

监督学习

主讲：刘夏雷、郭春乐、王亚星

南开大学计算机学院

<https://mmcheng.net/xliu/>

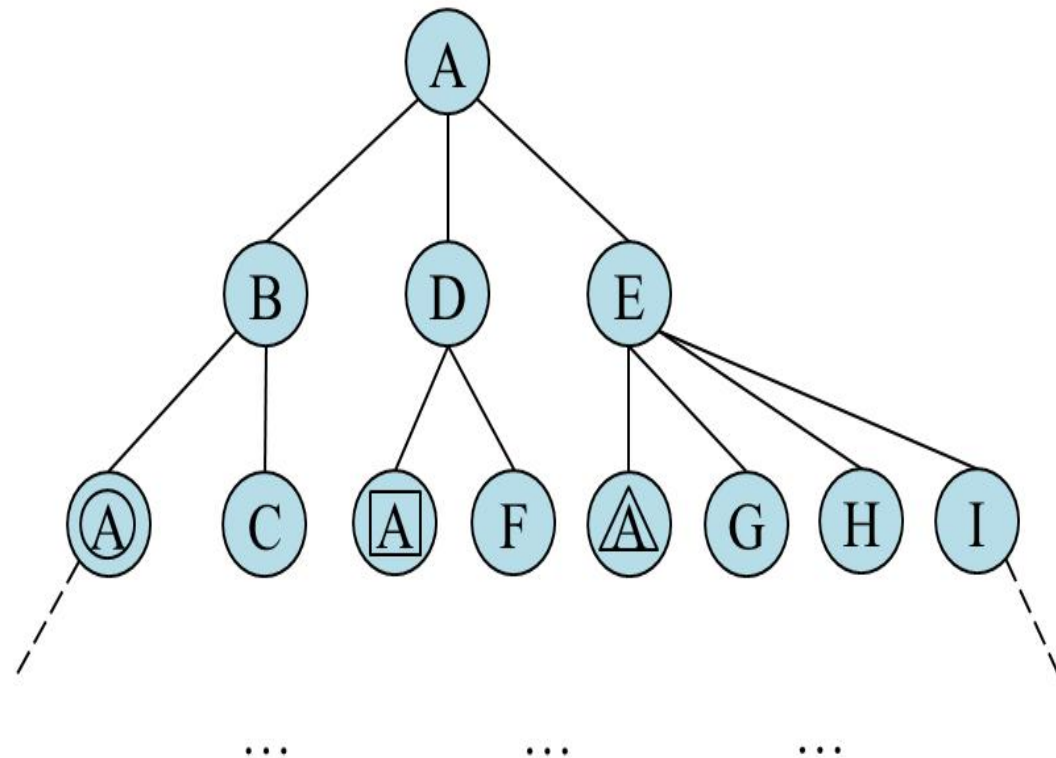
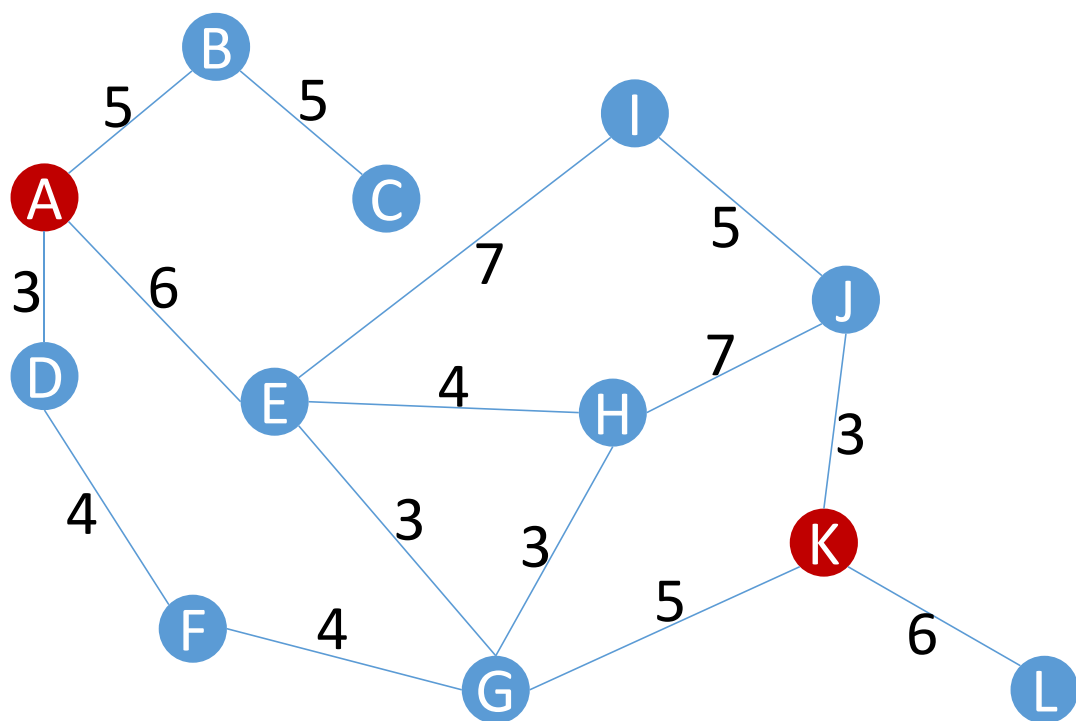
致谢：本课件主要内容来自浙江大学吴飞教授、
南开大学程明明教授

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

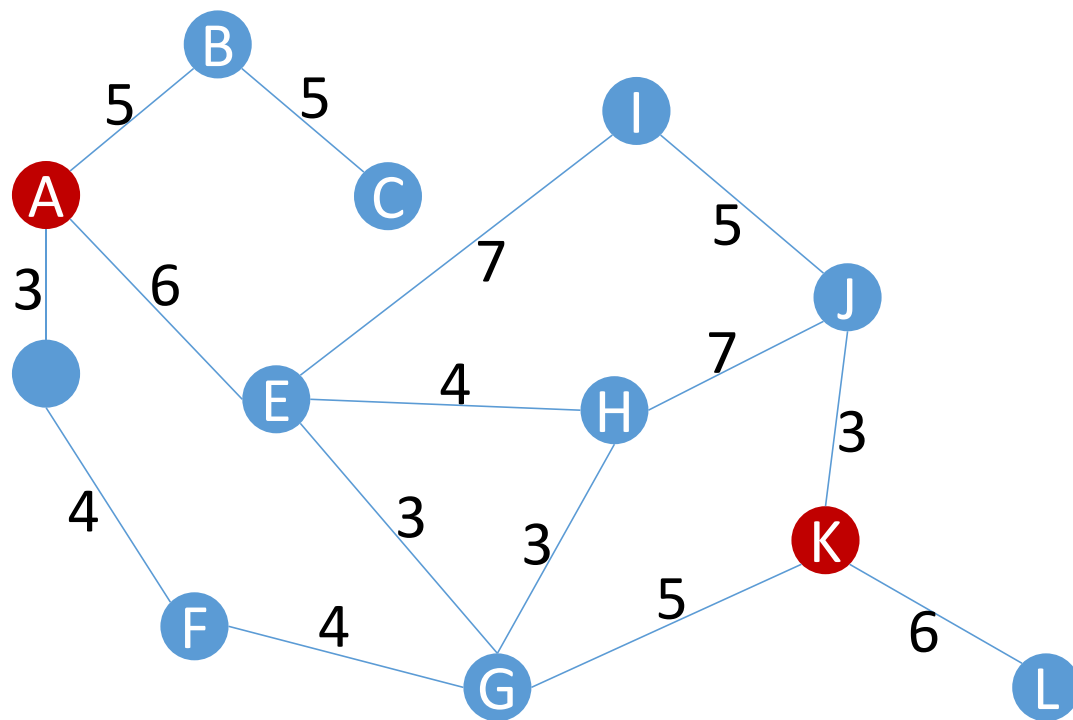
搜索树：用一棵树来记录算法探索过的路径

- 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。

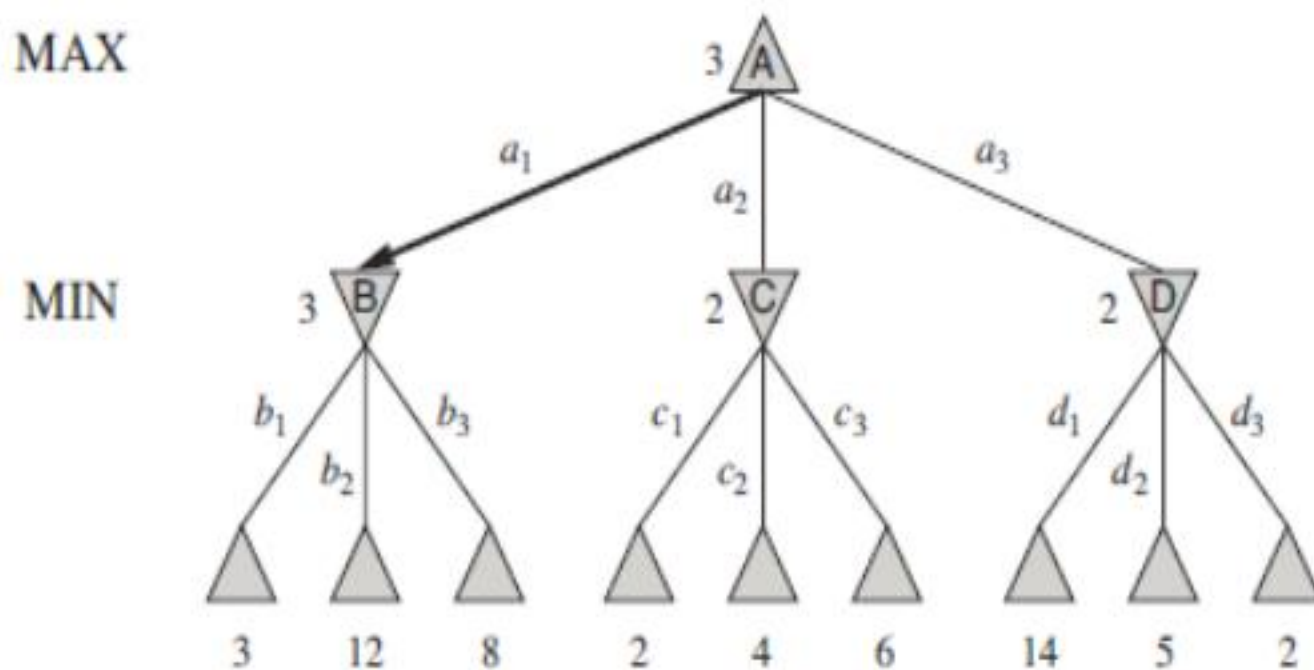
- **蒙特卡洛树搜索(Monte-Carlo Tree Search)**

- 通过采样而非穷举方法来实现搜索。

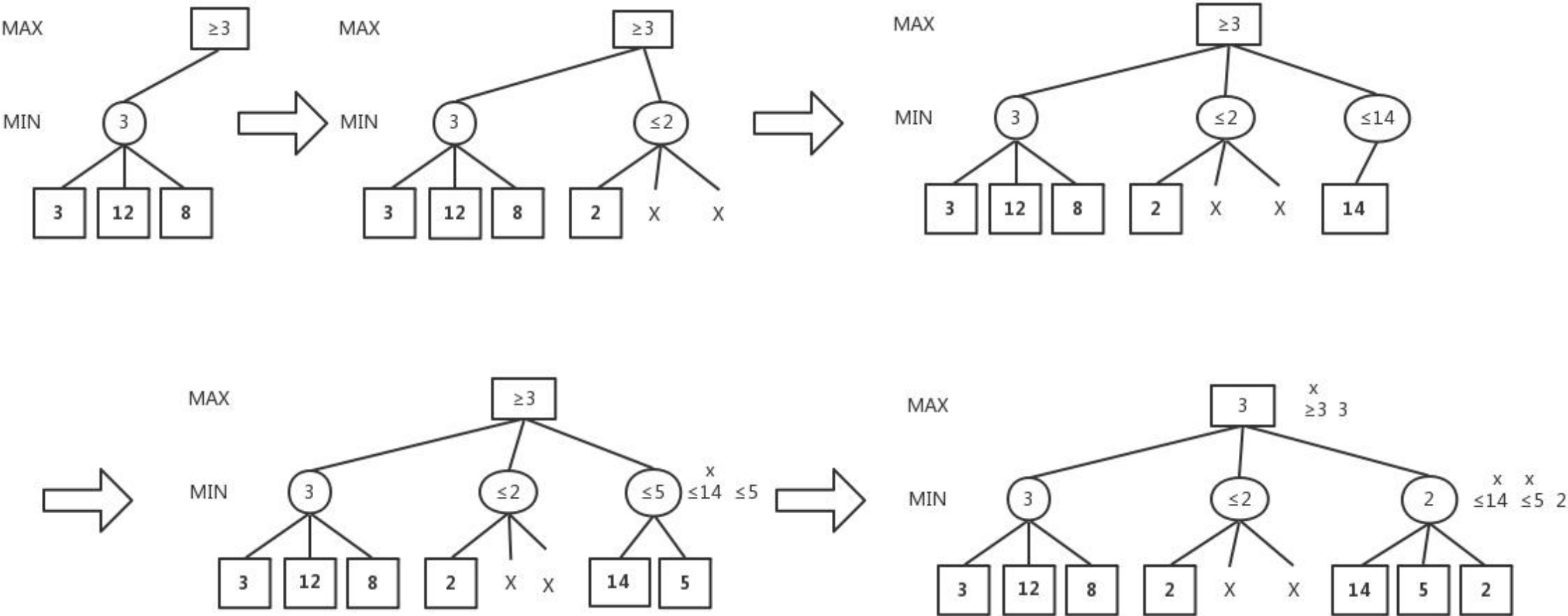
对抗搜索：Alpha-Beta 剪枝搜索

- 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。



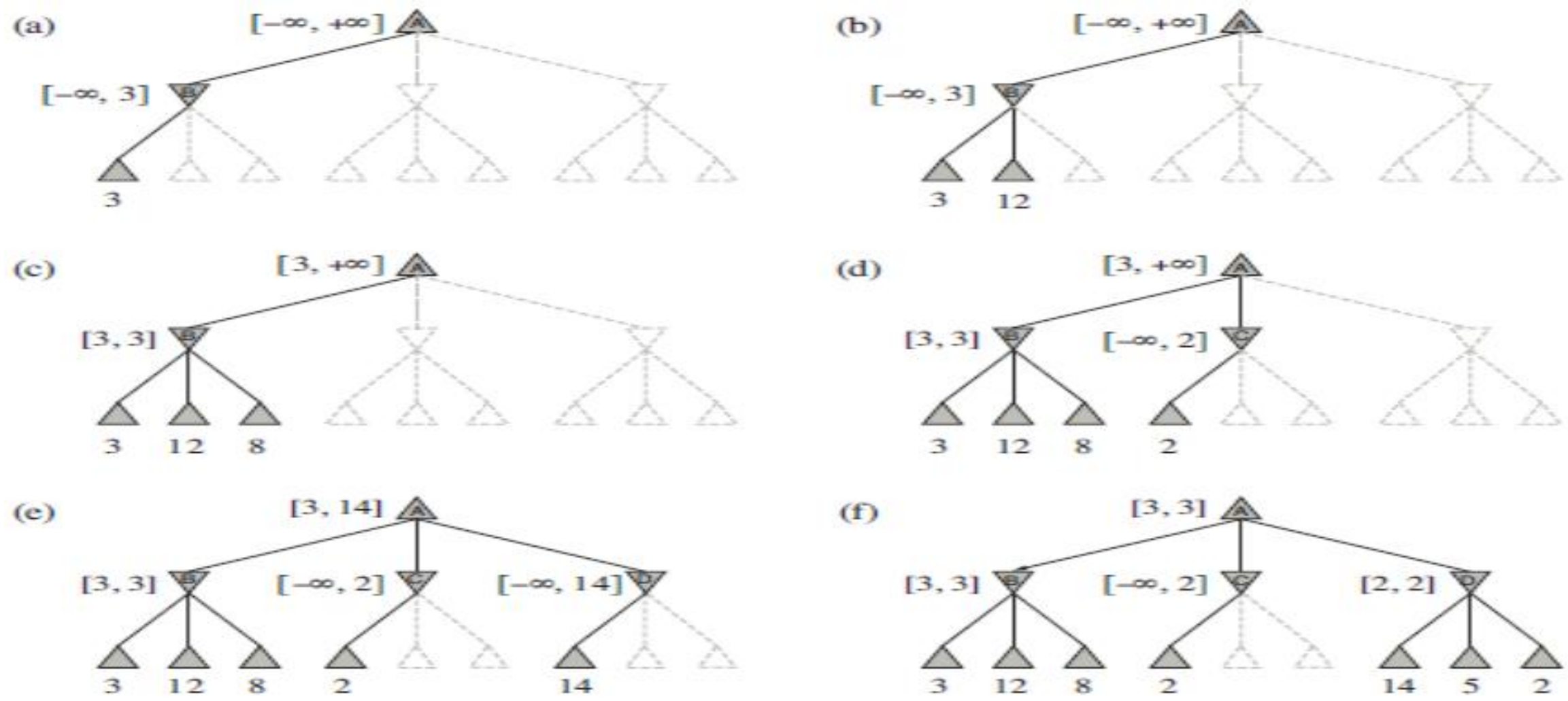
对抗搜索：Alpha-Beta 剪枝搜索



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

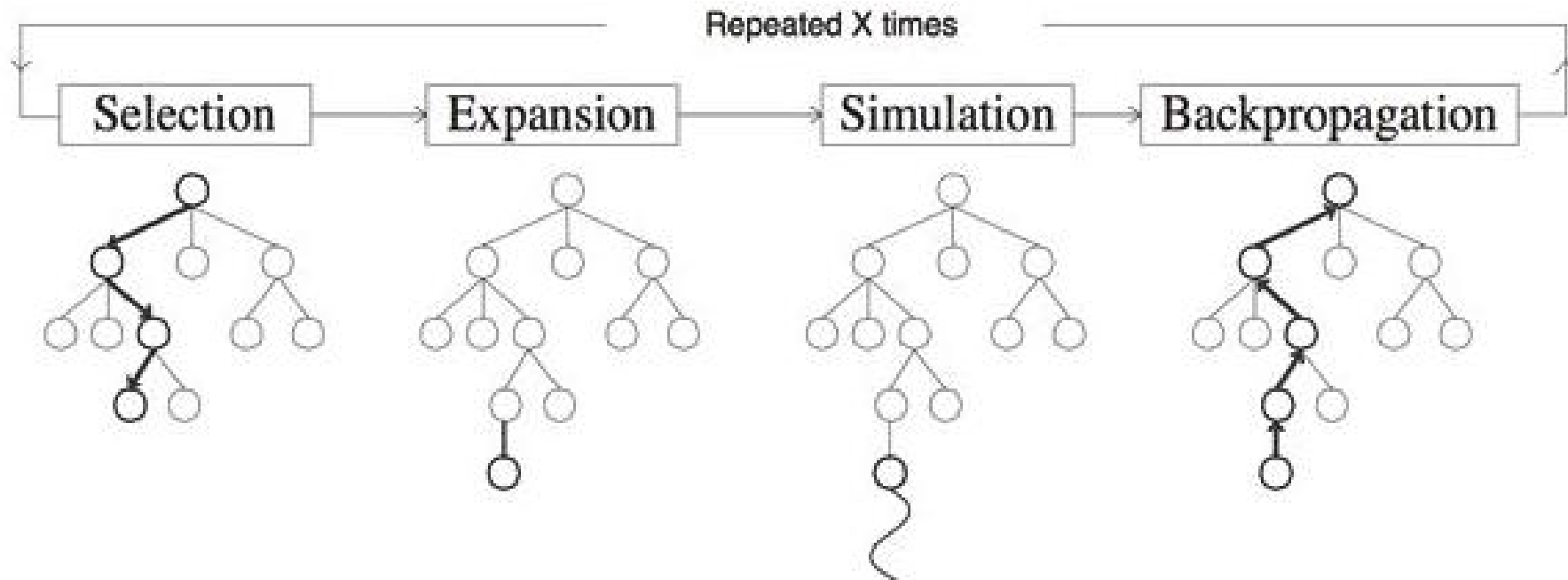
- 从 α 和 β 的变化来理解剪枝过程



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

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



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。

习题

第1题 | 单选题 1分

1.下列关于探索与利用的说法中,不正确的是()。

- ☐ A 在多臂赌博机问题中,过度探索会导致算法很少主动去选择比较好的摇臂。
- ☐ B 在多臂赌博机问题中,过度利用可能导致算法对部分臂膀额奖励期望估计不准确。
- ☐ C 在 ϵ 贪心算法中, ϵ 的值越大,表示算法越倾向于探索。
- ☒ D 在多臂赌博机问题中,某时刻UCB1算法选择的臂膀置信上界为 R ,则此时任意摇动一个臂膀,得到的硬币数量不会超过 R 。

第2题 | 单选题 1分

2.下列关于蒙特卡洛树搜索算法的说法中,不正确的是()。

- ☐ A 选择过程体现了探索与利用的平衡。
- ☒ B 算法进入扩展步骤时,当前节点的所有子节点必然都未被扩展。
- ☐ C 模拟步骤采取的策略与选择步骤不一定要相同。
- ☐ D 反向传播只需要更新当前路径上已被扩展的节点。

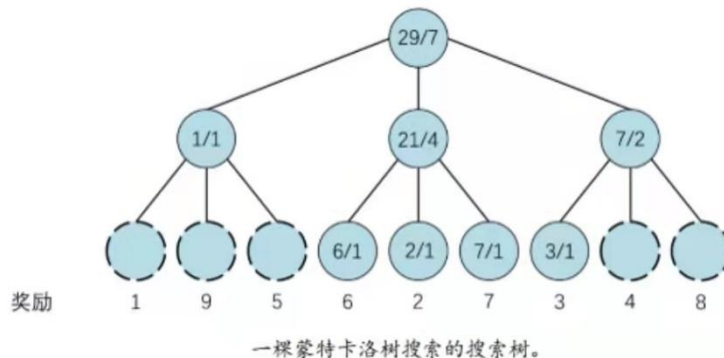
第3题 | 主观题 10分

10.下图展示了一个蒙特卡洛树搜索的例子。其中每个叶子节点(终止节点)下标出了该节点对应的奖励。为了最大化取得的奖励,可利用蒙特卡洛树搜索求解奖励最大的路径。假设执行了若干步骤后,算法的状态如图3所示,节点内的数字分别表示“总奖励/访问次数”,虚线节点表示尚未扩展的节点。算法此时正要开始下一轮选择-扩展-模拟-反向传播的迭代

(1)假设UCB1算法中的超参数,请计算并画出算法选择过程经过的路径。

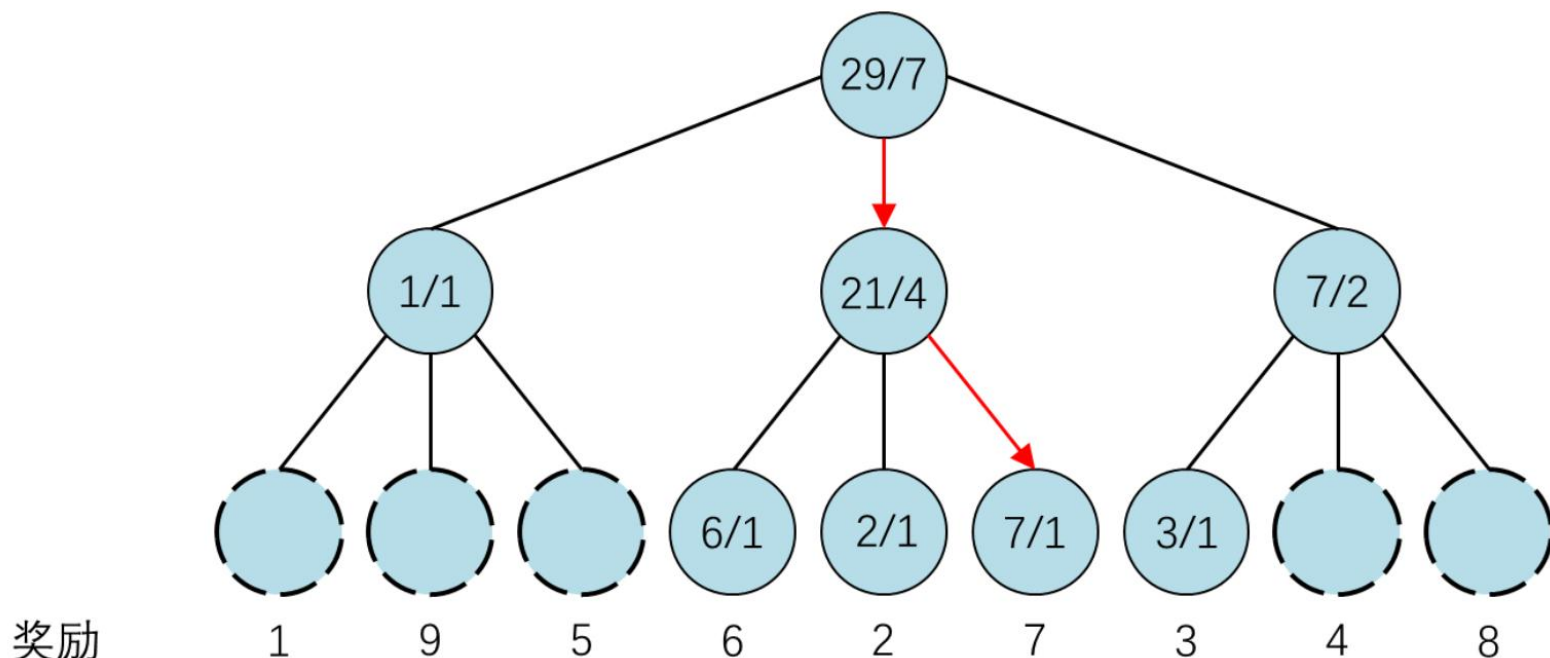
(2)请继续执行扩展、模拟、反向传播步骤,并画出完成后的搜索树状态。(为了避免随机性,假设扩展总是扩展最左侧的未扩展节点,模拟总是选择最左侧的路径。)

(3)尝试进行若干次迭代,请问此时算法是否能有效地找到奖励最大的叶子结点(奖励为9),那么进行足够多次迭代以后又如何?如果希望提高算法的效率,应该做出怎样的调整?



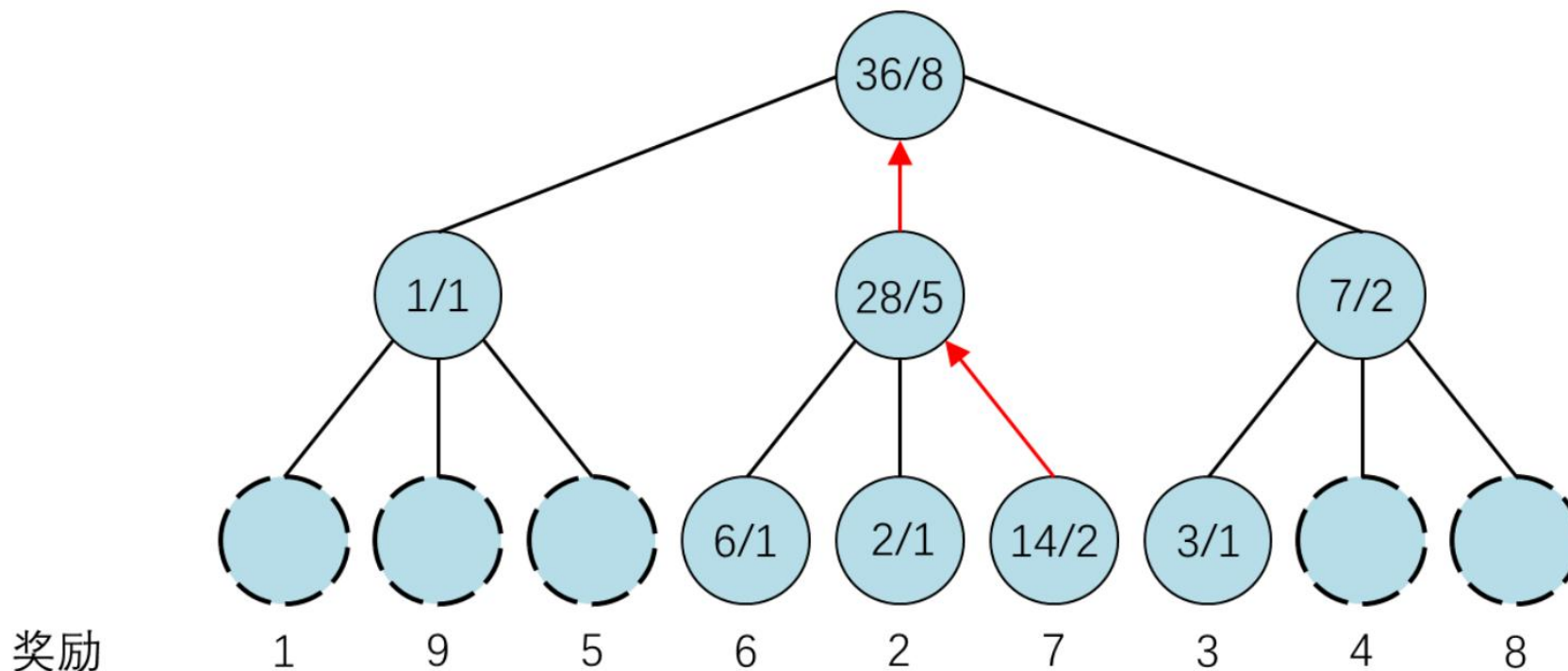
课上习题

(1) 第一步三个节点的 UCB 值从左到右分别为 $\frac{1}{1} + \sqrt{\frac{2\ln 7}{1}} = 2.97$, $\frac{21}{4} + \sqrt{\frac{2\ln 7}{4}} = 6.24$, $\frac{7}{2} + \sqrt{\frac{2\ln 7}{2}} = 4.89$, 因此第一步选择第二层中间的节点。第二步三个节点的 UCB 值从左到右分别为 $\frac{6}{1} + \sqrt{\frac{2\ln 4}{1}} = 7.67$, $\frac{2}{1} + \sqrt{\frac{2\ln 4}{1}} = 3.67$, $\frac{7}{1} + \sqrt{\frac{2\ln 4}{1}} = 8.67$, 因此第二步选择奖励为 7 的节点。如下图所示。



课上习题

(2) 由于此时已经到达叶子节点，因此不需要进行扩展和模拟过程，反向传播后结果如下图所示



课上习题

- (3) 算法在很长一段时间内都会选择奖励为 7 的节点，而不会探索奖励为 9 的节点。当实验次数足够多时，第二层左侧的节点的 UCB 值最终会超过第二层中间节点的 UCB 值，因此只要实验次数足够多，算法是有可能探索到奖励为 9 的节点的。如果希望提高算法的效率，可考虑加大探索的力度，即取一个更大的超参数 C 。不难验证，在原题中的状态下，取 $C = 10$ 即可令算法选择第二层左侧的节点。

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

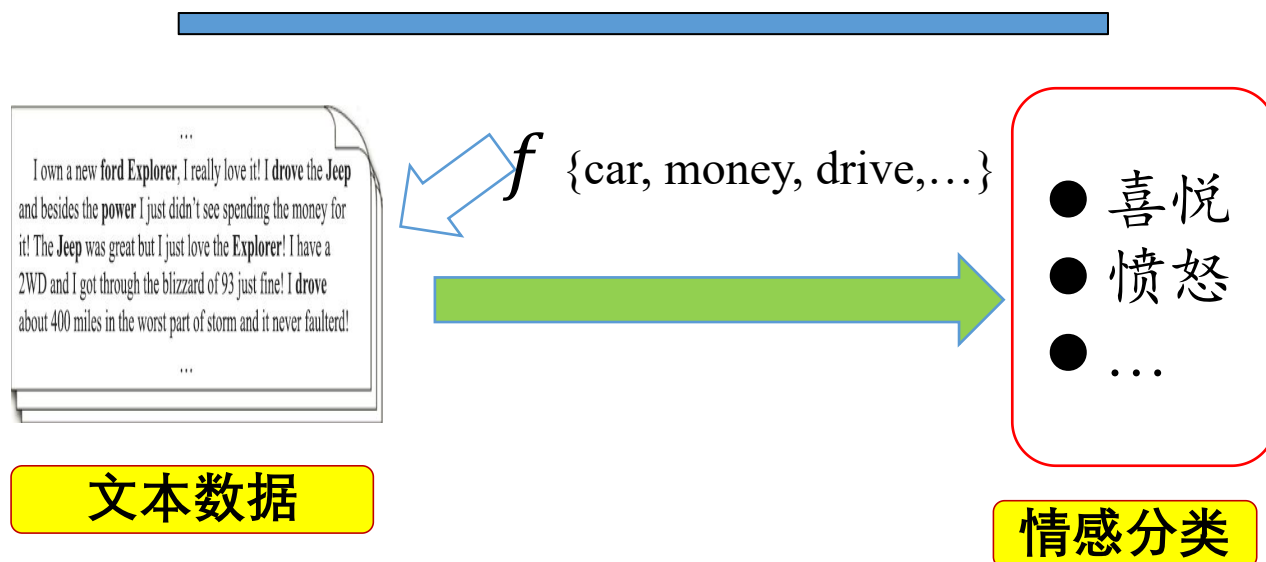
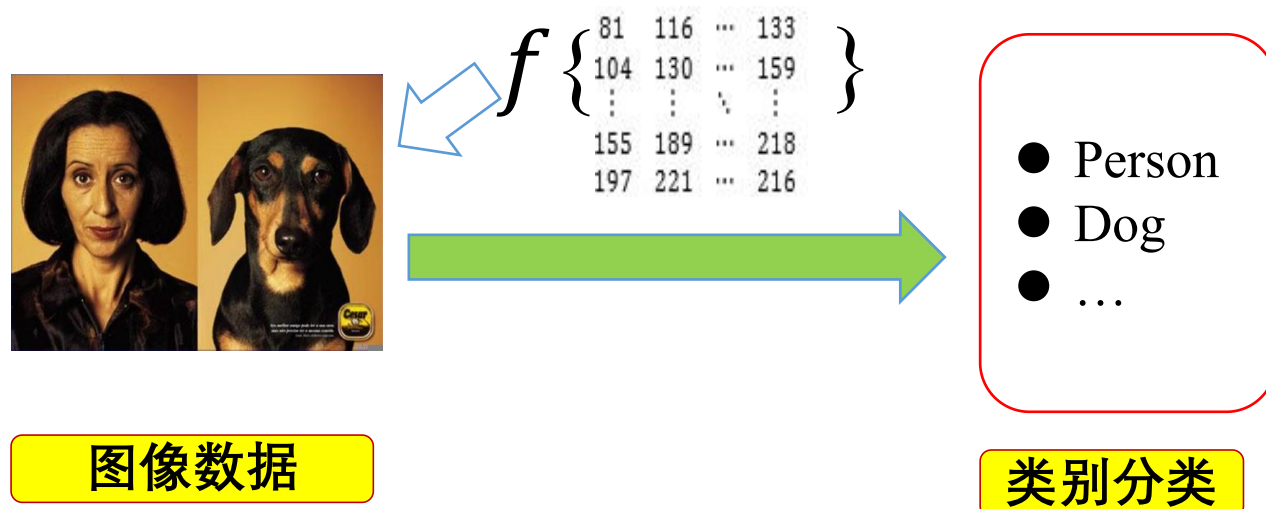
七、生成学习模型

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)

机器学习的分类

Step 1

Collect demonstration data and train a supervised policy.

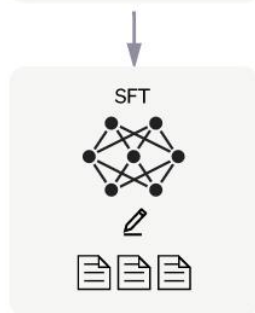
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

Collect comparison data and train a reward model.

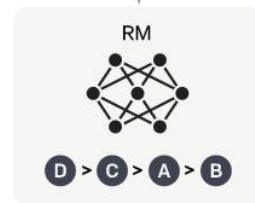
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



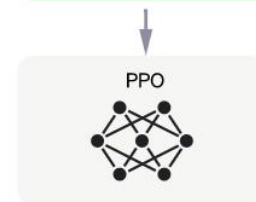
Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.



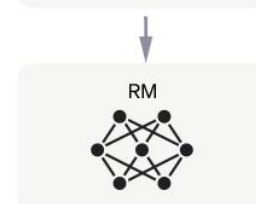
The PPO model is initialized from the supervised policy.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



监督学习：重要元素

标注数据

■ 标识了类别信息的数据
学什么

学习模型

■ 如何学习得到映射模型
如何学

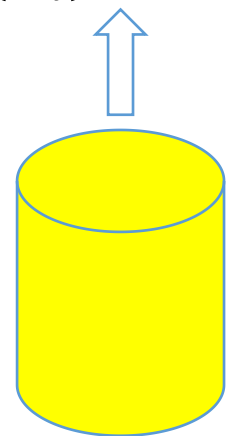
损失函数

■ 如何对学习结果进行度量
学到否

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



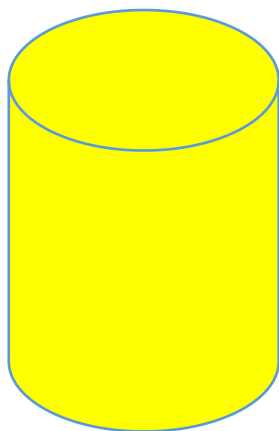
训练数据集
 $(x_i, y_i), i = 1, \dots, n$

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	$Loss(y_i, P(y_i x_i)) = -\log P(y_i x_i)$

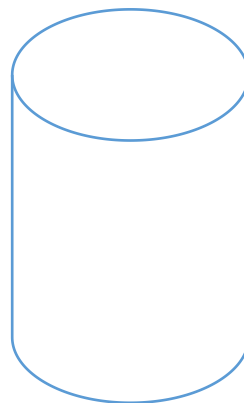
监督学习：训练数据和测试数据

从**训练数据集**学习
得到映射函数 f



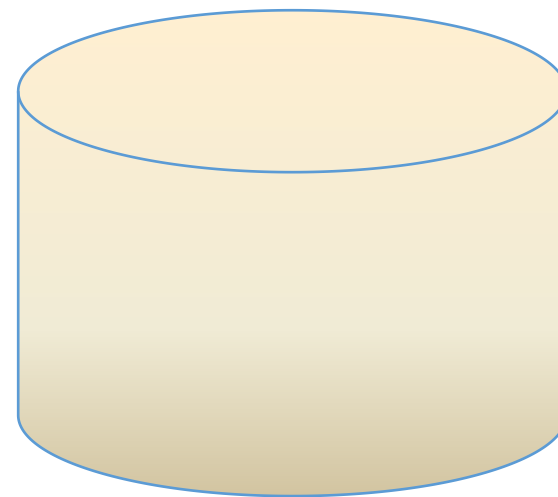
训练数据集
 $(x_i, y_i), i = 1, \dots, n$

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。

在测试数据集测试映射函数 f

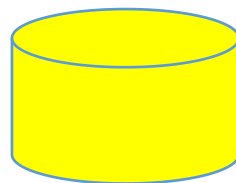
期望风险(expected risk)

当测试集中存在无穷多数据时产生的损失。期望风险越小，学习所得模型越好。

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。
- 根据大数定律，当样本容量趋于无穷时，经验风险趋于期望风险。所以在实践中很自然用经验风险来估计期望风险。
- 由于现实中训练样本数目有限，用经验风险估计期望风险并不理想，要对经验风险进行一定的约束。

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

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
经验风险小	期望风险大	过学习 (模型过于复杂)
经验风险大	期望风险大	欠学习
经验风险大	期望风险小	“神仙算法”或“黄粱美梦”

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)：

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

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：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。

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

$$P(\text{人脸} | \text{人脸}) = 0.99$$

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）

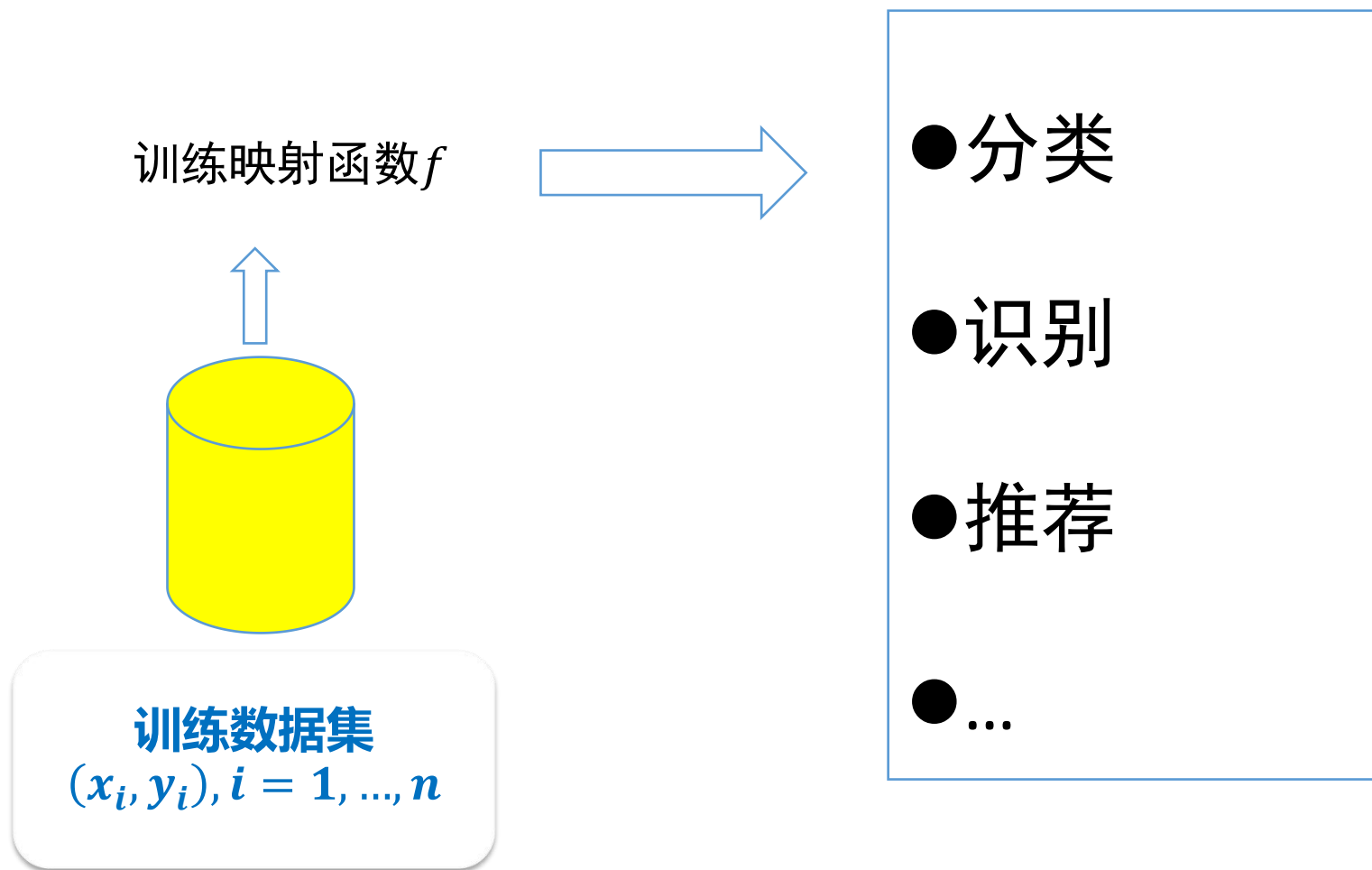
$$P(Y|X) = \frac{P(X, Y)}{P(X)} \quad \text{或者} \quad P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)}$$

- 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：应用



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

- 在现实生活中，往往需要分析若干变量之间的关系，如碳排放量与气候变暖之间的关系、某一商品广告投入量与该商品销售量之间的关系等，这种分析不同变量之间存在关系的研究叫**回归分析**，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**。
- 一旦确定了回归模型，就可以进行预测等分析工作，如从碳排放量预测气候变化程度、从广告投入量预测商品销售量等。

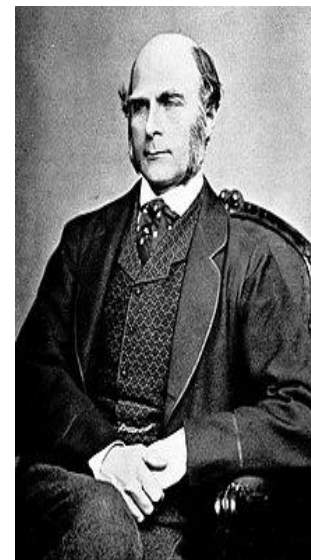
线性回归：一元线性回归

$$y = 33.73(\text{英寸}) + 0.516x$$

y : 子女平均身高

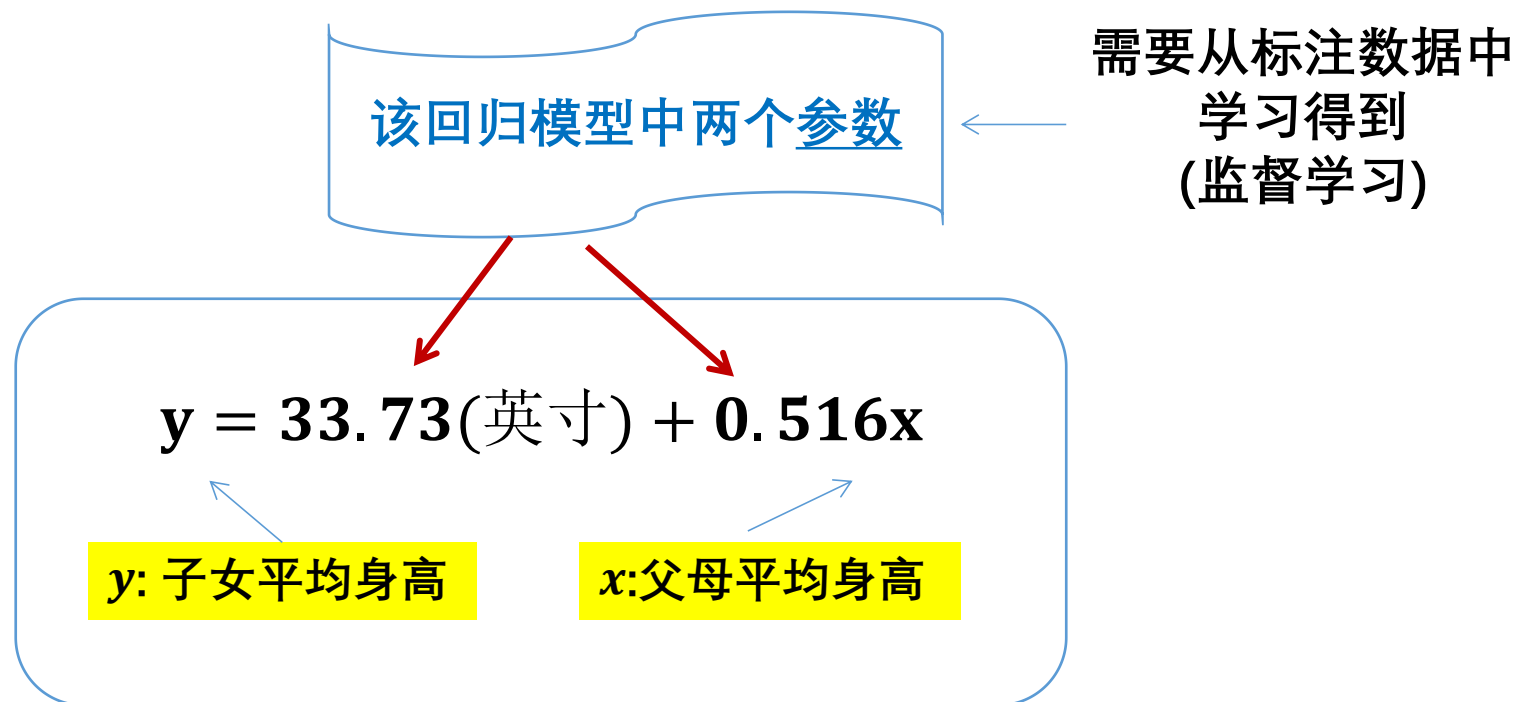
x : 父母平均身高

- 父母平均身高每增加一个单位, 其成年子女平均身高只增加0.516个单位, 它反映了这种“衰退(regression)”效应 (“回归”到正常人平均身高)。
- 虽然 x 和 y 之间并不总是具有“衰退”(回归)关系, 但是“线性回归”这一名称就保留下来了。



英国著名生物学家兼
统计学家高尔顿
Sir Francis Galton
(1822-1911)

线性回归：一元线性回归



- 给出任意一对父母平均身高，则可根据上述方程，计算得到其子女平均身高
- 从父母平均身高来预测其子女平均身高
- 如何求取上述线性方程（预测方程）的参数？

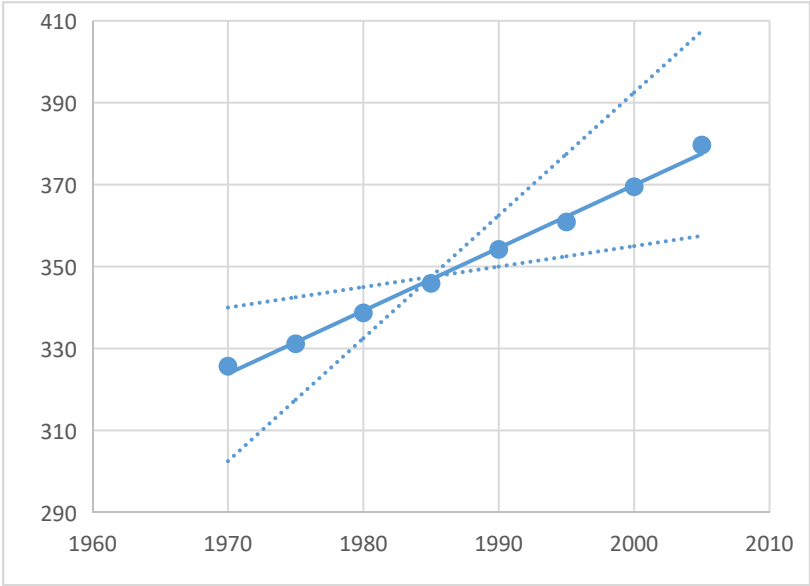
线性回归：一元线性回归

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



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）

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

↓ 代入

回归模型： $y = ax + b$

求取：最佳回归模型是最小化残差平方和的均值，即要求8组 (x, y) 数据得到的残差平均值 $\frac{1}{N} \sum (y - \tilde{y})^2$ 最小。残差平均值最小只与参数 a 和 b 有关，最优解即是使得残差最小所对应的 a 和 b 的值。

线性回归：一元线性回归

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

- 记在当前参数下第 i 个训练样本 x_i 的预测值为 \hat{y}_i
- x_i 的标注值（实际值） y_i 与预测值 \hat{y}_i 之差记为 $(y_i - \hat{y}_i)^2$
- 训练集中 n 个样本所产生误差总和为： $L(a, b) =$

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

目标：寻找一组 a 和 b ，使得误差总和 $L(a, b)$ 值最小。在线性回归中，解决如此目标的方法叫最小二乘法。

一般而言，要使函数具有最小值，可对 $L(a, b)$ 参数 a 和 b 分别求导，令其导数值为零，再求取参数 a 和 b 的取值。

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： $y_i = ax_i + b$ ($1 \leq i \leq n$)

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

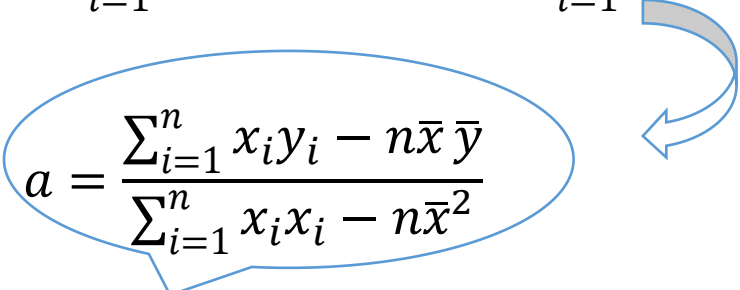
将 $b = \bar{y} - a\bar{x}$ ($\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$, $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$)
代入上式

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

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

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

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


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

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

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

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

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 = $1.5344 \times$ 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：

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

为了方便，使用矩阵来表示所有的训练数据和数据标签。

$$X = [\mathbf{x}_1, \dots, \mathbf{x}_m], \quad \mathbf{y} = [y_1, \dots, y_m]$$

其中每一个数据 \mathbf{x}_i 会扩展一个维度，其值为1，对应参数 a_0 。均方误差函数可以表示为：

$$J_m(\mathbf{a}) = (\mathbf{y} - X^T \mathbf{a})^T (\mathbf{y} - X^T \mathbf{a})$$

均方误差函数 $J_n(\mathbf{a})$ 对所有参数 \mathbf{a} 求导可得：

$$\nabla J(\mathbf{a}) = (-2X(\mathbf{y} - X^T \mathbf{a}))^T$$

因为均方误差函数 $J_n(\mathbf{a})$ 是一个二次的凸函数，所以函数只存在一个极小值点，也同样是极小值点，所以令 $\nabla J(\mathbf{a}) = 0$ 可得

$$\begin{aligned} XX^T \mathbf{a} &= X\mathbf{y} \\ \mathbf{a} &= (XX^T)^{-1} X\mathbf{y} \end{aligned}$$

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

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

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

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

MORE VIDEOS

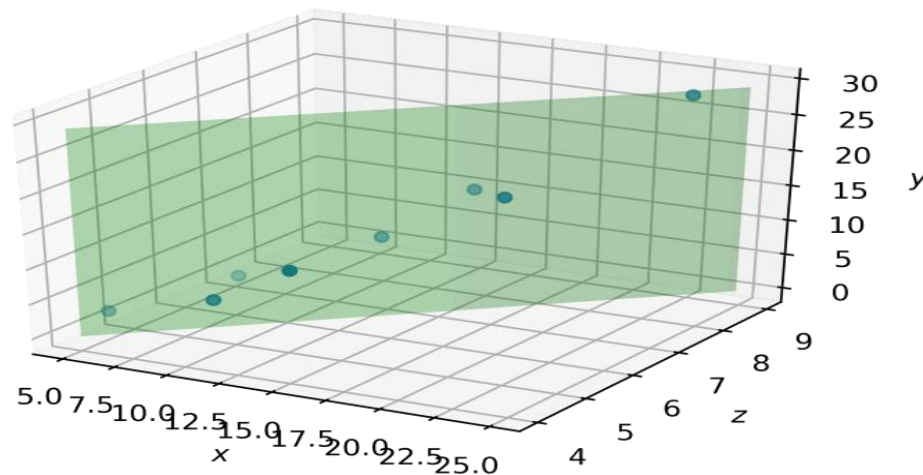
线性回归：多元线性回归

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

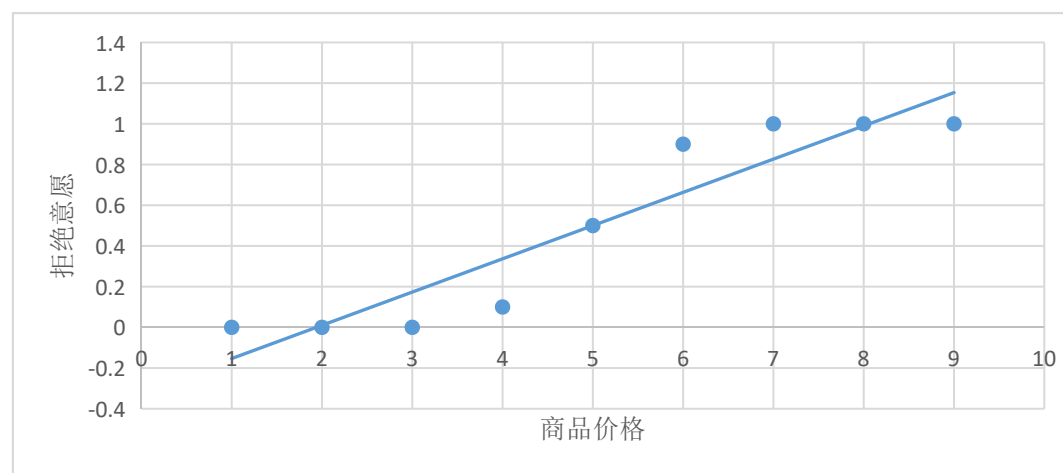
$$\mathbf{a} = [1.312 \quad 0.626 \quad -9.103]$$
$$\mathbf{y} = -9.103 + 1.312x + 0.626z$$



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]。

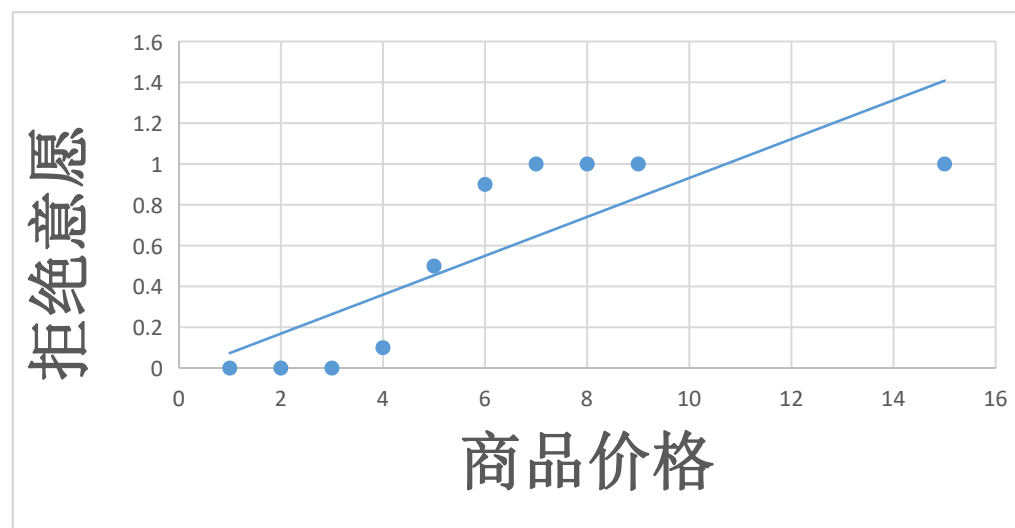


用户对某件商品拒绝购买意愿（选择不购买商品的人数/受调查的总人数）与商品价格之间的关系

线性回归：逻辑回归/对数几率回归

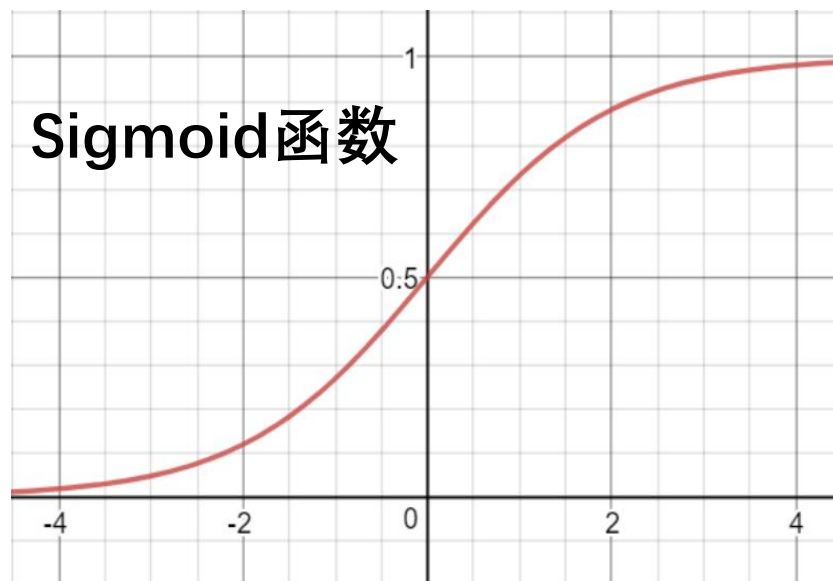
逻辑回归/对数几率回归模型例子

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]。



加入一个离群点，该点表示当商品价格为15时，用户拒绝意愿为1
(即用户不愿意购买该商品)

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

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）。

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

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）。

$y = \frac{1}{1+e^{-(\mathbf{w}^T \mathbf{x} + b)}}$ 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 $\log\left(\frac{p}{1-p}\right)$ 。

显然，可以得到 $p(y = 1|\mathbf{x}) = h_{\theta}(\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{w}^T \mathbf{x} + b)}}$ 和 $p(y = 0|\mathbf{x}) = 1 - h_{\theta}(\mathbf{x}) = \frac{e^{-(\mathbf{w}^T \mathbf{x} + b)}}{1+e^{-(\mathbf{w}^T \mathbf{x} + b)}}$ 。 θ 表示模型参数（ $\theta = \{\mathbf{w}, b\}$ ）。于是有：

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

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（ $\theta = \{\mathbf{w}, b\}$ ）。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：

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

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

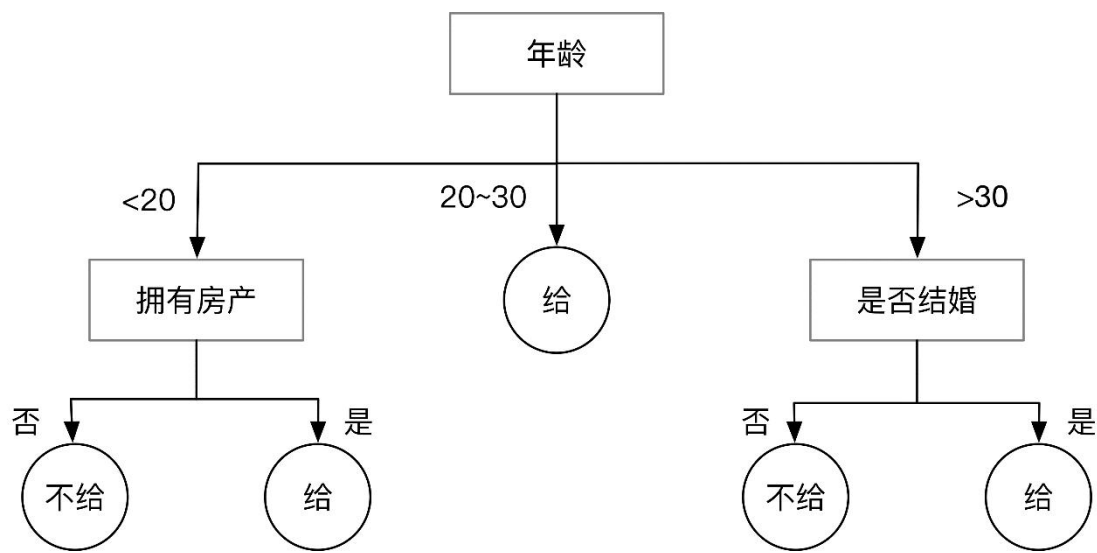
五、Ada Boosting

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： 案例

决策树分类案例

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

信息熵 (entropy)

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。

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

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-)

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

$$\text{"年龄"} 20 \sim 30: Ent(D_1) = -\left(\frac{4}{4} \times \log_2 \frac{4}{4} + 0\right) = 0$$

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

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：

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

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：

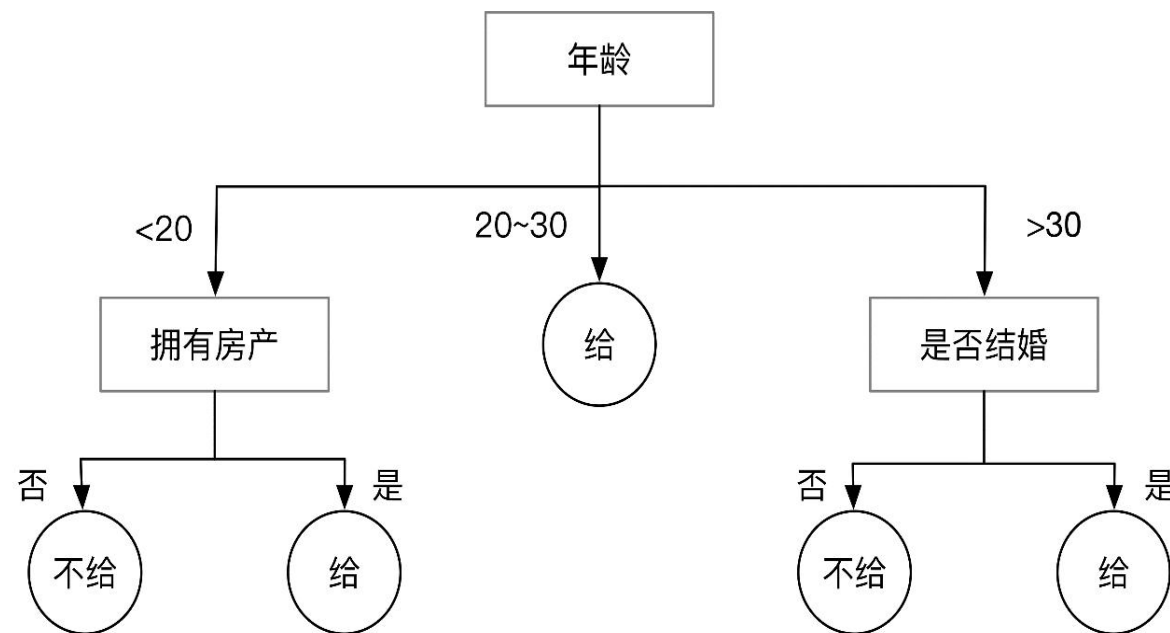
$$info = - \sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$$

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

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：

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

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	-

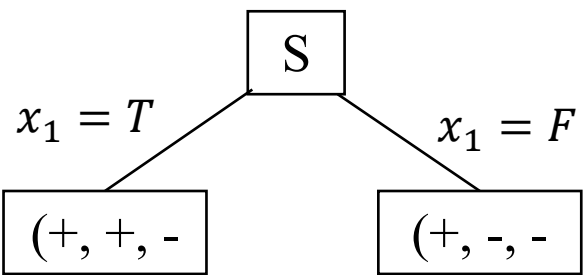
决策树：例题

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!