模式识别与机器学习

1. 基本概念

随着计算机网络的飞速发展，采集、存储、传输的信息规模达到了前所未有的水平，产生了海量的信息。例如，根据国际权威机构 Stabista 的统计和预测，全球数据量在 2019 年约达 41ZB，即 41 万亿个 GB。面对如此庞大的信息，单纯依靠人力处理是非常困难的，而模式识别(Pattern Recognition)与机器学习(Machine Learning)是非常有效的信息处理技术手段。

模式识别是研究如何使机器具有感知能力的科学，主要研究视觉模式和听觉模式的识别，如识别物体、地形、图像、字体(如签字)等，在日常生活各方面以及军事上都有广泛应用。具体而言，模式识别是指对表征事物或现象各种形式的(数值、文字和逻辑关系)信息进行处理和分析，以对事物或现象进行描述、辨认、分类和解释的过程，是信息科学和人工智能的重要组成部分。模式识别算法所识别的事件或过程可以是文字、声音、图像等具体对象，也可以是状态、程度等抽象对象。这些对象与数字形式的信息相区别，称为模式信息。模式识别从 19 世纪 50 年代兴起，是信息科学和人工智能的重要组成部分，主要被应用于图像分析与处理、语音识别、声音分类、通信、计算机辅助诊断、数据挖掘等方面。

机器学习是研究如何使机器(计算机)从经验和数据中获得知识或提高自身能力的科学。机器学习可以描述为：对于某类任务和性能度量参数，如果一个计算机程序在任务上以度量参数衡量的性能随着经验的积累能够自我完善，那么称这个计算机程序可从经验中学习。不同于模式识别中强调提取特征给机器，从而让机器对未知的事物进行判断；机器学习更加注重从已知的经验数据(样本)中，通过某种特定的方法(算法)，自己去寻找提炼(训练/学习)出一些规律(模型)，进而用来判断或者预测一些未知的事情。

模式识别和机器学习是分别从实际工程和计算机科学的角度发展起来的知识。尽管模式识别、机器学习技术不断迭代，但不可否认的是，新技术的发展总是建立在原有技术的基础之上。尽管新的技术会不断占据潮流，但这并不意味着旧有技术已经过时。同时随着技术和应用的发展，它们越来越融合，解决了很多共同问题(分类、聚类、特征选择、信息融合等)，这两个领域的界限也越来越模糊。模式识别和机器学习的理论和方法可用来解决很多机器感知和与信息处理的问题，其中包括图像/视频分析、文档分析、信息检索和网络搜索等，吸引了越来越多的研究者，理论和方法的进步促进了工程应用中识别性能的明显提高。

在人工智能领域，模式识别和机器学习技术都可完成判断或者预测，互有其独特和补充作用。例如，在图像识别等高维数据处理方向，机器学习的算法更加有效，但是在一些简单的色彩识别领域，参数维度相对单一，界定也相对明显，如果用大数据去建模计算，无疑是一种“大材小用”，而传统的模式识别算法更加合适。因此不同的算法，可以在不同领域发挥各自的效用。

人类是一个富有创造力的种族，能够通过已经发生的或者经历过的事情不断积累经验，并利用这些经验去应对新鲜事物和未知的世界，这就是人类无与伦比的学习能力。随着计算机技术的不断发展，在越来越多的领域，智能机器正在代替人类来完成人们的日常生产活动，并且取得了不错的效果。然而，在这些生产活动中，大多数还是体力劳动或者重复劳动，较少涉及智力活动，这就是机器的致命弱点——不具备思维性。假使机器能够像人一样思维、学习甚至创造，那么机器就能帮助人分担更多的生产活动(主要是智力活动)，从而进一步提高生产效率。在这种情况下不断丰富和完善模式识别和机器学习领域的理论和实践知识，其最终目标是让机器模拟人类的思维和学习。

从广义的角度看，模式识别和机器学习就是利用机器对人思维的模拟，使机器学到新的技能，以实现机器性能的提高。本章将结合模式识别技术，重点介绍机器学习及相关算法。

1.1 研究分类

机器学习主要使用计算机模拟人类的学习活动，它是研究计算机识别现有知识、获取新知识、不断改善性能和实现自身完善的方法。因此，机器学习侧重于如何提高学习系统的泛化能力，或者说是机器在数据中发现模式并具有良好的推广能力。模式识别是指对表征事物或现象各种形式的(数值、文字和逻辑关系)信息进行处理和分析，以对事物或现象进行描述、辨认、分类和解释的过程。模式识别侧重于利用计算机对要分析的客观事物通过某种模式算法对其进行分类，使识别到的结果最接近于待识别的客观事实。

1. 按机器学习策略划分

机器学习是学习与推理的紧密结合，在学习中使用的推理方法就是学习策略。目前在机器学习领域内的主流学习策略可以分为三类：搜索型策略、构造型策略和规划型策略。

(1) 搜索型策略

顾名思义，搜索型策略是以搜索策略为基础去解决问题，常见的算法有状态空间法和误差反向传播算法。以状态空间法为例，它主要用一个三元组(即 x 表示当前状态的集合向量、II表示操作的集合向量、y 表示目标状态的集合向量)。状态向量随着时间的变化在空间里形

成一条轨迹，问题就转化为从某个初始状态出发去搜索一条能够到达目标状态的路径，该方法比较适合于简单问题的解决。

(2) 构造型策略

构造型策略就如同建造大楼，总是先从最底层的地基开始，然后逐层往上，重复构造直至最后大楼建好。它主要体现了分层的思想，将学习的系统分为若干独立的子功能模块，该策略的设计过程比较复杂，可理解性比较差，难以处理海量数据。

(3) 规划型策略

规划型策略就是将学习的过程转化为数学规划问题，最典型的就是支持向量机算法。此方法适合解决非线性问题，样本在预处理后被映射到一个高维的特征向量中，从而将非线性的实际问题转化为可用数学模型解决的线性问题。

2. 按机器学习方式划分

从机器学习的方式划分，机器学习可以分为五类：

(1) 记忆学习

机器学习是机器通过记忆学习资料，避免接触其内部复杂的逻辑和关系的学习方式，是最简单的学习方式，故又被称作记忆学习。

(2) 传授学习

传授学习又被称为指点学习，就是在学习的过程中，外部人为输入一些知识表达式以帮助学习过程的方式。

(3) 演绎学习

演绎学习是学习系统已经习得一套知识体系，学习系统根据已有的知识体系对未知情况进行合理的推理，并将新的结论存储到知识体系中，包括知识改造、知识编译、宏操作等保真变换。

(4) 归纳学习

归纳学习就是应用归纳法的一类学习方法，它又分为实例学习和观察与发现学习。实例学习就是系统提供各种实际的样例，从中归纳出这些样例的一般性规律；观察与发现学习就是从一般性环境中发现观察到的现象并形成理论，包括概念聚类、曲线拟合和构造分类等。

(5) 类比学习

类比学习就是利用之前学习到的类似问题的解决方法来解决现有问题，发现现有问题和已知问题的共同点是这类学习的关键所在。

3. 按机器学习形式划分

从机器学习的形式划分，机器学习可以分为四类：

(1) 监督学习

监督学习中最常用的就是分类问题，主要利用已知类别的样本构造分类器或调整分类器的参数，以达到要求的性能。

(2) 无监督学习

该方法对不带类别的样本信息进行学习，常用的就是聚类。

(3) 半监督学习

该方法是在大量文本信息未标注的基础上标注少量文本来辅助分类，该方法减少了标注所需的代价，某种意义上提高了机器学习的性能。

(4) 强化学习

强化学习就是智能系统从环境到行为映射的学习，以使奖励信号函数值最大，也就是观察后再采取行动。

1.2 研究模型

任何一个过程都可以泛化成一个模型。机器学习的基本模型如图 6-1 所示，主要包括四个基本组成部分：环境、学习环节、知识库和执行环节。

在这个模型中，学习环节和执行环节是两个过程，学习环节通过对环境的学习构建知识库，同时不断地通过学习来改进知识库，而执行环节就是利用已有的知识库来解决当前的问题。

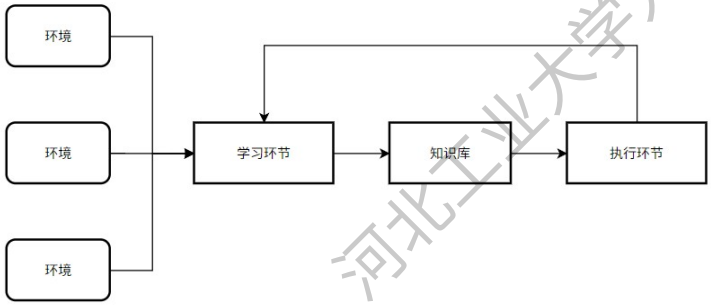

图 1 机器学习基本模型

1. 环境

在机器学习的基本模型中，环境是系统外部的信息源，主要为学习提供信息和样本。环境信息的表现形式决定了机器学习能够解决的问题。例如，高度抽象化的信息适合解决广泛性的问题；低抽象化的信息比较适合解决具体的或个别的问题。信息表现的质量决定了学习

过程的难易和效果，如果环境向系统提供的信息表述准确，机器在学习过程中能较容易归纳总结，取得不错的效果，否则达不到预期效果。

2. 学习环节

学习环节就是通过对外部环境所提供的信息进行学习，归纳总结出知识并不断反馈完善知识。环境所提供的信息必须经过学习环节反复的分析、对比、归纳、总结等过程才能获得相关知识。

3. 知识库

知识库是机器学习模型中用于存放学习环节获得知识的地方。知识库的表现形式和存储结构也是影响模型好坏的重要因素，在知识的表示方面应参照以下基本原则：表达能力的强弱、推理的难度大小、修改的难易程度、是否便于扩充。

4. 执行环节

执行环节是学习系统最重要的环节，执行环节的最终效果也是衡量一个系统是否成功的指标。这个环节主要解决当前所面临的现实问题，将知识库中的知识应用于解决实际问题。同时，每次执行环节的结果都将反馈回学习环节中，从而进一步完善系统的学习。

1.3 研究内容

如果给定一个样本特征，希望预测其对应的属性值，如果其属性值是离散的，那么这是一个分类问题；反之，如果其属性值是连续的实数，则是一个回归问题。如果给定一组样本特征，没有对应的属性值，而是想发掘这组样本在维空间的分布，比如分析哪些样本靠得更近、哪些样本之间离得很远，这属于聚类问题。

无论是分类还是回归，都是想建立一个预测模型，给定一个输入，可以得到一个输出。区别只是在分类问题中，属性值是离散的；而在回归问题中是连续的。总的来说，两种问题的学习算法都很类似，所以在分类问题中用到的学习算法，在回归问题中也能使用。

分类问题最常用的学习算法包括贝叶斯估计(Bayes Estimate)、支持向量机(Support Vector Machine，SVM)、随机梯度下降(Stochastic Gradient Descent，SGD)、集成学习(Ensemble Learning)、k 最近邻(k-Nearest Neighbor，kNN)、决策树学习等；聚类算法包括 k-均值(k-means)、高斯混合模型(Gaussian Mixture Model，GMM)、基于密度的噪声应用空间聚类(Density - Based Spatial Clustering of Applications with Noise，DBSCAN)等几种；而回归问题也能使用最小二乘法、逻辑回归等算法以及其他线性回归算法；降维算法多采用主成分分析(Principal

Component Analysis，PCA)等算法，近些年深度学习以及相关算法更加丰富了该领域的应用和研究。

2. 分类算法

2.1 二分类

首先考虑二类别分类问题 $y\in\{+1，-1\}$。在这种情况下，分类器的学习问题可以近似地定义为取值为+1、-1 的二值函数问题。

二值函数可以使用最小二乘法进行与回归算法相同的学习。测试模式 x 所对应的类别 y 的预测值 y'是由学习后的输出结果的符号决定的：

$$\hat{y}=\operatorname{sgn}(f_\theta(x))=\begin{cases}+1 & (f_\theta(x)>0)\\ 0 & (f_\theta(x)=0)\\ -1 & (f_\theta(x)<0)\end{cases} \tag{1}$$

式中，$f_\theta(x)=0$ 是指实际上不会发生的事件，也就是小概率事件。

如果利用输入为线性的模型为

$$f_\theta(x)=\theta^\top x \tag{2}$$

训练输出 y_i 表示为$\{+1/n_+，-1/n_-\}$。其中，n_+和 n_-分别代表正、负训练样本个数。通过设定，利用最小二乘学习进行模式识别，与线性判别分析算法一致。在线性判别分析中，当正、负两类样本的模式与协方差矩阵服从相同的高斯分布时，可以获得最佳的泛化能力。

分类问题使用函数的正、负符号来进行模式判断，函数值本身的大小并不重要。因此，分类问题中应用如式(3)所示的 0/1 损失，比 L2 损失得到的结果更佳。有

$$\frac{1}{2}\left(1-\operatorname{sgn}\left(f_\theta(x)y\right)\right) \tag{3}$$

上式 0/1 损失等价为

$$\sigma\left(\operatorname{sgn}\left(f_\theta(x)\neq y\right)\right)=\begin{cases}1 & \operatorname{sgn}\left(f_\theta(x)\neq y\right)\\ 0 & \operatorname{sgn}\left(f_\theta(x)=y\right)\end{cases} \tag{4}$$

函数结果为 1 表示分类错误；函数结果为 0 表示分类正确。因此，0/1 损失可以用来对错误分类的样本个数进行统计。

2.2 多类别分类

在实际问题中，类别不止两类，比如英文字母的手写识别是 26 个类别，而汉字的识别则需要成百上千类别。下面介绍两种利用二分类解决多类别分类问题的方法。

1. 一对多法

$$\left.\begin{array}{lll} \text{类别 1} & \text{vs} & \text{类别 1 以外} \rightarrow \overrightarrow{f_1} \\ \text{类别 2} & \text{vs} & \text{类别 2 以外} \rightarrow \overrightarrow{f_2} \\ & \vdots & \\ \text{类别 } c & \text{vs} & \text{类别 } c \text{ 以外} \rightarrow \overrightarrow{f_c} \end{array}\right\} \hat{y}=\underset{y=1\rightarrow c}{\operatorname{argmax}} \hat{f}_y(x) \tag{5}$$

对于所有与 y=1,2,⋯,c 相对应的类别，设其标签为+1，剩余的 y 以外的所有类别，则设其标签为-1。在对样本 x 进行分类时，利用从各个二类别分类问题中得到的 c 个识别函数

$$\hat{f}_1(x), \hat{f}_2(x), \cdots, \hat{f}_c(x) \tag{6}$$

对训练样本进行预测，并计算其函数值，其预测类别 y'即为函数值最大的对应类。有

$$\hat{y}=\underset{y=1\cdots c}{arg\,max}\, \hat{f}_y(x) \tag{7}$$

2. 一对一法

对于所有与 y，y'=1,2,⋯ ,c 相对应的类别，在任意两类之间训练一个分类器，属于类别 y 的标签设为+1，属于类别 y'的标签设置为-1，如式(8)所示，利用二分类算法进行求解。有

$$\operatorname{sgn}\left(\hat{f}_{y,y'}(x)\right)=\begin{cases} +1 & \Rightarrow \text{投票给类别} y \\ 0 & \Rightarrow \text{不给任何类投票} \\ -1 & \Rightarrow \text{投票给类别} y' \end{cases} \tag{8}$$

多类别分类：一对一法见**表 1**。

表 1 多类别分类：一对一法

/	类别 1	类别 2	类别 3	...	类别 c
类别 1	/	$\hat{f}_{12}$	$\hat{f}_{13}$	...	$\hat{f}_{1c}$
类别 2	/	/	$\hat{f}_{23}$	...	$\hat{f}_{2c}$
类别 3	/	/	/	...	$\hat{f}_{3c}$
...	/	/	/	/	...
类别 c	/	/	/	/	/

对样本 x 进行分类时，利用从各个二分类问题中得到的 $c(c\text{-}1)/2$ 个识别函数对训练样本进行预测，再用投票法决定其最终类别，得票数最多的类别就是样本 x 所属的类别。

一对多法和一对一法的主要区别有两方面：一方面，在一对一法中，对二类别问题进行了 c 次求解，而一对多法进行了 $c(c\text{-}1)/2$ 次求解。另一方面，对于每个二类别分类器，一对一法中需要两类的训练样本即可完成训练；而在一对多法中，每个二类别分类器需要所有类别的训练样本都参与才能完成。

2.3 朴素贝叶斯分类

朴素贝叶斯(Naive Bayes)是一种非常简单的分类算法。由于朴素贝叶斯基于概率模型，所以它的优点在于可以对预测标签给出理论上完美的可能性估计，但它也要求数据多维特征之间相互独立。

1 基础概率

概率是对未来事件发生可能性的表述，它有如下关键概念。

- 概率值常用 P 表示，古典概率取值范围是 0～1 之间。比如：如果事件 A 一定不会发生，则有 $P(A)=0$。
- 条件概率：用形如 $P(A|B)$的方式表达，其含义是“如果 B 已经发生，那么 A 发生的概率是多少”。
- 联合概率：是用来描述两个事件共同发生的概率，表达式 $P(AB)$、$P(A, B)$或 $P(A\cap B)$都是联合概率的表示符，其含义是“事件 A、B 同时发生的概率是多少”。
- 事件之间并的概率用 $P(A\cup B)$表示，其含义是“A 或 B 至少一个事件发生的概率”。
- 加法原理：$P(A\cup B)=P(A)+P(B)-P(A\cap B)$。
- 乘法原理：$P(A\cap B)=P(B)\bullet P(A|B)=P(A)\bullet P(B|A)$。
- 两事件独立的充分必要条件是 $P(A\cap B)=P(A)\bullet P(B)$，也就是事件 B 的发生对事件 A 是否发生没有任何影响，即 $P(A|B)=P(A)$。反之也是如此。

从概率定义本身出发，某个事件的概率可以通过公式 $P=\dfrac{\text{构成事件的元素数目}}{\text{整个空间的元素数目}}$ 计算获得，其也称为古典概率公式。但有些时候这样计算不够直观，需要通过其他事件的概率推导获得。举个例子，已知有两袋糖衣巧克力，它们分别装有不同颜色的巧克力。

袋 a: 4 个红色，3 个绿色，3 个黄色巧克力。

袋 b: 2 个红色，7 个绿色，11 个黄色巧克力

假设巧克力的颜色在将其从袋子中取出后才能看到，请读者思考如下两个问题。

问题 1：从袋 *a* 中任意取出一个巧克力，它是红色的概率是多少？

问题 2：任取一袋，再从中取出一颗巧克力发现其为红色，那么它来自袋 *a* 的概率是多少？

相信读者可以很快想到问题 1 的答案，因为其可以通过古典概率直接计算获得，即 $P=\frac{\text{袋}a\text{中红色巧克力个数}}{\text{袋}a\text{中红色}+\text{绿色}+\text{黄色巧克力个数}}=\frac{4}{4+3+3}=\frac{2}{5}$。但对于问题 2 就不那么直观了，应该如何计算呢？贝叶斯定理正是为解决这类问题而来，其定义为：

$$P(A|B)=\frac{P(A)P(B|A)}{P(B)} \tag{9}$$

从公式可知其是描述了两个事件的条件概率之间的关系，在该公式中的每个元素又有其各自的名称：$P(A|B)$是后验概率、$P(A)$是先验概率、$P(B|A)$是似然度、$P(B)$是标准化常量。

现在回到取巧克力的情景，可以根据问题作定义：事件 *A* 是“取到袋 *a*”，事件 *B* 是“取出了一个红色巧克力”。所以问题二演变成了用贝叶斯公式求 $P(A|B)$，此时贝叶斯公式(9)中的各项元素就是：

- 由于任意取了一袋，先验概率 $P(A)=\frac{1}{\text{巧克力袋数}}=\frac{1}{2}$。
- 似然度 $P(B|A)$其实就是前面的问题 1，即 $P(B|A)=\frac{2}{5}$。
- $P(B)$=P(取袋 *a*)×P(从袋 *a* 中取到红色)+P(取袋 *b*)×P(从袋 *b* 中取到红色)$=\frac{1}{2}\times\frac{4}{10}+\frac{1}{2}\times\frac{2}{20}=\frac{1}{4}$。

这样就可以很顺利地获得问题 2 的答案了，即：

$$P(A|B)=\frac{P(A)\times P(B|A)}{P(B)}=\frac{\left(\frac{1}{2}\right)\times\left(\frac{2}{5}\right)}{\frac{1}{4}}=\frac{4}{5} \tag{10}$$

所以虽然任意取到袋 *a* 的概率只有 0.5，但一旦发现从其中取出了一颗红色巧克力，那么它来自袋 *a* 的概率则会达到 0.8。

2 贝叶斯分类原理

朴素贝叶斯是应用贝叶斯定理进行有监督学习的一种分类模型。在该模型中，将贝叶斯定理公式 $P(A|B)=\frac{P(A)P(B|A)}{P(B)}$ 中的事件 A 看成被分类标签，事件 B 看成数据特征。由于通常数据特征是 n 维向量，所以 $P(B)$演变成了 n 个特征的联合概率。因此机器学习语境下的贝叶斯公式变成了：

$$P(y|x_1,x_2,\ldots,x_n)=\frac{P(y)P(x_1,x_2,\ldots,x_n|y)}{P(x_1,x_2,\ldots,x_n)} \tag{11}$$

其中 $x_1, x_2, \ldots, x_n$ 是数据的 n 维特征，y 是预测标签。

(1) 预测

利用贝叶斯公式对某样本进行分类的伪代码如下：

```
for label in 所有标签:
    用贝叶斯公式计算在给定特征情况下出现该 label 的后验概率;

预测标签 ← 获得最高后验概率的 label
```

由于实际上计算了所有标签的后验概率，所以贝叶斯分类不仅可以提供该组特征最可能的标签，还能给出第 2、3、4……高可能性的标签，做出诸如“是苹果的可能性为 50%，是橘子的可能性为 20%，是桃子的可能性为 10%……”这样的预测形式。这是梯度下降、SVM 等分类器所做不到的，在某些应用场景中很有吸引力。

另外，由于对单条样本来说在计算所有标签的后验概率时公式中的 $P(x_1, x_2, ..., x_n)$保持不变，所以出于性能考虑在实践中无须将该标准化常量加入训练与预测的计算中。

(2) 训练

对于训练来说关注的是贝叶斯公式中右侧的先验概率、似然度。

- 先验概率：可以由训练者根据经验直接给出，也可以自动计算：统计训练数据中每个标签的出现次数，除以训练总数就可以得到每个标签的先验概率 $P(y)$。
- 似然度：假定 n 维特征的条件概率符合某种联合分布，根据训练样本估计该分布的参数。比如对于高斯分布来说，学习的参数有期望值和方差。

(3) 独立性假设

在如上所述训练与预测的原理中，难点最终归结为计算 n 维特征的联合分布。在此处朴素贝叶斯模型有一个约定，就是其假设所有 n 维特征之间是相互独立的。这大大简化了联合分布的计算难度，由此事件独立性的充分必要条件有：

$$P(x_1, x_2, \ldots, x_n) = P(x_1) \times P(x_2) \times \ldots \times P(x_n) \tag{12}$$

因此似然度函数为：

$$P(x_1, x_2, \ldots, x_n \mid y) = P(x_1 \mid y) \times P(x_2 \mid y) \times \ldots \times P(x_n \mid y) \tag{13}$$

这意味着可以将目标从计算联合分布的条件概率简化为计算各特征独自的条件概率，这也是模型名称中“朴素”(naive)的由来。

(4) 高斯朴素贝叶斯

训练似然度条件概率中的元素 $P(x_1|y)$, $P(x_2|y)$…的方法是假定特征符合某种分布，然后通过训练集数据估计该分布的参数。高斯朴素贝叶斯使用的高斯分布就是常说的正态分布，假定所有特征条件分布符合：

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{14}$$

其中μ_y、σ_y^2被学习的模型参数特征期望值与方差。具体学习方法涉及统计学的参数估计知识，此处不再展开。

(5) 多项式朴素贝叶斯

多项式朴素贝叶斯(Multinomial Naive Bayes)是用多项分布(Multinomial Distribution)作为似然度概率模型的分类器。由于其实际衡量的是特征在不同标签之间的分布比例关系，所以特别适合文本分类场景(每个单词在不同类型文章中有一定的分布比例)。

先简单介绍多项分布的概念：假设某事件的结果有 k 种可能，在实验了 n 次之后每种结果出现了若干次，多项分布就是用于描述在实验了 n 次之后每种结果发生次数概率的分布。比如普通的骰子有 6 个面，那么掷骰子的结果就是符合 k=6 的多项分布。而掷了 n 次之后结果是 1 的次数有多少呢？这取决于骰子是否均匀，也就是骰子本身掷为 1 的概率是多少。所以在应用多项分布估算 n 次实验的结果之前，还要知道单次实验每种结果发生的可能性是多少，这就是多项分布的超参数。比如掷骰子的超参数＜0.1、0.1、0.1、0.1、0.1、0.5＞就具体化了一个多项分布，其代表的是一种极为不均匀的骰子，因为某面出现的概率达到了 50%。

设训练集中有 m 种分类标签，多项式朴素贝叶斯假定每个特征都符合参数是向量 $\langle \theta_{y_1}, \theta_{y_2}, \ldots, \theta_{y_m} \rangle$ 的多项式分布，向量中的每个值是：

$$\theta_{y_i} = \frac{N_{y_i} + \alpha}{N_y + \alpha \cdot M} \tag{15}$$

其中 N_{yi} 是特征 i 在当前标签中的总数，N_y 是当前标签所有特征的总数。超参数 α 是平滑先验，其目的是防止 N_{yi} 为零从而导致 θ_{yi} 也为零的情况发生。α 通常设为 1，也就是拉普拉斯平滑(Laplace Smoothing)。

为什么要用平滑参数防止 θ_{yi} 为零呢？这是由训练样本的有限性造成的。对于出现概率很小的特征来说，没有出现在某标签的训练样本中并不代表其以后也永远不会出现，所以合理的方法是用平滑参数赋予其一个很小的概率值，而不是零。

(6) 伯努利朴素贝叶斯

伯努利朴素贝叶斯(Bernoulli Naive Bayes)使用伯努利分布(Bernoulli Distribution)作为似然度概率模型。所谓伯努利分布也称二值分布，用来描述一次实验只可能出现两种结果的事件概率分布。由于伯努利分布只能描述二值结果，因而在该学习模型中要求数据中的所有特征都是布尔/二值类型。它用如下公式计算第 i 个特征的似然度：

$$P(x_i|\ y) = P(i|\ y)x_i + (1 - P(i|\ y))(1 - x_i) \tag{16}$$

其中 $P(i|y)$ 是第 i 个特征在所有该标签训练数据中出现的比。

2.4 决策树

决策树(Decision Tree)是一个非常成熟的算法，目前最常用的三种决策树算法 ID3、C4.5、CART 均出现于 20 世纪 80 年代。虽然其理论基础比较简单，但由于训练后可以产生非常直观易懂的树形图，决策树至今仍被广泛应用。

1 最易于理解的模型

决策树最初被用来解决分类问题，目标是从大量的样本数据特征中找到分类决策路径。比如有一个银行 VIP 客户识别系统，定义了如**表 2** 所示的已有样本

表 2 客户分类样本数据

特征向量 *X*			**目标值 *Y***
年龄	**月收入**	**存款**	**客户级别**
20	30000	400	VIP
37	13000	0	普通
50	26000	0	普通
28	10000	3000	普通
31	19000	1500000	VIP
46	7000	6000	普通

虽然只是一组三维特征数据，普通人可能还是无法一下子识别出客户级别划分的依据是什么，但是通过训练可以获得如**表 2** 所示的决策树

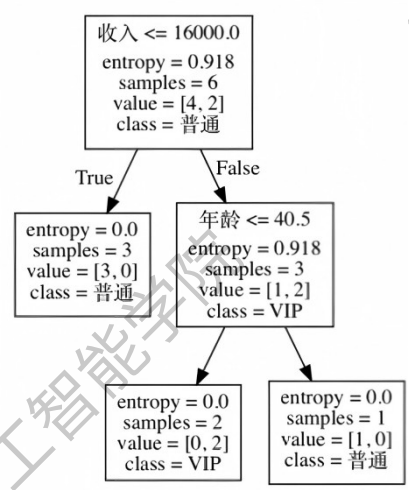

图 2 客户分类决策树

这是一棵 CART 二叉决策树，预测时从根开始每个非叶子节点执行一次某个特征的逻辑判断，当条件为真时走向左子树，否则走向右子树。如此走到叶子结点时就可以知道被预测的标签(图中叶子结点的 class 属性)。从本例训练的结果看，实际只有两个特征在分类预测中起了作用：收入是否小于 16000，年龄是否小于 40.5。

在**表 2** 中每个结点还有另外三个属性。samples：本子树一共包含了多少训练数据；value：本子树训练数据中各种标签类型的样本数量，比如[4, 2]说明第一个类型的样本有 4 条，第二个类型的样本有 2 条；entropy：该结点的信息熵值。熵和信息增益是决策树训练过程中的核心概念。

2 熵的作用

在热力学中熵(entropy)被用来衡量系统的不稳定程度。信息熵的概念由信息论奠基人香农于 1948 年在论文《通信的数学原理》中提出，其目的是用于量化数字信息的价值。可能是由于人们生活中习惯于只对信息做定性判断，直到今天熵仍然不是一个被大众所熟知的概念。人们可能会说“这份情报很有价值”“总是收到垃圾短信”，但却很少思考该条信息好到何种程度或无用到什么地步。

(1) 信息熵的定义

香农提出了量化信息的方式，即

$$\text{随机事件的熵} = H(R_1, P_2, \ldots, P_n) = -\sum_{i=1}^{n} P_i \cdot \log_2(P_i) \tag{17}$$

公式中 $P_1, P_2 \cdots P_n$ 是随机事件每种可能结果的发生概率，所以必有 $P_1+P_2+\cdots P_n=1$。熵 $H(\ )$的结果是一个大于等于零的值，熵越高说明事件的不确定性越大。而当有信息表明不确定性越大事件的结果时，该条信息的价值越高。

(2) 熵的通俗解释

打个比方，假设有人从未来穿越回来告诉您两件事：a)下一次抛硬币的结果，b)下一次掷骰子的结果。哪条信息的价值更高呢？通过熵的公式可以有如下计算结果：

假设硬币是均匀的，每面发生的概率是 1/2，则熵 $H=-\left(\frac{1}{2}\times\log_2\left(\frac{1}{2}\right)+\frac{1}{2}\times\log_2\left(\frac{1}{2}\right)\right)=1$

假设骰子是均匀的，每面的概率是 1/6，则熵 $H=-\left(\frac{1}{6}\times\log_2\left(\frac{1}{6}\right)+\ldots\right)\approx 2.59$ 。

因此抛硬币事件结果的信息价值是不如掷骰子结果的价值的。

现在想象另一种情况，如果老千制作了一枚不均匀的骰子，其中某一面发生的概率达到了 0.9，而另五面的概率都只有 0.02，那么获知此枚骰子掷出结果的信息熵的值是 $H=-\left(\frac{9}{10}\times\log_2\left(\frac{9}{10}\right)+\frac{1}{20}\times\log_2\left(\frac{1}{20}\right)\times 5\right)\approx 1.22$，所以该条信息的价值已经远小于均匀骰子了。这很好理解，因为即使没人告知，也可以断定该枚骰子的结果肯定是发生在概率达到 0.9 的那一面。通过上述例子，可以很通俗地解释信息熵了：

某随机事件结果的种类越多，则该事件的熵越大。

某随机事件的各种可能发生的结果概率越均匀，则该事件的熵越大。

(3) 基尼指数(Gini index)

根据香农的论文，用信息熵衡量信息的价值有完美的理论依据。但有一个问题是：在当前计算机 CPU 架构中计算 $log_2(\ )$函数非常耗时，而在熵的计算公式中大量依赖该函数。因此出现了另一个衡量信息价值的指标——基尼指数：

$$\text{随机时间的基尼指数}=G(P_1,P_2,\ldots,P_n)=1-\sum_{i=1}^{n}P_i^2 \tag{18}$$

该公式有闭合的值域范围 0～1，数值越大表示事件越无序。基尼指数衡量信息价值的能力略逊于熵，但是它的公式由普通乘法和加法构成，并且有封闭的取值范围，因此非常适合在实际开发中使用。

(4) 建树策略

决策树的训练是一个利用已有样本从根开始逐步挑选特征并建树的过程。比如最开始的任务是寻找用哪一个特征作为根结点，而选择的依据是在用该特征进行数据划分后得到的信息增益最大。

说明：在用某个特征将数据集划分到不同的子树后，所有子树信息价值(熵/基尼指数)的和必定小于等于原来整体数据集的信息价值，信息增益用来衡量减少的程度。

设有建树函数 build( )，很容易写出递归训练决策树的算法思想。

```
def build(D=数据集，):
if D中所有数据目标值y都相同:
    return                                  # 本数据集可以作为叶子结点
for i in D中的所有特征:
    计算用i划分子树后获得的信息增益
if 所有的特征都没有大于零的信息增益:          # 已经无法再分
    return
被选择的特征x = 具有最大信息增益的特征
for sub in 按照x划分子树后的数据集:
    build(sub)                              # 递归寻找下一个决策特征
```

有了上述算法思想后还有很多细节要处理，比如：

- 信息增益如果直接用划分前后的熵差计算，则会导致倾向于先分类取值比较多的特征(回想抛硬币与掷骰子哪个信息熵更高的例子)。
- 连续值类型的特征如何计算？
- 强制要求每个叶子结点只有一个目标值容易导致预测过度拟合，如何适当归并叶子结点——剪枝(prune)？
- 叶子结点只能保存分类问题的目标值，那么如何用决策树处理回归问题？

根据对这些问题的处理策略不同，决策树家族又陆续衍生出了很多具体算法，目前有较大影响的是 ID3、C4.5 和 CART 算法。其中 ID3 只能使用熵的信息增益处理离散特征的分类问题；C4.5 在其中加入了：

- 使用信息增益比的概念去除先选择多值特征的倾向。
- 支持连续特征，在计算信息增益比之前首先将其离散化。
- 在训练后检测训练集的正确分类比，并剪枝产生错误较多的叶子结点。

而 CART 算法主要的不同在于使用基尼系数代替熵进行信息增益计算、只使用二叉树、并提供了解决回归问题的能力，因此综合看来 CART 比另两种算法能适应更多的场景。

3. 无监督学习：聚类

聚类分析，作为无监督学习的一种典型代表，旨在将对象集合划分为多个类别，其中每个类别由彼此相似的对象组成。该分析过程是一种多元统计方法，依据某一特征对研究对象进行分类，而忽略特征及变量间的因果关系。分类结果应确保类别间个体差异显著，而同类个体差异最小化。与回归分析、支持向量机和决策树等监督学习方法不同，聚类分析在缺乏输出信息和预设分类标准的情况下，仅依赖输入样本信息，依据样本相似度进行分组。

该算法致力于基于数据的内在结构，探索观察样本的自然群体，即集群。聚类算法在多个领域得到广泛应用，包括客户细分、新闻聚类、文章推荐等。

聚类算法的种类繁多，包括 k-均值(k-means)、吸引子传播(Afinity Propagation，AP)、层次聚类(Hierarchical/Agglomerative)、DBSCAN、CMM 等。

k-均值聚类作为一种通用型算法，其聚类度量基于样本点间的几何距离(即在坐标平面中的距离)。集群围绕聚类中心形成，通常呈现类球状且大小相似。鉴于其简单性和灵活性，k-均值聚类算法常被推荐给初学者，适用于解决大多数问题。其优点在于：作为聚类算法中最流行的代表，k-均值聚类因其快速、简便以及在有效的数据预处理和特征工程下的高度灵活性而备受青睐。然而，该算法也存在局限性：需要预先指定集群数量，而 k 值的选择往往难以确定；此外，若训练数据中的真实群体并非类球状，k-均值聚类可能会产生质量较差的集群。

AP 聚类算法是一种相对较新的聚类方法，该方法基于样本点间的图形距离(GraphDistances)来确定聚类。采用该聚类方法的聚类具有更小且不均等的规模。其优势在于：该算法无需预先指定聚类数量(尽管需要设定「samplep reference」和「damping」等超参数)。然而，AP 聚类算法的主要局限性在于其训练速度较慢，且需要大量内存资源，这使得其难以扩展至大规模数据集；此外，该算法还假定潜在的聚类形状为类球形。

层次聚类方法初始时将每个数据点视为一个独立的聚类，随后基于统一的标准逐步合并聚类，直至最终形成单一的聚类层次结构。该方法的主要优势在于：层次聚类无需假定聚类形状为类球形，且能够适用于大规模数据集。其局限性在于：层次聚类类似于 k-均值聚类，需要预先设定聚类数量(即算法完成后需要保留的层次数)。

DBSCAN 是一种基于密度的聚类算法，它将样本点的密集区域划分为一个聚类。最近，一种名为 HDBSCAN 的新算法被提出，它允许对密度聚类进行调整。DBSCAN 的优势在于：它不假定聚类形状为球形，并且其性能具有良好的可扩展性；此外，它允许样本点不被强制分配至任何聚类中，从而减少了异常数据的影响。然而，DBSCAN 的局限性在于：用户必须调整聚类密度相关的超参数，而 DBSCAN 对这些超参数的选择非常敏感。

3.1 k-means 聚类

相较于监督式分类/回归问题，聚类问题可视为一个仅包含数据特征而无目标值的数据集。k-means 算法作为聚类分析中直观性最强的算法之一，其应用过程中需预先设定聚类数目，随后算法依据样本向量间的距离最小化原则，将数据点分配至相应的簇中。

3.1.1 算法

1 目标

k-means 假定在聚类的每个分组中有一个中心点，算法的目的就是找到这些中心点的合适坐标，使得所有样本向量到其分组中心点距离的平方和最小。用数学公式表达，设训练集有 n 个样本，则 k-means 的目标是最小化：

$$C = \sum_{i=1\ldots N} \text{distant}(\text{centerOf}(\boldsymbol{x}_i) - \boldsymbol{x}_i)^2 \tag{19}$$

其中 x_i 是样本的特征向量，centerOf(x_i)是样本所在组的中心点向量(一个样本与哪个中心点更近，则被分为与该中心点一组)，distant()用于计算两个向量之间的距离。该公式计算结果在下文中简称 C 值。

2 步骤

假设数据集需要被划分为 M 个组，k-means 使用逐步迭代的方式找到每个中心点的合适位置。最初，可以用固定值或随机的方式选取任意 M 个点作为每组的中心点，然后迭代执行如下两步操作：

- 为每个中心点找到各自的样本数据，一个样本数据距离哪个中心点近则被划分给哪个中心点。
- 在每个组内用该组成员重新计算出中心点，新的中心点坐标是组内每个成员在各维度上的算数平均值，新中心点确定后的 C 值一定低于或等于原来的 C 值。

上述循环在到达算法收敛条件后退出。所谓收敛条件是 C 值低于某个要求或者每组中被推举的中心点不再发生变化。

注意：k-means 的中心点向量不一定是训练样本中某成员的位置。

3 演示

出于便于理解的目的，使用二维特征向量演示用 k-means 算法逐步找到中心点合适位置的算法过程。假设平面上有如图所示的若干个点，试图使用 k-means 算法将它们分为三组。

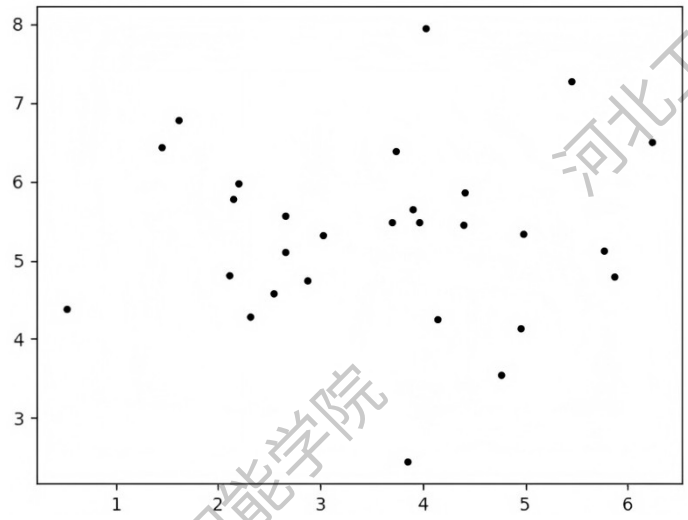

图 3 二维特征样本集

首先可以用随机方法选定三个中心点，然后划分样本数据到各自管辖范围内。使用三种图形和颜色分别表示三个组，第一个迭代的结果如图 4 所示

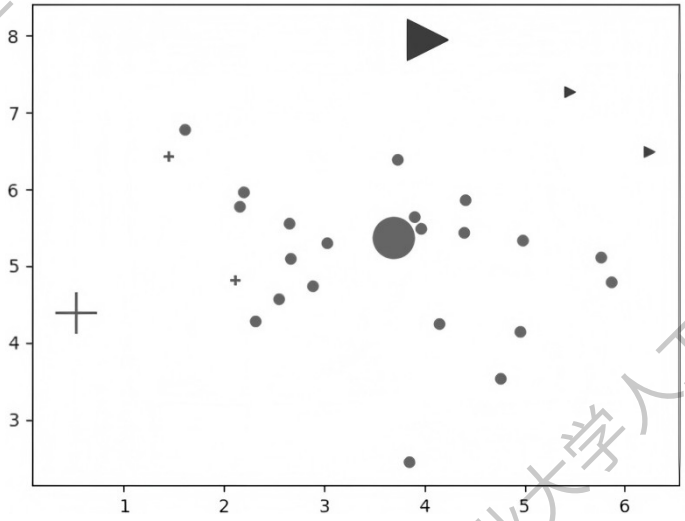

图 4 k-means 迭代的结果 1

在图示中，大散点代表了随机选取的三个样本数据作为初始聚类中心。在确定这些中心点后，所有训练数据样本(小散点)被分配至相应的类别中。经过计算，此时的聚类有效性指标 C 值约为 68.66。随后，启动第二轮迭代过程，依据当前的分类情况，重新选举出新的聚类中心，其结果如图 5 所示。

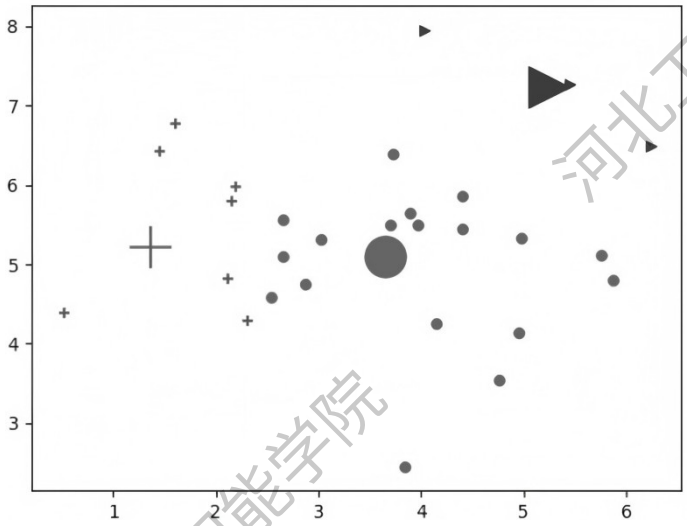

图 5 k-means 迭代的结果 2

在图 5 中，各组的“中心点”被重新定位至各自组内更为中心的位置，导致样本数据的分组情况发生相应变化。随着中心点的调整，系统的 *C* 值相应减小，当前 *C* 值约为 48.26。通过不断迭代，最终将达到一个临界点，即无法进一步调整中心点以减小 *C* 值。因此，系统将呈现图 6 所示的稳定状态。

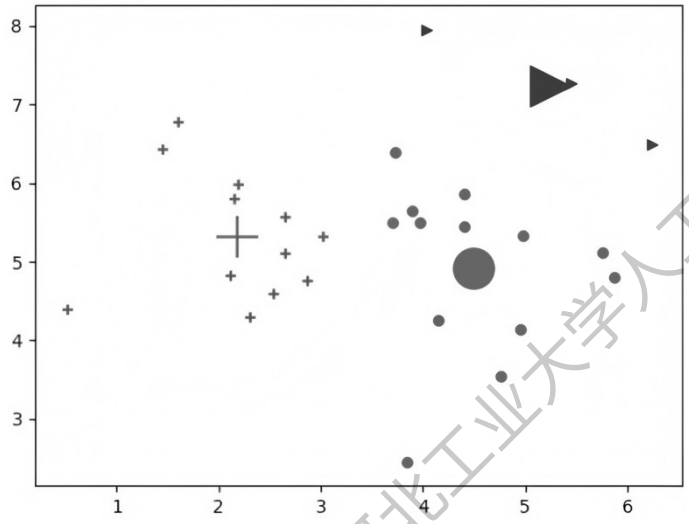

图 6 k-means 迭代的最终状态

此时，*C* 值达到 36.32，且无法进一步缩减，表明系统已实现收敛至稳定状态，k-means 算法的迭代过程至此终止。