北京交通大学

BEIJING JIAOTONG UNIVERSITY

Artificial Intelligence a Modern Approach

CHAPTER 14：Probabilistic Reasoning over time

第 14 章：时间上的概率推理

姓　　名：包凯丰

学　　号：19211290

专　　业：通信工程

班　　级：通信 1909

授课教师：陈一帅

目录

1. Abstract 摘要

我读的是 Artificial Intelligence A Modern Approach 的第 14 章，Probabilistic Reasoning over time，时间上的概率推理。在这一章中，每个时间点对世界状态的每一个方面用一个变量来表示，通过这种方式对变化的世界进行建模。在时间和不确定性里包含**转移模型**和**传感器模型**。在后边几节有三种特殊类型的模型：**隐马尔科夫模型（Hidden Markov Models）**、**卡尔曼滤波器（Kalman Filter）**以及**动态贝叶斯网络（Dynamic Bayesian Networks）**，前两个模型是动态贝叶斯网络的特殊情况。下边我将具体分享上述内容。

2. Time and Uncertainty 时间与不确定性

这里分为**静态**和**动态**两种，静态就是在整个过程中状态是不随时间变化的，例如汽车修理过程中，一个故障是一直保持到修理结束的；而动态就是整个过程中状态是随着时间变化的，例如跟着机器人的位置、治疗病人时病人的医学状态等等。为了根据过去的数据，评估当前的状态，预测未来的变化，要对这些变化进行建模。

2.1 States and observations 状态和观察

首先我们将问题转换成数学语言，我们考虑离散数学模型，将世界看作是一系列的时间片（time slice），就像动画片一样，每一个时间片都包含了一组随机变量，其中一部分是可以观察到的，另一部分是不可观察的。我们用E_t表示 t 时刻可观察到变量集（Evidence），用X_t表示 t 时刻不可观察的状态变量集。时刻 t 的观察结果为$E_t = e_t$。

比如我们考虑课上讲到的例子：门口的警卫想知道今天是否会下雨，只能通过观察高管进来时是否带着雨伞来判断，所以此时的$X_t = Rain_t$，$E_t = Umbrella_t$。

我们将世界的状态看作是一系列的时间片，而具体时间片之间的间隔需要随着问题的变化而灵活变化。而这一章节中，默认时间间隔是固定的，所以就可以使用整数下标来标注时间，将状态序列从$t=0$时刻开始，比如X_0，X_1，…，假设证据变量从$t=1$开始，比如E_1，E_2，…，这样表示。

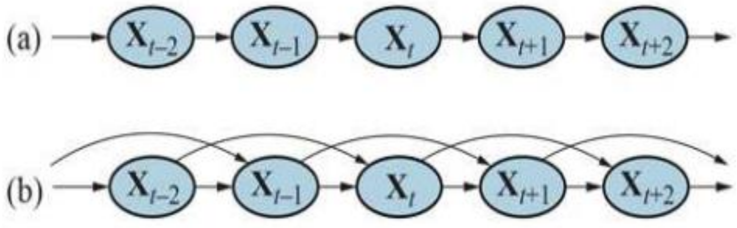
2.2 Transition and sensor models 转移模型和传感器模型

一旦我们知道了给定问题的状态变量和证据变量的集合，下一步的工作便是要制定世界如何演变（**转移模型**）以及证据变量如何得到它们的取值（**传感器模型**）。

转移模型：在给定过去的状态变量值之后，确定最新状态变量的概率分布$P(X_t|X_{0:t-1})$。

转移模型的简化：这里采用两个假设：马尔可夫假设和稳态过程假设。

马尔可夫假设：考虑到随着时间 t 的增长，集合$X_{0:t-1}$的大小没有限制，所以我们采用马尔可夫假设——当前状态只取决于有限的过去状态，与更早的状态无关，这样的过程称为马尔可夫过程，一阶 Markov 过程如下：$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$，二阶 Markov 过程如下：$P(X_t|X_{0:t-1}) = P(X_t|X_{t-2}, X_{t-1})$。其对应的贝叶斯网络结构如下：



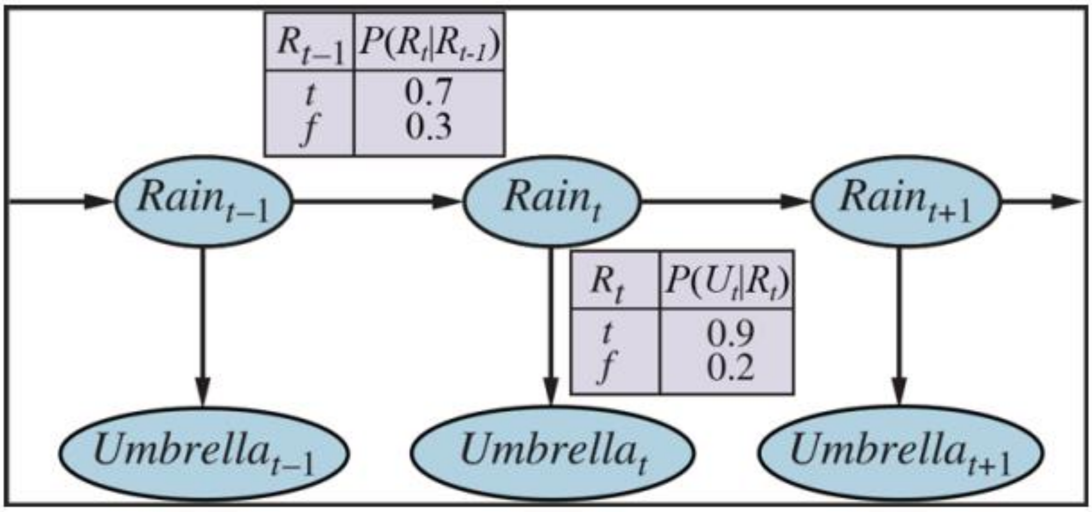
稳态过程假设：即使我们有了马尔可夫假设，但是时间 t 有无穷多个取值，我们是否需要为每个时间步骤确定一个不同的分布呢？对这个问题，假设世界状态的变化是由一个稳态过程引起的——变化的过程是由本身不随时间变化的规律支配的。于是课上的雨伞模型中，下雨的条件概率$P(Rain_t|Rain_{t-1})$对于所有的时间片 t 都是相同的。

传感器模型：通过之前的变量以及当前的状态变量，来确定证据变量的概率分布$P(E_t|X_{0:t},E_{0:t-1})$。也叫**观察模型**。

传感器模型的简化：如果一个当前的状态足够稳健，那么对于产生当前的传感器值应该足够了，所以做如下的**传感器马尔可夫假设**：　$P(E_t|X_{0:t},E_{0:t-1})=P(E_t|X_t)$。

下边我只要知道事情是如何开始的，就可以一步一步得到所有状态，即要知道 t=0 时刻的状态$P(X_0)$。

对于下雨和雨伞模型，如果整体结构采用一阶马尔可夫假设和传感器马尔可夫假设，在初始状态已知的情况下，得到如下结构和联合概率分布：



Bayesian network structure and conditional distributions describing the umbrella world. The transition model is $\mathbf{P}(Rain_t|Rain_{t-1})$ and the sensor model is $\mathbf{P}(Umbrella_t|Rain_t)$.

对于任何时间 t 的联合概率分布如下：

$$P(X_{0:t},E_{1:t})=P(X_0)\prod_{i=1}^{t}{\color{blue}P(X_i|X_{i-1})}{\color{red}P(E_i|X_i)}$$

其中蓝色为转移模型，红色为传感器模型。

2.3 My idea 我的想法

这一节的整体思路来看，其实就是一个数学建模过程，首先将问题转化为数学问题，再假设符号，建立关系。之后在考虑所有因素的情况下，慢慢简化模型，得到最后的结果。我认为最重要的两点是：一是要寻找一个强有力的关系，二就是要合理的简化。这才是解决问题的关键所在。

如何得到更加精确的模型：最简单的方法是提高马尔可夫过程的阶数，另外可以考虑**多传感器模型**，即不止考虑雨伞这一个因素，还可以考虑人们的穿衣类型、哪年哪月哪号以及是否是雨季等因素。

3. Inference in Temporal Models 时序模型的推理

建立了一般时序模型的结构之后，可以将要解决的任务列为以下几个部分：

1、**滤波（filtering）**——即给定目前为止的所有证据参数，计算当前状态的后验概率分布，滤波也称为**状态估计（state estimation）**。即计算$P(X_t|e_{1:t})$。在课上雨伞的例子中，给定目前位置雨伞携带者的过去的所有观察数据，计算今天下雨的概率。

2、**预测（prediction）**——给定目前为止的所有证据，计算未来状态的后验概率分布。也就是计算$P(X_{t+k}|e_{1:t})$，其中 k>0。在课上雨伞的例子中，给定目前为止对雨伞携带者的过去的所有观察数据，计算今天开始以后 k 天下雨的概率。

3、**平滑（smoothing）**——给定目前为止的所有证据，计算过去某一状态的后验概率。也就是计算$P(X_k|e_{1:t})$，其中 0≤k<t。在课上雨伞的例子中，给定目前为止对雨伞携带者的过去的所有观察数据，计算某一天下雨的概率，平滑结合之前之后的证据，可以为该状态提供一个比当时计算得到的结果更好的估计。

4、**最可能解释（most likely explanation）**——在给定观察序列，希望找到最可能生成这些观察结果的状态序列。也就是计算$arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$。在课上雨伞的例子中，如果前三天每天都出现雨伞，但第四天没有出现，那么最可能的解释就是前三天下了雨，第四天没有下。

5、**学习（learning）**——如果还不知道转移模型和传感器模型，则可以从观察中学习。整个学习算法是期望最大化算法（Expectation-Maximization：EM 算法）的一个特例，这个在后续其他章节有讲解。这一章主要讲述上边的四种任务。

3.1 Filtering and prediction 滤波和预测

滤波：给定知道时刻 t 的滤波结果，Agent 需要根据新的证据e_{t+1}来计算时刻 t+1 的滤波结果。也就是存在某个函数 f 满足：

$$P(X_{t+1}|e_{1:t+1})=f(e_{t+1},P(X_t|e_{1:t}))$$

这个过程也称为**递归估计（recursive estimation）**。我们可以将计算分解成两部分：首先，将当前的状态分布由时刻 t 向前投影到时刻 t+1；然后，通过新的证据$e_{1:t}$进行矫正（更新）。如果重排公式，这两步的过程能够很容易得到：

$$\begin{aligned}P(X_{t+1}|e_{1:t+1})&=P(X_{t+1}|e_{1:t},e_{t+1}) && \text{(证据分解)}\\&=\alpha P(e_{t+1}|X_{t+1},e_{1:t})P(X_{t+1}|e_{1:t}) && \text{(使用贝叶斯规则)}\\&=\alpha P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t}) && \text{(使用传感器马尔可夫假设)}\end{aligned}$$

这里涉及到的α均表示一个归一化常数以保证所有概率的和为 1。式子中$P(X_{t+1}|e_{1:t})$表示对下一个状态的单步预测，而第一项$P(e_{t+1}|X_{t+1})$可以从传感器模型中直接得到。现在我们就可以通过当前状态X_t预测下一个状态：

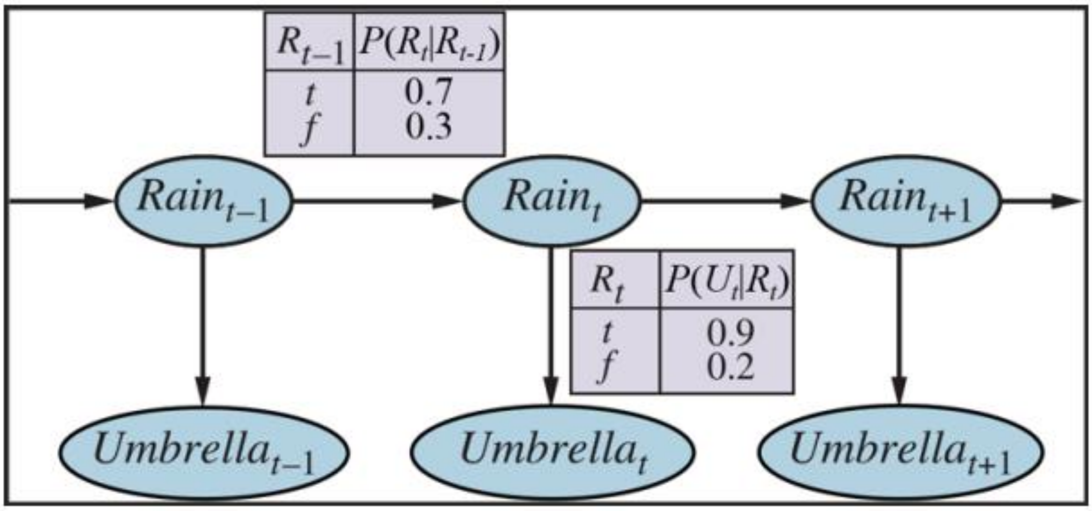
$$\begin{aligned}P(X_{t+1}|e_{1:t+1})&=\alpha P(e_{t+1}|X_{t+1})\sum_{x_t}P(X_{t+1}|x_t,e_{1:t})P(x_t|e_{1:t})\\&=\alpha P(e_{t+1}|X_{t+1})\sum_{x_t}{\color{blue}P(X_{t+1}|x_t)}{\color{red}P(x_t|e_{1:t})} \quad (Markov\text{假设})\end{aligned}$$

其中蓝色的一项为转移模型，红色的一项为当前状态分布。因此我们就得到了递归公式。我们可以认为滤波估计$P(X_t|e_{1:t})$是沿着序列向前传播的“消息”$f_{1:t}$，他在每次转移时得到修正，并根据每个新的观察进行更新。这个过程时：

$$f_{1:t+1}=\alpha FORWARD(f_{1:t},e_{t+1})$$

其中 FORWARD 即为上述的公式描述的更新过程，开始时$f_{1:0}=P(X_0)$。

下边以上述的雨伞模型为例，说明这个两部滤波过程，按照如下方式计算$P(R_2|u_{1:2})$：结构图如下：



Bayesian network structure and conditional distributions describing the umbrella world. The transition model is $\mathbf{P}(Rain_t|Rain_{t-1})$ and the sensor model is $\mathbf{P}(Umbrella_t|Rain_t)$.

第 0 天：观察还未开始，只有警卫的先验经验，假设$P(R_0)=\langle 0.5,0.5\rangle$

第 1 天：出现了雨伞，所以$U_1=true$。从 t=0 到 t=1 的预测结果为：

$$P(R_1)=\sum_{r_0}P(R_1|r_0)P(r_0)=\langle 0.7,0.3\rangle\times 0.5+\langle 0.3,0.7\rangle\times 0.5=\langle 0.5,0.5\rangle$$

然后更新步骤用 t=1 时刻的证据的概率相乘并规范化，得到更新结果，如下：

$$P(R_1|u_1)=\alpha P(u_1|R_1)P(R_1)=\alpha\langle 0.9,0.2\rangle\langle 0.5,0.5\rangle=\alpha\langle 0.45,0.1\rangle\approx\langle 0.818,0.182\rangle$$

第 2 天：又出现了雨伞，因此$U_2=true$。由 t=1 到 t=2 的预测结果为：

$$P(R_2|u_1)=\sum_{r_1}P(R_2|r_1)P(r_1|u_1)=\langle 0.7,0.3\rangle\times 0.818+\langle 0.3,0.7\rangle\times 0.182\approx\langle 0.672,0.373\rangle$$

根据 t=2 时刻的证据进行更新，得到：

$$P(R_2|u_1,u_2)=\alpha P(u_2|R_2)P(R_2|u_1)=\alpha\langle 0.9,0.2\rangle\langle 0.627,0.373\rangle$$
$$=\alpha\langle 0.565,0.075\rangle\approx\langle 0.883,0.117\rangle$$

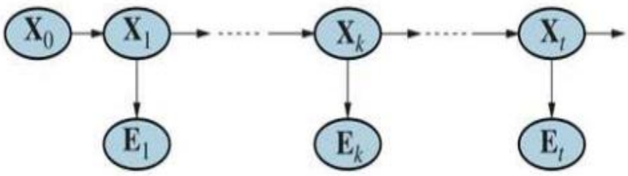
我们通过数据可以知道，下雨的概率从第 1 天到第 2 天提高了。

预测：预测就是没有增加新证据条件下的滤波。所以预测可以看成是一个简化的滤波，比如我们根据$t+k$时刻的预测能够很容易地推导出对$t+k+1$的状态预测，递归过程如下，且仅涉及转移模型：

$$P(X_{t+k+1}|e_{1:t})=\sum_{x_{t+k}}P(X_{t+k+1}|x_{t+k})P(x_{t+k}|e_{1:t})$$

3.2 Smoothing 平滑

平滑：给定直到现在的已知证据，来计算过去的状态的分布的过程；也就是计算$P(X_k|e_{1:t})$，其中 0≤k<t。



对于这个的计算我们分成两部分：

$$\begin{aligned} P(X_k|e_{1:t}) &= P(X_k|e_{1:k}, e_{k+1:t}) \\ &= \alpha P(X_k|e_{1:k}) P(e_{k+1:t}|X_k, e_{1:t}) \qquad \text{(使用贝叶斯规则)} \\ &= \alpha P(X_k|e_{1:k}) P(e_{k+1:t}|X_k) \qquad \text{(使用条件独立性)} \\ &= \alpha f_{1:k} \times b_{k+1:t} \end{aligned}$$

其中“×”表示向量的逐点相乘。$f_{1:k}$：类似于前边提到的 FORWRAD，这里再定义了“后向 BACKWARD”消息$b_{k+1:t} = P(e_{k+1:t}|X_k)$。前向的消息可以通过时刻 1 到时刻 k 的前向滤波过程计算，即由 3.1 的公式进行计算。而后向的消息$b_{k+1:t}$可以通过一个从 t 开始向后运行的递归过程来计算：

$$\begin{aligned} P(e_{k+1:t}|X_k) &= \sum_{x_{k+1}} P(e_{k+1:t}|X_k, x_{k+1}) P(x_{k+1}|X_k) \qquad \text{(将} X_{k+1} \text{条件化)} \\ &= \sum_{x_{k+1}} P(e_{k+1:t}|x_{k+1}) P(x_{k+1}|X_k) \qquad \text{(根据条件独立性)} \\ &= \sum_{x_{k+1}} P(e_{k+1}, e_{k+2:t}|x_{k+1}) P(x_{k+1}|X_k) \\ &= \sum_{x_{k+1}} P(e_{k+1}|x_{k+1}) P(e_{k+2:t}|x_{k+1}) P(x_{k+1}|X_k) \end{aligned}$$

其中红色的一项为传感器模型，蓝色的一项是转移模型，剩下一项是递归模型。使用消息符号，有$b_{k+1:t} = BACKWARD(b_{k+2:t}, e_{k+1})$，其中 BACKWARD 即为上述公式。其中后向的消息的初始值为$b_{t+1:t} = P(e_{t+1:t}|X_t) = P(|X_t) = 1$，其中 1 表示由 1 组成的向量，因为$e_{t+1:t}$是一个空序列，概率为 1。

下边继续以上述的雨伞模型为例，给定第 1 天和第 2 天都观察到雨伞，要计算 k=1 时下雨的概率的平滑估计。根据公式有：

$$P(R_1|u_1, u_2) = \alpha P(R_1|u_1) P(u_2|R_1)$$

由前边描述的前向滤波过程知道，第一项$P(R_1|u_1) = \langle 0.818, 0.182 \rangle$。通过应用后向递归过程可以计算出第二项：

$$P(u_2|R_1) = \sum_{r_2} P(u_2|r_2) P(|r_2) P(r_2|R_1)$$

$$= (0.9 \times 1 \times \langle 0.7, 0.3 \rangle) + (0.2 \times 1 \times \langle 0.3, 0.7 \rangle) = \langle 0.69, 0.41 \rangle$$

所以可以求得第 1 天下雨的平滑估计为：

$$P(R_1|u_1, u_2) = \alpha P(R_1|u_1) P(u_2|R_1)$$

$$= \alpha \langle 0.818, 0.182 \rangle \times \langle 0.69, 0.41 \rangle \approx \langle 0.883, 0.117 \rangle$$

而这个值是高于滤波估计的 0.818 这个值的，这是因为第 2 天出现雨伞的证据使第 2 天下雨的可能性增大了；进一步，由于下雨天气倾向于持续，这又使得第 1 天下雨的可能性也增大了。

前向-后向算法（forward-backward algorithm）这其实就是我们上述提到的方法，即我们可以重复使用前向滤波阶段的结果。线性时间 O(t)复杂度的算法的关键是记录对整个序列进行前向滤波的结果。然后我们在时刻 t 到 1 运行后向递归，在每个步骤 k，根据已经计算出来的后向的消息$b_{k+1:t}$和所储存的前向的消息$f_{1:k}$计算平滑估计。算法的伪代码如下：

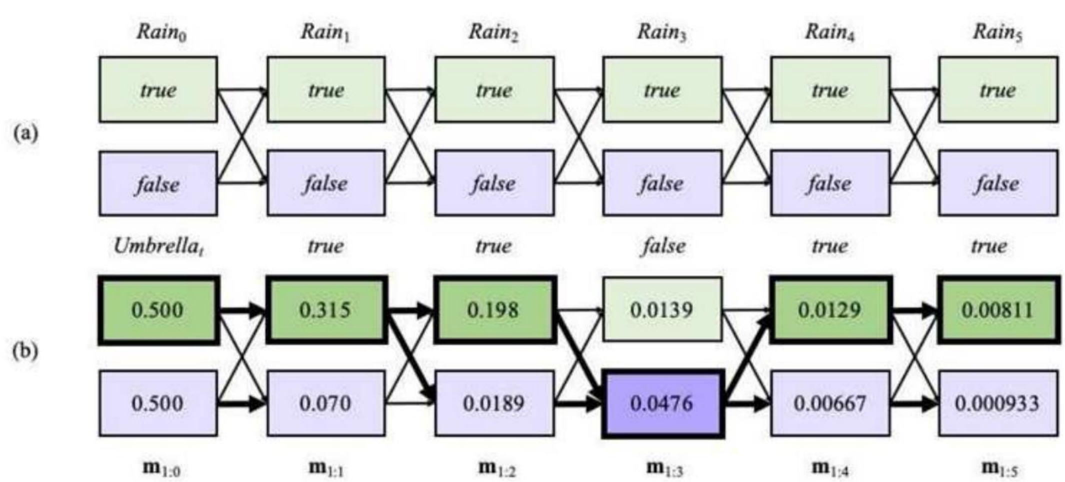
```
function FORWARD-BACKWARD(ev, prior) returns a vector of probability distributions
  inputs: ev, a vector of evidence values for steps 1,...,t
          prior, the prior distribution on the initial state, P(X0)
  local variables: fv, a vector of forward messages for steps 0,...,t
                   b, a representation of the backward message, initially all 1s
                   sv, a vector of smoothed estimates for steps 1,...,t

  fv[0] ← prior
  for i = 1 to t do
     fv[i] ← FORWARD(fv[i − 1], ev[i])
  for i = t down to 1 do
     sv[i] ← NORMALIZE(fv[i] × b)
     b ← BACKWARD(b, ev[i])
  return sv
```

前向后向算法的缺点：一是如果序列很长是，算法的空间复杂度可能会过高。二是联机环境工作下，算法必须在新证据不断追加到序列末尾的同时，为以前的时间片计算平滑估计。对于这一项缺点，有一个很常见的要求是：**固定延迟平滑（fixed-lag smoothing）**，就像固定窗口长度一样，某种意义上可以认为是**广义的马尔可夫假设（个人理解）**。但是这样效率也不是很高，在隐马尔可夫模型中，某些情况下固定延迟平滑能够实现每次更新都在常数时间内完成，而且与窗口长度无关。

3.3 Finding the most likely sequence 寻找最可能序列

假设[true，true，false，true，true]是警卫观察到的前 5 天的雨伞序列。解释这个序列的最可能的天气序列是什么？第 3 天没有出现雨伞是否意味着这天没下雨还是主管忘记了带伞？如果第 3 天没有下雨，也许（因为天气的持续性）第 4 天也不会下雨，而主管第四天带了雨伞只是为了以防万一。我们可选择的天气序列总共有2^5。是否有方法能找到最可能的序列而不用把所有序列枚举出来呢？

我们可以尝试一下下面这个线性时间过程：用平滑算法找到每个时间步上的天气后验分布；然后用每个时间步上的后验分布最可能一致的天气来构造这个序列。这其实就是通信系统中的**译码过程**。这里采用 **Viterbi 译码**，还是以雨伞的例子为例：



我们考虑如何在这个图中寻找到最可能的路径。每个路径的似然是沿着该路径的转移概

率和每个状态的给定观察结果的概率乘积。

首先我们把注意到$Rain_5 = true$的路径。由于马尔可夫特性，最可能到达状态$Rain_5 = true$的路径包含了到达时刻 4 的某个状态的最可能路径和到达$Rain_5 = true$的转移；而在时刻 4 将成为到达$Rain_5 = true$的路径的一部分的状态就是使该路径的似然达到最大值的那个状态。简单来说就是，**在每个状态x_{t+1}的最可能路径与到达每个状态x_t的最可能路径之间存在递归关系**。我们可以将这种关系写成与路径的概率有关的公式：

我们定义最可能路径决定公式：$m_{1:t} = \max_{x_{1:t-1}} P(x_{1:t-1}, X_t, e_{1:t})$，下边寻找其递归关系：

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

我们将这个公式与滤波公式对比：

$$m_{1:t+1} = P(e_{t+1}|X_{t+1}) \max_{x_t} P(X_{t+1}|x_t) \ \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t) P(x_t|e_{1:t}) \text{——滤波公式}$$

除了将求和符号换成最大值符号之外，和没有归一化符号α以外，其余都是相同的，所有我们也可以将这个过程看成滤波。起始在 t=0 时刻，$m_{1:0} = P(X_0)$，通过迭代计算出的结果在上边图片（b）中，最后我们可以很容易的选择总体上最可能的序列，即图中粗箭头所示。

3.4 My idea 我的想法

虽然这一部分公式多且繁琐，但是整体思路和公式总结起来，无疑是用到了以下几点：**条件概率公式及一阶马尔可夫假设的联合应用、贝叶斯全概率公式、独立事件的联合概率等于其概率乘积、前向-后向算法在平滑计算的应用、Viterbi 译码寻找最可能序列、递归公式的寻找**等。

这里我从一个通信学生的角度来看，一个简单的**通信系统**中包含信源、编码、信道（噪声）、解码、信宿，这一节的内容正好可以看作与之**对应**。

比如拿雨伞的例子举例：

信源——天气：产生下雨或者不下雨两种信号。

编码——行人：是否拿雨伞（有失真编码，因为行人可能未雨绸缪，也可能忘了带）。

信道——传感器模型：警卫通过观察行人，来确定是否下雨，这一部分也是有噪声的。

解码——滤波：这里就包含滤波、预测、平滑、最可能序列的确定等步骤。

信宿——结果：最后我们才可以判断天气给我们传送的到底是什么信号。

这样一一对应之后这一部分内容就变得很顺畅。其中我**感受最深的两个方面：一是传感器**，因为在读这一节之前我对于传感器的理解仍然停留在电子电表，指示灯这样的水平，但是现在我发现，任何东西都可以是传感器，只要我们能通过他感知世界的变化或者得到某种有用的信息，并且传感器也可以当作是通信系统的编码器（有噪声或者无噪声），因为我们可以通过传感器来转换符号或信号的存在形式，来进行信息处理等工作。**二是滤波器**，在读

这篇文章之前，我还只认为滤波器是我们通信里边截取一个固定的带宽用的，比如高通、低通、带通、带阻滤波器等等，但是现在我发现我之前用的最小二乘法也是一个滤波过程，广义上来说，只要把我们不需要的东西滤掉的都是滤波器。我觉得这两个方面把通信和人工智能或者现实世界拉的更近了。

这里提到的预测、滤波和平滑是有递进关系的。预测——用过去预测未来；滤波——用过去预测，并通过现在修正；平滑——通过未来的数据，反馈回去，重新评估计算的数据。

4. Hidden Markov Models 隐马尔可夫模型

隐马尔可夫模型（Hidden Markov Models，HMM）是单个离散随机变量描述过程状态的属虚概率模型。该变量的可能取值就是世界的可能状态。比如前边提到的雨伞例子，只有一个状态变量 Rain。如果模型具有多个变量，可以整合成一个多维大变量。这样他仍然适用于隐马尔可夫模型。

4.1 Simplified matrix algorithms 简化的矩阵算法

使用单个的、离散的状态变量X_t，我们能够给出表示转移模型、传感器模型以及前向与后向的消息的具体形式。设状态变量X_t用整数 1，…，S 表示，其中 S 表示可能状态的数目。转移模型$P(X_t|X_{t-1})$成为一个 S×S 的矩阵 T，其中：

$$T_{ij}=P(X_t=j|X_{t-1}=i)$$

比如雨伞的例子中，转移矩阵为：

$$T_{ij}=P(X_t|X_{t-1})=\begin{pmatrix}0.7 & 0.3\\ 0.3 & 0.7\end{pmatrix}$$

我们同样可以将传感器模型用矩阵形式表示。由于证据变量E_t的取值在 t 时刻是已知的，称作e_t，我们只需要为每个状态指定这个状态使e_t出现的概率是多少。为了数学计算方便，我们同样把这些值放入 S×S 的矩阵O_t中，他的第 i 个对角线元素是$P(e_t|X_t=i)$，其他元素是 0。

比如雨伞的例子中，第 1 天($U_1=true$)、第 3 天($U_3=false$)的传感器矩阵分别为：

$$O_1=\begin{pmatrix}0.9 & 0\\ 0 & 0.2\end{pmatrix};O_3=\begin{pmatrix}0.1 & 0\\ 0 & 0.8\end{pmatrix}$$

现在我们用列向量表示前向与后向的消息，整个计算过程变成如下的矩阵-向量运算。

前向公式：

$$P(X_{t+1}|e_{1:t+1})=\alpha P(e_{t+1}|X_{t+1})\sum_{x_t}P(X_{t+1}|x_t)P(x_t|e_{1:t})\Rightarrow f_{1:t+1}=\alpha O_{t+1}T^T f_{1:t}$$

后向公式：

$$P(e_{k+1:t}|X_k)=\sum_{x_{k+1}}P(e_{k+1}|x_{k+1})P(e_{k+2:t}|x_{k+1})P(x_{k+1}|X_k)\Rightarrow b_{k+1:t}=TO_{k+1}b_{k+2:t}$$

这样简化的矩阵算法提供了什么好处呢？最主要的是算法改进！主要有以下两个方面：

一是前向-后向算法的简单变形：

使得算法可以在常数空间内进行执行平滑，而与序列长度无关。他的主要思想是**矩阵的可逆**，因为前边提到的前向后向算法对于任何特定时间片 k 的平滑估计都需要同时给出前向和后向消息，即$P(X_k|e_{1:t})=\alpha f_{1:k}\times b_{k+1:t}$公式中的$f_{1:k}$和$b_{k+1:t}$。前向后向算法中是将前向过程运行之后结果保存起来以供后边使用。如果采用矩阵改进的话只运行一次，这一

次里同时向着相同的方向传递 f 和 b。例如，对公式进行矩阵左边乘逆的操作，$f_{1:t+1}=\alpha O_{t+1}T^{T}f_{1:t} \Longrightarrow f_{1:t}=\alpha'(T^{T})^{-1}O_{t+1}^{-1}f_{1:t+1}$。但是这个算法的最大缺点是转移矩阵必须可逆，而且传感器模型没有零元素——在每个状态下每个观察值都是可能的。

二是具有固定延迟的联机平滑中：

平滑能够在常数空间内完成，从这里我们可以知道联机平滑应该存在一种高效的递归算法——即一种时间复杂度与延迟长度无关的算法。如果我们假设延迟为 d；我们需要对时间片 t-d 进行平滑，其中 t 表示当前时间。即计算$\alpha f_{1:t-d}\times b_{t-d+1:t}$，然而有了新的观察值之后，需要为时间片 t-d+1 进行计算$\alpha f_{1:t-d+1}\times b_{t-d+2:t+1}$，那么这可不可以通过增量方式实现呢？

前向增量计算可以直接通过递归迭代，那么后向增量计算如何得到呢？具体过程如下：$b_{k+1:t}=TO_{k+1}b_{k+2:t}$，零 k=t-d，并且迭代 d 词，得到：

$$b_{t-d+1:t}=\left(\prod_{i=t-d+1}^{t}TO_i\right)b_{t+1:t}=B_{t-d+1:t}\vec{1}$$

这里因为$b_{t+1:t}=\vec{1}$，$B_{t-d+1:t}$是 T 和 O 矩阵序列的乘积。B 可以认为是一个变换算子。联机状态下有了下一个观察之后，对于新的后向的消息有类似的连乘公式：

$$b_{t-d+2:t+1}=\left(\prod_{i=t-d+2}^{t+1}TO_i\right)b_{t+2:t+1}=B_{t-d+2:t+1}\vec{1}$$

比较上述两个公式可以得到关于算子 B 的迭代关系：

$$B_{t-d+2:t+1}=O_{t-d+1}^{-1}T^{-1}B_{t-d+1:t}TO_{t+1}$$

这就得到了增量式的迭代关系，所以得到矩阵简化优化后具有固定时间延迟 d 的完整平滑算法如下：

```
function FIXED-LAG-SMOOTHING(e_t, hmm, d) returns a distribution over X_{t-d}
  inputs: e_t, the current evidence for time step t
          hmm, a hidden Markov model with S × S transition matrix T
          d, the length of the lag for smoothing
  persistent: t, the current time, initially 1
              f, the forward message P(X_t | e_{1:t}), initially hmm.PRIOR
              B, the d-step backward transformation matrix, initially the identity matrix
              e_{t-d:t}, double-ended list of evidence from t − d to t, initially empty
  local variables: O_{t-d}, O_t, diagonal matrices containing the sensor model information

  add e_t to the end of e_{t-d:t}
  O_t ← diagonal matrix containing P(e_t | X_t)
  if t > d then
     f ← FORWARD(f, e_{t-d})
     remove e_{t-d-1} from the beginning of e_{t-d:t}
     O_{t-d} ← diagonal matrix containing P(e_{t-d} | X_{t-d})
     B ← O_{t-d}^{-1} T^{-1} B T O_t
  else B ← B T O_t
  t ← t + 1
  if t > d + 1 then return NORMALIZE(f × B1) else return null
```

其中 NORMALIZE(f×B1)就是最终的输出。

4.2 Hidden Markov model example：Localization 定位

这一小节主要介绍了 HMM 的一个实际例子：**机器人的定位**。

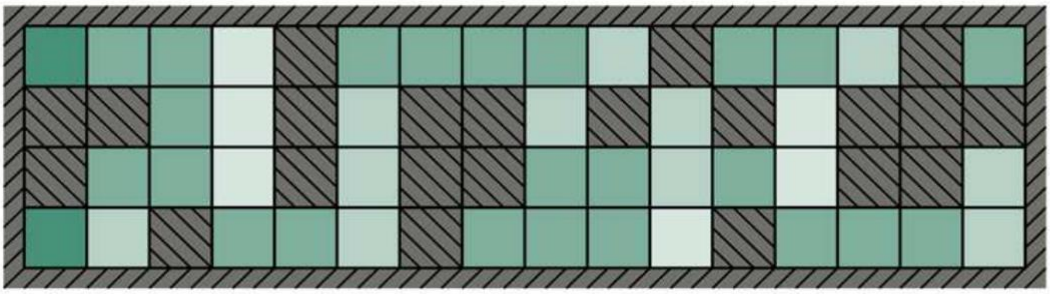
要解决的问题：一开始我们并不知道机器人在迷宫中的位置，可以**用滤波估计当前位置**，

也可以**使用平滑寻找任何给定过去时间所处的位置**，使用 **Viterbi 算法找出到达目前所处位置的最可能路径**。

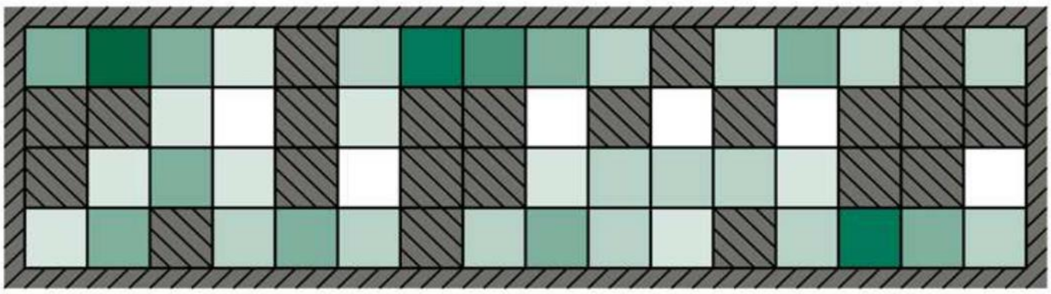
先前假设：状态变量X_t表示机器人在离散网格里的位置；这个变量的定义域是空方块组成的集合$\{s_1, \dots, s_n\}$。设 NEIGHBORS(s)是与 s 相邻的空方块集合，并设 N(s)是这个集合的大小。然后机器人只有一个非确定性行为 MOVE，在这个 MOVE 转移模型中，机器人等可能地移动到各相邻方块：

$$P(X_{t+1}=j|X_t=i)=T_{ij}=\begin{cases} 1/N(i) \ \ if\ j \in NEIGHBORS(i) \\ 0 \qquad\qquad otherwise \end{cases}$$

我们并不知道机器人的初始位置，所以我们将假设在各方块是均匀分布的；也就是$P(X_0=i)=1/n$。我们考虑 n=42 的特定环境，如下图：



(a) Posterior distribution over robot location after $E_1 = 1011$



(b) Posterior distribution over robot location after $E_1 = 1011$, $E_2 = 1010$

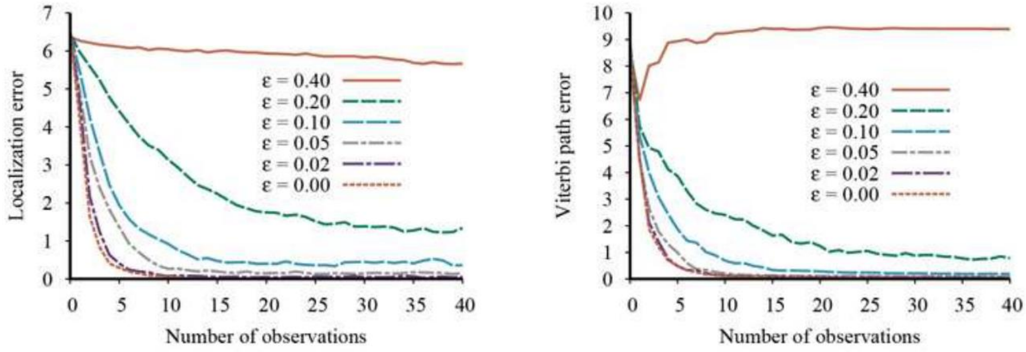
则转移矩阵 T 有 42×42=1764 个元素。传感器变量E_t有 16 个取值，采用 4bit 编码，用来描述东南西北四个方向是否有障碍物。我们用符号记录，比如 NS 表示北边和南边有障碍物，而东边和西边没有障碍物。设每个传感器的错误率为ϵ，而且四个方向的错误率相互独立，所以四个位置都是正确的概率为$(1-\epsilon)^4$，都是错误的概率为ϵ^4。此外，如果对于方块 i 的真实值与实际读数e_t之间的差异——不同位置的数量——是d_{it}（通信中的 hamming 距离），那么我们就可以得到方块 i 中的而机器人得到传感器读书e_t的概率是

$$P(E_t=e_t|X_t=i)=O_{t_{ii}}=(1-\epsilon)^{4-d_{it}}\epsilon^{d_{it}}$$

例如，北边和南边有障碍物的方块得到传感器读数 NSE 的概率为$(1-\epsilon)^3\epsilon^1$。

给定矩阵 T 和 O 之后，机器人就可以使用上一节的前向公式：$f_{1:t+1}=\alpha O_{t+1}T^T f_{1:t}$计算位置的概率分布——**也就是找出机器人所处的位置**。上图给出了$P(X_1|E_1=NSW)$和$P(X_2|E_1=NSW, E_2=NS)$。

除了估计当前位置以外，还可以计算平滑，并使用 Viterbi 译码得到到达目前所处位置的最可能正确路径。下图给出了传感器错误率ϵ取不同值时的位置错误与 Viterbi 路径的错误率。



(a) (b)

可以看到：

①$\epsilon=40\%$时，这个模型的鲁棒性很差；

②$\epsilon=20\%$时，即使在$1-(1-\epsilon)^4=59.04\%$的概率都是错误的情况下，也能在计算20个观察点之后将位置锁定在1到2个方块中间。这是因为算法考虑了转移模型施加在位置序列上的概率约束；

③$\epsilon\leqslant10\%$时，计算小于等于6个观察点之后，就可以将位置锁定，与理想传感器$\epsilon=0$相媲美。

总的来说，即使使用的模型有很多错误，可以保持高水平的定位和路径的正确率。

4.3 My idea 我的想法

这一小节主要是讲了隐马尔可夫模型和通过矩阵来简化算法的步骤。

对于这一节我有一个问题想了很久：那就是**如果矩阵不可逆**的话我们这个模型不再适用，那有没有什么方法呢？

我想到的方法就是在不可逆矩阵的每个元素上，在不影响矩阵整体性能和数字分布的情况下，增加小的随机扰动，令其可逆，但是不知道求出来的逆矩阵的性质该如何变化，是否会出现非常极端的数字？

而且，如果我们的模型有很多状态变量，那样计算量将会非常大，那应该如何简化呢？还有极端情况下：如果我们的机器人的移动是连续的而不是在离散网格里移动，则会有无限多个状态，那又该如何解决这个问题呢？

5. Kalman Filters 卡尔曼滤波

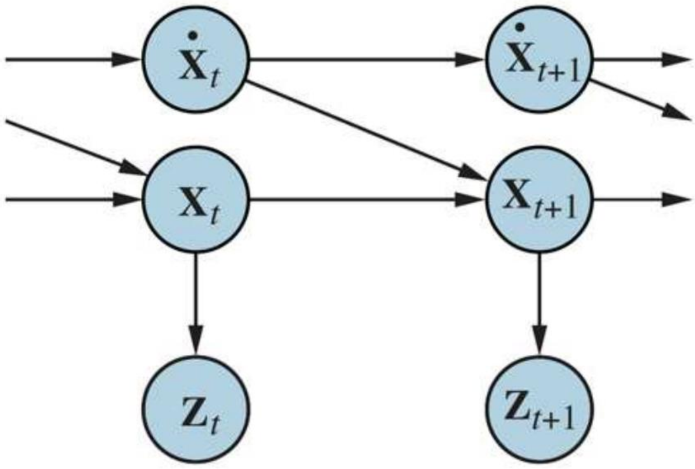
卡尔曼滤波：根据随时间**连续**变化并且带有噪声的观察数据去状态变量。如果是随着时间离散变化的话，我们可以用HMM隐马尔可夫模型。比如你在傍晚时分看着一只小鸟飞行穿过浓密的丛林：你只能瞥见小鸟隐隐约约、断断续续的闪现；你试图努力地猜测小鸟在哪里以及下一时刻它会出现在哪里，以至于不失去他的行踪。这就是这一节要做的内容。

小鸟的飞行可以用每个时间点的6个连续变量来描述，三个变量用于位置(X_t,Y_t,Z_t)，三个变量用于速度$\left(\dot{X}_t,\dot{Y}_t,\dot{Z}_t\right)$。还需要合适的条件概率密度来表示转移模型和传感器模型。

这里我们适用**线性高斯**分布。这意味着下一个状态X_{t+1}必须是当前状态X_t的线性函数，再加上某个高斯噪声，这实际上是非常合理的。比如，考虑小鸟的X坐标，暂时忽略其他坐标。令观测之间的间隔为 Δ ，并假设在观测间隔里速度不变；那么位置更新由$X_{t+\Delta}=X_t+\dot{X}_t\Delta$给出。增加高斯噪声后，我们得到一个线性高斯转移模型：

$$P\left(X_{t+\Delta}=x_{t+\Delta}|X_t=x_t,\dot{X}_t=\dot{x}_t\right)=N(x_t+\dot{x}_t\Delta,\sigma^2)(x_{t+\Delta})$$

下图给出了包含位置X_t和速度$\dot{X}_t$的系统的贝叶斯网络结构。这是一种形式特定的线性高斯模型。



一个含有d个变量的多元高斯分布可以用一个d元均匀向量μ和一个$d\times d$的协方差矩阵Σ来决定。

5.1 Updating Gaussian distributions 更新高斯分布

我们知道，线性高斯分布有一个关键的性质：在标准贝叶斯网络操作下这个运算是**封闭**的。比如：

如果当前分布$P(X_t|e_{1:t})$是高斯分布，并且转移模型$P(X_{t+1}|x_t)$是线性高斯的，那么由$P(X_{t+1}|e_{1:t})=\int_{x_t}P(X_{t+1}|x_t)P(x_t|e_{1:t})dx_t$给出的单步预测分布也是高斯分布。

如果预测分布$P(X_{t+1}|e_{1:t})$是高斯分布，传感器模型$P(e_{t+1}|X_{t+1})$是线性高斯的，那么条件化新证据后，更新后的分布$P(X_{t+1}|e_{1:t+1})=\alpha P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$也是高斯分布。

因此卡尔曼滤波的 FORWARD 算子选取一个高斯前向消息，该消息由均值μ_t和协方差矩阵Σ_t确定；并产生一个新的多元高斯前向的消息$f_{1:t+1}$，该消息由均值μ_{t+1}和协方差矩阵Σ_{t+1}确定。因此，如果我们从高斯先验概率$f_{1:0}=P(X_0)=N(\mu_0,\Sigma_0)$出发，用一个线性高斯模型进行滤波，在任何时间片都会产生一个高斯状态分布。

这一点这么重要的原因在于，除了一些像这样的特殊情况外，连续或者离散连续混合的网络的滤波过程会生成的状态分布其表示的规模随时间增长而趋于无界。

5.2 A simple one-dimensional example 简单的一维实例

前边提到，卡尔曼滤波器中的 FORWARD 算子将一个高斯分布映射到另一个新的高斯

分布。这其实转变成了我们寻找递归关系的思路，即根据原有的均值与协方差矩阵计算新的均值与协方差矩阵的过程。我们先考虑简单的一维情况。

我们考虑的时序模型描述了有噪声观察Z_t的单一连续状态变量X_t的**随机行走**（random walk）。一个例子是“消费者信心”指数，信心指数每个月会发生一次随机的服从高斯分布的变化，同时通过对一个随机的消费调查来度量，这个调查当然也会引入一个高斯采用噪声。假设其**先验概率**具有方差σ_0^2的高斯分布：

$$P(x_0)=\alpha e^{-\frac{1}{2}\left(\frac{(x-\mu_0)^2}{\sigma_0^2}\right)}$$

转移模型在当前状态中增加了一个具有常数方差σ_x^2的高斯扰动：

$$P(x_{t+1}|x_t)=\alpha e^{-\frac{1}{2}\left(\frac{(x_{t+1}-x_t)^2}{\sigma_x^2}\right)}$$

假设**传感器模型**具有方差为σ_z^2的高斯噪声：

$$P(z_t|x_t)=\alpha e^{-\frac{1}{2}\left(\frac{(z_t-x_t)^2}{\sigma_z^2}\right)}$$

现在，已知先验分布$P(X_0)$，我们可以使用公式，得到单步预测结果

$$P(X_{t+1}|e_{1:t})=\int_{x_t}P(X_{t+1}|x_t)P(x_t|e_{1:t})dx_t$$

$$P(x_1)=\int_{-\infty}^{+\infty}P(x_1|x_0)P(x_0)dx_0=\alpha\int_{-\infty}^{+\infty}e^{-\frac{1}{2}\left(\frac{(x_1-x_0)^2}{\sigma_x^2}\right)}e^{-\frac{1}{2}\left(\frac{(x_0-\mu_0)^2}{\sigma_0^2}\right)}dx_0$$

$$=\alpha\int_{-\infty}^{+\infty}e^{-\frac{1}{2}\left(\frac{\sigma_0^2(x_1-x_0)^2+\sigma_x^2(x_0-\mu_0)^2}{\sigma_0^2\sigma_x^2}\right)}dx_0$$

这个积分虽然看起来非常复杂，但是还是可以简化的，一般化的简化步骤采用**配方法**：

因为任意一个二次多项式都可以改写为：$ax_0^2+bx_0+c=a\left(x_0-\frac{-b}{2a}\right)^2+\left(c-\frac{b^2}{4a}\right)$

关于x_0的多项式$\frac{\sigma_0^2(x_1-x_0)^2+\sigma_x^2(x_0-\mu_0)^2}{\sigma_0^2\sigma_x^2}$，我们可以令其系数为

$$a=\frac{\sigma_0^2+\sigma_x^2}{\sigma_0^2\sigma_x^2},b=-\frac{2(\sigma_0^2x_1+\sigma_x^2\mu_0)}{\sigma_0^2\sigma_x^2},c=\frac{\sigma_0^2x_1^2+\sigma_x^2\mu_0^2}{\sigma_0^2\sigma_x^2}$$

而且由于高斯积分$\int_{-\infty}^{+\infty}e^{-\frac{1}{2}\left(a\left(x_0-\frac{-b}{2a}\right)^2\right)}dx_0=1$，所积分后只剩下$P(x_1)=\alpha e^{-\frac{1}{2}\left(c-\frac{b^2}{4a}\right)}$

即

$$P(x_1)=\alpha e^{-\frac{1}{2}\left(\frac{(x_1-\mu_0)^2}{\sigma_0^2+\sigma_x^2}\right)}$$

也就是说单步预测结果是一个具有相同均值μ_0的高斯分布，而其方差等于原来方差σ_0^2与转移方差σ_x^2的和。

单步预测后，我们要通过传感器模型完成更新步骤，即将第 1 个时间片的观察条件化，根据公式$P(X_{t+1}|e_{1:t+1})=\alpha P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$，我们可以得到：

$$P(x_1|z_1)=\alpha P(z_1|x_1)P(x_1)=\alpha e^{-\frac{1}{2}\left(\frac{(z_1-x_1)^2}{\sigma_z^2}\right)}e^{-\frac{1}{2}\left(\frac{(x_1-\mu_0)^2}{\sigma_0^2+\sigma_x^2}\right)}$$

对这个公式，再一次采用配方法，具体步骤省略，得到

$$P(x_1|z_1) = \alpha e^{-\frac{1}{2}\left(\frac{\left(x_1 - \frac{(\sigma_0^2+\sigma_x^2)z_1+\sigma_z^2\mu_0}{\sigma_0^2+\sigma_x^2+\sigma_z^2}\right)^2}{(\sigma_0^2+\sigma_x^2)\sigma_z^2/(\sigma_0^2+\sigma_x^2+\sigma_z^2)}\right)}$$

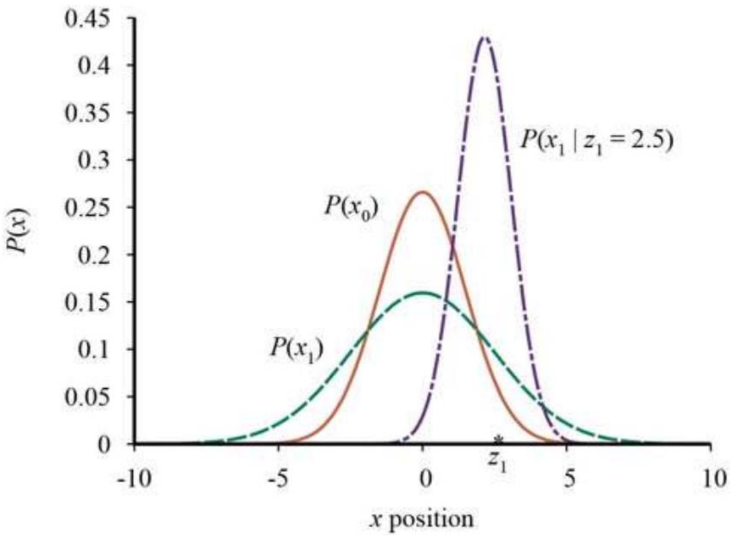
于是，经过更新后，得到了状态变量的一个新的高斯分布。

可以知道，通过上述更新公式，新的均值和标准差都可以由原来的均值和标准差按照下边的迭代公式计算：

$$\mu_{t+1} = \frac{(\sigma_t^2+\sigma_x^2)z_{t+1}+\sigma_z^2\mu_t}{\sigma_t^2+\sigma_x^2+\sigma_z^2} \text{和} \sigma_{t+1}^2 = \frac{(\sigma_t^2+\sigma_x^2)\sigma_z^2}{\sigma_t^2+\sigma_x^2+\sigma_z^2}$$

下图给出了对转移模型和传感器模型的特定取值的一轮更新。

各参数取值为：$\mu_0=0.0$、$\sigma_0=1.0$、$\sigma_x=2.0$、$\sigma_z=0.0$、$z_1=2.5$。



我们可以看到其实卡尔曼滤波（连续状态变量）的公式$\mu_{t+1} = \frac{(\sigma_t^2+\sigma_x^2)z_{t+1}+\sigma_z^2\mu_t}{\sigma_t^2+\sigma_x^2+\sigma_z^2}$、$\sigma_{t+1}^2 = \frac{(\sigma_t^2+\sigma_x^2)\sigma_z^2}{\sigma_t^2+\sigma_x^2+\sigma_z^2}$与 HMM（离散状态变量）中的公式$f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t}$，其作用都是一样的。

除了性质一样，因为是高斯分布，所以有一些很**奇特的性质，一是**：观察μ_{t+1}的公式，其实是将观察值z_{t+1}与旧均值μ_t进行了加权平均，而加权平均系数取决于方差，如果观察不可靠，那么σ_z^2一定很大，那么旧均值的系数更大，我们更加关注均值；如果旧的均值不可靠（σ_t^2很大）或者这个过程不好预测（σ_x^2很大），那么我们更加关注于观察值。**二是：** 观察σ_{t+1}^2的公式，其实是独立于观察值的，因此我们可以事先计算出方差取值的序列。

5.3 The general case 一般情况

前边我们推导了卡尔曼滤波工作的基础——高斯分布的关键性质：指数是二次多项式形式的。完整的**多元高斯分布**具有如下形式：

$$N(x;\mu,\Sigma) = \alpha e^{-\frac{1}{2}((x-\mu)^T\Sigma^{-1}(x-\mu))}$$

观察指数项，可以看到指数部分也是x中的x_i的二次函数。

首先，用卡尔曼滤波**定义一般的时序模型**。转移模型和传感器模型都允许附加高斯噪声

的线性变换，因此我们有：

$$P(x_{t+1}|x_t) = N(x_{t+1}; Fx_t, \Sigma_x)$$

$$P(z_t|x_t) = N(z_t; Hx_t, \Sigma_z)$$

其中F和Σ_x是描述线性转移模型和转移噪声协方差的矩阵，而H和Σ_z是传感器模型的相应矩阵。现在的均值于协方差的更新公式为：

$$\mu_{t+1} = F\mu_t + K_{t+1}(z_{t+1} - HF\mu_t)$$

$$\Sigma_{t+1} = (I - K_{t+1}H)(F\Sigma_t F^T + \Sigma_x)$$

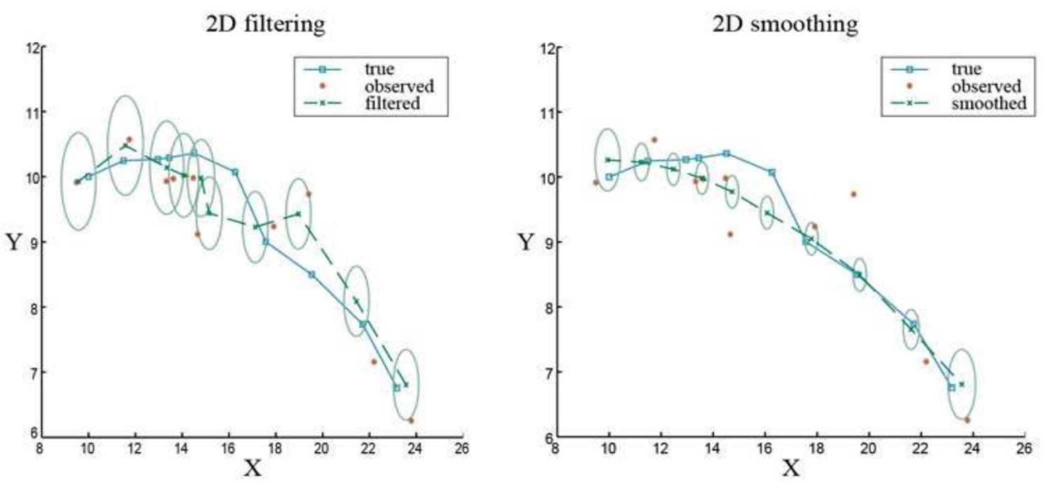
其中$K_{t+1} = (F\Sigma_t F^T + \Sigma_x)H^T(H(F\Sigma_t F^T + \Sigma_x)H^T + \Sigma_t)^{-1}$被称为**卡尔曼增益矩阵（Kalman gain matrix）**。

那这**三个公式的意义**是什么呢？其实我们对比之前一维的μ_{t+1}和σ_{t+1}^2的迭代公式，可以看出来：$F\mu_t$是 t+1 时刻的预测状态，$HF\mu_t$是预测观察值。因此$z_{t+1} - HF\mu_t$表示预测观察值的误差。我们可以将其乘K_{t+1}来修正预测状态；因此K_{t+1}是相对于预测的一个加权系数。

那我们具体看一看**三个公式的应用**，我们将他们应用到 X-Y 平面上的运动物体跟踪问题（二维问题）。那么状态变量为$X = \left(X, Y, \dot{X}, \dot{Y}\right)^T$，因此$F$、$\Sigma_x$、$H$、$\Sigma_z$都是 4×4 的矩阵。下图（a）给出了物体的真实轨迹、带有噪声的观察结果以及通过卡尔曼滤波得到的轨迹。这里的椭圆表示方差的大小，如期望所示，方差很快达到了一个不动点。

和滤波过程一样，也可以用线性高斯模型推导出平滑公式。平滑过程如下图（b）所示。方差是急剧减小的，除了轨迹末端，而且估计的轨迹变得很平滑。

我认为末端的方差很大，原因就是加权系数K_{t+1}的问题。但是如果分析得话，我们要重复上述过程，得到关于平滑的递推公式。



(a) (b)

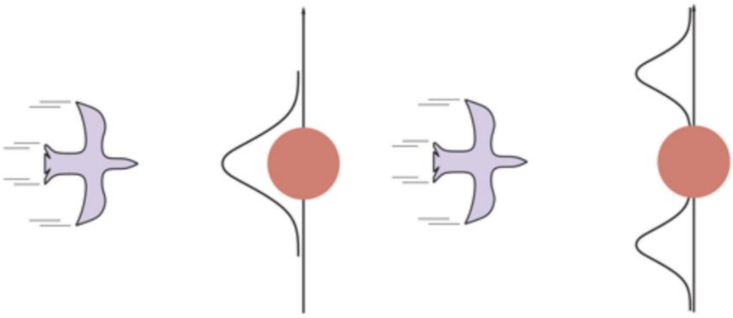
5.4 Applicability of Kalman filtering 卡尔曼滤波的适用性

卡尔曼滤波的应用有很多，比如对飞行器及导弹的雷达跟踪；潜艇和地面车辆的声学跟

踪；车辆和人的视觉追踪等。但是卡尔曼滤波的仅是物体的运动轨迹跟踪，正如在这一节开始时提到的：**任何通过连续状态变量与噪声测量来刻画的系统都可以应用卡尔曼滤波**。

卡尔曼滤波能够应用于一个系统，并不是其结果一定是有效或者有用的。而且这里面临着同样的问题，即如果矩阵不可逆，那应该如何变化？即非线性问题。而**扩展的卡尔曼滤波器（extended Kalman filter）**试图解决这个问题。其工作机理是将$x_t = \mu_t$的区域中的状态x_t当作局部线性的，在此基础上对系统进行建模——即是微分思想（**个人理解**）。而具体的内容在课本中放到了第 26 章。

那我们以小鸟在树林飞行的例子来解释一下什么是非线性的？比如我们追踪一只鸟飞行时正以很快的速度笔直朝着一个树桩飞过去。不论是常规卡尔曼滤波器还是扩展卡尔曼滤波器，都只会对鸟的位置做出一个高斯预测，而高斯分布的均值将以树桩作为中心，如下图左边所示。



但是，关于鸟的合理模型应该要预测到鸟的躲避树桩的行为，从树桩的某一侧绕过去，如下图右边所示。这样的模型其实就是高度非线性的，因为鸟的决策的变化高度依赖于它相对于树桩的精确位置，

对于这种问题的标准解决方法是使用**切换卡尔曼滤波器（switching Kalman filter）**。在这种方法中多个卡尔曼滤波器并行的运行，其中每个都是用不同的系统模型——比如，一个直行，一个向左急转，一个向右急转。我们使用这些预测结果的加权和，其中权重依赖于每个滤波器对当前数据的适合程度。这其实跟我们后边课上讲的**匈牙利算法**很相似。而且这其实是**动态贝叶斯网络模型**中的一种特殊情况。

5.5 My Idea 我的想法

这一节讲述了卡尔曼滤波，这与前一节的隐马尔可夫模型 HMM 形成一个互补的关系。因为**卡尔曼滤波**是对于**连续的状态变量**的求滤波，平滑，寻找最可能路径的方法，而**隐马尔可夫模型**是对于**离散的状态变量**的求滤波，平滑，寻找最可能路径的方法。

这一节的卡尔曼滤波的主要方法是将隐马尔可夫过程连续化，这里使用**线性高斯分布**，一开始研究简单的**一维高斯分布**的滤波和平滑，然后将向量矩阵化，得到**多维高斯分布**的滤波和平滑，再以**二维高斯分布**具体举例，观察了 X-Y 平面上的运动物体跟踪观察的结果。之后再介绍了**扩展的卡尔曼滤波器**和**切换卡尔曼滤波器**以适应更加复杂的并且灵活的模型。

虽然这一节公式繁多，但是与 HMM 模型都可以比拟来计算出来。简化过程中，主要使用了**高斯分布的性质**和**配方法**联合化简积分结果，得到递归关系。

但是借助矩阵来简化，必定会引入**矩阵不可逆**问题——非线性问题，对此我觉得可以考

虑加入随机扰动，或是参考下一节的应用范围更加广泛的**动态贝叶斯网络（DBN，Dynamic Bayesian network）**。

6. Summary and inspire thinking 总结和启发思考

6.1 Summary 总结

本章主要讲述了概率时序过程的表示与推理的一般问题，总结为以下几点：

- 世界中变化的状态是通过一个随机变量集来表示每个时间点的状态来处理的。
- 随时间推移的过程，我们可以使用**马尔可夫假设**简化，这样未来不再依赖于过去，再结合**稳态过程假设**就可以大大简化我们的模型。
- 再有就是对我印象最深的**转移模型**和**传感器模型**。
- 还有就是贯穿整个章节的四个计算：**滤波**、**预测**、**平滑**以及**寻找最可能解释**。这里用到**递归算法**以及**前向-后向算法**。
- 还有对于离散状态变量使用到的**隐马尔可夫模型**、对连续状态变量使用到的**卡尔曼滤波器**。这里简化过程还用到了**配方法**和**高斯积分**。

6.2 Inspire thinking 启发思考

启发思考我想总结为以下几个部分，并且与通信系统的框图进行比拟：

- **首先是我们主要的四个计算：滤波、预测、平滑以及寻找最可能解释。**
  学完这一部分，我认识到：**预测是没有考虑当前观察值的滤波，滤波是没有考虑未来数据的平滑**。这样的递进关系可以令我的学习变得很容易，因为只要我知道平滑的具体算法，删减或者增加条件就可以实现与滤波、预测同样的效果。而这些在通信系统中其实就是对解码器的设计和优化过程。**寻找最可能解释**其实就是通信中的 Viterbi 译码，得到实际信源发出的信息。
- **其次是印象较为深刻的转移模型和传感器模型。**
  转移模型和传感器模型，就像我在对于章节写的想法一样，它其实是把我们电子系统中的转移模型和传感器模型带到了生活。对于这一部分我想的是：如果要优化传感器模型，一个简单的方法是提高马尔可夫模型的阶数；另一个方法就是使用**多传感器模型**，虽然可以增加精度，但是对于复杂度的考虑还是欠缺的。
- **还有就是如何简化模型？**
  什么时候需要简化模型呢？一是我们的模型太过复杂，此时可以使用**马尔可夫假设**和**稳态过程假设**，进而考虑的时间缩短，但是这需要过去和当前以及未来之间的一定的独立性。二是我们需要得到明显的关系，并扩展到更高的维度时，需要简化模型，比如我们使用到的**简化矩阵算法，线性高斯假设**，这样可以很容易的得到简单的关系式，并且易于扩展，比如从一维高斯分布拓展到多维高斯分布。
- **简化之后会出现什么问题？**
  简化后一个很明显的问题就是计算精度会下降，毕竟我们考虑的因素变小了。还有一个问题就是当我们引入矩阵运算时，会出现**矩阵不可逆——非线性问题**，那么这就会导致我们的模型适用范围变窄，此时卡尔曼滤波可以使用**切换卡尔曼滤波器**或者是使用**动态贝叶斯网络**模型来拓展适用范围。
- **隐马尔可夫模型和卡尔曼滤波器带给我的思考。**

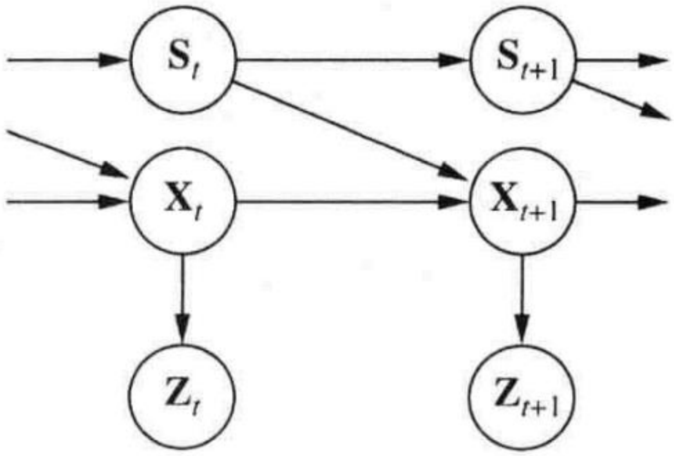
这两个模型，HMM 是对应离散的，而卡尔曼滤波器是对应连续的，那么从通信的角度来看，对卡尔曼滤波器进行“抽样”，是否可以得到 HMM 模型呢？或者是将时间片之间的间隔无限缩小，是否可以从 HMM 模型得到卡尔曼滤波器呢？如果可以的话，我认为应该存在对应的**切换隐马尔可夫模型**等等。

7. Exercise 课后练习题

这里我挑选了课后习题 14.10，是关于**切换卡尔曼滤波器**的习题。因为切换卡尔曼滤波器解决了上述的小鸟飞行中飞向树桩的问题。而且应用情况很多，可以看成是多个并行的卡尔曼滤波器。

7.1 Requirement 题目要求

14.10 当我们监控一个连续状态系统，其行为在 k 个不同“模式”间以不可预知的方式来回切换。例如，试图躲避导弹攻击的飞行器可能会做出一系列不同的机动飞行动作，而导弹则试图跟踪这些东西。下图给出了这种切换卡尔曼滤波器模型的一种贝叶斯网络表示。



A．假设离散状态S_t有 k 种可能取值，并且其先验连续状态估计$P(X_0)$是一个多元高斯分布。证明：预测$P(X_1)$是一个**混合高斯分布（mixture of Gaussians）**——即多个高斯分布的加权和，其中权值和等于 1。

B．证明：如果当前的联系状态估计$P(X_t|e_{1:t})$是 m 个高斯分布的混合，那么在一般情况下，更新后的状态估计$P(X_{t+1}|e_{1:t+1})$将是 km 个高斯分布的混合。

C. 在混合高斯表示中的权值表示了时序过程的哪一方面？

7.2 Solution 解答

A. 依据我们文中的推导，预测应该是由 k 个可能取值的加权和。

$$P(X_1) = \sum_{s_0=1}^{k} P(s_0) \int_{x_0} P(x_0) P(X_1|x_0, s_0)$$

由卡尔曼滤波器的性质，我们知道积分对每个不同S_0都给出了一个高斯分布。因此，预测分布是 k 个高斯分布的混合，每个高斯分布的权重为$P(s_0)$。

B. 更新后的切换卡尔曼滤波器的方程为：

$$P(X_{t+1},S_{t+1}|e_{1:t+1})$$
$$=\alpha P(e_{t+1}|X_{t+1},S_{t+1})\sum_{s_t=1}^{k}\int_{x_t}P(x_t,s_t|e_{1:t})P(X_{t+1},S_{t+1}|x_t,s_t)$$
$$=\alpha P(e_{t+1}|X_{t+1})\sum_{s_t}^{k}P(s_t|e_{1:t})P(S_{t+1}|s_t)\int_{x_t}P(x_t|e_{1:t})P(X_{t+1}|x_t,s_t)$$

因为$P(X_t|e_{1:t})$是 m 个高斯分布的混合。每个高斯分布都经过 k 个不同的线性高斯投影，然后再通过线性高斯观测值进行更新，因此我们就获得了 km 个高斯的混合。

C. 权值表示了时序过程中，每一个可能转换值得可能性。比如小鸟飞向树桩时，在做动作之前得前一个无穷小得时间点，直行的卡尔曼滤波器的取值可能性（权重）一定是最小的（因为小鸟本能的躲避树桩。）而左转或者右转的卡尔曼滤波器的取值可能性（权重）可能很大而且近似，比如可能存在这种情况：1%的概率直行，49%的概率左转，49%的概率右转，剩下 1%为别的情况。