

监督学习

主讲：王亚星、刘夏雷、郭春乐
南开大学计算机学院

致谢：本课件主要内容来自浙江大学吴飞教授、
南开大学程明明教授

作业

- 资料

https://nankai.feishu.cn/drive/folder/KHARfPDsLlit6hdkPnxcnT8FnPc?from=from_copylink

- 作业递交：

https://nankai.feishu.cn/sheets/FKISs8GRvhxPXRtRR2ccBWFDnng?from=from_copylink

- 英文递交

作业

新建
新建文档开始协作

上传
上传本地文件

添加
添加云文档的快捷方式

模板
选择模板

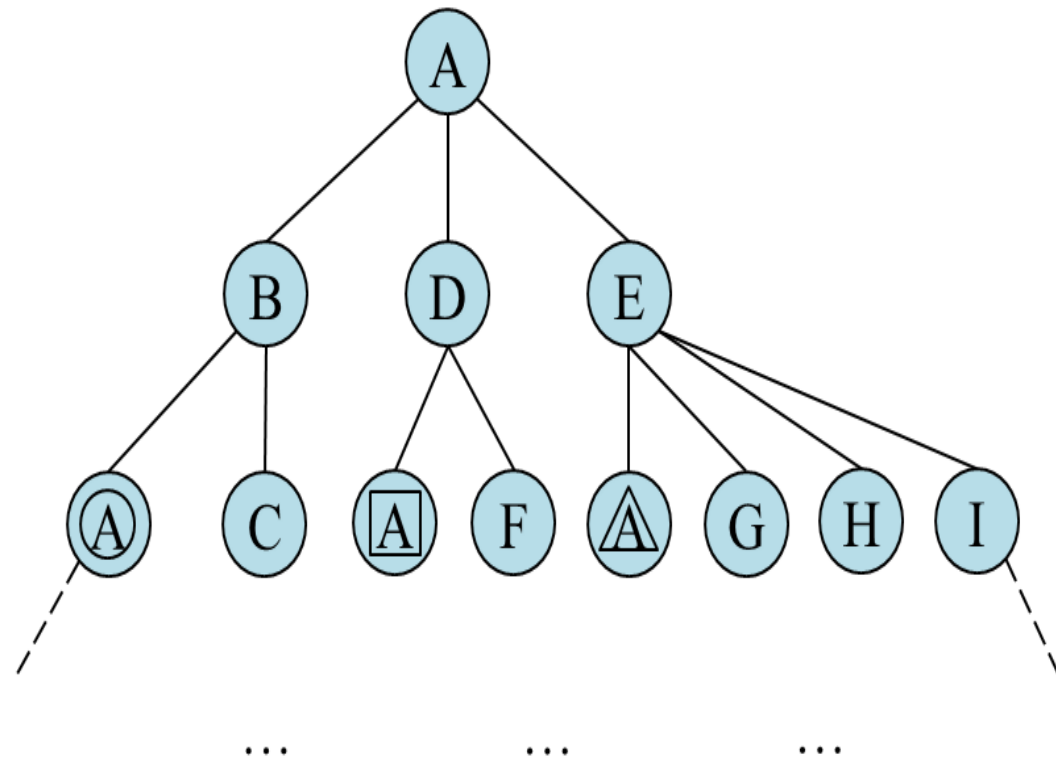
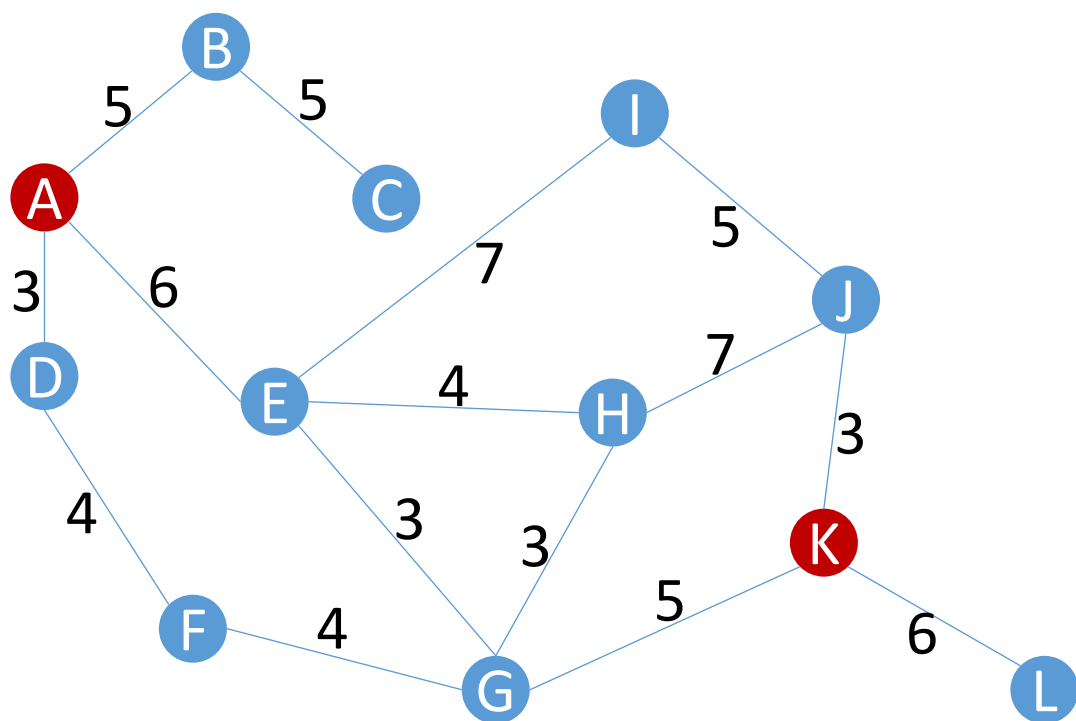
名称 ↑	所有者 ▾	修改时间 ▾	田 >
 overleaf_study.mov	王亚星	今天 11:01	...
 sora.zip	王亚星	今天 10:53	...
 sora学习	王亚星	今天 11:03	...

课程回顾：搜索求解

- 搜索算法基础
- 启发式搜索
- 对抗搜索
- 蒙特卡洛树搜索

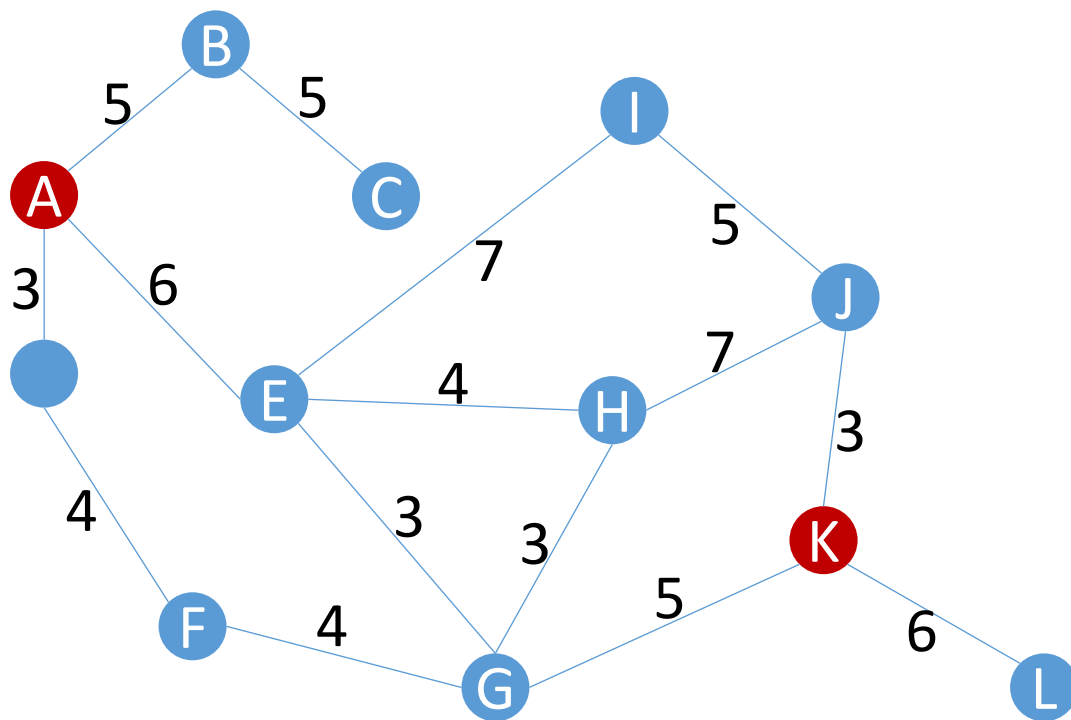
搜索树：用一棵树来记录算法探索过的路径

- 搜索算法会时刻记录所有从初始结点出发已经探索过的路径，每次从中选出一条，从该路径末尾状态出发进行一次状态转移，探索一条尚未被探索过新路径。



搜索算法：启发式搜索（有信息搜索）

- 在搜索的过程中利用与所求解问题相关的辅助信息，其代表算法为**贪婪最佳优先搜索**(Greedy best-first search)和**A*搜索**。



寻找从城市A到城市K之间最短路线？

搜索算法：启发式搜索(有信息搜索)

辅助信息	所求解问题之外、与所求解问题相关的特定信息或知识。	
评价函数 $f(n)$ (evaluation function)	从当前节点 n 出发，根据评价函数来选择后续结点。	下一个结点是谁？
启发函数 $h(n)$ (heuristic function)	从结点 n 到目标结点之间所形成路径的最小代价值，这里用两点之间的直线距离。	完成任务还需要多少代价？

• 贪婪最佳优先搜索：评价函数 $f(n)$ =启发函数 $h(n)$

• 辅助信息(启发函数)

• 任意一个城市与
终点城市K之间的直线距离

状态	A	B	C	D	E	F	G	H	I	J	K	L
$h(n)$	13	10	6	12	7	8	5	3	6	3	0	6

辅助信息：任意一个城市与终点城市
K之间的直线距离

搜索算法：A*算法

- 评价函数： $f(n) = g(n) + h(n)$
 - $g(n)$ 表示从起始结点到结点 n 的开销代价值
 - $h(n)$ 表示从结点 n 到目标结点路径中所估算的最小开销代价值
 - $f(n)$ 可视为经过结点 n 、具有最小开销代价值的路径。

$$\underbrace{f(n)}_{\text{评价函数}} = \underbrace{g(n)}_{\substack{\text{起始结点到结点}n\text{代价} \\ \text{(当前最小代价)}}} + \underbrace{h(n)}_{\substack{\text{结点}n\text{到目标结点代价} \\ \text{(后续估计最小代价)}}}$$

对抗搜索：主要内容

- **最小最大搜索(Minimax Search)**

- 最小最大搜索是在对抗搜索中最为基本的一种让玩家来计算最优策略的方法

- **Alpha-Beta剪枝搜索(Pruning Search)**

- 一种对最小最大搜索进行改进的算法，即在搜索过程中可剪除无需搜索的分支节点，且不影响搜索结果。

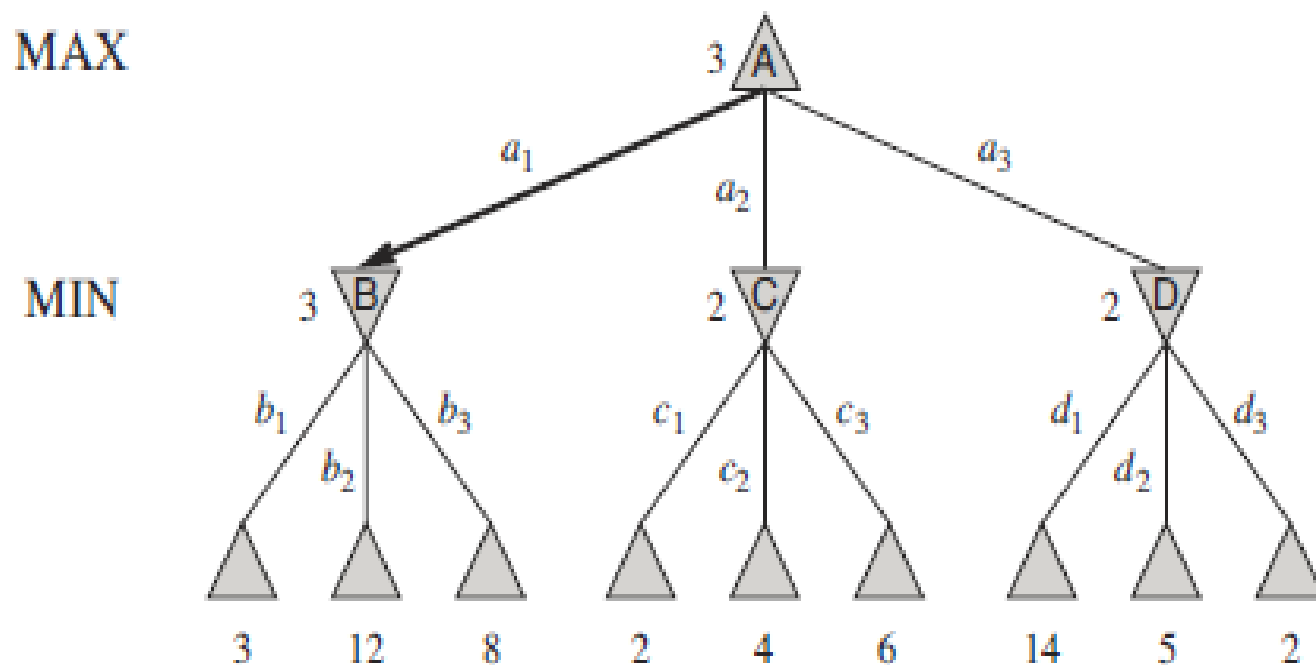
- **蒙特卡洛树搜索(Monte-Carlo Tree Search)**

- 通过采样而非穷举方法来实现搜索。

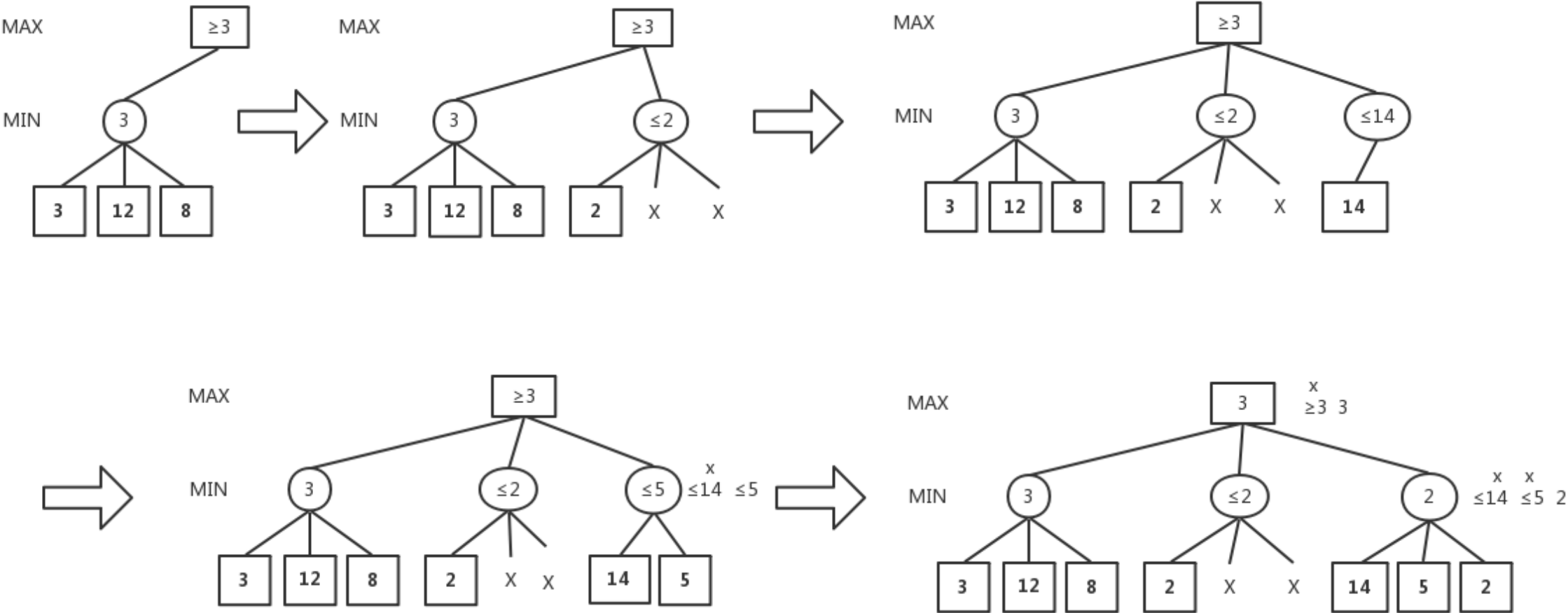
对抗搜索：Alpha-Beta 剪枝搜索

- 在极小化极大算法(minimax算法)中减少所搜索的搜索树节点数。该算法和极小化极大算法所得结论相同，但剪去了不影响最终结果的搜索分枝。

图中MIN选手所在的节点C下属分支4和6与根节点最终优化决策的取值无关，可不被访问。



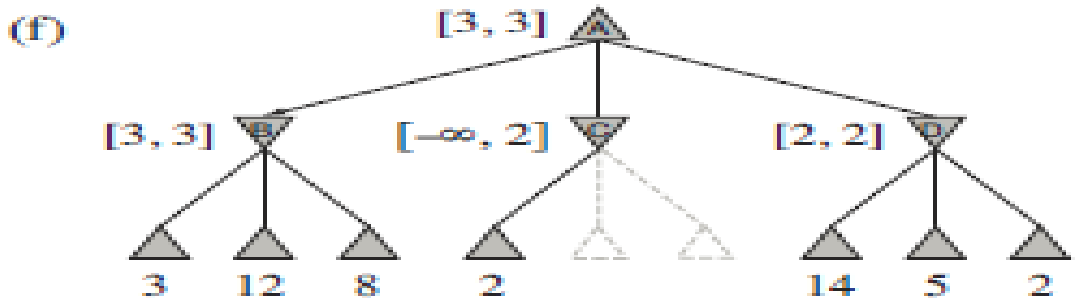
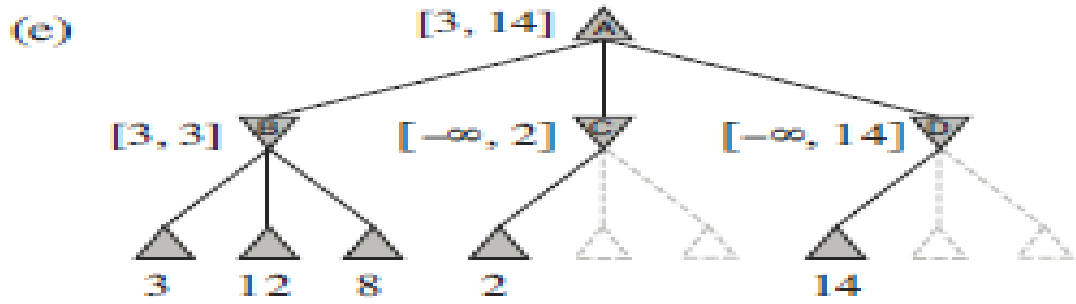
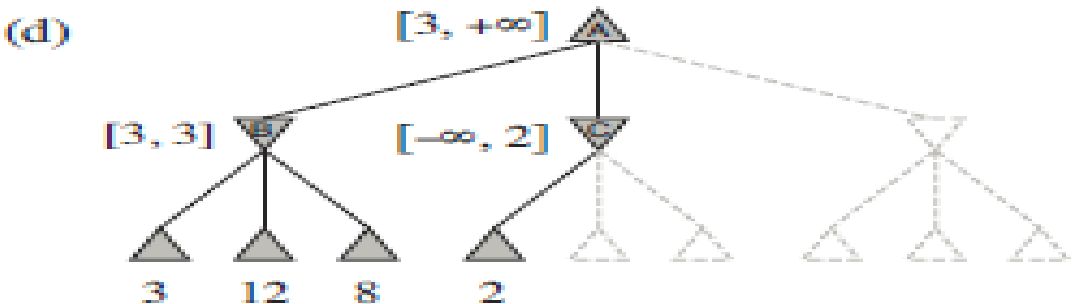
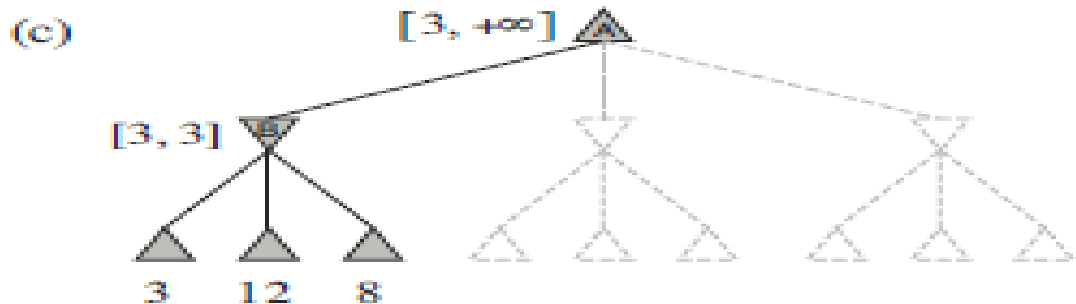
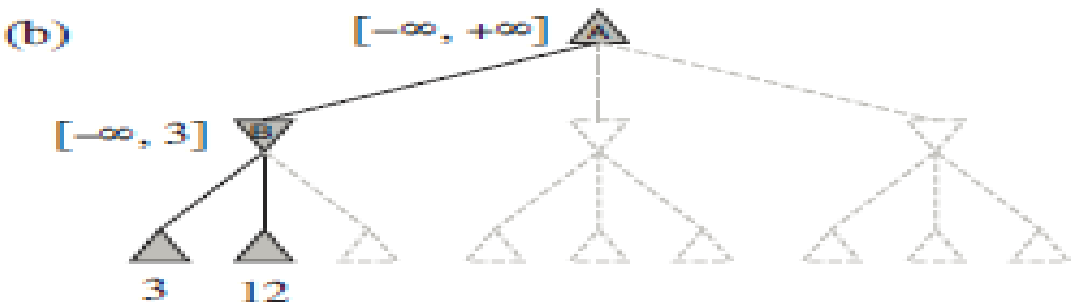
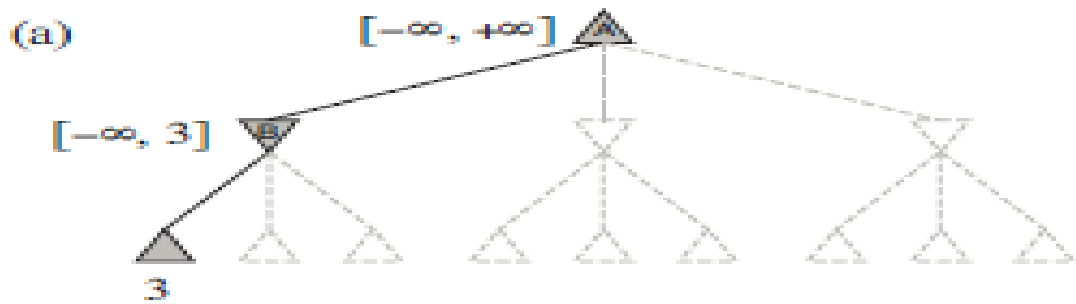
对抗搜索：Alpha-Beta 剪枝搜索



Alpha值(α)	MAX节点目前得到的最高收益
Beta值(β)	MIN节点目前可给对手的最小收益
α 和 β 的值初始化分别设置为 $-\infty$ 和 ∞	

对抗搜索：Alpha-Beta 剪枝搜索

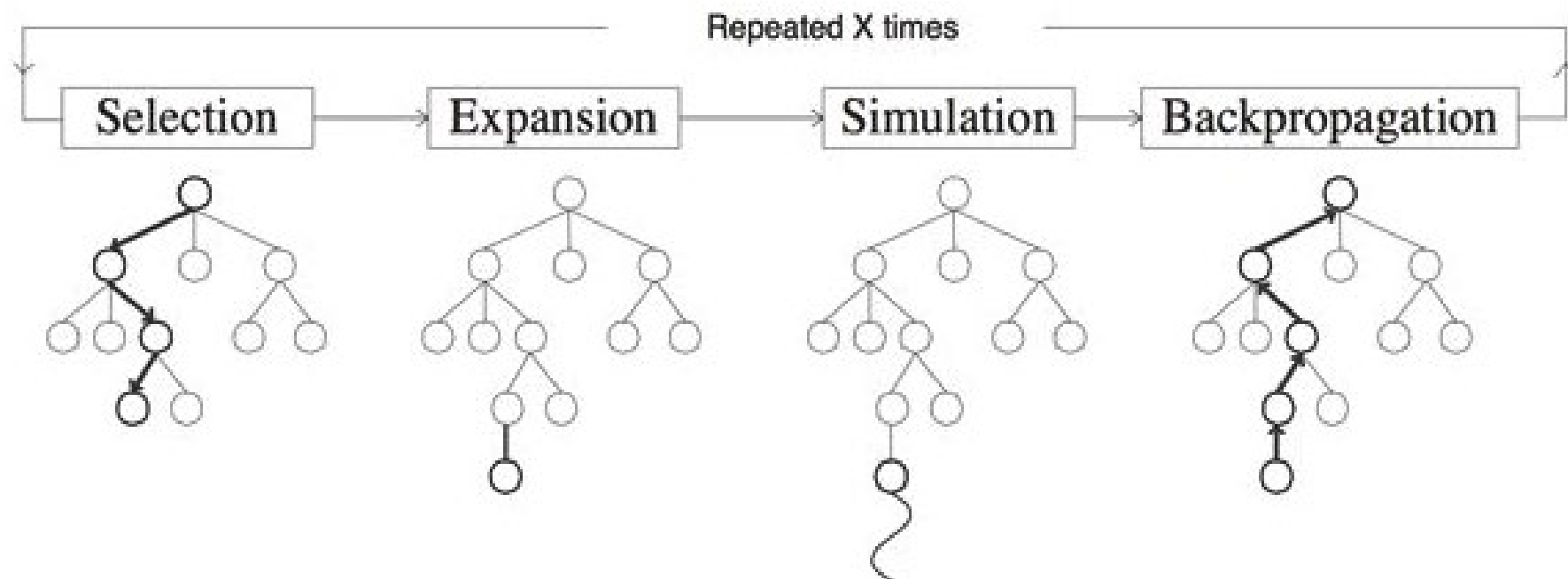
- 从 α 和 β 的变化来理解剪枝过程



蒙特卡洛树搜索

- **选择：** 从根节点 R 开始，递归选择子节点，直至到达叶节点或到达具有还未被扩展过的子节点的节点 L。
- 具体来说，通常用UCB1 (Upper Confidence Bound, 上限置信区间)选择最具“潜力”的后续节点

$$UCB = \bar{X}_j + \sqrt{\frac{2 \ln n}{n_j}}$$



蒙特卡洛树搜索

- **扩展：**

- 如果 L 不是一个终止节点，则随机创建其后的一个未被访问节点，选择该节点作为后续子节点 C 。

- **模拟：**

- 从节点 C 出发，对游戏进行模拟，直到博弈游戏结束。

- **反向传播**

- 用模拟所得结果来回溯更新导致这个结果的每个节点中获胜次数和访问次数。

一、机器学习基本概念

二、回归分析

三、决策树

四、线性区别分析

五、Ada Boosting

六、支持向量机

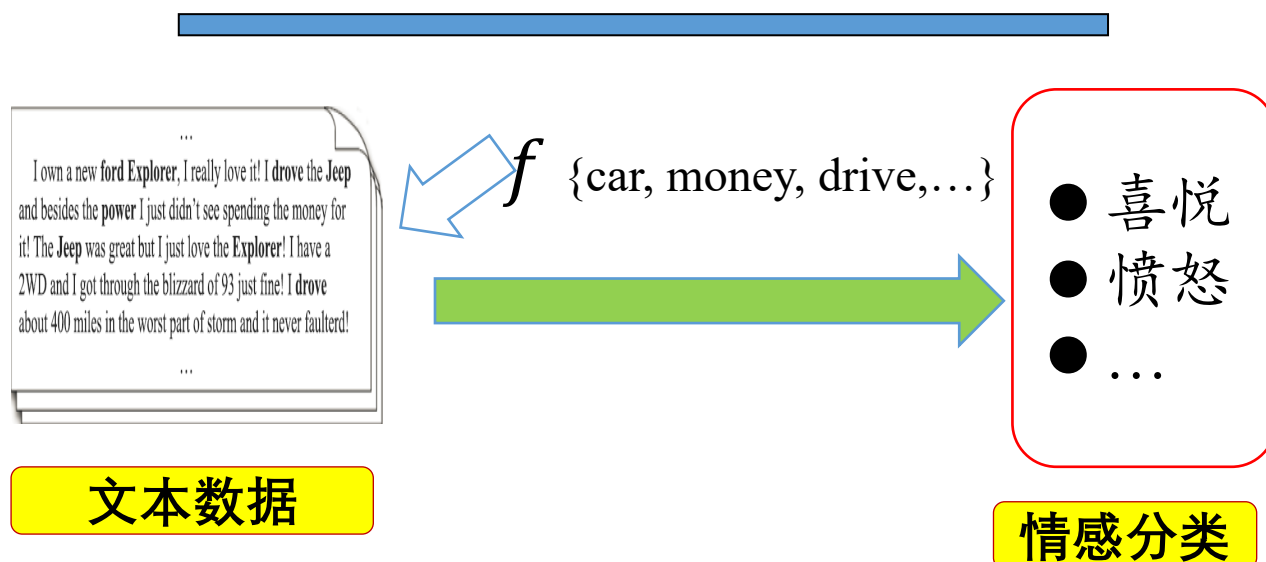
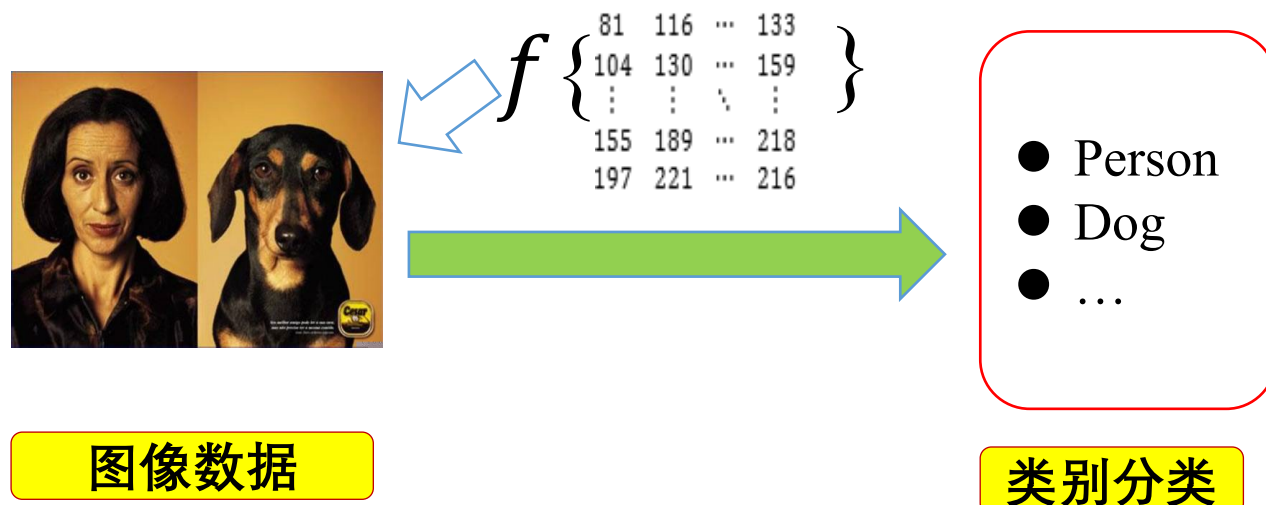
七、生成学习模型

机器学习: 从数据中学习知识

1. 原始数据中提取特征

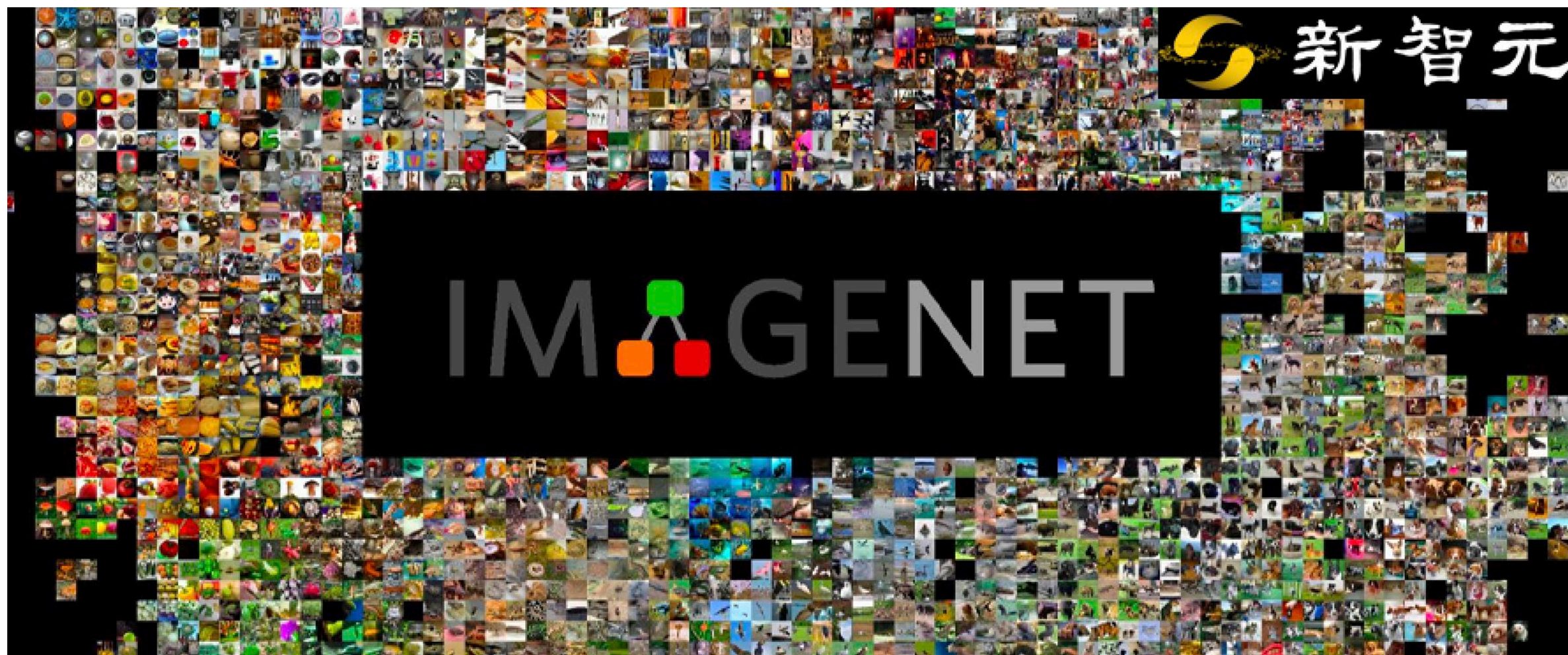
2. 学习映射函数 f

3. 通过映射函数 f 将原始数据映射到语义空间，即寻找数据和任务目标之间的关系



机器学习: 从数据中学习知识

Imagenet: 1M, 1000 classes



机器学习的分类

监督学习(supervised learning)
数据有标签、一般为回归或分类等任务

无监督学习(un-supervised learning)
数据无标签、一般为聚类或若干降维任务

强化学习(reinforcement learning)
序列数据决策学习，一般为与从环境交互中学习

半监督学习
(semi-supervised learning)

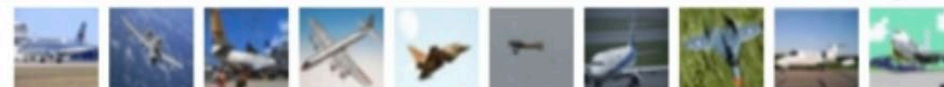
监督学习：重要元素

标注数据

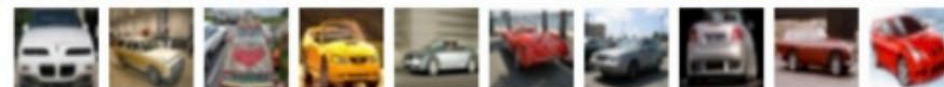
学习模型

损失函数

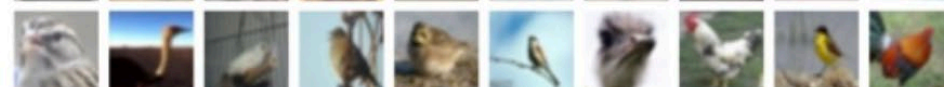
airplane



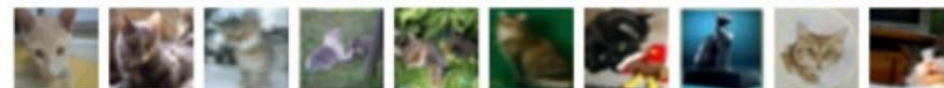
automobile



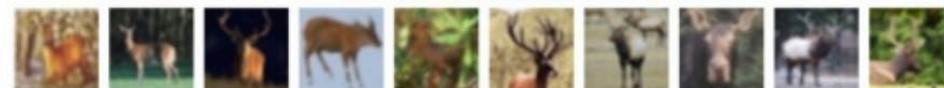
bird



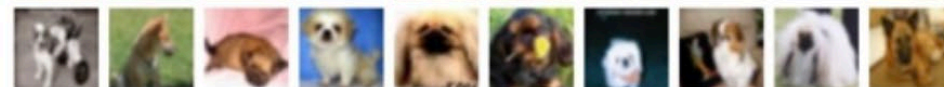
cat



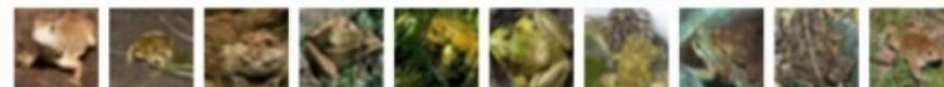
deer



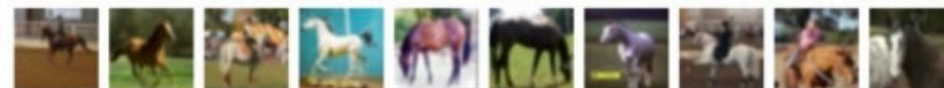
dog



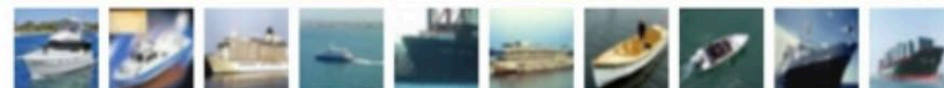
frog



horse



ship



truck

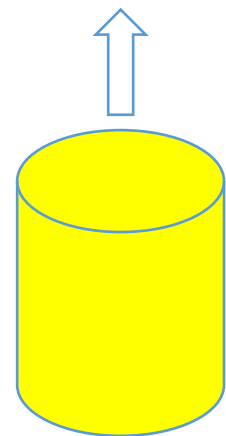


<https://imgur.com/gallery/20190110>

监督学习：损失函数

- 训练集共有 n 个标注数据，第 i 个记为 (x_i, y_i)
- 从训练数据中学习映射函数 $f(x_i)$
 - 损失函数就是真值 y_i 与预测值 $f(x_i)$ 之间差值的函数。
- 在训练过程中希望映射函数在训练数据集上得到“损失”最小
 - 即 $\min \sum_{i=1}^n \text{Loss}(f(x_i), y_i)$ 。

训练映射函数 f
使得 $f(x_i)$ 尽量等于 y_i



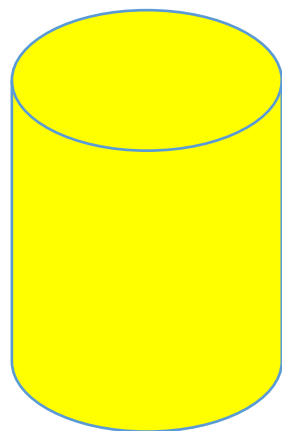
训练数据集
 $(x_i, y_i), i = 1, \dots, n$

监督学习：常见的损失函数

损失函数名称	损失函数定义
0-1损失函数	$Loss(y_i, f(x_i)) = \begin{cases} 1, & f(x_i) \neq y_i \\ 0, & f(x_i) = y_i \end{cases}$
平方损失函数	$Loss(y_i, f(x_i)) = (y_i - f(x_i))^2$
绝对损失函数	$Loss(y_i, f(x_i)) = y_i - f(x_i) $
对数损失函数/对数似然损失函数	$Loss(y_i, P(y_i x_i)) = -\log P(y_i x_i)$

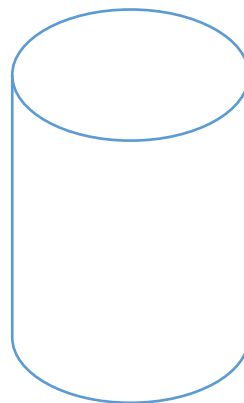
监督学习：训练数据和测试数据

从**训练数据集**学习
得到映射函数 f



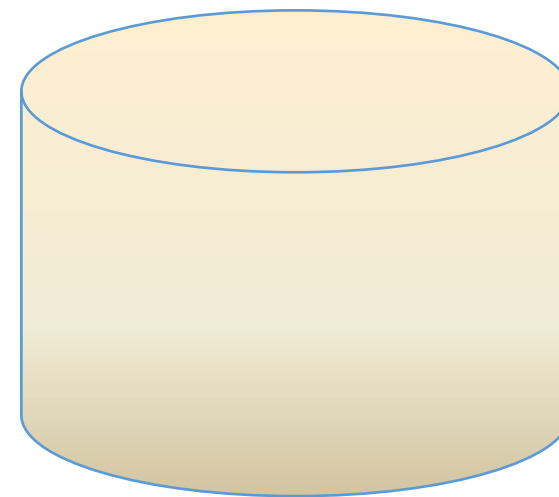
训练数据集
 $(x_i, y_i), i = 1, \dots, n$

在**测试数据集**
测试映射函数 f



测试数据集
 $(x_i', y_i'), i = 1, \dots, m$

未知数据集
上测试映射函数 f



监督学习：经验风险和期望风险

从训练数据集学习映射函数 f

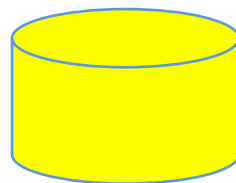
经验风险(empirical risk)

训练集中数据产生的损失。经验风险越小说明学习模型对训练数据拟合程度越好。

监督学习：经验风险和期望风险

- 映射函数训练目标：经验风险最小化
 - Empirical risk minimization

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i))$$



训练数据集
 $(x_i, y_i), i = 1, \dots, n$

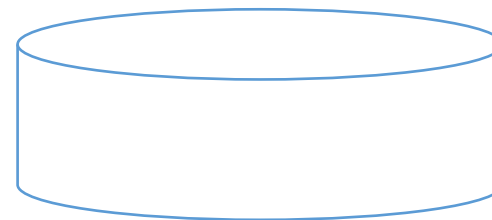
选取一个使得训练集所有数据
损失平均值最小的映射函数。
这样的考虑是否够？

监督学习：经验风险和期望风险

- 映射函数训练目标：期望风险最小化
 - Expected risk minimization

$$\min_{f \in \Phi} \int_{x,y} Loss(y, f(x)) P(x, y) dx dy$$

测试数据集数据无穷多
 $(x_i', y_i'), i = 1, \dots, \infty$



监督学习：经验风险和期望风险

- 期望风险是模型关于联合分布期望损失，经验风险是模型关于训练样本集平均损失。

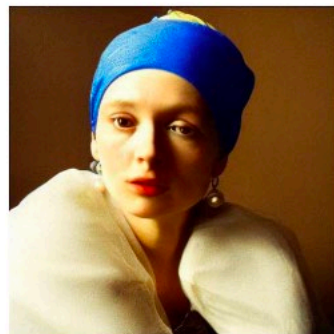
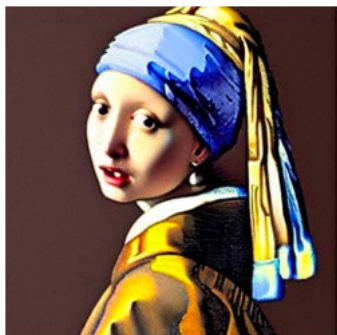
监督学习：经验风险和期望风险

- 模型泛化能力与经验风险、期望风险的关系

对应举一反三

训练集上表现	测试集上表现	
经验风险小	期望风险小	泛化能力强

监督学习：经验风险和期望风险



训练

过拟合

最佳

监督学习：结构风险最小化

- 结构风险最小化(structural risk minimization)

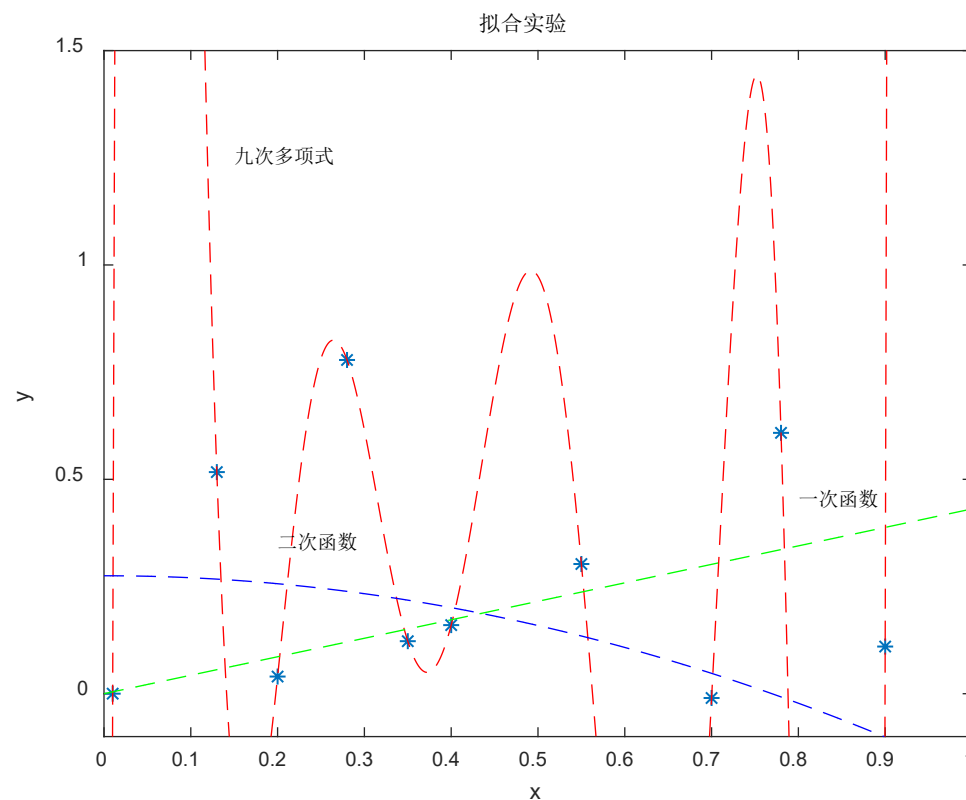
- 为了防止过拟合，在经验风险上加上表示模型复杂度的正则化项(regularizer)或惩罚项(penalty term)：

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda J(f)$$

实际中L2

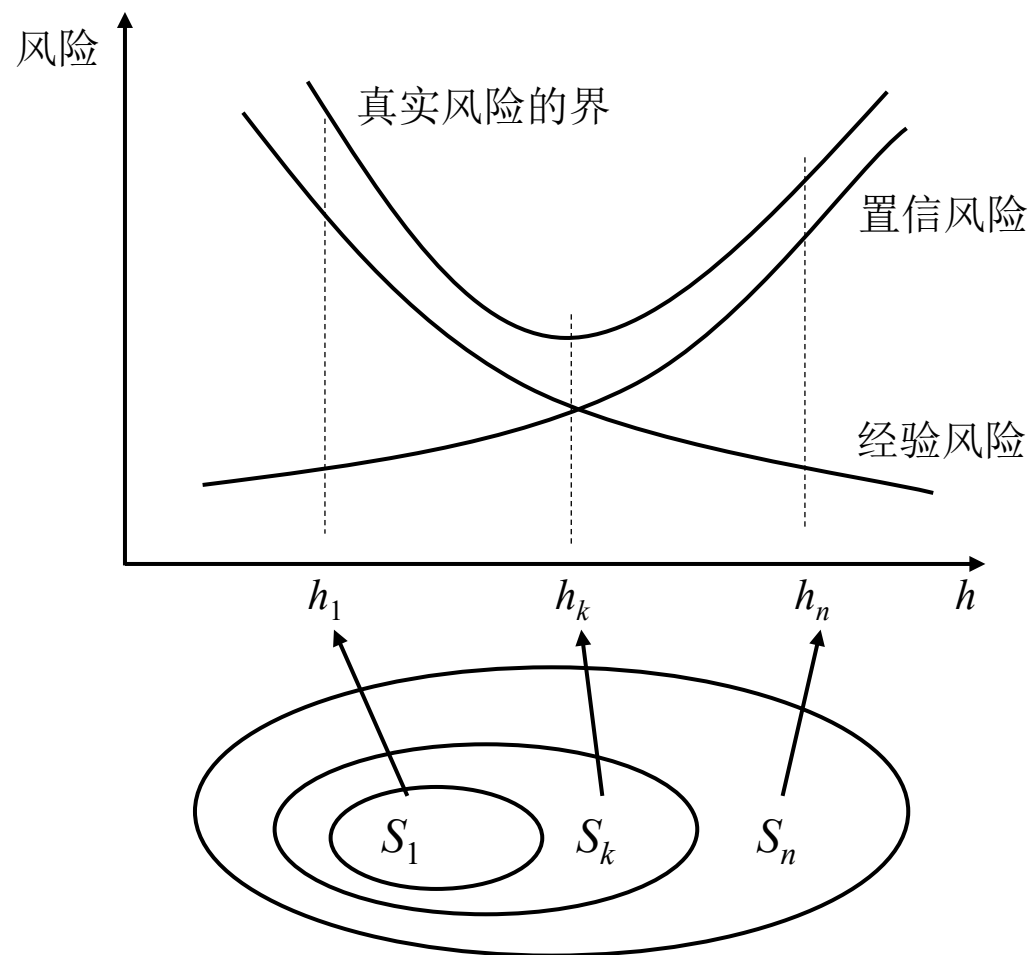
在最小化经验风险与降低模型复杂度之间寻找平衡

监督学习：结构风险最小化



有限样本情况下的拟合实验

监督学习：结构风险最小化



监督学习：判别模型与生成模型

- 监督学习方法又可以分为**生成**方法(generative approach)和**判别**方法(discriminative approach)。
- 所学到的模型分别称为生成模型(generative model)和判别模型(discriminative model).

监督学习：判别模型与生成模型

- 判别方法直接学习判别函数 $f(X)$ 或者条件概率分布 $P(Y|X)$ 作为预测的模型，即判别模型。
- 判别模型关心在给定输入数据下，预测该数据的输出是什么。
- 典型判别模型包括回归模型、神经网络、支持向量机和Ada boosting等。

$$f(\text{人脸}) \rightarrow \text{人脸}$$

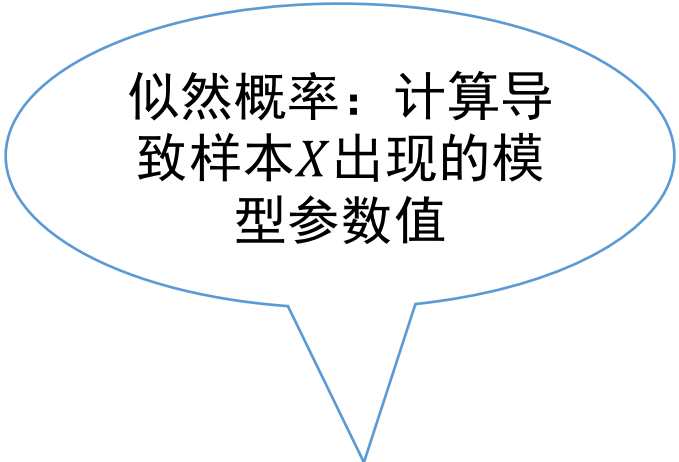
$$P(\text{人脸} | \text{人脸}) = 0.99$$

监督学习：判别模型与生成模型

- 生成模型从数据中学习联合概率分布 $P(X, Y)$ （通过似然概率 $P(X|Y)$ 和类概率 $P(Y)$ 的乘积来求取）

$$P(Y|X) = \frac{P(X, Y)}{P(X)} \text{ 或者 } P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

- 典型方法为贝叶斯方法、隐马尔可夫链
- 授之于鱼、不如授之于“渔”
- 联合分布概率 $P(X, Y)$ 或似然概率 $P(X|Y)$ 求取很困难



似然概率：计算导致样本 X 出现的模型参数值

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

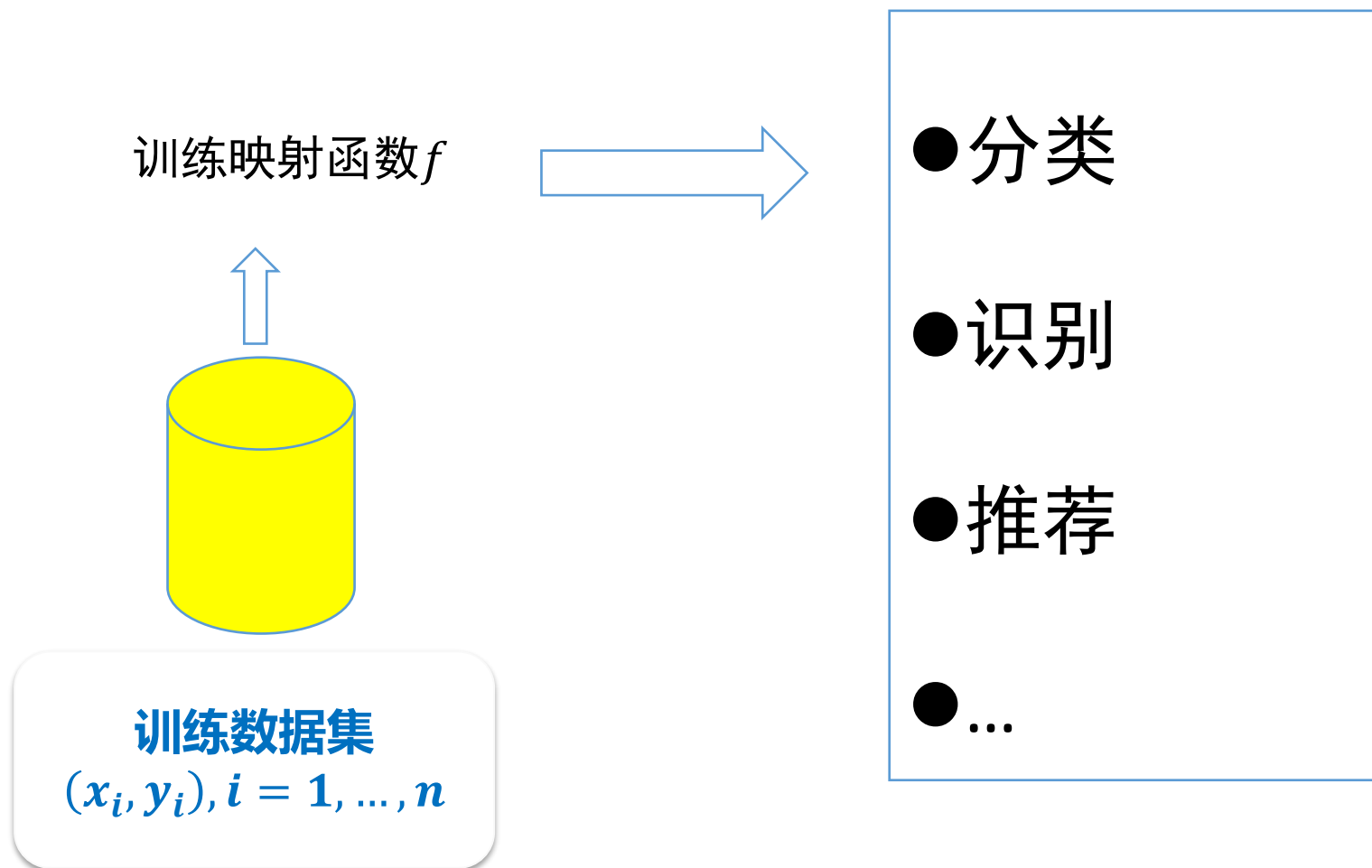
监督学习：判别模型与生成模型



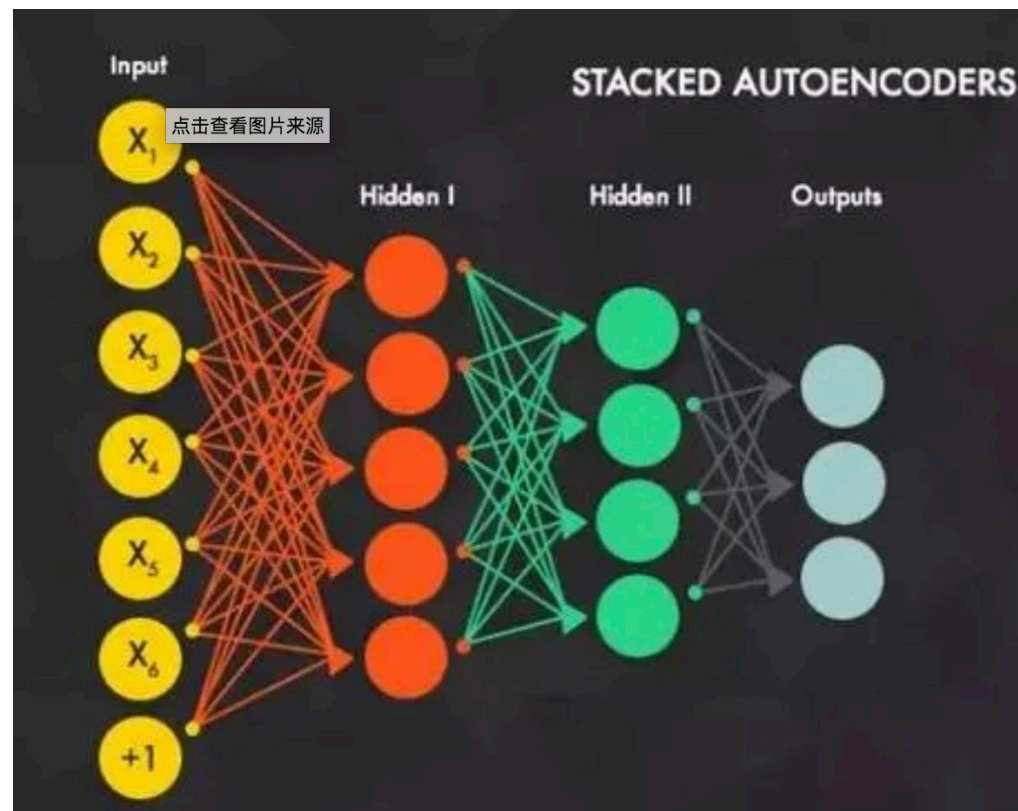
监督学习：判别模型与生成模型



监督学习：应用



监督学习：应用



0.9	Dog
0.05	Cat
0.05	Car

一、机器学习基本概念

二、回归分析

三、决策树

四、线性区别分析

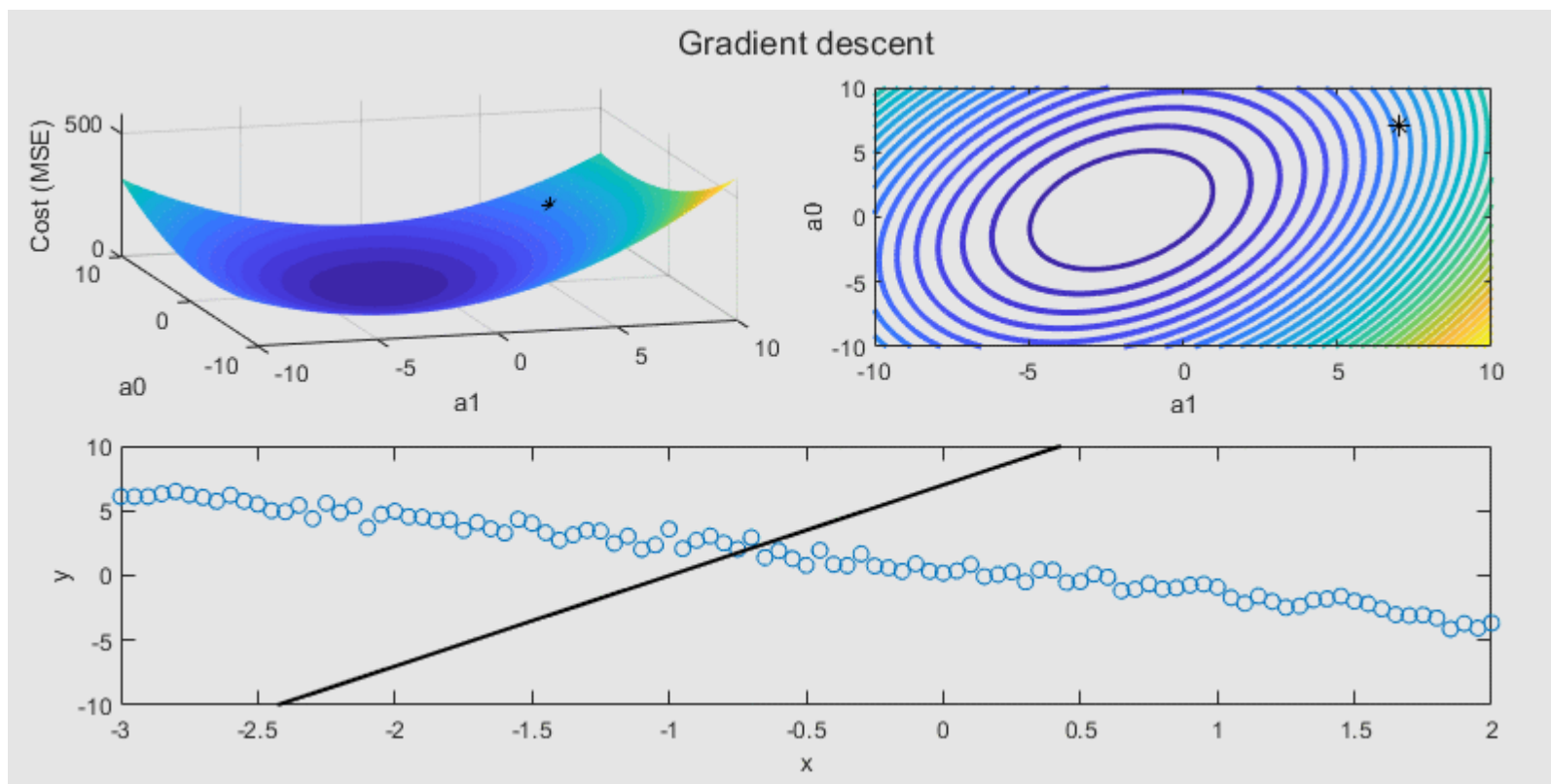
五、Ada Boosting

六、支持向量机

七、生成学习模型

线性回归

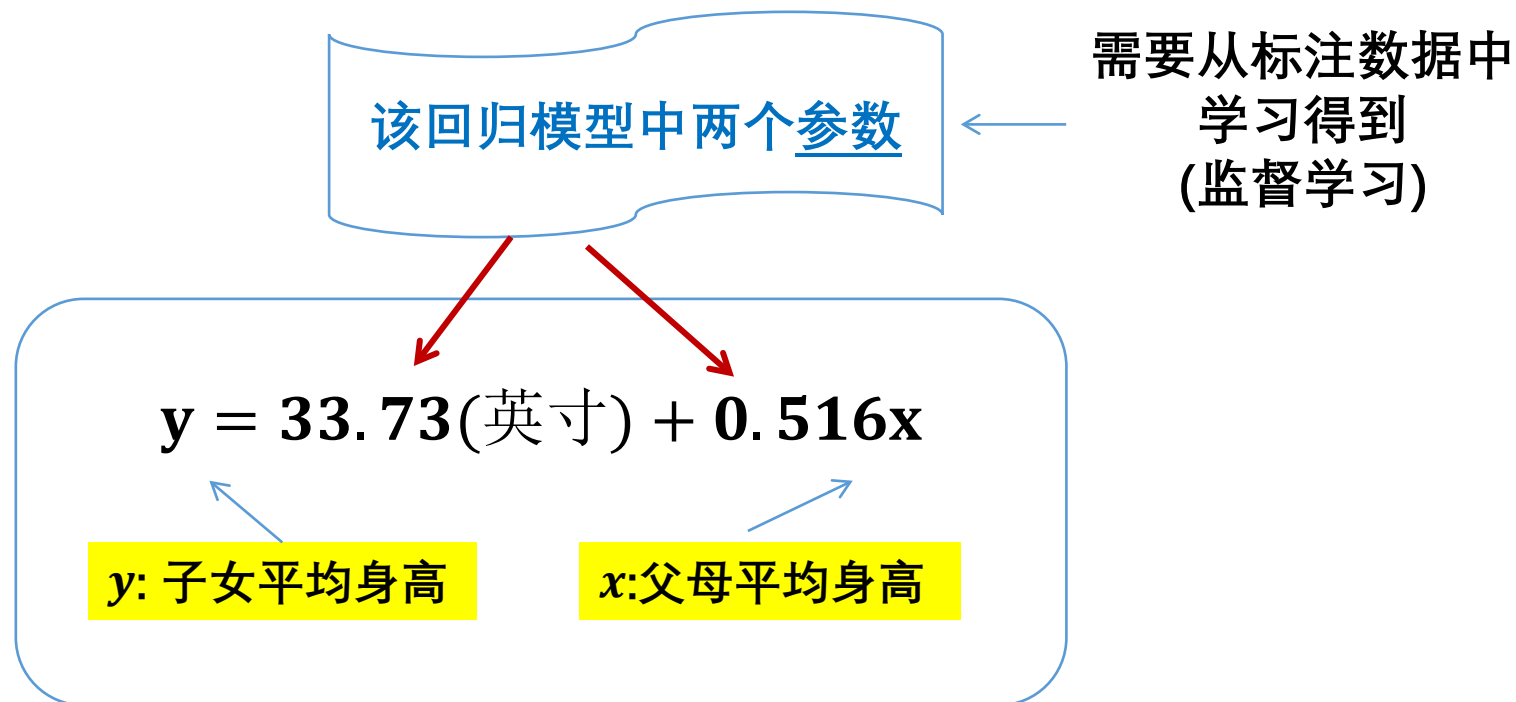
- 刻画不同变量之间关系的模型被称为回归模型。如果这个模型是线性的，则称为线性回归模型。



线性回归：一元线性回归

$$y = 33.73(\text{英寸}) + 0.516x$$

线性回归：一元线性回归



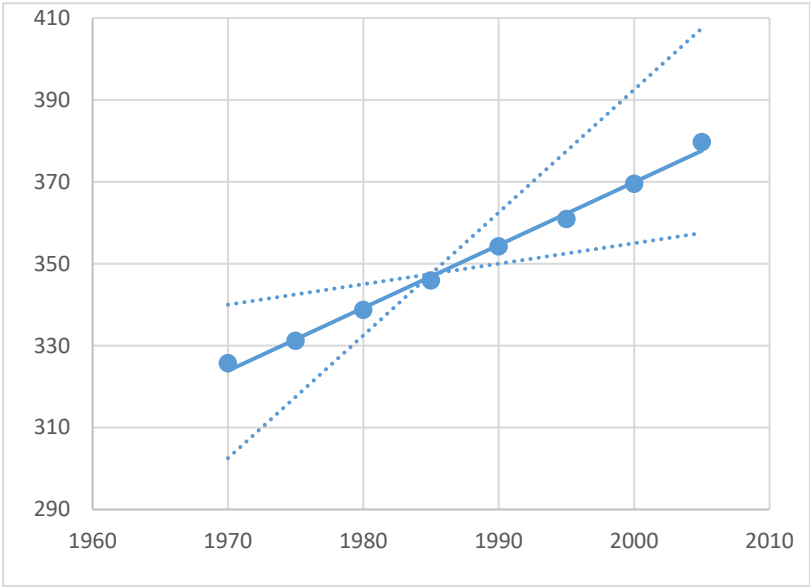
线性回归：一元线性回归

下表给出了莫纳罗亚山（夏威夷岛的活火山）从1970年到2005年每5年的二氧化碳浓度，单位是百万分比浓度（Parts Per Million, ppm）。

年份(x)	1970	1975	1980	1985	1990	1995	2000	2005
CO2 (y)	325.68	331.15	338.69	345.90	354.19	360.88	369.48	379.67

问题： 1) 给出1984年二氧化碳浓度值； 2) 预测2010年二氧化碳浓度值

线性回归：一元线性回归



年份(x)	1970	1975	1980	1985	1990	1995	2000	2005
CO2 (y)	325.68	331.15	338.69	345.90	354.19	360.88	369.48	379.67

莫纳罗亚山地区时间年份与二氧化碳浓度之间的一元线性回归模型（实线为最佳回归模型）

线性回归：一元线性回归

回归模型参数求取： $y_i = ax_i + b \ (1 \leq i \leq n)$

线性回归：一元线性回归

回归模型参数求取： $y_i = ax_i + b$ ($1 \leq i \leq n$)

$$\frac{\partial L(a, b)}{\partial b} = \sum_{i=1}^n 2(y_i - ax_i - b)(-1) = 0$$

$$\rightarrow \sum_{i=1}^n (y_i - ax_i - b) = 0$$

$$\rightarrow \sum_{i=1}^n (y_i) - a \sum_{i=1}^n x_i - \sum_{i=1}^n b = 0$$

$$\rightarrow n\bar{y} - an\bar{x} - nb = 0$$



$$b = \bar{y} - a\bar{x}$$

$$\min_{a,b} L(a, b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$$

可以看出：只要给出了训练样本 (x_i, y_i) ($i = 1, \dots, n$)，我们就可以从训练样本出发，建立一个线性回归方程，使得对训练样本数据而言，该线性回归方程预测的结果与样本标注结果之间的差值和最小。

线性回归：一元线性回归

回归模型参数求取： $y_i = ax_i + b$ ($1 \leq i \leq n$)

$$\frac{\partial L(a, b)}{\partial a} = \sum_{i=1}^n 2(y_i - ax_i - b)(-x_i) = 0$$

将 $b = \bar{y} - a\bar{x}$ ($\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$, $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$)

代入上式

$$\rightarrow \sum_{i=1}^n (y_i - ax_i - \bar{y} + a\bar{x})(x_i) = 0$$

$$\rightarrow \sum_{i=1}^n (y_i x_i - ax_i x_i - \bar{y} x_i + a\bar{x} x_i) = 0$$

$$\min_{a,b} L(a, b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$$

$$\rightarrow \sum_{i=1}^n (y_i x_i - \bar{y} x_i) - a \sum_{i=1}^n (x_i x_i - \bar{x} x_i) = 0$$

$$\rightarrow \left(\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} \right) - a \left(\sum_{i=1}^n x_i x_i - n\bar{x}^2 \right) = 0$$

$$a = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i x_i - n\bar{x}^2}$$

线性回归：一元线性回归

回归模型参数求取： $y_i = ax_i + b \ (1 \leq i \leq n)$ $\min_{a,b} L(a,b) = \sum_{i=1}^n (y_i - a \times x_i - b)^2$

$$b = \bar{y} - a\bar{x}$$

$$a = \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2}$$

年份(x)	1970	1975	1980	1985	1990	1995	2000	2005
CO2 (y)	325.68	331.15	338.69	345.90	354.19	360.88	369.48	379.67

训练样本数据

$$a = \frac{x_1 y_1 + x_2 y_2 + \cdots + x_8 y_8 - 8\bar{x}\bar{y}}{x_1^2 + x_2^2 + \cdots + x_8^2 - 8\bar{x}^2} = 1.5344$$

$$b = \bar{y} - a\bar{x} = -2698.9$$

预测莫纳罗亚山地区二氧化碳浓度的一元线性回归模型为“二氧化碳浓度 = $1.5344 \times \text{时间年份} - 2698.9$ ”，即 $y = 1.5344x - 2698.9$ 。

线性回归：多元线性回归

多元线性回归模型例子

接下来扩展到数据特征的维度是多维的情况，在上述数据中增加一个影响火灾影响面积的潜在因素—风力。

气温 x	5.1	8.2	11.5	13.9	15.1	16.2	19.6	23.3
风力 z	4.5	5.8	4	6.3	4	7.2	6.3	8.5
火灾影响面积 y	2.14	4.62	8.24	11.24	13.99	16.33	19.23	28.74

多维数据特征中线性回归的问题定义如下：假设总共有 m 个训练数据 $\{(x_i, y_i)\}_{i=1}^m$ ，其中 $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}] \in \mathbb{R}^D$ ， D 为数据特征的维度，线性回归就是要找到一组参数 $a = [a_0, a_1, \dots, a_D]$ ，使得线性函数：

$$f(x_i) = a_0 + \sum_{j=1}^D a_j x_{i,j} = a_0 + \mathbf{a}^T \mathbf{x}_i$$

线性回归：多元线性回归

最小化均方误差函数：

$$J_m = \frac{1}{m} \sum_{i=1}^m (y_i - f(\mathbf{x}_i))^2$$

$$\mathbf{y} = \mathbf{A} \rightarrow \frac{\delta y}{\delta x} = \mathbf{0}$$

$$\mathbf{y} = \mathbf{A}\mathbf{x} \rightarrow \frac{\delta y}{\delta x} = \mathbf{A}$$

$$\mathbf{y} = \mathbf{x}\mathbf{A} \rightarrow \frac{\delta y}{\delta x} = \mathbf{A}^T$$

$$\mathbf{y} = \mathbf{x}^T \mathbf{A} \mathbf{x} \rightarrow \frac{\delta y}{\delta x} = 2\mathbf{x}^T \mathbf{A}$$

MORE VIDEOS

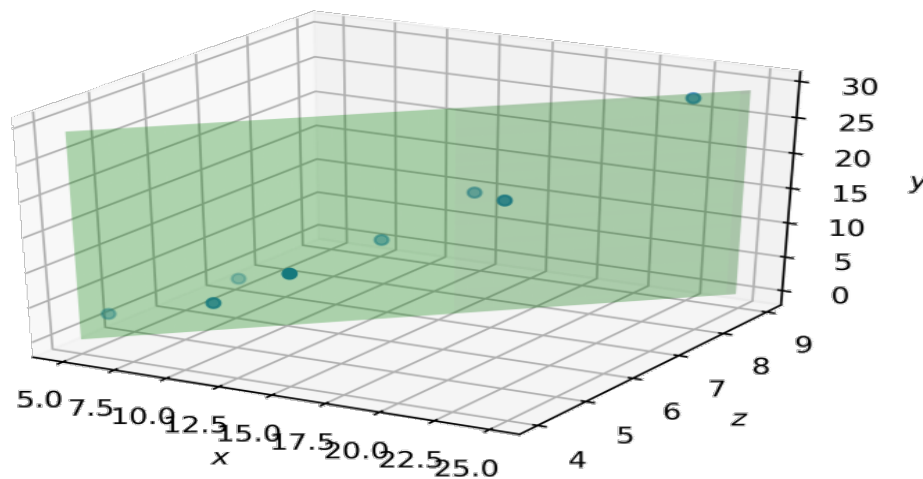
线性回归：多元线性回归

对于上面的例子，转化为矩阵的表示形式为：

$$X = \begin{bmatrix} 5.1 & 8.2 & 11.5 & 13.9 & 15.1 & 16.2 & 19.6 & 23.3 \\ 4.5 & 5.8 & 4. & 6.3 & 4. & 7.2 & 6.3 & 8.5 \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \end{bmatrix}$$
$$\mathbf{y} = [2.14 \quad 4.62 \quad 8.24 \quad 11.24 \quad 13.99 \quad 16.33 \quad 19.23 \quad 28.74]^T$$

其中矩阵 X 多出一行全1，是因为常数项 a_0 ，可以看作是数值为全1的特征的对应系数。计算可得

$$\mathbf{a} = [1.312 \quad 0.626 \quad -9.103]$$
$$\mathbf{y} = -9.103 + 1.312x + 0.626z$$



线性回归：逻辑回归/对数几率回归

逻辑回归/对数几率回归模型例子

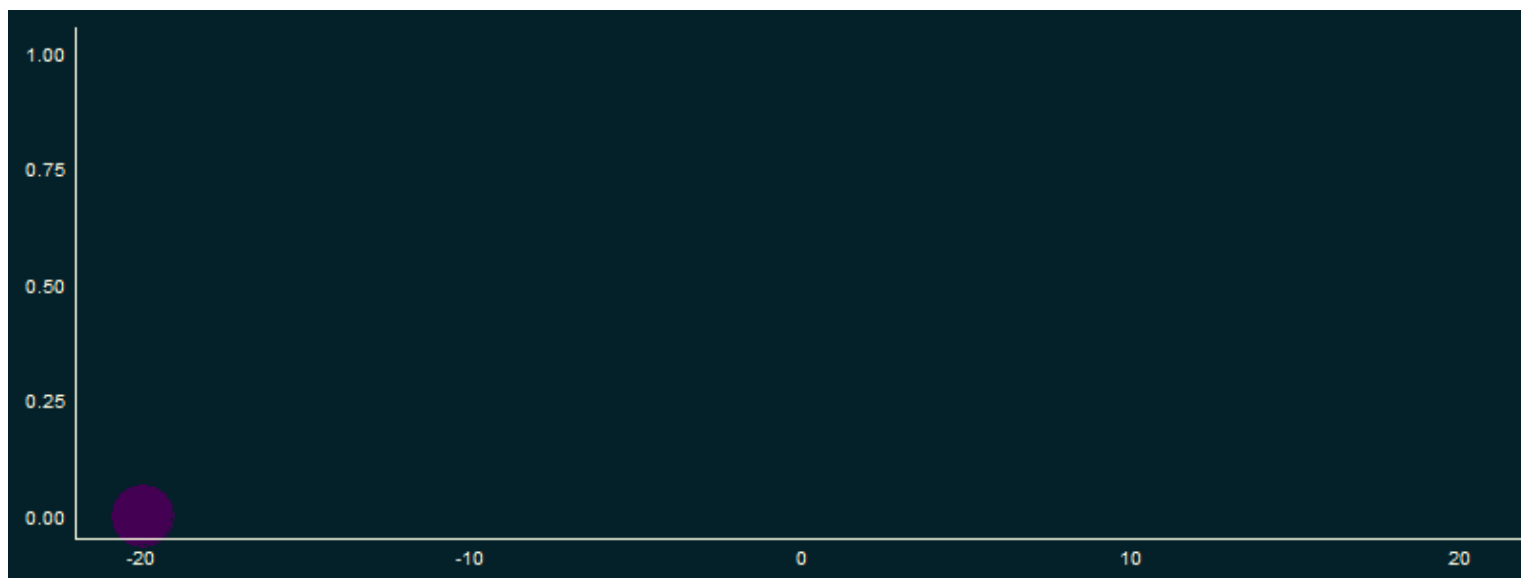
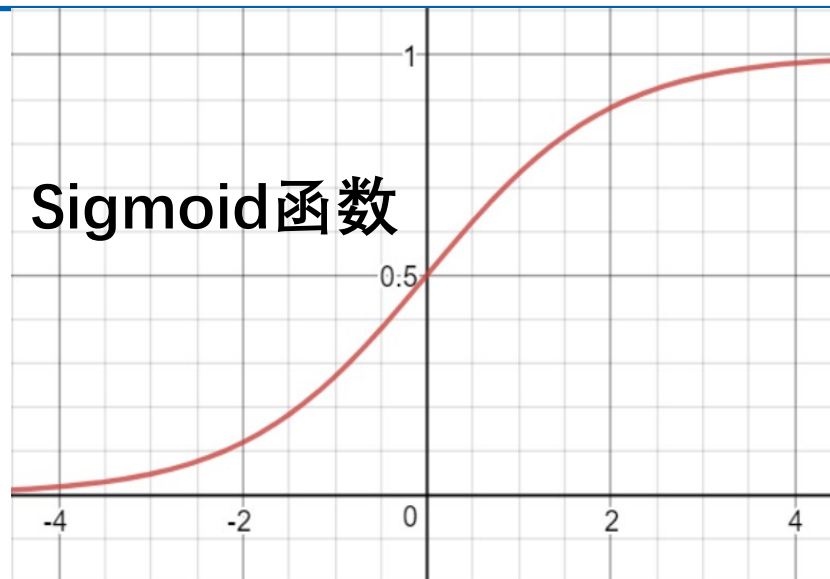
线性回归一个明显的问题是对离群点（和大多数数据点距离较远的数据点，outlier）非常敏感，导致模型建模不稳定，使结果有偏，为了缓解这个问题（特别是在二分类场景中）带来的影响，可考虑逻辑回归(logistic regression)[Cox 1958]。

线性回归：逻辑回归/对数几率回归

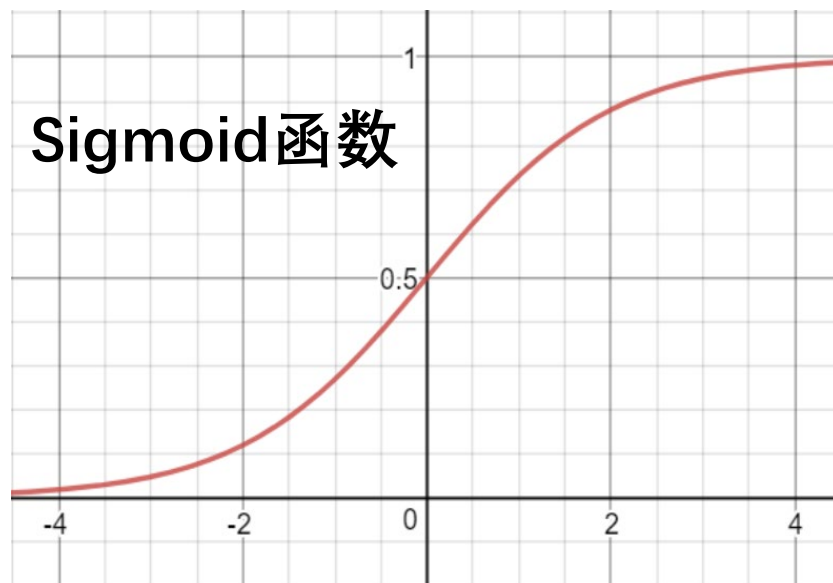
逻辑回归/对数几率回归模型例子

线性回归一个明显的问题是对离群点（和大多数数据点距离较远的数据点，outlier）非常敏感，导致模型建模不稳定，使结果有偏，为了缓解这个问题（特别是在二分类场景中）带来的影响，可考虑逻辑回归(logistic regression)[Cox 1958]。

线性回归：逻辑回归/对数几率回归



线性回归：逻辑回归/对数几率回归



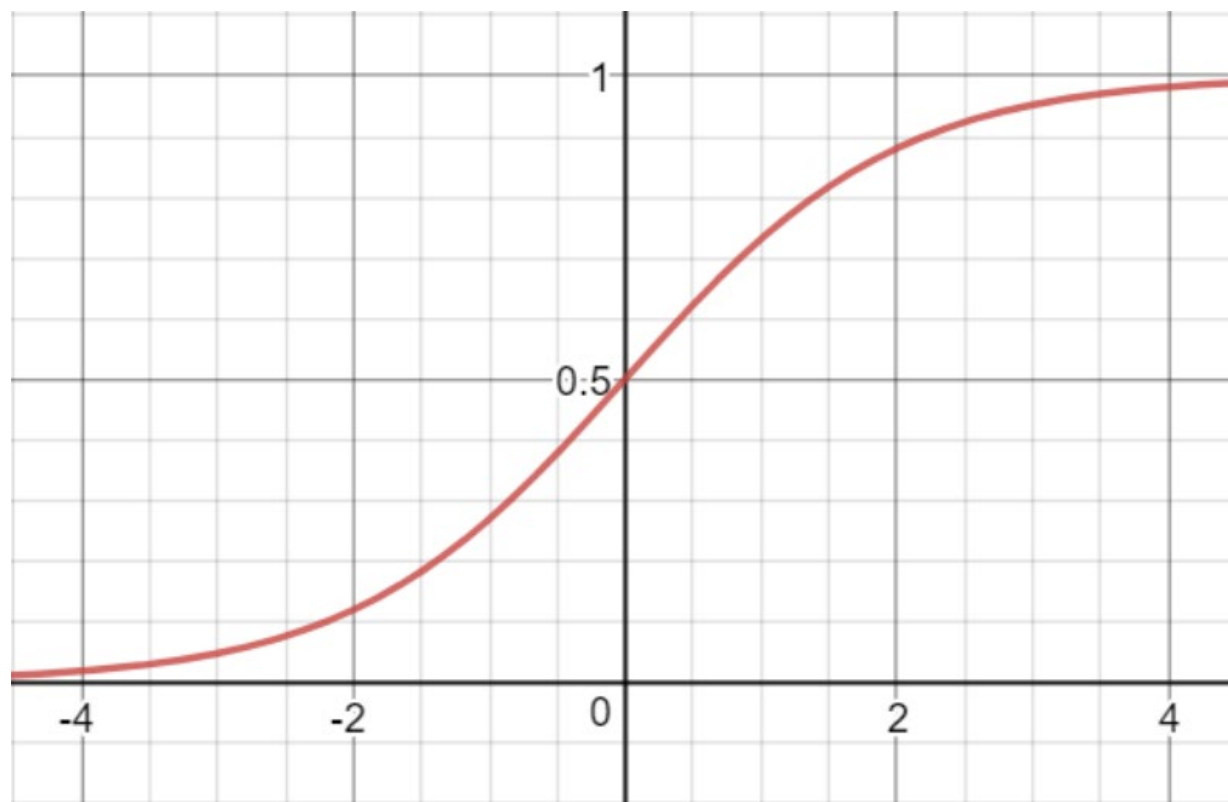
逻辑回归(logistic regression)就是在回归模型中引入 sigmoid函数的一种回归模型。
Logistic回归模型可如下表示：

$$y = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\mathbf{w}^T \mathbf{x} + b)}} \quad , \quad \text{其中 } y \in (0,1), z = \mathbf{w}^T \mathbf{x} + b$$

这里 $\frac{1}{1+e^{-z}}$ 是sigmoid函数、 $\mathbf{x} \in \mathbb{R}^d$ 是输入数据、 $\mathbf{w} \in \mathbb{R}^d$ 和 $b \in \mathbb{R}$ 是回归函数的参数。

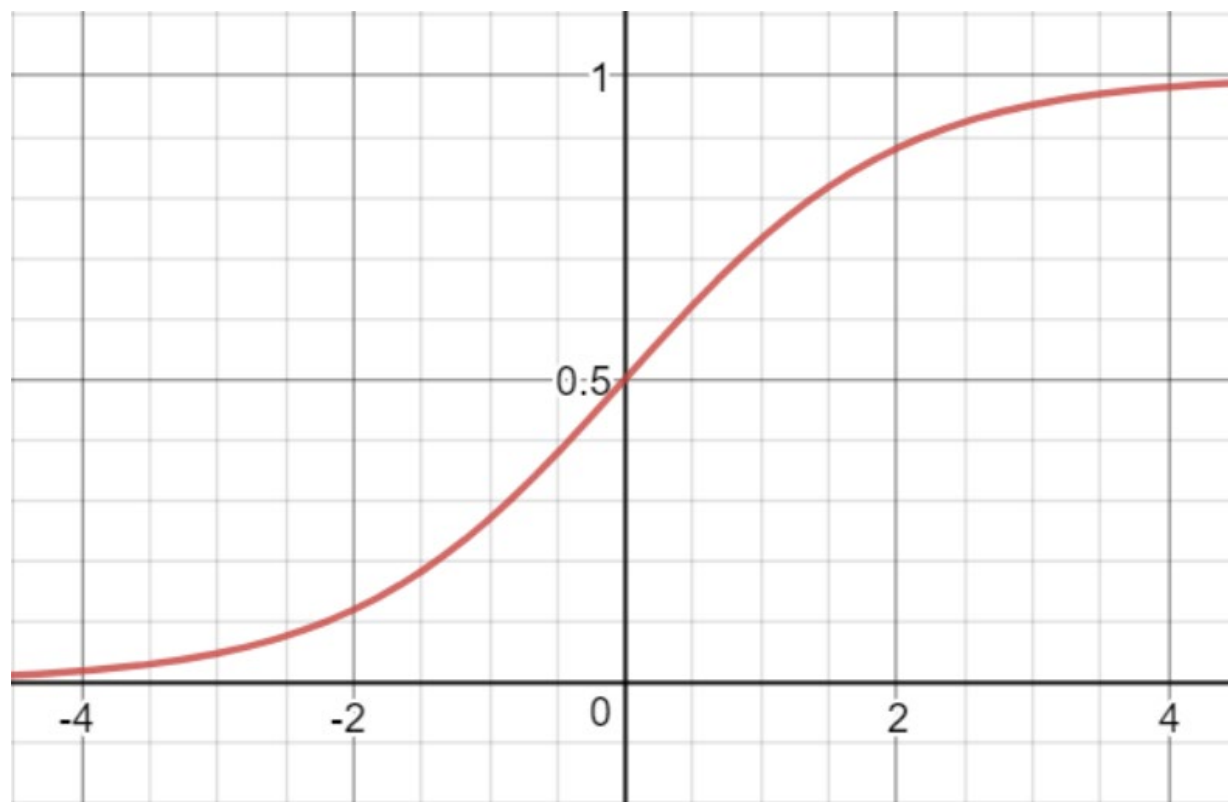
线性回归：逻辑回归/对数几率回归

逻辑回归虽可用于对输入数据和输出结果之间复杂关系进行建模，但由于逻辑回归函数的输出具有概率意义，使得逻辑回归函数更多用于二分类问题（ $y = 1$ 表示输入数据 \mathbf{x} 属于正例， $y = 0$ 表示输入数据 \mathbf{x} 属于负例）。



线性回归：逻辑回归/对数几率回归

$y = \frac{1}{1+e^{-(w^T x+b)}}$ 可用来计算输入数据 \mathbf{x} 属于正例概率，这里 y 理解为输入数据 \mathbf{x} 为正例的概率、 $1 - y$ 理解为输入数据 \mathbf{x} 为负例的概率，即 $p(y = 1|\mathbf{x})$ 。



线性回归：逻辑回归/对数几率回归

$y = \frac{1}{1+e^{-(\mathbf{w}^T \mathbf{x} + b)}}$ 可用来计算输入数据 \mathbf{x} 属于正例概率，这里 y 理解为输入数据 \mathbf{x} 为正例的概率、 $1 - y$ 理解为输入数据 \mathbf{x} 为负例的概率，即 $p(y = 1|\mathbf{x})$ 。

我们现在对比值 $\frac{p}{1-p}$ 取对数(即 $\log\left(\frac{p}{1-p}\right)$) 来表示输入数据 \mathbf{x} 属于正例概率。 $\frac{p}{1-p}$ 被称为几率(odds)，反映了输入数据 \mathbf{x} 作为正例的相对可能性。 $\frac{p}{1-p}$ 的对数几率(log odds)或logit函数可表示为 $\log\left(\frac{p}{1-p}\right)$ 。

线性回归：逻辑回归/对数几率回归

可以得到

$$p(y = 1|\mathbf{x}) = h_{\theta}(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

$$p(y = 0|\mathbf{x}) = 1 - h_{\theta}(\mathbf{x}) = \frac{e^{-(\mathbf{w}^T \mathbf{x} + b)}}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

θ 表示模型参数 $\theta = \{\mathbf{w}, b\}$ ，于是有：

$$\begin{aligned} & \text{logit}(p(y = 1|\mathbf{x})) \\ &= \log\left(\frac{p(y = 1|\mathbf{x})}{p(y = 0|\mathbf{x})}\right) \\ &= \log\left(\frac{p}{1 - p}\right) \\ &= \mathbf{w}^T \mathbf{x} + b \end{aligned}$$

线性回归：逻辑回归/对数几率回归

- 如果输入数据 \mathbf{x} 属于正例的概率大于其属于负例的概率，即 $p(y = 1|\mathbf{x}) > 0.5$ ，则输入数据 \mathbf{x} 可被判断属于正例。这一结果等价于

$$\frac{p(y = 1|\mathbf{x})}{p(y = 0|\mathbf{x})} > 1, \quad \text{即} \log \left(\frac{p(y = 1|\mathbf{x})}{p(y = 0|\mathbf{x})} \right) > \log 1 = 0, \quad \text{也就是} \mathbf{w}^T \mathbf{x} + b > 0$$

成立。

- 从这里可以看出，logistic回归是一个广义线性模型。在预测时，可以计算线性函数 $\mathbf{w}^T \mathbf{x} + b$ 取值是否大于0来判断输入数据 \mathbf{x} 的类别归属。

线性回归：逻辑回归/对数几率回归

模型参数的似然函数被定义为 $\mathcal{L}(\theta|\mathcal{D}) = p(\mathcal{D}|\theta)$ ，其中 $\mathcal{D} = \{(x_i, y_i) | 1 \leq i \leq n\}$ 表示所有观测数据（或训练数据）， θ 表示模型参数（ $\theta = \{\mathbf{w}, b\}$ ）。

线性回归：逻辑回归/对数几率回归

模型参数的似然函数被定义为 $\mathcal{L}(\theta|\mathcal{D}) = p(\mathcal{D}|\theta)$ ，其中 $\mathcal{D} = \{(x_i, y_i) | 1 \leq i \leq n\}$ 表示所有观测数据（或训练数据）， θ 表示模型参数（ $\theta = \{\mathbf{w}, b\}$ ）。

在最大化对数似然函数过程中，一般假设观测所得每一个样本数据是独立同分布 (independent and identically distributed, i.i.d)，于是可得：

$$\mathcal{L}(\theta|\mathcal{D}) = p(\mathcal{D}|\theta) = \prod_{i=1}^n p(y_i|x, \theta) = \prod_{i=1}^n (h_{\theta}(x_i))^{y_i} (1 - h_{\theta}(x_i))^{1-y_i}$$

线性回归：逻辑回归/对数几率回归

模型参数的似然函数被定义为 $\mathcal{L}(\theta|\mathcal{D}) = p(\mathcal{D}|\theta)$ ，其中 $\mathcal{D} = \{(x_i, y_i) | 1 \leq i \leq n\}$ 表示所有观测数据（或训练数据）， θ 表示模型参数（ $\theta = \{\mathbf{w}, b\}$ ）。

在最大化对数似然函数过程中，一般假设观测所得每一个样本数据是独立同分布 (independent and identically distributed, i.i.d)，于是可得：

$$\mathcal{L}(\theta|\mathcal{D}) = p(\mathcal{D}|\theta) = \prod_{i=1}^n p(y_i|x_i, \theta) = \prod_{i=1}^n (h_{\theta}(x_i))^{y_i} (1 - h_{\theta}(x_i))^{1-y_i}$$

对上述公式取对数：

$$l(\theta) = \log(\mathcal{L}(\theta|\mathcal{D})) = \sum_{i=1}^n (y_i \log(h_{\theta}(x_i)) + (1 - y_i) \log(1 - h_{\theta}(x_i)))$$

线性回归：逻辑回归/对数几率回归

最大似然估计目的是计算似然函数的最大值，而分类过程是需要损失函数最小化。因此，在上式前加一个负号得到损失函数(交叉熵)：

$$\begin{aligned}\mathcal{J}(\theta) &= -l(\theta) = -\log(L(\theta|\mathcal{D})) \\ &= -\left(\sum_{i=1}^n y_i \log(h_{\theta}(x_i)) + (1 - y_i)\log(1 - h_{\theta}(x_i))\right)\end{aligned}$$

$$\mathcal{J}(\theta)\text{等价于: } \mathcal{J}(\theta) = \begin{cases} -\log(h_{\theta}(x_i)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x_i)) & \text{if } y = 0 \end{cases}$$

线性回归：逻辑回归/对数几率回归

需要最小化损失函数来求解参数。数损失函数对参数 θ 的偏导如下（其中， $h'_\theta(x) = h_\theta(x)(1 - h_\theta(x))$, $\log' x = \frac{1}{x}$ ）

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta_j} &= - \sum_{i=1}^n \left(y_i \frac{1}{h_\theta(x_i)} \frac{\partial h_\theta(x_i)}{\partial \theta_j} + (1 - y_i) \frac{1}{1 - h_\theta(x_i)} \frac{\partial (1 - h_\theta(x_i))}{\partial \theta_j} \right) \\ &= - \sum_{i=1}^n \frac{\partial h_\theta(x_i)}{\partial \theta_j} \left(\frac{y_i}{h_\theta(x_i)} - \frac{1 - y_i}{1 - h_\theta(x_i)} \right) \\ &= - \sum_{i=1}^n x_i h_\theta(x_i) (1 - h_\theta(x_i)) \left(\frac{y_i}{h_\theta(x_i)} - \frac{1 - y_i}{1 - h_\theta(x_i)} \right) \\ &= - \sum_{i=1}^n x_i (y_i (1 - h_\theta(x_i)) - (1 - y_i) h_\theta(x_i)) \\ &= \sum_{i=1}^n (h_\theta(x_i) - y_i) x_i\end{aligned}$$

将求导结果代入梯度下降迭代公式得：（勘误：P124）

$$\theta_j = \theta_j - \eta \sum_{i=1}^n (h_\theta(x_i) - y_i) x_i$$

一、机器学习基本概念

二、回归分析

三、决策树

四、线性区别分析

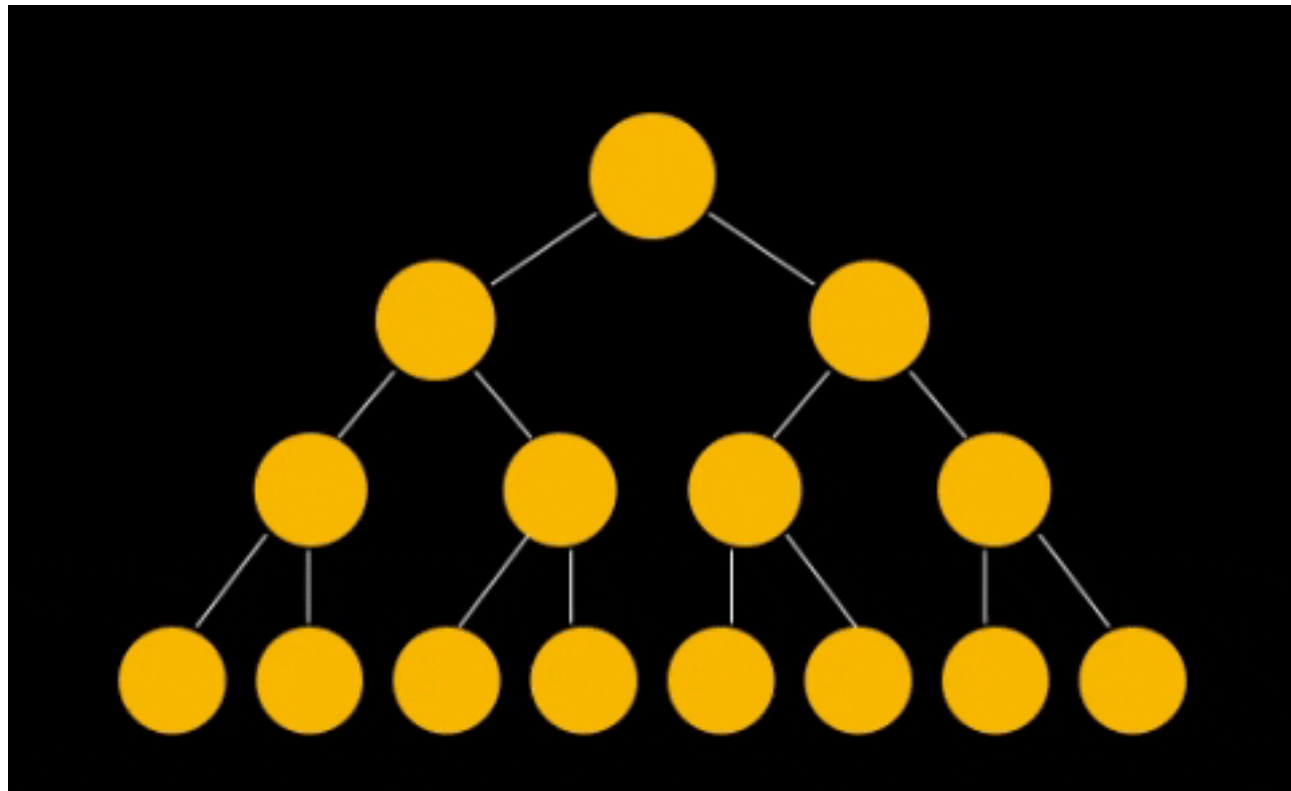
五、Ada Boosting

六、支持向量机

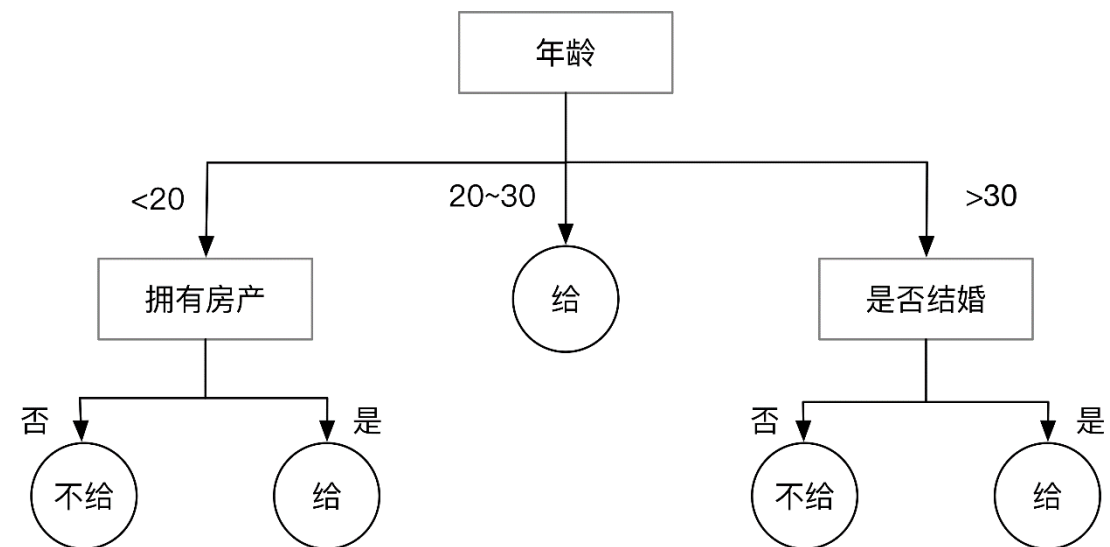
七、生成学习模型

决策树

决策树是一种通过树形结构来进行分类的方法。在决策树中，树形结构中每个非叶子节点表示对分类目标在某个属性上的一个判断，每个分支代表基于该属性做出的一个判断，最后树形结构中每个叶子节点代表一种分类结果，所以决策树可以看作是一系列以叶子节点为输出的决策规则（Decision Rules）[Quinlan 1987]。



决策树



决策树： 案例

贷款与其它因素的关系

序号	年龄	银行流水	是否结婚	拥有房产	是否给予贷款
1	>30	高	否	是	否
2	>30	高	否	否	否
3	20~30	高	否	是	是
4	<20	中	否	是	是
5	<20	低	否	是	是
6	<20	低	是	否	否
7	20~30	低	是	否	是
8	>30	中	否	是	否
9	>30	低	是	是	是
10	<20	中	否	是	是
11	>30	中	是	否	是
12	20~30	中	否	否	是
13	20~30	高	是	是	是
14	<20	中	否	否	否

决策树：信息熵



决策树：信息熵

假设有 K 个信息，其组成了集合样本 D ，记第 k 个信息发生的概率为 $p_k (1 \leq k \leq K)$ ”。如下定义这 K 个信息的信息熵：

$$E(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

$E(D)$ 值越小，表示 D 包含的信息越确定，也称 D 的纯度越高。需要指出，所有 p_k 累加起来的和为1。

决策树：信息熵

要点：构建决策树时划分属性的顺序选择是重要的。性能好的决策树随着划分不断进行，决策树分支结点样本集的“纯度”会越来越高，即其所包含样本尽可能属于相同类别。

年龄属性划分后子样本集情况统计

年龄属性 取值 a_i	">30"	"20~30"	"<20"
对应样 本数 $ D_i $	5	4	5
正负样本 数量	(2+, 3-)	(4+, 0-)	(3+, 2-)

决策树：信息熵

序号	年龄	是否给予贷款
1	>30	否
2	>30	否
3	20~30	是
4	<20	是
5	<20	是
6	<20	否
7	20~30	是
8	>30	否
9	>30	是
10	<20	是
11	>30	是
12	20~30	是
13	20~30	是
14	<20	否

$$E(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

年龄属性划分后子样本集情况统计

年龄属性取值	">30"	"20~30"	"<20"
对应样本数	5	4	5
正负样本数量	(2+, 3-)	(4+, 0-)	(3+, 2-)

决策树：信息熵

序号	年龄	是否给予贷款
1	>30	否
2	>30	否
3	20~30	是
4	<20	是
5	<20	是
6	<20	否
7	20~30	是
8	>30	否
9	>30	是
10	<20	是
11	>30	是
12	20~30	是
13	20~30	是
14	<20	否

$$E(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

年龄属性划分后子样本集情况统计

年龄属性取值	">30"	"20~30"	"<20"
对应样本数	5	4	5
正负样本数量	(2+, 3-)	(4+, 0-)	(3+, 2-)

“年龄 > 30”: $Ent(D_0)$

$$= - \left(\frac{2}{5} \times \log_2 \frac{2}{5} + \frac{3}{5} \times \log_2 \frac{3}{5} \right) = 0.971$$

决策树：信息熵

序号	年龄	是否给予贷款
1	>30	否
2	>30	否
3	20~30	是
4	<20	是
5	<20	是
6	<20	否
7	20~30	是
8	>30	否
9	>30	是
10	<20	是
11	>30	是
12	20~30	是
13	20~30	是
14	<20	否

$$E(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

年龄属性划分后子样本集情况统计

年龄属性取值	">30"	"20~30"	"<20"
对应样本数	5	4	5
正负样本数量	(2+, 3-)	(4+, 0-)	(3+, 2-)

“年龄 > 30”: $Ent(D_0)$

$$= - \left(\frac{2}{5} \times \log_2 \frac{2}{5} + \frac{3}{5} \times \log_2 \frac{3}{5} \right) = 0.971$$

“年龄 20~30”: $Ent(D_1) = - \left(\frac{4}{4} \times \log_2 \frac{4}{4} + 0 \right) = 0$

决策树：信息熵

序号	年龄	是否给予贷款
1	>30	否
2	>30	否
3	20~30	是
4	<20	是
5	<20	是
6	<20	否
7	20~30	是
8	>30	否
9	>30	是
10	<20	是
11	>30	是
12	20~30	是
13	20~30	是
14	<20	否

$$E(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

年龄属性划分后子样本集情况统计

年龄属性取值	">30"	"20~30"	"<20"
对应样本数	5	4	5
正负样本数量	(2+, 3-)	(4+, 0-)	(3+, 2-)

“年龄 > 30”: $Ent(D_0)$
 $= - \left(\frac{2}{5} \times \log_2 \frac{2}{5} + \frac{3}{5} \times \log_2 \frac{3}{5} \right) = 0.971$

“年龄 20~30”: $Ent(D_1) = - \left(\frac{4}{4} \times \log_2 \frac{4}{4} + 0 \right) = 0$

“年龄 < 20”: $Ent(D_2)$
 $= - \left(\frac{3}{5} \times \log_2 \frac{3}{5} + \frac{2}{5} \times \log_2 \frac{2}{5} \right) = 0.971$

决策树：信息增益

得到上述三个的信息熵后，可进一步计算使用年龄属性对原样本集进行划分后的信息增益，计算公式如下：

$$Gain(D, A) = Ent(D) - \sum_{i=1}^n \frac{|D_i|}{|D|} Ent(D_i)$$

决策树：信息增益

得到上述三个的信息熵后，可进一步计算使用年龄属性对原样本集进行划分后的信息增益，计算公式如下：

$$Gain(D, A) = \boxed{Ent(D)} - \sum_{i=1}^n \frac{|D_i|}{|D|} Ent(D_i)$$

$$E(D) = - \sum_{k=1}^K p_k \log_2 p_k$$

= 0.940

序号	是否给予贷款
1	否
2	否
3	是
4	是
5	是
6	否
7	是
8	否
9	是
10	是
11	是
12	是
13	是
14	否

决策树：信息增益

得到上述三个的信息熵后，可进一步计算使用年龄属性对原样本集进行划分后的信息增益，计算公式如下：

$$Gain(D, A) = Ent(D) - \sum_{i=1}^n \frac{|D_i|}{|D|} Ent(D_i)$$

将 $A = \text{年龄}$ 代入。于是选择年龄这一属性划分后的信息增益为：

$$Gain(D, \text{年龄}) = 0.940 - \left(\frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 \right) = 0.246$$

同理，可以计算银行流水、是否结婚、是否拥有房产三个人物属性的信息增益。通过比较四种属性信息增益的高低来选择最佳属性对原样本集进行划分，得到最大的“纯度”。如果划分后的不同子样本集都只存在同类样本，那么停止划分。

决策树：构建决策树

$info$ 和 $Gain - ratio$ 计算公式如下：

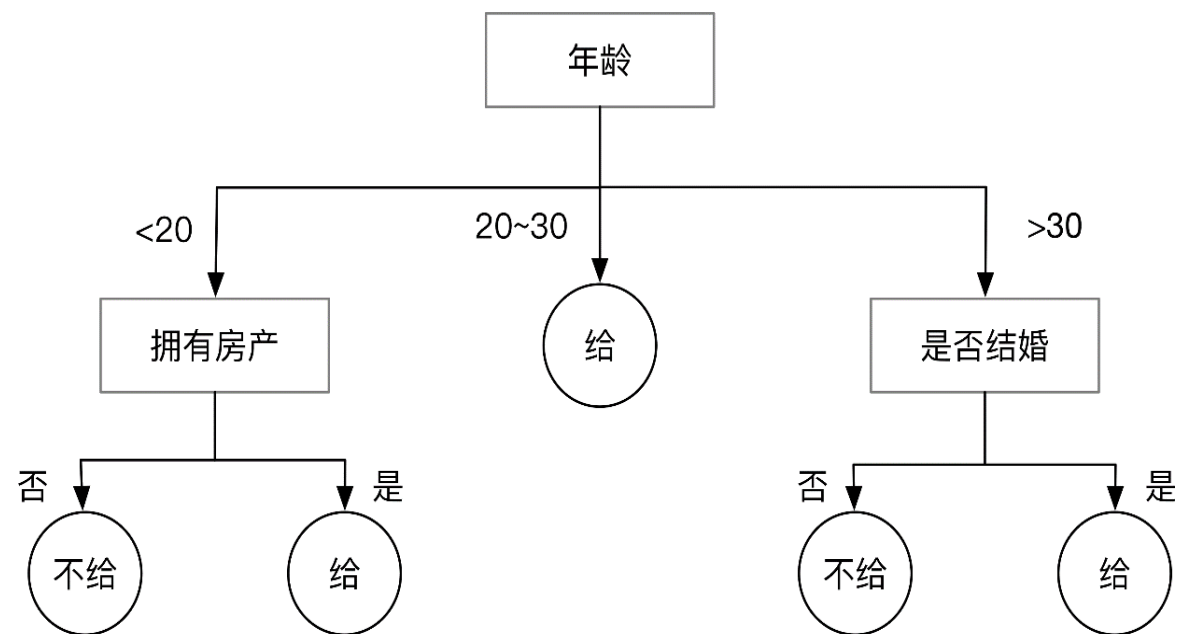
$$info = - \sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$$

$$Gain - ratio = Gain(D, A) / info$$

另一种计算更简的度量指标是如下的Gini系数：

$$Gini(D) = 1 - \sum_{k=1}^K p_k^2$$

相对于信息熵的计算 $E(D) = - \sum_{k=1}^K p_k \log_2 p_k$ ，不用计算对数 \log ，计算更为简易。



决策树：例题

样本	属性		分类
	x_1	x_2	
1	T	T	+
2	T	T	+
3	T	F	-
4	F	F	+
5	F	T	-
6	F	T	-

设训练集如表所示，请用经典的 **ID3 算法** 完成其学习过程。
(注意： $\log_2(x/y) = \log_2 x - \log_2 y$, $\log_2 1 = 0$, $\log_2 2 = 1$,
 $\log_2 3 = 1.585$, $\log_2 4 = 2$, $\log_2 5 = 2.322$, $\log_2 6 = 2.585$)

思路：使用ID3算法，计算信息增益

决策树：例题

步骤一：计算出集合D的**总信息熵**

在决策树学习开始时，根结点包含D中的所有样例，其中正例占 $p_1 = \frac{3}{6} = \frac{1}{2}$ ，反例占 $p_2 = \frac{1}{2}$ ，于是根结点的信息熵为：

$$Ent(D) = -(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}) = 1$$

样本	属性		分类
	x_1	x_2	
1	T	T	+
2	T	T	+
3	T	F	-
4	F	F	+
5	F	T	-
6	F	T	-

决策树：例题

步骤二：计算每个属性的信息熵

属性 x_1 ：包含 $D^1(T)$ 和 $D^2(F)$ ，各占 $\frac{1}{2}$

$D^1(T)$ ：正例占 $p_1 = \frac{2}{3}$ ，反例占 $p_2 = \frac{1}{3}$

$$Ent(D^1) = -(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}) = 0.9183$$

$D^2(F)$ ：正例占 $p_1 = \frac{1}{3}$ ，反例占 $p_2 = \frac{2}{3}$

$$Ent(D^2) = -(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}) = 0.9183$$

$$\text{因此 } Ent(x_1) = \frac{1}{2}Ent(D^1) + \frac{1}{2}Ent(D^2) = 0.9183$$

样本	属性		分类
	x_1	x_2	
1	T	T	+
2	T	T	+
3	T	F	-
4	F	F	+
5	F	T	-
6	F	T	-

决策树：例题

步骤二：计算每个属性的信息熵

属性 x_2 ：包含 $D^1(T)$ 和 $D^2(F)$ ，分别占 $\frac{2}{3}$ 和 $\frac{1}{3}$

$D^1(T)$ ：正例占 $p_1 = \frac{1}{2}$ ，反例占 $p_2 = \frac{1}{2}$

$$Ent(D^1) = -(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}) = 1$$

$D^2(F)$ ：正例占 $p_1 = \frac{1}{2}$ ，反例占 $p_2 = \frac{1}{2}$

$$Ent(D^2) = -(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}) = 1$$

$$\text{因此 } Ent(x_2) = \frac{2}{3} Ent(D^1) + \frac{1}{3} Ent(D^2) = 1$$

样本	属性		分类
	x_1	x_2	
1	T	T	+
2	T	T	+
3	T	F	-
4	F	F	+
5	F	T	-
6	F	T	-

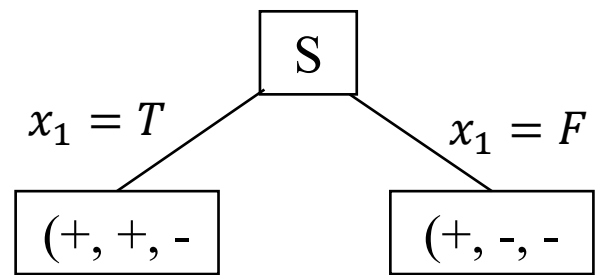
决策树：例题

步骤三：计算每个属性的信息增益

$$\begin{aligned} \text{Gain}(D, x_1) &= \text{Ent}(D) - \text{Ent}(x_1) = 1 - 0.9183 \\ &= 0.0817 \end{aligned}$$

$$\text{Gain}(D, x_2) = \text{Ent}(D) - \text{Ent}(x_2) = 1 - 1 = 0$$

选择**信息增益大**的作为第一个属性，即选择属性 x_1 对根节点进行扩展



样本	属性		分类
	x_1	x_2	
1	T	T	+
2	T	T	+
3	T	F	-
4	F	F	+
5	F	T	-
6	F	T	-

谢谢!