

BUSS 3620.人工智能导论

# 不确定性

刘佳璐

安泰经济与管理学院

上海交通大学

# 为什么需要引入不确定性?

- 搜索: 行动
  - 状态是**确定的**
- 状态不确定
  - 根据一些现有证据,  
形成对状态的一种**认知**

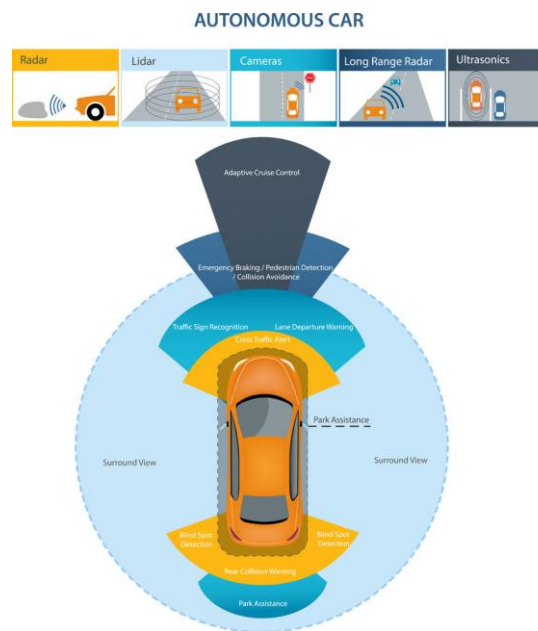


# 为什么需要引入不确定性?

- 解决的问题类型
  - 基于观测值推理真实值



侦探/鉴宝



感知系统  
(雷达摄像机)



推荐系统等

BUSS 3620.人工智能导论

# #1. 概率复习

刘佳璐

安泰经济与管理学院

上海交通大学

# 基础概念

---

- 随机变量 Random Variable: 表示随机现象各种结果的变量
  - 骰子 *Die*:  $\{1, 2, 3, 4, 5, 6\}$
- 概率 Probability:  $P(\omega)$ : 事件  $\omega$  发生的概率
  - $0 \leq P(\omega) \leq 1$
  - $\sum_{\omega \in \Omega} P(\omega) = 1$
- 概率分布 Probability distribution: 表述随机变量取值的概率规律
  - $\mathbf{P}(\text{Die}) = \langle 1/6, 1/6, 1/6, 1/6, 1/6, 1/6 \rangle$

# 基础概念

---

- 概率 Probability
  - 非条件概率 Unconditional probability
    - 在没有任何限制的条件或证据的情况下,某一事件发生的概率
  - 条件概率 Conditional probability
    - 基于一些已知证据,某一事件发生的概率
    - $P(a / b)$ 
      - $P(\text{rain today} / \text{knif})$
      - $P(\text{route change} / \text{traffic conditions})$
      - $P(\text{disease} / \text{test results})$

# 条件概率 Conditional Probability

---

$$P(a \mid b) = \frac{P(a \wedge b)}{P(b)}$$

小练习:

$$P(\text{sum to 12} \mid \text{first die is 6}) = ?$$

- 独立事件
  - 事件A的发生 **不影响** 事件B的概率值
  - $P(\text{sum to 12} \mid \text{first die is 6}) = P(\text{second die is 6} \mid \text{first die is 6}) = P(\text{second die is 6})$

# 贝叶斯法则 Bayes' Rule

---

$$P(a | b) = \frac{P(a \wedge b)}{P(b)}$$

$$P(a \wedge b) = P(b)P(a | b)$$

$$P(a \wedge b) = P(a)P(b | a)$$

$$P(a)P(b | a) = P(b)P(a | b)$$

$$P(b | a) = \frac{P(b)P(a | b)}{P(a)} = \frac{P(a | b)P(b)}{P(a)}$$



# 练习 #1

---

- 某城市发生了一起银行抢劫案。监控视频显示，抢劫者戴着面具。警方有如下数据：
  - 惯犯在抢劫时戴面具的概率为80%
  - 城市中30%的抢劫案由惯犯实施
  - 在所有抢劫案中，25%涉及戴面具的劫匪
- 问：已知这起案件中的抢劫者戴着面具，计算抢劫者是惯犯的概率

# 贝叶斯法则的应用

- 计算反向条件概率

- 尤其是其中一个条件概率很容易获得的时候

已知:  $P(\text{面具} | \text{惯犯})$

可计算:  $P(\text{惯犯} | \text{面具})$

已知:  $P(\text{观测值} | \text{真实值})$

可计算:  $P(\text{真实值} | \text{观测值})$

已知:  $P(\text{医疗检测阳性} | \text{疾病})$

可计算:  $P(\text{疾病} | \text{医疗检测阳性})$

已知:  $P(\text{车速} = 5 | \text{堵车})$

可计算:  $P(\text{堵车} | \text{车速} = 5)$

# 联合概率 Joint Probability

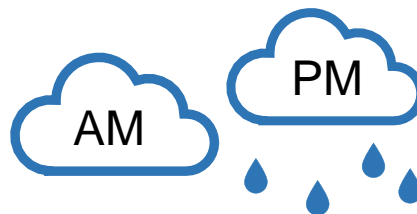
- 多个事件同时发生的概率。



$C = \text{cloud}$	$C = \neg \text{cloud}$
0.4	0.6



$R = \text{rain}$	$R = \neg \text{rain}$
0.1	0.9



	$R = \text{rain}$	$R = \neg \text{rain}$
$C = \text{cloud}$	0.08	0.32
$C = \neg \text{cloud}$	0.02	0.58

# 已知联合概率求条件概率

---

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

$$P(A|B) = \left\langle \frac{P(A = a_1, B)}{P(B)}, \dots, \frac{P(A = a_k, B)}{P(B)} \right\rangle$$

$$P(A|B) = \alpha \langle P(A = a_1, B), \dots, P(A = a_k, B) \rangle$$

$$\alpha(P(A = a_1, B) + \dots + P(A = a_k, B)) = 1$$

条件概率与其联合概率**成比例**

## 练习 #2

- 一个电商平台希望通过分析用户行为改进推荐系统。统计数据显示如下

	R = 推流	R = $\neg$ 推流
B = 购买	0.12	0.03
B = $\neg$ 购买	0.28	0.57

- 求:  $P(B \mid \text{推流})$

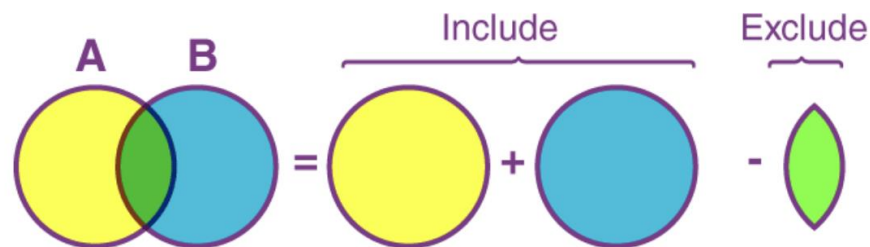
# 其他的一些概率公式

- 否定 Negation

$$P(\neg a) = 1 - P(a)$$

- 容斥原理 Inclusion – Exclusion

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$



- 边缘化 Marginalization

- 已知联合概率求单独的概率

$$P(a) = P(a, b) + P(a, \neg b)$$

$$P(X = x_i) = \sum_j P(X = x_i, Y = y_j)$$

## 练习 #3

- 在一场刑侦调查中，嫌疑人是否有犯罪记录与其DNA是否在现场找到有关系。

联合概率表如下

	M = DNA匹配	R = $\neg$ DNA匹配
C = 犯罪记录	0.06	0.04
C = $\neg$ 犯罪记录	0.10	0.80

- 求：  $P(M = \text{DNA匹配})$

# 边缘化 + 贝叶斯法则

---

$$P(a) = P(a, b) + P(a, \neg b)$$

$$P(a) = P(a \mid b)P(b) + P(a \mid \neg b)P(\neg b)$$

$$P(X = x_i) = \sum_j P(X = x_i, Y = y_j)$$

$$P(X = x_i) = \sum_j P(X = x_i \mid Y = y_j)P(Y = y_j)$$



# 有问题吗？

---

- 请随时举手提问。



BUSS 3620.人工智能导论

# #2. 贝叶斯网络

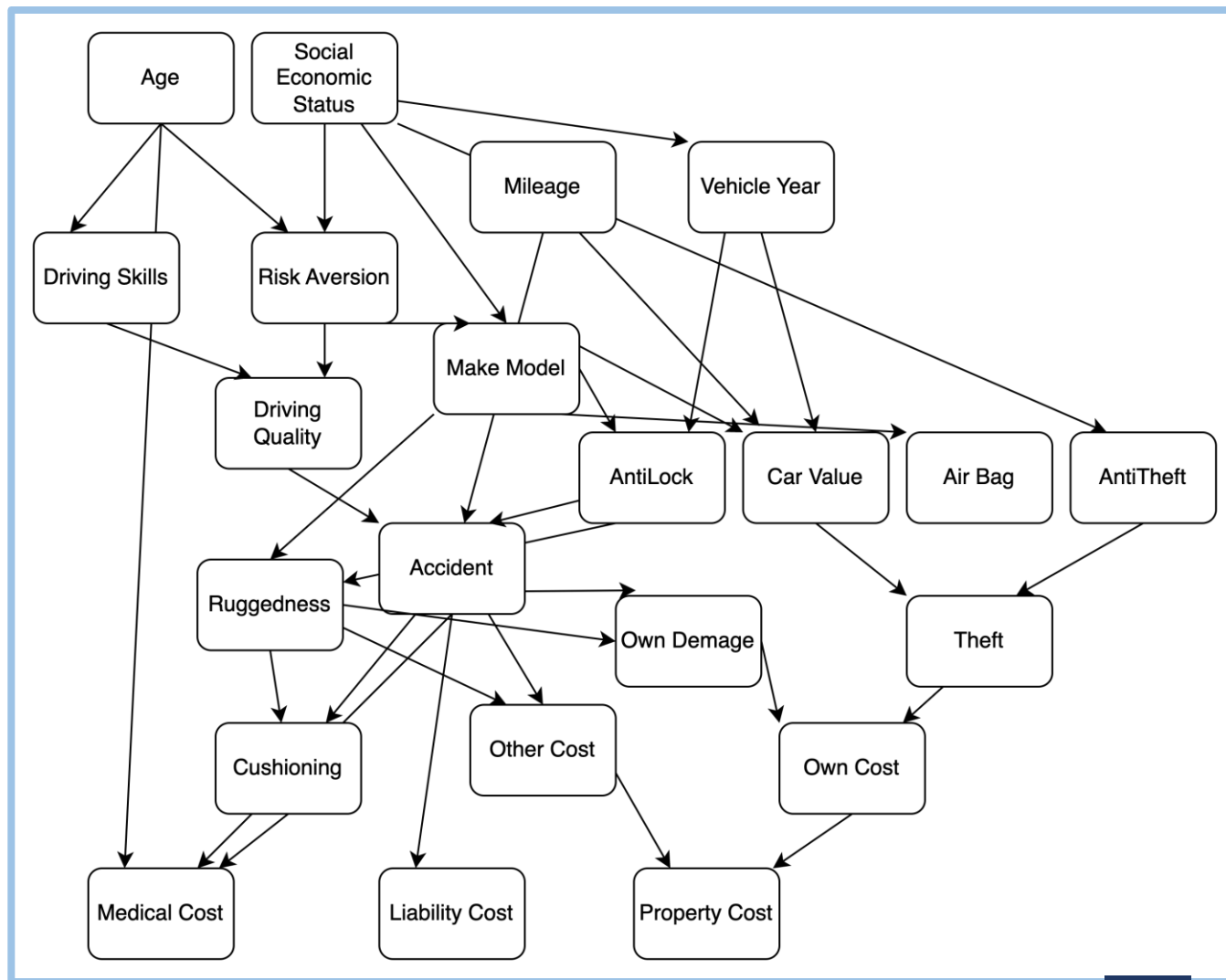
刘佳璐

安泰经济与管理学院

上海交通大学

# 贝叶斯网络(Bayesian Network)示例：保险

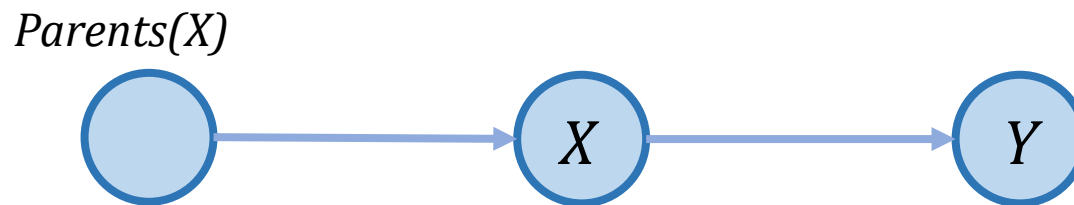
- 为什么需要BN?
  - 对复杂世界建模
  - 原因
    - 为什么会发生车祸?
  - 预测
    - 如果。。。?
  - 抽样



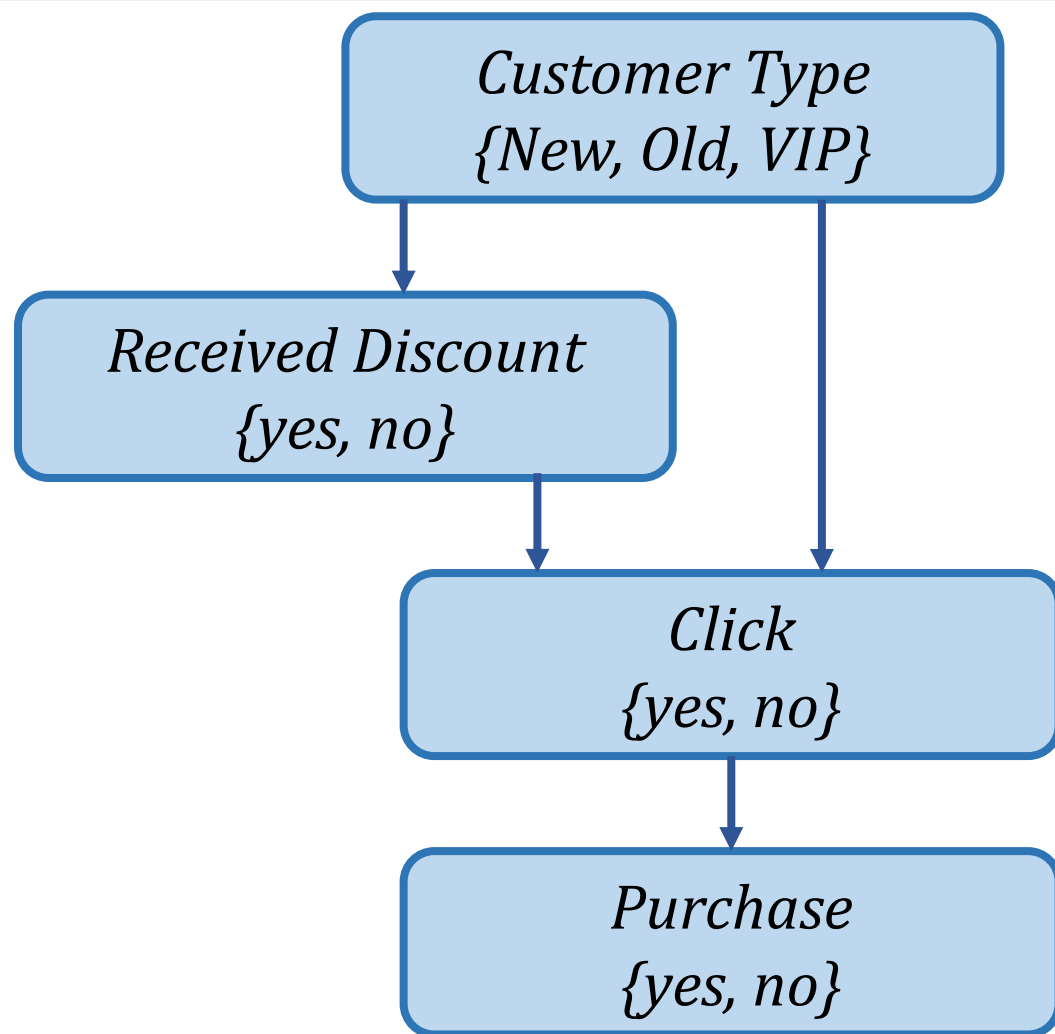
# 贝叶斯网络 Bayesian Network

- 表示随机变量之间**相互关系**的数据结构
- **有向图 Directed** graph (也称为图形模型graphical model)
- 每个节点代表一个随机变量
- 从 $X$ 到 $Y$ 的箭头代表 $X$ 是 $Y$ 的父节点
- 每个节点 $X$ 都有概率分布

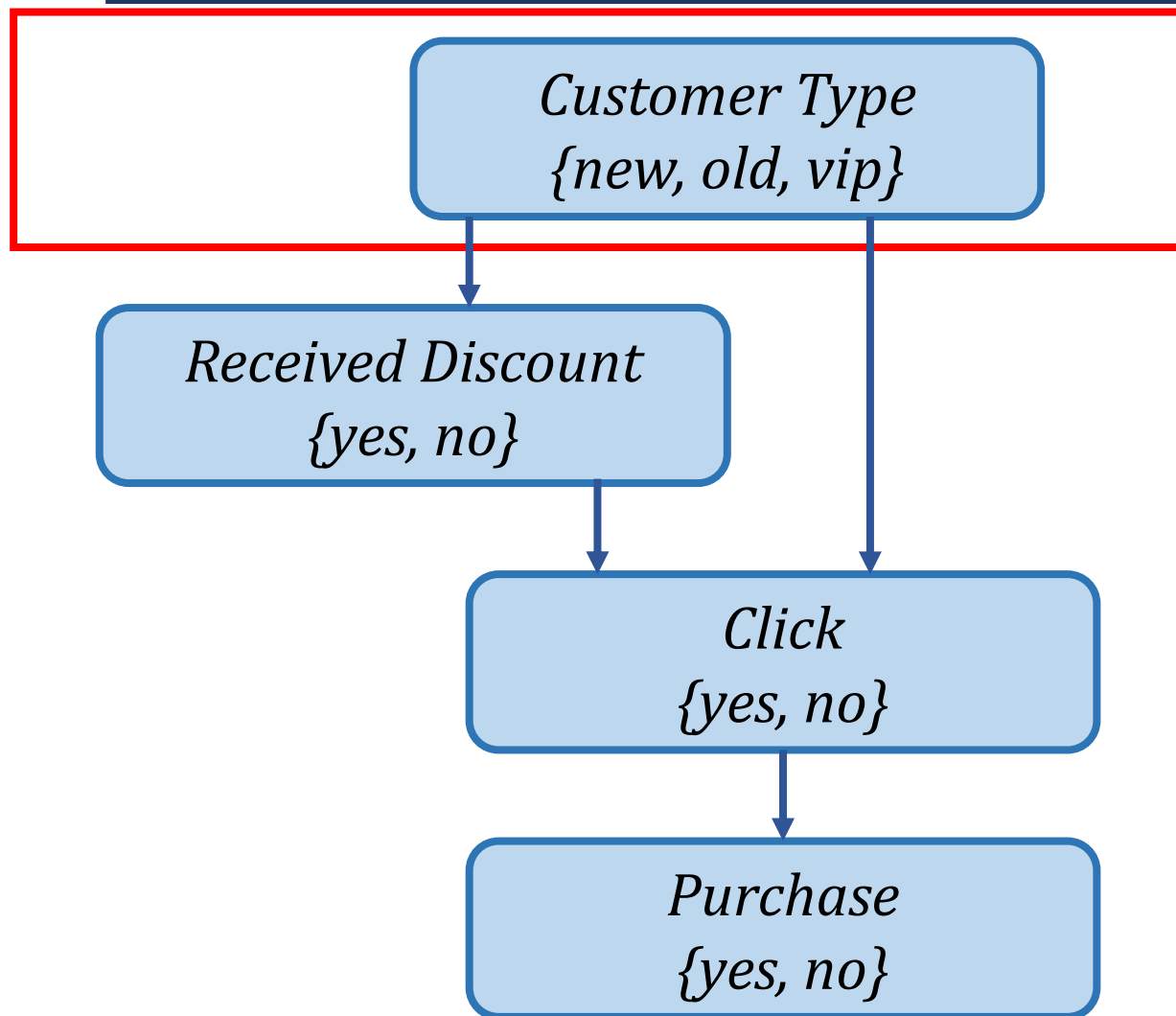
$$\mathbf{P}(X \mid \text{Parents}(X))$$



# 贝叶斯网络示例：智能营销

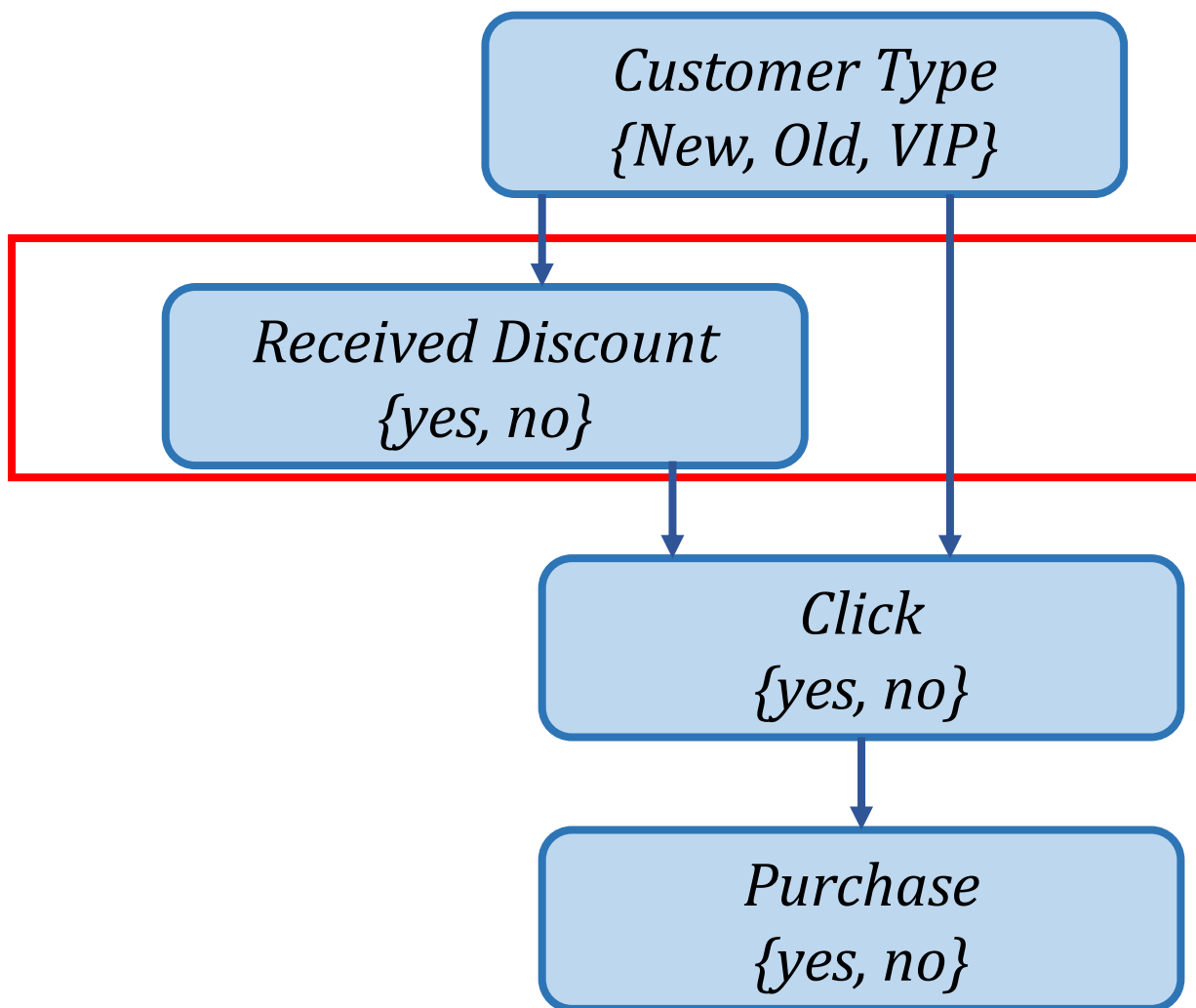


# 贝叶斯网络示例：智能营销



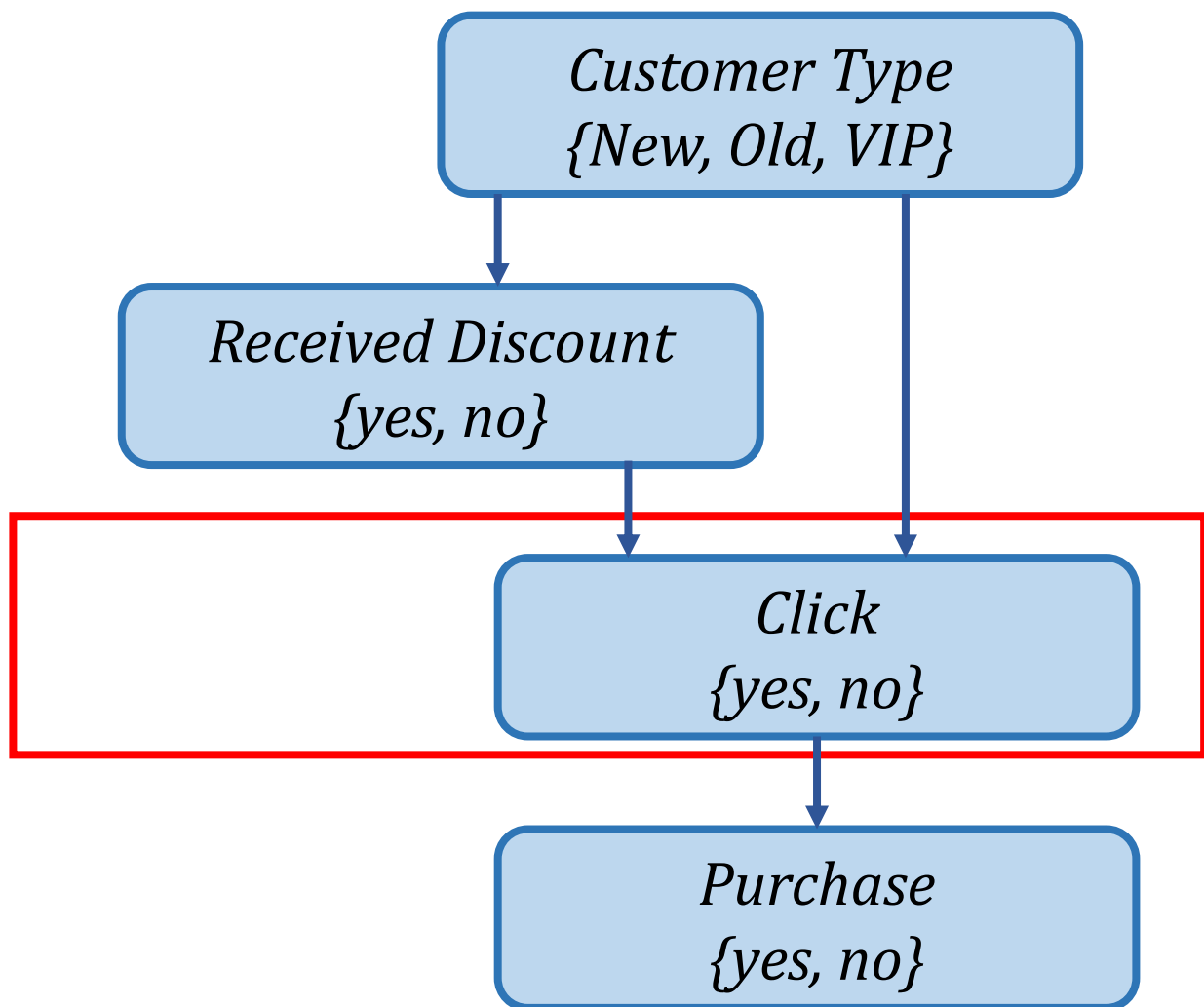
<i>new</i>	<i>old</i>	<i>vip</i>
0.3	0.6	0.1

# 贝叶斯网络示例：智能营销



<i>type</i>	<i>yes</i>	<i>no</i>
<i>new</i>	0.6	0.4
<i>old</i>	0.4	0.6
<i>vip</i>	0.8	0.2

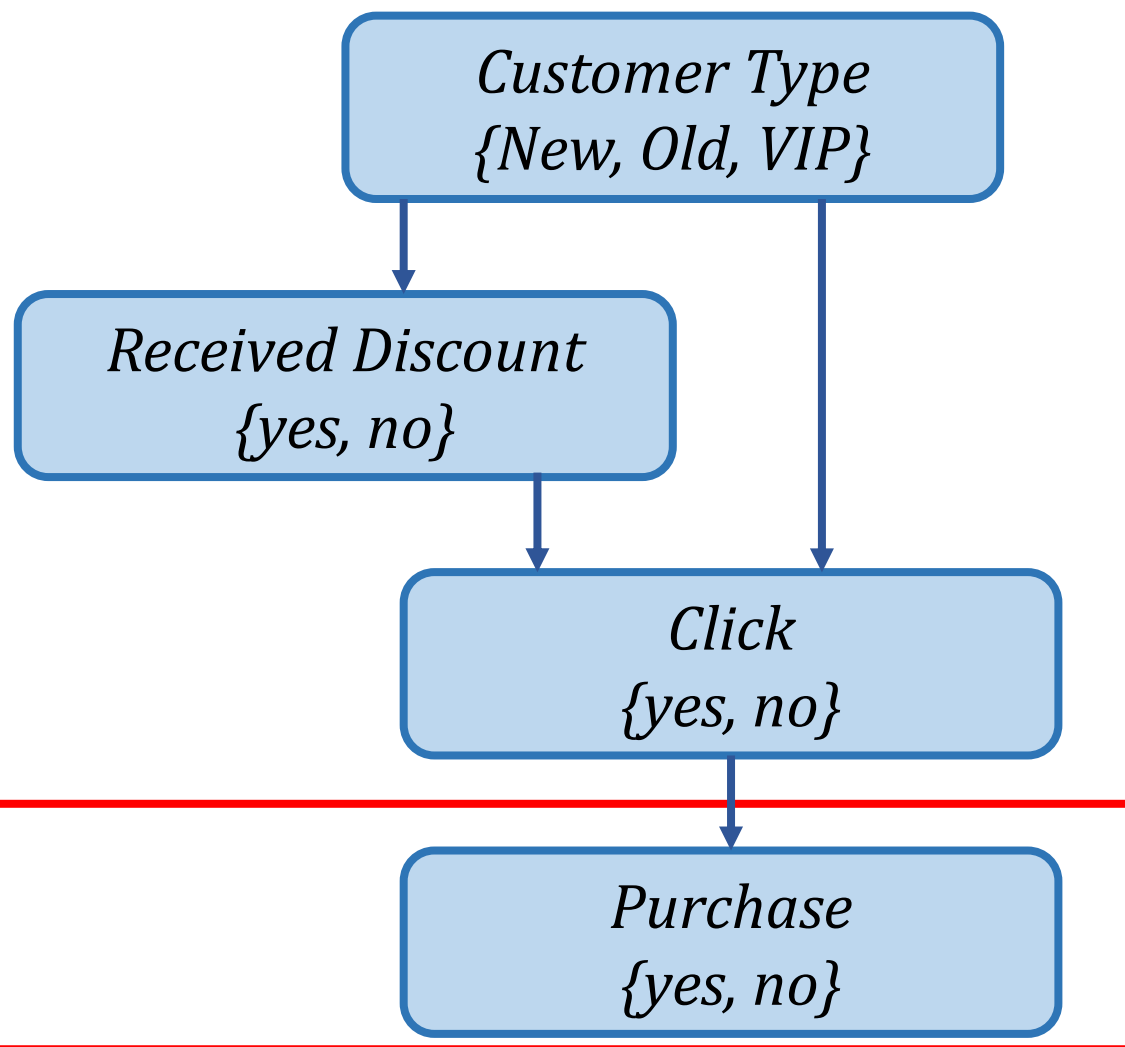
# 贝叶斯网络示例：智能营销



<i>type</i>	<i>Discount</i>	<i>yes</i>	<i>no</i>
<i>new</i>	<i>yes</i>	0.3	0.7
<i>new</i>	<i>no</i>	0.1	0.9
<i>old</i>	<i>yes</i>	0.5	0.5
<i>old</i>	<i>no</i>	0.3	0.7
<i>vip</i>	<i>yes</i>	0.7	0.3
<i>vip</i>	<i>no</i>	0.5	0.5

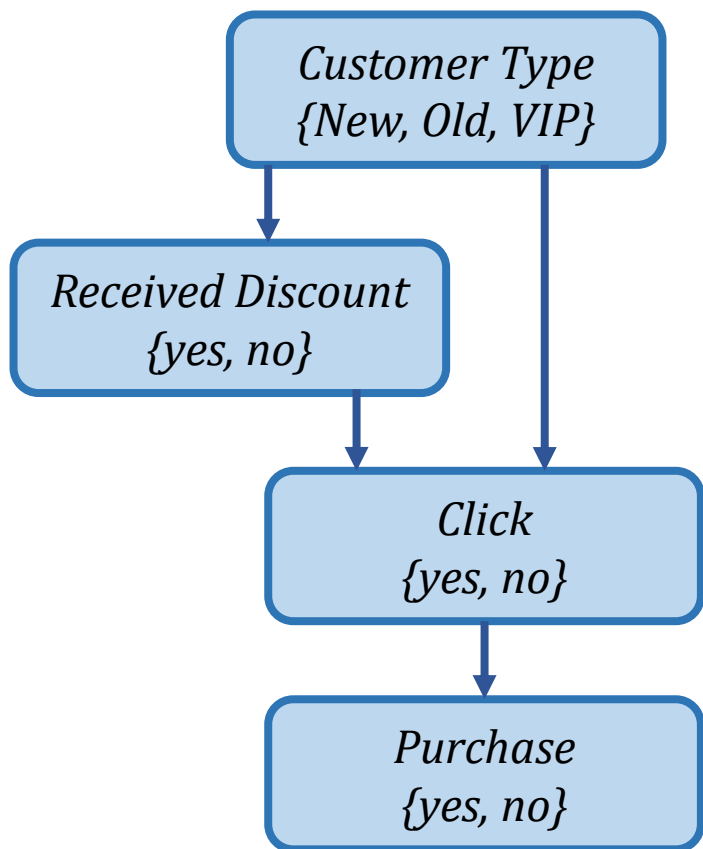


# 贝叶斯网络示例：智能营销



<i>Click</i>	<i>yes</i>	<i>no</i>
<i>yes</i>	0.6	0.4
<i>no</i>	0.2	0.8

# 练习 #4 计算联合分布



<i>new</i>	<i>old</i>	<i>vip</i>
0.3	0.6	0.1

<i>type</i>	<i>yes</i>	<i>no</i>
<i>new</i>	0.6	0.4
<i>old</i>	0.4	0.6
<i>vip</i>	0.8	0.2

<i>type</i>	<i>Discount</i>	<i>yes</i>	<i>no</i>
<i>new</i>	<i>yes</i>	0.3	0.7
<i>new</i>	<i>no</i>	0.1	0.9
<i>old</i>	<i>yes</i>	0.5	0.5
<i>old</i>	<i>no</i>	0.3	0.7
<i>vip</i>	<i>yes</i>	0.7	0.3
<i>vip</i>	<i>no</i>	0.5	0.5

<i>Click</i>	<i>yes</i>	<i>no</i>
<i>yes</i>	0.6	0.4
<i>no</i>	0.2	0.8

求:

$$P(T = \text{old}, D = \text{no})$$

$$P(T = \text{old}, D = \text{no}, C = \text{yes})$$

$$P(T = \text{old}, D = \text{no}, C = \text{yes}, P = \text{yes})$$

BUSS 3620.人工智能导论

# #3. 贝叶斯推理

刘佳璐

安泰经济与管理学院

上海交通大学

# 推理 Inference

---

- 查询 Query  $X$ : 要计算变量  $X$  的概率分布
  - 例: 最终购买决策
- 可观察到的变量 Evidence variable  $E$ : 可以观测到事件  $e$ 
  - 例: 客户是否被推送了广告
- 隐变量 Hidden variables  $Y$ : 无法观测到也无法查询的变量
  - 例: 客户类型、客户是否点击广告
- 目标: 计算  $P(X | e)$

# 推理 Inference

---

- 精确推理 Exact Inference
  - 通过枚举进行精确推理 Exact Inference by Enumeration
- 近似推理 Approximate Inference
  - 通过抽样进行近似推理 Approximate Inference through sampling

# 枚举法 Inference by Enumeration

- 基于已观测到的证据 $e$ 和一些隐变量 $Y$ ，计算变量 $X$ 的概率分布的过程。

$$P(X | e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

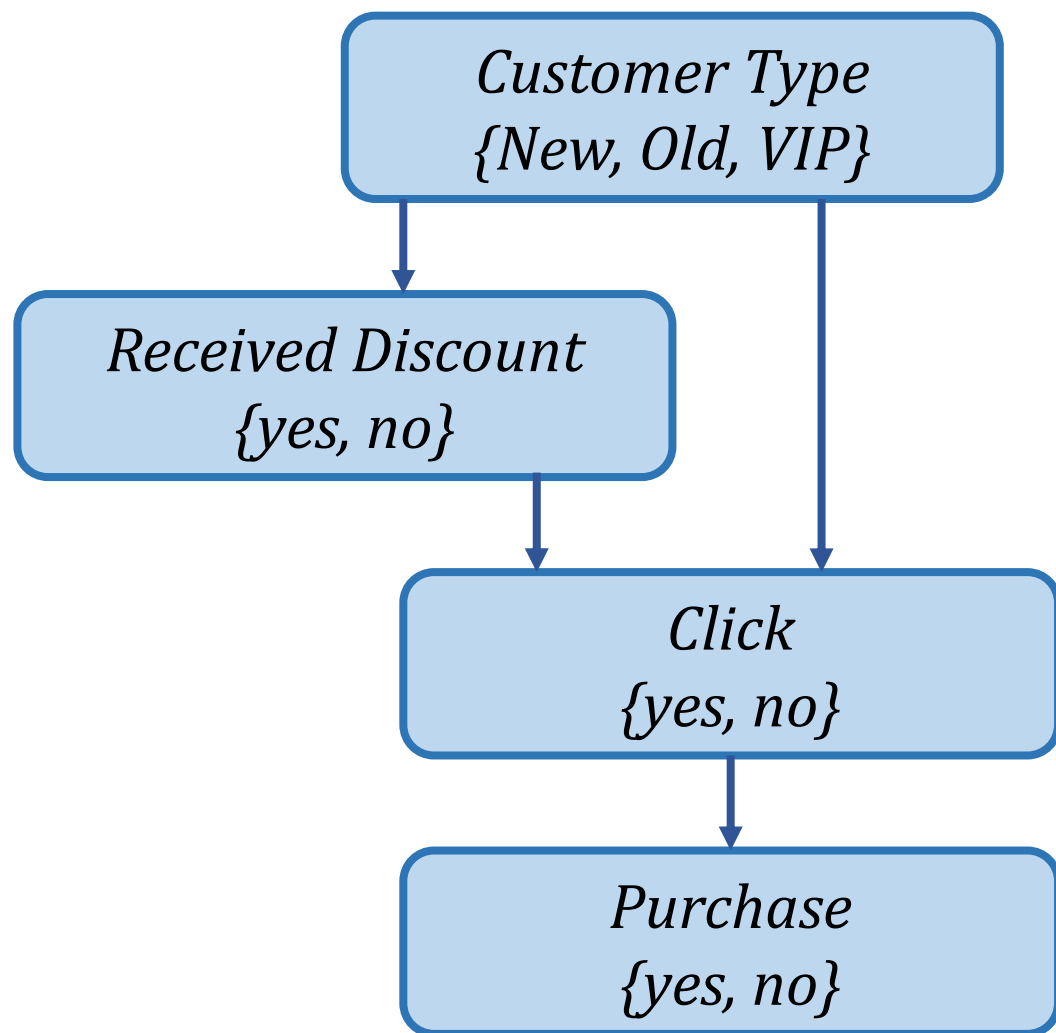
$X$ 是要计算的变量

$e$ 是已观测到的证据

$y$ 的取值范围是隐变量的可能的值

$\alpha$ 标准化概率（和为1）

# 枚举法示例 Inference by Enumeration Example



$$P(\text{Purchase} \mid T = \text{old}, D = \text{no})$$

$$= \alpha P(\text{Purchase}, T = \text{old}, D = \text{no})$$

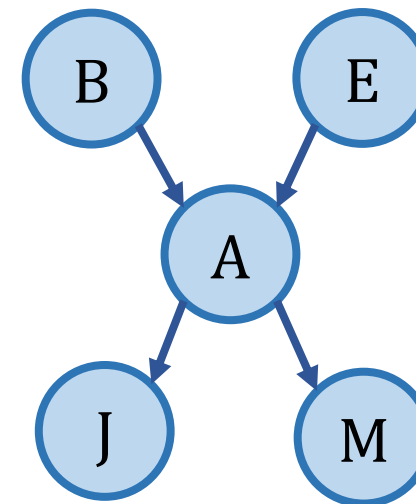
$$= \alpha [P(\text{Interview}, T = \text{old}, D = \text{no}, C = \text{yes}) + P(\text{Interview}, T = \text{old}, D = \text{no}, C = \text{no})]$$

$$\begin{aligned} = \alpha [ & P(T = \text{old})P(D = \text{no} \mid T = \text{old})P(C = \text{yes} \mid T = \text{old}, D = \text{no}) \times \\ & P(\text{Interview} \mid T = \text{old}, D = \text{no}, C = \text{yes}) + \\ & P(T = \text{old})P(D = \text{no} \mid T = \text{old})P(C = \text{no} \mid T = \text{old}, D = \text{no}) \times \\ & P(\text{Interview} \mid T = \text{old}, D = \text{no}, C = \text{no})] \end{aligned}$$

## 练习 #5

---

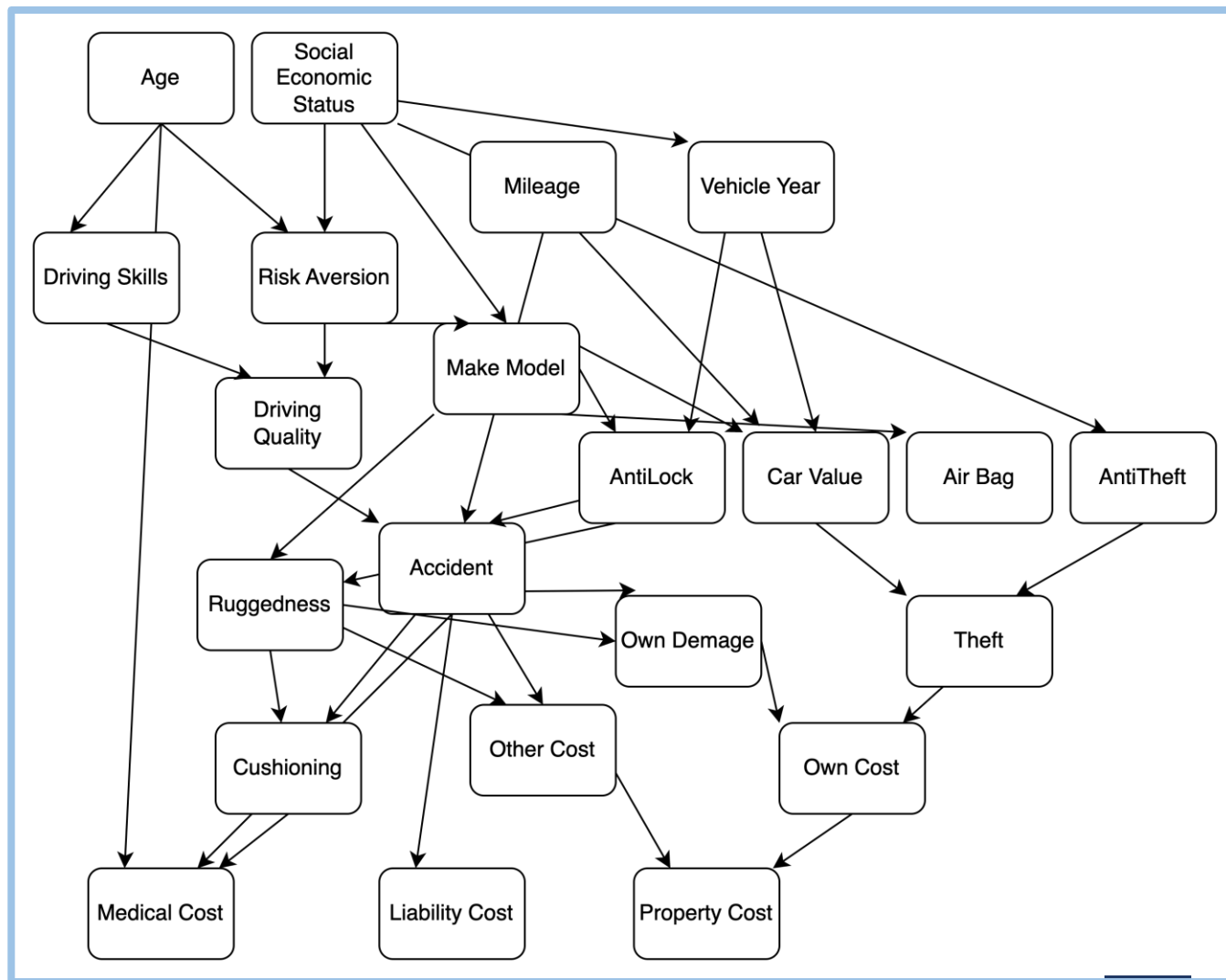
- 证据:  $+j, +m$
- 求  $P(B | +j, +m)$





# 精确推理 Exact Inference

- $P(\text{Accident}|\text{Make Model})$
- 又慢又复杂
  - 遍历隐变量组合



# 近似推理 Approximate Inference

- 近似推理以牺牲准确性来获得更短的运算时间

- 抽样 Sampling

- 每个变量根据其概率分布采样一个数值

- 如何抽样？

- 在 $[0, 1)$ 的均匀分布上产生一个随机数 $u$

- Python: `random.random()`

- $0 \leq u < 0.3 \rightarrow \text{new}; 0.3 \leq u < 0.9 \rightarrow \text{old}; 0.9 \leq u < 1 \rightarrow \text{vip}$

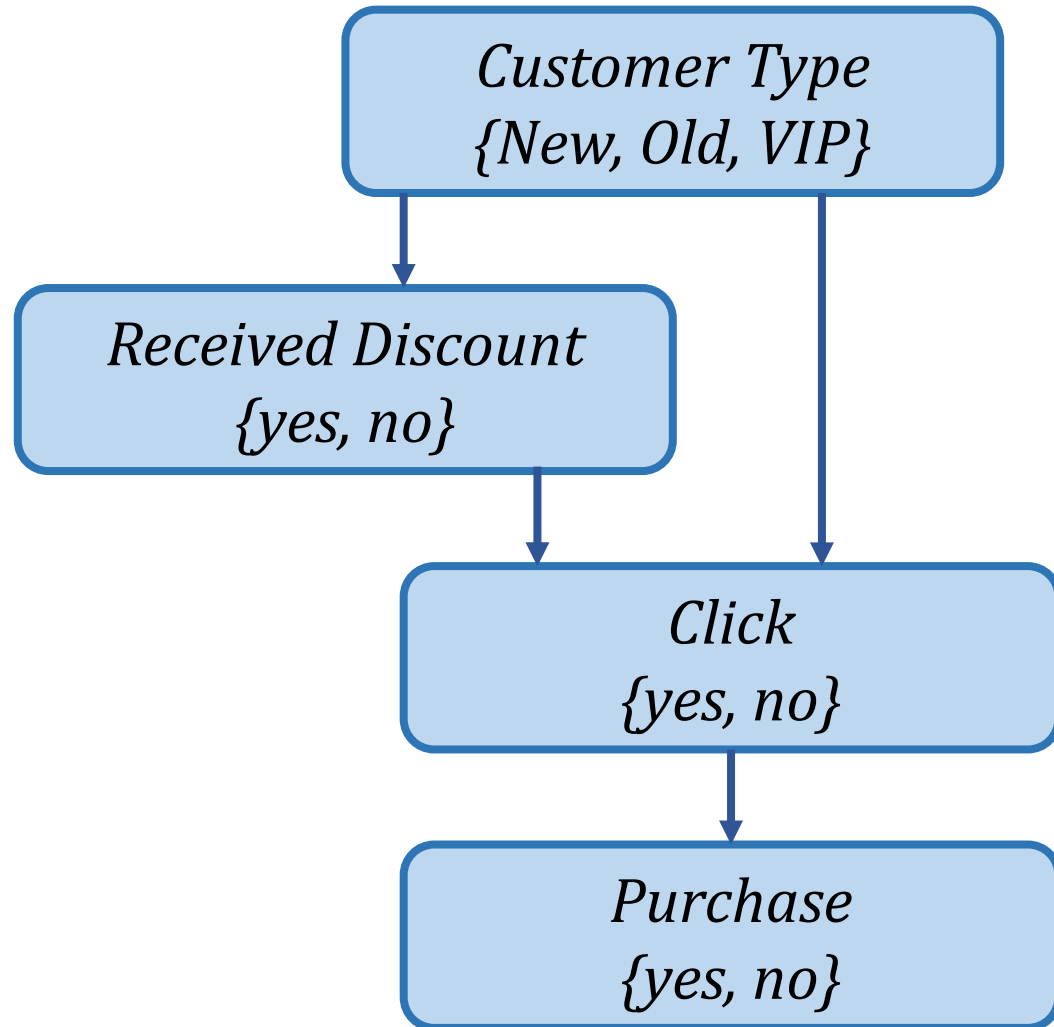
<i>new</i>	<i>old</i>	<i>vip</i>
0.3	0.6	0.1

# 近似推理的抽样方法

---

- 先验抽样 Prior Sampling
- 拒绝抽样 Rejection Sampling
- 加权抽样 Likelihood weighting Sampling

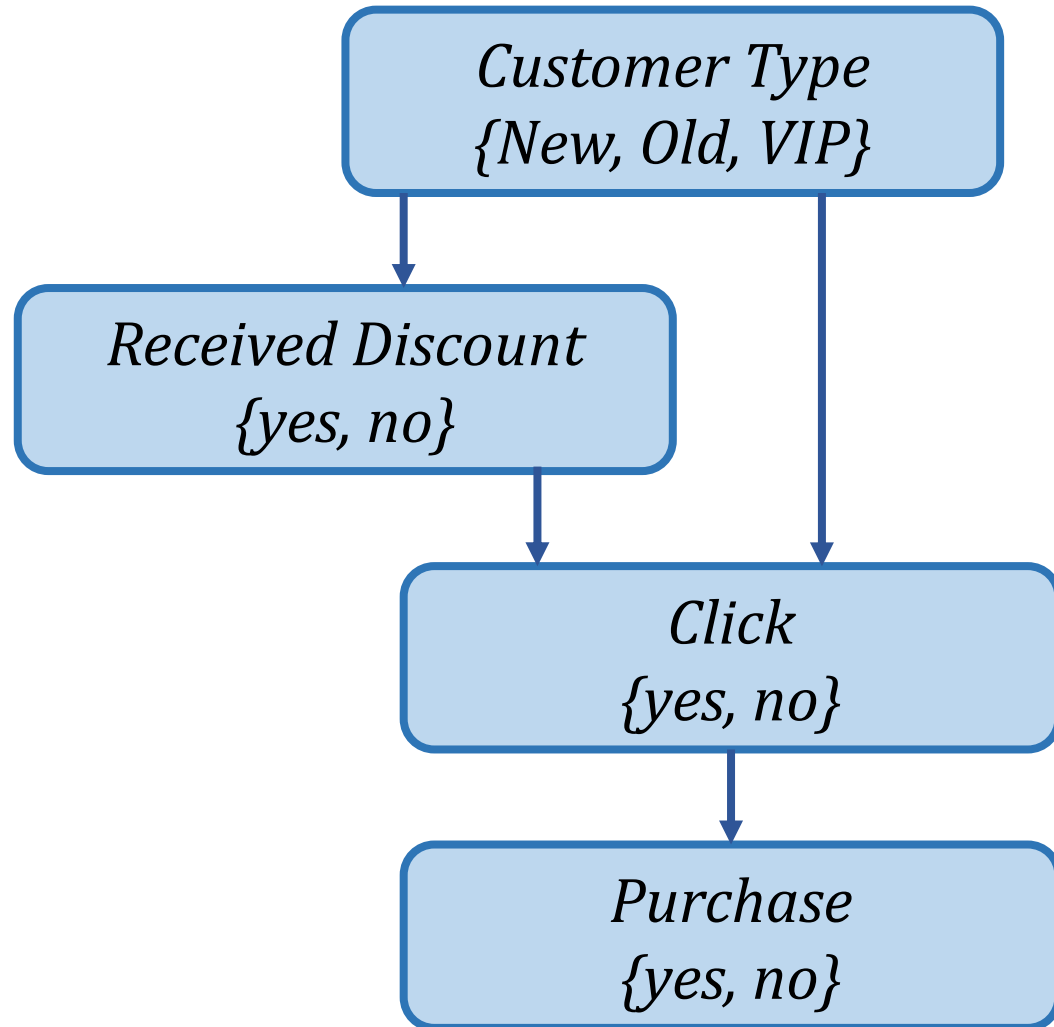
# 先验抽样 Prior Sampling



<i>new</i>	<i>old</i>	<i>vip</i>
0.3	0.6	0.1

$T = \textit{new}$

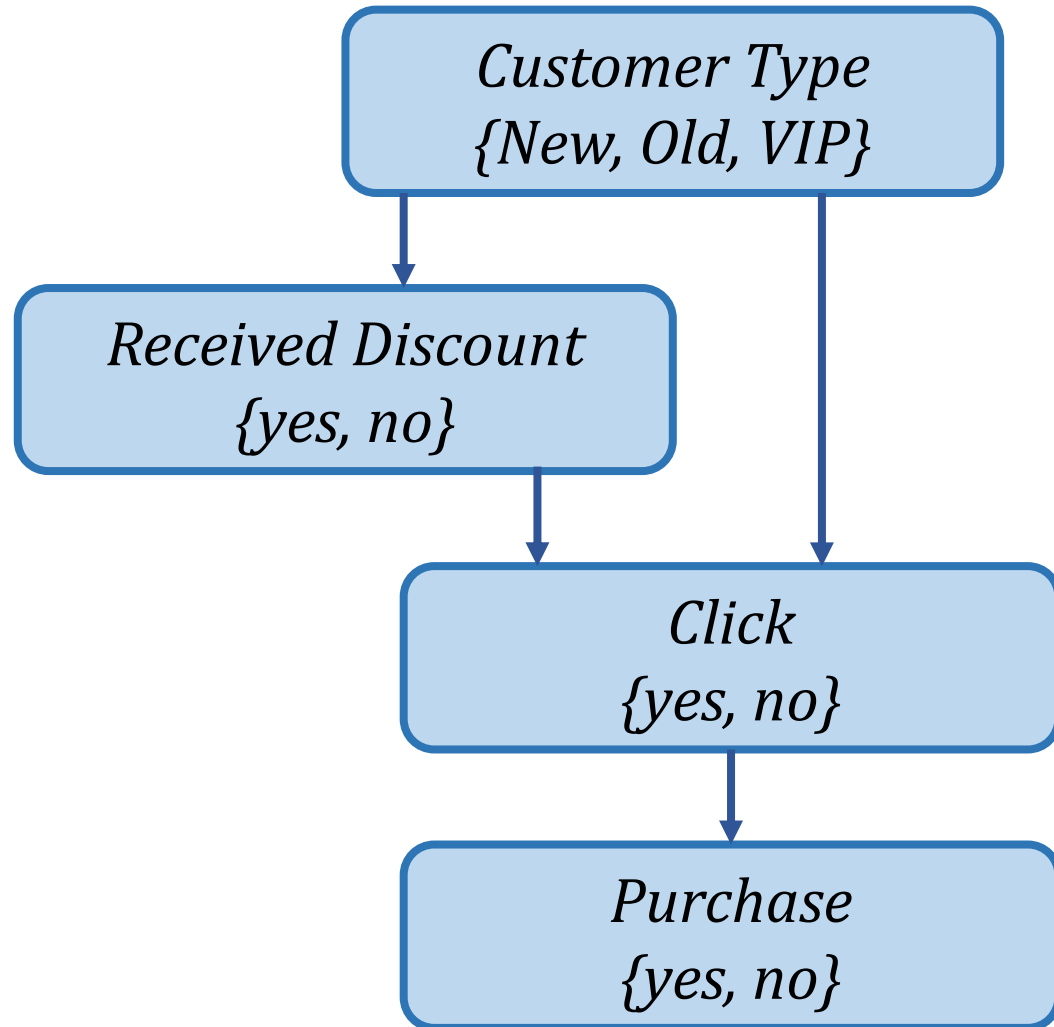
# 先验抽样 Prior Sampling



$T = \text{new}$   
 $D = \text{yes}$

<i>type</i>	<i>yes</i>	<i>no</i>
<i>new</i>	0.6	0.4
<i>old</i>	0.4	0.6
<i>vip</i>	0.8	0.2

# 先验抽样 Prior Sampling



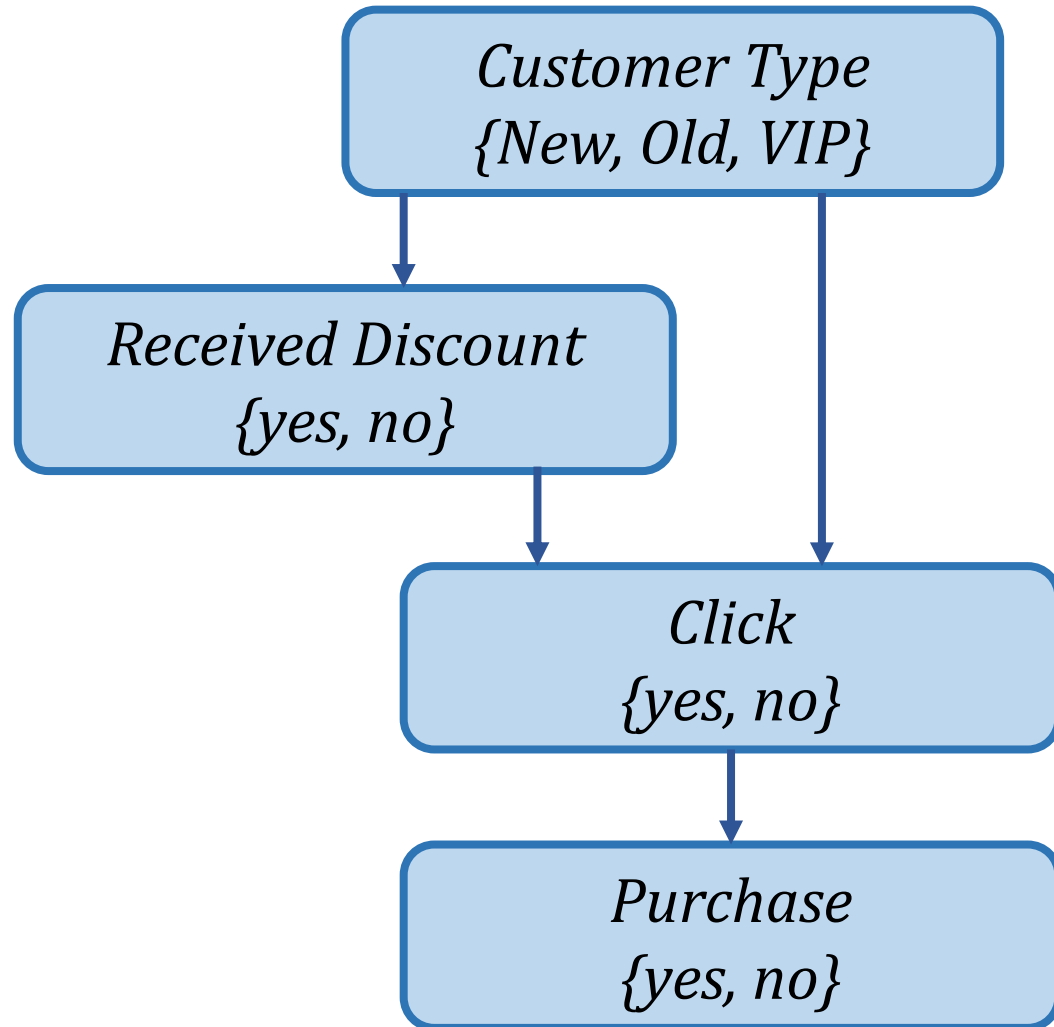
$T = new$

$D = yes$

$C = yes$

<i>type</i>	<i>Discount</i>	<i>yes</i>	<i>no</i>
<i>new</i>	<i>yes</i>	0.3	0.7
<i>new</i>	<i>no</i>	0.1	0.9
<i>old</i>	<i>yes</i>	0.5	0.5
<i>old</i>	<i>no</i>	0.3	0.7
<i>vip</i>	<i>yes</i>	0.7	0.3
<i>vip</i>	<i>no</i>	0.5	0.5

# 先验抽样 Prior Sampling



$T = new$
$D = yes$
$C = yes$
$P = yes$

<i>Click</i>	<i>yes</i>	<i>no</i>
<i>yes</i>	0.6	0.4
<i>no</i>	0.2	0.8

# 先验抽样 Prior Sampling

$T = \text{new}$
$D = \text{no}$
$C = \text{yes}$
$P = \text{no}$

$T = \text{new}$
$D = \text{yes}$
$C = \text{no}$
$P = \text{yes}$

$T = \text{new}$
$D = \text{yes}$
$C = \text{yes}$
$P = \text{yes}$

$T = \text{old}$
$D = \text{no}$
$T = \text{yes}$
$P = \text{yes}$

$T = \text{vip}$
$D = \text{yes}$
$C = \text{yes}$
$P = \text{yes}$

$T = \text{old}$
$D = \text{yes}$
$C = \text{yes}$
$P = \text{yes}$

$T = \text{old}$
$D = \text{no}$
$C = \text{no}$
$P = \text{no}$

$T = \text{new}$
$D = \text{no}$
$C = \text{yes}$
$P = \text{yes}$



# 拒绝抽样 Rejection Sampling

$T = \text{new}$
$D = \text{no}$
$C = \text{yes}$
$P = \text{no}$

$T = \text{new}$
$D = \text{yes}$
$C = \text{no}$
$P = \text{yes}$

$T = \text{vip}$
$D = \text{yes}$
$C = \text{yes}$
$P = \text{yes}$

$T = \text{old}$
$D = \text{no}$
$T = \text{yes}$
$P = \text{yes}$

$T = \text{vip}$
$D = \text{yes}$
$C = \text{yes}$
$P = \text{yes}$

$T = \text{old}$
$D = \text{yes}$
$C = \text{yes}$
$P = \text{yes}$

$T = \text{old}$
$D = \text{no}$
$C = \text{no}$
$P = \text{no}$

$T = \text{new}$
$D = \text{no}$
$C = \text{yes}$
$P = \text{yes}$

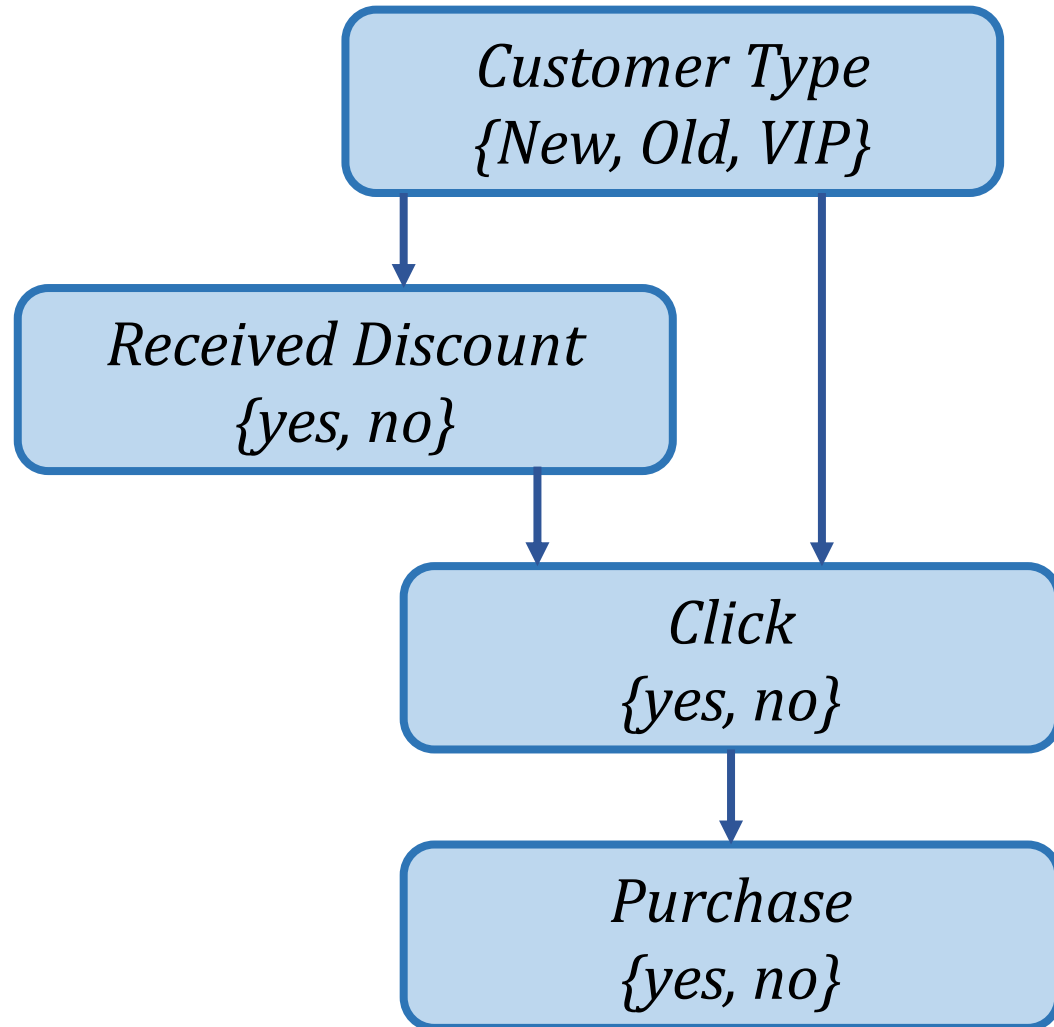
$$P(\text{Type} = \text{new} \mid \text{Click} = \text{yes}) = ? \quad 1/3$$

# 拒绝抽样的劣势

---

- 当证据  $e$  不太可能发生时，**效率很低**，因为抽出了大量无用的样本
- 解决办法: 加权抽样 likelihood weighting sampling
  - 首先固定证据变量的值
  - 使用贝叶斯网络中的条件概率对非证据变量进行抽样
  - 按**可能性**对每个抽样的样本进行加权
    - 可能性: 样本出现观测到证据的概率
  - 例:  $P(\text{Type} = \text{old} \mid \text{Click} = \text{yes})$

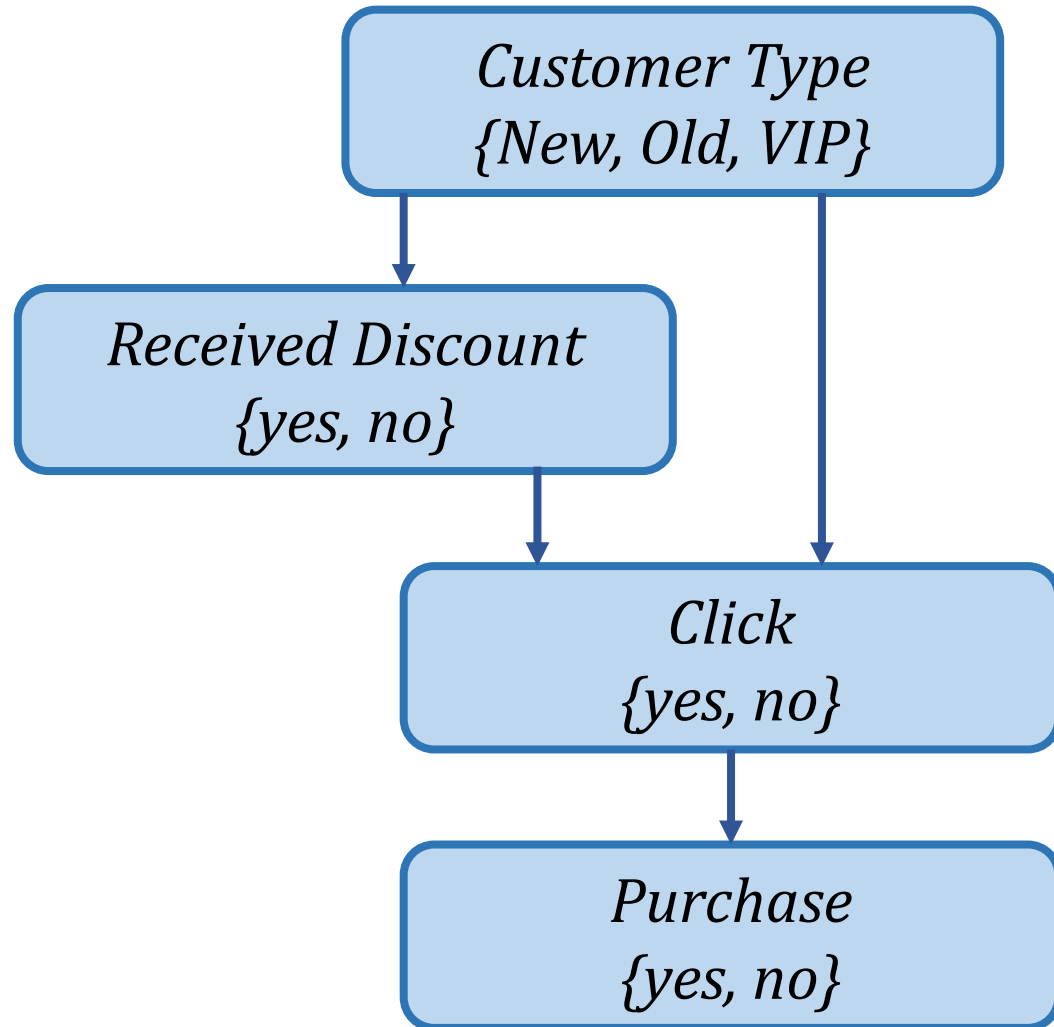
# 加权抽样 Likelihood weighting Sampling



<i>new</i>	<i>old</i>	<i>vip</i>
0.3	0.6	0.1

$T = old$
$C = yes$

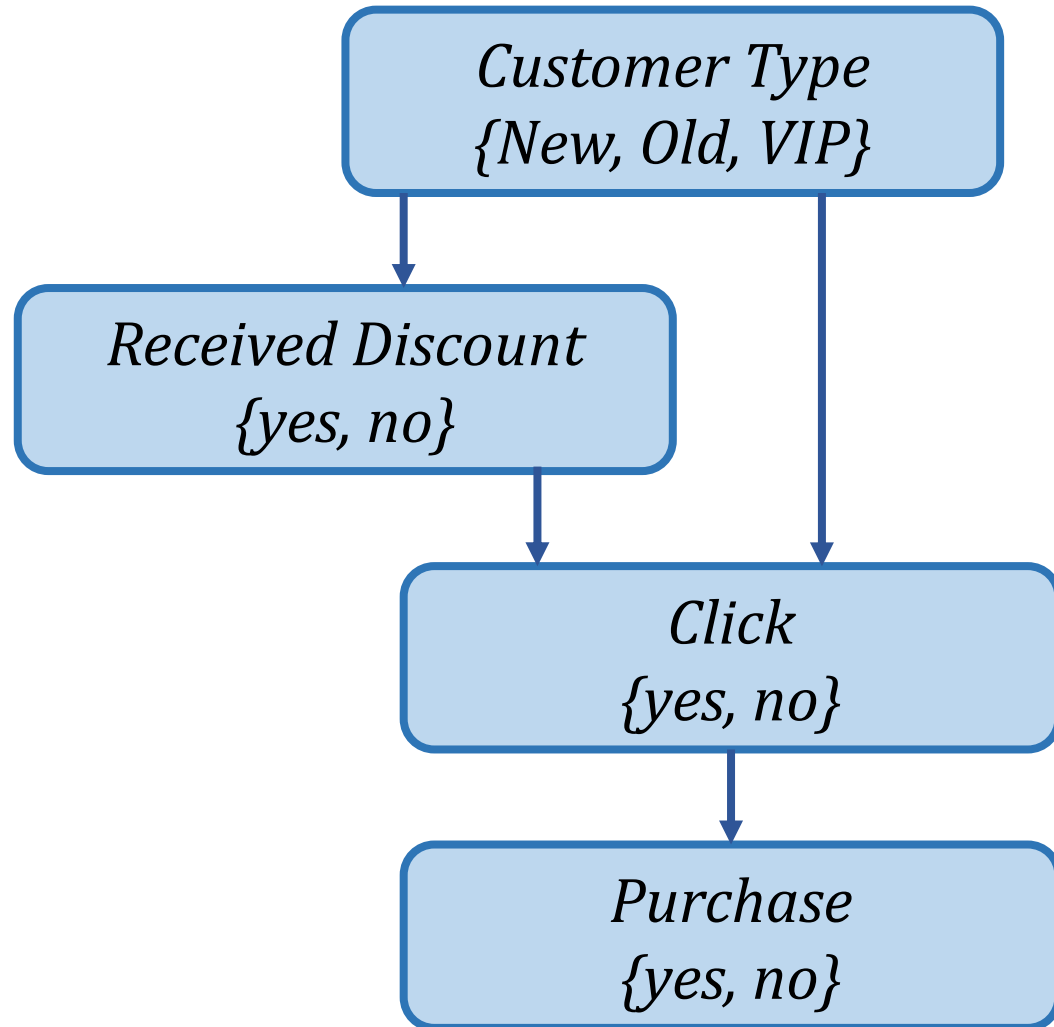
# 加权抽样 Likelihood weighting Sampling



<i>type</i>	<i>yes</i>	<i>no</i>
<i>new</i>	0.6	0.4
<i>old</i>	0.4	0.6
<i>vip</i>	0.8	0.2

$T = old$
$D = yes$
$C = yes$

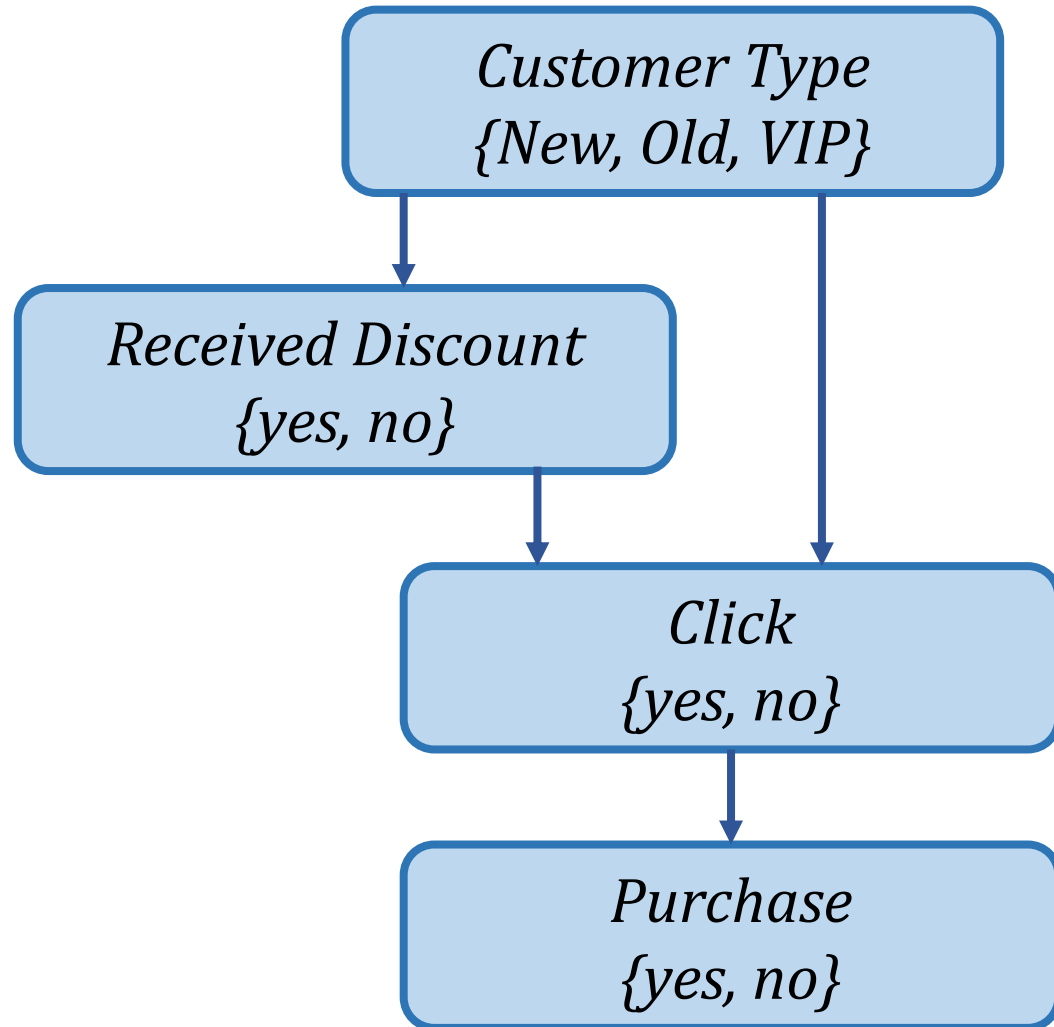
# 加权抽样 Likelihood weighting Sampling



$T = old$
$D = yes$
$C = yes$
$P = yes$

<i>Click</i>	<i>yes</i>	<i>no</i>
<i>yes</i>	0.6	0.4
<i>no</i>	0.2	0.8

# 加权抽样 Likelihood weighting Sampling



$T = old$
0.5 $D = yes$
$C = yes$
$P = yes$

<i>type</i>	<i>Discount</i>	<i>yes</i>	<i>no</i>
<i>new</i>	<i>yes</i>	0.3	0.7
<i>new</i>	<i>no</i>	0.1	0.9
<i>old</i>	<i>yes</i>	0.5	0.5
<i>old</i>	<i>no</i>	0.3	0.7
<i>vip</i>	<i>yes</i>	0.7	0.3
<i>vip</i>	<i>no</i>	0.5	0.5

# 加权抽样 Likelihood weighting Sampling

$T = old$	$T = new$	$T = vip$	$T = old$	$T = vip$
$D = no$	$D = yes$	$D = yes$	0.5 $D = yes$	$D = no$
$C = yes$	$C = yes$	$C = yes$	$C = yes$	$C = yes$
0.3 $P = no$	0.3 $P = yes$	0.7 $P = yes$	$P = yes$	0.5 $P = yes$

$T = old$
$D = no$
$C = yes$
0.3 $P = yes$

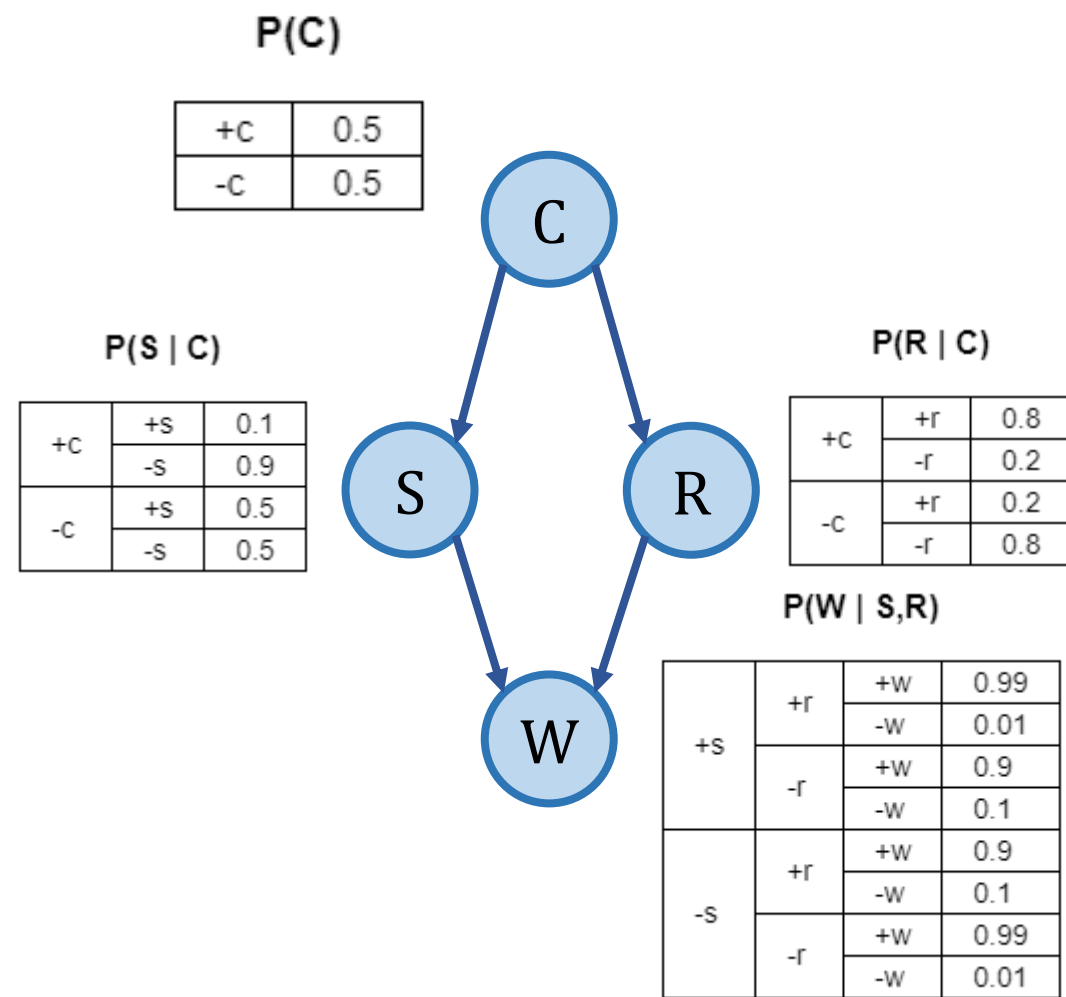
$P(\text{Type} = old \mid \text{Click} = yes)$

$$= \frac{0.3 + 0.5 + 0.3}{0.3 + 0.3 + 0.7 + 0.5 + 0.5 + 0.3} = 0.42$$

type	Discount	yes	no
new	yes	0.3	0.7
new	no	0.1	0.9
old	yes	0.5	0.5
old	no	0.3	0.7
vip	yes	0.7	0.3
vip	no	0.5	0.5

# 练习 #6

- 加权抽样
- +C, +S, -r, -w: 15次
- +C, +S, -r, -w: 4次
- +C, -S, -r, +w: 78次
- +C, -S, -r, -w: 3次
- 求:  $P(+s|+c, -r)$





# 有问题吗？

---

- 请随时举手提问。



BUSS 3620.人工智能导论

# #4. 马尔科夫链

刘佳璐

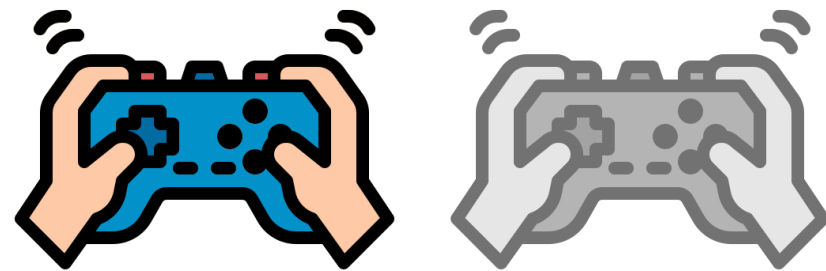
安泰经济与管理学院

上海交通大学

# 目前为止

- 某一个随机变量的不确定性
  - 离散的
- 无法处理一个变量会随时间而变化的情况
  - 每个时间点都有一个值
  - $X_t$ :  $t$  时的客户状态  $\{active, inactive\}$
  - 使用过去每一天的数据来预测明天的客户类型是**不可行的**

*Customer Type*  
 $\{new, old, vip\}$




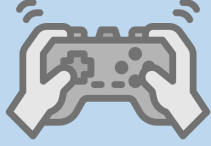

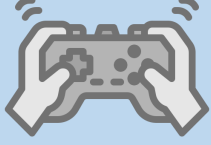
# 一些让问题简单的假设


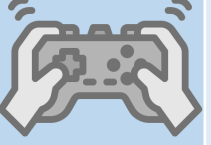
---

- 马尔可夫假设 Markov assumption
  - 假设当前状态仅依赖于有限的固定数量的过去状态
    - 例如，今天的客户状态仅取决于其昨天的客户状态
- 马尔可夫链 Markov Chain
  - 马尔可夫链是随机变量 $X_1, X_2, X_3, \dots$ 的一个数列，其中每个变量都遵从马尔科夫假设。

# 马尔科夫链示例

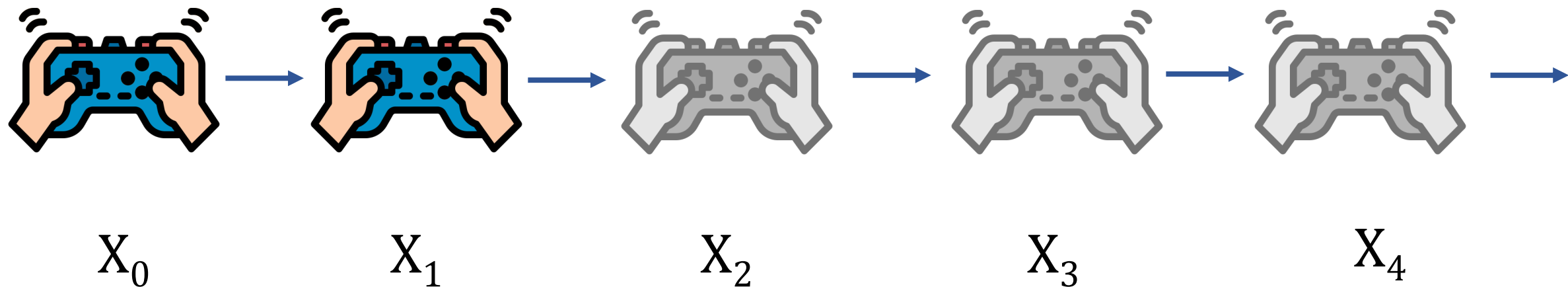
- 预测明天的客户状态  $\{active, inactive\}$
- 需要一个**转换模型/转移模型(transition model)**来构建马尔可夫链
- 对链进行抽样
  - 需要**初始概率分布(initial distribution)**

		Tomorrow ( $X_{t+1}$ )	
Today ( $X_t$ )			
		0.8	0.2
		0.3	0.7

	$P(X_0)$
	0.5
	0.5

# 马尔科夫链示例




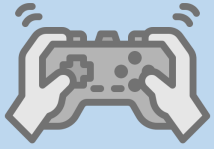
- 相当于一个贝叶斯网络

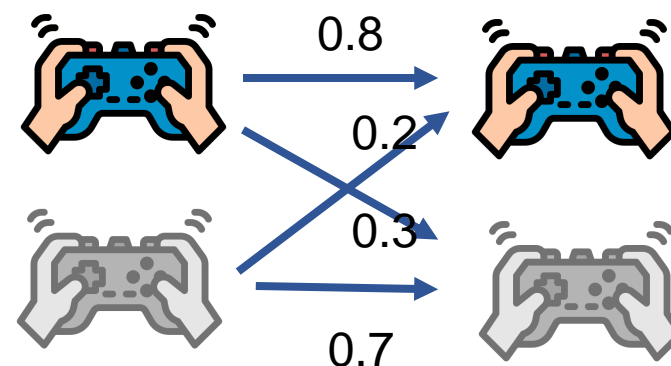
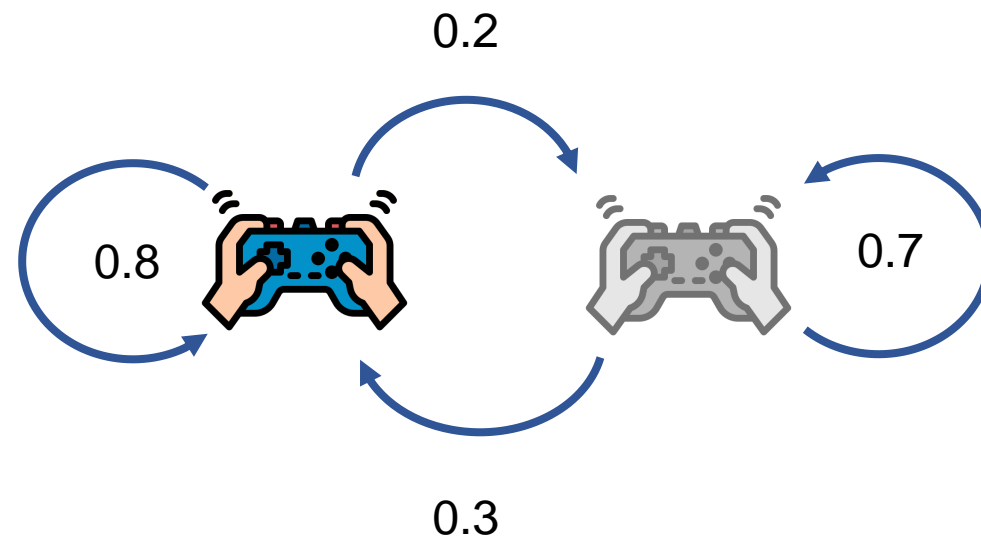


# 三种方法表示转移模型 transition model



Tomorrow ( $X_{t+1}$ )

Today ( $X_t$ )







		
	0.8	0.2
	0.3	0.7



# 练习 #7

	$P(X_0)$
	0.5
	0.5

初始概率

	Tomorrow ( $X_{t+1}$ )	
Today ( $X_t$ )		
		
	0.8	0.2
	0.3	0.7

转移模型

求:  $P(X_1=active)$



# 迷你前向算法 Mini-Forward Algorithm

---

- 求  $P(X_t)$

$$P(X_t) = \sum_{X_{t-1}} P(X_t, X_{t-1}) \quad \text{边缘化}$$

$$P(X_t) = \sum_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}) \quad \text{贝叶斯法则}$$

# 迷你前向算法示例

$$X_t \\ \{active, inactive\}$$

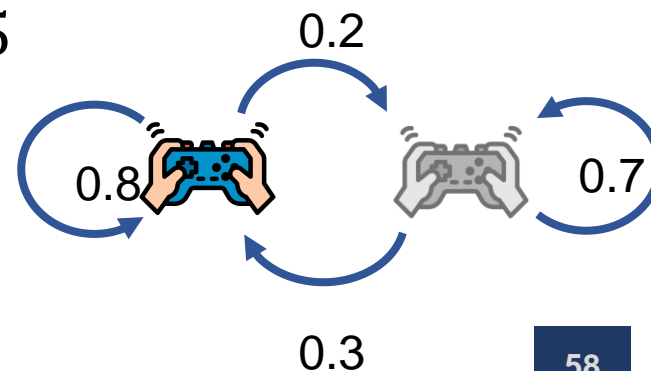
$P(X_0)$	$P(X_1)$	$P(X_2)$	$P(X_3)$	.....	$P(X_\infty)$
$=\langle 0.5, 0.5 \rangle$	$=\langle 0.55, 0.45 \rangle$	$=\langle 0.575, 0.425 \rangle$	$=\langle 0.5875, 0.4125 \rangle$		$=\langle 0.6, 0.4 \rangle$

$$P(X_1 = active) = 0.8 \times 0.5 + 0.3 \times 0.5 = 0.55$$

$$P(X_2 = active) = 0.8 \times 0.55 + 0.3 \times 0.45 = 0.575$$

$$P(X_3 = active) = 0.8 \times 0.575 + 0.3 \times 0.425 = 0.5875$$

$$P(X_{\infty+1} = active) = 0.8 \times 0.6 + 0.3 \times 0.4 = 0.6$$



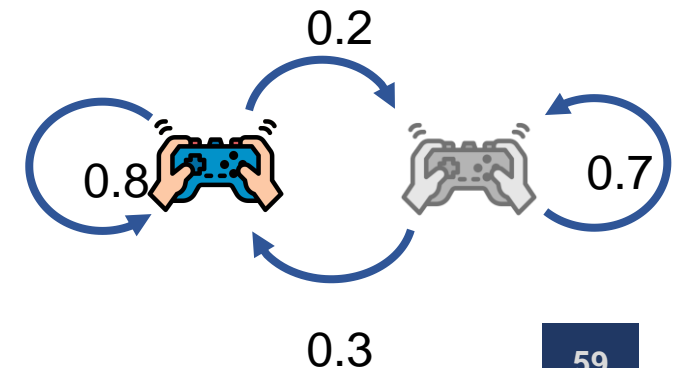
# 平稳分布 Stationary Distribution

$X_t$   
 $\{active, inactive\}$

$P(X_0)$	$P(X_1)$	$P(X_2)$	$P(X_3)$	.....	$P(X_\infty)$
$=\langle 0.5, 0.5 \rangle$	$=\langle 0.55, 0.45 \rangle$	$=\langle 0.575, 0.425 \rangle$	$=\langle 0.5875, 0.4125 \rangle$		$=\langle 0.6, 0.4 \rangle$
$=\langle 0.7, 0.3 \rangle$	$=\langle 0.65, 0.35 \rangle$	$=\langle 0.625, 0.375 \rangle$	$=\langle 0.6125, 0.3875 \rangle$		$=\langle 0.6, 0.4 \rangle$
$=\langle 0, 1 \rangle$	$=\langle 0.3, 0.7 \rangle$	$=\langle 0.45, 0.55 \rangle$	$=\langle 0.525, 0.475 \rangle$		$=\langle 0.6, 0.4 \rangle$

初始概率分布  $P(X_0)$  无关紧要

$$P(X_{\infty+1}) = \sum_{X_\infty} P(X_{\infty+1}|X_\infty)P(X_\infty) = P(X_\infty)$$



# 平稳分布 Stationary Distribution

- 平稳分布时马尔可夫链的上随着时间变化概率分布保持不变的概率分布。它满足:

$$P(X_{\infty+1}) = \sum_{X_{\infty}} P(X_{\infty+1}|X_{\infty})P(X_{\infty}) = P(X_{\infty})$$

$$P(X_{\infty+1} = active) = 0.8 \times P(X_{\infty} = active) + 0.3 \times P(X_{\infty} = inactive) = P(X_{\infty} = active)$$

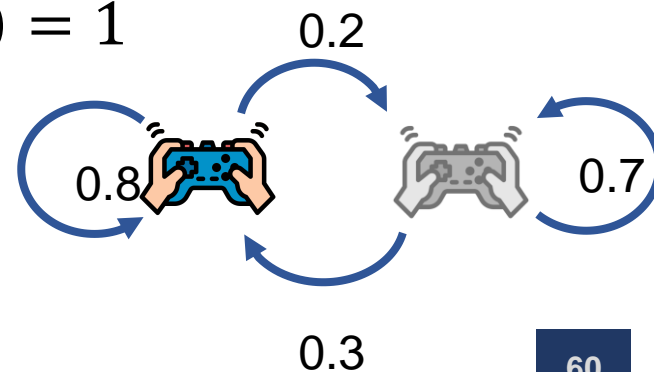
$$0.3 \times P(X_{\infty} = inactive) = 0.2 \times P(X_{\infty} = active)$$

$$P(X_{\infty} = active) + P(X_{\infty} = inactive) = 1$$

$$P(X_{\infty} = active) = 0.6$$

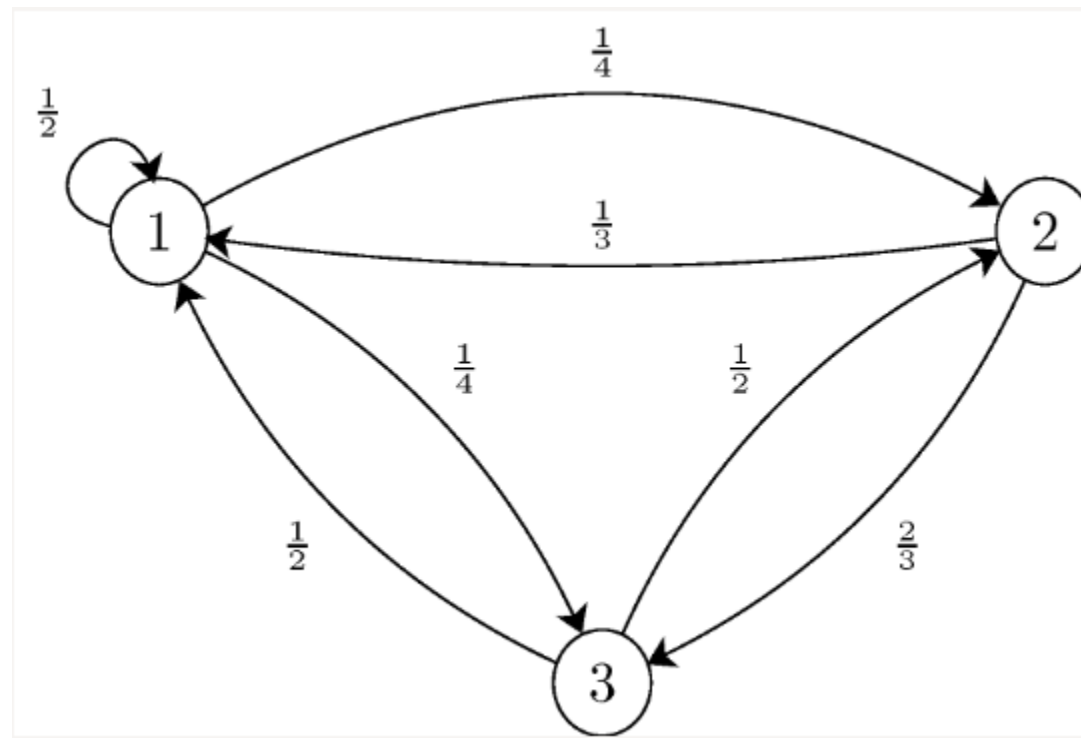
$$P(X_{\infty} = inactive) = 0.4$$

$$P(X_{\infty}) = \langle 0.6, 0.4 \rangle$$



## 练习 #8

- 列出求右图马尔可夫链的平稳分布的式子



# 有问题吗？

---

- 请随时举手提问。



BUSS 3620.人工智能导论

# #5. 马尔科夫链示例： 谷歌PageRank(代码)

刘佳璐

安泰经济与管理学院

上海交通大学

# PageRank

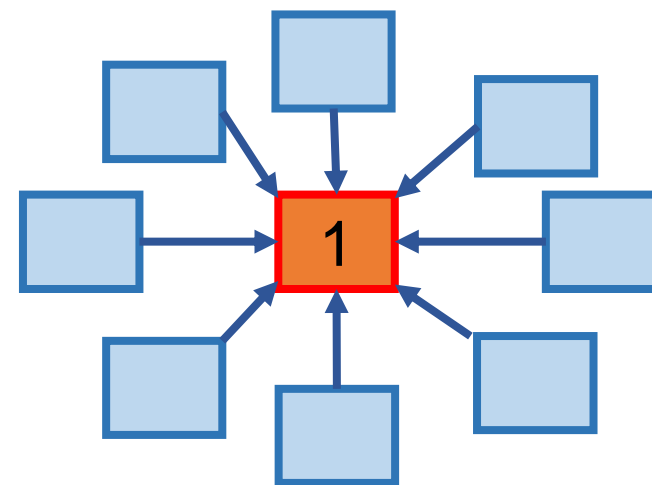
---

- 假设我们搜索人工智能。有太多的网页包含这个词。我们应该先显示哪一个？
  - 随机显示
  - 工作人员人工排序
  - 根据字数
  - .....
- 类似的问题
  - 谁是现今中国人中最具影响力的人物？
    - 主席
    - 每个中国人都认识主席
  - 谁是最具影响力的带货主播？
    - 李佳琪
    - 大部分人都听说过
  - 哪篇论文是你领域中的最具影响力的论文？
    - 引用次数多的论文
    - 其他论文引用过这篇论文



# PageRank

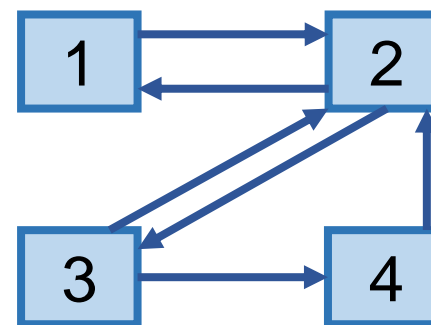
- 重要的网页：从其他网页可以跳转/链接到这个网页→其他网页含有这个网页的超链接
  - 给每个网页打一个重要性的分数(Page Rank Score)
    - 来自其他网页的的跳转超链接(incoming link)的数量
    - 潜在问题
      - 人为的创建许多网页，都含有超链接到所需网页，从而提高排名
    - 解决办法
      - 同时考虑：
        - 有多少其他网页能够跳转到这个网页
        - 其他网页的重要性



# 随机游走模型 Random surfer model

---

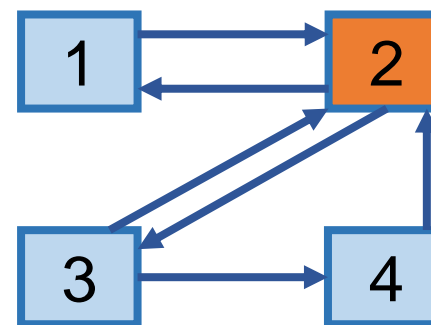
- 描述一个用户随机访问某一个网页的概率的图模型



# 随机游走模型 Random surfer model

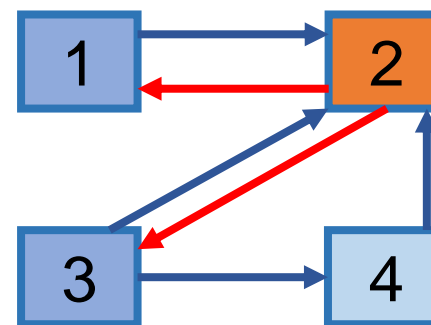
---

- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始



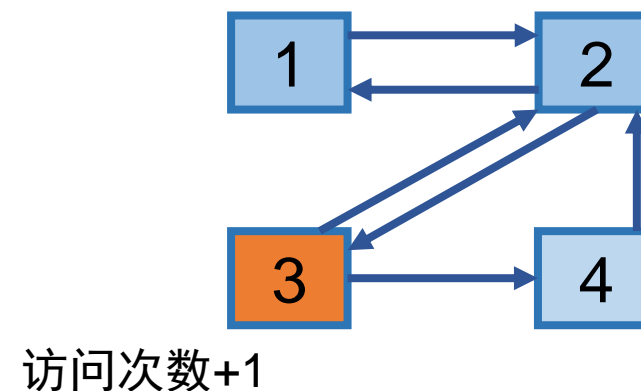
# 随机游走模型 Random surfer model

- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始
  - 选择下一个网页



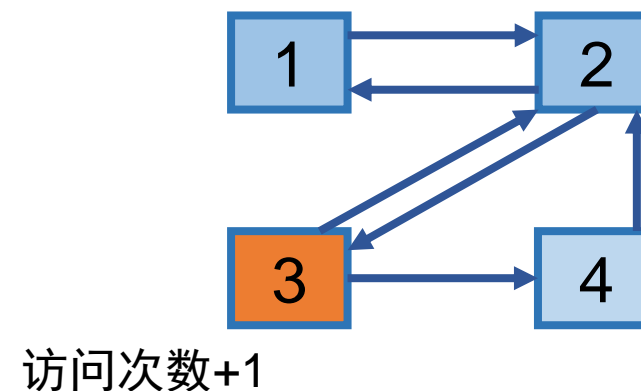
# 随机游走模型 Random surfer model

- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始
  - 选择下一个网页



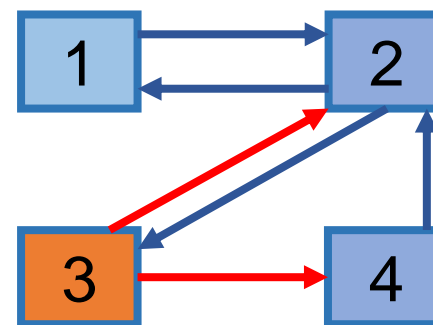
# 随机游走模型 Random surfer model

- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始
  - 选择下一个网页
  - 重复 ...



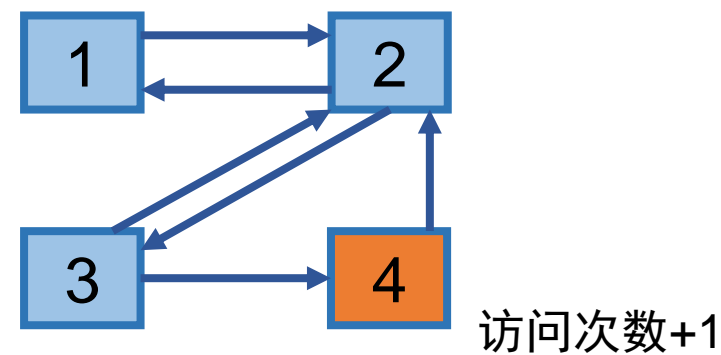
# 随机游走模型 Random surfer model

- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始
  - 选择下一个网页
  - 重复 ...



# 随机游走模型 Random surfer model

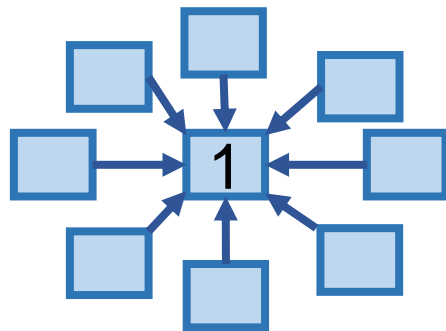
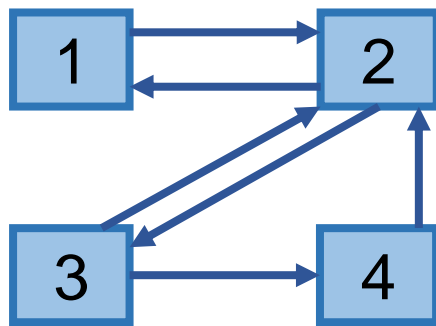
- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始
  - 选择下一个网页
  - 重复 ...





# 随机游走模型 Random surfer model

- 描述一个用户随机访问某一个网页的概率的图模型
  - 从随机一个网页开始
  - 选择下一个网页
  - 重复 ...
- Page Rank 分数
  - 跟踪  $T$  个时刻
  - 访问这个网页的次数 /  $T$



# 随机游走模型 Random surfer model

- 潜在问题

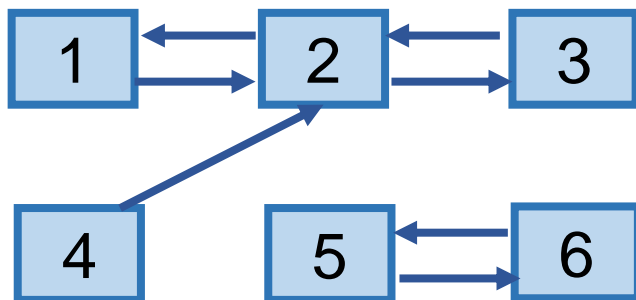
- 从网页5开始

- 永远无法跳转到网页1 – 4

- PageRank 分数

- 1 – 4: 0

- 5 – 6: 0.5



- 解决办法

- 阻尼因子 Damping factor  $d$

- 概率 $d$ : 根据当前网页中的超链接跳转到下一个网页

- 概率 $1-d$ : 从所有网页中随机选一个跳转

- $d=0.85$

# PageRank 分数的公式

- $PR(p)$ : 一个用户随机访问页面 $p$ 的概率

$$PR(p) = \frac{1-d}{N} + d \sum_{P_0} \frac{1}{Numlinks(P_0)} PR(P_0)$$

- $PR(p) = \sum_{p_0} P(P|P_0) PR(P_0)$

→ 概率  $d$ : 根据当前网页中的超链接跳转到下一个网页

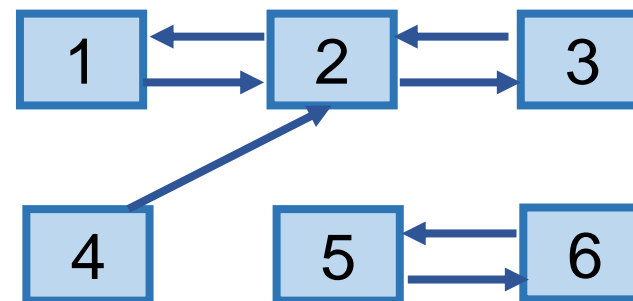
- $d \sum_{P_0} \frac{1}{Numlinks(P_0)} PR(P_0)$

→ 概率  $1-d$ : 从全部 $N$ 个网页中随机选一个跳转

- $(1-d) \sum_{P_0} \frac{1}{N} PR(P_0) = \frac{1-d}{N}$

## 马尔可夫链

- 到某网页的概率只和前一个网页有关
- 转移模型 Transition model



# 马尔科夫链的平稳分布 Stationary Distribution

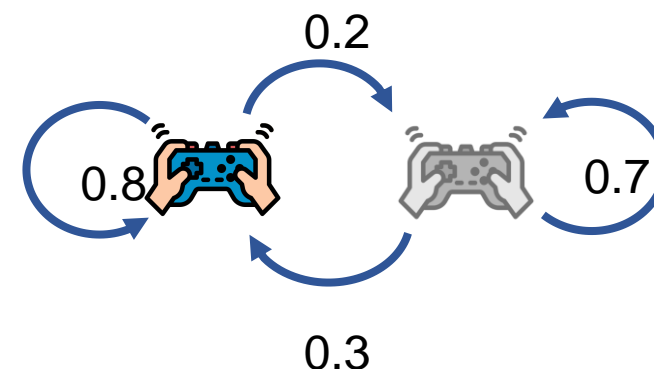
$$X_t \\ \{active, inactive\}$$

$P(X_0)$	$P(X_1)$	$P(X_2)$	$P(X_3)$	.....	$P(X_\infty)$
$=\langle 0.5, 0.5 \rangle$	$=\langle 0.55, 0.45 \rangle$	$=\langle 0.575, 0.425 \rangle$	$=\langle 0.5875, 0.4125 \rangle$		$=\langle 0.6, 0.4 \rangle$
$=\langle 0.7, 0.3 \rangle$	$=\langle 0.65, 0.35 \rangle$	$=\langle 0.625, 0.375 \rangle$	$=\langle 0.6125, 0.3875 \rangle$		$=\langle 0.6, 0.4 \rangle$

初始概率分布 $P(X_0)$  无关紧要

$$P(X_{\infty+1}) = \sum_{X_\infty} P(X_{\infty+1}|X_\infty)P(X_\infty) = P(X_\infty)$$

平稳分布即为PageRank的分数，可以直接求出来



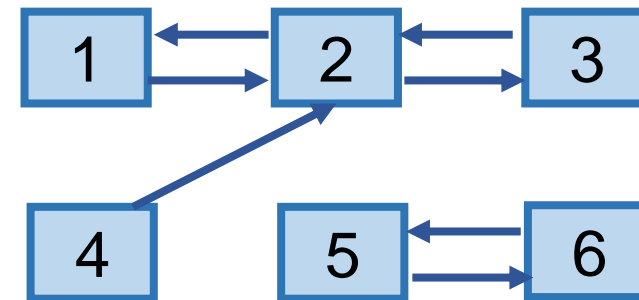
# 迭代算法 Iterative Algorithm

- 转换模型 Transition model

- $PR(p) = \frac{1-d}{N} + d \sum_{P_0} \frac{1}{Numlinks(P_0)} PR(P_0)$

- 算法步骤:

- 将每个网页的初始概率设置为  $\frac{1}{N}$
  - 重复:
    - 使用转换模型计算新的概率
    - 直到它收敛(变化不超过一个自定义的小阈值)



# 实现 PageRank

- 爬取网站并储存 (假设已经做好了)
  - 每个搜索单词(query)包含哪些网页也已经分类完成
- 抽取网页之间的链接
- 计算 PageRank 分数
- 输出结果

```
if __name__ == "__main__":  
    # command line argument format  
    if len(sys.argv) != 2:  
        sys.exit("Usage: python pagerank.py corpus")  
  
    # Retrive the links between pages  
    corpus = crawl(sys.argv[1])  
  
    # Generate PageRank Score using Random Suffer Model  
    ranks = sample_pagerank(corpus, DAMPING, SAMPLES)  
  
    # Print out the result in the terminal  
    print(f"PageRank Results from Sampling (n = {SAMPLES})")  
    for page in sorted(ranks):  
        print(f"  {page}: {ranks[page]:.4f}")  
  
    # Generate PageRank Score using Iterative Algorithm  
    ranks = iterate_pagerank(corpus, DAMPING)  
  
    # Print out the result in the terminal  
    print(f"PageRank Results from Iteration")  
    for page in sorted(ranks):  
        print(f"  {page}: {ranks[page]:.4f}")
```

# 爬取网站并储存

- 编写网页的语言是HTML  
(Hypertext Markup Language)
  - 不是编程语言
  - 用于排版网页（审美上的）

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <title>
```

```
      hello, title
```

```
    </title>
```

```
  </head>
```

```
  <body>
```

```
    Visit <a href="https://www.liujialu.org">Me</a>.
```

```
  </body>
```

```
</html>
```

# 抽取网页之间的链接

- 网页之间的联系可以储存在一个字典中
  - {'1.html': {'2.html'}}
  - 网页1包含到网页2的链接
  - 可以从网页1跳转到网页2
- 读取 html 网页文件
- 找到哪一行文本匹配<a href="*SomeLink*">
- 同一网页中的重复链接被视为一个链接
- 忽略从一个网页到自身的超链接
- 只包含同一个网页库(corpus)的网页的链接

```
9 def crawl(directory):
10     """
11     Parse a directory of HTML pages and check for links to other pages.
12     Return a dictionary where each key is a page, and values are
13     a list of all other pages in the corpus that are linked to by the page.
14     """
15     pages = dict()
16
17     # Extract all links from HTML files
18     for filename in os.listdir(directory):
19         if not filename.endswith(".html"):
20             continue
21         with open(os.path.join(directory, filename)) as f:
22             contents = f.read()
23             links = re.findall(r"<a\s+(?:[^\>]*?)href=\"([^\"]*)\"", contents)
24             pages[filename] = set(links) - {filename}
25
26     # Only include links to other pages in the corpus
27     for filename in pages:
28         pages[filename] = set(
29             link for link in pages[filename]
30             if link in pages
31         )
32
33     return pages
```



# 计算 PageRank 分数：随机游走模型

- 练习 #9: 编写函数 `transition_model()`:
- 输入: 网页库 `corpus`, 一个网页  $p$ , 阻尼系数  $d$
- 功能: 返回从网页  $p$  接下来要能跳转到哪个页面及其概率 (转换模型 `transition model`)
  - 阻尼系数 Damping factor  $d$ 
    - 概率  $d$ : 根据当前网页中的超链接跳转到下一个网页
    - 概率  $1-d$ : 从所有网页中随机选一个跳转
  - $$PR(p) = \frac{1-d}{N} + d \sum_{P_0} \frac{1}{Numlinks(P_0)} PR(P_0)$$
- 输出: 一个字典, key是可以跳转的网页, value是跳转的概率

- 例: `{"1.html": {"2.html", "3.html"}}, d=0.85`
  - 当且页面是 1.html, 网页库一共含有3个页面
    - 1.html:  $(1 - 0.85) \times \frac{1}{3} = 0.05$
    - 2.html:  $(1 - 0.85) \times \frac{1}{3} + 0.85 \times \frac{1}{2} = 0.475$
    - 3.html:  $(1 - 0.85) \times \frac{1}{3} + 0.85 \times \frac{1}{2} = 0.475$
    - 返回 `{"1.html": 0.05, "2.html": 0.475, "3.html": 0.475}`
- 如果某一个网页没有跳转到其他网页的超链接, 会随机从所有页面中选择选择一个网页跳转

# 计算 PageRank 分数：随机游走模型

- 练习 #10
- 编写函数sample\_pagerank():
  - 输入：网页库corpus，阻尼系数d,需要观测的时间T
  - 功能：观察T个时刻，计算一个用户随机停留在corpus中每一个网页的概率
  - 输出：一个字典，key是corpus中每一个网页，value是计算的Page Rank 分数（停留在每一个网页的概率）
- 提示：随机游走模型
- 从网页库Corpus中随机选择一个网页
  - 该网页的访问次数增加 1
- 基于转换模型，选择跳转的网页
  - 该网页的访问次数增加 1
- 计算概率
  - 跟踪 T 个时刻
  - 访问这个网页的次数 / T

# 计算 PageRank 分数：迭代算法

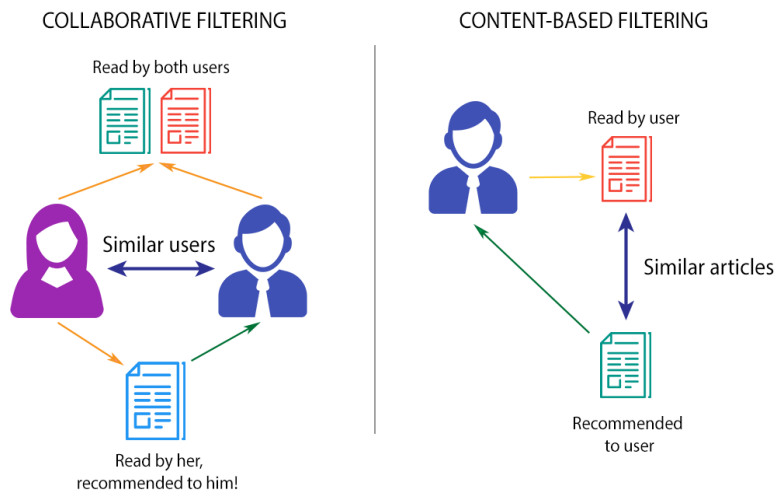
- 练习 #11
- 编写函数 `iterate_pagerank()` :
  - 输入：网页库 `corpus`，阻尼系数 `d`
  - 功能：用迭代算法计算马尔科夫链的平稳分布
  - 输出：一个字典，key是 `corpus` 中每一个网页，value是计算的 Page Rank 分数（每一个网页在马尔科夫链的概率分布）

- 提示：迭代算法
- 将每个网页的初始概率设置为  $\frac{1}{N}$
- 重复：
  - 使用转换模型计算新的概率
  - 直到它收敛(变化不超过一个自定义的小阈值0.001)

$$PR(p) = \frac{1-d}{N} + d \sum_{P_0} \frac{1}{Numlinks(P_0)} PR(P_0)$$

# 推荐算法

- 产品/视频/文章
- 给每个产品/视频打分
  - 产品或视频之间没有超链接
  - 找到产品或视频之间**连接/关联**的地方



- 关联的link
  - **共同**的消费习惯
    - 购买此产品的人也会购买其他产品
    - 观看此视频的人也会观看另一个视频
  - **共同**话题
    - 产品属于同一类别
    - 视频具有类似的内容
- 阻尼系数damping factors  $d$  又指?
  - 概率  $d$ : 根据联系进行推荐
  - 概率  $1-d$ : **推荐新产品/视频**

# 有问题吗？

---

- 请随时举手提问。



BUSS 3620.人工智能导论

# #6. 隐马尔可夫模型

刘佳璐

安泰经济与管理学院

上海交通大学





# 真实的状态是未知的

- 不知道世界的真实状态（例如，用户是否感兴趣）
- 需要一些传感器sensor来观察

AI任务	隐状态	观测
机器人定位	机器人的位置	机器人的传感器数据
语音识别	说的话	音频波形
用户关注	用户是否感兴趣	浏览时间\互动情况
信用卡欺诈检测	交易的真实性	交易特征（如金额、时间）
金融市场分析	市场状态（牛市、熊市）	每日的股票价格变化

# 隐马尔可夫模型 Hidden Markov Model

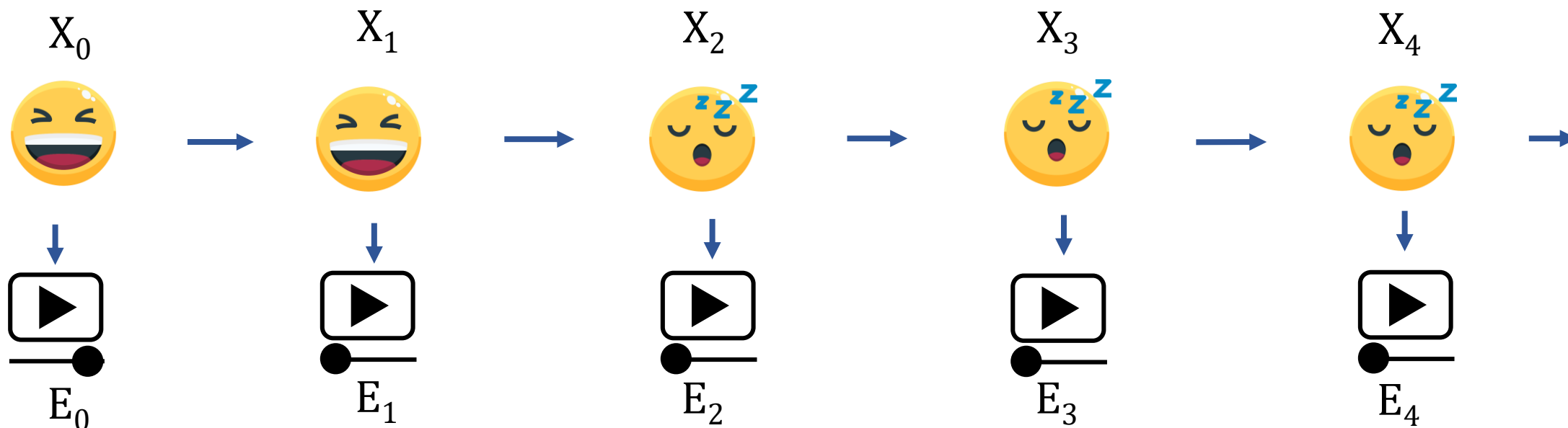
- 描述一个含有**隐含未知参数**的马尔可夫过程。其难点是从**可观察的参数**中确定该过程的隐含参数。
- 除了初始分布 initial distribution 和转换模型 transition model
  - 还需要**传感器模型 sensor model** (又称观测概率 emission probabilities)

		Observation ( $E_t$ )	
			
State ( $X_t$ )		0.2	0.8
		0.9	0.1



# 观测独立性假设 Sensor Markov assumption

- 任意时刻的观测到的变量只依赖当前时刻的状态，与其他状态及观测值无关
- 例如，当前时刻是否看完视频完全取决于当前时刻用户的情绪



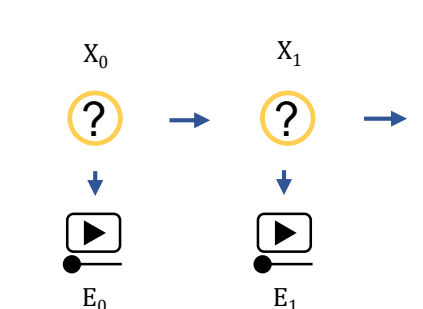
# 隐马尔可夫模型的应用

---

- 滤波 filtering（关注当下）
  - 根据全部历史观测估计当前的状态。
  - 例：电视连环杀人案，根据之前不同死法，估计当前人的死法的可能性。根据之前的办案经验，判断当前案件的可能性。
- 预测 prediction（关注未来）
  - 根据全部历史观测估计未来的状态。
  - 例：电视连环杀人案，根据之前不同的死亡案件，估计下一个死者可能是谁。
- 平滑 smoothing（关注过去）
  - 根据全部历史观测估计过去的状态。
  - 例：复盘，往前想想往后看看，根据所有人的情况，去分析某一个人的死亡原因(或者别的想法)。
- 最可能的解释 Most likely explanation
  - 根据全部历史观测估计，计算最可能的状态序列

# 计算当前和未来状态的概率

- 根据全部历史观测估计当前及未来的状态



$X_t$   
 $\{engaged, bored\}$

$E_t$   
 $\{short, long\}$

$P(X_0)$

$P(X_0|E_0 = short)$

$P(X_1|E_0 = short)$

$P(X_1|E_0 = take, E_1 = take)$

$=\langle 0.5, 0.5 \rangle$

$=\langle 0.18, 0.82 \rangle$

$=\langle 0.39, 0.61 \rangle$

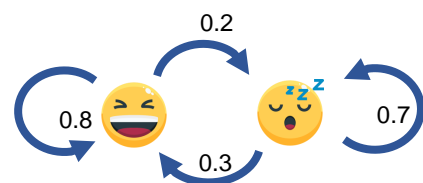
$=\langle 0.12, 0.88 \rangle$

没有任何观测时

观测到  $E_0$

观测到  $E_0$

观测到  $E_0, E_1$



Observation ( $E_t$ )

State ( $X_t$ )		
	0.2	0.8
	0.9	0.1

$$P(X_0|E_0 = short) = \alpha P(X_0, E_0 = short) = \alpha P(E_0 = short|X_0)P(X_0) = \alpha \langle 0.2 \times 0.5, 0.9 \times 0.5 \rangle = \langle 0.18, 0.82 \rangle$$

$$P(X_1|E_0 = short) = P(X_1, X_0 = engaged|E_0 = short) + P(X_1, X_0 = bored|E_0 = short)$$

$$= P(X_1|X_0 = engaged, E_0 = short)P(X_0 = engaged|E_0 = short) + P(X_1|X_0 = bored, E_0 = short)P(X_0 = bored|E_0 = short)$$

$$= P(X_1|X_0 = engaged)P(X_0 = engaged|E_0 = short) + P(X_1|X_0 = bored)P(X_0 = bored|E_0 = short)$$

$$= \langle 0.8 \times 0.18 + 0.3 \times 0.82, 0.2 \times 0.18 + 0.7 \times 0.82 \rangle = \langle 0.39, 0.61 \rangle$$

$$P(X_1|E_0 = short, E_1 = short) = \alpha P(X_1, E_1 = short|E_0 = short) = \alpha P(E_1 = short|X_1, E_0 = short)P(X_1|E_0 = short)$$

$$= \alpha P(E_1 = short|X_1)P(X_1|E_0 = short) = \alpha \langle 0.2 \times 0.39, 0.9 \times 0.61 \rangle = \langle 0.12, 0.88 \rangle$$

# 近似推断 Approximate inference

---

- 有时随机变量 $x$ 可能因为定义域太大而无法使用精确推理Exact inference
  - 例如，连续变量（位置）
- 解决办法: 粒子滤波 Particle Filtering (抽样方法)
  - 粒子只是**样本**的新名称
  - 通常，抽取的样本数小于实际的状态数

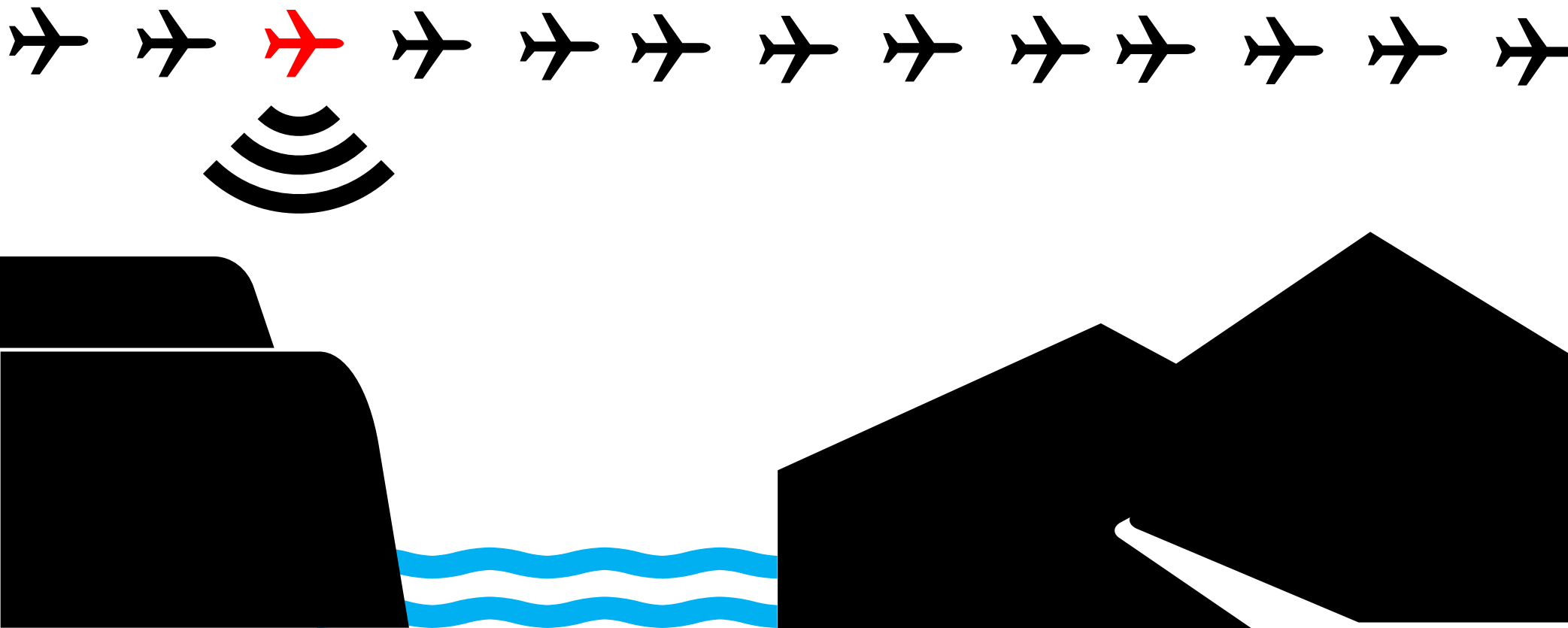
# 粒子滤波 Particle Filtering: 定位

---



# 粒子滤波 Particle Filtering: 定位

在飞机可能出现的地方随机抽样



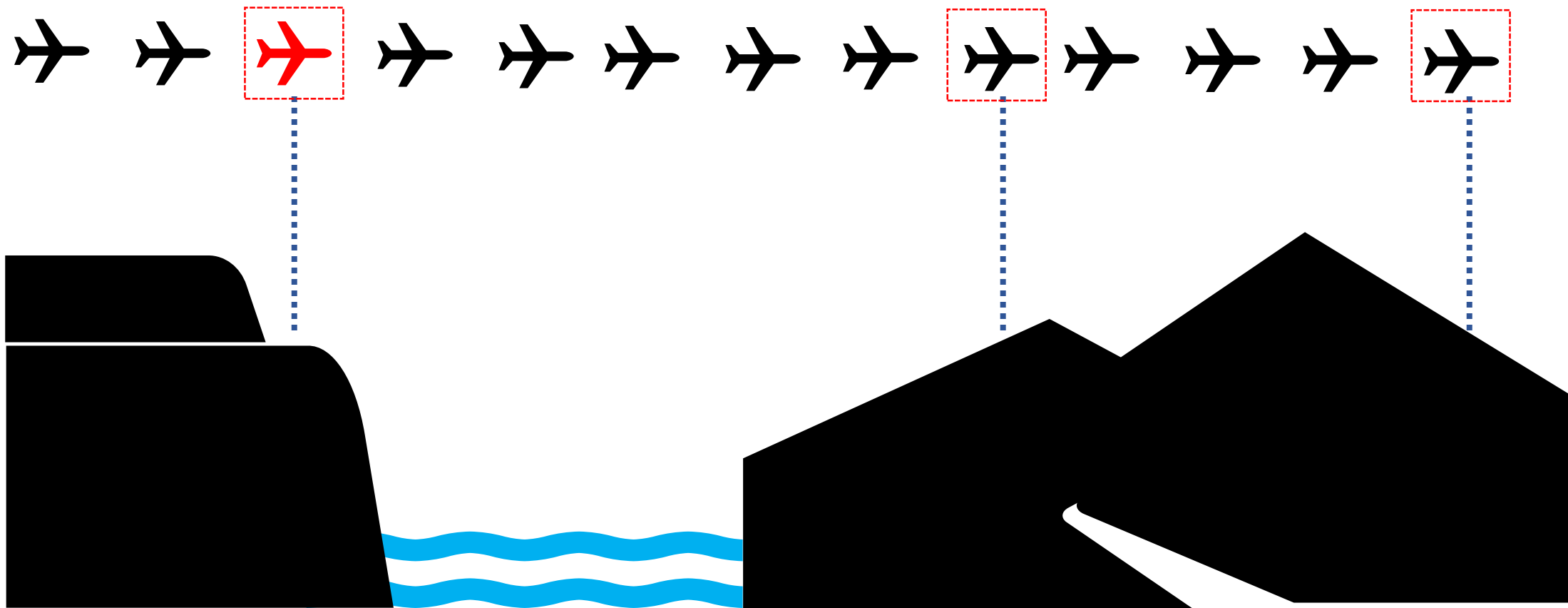
# 粒子滤波 Particle Filtering: 定位

观察传感器的数据



# 粒子滤波 Particle Filtering: 定位

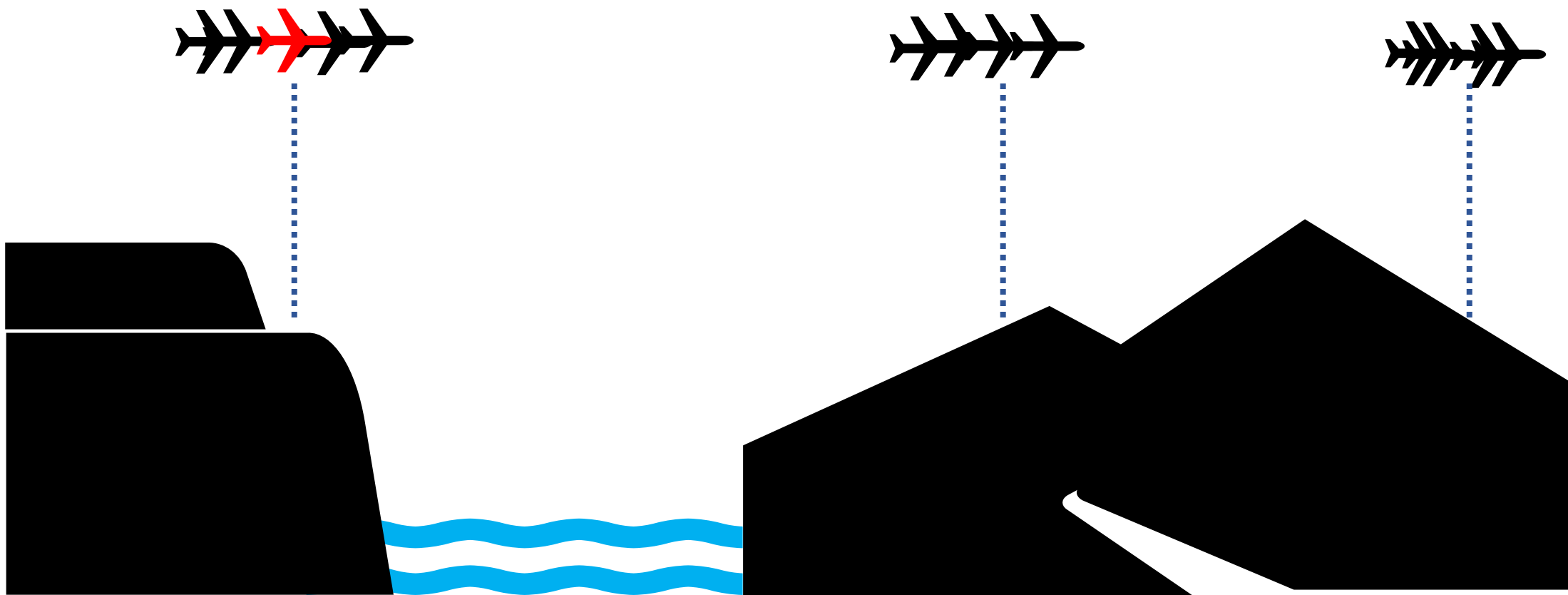
判断哪些样本更有概率是飞机目前的位置





# 粒子滤波 Particle Filtering: 定位

重新在这些地方抽样



# 粒子滤波 Particle Filtering: 定位

时间在流逝，飞机也在向前飞



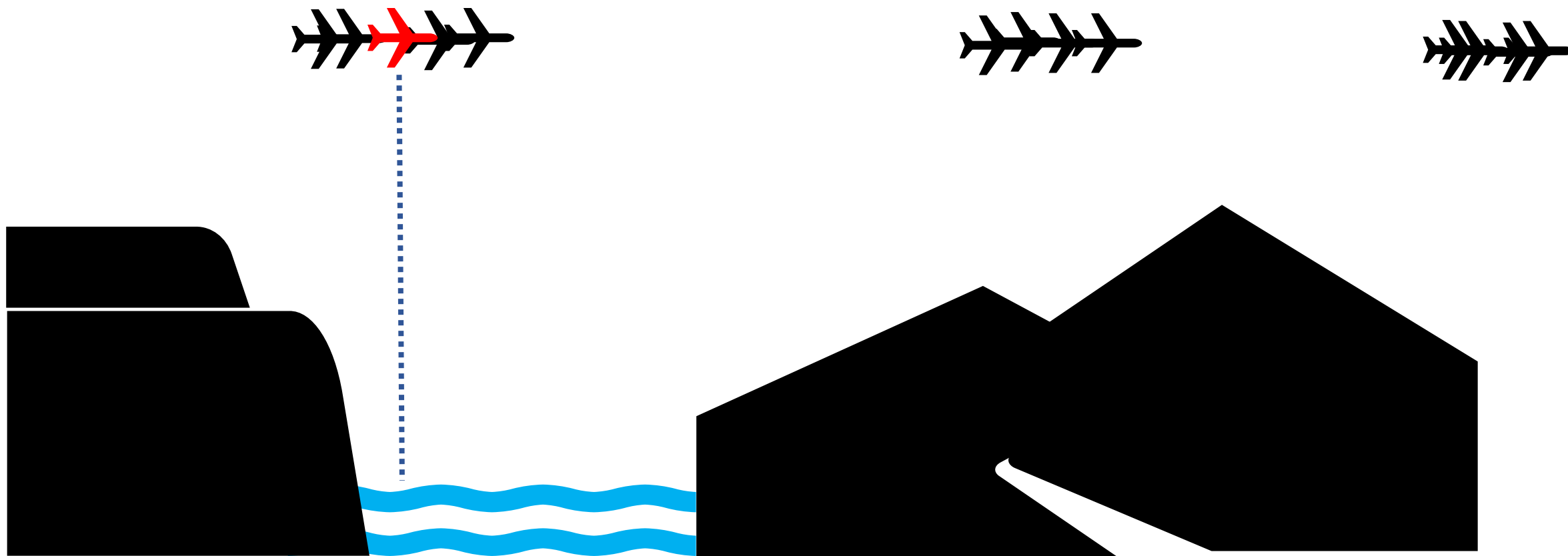
# 粒子滤波 Particle Filtering: 定位

时间在流逝，飞机也在向前飞



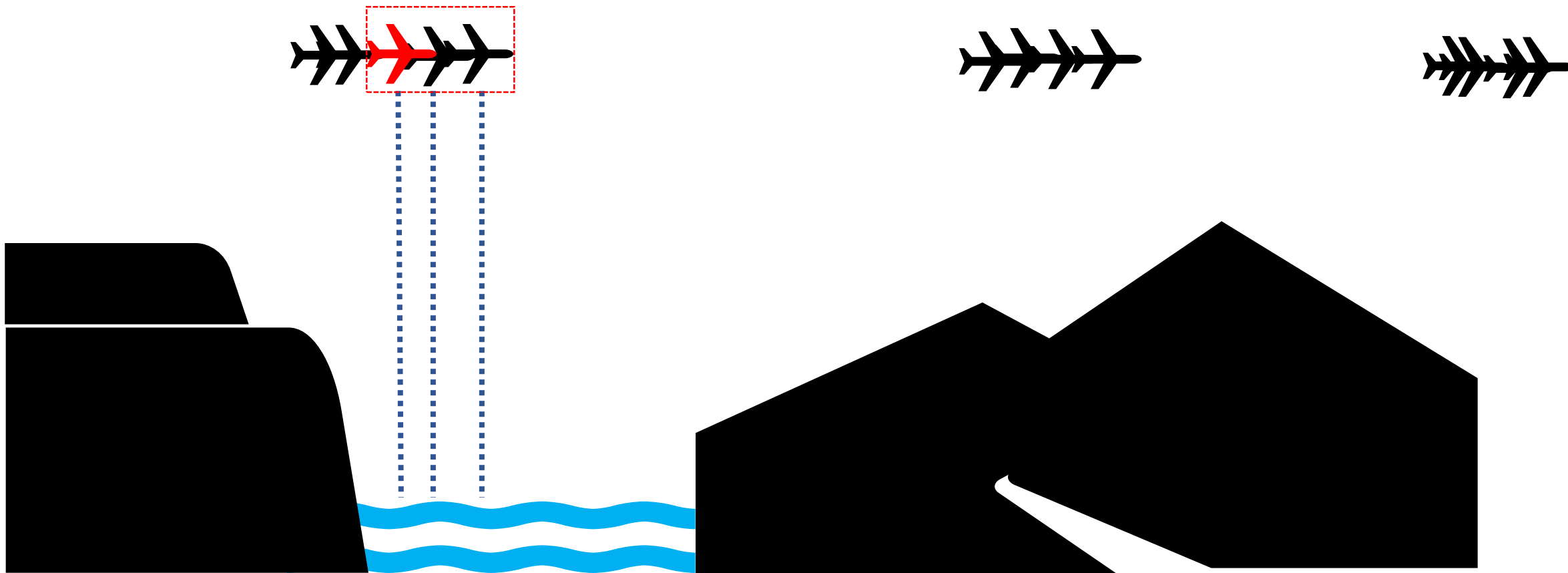
# 粒子滤波 Particle Filtering: 定位

观察传感器的数据

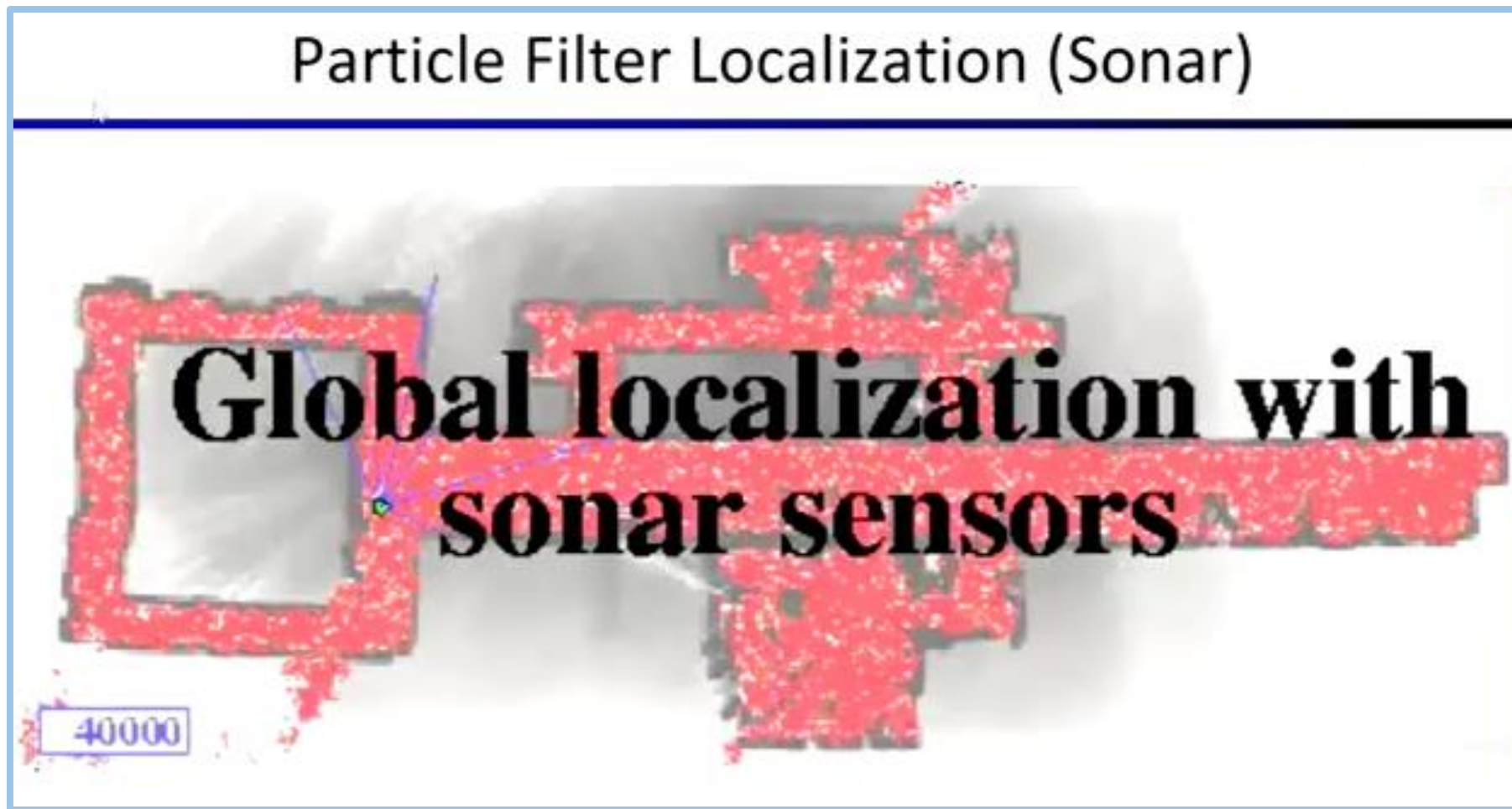


# 粒子滤波 Particle Filtering: 定位

判断哪些样本更有概率是飞机目前的位置

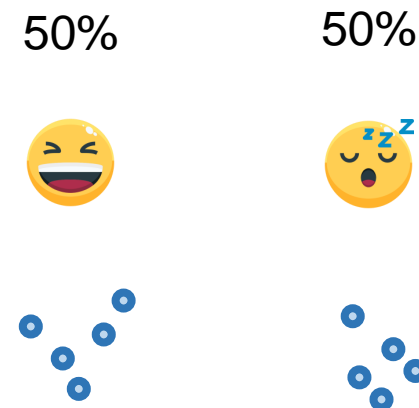


# 粒子滤波应用：机器人定位







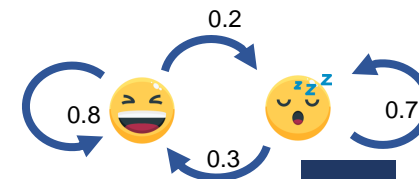
# 粒子滤波 Particle Filtering

- 根据初始分布随机产生N个样本
  - 假设取10个样本
  - 将每个样本储存成一个列表[1, 1, 1, 1, 1, 0, 0, 0, 0, 0]




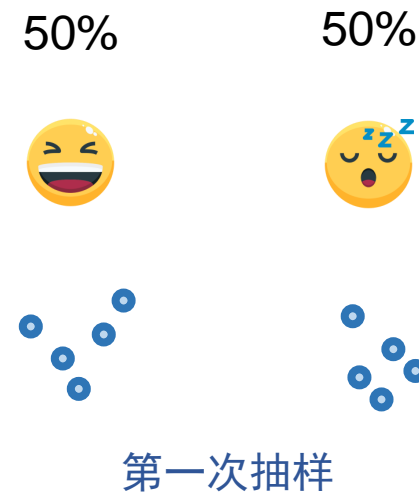
第一次抽样







Observation ( $E_t$ )		
		
State ( $X_t$ )		
	0.2	0.8
	0.9	0.1

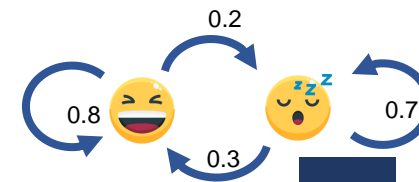


# 粒子滤波 Particle Filtering

- 根据初始分布随机产生N个样本
  - 假设取10个样本
  - 将每个样本储存成一个列表[1, 1, 1, 1, 1, 0, 0, 0, 0, 0]
- 进行观测
  - 假设观测到 
  - 根据观测，计算每个样本的权重，权重为  $P(e/x)$ 
    - 因此权重列表更新为 [0.2, 0.2, 0.2, 0.2, 0.2, 0.9, 0.9, 0.9, 0.9, 0.9]
  - 计算隐变量更新后的概率分布  $\langle \frac{0.2 \times 5}{0.2 \times 5 + 0.9 \times 5}, \frac{0.9 \times 5}{0.2 \times 5 + 0.9 \times 5} \rangle$   
 $= \langle 0.18, 0.82 \rangle$



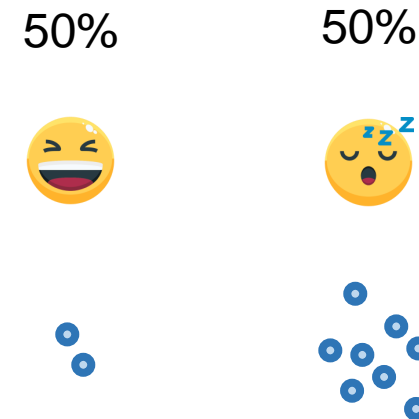
Observation ( $E_t$ )		
State ( $X_t$ )		
	 0.2	 0.8
	 0.9	 0.1











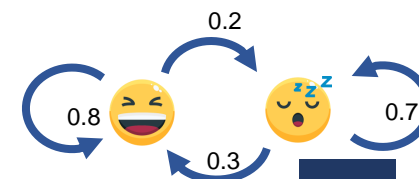
# 粒子滤波 Particle Filtering

- 根据更新过的概率分布  $\langle 0.18, 0.82 \rangle$  随机产生  $N$  个样本
  - 假设取 10 个样本
  - 将每个样本储存成一个列表  $[1, 1, 0, 0, 0, 0, 0, 0, 0, 0]$



第二次抽样

Observation ( $E_t$ )		
State ( $X_t$ )		
	 0.2	 0.8
	 0.9	 0.1



# 粒子滤波 Particle Filtering

- 根据更新过的概率分布 $\langle 0.18, 0.82 \rangle$ 随机产生N个样本
  - 假设取10个样本
  - 将每个样本储存成一个列表 $[1, 1, 0, 0, 0, 0, 0, 0, 0, 0]$
- 时间流转 to 下一时刻
  - 根据转移模型记录样本下一时刻的状态
  - 样本变化如下:  $[1, 1, 1, 1, 0, 0, 0, 0, 0, 0]$

50%

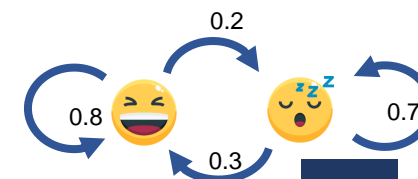


50%




时间流转 to 下一时刻

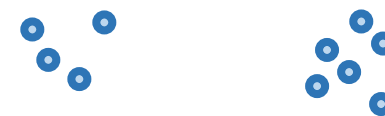
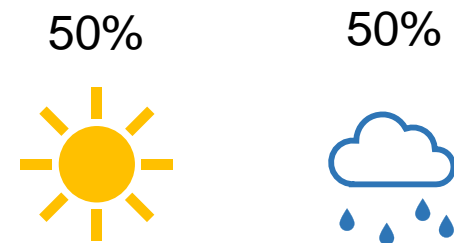
		Observation ( $E_t$ )	
State ( $X_t$ )		0.2	0.8
		0.9	0.1







# 粒子滤波 Particle Filtering

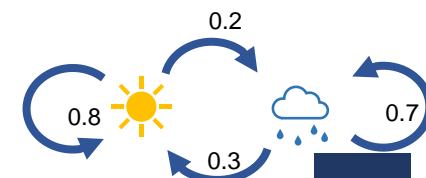
## 重复:

- 根据更新过的概率分布  $\langle 0.18, 0.82 \rangle$  随机产生  $N$  个样本
  - 假设取 10 个样本
  - 将每个样本储存成一个列表  $[1, 1, 0, 0, 0, 0, 0, 0, 0, 0]$
- 时间流转 to 下一时刻
  - 根据转移模型记录样本下一时刻的状态
  - 样本变化如下:  $[1, 1, 1, 1, 0, 0, 0, 0, 0, 0]$
- 进行观测
  - 假设观测到 
  - 根据观测, 计算每个样本的权重, 权重为  $P(e/x)$ 
    - 因此权重列表更新为  $[0.2, 0.2, 0.2, 0.2, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9]$
  - 计算隐变量更新后的概率分布  $\langle \frac{0.2 \times 4}{0.2 \times 5 + 0.9 \times 5}, \frac{0.9 \times 6}{0.2 \times 5 + 0.9 \times 5} \rangle = \langle 0.12, 0.88 \rangle$



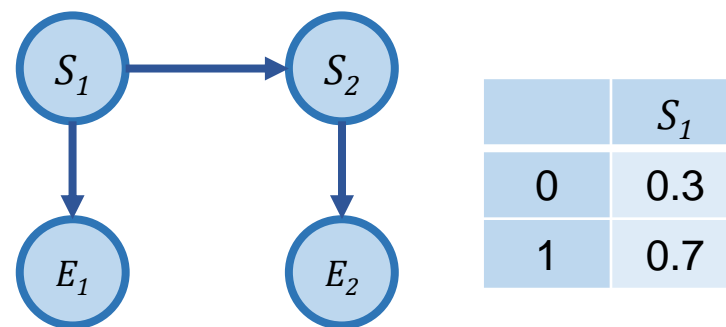
时间流转 to 下一时刻

		Observation ( $E_t$ )	
			
State ( $X_t$ )		0.2	0.8
		0.9	0.1



# 练习 #12

- 初始产生了2个样本[0, 1]
- 用例子滤波推算 $P(S_2|E_1 = a, E_2 = b)$ 
  - 过程中产生的随机数为[0.22, 0.05, 0.33, 0.20]



		$E_t$	
		a	b
$S_t$	0	0.9	0.1
	1	0.5	0.5

		$S_{t+1}$	
		0	1
$S_t$	0	0.4	0.6
	1	0.8	0.2

# 有问题吗？

---

- 请随时举手提问。

