第5章 信息内容分析

字符串匹配技术实现了关键字是否存在于正文中的研判，下一步需要分析信息内容是否为非法非授权信息，研判是否涉及谣言传播、敏感信息泄露、意识形态渗透、恶意舆论操纵等，为后续精准实施安全管理策略提供决策依据。

本章内容包括基本的分类和聚类算法、情感分析等深层语义理解、热点话题分析、与社交网络群体发现等。这些技术共同为信息内容安全管理提供了重要的技术支撑与分析视角，是实现风险源头精准识别、恶意内容理解、威胁影响范围评估等的核心基础。

5.1 文本分类

5.1.1 分类的基本概念

文本分类(text categorization 或 text classification)起源于人们对信息快速检索和高效管理的需求。

具体来说，文本分类是指在预先定义的分类模型下，自动地将自然语言文本按照其内容归属到一个或多个预定的类别。这个过程本质上是知识学习和应用的过程。首先，分类器通过学习每个类别中的若干样本文本，总结出各个类别的特征规律，并建立相应的判别模型和规则。这一阶段是知识学习过程。随后，当面对新的文本时，分类器根据已总结的规则判断这些文本的类别，这是知识应用过程。

总之，文本分类是一种将文本(例如文章、网页内容、邮件等)根据其内容自动分类到一个或多个预定义类别的过程，在信息内容安全领域有着广泛的应用。下面具体介绍文本分类问题。

假定有以下元素：

(1) 实例的描述($x \in X$)。这里 X 是实例空间，即所有可能文本的集合。x 是其中的一个具体实例，例如一篇文章或一封邮件。

(2) 固定的文本分类体系 $C=\{c_1, c_2, \cdots, c_n\}$。这是预先定义好的一组分类标签，每个标签 c_i 代表一个特定的类别。

(3) 有监督的分类。由于类别是事先定义好的，分类过程是有指导的(即有监督的)，这意味着有一组标记过类别的文本作为训练数据指导分类器如何分类。

在这个框架下，目标是确定实例 x 的类别 $c(x)\in C$。其中，$c(x)$是一个分类函数，它的定义域是实例空间 X，值域是分类体系 C。

文本分类问题可以具体分为以下 3 类：

(1) 二类问题(binary classification)。每个实例只分为两类之一，例如“属于”或“不属于”某个特定类别。

(2) 多类问题(multi-class classification)。每个实例可以被分类到多个类别中的一个。这种类型的问题可以拆分成多个二类问题。

(3) 多标签问题(multi-label classification)。每个实例可以同时属于多个类别。

分类体系通常由人工构造，以反映特定领域或需求的类别结构。例如：

(1) 广泛的领域分类，如“政治”“体育”“军事”等。

(2) 更具体的主题分类，如“网络诈骗”“恐怖事件”等。

(3) 已经存在的分类体系，例如路透社(Reuters)分类体系、中国图书馆分类法(中图法)分类体系等。

要能够实现文本的自动分类，必须有完整的文本分类系统和一整套的数据处理流程。图 5-1 是一个典型的文本分类系统的结构。一般来说，一个完整的文本分类系统通常包括如下几个主要阶段：文本预处理、文本集合的表示、维数约减、分类器的学习、分类器的测试以及分类器性能评价。每个阶段具体工作如下：

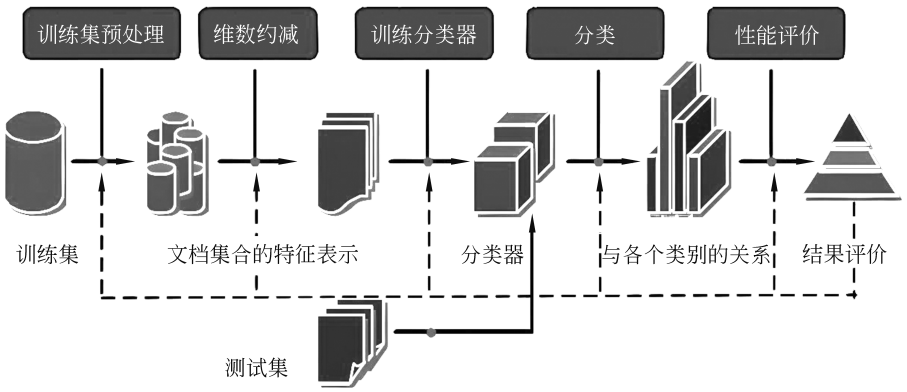


图 5-1　典型的文本分类系统的结构

(1) 文本预处理。包括对文档集合进行格式分析并提取重要内容，进行中文分词、英文词干化、剔除停用词等操作。对于英文文本，预处理技术相对成熟，例如词干化用于还原词汇到基本形式。对于中文文本，分词是一个主要挑战，因为中文文本没有明显的词间分隔符。常用的处理方法包括基于词典的方法、自然语言处理技术和基于统计的方法。

(2) 文本集合的表示。文本被视为出现在其中的关键词的集合，这些关键词成为特征。通常使用向量空间模型(Vector Space Model，VSM)表示文本集合，其中文本被表示为特征组成的向量。

(3) 维数约减。从文本集合提取的特征数量通常很大，过多的特征不仅无助于提高分类效率，反而可能导致维度灾难。因此，需要通过某些方法(如主成分分析、特征选择算法等)从大量特征中抽取最有利于文本分类的特征，并以一定的描述模型对文本进行特征表示。

(4) 分类器的学习。这是文本分类系统的核心环节。从文本集合中选取一部分文本作为训练集。利用机器学习算法针对训练集进行学习,确定分类器的参数或阈值,最终构建分类器。常用的分类器包括朴素贝叶斯分类器、支持向量机、决策树等。

(5) 分类器的测试。使用分类器对文本集合的测试集进行分类,以获取分类结果。测试分为封闭测试(closed testing)和开放测试(open testing),其中,封闭测试使用训练集中的数据进行测试,而开放测试使用未用于训练的数据。

(6) 分类器性能评价。使用精确度、召回率、F1 分数等评价指标对分类结果进行评估。如果分类结果不符合预期,可能需要返回前面的某一步骤进行调整,如重新选择特征、调整分类器参数等。

5.1.2　基于规则归纳的分类方法

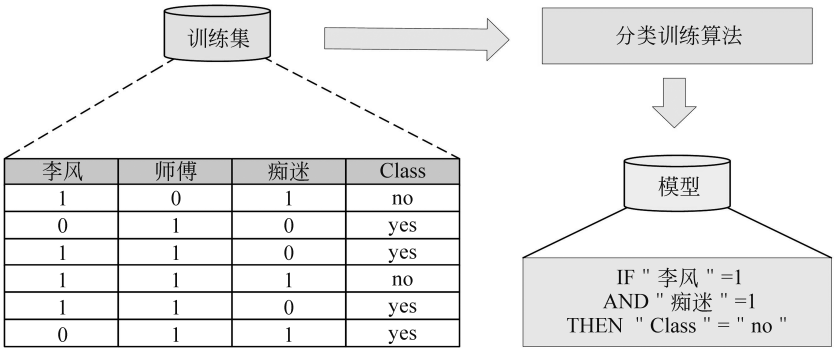
CN2 是一种基于规则的机器学习算法,用于从一系列已标记的示例中提取有意义的分类规则。CN2 作为一个有效的规则生成工具,通过学习用户提供的大量已知属性的示例生成用于分类的规则,以便对新的、属性未知的示例进行评估和分类。

具体而言,CN2 通过比较大量的示例识别分类的共同点和区别点。这些共同点和区别点是由示例中的特征值决定的。换句话说,一组特定的特征值组合可以成为分类的依据。例如,假设有许多不同的生物作为示例群体,可以将这些生物分为鱼类和兽类。如何进行分类呢? 如果确定“4 条腿且用肺呼吸”是兽类的特征,则可以形成一个规则:“如果腿的数量为 4 且呼吸方式为肺呼吸,则该生物属于兽类”。

要应用 CN2 这种工具,首先需要提供一批生物的数据,并且已知哪些是兽类,哪些是鸟类,以及每种生物的腿的数量、呼吸方式、繁殖方式等属性。CN2 通过学习这些示例的特征值和对应的分类归纳出分类规则,即特征组合的表达式。然后,CN2 可以应用这些规则判断具有不同特征组合(例如“腿的数量为 3 且不通过肺呼吸”)的生物属于哪个类别。

如图 5-2 所示,假设训练集中包含一个数据表,表中有 4 个数据项,分别是"李风"、"师傅"、"痴迷"和"Class"。通过对这些数据的训练和归纳,CN2 能够提炼出一个分类规则。例如,规则可能是

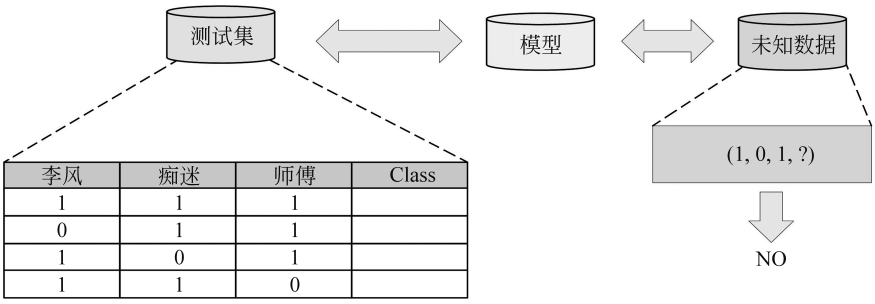
```
IF "李风"=1 AND "痴迷"=1,THEN "Class"="no"
```



李风	师傅	痴迷	Class
1	0	1	no
0	1	0	yes
1	1	0	yes
1	1	1	no
1	1	0	yes
0	1	1	yes

图 5-2　CN2 生成规则示例

训练结束后，就可以用测试集测试模型。如图 5-3 所示，假设有一组未知类别的测试数据：(1,0,1,?)，将这组数据输入模型后，模型会根据其学习到的规则预测这组数据的类别，给出分类结果为 NO。



李风	痴迷	师傅	Class
1	1	1	
0	1	1	
1	0	1	
1	1	0	

图 5-3　CN2 的模型测试示例

5.1.3　决策树分类方法

决策树(decision tree)是数据建模中常用的一种方法，它通过一系列规则对数据进行分类或预测。决策树的基本思想是：从训练数据中选取一个最能区分不同类别的样本的属性，将其作为树的根节点，并基于这个属性将训练集分成几个子集。接下来，算法从每个子集中选出区分度较大的属性作为下一层的节点。这个过程一直重复，直到达到某个停止条件，例如所有的叶子节点都只包含单一类别的样本，或达到预设的树的最大深度。

决策树的基本组成部分包括决策节点、分支和叶子节点。树中的每个内部节点(非叶子节点)代表一个属性上的测试，这些测试的结果决定了样本将走向哪个分支。每个叶子节点代表一个类别，表示从根节点到该叶子节点的路径对应的决策规则所归纳的类别。

如图 5-4 所示，以购房贷款申请的风险评估为例，决策树可以帮助银行或金融机构快速有效地做出判断。在这种应用中，决策树的每个节点可能代表贷款申请人的不同属性，如年龄、收入、信用历史等。通过对这些属性进行一系列判断，决策树最终将每个申请分类为高风险或低风险。例如，决策树的一个节点可能是“年收入是否超过 5 万美元”，根据是或否的答案，样本将被分到不同的子节点。最终，每个叶子节点代表了一种风险评估的结果。

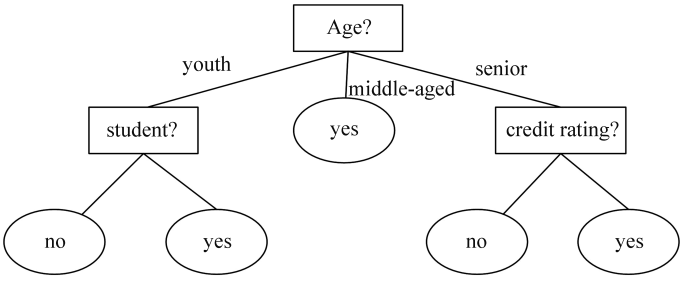


图 5-4　决策树分类示例

决策树的优点是模型直观且易于理解，不需要复杂的数学知识即可解释模型的决策过程。然而，决策树也存在一些缺点，如容易过拟合、对数据中的小变化敏感等，因此在实际应用中通常需要通过剪枝等技术优化模型的泛化能力。

在决策树模型中，最顶端的节点被称为根节点，它标志着整个决策过程的开始。根节点基于选定的属性将数据集分成子集，这些子集由根节点的子节点（或分支）表示。决策树中每个节点可以拥有的子节点数取决于使用的决策树构建算法。例如，在 CART（Classification And Regression Tree，分类与回归树）算法中，每个节点产生两个分支，形成的是一种二叉树结构。而在其他算法（如 ID3 或 C4.5）中，一个节点可能产生多于两个分支，形成的则是多叉树。

在决策树中，每个分支可以是一个新的决策节点，或者是一个叶子节点。叶子节点表示分类决策的最终结果，即数据的类别。当从根节点沿着树向下搜索时，在每个决策节点上都会根据某个属性的不同值选择不同的分支。这个过程一直持续，直到达到叶子节点。到达叶子节点时，就完成了对一个数据实例的分类。在这个过程中，每个决策节点上的问题对应一个属性，而每个叶子节点代表一个可能的类别。

决策树中一个重要的环节是如何选择具有高区分度的属性。在许多决策树算法中，通常认为具有最高信息增益（information gain）的属性是最具区分度的。信息增益是一种基于信息论的度量，用来评估每个属性在分类过程中提供的信息量。选择信息增益最高的属性作为决策节点，可以最大限度地减少分类时的不确定性。因此，通过计算每个属性的信息增益，可以确定属性的重要性排序，进而有效地构建决策树。以下是根据一个数据划分 D 的训练元组产生决策树的算法伪代码：

```
算法：Generate_decision_tree:
输入：数据划分 D，训练元组和对应类标号的集合；
      attribute_list，候选属性的集合；
      Attribute_selection_method，一个确定最好地划分数据元组为个体类的分裂准则的过
      程，这个准则由分裂属性和分裂点或分裂子集组成
输出：一棵决策树
方法：
创建一个节点 N
if  D 中的元组都是同一类 C then
    返回 N 作为叶子节点，以类 C 标记
if  attribute_list 为空 then
    返回 N 作为叶子节点，标记为 D 中的多数类          //多数表决
    使用 attribute_selection_method(D, attribute_list)，找出最好的 splitting_
    criterion 用 splitting_criterion 标记节点 N
if  splitting_attribute 是离散值并且允许多路划分 then          //不限于二叉树
    attribute_list ← attribute_list — splitting_attribute
                                                          //删除划分属性
for splitting_criterion 的每个输出 j                //划分元组并对每个划分产生子树
    设 Dj 是 D 中满足输出的 j 的数据元组的集合          //一个划分
    if  Dj 为空 then
      加一个叶子节点到节点 N，标记为 D 中的多数类
    else 加一个由 Generate_decision_tree(Dj，attribute_list)返回的节点到节点 N
  end for
返回 N
```

决策树算法的核心是贪心算法，它采用自上而下分而治之的方法。在构建决策树的过程中，初始时刻，所有的数据都集中在根节点。随后，算法递归地使用选定的属性对数据集进行分割，形成树的各个分支。这一分割过程会持续进行，直到达到某个停止条件。通常这

些条件包括：所有位于当前节点的数据都属于同一类别，没有更多的属性可用于进一步分割数据，或者达到了树的预设最大深度。

在每次分割过程中，算法都会尽力确保生成的子组之间的差异最大化。这种差异可以通过各种统计度量衡量，例如信息增益、增益率(gain ratio)或基尼不纯度(gini impurity)。具体计算公式见表 5-1。不同的决策树算法(如 ID3、C4.5 或 CART)之间的主要区别在于它们衡量差异的方式不同。

表 5-1　决策树统计度量值计算公式

度　量　值	公　　式	含　　义
信息熵(Entropy)	$E(D)=-\sum_{i=1}^{m}p_i\log_2 p_i$	衡量样本集 D 的纯度，值越小表示越纯。p_i 表示第 i 类样本所占比例
条件熵(Conditional Entropy)	$E(Y\mid X)=\sum_{i=1}^{m}p(X=x_i)E(Y\mid X=x_i)$	衡量在已知某一特征 X 的情况下，随机变量 Y 的不确定性有多大。通过对所有可能的 X 取值的熵进行加权平均来计算
信息增益(Information Gain)	$\mathrm{Gain}(Y,X)=E(Y)-E(Y\mid X)$	表示在特征 X 的条件下，随机变量 Y 的信息不确定性减少的程度
信息增益率(Information Gain Ratio)	$\mathrm{Gain\_ratio}(Y,X)=\dfrac{\mathrm{Gain}(Y\mid X)}{E(X)}$	信息增益率可以对信息增益进行归一化，减少对取值较多属性的偏好。其中，$E(X)$表示特征 X 的信息度量值
基尼不纯度(Gini Impurity)	$\mathrm{Gini}(D)=1-\sum_{i=1}^{m}p_i^2$	衡量数据集不纯度，值越大表示样本混杂度越高

在选择用于分割数据的属性时，通常会基于某种启发式规则或统计度量做出选择。例如，信息增益是一种衡量属性分割数据有效性的统计度量。通常，选取的属性是分类属性(离散的)。如果遇到连续的属性，则需要先进行离散化处理，如通过设定阈值将其划分为不同的类别。

以 C4.5 为例，使用信息增益率来选择当前节点进行分叉的特征，每个节点进行分裂的算法流程为：

```
while(当前节点不纯)
    1 计算当前节点的类别熵 E(D)(以类别取值计算)；
    2 计算当前节点的属性熵 E(D|Aᵢ)(按照属性取值下的类别取值计算)；
    3 计算各个属性的信息增益 Gain(D,Aᵢ)=E(D)-E(D|Aᵢ)；
    4 计算各个属性的分类信息度量值 E(Aᵢ)(按照属性取值计算)；
    5 计算各个属性的信息增益率 Gain_ratio(D,Aᵢ)=Gain(D,Aᵢ)/E(Aᵢ)；
end while
当前节点设置为叶子节点
```

例 5-1 是基于 C4.5 算法进行决策树分割的实例。

【例 5-1】 表 5-2 给定了一组根据天气状况决定是否举行某项活动的数据样本。数据样本描述天气状况的属性包括天气、温度、湿度、风速，类别标签有两个，分别表示活动是否举行：类别集合 C=｛进行，取消｝。根据 C.5 算法进行决策树分割。

表 5-2　例 5-1 的数据样本

序号	天气	温度	湿度	风速	活动
1	晴	炎热	高	弱	取消
2	晴	炎热	高	强	取消
3	阴	炎热	高	弱	进行
4	雨	适中	高	弱	进行
5	雨	寒冷	正常	弱	进行
6	雨	寒冷	正常	强	取消
7	阴	寒冷	正常	强	进行
8	晴	适中	高	弱	取消
9	晴	寒冷	正常	弱	进行
10	雨	适中	正常	弱	进行
11	晴	适中	正常	强	进行
12	阴	适中	高	强	进行
13	阴	炎热	正常	弱	进行
14	雨	适中	高	强	取消

首先，从根节点进行决策树分裂，计算第一个分裂的属性。

(1) 计算类别信息熵：表示的是所有样本中各种类别出现的不确定性之和。共有两个类别，表示活动“进行”或者“取消”。其中，样本总数为 14，类别“进行”有 9 个样本，类别“取消”有 5 个样本。

$$\begin{aligned}E(\text{活动}) &= -(p_{\text{活动}=\text{"进行"}}\log_2 p_{\text{活动}=\text{"进行"}} + p_{\text{活动}=\text{"取消"}}\log_2 p_{\text{活动}=\text{"取消"}}) \\ &= -\left[\frac{9}{14}\times\log_2\left(\frac{9}{14}\right)+\frac{5}{14}\times\log_2\left(\frac{5}{14}\right)\right] \\ &\approx 0.940\end{aligned}$$

(2) 计算每个属性的信息熵：每个属性的信息熵相当于一种条件熵，表示的是在某种属性的条件下，各种类别出现的不确定性之和。属性的信息熵越大，表示这个属性中拥有的样本类别越“不纯”。

$$\begin{aligned}E(\text{活动}\mid\text{天气}) &= p_{\text{天气}=\text{"晴"}}E(\text{活动}\mid\text{天气}=\text{"晴"}) + p_{\text{天气}=\text{"阴"}}E(\text{活动}\mid\text{天气}=\text{"阴"}) + \\ &\quad p_{\text{天气}=\text{"雨"}}E(\text{活动}\mid\text{天气}=\text{"雨"}) \\ &= \frac{5}{14}\times\left[-\frac{2}{5}\times\log_2\frac{2}{5}-\frac{3}{5}\times\log_2\frac{3}{5}\right]+\frac{4}{14}\times\left[-\frac{4}{4}\times\log_2\frac{4}{4}\right]+ \\ &\quad \frac{5}{14}\times\left[-\frac{3}{5}\times\log_2\frac{3}{5}-\frac{2}{5}\times\log_2\frac{2}{5}\right] \\ &\approx 0.694\end{aligned}$$

$$\begin{aligned}E(\text{活动}\mid\text{温度}) &= p_{\text{温度}=\text{"炎热"}}E(\text{活动}\mid\text{温度}=\text{"炎热"}) + p_{\text{温度}=\text{"适中"}}E(\text{活动}\mid\text{温度}=\text{"适中"}) + \\ &\quad p_{\text{温度}=\text{"寒冷"}}E(\text{活动}\mid\text{温度}=\text{"寒冷"}) \\ &= \frac{4}{14}\times\left[-\frac{2}{4}\times\log_2\frac{2}{4}-\frac{2}{4}\times\log_2\frac{2}{4}\right]+\frac{6}{14}\times\left[-\frac{4}{6}\times\log_2\frac{2}{6}\right]+\end{aligned}$$

$$\frac{4}{14}\times\left[-\frac{3}{4}\times\log_2\frac{3}{4}-\frac{1}{4}\times\log_2\frac{1}{4}\right]$$

$$\approx 0.911$$

$$E(\text{活动}\mid\text{湿度})=p_{\text{湿度="高"}}E(\text{活动}\mid\text{湿度="高"})+p_{\text{湿度="正常"}}E(\text{活动}\mid\text{湿度="正常"})$$

$$=\frac{7}{14}\times\left[-\frac{3}{7}\times\log_2\frac{3}{7}-\frac{4}{7}\times\log_2\frac{4}{7}\right]+\frac{7}{14}\times\left[-\frac{6}{7}\times\log_2\frac{6}{7}-\frac{1}{7}\times\log_2\frac{1}{7}\right]$$

$$\approx 0.789$$

$$E(\text{活动}\mid\text{风速})=p_{\text{风速="强"}}E(\text{活动}\mid\text{风速="强"})+p_{\text{风速="弱"}}E(\text{活动}\mid\text{风速="弱"})$$

$$=\frac{6}{14}\times\left[-\frac{3}{6}\times\log_2\frac{3}{6}-\frac{3}{6}\times\log_2\frac{3}{6}\right]+\frac{8}{14}\times\left[-\frac{6}{8}\times\log_2\frac{6}{8}-\frac{2}{8}\times\log_2\frac{2}{8}\right]$$

$$\approx 0.892$$

(3) 计算信息增益：信息增益＝熵－条件熵，在这里就是类别信息熵－属性信息熵，它表示的是信息不确定性减少的程度。如果一个属性的信息增益越大，就表示用这个属性进行样本划分可以更好地减少划分后样本的不确定性，当然，选择该属性就可以更快更好地完成我们的分类目标。

$$\text{Gain}(\text{活动},\text{天气})=E(\text{活动})-E(\text{活动}\mid\text{天气})=0.940-0.694=0.246$$

$$\text{Gain}(\text{活动},\text{温度})=E(\text{活动})-E(\text{活动}\mid\text{温度})=0.940-0.911=0.029$$

$$\text{Gain}(\text{活动},\text{湿度})=E(\text{活动})-E(\text{活动}\mid\text{湿度})=0.940-0.789=0.151$$

$$\text{Gain}(\text{活动},\text{风速})=E(\text{活动})-E(\text{活动}\mid\text{风速})=0.940-0.892=0.048$$

(4) 计算属性分裂的信息度量：用分裂信息度量来考虑某种属性进行分裂时分支的数量信息和尺寸信息，我们把这些信息称为属性的内在信息。信息增益率用信息增益/内在信息，会导致属性的重要性随着内在信息的增大而减小(也就是说，如果这个属性本身不确定性就很大，那就越不倾向于选取它)，这样算是对单纯用信息增益的补偿。

$$E(\text{天气})=-(p_{\text{天气="晴"}}\log_2 p_{\text{天气="晴"}}+p_{\text{天气="阴"}}\log_2 p_{\text{天气="阴"}}+p_{\text{天气="雨"}}\log_2 p_{\text{天气="雨"}})$$

$$=-\left[\frac{5}{14}\times\log_2\left(\frac{5}{14}\right)+\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)+5/14\times\log_2\left(\frac{5}{14}\right)\right]$$

$$\approx 1.577$$

$$E(\text{温度})=-(p_{\text{温度="炎热"}}\log_2 p_{\text{温度="炎热"}}+p_{\text{温度="适中"}}\log_2 p_{\text{温度="适中"}}+p_{\text{温度="寒冷"}}\log_2 p_{\text{温度="寒冷"}})$$

$$=-\left[\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)+\frac{6}{14}\times\log_2\left(\frac{6}{14}\right)+\frac{4}{14}\times\log_2\left(\frac{4}{14}\right)\right]$$

$$\approx 1.556$$

$$E(\text{湿度})=-(p_{\text{湿度="高"}}\log_2 p_{\text{湿度="高"}}+p_{\text{湿度="正常"}}\log_2 p_{\text{湿度="正常"}})$$

$$=-\left[\frac{7}{14}\times\log_2\left(\frac{7}{14}\right)+\frac{7}{14}\times\log_2\left(\frac{7}{14}\right)\right]=1.0$$

$$E(\text{风速})=-(p_{\text{风速="强"}}\log_2 p_{\text{风速="强"}}+p_{\text{风速="弱"}}\log_2 p_{\text{风速="弱"}})$$

$$=-\left[\frac{6}{14}\times\log_2\left(\frac{6}{14}\right)+\frac{8}{14}\times\log_2\left(\frac{8}{14}\right)\right]\approx 0.985$$

(5) 计算信息增益率：

$$\text{Gain\_ratio}(\text{活动},\text{天气})=\frac{\text{Gain}(\text{活动}\mid\text{天气})}{E(\text{天气})}=0.246/1.577\approx 0.155$$

$$\text{Gain\_ratio}(\text{活动},\text{温度})=\frac{\text{Gain}(\text{活动}\mid\text{温度})}{E(\text{温度})}=0.029/1.556\approx 0.0186$$

$$\text{Gain\_ratio}(\text{活动},\text{湿度})=\frac{\text{Gain}(\text{活动}\mid\text{湿度})}{E(\text{湿度})}=0.151/1.0=0.151$$

$$\text{Gain\_ratio}(\text{活动},\text{风速})=\frac{\text{Gain}(\text{活动}\mid\text{风速})}{E(\text{风速})}=0.048/0.985\approx 0.048$$

那么，在根节点时，天气的信息增益率最高，选择天气为当前的分裂属性。在第一次分裂之后，会出现三个节点。对于每个节点判断其是否类别唯一，如果类别唯一，则该节点被定义为叶子节点。在例 5-1 中，根节点分裂后，天气为“阴”的条件下，类别唯一，则它定义为叶子节点。其余两个节点为类别不唯一节点，继续按分裂根节点的方法进行分裂。读者可以自行尝试。

由于决策树的构建过程通常只需要对数据集进行有限次数的扫描，因此可以比较快地建立，适合大型数据集的分类任务。这种快速构建的特性，加上决策树模型的易解释性，使得它在各种应用场景中都非常受欢迎。

5.1.4　朴素贝叶斯分类方法

朴素贝叶斯(Naive Bayes)分类是一种基于贝叶斯定理的简单概率分类，它假设在给定类别的情况下各特征相互独立。这种算法特别适用于大量数据的分类，可以有效地预测给定样本属于特定类别的可能性。下面是朴素贝叶斯分类的关键概念解释：

(1) 先验概率。这是在没有考虑任何特征的影响前一个样本属于类别 C_k 的概率。它是基于训数据集中的类别分布直接计算得出的，表示为 $P(C_k)$，其中 C_k 表示第 k 个类别。

(2) 后验概率。这是在考虑了特征 x 的影响后一个样本属于类别 C_k 的概率。它是结合先验概率和类条件概率，通过贝叶斯公式计算得出的，表示为 $P(C_k\mid x)$。

(3) 类条件概率。这是在已知样本属于类别 C_k 的条件下具有某些特征 x 的概率。这个概率是基于特征在该类别的训练样本中的分布计算的。

(4) 贝叶斯公式。贝叶斯定理是整个朴素贝叶斯分类的核心，它的公式为

$$P(C_k\mid x)=\frac{P(x\mid C_k)P(C_k)}{P(x)}$$

其中，$P(C_k\mid x)$是后验概率，$P(x\mid C_k)$是类条件概率，$P(C_k)$是先验概率，$P(x)$是特征 x 的概率。在具体应用中，每个 $P(x\mid C_k)$和 $P(C_k)$需要从训练数据中计算得出，而 $P(x)$通常通过对所有类别的 $P(x\mid C_k)P(C_k)$求和获得。

朴素贝叶斯分类器的“朴素”之处在于它假设各特征相互独立，这使得类条件概率 $P(x\mid C_k)$的计算简化为各个独立特征的概率的乘积。

朴素贝叶斯分类的具体步骤如下：

(1) 数据准备。收集数据，准备好训练集，其中每个样本都是特征-类别对。

(2) 计算先验概率。计算训练集中每个类别的先验概率。

(3) 计算类条件概率。对每个类别,计算其下每个特征的概率分布。

(4) 分类决策。对于一个新的样本,计算其属于每个类别的后验概率。这涉及将该样本的特征值代入贝叶斯公式中。

(5) 结果输出。选择具有最高后验概率的类别作为该样本的预测分类。

朴素贝叶斯分类因其简单性和高效性,在文本分类、垃圾邮件检测等领域得到了广泛应用。使用朴素贝叶斯方法进行数据分类的具体算法流程如下:

(1) 每个数据样本用一个 n 维特征向量 $\boldsymbol{X}=[x_1\ x_2\cdots x_n]$ 表示,分别描述对 n 个性质 $A_1,A_2,\cdots,A_n$ 产生的 n 个度量。

(2) 假定有 m 个类 $C_1,C_2,\cdots,C_m$。给定一个未知的数据样本 $\boldsymbol{X}$(即没有类标号),分类法将预测 $\boldsymbol{X}$ 属于具有最高后验概率(条件 $\boldsymbol{X}$ 下)的类。即朴素贝叶斯分类将未知的样本分配给类 C_i,这样最大化 $P(C_i|\boldsymbol{X})$。其中 $P(C_i|\boldsymbol{X})$最大的类 C_i 称为最大后验假定。

(3) 由于 $P(\boldsymbol{X})$对于所有类为常数,只需要 $P(\boldsymbol{X}|C_i)P(C)$最大即可。如果类的先验概率未知,则通常假定这些类是等概率的,即 $P(C_1)=P(C_2)=\cdots=P(C_m)$,并据此对 $P(C_i|\boldsymbol{X})$最大化。否则,最大化 $P(\boldsymbol{X}|C_i)P(C_i)$。注意,类的先验概率可以用 $P(C_i)=s_i/s$ 计算,其中 s_i 是类 C_i 中的训练样本数,而 s 是训练样本总数。

(4) 给定具有许多属性的数据集,计算 $P(\boldsymbol{X}|C_i)$ 的开销可能非常大。为降低计算开销,可以做类条件独立的朴素假定。给定样本的类标号,假定各属性值相互条件独立,即在属性间不存在依赖关系。

(5) 对未知样本 $\boldsymbol{X}$ 进行分类,对每个类 C_i,计算 $P(\boldsymbol{X}|C_i)P(C_i)$。样本 $\boldsymbol{X}$ 被指派到类 C_i,换言之,$\boldsymbol{X}$ 被指派到其 $P(\boldsymbol{X}|C_i)P(C_i)$最大的类 C_i。

在理论上,与其他所有分类算法相比,贝叶斯分类具有最小的出错率。然而,实践中并非总是如此。这是由于对其应用的假定(如类条件独立性)不准确以及缺乏可用的概率数据造成的。尽管如此,种种实验研究表明,在某些领域,贝叶斯分类算法可以与决策树和神经网络分类算法相媲美。

【例 5-2】 表 5-3 的样本为一组顾客是否购买计算机的调查数据,样本用属性"年龄"、"收入"、"是否为学生"、"信用评级"描述。类标号属性"是否购买计算机"具有两个不同值,即{yes,no}。根据该样本集,利用朴素贝叶斯分类对一个具有属性{youth,medium,yes,fair}的未知样本 $\boldsymbol{X}$ 进行分类。

表 5-3 例 5-2 的数据样本集

序号	年龄	收入	是否为学生	信用评级	是否购买计算机
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle-aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no