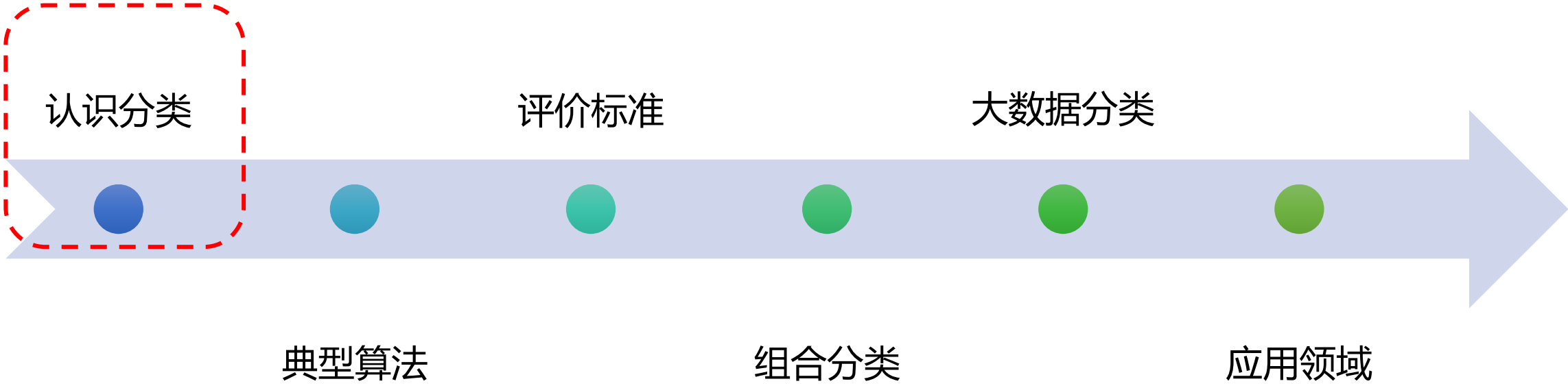
大数据分类技术介绍

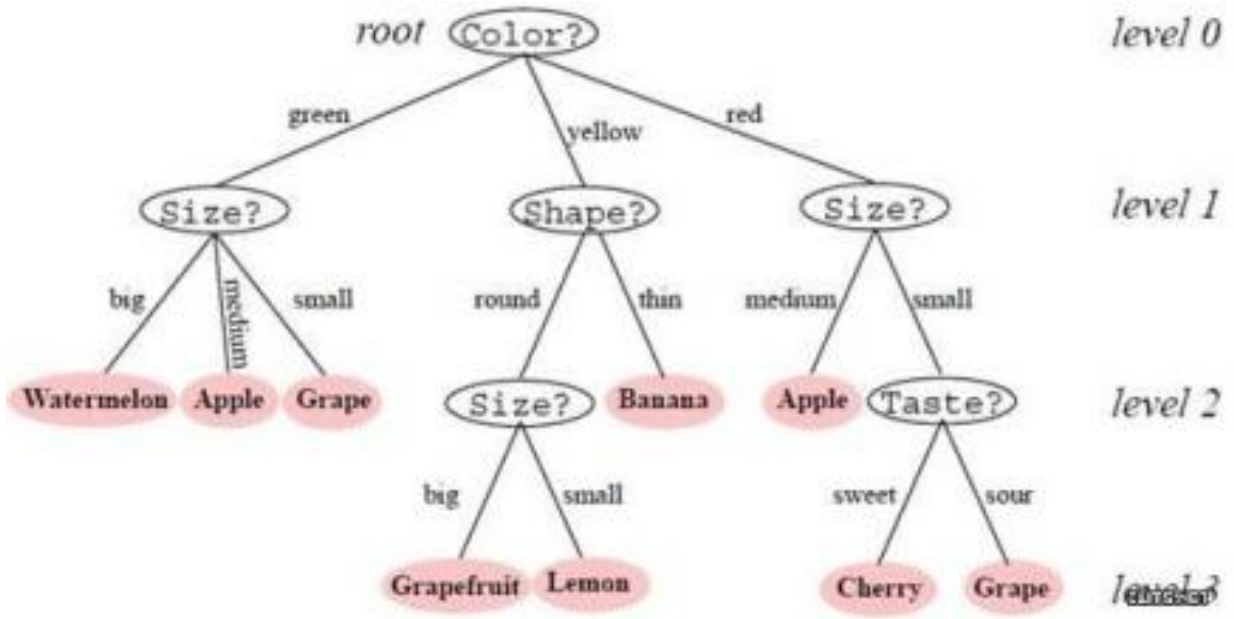
牛琨　副教授，博士生导师
北京邮电大学软件学院
2019年10月

一、认识分类



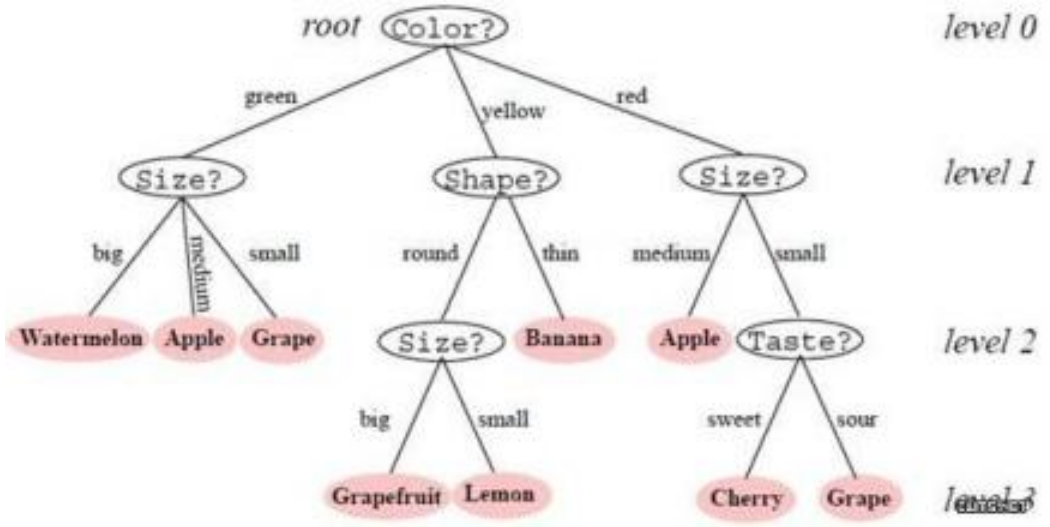
认识分类

- 分类的目的是获得一个**分类函数或分类模型**（也常常称作**分类器**），该模型能把数据库中的数据项映射到某一个给定类别。
- 分类可用于提取描述重要数据类的模型或预测未来的数据趋势。



分类和预测

- **分类**是预测分类(离散、无序的)标号
- **预测**是建立连续值函数模型



什么是分类？

- 银行贷款员需要分析数据，搞清楚哪些贷款申请者是“安全的”，银行的“风险”是什么
- 百货公司的市场经理需要数据分析，以便帮助他猜测具有某些特征的顾客是否会购买一台新的计算机
- 医学研究者希望分析乳腺癌数据，预测病人应当接受三种具体治疗方案的哪一种
- 这种类属性可以用离散值表示，其中值之间的序没有意义

什么是预测？

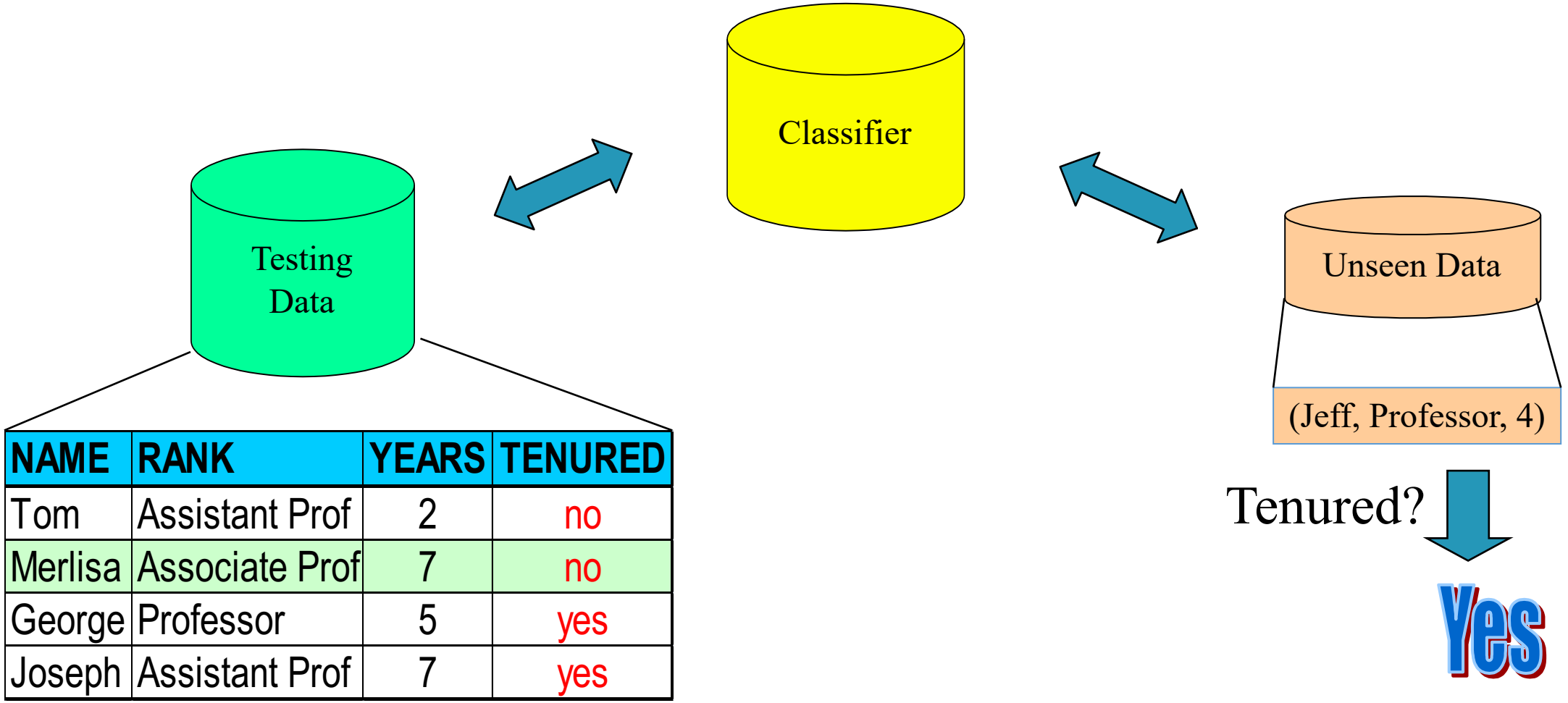
- 百货公司的市场经理希望预测一位给定的顾客在AllElectronics的一次销售期间将花费多少钱
- 该数据分析任务就是数值预测的例子，其中所构造的模型预测一个连续值函数或有序值，与类标号不同
- 这种模型是预测器(predictor)，回归分析(regression analysis)是数值预测最常用的统计方法

分类的实现

- 构建模型：预设分类类别
  - 对每个样本进行类别标记
  - 训练集构成分类模型
  - 分类模型可表示为：分类规则、决策树或数学公式

- 使用模型：识别未知对象的所属类别
  - 模型正确性的评价
    - 已标记分类的测试样本与模型的实际分类结果进行比较
    - 模型的正确率是指测试集中被正确分类的样本数与样本总数的百分比。测试集与训练集相分离，否则将出现**过拟合（over-fitting）**现象。

分类的实现—利用模型预测



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

分类与聚类

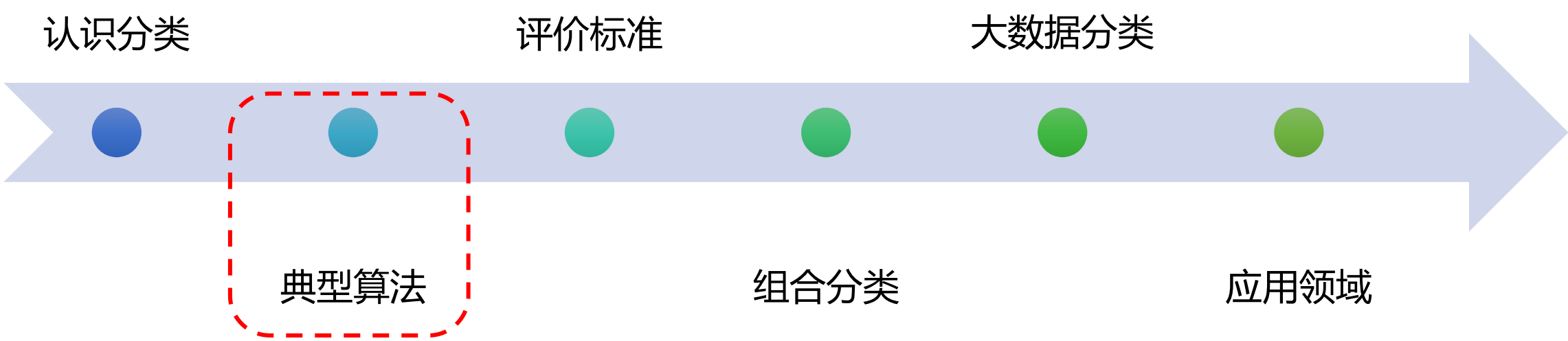
分类是有监督的学习

- Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
- New data is classified based on the training set

聚类是无监督的学习

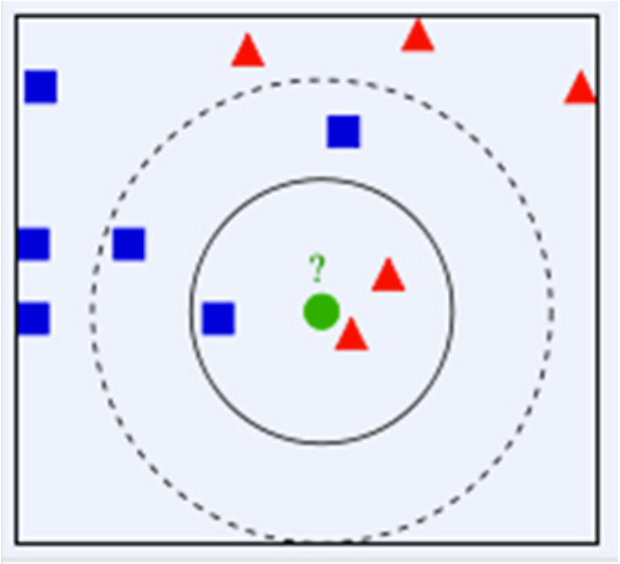
- The class labels of training data is unknown
- Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

二、典型算法



K-NN分类

- K-NN分类，即K最近邻法，最初由Cover和Hart于**1968年**提出的，是一个理论上比较成熟的方法。
- 该方法的思路非常简单直观：**如果一个样本在特征空间中的k个最相似（即特征空间中最邻近）样本中的大多数属于某一个类别，则该样本也属于这个类别。**
- 该方法在分类决策上只依据最邻近的一个或者几个样本的类别来决定待分类样本所属的类别。
- 该算法较适用于**样本容量比较大的类域的自动分类**，而那些样本容量较小的类域采用这种算法比较容易产生误分。



SVM分类

- SVM分类方法
  - 即支持向量机（Support Vector Machine）法，由Vapnik等人于1995年提出，具有相对优良的性能指标。
  - 该方法是建立在统计学习理论基础上的机器学习方法。通过学习，SVM可以自动寻找出那些对分类有较好区分能力的支持向量，由此构造出的分类器可以**最大化类与类的间隔**，因而有较好的适应能力和较高的分准率。
  - 该方法只需要由各类域的边界样本的类别来决定最后的分类结果。
  - SVM法对**小样本情况下的自动分类**有着较好的分类结果。

SVM分类算法举例

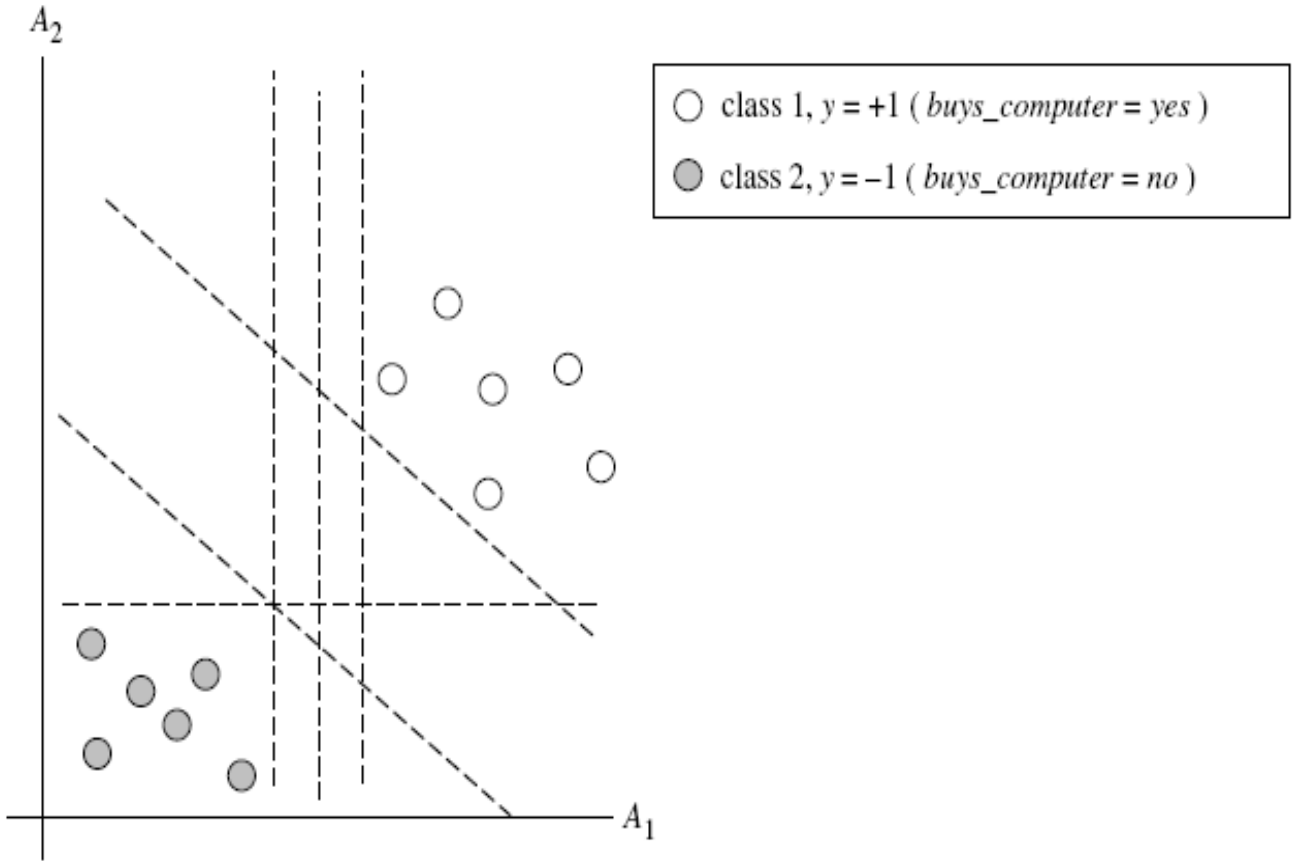


Figure 6.20 The 2-D training data are linearly separable. There are an infinite number of (possible) separating hyperplanes or "decision boundaries." Which one is best?

决策树算法

- 决策树归纳是一种经典的分类算法。
- 它采用**自顶向下、递归的、各个击破**的方式构造决策树。树的每一个结点上使用信息增益度量选择属性，可以从所生成的决策树中提取出分类规则。
- 决策树分类是用属性值对样本集逐级划分，直到一个节点仅含有同一类的样本为止。
- 决策树首先起源于Hunt等人提出的概念学习系统（Concept Learning System,CLS），然后发展到**Quinlan的ID3算法**，最后演化为能处理连续属性值的**C4.5算法**。

决策树算法

- **决策树输入**
  - 一组带有类别标记的样本
- **决策树输出**
  - 一棵二叉或多叉树。
  - 二叉树的内部节点（非叶子节点）一般表示为一个逻辑判断，如形式为(ai=vi)的逻辑判断，其中ai是属性，vi是该属性的某个属性值；树的边是逻辑判断的分支结果。
  - 多叉树（ID3）的内部节点是属性，边是该属性的所有取值，有几个属性值，就有几条边。树的叶子节点则是类别标记。

决策树算法举例

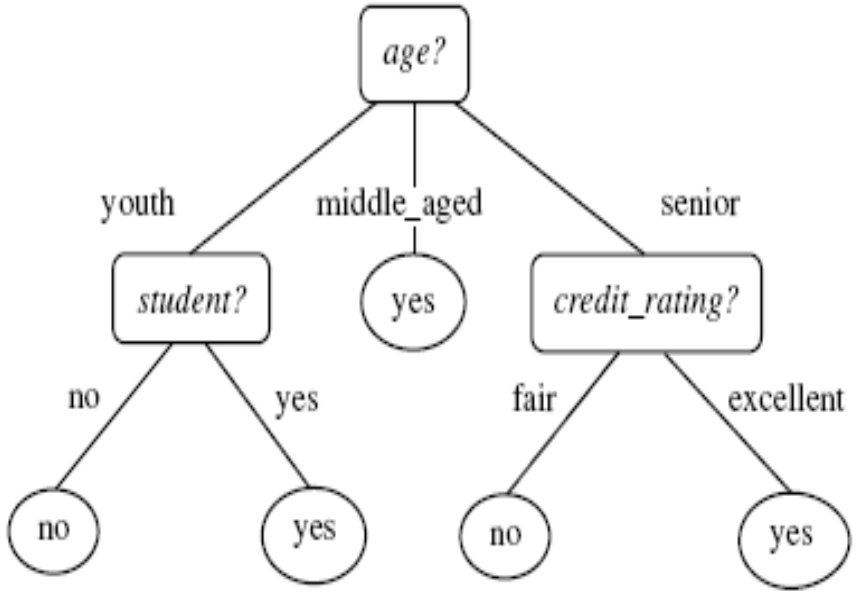


Figure 6.2 A decision tree for the concept *buys_computer*, indicating whether a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer* = *yes* or *buys_computer* = *no*).

构造决策树

- 决策树的构造采用**自上而下**的递归构造。

如果训练样本集中所有样本是同类的，则将它作为叶子节点，节点内容即是该类别标记；

否则，根据某种策略选择一个属性，按照属性的不同取值，将样本集划分为若干子集，使得每个子集上的所有样本在该属性上具有同样的属性值。

然后再依次处理各个子集。

- 实际上就是**“分而治之”（divide-and-conquer）**的策略。

决策树构造条件

- 决策树构造的条件
  - 构造好的决策树的关键是：**如何选择好的逻辑判断或属性**。
  - 对于同样一组样本，可以有很多决策树能符合这组样本。研究表明，一般情况下，树越小则树的预测能力越强。要构造尽可能小的决策树，关键在于选择恰当的逻辑判断或属性。由于构造最小的树是NP问题，因此只能采用启发式策略选择好的逻辑判断或属性。

决策树算法

- 实际中，用于模型学习的训练数据往往不是完美的，原因是：

某些属性字段上缺值（missing values）；

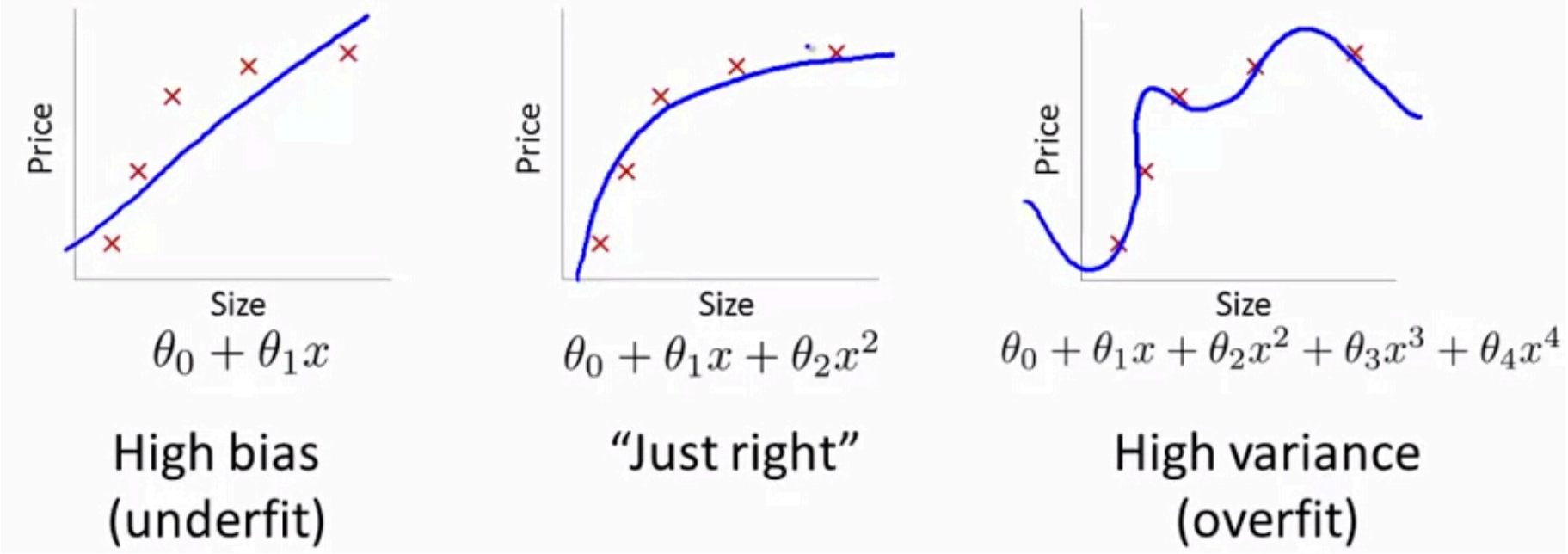
缺少必需的数据而造成数据不完整；

数据不准确含有噪声甚至是错误的。

- 此时，需要克服噪声和决策树剪枝。

决策树算法：过拟合

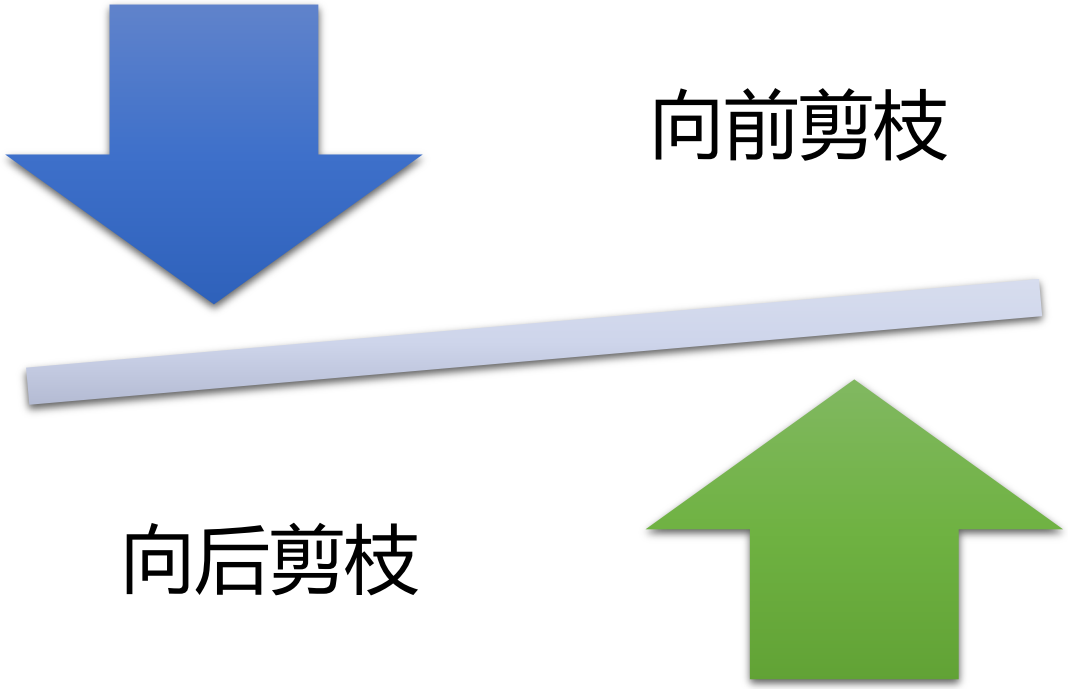
- 基本的决策树构造算法没有考虑噪声，生成的决策树完全与训练样本拟合。
- 在有噪声的情况下，完全拟合将导致**过分拟合（overfitting）**，即对训练数据的完全拟合反而不具有很好的预测性能。



决策树算法：剪枝

剪枝技术是一种克服噪声的技术，同时它也能使树得到简化而变得更容易理解。

向后剪枝（backward pruning）是一种两阶段法：拟合 - 化简（fitting-and-simplifying），首先生成与训练数据完全拟合的一棵决策树，然后从树的叶子开始剪枝，逐步向根的方向剪。



向前剪枝（forward pruning）在生成树的同时决定是继续对不纯的训练子集进行划分还是停机。

决策树算法：剪枝

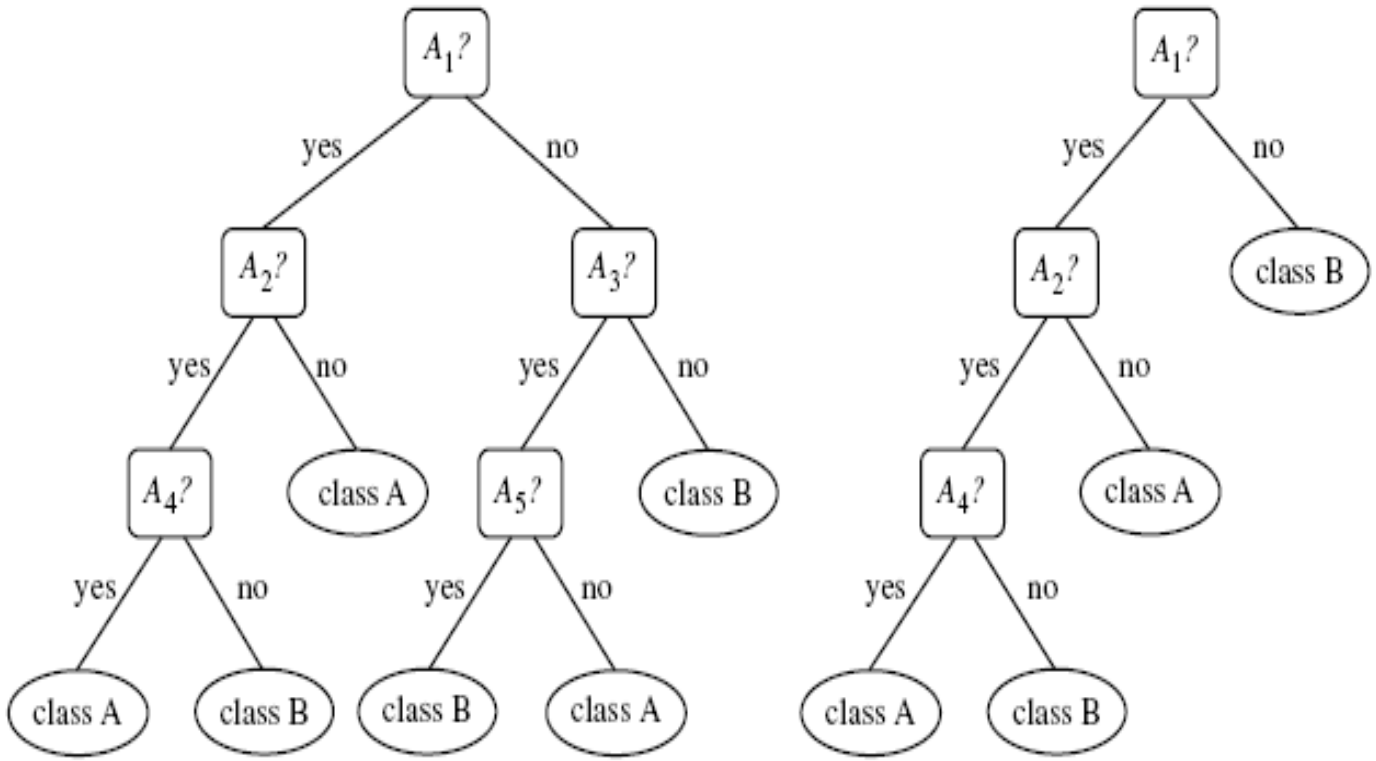


Figure 6.6 An unpruned decision tree and a pruned version of it.

剪枝局限性

- 剪枝的局限性
  - 剪枝并不是对所有的数据集都好，就像最小树并不是最好（具有最大的预测率）的树。
  - 当数据稀疏时，要**防止过分剪枝（over-pruning）**。
  - 从某种意义上而言，剪枝也是一种偏向（bias），对有些数据效果好而有些数据则效果差。

决策树举例

- **根据加薪百分比、工作时长、法定节假日、及医疗保险三个属性来判断一个企业的福利状况。**

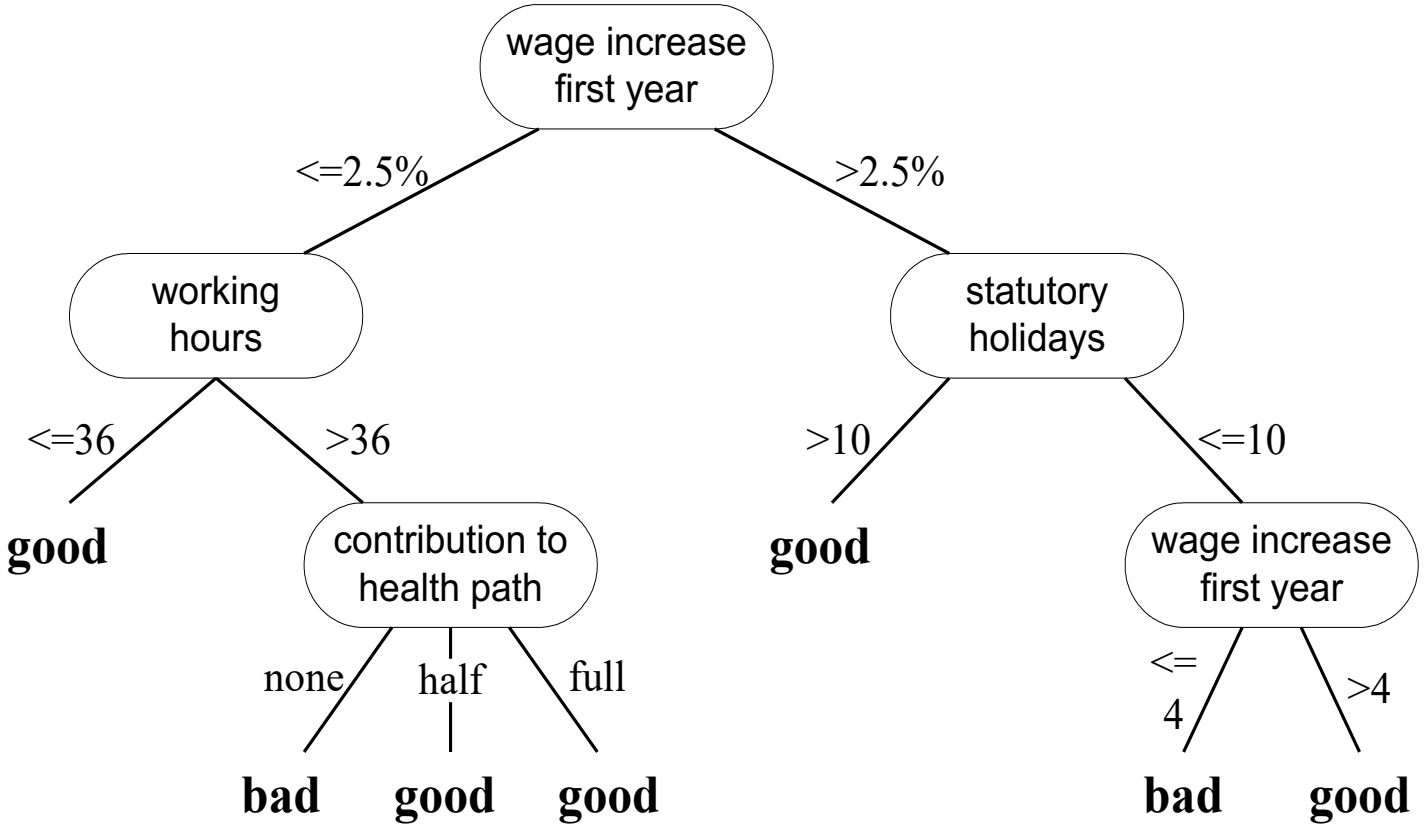


图　　公司福利条件决策树示例

ID3算法

- Quinlan提出的ID3算法，对CLS算法做出了改进。它的基本算法仍然来自于CLS，但使用熵来选择属性，效果非常理想。
- ID3使用**信息增益**作为属性选择度量，设节点N代表或存放划分D的元组，选择具有最高信息增益的属性作为节点N的分裂属性。该属性使结果划分中的元组分类所需的信息量最小。

信息增益

- 对D中的元组分类所需的期望信息由下式给出：

$$info(D) = -\sum_{i=1}^{m} p_i \, log_2(p_i)$$

- 其中，p_i是D中任意元组属于类c_i的概率，并用$|c_i, D|/|D|$估计。使用以2为底的对数函数，因为信息用二进位编码。 $Info(D)$是识别D中元组的类标号所需要的平均信息量。注意，我们这里所具有的信息只是每个类的元组所占百分比，$Info(D)$又称D的熵。

信息增益

- 假设我们要按属性A划分D中的元组，其中属性A根据训练数据的观测具有v个不同的值$\{a_1, a_2, .., a_v\}$
- 为了得到准确的分类我们还需要多少信息？这个量由下式度量：

$$info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times info(D_j)$$

其中，项$|D_j|/|D|$充当第j个划分的权重。$info_A(D)$是基于按A划分对D的元组分类所需要的期望信息，还需要的期望信息越小，划分的纯度越高。

信息增益

- 信息增益定义为原来的信息需求(即仅基于类比例)与新的需求(即对A划分之后得到的)之间的差，即：

$$Gain(A) = Info(D) - info_A(D)$$

- 换言之，$Gain(A)$告诉我们通过A的划分我们得到了多少，它是知道A的值而导致的信息需求的期望减少。选择具有最高信息增益$Gain(A)$的属性A作为节点N的分裂属性。这等价于按能做“最佳分类”的属性A划分，使得完成元组分类还需要的信息最小。

ID3算法举例

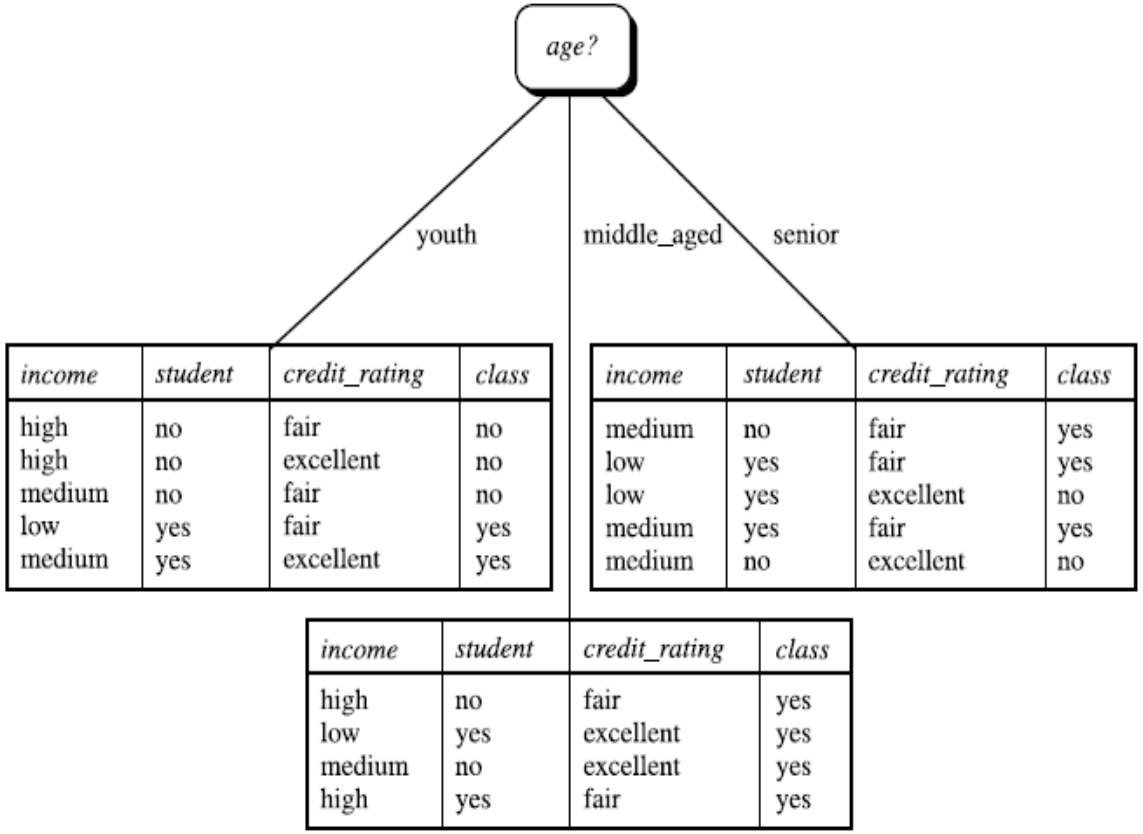
Table 6.1 Class-labeled training tuples from the *AllElectronics* customer database.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

ID3 算法举例

- $info(D) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$
- $info_{age}(D) = \frac{5}{14}\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14}\left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) + \frac{5}{14}\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.694$
- 因此，这种划分的信息增益是$Gain(age) = Info(D) - info_{age}(D) = 0.940 - 0.694 = 0.246$
- 类似地，可得：
  - $Gain(income) = 0.029$
  - $Gain(student) = 0.151$
  - $Gain(credit\_rating) = 0.048$
- 因此，age具有最大的信息增益，因此age被选为根结点并向下扩展，通过类似的方法，得到相应的ID3决策树。

ID3 算法举例



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Figure 6.5 The attribute *age* has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of *age*. The tuples are shown partitioned accordingly.

从决策树提取规则

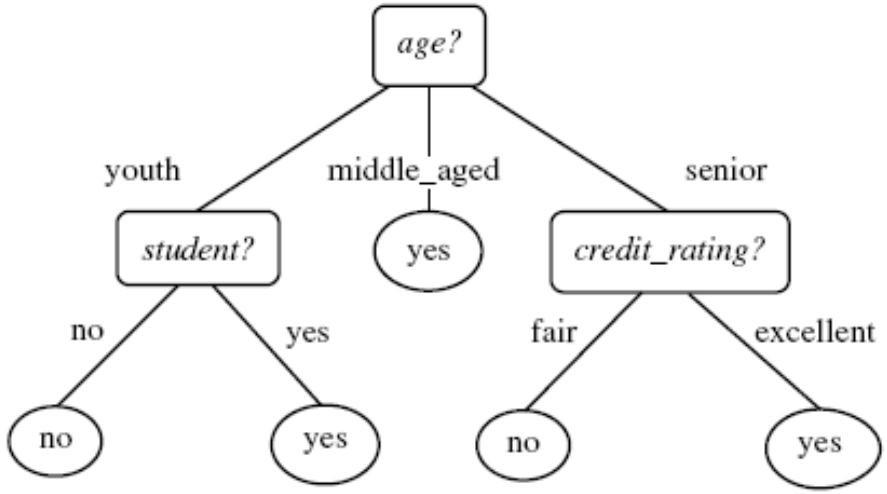


Figure 6.2 A decision tree for the concept *buys_computer*, indicating whether a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer* = *yes* or *buys_computer* = *no*).

从决策树提取规则

Extracting classification rules from a decision tree. The decision tree of Figure 6.2 can be converted to classification IF-THEN rules by tracing the path from the root node to each leaf node in the tree. The rules extracted from Figure 6.2 are

*R*1: IF *age* = *youth* AND *student* = *no* THEN *buys_computer* = *no*
*R*2: IF *age* = *youth* AND *student* = *yes* THEN *buys_computer* = *yes*
*R*3: IF *age* = *middle_aged* THEN *buys_computer* = *yes*
*R*4: IF *age* = *senior* AND *credit_rating* = *excellent* THEN *buys_computer* = *yes*
*R*5: IF *age* = *senior* AND *credit_rating* = *fair* THEN *buys_computer* = *no*

ID3算法

优势

- 算法理论清晰
- 方法简单
- 学习能力较强

不足之处

- 对较小的数据集有效
- 对噪声比较敏感
- 当数据集增大时，决策树可能会改变。

C4.5算法

- C4.5算法
  - ID3有很多改进算法，其中Quinlan在1994年开发出的C4.5算法流行最广。
- C4.5的改进主要体现在两方面：
  - 解决了**连续数据值**的学习问题；
  - 提供了将**学习结果决策树到等价规则集的转换**功能。

神经网络算法

- 人工神经网络（Artificial Neural Network，ANN）是20世纪80年代后期迅速发展起来的人工智能技术。
- 算法对**噪声数据具有很高的承受能力**，对未经训练的数据具有分类模拟的能力，因此在网站信息、生物信息和基因以及文本的数据挖掘等领域得到了越来越广泛的应用。
- 在多种ANN模型中，**反向传播（Back Propagation，BP）网络**是应用最广的一种。

神经网络算法举例

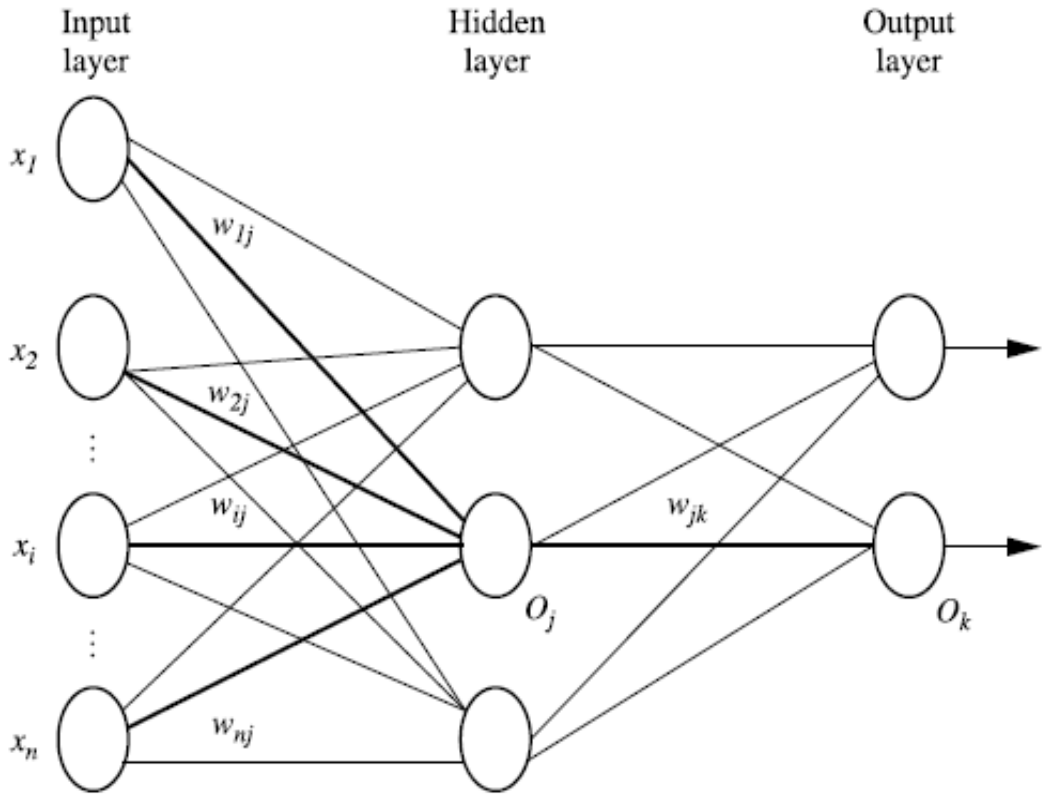
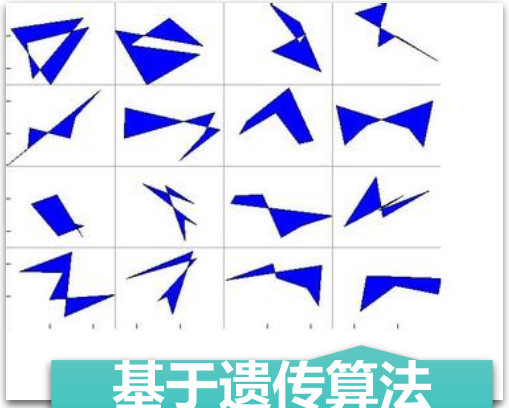


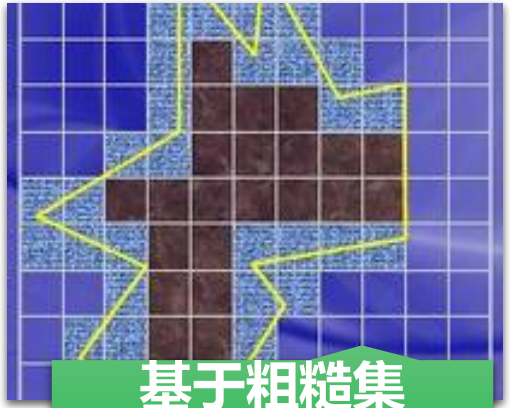
Figure 6.15 A multilayer feed-forward neural network.

其它的分类算法

基于案例推理的分类



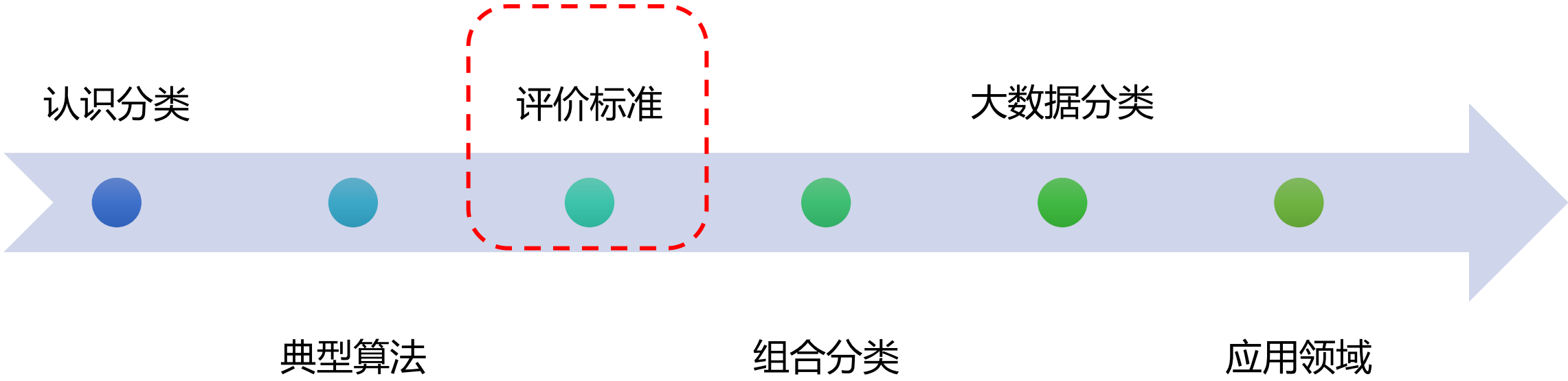
基于遗传算法的分类



基于粗糙集的分类

基于模糊集的分类

三、评价标准



分类的评价标准

- **预测的正确性**
- **时间**
  - 构建模型的时间
  - 使用模型所需的时间
- **健壮性**
  - 处理噪声及缺失值的能力
- **可扩展性**
- **可操作性**
- **规则的优化**
  - 决策树的大小
  - 分类规则的简洁性

准确性：混淆矩阵

- 准确率：$\frac{TP+TN}{P+N}$
- 错误率：$\frac{FP+FN}{P+N}$
- 召回率：$\frac{TP}{P}$
- 精度：$\frac{TP}{TP+FP}$
- F1分数：$\frac{2\times precision \times recall}{precision+recall}$

预测的类

实际的类

	yes	no	合计
yes	TP	FN	P
no	FP	TN	N
合计	P'	N'	$P+N$

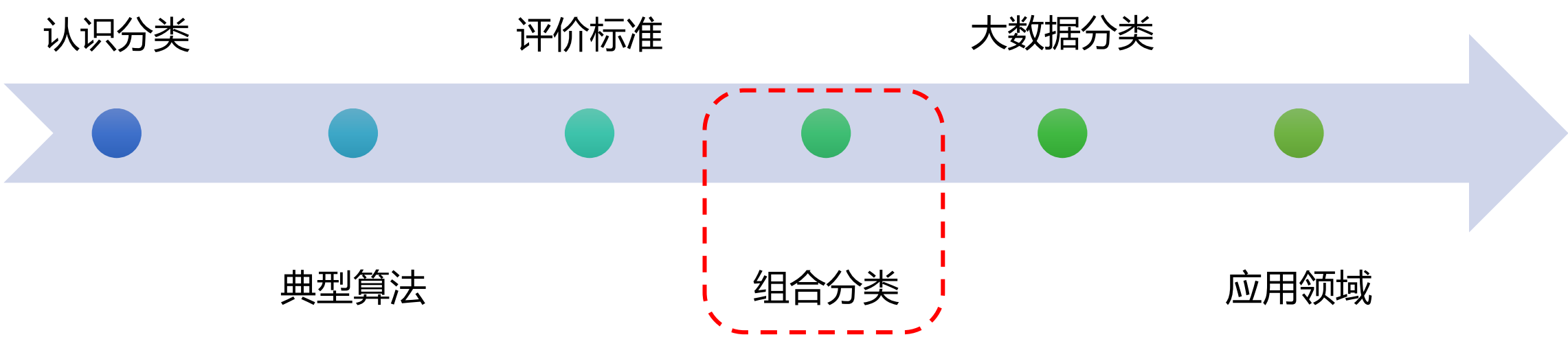
交叉验证

- **交叉验证的基本思想**
  - **数据分组**：把在某种意义下将原始数据(dataset)进行分组，一部分做为训练集(train set)，另一部分做为验证集(validation set or test set)。
  - **模型训练**：首先用训练集对分类器进行训练，再利用验证集来测试训练得到的模型(model)，以此来做为评价分类器的性能指标。
- **常见交叉验证模式**
  - Holdout验证
  - **K-fold cross-validation**
  - 留一验证

ROC曲线

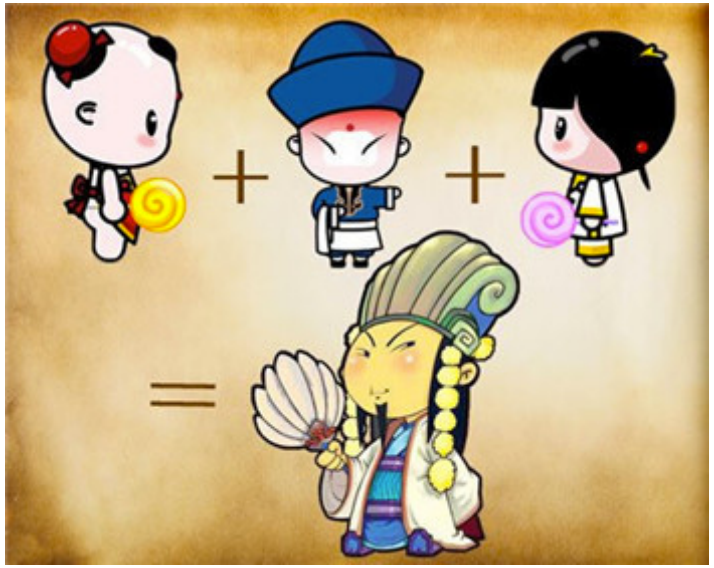
- ROC曲线是衡量分类模型效果的最重要的图形展现形式
- 绿色折线是**理想曲线（Ideal Curve）**，代表目标类别完全识别，无一漏网也无一错认
- 红色直线是**随机曲线（Random Curve）**，代表目标类别完全无法识别，随机检测
- 蓝色曲线是**性能曲线（Performance Curve）**，代表目标类别被模型识别效果，**越接近绿线性能越好**

四、组合分类

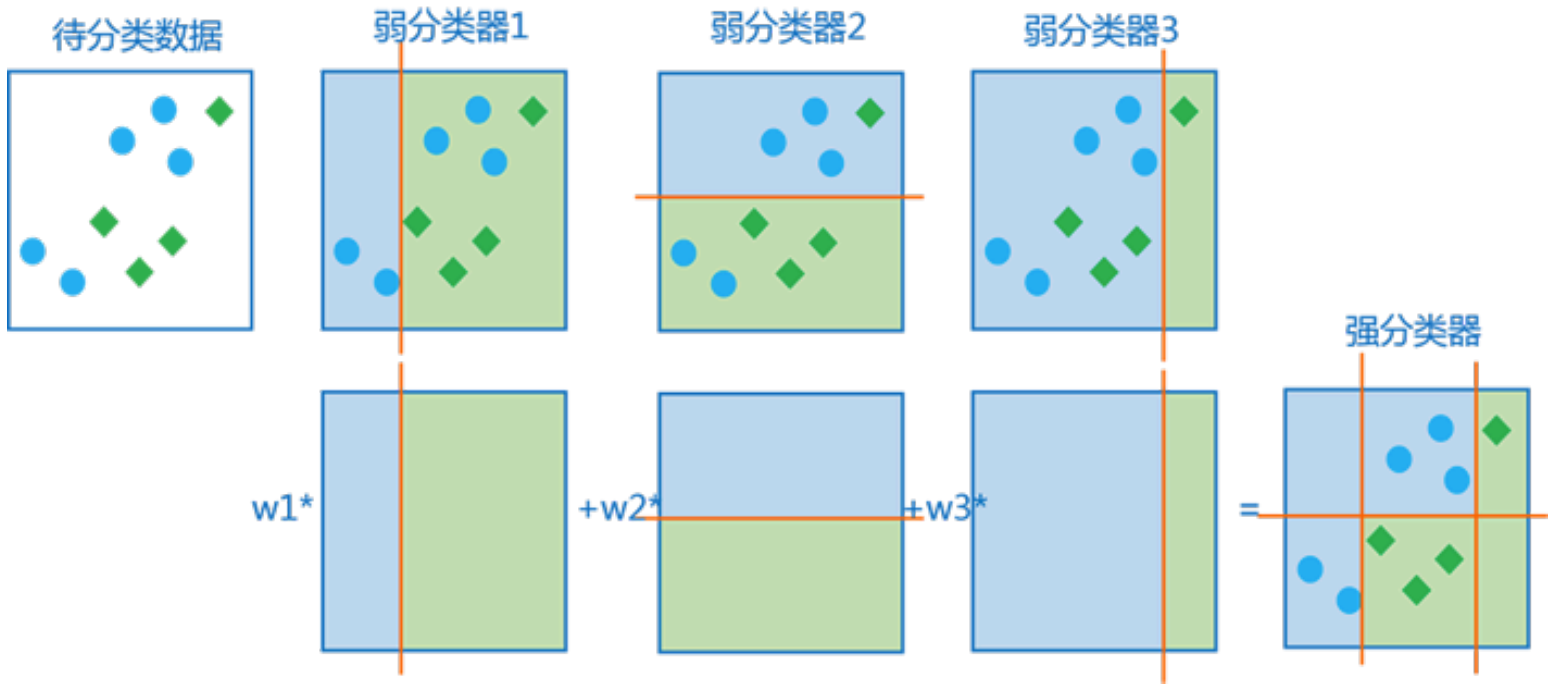


三个臭皮匠，顶过一个诸葛亮

- 模型组合与**决策树**相关的算法比较多。
- 算法最终的结果是生成**N棵树**(可能会有几百棵以上），这样可以大大减少单决策树带来的缺点。这几百棵决策树中的每一棵都很简单，但组合的力量是强大的。
- 代表算法：
  - 随机森林；
  - Boosting；
  - GBDT。



组合分类

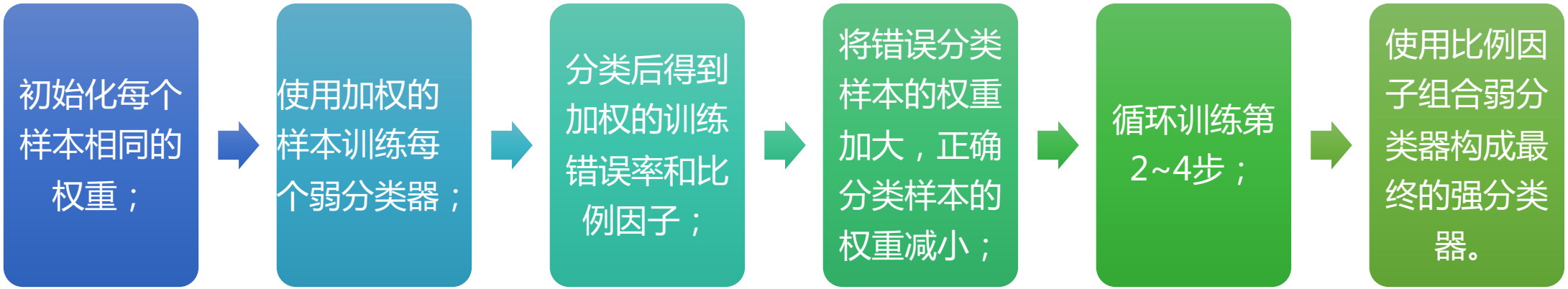


随机森林

- **随机森林，用随机的方式建立一个森林。**
- 森林里面有很多的决策树组成，随机森林的**每一棵决策树之间是没有关联的**。
- “**投票分类**”：当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别判断该样本应属哪一类，然后看看哪一类被选择最多，就预测该样本为那一类。
- 随机森林算法比喻说法：
  - 每一棵决策树就是一个精通于某一个**窄领域的研究员**；
  - 在随机森林中就有了很多个**精通不同领域的研究员**；
  - 对一个新的问题，可以用不同的角度去看待它，最终由所有研究员**投票**得到结果。

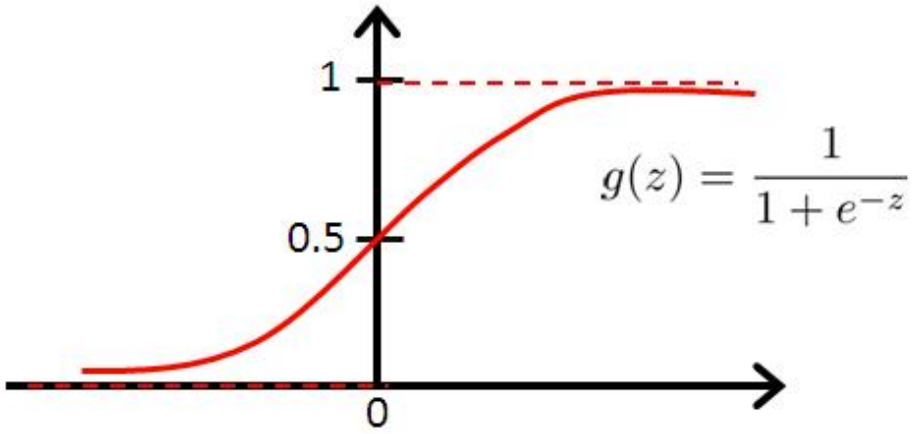
传统Boosting算法

- Boosting：提升、促进。
- 一般Boosting算法都是一个**迭代**的过程，每一次新的训练都是为了**改进上一次的结果**。
- 算法流程：

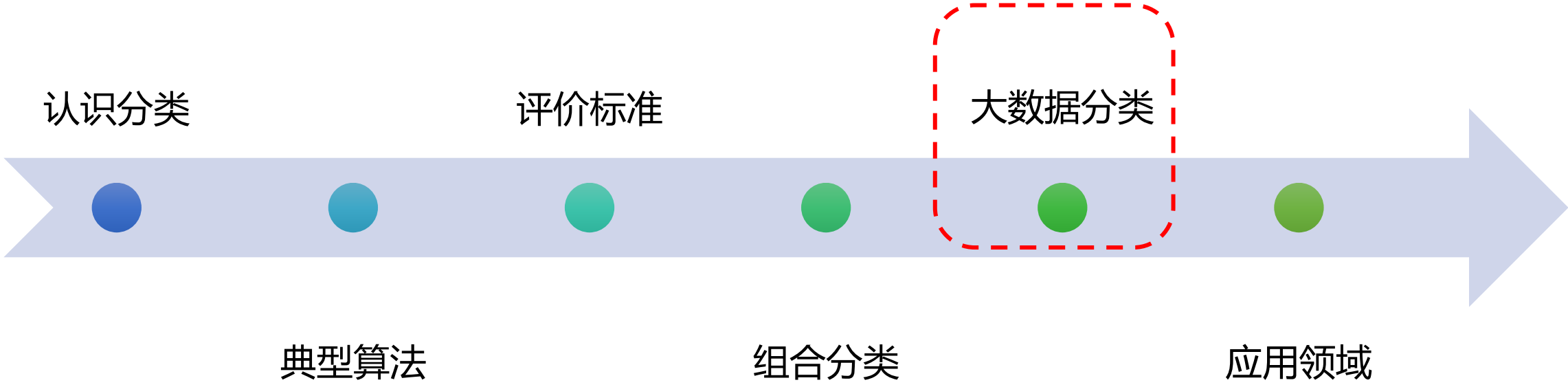


GBDT: Gradient Boost Decision Tree

- Gradient：梯度、倾斜度。
- 与传统的Boost的区别：**每一次的计算是为了减少上一次的残差(residual)**。为了消除残差，算法在残差减少的梯度(Gradient)方向上建立一个新的模型。
  - 传统Boost：对正确、错误的样本进行加权。
  - GBDT：新模型建立是为了使之前模型的残差往梯度方向减少。



五、大数据分类



大数据挖掘面临的挑战

单一学习变为分布式学习

- 传统的分类挖掘方法以单一学习样本集为基础，而大数据的分布式收集特性决定分类学习需要分布式进行，因而对应的分布式学习策略和方法需要研究。

静态数据变为动态流数据

- 动态流动的流式大数据和传统数据库存储的静态数据有显著的不同，不可能一次性将所有数据存储起来再进行离线式的挖掘，必须探索在线实时的收集技术和随时间变化的增量式的挖掘方法。

分布式很难保证学习样本的纯度要求

- 传统的分类挖掘技术对学习样本集要求较高，而分布式、流式大数据的分类挖掘需要多节点、多步骤协同处理，很难保证学习样本集的纯度，所以必须针对这类大数据的挖掘特点来探索鲁棒性能好的分类技术。

SLIQ算法改进C4.5

预排序

- 对于连续属性在每个内部结点寻找其最优分裂标准时，都需要对训练集按照该属性的取值进行排序，操作需要大量时间。为此，SLIQ算法 采用了预排序技术，针对每个属性的取值，把所有的记录按照从小到大的顺序进行排序，以消除在决策树的每个结点对数据集进行的排序。

广度优先策略

- C4.5算法中，树的构造是按照深度优先策略完成的，需要对每个属性列表在每个结点处都进行一遍扫描，费时很多。SLIQ采用 广度优先策略构造决策树，即在决策树的每一层只需对每个属性列表扫描一次，就可以为当前决策树中每个叶子结点找到最优分裂标准。

分布式改进算法

分布式平台技术与分类算法的结合

- Apache Storm是一个分布式的，可靠的，容错的数据流处理系统。
- Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。
- Hadoop是一个由Apache基金会所开发的分布式系统基础架构。

分布式改进算法

局部挖掘

- 在每个局部节点依据数据模型来收集当前数据，然后对上一挖掘点所维护的局部模型进行增量式更新，形成当前新的模式。

模式传输

- 当一个局部节点的模式更新完成后就通过网络把它传送到中心节点。

全局挖掘

- 当所有局部节点的当前模式都被成功地送到中心节点后，中心节点就进行全局集成分类器的学习，将全局模式更新到当前状态。

Spark优化随机森林

Spark在实现随机森林时，采用了下面几个优化策略：

切分点抽样

- MlLib通过抽样的方法在样本上进行排序，并且根据样本获取切分点。

Feature装箱

- 箱子由相邻样本切分点构成，通过计算每个箱子中不同种类的占比计算出最优切分点。

分区统计

- RDD分区中装箱数据单独统计，通过reduce合并每个分区的数据得到总体的装箱数据。

逐层计算

- MlLib采用的策略是逐层构建树节点（广度优先），遍历所有数据的次数等于所有树的最大层数。

使用这些策略，原因在于RDD的数据分布在不同服务器上，为了避免过多的I/O，必须在原始算法上做出一些优化，否则执行时间可能难以接受。

流数据处理模型

界标模型（Landmark model）：

- 处理数据范围从一个固定时间戳到当前时间戳。
- 创建基于界标模型的概要数据结构，要求这个结构能够近似模拟这个数据集合的特征。

滑动窗口模型（sliding window model）：

- 仅关心数据流中最新的W(W 也称为滑动窗口大小)个数据，随着数据的不断到达，窗口中的数据也不断平移。
- 其挑战性在于，不仅新数据不断到达，而且旧数据会过期。

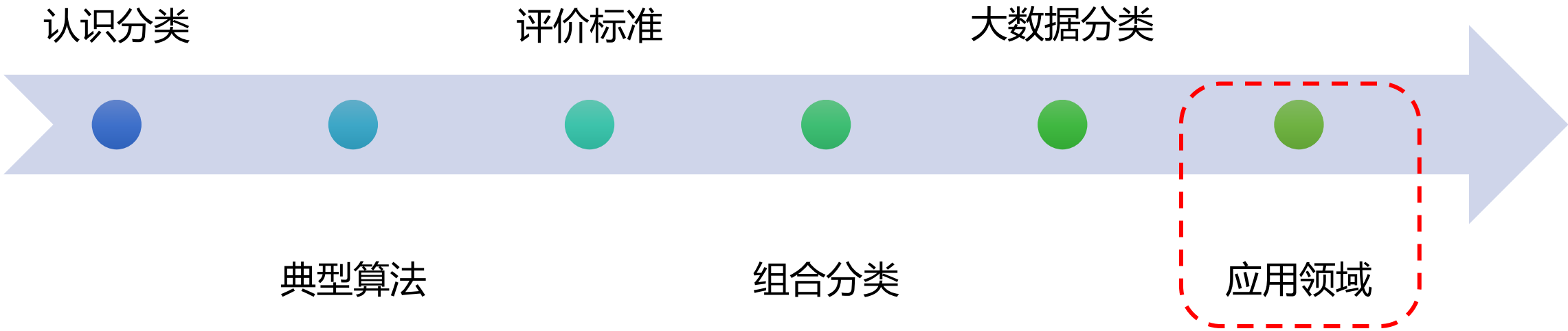
快照模型（snapshot model）：

- 将操作限制在两个预定义的时间戳之间。

分治策略

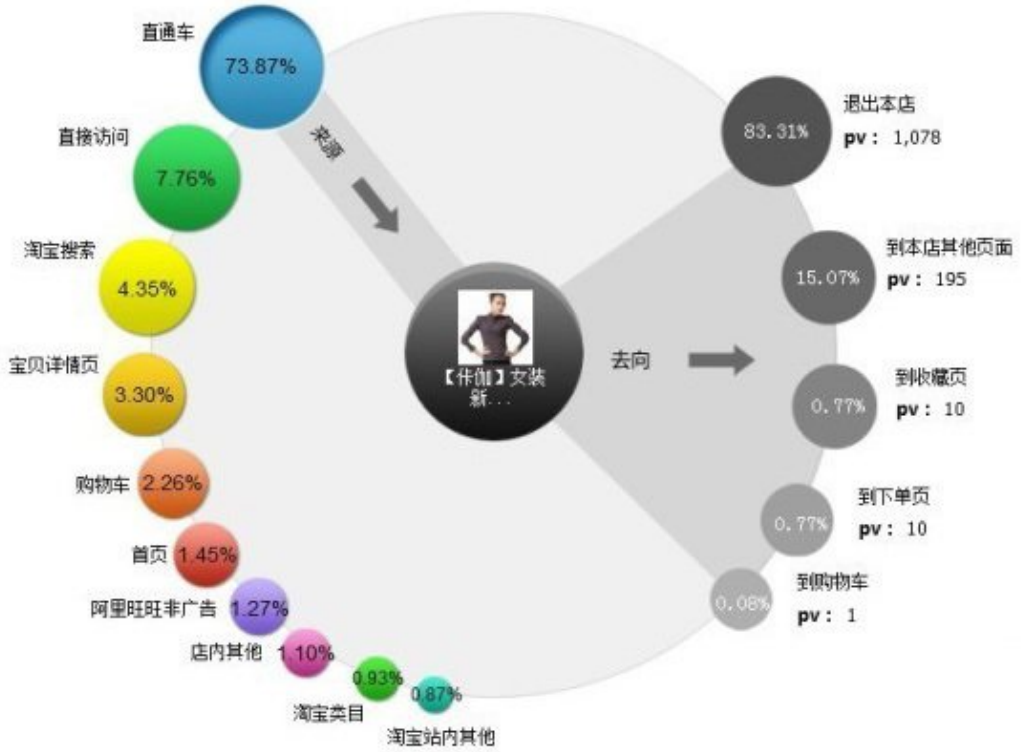
- 数据分治与并行处理策略是大数据处理的基本策略。
- 但目前的分治与并行处理策略较少利用大数据的分布知识，且需考虑影响大数据处理的负载均衡与计算效率。
- 如何学习大数据的分布知识用于优化负载均衡是一个亟待解决的问题。

六、应用领域



目标客户细分

- 运用大数据挖掘技术来分析庞大的用户数据、实现客户管理（CRM）、识别客户消费习惯差异和深入了解客户消费行为特征，设计高价值、个性化和多元化的服务与产品，是企业核心竞争力之一。



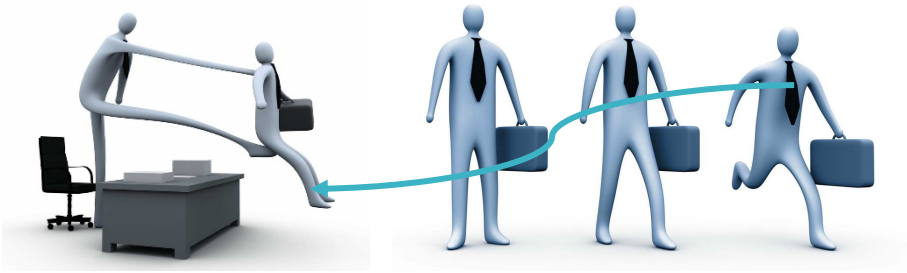
淘宝魔方大数据平台客户分类数据

客户流失预警

- **客户流失预警模型**：根据客户特征对客户进行分群、建模，筛选出可能即将要流失的用户，辅助业务部门提高客户维系挽留工作的效率、提高维系成本的使用效率，降低客户流失率。
- **招商银行**通过数据分析识别出招行信用卡高价值客户经常出现在星巴克、DQ、麦当劳等场所后，通过构建客户流失预警模型，对**流失率等级前20%的客户**发售高收益理财产品予以挽留，使得金卡和金葵花卡客户流失率分别降低了**15个和7个百分点**；

招商银行 CHINA MERCHANTS BANK

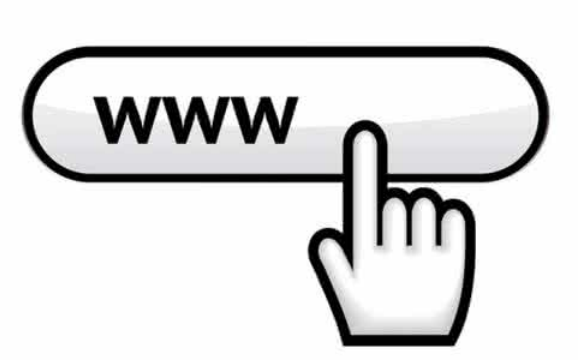
金葵花理财 SUNFLOWER 专注您所关注



征信评估

- 通过大数据分类进行征信可以进行风险评估，依据评估分数，**预测还款人的还款能力、还款意愿、以及欺诈风险。**
- 信用评估的来源更加广泛，社交网络与电子商务行为中产生的海量数据，都能给用户行为提供侧面支持。
- 将海量数据纳入征信体系，并以多个信用模型进行多角度分析，可以使信用评价更精准。
- 大数据带来了更具时效性的评判标准。大数据征信能够实时监测到信用主体的信用变化，企业可以及时拿出解决方案，避免不必要的风险。

广告点击行为预测



- 对用户主动或者被动留下的大量行为数据和与用户行为相关的海量情境信息使用大数据分类算法进行处理进行广告点击行为预测，既可以针对用户进行精准的广告推荐，也可以有效的判断广告投放的效果。
- 确保广告对用户的“伤害”定量化地控制在一定范围之内。
- 可以基于广告投放成果进行收费，简化了广告投放并提升整个广告体系的效率。

垃圾邮件/短信处理

- 目前在垃圾邮件/短信过滤的分类技术中最常见的是**贝叶斯算法**。
- 贝叶斯算法的主要原理是，根据邮件正文中的单词，利用贝叶斯条件概率，计算属于垃圾邮件的概率和正常邮件的概率。如果结果表明，属于垃圾邮件的概率大于正常邮件的概率，那么该邮件就会被划为垃圾邮件。

自动文本分类

- 新闻网站包含大量报道文章，基于文章内容，需要将这些文章按题材进行自动分类，例如自动划分政治、经济、军事、体育、娱乐等。媒体每日有大量投稿，依靠文本分类技术，能够对**文章进行自动审核**，标记投稿中的暴力、政治、垃圾广告等违规内容。
- 文本分类一般包括了文本的表达、 分类器的选择与训练、分类结果的评价与反馈等过程，其中文本的表达又可细分为文本预处理、索引和统计、特征抽取等步骤。
- 文本分类常用的分类算法有：**决策树，Rocchio，朴素贝叶斯，和支持向量机**等。

安全领域入侵检测

- 在入侵检测中，监视、分析用户及系统活动，并予以迅速的反应，正是大数据分类技术的核心内容，因此将其应用于网络安全的入侵检测是**可行并且有效**的。
- 入侵检测体系构建过程：
  - **数据抽取**，对网络中的各种行为进行行为特征的抽取；
  - **数据预处理**，将抽取得到的行为特征数据进行清洗、集成、转换预处理，形成相对一致性的数据格式；
  - 通过**构建模型**，建立入侵检测的行为模型；
  - **安全防护检测**，对非法的网络行为进行拦截和响应。

自动驾驶

■ 目前的自动驾驶仍然停留在激光雷达对外部环境的识别层面，对于外部动态数据的应用比较有限，而随着大数据分类技术逐步应用于联网车辆，**自动驾驶将会具备超越人类的决策能力**。

■ 自动驾驶汽车依靠人工智能、视觉计算、雷达、监控装置和全球定位系统协同合作，让电脑可以**在没有任何人类主动的操作下，自动安全地操作机动车辆**。

■ 2014年12月中下旬，谷歌首次展示自动驾驶原型车成品，该车可全功能运行。

■ 2015年5月，谷歌宣布将于2015年夏天在加利福尼亚州山景城的公路上测试其自动驾驶汽车。

联系方式

- 北京邮电大学软件学院
- 联系人：牛琨
- 手机：18911815860
- 固话：010-62282761
- 地址：北京邮电大学明光楼309