagation	吸引力传播聚类，2007 年提出，几乎优于所有其他方法，不需要指定聚类数，但运行效率较低	sklearn.cluster
MeanShift	均值漂移聚类算法	sklearn.cluster
SpectralClustering	谱聚类，具有效果比 K 均值好，速度比 K 均值快等特点	sklearn.cluster
AgglomerativeClustering	层次聚类，给出一棵聚类层次树	sklearn.cluster
DBSCAN	具有噪声的基于密度的聚类方法	sklearn.cluster
BIRCH	综合的层次聚类算法，可以处理大规模数据的聚类	sklearn.cluster

关联规则

常用关联规则算法

算法名称	算法描述
Apriori	关联规则最常用也是最经典的挖掘频繁项集的算法，其核心思想是通过连接产生候选项及其支持度然后通过剪枝生成频繁项集
FP-Tree	针对 Apriori 算法的固有的多次扫描事务数据集的缺陷，提出的不产生候选频繁项集的方法。Apriori 和 FP-Tree 都是寻找频繁项集的算法
Eclat 算法	Eclat 算法是一种深度优先算法，采用垂直数据表示形式，在概念格理论的基础上利用基于前缀的等价关系将搜索空间划分为较小的子空间
灰色关联法	分析和确定各因素之间的影响程度或是若干个子因素（子序列）对主因素（母序列）的贡献度而进行的一种分析方法

Apriori 算法

通过连接产生候选项与其支持度，然后通过剪枝生成频繁项集。

1. 关联规则和频繁项集

(1) 关联规则的一般形式

项集 A、B 同时发生的概率称为关联规则的支持度（也称相对支持度）。

$$Support(A \Rightarrow B) = P(A \cup B) \quad (5-33)$$

项集 A 发生, 则项集 B 发生的概率为关联规则的置信度。

$$Confidence(A \Rightarrow B) = P(B|A) \quad (5-34)$$

(2) 最小支持度和最小置信度

最小支持度是用户或专家定义的衡量支持度的一个阈值, 表示项目集在统计意义上的最低重要性; 最小置信度是用户或专家定义的衡量置信度的一个阈值, 表示关联规则的最低可靠性。同时满足最小支持度阈值和最小置信度阈值的规则称作强规则。

(3) 项集

项集是项的集合。包含 k 个项的项集称为 k 项集, 如集合 {牛奶, 麦片, 糖} 是一个 3 项集。

项集的出现频率是所有包含项集的事务计数, 又称作绝对支持度或支持度计数。如果项集 I 的相对支持度满足预定义的最小支持度阈值, 则 I 是频繁项集。频繁 k 项集通常记作 k 。

(4) 支持度计数

项集 A 的支持度计数是事务数据集中包含项集 A 的事务个数, 简称为项集的频率或计数。

已知项集的支持度计数, 则规则 $A \Rightarrow B$ 的支持度和置信度很容易从所有事务计数、项集 A 和项集 $A \cup B$ 的支持度计数推出。

$$Support(A \Rightarrow B) = \frac{A, B \text{ 同时发生的事务个数}}{\text{所有事务个数}} = \frac{Support_count(A \cap B)}{Total_count(A)} \quad (5-35)$$

$$Confidence(A \Rightarrow B) = P(A|B) = \frac{Support(A \cap B)}{Support(A)} = \frac{Support_count(A \cap B)}{Support_count(A)} \quad (5-36)$$

也就是说, 一旦得到所有事务个数, A, B 和 $A \cap B$ 的支持度计数, 就可以导出对应的关联规则 $A \Rightarrow B$ 和 $B \Rightarrow A$, 并可以检查该规则是否是强规则。

2. Apriori 算法: 使用候选产生频繁项集

Apriori 算法的主要思想是找出存在于事务数据集中的最大的频繁项集, 在利用得到的最大频繁项集与预先设定的最小置信度阈值生成强关联规则。

(1) Apriori 的性质

频繁项集的所有非空子集也必须是频繁项集。根据该性质可以得出: 向不是频繁项集 I 的项集中添加事务 A, 新的项集 $I \cup A$ 一定也不是频繁项集。

(2) Apriori 算法实现的两个过程如下。

1) 找出所有的频繁项集 (支持度必须大于等于给定的最小支持度阈值), 在这个过程中连接步和剪枝步互相融合, 最终得到最大频繁项集 L_k 。

连接步:

连接步的目的是找到 K 项集。对给定的最小支持度阈值, 分别对 1 项候选集 C_1 , 剔除小于该阈值的项集得到 1 项频繁集 L_1 ; 下一步由 L_1 自身连接产生 2 项候选集 C_2 , 保留 C_2 中满足约束条件的项集得到 2 项频繁集, 记为 L_2 ; 再下一步由 L_2 与 L_3 连接产生 3 项候选集 C_3 , 保留 C_2 中满足约束条件的项集得到 3 项频繁集, 记为 L_3 ……这样循环下去, 得到最大频繁项集 L_k 。

剪枝步：

剪枝步紧接着连接步，在产生候选项 C_k 的过程中起到减小搜索空间的目的。由于 C_k 是 L_{k-1} 与 L_1 连接产生的，根据 Apriori 的性质频繁项集的所有非空子集也必须是频繁项集，所以不满足该性质的项集不会存在于 C_k 中，该过程就是剪枝。

2) 由频繁项集产生强关联规则：由过程 1) 可知未超过预定的最小支持度阈值的项集已被剔除，如果剩下这些规则又满足了预定的最小置信度阈值，那么就挖掘出了强关联规则。

时序模式

时间序列算法

模型名称	描述
平滑法	平滑法常用于趋势分析和预测，利用修匀技术，削弱短期随机波动对序列的影响，使序列平滑化。根据所用平滑技术的不同，可具体分为移动平均法和指数平滑法
趋势拟合法	趋势拟合法把时间作为自变量，相应的序列观察值作为因变量，建立回归模型。根据序列的特征，可具体分为线性拟合和曲线拟合
组合模型	时间序列的变化主要受到长期趋势 (T)、季节变动 (S)、周期变动 (C) 和不规则变动 (ε) 这 4 个因素的影响。根据序列的特点，可以构建加法模型和乘法模型 加法模型： $x_t = T_t + S_t + C_t + \varepsilon_t$ 乘法模型： $x_t = T_t \times S_t \times C_t \times \varepsilon_t$
AR 模型	$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t$ 以前 p 期的序列值 $x_{t-1}, x_{t-2}, \cdots, x_{t-p}$ 为自变量、随机变量 X_t 的取值 x_t 为因变量建立线性回归模型
MA 模型	$x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$ 随机变量 X_t 的取值 x_t 与以前各期的序列值无关，建立 x_t 与前 q 期的随机扰动 $\varepsilon_{t-1}, \varepsilon_{t-2}, \cdots, \varepsilon_{t-q}$ 的线性回归模型
ARMA 模型	$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$ 随机变量 X_t 的取值 x_t 不仅与以前 p 期的序列值有关，还与前 q 期的随机扰动有关
ARIMA 模型	许多非平稳序列差分后会显示出平稳序列的性质，称这个非平稳序列为差分平稳序列。对差分平稳序列可以使用 ARIMA 模型进行拟合
ARCH 模型	ARCH 模型能准确地模拟时间序列变量的波动性的变化，适用于序列具有异方差性并且异方差函数短期自相关
GARCH 模型及其衍生模型	GARCH 模型称为广义 ARCH 模型，是 ARCH 模型的拓展。相比于 ARCH 模型，GARCH 模型及其衍生模型更能反映实际序列中的长期记忆性、信息的非对称性等性质

时间序列的预处理

拿到一个观察值序列后，首先要对它的纯随机性和平稳性进行检验，这两个重要的检验称为序列的预处理。根据检验结果可以将序列分为不同的类型，对不同类型的序列会采取不同的分析方法。

对于纯随机序列,又称为白噪声序列,序列的各项之间没有任何相关关系,序列在进行完全无序的随机波动,可以终止对该序列的分析。白噪声序列是没有信息可提取的平稳序列。

对于平稳非白噪声序列,它的均值和方差是常数,现已有一套非常成熟的平稳序列的建模方法。通常是建立一个线性模型来拟合该序列的发展,借此提取该序列的有用信息。ARMA 模型是最常用的平稳序列拟合模型。

对于非平稳序列,由于它的均值和方差不稳定,处理方法一般是将其转变为平稳序列,这样就可以应用有关平稳时间序列的分析方法,如建立 ARMA 模型来进行相应的研究。如果一个时间序列经差分运算后具有平稳性,则该序列为差分平稳序列,可以使用 ARIMA 模型进行分析。

1. 平稳性检验

(1) 平稳时间序列的定义

对于随机变量 X , 可以计算其均值 (数学期望) μ 、方差 σ^2 ; 对于两个随机变量量 X 和 Y , 可以计算 X, Y 的协方差 $\text{cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)]$ 和相关系数 $\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$, 它们度量了两个不同事件之间的相互影响程度。

对于时间序列 $\{X_t, t \in T\}$, 任意时刻的序列值 X_t 都是一个随机变量, 每一个随机变量都会有均值和方差, 记 X_t 的均值为 μ_t , 方差为 σ_t ; 任取 $t, s \in T$, 定义序列 $\{X_t\}$ 的自协方差函数 $\gamma(t, s) = E[(X_t - \mu_t)(X_s - \mu_s)]$ 和自相关系数 $\rho(t, s) = \frac{\text{cov}(X_t, X_s)}{\sigma_t \sigma_s}$ (特别地, $\gamma(t, t) = \gamma(0) = 1, \rho_0 = 1$), 之所以称它们为自协方差函数和自相关系数, 是因为它们衡量的是同一个事件在两个不同时期 (时刻 t 和 s) 之间的相关程度, 形象地讲就是度量自己过去的行为对自己现在的影响。

如果时间序列 $\{X_t, t \in T\}$ 在某一常数附近波动且波动范围有限, 即有常数均值和常数方差, 并且延迟 k 期的序列变量的自协方差和自相关系数是相等的或者说延迟 k 期的序列变量之间的影响程度是一样的, 则称 $\{X_t, t \in T\}$ 为平稳序列。

(2) 平稳性的检验

对序列的平稳性的检验有两种检验方法, 一种是根据时序图和自相关图的特征做出判断的图检验, 该方法操作简单、应用广泛, 缺点是带有主观性; 另一种是构造检验统计量进行检验的方法, 目前最常用的方法是单位根检验。

1) 时序图检验。根据平稳时间序列的均值和方差都为常数的性质, 平稳序列的时序图显示该序列值始终在一个常数附近随机波动, 而且波动的范围有界; 如果有明显的趋势性或者周期性, 那它通常不是平稳序列。

2) 自相关图检验。平稳序列具有短期相关性, 这个性质表明对平稳序列而言通常只有近期的序列值对现时值的影响比较明显, 间隔越远的过去值对现时值的影响越小。随着延迟期数 k 的增加, 平稳序列的自相关系数 ρ_k (延迟 k 期) 会比较快的衰减趋向于零, 并在零附近随机波动, 而非平稳序列的自相关系数衰减的速度比较慢, 这就是利用自相关图进行平稳性检验的标准。

3) 单位根检验。单位根检验是指检验序列中是否存在单位根，如果存在单位根就是非平稳时间序列了。

2. 纯随机性检验

如果一个序列是纯随机序列，那么它的序列值之间应该没有任何关系，即满足 $\gamma(k) = 0$, $k \neq 0$ 这是一种理论上才会出现的理想状态，实际上纯随机序列的样本自相关系数不会绝对为零，但是很接近零，并在零附近随机波动。

纯随机性检验也称白噪声检验，一般是构造检验统计量来检验序列的纯随机性，常用的检验统计量有 Q 统计量、LB 统计量，由样本各延迟期数的自相关系数可以计算得到检验统计量，然后计算出对应的 p 值，如果 p 值显著大于显著性水平 α ，则表示该序列不能拒绝纯随机的原假设，可以停止对该序列的分析。

平稳时间序列分析

ARMA 模型的全称是自回归移动平均模型，它是目前最常用的拟合平稳序列的模型。它又可以细分为 AR 模型、MA 模型和 ARMA 三大类。都可以看作是多元线性回归模型。

1. AR 模型

具有如下结构的模型称为 p 阶自回归模型，简记为 $AR(p)$ 。

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t \quad (5-37)$$

即在 t 时刻的随机变量 X_t 的取值 x_t 是前 p 期 $x_{t-1}, x_{t-2}, \cdots, x_{t-p}$ 的多元线性回归，认为 x_t 主要是受过去 p 期的序列值的影响。误差项是当期的随机干扰 ε_t ，为零均值白噪声序列。

统 计 量	性 质	统 计 量	性 质
均值	常数均值	自相关系数 (ACF)	拖尾
方差	常数方差	偏自相关系数 (PACF)	p 阶截尾

(1) 均值

对满足平稳性条件的 $AR(p)$ 模型的方程，两边取期望，得：

$$E(x_t) = E(\phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t) \quad (5-38)$$

已知 $E(x_t) = \mu$, $E(\varepsilon_t) = 0$ ，所以有 $\mu = \phi_0 + \phi_1 \mu + \phi_2 \mu + \cdots + \phi_p \mu$ ，

解得：

$$\mu = \frac{\phi_0}{1 - \phi_1 - \phi_2 - \cdots - \phi_p} \quad (5-39)$$

(2) 方差

平稳 $AR(p)$ 模型的方差有界，等于常数。

(3) 自相关系数 (ACF)

平稳 $AR(p)$ 模型的自相关系数 $\rho_k = \rho(t, t-k) = \frac{\text{cov}(X_t, X_{t-k})}{\sigma_t \sigma_{t-k}}$ 呈指数的速度衰减，始终有非零取值，不会在 k 大于某个常数之后就恒等于零，这个性质就是平稳 $AR(p)$ 模型的自相关系数 ρ_k 具有拖尾性。

(4) 偏自相关系数 (PACF)

对于一个平稳 $AR(p)$ 模型，求出延迟 k 期自相关系数 ρ_k 时，实际上的得到的并不是 X_t 与

X_{t-k} 之间单纯的相关关系, 因为 X_t 同时还会受到中间 $k-1$ 个随机变量 $X_{t-1}, X_{t-2}, \dots, X_{t-k}$ 的影响, 所以自相关系数 ρ_k 里实际上掺杂了其他变量对 X_t 与 X_{t-k} 的相关影响, 为了单纯地测度 X_{t-k} 对 X_t 的影响, 引进偏自相关系数的概念。

可以证明平稳 $AR(p)$ 模型的偏自相关系数具有 p 阶截尾性。这个性质连同前面的自相关系数的拖尾性是 $AR(p)$ 模型重要的识别依据。

2. MA 模型

具有如下结构的模型称为 q 阶自回归模型, 简记为 $MA(q)$ 。

$$x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (5-40)$$

即在 t 时刻的随机变量 X_t 的取值 x_t 是前 q 期的随机扰动 $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-q}$ 的多元线性函数, 误差项是当期的随机干扰 ε_t , 为零均值白噪声序列, μ 是序列 $\{X_t\}$ 的均值。认为 x_t 主要是受过去 q 期的误差项的影响。

统 计 量	性 质	统 计 量	性 质
均值	常数均值	自相关系数 (ACF)	q 阶截尾
方差	常数方差	偏自相关系数 (PACF)	拖尾

3. ARMA 模型

具有如下结构的模型称为自回归移动平均模型, 简记为 $ARMA(p, q)$ 。

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (5-41)$$

即在 t 时刻的随机变量 X_t 的取值 x_t 是前 p 期 $x_{t-1}, x_{t-2}, \dots, x_{t-p}$ 和前 q 期 $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-q}$ 的多元线性函数, 误差项是当期的随机干扰 ε_t , 为零均值白噪声序列。认为 x_t 主要是受过去 p 期的序列值和过去 q 期的误差项的共同影响。

特别的, 当 $q = 0$ 时, 是 $AR(p)$ 模型; 当 $p = 0$ 时, 是 $MA(q)$ 模型。

统 计 量	性 质	统 计 量	性 质
均值	常数均值	自相关系数 (ACF)	拖尾
方差	常数方差	偏自相关系数 (PACF)	拖尾

4. 平稳时间序列建模

某个时间序列经过预处理, 被判定为平稳非白噪声序列, 就可以利用 ARMA 模型进行建模。计算出平稳非白噪声序列 $\{X_t\}$ 的自相关系数和偏自相关系数, 再由 $AR(p)$ 模型、 $MA(q)$ 和 $ARMA(p, q)$ 的自相关系数和偏自相关系数的性质, 选择合适的模型。平稳时间序列建模步骤如图 5-17 所示。

1) 计算 ACF 和 PACF。先计算非平稳白噪声序列的自相关系数 (ACF) 和偏自相关系数 (PACF)。

2) ARMA 模型识别。也称为模型定阶, 由 $AR(p)$ 模型、 $MA(q)$ 和 $ARMA(p, q)$ 的自相关系数和偏自相关系数的性质, 选择合适的模型。识别的原则见表 5-24。

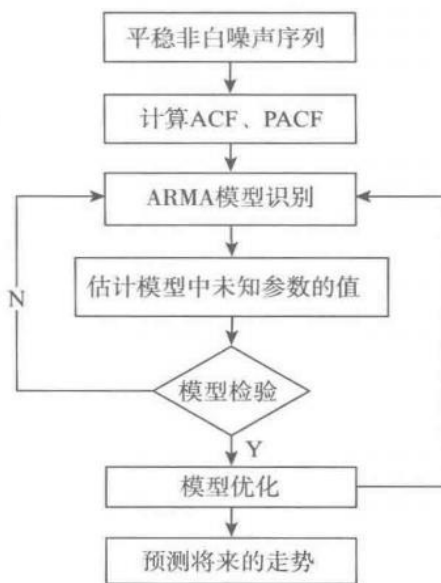


图 5-17 平稳时间序列 ARMA 模型建模步骤

模 型	自相关系数 (ACF)	偏自相关系数 (PACF)
$AR(p)$	拖尾	p 阶截尾
$MA(q)$	q 阶截尾	拖尾
$ARMA(p, q)$	p 阶拖尾	q 阶拖尾

3) 估计模型中未知参数的值并进行参数进行检验。

4) 模型检验。

5) 模型优化。

6) 模型应用：进行短期预测。

非平稳时间序列分析

对非平稳时间序列的分析方法可以分为确定性因素分解的时序分析和随机时序分析两大类。

确定性因素分解的方法把所有序列的变化都归结为 4 个因素（长期趋势、季节变动、循环变动和随机波动）的综合影响，其中长期趋势和季节变动的规律性信息通常比较容易提取，而由随机因素导致的波动则非常难确定和分析，对随机信息浪费严重，会导致模型拟合精度不够理想。

随机时序分析法的发展就是为了弥补确定性因素分解方法的不足。根据时间序列的不同特点，随机时序分析可以建立的模型有 ARIMA 模型、残差自回归模型、季节模型、异方差模型等。本节重点介绍使用 ARIMA 模型对非平稳时间序列进行建模的方法。

1. 差分运算

(1) p 阶差分

相距一期的两个序列值之间的减法运算称为 1 阶差分运算。

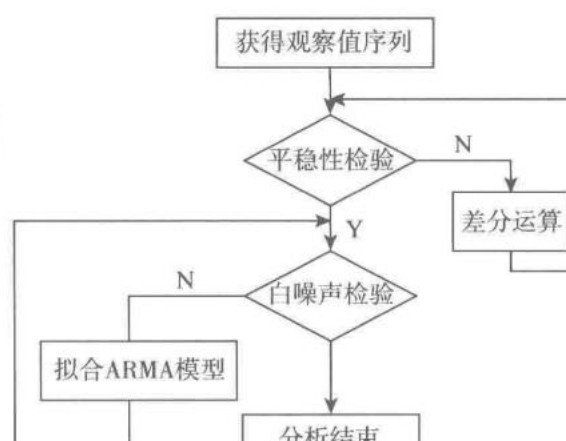
(2) k 步差分

相距 k 期的两个序列值之间的减法运算称为 k 步差分运算。

2. ARIMA 模型

差分运算具有强大的确定性信息提取能力，许多非平稳序列差分后会显示出平稳序列的性质，这时称这个非平稳序列为差分平稳序列。对差分平稳序列可以使用 ARMA 模型进行拟合。ARIMA 模型的实质就是差分运算与 ARMA 模型的组合，掌握了 ARMA 模型的建模方法和步骤以后，对序列建立 ARIMA 模型是比较简单的。

差分平稳时间序列建模步骤如图 5-18 所示。



Python 主要时序模式算法

函 数 名	函 数 功 能	所 属 工 具 箱
acf()	计算自相关系数	statsmodels.tsa.stattools

plot_acf()	画自相关系数图	statsmodels.graphics.tsaplots
pacf()	计算偏相关系数	statsmodels.tsa.stattools
plot_pacf()	画偏相关系数图	statsmodels.graphics.tsaplots
adfuller()	对观测值序列进行单位根检验	statsmodels.tsa.stattools
diff()	对观测值序列进行差分计算	Pandas 对象自带的方法
ARIMA()	创建一个 ARIMA 时序模型	statsmodels.tsa.arima_model
summary() 或 summaty2	给出一份 ARIMA 模型的报告	ARIMA 模型对象自带的方法
aic/bic/hqic	计算 ARIMA 模型的 AIC/BIC/HQIC 指标值	ARIMA 模型对象自带的变量
forecast()	应用构建的时序模型进行预测	ARIMA 模型对象自带的方法
acorr_ljungbox()	Ljung-Box 检验，检验是否为白噪声	statsmodels.stats.diagnostic

离群点检测

因为离群点的属性值明显偏离期望的或常见的属性值，所以离群点检测也称偏差检测。

(1) 离群点的成因

离群点的主要成因有：数据来源于不同的类、自然变异、数据测量和收集误差。

(2) 离群点类型

分类标准	分类名称	分类描述
从数据范围	全局离群点和局部离群点	从整体来看，某些对象没有离群特征，但是从局部来看，却显示了一定的离群性。如图 5-25 所示，C 是全局离群点，D 是局部离群点
从数据类型	数值型离群点和分类型离群点	这是以数据集的属性类型进行划分的
从属性的个数	一维离群点和多维离群点	一个对象可能有一个或多个属性

离群点检测方法

离群点检测方法	方法描述	方法评估
基于统计	大部分的基于统计的离群点检测方法是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离群点	基于统计模型的离群点检测方法的前提是必须知道数据集服从什么分布；对于高维数据，检验效果可能很差
基于邻近度	通常可以在数据对象之间定义邻近性度量，把远离大部分点的对象视为离群点	简单，二维或三维的数据可以做散点图观察；大数据集不适用；对参数选择敏感；具有全局阈值，不能处理具有不同密度区域的数据集
基于密度	考虑数据集可能存在不同密度区域这一事实，从基于密度的观点分析，离群点是在低密度区域中的对象。一个对象的离群点得分是该对象周围密度的逆	给出了对象是离群点的定量度量，并且即使数据具有不同的区域也能够很好处理；大数据集不适用；参数选择是困难的
基于聚类	一种利用聚类检测离群点的方法是丢弃远离其他簇的小簇；另一种更系统的方法，首先聚类所有对象，然后评估对象属于簇的程度（离群点得分）	基于聚类技术来发现离群点可能是高度有效的；聚类算法产生的簇的质量对该算法产生的离群点的质量影响非常大

基于统计模型的离群点检测方法需要满足统计学原理，如果分布已知，则检验可能非常有效。基于邻近度的离群点检测方法比统计学方法更一般、更容易使用，因为确定数据集有意义的邻近度量比确定它的统计分布更容易。基于密度的离群点检测与基于邻近度的离群点检测密切相关，因为密度常用邻近度定义：一种是定义密度为到 K 个最邻近的平均距离的倒数，如果该距离小，则密度高；另一种是使用 DBSCAN 聚类算法，一个对象周围的密度等于该对象指定距离 d 内对象的个数。

基于模型的离群点检测方法

通过估计概率分布的参数来建立一个数据模型。如果一个数据对象不能很好地同该模型拟合，即如果它很可能不服从该分布，则它是一个离群点。

一元正态分布中的离群点检测

混合模型的离群点检测

混合模型将数据看作从不同的概率分布得到的观测值的集合。概率分布可以是任何分布，但是通常是多元正态的，因为这种类型的分布不难理解，容易从数学上进行处理，并且已经证明在许多情况下都能产生好的结果。这种类型的分布可以对椭圆簇建模。

总的来说，混合模型数据产生过程为：给定几个类型相同但参数不同的分布，随机地选取一个分布并由它产生一个对象。重复该过程 m 次，其中 m 是对象的个数。

具体地讲，假定有 K 个分布和 m 个对象 $\chi = \{x_1, x_2, \dots, x_m\}$ 。设第 j 个分布的参数为 α_j ，并设 A 是所有参数的集合，即 $A = \{\alpha_1, \alpha_2, \dots, \alpha_K\}$ 。则 $P(x_i | \alpha_j)$ 是第 i 个对象来自第 j 个分布的概率。选取第 j 个分布产生一个对象的概率由权值 $w_j (1 \leq j \leq K)$ 给定，其中权值（概率）受限于其和为 1 的约束，即 $\sum_{j=1}^K w_j = 1$ 。于是，对象 x 的概率由以下公式给出：

$$P(x | A) = \sum_{j=1}^K w_j P_j(x | \theta_j) \quad (5-42)$$

如果对象以独立的方式产生，则整个对象集的概率是每个个体对象 x_i 的概率的乘积，公式如下。

$$P(\chi | \alpha) = \prod_{i=1}^m P(x_i | \alpha) = \prod_{i=1}^m \sum_{j=1}^K w_j P_j(x_i | \alpha_j) \quad (5-43)$$

对于混合模型，每个分布描述一个不同的组，即一个不同的簇。通过使用统计方法，可以由数据估计这些分布的参数，从而描述这些分布（簇）。也可以识别哪个对象属于哪个簇。然而，混合模型只是给出具体对象属于特定簇的概率。

聚类时，混合模型方法假定数据来自混合概率分布，并且每个簇可以用这些分布之一识别。同样，对于离群点检测，用两个分布的混合模型建模，一个分布为正常数据，而另一个为离群点。

聚类和离群点检测的目标都是估计分布的参数，以最大化数据的总似然。

我们提供一种离群点检测常用的简单的方法：先将所有数据对象放入正常数据集，这时离群点集为空集；再用一个迭代过程将数据对象从正常数据集转移到离群点集，该转移能提高数据的总似然。

具体操作如下。

假设数据集 U 包含来自两个概率分布的数据对象： M 是大多数（正常）数据对象的分布，而 N 是离群点对象的分布。数据的总概率分布可以记作：

$U(x) = (1-\lambda)M(x) + \lambda N(x)$ 其中， x 是一个数据对象； $\lambda \in [0, 1]$ ，给出离群点的期望比例。分布 M 由数据估计得到，而分布 N 通常取均匀分布。设 M_t 和 N_t 分别为时刻 t 正常数据和离群点对象的集合。初始 $t=0$ ， $M_0=D$ ，而 $N_0 \neq \emptyset$ 。

根据公式混合模型中公式 $P(x|A) = \sum_{j=1}^K w_j P_j(x|\alpha_j)$ 推导，在整个数据集的似然和对数似然可分别由下面两式给出。

$$L_t(U) = \prod_{x_i \in U} P_U(x_i) = ((1-\lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i)) (\lambda^{|N_t|} \prod_{x_i \in N_t} P_{N_t}(x_i)) \quad (5-44)$$

$$\ln L_t(U) = |M_t| \ln(1-\lambda) + \sum_{x_i \in M_t} \ln P_{M_t}(x_i) + |N_t| \ln \lambda + \sum_{x_i \in N_t} \ln P_{N_t}(x_i) \quad (5-45)$$

其中 P_D 、 P_{M_t} 、 P_{N_t} 分别是 D 、 M_t 、 N_t 的概率分布函数。

因为正常数据对象的数量比离群点对象的数量大很多，因此当一个数据对象移动到离群点集后，正常数据对象的分布变化不大。在这种情况下，每个正常数据对象的总似然的贡献保持不变。此外，如果假定离群点服从均匀分布，则移动到离群点集的每一个数据对象对离群点的似然贡献一个固定的量。这样，当一个数据对象移动到离群点集时，数据总似然的改变粗略地等于该数据对象在均匀分布下的概率（用 λ 加权）减去该数据对象在正常数据点的分布下的概率（用 $1-\lambda$ 加权）。从而，离群点由这样一些数据对象组成。这样，数据对象在均匀分布下的概率比正常数据对象分布下的概率高。

基于聚类的离群点检测方法

聚类分析用于发现局部强相关的对象组，而异常检测用来发现不与其他对象强相关的对象。因此，聚类分析非常自然地可以用于离群点检测。本节主要介绍两种基于聚类的离群点检测方法。

（1）丢弃远离其他簇的小簇

一种利用聚类检测离群点的方法是丢弃远离其他簇的小簇。通常，该过程可以简化为丢弃小于某个最小阈值的所有簇。

（2）基于原型聚类

基于原型的聚类是另一种更系统的方法。首先聚类所有对象，然后评估对象属于簇的程度（离群点得分）。在这种方法中，可以用对象到它的簇中心的距离来度量属于簇的程度。特别的，如果删除一个对象导致该目标的显著改进，则可将该对象视为离群点。

对于基于原型的聚类，主要有两种方法评估对象属于簇的程度：一是度量对象到簇原型的距离，并用它作为该对象的离群点得分；二是考虑到簇具有不同程度的密度，可以度量簇到原型的相对距离，相对距离是点到质心的距离与簇中所有点到质心的距离的中位数之比。

诊断步骤如下。

- 1) 进行聚类。选择聚类算法（如 K-Means 算法），将样本集聚为 K 簇，并找到各簇的质心。
- 2) 计算各对象到它的最近质心的距离。
- 3) 计算各对象到它的最近质心的相对距离。
- 4) 与给定的阈值作比较。

如果某对象距离大于该阈值，就认为该对象是离群点。

基于聚类的离群点检测的改进如下。

1) 离群点对初始聚类的影响：通过聚类检测离群点时，离群点会影响聚类结果。为了处理该问题，可以使用方法：对象聚类，删除离群点，对象再次聚类（这个不能保证产生最优结果）。

2) 还有一种更复杂的方法：取一组不能很好地拟合任何簇的特殊对象，这组对象代表潜在的离群点。随着聚类过程的进展，簇在变化。不再强属于任何簇的对象被添加到潜在的离群点集合；测试当前在该集合中的对象，如果它现在强属于一个簇，就可以将它从潜在的离群点集合中移除。聚类过程结束时还留在该集合中的点被分类为离群点（这种方法也不能保证产生最优解，甚至不比前面的简单算法好，在使用相对距离计算离群点得分时，这个问题特别严重）。

对象是否被认为是离群点可能依赖于簇的个数（如 k 很大时的噪声簇）。该问题也没有简单的答案。一种策略是对于不同的簇个数重复该分析。另一种方法是找出大量小簇，其想法如下。

1) 较小的簇倾向于更加凝聚；

2) 如果存在大量小簇时，一个对象是离群点，则它多半是一个真正的离群点。

不利的一面是一组离群点可能形成小簇从而逃避检测。