

# 第五章 机器学习

- 5.1 机器学习概述
- 5.2 归纳学习
- 5.3 统计学习
- 5.4 聚类
- 5.5 进化计算
- 5.6 群体智能

# 5.1 机器学习概述

- 5.1.1 什么是机器学习？
- 5.1.2 为什么要研究机器学习？
- 5.1.3 机器学习发展阶段
- 5.1.4 机器学习能做什么？
- 5.1.5 机器学习模型
- 5.1.6 机器学习分类
- 5.1.7 机器学习的基本问题



# 5.1.1 什么是机器学习？

- 计算机技术的发展
  - 海量数据（存储和处理的能力）
  - 计算机网络（远程访问数据的能力）
- 例如：
  - 连锁超市遍布全国各地，商品上千种，顾客数百万。
  - 销售终端记录每笔交易的详细资料，包括日期，购买商品和数量、销售价格和总额，顾客标识码等。
  - 我们不能确切的知道哪些人比较倾向于购买哪些特定的商品，也不知道应该向喜欢看尼古拉斯凯奇电影的人推荐哪些其他的电影。
- 我们已经掌握的，就是历史的数据（经验）。
- 我们期望从数据中提取出这些问题或相似问题的答案。

# 5.1.1 什么是机器学习？

- 已经观测到的数据产生是随机的么？其中是否隐含一些规律？
  - 当你去超市买面包的时候，你是不是同时也会买点牛奶？
  - 夏天的时候你是不是经常买雪糕？冬天则很少？
- 数据中存在一些确定的**模式或规律**！



# 5.1.1 什么是机器学习？

- 机器学习？
  - 从历史数据中，发现某些模式或规律（描述）
  - 利用发现的模式和规律进行预测
- 机器学习的定义
  - 基于历史经验的，描述和预测的理论、方法和算法。
  - 从人工智能的角度看，机器学习是一门研究使用计算机获取新的知识和技能，提高现有计算机求解问题能力的科学
- 机器学习可行性的保证
  - 将来，至少是不远的将来，情况不会与收集的样本数据时有很大的不同，因此未来的预测也将有望是正确的。

## 5.1.2 为什么要研究机器学习？

### □ 必要性：

- 理解学习的本质和建立学习系统是AI研究的目标之一
- 现有的大多数AI系统都是演绎的，没有归纳推理，因而不能自动获取和生成知识

### □ 可行性：

- 学习的过程是信息处理的过程，这包括直接记忆和经过推理
- 已有工作说明可以实现一定程度的机器学习



## 5.1.2 为什么要研究机器学习？

### □ 研究目标：

- 通用学习算法：理论分析任务和开发用于非实用学习任务的算法
- 认知模型：研究人的学习的计算模型和实验模型
- 工程目标：解决专门的实际问题，并开发完成这些任务的工程系统

### □ 困难：

- 学习系统性能的预测更加困难
- 获取知识的本质还是猜想。由特定的观察和类比生成的知识不可能证明其正确性。

# 5.1.3 机器学习发展阶段

机器学习的研究大致可以分为三个阶段：

## ■ （一） 五六十年的探索阶段：

主要受神经生理学、生理学和生物学的影响，研究主要侧重于非符号的神经元模型的研究，主要研制通用学习系统，即神经网络或自组织系统。

主要成果有：

- 感知机（Perceptron）
- Friedberg等模拟随机突变和自然选择过程的程序，
- Hunt等的决策树归纳程序CLS。



## ■ （二） 七十年代的发展阶段：

由于当时专家系统的蓬勃发展，知识获取成为当务之急，这给机器学习带来了契机，主要侧重于**符号学习的研究**。机器学习的研究脱离了基于统计的以优化理论为基础的研究方法，提出了基于符号运算为基础的机器学习方法，并产生了许多相关的学习系统，

主要系统和算法包括：

- Winston的积木世界学习系统；
- Michalski基于逻辑的归纳学习系统AQVAL；
- Michalski和Chilausky的AQ11；
- Quinlan的**ID3程序**
- Mitchell的版本空间方法。

### （三）八九十年代至今的鼎盛阶段。

理论研究和应用研究也有了新的突破，机器学习的研究进入了全面的、系统化的时期。

主要成果有：

- 一方面传统的符号学习的各种方法已日臻完善。Michalski等将AQ11扩充为一个多功能学习系统AQ15，ID3算法中使用了熵，从而使**决策树归纳**得到了很大的改进。
- 科学发现系统BACON开辟了**无导师学习**的两个重要研究领域。
- **神经网络学习**在消沉了一段时期后又重新蓬勃发展起来了，同时计算机硬件技术的高速发展也为开展大规模和高性能的人工神经网络提供了保障，使得基于神经网络的连接学习从低谷走出，发展迅猛。其中Rumelhart等人提出的**BP模型**，提供了一个训练多层网络的实际可行的方法，克服了Perceptron的大部分局限性。



- 另一方面，机器学习的基础理论的研究越来越引起人们的重视。
- 1984年美国学者Valiant提出了基于概率近似正确性的学习理论（**PAC学习**），对布尔函数的一些特殊子类的可学习性进行了探讨，将可学习性与计算复杂性联系在一起，并由此派生出了“计算学习理论”（COLT）
- 1995年，Vapnik出版了“统计学习理论”一书。
- 对PAC的研究是一种理论性，存在性的；Vapnik的研究却是构造性的，他将这类研究模型称为**支持向量机**SVM（Support Vector Machine）。

## 5.1.4 机器学习能做什么？

- 机器学习也是人工智能的组成部分。
- 授予鱼不如授予渔
  - 为了智能化，处于变化环境中的系统不需具备学习能力。如果系统能够学习并且适应这些变化，那么系统设计者就不必预见所有情况，并为它们提供解决方案了。

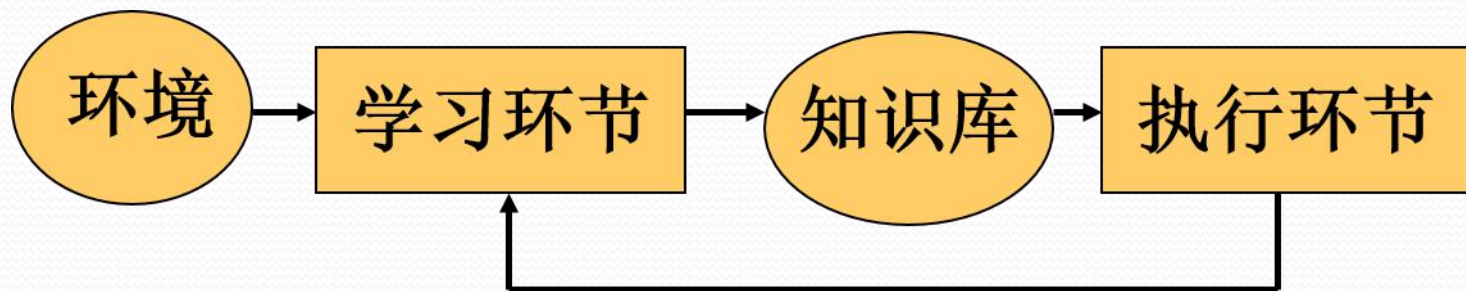


## 5.1.4 机器学习能做什么？

- 机器学习还可以解决视觉、语音识别以及机器人方面的许多问题。
- 模式识别
  - 图像和音频的获得很容易，机器如何做到识别？让机器人识别人脸？辨别声音？
  - 一个图像并非是像素点的随机组合，人脸是有结构、对称的。人脸上的器官是有组合模式的。
  - 通过分析一个人的脸部图像的多个样本，学习程序是可以捕获到那个人特有的模式。然后进行辨认。

## 5.1.5 机器学习模型

- 学习的一种模型



- 环境：外部信息的来源，它将为系统的学习提供有关信息
- 知识库：代表系统已经具有的知识
- 学习环节：系统的学习机构，它通过对环境的感知取得外部信息，然后经分析、综合、类比、归纳等思维过程获得知识，生成新的知识或改进知识库的组织结构。
- 执行环节：基于学习后得到的新的知识库，执行一系列任务，并将运行结果报告学习环节，以完成对新知识库的评价，指导进一步的学习工作，是该模型的核心。



## 5.1.6 机器学习分类

- 按照有无指导来分：
  - 有监督学习（或有导师学习）、无监督学习（或无导师学习）和强化学习（或增强学习）。
- 按学习方法来分：
  - 有机械式学习、指导式学习、范例学习、类比学习、解释学习。
- 按推理策略来分：
  - 有演绎学习、归纳学习、类比学习、解释学习等。
- 综合多因素的分类：
  - 有人工神经网络学习、进化学习、概念学习、分析学习、基于范例的学习等等。

## 5.1.7 机器学习的基本问题

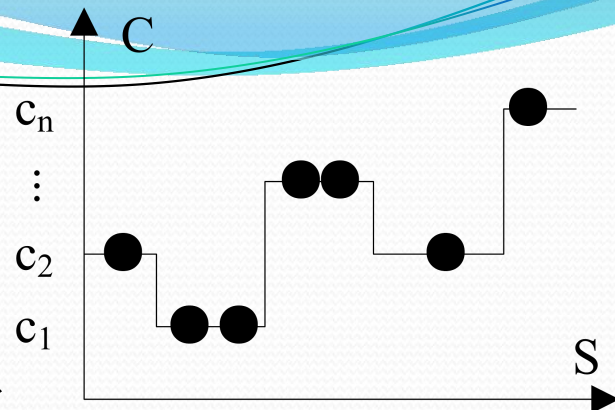
- 机器学习中解决的基本问题主要有：
  - 分类、聚类、预测、联想、优化。
- 令 $S$ 表示数据空间， $Z$ 表示目标空间。

机器学习就是在现有观察的基础上求得一个函数  $L:S \rightarrow Z$ ，实现从给定数据到目标空间的映射。

- 不同特征的学习函数实际上表示了不同的基本问题。

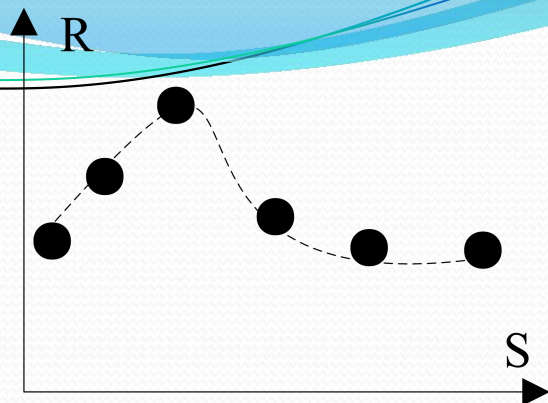


# 分类问题



- 目标空间是已知有限离散值空间，  
即，  $Z=C=\{c_1, c_2, \dots, c_i, \dots, c_n\}$   
待求函数就是分类函数（分类器/分类模型）。
- 分类问题所用的训练数据是  $\langle D, C \rangle$ ，  $D \subset S$ 。
- 由于学习时目标类别已知，所以分类算法都是有监督学习。
- 常用的方法：
  - 决策树方法、贝叶斯方法、前馈神经网络BP算法、支持向量机方法等

# 预测问题



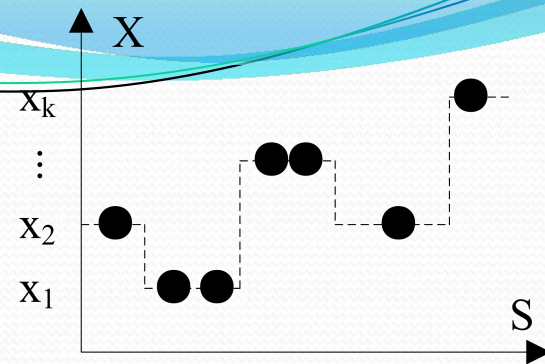
- 目标空间是连续值空间，待求函数就是回归（拟合）曲线（面）。

此时机器学习解决预测问题，也就是求一个数据在目标空间中符合某观测规律的象。

- 预测问题所用的训练数据是  $\langle D, R \rangle$ ,  $D \subset S$ .
  - 一般情况下我们事先已知（或者选择了）曲线（面）模型，需要学习的是模型中的参数。
  - 例如已知多项式模型，但是要学习各项的系数。
- 常用的方法：
  - 人工神经网络方法、线性回归、非线性回归、灰色预测模型等。

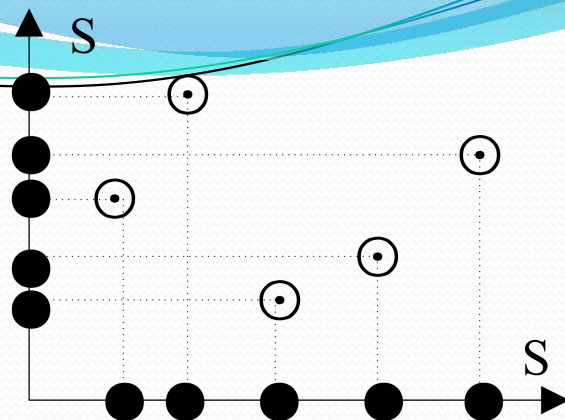


# 聚类问题



- 目标空间是未知有限离散值空间，即， $Z=X=\{x_1, x_2, \dots, x_k\}$   
待求函数就是聚类函数，也称为聚类模型。
- 聚类问题就是把已知数据集划分为不同子集（类别），并且不同类别之间的差距越大越好，同一类别内的数据差距越小越好。
- 聚类问题所用的训练数据是  $D$  ( $D \subset S$ )。
- 聚类问题要用无监督学习
- 常用的方法：
  - 划分聚类法、层次聚类法、基于密度的聚类、基于网格的聚类、自组织特征映射网络等等。

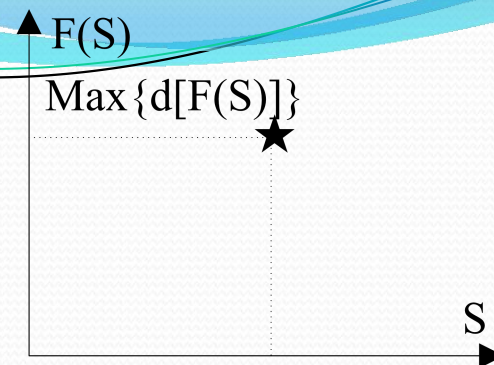
# 联想问题



- 目标空间就是数据空间本身，  
即， $Z=S$   
待求函数就是求自身内部的一种映射。
- 联想问题，也称为相关性分析或者关联问题
  - 就是发现不同数据（属性）之间的相互依赖关系。
  - 简单地说，就是从事物A推出事物B，即 $A \rightarrow B$
- 常用的方法：
  - 反馈神经网络、关联规则、回归分析等等。



# 优化问题



- 目标空间是数据空间上的某种函数（用 $F(S)$ 表示），且学习目标为使对函数 $F(S)$ 的某种度量 $d[F(S)]$ 达到极值。
- 解决优化问题，就是在给定数据范围内寻找使某值达到最大（最小）的方法。
- 优化问题一般都有有一些约束条件
  - 例如时空资源的限制等等。
  - 典型代表就是NP问题，这也是计算机科学中的一类经典问题。
- 解决优化问题对于提高系统效率，保证系统实用性有重要意义。
- 常用的方法有：
  - 遗传算法、Hopfield神经网络、线性规划方法等等。

## 5.2 归纳学习

- 5.2.1 基本概念
- 5.2.2 变形空间学习
- 5.2.3 决策树



## 5.2.1 基本概念

- 归纳学习 (Inductive Learning)
  - 就是从个别到一般，根据某个概念的一系列已知的正例和反例，从中归纳出一个一般的概念描述
  - 旨在从大量的经验数据中归纳抽取出一般的判定规则和模式。
  - 是机器学习中最核心、最成熟的分支。
- 归纳学习也称为：
  - 经验学习：归纳学习依赖于经验数据
  - 基于相似性的学习：归纳学习依赖于数据间的相似形
- 归纳的操作：
  - **泛化(Generalization)**：扩展某假设的语义信息，使其能够包含更多的正例
  - **特化(Specialization)**：泛化的相反操作，用于限制概念描述的应用范围

- 归纳学习的分类和研究领域：

- 符号学习

- 监督学习：

- 实例学习：系统事先将训练例子（经验数据）分类：正、负例子。由于它产生规则，所以也称为概念学习

- 无监督学习：事先不知道训练例子的分类

- 概念聚类

- 机器发现

- 神经网络：本质上是实例学习，为区别起见，称为联结学习

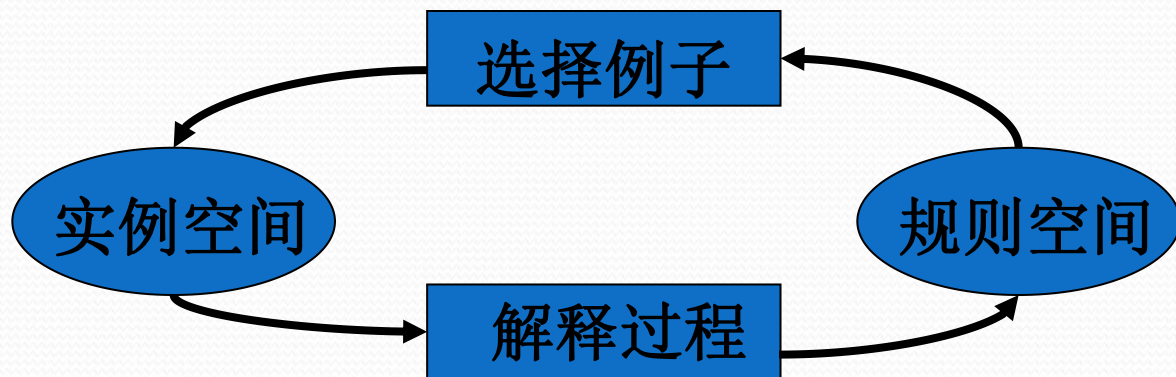
- 学习的计算理论

- 传统的算法复杂性分析

- 概率近似正确性学习研究（计算学习理论）



## 5.2.1 基本概念



- 实例空间要考虑的问题：
  - 1) 示教例子的质量
  - 2) 实例空间的组织和搜索方法
- 规则空间要考虑的问题
  - 1) 形成知识的归纳推理方法
  - 2) 搜索规则空间的方法
  - 3) 对规则空间的要求

## ● 按规则空间搜索方法分类：

### ● (1) 数据驱动方法：

- **变型空间方法**：采用统一的形式表示规则和例子。
- **改进假设方法**：例子和规则的表示不统一。程序根据例子选择一种操作，用该操作修改H中的规则

### ● (2) 模型驱动方法：

- **产生和测试方法**：针对示教例子反复产生和测试假设的规则。利用基于模型的知识产生假设的规则，便于只产生可能合理的假设
- **方案示例方法**：使用规则方案的集合来限制可能合理的规则形式，最符合示教例子的规则被认为是最合理的规则



## 5.2.2 变型空间方法

- 基本思想：以整个规则空间为初始的假设规则集合H，根据示教例子中的信息，对集合H进行一般化或特殊化处理，逐步缩小集合H，最后使H收敛为只含要求的规则。
- 规则空间中的偏序关系：它是按一般性和特殊性来建立的一种概念之间的关系
- 排序后的变形空间：
  - 最上面：是最一般的 规则(概念)，是没有描述的点，所有的例子都符合这一概念
  - 最下面一行的各点：是示教正例对应的概念，每个点的概念只符合一个正例

## 5.2.2 变型空间方法

- 假设规则的集合H:
  - H是规则空间的子集
  - H中最一般的元素组成的子集称为G集合
  - H中最特殊的元素组成的子集称为S集合
  - 在规则空间中，H是G和S中间的一段。
  - 可以用G和S来表示H
- 变型空间方法：
  - 初始：G是最上面一个点，S是最下面的直线（示教正例），H为整个规则空间
  - 搜索过程：G下移，S上移，H逐步缩小。
  - 结果：H收敛为只含一个要求的概念



## 5.2.2 变型空间方法

- 消除候选元素算法

(1) 正规的初始H集是整个规则空间，这时S包含所有可能的示教正例（最特殊的概念）。

(2) 接收一个新的示教例子。

如果是正例：去掉G中不覆盖新正例的概念，然后修改S为由新正例和S原有的元素共同归纳出的最特殊的结果

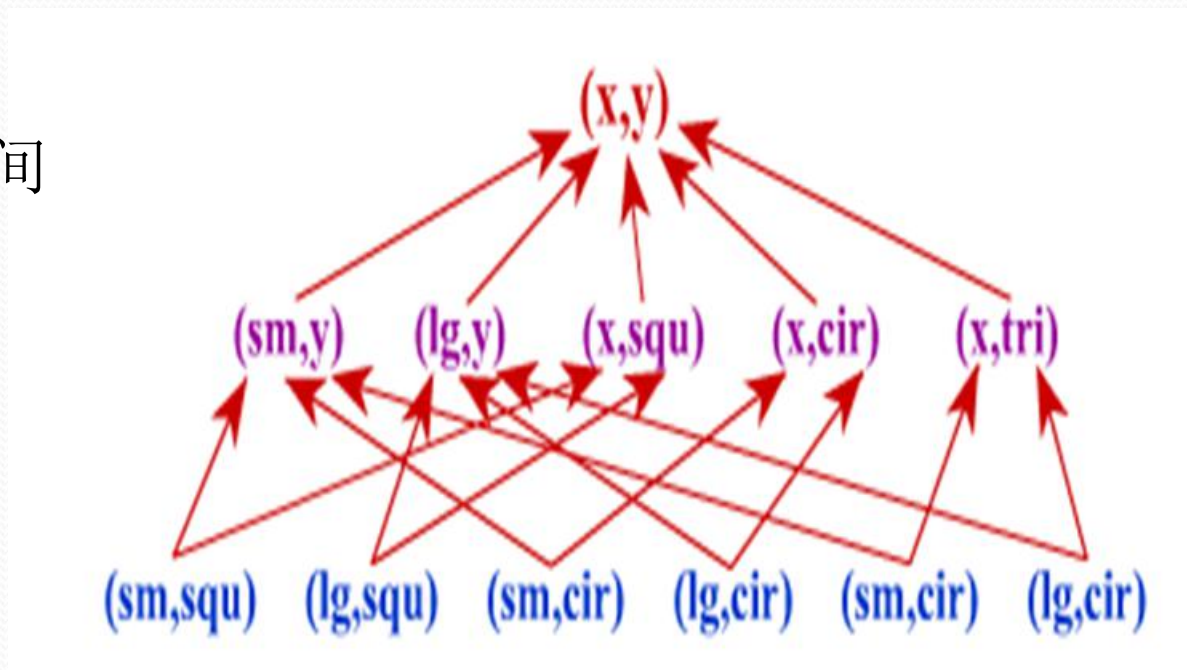
如果是反例：从S中去掉覆盖该反例的概念；然后修改G为由新反例和G原有元素共同特殊化为最一般的结果

(3) 若 $G=S$ ，且是单元集合，则转(4)，否则转(2)

(4) 输出H中的概念（即G和S）

- 物体的大小用大(lg)或小(sm)来描述；物体的形状用圆(cir)、方(squ)或三角(tri)来描述。若用x表示大小,用y表示形状。用删除候选算法学习“圆”的概念。

初始变形空间

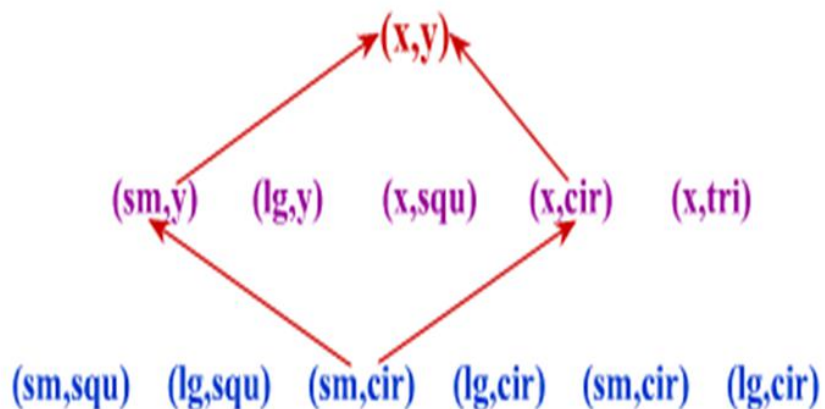


$$G=\{(x, y)\}$$

$$S=\{(sm, squ), (sm, cir), (sm, tri), (lg, squ), (lg, cir), (lg, tri)\}$$

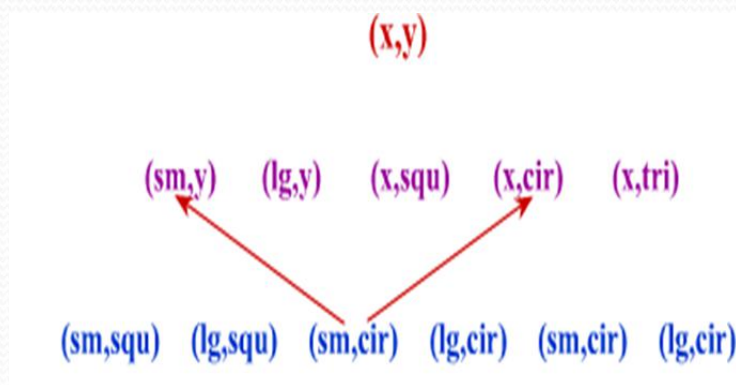


- 第一个示教例子正例(sm, cir), 表示小圆是圆后的变形空间



$$G=\{(x, y)\}, S=\{(sm, cir)\}$$

- 第二个示教例子反例(lg, tri)。这表示大三角不是圆后的变形空间



$$G=\{(x, cir), (sm, y)\}$$

$$S=\{(sm, cir)\}$$

第三个示教例子正例(lg, cir), 表示大圆是圆。

$$G=\{(x, cir)\} S=\{(x, cir)\}$$

## 5.2.2变型空间方法

- 变型空间法的缺点

(1) 抗干扰能力差

- 变形空间法是数据驱动的方法，所有数据驱动的方法都难以处理有干扰的训练例子
- 算法得到的概念应满足每个示教例子的要求，所以一个错误的例子会造成很大的影响

(2) 无法发现析取概念



## 5.2.3 决策树

决策树(decision tree)也称判定树，它是由对象的若干属性、属性值和有关决策组成的一棵树。其中的节点为属性（一般为语言变量），分枝（边）为相应的属性值（一般为语言值）。

- 从同一节点出发的各个分枝之间是逻辑“或”关系；
- 根节点为对象的某一个属性；
- 从根节点到每一个叶子节点的所有节点和边，按顺序串连成一条分枝路径，位于同一条分枝路径上的各个“属性-值”对之间是逻辑“与”关系，叶子节点为这个与关系的对应结果，即决策。

- 决策树学习是以实例为基础的归纳学习。
- 从一类无序、无规则的事物（概念）中推理出决策树表示的分类规则。
- 概念分类学习算法：来源于
  - Hunt, Marin和Stone 于1966年研制的**CLS**学习系统，用于学习单个概念。
  - 1979年, J.R. Quinlan 给出ID<sub>3</sub>算法，并在1983年和1986年对**ID<sub>3</sub>**进行了总结和简化，使其成为决策树学习算法的典型。
  - Schlimmer 和Fisher 于1986年对ID<sub>3</sub>进行改造，在每个可能的决策树节点创建缓冲区，使决策树可以递增式生成，得到**ID<sub>4</sub>**算法。
  - 1988年，Utgoff 在ID<sub>4</sub>基础上提出了**ID<sub>5</sub>**学习算法，进一步提高了效率。
  - 1993年，Quinlan 进一步发展了ID<sub>3</sub>算法，改进成**C<sub>4.5</sub>**算法。
  - 另一类决策树算法为**CART**，与C<sub>4.5</sub>不同的是，CART的决策树由二元逻辑问题生成，每个树节点只有两个分枝，分别包括学习实例的正例与反例。



## ● 一、CLS决策树构造算法

在CLS算法生成的决策树中，节点对应于待分类对象的属性，由某一节点引出的弧对应于这个属性的取值，叶节点对应于分类的结果。

设分类对象的属性表为 $\text{AttrList}=\{A_1, A_2, \dots, A_n\}$ ，每一个属性 $A_i$ 的值域记为 $\text{Value}(A_i)$ ，需要分类的分类结果组成分类结果表 $\text{Class}=\{C_1, C_2, \dots, C_m\}$ 。一般应有 $n \geq 1$ ， $m \geq 2$ 。训练实例集 $\text{TR}=\{\langle X, C \rangle\}$ 中的一个元素就是一个训练实例 $\langle X, C \rangle$ ，训练实例 $\langle X, C \rangle$ 的特征向量为 $X=(a_1, a_2, \dots, a_n)$ ，其中 $a_i$ 为这个实例的第 $i$ 个属性 $A_i$ 的取值， $C$ 为这个实例的分类结果， $C \in \text{Class}$ 。

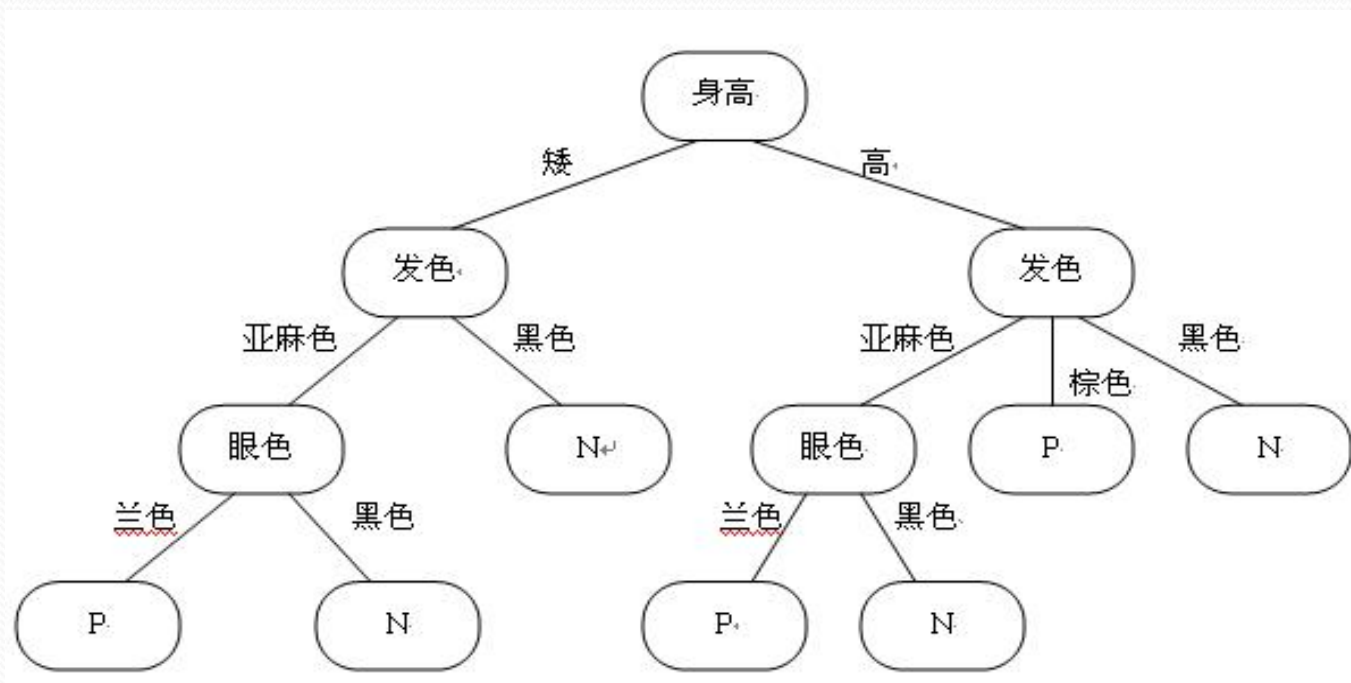
## • CLS算法

- 1) 如果TR中所有实例的分类结果均为 $C_i$ ，则返回 $C_i$ ；
- 2) 从属性表中选择某一属性A作为检测属性；
- 3) 假设属性 $A_i$ 的值域表 $\text{Value}(A_i)$ 有k个值，则将TR划分为k个子集 $TR_1, TR_2, \dots, TR_k$ ，同一个子集节点中的实例属性 $A_i$ 有相同的属性值；
- 4) 从属性表中删除已做检验的属性A；
- 5) 对每一个i，用 $TR_i$ 和新的属性表递归调用CLS生成 $TR_i$ 的决策树 $DTR_i$ ；
- 6) 返回以属性A为根， $DTR_1, DTR_2, \dots, DTR_k$ 为子树的决策树。



## 例根据人员的外貌特征属性对人员进行分类

序号	身高	发色	眼色	类别	序号	身高	发色	眼色	类别
1	矮	亚麻色	<u>兰色</u>	P	5	高	黑色	<u>兰色</u>	N
2	高	亚麻色	黑色	N	6	高	亚麻色	<u>兰色</u>	P
3	高	棕色	<u>兰色</u>	P	7	高	黑色	黑色	N
4	矮	黑色	<u>兰色</u>	N	8	矮	亚麻色	黑色	N



## 二、ID3算法

- 1、是利用信息论原理对大量样本的属性进行分析和归纳而产生的。
- 2、决策树的根结点是所有样本中信息量最大的属性。树的中间结点是该结点为根的子树所包含的样本子集中信息量最大的属性。决策树的叶结点是样本的类别值。
- 3、用信息增益（即信息论中的互信息）来选择属性作为决策树的结点。



## 二、ID3算法

- 熵 (entropy)：给定有关某概念的正例和负例的集合S。

对此BOOLEAN分类的熵为：

$$\text{Entropy}(S) = - \text{pos} \log_2(\text{pos}) - \text{neg} \log_2(\text{neg})$$

“pos” 和 “neg” 分别表示S中正例和负例的比例。

并定义： $0 \log_2(0) = 0$

- 如果分类器有c个不同的输出，则：

$$\text{Entropy}(S) = - \sum_{i=1}^c p_i \log_2(p_i)$$

$p_i$  表示S中属于类i的比例

## 二、ID3算法

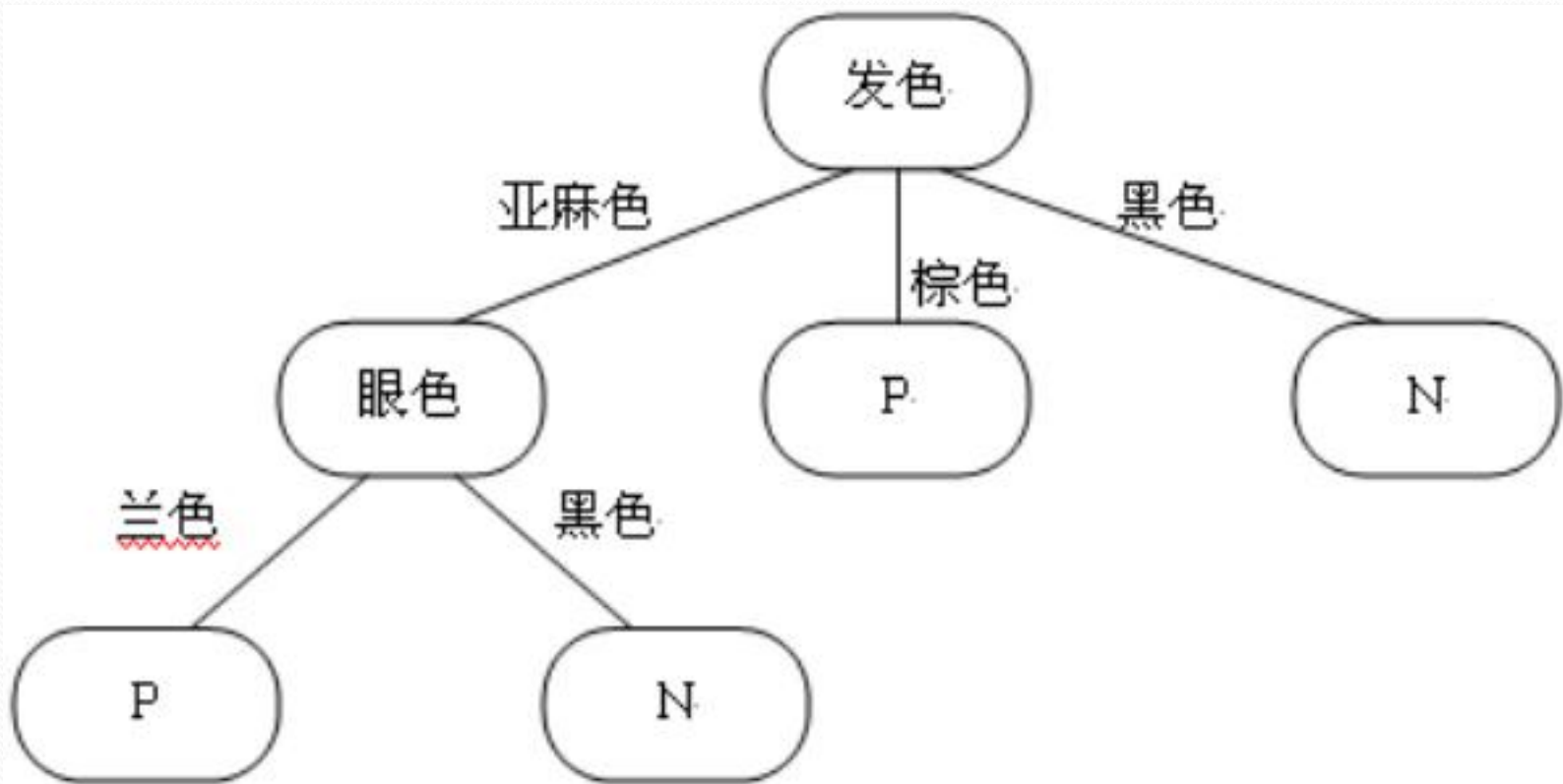
- 实例集合S中属性A的信息增益为：

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{values of } A} (|S_v| / |S|) \text{Entropy}(S_v)$$

$S_v$ 表示S的子集，其属性A的值为V



序号	身高	发色	眼色	类别	序号	身高	发色	眼色	类别
1	矮	亚麻色	<u>兰色</u>	P	5	高	黑色	<u>兰色</u>	N
2	高	亚麻色	黑色	N	6	高	亚麻色	<u>兰色</u>	P
3	高	棕色	<u>兰色</u>	P	7	高	黑色	黑色	N
4	矮	黑色	<u>兰色</u>	N	8	矮	亚麻色	黑色	N



## 5.3 统计学习

- 5.3.1 统计学习概述
- 5.3.2 支持向量机



## 5.3.1 统计学习概述

- 统计方法是从事物的外在数量上的表现去推断该事物可能的规律性。

科学规律性的东西一般总是隐藏得比较深，最初总是从其数量表现上通过统计分析看出一些线索，然后提出一定的假说或学说，作进一步深入的理论研究。当理论研究提出一定的结论时，往往还需要在实践中加以验证。就是说，观测一些自然现象或专门安排的实验所得资料，是否与理论相符、在多大的程度上相符、偏离可能是朝哪个方向等等问题，都需要用统计分析的方法处理。

- 统计方法处理过程可以分为三个阶段：

- ① 搜集数据：采样、实验设计
- ② 分析数据：建模、知识发现、可视化
- ③ 进行推理：预测、分类

## □ 统计学习方法：

- ✓ 传统方法：渐近理论，即当样本趋向于无穷多时的统计性质。统计方法主要考虑测试预想的假设和数据模型拟合。它依赖于显式的基本概率模型。
- ✓ 模糊集
- ✓ 粗糙集
- ✓ 支持向量机

## □ 常见的统计方法有：

- ✓ 回归分析（多元回归、自回归等）
- ✓ 判别分析（贝叶斯判别、费歇尔判别、非参数判别等）
- ✓ 聚类分析（系统聚类、动态聚类等）
- ✓ 探索性分析（主元分析法、相关分析法等）等。



## 5.3.2 支持向量机

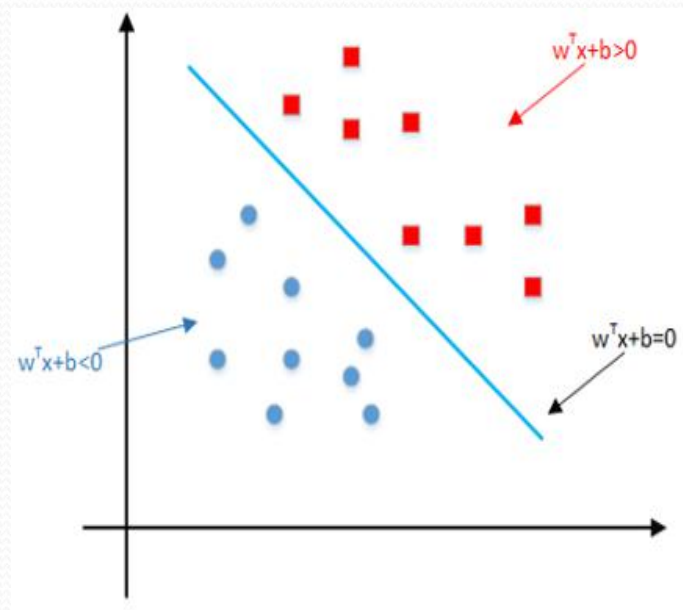
- 支持向量机 (support vector machine: SVM) 是一种二类分类方法，它的基本模型是定义在特征空间上的间隔最大的线性分类器。
- 支持向量机方法是建立在统计学习理论的VC 维理论和结构风险最小原理基础上的。
- 在解决小样本、非线性及高维模式识别中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中。

# 一、线性可分

对于分离超平面： $\mathbf{w}^T \mathbf{X} + b = 0$

可以将方点和圆点表示的两类不同数据完全分离在该超平面的两侧，使其满足：

- 1、 $\mathbf{w}^T \mathbf{X} + b > 0$ 的所有数据属于一类；
- 2、 $\mathbf{w}^T \mathbf{X} + b < 0$ 的所有数据属于另一类。



称样本数据集为**线性可分**；  
称 $\mathbf{w}^T \mathbf{X} + b = 0$ 为**分离超平面**。

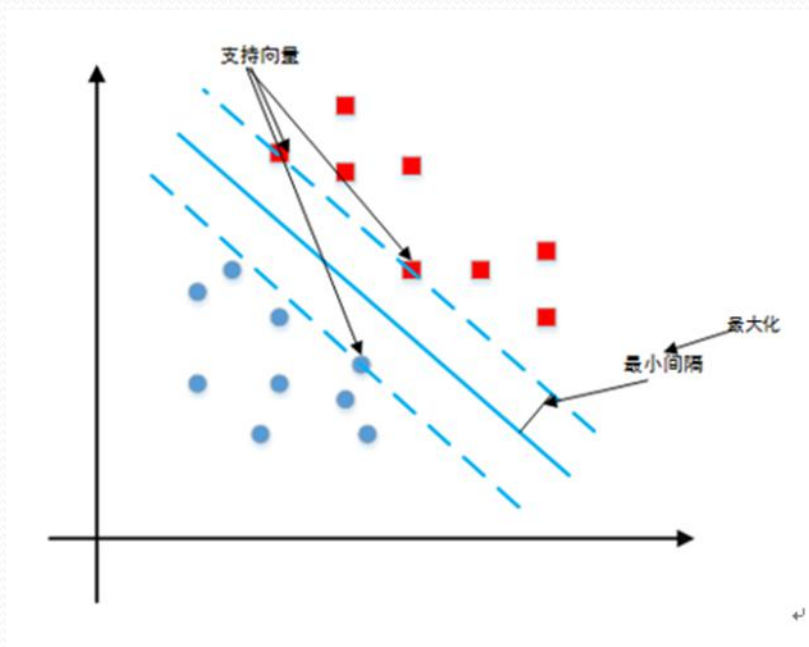
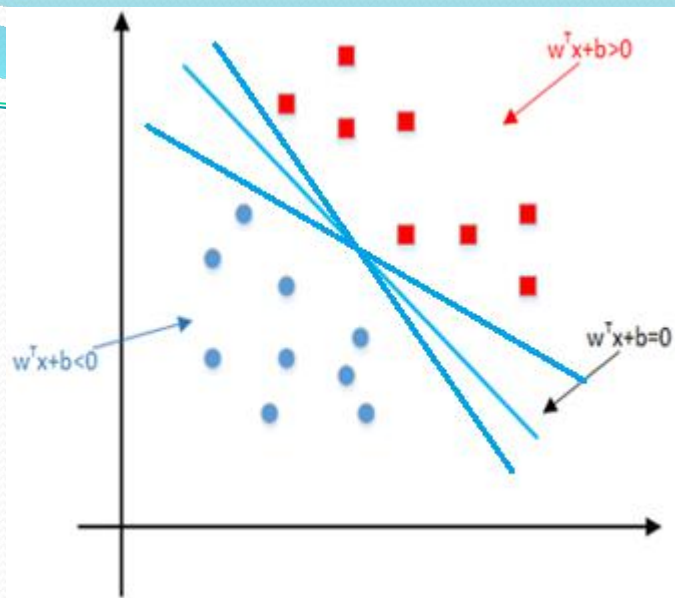


# 一、线性可分

对于一个线性可分性的样本数据集，其分离超平面通常不止一个。

**SVM模型：**分离超平面使得两类样本数据与该分离超平面形成的间隔均为最大。

SVM只输出样本类别而不输出样本属于某一类别的概率。

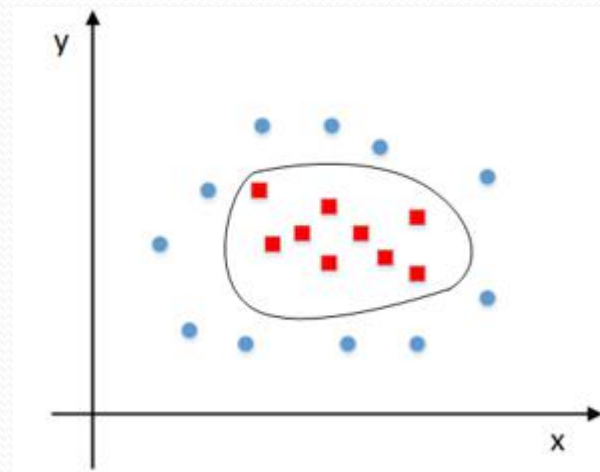


## 二、核函数

带标签样本数据集： $D = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\}$

图表示 $D$ 中样本数据点在二维特征空间中的分布。

显然无法直接找到一个超平面对图中两类样本数据点实现精确或大致分离。



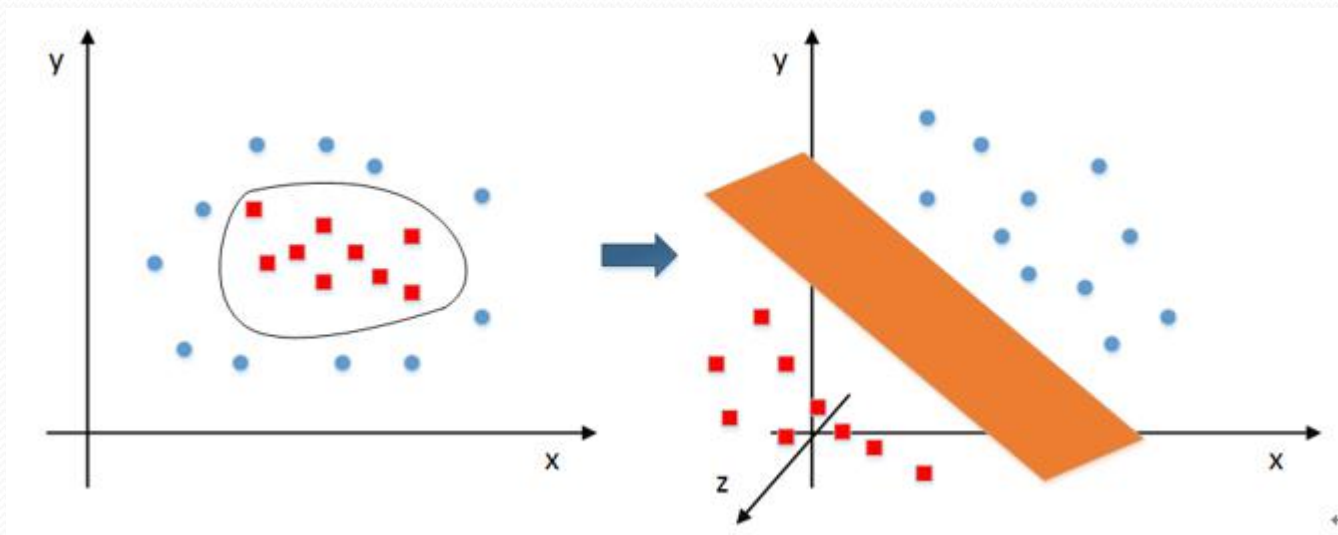
**解决方法：**通过适当方法将其转化为一个线性可分数据集 $D'$ 。



使用映射函数 $\varphi(X)$ 作用于 $D$ 中所有样本数据，将 $D$ 映射到高维空间，使得 $D$ 在高维空间的映像 $D' = \varphi(D)$ 满足线性可分性，即有：

$$D' = \{(\varphi(X_1), y_1), (\varphi(X_2), y_2), \dots, (\varphi(X_n), y_n)\}$$

**关键：**映像函数 $\varphi(X)$ 的构造。



# 三、结构风险分析

训练样本集:  $S = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\}$

SVM模型的经验风险:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{k=1}^n L(y_k, f(X_k))$$

训练样本数足够多时: 此时经验风险大体能够代表泛化误差, 使用经验风险来代替泛化误差作为模型泛化性能的度量指标;



### 三、结构风险分析

- 训练样本数目较少时：经验风险和泛化误差之间通常会有较大差别，采用结构风险。
- 结构风险：经验风险+置信风险。
- 模型 $f$ 结构风险定义为：

$$R_{\text{srm}}(f) = R_{\text{emp}}(f) + \alpha\lambda(f)$$

其中 $\lambda(f)$ 表示模型的复杂程度， $\alpha\lambda(f)$ 可以称为置信风险。

- 置信风险与模型复杂程度成正比：模型越复杂，则其泛化能力通常越弱，此时置信风险就越大；
- 置信风险与训练样本数成反比：训练样本较多时，模型泛化能力较强，置信风险则较小

- 1、SVM模型优化训练采用的是基于结构风险最小的优化策略。
- 2、通过对支持向量分类间隔最大化的方式最小化置信风险，由此获得泛化能力较强的SVM模型。
- 3、对于训练样本集线性不可分的情形，由于模型训练的经验风险不为0，故此时综合考虑经验风险和置信风险取值，使得经验风险和置信风险的整体取值达到最小，从而获得具有较强泛化能力的SVM模型。

**具体的思想是：**通过将决策函数候选集： $B = \{f(x, w), w \in \Omega\}$ 划分为多个子集。对于每个子集按照VC维度排列，在每个子集中寻找最小经验风险，然后在子集之间折衷考虑经验风险和置信风险之和最小，得到的泛化误差界最小。

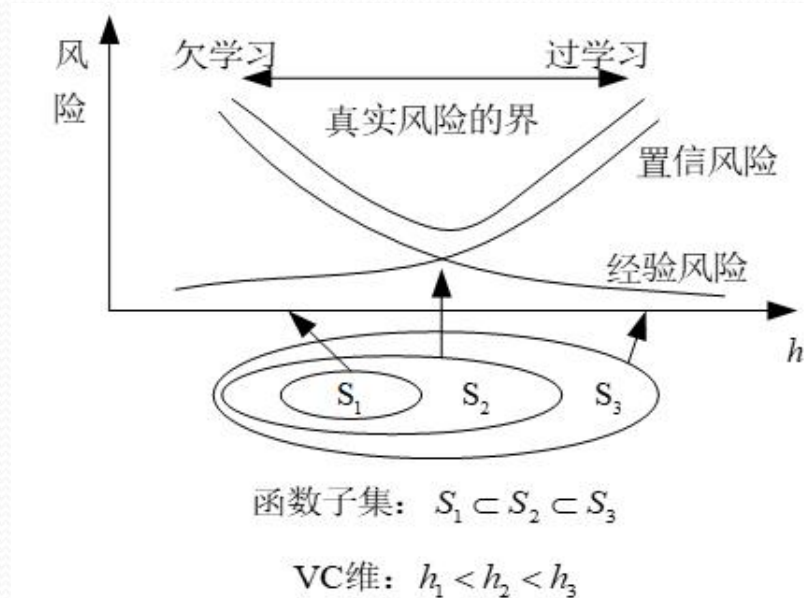


图 3-23 结构风险最小化基本原理<sup>52</sup>



## 5.4 聚类

- 5.4.1 聚类概述
- 5.4.2 划分聚类

## 5.4.1 聚类概述

- 聚类就是对一堆观测数据（对象）进行划分，使得同簇（Cluster）内的数据彼此相似，而不同簇之间不相似。
- 定义

对于观测空间 $S$ 上的数据集 $D$ （ $D \subset S$ ），求 $D$ 上的一个划分 $X = \{ x_i \mid x_i \subset D, \bigcup_i x_i = D, i = 1, 2, \dots, n \}$ ，使得 $D$ 中的任意一对数据满足：

$$\text{sim}(d_k, d_l) > \text{sim}(d_p, d_q)$$

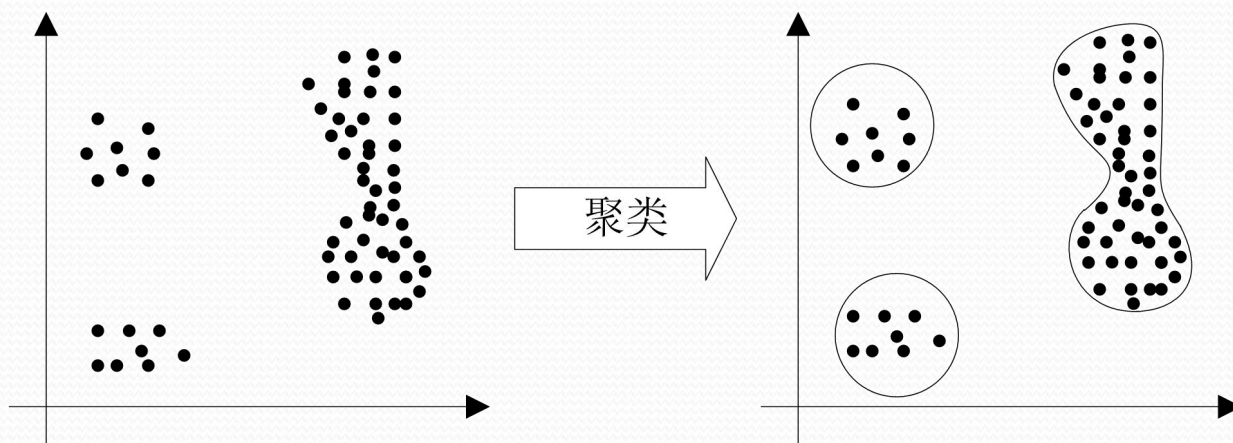
$$d_k, d_l \in x, k \neq l, \quad d_p \in y, d_q \in z, p \neq q, y \neq z, \quad x, y, z \subset D$$

其中若  $x_i \cap x_j = \Phi$ ,  $x_i, x_j \subset D$ ，即任意两个簇之间没有共享数据点，亦即一个数据点只能属于一个簇，则称之为硬聚类。否则称之为软聚类。



## 5.4.1 聚类概述

- 一般而言，我们要求同簇内数据的相似度尽可能大，不同簇间数据的相似度尽可能小。



## 5.4.1 聚类概述

- 两个不同数据点（对象）之间的相似度 $\text{sim}(d_k, d_l)$ 可以根据问题需要进行不同定义。
  - 按照距离定义
    - 距离越大则相似度越小，距离越小则相似度越大
  - 按照密度定义
    - 密度越大，则相似度越大，密度越小则相似度越小
  - 按照概念定义
    - 具有相同（或者相近）概念的数据（对象）相似度大，反之则小



# 常用距离公式

$$d(x, y) = \left( \sum_{k=1}^m |x_k - y_k|^q \right)^{\frac{1}{q}}, \quad q \geq 1$$

- 曼哈顿距离（Manhattan Distance）
  - $q=1$
- 欧几里德距离（Euclidean Distance）
  - $q=2$
- 明科夫斯基距离（Minkowski Distance）
  - $q>2$

## 5.4.1 聚类概述

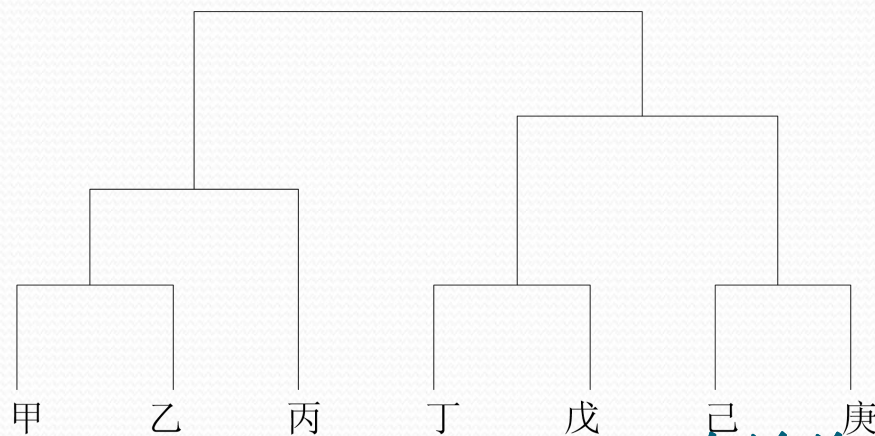
- 常用的聚类方法
  1. 分层聚类
  2. 划分聚类
  3. 基于密度的聚类
  4. 基于网格的聚类
  5. 基于模型的聚类
- 聚类研究中面临的主要问题
  - 如何降低高维、海量数据集上聚类算法的时间复杂度。
  - 如何有效定义数据之间的相似度。
  - 如何解释聚类结果。



## 一、分层聚类方法

## ● 基本思想

- 在聚类过程中生成一个聚类树。完整聚类树的最顶端代表把整个数据集划作为一个簇；最底端代表把数据集中每一个数据都当作一个簇；树中父节点对应的簇包含着所有子节点对应的簇。
- 聚类树的不同层次可以表示聚类的不同粒度。



# 常用的分层聚类算法

- Linkage算法
  - 只能聚类凸集数据，其时间复杂度为 $O(N^2)$ ，其中 $N$ 为数据个数。
- CURE算法
  - 可以聚类任意形状的数据集，但是不能处理具有分类属性的数据。
- CHAMELEON算法
  - 可以聚类任意形状的数据集
- BIRCH算法
  - 最少只扫描一遍数据集，时间复杂度为 $O(N)$ 。
  - 所以非常适合于大规模数据的聚类。
  - 但是难以聚类非凸数据集。



# 分层聚类的特点

- 在聚类过程中一次性就建好了聚类树，没有回溯调整操作。
  - 一个数据点一旦属于每个簇之后就一直属于该簇，不会更改。
  - 一个簇一旦被合并或者分裂之后，也不会再调整其中的数据点了。
- 优点：
  - 算法简单，适用性强，数据扫描顺序对聚类结果无影响，不用担心组合数目的不同选择。
- 缺点：
  - 没有全局优化，如果某一步没有很好地合并或者分裂，则必将导致低质量的聚类结果。
- BIRCH算法实际上结合了分层聚类和划分聚类的一些特点，对最终聚类结果进行了适当修整和优化。

## 二、基于密度的聚类方法

- 基本思想

- 将簇看作是数据空间中被低密度区域分割开的高密度区域。
- 只要邻近区域的密度（对象或数据点的数目）超出了某个阈值，就继续聚类。
- 即，对于给定数据集，在一个给定范围的区域中必须至少包含某个数目的点。
- 这样的方法可以用来过滤“噪声”孤立点数据，发现任意形状的簇。



# 常见的基于密度的聚类方法

- DBSCAN算法
  - 依赖于邻域半径和密度阈值两个参数。
  - 但是这两个参数并不易确定最优值。
- OPTICS算法
  - 通过一系列的邻域半径来控制簇生长。
- DENCLUE算法
  - 用密度分布函数来聚类。

## 三、基于网格的聚类方法

- 基本思想

- 把数据空间量化为有限数目的单元，形成一个网格结构。
- 所有的聚类操作都在这个网格结构（即量化的空间）上进行。

- 主要优点

- 处理速度快，
  - 其处理时间主要与量化空间每一维上的网格（单元）数目有关。
- 所以聚类高维、海量数据时，往往使用这种思想。



# 常见的基于网格的聚类方法

- 有STING算法
  - 利用存储在网格单元中的统计信息聚类。
- WaveCluster算法
  - 利用小波变换方法聚类。
- CLIQUE算法
  - 结合网格法和密度法在子空间中进行聚类。

## 5.4.2 划分聚类

- 基本思想
  - 首先把数据集划分为 $k$ 个簇。
  - 然后逐一把数据点放入合适的簇中。
  - 为了达到全局优化，算法需要重复扫描数据集多次。
- 划分聚类方法大多数使用距离定义数据相似度。
- 一个点 $x$ 到一个簇 $C$ 的距离，
  - 选取一个点 $y$ 作为簇 $C$ 的代表，然后计算 $x$ 和 $y$ 两点间的距离就是点 $x$ 到簇 $C$ 的距离。



## 5.4.2 划分聚类

- K平均（K-means）方法
  - 簇的代表点是簇的理论中心（centroid）。
  - 理论中心点不一定是簇内真实存在的数据点。
- K代表点（K-medoids）方法
  - 簇的代表点是簇内最靠近理论中心的数据点（即最有代表性的数据点）。

# 一、K平均（K-means）聚类方法

- 基本过程：
  - 第一步 从数据集中选择K个数据点作为初始簇代表点；
  - 第二步 数据集中每一个数据点按照距离，被分配给与其最近的簇；
  - 第三步 重新计算每个簇的中心，获得新的代表点；
  - 第四步 如果所有簇的新代表点均无变化，则算法结束；否则转至第二步。
- 簇理论中心常用算术平均公式计算：

$$r = \frac{1}{|C_i|} \sum_{x \in C_i} x$$



图4-1是对具有两个属性特征 $X_1, X_2$ 的某示例样本数据集进行聚类的效果，图中取聚类簇数 $k = 3$ 且每个聚簇的聚类中心坐标值为该簇中所有示例样本特征的均值

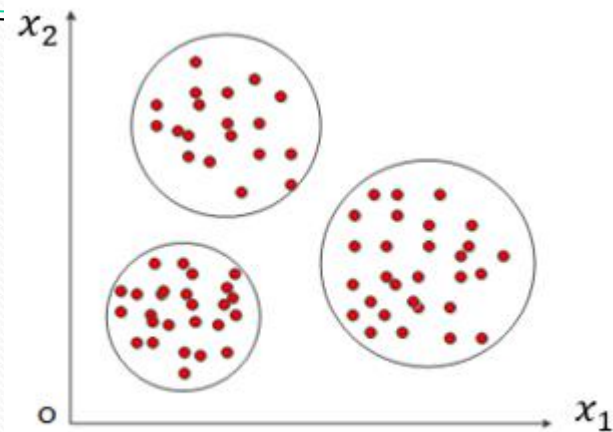


图 4-1 样本示例的聚类效果

图4.2是K-均值算法从选择初始聚类中心经过迭代到收敛的过程。

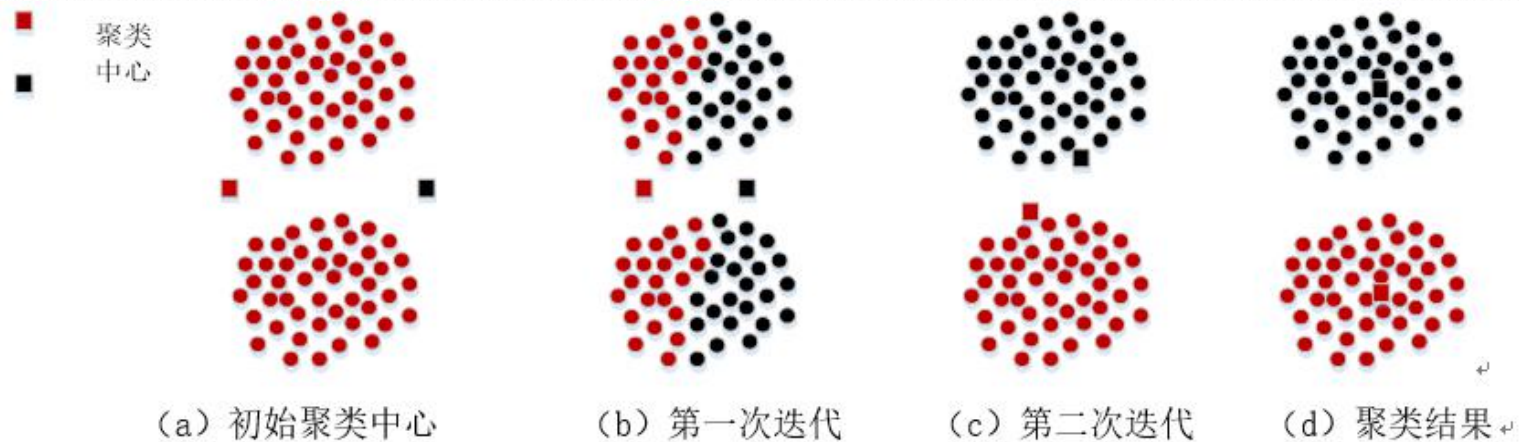


图 4-2  $k$ -均值算法的收敛过程

## 二、K代表点聚类方法

- 与K平均聚类方法的过程基本相同。
- 在选择簇代表点时，
  - 计算候选代表点与簇内其它所有点间相似度之和。
  - 然后取相似度和最大的点作为簇代表点。

$$x^* = \arg \max_{C_i} \sum sim(x, x^*)$$

- 如果采用距离定义相似度，则上式就变为使其它点到代表点的距离之和最小。
- 相似度可以有各种不同定义，
  - 所以K代表点方法不局限于可度量数据，还可以处理具有分类属性的数据。



# 划分聚类的特点

- 时间复杂度与数据集大小成线性关系。
- 对于非凸集合以及簇大小相差悬殊的数据效果不好，并且数据扫描顺序会影响选择簇中心。
- K平均方法由于要求理论中心，所以只能处理可度量的数据，难以处理具有分类属性的数据。
- K代表点方法则可以处理任何数据。
- 孤立点和噪声数据对K平均方法的影响更大一些。
- 簇个数 $k$ 对于划分聚类方法很重要。
  - 如何确定最优 $k$ 值仍然依赖于经验。
- 初始簇中心（代表点）对于划分聚类结果的影响很关键。
  - 随机选择的初始簇中心往往不能获得较好结果。

【例题4.1】表4-1为某机构15支足球队在2017-2018年间的积分，各队在各赛事中的水平发挥有所不同。若将球队的水平分为三个不同的层次水平，试用k-均值聚类方法分析哪些队伍的整体水平比较相近。

表4 - 1

队伍	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
赛事1	50	28	17	25	28	50	50	50
赛事2	50	9	15	40	40	50	40	40
赛事3	9	4	3	5	2	1	9	9
队伍	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	
赛事1	40	50	50	50	40	40	50	
赛事2	40	50	50	50	40	32	50	
赛事3	5	9	5	9	9	17	9	



【解】由于各队在各赛事上的发挥水平有所不同，故先对积分数据进行归一化处理，使用最小-最大标准化策略将积分数据映射到[0,1]区间内，具体计算公式为：

$$a'_i = \frac{a_i - \min(a_i)}{\max(a_i) - \min(a_i)}$$

$\min(a_i)$ 和 $\max(a_i)$ 分别表示第*i*个属性值 $a_i$ 在所有球队中的最小值和最大值。使用上述公式对表4-1中数据进行归一化计算。

由于需将球队分为3个层次水平，故取聚类的簇数 $k = 3$ 。通过随机采样选择编号为2、13、15的三支队伍所对应数据点作为初始聚类中心，即三个簇的聚类中心分别为： $\mu_1 = (0.3, 0, 0.19)$ ,  $\mu_2 = (0.7, 0.76, 0.5)$ ,  $\mu_3 = (1, 1, 0.5)$

队伍	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
赛事1	1	0.3	0	0.24	0.3	1	1	1
赛事2	1	0	0.15	0.76	0.76	1	0.76	0.76
赛事3	0.5	0.19	0.13	0.25	0.06	0	0.5	0.5
队伍	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	
赛事1	0.7	1	1	1	0.7	0.7	1	
赛事2	0.76	1	1	1	0.76	0.68	1	
赛事3	0.25	0.5	0.25	0.5	0.5	1	0.5	

计算每个数据  
点到聚类中心  
的欧氏距离

队伍	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
$\mu_1$	1.2594	0	0.3407	0.7647	0.7710	1.2354	1.0787	1.0787
$\mu_2$	0	0.9131	0.9995	0.5235	0.5946	0.6306	0.3000	0.3000
$\mu_3$	0.3407	1.2594	1.3636	0.8353	0.8609	0.5000	0.2400	0.2400
队伍	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	
$\mu_1$	0.8609	1.2594	1.2221	1.2594	0.9131	1.1307	1.2594	
$\mu_2$	0.2500	0.3842	0.4584	0.3842	0	0.5064	0.3842	
$\mu_3$	0.4584	0	0.2500	0	0.3842	0.6651	0	

分别将每个数据点分配到聚类中心与其距离最近的簇中，得到第一次聚类结果为：

$$C_1 = \{X_2, X_3\}; C_2 = \{X_4, X_5, X_9, X_{13}, X_{14}\};$$

$$C_3 = \{X_1, X_6, X_7, X_8, X_{10}, X_{11}, X_{12}, X_{15}\}$$

根据上述第一次聚类结果，对聚类中心做调整。对于 $C_1$ ，有：

$$\mu'_1 = \left( \frac{0.3 + 0}{2}, \frac{0.15 + 0}{2}, \frac{0.19 + 0.13}{2} \right) = (0.15, 0.075, 0.16)$$

同理可将第二个簇 $C_2$ 和第三个簇 $C_3$ 的聚类中心进行调整，分别得到

$$\mu'_2 = (0.528, 0.744, 0.412), \mu'_3 = (1, 0.94, 0.40625)$$



计算各数据点与更新后的聚类中心的距离。

队伍	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
$\mu'_1$	1.3014	0.1704	0.1704	0.6967	0.7083	1.2664	1.1434	1.1434
$\mu'_2$	0.5441	0.8092	0.8443	0.3308	0.4197	0.6768	0.4804	0.4804
$\mu'_3$	0.1113	1.1918	1.3040	0.7965	0.8014	0.4107	0.2030	0.2030
队伍	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	
$\mu'_1$	0.8831	1.3014	1.2595	1.3014	0.9420	1.1722	1.3014	
$\mu'_2$	0.2368	0.5441	0.5609	0.5441	0.1939	0.6160	0.5441	
$\mu'_3$	0.3832	0.1113	0.1674	0.1113	0.3622	0.7142	0.1113	

得到第二次聚类结果如下： $C_1 = \{X_2, X_3\}$ ； $C_2 = \{X_4, X_5, X_9, X_{13}, X_{14}\}$ ； $C_3 = \{X_1, X_6, X_7, X_8, X_{10}, X_{11}, X_{12}, X_{15}\}$

聚类结果并未发生变化，故聚类中心收敛，停止迭代。

由上述聚类结果可知， $X_2$ 、 $X_3$ 两支球队的整体水平比较相近， $X_4, X_5, X_9, X_{13}, X_{14}$ 的整体水平比较相近，其余球队的整体水平比较相近

## 5.5 进化计算

- 5.5.1 达尔文进化算法
- 5.5.2 遗传算法



## 5.5 进化计算

- ❑ 进化计算(evolutionary computation)是研究利用自然进化和适应思想的计算系统。
- ❑ 达尔文进化论是一种稳健的搜索和优化机制，对计算机科学，特别是对人工智能的发展产生了很大的影响。
- ❑ 大多数生物体是通过自然选择和有性生殖进行进化。自然选择决定了群体中哪些个体能够生存和繁殖，有性生殖保证了后代基因中的混合和重组。自然选择的法则是适应者生存，不适应者被淘汰,简言之称为优生劣汰。

## 5.5.1 达尔文进化算法

- 1) 建立原始种体。
- 2) 通过突变建立子孙。

$$s'_1 = sg_1$$

$$x'_1 = x + s'_1 Z_1$$

$$s'_\lambda = sg_\lambda$$

$$x'_\lambda = x + s'_\lambda Z_\lambda$$

- 3) 选择：

$$Q(x) = \max_{1 \leq i \leq \lambda} \{Q(x'_i)\}$$

- 4) 返回到步骤(1)。



## 5.5.2 遗传算法

- 遗传算法由美国Michigan大学J.H.Holland于60年代提出，是模仿生物遗传学和自然选择机理，通过人工方式所构造的一类优化搜索算法，是对生物进化过程进行的一种数学仿真，是进化计算的最重要的形式。
- 进化计算和遗传算法借鉴了生物科学中的某些知识，这也体现了人工智能这一交叉学科的特点。

# 生物进化与遗传算法之间的对应关系

生物进化中的概念	遗传算法中的作用
环境	适应函数
适应性	适应值函数
适者生存	适应函数值最大的解被保留的概率最大
个体	问题的一个解
染色体	解的编码
基因	编码的元素
群体	被选定的一组解
种群	根据适应函数选择的一组解
交叉	以一定的方式由双亲产生后代的过程
变异	编码的某些分量发生变化的过程



# 一、遗传算法的基本机理

- 霍兰德的遗传算法通常称为简单遗传算法(SGA)。现以此作为讨论主要对象，加上适应的改进，来分析遗传算法的结构和机理。

- ① 编码与解码
- ② 适应度函数
- ③ 遗传操作

# 一、遗传算法的基本机理

## □ (1) 编码与解码

- 将问题结构变换为位串形式编码表示的过程叫**编码**；而相反将位串形式编码表示变换为原问题结构的过程叫**解码**或译码。
- 把位串形式编码表示叫**染色体** (chromosome)，有时也叫**个体**。

“甲”的编码：0100**1**11010100011
- 染色体的每位称为**基因** (gene)。

“甲”染色体的第5个基因取值为**1**。



# 一、遗传算法的基本机理

## □ (2) 适应度函数

- 为了体现染色体的适应能力，引入了对问题中的每一个染色体都能进行度量的函数，叫**适应度函数**(fitness function)。
- 适应度函数要有效反映每一个染色体与问题的最优解染色体之间的差距。适应度函数的取值大小与求解问题对象的意义有很大的关系。
- TSP的目标是路径总长度为最短，自然地，由路径总长度就可**导出**TSP问题的适应度函数。

$$f(w_1, w_2, \dots, w_n) = \frac{1}{\sum_{j=1}^n d(w_j, w_{j+1})}$$

大连海事大学

### □ (3) 遗传操作:选择、交叉、变异

#### ◆ 1) 选择(selection)

- 根据适者生存原则选择下一代的个体。在选择时，以适应度为选择原则。适应度准则体现了适者生存，不适应者淘汰的自然法则。
- 给出目标函数 $f$ ，则 $f(b_i)$ 称为个体 $b_i$ 的适应度。通常选中 $b_i$ 为下一代个体的次数可用下式来计算。

$$P\{\text{选中 } b_i\} = \frac{f(b_i)}{\sum_{j=1}^n f(b_j)} \times n$$

#### ◆ 2) 交叉(crossover)将两个个体的部分编码进行交换。

10001110		10001001
11011001	→	11011110

Diagram illustrating crossover: Two parent binary strings (10001110 and 11011001) are shown. A vertical double-headed arrow indicates a swap of the third and fourth bits. The resulting offspring strings are 10001001 and 11011110.

#### ◆ 3) 变异(mutation)改变某位的值。

10100110	→	10110110
----------	---	----------

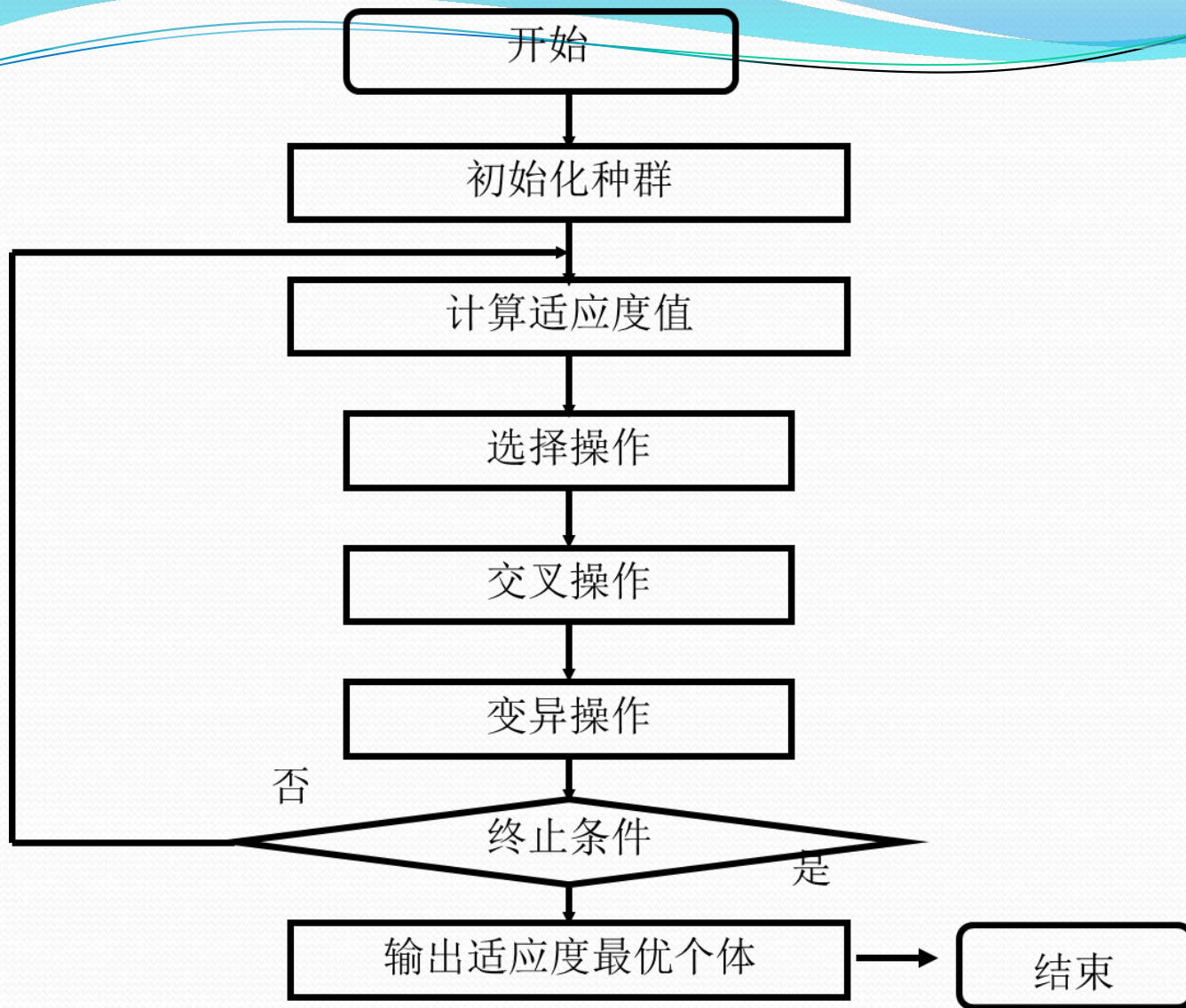
Diagram illustrating mutation: A parent binary string (10100110) is shown. A vertical double-headed arrow indicates a flip of the third bit. The resulting offspring string is 10110110.



## 二、遗传算法的求解步骤

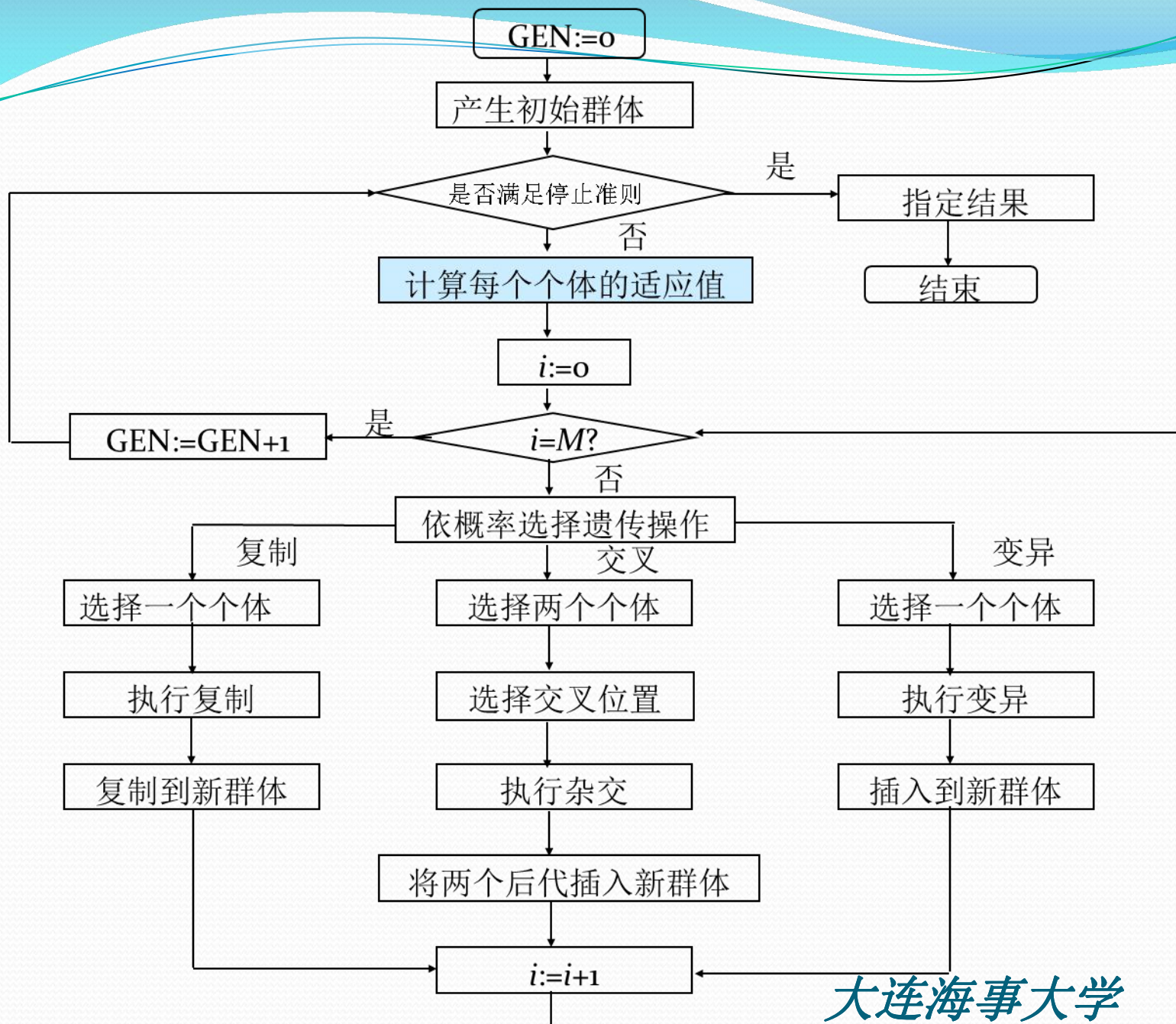
遗传算法(SGA)按如下过程进行：

- (1) 对待解决问题进行编码；
- (2) 随机初始化群体 $X(0):=(x_1, x_2, \dots, x_n)$ ；
- (3) 对当前群体 $X(t)$ 中每个个体 $x_i$ 计算其适应度 $F(x_i)$ ，适应度表示了该个体的性能好坏；
- (4) 应用选择算子产生中间代 $X_r(t)$ ；
- (5) 对 $X_r(t)$ 应用交叉和变异算子，产生新一代群体 $X(t+1)$ ，这些算子的目的在于扩展有限个体的覆盖面，体现全局搜索的思想；
- (6)  $t:=t+1$ ；如果不满足终止条件继续(3)。



遗传算法流程图





例1 求函数 $f(x)=x^2$ 的最大值，变量 $x$ 为0~31之间的整数。

解：

- 为用GA解此问题,容易想到将决策变量 $x$ 取的值以二进位数表示,从而得到一种自然的编码;
- 每一个体均为长度是5的二进制位串。初始群体的容量取4。于是,从总体中随机抽取4个个体组成第一代群体,即初始群体。
- 交配概率 $p_c=100\%$ ,变异概率 $p_m=1\%$ 。
- 具体操作可通过掷硬币确定。

例如,将一枚硬币连续掷20次,或指定了顺序的5枚硬币各掷4次,正面为1,反面为0,得4个5位二进制字符串,不妨记为(01101)、(11000)、(01000)、(10011)。



## 第0代情况表

串 NO.	初始群体 (随机生成)	$x$	$f(x)=x^2$	$\frac{f}{\sum f}$	$\frac{nf}{\sum f}$	实际生存数 (由转盘决定)
1	01101	13	169	0.144	0.576	1
2	11000	24	576	0.492	1.968	2
3	01000	8	64	0.055	0.220	0
4	10011	19	361	0.309	1.236	1

## 第0代种群的交叉情况

串NO.	种群	配对 (随机选择)	交叉点 (随机选择)	新种群	$x$	$f(x)=x^2$
1	01101	2	4	01100	12	144
2	11000	1	4	11001	25	625
3	10000	4	2	11011	27	729
4	10011	2	2	10000	16	256

## 5.6 群体智能

- 5.6.1 蚁群算法
- 5.6.2 粒子群算法



## 5.6 群体智能

- 指无智能的个体通过合作表现出智能行为的特性，在没有集中控制且不提供全局模型的前提下，为复杂问题求解提供了基础。
- 特点
  - 分布式：能够适应当前网络环境下的工作状态；
  - 鲁棒性：没有中心的控制与数据，个体的故障不影响整个问题的求解；
  - 扩充性：个体的增加，系统的通信开销增加小；
  - 简单性：个体简单，实现也比较简单。

## 5.6.1 蚁群算法

- 蚁群算法(ant colony algorithm)是一种模拟进化算法。是由意大利学者M. Dorigo等人在对自然界中真实蚁群的集体行为的研究基础上，于1991年首先提出的。
- 蚁群算法模拟了自然蚂蚁的协作过程，用一定数目的蚂蚁共同求解，用蚂蚁的移动线路表示所求问题的可行解集，通过正反馈、分布式协作和隐并行性找最优解。
- 蚁群算法已成功应用于求解TSP 问题、任务分配问题、调度问题等组合优化问题，并取得了较好的实验结果。受其影响，蚁群系统模型逐渐引起了其它研究者的注意，并用该算法来解决一些实际问题。



# 一、蚁群算法原理

蚂蚁属于群居昆虫，个体行为极其简单，而群体行为却相当复杂。

- ◆ 协作能力：一群蚂蚁很容易找到从蚁巢到食物源的最短路径，而单个蚂蚁则不能。
- ◆ 自适应能力：例如在蚁群的运动路线上突然出现障碍物时，它们能够很快地重新找到最优路径。

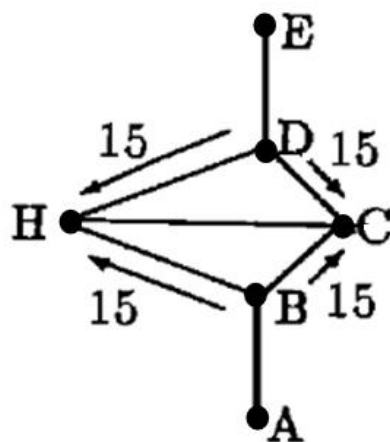
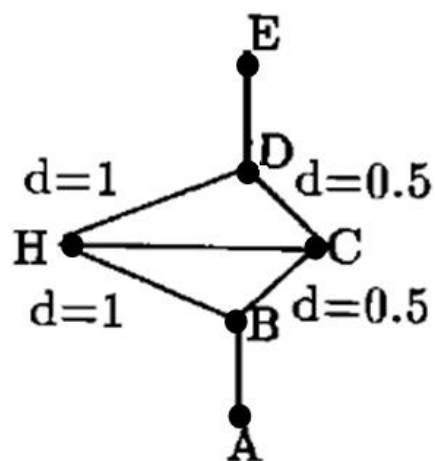
仿生学家经过大量细致观察研究发现，蚂蚁个体之间是通过一种称之为**信息素(pheromone)**的物质进行信息传递，从而能相互协作，完成复杂的任务。蚁群之所以表现出复杂有序的行为，个体之间的信息交流与相互协作起着重要的作用。

M. Dorigo 用下面的例子来具体说明蚁群系统的原理。

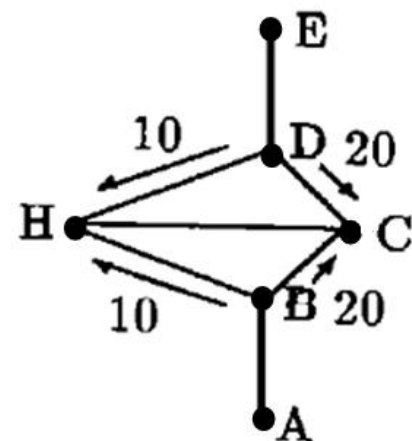
设A是巢穴，E是食物源，HC为障碍隘物。各点之间距离如图所示。假设每个时间单位有30只蚂蚁由A到达B，有30只蚂蚁由E到达D点，蚂蚁过后留下的激素物质质量为1，设该物质停留时间为1。

(1) 初始时，路径上均无信息存在，蚂蚁等概率选择路径。

(2) 经过一个时间单位后，在路径BCD上的信息量是路径BHD上信息量的二倍。将有更多的蚂蚁选择路径BCD。



$t=0$  时刻



$t=1$  时刻



## 二、蚁群系统模型

蚁群算法基本原理吸收了生物界中蚂蚁群体行为的某些显著特征：

- ◆ 能察觉小范围区域内状况并判断出是否有食物或其他同类的信息素轨迹；
- ◆ 能释放自己的信息素；
- ◆ 所遗留的信息素数量会随时间而逐步减少。

蚁群算法的核心有三条：

- (1) 选择机制：信息素越多的路径，被选中的概率越大；
- (2) 信息素更新机制：路径越短，信息素增加越快；
- (3) 协作机制：个体之间通过这种信息素进行交流。

例：求解 $n$  个城市的TSP问题。

假设将 $m$ 只蚂蚁随机放入到 $n$ 个城市中,为模拟实际蚂蚁的行为，首先引进如下记号：

$d_{ij}$ ：表示城市 $i$ 和城市 $j$  之间的距离，  $i, j= 1, 2, \cdots, n$

$\tau_{ij}(t)$ ：表示 $t$ 时刻在 $(i, j)$ 连线上残留的信息量。初始时刻，各条路径上信息量相等，设 $\tau_{ij}(0) = C$  ( $C$  为常数)

$\alpha$ ：表示路径上信息量的相对重要性( $\alpha \geq 0$ )

$\eta_{ij}$ ：表示由城市 $i$  转移到城市 $j$  期望值，可根据某种启发式算法具体确定。对于TSP问题一般可取 $\eta_{ij}=1/d_{ij}$

$\beta$ ：表示期望值的相对重要性( $\beta \geq 0$ )

$p_{ij}^k(t)$ ：表示在 $t$ 时刻蚂蚁 $k$ 由位置 $i$  转移到位置 $j$  的概率：

$\Delta\tau_{ij}^k$ ：表示第 $k$  只蚂蚁在本次循环中留在路径 $(i, j)$ 上的信息量：



采用蚁群方法进行求解TSP的主要步骤可叙述如下：

- ①  $nc \leftarrow 0$  ( $nc$  为迭代步数或搜索次数)；各  $\tau_{ij}$  和  $\Delta\tau_{ij}$  的初始化；将  $m$  个蚂蚁置于  $n$  个顶点上；
- ② 将各蚂蚁的初始出发点置于当前  $tabu_k$  中；对每个蚂蚁  $k$  按概率  $P_{ij}^k$  移至下一顶点  $j$ ；将顶点  $j$  置于当前解集；

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \eta_{is}^\beta} & j \in allowed_k \\ 0 & \text{其它} \end{cases}$$

- ③ 计算各蚂蚁的目标函数值  $Z_k$  ( $k=1, \dots, m$ )；记录当前的最好解；
- ④ 按更新方程修改信息强度；

$$\tau_{ij}(t+n) = \rho \tau_{ij}(t) + \Delta \tau_{ij}$$
$$\Delta \tau_{ij} = \sum_{k=1}^{k=m} \Delta \tau_{ij}^k$$
$$\Delta \tau_{ij}^k = \begin{cases} Q/L_k & \text{若第} k \text{只蚂蚁在本次循环中经过}(i, j) \\ 0 & \text{其它} \end{cases}$$

- ⑤ 对各边弧  $(i, j)$ ，置  $\Delta\tau_{ij} \leftarrow 0$ ； $nc \leftarrow nc + 1$ ；
- ⑥ 若  $nc <$  预定的迭代次数且无退化行为(即找到的都是相同解)，则转步骤②；

## 5.6.2 粒子群算法

- 粒子群优化PSO（Particle Swarm Optimization）算法是一种基于群智能的演化计算方法，由Kennedy和Eberhart于1995年提出。该算法源于对鸟类捕食行为的模拟。
- 鸟群：

假设一个区域，所有的鸟都不知道食物的位置，但是它们知道当前位置离食物还有多远。
- PSO算法

每个解看作一只鸟，称为“粒子(particle)”，所有的粒子都有一个适应值，每个粒子都有一个速度决定它们的飞翔方向和距离，粒子们追随当前最优粒子在解空间中搜索。



- 粒子速度和位置的更新

$$v_{id}^{k+1} = wv_{id}^k + c_1 \text{rand}() (p_{id} - x_{id}^k) + c_2 \text{rand}() (p_{gbest} - x_{id}^k)$$
$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}$$

$i = 1, 2, \dots, m; d = 1, 2, \dots, D$

“惯性部分”，  
对自身运动状态  
的信任

“认知部分”，对微粒  
本身的思考，即来源于  
自己经验的部分

“社会部分”，微粒间的  
信息共享，来源于群体中  
的其它优秀微粒的经验

# 粒子群算法过程

- (1) 随机初始化粒子群，即 $t=0$ 时随机为每个粒子指定一个位置 $X_i(0)$ 及速度 $V_i(0)$ ;
- (2) 计算每个粒子的适应度值 $f(X_i(t))$ ;
- (3) 比较每个粒子的当前适应度值 $f(X_i(t))$ 和个体最优值 $f(P_i)$ ，如果 $f(X_i(t)) < f(P_i)$ ，那么 $P_i = X_i(t)$ ;
- (4) 比较每个粒子的当前适应度值 $f(X_i(t))$ 和全局最优值 $f(P_g)$ ，如果 $f(X_i(t)) < f(P_g)$ ，那么 $P_g = X_i(t)$ ;
- (5) 按下列公式更改每个粒子的速度矢量和位置：

$$\begin{cases} V_i(t+1) = wV_i(t) + c_1r_1(P_i - X_i(t)) + c_2r_2(P_g - X_i(t)) \\ X_i(t+1) = X_i(t) + V_i(t+1) \end{cases}$$

- (6) 如果满足终止条件，则输出 $P_g$ ；否则， $t=t+1$ ，转(2)。