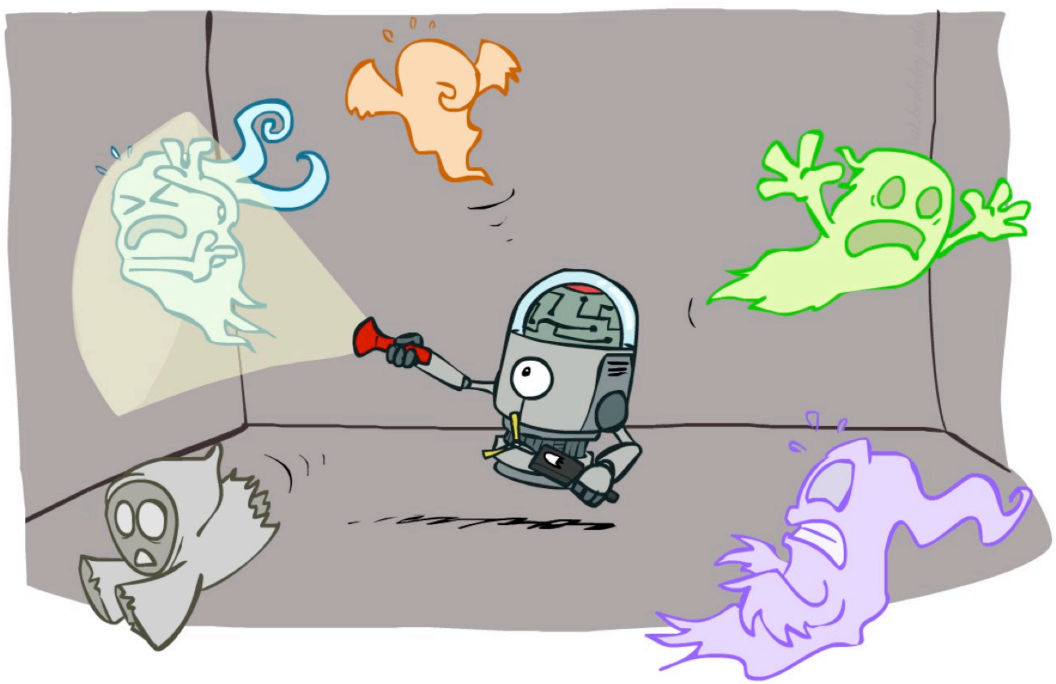
隐式马科夫模型（HMM）， 粒子过滤
(Particle Filters)



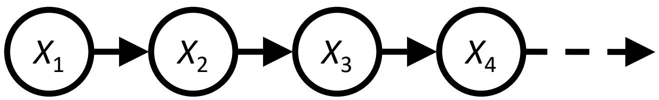
今天内容

- HMMs
  - Particle filters 粒子滤波
  - Demos! 实例展示
  - Most-likely-explanation queries

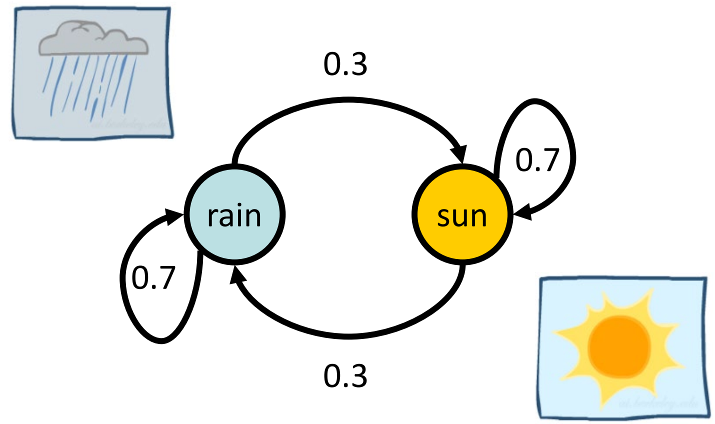
- Applications:
  - Robot localization / mapping 应用：机器人定位和地图绘制
  - Speech recognition (later)

复习: 时间迁移上的推理

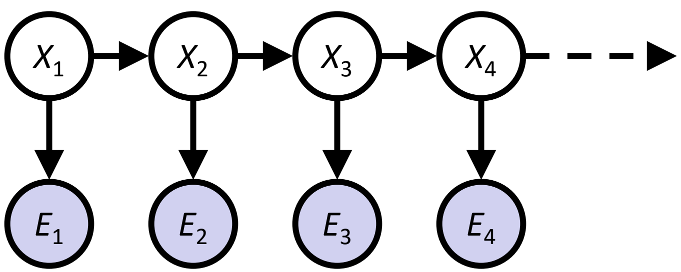
- Markov models 马科夫模型



$P(X_1)$ $P(X|X_{-1})$



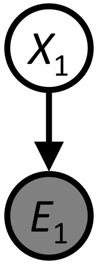
- Hidden Markov models 隐式马科夫模型

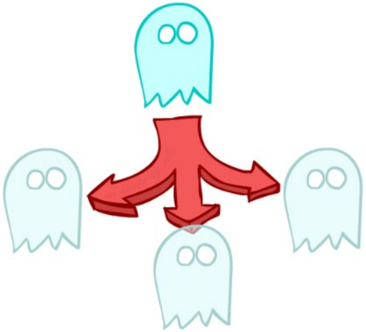


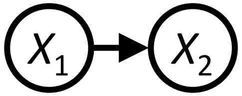
$P(E|X)$

X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

推理: 两种基本情况







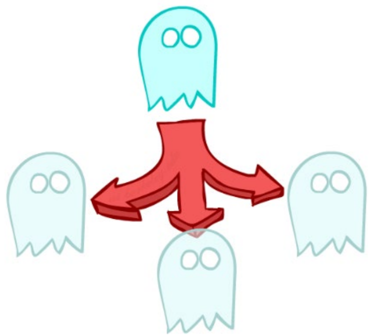
$$P(X_1|e_1)$$

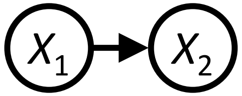
$$\begin{aligned} P(x_1|e_1) &= P(x_1, e_1)/P(e_1) \\ &\propto_{X_1} P(x_1, e_1) \\ &= P(x_1)P(e_1|x_1) \end{aligned}$$

$$P(X_2)$$

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1)P(x_2|x_1) \end{aligned}$$

推理: 基本情况





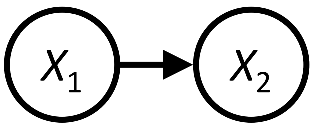
$$P(X_2)$$

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1) P(x_2|x_1) \end{aligned}$$

时间转移

- 给定当前的置信分布 P(X | evidence to date)

$$B(X_t) = P(X_t|e_{1:t})$$



- 然后, 在经过一个时间步长后:

$$\begin{aligned} P(X_{t+1}|e_{1:t}) &= \sum_{x_t} P(X_{t+1}, x_t|e_{1:t}) \\ &= \sum_{x_t} P(X_{t+1}|x_t, e_{1:t}) P(x_t|e_{1:t}) \\ &= \sum_{x_t} P(X_{t+1}|x_t) P(x_t|e_{1:t}) \end{aligned}$$

- Or compactly:

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t) B(x_t)$$

- Basic idea: beliefs get “pushed” through the transitions
  - With the “B” notation, we have to be careful about what time step t the belief is about, and what evidence it includes

举例: 时间迁移

- 随着时间的迁移, 不确定性 “累积”

(Transition model: ghosts usually go clockwise)

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	1.00	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

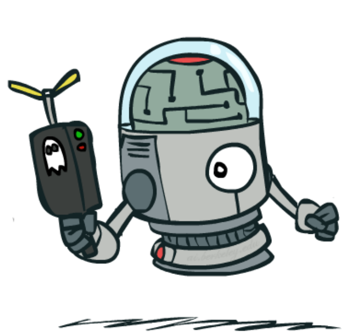
T = 1

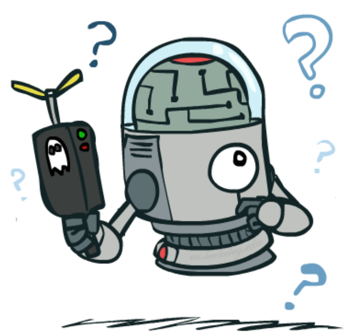
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

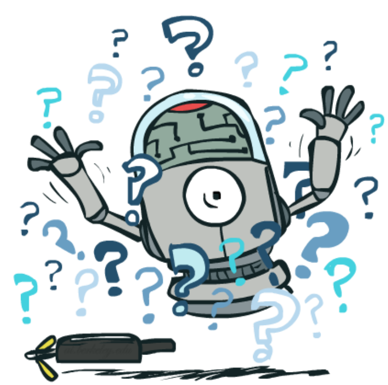
T = 2

0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

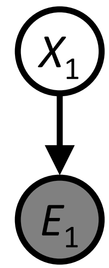
T = 5







推理: 基本情况



$$P(X_1|e_1)$$

$$\begin{aligned} P(x_1|e_1) &= P(x_1, e_1)/P(e_1) \\ &\propto_{X_1} P(x_1, e_1) \\ &= P(x_1)P(e_1|x_1) \end{aligned}$$

给定观察变量下的推理

- 假设给定当前置信分布 P(X | previous evidence):

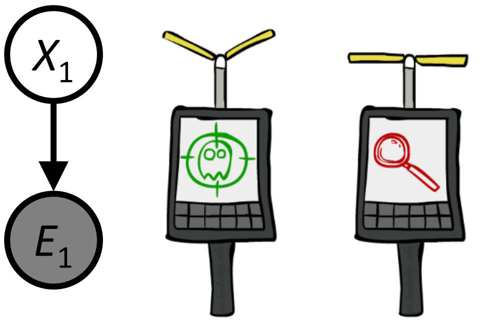
$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- 然后, 当前时刻观察结果获得后:

$$\begin{aligned} P(X_{t+1}|e_{1:t+1}) &= \ P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t}) \\ &\propto_{X_{t+1}} \ P(X_{t+1}, e_{t+1}|e_{1:t}) \\ &= P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t}) \\ &= P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t}) \end{aligned}$$

- 或表示为:

$$B(X_{t+1}) \propto_{X_{t+1}} \ P(e_{t+1}|X_{t+1})B'(X_{t+1})$$



- Basic idea: beliefs “reweighted” by likelihood of evidence
- Unlike passage of time, we have to renormalize

举例: 观察过滤

- 当我们获得观察值后, 置信分布被重新加权修正, 不确定性 “降低”

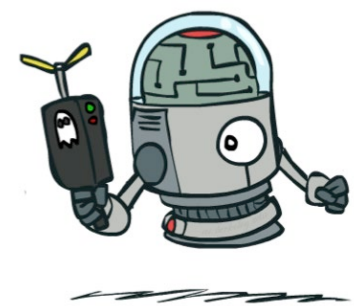
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

Before observation

<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

After observation

$$B(X) \propto P(e|X)B'(X)$$



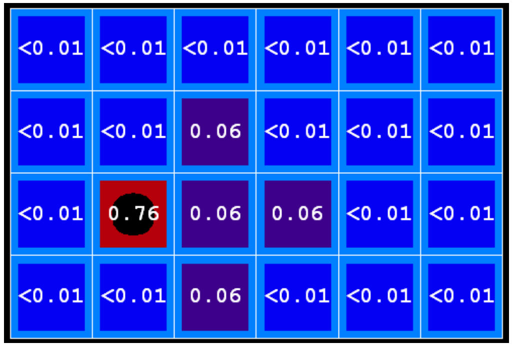
Filtering(过滤法)

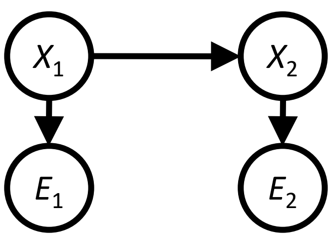
Elapse time: compute P(X_t | $e_{1:t-1}$)

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

Observe: compute P(X_t | $e_{1:t}$)

$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$

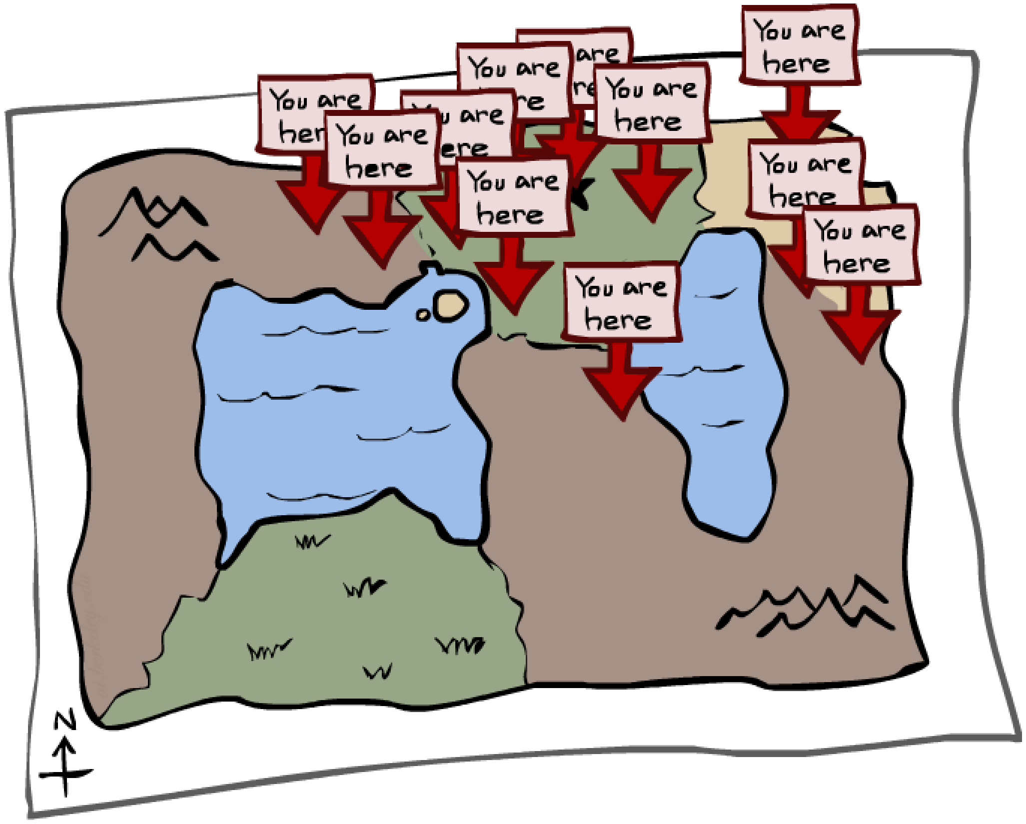




Belief: <P(rain), P(sun)>

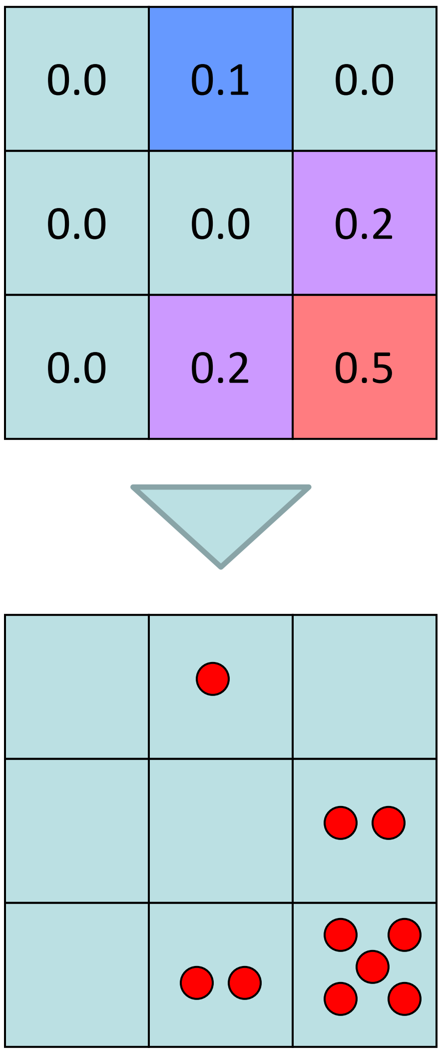
$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

粒子滤波(过滤)
Particle Filtering



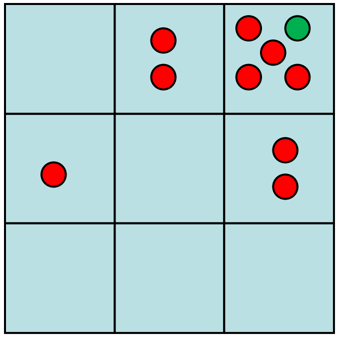
粒子滤波

- 滤波：近似解
- 有时 |X| 太大不能用精确推理计算
  - |X| 也许太大而不能存储
  - 例如， X 是连续的
- 解决办法：近似推理
  - 追踪 X 的样本，不需要追踪其所有的值
  - 样本这里也叫**粒子**
  - 每步计算的时间与样本数量成线性关系
  - 但是：需要的样本数也许会很大
  - 存储的是：一系列的样本，不是状态
- 机器人定位在实践中就是利用这个方法
- 粒子在这里只是样本的一个新的名字



表示：粒子

- P(X) 是由 N 个粒子（样本）表示的
  - 通常，N 远小于 |X|
  - 存储的是一列样本，而不是从状态 X 到其数量(概率值)的一个映射
- P(x) 是由关于 x 的粒子的数量近似计算的
  - 所以许多状态 x 可能是 P(x) = 0!
  - 粒子越多，近似越准确
- 目前为止,所有的粒子的权值均是 1



粒子:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

粒子过滤：时间迁移

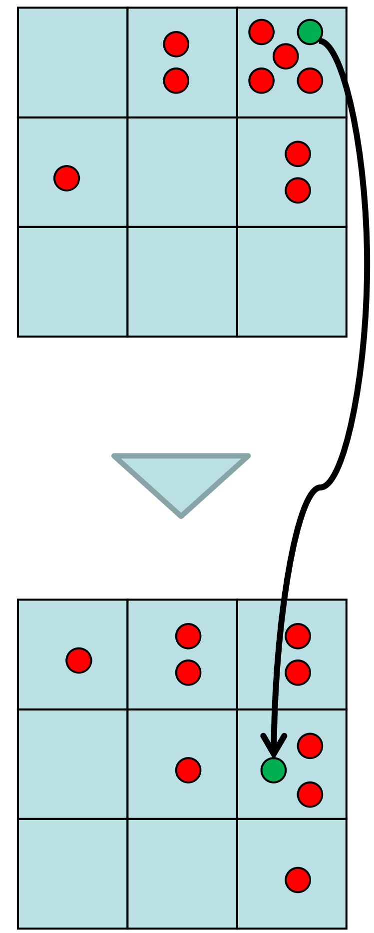
- 每个粒子的移动是通过采样相应的转移模型来完成的，即：

$$x' = \text{sample}(P(X'|x))$$

  - 这一步像是先验采样 – 样本出现的频率反映了相应的转移概率分布
  - 在这个例子里，（幽灵移动顺时针）所有的样本顺时针移动，但也有些向另一个方向移动或呆在原地不动

- 这一步描述了时间上的迁移
  - 如果样本数足够，就会接近真实值，无论是采样之前还是之后（保持与概率分布的一致性）

Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)



Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

粒子过滤：在观察时

- 获取传感观察值后，固定它
- 像是 似然性加权，根据所谓的证据（观察变量的值），给每个样本添加一个权值

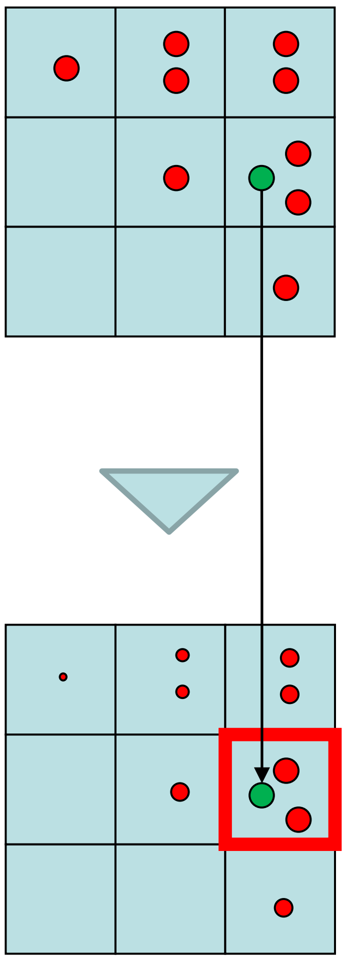
$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

- 像之前一样，这里的概率值加和还不为 1，（实际上现在它们加和后等于概率P(e)的近似值的N倍）

Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4



粒子过滤：重新采样阶段

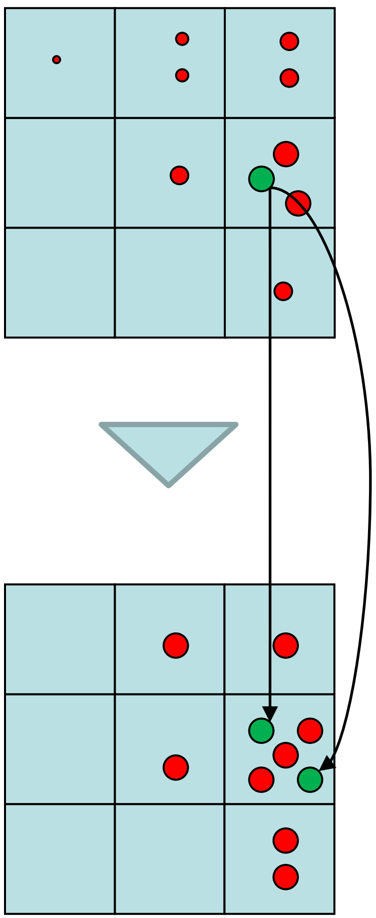
- 暂时不再追踪加权后的样本在时间上的迁移，而是重新进行采样
- *从加权后的当前样本空间分布里*，重新采样 N 次，（换句话说，抽出后放回采样）
- 这相当于重新规整（正规化/归一化）之前加权的概率分布
- 截至已完成在这一时刻的样本分布更新，将开始下一时刻的样本迁移过程

Particles:

(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

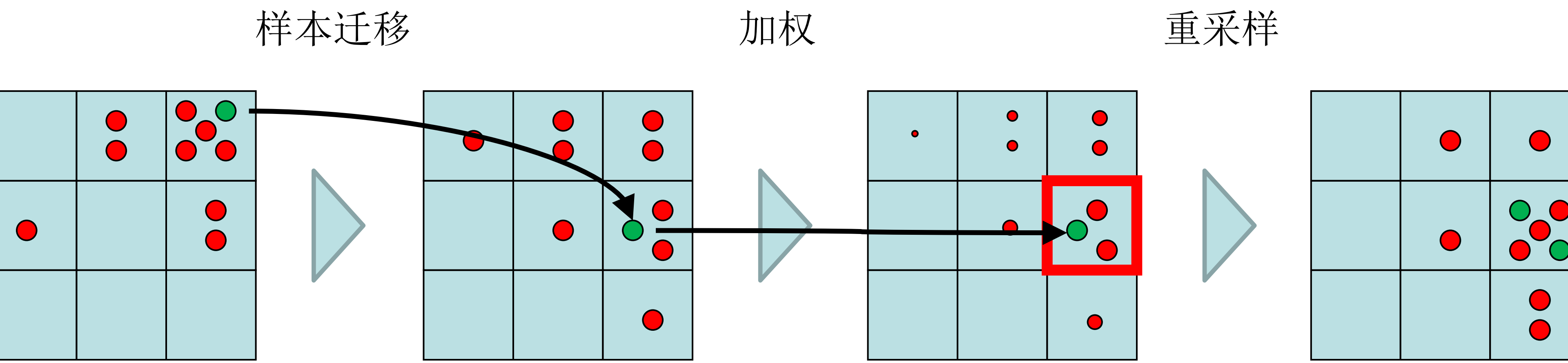
(New)
Particles:

(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(3,2)



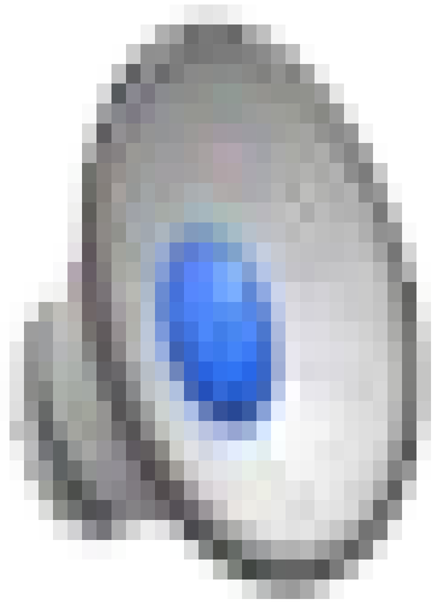
粒子过滤一个时间段里的完整过程

- 粒子: 追踪状态的样本空间分布，而不是显示地(直接地)表示状态的概率分布



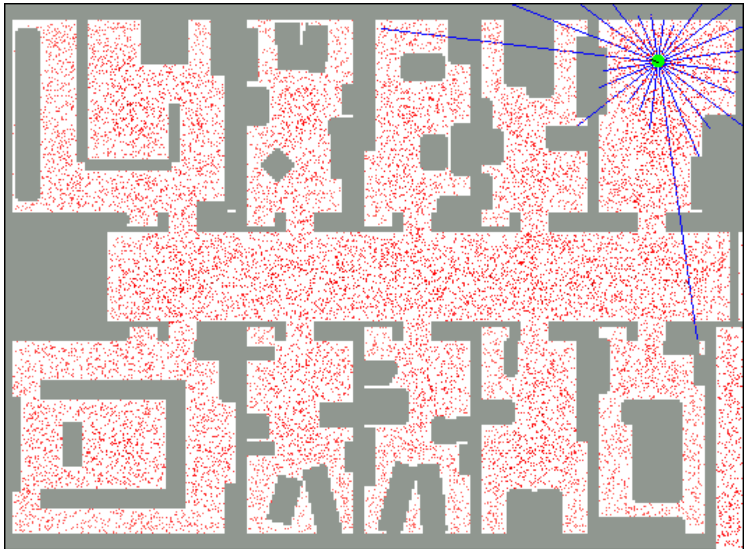
Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

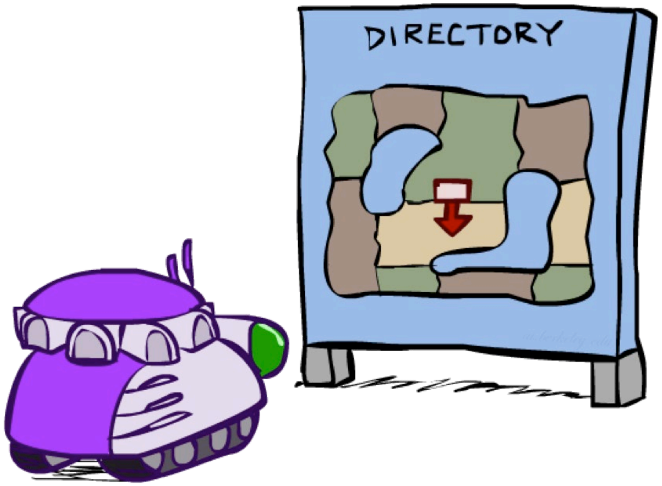
幽灵追踪视频演示
粒子过滤（中等数量的粒子）



机器人定位

- 地图已知，但不知道机器人的位置
- 观察到的值 可以是测距仪器返回的读数相量(多个方向)
- 状态空间和距离读数通常是连续数值（可以用细分的网格来近似)，所以我们无法存储 B(X)
- 粒子过滤是这里应用的一个主要的技术

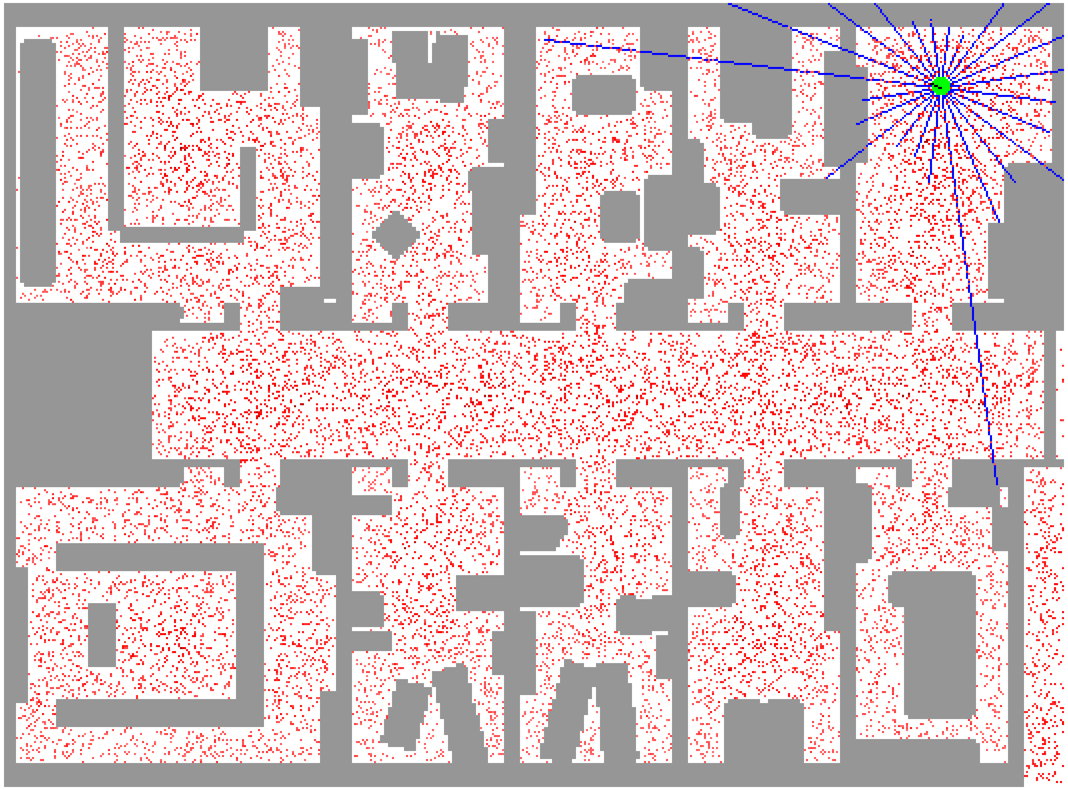




粒子过滤定位(利用声波测距)

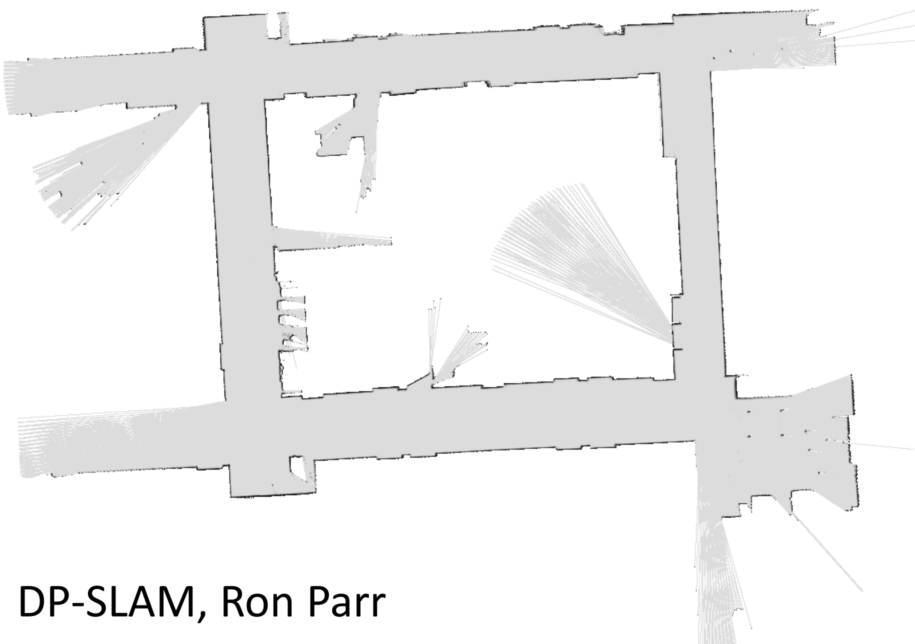
[Dieter Fox, et al.]

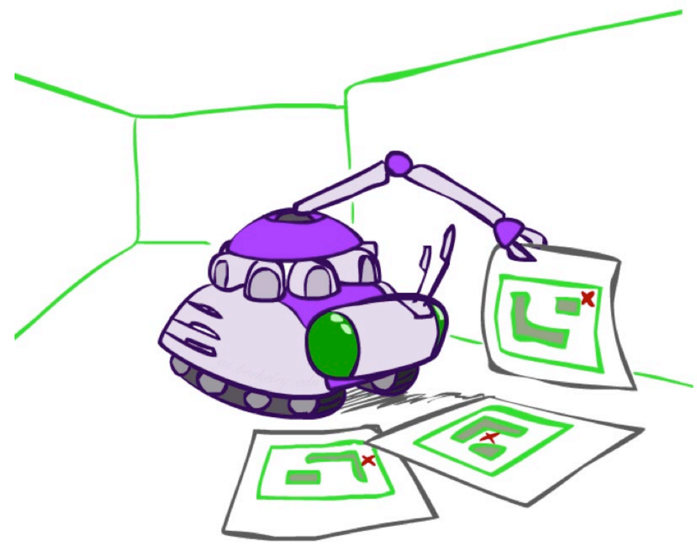
Particle Filter Localization (Laser)



机器人绘地图

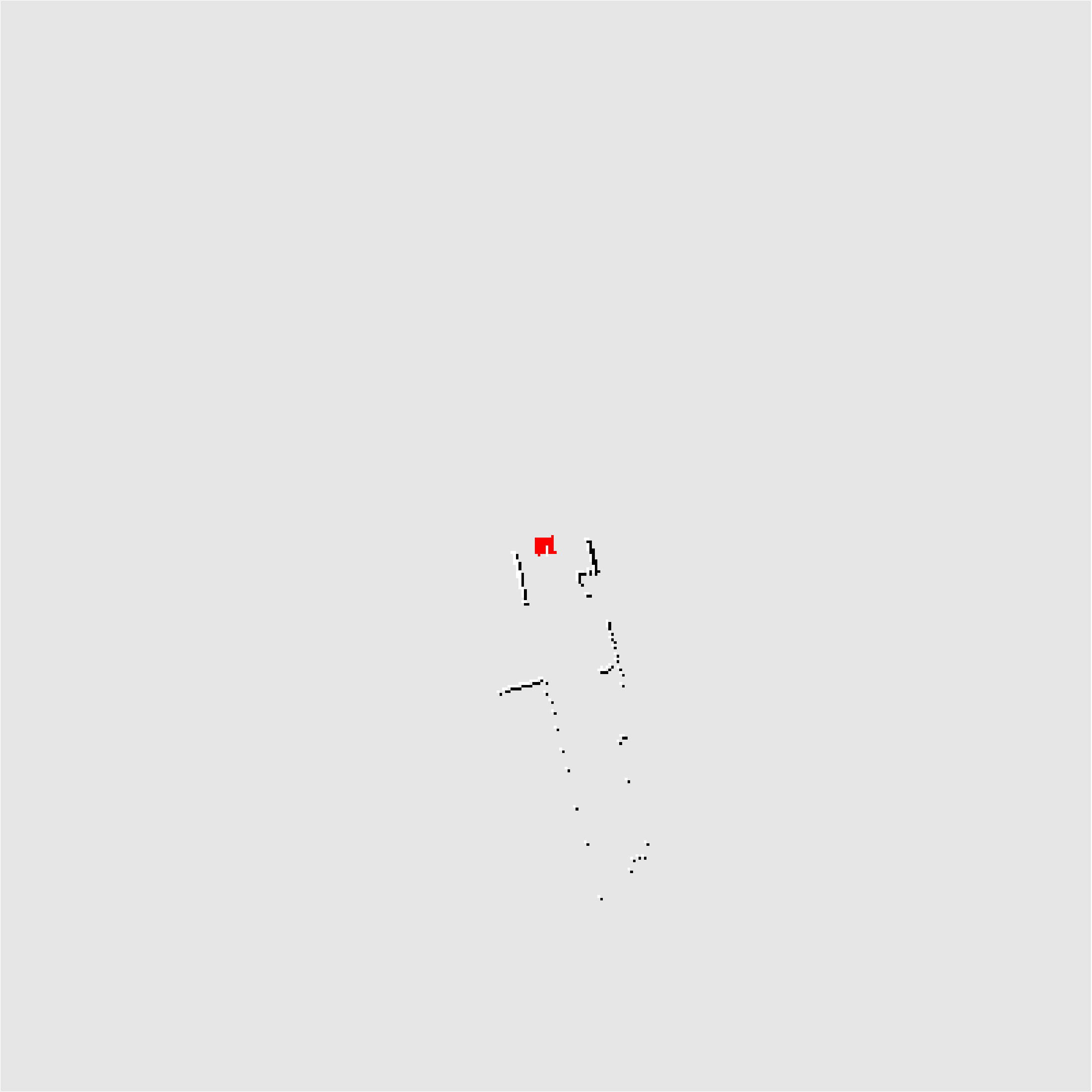
- SLAM: Simultaneous Localization And Mapping（同时定位和绘图）
  - 我们不知道地图和我们的位置
  - 状态由位置和地图构成!
  - 主要技术: 卡尔曼滤波器 Kalman filtering (Gaussian HMMs) 和 基于粒子的方法





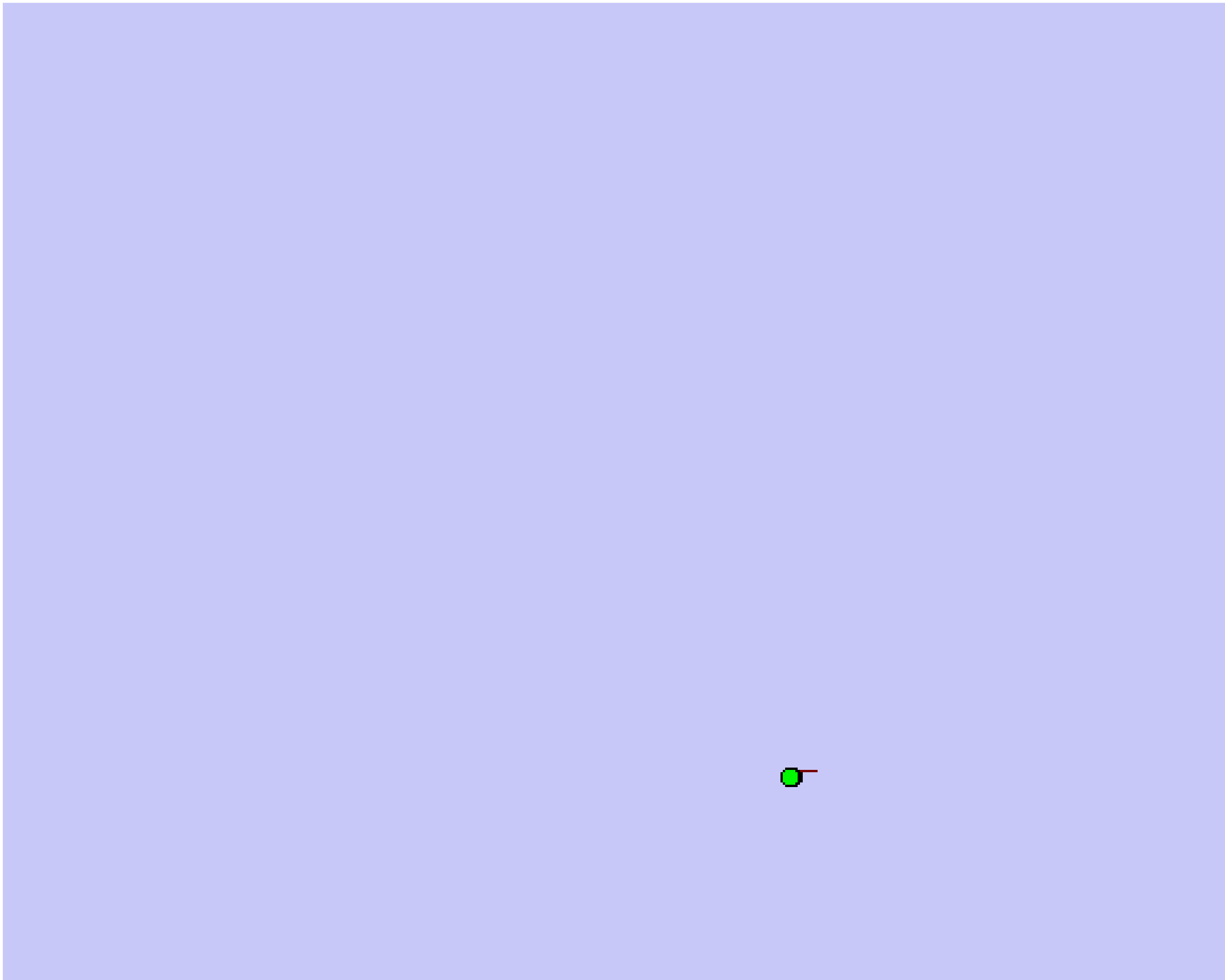
DP-SLAM, Ron Parr

粒子过滤 SLAM – Video 1



[Sebastian Thrun, et al.]

粒子过滤 SLAM – Video 2

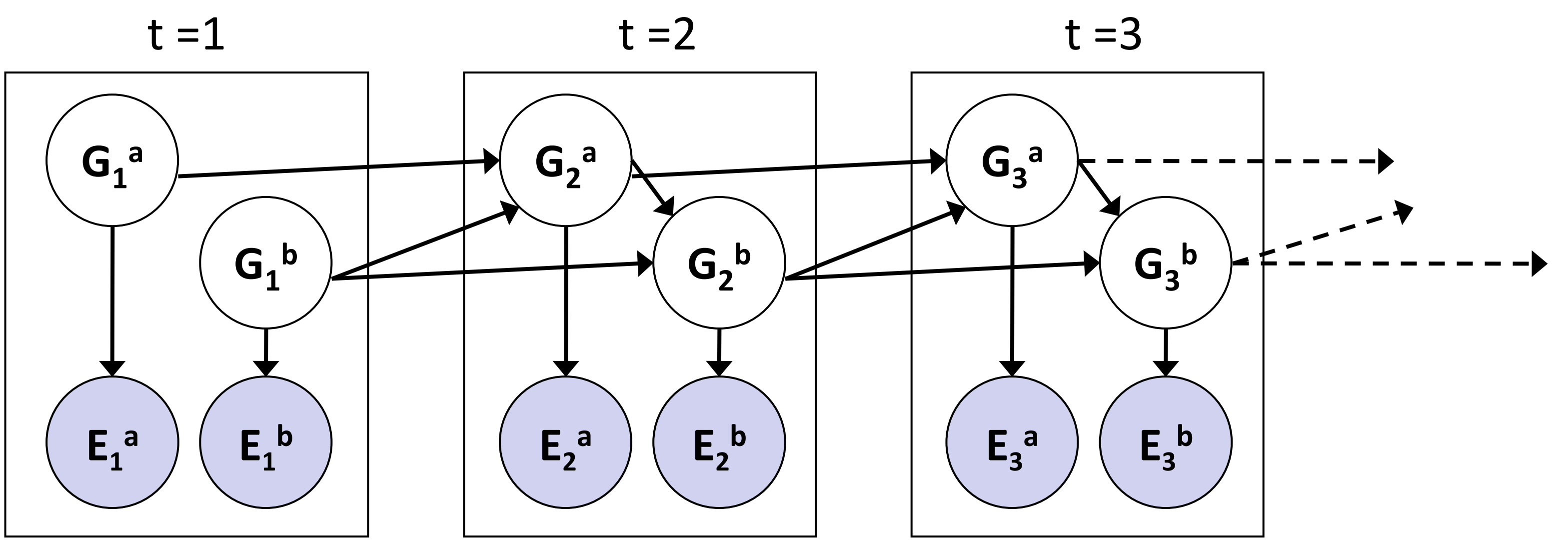


[Dirk Haehnel, et al.]

动态贝叶斯网络
Dynamic Bayes Nets

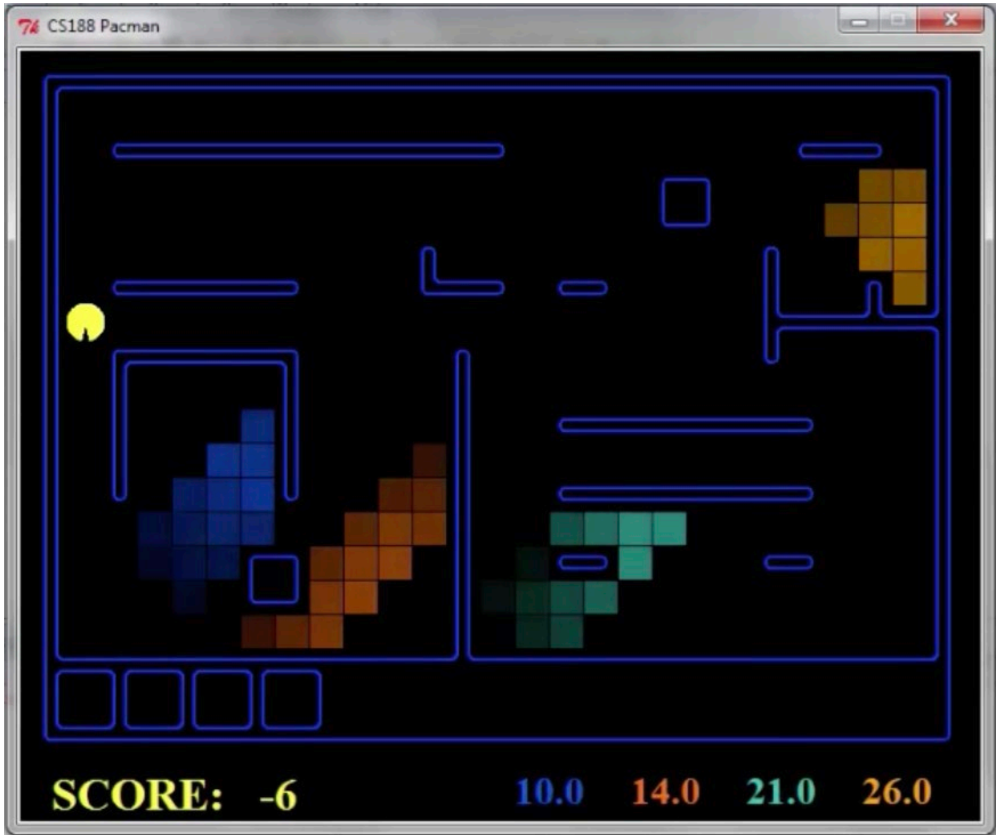
动态贝叶斯网络(DBNs)

- 我们想要随时间的发展追踪多个变量，使用多个可观察变量
- 思想：在每一时刻重复一个固定的贝叶斯网络结构
- 时刻t的变量可能条件基于时刻t-1的变量

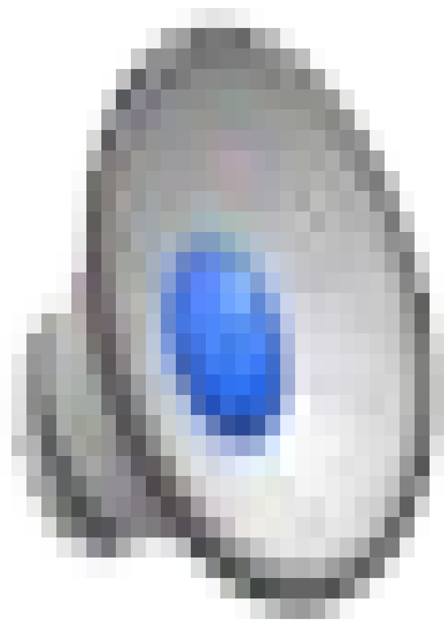


- 动态贝叶斯网络是一种广义化的隐式马科夫模型(HMMs)

Pacman – Sonar (P4)

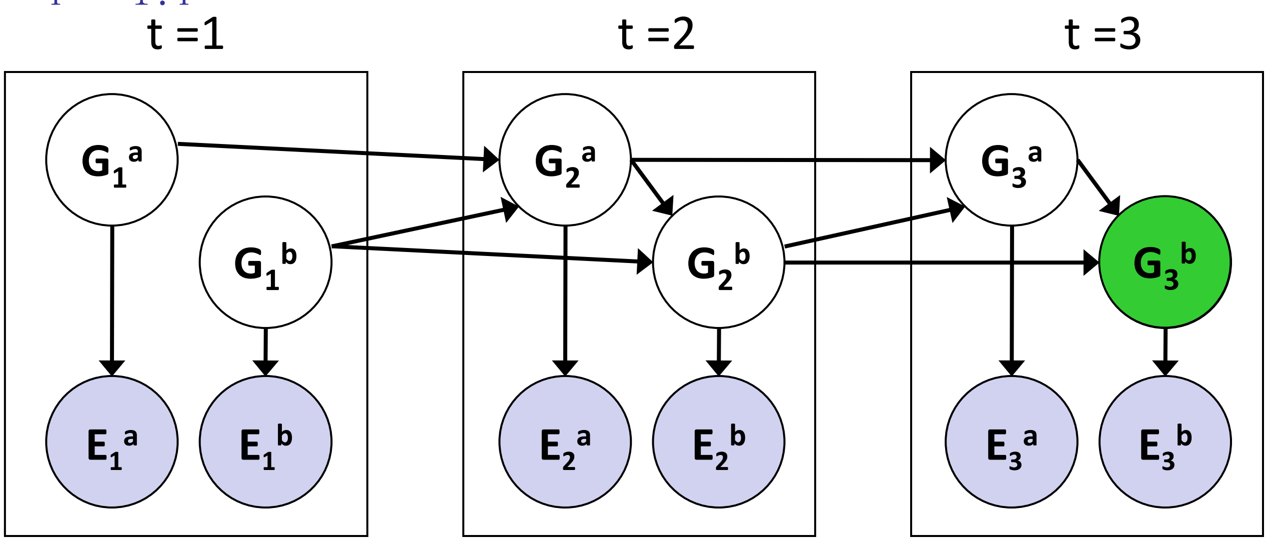


Pacman 追踪幽灵视频演示 Ghost DBN 模型



DBNs中的精确推理

- 变量消除法同样应用于动态贝叶斯网络
- 步骤：“展开”T 个时刻的网络结构，然后消除变量，直到计算出 $P(X_T|e_{1:T})$



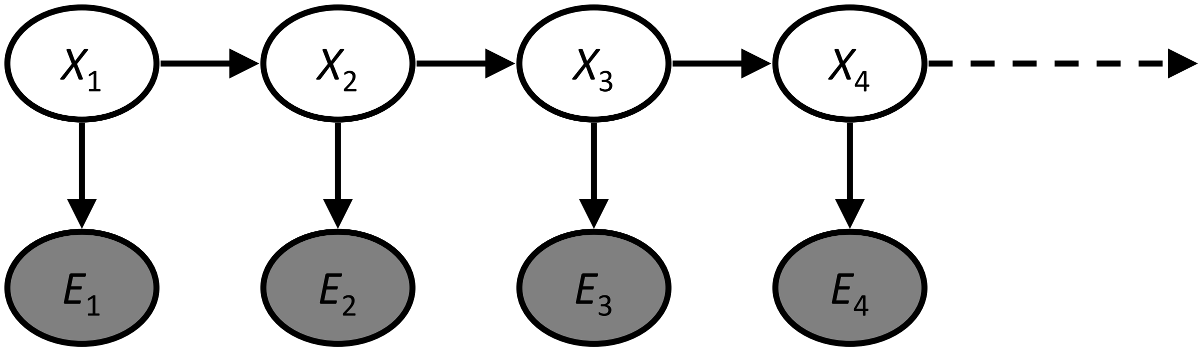
- 在线置信度更新：Eliminate all variables from the previous time step; store factors for current time only

DBN中的粒子过滤方法

- 一个粒子是一个时刻的一个*完整*样本
- **初始化**：产生t=1时刻贝叶斯网络的N个先验样本
  - 粒子举例: $G_1^a = (3,3)$ $G_1^b = (5,3)$
- **时间迁移**：对于每个粒子采样一个其后继样本(粒子)
  - 比如: $G_2^a = (2,3)$ $G_2^b = (6,3)$
- **观察**：权重每个 *完整的* 样本，通过观察值条件于样本的似然值
  - 似然权值: $P(E_1^a \mid G_1^a) * P(E_1^b \mid G_1^b)$
- **重新采样**：在加权后的分布上，重新采集先验样本(粒子)

最可能的解释
Most Likely Explanation

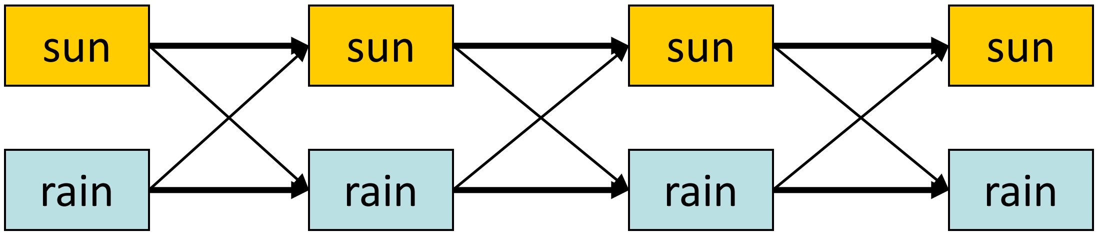
HMMs: MLE 查询



- HMMs 的定义
  - 状态 X
  - 观察变量 E
  - 初始分布: $P(X_1)$
  - 转移模型: $P(X_t|X_{t-1})$
  - 输出或观察模型: $P(E|X)$
- 新的查询: 最有可能的解释: $\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$
- 方法: 维特比算法 Viterbi algorithm

状态网格 State Trellis

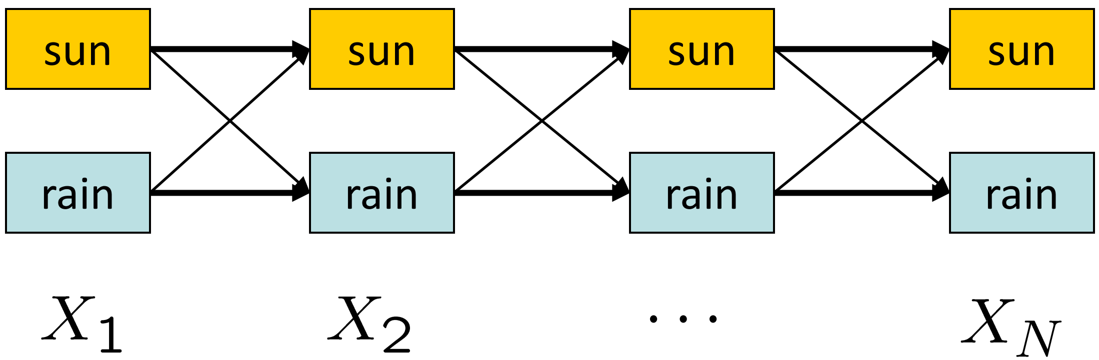
- 状态网格：状态及状态随时间过渡图



X_1 X_2 $\cdots$ X_N

- 每条边代表某个过渡转移 $x_{t-1} \rightarrow x_t$
- 每条边有一个权值 $P(x_t|x_{t-1})P(e_t|x_t)$
- 每条路径是一序列的状态组成
- 一条路径上所有权值的乘积是这个状态序列及相应观察序列的联合概率
- 前向算法计算路径之和，Viterbi计算最佳(最可能解释的)路径

前向 与 维特比算法



前向算法(求和)

$$f_t[x_t] = P(x_t, e_{1:t})$$

$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

维特比算法(求最大)

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$