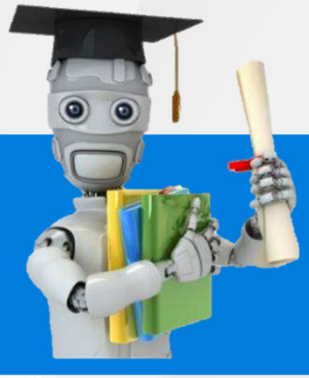


06 机器学习

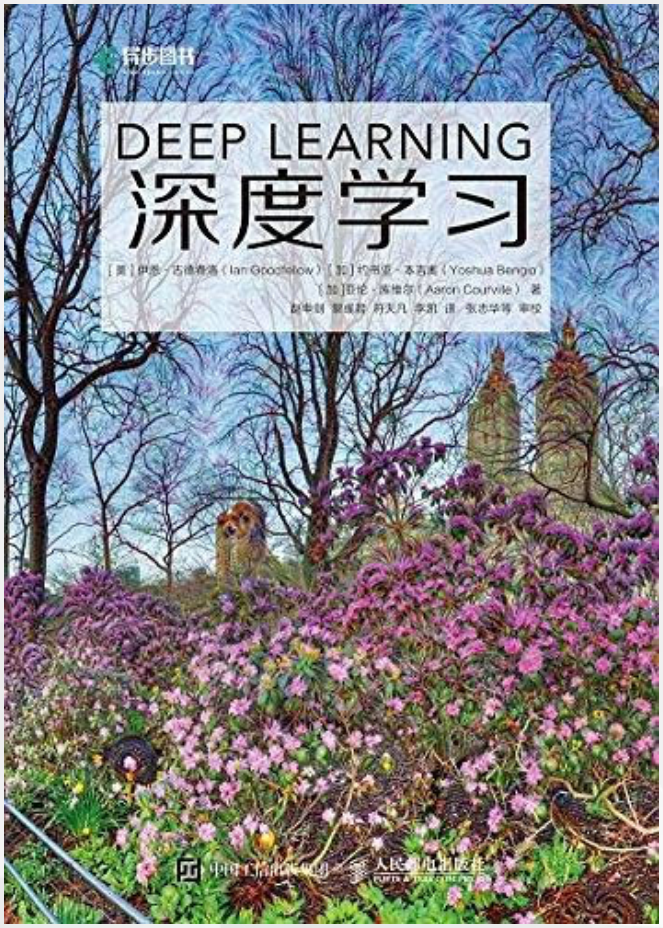
参考书目

机器学习

周志华

清华大学出版社

参考书目



机器学习

Ian Goodfellow
Yoshua Bengio

清华大学出版社

教学视频

吴恩达 《机器学习》课程

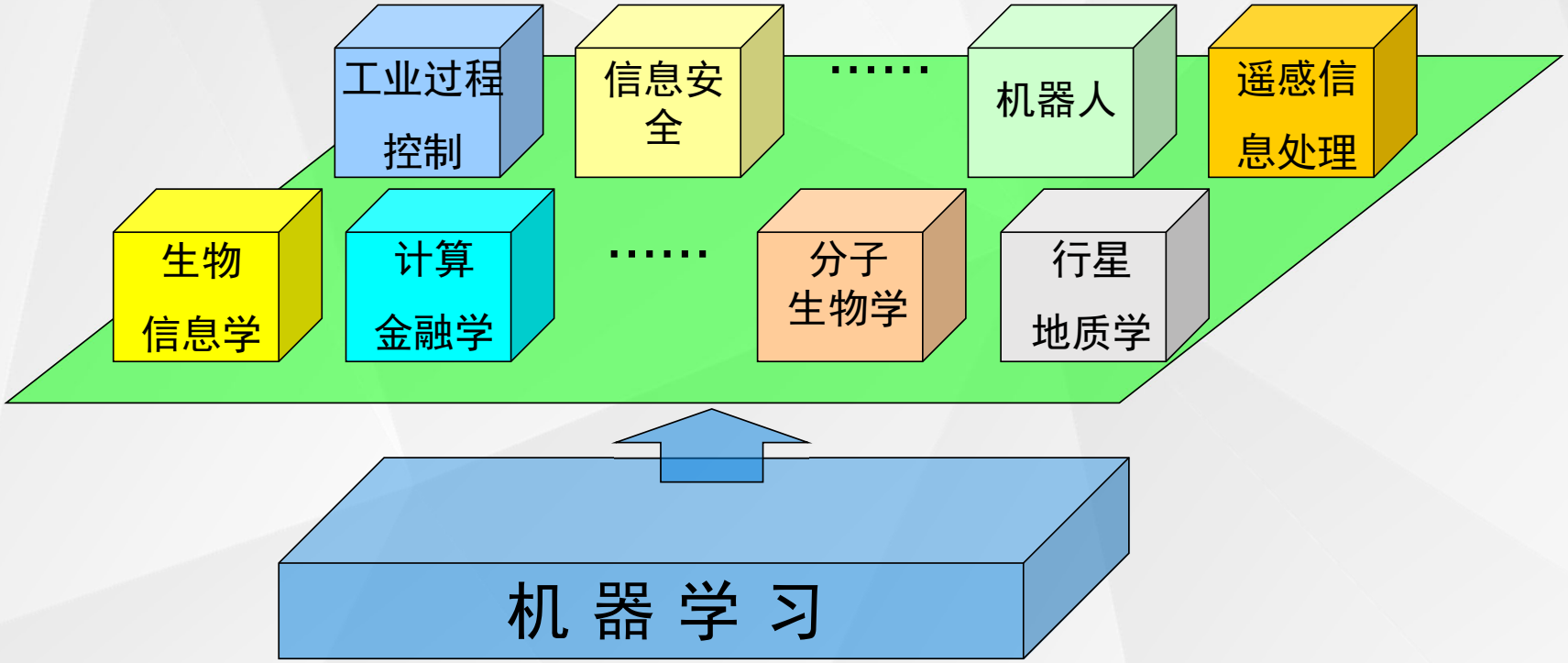
林轩田 《机器学习基石》和《机器学习技法》

台大李宏毅 《机器学习》课程

CV 方向的斯坦福 CS231n 课程；

NLP 方向的斯坦福 CS224n 课程；

6.1 机器学习概述



美国航空航天局JPL实验室的科学家在《Science》（2001年9月）上撰文指出：机器学习对科学研究的整个过程正起到越来越大的支持作用，……，该领域在今后的若干年内将取得稳定而快速的发展。

6.1.1 机器学习的定义

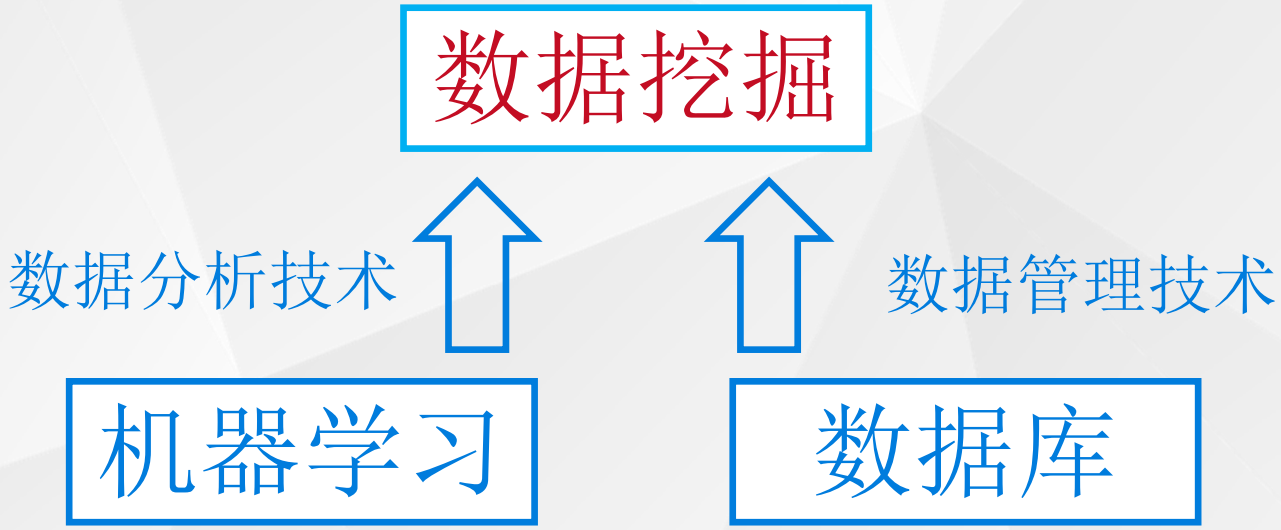
- 学习可能只是一个简单的联想过程，给定了特定的输入，就会产生特定的输出。如：狗
  - 命令“坐”
  - 行为“坐”
- 西蒙（Simon, 1983）：学习就是系统中的适应性变化，这种变化使系统在重复同样工作或类似工作时，能够做得更好或效率更高。
- 明斯基（Minsky, 1985）：学习是在人们头脑里（心理内部）有用的变化。
- 综合众多观点，学习是一个有特定目的的知识获取和能力增长过程，其内在行为是获得知识、积累经验、发现规律等，其外部表现是改进性能、适应环境和实现系统的自我完善。

机器学习本质

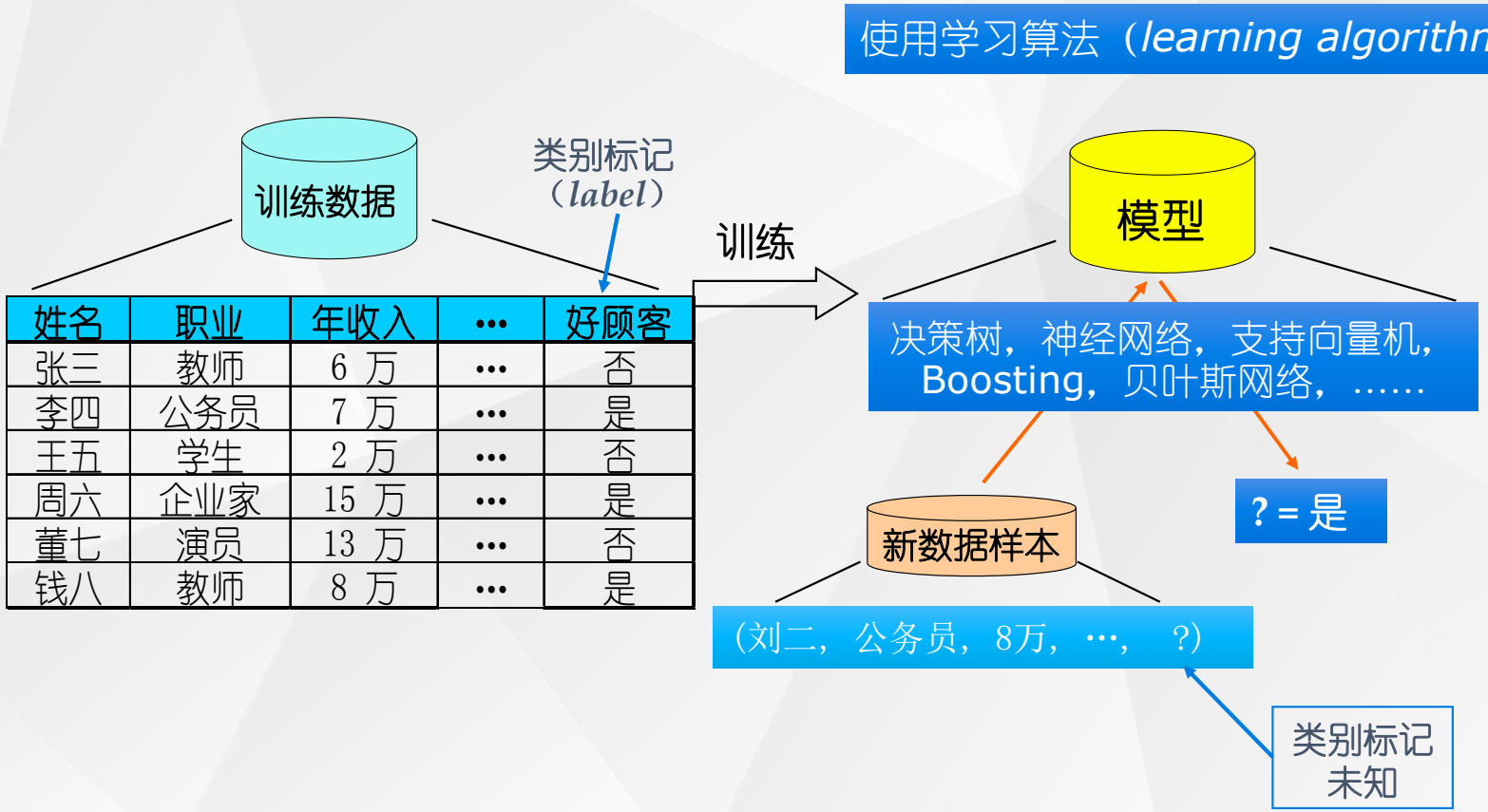
“假设用P来评估计算机程序在某任务类T上的性能，若一个程序通过利用经验E在T中任务上获得了性能改善，则我们就说关于T和P，该程序对E进行了学习”

机器学习致力于研究如何通过计算的手段，利用经验来改善系统自身的性能，从而在计算机上从数据中产生“模型”，用于对新的情况给出判断。

机器学习与数据挖掘



典型的机器学习过程



姓名	职业	年收入	…	好顾客
张三	教师	6 万	…	否
李四	公务员	7 万	…	是
王五	学生	2 万	…	否
周六	企业家	15 万	…	是
董七	演员	13 万	…	否
钱八	教师	8 万	…	是

6.1.1 机器学习的定义

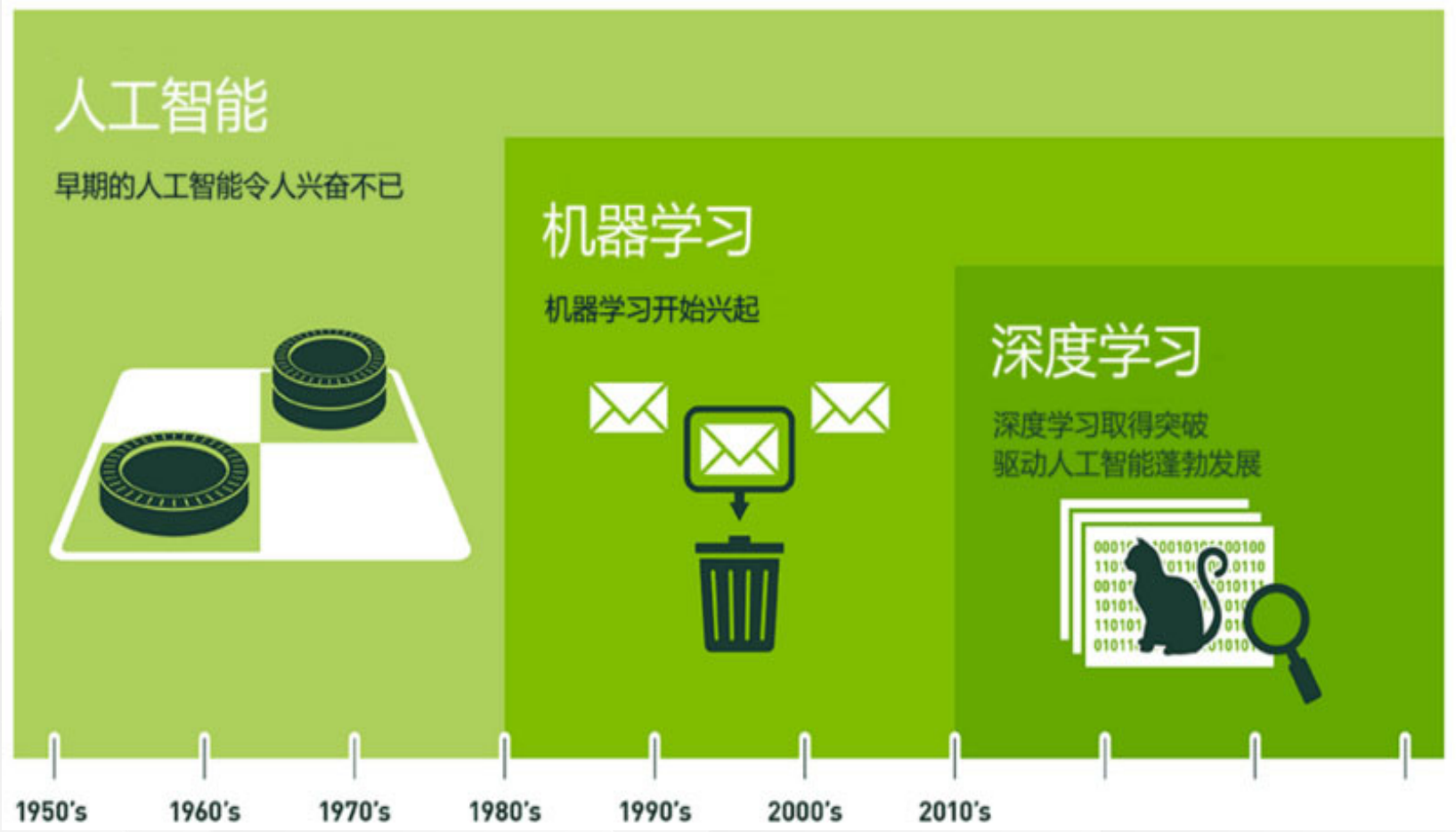
- 学习的成功是多种多样的：
  - 学习识别客户的购买模式以便能检测出信用卡欺诈行为，
  - 对客户进行扼要描述以便能对市场推广活动进行定位，
  - 对网上内容进行分类并按用户兴趣自动导入数据，
  - 贷款申请人的信用打分，
  - 燃气涡轮的故障诊断等。

6.1.1 机器学习的定义-学科角度

- 机器学习就是让机器来模拟人类的学习功能，是一门研究怎样用机器来模拟或实现人类学习活动的一门学科。机器学习是人工智能中最具有智能特征的前沿研究领域之一。

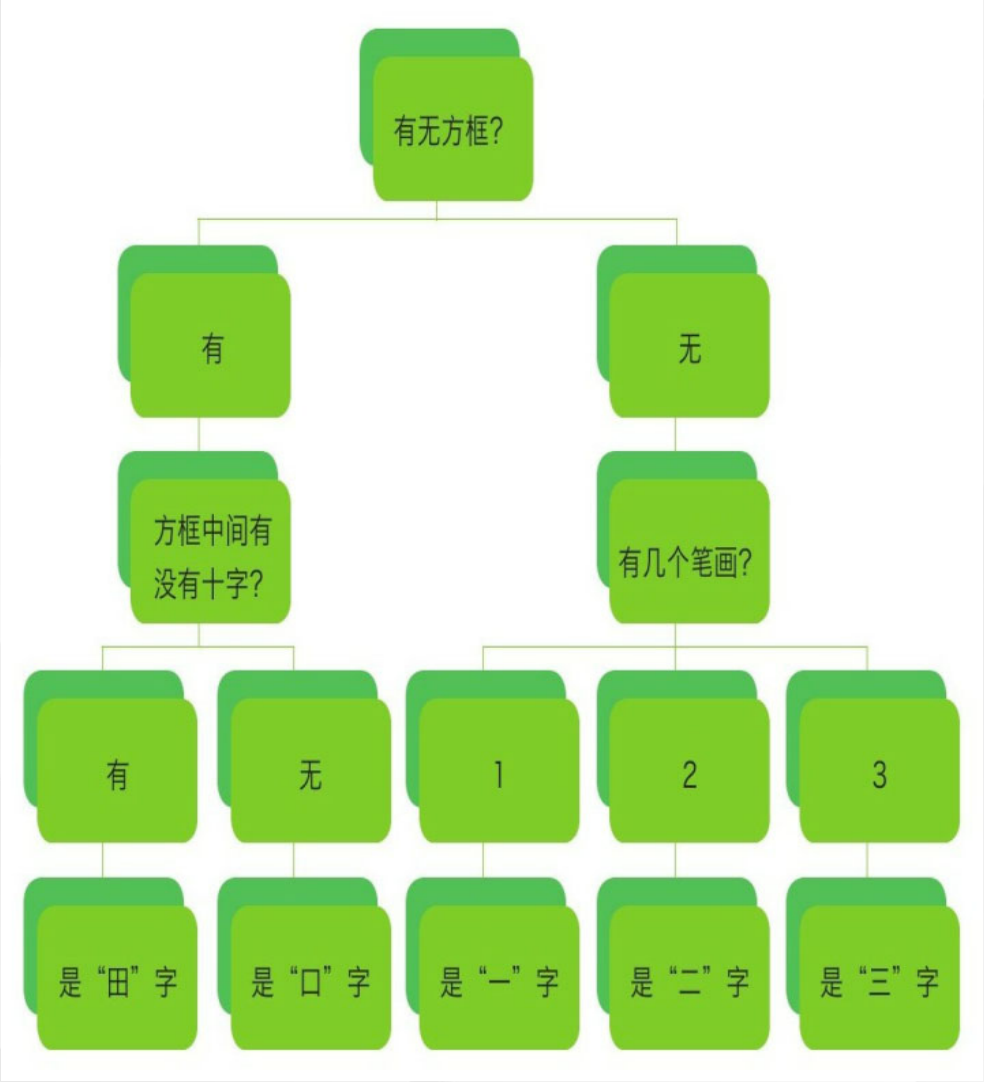
- 机器学习的研究主要集中在以下三个方面
  - 认知模型的研究
  - 理论学习的研究
  - 面向任务的研究

6.1.1 机器学习的定义



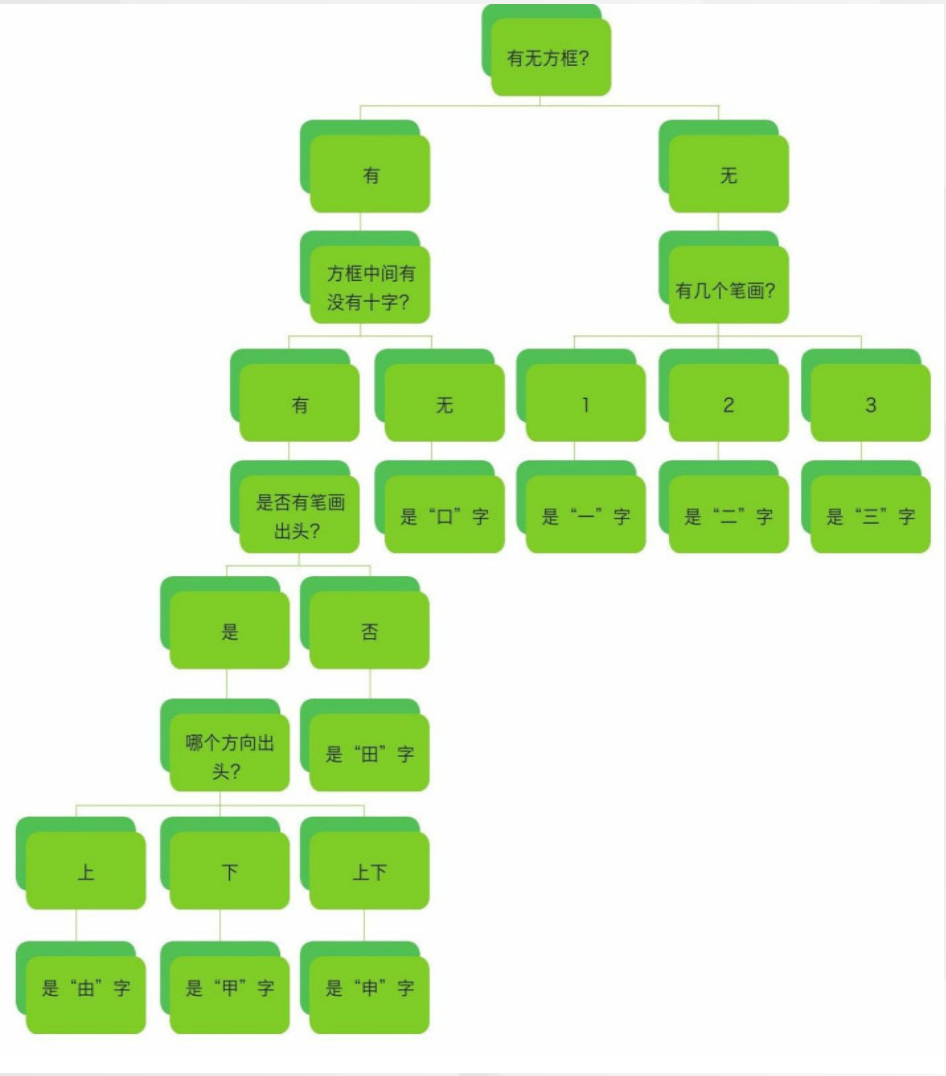
人工智能、机器学习和深度学习三者之间的关系

6.1.1 机器学习的定义



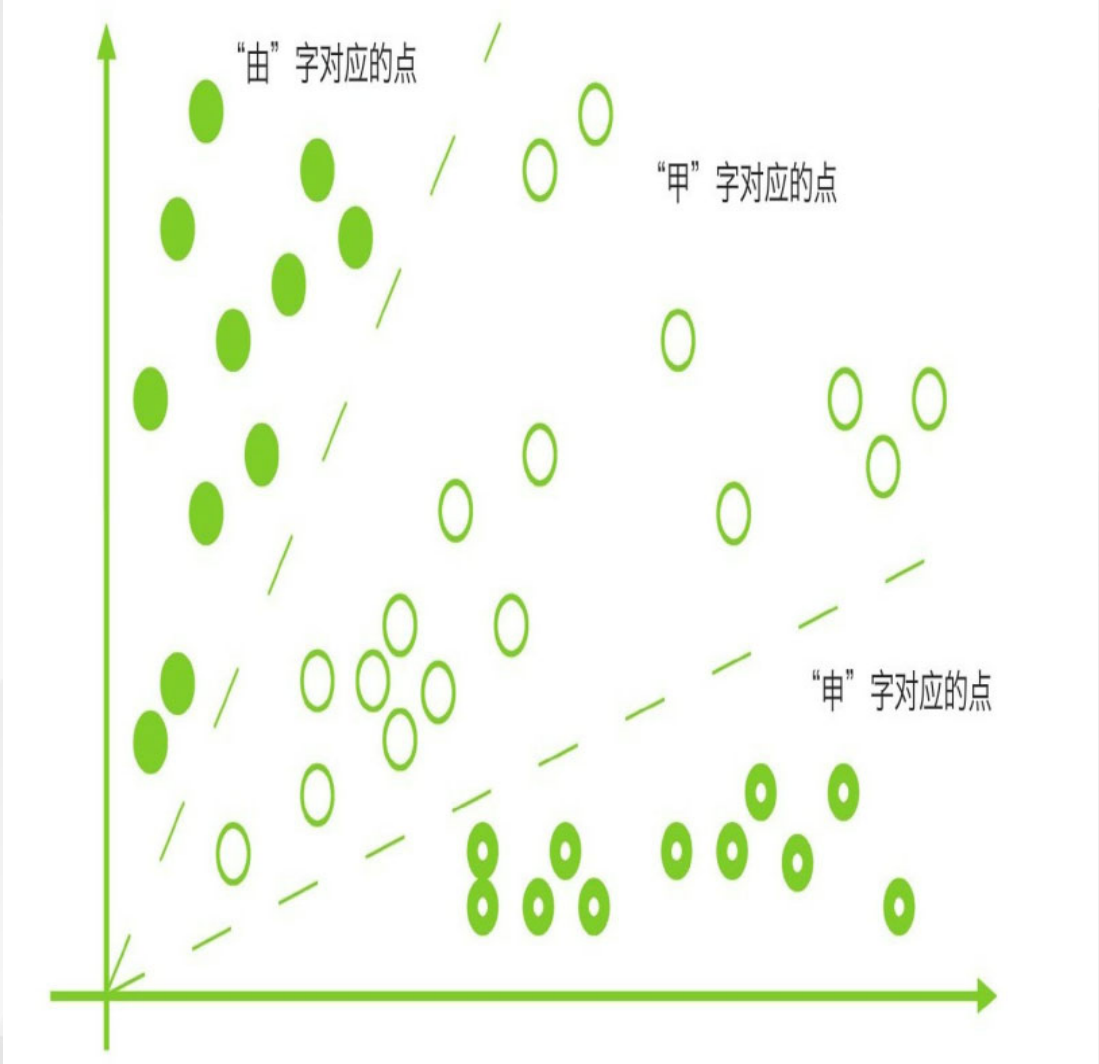
计算机分辨“一” “二” “三” “口” “田”的决策树

6.1.1 机器学习的定义



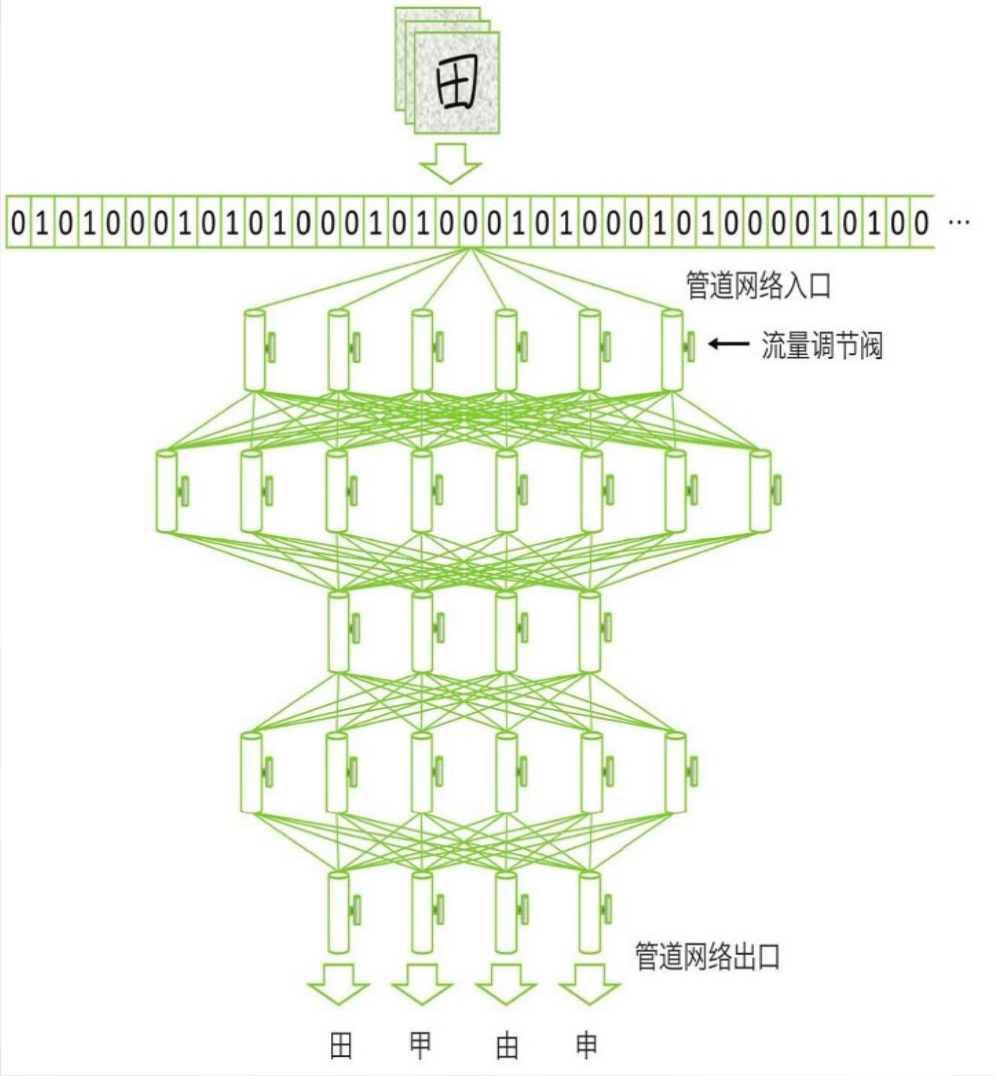
进一步扩展的决策树

6.1.1 机器学习的定义



使用空间分割法的机器学习

6.1.1 机器学习的定义



用“水管网络”来描述教计算机识字的深度学习过程

6.1.2 机器学习的发展历程

- 在20世纪50年代中叶到60年代中叶，属于热烈时期。
  - 在这个时期，所研究的是“没有知识”的学习，即“无知”学习；其研究目标是各类自组织系统和自适应系统；其主要研究方法是不断修改系统的控制参数以改进系统的执行能力，不涉及与具体任务有关的知识。
  - 指导本阶段研究的理论基础是早在20世纪40年代就开始研究的神经网络模型。
  - 我国研究了数字识别学习机。

6.1.2 机器学习的发展历程

- 20世纪60年代中叶至70年代中叶，属于冷静时期。
  - 主要研究模拟人类的学习过程，采用逻辑结构或图结构作为机器内部描述。
  - 代表性工作是温斯顿的结构学习系统和海斯、罗斯等人提出的基于逻辑的归纳学习系统
  - 只能学习单一的概念，而且未能投入实际应用。
  - 神经网络学习机因理论缺陷未能达到预期效果，机器学习的研究转入低潮。

6.1.2 机器学习的发展历程

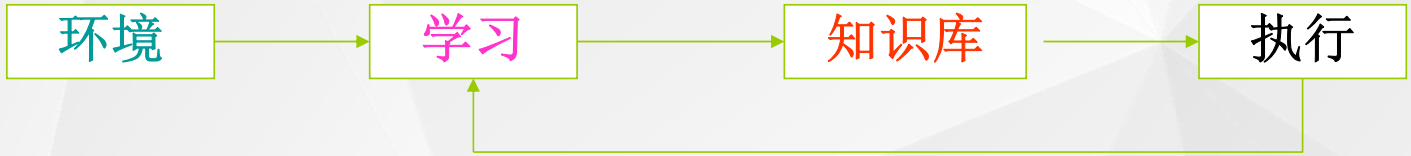
- 从20世纪70年代中叶至80年代中叶，称为复兴时期。
  - 在这个时期，人们从学习单个概念扩展到学习多个概念，探索不同的学习策略和各种学习方法。
  - 本阶段开始把学习系统与各种现实应用结合起来，促进了机器学习的发展。
  - 1980年，在CMU召开了第一届机器学习国际研讨会，标志着机器学习在全世界范围内的全面兴起。
  - 20世纪70年代末，中国科学院自动化研究所进行了质谱分析和模式文法推断研究，表明我国的机器学习研究得到恢复。

6.1.2 机器学习的发展历程

- 机器学习的最新阶段始于1986年。
  - 神经网络的研究再度兴起，使得机器学习进入了连接学习的研究阶段。
  - 传统的符号学习研究也取得了很大的发展。连接学习和符号学习各有所长，并具有很大的互补性。
  - 深度学习异军突起，已经成为机器学习研究中的一个重要领域，其动机在于建立、模拟人脑进行分析学习的神经网络，并模仿人脑的机制来解释数据。

6.1.3 学习系统的基本模型★

- 学习系统的基本结构如图所示。



- 环境向系统的学习部分提供某些信息，
- 学习部分利用这些信息修改知识库，以增进系统执行部分完成任务的效能，
- 执行部分根据知识库完成任务，同时把获得的信息反馈给学习部分。
- 在具体的应用中，环境、知识库和执行部分决定了具体的工作内容，学习部分所需要解决的问题完全由上述三部分确定。

6.1.3 学习系统的基本模型★

- 影响学习系统设计的最重要的因素是环境向系统提供的信息。
- 知识库里存放的是指导执行部分动作的一般原则，但环境向学习系统提供的信息却是各种各样的。
- 如果信息的质量比较高，与一般原则的差别比较小，则学习部分就比较容易处理。
- 如果向学习系统提供的是杂乱无章的指导执行具体动作的具体信息，则学习系统需要在获得足够数据之后，删除不必要的细节，进行总结推广，形成指导动作的一般原则，放入知识库。
- 这样，学习部分的任务就比较繁重，设计起来也较为困难。

6.1.4 机器学习分类

- 机器学习算法可以按照不同的标准来进行分类。
  - 比如按照使用函数的不同，机器学习算法可以分为线性模型和非线性模型
  - 按照学习准则的不同，机器学习算法可以分为统计方法和非统计方法
  - 一般来说，通常会按照训练样本提供的信息以及反馈方式的不同，将机器学习算法分为监督学习、无监督学习和强化学习。 ★

6.1.4 机器学习分类

- 监督学习（Supervised Learning）
  - 通过已有的训练样本（即已知数据以及其对应的输出）进行训练，从而得到一个最优模型，再利用这个模型将所有新的数据样本映射为相应的输出结果，对输出结果进行简单的判断从而实现分类的目的。
  - 根本目标是训练机器学习的泛化能力。
  - 典型算法有：决策树、支持向量机（SVM）、*k*-近邻算法（KNN）等。
  - 在文字识别、垃圾邮件分类与拦截、网页检索、基因诊断以及股票预测等各个领域都有着广泛的应用

6.1.4 机器学习分类

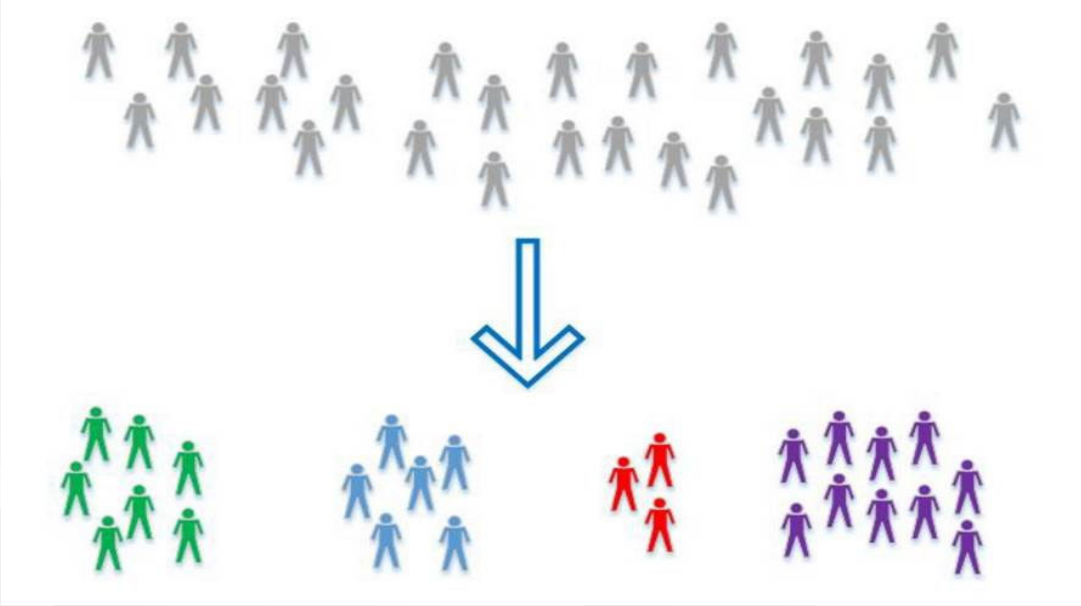
监督学习:给历史数据打上标签，运用模型预测结果。

监督学习需要标记数据

数据标记员:隐身于人工智能产业背后的工兵

6.1.4 机器学习分类

- 无监督学习（Unsupervised Learning）
  - 指用来学习的数据在没有任何类别信息以及给定目标值的情况下，通过学习寻求数据间的内在模式和统计规律，从而获得样本数据的结构特征。
  - 根本目标是在学习过程中根据相似性原理进行区分。
  - 典型算法有：*k*-均值(*K-means*)、DBSCAN密度聚类算法、最大期望算法等
  - 在人造卫星故障诊断、视频分析、社交网站解析和异常检测以及数据可视化以及作为监督学习方法的预先处理方面有着广泛的应用

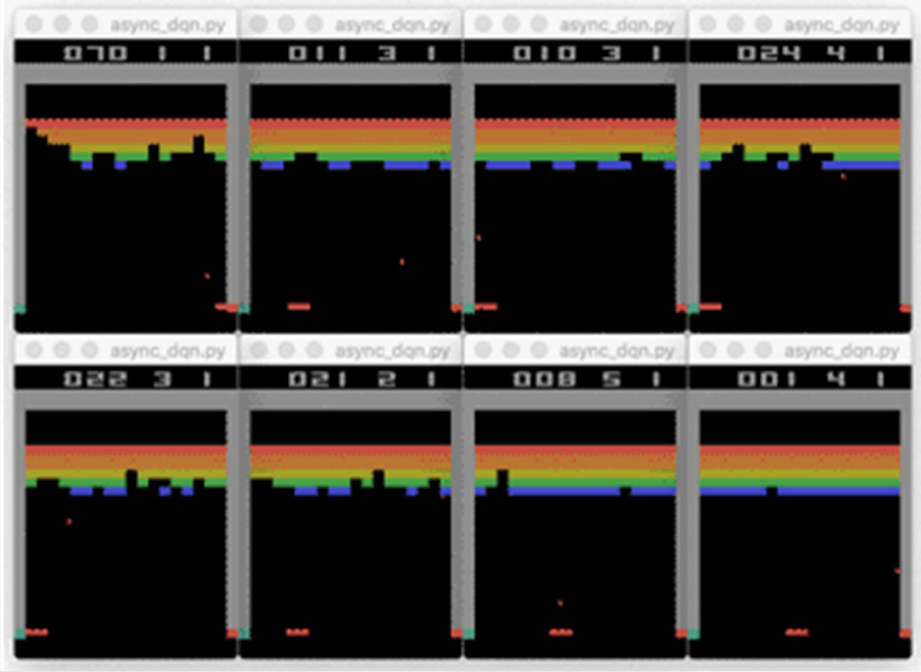


6.1.4 机器学习分类

无监督学习:从信息出发自动寻找规律，并将其分成各种类别，有时也称"聚类问题"。

6.1.4 机器学习分类

- 强化学习（Reinforcement Learning）
  - 通过交互来学习的机器学习算法。
  - 智能体根据环境状态做出一个动作，并得到即时或延时的激励。智能体在和环境的交互中不断学习并调整策略，以取得最大化的期望。
  - 主要用于智能控制领域，比如机器人控制、电梯调度、电信通讯等，如今已经在自动驾驶、自然语言处理和语音交互领域都有相关的应用。



6.1.4 机器学习分类

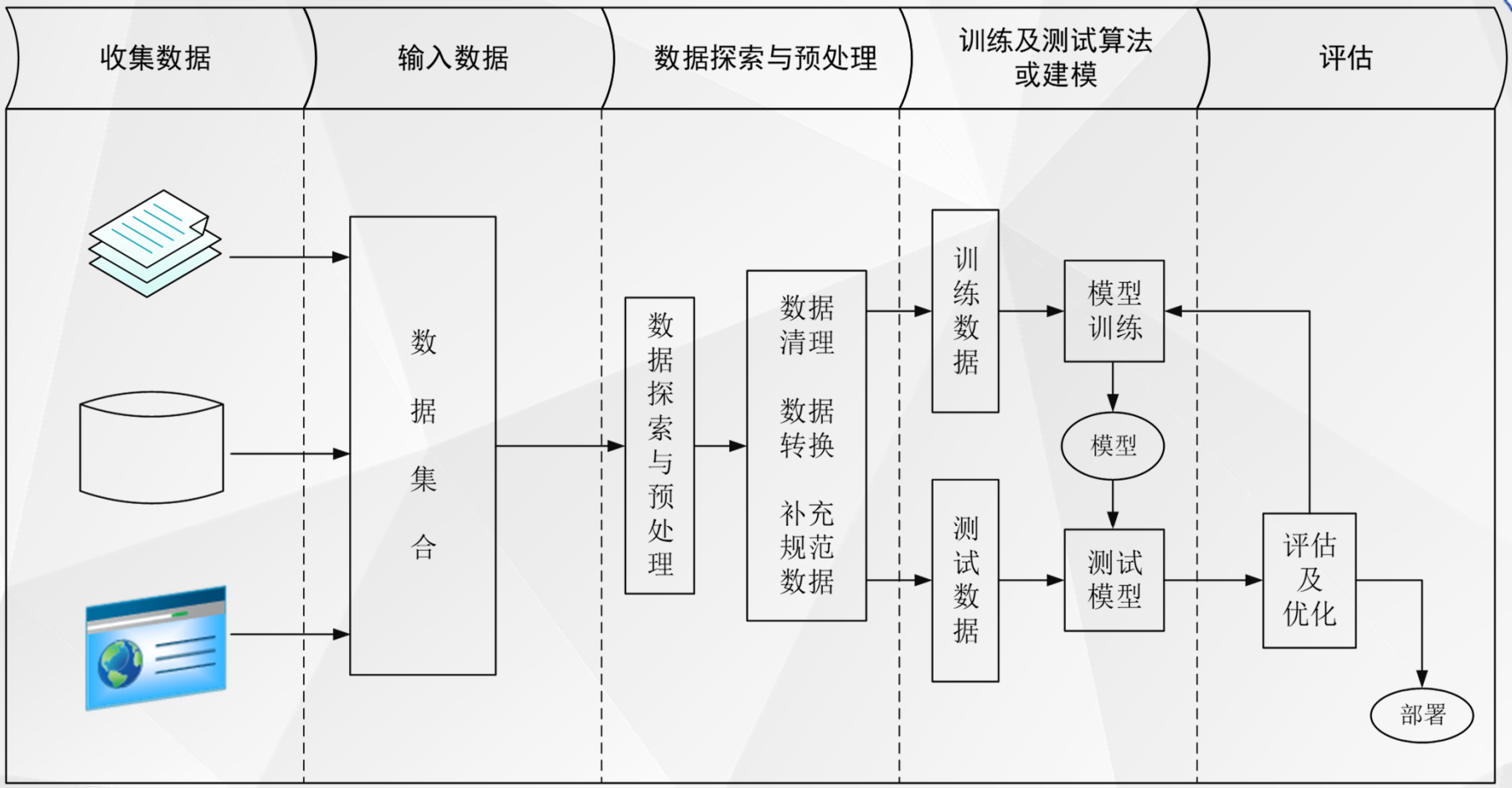
	监督学习	无监督学习	强化学习
输入	有标记数据	无标记数据	决策过程
反馈方式	直接反馈	无反馈	激励
目标	分类、预测	聚类、降维	动作

三种机器学习类型的比较

6.1.5 机器学习与人类思考的类比

机器学习与人类思考的类比

6.1.6 解决机器学习问题的一般步骤



机器学习算法处理问题的一般流程

6.1.7 相关的基本术语—数据

特征　　标记

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

训练集

1	青绿	蜷缩	沉闷	?

测试集

6.1.7 相关的基本术语—任务

- 预测目标：
  - 分类:离散值
    - 二分类:好瓜;坏瓜
    - 多分类:冬瓜;南瓜;西瓜
  - 回归:连续值

    瓜的成熟度(色泽、根蒂、敲声)
  - 聚类:无标记信息

6.1.7 相关的基本术语—任务

- 有无标记信息
  - 监督学习：分类、回归
  - 无监督学习：聚类
  - 半监督学习：两者结合

6.1.7 相关的基本术语—泛化能力

机器学习的目标是使得学到的模型能很好的适用于“新样本”,而不仅仅是训练集合，我们称模型适用于新样本的能力为泛化(generalization)能力。

6.1.8 机器学习编程语言

LISP

- 见证了人工智能的早期发展和实践
- 代表系统：SHRDLU

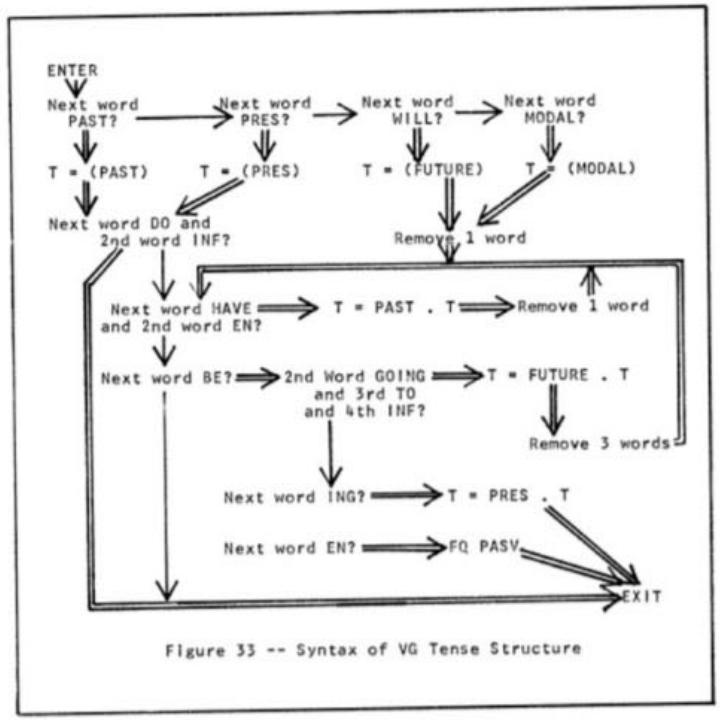
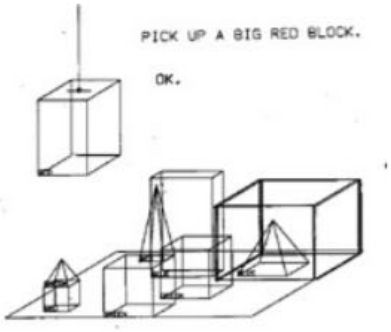
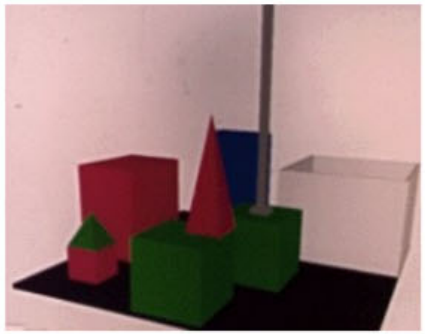


Figure 33 -- Syntax of VG Tense Structure





```
(CLAUSE MAJOR IMPER PRT TRANS)  (pick up a big red block)
     (VG IMPER)                          (pick)
          (VB MVB INF TRANS VPRT)         pick
     (PRT)                               up
     (NG OBJ OBJ1 DET INDEF NS)     (a big red block)
          (DET INDEF NS)                  a
          (ADJ)                           big
          (ADJ)                           red
          (NOUN NS)                       block.
```

Figure 41 -- Syntactic Structure 3

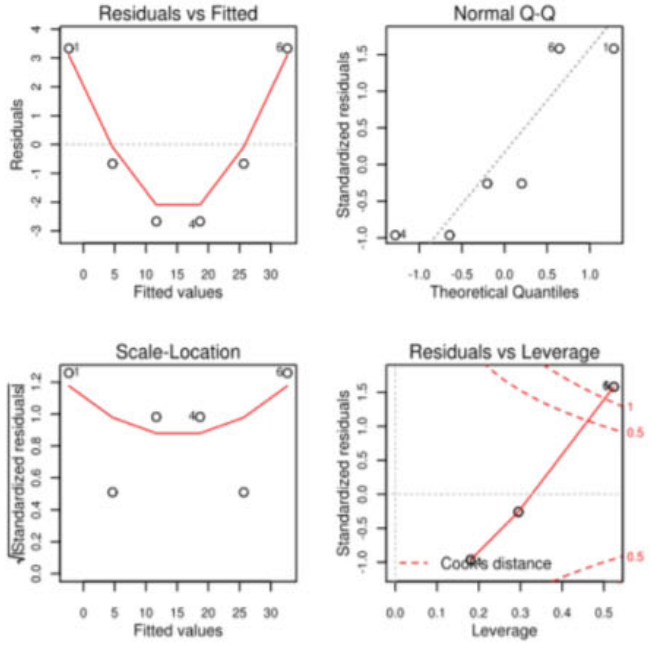
Source: Terry Winograd, 1971, Procedures as a Representation for Data in a Computer Program for Understanding Natural Language

6.1.8 机器学习编程语言

R

- 数据科学家和机器学习领域的传统语言
- 目前仍在机器学习领域扮演重要角色

```
kalythos <- data.frame(x = c(20,35,45,55,70), n = rep(50,5),
                       y = c(6,17,26,37,44))
kalythos$Ymat <- cbind(kalythos$y, kalythos$n - kalythos$y)
fmp <- glm(Ymat ~ x, family = binomial(link=probit), data = kalythos)
fml <- glm(Ymat ~ x, family = binomial, data = kalythos)
summary(fmp)
summary(fml)

fit <- rpart(Kyphosis ~ Age + Number + Start, data = kyphosis)
fit2 <- rpart(Kyphosis ~ Age + Number + Start, data = kyphosis,
              parms = list(prior = c(.65,.35), split = "information"))
fit3 <- rpart(Kyphosis ~ Age + Number + Start, data = kyphosis,
              control = rpart.control(cp = 0.05))
par(mfrow = c(1,2), xpd = NA) # otherwise on some devices the text is clipped
plot(fit)
text(fit, use.n = TRUE)
plot(fit2)
text(fit2, use.n = TRUE)
```



Source: https://cran.r-project.org/doc/manuals/r-release/R-intro.html
Source: https://cran.r-project.org/web/packages/rpart/rpart.pdf

6.1.8 机器学习编程语言

Python

- 机器学习（特别是深度学习）领域的第一语言
- 在TensorFlow等主流框架中占据核心位置
- 不仅仅在科研领域流行，也在工程领域流行
- 可以和C++、Java、JavaScript等主流工程开发语言高效衔接
- Python本身存在的运行性能问题近年已经得到了巨大的改进
- 基于Python的基础开源库和开发工具构成了一个强大、便捷的生态系统
  - jupyter notebook, pip/pip3, docker, etc.
  - panda, numpy, scikit-learn, etc.

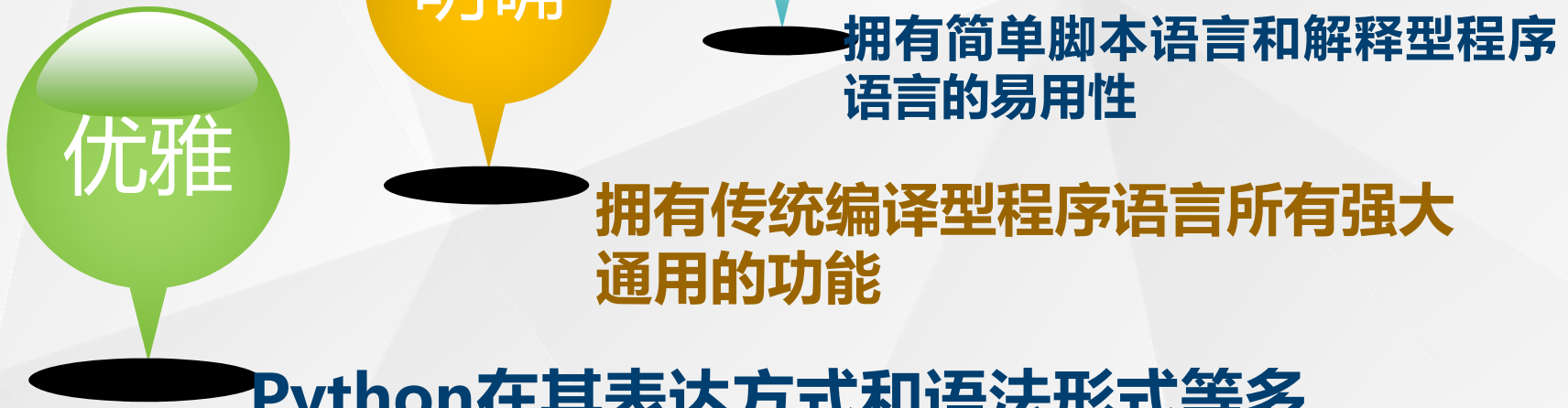

```python
# Input Layer
input_layer = tf.reshape(features["x"], [-1, 28, 28, 1])

# Convolutional Layer #1
conv1 = tf.layers.conv2d(
    inputs=input_layer,
    filters=32,
    kernel_size=[5, 5],
    padding="same",
    activation=tf.nn.relu)

# Pooling Layer #1
pool1 = tf.layers.max_pooling2d(inputs=conv1, pool_size=[2, 2], strides=2)

# Convolutional Layer #2 and Pooling Layer #2
conv2 = tf.layers.conv2d(
    inputs=pool1,
    filters=64,
    kernel_size=[5, 5],
    padding="same",
    activation=tf.nn.relu)
pool2 = tf.layers.max_pooling2d(inputs=conv2, pool_size=[2, 2], strides=2)

# Dense Layer
pool2_flat = tf.reshape(pool2, [-1, 7 * 7 * 64])
dense = tf.layers.dense(inputs=pool2_flat, units=1024, activation=tf.nn.relu)
dropout = tf.layers.dropout(
    inputs=dense, rate=0.4, training=mode == tf.estimator.ModeKeys.TRAIN)

# Logits Layer
logits = tf.layers.dense(inputs=dropout, units=10)
```

6.1.8 机器学习编程语言

- Python应用实例

6.1.8 机器学习编程语言

- Python的哲学



拥有简单脚本语言和解释型程序语言的易用性

拥有传统编译型程序语言所有强大通用的功能

Python在其表达方式和语法形式等多个方面均体现其优雅

语句块的强制缩进

语法糖：x, y = y, x

6.1.8 机器学习编程语言

- 强大的Python库

6.1.8 机器学习编程语言

- 实例：电影圈的那些事儿

6.1.8 机器学习编程语言

- Jieba、wordcloud- 电影主题分析

Tag：犯罪

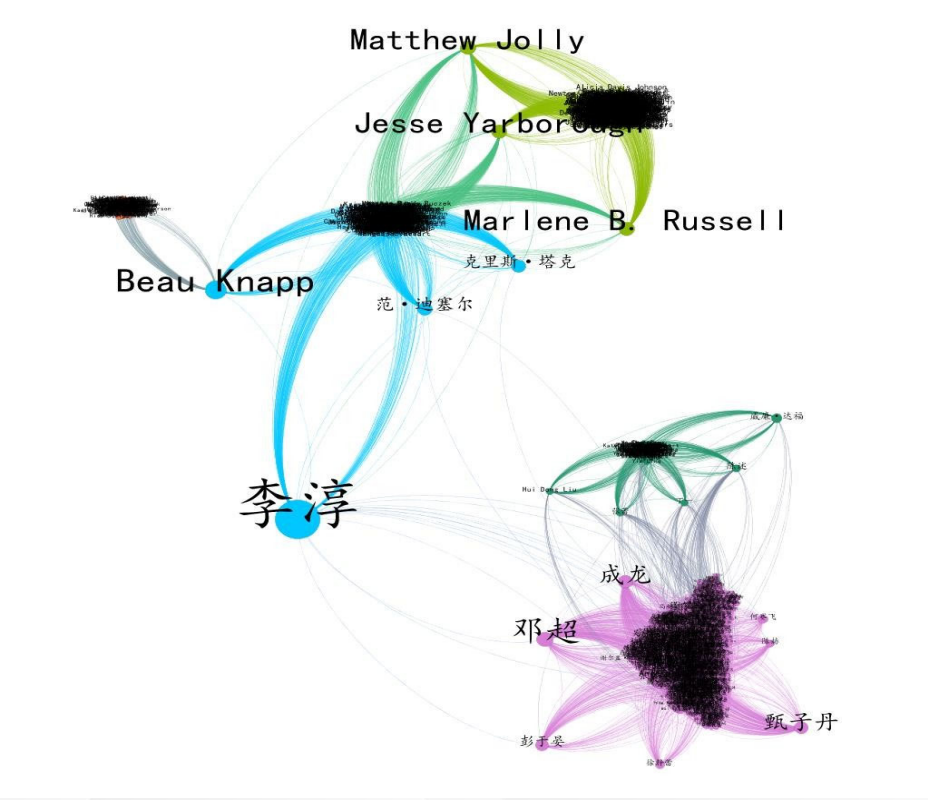
Tag：科幻

Tag：战争

Tag：爱情

6.1.8 机器学习编程语言

- Networkx – 电影合作网络分析



6.1.8 机器学习编程语言

计算机视觉的基本应用：面包识别

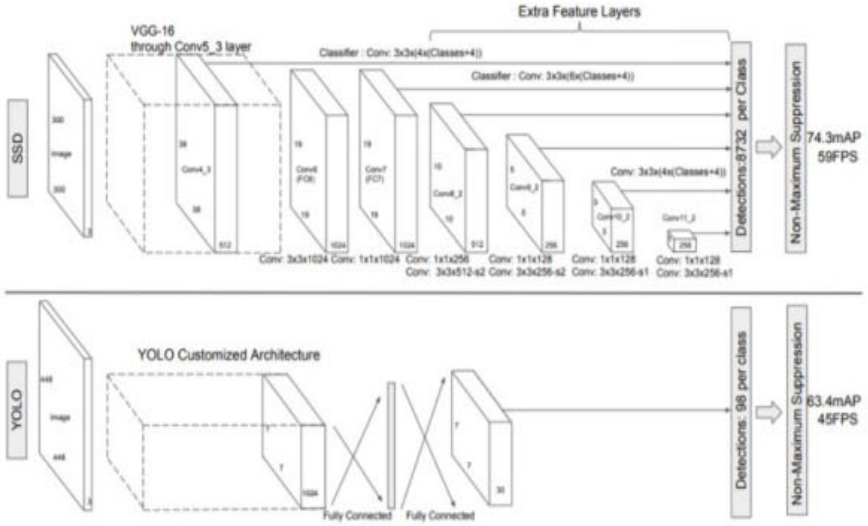


Fig. 2: A comparison between two single shot detection models: SSD and YOLO [5]. Our SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences. SSD with a 300 × 300 input size significantly outperforms its 448 × 448 YOLO counterpart in accuracy on VOC2007 `test` while also improving the speed.

商品识别工程的挑战

- 算法层面
  - 多物体的分类（Classification）和检测（Detection）
  - 部分重叠物体的检测（Overlapping Instances Detection）
  - 深度信息（Depth Information）如何帮助视觉检测？重力信息呢？
- 算法封装层面
  - 如何将模型压缩到 ## MB 以内？
  - 如何将模型封装成目标平台的部署格式？
- 模型训练和管理层面
  - 如何快速支持新的商品品类识别
  - 如何快速/实时更新模型
  - 识别结果是否能反馈到模型训练过程，以持续改进识别能力？

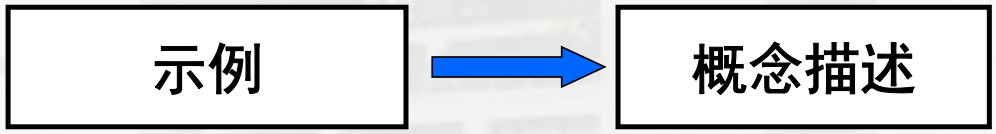
商品识别工程的挑战（续）

- 计算层面
  - 推理（Inference）计算应该发生在CPU上、GPU上还是FPGA上？
  - 推理计算的效率是否满足业务需求？
  - 推理计算的功耗是否满足节能需求？
- 硬件平台层面
  - 推理计算使用PC、移动终端还是嵌入式系统？系统的功耗和稳定性如何？
  - 摄像头的类型（单目、深度），视角，分辨率，功耗，稳定性……
  - 灯光的设计：色温、位置、照射方向、环境光干扰、功耗、稳定性……
- 网络服务层面
  - 计算终端数量达到一定规模后，终端和云服务之间的数据同步如何优化？
  - 如何与现有的POS系统整合？如何整合支付模块？如何与ERP系统整合？

6.2 归纳(示例)学习

- 在专家系统中经常遇到
- 示例学习任务：
  - 从一系列示例出发：
    - 正例；
    - 反例；
  - 生成一个反映这些示例本质的定义（概念描述）：
    - 覆盖所有的正例，而不包含任何反例；
    - 可用来指导对新例子的分类识别；

解描述



示例学习

1、概念描述的搜索和获取

例子空间和假设空间

- 例子空间：
  - 所有可能的正例、反例构成的空间；
- 假设空间（概念空间）：
  - 所有可能的假设（概念描述）构成的空间；
- 假设空间中每一假设都对应于例子空间中一个子集
  - 子集中的例子均是该假设的例子；

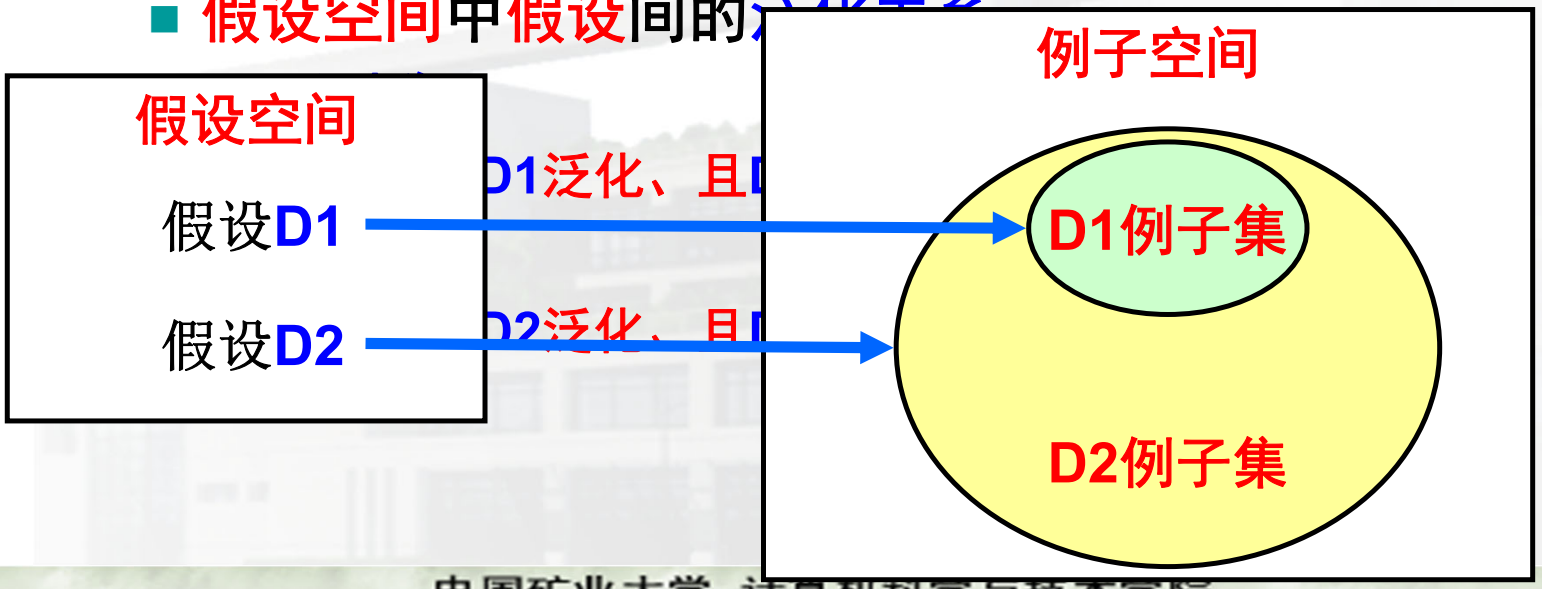
示例学习

- 1、概念描述的搜索和获取
  - 假设的泛化和特化：
    - D1对应例子集是D2对应例子集的子集；
    - D2比D1泛化；
    - D1比D2特化；
  - 假设空间中假设间的泛化关系

假设空间

假设D1

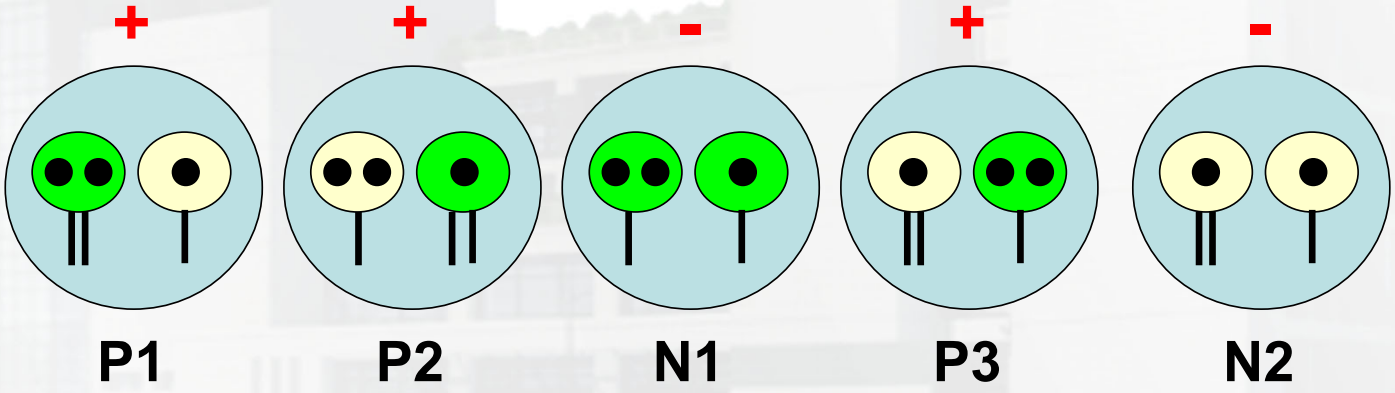
假设D2



示例学习

1、概念描述的搜索和获取

- 例1：病态细胞的分类识别（找到病态细胞的概念）
  - 每个细胞由2个细胞体组成；
  - 每个细胞体具有3个属性——胞核数(1-2)，尾巴数(1-2)及染色状（浅或深）；
  - 细胞P1, P2, P3有病状X；
  - N1, N2是正常细胞；



示例学习

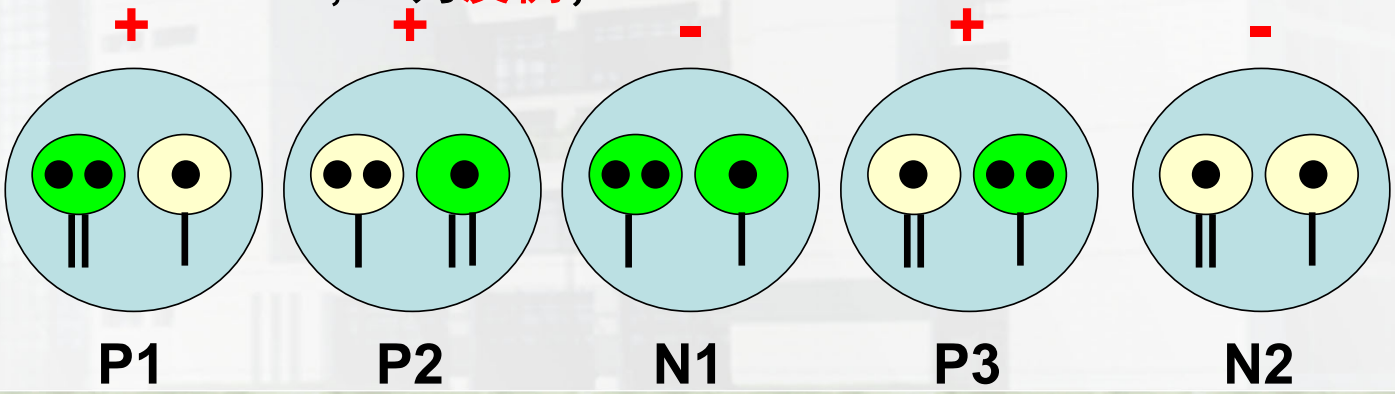
1、概念描述的搜索和获取

- 例1：病态细胞的分类识别

学习任务

从例子空间中归纳出有病状X的细胞概念描述

- 细胞i表示为((2, 2, 深), (1, 1, 浅))
- 例子空间由P1,P2,P3,N1,N2组成；
  - P1,P2,P3为正例；
  - N1,N2为反例；

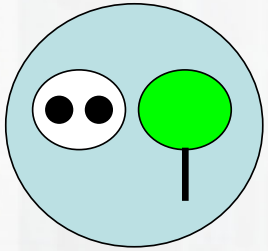


示例学习

1、概念描述的搜索和获取

- 例1：病态细胞的分类识别
  - 假设空间表示为假设的集合；
  - 假设不必给每个特性（属性）都指明应取值：
    - 假设a：{(2, ?, ?) (?, 1, 深)}，表示：
    - 如果：
      - 细胞中一个细胞体有2个胞核；
      - 另一个有1个尾巴，且染色是深的；
    - 则：
      - 该细胞有病症X。
    - “?”指
      - 相应的属性对病细胞的判断是无关紧要；

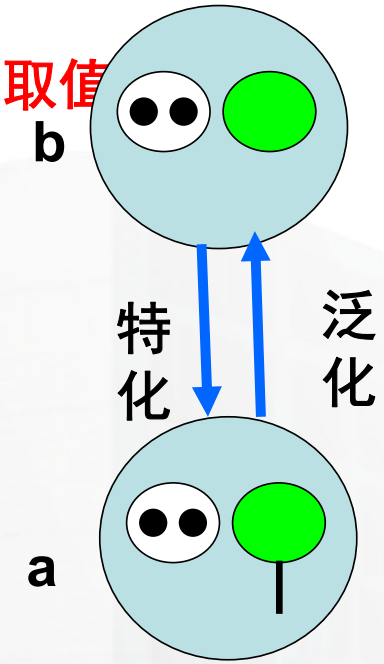
a



示例学习

1、概念描述的搜索和获取

- 例1：病态细胞的分类识别
  - 假设空间表示为假设的集合；
  - 假设不必给每个特性（属性）都指明应取值
    - 假设a：{(2, ?, ?) (?, 1, 深)}
    - 假设b： {(2, ?, ?) (?, ?, 深)}
    - 覆盖更多的例子



假设b比假设a泛化

假设a比假设b特化

示例学习

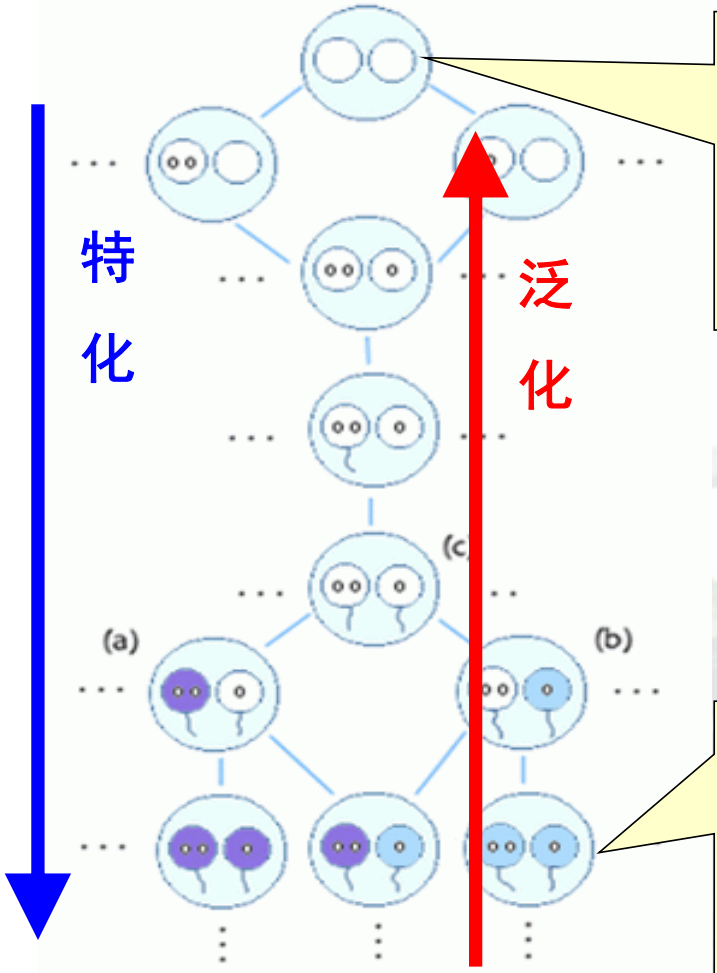


图6.5 假设空间的半序图

顶层假设

(1)最泛化的概念描述；

(2)不指定任何具体的特性值 ；

(3)表示为{(？ ？ ?)，(？ ？ ?)}；

完全的假设空间

底层假设

(1)最特化（具体）的概念描述；

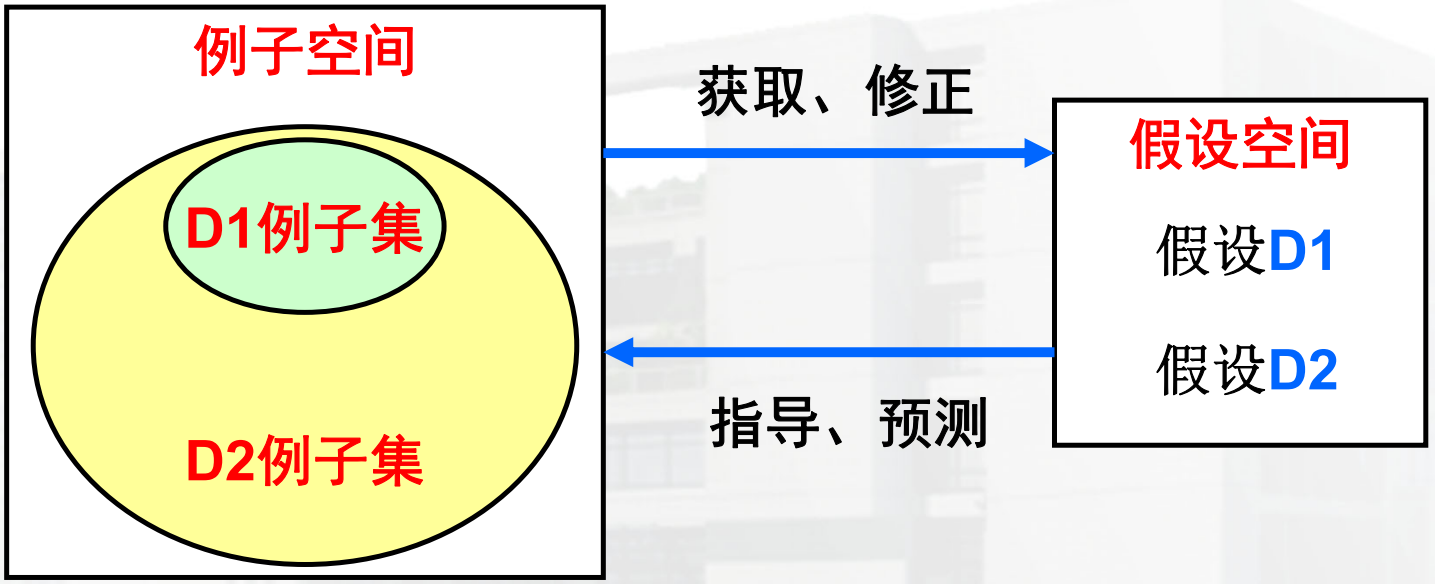
(2)所有特性都给定特性值；

(3)对应于例子空间中的一个例子；

示例学习

1、概念描述的搜索和获取

- 示例学习的过程（T.Mitchell, 1982）：
  - 在假设空间中搜索的过程。
- 学习过程中假设空间可以动态扩展；



示例学习

1、概念描述的搜索和获取

假设空间中的搜索方法

①特化搜索

- 从最泛化的假设（概念描述）出发；
- 每次取用一个新的例子，产生一些特化的描述；
- 直到产生出足够特化的解描述；

②泛化搜索

- 从最特化的假设（例子空间中的一个正例）开始；
- 每次取用一个新的例子，产生一些泛化的描述；
- 直到产生出足够泛化的解描述。

示例学习

1、概念描述的搜索和获取

- 假设空间中的搜索方法
  - ①特化搜索
  - ②泛化搜索
  - 大多数示例学习方法都采用这二种方法或这二个方法的结合。
- 任何的示例学习的过程都可以看成假设空间中的搜索过程，不同的搜索方式对应于不同的学习策略：
  - ①逐步泛化的学习策略——自底向上的泛化搜索；
  - ②逐步特化的学习策略——自顶向下的特化搜索；
  - ③双向学习策略——双向搜索。

示例学习

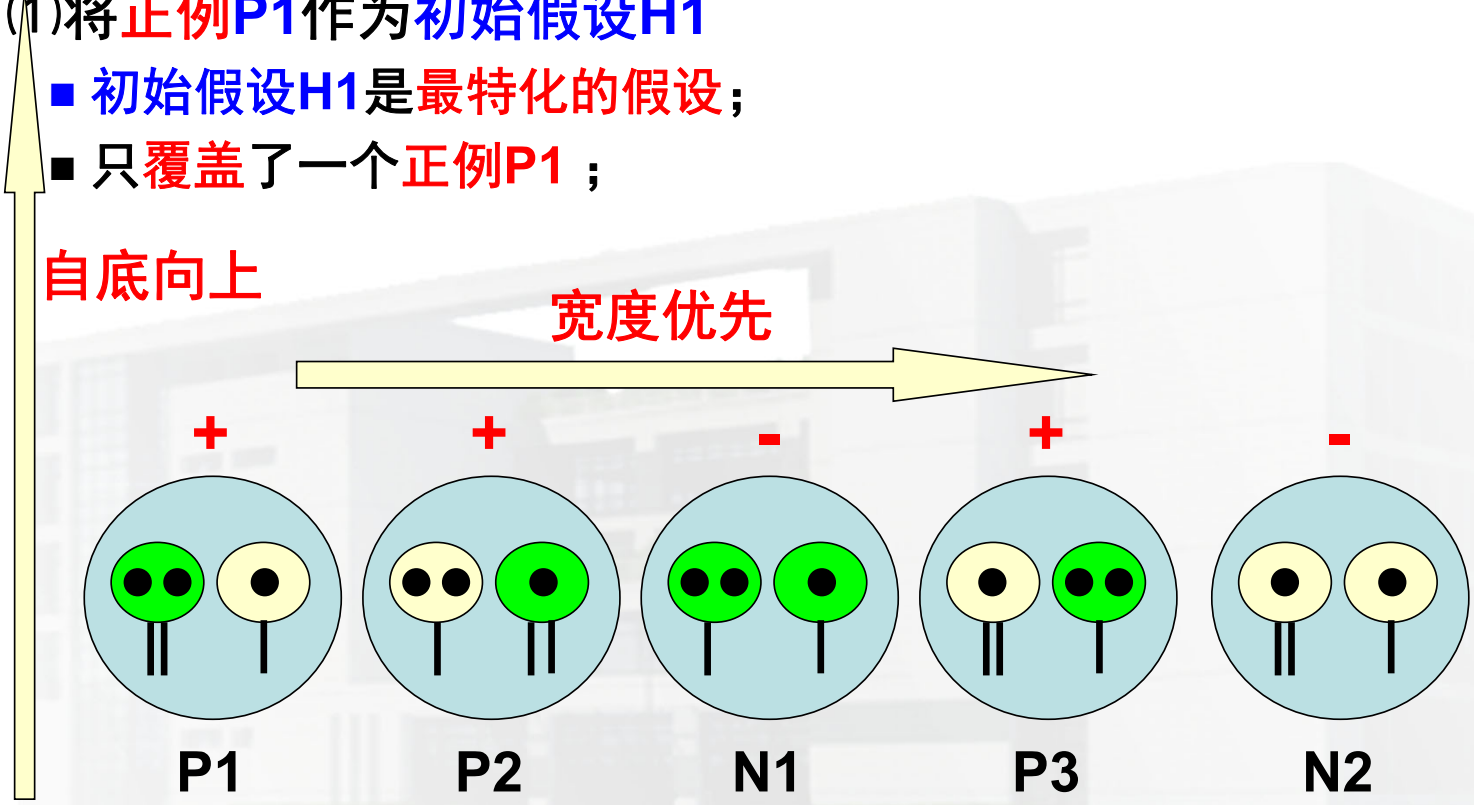
2、逐步泛化的学习策略

- 采用宽度优先、自底向上的泛化搜索方式；
- 基本策略：
  - (1)从第一个正例出发，作为初始假设；
  - (2)遇见正例就泛化某些假设以保证假设的完全描述性（覆盖所有正例）；
  - (3)遇见反例则删去某些假设以保证假设的一致描述性（不覆盖所有反例）；
  - 直至得到一个既完全又一致的解描述(假设)为止；
  - 解描述作为学习系统获得的新知识，满足给定例子集的概念定义。

示例学习

2、逐步泛化的学习策略

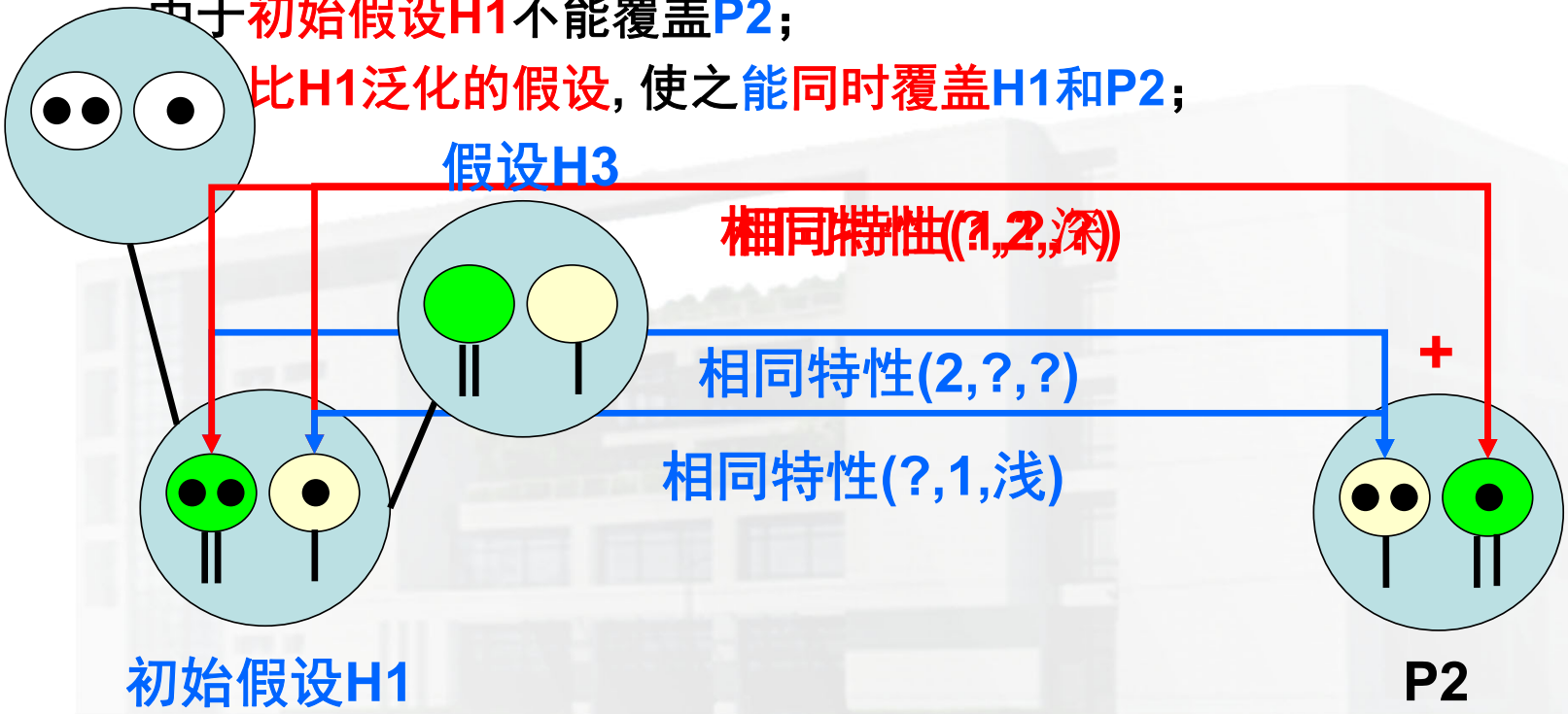
- 采用宽度优先、自底向上的泛化搜索方式：
  - (1)将正例P1作为初始假设H1
    - 初始假设H1是最特化的假设；
    - 只覆盖了一个正例P1 ；



示例学习

2、逐步泛化的学习策略

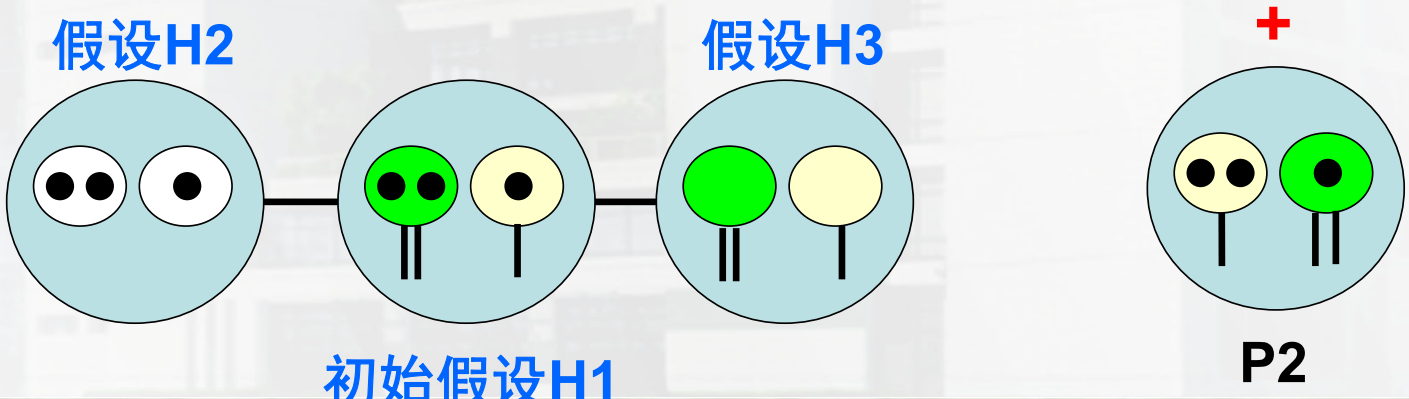
- 采用宽度优先、自底向上的搜索方式：
  - (2)取出下一个正例P2
    - 由于初始假设H1不能覆盖P2；
    - 比H1泛化的假设，使之能同时覆盖H1和P2；



示例学习

2、逐步泛化的学习策略

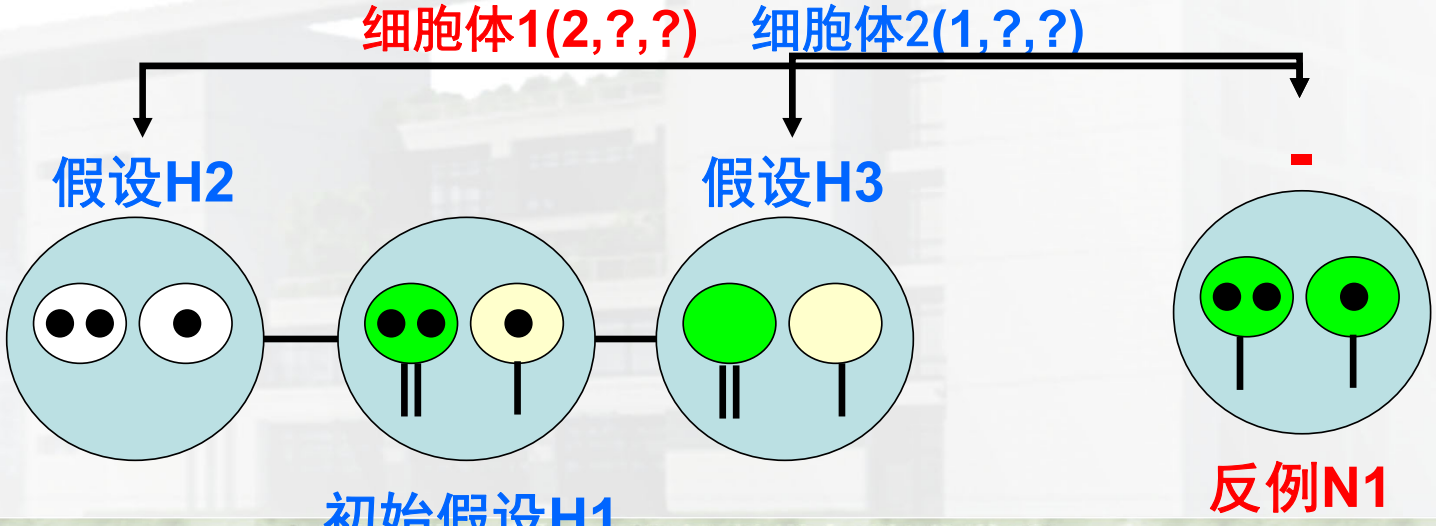
- 采用宽度优先、自底向上的搜索方式：
  - (2)取出下一个正例P2
    - 正例P2指导系统生成泛化的假设H2和H3；
    - 采用“最低限度的泛化”的原则
      - 新的假设刚好覆盖现有的“假设/例子”，
      - 如，H2和H3刚好覆盖H1/P2；



示例学习

2、逐步泛化的学习策略

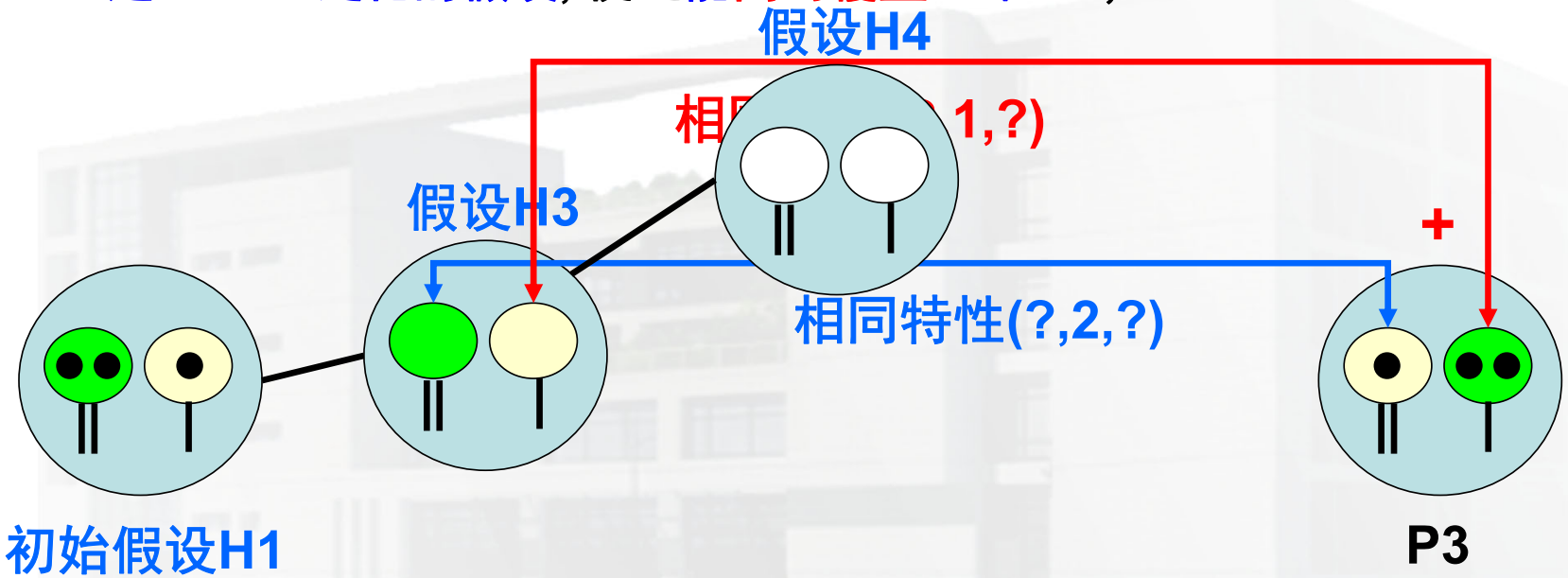
- 采用宽度优先、自底向上的搜索方式：
  - (3)取出下一个反例N1
    - 反例用来删除过于泛化的假设；
    - 假设H2覆盖了反例N1；
    - 假设H2是过于泛化的假设，应该剪去；



示例学习

2、逐步泛化的学习策略

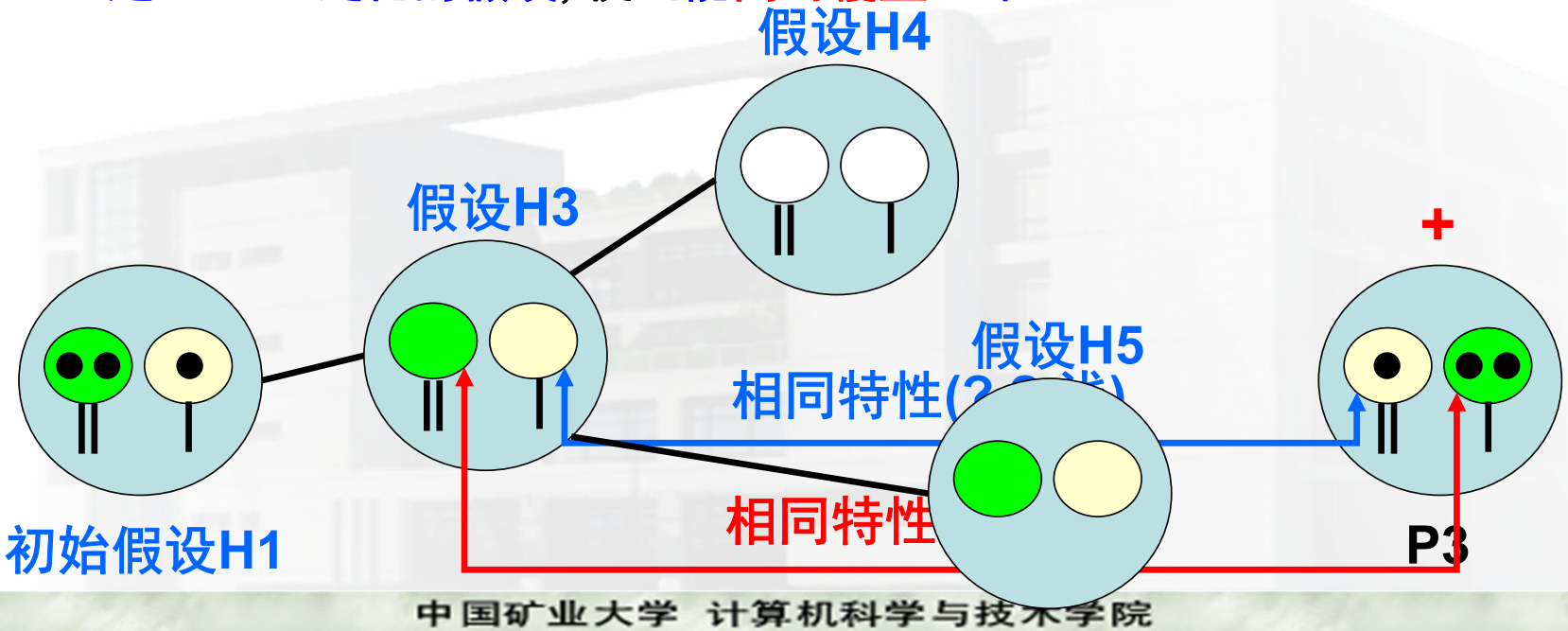
- 采用宽度优先、自底向上的搜索方式：
  - (4)取出下一个正例P3
    - 由于假设H3不能覆盖P3；
    - 建立比H3泛化的假设，使之能同时覆盖H3和P3；



示例学习

2、逐步泛化的学习策略

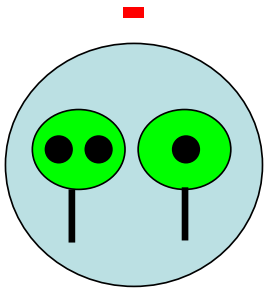
- 采用宽度优先、自底向上的搜索方式：
  - (4)取出下一个正例P3
    - 由于假设H3不能覆盖P3；
    - 建立比H3泛化的假设，使之能同时覆盖H3和P3



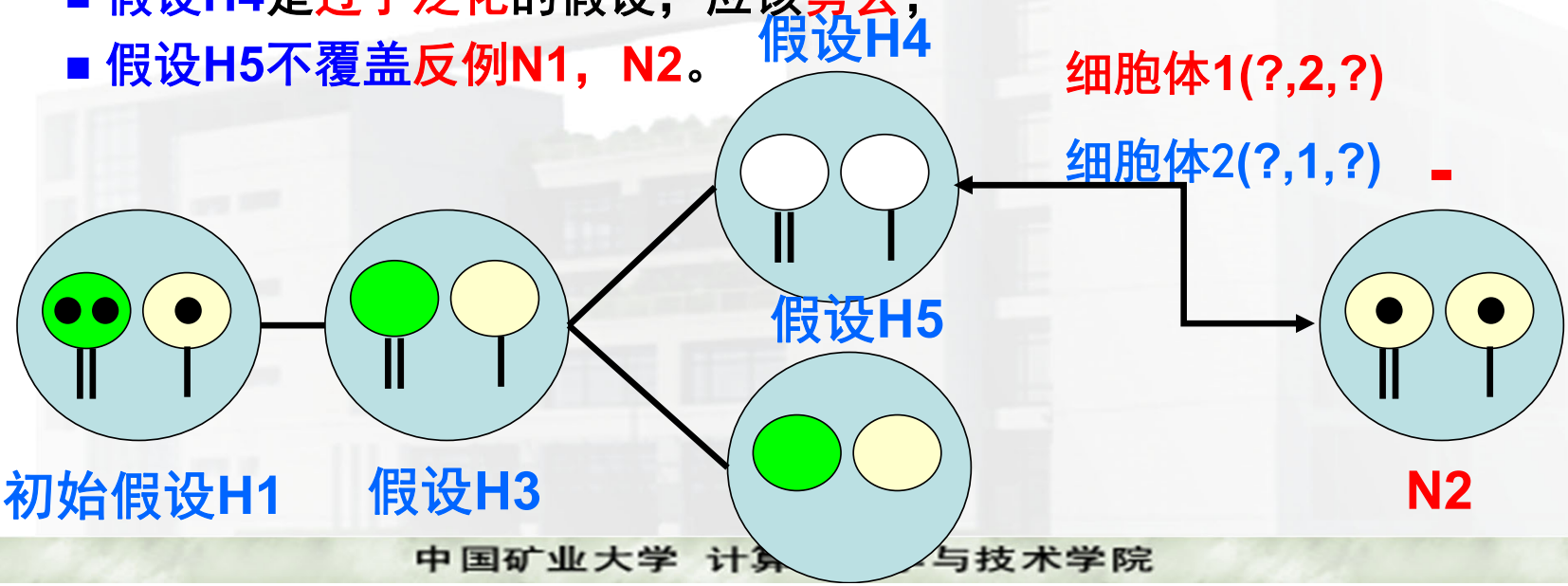
示例学习

2、逐步泛化的学习策略

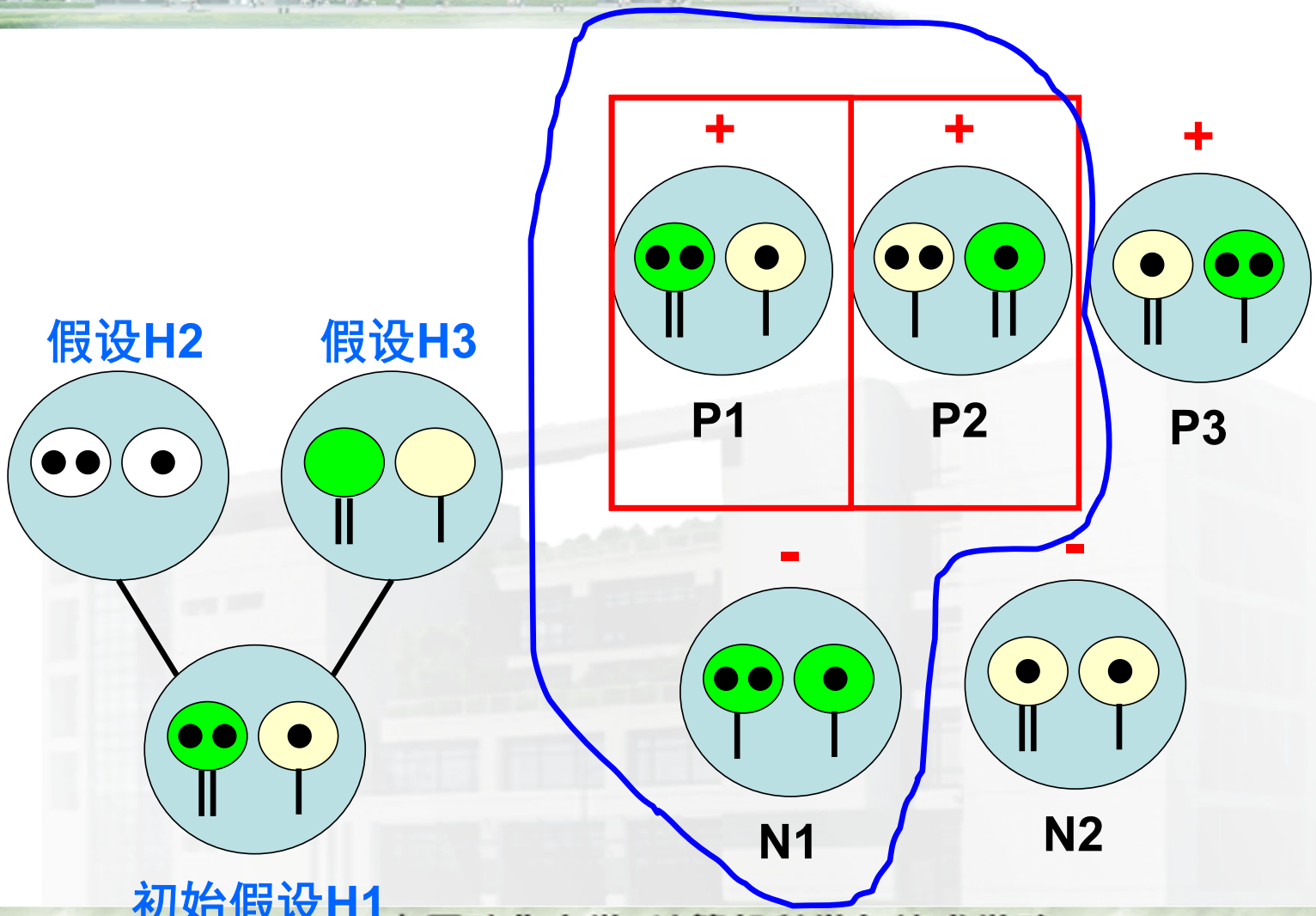
- 采用宽度优先、自底向上的搜索方式：
  - (5)取出下一个反例N2
    - 反例用来删除过于泛化的假设；
    - 假设H4覆盖了反例N2；
    - 假设H4是过于泛化的假设，应该剪去；
    - 假设H5不覆盖反例N1，N2。



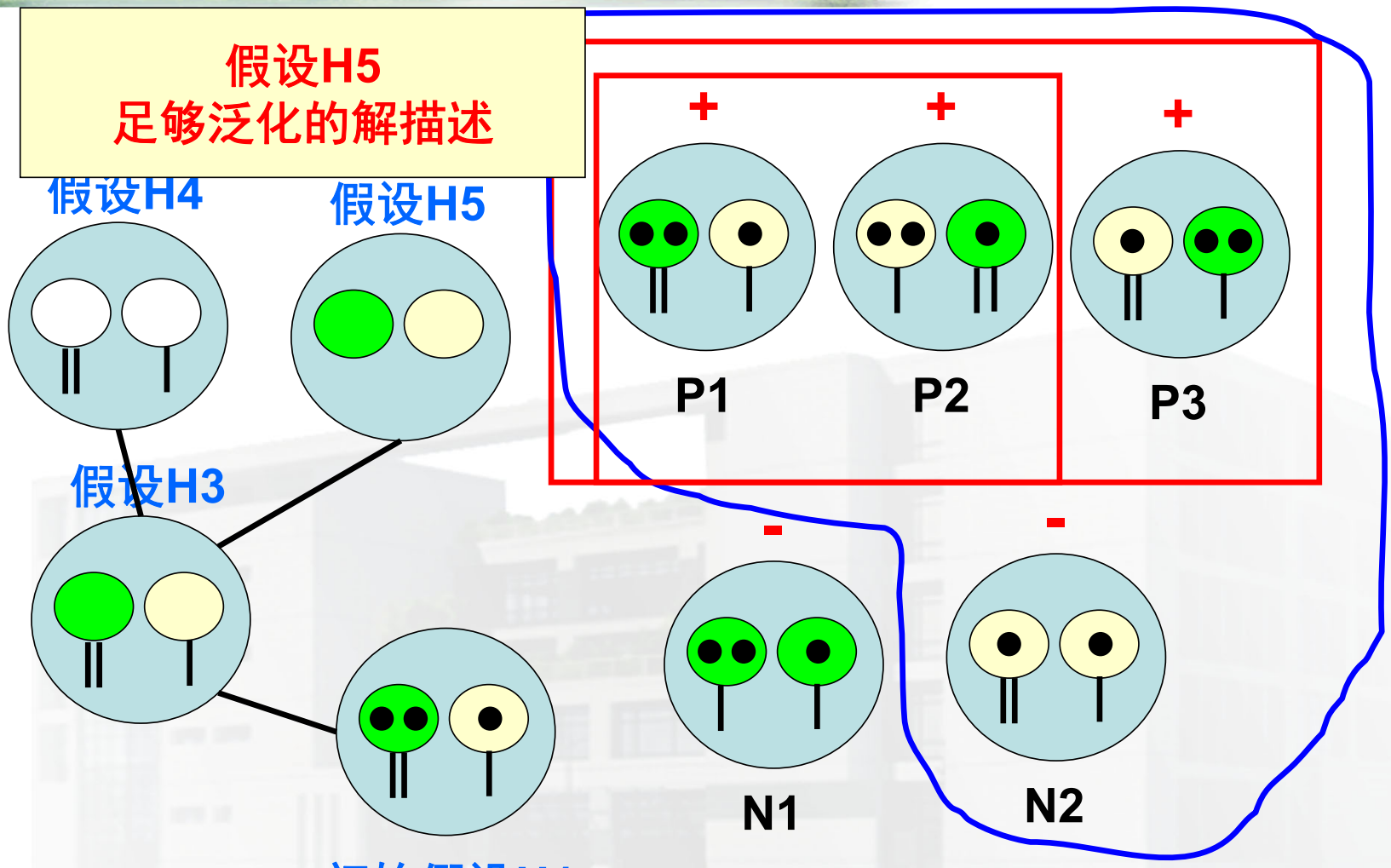
反例N1



示例学习



示例学习



示例学习

2、逐步泛化的学习策略

- 符号说明：
  - H：当前的假设集，初始值为{第一个观察的正例}；
  - N：已观察到的反例集，初始值为空集{}；
  - i：观察的下一个例子；
- 算法描述：
  - IF i是正例 THEN {
    - (1)对每一个不覆盖i的假设h∈H, 用能覆盖i和h（假设/例子），且泛化程度又最低的假设（可以有多个）代替h；
    - (2)移去H中能覆盖已往观察到的反例n∈N的假设(以保证一致性)；

    }
  - ELSE//i是反例{
    - (1)把i加入到反例集N；
    - (2)移去H中能覆盖i的假设；

    }

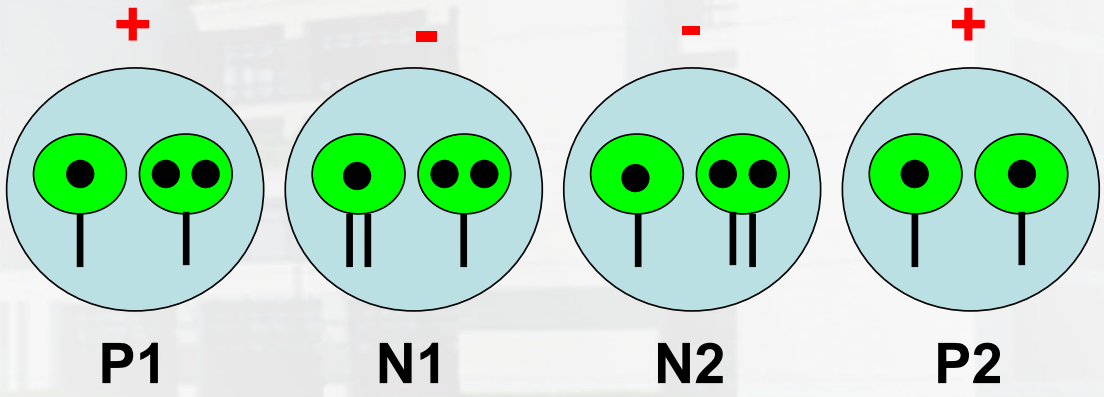
示例学习

- 1、概念描述的搜索和获取
  - 示例学习的过程（T.Mitchell, 1982）：
    - 在假设空间中搜索的过程。
  - 假设空间中的搜索方法
    - ①泛化搜索
      - 从最特化的假设（例子空间中的一个正例）开始；
      - 每次取用一个新的例子，产生一些泛化的描述；
      - 直到产生出足够泛化的解描述。
    - ②特化搜索
      - 从最泛化的假设（概念描述）出发；
      - 每次取用一个新的例子，产生一些特化的描述；
      - 直到产生出足够特化的解描述；

示例学习

3、逐步特化的学习策略

- “泛化策略”：
  - 采用宽度优先、自底向上的搜索方式；
- “特化策略”：
  - 采用宽度优先、自顶向下的搜索方式；
- 【相同点】
  - 新例子的加入会导致新假设的增加和已存在假设的删除；



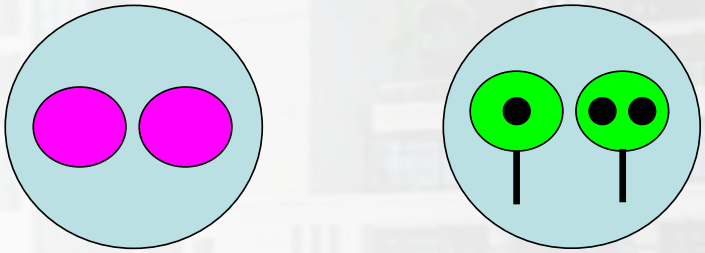
□3、逐步特化的学习策略

- 正例和反例所起的作用与泛化策略相反：
  - 反例——生成一些特化假设；
    - * 采用保守的原则——最低限度的特化：
      - – 新的假设在覆盖已有正例的同时只是刚好能排斥反例；
  - 正例——剪裁过于特化的假设。

示例学习

- 3、逐步特化的学习策略
  - 采用宽度优先、自顶向下的搜索方式；
  - (1)最泛化的假设H1={(?,?,?), (?,?,?)}
    - 细胞简化成2个细胞体,不附有任何的属性；
  - (2)取出第一个正例P1
    - H1正确地覆盖了正例P1，不必修改；
    - 正例P1将放入正例集，备用；

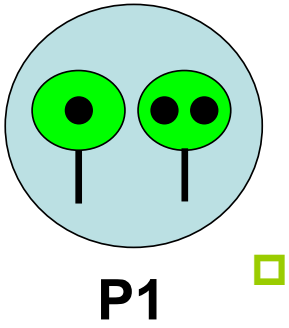
初始假设H1 +

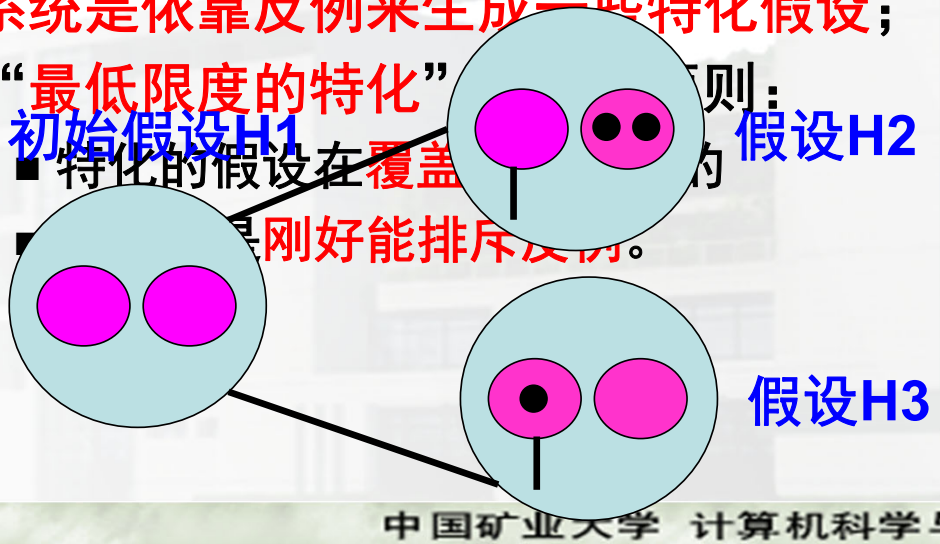


P1

示例学习

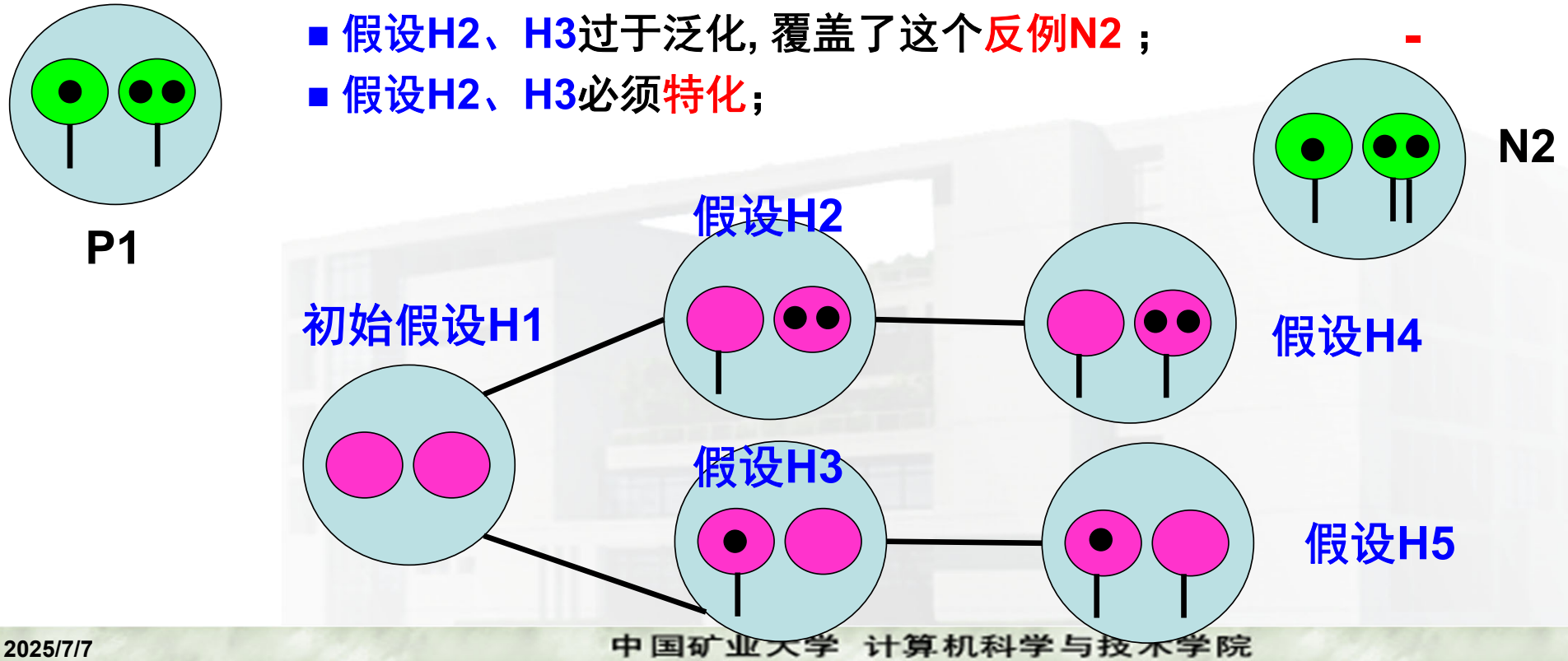
- 3、逐步特化的学习策略
  - 采用宽度优先、自顶向下的搜索方式；
    - (3)取出下一个反例N1
      - 初始假设H1过于泛化，覆盖了这个反例N1 ；
      - 假设H1必须特化，至少得到特化假设H2、H3；
      - 假设H2、H3排斥反例N1；
    - 系统是依靠反例来生成一些特化假设；
    - “最低限度的特化”[illegible]则：
      - 特化的假设在覆盖[illegible]的
      - [illegible]刚好能排斥反例。





示例学习

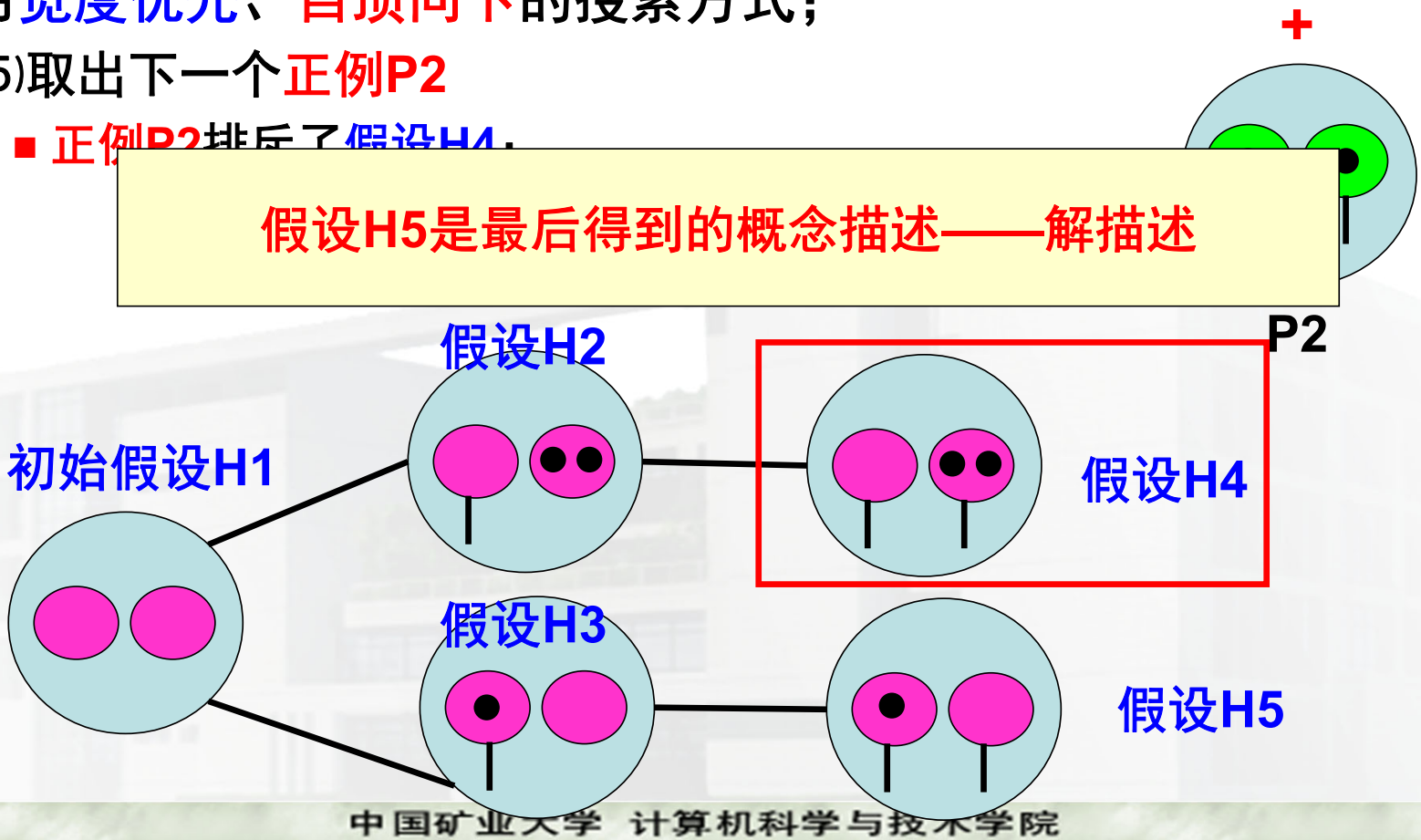
- 3、逐步特化的学习策略
  - 采用宽度优先、自顶向下的搜索方式；
    - (4)取出下一个反例N2
      - 假设H2、H3过于泛化, 覆盖了这个反例N2 ；
      - 假设H2、H3必须特化；



示例学习

- 3、逐步特化的学习策略
  - 采用宽度优先、自顶向下的搜索方式；
    - (5)取出下一个正例P2
      - 正例P2排斥了假设H4；

假设H5是最后得到的概念描述——解描述



示例学习

- 3、逐步特化的学习策略
  - 符号说明：
    - H：当前的假设集，初始值为{最泛化的假设}；
    - P：已观察到的正例集，初始值为空集{}；
    - i：观察的下一个例子；
  - 算法描述：
    - IF i是反例THEN {
      - (1)对每一个覆盖i的假设h∈H, 用可被h覆盖但排斥i，且特化程度最低的假设代替h；
      - (2)移去H中不覆盖已往观察到的正例p∈P的假设；

      }
    - ELSE//i是正例 {
      - (1)把i加入到正例集P；
      - (2)移去H中所有不覆盖i的假设；

      }

示例学习

- 泛化策略：
  - 采用自底向上的搜索假设空间的方式；
  - 从第一个正例表示的最特化的假设开始；
  - 系统依靠正例生成泛化的假设；
  - 反例用来剪裁过于泛化的假设；
  - 解描述——泛化程度最低；
- 特化策略：
  - 采用自顶向下的搜索假设空间的方式；

如果给出充分多的例子，
那么二者的结果就可能会是相同的概念描述。

  - 解描述——特化程度最低；

示例学习

4、双向学习策略

- 结合“泛化策略”和“特化策略”，同时从2个方向搜索假设空间。
- 版本空间法(Version Space）
  - 假设集S——泛化搜索的假设空间；
    - 遇见一个新的正例时，如未被S集包含，则在该集中进行泛化搜索；
  - 假设集G——特化搜索的假设空间；
    - 一个新的反例产生时，如被G集包含，则在该集中进行特化搜索；

示例学习

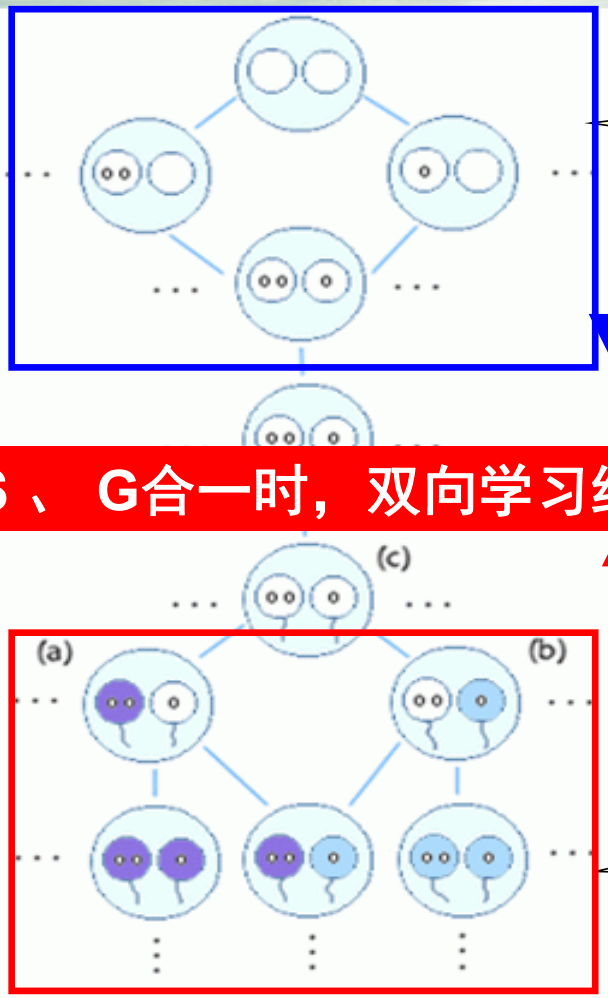


图6.5 假设空间的半序图

特化搜索

假设集G
G能覆盖新的反例i，
则在G中进行特化搜索

当S 、 G合一时，双向学习结束

完全的假设空间

范化搜索

假设集S
S不能覆盖新的正例i
则在S中进行泛化搜索

示例学习

4、双向学习策略

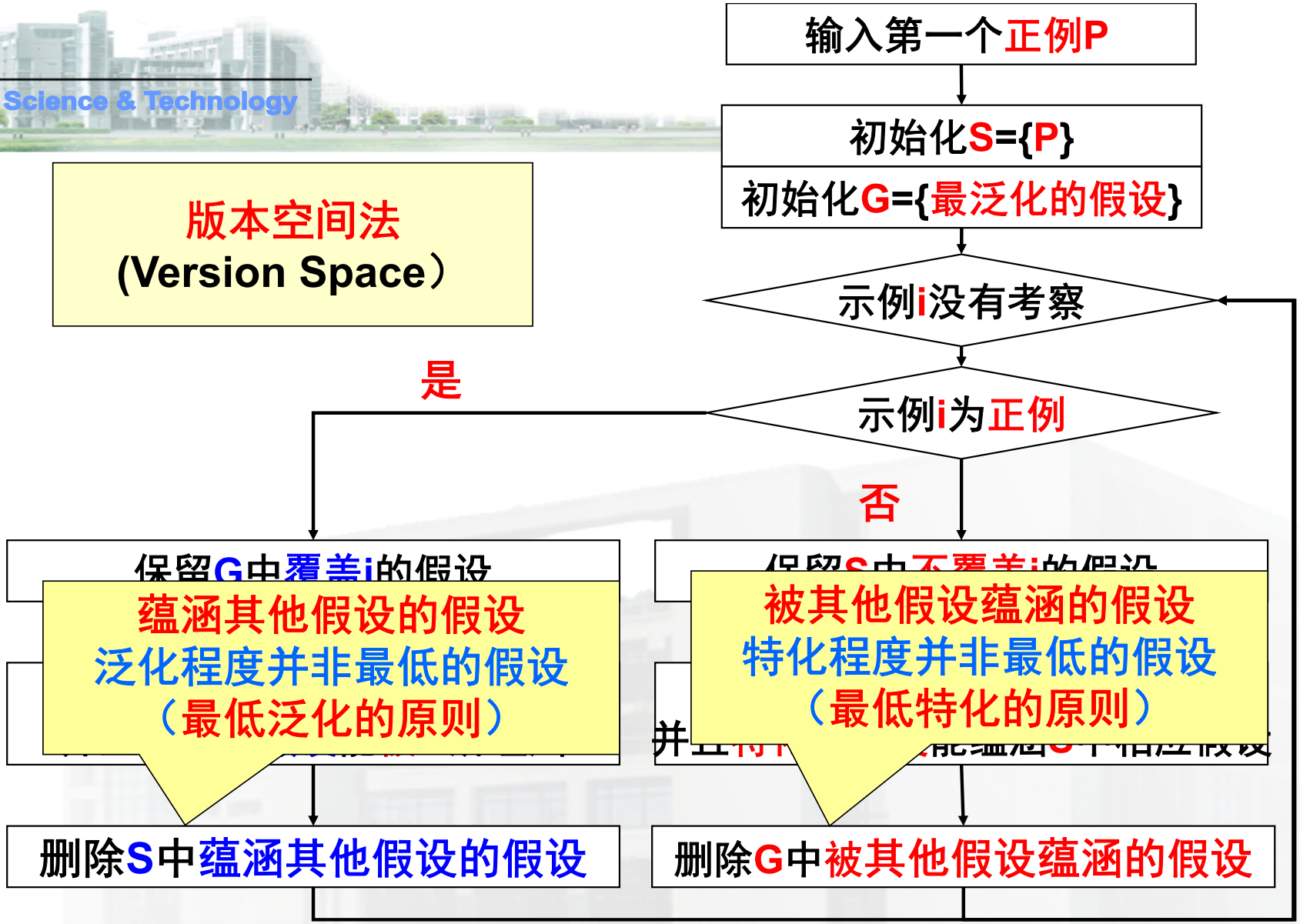
- 结合“泛化策略”和“特化策略”，同时从2个方向搜索假设空间。
- 版本空间法(Version Space）
  - 假设集S——泛化搜索的假设空间；
    - 期望获取的最终解描述下界；
  - 假设集G——特化搜索的假设空间；
    - 期望获取的最终解描述上界；

示例学习

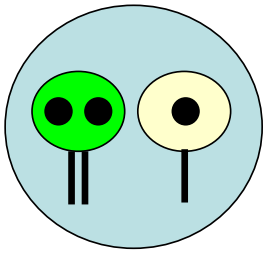
4、双向学习策略

- 版本空间法(Version Space）优点：
  - (1)系统不必保留正例（特化策略）和反例（泛化策略）：
    - S蕴涵了已取用的所有正例，删除G中过于特化的假设；
    - G蕴涵了对所有已取用反例的排斥，删除S中过于泛化的假设。
  - (2)系统知道何时推理任务完成；
    - 当S 、 G合一时，双向学习结束；
    - “泛化”和“特化”策略只能搜索完所有示例；

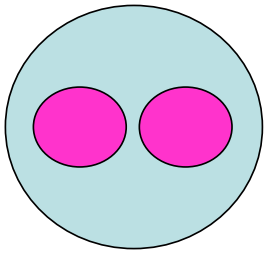
版本空间法
(Version Space）



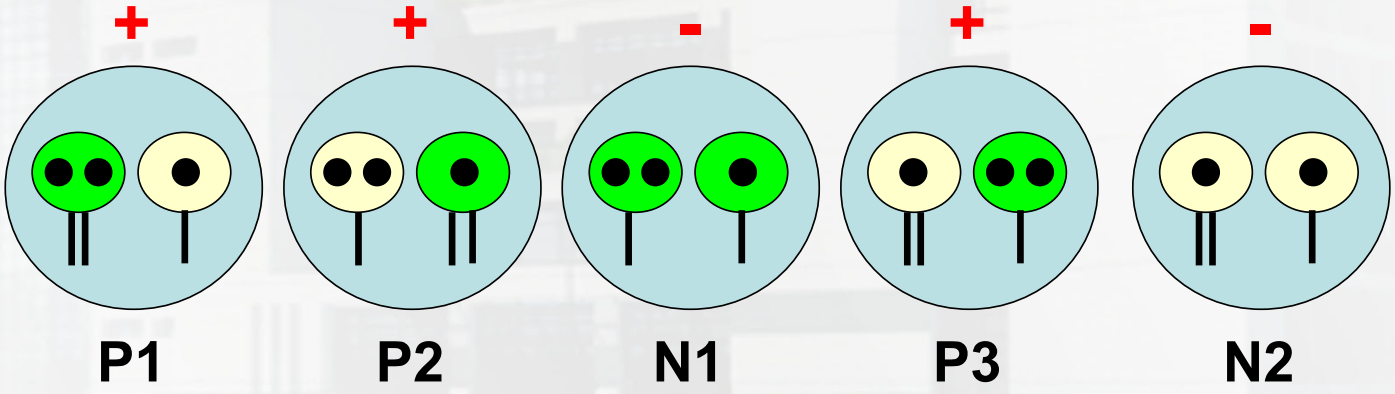
输入第一个正例P1



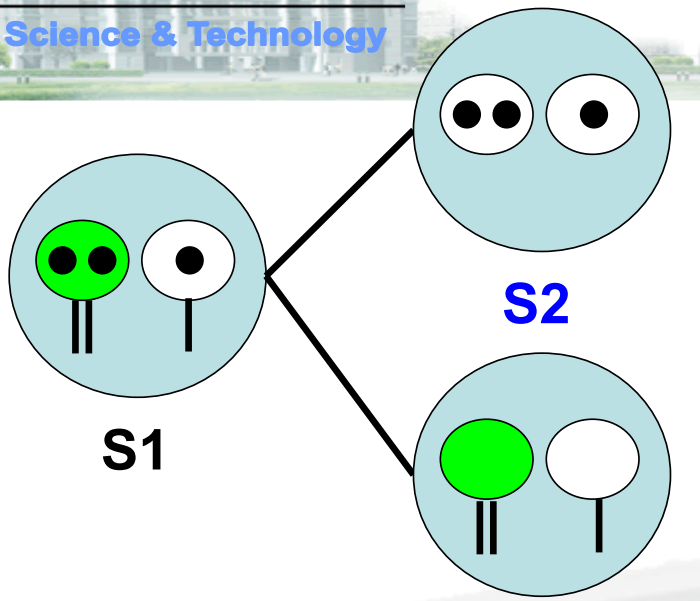
S1

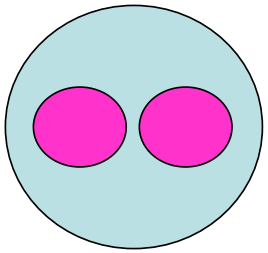


G1



正例P2

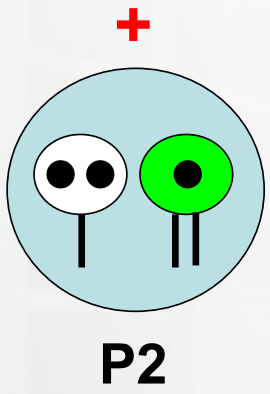




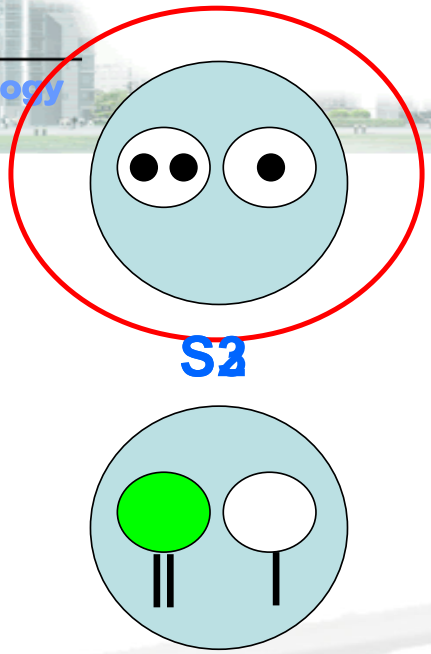
保留G中覆盖i的假设

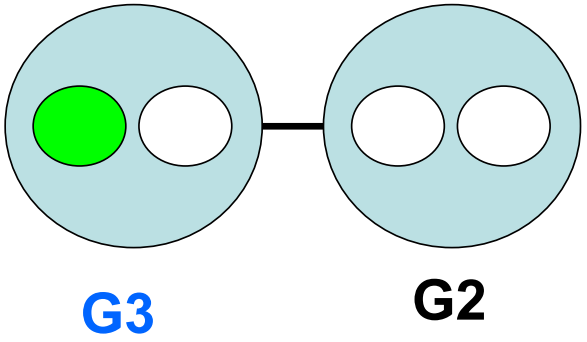
S中不覆盖i的假设泛化，
并且泛化的假设能被G所蕴涵

删除S中蕴涵其他假设的假设



反例N1

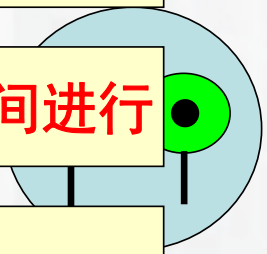




S3和G3中的假设构成了满足已知正、反例的概念描述

进一步的“泛化”、“特化”搜索只能在S3和G3之间进行

并且特化的假设能蕴涵S中相相应假设



示例足够多时，S3和G3就会合而为一

删除G中可以被其他假设蕴含的假设

6.3 距离及相似度度量方法

- 在机器学习算法中，经常需要评估两个不同样本之间的“相似性”，进而确定样本属于哪一类。
  - 如，*k*-近邻算法（KNN）、*k*-均值算法（K-Means）等等
- 常见的距离计算方式包括欧氏距离、曼哈顿距离、夹角余弦和Jaccard系数等。
- 根据数据特性的不同，可以采用不同的度量方法。
- 定义一个距离函数，需要满足下面几个准则：
  - 距离永远非负，只有点到自身的距离为0；
  - 距离具有对称性，在计算点之间的距离时考虑点的顺序无所谓先后；
  - 距离遵守三角不等式，即*x*到*y*的距离加上*y*到*z*的距离不低于直接*x*到*z*的距离。

6.3.1 距离度量

- 欧氏距离
  - 在*m*维空间中两个点之间的真实距离
  - 即向量的自然长度（即该点到原点的距离）
  - 在二维和三维空间中的欧氏距离就是两点之间的实际距离。
    - 对于二维平面上两点 与 间的欧氏距离计算公式为

$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

    - 对于三维平面上两点 与 间的欧氏距离计算公式为

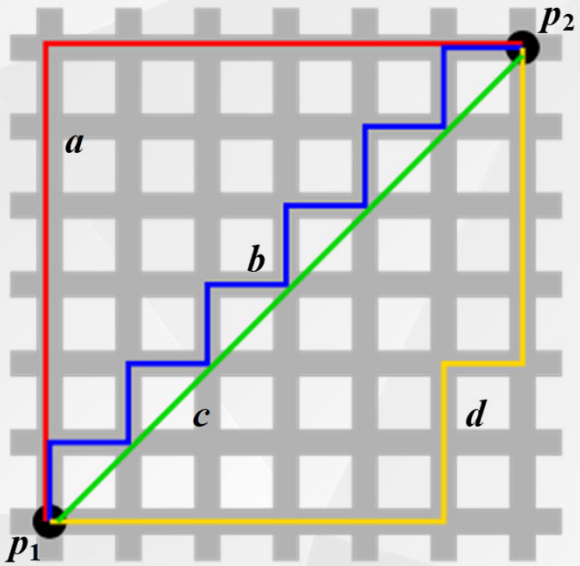
$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$$

6.3.1 距离度量

曼哈顿距离

- 标明两个点在标准坐标系上的绝对轴距总和
- 表示的不是两点的直线距离，而是实际从p_1点到达p_2点的距离
  - 对于二维平面上的两点 与 之间的曼哈顿距离，计算公式为

$$d = |x_1 - x_2| + |y_1 - y_2|$$



曼哈顿距离示例

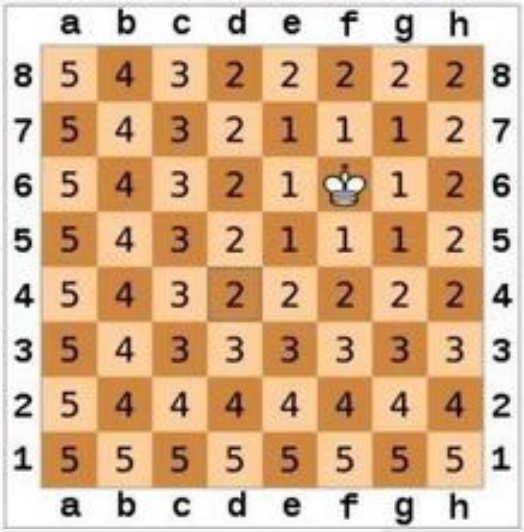
6.3.1 距离度量

- 切比雪夫距离
  - 其各坐标数值差的最大值
  - 在国际象棋中指王从一个位置移至另一个位置需要走的步数
  - 也称为棋盘距离
    - 二维平面上的两点 与 之间的切比雪夫距离计算公式为

$$d = \max(|x_1 - x_2|, |y_1 - y_2|)$$

    - n维空间中两点 与 之间的切比雪夫距离计算公式为

$$d = \max_i(|x_{1i} - x_{2i}|)$$



切比雪夫距离示例

6.3.1 距离度量

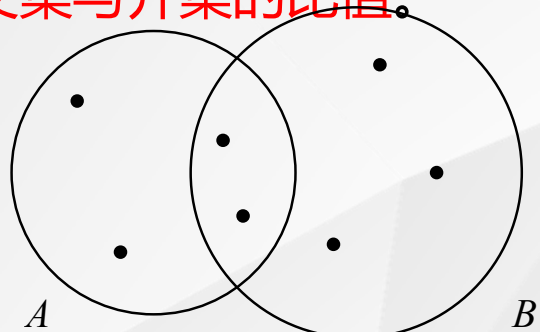
- 海明距离
  - 两个等长字符串之间的海明距离是两个字符串对应位置的不同字符的个数
  - 即将一个字符串变换成另外一个字符串所需要替换的字符个数
    - “1001101”与“1001011”之间的海明距离是2
    - “2143896”与“2233796”之间的海明距离是3
    - “toned”与“roses”之间的海明距离是3

6.3.2 相似度度量

- 相似度度量（Similarity），即计算样本之间的相似程度，与距离度量相反，相似度度量的值越小，说明样本间的相似度越小，差异越大。
- Jaccard相似系数
  - 也称雅可比相似度系数，它是通过计算样本的交集与并集之间差异的相似度算法
  - 一般用*J*表示，计算公式如下

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

  - 在集合*A*和集合*B*中，两者的交集越多，则表示两者相似度越高。
  - *Jaccard*的相似性与集合的顺序无关，仅与在集合中是否出现有关，即为二值数据，不是0 则是1，是一种简单的相似性方法，实质是集合交集与并集的比值。



6.3.2 相似度度量

- 夹角余弦相似度
  - 又称为余弦相似性，是通过计算两个向量的夹角余弦值来评估它们的相似性
  - 将向量根据坐标值，绘制到向量空间中。
  - 二维空间中向量 a(x_1,y_1)与向量b(x_2,y_2) 的夹角余弦公式为：

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

  - 对于任意两个*n*维样本向量a(x_1, x_2,..., x_n) 和b(y_1, y_2,..., y_n) ，它们之间的夹角余弦公为：

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}.$$

  - 夹角余弦取值范围为[-1,1]。其值越大表示两个向量的夹角越小，越小表示两向量的夹角越大。
  - 当两个向量的方向重合时夹角余弦取最大值1，当两个向量的方向完全相反夹角余弦取最小值-1。

6.3.2 相似度度量

例:给定下列两个文本，分别使用Jaccard相似系数和夹角余弦相似度计算这两个文本的相似程度。

文本t_1：机器学习促进了人工智能的发展。

文本t_2：人工智能的发展改变了人们的生活。

解：（1）使用Jaccard相似系数计算文本t_1和t_2之间的相似度

第一步：分词处理。将文本 t_1和t_2 分别进行分词处理后，结果是“机器学习 促进 人工智能 发展”和“人工智能 发展 改变 人们 生活”。

第二步：将文本 t_1的分词结果当做一个集合，则t_1 ={机器学习，促进，人工智能，发展}；同理， t_2 ={人工智能，发展，改变，人们，生活}。

第三步：计算集合 t_1与集合 t_2的交集和并集。集合 t_1与集合 t_2的交集为{人工智能，发展}，并集为{机器学习，促进，人工智能，发展，改变，人们，生活}。

第四步：相似度计算。

$$J(t_1,t_2)=\frac{|t_1 \cap t_2|}{|t_1 \cup t_2|}=\frac{2}{7}=0.286$$

通过上述过程，最终计算出“文本 t_1：机器学习促进了人工智能的发展”与“文本 t_2：人工智能的发展改变了人们的生活”的Jaccard相似度为0.286。

6.3.2 相似度度量

解：（2）使用夹角余弦相似度计算文本 t_1和 t_2之间的相似度

第一步：分词处理。将文本t_1和t_2分别进行分词处理后，结果是“机器学习 促进 人工智能 发展”和“人工智能 发展 改变 人们 生活”。

第二步：构建特征集。根据第一步的分词结果，将所有词语进行合并，出现同一词语仅记一次，获得的特征集为{机器学习，促进，人工智能，发展，改变，人们，生活}。

第三步：形成特征向量。特征向量以特征集为基础，词语在文本中出现，则在相应位置记1，未出现记0。则针对文本 t_1的特征向量是t_1= （1,1,1,1,0,0,0），针对文本 t_2的特征向量是 t_2= （0,0,1,1,1,1,1），。

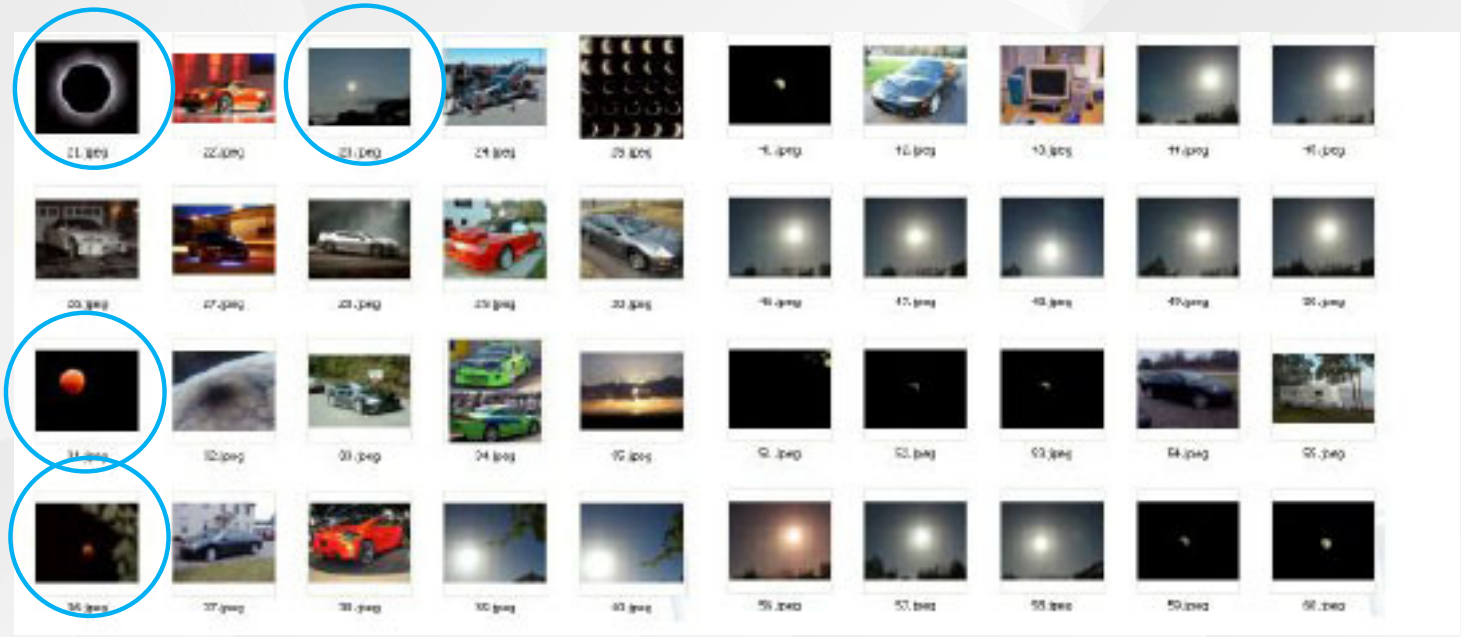
第四步：相似度计算。通过第三步已经获得了文本 t_1和t_2 的特征向量，将问题转化为对向量求夹角余弦值。根据夹角余弦公式，计算可得：

$$\cos(t_1,t_2)=\frac{t_1 \square t_1}{\sqrt{t_1^2}\times\sqrt{t_2^2}}=\frac{1\times0+1\times0+1\times1+1\times1+0\times1+0\times1+0\times1}{\sqrt{1^2+1^2+1^2+1^2+0^2+0^2+0^2}\times\sqrt{0^2+0^2+1^2+1^2+1^2+1^2+1^2}}$$

通过上述过程，最终计算出“文本 t_1：机器学习促进了人工智能的发展”与“文本 t_2：人工智能的发展改变了人们的生活”的余弦相似度为0.447。

6.4 分类算法分析

- 分类的任务就是确定对象属于哪个预定义的目标类。
  - 根据电子邮件的标题和内容检查出垃圾邮件
  - 对一大堆照片区分出哪些是风景照、哪些是人物照



图片分类案例：从图库中识别出“日蚀”图片

6.4.1 分类概述

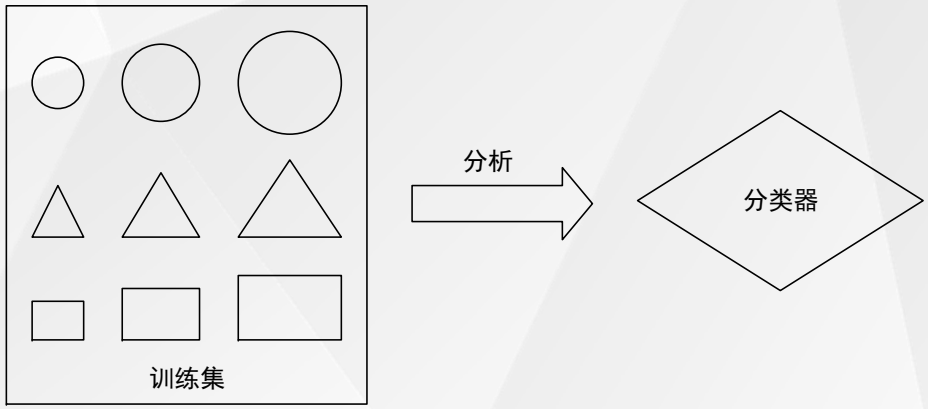
- 分类（Classification）是机器学习的一项重要任务，是指在数据库的各个对象中找出共同特征，并按照分类模型把它们进行分类。
  - 分类分析时将依据“训练集”数据的类标号，对类进行准确的描述或者建立模型，该模型能够很好地拟合输入数据中类标号和属性集之间的联系，还要能正确地预测未知样本的类标号。
  - 属于典型的监督学习。

6.4.1 分类概述

- 解决分类问题的一般过程

（1）建立分类模型

分类模型的建立是通过分析训练样本数据总结得出的一般性的分类规则，模型以分类规则、决策树或数学公式的形式给出。



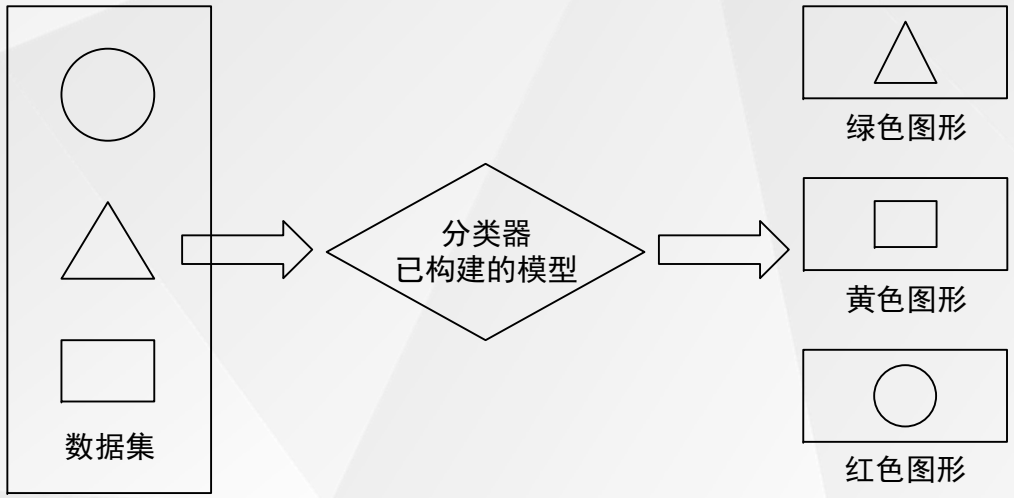
分类模型的建立

6.4.1 分类概述

- 解决分类问题的一般过程

（2）分类模型的应用

在对建立的分类模型进行应用之前，需要对模型进行评估，在确保分类模型的准确性和精度的情况下，才能运用该分类模型对未知其类型的样本进行分类处理。



分类模型的应用

6.4.1 分类概述

分类模型的评估

- 分类模型的性能根据模型正确的和错误的测试记录计数进行评估，这些计数存放在混淆矩阵里。
- 对于类别c来说，模型在测试集上的结果可以分为以下四种情况：
  - 真正例（True Positive, TP）：一个样本的真实类别为c并且模型正确地预测为类别c；
  - 假负例（False Negative, FN）：一个样本的真实类别为c，模型错误地预测为其它类；
  - 假正例（False Positive, FP）：一个样本的真实类别为其它类，模型错误地预测为类c；
  - 真负例（True Negative, TN）：一个样本的真实类别为其它类，模型也预测为其它类。

真实类别	预测类别	
	正例	负例
正例	TP	FN
负例	FP	TN

类别预测结果的混淆矩阵

6.4.1 分类概述

分类模型的评估

- 不同的机器学习任务有着不同的评价指标，同时同一种机器学习任务也有着不同的评价指标，每个指标的着重点不一样。并且很多指标可以对多种不同的机器学习模型进行评价。
  - 查准率（Precision），也叫精确率或精度，类别c的查准率为在所有预测为类别c的样本中，预测正确的比例。

$$P_c = \frac{\text{预测正确的样本数}}{\text{预测总数}} = \frac{TP_c}{TP_c + FP_c}$$

  - 查全率（Recall），也叫召回率，类别c的查全率为在所有真实标签为类别c的样本中，预测正确的比例。

$$R_c = \frac{\text{预测正确的样本数}}{\text{正确的样本总数}} = \frac{TP_c}{TP_c + FN_c}$$

6.4.1 分类概述

分类模型的评估

- 不同的机器学习任务有着不同的评价指标，同时同一种机器学习任务也有着不同的评价指标，每个指标的着重点不一样。并且很多指标可以对多种不同的机器学习模型进行评价。
  - F-Measure（又称为F-Score），是一个综合评价指标，是查准率和查全率的加权调和平均。

$$F_c = \frac{(1+\beta^2) \times P_c \times R_c}{\beta^2 \times P_c + R_c}$$

  - 其中，β 用于平衡查全率和查准率的重要性，一般取值为1，此时F值又称为F1值。

6.4.1 分类概述

例:某池塘有1400条鲤鱼，300只虾，300只鳖。现在以捕捞鲤鱼为目的，渔民往池塘撒一大网，收获700条鲤鱼的同时，也捞出了200只虾和100只鳖。请问此次捕捞的精确率、召回率以及F1值分别是多少？如果把池塘里的所有的鲤鱼、虾和鳖都一网收获，这些指标又有什么变化?

解：收获700条鲤鱼的同时，也捞出了200只虾和100只鳖的精确率、召回率以及F1值计算如下：

$$Precision = \frac{\text{捕捞正确的样本数}}{\text{捕捞总数}} = \frac{700}{700+200+100} = 70\%$$

$$Recall = \frac{\text{捕捞正确的样本数}}{\text{需要捕捞的样本数}} = \frac{700}{1400} = 50\%$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0.7 \times 0.5}{0.7 + 0.5} = 58.3\%$$

6.4.1 分类概述

例:某池塘有1400条鲤鱼，300只虾，300只鳖。现在以捕捞鲤鱼为目的，渔民往池塘撒一大网，收获700条鲤鱼的同时，也捞出了200只虾和100只鳖。请问此次捕捞的精确率、召回率以及F1值分别是多少？如果把池塘里的所有的鲤鱼、虾和鳖都一网收获，这些指标又有什么变化?

解：把池塘里的所有的鲤鱼、虾和鳖都一网收获，此时精确率、召回率以及F1值计算如下：

$$Precision = \frac{1400}{1400+300+300} = 70\%$$

$$Recall = \frac{1400}{1400} = 100\% \qquad F1 = \frac{2 \times 0.7 \times 1}{0.7+1} = 82.35\%$$

总结：精确率是评估捕获成果中目标成果所占的比例；召回率是从所关注的领域中召回目标类别的比例；而F值是综合这两个指标的评价指标，用于综合反映事件整体情况。

6.4.1 分类概述

- 分类算法的应用场景：解决各种模式识别问题
  - 图库软件的图片分类
  - 新闻网站的话题分类
  - 银行贷款客户的风险分类
  - 医院对患者病因的分类
  - ……

6.4.2 分类分析方法

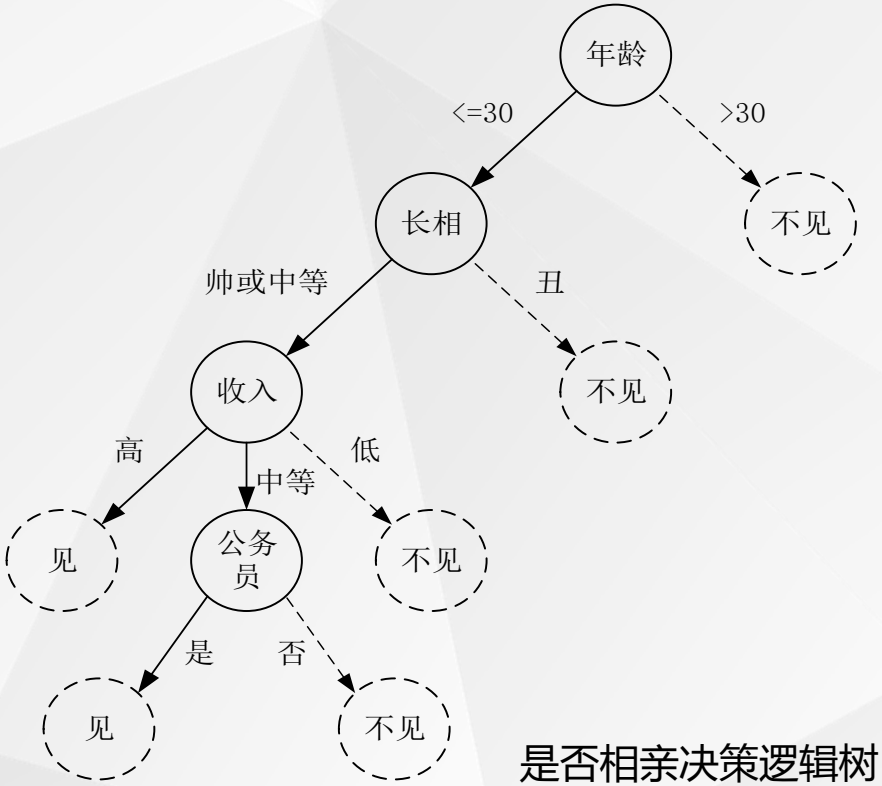
分类的目的是根据数据集的特点构造一个分类器，把未知类别的样本映射到给定类别中的某一个。

- 单一的分类方法主要包括：
  - 决策树、K-近邻、贝叶斯、支持向量机、基于关联规则的分类等
- 集成学习算法：组合单一分类方法
  - 如Bagging和Boosting等
- 分类算法需要对训练数据集进行标识、即事先确定好类别，属于监督学习

6.4.3 典型的分类算法介绍—决策树算法

- 决策树（Decision Tree）是一种基本的分类与回归方法，模型呈树形结构，在分类问题中，表示基于特征对实例进行分类的过程。
  - 构造决策树的目的是找出属性和类别间的关系，用来预测将来未知类别的记录的类别。

母亲：给你介绍个对象。
女儿：年纪多大了？
母亲：26。
女儿：长的帅不帅？
母亲：挺帅的。
女儿：收入高不？
母亲：不算很高，中等情况。
女儿：是公务员不？
母亲：是，在税务局上班呢。
女儿：那好，我去见见。



是否相亲决策逻辑树

6.4.3 典型的分类算法介绍—决策树算法

- 决策树学习——归纳学习方法的一个变种；
  - 任务：从大的已经分类的例子集，归纳分类概念；
    - 例子表示为一组“属性-值”；
      - 每一个例子用相同的一组属性来表示；
      - 每一个属性又有自身的属性值集；
  - ID3算法，昆兰（J.R.Quinlan，1975）； ★
    - 输入：
      - (1)描述已知类别例子的列表；
      - (2)例子由预先定义的“属性-值”对来表示；
    - 结果：
      - 决策树——可以正确地区分所有给定例子的类别；

数学基础

使用信息论指导决策树构造，提高决策树的工作效率

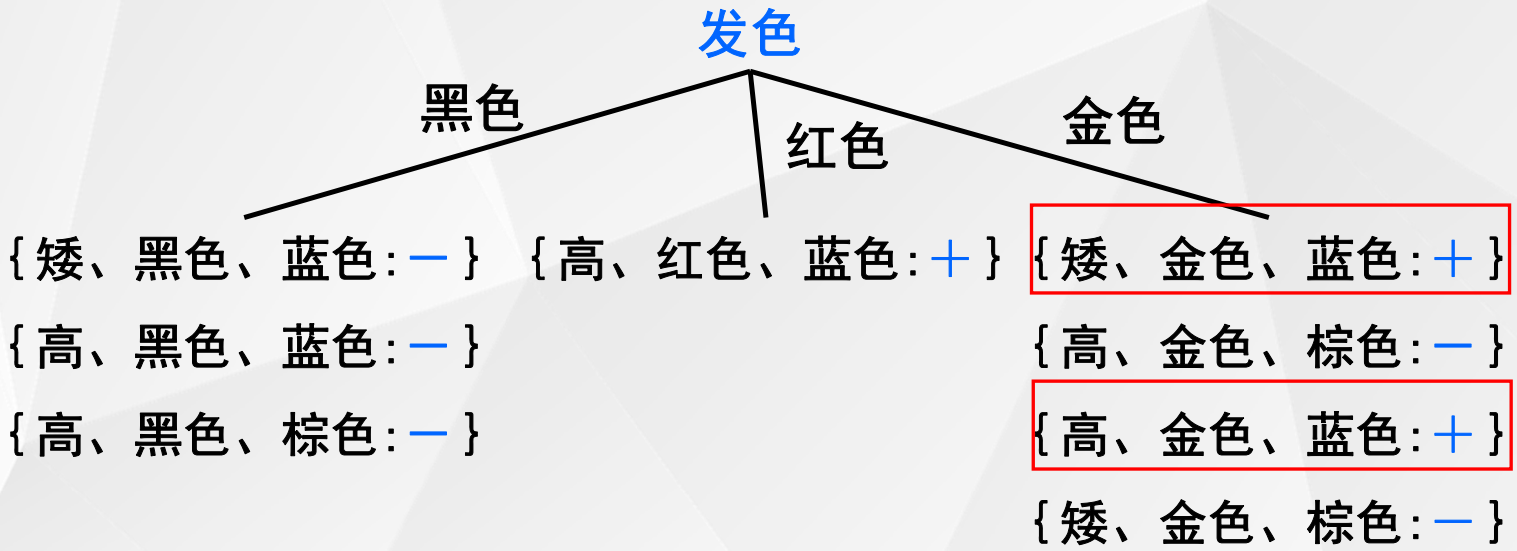
6.4.3 典型的分类算法介绍—决策树算法

- 例：预先定义一组属性及其可取值：
  - 高度{高，矮}；
  - 发色{黑色, 红色，金色}；
  - 眼睛{蓝色，棕色}；
- 人分为两类：
  - “＋”
  - “－”

高度	发色	眼睛	类别
矮	黑色	蓝色	-
高	黑色	蓝色	-
矮	金色	蓝色	+
高	金色	棕色	-
高	黑色	棕色	-
矮	金色	棕色	-
高	金色	蓝色	+
高	红色	蓝色	+

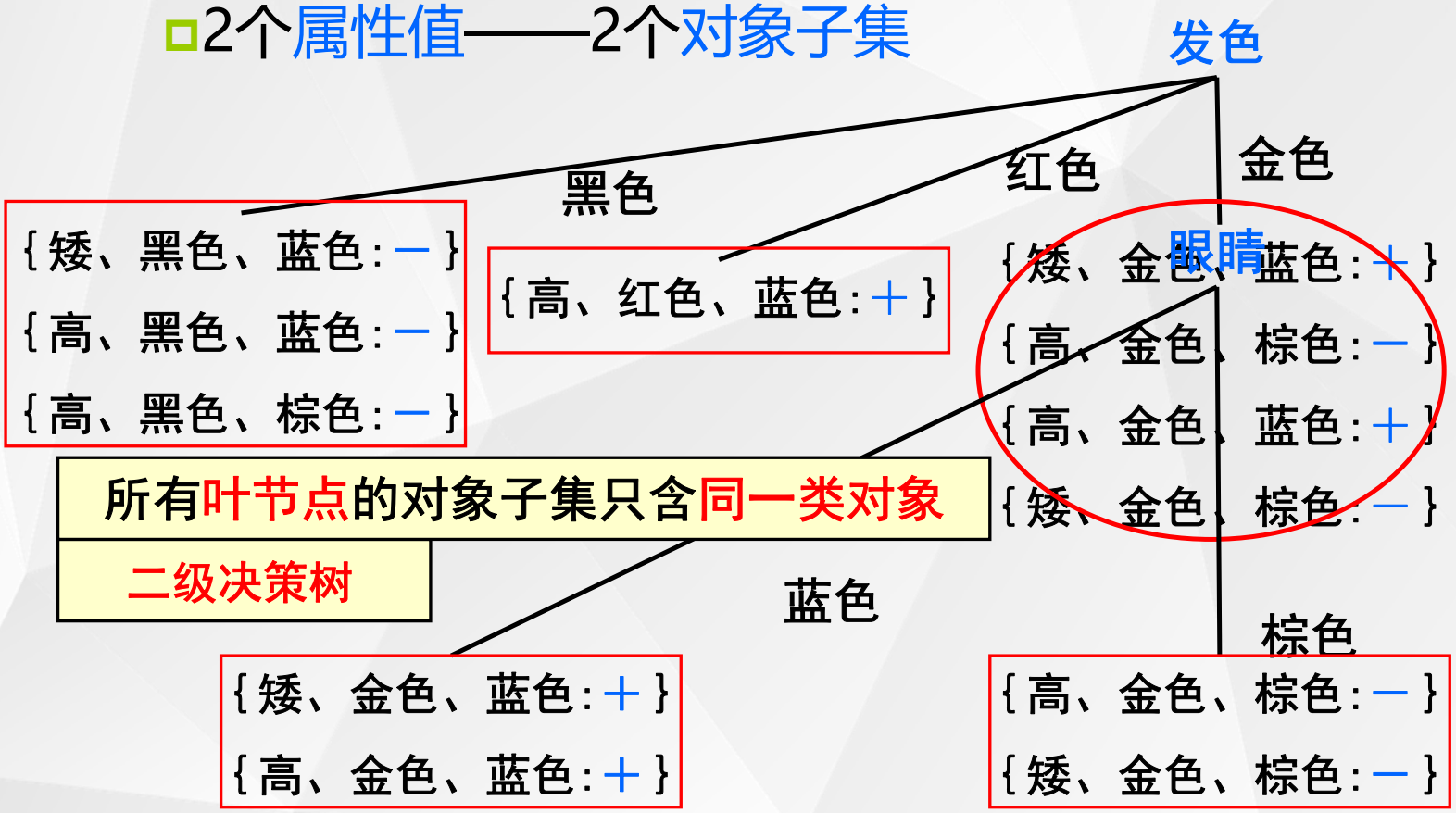
6.4.3 典型的分类算法介绍—决策树算法

- 选取“发色”为树的根节点：
  - 3个属性值——3个对象子集

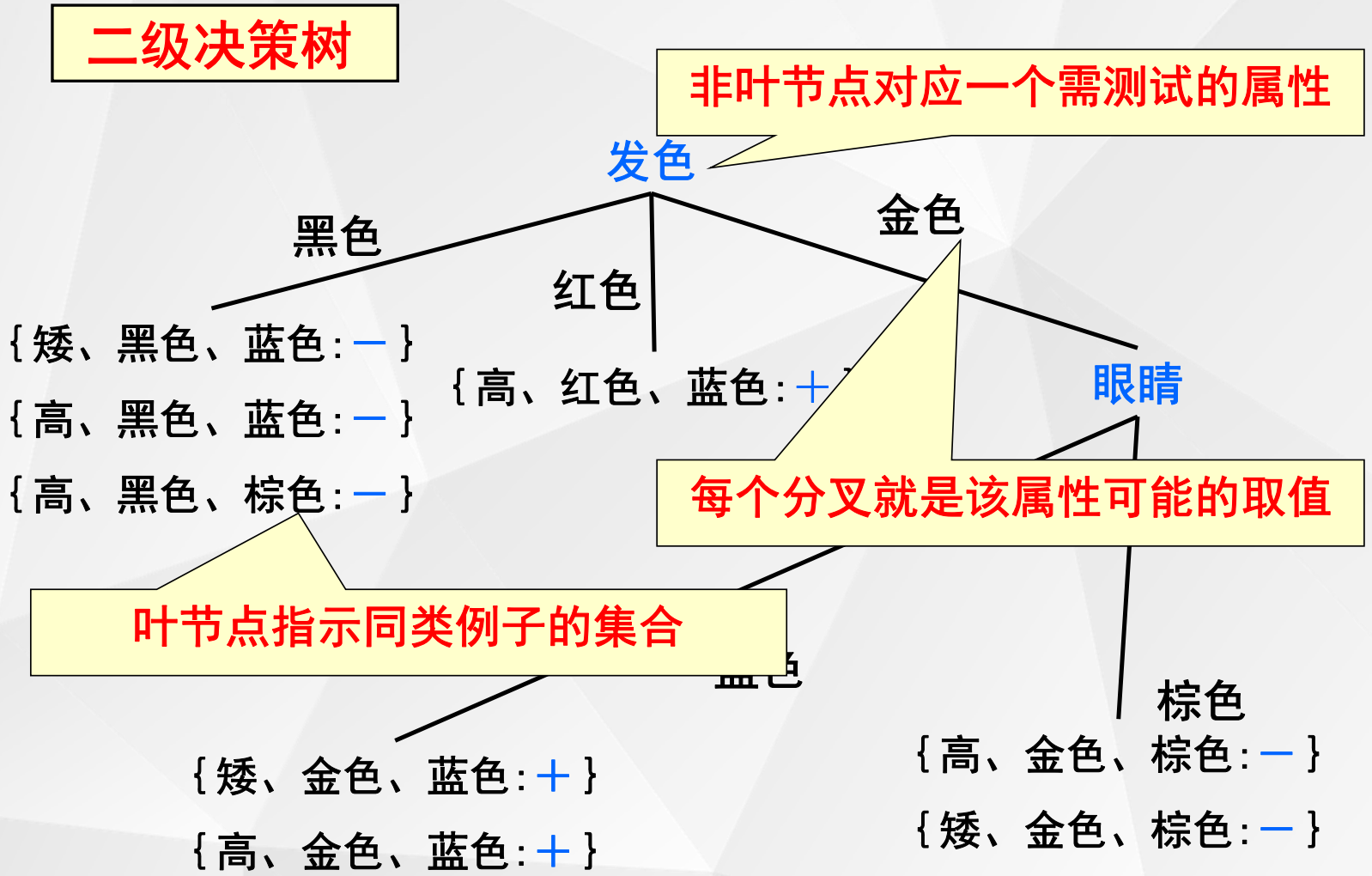


6.4.3 典型的分类算法介绍—决策树算法

- 按属性“眼睛”划分“金色”分支：
  - 2个属性值——2个对象子集

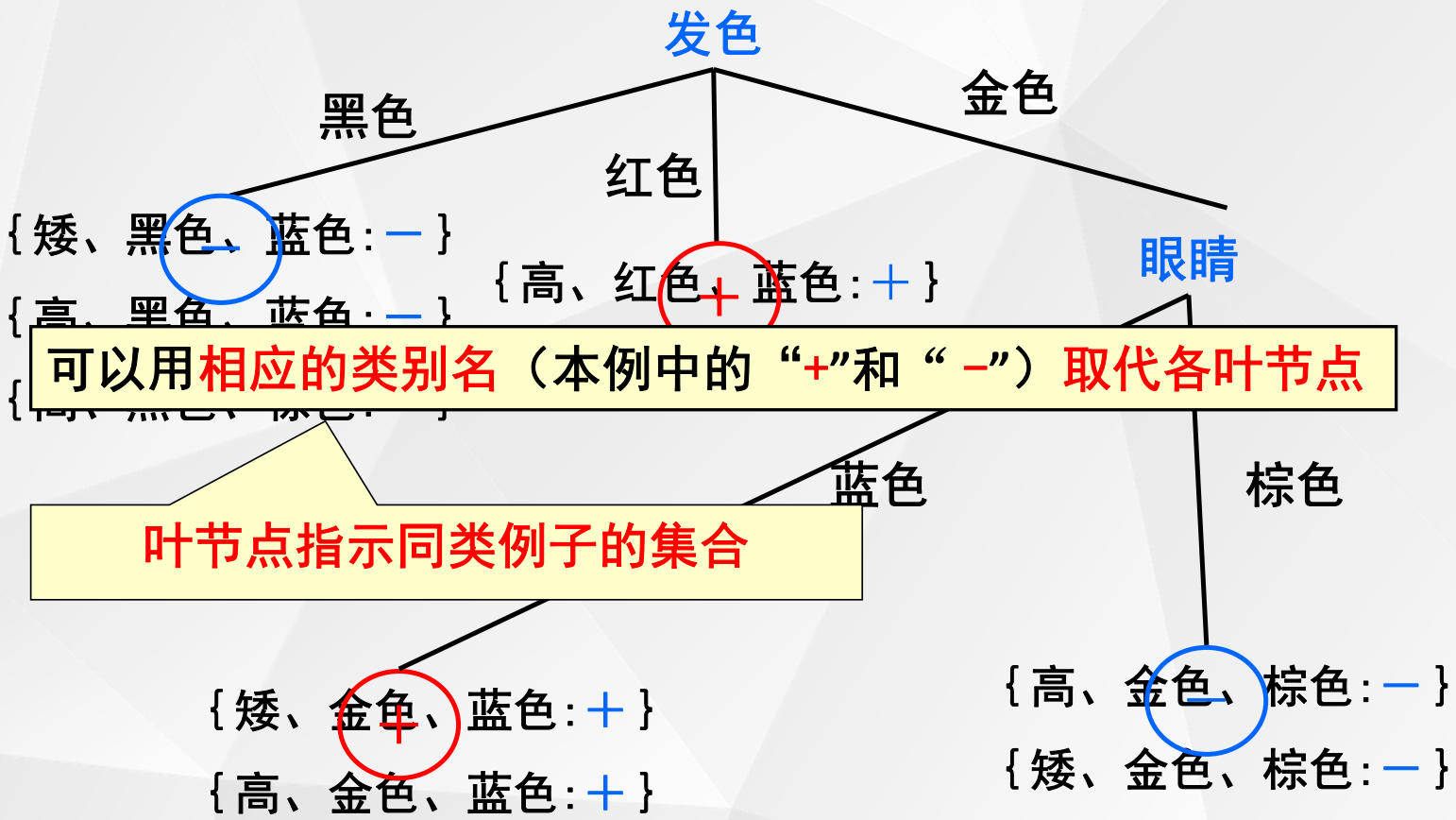


6.4.3 典型的分类算法介绍—决策树算法

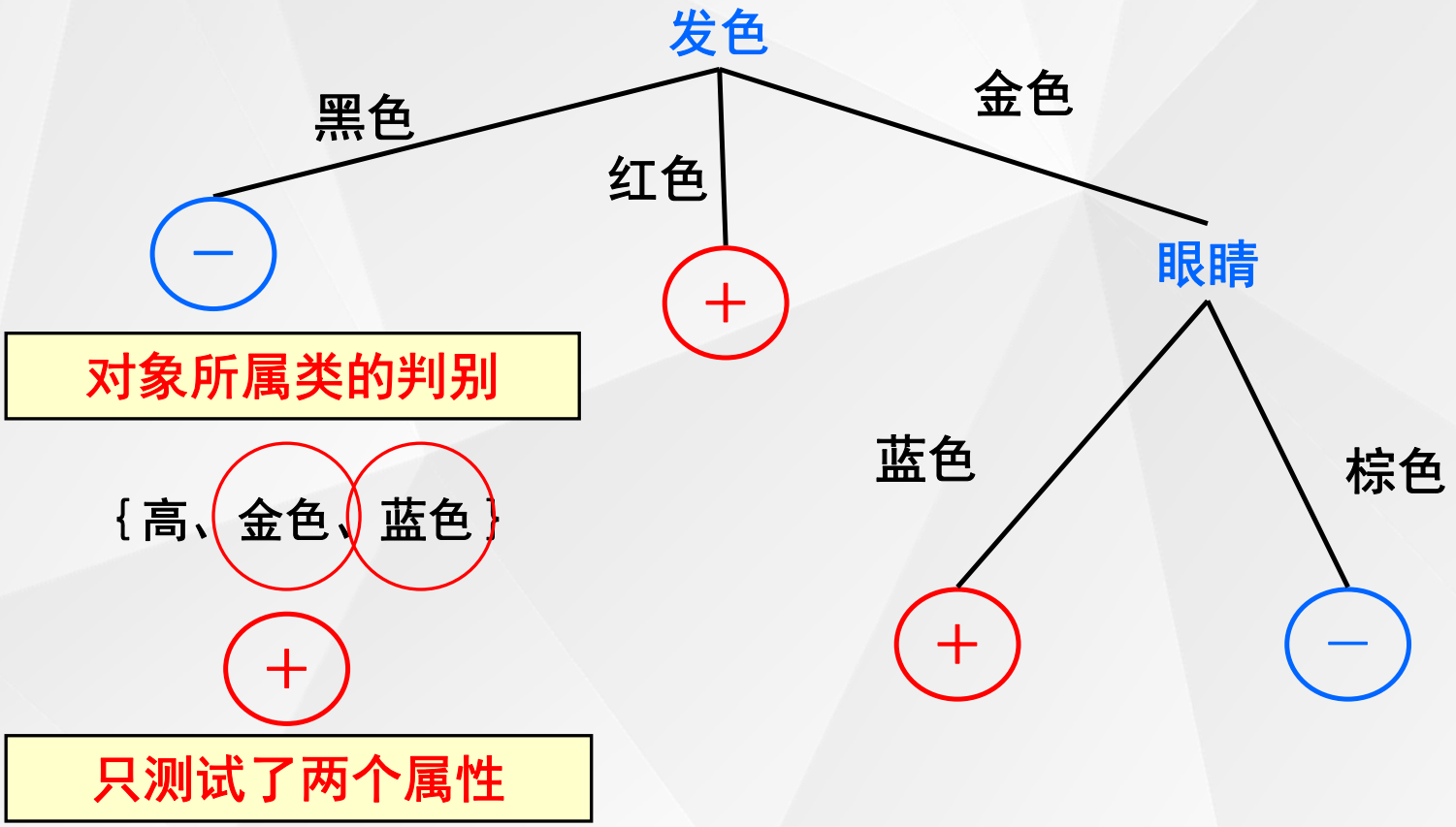


6.4.3 典型的分类算法介绍—决策树算法

二级决策树生成



6.4.3 典型的分类算法介绍—决策树算法



6.4.3 典型的分类算法介绍—决策树算法

- 预先定义一组属性及其可取值：
  - 高度{高，矮}；
  - 发色{黑色, 红色，金色}；
  - 眼睛{蓝色，棕色}；
- 人分为两类：
  - “+”
  - “-”

高度

眼睛

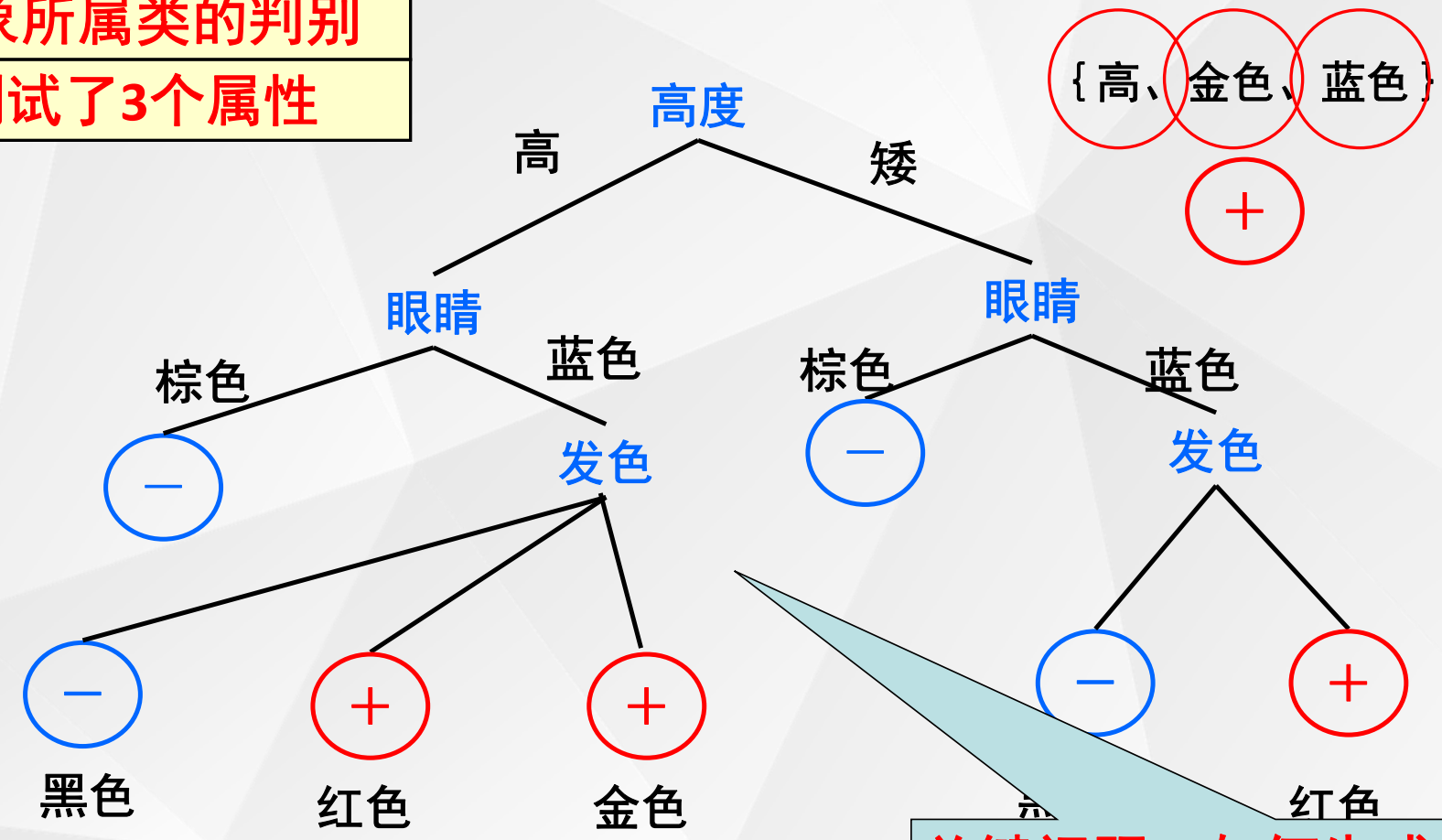
发色

高度	发色	眼睛	类别
矮	黑色	蓝色	-
高	黑色	蓝色	-
矮	金色	蓝色	+
高	金色	棕色	-
高	黑色	棕色	-
矮	金色	棕色	-
高	金色	蓝色	+
高	红色	蓝色	+

6.4.3 典型的分类算法介绍—决策树算法

对象所属类的判别

测试了3个属性



关键问题：如何生成最小决策树？

6.4.3 典型的分类算法介绍—决策树算法

补充知识：信息的定量描述

- 衡量信息多少的物理量称为信息量。
  - 若概率很大，受信者事先已有所估计，则该消息信息量就很小；
  - 若概率很小，受信者感觉很突然，该消息所含信息量就很大。
- 根据客观事实和人们的习惯概念，信息量函数f(p)应满足以下条件：
  - f(p)应是概率p的严格单调递减函数，即当p1>p2， f(p1)<f(p2)；
  - 当p=1时，f(p)=0；
  - 当p=0时，f(p)=∞；
  - 两个独立事件的联合信息量应等于它们分别的信息量之和。

6.4.3 典型的分类算法介绍—决策树算法

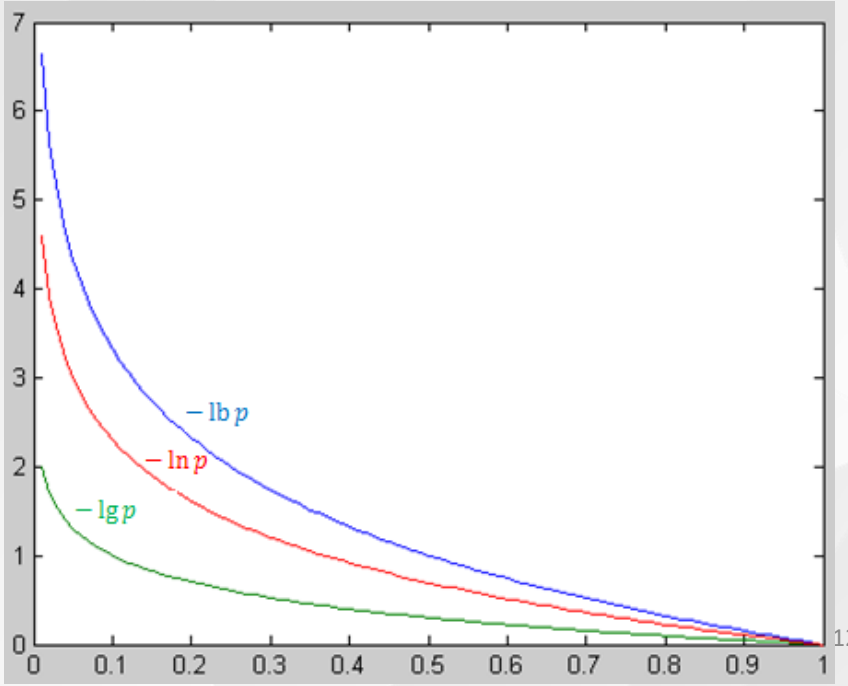
补充知识：信息的定量描述

- 信息量的定义
  - 若一个消息x出现的概率为p，则这一消息所含的信息量为

$$I = -\log p$$

其中，对数的底大于1

- 信息量单位
  - 以2为底时，单位为 bit（binary unit，比特）
  - 以e为底时，单位为 nat（natural unit，奈特）
  - 以10为底时，单位为 hart（Hartley，哈特）



6.4.3 典型的分类算法介绍—决策树算法

补充知识：信息的定量描述

- 抛一枚均匀硬币，出现正面与反面的信息量是多少?

  解：出现正面与反面的概率均为0. 5，它们的信息量是

  I(正)= - $\log_2 p$(正)= - $\log_2$ 0.5=1b

  I(反)= - $\log_2 p$(反)= - $\log_2$ 0.5=1b

- 抛一枚畸形硬币，出现正面与反面的概率分别是1/4，3/4，出现正面与反面时的信息量是多少?

  解：出现正面与反面的概率分别是1/4，3/4，它们的信息量是

  I(正)= - $\log_2 p$(正)= - $\log_2$ 1/4=2b

  I(反)= - $\log_2 p$(反)= - $\log_2$ 3/4=0.415b

6.4.3 典型的分类算法介绍—决策树算法

补充知识：信息的定量描述

- 信源含有的信息量是信源发出的所有可能消息的平均不确定性，香农把信源所含有的信息量称为信息熵，是指每个符号所含信息量的统计平均值。*m*种符号的平均信息量为

$$H(X)=\sum_i p(x_i)I(x_i)=-\sum_i p(x_i)\log p(x_i)$$

6.4.3 典型的分类算法介绍—决策树算法

补充知识：信息的定量描述

- 抛一枚均匀硬币的信息熵是多少?

  解：出现正面与反面的概率均为0. 5，信息熵是

$$H(x) = -\sum_{i=1}^{q} p(x_i)\log\ p(x_i)$$

$$= -(0.5\log\ 0.5 + 0.5\log\ 0.5)$$

$$= 1\mathrm{b}$$

6.4.3 典型的分类算法介绍—决策树算法

补充知识：信息的定量描述

- 抛一枚畸形硬币，出现正面与反面的概率分别是1/4，3/4，出现正面与反面时的信息熵是多少?

  解：出现正面与反面的概率分别是1/4，3/4，信息熵是

$$H(x) = -\sum_{i=1}^{q} p(x_i)\log p(x_i)$$

$$= -(1/4\log 1/4 + 3/4\log 3/4)$$

$$= 0.811\text{b}$$

6.4.3 典型的分类算法介绍—决策树算法

- 面临的问题：
  - 如何选择属性，使生成的决策树最小的?
- ID3算法采用了香农（Shannon）信息论：
  - 目标：使分类时平均的测试次数最小；
  - 给定的例子集C：
    - M(C)：从C判别一个对象的类属所要求的总
      信息量；
- 人分类问题：$M(C) = -P^{+}\log_2 P^{+} - P^{-}\log_2 P^{-}$
  - “+”类消息的概率P^{+}；
  - “-”类消息的概率P^{-}；

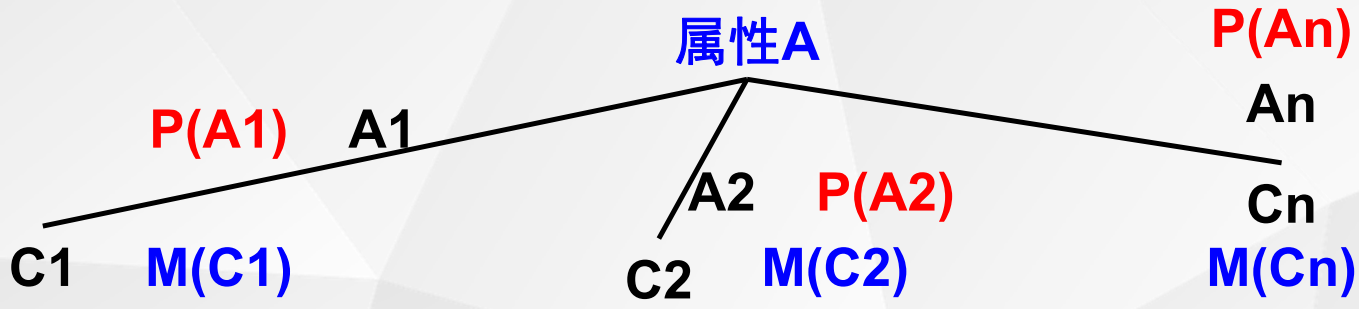
高度	发色	眼睛	类别
矮	黑色	蓝色	-
高	黑色	蓝色	-
矮	金色	蓝色	+
高	金色	棕色	-
高	黑色	棕色	-
矮	金色	棕色	-
高	金色	蓝色	+
高	红色	蓝色	+

概率近似地表示为相对频率
$P^{+}=3/8$

对于上述例子，C集有 8 个例子，3个为“+”，5为"-"，则
$M(C) = -(3/8)\log_2(3/8) - (5/8)\log_2(5/8) = 0.954$ bits

6.4.3 典型的分类算法介绍—决策树算法

- A为构造C的决策树时下一个可能选取的属性；
  - {A1,A2,..,An}为属性A的值且是互斥的；
  - 属性A将集合C划分为若n个子集合；
    - {C1，C2，...，Cn}
- M(Ci)是子集Ci判别一个对象的类属所要求的总的期望信息量；
- B(C, A)：属性A构造决策树后需要的期望信息量：
- ∑(集合C中A值为Ai的概率P(Ai)) * M(Ci)



6.4.3 典型的分类算法介绍—决策树算法

- $M(C) = \sum(-P^i \log_2 P^i)$
  - C判别一个对象的类属所要求的总的期望信息量；
- M(Ci)
  - Ci判别一个对象的类属所要求的总的期望信息量；
- B(C, A) = ∑(C中A值为Ai的概率P(Ai)) * M(Ci)
  - C按属性A构造决策树后需要的期望信息量 ；
- M(C)-B(C, A)越大
  - 说明测试这个属性A所能传递的信息量越大；
  - 判别的速度也就越快；
- 关键：选择M(C)-B(C, A)最大的属性A生成决策树； ★

6.4.3 典型的分类算法介绍—决策树算法

“高”的分支的所需期望信息量为：$M(C_{高})$

$-(2/5)\log_2(2/5)-(3/5)\log_2(3/5)=0.971$ bits

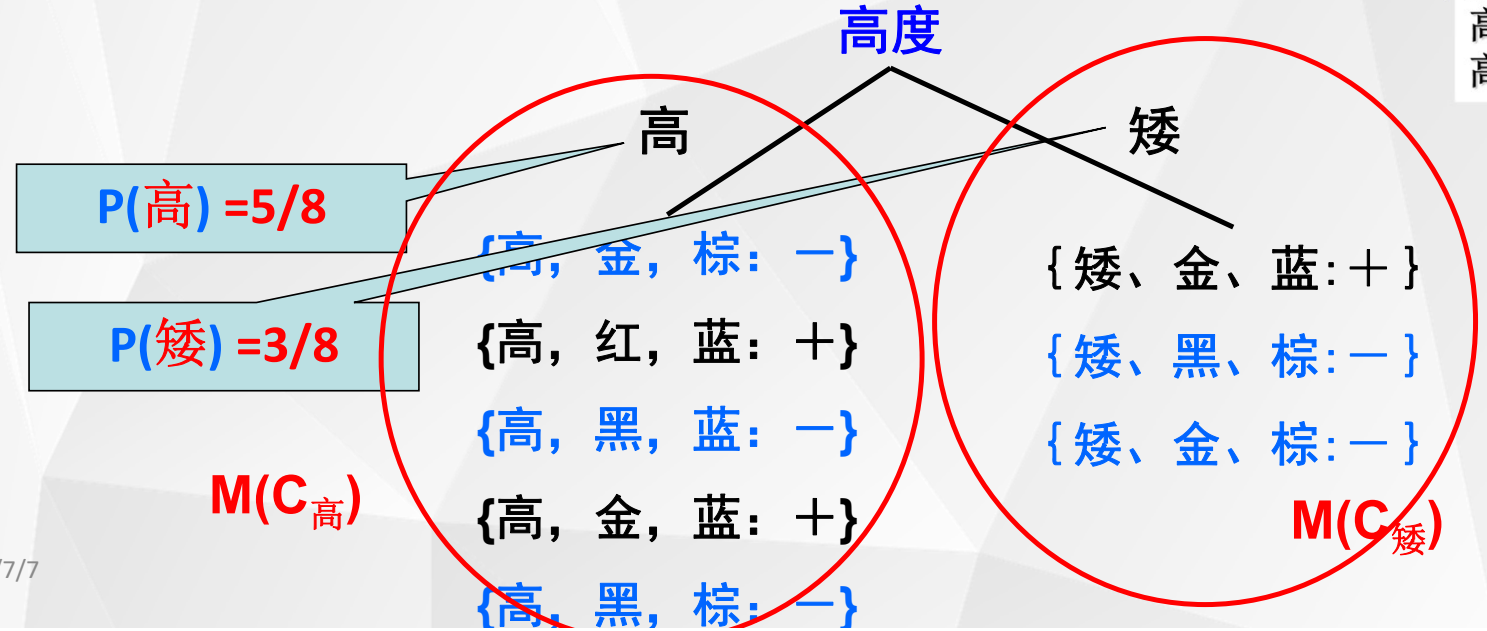
“矮”的分支的所需期望信息量为：$M(C_{矮})$

$-(1/3)\log_2(1/3)-(2/3)\log_2(2/3)=0.918$ bits

C以属性“高度”作划分后进一步判别所需的期望信息量为：

$B(C,\text{“高度”})=5/8\times M(C_{高})+3/8\times M(C_{矮})=0.951$ bits

高度	发色	眼睛	类别
矮	黑色	蓝色	—
高	黑色	蓝色	—
矮	金色	蓝色	+
高	金色	棕色	—
高	黑色	棕色	—
矮	金色	棕色	—
高	金色	蓝色	+
高	红色	蓝色	+



6.4.3 典型的分类算法介绍—决策树算法

以属性“高度”作划分后进一步判别所需的期望信息量为：

$$B(C,\text{"高度"}) = 5/8 \times 0.971 + 3/8 \times 0.918 = 0.951 \text{ bits}$$

对于上述例子，C集有 8 个例子，3个为“+”，5为"-"，则

$$M(C) = -(3/8)\log_2(3/8) - (5/8)\log_2(5/8) = 0.954 \text{ bits}$$

测试这属性“高度”传递的信息为：

$$M(C)-B(C,\text{"高度"}) = 0.954 - 0.951 = 0.003 \text{ bits}$$

属性“发色”作为根节点构造决策树

6.4.3 典型的分类算法介绍—决策树算法

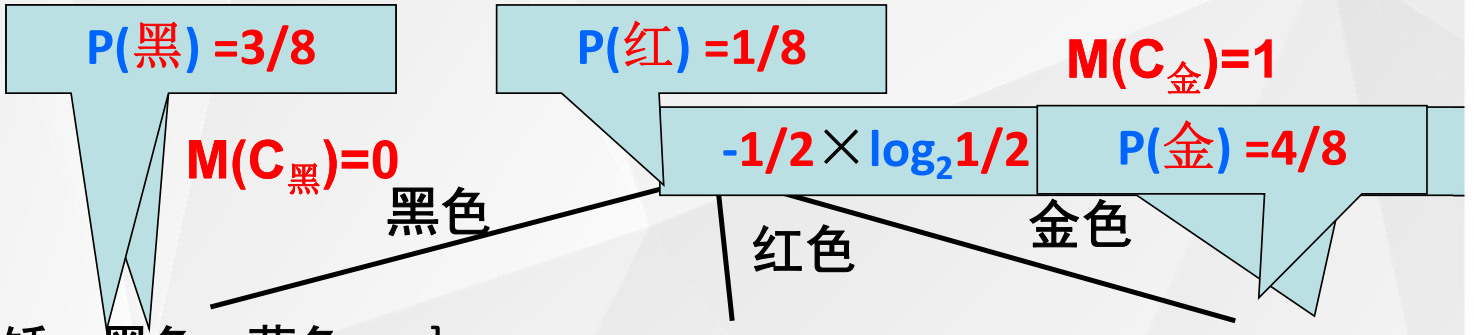
以属性“发色”作划分后进一步判别所需的期望信息量为：

$B(C,\text{“发色”}) = 3/8 \times 0 + 1/8 \times 0 + 4/8 \times 1 = 0.5$ bits

测试这属性“发色”传递的信息为：

$M(C)-B(C,\text{"发色"}) = 0.954 - 0.5 = 0.454$ bits

高度	发色	眼睛	类别
矮	黑色	蓝色	—
高	黑色	蓝色	—
矮	金色	蓝色	+
高	金色	棕色	—
高	黑色	棕色	—
矮	金色	棕色	—
高	金色	蓝色	+
高	红色	蓝色	+



{矮、黑色、蓝色：—}

{高、黑色、蓝色：—}

{高、黑色、棕色：—}

{高、红色、蓝色：+}

$M(C_{红})=0$

$-1 \times \log_2 1 = 0$

{矮、金色、蓝色：+}

{高、金色、棕色：—}

{高、金色、蓝色：+}

{矮、金色、棕色：—}

6.4.3 典型的分类算法介绍—决策树算法

对于上述例子，C集有 8 个例子，3个为“+”，5为"-"，则

$M(C) = -（3/8）\log_2（3/8）-（5/8）\log_2（5/8）= 0.954$ bits

测试这属性“高度”传递的信息为：

M(C)-B(C,"高度") = 0.954 - 0.951 = 0.003 bits

测试这属性“发色”传递的信息为：

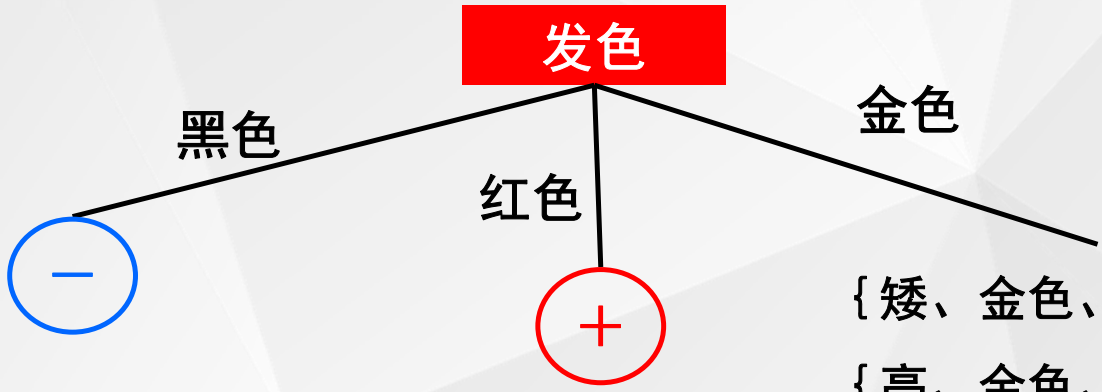
M(C)-B(C,"发色") = 0.954 - 0.5 = 0.454 bits

测试这属性“眼睛”传递的信息为：

M(C)-B(C,"眼睛") = 0.347 bits

高度	发色	眼睛	类别
矮	黑色	蓝色	−
高	黑色	蓝色	−
矮	金色	蓝色	+
高	金色	棕色	−
高	黑色	棕色	−
矮	金色	棕色	−
高	金色	蓝色	+
高	红色	蓝色	+

6.4.3 典型的分类算法介绍—决策树算法



{矮、金色、蓝色:＋}

{高、金色、棕色:－}

{高、金色、蓝色:＋}

{矮、金色、棕色:－}

6.4.3 典型的分类算法介绍—决策树算法

C1集有4个例子，2个为“+”，2为“-”，则

$M(C1) = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$ bits

以属性“眼睛”作划分后进一步判别所需的期望信息量为：

$B(C1,\text{“眼睛”}) = 1/2 \times 0 + 1/2 \times 0 = 0$ bits

测试这属性“眼睛”传递的信息为：

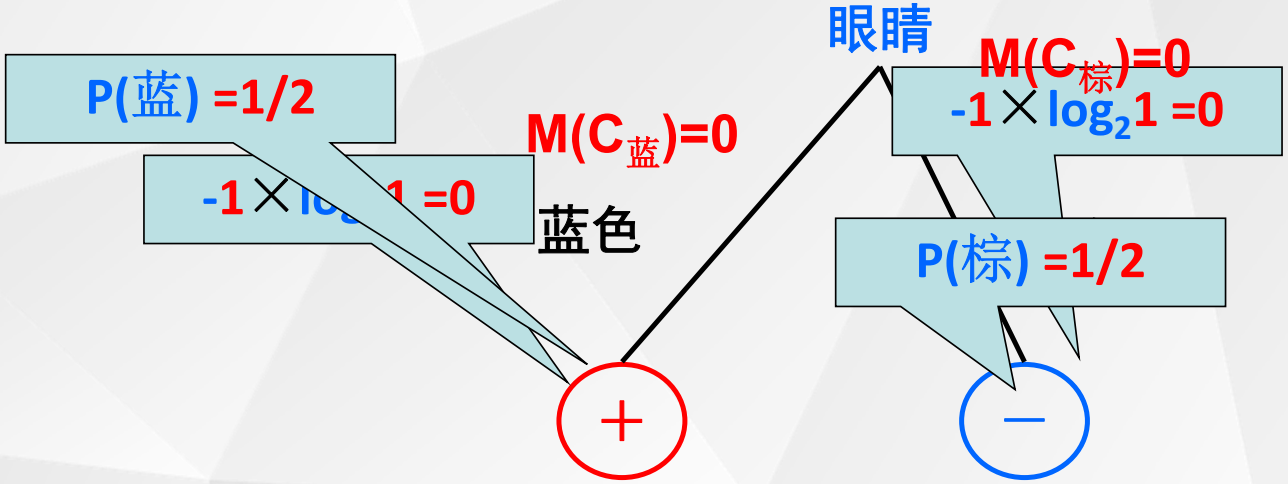
$M(C1) - B(C1,\text{“眼睛”}) = 1 - 0 = 1$ bits

{矮、金色、蓝色：＋}

{高、金色、棕色：－}

{高、金色、蓝色：＋}

{矮、金色、棕色：－}



6.4.3 典型的分类算法介绍—决策树算法

C1集有4个例子，2个为“+”，2为“-”，则

$M(C1) = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$ bits

以属性“高度”作划分后进一步判别所需的期望信息量为：

$B(C1,\text{“高度”}) = 1/2 \times 1 + 1/2 \times 1 = 1$ bits

测试这属性“高度”传递的信息为：

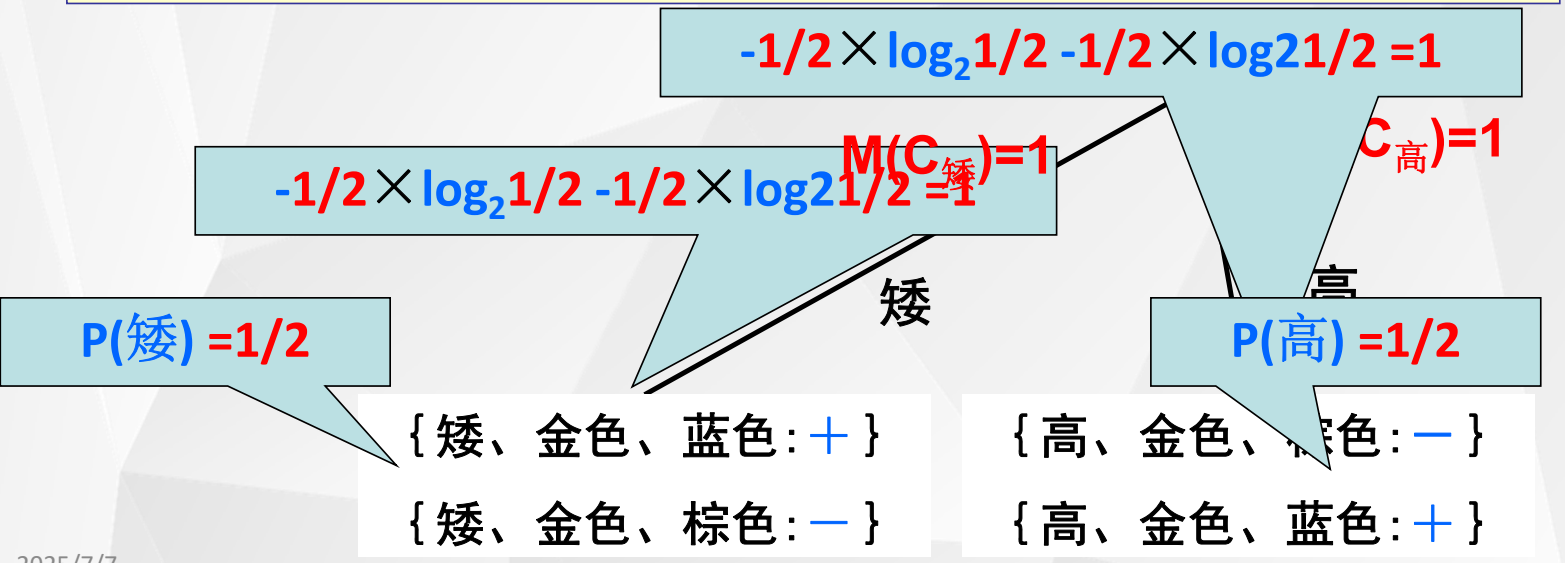
$M(C1) - B(C1,\text{“高度”}) = 1 - 1 = 0$ bits

{矮、金色、蓝色:＋}

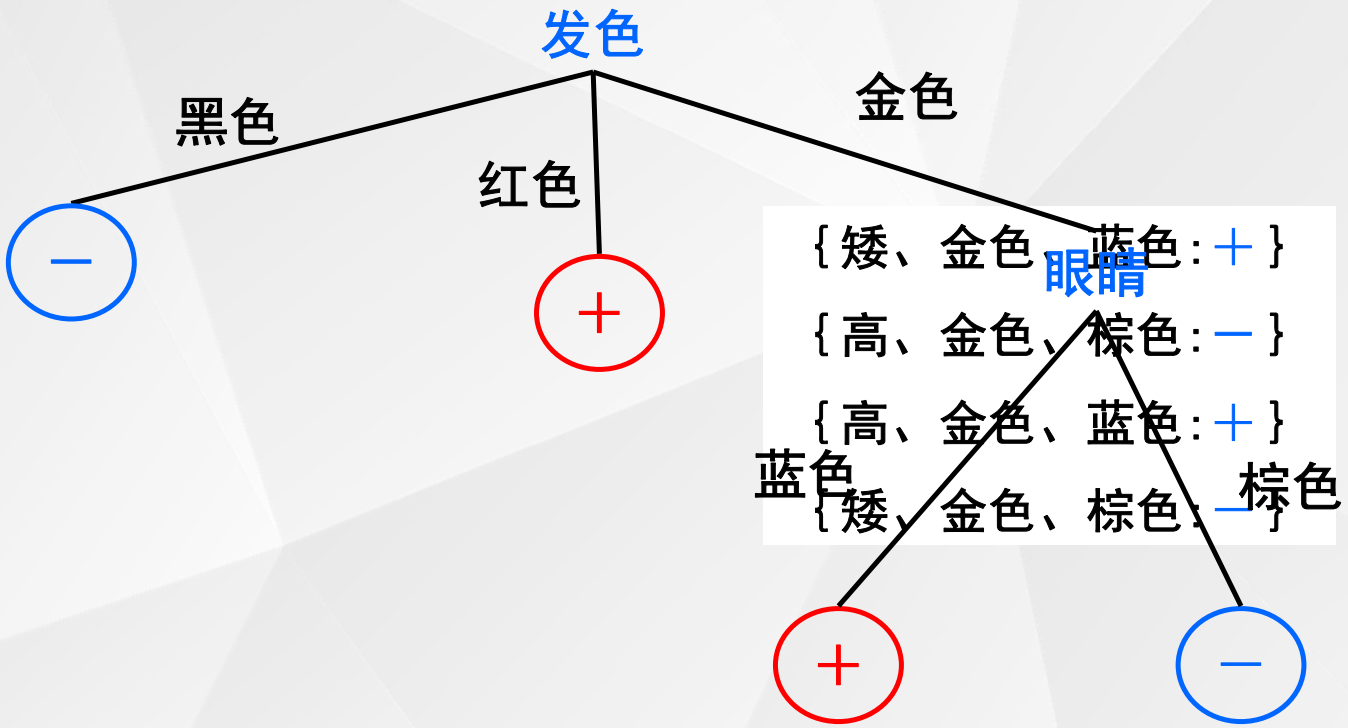
{高、金色、棕色:－}

{高、金色、蓝色:＋}

{矮、金色、棕色:－}



6.4.3 典型的分类算法介绍—决策树算法



测试这属性“眼睛”传递的信息为：

M(C1)-B(C1,“眼睛") = 1 - 0 = 1 bits

测试这属性“高度”传递的信息为：

M(C1)-B(C1,“高度") = 1 - 1 = 0 bits

主观题 10分

设置

练习：构造是否进行打网球的决策树
要求：只要求写出按不同状况的根节点值的算式，不要求计算结果

天气	温度	湿度	风速	活动
晴	炎热	高	弱	取消
晴	炎热	高	强	取消
阴	炎热	高	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
雨	寒冷	正常	强	取消
阴	寒冷	正常	强	进行
晴	适中	高	弱	取消
晴	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
晴	适中	正常	强	进行
阴	适中	高	强	进行
阴	炎热	正常	弱	进行
雨	适中	高	强	取消

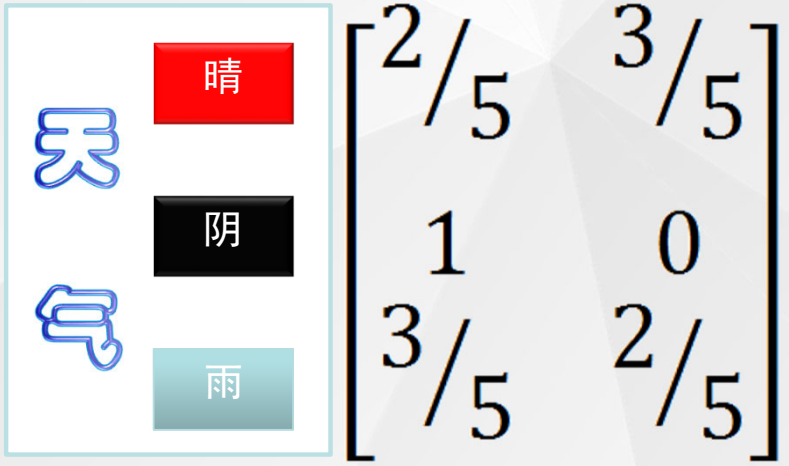
作答

6.4.3 典型的分类算法介绍—决策树算法

天气	温度	湿度	风速	活动
阴	炎热	高	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
阴	寒冷	正常	强	进行
晴	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
晴	适中	正常	强	进行
阴	适中	高	强	进行
阴	炎热	正常	弱	进行
晴	炎热	高	弱	取消
晴	炎热	高	强	取消
雨	寒冷	正常	强	取消
晴	适中	高	弱	取消
雨	适中	高	强	取消

M(C) = = - (9/14)* **$\log_2$** (9/14) - (5/14)* **$\log_2$** (5/14) = **0.94**

6.4.3 典型的分类算法介绍—决策树算法



天气有三个属性值，晴，阴和雨。其熵分别为：

$M(C_{晴}) = -(2/5)*\log_2(2/5) - (3/5)*\log_2(3/5) = 0.971$

$M(C_{阴}) = -(4/4)*\log_2(4/4) = 0$

$M(C_{雨}) = -(3/5)*\log_2(3/5) - (2/5)*\log_2(2/5) = 0.971$

6.4.3 典型的分类算法介绍—决策树算法

温度	湿度	风速	天气	活动
寒冷	正常	弱	晴	进行
适中	正常	强	晴	进行
炎热	高	弱	晴	取消
炎热	高	强	晴	取消
适中	高	弱	晴	取消
炎热	高	弱	阴	进行
寒冷	正常	强	阴	进行
适中	高	强	阴	进行
炎热	正常	弱	阴	进行
适中	高	弱	雨	进行
寒冷	正常	弱	雨	进行
适中	正常	弱	雨	进行
寒冷	正常	强	雨	取消
适中	高	强	雨	取消

$$B(C,\text{“天气”}) = 5/14 * M(C_{\text{晴}}) + 4/14 * M(C_{\text{阴}}) + 5/14 * M(C_{\text{雨}})$$

$$= (5/14)*0.971 + (4/14)*0 + (5/14)*0.971$$

$$= 0.693$$

6.4.3 典型的分类算法介绍—决策树算法

天气	湿度	风速	温度	活动
阴	高	弱	炎热	进行
阴	正常	弱	炎热	进行
晴	高	弱	炎热	取消
晴	高	强	炎热	取消
雨	高	弱	适中	进行
雨	正常	弱	适中	进行
晴	正常	强	适中	进行
阴	高	强	适中	进行
晴	高	弱	适中	取消
雨	高	强	适中	取消
雨	正常	弱	寒冷	进行
阴	正常	强	寒冷	进行
晴	正常	弱	寒冷	进行
雨	正常	强	寒冷	取消

$B(C, \text{“温度”}) = 0.911$

6.4.3 典型的分类算法介绍—决策树算法

天气	温度	风速	湿度	活动
阴	炎热	弱	高	进行
雨	适中	弱	高	进行
阴	适中	强	高	进行
晴	炎热	弱	高	取消
晴	炎热	强	高	取消
晴	适中	弱	高	取消
雨	适中	强	高	取消
雨	寒冷	弱	正常	进行
阴	寒冷	强	正常	进行
晴	寒冷	弱	正常	进行
雨	适中	弱	正常	进行
晴	适中	强	正常	进行
阴	炎热	弱	正常	进行
雨	寒冷	强	正常	取消

B(C, “湿度”) = 0.789

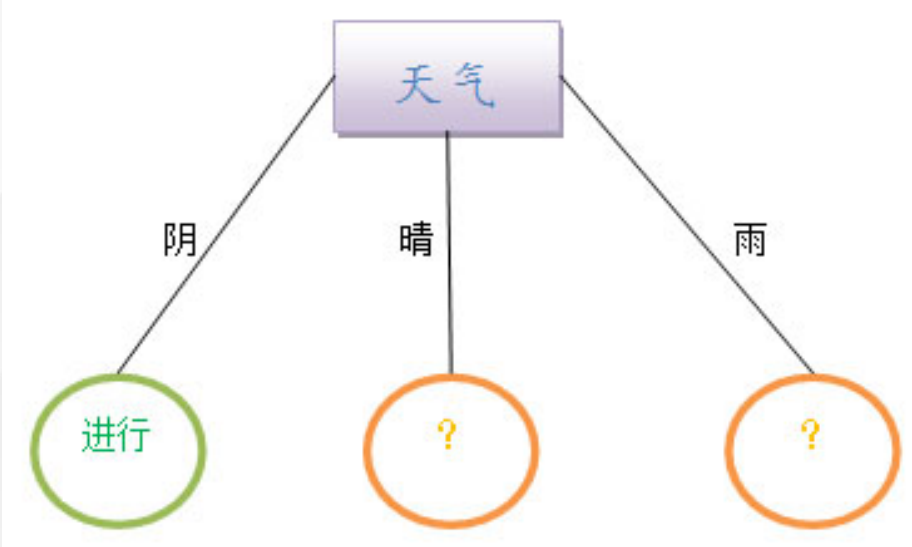
6.4.3 典型的分类算法介绍—决策树算法

天气	温度	湿度	风速	活动
阴	寒冷	正常	强	进行
晴	适中	正常	强	进行
阴	适中	高	强	进行
晴	炎热	高	强	取消
雨	寒冷	正常	强	取消
雨	适中	高	强	取消
阴	炎热	高	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
晴	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
阴	炎热	正常	弱	进行
晴	炎热	高	弱	取消
晴	适中	高	弱	取消

$B(C, \text{“风速”}) = 0.892$

6.4.3 典型的分类算法介绍—决策树算法

- M(C)-B(C,“天气") = 0.94- 0.693 = 0.246
- M(C)-B(C,“温度") = 0.94- 0.911 = 0.029
- M(C)-B(C,“湿度") = 0.94- 0.789 = 0.151
- M(C)-B(C,“风速") = 0.94- 0.892 = 0.048



6.4.3 典型的分类算法介绍—决策树算法

天气	温度	湿度	风速	活动
晴	炎热	高	弱	取消
晴	炎热	高	强	取消
阴	炎热	高	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
雨	寒冷	正常	强	取消
阴	寒冷	正常	强	进行
晴	适中	高	弱	取消
晴	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
晴	适中	正常	强	进行
阴	适中	高	强	进行
阴	炎热	正常	弱	进行
雨	适中	高	强	取消

天气	温度	湿度	风速	活动
晴	寒冷	正常	弱	进行
晴	适中	正常	强	进行
晴	炎热	高	弱	取消
晴	炎热	高	强	取消
晴	适中	高	弱	取消
阴	炎热	高	弱	进行
阴	寒冷	正常	强	进行
阴	适中	高	强	进行
阴	炎热	正常	弱	进行
雨	适中	高	弱	进行
雨	寒冷	正常	弱	进行
雨	适中	正常	弱	进行
雨	寒冷	正常	强	取消
雨	适中	高	强	取消

晴

温度	湿度	风速	活动
寒冷	正常	弱	进行
适中	正常	强	进行
炎热	高	弱	取消
炎热	高	强	取消
适中	高	弱	取消

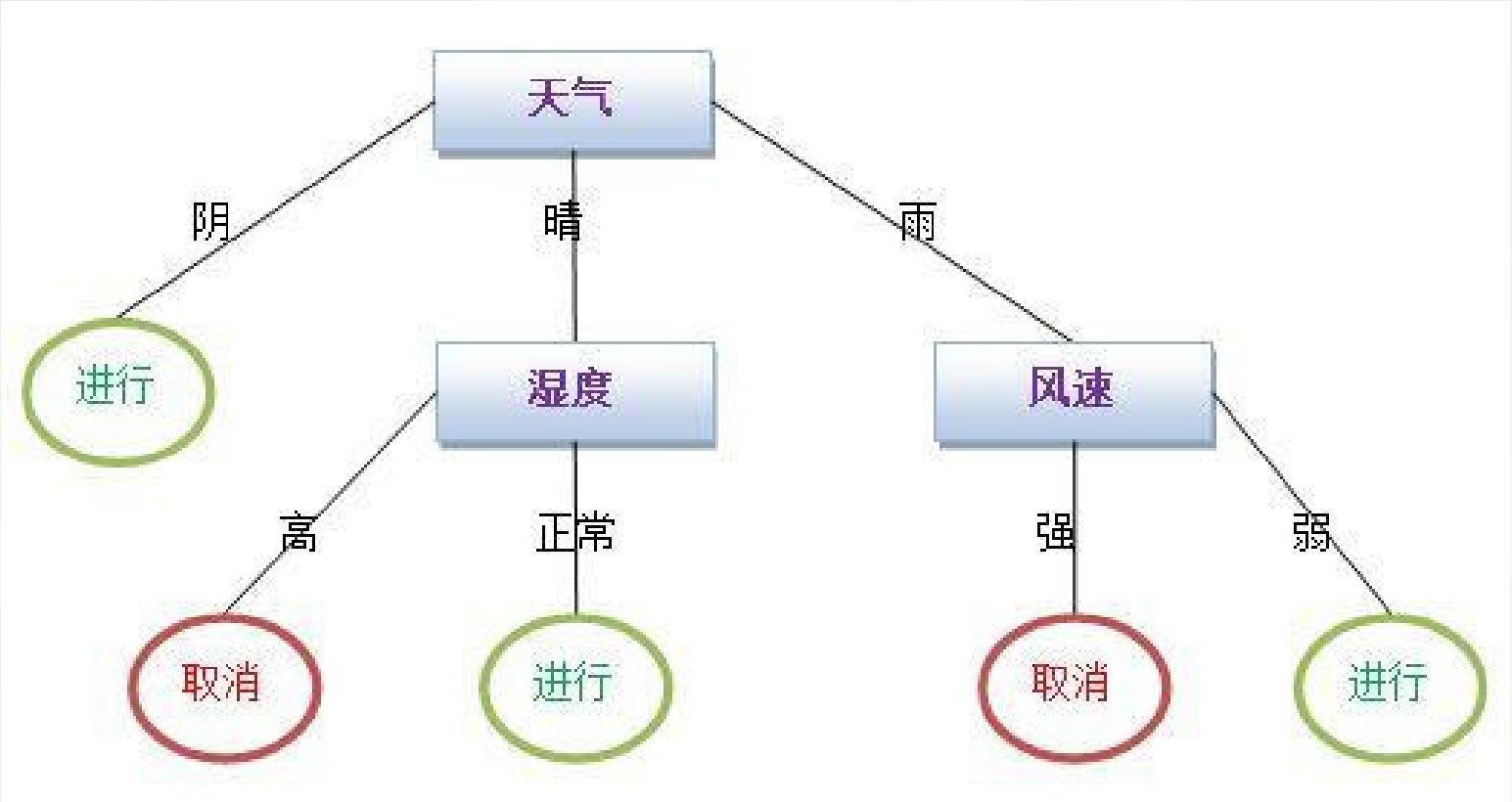
阴

温度	湿度	风速	活动
炎热	高	弱	进行
寒冷	正常	强	进行
适中	高	强	进行
炎热	正常	弱	进行

雨

温度	湿度	风速	活动
适中	高	弱	进行
寒冷	正常	弱	进行
适中	正常	弱	进行
寒冷	正常	强	取消
适中	高	强	取消

6.4.3 典型的分类算法介绍—决策树算法



最终生成的决策树

主观题 10分

设置

练习

小明在玩穿越火线游戏时会根据自己的武器、子弹数量、生命值的血量采取不同的策略行为，请利用ID3算法画出他玩此游戏的最小决策树。

（注：计算过程中可能用到的对数值如下：

$\log_2(2/3)=-0.585$ $\log_2(1/3)=-1.585$

不要求画出完整决策树，只要求指出根节点。）

武器	子弹数量	生命值	行为
机枪	多	少	战斗
机枪	少	多	逃跑
手枪	少	多	战斗
手枪	少	少	逃跑

作答

解析

武器	子弹数量	生命值	行为
机枪	多	少	战斗
机枪	少	多	逃跑
手枪	少	多	战斗
手枪	少	少	逃跑

$M(C) = = -(1/2)*\log_2(1/2) - (1/2)*\log_2(1/2) = 1$

武器有两个属性值，机枪和手枪。其信息量分别为：

$M(C_{机枪}) = -(1/2)*\log_2(1/2) - (1/2)*\log_2(1/2) = 1$

$M(C_{手枪}) = -(1/2)*\log_2(1/2) - (1/2)*\log_2(1/2) = 1$

$B(C,\text{“武器”}) = (2/4)*1+(2/4)*1=1$

子弹数量有两个属性值，多和少。其信息量分别为：

$M(C_{子弹多}) = -1 \times \log_2 1 = 0$

$M(C_{子弹少}) = -(2/3)*\log_2(2/3) - (1/3)*\log_2(1/3) = 0.918$

$B(C,\text{“子弹数量”}) = (1/4)*0+(3/4)*0.918=0.689$

解析

武器	子弹数量	生命值	行为
机枪	多	少	战斗
机枪	少	多	逃跑
手枪	少	多	战斗
手枪	少	少	逃跑

生命值血量有两个属性值，多和少。其信息量分别为：

$M(C_{生命值多}) = -(1/2)*\log_2(1/2) - (1/2)*\log_2(1/2) = 1$

$M(C_{生命值少}) = -(1/2)*\log_2(1/2) - (1/2)*\log_2(1/2) = 1$

B(C,“生命值”) =(2/4)*1+(2/4)*1=1

M(C)-B(C,“武器") = 1- 1 = 0

M(C)-B(C,“子弹数量") = 1- 0.689 = 0.311

M(C)-B(C,“生命值") = = 1- 1 = 0

所以，根节点应该选取子弹数量作为属性。

解析

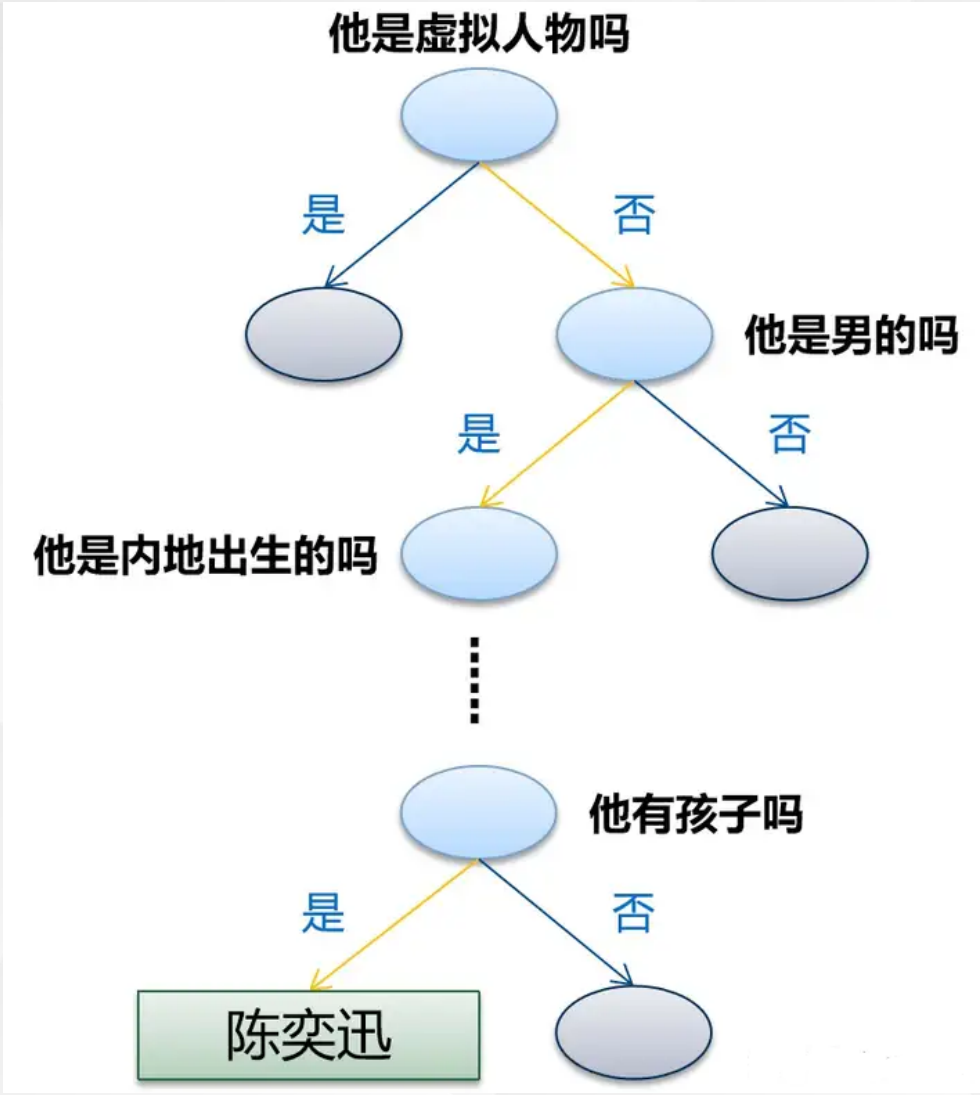
由于武器类型和生命值血量的增益都是一样，所以可以有两种合理的画法，下面给出其中一种画法。

决策树算法示例


微软小冰 • 读心术
规则很简单。你在心里想好一个人的名字，然后按下【开始】。我将问你15个问题。之后，我就会轻松地猜到那个人是谁。吼吼，我已经准备好了，开始吧?
开始
我在观察你的人类心理模型。。。开始！听好，第一个问题是 ——他是虚拟的吗?
不是
他是男的吗?
是
他是在中国内地出生的吗?


微软小冰 • 读心术
最后一个问题！他有孩子吗?
好！现在是见证奇迹的时刻。你心里想的这个人，就是 ——
是陈奕迅!

决策树算法示例



小冰的读心术

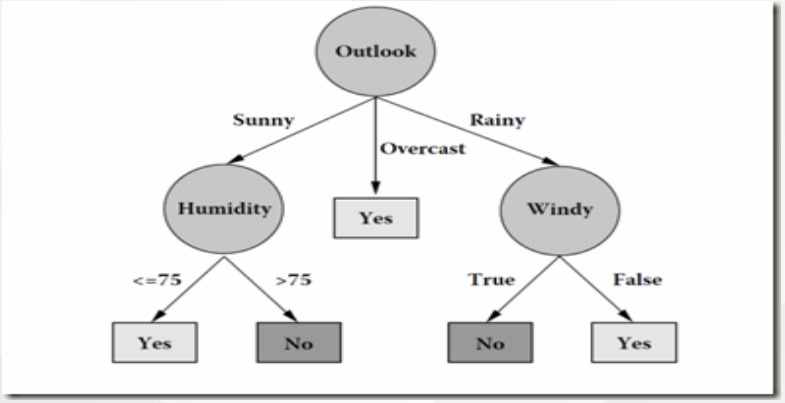
6.4.3 典型的分类算法介绍—决策树算法

- ID3算法总结
  - 优点：
    - 计算复杂度不高
    - 输出结果易于理解
    - 可以处理不相关特征数据
  - 缺点：
    - 不能处理带有缺失值的数据集
    - 在进行算法学习之前需要对数据集中的缺失值进行预处理

决策树算法总结

- 决策树
  - 决策树是用于分类的主要技术，是以实例为基础的归纳学习算法，它着眼于从一组无次序、无规则的实例中推理出以决策树表示的分类规则
  - 构造决策树的目的是找出属性和类别间的关系，用来预测将来未知类别的记录的类别
  - 它采用自顶向下的递归方式，在决策树的内部节点进行属性的比较，并根据不同属性值判断从该节点向下的分支，在决策树的叶节点得到结论
- 主要的决策树算法有ID3、**C4.5**（C5.0）、CART、PUBLIC、SLIQ和SPRINT算法等
- 它们在选择测试属性采用的技术、生成的决策树的结构、剪枝的方法以及时刻，能否处理大数据集等方面都有各自的不同之处
- 决策树模型的缺点：
  - 处理缺失数据时的困难
  - 过度拟合问题
  - 忽略数据集中属性之间的相关性

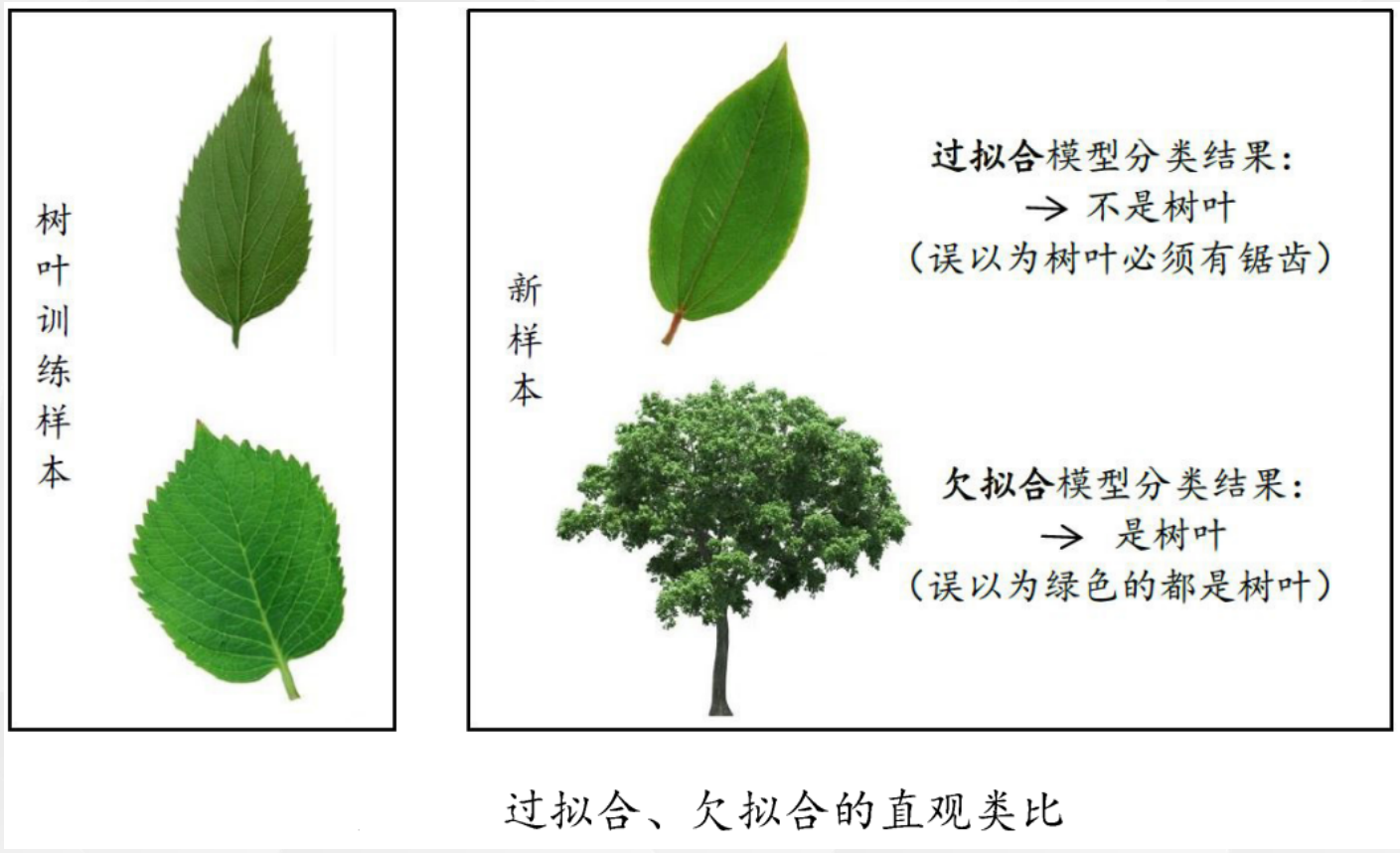
Day	Outlook	Temperature	Humidity	Windy	Play Golf?
1	Sunny	85	85	False	No
2	Sunny	80	90	True	No
3	Overcast	83	78	False	Yes
4	Rainy	70	96	False	Yes
5	Rainy	68	80	False	Yes
6	Rainy	65	70	True	No
7	Overcast	64	65	True	Yes
8	Sunny	72	95	False	No
9	Sunny	69	70	False	Yes
10	Rainy	75	80	False	Yes
11	Sunny	75	70	True	Yes
12	Overcast	72	90	True	Yes
13	Overcast	81	75	False	Yes
14	Rainy	71	80	True	No



经验误差与过拟合

- 过拟合:
  - 学习器把训练样本学习的“太好”，将训练样本本身的特点当做所有样本的一般性质，导致泛化性能下降
  - 优化目标加正则项
  - early stop
- 欠拟合:
  - 对训练样本的一般性质尚未学好
  - 决策树:拓展分支
  - 神经网络:增加训练轮数

经验误差与过拟合



过拟合、欠拟合的直观类比

过拟合：学习器把训练样本本身特点当做所有潜在样本都会具有的一般性质.
欠拟合：训练样本的一般性质尚未被学习器学好.

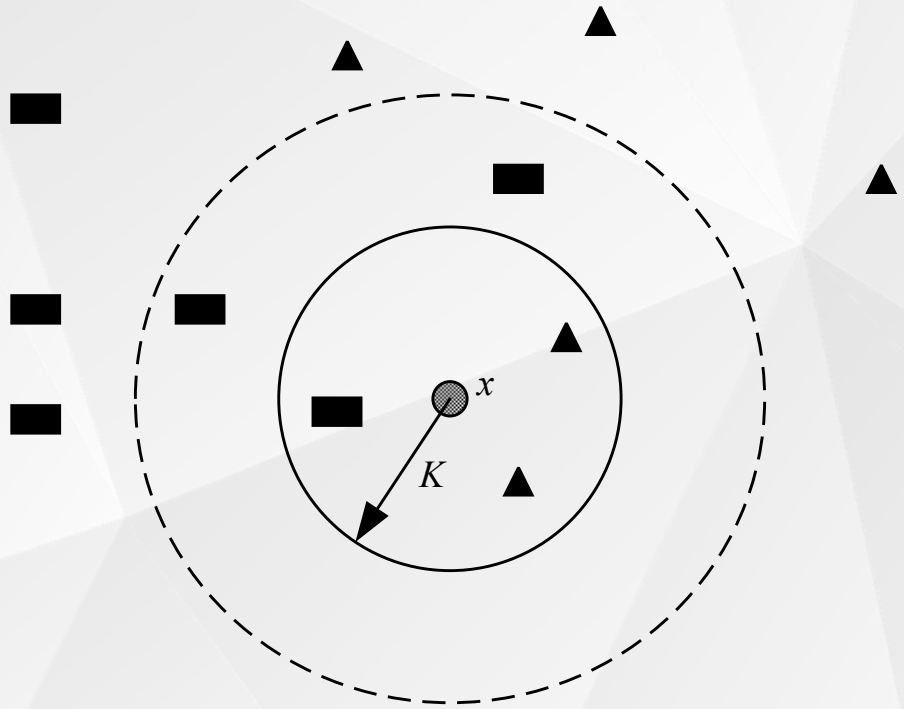
6.4.4 典型的分类算法介绍— K近邻算法

- K近邻算法（K-Nearest Neighbor，KNN）
  - 一种监督学习分类算法；
    - 算法没有学习的过程；
      - 在分类时通过类别已知的样本对新样本的类别进行预测；
      - 属于基于实例的推理方法；
  - 基本思路
    - 通过以某个数据为中心：
      - 分析离其最近的*K*个邻居的类别；
      - 获得该数据可能的类别
    - 如果取*K*=1
      - 待分类样本的类别就是最近邻居的类别

只要训练样本足够多，K近邻算法就能达到很好的分类效果

6.4.4 典型的分类算法介绍— K近邻算法

- 一个实例



■当K=3时，即选择最近的3个点，由于三角形样本所占近邻样本的比例为2/3，于是可以得出圆形输入实例应该为三角形；

■当K=5时，由于长方形样本占近邻样本的比例为3/5，此时测试样本被归为长方形类别。

6.4.4 典型的分类算法介绍— K近邻算法

- 算法原理与描述
  - 原理：利用*K*近邻算法进行数据分类
    - 简单来说，对一个测试样本进行类别分析时
      - 只需要利用测试样本扫描其附近的数据中最大可能属于的分类类别；
      - 然后依据少数服从多数的原理，就可以对测试样本进行分类
    - K近邻算法比较容易理解，但是计算过程却相对复杂，它需要计算出每个数据中心与其他数据的关系，并且进行排序
  - 算法描述
    - 计算测试数据与每一个训练数据之间的距离；
    - 按照距离的递增关系进行排序；
    - 选取距离最小的K个点；
    - 确定前K个点所在类别的出现频率；
    - 返回前K个点中出现频率最高的类别作为测试数据的预测分类。

6.4.4 典型的分类算法介绍— K近邻算法

- 例：某电影院最近要上映一部新的电影《绿皮书》，推广部要策划一系列活动邀请部分会员参加，已知A、B、C、D、E、F等六位VIP会员的近期观影记录，如表1所示。其中，会员A、B、D对该电影表现出观影兴趣，因此推广部试图根据表1的近期观影数据了解其它观众的观影兴趣，以便确定活动邀请名单。

序号	电影名称	电影类型	会员
1	《无名之辈》	剧情 | 搞笑	C、D、F
2	《邪不压正》	搞笑 | 爱情 | 动作	B、D、E
3	《流浪地球》	科幻	A、B、C、D、F
4	《疯狂地球人》	喜剧 | 剧情 | 科幻	B、C、D、E、F
5	《熊出没·原始时代》	剧情 | 搞笑 | 喜剧	E
6	《飞驰人生》	喜剧 | 动作	A、C、D、E
7	《阿丽塔·战斗天使》	动作 | 冒险 | 爱情	A、B、E
8	《一出好戏》	喜剧 | 动画 | 冒险	B、C、D、F
9	《我不是药神》	搞笑 | 剧情	A、B、C、D、E、F
10	《廉政风云》	犯罪 | 悬疑	C、F

表1 观看电影的会员与电影类型示例

6.4.4 典型的分类算法介绍— K近邻算法

- 解：第一步：将表1转换为每个会员的观影记录，并通过6.3.2的方法将会员的观影记录转化为特征向量，如表2所示，为下一步计算用户间相似度做好准备。

会员	电影名称	特征向量
A	《流浪地球》|《飞驰人生》|《阿丽塔·战斗天使》|《我不是药神》	[0 0 1 0 0 1 1 0 1 0]
B	《邪不压正》|《流浪地球》|《疯狂地球人》|《阿丽塔·战斗天使》|《一出好戏》|《我不是药神》	[0 1 1 1 0 0 1 1 1 0]
C	《无名之辈》|《流浪地球》|《疯狂地球人》|《飞驰人生》|《一出好戏》|《我不是药神》|《廉政风云》	[1 0 1 1 0 1 0 1 1 1]
D	《无名之辈 》|《流浪地球》|《疯狂地球人》|《飞驰人生》|《一出好戏》|《我不是药神》	[1 1 1 1 0 1 0 1 1 0]
E	《邪不压正》|《疯狂地球人》|《熊出没·原始时代》|《飞驰人生》|《阿丽塔·战斗天使》|《我不是药神》	[0 1 0 1 1 1 1 0 1 0]
F	《无名之辈》|《流浪地球》|《疯狂地球人》|《一出好戏》|《我不是药神》|《廉政风云》	[1 0 1 1 0 0 0 1 1 1]

表2 会员观看电影的记录

6.4.4 典型的分类算法介绍— K近邻算法

- 第二步：利用余弦相似性或者Jaccard相似系数计算会员之间的观影相似程度，如采用余弦相似性，计算结果如表3所示。

	会员A	会员B	会员C	会员D	会员E	会员F
会员A	1	0.61	0.57	0.57	0.61	0.41
会员B	0.61	1	0.62	0.77	0.67	0.67
会员C	0.57	0.62	1	0.86	0.46	0.93
会员D	0.57	0.77	0.86	1	0.62	0.77
会员E	0.61	0.67	0.46	0.62	1	0.33
会员F	0.41	0.67	0.93	0.77	0.33	1

表3 会员之间的观影相似度（余弦相似度）

6.4.4 典型的分类算法介绍— K近邻算法

- 第三步：根据表3，设定会员之间的观影相似度阈值为0.5，若相似度大于阈值，则视两会员间存在观影关联关系，对关联会员进行权重排序，可得到表4。

会员	存在观影关联关系的会员
会员A	会员B（0.61）、会员E（0.61）、会员C（0.57）、会员D（0.57）
会员B	会员D（0.77）、会员E（0.67）、会员F（0.67）、会员C（0.62）、会员A（0.61）
会员C	会员F（0.93）、会员D（0.86）、会员B（0.62）、会员A（0.57）
会员D	会员C（0.86）、会员B（0.77）、会员F（0.77）、会员E（0.62）、会员A（0.57）
会员E	会员B（0.67）、会员D（0.62）、会员A（0.61）
会员F	会员C（0.93）、会员D（0.77）、会员B（0.67）

表4 对每个会员分析与其存在观影关联关系的会员

6.4.4 典型的分类算法介绍— K近邻算法

- 第四步：通过上述过程，已经划分出每个会员最相近的邻居会员，根据这些数据可以有效地进行很多后续操作。已知会员A、B、D对电影《绿皮书》表现出兴趣，则对应到表4，分析会员C、E、F对电影《绿皮书》系列宣传活动的兴趣程度，通过相似度叠加的方式确定兴趣值。约定*K*=3，即分别与会员C、E、F关联关系最强的三个会员的范围内，与存在观影兴趣会员的相似度进行平均值分析，即平均相似度，得到表5所示内容。

会员	*K*=3的邻居会员	兴趣值
会员C	会员F（0.93）、会员D（0.86）、会员B（0.62）	(0.86+0.62) / 2 = 0.74
会员E	会员B（0.67）、会员D（0.62）、会员A（0.61）	(0.67+0.62+0.61) / 3 = 0.63
会员F	会员C（0.93）、会员D（0.77）、会员B（0.67）	(0.77+0.67) / 2 = 0.72

表5 会员与K=3的邻居会员的平均兴趣值

6.4.4 典型的分类算法介绍— K近邻算法

表5 会员与K=3的邻居会员的平均兴趣值

会员	*K*=3的邻居会员	兴趣值
会员C	会员F（0.93）、会员D（0.86）、会员B（0.62）	(0.86+0.62) / 2 = 0.74
会员E	会员B（0.67）、会员D（0.62）、会员A（0.61）	(0.67+0.62+0.61) / 3 = 0.63
会员F	会员C（0.93）、会员D（0.77）、会员B（0.67）	(0.77+0.67) / 2 = 0.72

- 结论：通过表5可以发现会员C是比较可能去参加电影《绿皮书》的宣传活动，会员E和会员F也有很大可能去参加宣传活动。上述过程是通过K近邻的思想，对具有观影关联关系的会员进行分析，来判定其他会员的可能行为。

6.4.4 典型的分类算法介绍— K近邻算法

- K近邻算法总结
  - 优点：
    - 简单，便于理解和实现
    - 应用范围广
    - 分类效果好
    - 无须进行参数估计
  - 缺点：
    - 样本小时误差难控制
    - 存储所有样本，需要较大存储空间
    - 对于大样本的计算量大
    - *K*值的设定也对算法的结果有较大的影响
      - 如果*K*值过小，将会对数据中存在的噪声过于敏感；
      - 如果*K*值过大，近邻中可能包含属于其他类的样本

6.4.4 典型的分类算法介绍— 贝叶斯

- 贝叶斯
  - 贝叶斯（Bayes）分类算法是一类利用概率统计知识进行分类的算法，如朴素贝叶斯（Naive Bayes）算法
  - 主要利用Bayes定理来预测一个未知类别的样本属于各个类别的可能性，选择其中可能性最大的一个类别作为该样本的最终类别
- 由于贝叶斯定理的成立本身需要一个很强的条件独立性假设，而此假设在实际情况中经常是不成立的，其分类准确性就会下降
- 为此，出现了许多降低独立性假设的贝叶斯分类算法，如TAN（Tree Augmented Naive Bayes)算法，可以考虑属性间的关联性

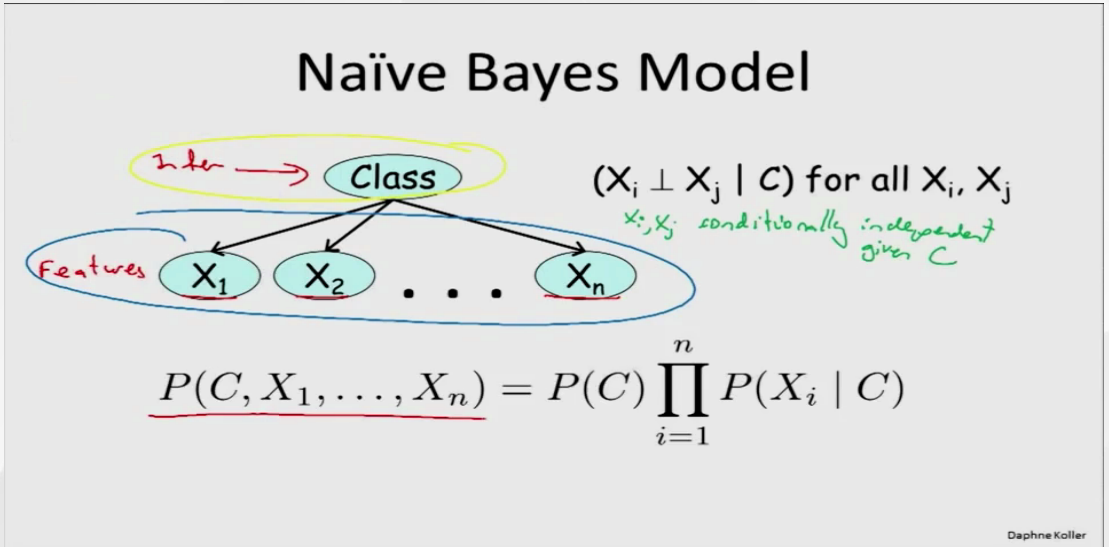
贝叶斯公式：

P(A|B)=P(B|A)*P(A)/P(B)

分类应用：

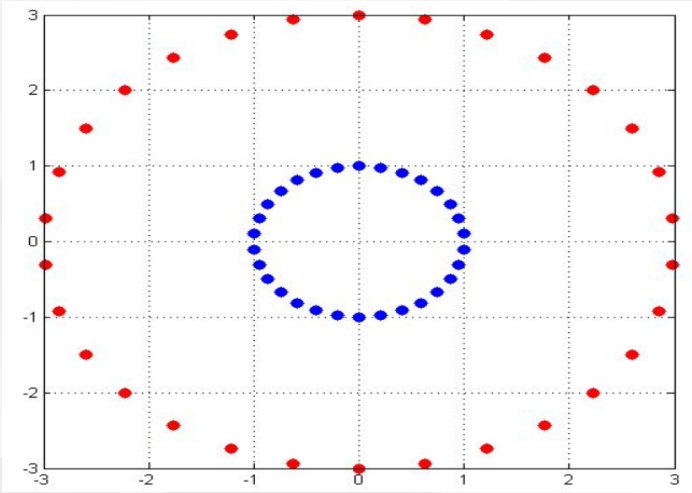
P(类别|样本)=P(样本|类别)*P(类别)

=ΠP（样本各属性|类别）*P(类别)



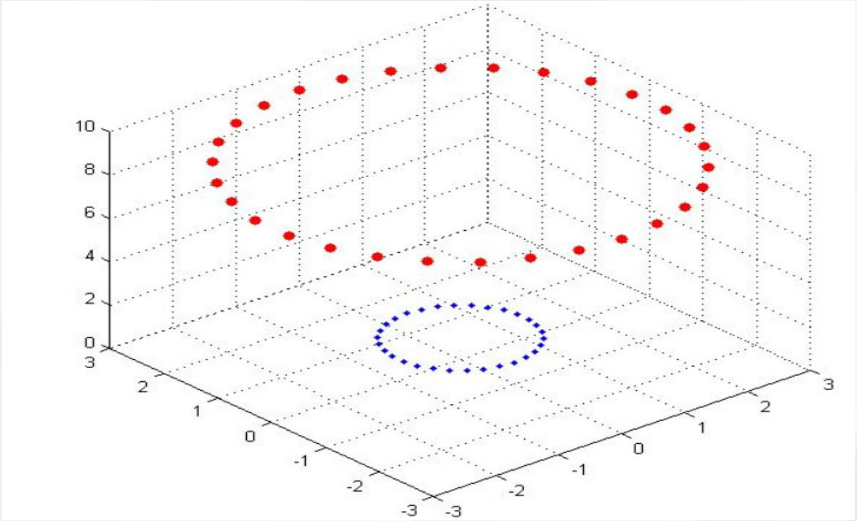
6.4.4 典型的分类算法介绍— SVM

- 支持向量机
  - 支持向量机（SVM，Support Vector Machine）是Vapnik根据统计学习理论提出的一种新的学习方法
  - SVM方法是通过一个非线性映射p，把样本空间映射到一个高维乃至无穷维的特征空间中（Hilbert空间），使得在原来的样本空间中非线性可分的问题转化为在特征空间中的线性可分的问题
  - 它的最大特点是根据结构风险最小化准则，以最大化分类间隔构造最优分类超平面来提高学习机的泛化能力，较好地解决了非线性、高维数、局部极小点等问题
  - SVM一般只能用在二类问题，对于多类问题效果不好



点集A服从{x,y|x^2+y^2=1}

点集B服从{x,y|x^2+y^2=9}



映射(x,y)->(x,y,x^2+y^2)

6.5 聚类算法分析

- 基于“物以类聚”的思想，通过计算对象个体或对象类之间的相似程度，将满足相似条件的对象个体或对象类分入同一类（簇）内；
- 将不满足相似条件的对象个体或对象类分入不同类（簇）中，使划分结果满足类内元素相似度高、类间元素相似程度低的要求。
  - 典型的无监督的学习方法
  - 目标是通过对无标记训练样本的学习来揭示数据的内在性质和规律，为进一步的数据分析提供基础

- **数据：**某文献下载商城某月活跃用户信息及相关下载记录
- **聚类维度：**选择最新下载的时间间隔(R)、下载的频率(F)以及费用(M)进行聚类
- 使用MongoDB统计每个用户对应的3个聚类属性
- 使用Mahout的Canopy+K-means聚类模型在Hadoop平台计算聚类结果

用户类别	下载次数	下载类别	付费总额	占比	特征
沉默型	0.92	0.86	0.06	48.6%	下载不活跃，几乎不贡献收入
试探型	1.42	1.36	8.81	15.3%	试探性下载，有付费意愿
免费型	4.8	4	0.36	22.6%	下载比较活跃，但基本下载免费应用
价值型	10.1	9.7	103.5	13.5%	下载次数种类都很多，付费意愿强烈

聚类案例：某文件下载商城用户聚类分析

6.5.1 聚类概述

- 本质：识别出内在的规则，按照这些规则把对象分成若干类。
- 典型的要求:
  - 可伸缩性
  - 处理不同属性类型的能力
  - 发现任意形状聚类的能力
  - 用于确定输入参数的领域知识的要求
  - 处理噪声数据的能力
  - 结果对数据输入顺序的无关性
  - 处理高维数据的能力
  - 增加限制条件后的聚类分析能力
  - 聚类结果的可解释性和可用性

6.5.2 聚类分析方法

聚类算法的任务是将相似的事物分成一类，**不依赖于事先确定好的组别**，属于**无监督学习**。聚类算法根据设计思想的不同主要有如下几种。

- 划分法(partitioning methods)
  - 给定一个有N个元组或者纪录的数据集，划分法将构造K个分组，每一个分组就代表一个聚类，K<N
  - 对于给定的K，算法首先给出一个初始的分组方法，以后通过反复迭代的方法改变分组，使得每一次改进之后的分组方案都较前一次好
    - **同一分组中的记录越近越好，而不同分组中的纪录越远越好**
  - 使用划分法思想的算法有：**K-MEANS、K-MEDOIDS、CLARANS**
- 密度法(density-based methods)
  - 密度法
    - 它不是基于各种各样的距离的，而是**基于密度**的
    - 克服基于距离的算法只能发现“类圆形”的聚类的缺点
  - 指导思想
    - 只要一个点所属区域的密度大过某个阈值，就把它加到与之相近的聚类中去
  - 代表算法有：**DBSCAN、OPTICS、DENCLUE**等
- 层次法(hierarchical methods)
  - 将数据转换为**树形结构**，实现不同层次上的聚类
- 网格法（Grid-based methods)
  - 基于网格的方法把对象空间划分为有限个单元，形成一个网格结构
  - 代表算法有： **STING算法、CLIQUE算法、WAVE-CLUSTER**算法等

6.5.3 典型聚类算法介绍 — K-means算法

- 也叫K均值算法，是一种无监督学习算法，广泛应用于机器学习领域。
  - 选取K个对象代表类中心，将N个对象分到K个类中，使得类内具有较高的相似度，而类间具有较低的相似度
  - 同时使类内对象到聚类中心的距离之和最小化，每个类的聚类中心是通过计算各个类内数据的平均值得来的
    - 对象之间的相似程度是采用某个距离函数作为度量方法，对象之间的距离越小，它们就越相似，反之亦然
- k-Means 算法流程： ★
  - 首先从n个数据对象任意选择 k 个对象作为初始聚类中心
  - 而对于所剩下其它对象，则根据它们与这些聚类中心的相似度（距离），分别将它们分配给与其最相似的（聚类中心所代表的）聚类
  - 然后再计算每个所获新聚类的聚类中心（该聚类中所有对象的均值）
  - 不断重复这一过程直到标准测度函数开始收敛为止
  - 一般都采用均方差作为标准测度函数

K-means算法举例

假定一个数据集**{1, 2, 30, 15, 10, 18, 3, 9, 8, 25}**，

用**K-means聚类算法**将这些数据进行聚类。

第一步：

首先给定K = 3，即将数据集聚成3类。

随机选取后面3个数据对象作为初始类的中心点，分别是：

$m_1 = 8$，$m_2 = 9$，$m_3 = 25$。

K-means算法举例

{1, 2, 30, 15, 10, 18, 3, 9, 8, 25}

第二步：

第一次迭代

分别计算数据集中每个对象到这3个类中心点的距离，并将其分配给距离最近的中心点所代表的类。

采用距离值为两个数的差的绝对值。

计算中心点与数据对象差的绝对值

初始中心点	$m_1 = 8$	$m_2 = 9$	$m_3 = 25$
1	**7**	8	24
2	**6**	7	23
30	22	21	**5**
15	7	**6**	10
10	2	**1**	15
18	10	9	**7**
3	**5**	6	22
9	1	**0**	16
8	**0**	1	17
25	15	16	**0**

第一次迭代结果： $K_1=\{1, 2, 3, 8\}$ $K_2=\{9, 10, 15\}$ $K_3=\{18, 25, 30\}$

K-means算法举例

{1, 2, 30, 15, 10, 18, 3, 9, 8, 25}

第三步：

K_1={1, 2, 3, 8}　　K_2={9, 10, 15}　　K_3={18, 25, 30}

重新计算每个类的均值。中心点更新为：

$m_1 = 3.5$，$m_2 = 11.3$，$m_3 = 24.3$

计算中心点与数据对象差的绝对值

中心点	$m_1 = 3.5$	$m_2 = 11.3$	$m_3 = 24.3$
1	**1.5**	10.3	23.3
2	**1.5**	9.3	22.3
30	26.5	18.7	**5.7**
15	11.5	**3.7**	9.3
10	6.5	**1.3**	14.3
18	14.5	6.7	**6.3**
3	**0.5**	8.3	21.3
9	5.5	**2.3**	15.3
8	4.5	**3.3**	16.3
25	21.5	13.7	**0.7**

第二次迭代结果： $K_1=\{1, 2, 3\}$ $K_2=\{8, 9, 10, 15\}$ $K_3=\{18, 25, 30\}$

K-means算法举例

{1, 2, 30, 15, 10, 18, 3, 9, 8, 25}

第四步：

第二次迭代：

K_1={1, 2, 3}　　K_2={8，9, 10, 15}　　K_3={18, 25, 30}

重新计算每个类的均值。更新后的中心点为：

$m_1 = 2$，$m_2 = 10.5$，$m_3 = 24.3$

K-means算法举例

{1, 2, 30, 15, 10, 18, 3, 9, 8, 25}

第三次迭代：

重复第一次迭代中的方法，得到新的3个类为：

K_1={1, 2, 3}， K_2={8, 9, 10, 15}， K_3={18, 25, 30}

比较第三次与第二次的迭代结果，发现：

每个类中的数据不再被重新分配，聚类结果稳定，则算法终止。

6.5.3 典型聚类算法介绍 — K-means算法

例:给定新闻标题文本集D，如下表所示，用K-means聚类算法将这些文本数据进行聚类（文本分词结果直接给出）。

编号	文本内容
t_1	世界杯亚洲足球进步快 中国足球差距被拉大
	世界杯 亚洲 足球 进步 快 中国 足球 差距 拉大
t_2	特朗普经济团队素质太低，中美贸易战美国必败无疑
	特朗普 经济 团队 素质 太低 中美 贸易战 美国 必败 无疑
t_3	没有国足的俄罗斯世界杯:10万中国游客贡献30亿元
	没有 国足 俄罗斯 世界杯 10万 中国 游客 贡献 30亿元
t_4	中美贸易战再度升级 华为宣布退出美国市场
	中美 贸易战 再度 升级 华为 宣布 退出 美国 市场
t_5	美国记者亲身体验中日韩俄高铁 中国不仅舒服速度还快
	美国 记者 亲身 体验 中日韩俄 高铁 中国 不仅 舒服 速度 还快
t_6	足球速报！东南亚劲旅爆冷胜世界杯球队，国足再迎一强敌
	足球 速报 东南亚 劲旅 爆冷 胜 世界杯 球队 国足 再迎 强敌
t_7	美媒体、议员担忧中美贸易战将使美国受损
	美 媒体 议员 担忧 中美 贸易战 美国 受损
t_8	中国高铁递出“国家名片”中国创新让世界共享
	中国 高铁 递出 国家名片 创新 世界 共享
t_9	从高铁速度感受中国发展强劲脉搏
	高铁 速度 感受 中国 发展 强劲 脉搏
t_{10}	特朗普贸易战背后:扼杀中国科技力量的崛起
	特朗普 贸易战 背后 扼杀 中国 科技 力量 崛起

6.5.3 典型聚类算法介绍 — K-means算法

解：从直观上看，上述新闻标题属于三大类：体育、经济、交通。但是从机器学习的角度分析，需要通过以下步骤进行聚类才可能将相似类别的新闻聚集在一起。

第一步：设定聚类个数以及选取中心点。

设定k=3，即将新闻标题文本集D聚成3类。在迭代计算之前，需要假定初始状态下三个聚类的中心点，通常是随机选择数据集中的k个样本（本题中即为文本）作为初始类中心。根据实际经验，应当选择彼此距离较远的三个点，有助于减少迭代次数。

第二步：对中心点进行调整，并不断地迭代计算，直至满足终止条件。

选择三个初始的类中心后，将其它文本分别与三个初始类中心计算相似度。例如，第m个文本，分别计算其与聚类k_1，k_2，k_3的类中心的相似度，分别是s_{m1}，s_{m2}，s_{m3}，若s_{m1}值最大，则说明在本次迭代中文本m属于聚类k_1，依次类推。完成一次迭代之后，需要重新确定聚类k_1，k_2，k_3的聚类中心。确定聚类中心文本的方式是计算该聚类中每个文本与其他所有文本相似度的平均值，取平均值较大的文本为该聚类簇的中心文本。确定类中心文本后，再次计算每个文本与类中心的相似度，将每个文本划分到对应的聚类簇中。迭代此过程，直至满足终止条件。

第三步：通过K-means不断迭代新闻标题数据集，最终聚类结果如下表

6.5.3 典型聚类算法介绍 — K-means算法

聚类	文本编号	文本内容
聚类k_1	t_1	世界杯亚洲足球进步快 中国足球差距被拉大
	t_3	没有国足的俄罗斯世界杯:10万中国游客贡献30亿元
	t_6	足球速报！东南亚劲旅爆冷胜世界杯球队，国足再迎一强敌
聚类k_2	t_2	特朗普经济团队素质太低，中美贸易战美国必败无疑
	t_4	中美贸易战再度升级 华为宣布退出美国市场
	t_7	美媒体、议员担忧中美贸易战将使美国受损
	t_{10}	特朗普贸易战背后:扼杀中国科技力量的崛起
聚类k_3	t_5	美国记者亲身体验中日韩俄高铁 中国不仅舒服速度还快
	t_8	中国高铁递出“国家名片”中国创新让世界共享
	t_9	从高铁速度感受中国发展强劲脉搏

6.5.3 典型聚类算法介绍 — K-means算法

- K-means算法总结
  - 优点：
    - 算法简单、易于理解、实现方便
    - 内存使用率低输出、结果易于理解
  - 缺点：
    - 初始聚类中心和聚类数随机选取的盲目性在很大程度上降低了算法的有效性
    - 当数据有噪声和孤立点时，算法变得不稳定

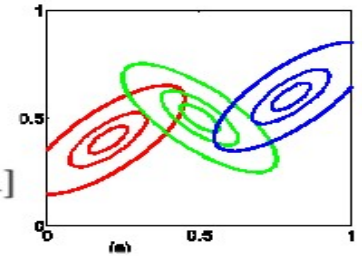
6.5.3 典型聚类算法介绍 — K-Medoids算法

- K-MEDOIDS
  - 对K-Means改进：选取一个对象叫做mediod来代替上面的中心的作用，这样的一个medoid就标识了这个类
    - 在K-means中，中心点取为当前cluster中所有数据点的平均值
    - 在 K-medoids算法中，从当前cluster 中选取这样一个点——它到其他所有（当前cluster中的）点的距离之和最小——作为中心点

6.5.3 典型聚类算法介绍 — 高斯混合模型GMM

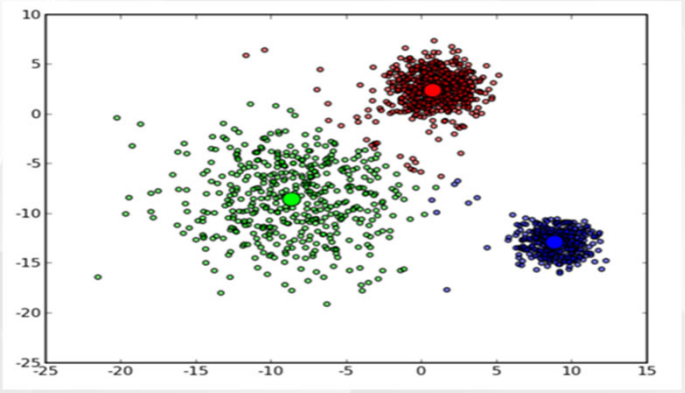
- GMM
  - 将k个高斯模型混合在一起，每个点出现的概率是几个高斯混合的结果
  - 假设有K个高斯分布，每个高斯对数据点的影响因子为πk，数据点为x，高斯参数为theta

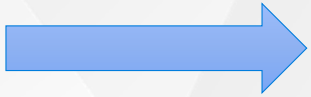
$$\begin{cases} p(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x};\boldsymbol{\theta}_k) \\ \boldsymbol{\Theta} = \{\pi_1,\cdots,\pi_K,\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_K\}, \sum_{k=1}^{k} \pi_k = 1, \pi_k \in [0,1] \\ p_k(\boldsymbol{x};\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) \end{cases}$$

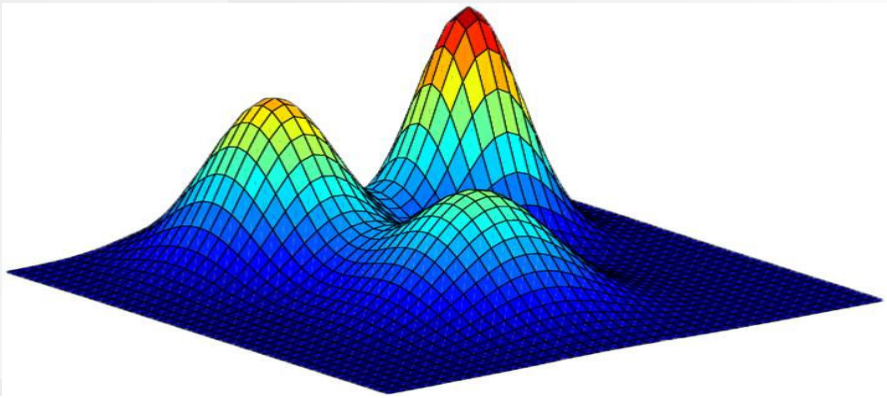


— Parameters to be estimated: $\pi_k, \boldsymbol{\mu}_k, \Sigma_k$

  - EM（expectation maximization）算法对产生样本数据的最大似然实现各个高斯模型估计
  - GMM可以给出每个数据属于各个类型的概率，可以对non-spherical数据进行聚类
  - 缺点是计算量大，不保证全局最优

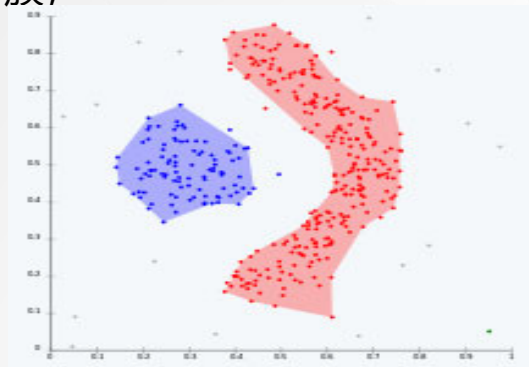






6.5.3 典型聚类算法介绍 — DBSCAN

- DBSCAN(Density-Based Spatial Clustering of Applications with Noise)
  - 几个定义：
    - E领域：给定对象半径为E内的区域称为该对象的E领域；
    - 核心对象：如果给定对象E领域内的样本点数大于等于MinPts，则称该对象为核心对象；
    - 直接密度可达：对于样本集合D，如果样本点q在p的E领域内，并且p为核心对象，那么对象q从对象p直接密度可达。
    - 密度可达：对于样本集合D，给定一串样本点p1,p2....pn，p= p1,q= pn,假如对象pi从pi-1直接密度可达，那么对象q从对象p密度可达。
    - 密度相连：存在样本集合D中的一点o，如果对象o到对象p和对象q都是密度可达的，那么p和q密度相联
  - DBSCAN目的是找到密度相连对象的最大集合
  - DBSCAN算法描述:
    - 输入: 包含n个对象的数据库，半径e，最少数目MinPts;
    - 输出:所有生成的簇，达到密度要求。
    - (1)Repeat
      - (2)从数据库中抽出一个未处理的点；
      - (3)IF抽出的点是核心点 THEN 找出所有从该点密度可达的对象，形成一个簇；
      - (4)ELSE 抽出的点是边缘点(非核心对象)，跳出本次循环，寻找下一个点；
    - (5)UNTIL 所有的点都被处理
  - 优点：
    - 不需要事先知道要形成的簇类的数量
    - 可以发现任意形状的簇类
    - 能够识别出噪声点
  - 缺点：不能适应变化的密度



基于聚类的应用——用户价值评估

数据来源和聚类模型

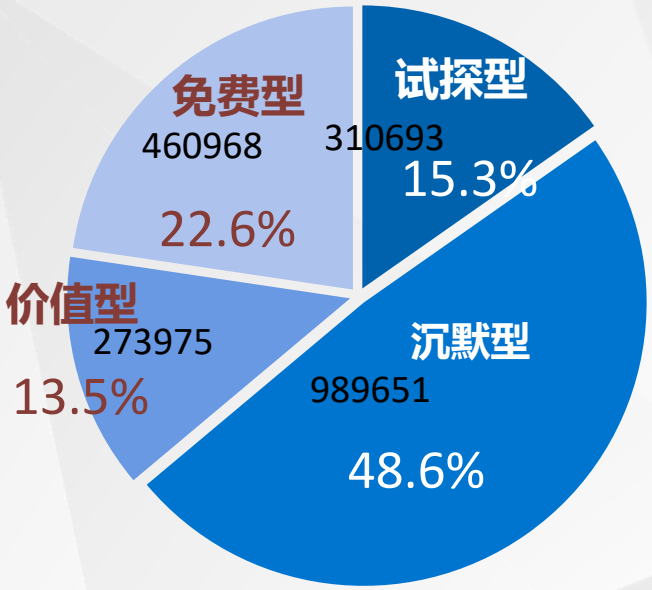
- **数据：**某文献下载商城某月活跃用户信息及相关下载记录
- **聚类维度：**选择最新下载的时间间隔(R)、下载的频率(F)以及费用(M)进行聚类
- 使用MongoDB统计每个用户对应的3个聚类属性
- 使用Mahout的Canopy+K-means聚类模型在Hadoop平台计算聚类结果

用户信息	应用信息	下载记录
• 约204万条，**73.6MB**	• 约12.7万条，14.5MB	• 约**2000万条**，**1.3GB**

聚类结果和分析

- 聚类的结果将204万用户分为4类，如图所示

用户类别	下载次数	下载类别	付费总额	占比	特征
沉默型	0.92	0.86	0.06	48.6%	下载不活跃，几乎不贡献收入
试探型	1.42	1.36	8.81	15.3%	试探性下载，有付费意愿
免费型	4.8	4	0.36	22.6%	下载比较活跃，但基本下载免费应用
价值型	10.1	9.7	103.5	13.5%	下载次数种类都很多，付费意愿强烈



算法性能

- 在3节点的Hadoop平台上完成204万用户的聚类计算只需要**10分钟**左右

Mahout算法库

Mahout 是 Apache Software Foundation（ASF） 旗下的一个开源项目，提供大量的可扩展的机器学习领域经典算法的实现。
由于Mahout是Hadoop 生态系统的标准算法库，二者结合紧密，Mahout算法可以有效地应用到Hadoop集群。

提供三方面的可扩展性

面向大数据的可扩展性

Mahout中的核心算法如聚类、分类、协同过滤等架构在Hadoop框架下，部分算法实现了细粒度的并行化

面向商业的可扩展性

开发人员在 Apache 许可下免费使用。
也可以通过获得商业许可，用于商用

面向开发社区的可扩展性

Mahout的宗旨是构建一个活跃的、快速响应的、多元性的开发者社区，讨论项目本身及应用

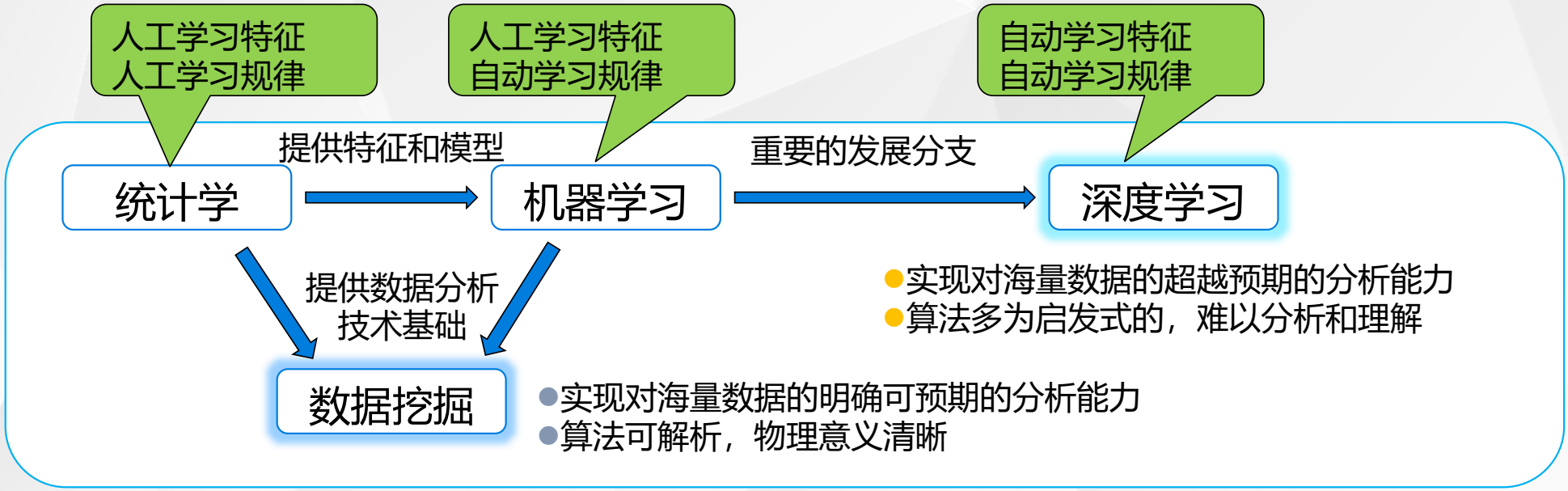
已实现基于Hadoop的数据挖掘/机器学习算法

- **Classification**：分类算法
- **Clustering**：聚类算法
- **Pattern Mining**：模式挖掘
- **Regression**：回归算法
- **Dimension reduction**：降维算法
- **Evolutionary Algorithms**：进化算法
- **Collaborative Filtering**：协同过滤算法
- **Vector Similarity**：向量相似度计算

6.6 机器学习 VS 大数据

如果数据是21世纪最宝贵的财富，大数据分析**就是当今最伟大的炼金术，可以从前所未有的大规模数据中发现前所未知的知识，实现不可限量的价值。**

大数据分析依赖的理论和方法主要包括传统的统计学、机器学习、数据挖掘，以及近10年来逐渐发展成熟的深度学习。

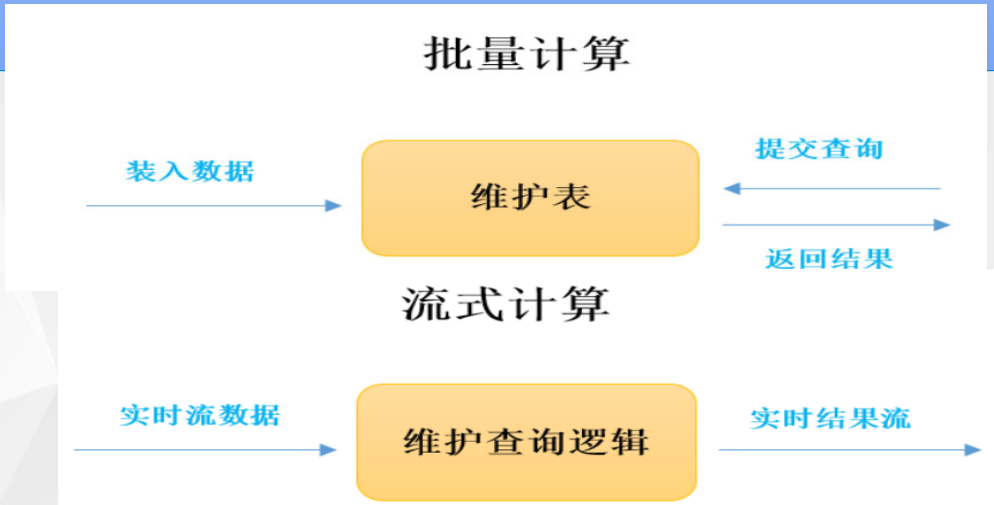


实现基础：数据库、数据仓库、分布式存储、并行计算、流式计算、GPU加速……

批量计算 VS 流式计算

收集数据 - 放到DB中 - 取出来分析 的传统的流程，叫做批量计算，顾名思义，将数据存起来，批量进行计算。

而流式计算，也跟名字一样，是对数据流进行实时计算，它不是更快的批计算，可以说，是完全不同的处理思路。



(1) 与批量计算那样慢慢积累数据不同，流式计算**将大量数据平摊到每个时间点**上，**连续地进行小批量的进行传输**，**数据持续流动**，**计算完之后就丢弃**。

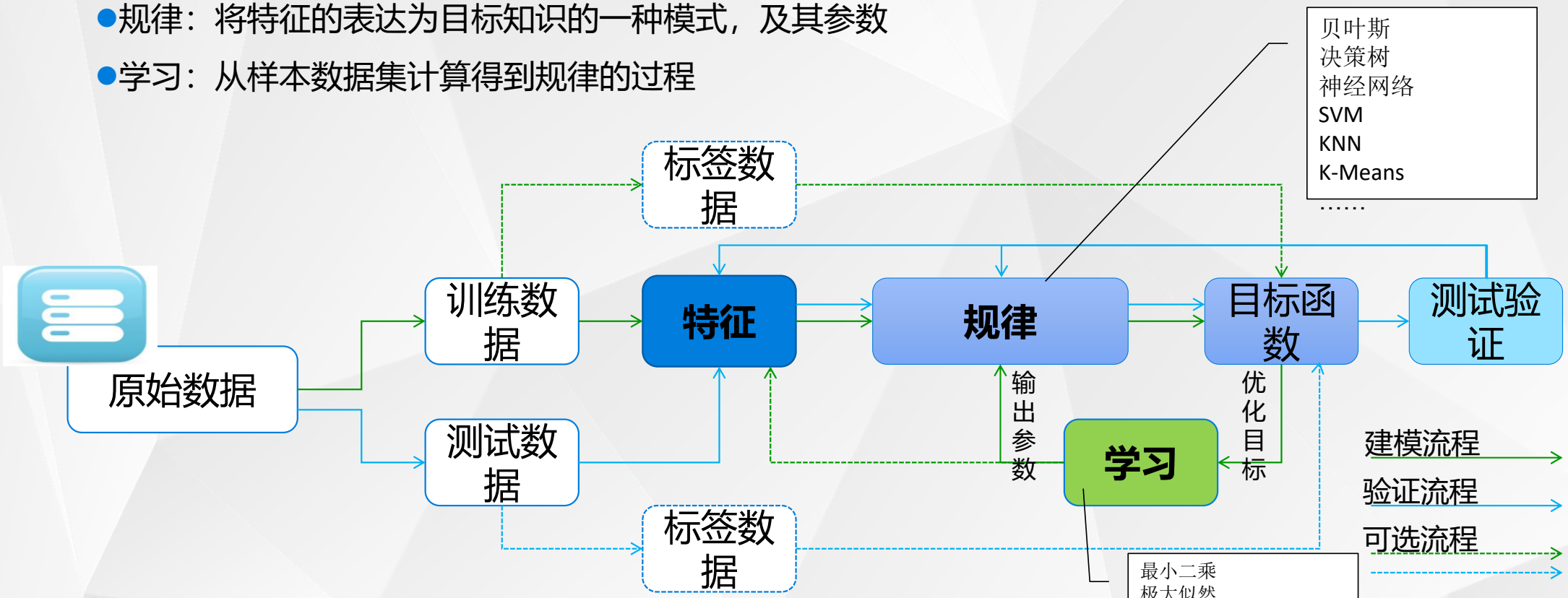
(2) 批量计算是维护一张表，**对表进行实施各种计算逻辑**。流式计算相反，是必须**先定义好计算逻辑，提交到流式计算系**统，这个计算作业逻辑在整个运行期间是不可更改的。

(3) 计算结果上，批量计算**对全部数据进行计算后传输结果**，流式计算是**每次小批量计算后，结果可以立刻投递到在线系统**，做到实时化展现。

关于特征、规律和学习

大数据建模分析的本质是通过构建数学模型，从数据中学习**特征和规律**，收获有用的**知识。**

- 特征：决定数据对象所蕴含的知识的关键属性
- 规律：将特征的表达为目标知识的一种模式，及其参数
- 学习：从样本数据集计算得到规律的过程



大数据建模分析的基本流程

大数据中的机器学习

学习能力是大数据分析建模的关键技术之一。根据反馈的不同，大数据中的学习技术可以分为监督学习（Supervised learning）、非监督学习（Unsupervised learning）、半监督学习（Semi-supervised learning）和强化学习（Reinforcement learning）四大类。

监督学习（Supervised learning）

- 使用有标签数据进行学习
- 典型场景：分类、回归

非监督学习（Unsupervised learning）

- 使用无标签数据进行学习
- 典型场景：聚类

半监督学习（Semi-supervised learning）

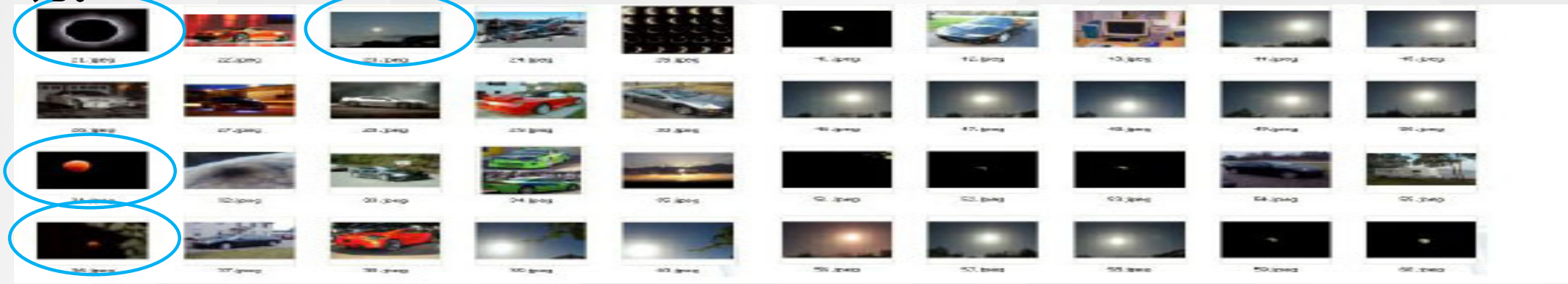
- 使用数据的一部分是有标签的，另一部分没有标签，无标签数据的数量>>有标签数据数量
- 典型场景：海量数据分类

强化学习（Reinforcement learning）

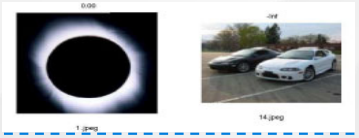
- 使用无标签但有反馈的数据进行学习
- 典型场景：策略推理

半监督（Semi-supervised) 学习案例

图片分类案例：从图库中识别出“日蚀”图片。当图库巨大时，人工标注耗时耗力。



步骤一: 用带有标识的图片训练分类器



步骤二: 对没有标识的数据进行分类，并按照信任度从大到小进行排序

步骤三: 将信任度最高的图片自动加入标识项

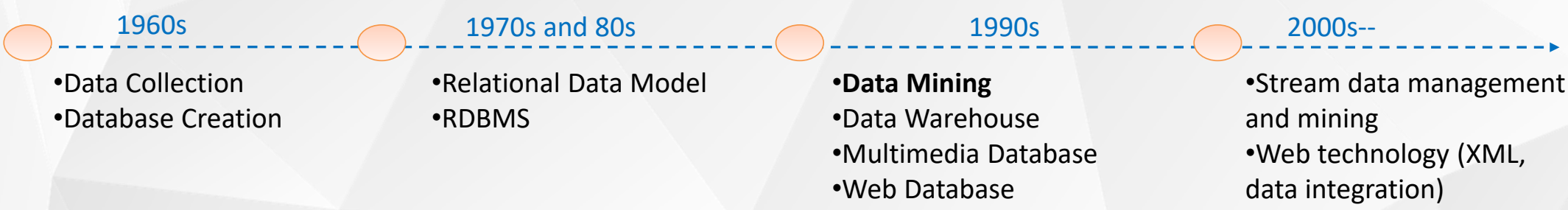
步骤四: 重新训练分类器并重复步骤二~步骤四

数据挖掘的概念和历史

数据挖掘（Data Mining）一词是在1989年8月召开的第十一届国际联合人工智能学术会议（JCAI’89）上正式形成的，其根源可追溯到**经典统计学、人工智能、机器学习**三个学科，关系型数据库、互联网的广泛应用两次推动了数据挖掘技术的发展。

数据挖掘定义的发展

- SAS研究所（1997）：“在大量相关数据基础之上进行数据探索和建立相关模型的先进方法”
- Bhavani（1999）：“使用模式识别技术、统计和数学技术，在大量的数据中发现有意义的新关系、模式和趋势的过程”
- Hand et al（2000）：“数据挖掘就是在大型数据库中寻找有意义、有价值信息的过程”
- **Jiawei Han(韩家炜)（2000）：从海量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中、人们事先不知道的、但又潜在有用的信息和知识的过程** --《数据挖掘:概念与技术》



主要的数据挖掘算法

随着数据挖掘应用多年来不断的扩展和深化，产生积累了大量的数据挖掘算法。根据应用场景及目标的不同，可以将数据挖掘算法分为如下几类。

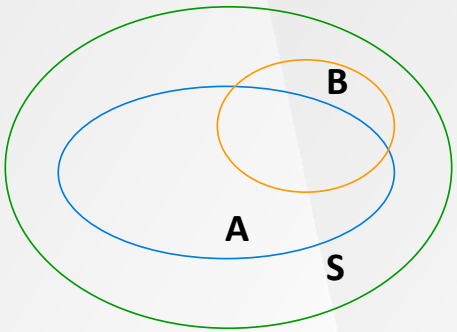
广义型知识	特征型知识	差异型知识	关联型知识	预测型知识	偏离型知识
• 反映同类事物共同性质的知识	• 反映事物各方面的特征知识	• 反映不同事物之间属性差别的知识	• 反映事物之间依赖或关联的知识	• 根据历史和当前数据推测未来数据	• 揭示事物偏离常规的异常现象

分类	聚类	关联规则	时序预测	异常检测
• 按照分析对象的属性、特征，建立不同的组类来描述事物	• 识别出内在的规则，按照这些规则把对象分成若干类	• 关联是某种事物发生时其他事物会发生的这样一种联系	• 把握分析对象发展的规律，对未来的趋势做出预见	• 对分析对象的少数极端的特例的描述，揭示内在的原因

关联规则挖掘算法

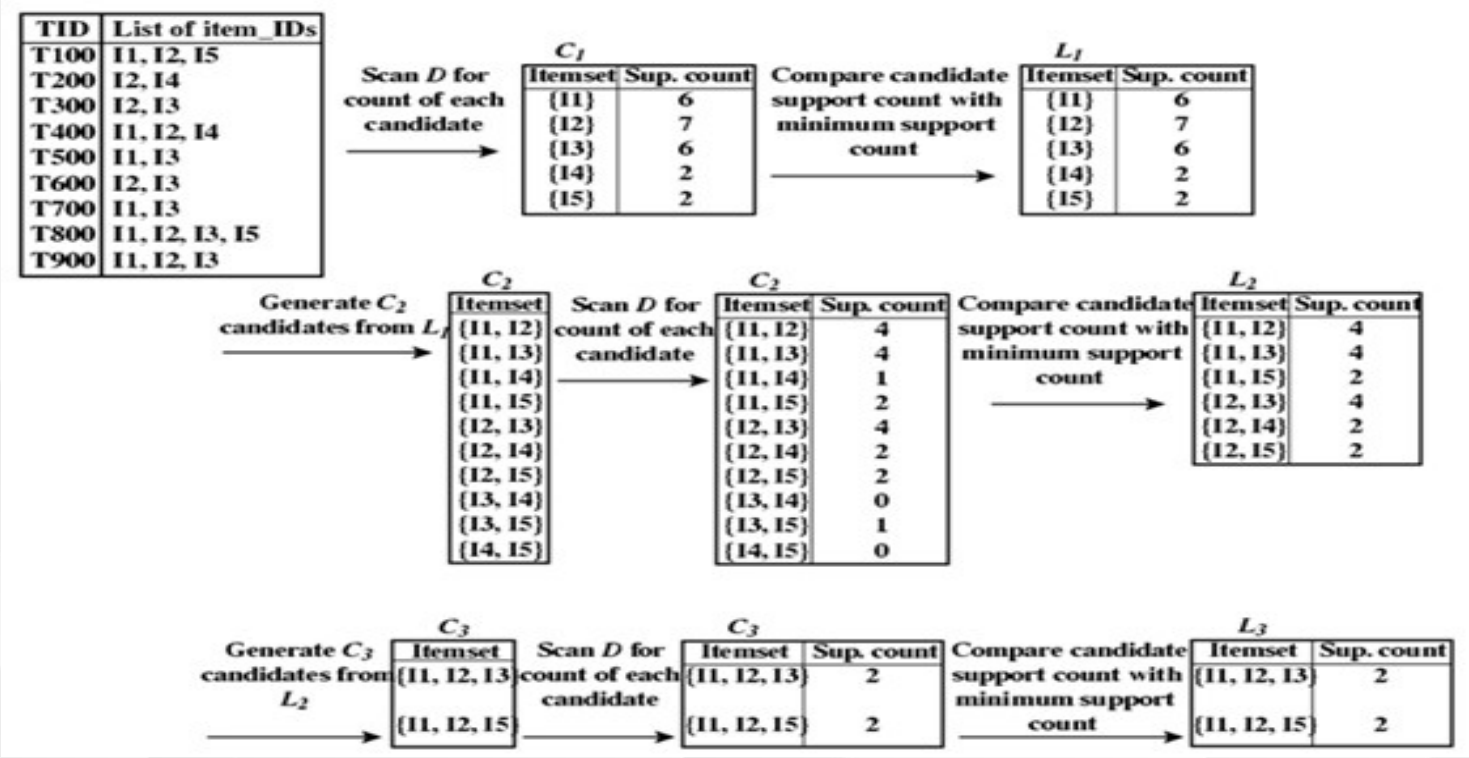
关联规则挖掘的目标是发现**有意义的**事物同时出现的规律。关联规则挖掘算法属于无监督学习的方法。最常用的关联规则算法有Apriori和FP-Growth两种。

- 有关关联关系的关键概念：
  - S-事件总集，A-出现A的事件集，B-出现B的事件集，AB-同时出现A和B的事件集
  - **A-B可信度**=|AB|/|A|
  - **支持度**=|AB|/|S|
  - **期望可信度**=|B|/|S|
  - **A-B作用度**=A-B可信度/B期望可信度
  - 可信度是对关联规则的准确度的衡量，支持度是对关联规则重要性的衡量，作用度描述了A 对B 的影响力
  - 为了发现有意义的关联规则，一般需要给定两个阈值：最小支持度和最小可信度



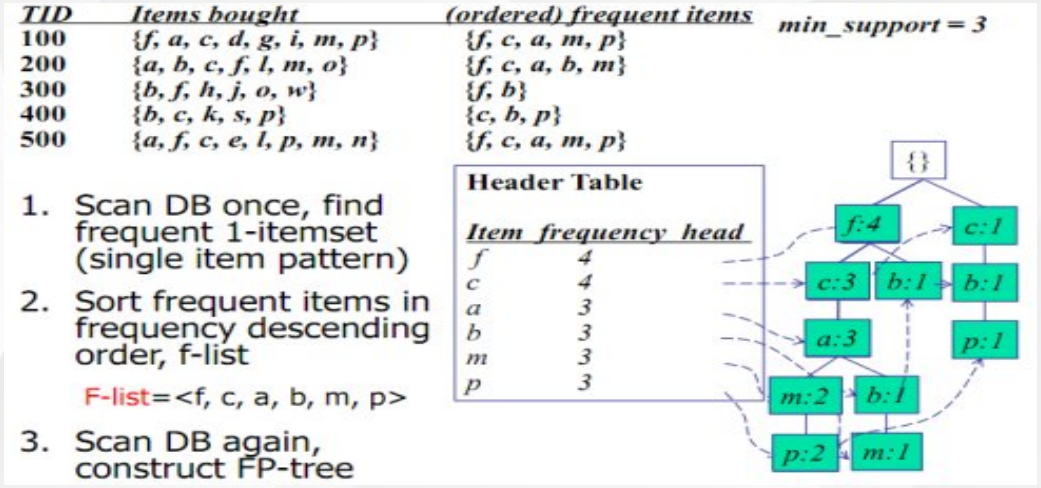
关联规则挖掘算法

- Apriori
  - 算法从少到多生成频繁项，根据前一次找到的频繁项来生成本次的频繁项，提升频繁项产生效率
  - 缺点：
    - 产生大量的候选集
    - 重复扫描数据库

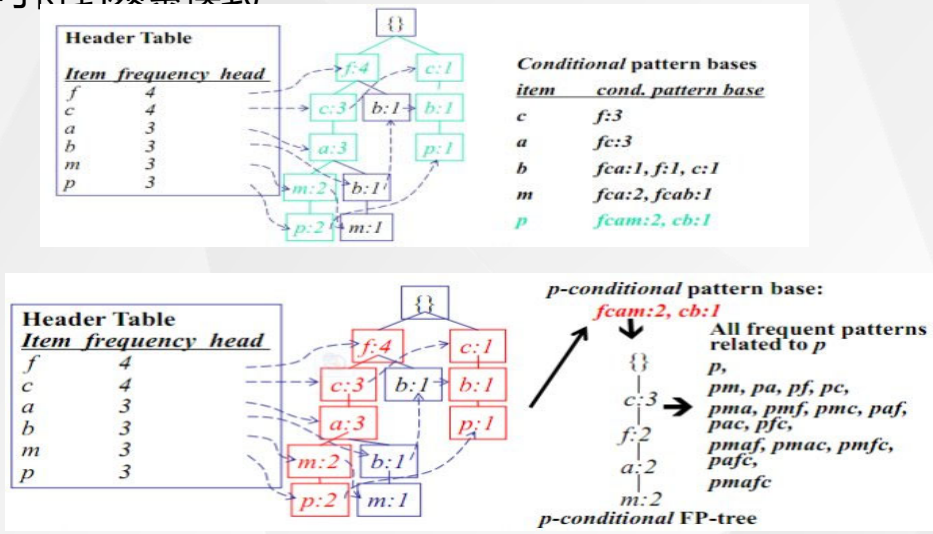


关联规则挖掘算法

- FP-Growth
  - J. Han等2000年提出
  - 不产生候选频繁集
  - 只需要两次遍历数据库
- FP-Growth算法流程：
  - 基本思路：不断地迭代FP-tree的构造和投影过程。
  - 对于每个频繁项，构造它的条件投影数据库和投影FP-tree。对每个新构建的FP-tree重复这个过程，直到构造的新FP-tree为空，或者只包含一条路径。当构造的FP-tree为空时，其前缀即为频繁模式；当只包含一条路径时，通过枚举所有可能组合并与此树的前缀连接即可得到频繁模式.



FP-Tree的生成



FP-Tree的分割

关联规则挖掘算法应用

针对某网站2个月的日志信息，采用Mahout PFPG关联规则挖掘算法进行模式挖掘，算法配置参数为：最小支持度600，最小置信度0.7，挖掘的模式数为15，具体结果如下：

模式挖掘结果：

_id	value
0060501	([1579614, 0060501],94), ([2490256, 0060501],94), ([1697685, 0060501],94), ([1579614, 2490256, 0060501],93), ([1579614, 1
0065440	([2522098, 0065440],85), ([2749892, 0065440],84), ([1579614, 0065440],84), ([2522098, 2749892, 0065440],83), ([2522098, 1
0066117	([1095907, 0066117],93), ([1134210, 0066117],93), ([0050370, 0066117],92), ([0001713, 0066117],92), ([2578686, 0066117],9
0066212	([2671454, 0066212],97), ([2671376, 0066212],97), ([2784248, 0066212],97), ([2617515, 0066212],97), ([2654776, 0066212],97), ([2671454, 2671376, 0066212],96), ([2671454, 2784248, 0066212],96), ([2671454, 26175...
0066213	([2671454, 0066213],94), ([2671376, 0066213],94), ([2784248, 0066213],94), ([2617515, 0066213],94), ([2654776, 0066213],94), ([2671454, 2671376, 0066213],94), ([2671454, 2784248, 0066213],94), ([2671454, 26175...
1093863	([2824620, 2825887],70), ([2824646, 2825887],69), ([2824646, 2824620, 2825887],68), ([2824646, 2826106, 2825887],67), ([2
1095734	([2734449, 1095734],82), ([0010457, 1095734],81), ([2522098, 1095734],81), ([0076723, 1095734],7
1095967	([0068902, 1095907],76), ([0007840, 1095907],70), ([0007840, 0068902, 1095907],66)
1096178	([0673864, 1096178],96), ([1407178, 1096178],94), ([0673864, 1407178, 1096178],92)
1096627	([2575008, 1096627],79), ([2431359, 1096627],79), ([2575008, 2431359, 1096627],78)
2825861	([2824590, 2825861],65), ([2825861, 2825869],64), ([2824590, 2825861, 2826096],62), ([2824590, 2825861, 2824691],61)
2825869	([2824646, 2825869],70), ([2824620, 2825869],70), ([2826118, 2825869],69), ([2826075, 2825869],69), ([2825861, 2825869],6
2825870	([2824590, 2825870],62), ([2824568, 2825870],61), ([2825870, 2825869],60), ([2824568, 2824590, 2825870],60), ([2825870, 2826129],60)
2825887	([2824620, 2825887],70), ([2824646, 2825887],69), ([2824646, 2824620, 2825887],68), ([2824646, 2826106, 2825887],67), ([2826106, 2824620, 2825887],67), ([2824646, 2826106, 2824620, 2825887],67), ([2826080, 28...

应用ID 置信度值%

应用模式	置信度
3G门户新闻，广州日报	0.94
重庆晨报移动版，广州日报	0.94
财经网，广州日报	0.94

应用模式	置信度
短信伴侣，灵犀语义助手	0.82
墨迹天气，短信伴侣	0.81
短信伴侣，360省电王	0.81
僵尸前线，短信伴侣	0.81
短信伴侣，淘宝	0.79

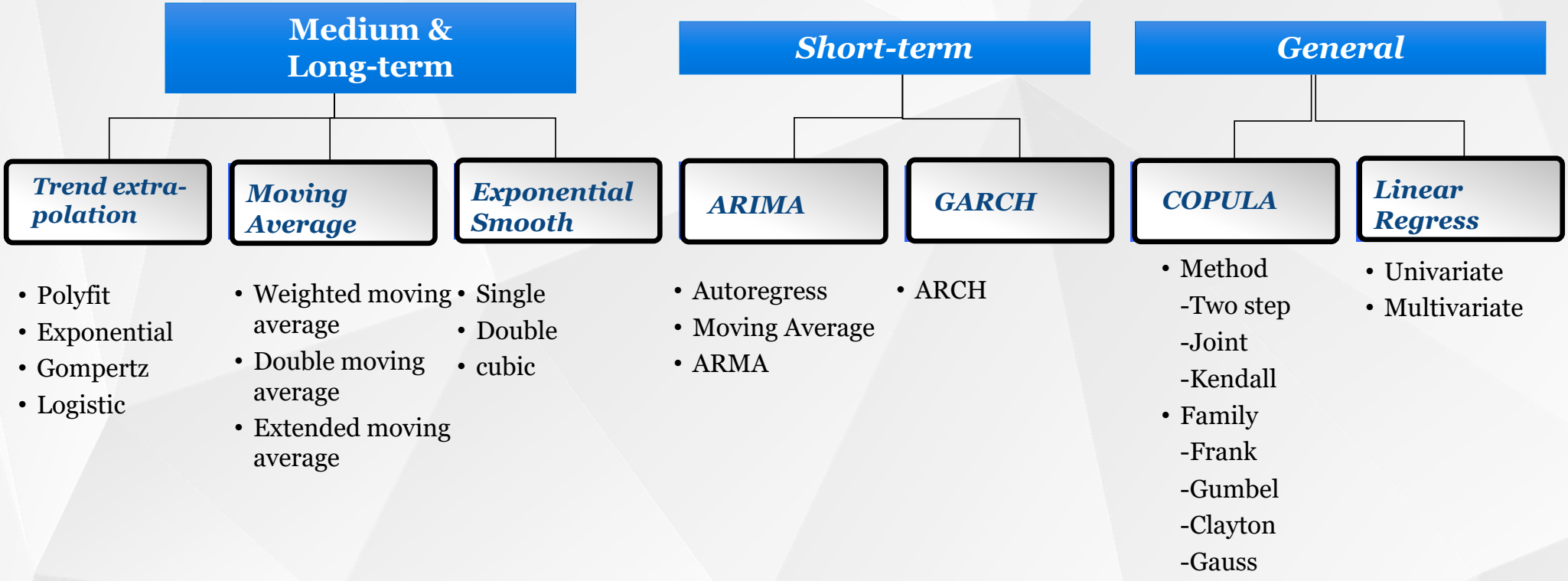
异常检测算法

在数据库中包含着少数的数据对象，它们与数据的一般行为或特征不一致，这些数据对象叫做异常点 (Outlier) ，也叫做孤立点。异常点的检测和分析是一种十分重要的数据挖掘类型，被称之为异常点挖掘。

- 基于统计的异常点检测算法
  - 通常用某个统计分布对数据点进行建模，再以假定的模型判断点的分布是否存在异常
  - 单样本多个离群检测算法 ESD、GESR
  - 缺点：模型拟合不准确、难以解释异常点
- 基于距离的异常点检测算法
  - 离群点被定义为数据集中与大多数点之间的距离都大于某个阈值的点，通常被描述为 DB (pct，d_min),数据集 T中一个记录 O称为离群点，当且仅当数据集T中至少有 pct部分的数据与 O 的距离大于 d_min
  - 孤立点是数据集中到第 k 个最近邻居的距离最大的 n 个对象
  - 孤立点是数据集中与其 k 个最近邻居的平均距离最大的 n个对象
  - 优点：直观、计算简单
  - 缺点：参数难以确定、高维数据效果差
- 基于密度的异常点检测算法
  - 将记录之间的距离和某一给定范围内记录数这两个参数结合起来，从而得到“密度”的概念，然后根据密度判定记录是否为离群点
  - 优点：可以发现局部异常

时间序列预测算法

时间序列预测是根据过去的变化趋势预测未来的发展，是数据挖掘的重要研究应用方向。



时间序列预测应用-运营收支预测

计算季节性指数

- 计算每期的中心移动平均值
- 采用中位数，作为概括性衡量标准
- 调整得到季节性指数

去除季节性影响

- 历史数据除以季节性指数作为预测模型的输入

预测

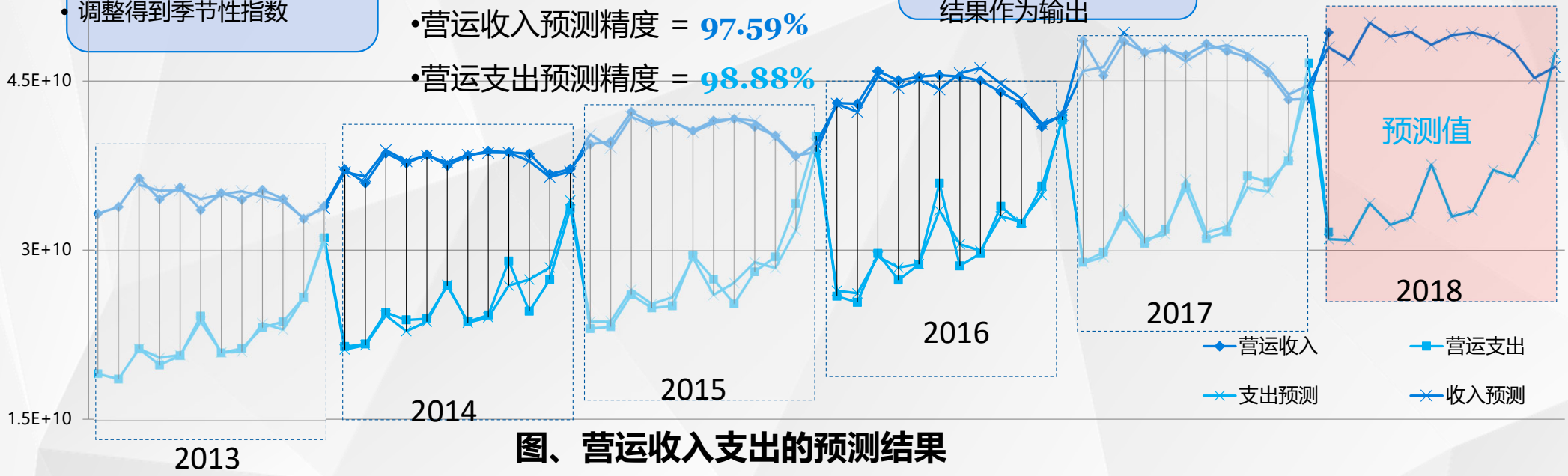
- 通过7类模型对输入数据进行预测
- 选择精度最高的模型结果作为输出

还原季节性影响

- 预测序列乘以季节性指数，还原季节性影响

•营运收入预测精度 = **97.59%**

•营运支出预测精度 = **98.88%**



图、营运收入支出的预测结果

注：预测以历史数据为依据，未考虑5G商用、宏观经济等突发事件的影响，如计算此因素对KPI的影响，可采用因素分解等模型；预测精度采用1-mean(abs((P_i-H_i)/H_i)衡量。

6.7 深度学习简介

深度学习的科普贴

- https://blog.csdn.net/qq_25424545/article/details/78867389

CSDN 博客 学院 下载 GitChat 论坛 问答 商城 VIP 活动 招聘 ITeye tinymind CSTO

T_tangcのblog

0 目录 收藏 评论 微信 微博 QQ

转 **最容易读进去的深度学习科普贴**

2017年12月21日 20:23:48 阅读数：279

注：自己写专业论文时候觉得很不错的一篇文章，分享在博客里

(一)

一

2016 年一月底，人工智能的研究领域，发生了两件大事。

先是一月二十四号，MIT 的教授，人工智能研究的先驱者，Marvin Minsky 去世，享年89 岁。

三天之后，谷歌在自然杂志上正式公开发表论文，宣布其以深度学习技术为基础的电脑程序 AlphaGo， 在 2015年 十月,连续五局击败欧洲冠军、职业二段樊辉。

这是第一次机器击败职业围棋选手。距离 97年IBM 电脑击败国际象棋世界冠军，一晃近二十年了。

极具讽刺意义的是，Minsky 教授，一直不看好深度学习的概念。他曾在 1969年 出版了 Perceptron (感知器) 一书，指出了神经网络技术(就是深度学习的前身) 的局限性。这本书直接导致了神经网络研究的将近二十年的长期低潮。

深度学习的背景

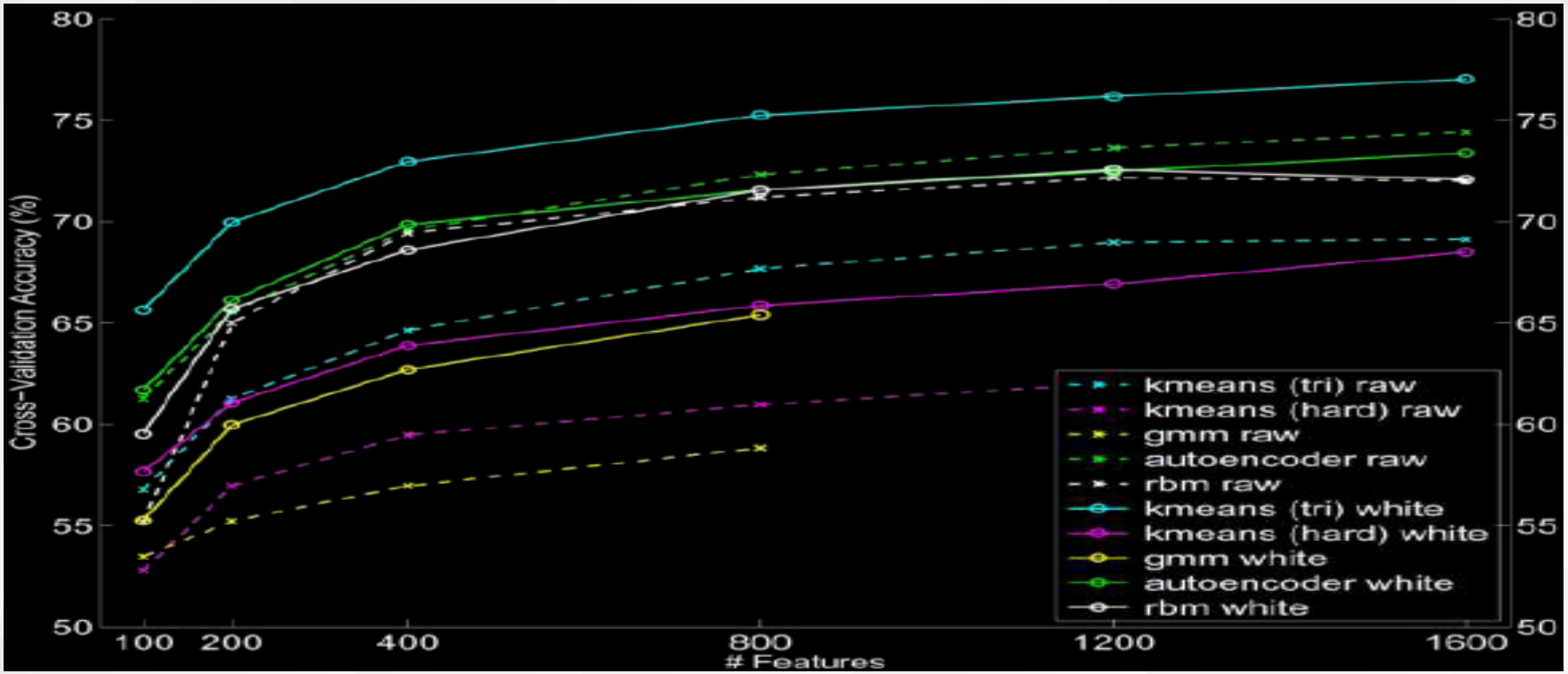
- 机器学习虽然发展了几十年，但还是存在很多没有良好解决的问题
- 例如图像识别、语音识别、自然语言理解、天气预测、基因表达、内容推荐等等。目前通过机器学习去解决这些问题的思路都是这样（以视觉感知为例子）

深度学习的背景

•从开始的通过传感器（例如CMOS）来获得数据。然后经过预处理、特征提取、特征选择，再到推理、预测或者识别。最后一个部分，也就是机器学习的部分，绝大部分的工作是在这方面做的，也存在很多的学术论文和研究。

•而中间最重要的部分，概括起来就是特征表达。良好的特征表达，对最终算法的准确性起了非常关键的作用，而且系统主要的计算和测试工作都耗在这一大部分。但是这块实际中一般都是人工完成的，即靠人工提取特征。

深度学习的背景



- 一般而言，特征越多，给出信息就越多，识别准确性会得到提升；
- 但特征多，计算复杂度增加，探索的空间大，可以用来训练的数据在每个特征上就会稀疏。
- 结论：不一定特征越多越好！需要有多少个特征，需要学习确定。

关于学习特征的总结

- 机器学习中，获得好的特征是识别成功的关键
- 目前存在大量人工设计的特征，不同研究对象特征不同，特征具有多样性，如：SIFT, HOG, LBP等
- 手工选取特征费时费力，需要启发式专业知识，很大程度上靠经验和运气
- 是否能自动地学习特征?

深度学习的概念

深度学习是一种特征学习方法，把原始数据通过一些简单的但是非线性的模型转变成为更高层次的，更加抽象的表达。深度学习的实质，是通过构建具有很多隐层的机器学习模型和海量的训练数据，来学习更有用的特征，从而最终提升分类或预测的准确性。**“深度模型”是手段，“特征学习”是目的。**

多层神经网络：

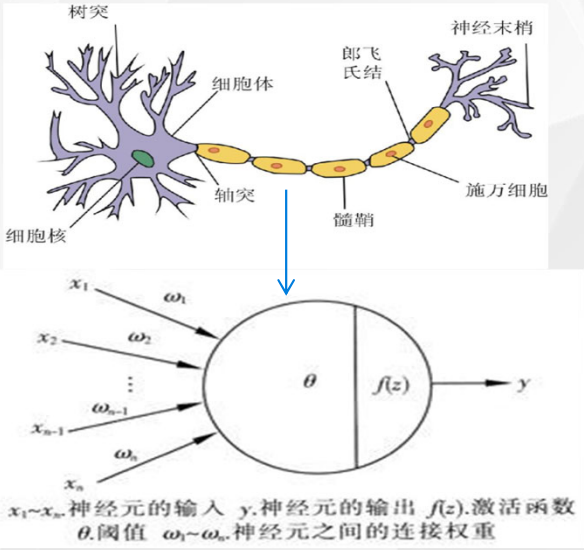
模型结构的深度，通常有5层、6层，甚至更多层

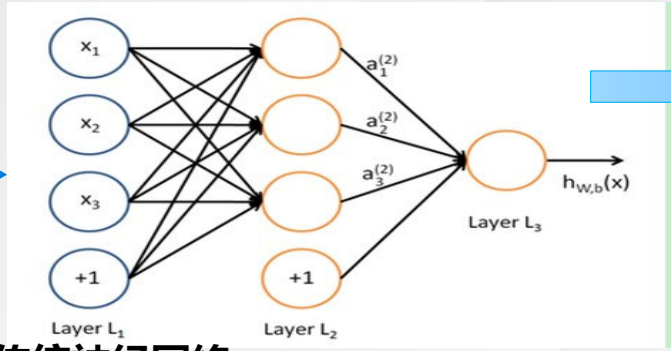
具有超强的泛化能力，非常复杂的函数也可以被学习，更利于发掘海量数据中蕴藏的丰富信息

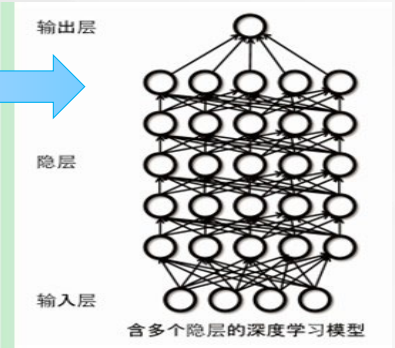
自动学习特征：

各层的特征**不是人工设计**的，而是**从数据本身中学到的**

同过逐层特征变换，将样本在原空间的特征表示变换到一个新特征空间，使分类或预测更加容易







传统神经网络：

- 一般只有两三层
- 使用BP算法训练
- 特征选取困难
- 比较容易过拟合，参数比较难调整
- 层次比较少的情况下效果并不比其它方法如SVM等更优

深度神经网络：

- 大于5层
- 更强大的拟合能力
- 有效的训练方法
- GPU/多核CPU
- 大规模数据集

- 深度学习起源于人工智能和机器学习的研究：
  - 从20世纪50年代到70年代初，人工智能研究处于“推理期”，目标是给机器赋予逻辑推理能力，但人们逐渐认识到，仅具有逻辑推理能力无法实现人工智能的，必须使机器拥有知识
  - 20世纪70年代中期开始，人工智能进入“知识期”，人们基于逻辑知识表示、通过领域知识获取来实现专家系统。但是，专家系统面临“知识工程瓶颈”，必须使机器自主学习知识
  - 20世纪80年代机器学习逐步成为人工智能研究领域的主流方向
- 1980年代末期，用于人工神经网络的反向传播算法（Back Propagation算法）的发明，掀起了机器学习热潮，各种机器学习算法在90年代大量涌现，其中包括卷积神经网络等深度学习的雏形
- 2000年以来随着互联网的高速发展，对大数据的智能化分析和预测产生了巨大需求，浅层学习模型在互联网应用上获得了巨大成功。最成功的应用包括搜索广告系统的广告点击率CTR预估、网页搜索排序、垃圾邮件过滤系统、基于内容的推荐系统等，**但是神经网络模型基本被冷落**

深度学习的起源和发展

- 2006年，加拿大多伦多大学教授Geoffrey Hinton在《科学》上发表论文提出：1. 具有很多隐层的人工神经网络具有优异的特征学习能力，学习得到的特征对数据有更本质的刻画，从而有利于可视化或分类；2. 深度神经网络在训练上的难度，可以通过“逐层初始化”（Layer-wise Pre-training）来有效克服，从而开启了深度学习在学术界和工业界的浪潮
- 目前，深度学习的理论研究还基本处于起步阶段，但在应用领域已显现出巨大能量
  - 2011年以来，微软研究院和Google的语音识别研究人员先后采用DNN技术降低语音识别错误率20%~30%，是语音识别领域十多年来最大的突破性进展
  - 2012年开始，DNN技术在图像识别领域取得惊人的效果，2013开始ImageNet大赛排名前列的算法都是自行学习特征的深度学习算法，在**2015年多个算法击败了人类**
  - **2016年3月，AlphaGO在五番棋中4：1绝对优势击败李世石九段，一直以来被视作Mission Impossible的围棋被深度学习所征服**

谁重新激活了神经元网络？

- **Geoffrey Hinton**
  - 出生于： 1947
  - 专业：
    - 学士，心理学，1970，
    - 博士，人工智能，1978
  - 多伦多大学教授
  - Google 研究中心
  - 1986： 神经元网络BP算法发明人之一
  - 深度学习主要贡献人

I GET VERY EXCITED WHEN WE DISCOVER A WAY OF MAKING NEURAL NETWORKS BETTER — AND WHEN THAT'S CLOSELY RELATED TO HOW THE BRAIN WORKS.'

谁重新激活了神经元网络？

图2：Montreal大学教授及AI研究者 Yoshua Bengio

图3：纽约大学AI研究者及Facebook人工智能研究院的主任Yann LeCun

- **NCAP:** 神经计算和自适应感知项目
  - 2004
  - NCAP Researchers
    - Yoshua Bengio
    - Yann Lecun (FaceBook)
    - Andrew Ng (Baidu)
    - 20~ Others

Core Team

深度学习实现的物理架构

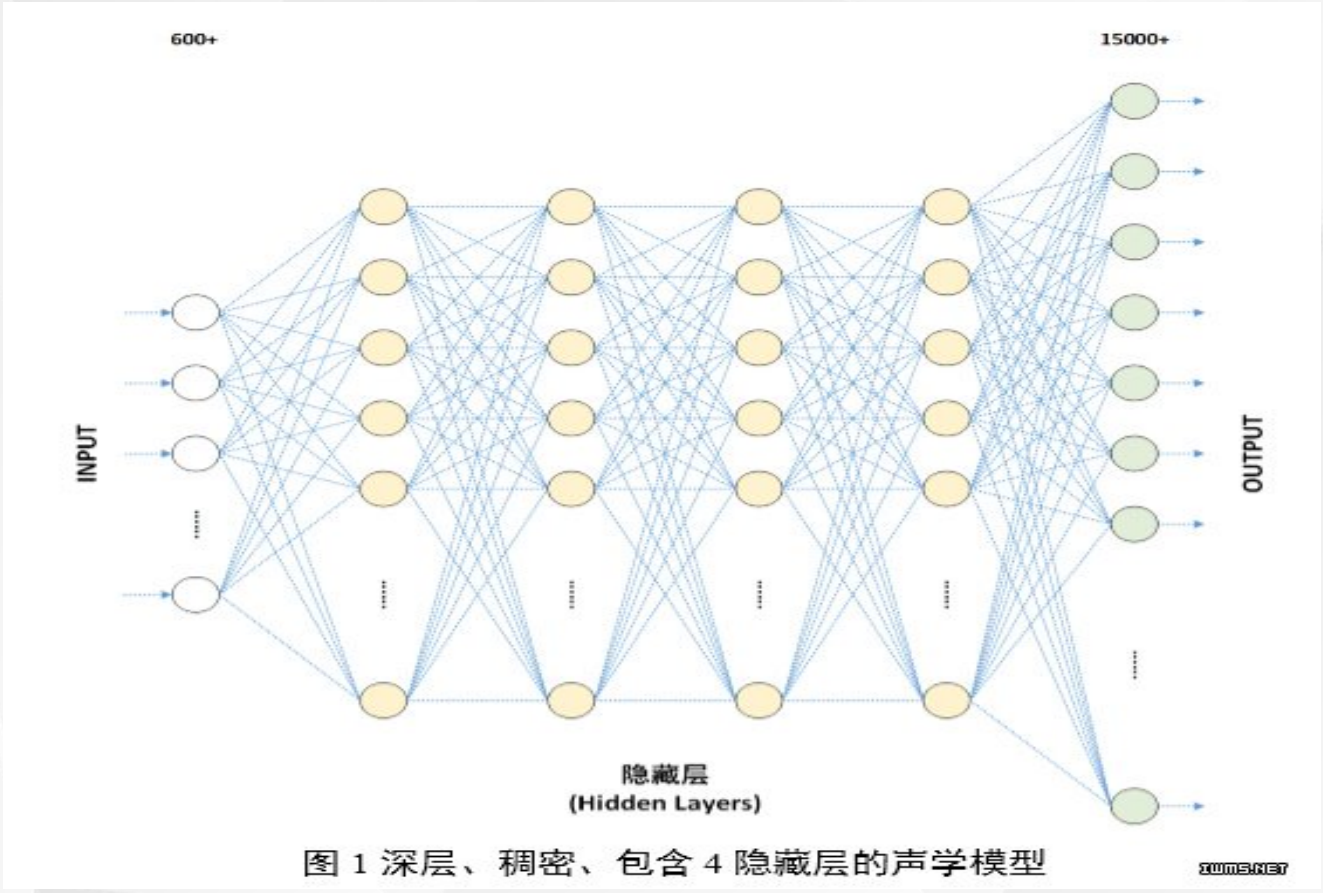


图 1 深层、稠密、包含 4 隐藏层的声学模型

几万神经元， 几千万参数

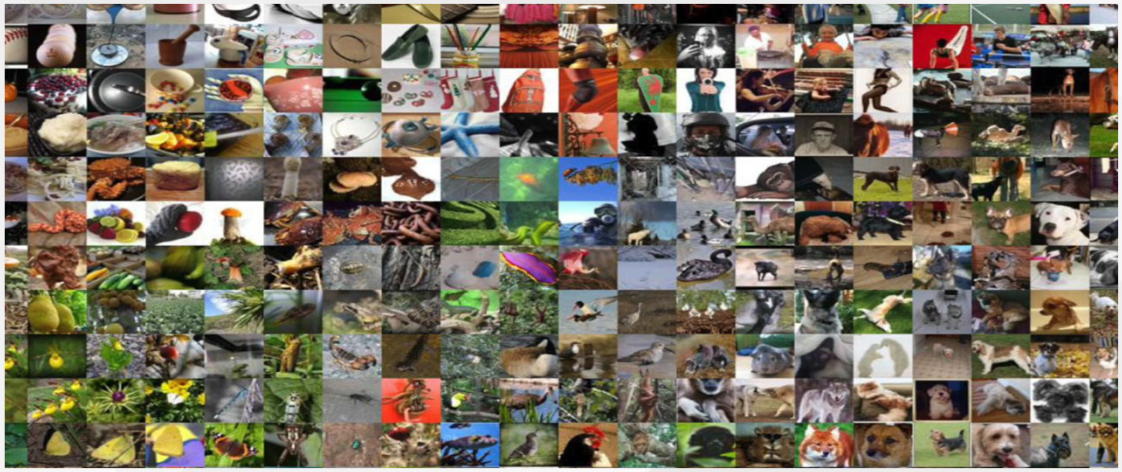
挑战：

◆训练数据规模庞大
◆ 计算开销大
◆训练过程收敛难
◆训练用时久

解决方案：
◆GPU 计算资源并行
◆CPU 集群

ImageNet竞赛

ImageNet提供的数据集包含约120万张训练图像、5万张验证图像和10万张测试图像，目标是将图像分为1000个不同的类别。



Team	Time	Place	Top-5 error
SuperVision	2012	1	16.42%
ISI	2012	2	26.17%
VGG	2012	3	26.98%
Clarifai	2013	1	11.74%
NUS	2013	2	12.95%
ZF	2013	3	13.51%
GoogLeNet	2014	1	6.66%
VGG	2014	2	7.32%
MSRA	2014	3	8.06%
Andrew Howard	2014	4	8.11%
DeeperVision	2014	5	9.51%
Human	2014	-	5.1%
MSRA PReLU-nets	2015.2	-	4.94%
BN-Inception	2015.2	-	4.82%
Deep Image	2015.5	-	4.58%

深度学习方法

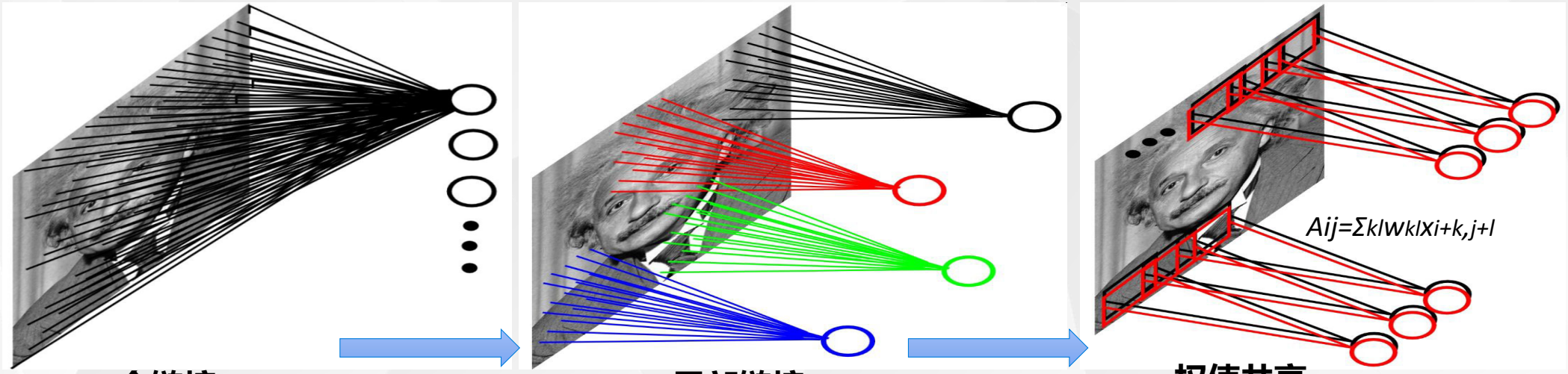
- 2015年2月，微软在ImageNet 2012分类数据集中的错误率已降低至4.94%
- 随后，google将错误率降至4.82%
- 2015年5月，百度宣布错误率降至**4.58%**
  - 百度使用的超级计算机Minwa配备72个CPU处理器和144个GPU处理器

卷积神经网络 Convolutional Neural Networks

CNN是第一个真正成功训练多层网络结构的学习算法。它利用空间关系减少需要学习的参数数目，以提高一般前向BP算法的训练性能。

在CNN中，图像的一小部分（局部感受区域）作为层级结构的最低层的输入，信息再依次传输到不同的层，每层通过一个数字滤波器去获得观测数据的最显著的特征。CNN能够提取对平移、缩放和旋转不变的观测数据的显著特征，在图像处理、语音处理等领域得到了广泛而深入的应用。

CNN使用**局部链接和权值共享**极大降低了模型的参数数量



全链接

- 200x200的图像
- 400,000个隐层节点
- 200x200x400000=**160亿个参数**
- 深度神经网络参数更多
- 参数太多

局部链接

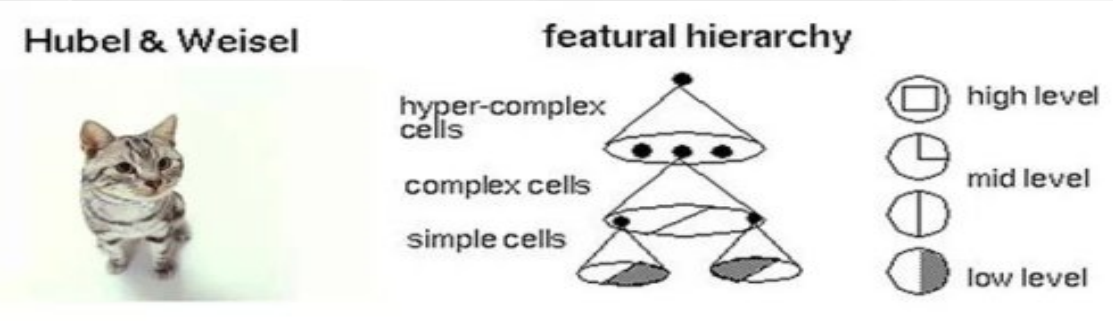
- 200x200的图像
- 10x10局部（感受野）
- 400,000个隐层单元
- **10x10x400000=4千万个参数**
- 模拟了神经细胞生理机制

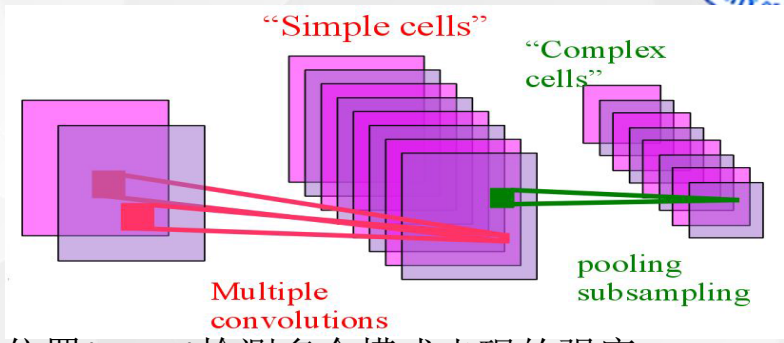
权值共享

- 200x200的图像
- 10个10x10的卷积核
- 400,000个隐层单元
- **10x10x10=1000个参数**
- 每个卷积核，反映了某种局部的模式

卷积神经网络 Convolutional Neural Networks

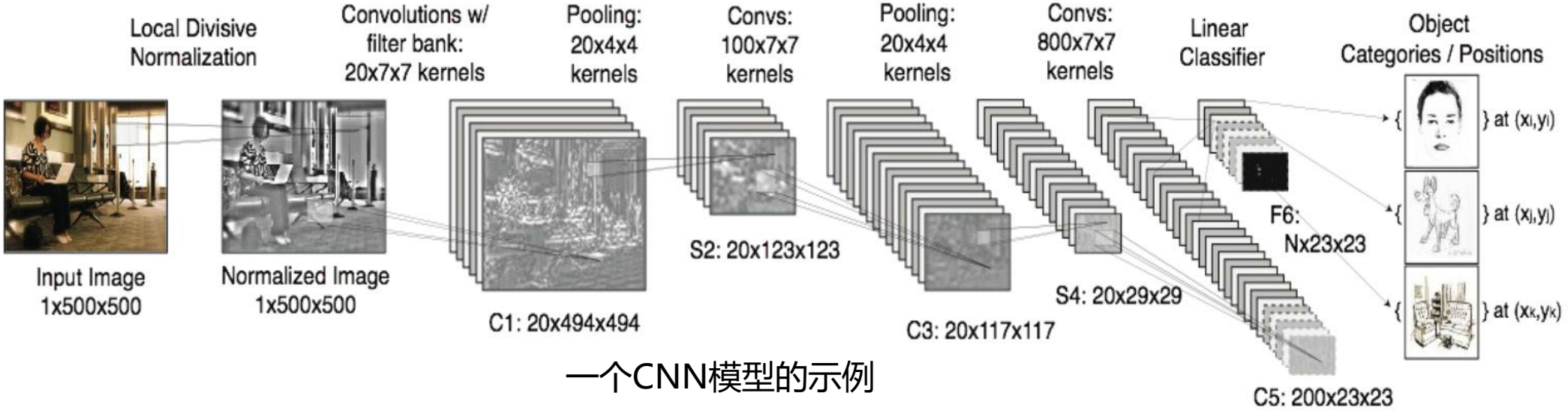
CNN深度学习模型由**多核卷积和池化亚采样**两个基本步骤交替构成。





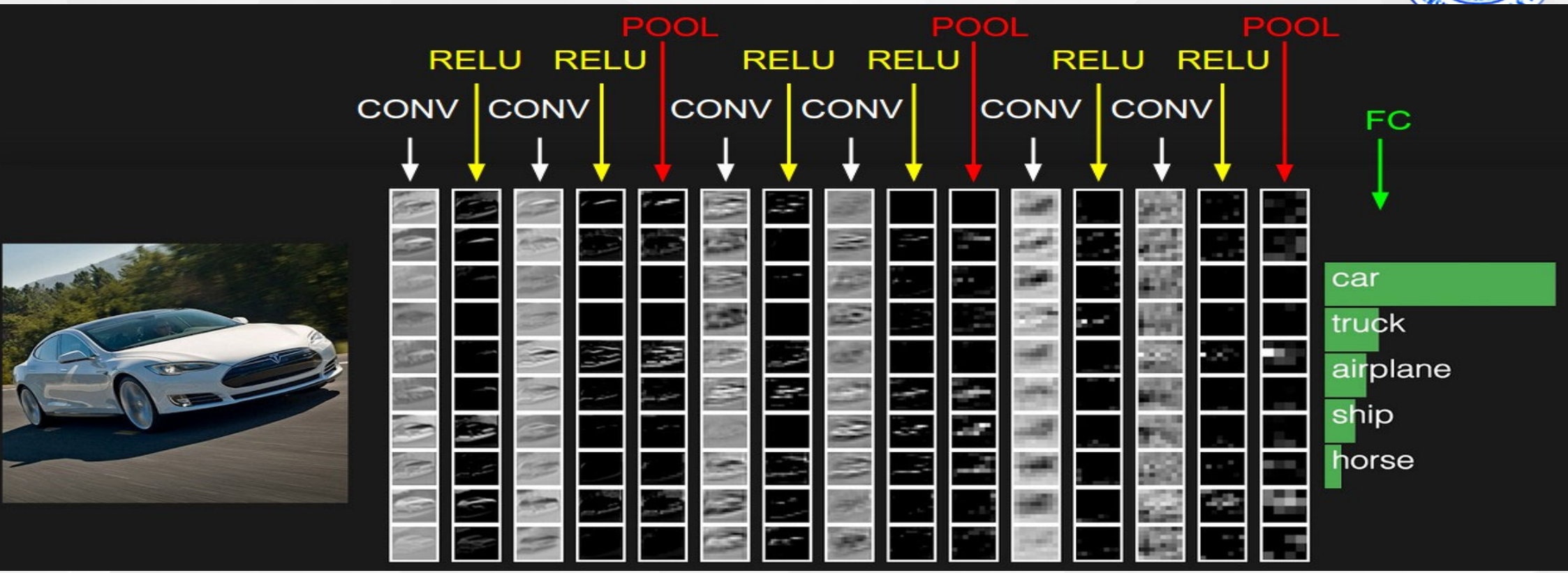
- 受Hubel和Wiesel对视皮层细胞的研究启发
- 简单细胞检测局部特征
- 复杂细胞融合感受野内简单细胞的输出

- 在每一个位置(patch)检测多个模式出现的强度
- 多个Feature Maps上对应同一个patch的响应构成了那个patch的特征向量



一个CNN模型的示例

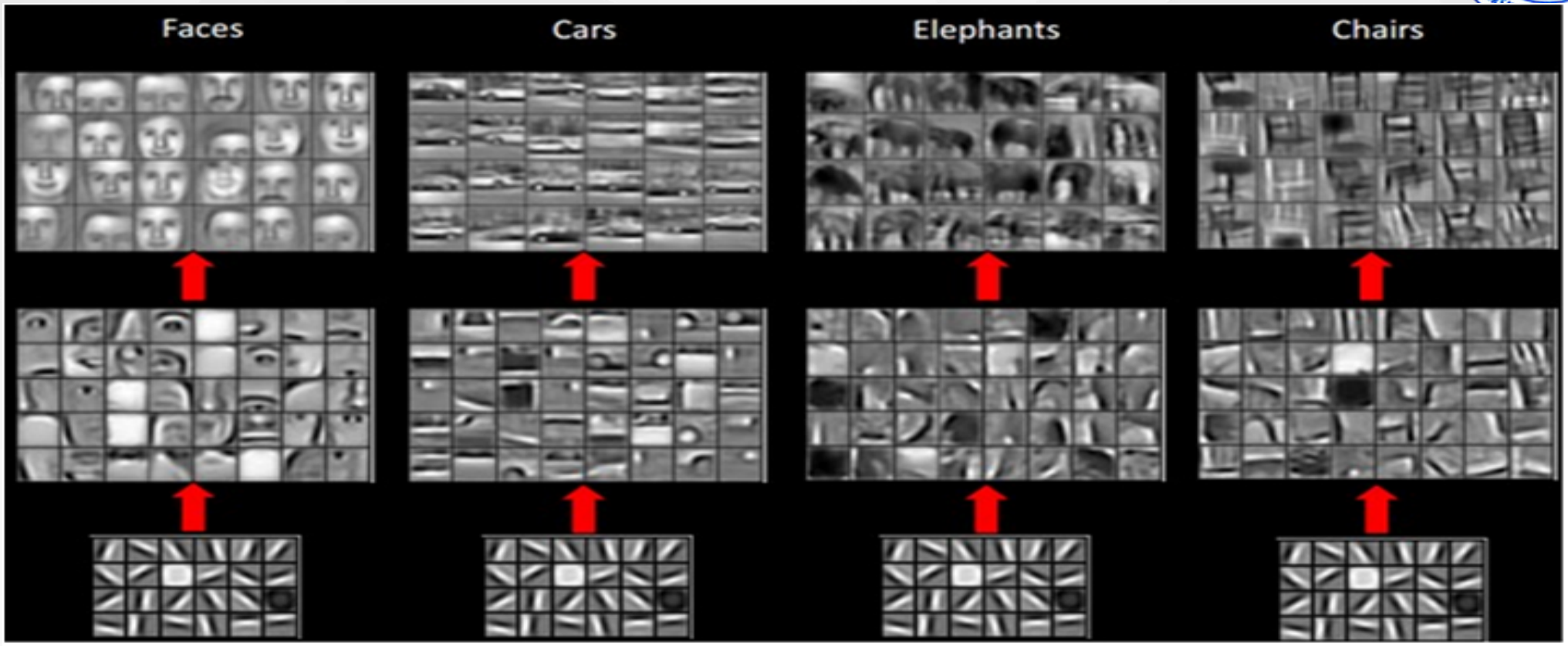
卷积神经网络到底学到了什么



- 具体
- 复杂

- 抽象
- 简单

卷积神经网络为什么有效



- 抽象
- 简单

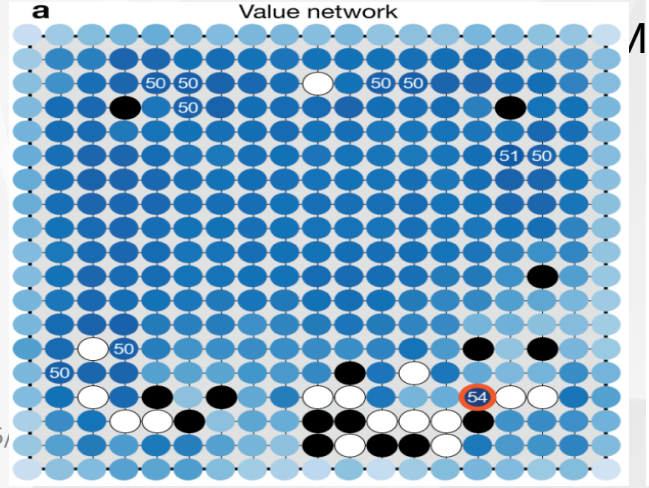
- 具体
- 复杂

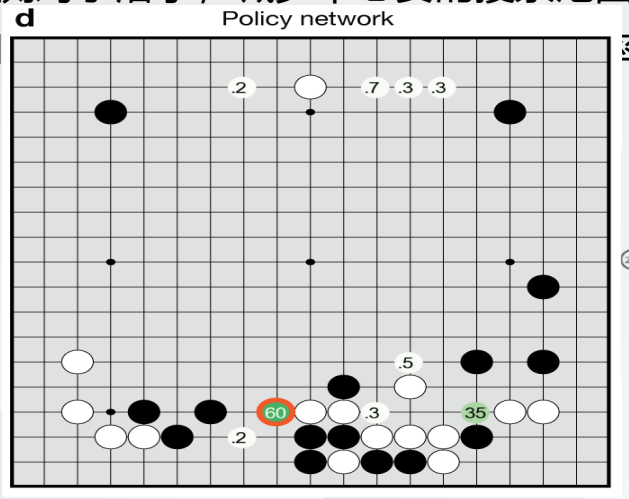
AlphaGo探秘

AlphaGo以4：1击败李世石九段震惊了全世界。虽然深蓝早在1997年就击败了国际象棋世界大师，虽然理论上围棋这种Perfect Information Games迟早会被“电脑”统治早已成为大家的共识，无人预计围棋顶级高手竟然会在今年就拜于“电脑”之“手”。因为围棋的搜索空间约为**250^150(~10^360)**远远大于国际象棋的**35^80(~10^124)。**

“电脑”是怎样下棋的?

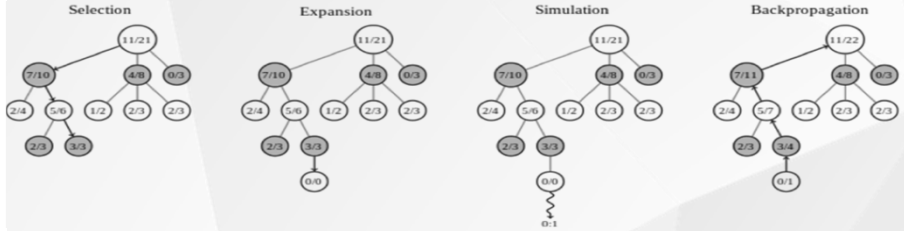
- **理想：评价函数**是求解Perfect Information Games的核心：V*(S)
- **现实：**搜索空间太大，模型计算量太大，无法遍历求解所有可能下法，无法得到完美的评价函数
- **方案：缩减搜索空间！从深度、广度、还有概率（手气）三方面入手**
  - 深度——近似评价函数Value： V*(S) 的一个近似，输出的不是胜负结果而是胜率，若干步后停止
  - 广度——策略函数Policy：预测对手落子，减少不必要的搜索范围





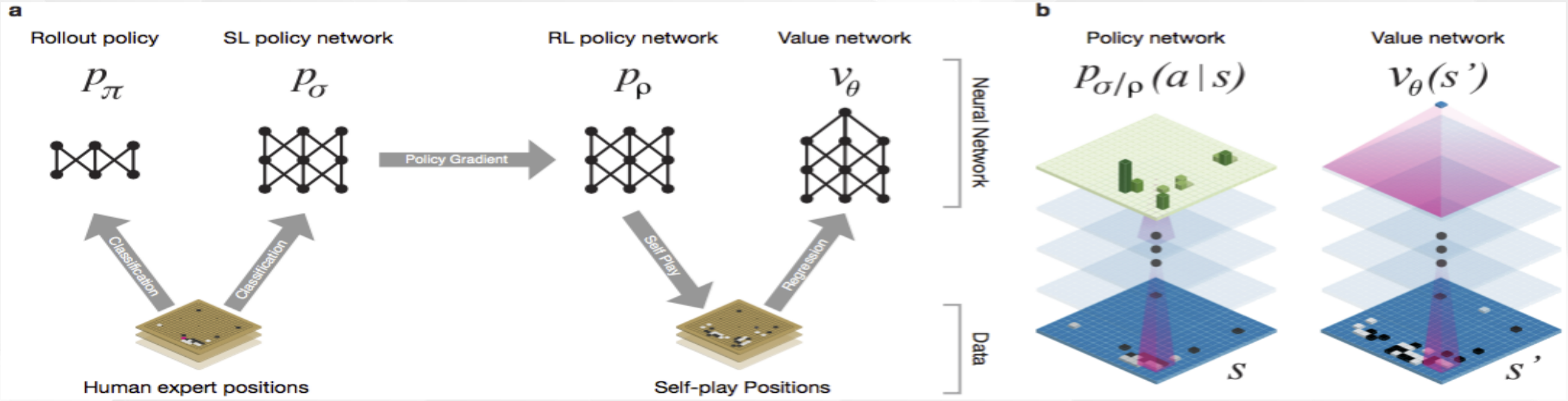
，在有限选项中计算最佳解

MCTS



AlphaGo探秘

- AlphaGo是深度学习创造的奇迹。 AlphaGo和前辈们比较起来，表面上实在没有多少区别，它的核心三大件还是评价函数、策略函数和MCTS，但是前辈们的评价函数和策略函数都是基于线性组合的浅层模型
- AlphaGo本质上就是一个深度学习的神经网络，AlphaGo将围棋棋局视为19×19的图片，策略函数和评价函数都使用了卷积神经网络模型（14层），卷积核为5×5，并先后使用监督学习和强化学习进行训练。



AlphaGo运行参数：

- 异步的多线程MCTS搜索，在CPU执行搜索，在GPU执行评价网络和策略网络的计算.
- 单机版AlphaGo ： 40线程，48CPUs， 8 GPUs
- 集群版AlphaGo： 40线程，1202CPUs， 176 GPUs

关于深度学习和人工智能

- AlphaGo的意义
  - AlphaGo取得了巨大成就，但其基本机制并没有什么颠覆性的东西
  - AlphaGo透过深度学习能够掌握更抽象的概念，但是计算机还是没有自我意识与思考
- 深度学习的局限
  - 在语音和图像识别方面的超人能力预示深度学习将会成为解锁通用人工智能的那把钥匙的一部分，但不是全部
  - 深度学习算法缺乏联想和推理能力，对信息处理的效率与人脑仍然有质的差距
  - 目前，包括深度学习的各种机器学习算法只有空间的概念，没有真正的时间的概念，而常识是与时间密切相关的，人工智能必须要有时间的概念、记忆关联的能力才能进一步超越人类
- 人工智能的潜在风险
  - 全知全能的存在将严重冲击人类社会的现有组织结构
  - 人类可能永远无法理解人工智能

人工智能就像一列火车，它临近时你听到了轰隆隆的声音，你在不断期待着它的到来。他终于到了，一闪而过，随后便远远地把你抛在身后

——王小川

本章小结

(1) 机器学习的相关概念和分类;

(2) 专家系统中的机器学习—归纳（示例）学习

(3) 距离及相似度度量方法;

(4) 分类算法的概念和原理

(5) 聚类算法的概念和原理

(6) 大数据和机器学习

(7) 深度学习简介