1 基本概念

1.1 实例理解

1.1.1 初始化阶段-提取跟踪目标特征

该阶段要人工指定跟踪目标（比如开始时人工用鼠标框出跟踪目标），通过程序计算跟踪目标的特征，可以采用目标的颜色特征，比如目标所在区域色调空间的直方图，直方图可以用一个向量来表示，所以目标特征就是一个 N*1 的向量 V。

1.1.2 搜索阶段-放狗

我们已经掌握了目标的特征，下面放出很多条狗，去搜索目标对象，这里的狗就是粒子。狗有很多种放法，比如均匀地放（在整个图像平面均匀防地布粒子），也可以在上一帧得到的目标附近按照高斯分布来放（靠近目标的地方多放，远离目标的地方少放）。狗放出去后，每条狗怎么搜索目标呢？就是按照初始化阶段得到的目标特征(色调直方图，向量 V)，每条狗计算它所处的位置处图像的颜色特征，得到一个色调直方图，向量 V_i，计算该直方图与目标直方图的相似性。相似性有多种度量，最简单的一种是计算归一化相似度 $V_i/sum(abs(V_i-V))$，使得所有的狗得到的归一化相似度加起来等于 1。

1.1.3 决策阶段

我们放出去的一条条聪明的狗向我们发回报告，“一号狗处图像与目标的相似度是 0.3”，“二号狗处图像与目标的相似度是 0.02”，“三号狗处图像与目标的相似度是 0.0003”，“N 号狗处图像与目标的相似度是 0.013”...那么目标究竟最可能在哪里呢？我们做次加权平均吧。设 N 号狗的图像像素坐标是(X_n,Y_n),它报告的相似度是 W_n，于是目标最可能的像素坐标 $X = sum(X_n*W_n),Y = sum(Y_n*W_n)$。

1.1.4 重采样阶段

既然我们是在做目标跟踪，一般说来，目标是跑来跑去乱动的。在新的一帧图像里，目标可能在哪里呢？还是让我们放狗搜索吧。但现在应该怎样放狗呢？让我们重温下狗狗们的报告吧。“一号狗处图像与目标的相似度是 0.3”，“二号狗处图像与目标的相似度是 0.02”，“三号狗处图像与目标的相似度是 0.0003”，“N 号狗处图像与目标的相似度是 0.013”...综合所有狗的报告，一号狗处的相似度最高，三号狗处的相似度最低，于是我们要重新分布警力，正所谓好钢用在刀刃上，我们在相似度最高的狗那里放更多条狗，在相似度最低的狗那里少放狗，甚至把原来那条狗也撤回来。这就是根据重要性重采样(更具重要性的位置放置更多的狗)。

搜索->决策->重采样->搜索如是反复循环，即完成了目标的动态跟踪。

1.2 学术理解

$$
\begin{aligned}
\mathrm{x}(t) &= f(x(t-1), w(t)) \\
\mathrm{y}(t) &= h(x(t), e(t))
\end{aligned}
$$

其中，x(t)为 t 时刻状态，w(t)和 e(t)分别为状态噪音和观测噪音。前一个是状态转移方程，后一个是观测方程。那么对于这样一个问题，粒子滤波怎么来从观测 y(t)和 x(t-1)得到真实状态 x(t)呢？

1.2.1 初始阶段

由于一开始对 x(0)一无所知，所有我们认为 x(0)在全状态空间内平均分布。然后将所有采样粒子输入状态转移方程，得到若干个预测粒子。

1.2.2 预测阶段

粒子滤波首先根据 x(t-1)的概率分布生成大量的采样（粒子），那么这些采样在状态空间中的分布实际是 x(t-1)的概率分布了。接下来依据状态转移方程可以对每一粒子得到下一个预测粒子。

1.2.3 校正阶段

观测值 y 得到后，可以利用条件概率 $p(y|x_i)$，对所有的粒子进行评价。直白的说，这个条件概率代表了假设真实状态 x(t)取第 i 个粒子 x_i 时获得观测 y 的概率。令这个条件概率为第 i 个粒子的权重。如此这样，继续对所有的粒子都进行这么个评价，那么越有可能获得观测 y 的粒子，当然获得的权重越高，可信度也就越高（越接近真实粒子状态）。

1.2.4 重采样

去除低权值的粒子，复制高权值的粒子，得到我们需要的真实状态 x(t)。而这些重采样后的粒子，就代表了真实状态的概率分布。下一轮滤波，再将重采样后的粒子集输入到状态转移方程中，直接就能够获得预测粒子了。

2 蒙特卡洛采样

现在假设我们获得了 t-1 时刻的一系列样本（粒子）$x_1, x_2, x_3 \ldots$，那么就可以根据这些粒子预测下一时刻的粒子状态 $f(x_1), f(x_2), f(x_3) \ldots$，若暂不考虑各个各个粒子的可信度，则可得到下一刻粒子的真实状态是 $E(f(x)) = \int f(x)p(x)dx$，而蒙特卡洛采样的思想就是用平均值来代替积分，得到 $E(f(x)) \approx \frac{f(x_1)+f(x_2)+\ldots+f(x_N)}{N}$，具体的推导过程如下：

根据之前的分析，贝叶斯后验概率的计算里要用到积分，为了解决这个积分难的问题，可以用蒙特卡洛采样来代替计算后验概率

$$\hat{p}(x_n|y_{1:k}) = \frac{1}{N}\sum_{i=1}^{N}\delta(x_n - x_n^{(i)})$$

$$E[f(x_n)] \approx \int f(x_n)\hat{p}(x_n|y_{1:k})dx_n$$

$$= \frac{1}{N}\sum_{i=1}^{N}\int f(x_n)\delta(x_n - x_n^{(i)})dx_n$$

$$= \frac{1}{N}\sum_{i=1}^{N} f(x_n^{(i)})$$

若直接以 $E(f(x)) \approx \dfrac{f(x_1)+f(x_2)+...+f(x_N)}{N}$ 作为 t 时刻的真实状态显然并不准确，因为 $x_1, x_2, x_3 ...$ 等若干个粒子的地位并不等价，有些粒子比较接近真实状态应该重点考虑，有些粒子则远远偏离于真实值在计算时应该有意识地忽略，所以相加求和时应根据重要性对每个粒子赋予不同的权重（在上面的推导中默认每个粒子的权重相等，都为 $\dfrac{1}{N}$ ）。

3 重要性采样（权重问题）

在前面我们已经提过狗的方法是多种多样的，一般情况下我们布粒子时就采用分布 $p(x_k | x_{k-1})$ ，但为了讨论更一般的情况，我们假设采用一个已知的分布 $q(x_k | x_{k-1})$ ，所以就可以从这个分布里去采样，这样上面的求期望问题就变成了：

$$\begin{aligned}
&E[f(x_k)] \\
&= \int f(x_k)\frac{p(x_k | y_{1:k})}{q(x_k | y_{1:k})}q(x_k | y_{1:k})dx_k \\
&= \int f(x_k)\frac{p(y_{1:k} | x_k)p(x_k)}{p(y_{1:k})q(x_k | y_{1:k})}q(x_k | y_{1k})dx_k \\
&= \int f(x_k)\frac{W_k(x_k)}{p(y_{1:k})}q(x_k | y_{1k})dx_k
\end{aligned}$$

其中由于每一次观测值 $y_1 \sim y_k$ 是确定的，所以像 $p(y_{1:k-1})$ 、 $p(y_k | y_{1:k-1})$ 等我们将其当做已已知的常数 C 处理：

$$W_k(x_{\mathrm{k}}) = \frac{p(y_{1:k} | x_k)p(x_k)}{q(x_k | y_{1:k})} = \frac{p(x_k | y_{1:k})}{q(x_k | y_{1:k})}p(y_{1:k}) \propto \frac{p(x_k | y_{1:k})}{q(x_k | y_{1:k})}$$

由于

$$p(y_{1:k}) = \int p(y_{1:k} \mid x_k) p(x_k) dx_k$$

结合蒙特卡洛采样思想，得：

$$\begin{aligned}
&E[f(x_k)] \\
&= \frac{1}{p(y_{1:k})} \int f(x_k) W_k(x_k) q(x_k \mid y_{1:k}) dx_k \\
&= \frac{\int f(x_k) W_k(x_k) q(x_k \mid y_{1:k}) dx_k}{\int p(y_{1:k} \mid x_k) p(x_k) dx_k} \\
&= \frac{\int f(x_k) W_k(x_k) q(x_k \mid y_{1:k}) dx_k}{\int W_k(x_k) q(x_k \mid y_{1:k}) dx_k} \\
&= \frac{E_{q(x_k \mid y_{1:k})}[f(x_k) W_k(x_k)]}{E_{q(x_k \mid y_{1:k})}[W_k(x_k)]} \\
&= \frac{\frac{1}{N} \sum_{i=1}^{N} f(x_k{}^{(i)}) W_k(x_k{}^{(i)})}{\frac{1}{N} \sum_{i=1}^{N} W_k(x_k{}^{(i)})} \\
&= \sum_{i=1}^{N} w_k(x_k{}^{(i)}) f(x_k{}^{(i)})
\end{aligned}$$

其中

$$w_k(x_k{}^{(i)}) = \frac{W_k(x_k{}^{(i)})}{\sum_{i=1}^{N} W_k(x_k{}^{(i)})}$$

这就是归一化以后的权重，它不再是之前式中所有的粒子状态直接相加求平均了，而是一种加权和的形式。不同的粒子都有它们相应的权重，如果粒子权重大，说明信任该粒子比较多。

4 权重的递推

到这里已经初步解决了权重问题，但是上面这种每个粒子的权重都直接计算的方法，效率低，因为每增加一个采样，$p(x_k \mid y_{1:k})$ 都得重新计算，并且还不好计算这个式子。所以求权重时能否避开计算 $p(x_k \mid y_{1:k})$ ？而最佳的形式是能够以递推的方式去计算权重，这就是所谓的序贯重要性采样（SIS）。

现在我们将已知的采样分布 $q(x_k \mid y_{1:k})$ 称为重要性概率密度函数，并记做 $q(x_{0:k} \mid y_{1:k})$ ，这

里 x 的下标是 0:k，也就是说粒子滤波是估计过去所有时刻的状态的后验（注意与贝叶斯滤波的区别），同样将 $p(x_k \mid y_{1:k})$ 改下为 $p(x_{0:k} \mid y_{1:k})$。

$$
\begin{aligned}
&P(X_n \mid Y_1, Y_2, \cdots, Y_n) \longleftarrow \text{Filter distribution} \\
&= \int \cdots \int P(X_0, X_1, \cdots, X_n \mid Y_1, Y_2, \cdots, Y_n)\, dX_{n-1} \cdots dX_0
\end{aligned}
$$

$p(x_{0:k} \mid y_{1:k})$ $q(x_{0:k} \mid y_{1:k})$ 推导如下：

$$
\begin{aligned}
&p(x_{0:k} \mid y_{1:k}) \\
&= \frac{p(x_{0:k}, y_{1:k})}{p(y_{1:k})} \\
&= \frac{p(x_{0:k}, y_{1:k-1}, y_k)}{p(y_{1:k-1}, y_k)} \\
&= \frac{p\left(x_{0:k}, y_k \mid y_{1:k-1}\right)}{p(y_k \mid y_{1:k-1})} \\
&= \frac{p\left(y_k \mid x_{0:k}, y_{1:k-1}\right) p\left(x_{0:k} \mid y_{1:k-1}\right)}{p(y_k \mid y_{1:k-1})} \\
&= \frac{p\left(y_k \mid x_{0:k}, y_{1:k-1}\right) p\left(x_k \mid x_{0:k-1}, y_{1:k-1}\right) p\left(x_{0:k-1} \mid y_{1:k-1}\right)}{p(y_k \mid y_{1:k-1})} \\
&= \frac{p\left(y_k \mid x_k\right) p\left(x_k \mid x_{k-1}\right) p\left(x_{0:k-1} \mid y_{1:k-1}\right)}{p(y_k \mid y_{1:k-1})} \\
&\propto p\left(y_k \mid x_k\right) p\left(x_k \mid x_{k-1}\right) p\left(x_{0:k-1} \mid y_{1:k-1}\right)
\end{aligned}
$$

$$
q(x_{0:k} \mid y_{1:k}) = q(x_{\ 0:k-1} \mid y_{1:k}) q(x_k \mid x_{0:k-1}, y_{1:k}) = q(x_{\ 0:k-1} \mid y_{1:k-1}) q(x_k \mid x_{0:k-1}, y_{1:k})
$$

$W_k(x_{\mathrm{k}})$ 递推如下：

$$
\begin{aligned}
&W_k^{(i)} \propto \frac{p(x^{(i)}{}_{0:k} \mid y_{1:k})}{q(x^{(i)}{}_{0:k} \mid y_{1:k})} \\
&= \frac{p\left(y_k \mid x_k^{(i)}\right) p\left(x_k^{(i)} \mid x_{k-1}^{(i)}\right) p\left(x_{0:k-1}^{(i)} \mid y_{1:k-1}\right)}{q(x_{0:k-1}^{(i)} \mid y_{1:k-1}) q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, y_{1:k})} \\
&= W^{(i)}{}_{k-1} \frac{p\left(y_k \mid x^{(i)}{}_k\right) p\left(x^{(i)}{}_k \mid x^{(i)}{}_{k-1}\right)}{q(x^{(i)}{}_k \mid x_{0:k-1}^{(i)}, y_{1:k})}
\end{aligned}
$$

在实际应用中我们可以假设重要性分布 q()满足：

$$q(x_k|x_{0:k-1}, y_{1:k}) = q(x_k|x_{k-1}, y_k)$$

这个假设说明重要性分布只和前一时刻的状态 x(k-1)以及测量 y(k)有关了

$$W_k^{(i)} \propto W_{k-1}^{(i)} \frac{p\left(y_k \mid x^{(i)}{}_k\right) p\left(x^{(i)}{}_k \mid x^{(i)}{}_{k-1}\right)}{q(x^{(i)}{}_k \mid x^{(i)}{}_{k-1}, y_k)}$$

注意，这种权重递推形式的推导是没有归一化的在实际应用中，递推计算出 w(k)后,要进行归一化，才能够代入式中去计算期望。

5 重采样

重采样的思路是：既然那些权重小的不起作用了，那就不要了。要保持粒子数目不变，得用一些新的粒子来取代它们。找新粒子最简单的方法就是将权重大的粒子多复制几个出来，至于复制几个？那就在权重大的粒子里面让它们根据自己权重所占的比例去分配 ,也就是老大分身分得最多，老二分得次多，以此类推。

前面已经说明了求某种期望问题变成了这种加权和的形式：

$$p(x_k|y_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta(x_k - x_k^{(i)})$$

通过重采样以后，希望表示成：

$$\tilde{p}(x_k|y_{1:k}) = \sum_{j=1}^{N} \frac{1}{N} \delta(x_k - x_k^{(j)}) = \sum_{i=1}^{N} \frac{n_i}{N} \delta(x_k - x_k^{(i)})$$

$x_k^{(i)}$ 是第 k 时刻的粒子，$x_k^{(j)}$ 是 k 时刻重采样以后的粒子。其中 n_i 是指粒子被复制的次数。第二个式子说明重采样以后，所有的粒子权重一样，都是 1/N，只是有的粒子多出现（被复制）n_i 了次。

6 算法流程

标准的粒子滤波算法流程为：

(1) 粒子集初始化，$k=0$：

对于 $i=1,2,\cdots,N$，由先验 $p(x_0)$ 生成采样粒子 $\{x_0^{(i)}\}_{i=1}^N$

(2) 对于 $k=1,2,\cdots$，循环执行以下步骤：

① 重要性采样：对于 $i=1,2,\cdots,N$，从重要性概率密度中生成采样粒子 $\{\tilde{x}_k^{(i)}\}_{i=1}^N$，计算粒子权值 $\tilde{w}_k^{(i)}$，并进行归一化；

② 重采样：对粒子集 $\{\tilde{x}_k^{(i)},\tilde{w}_k^{(i)}\}$ 进行重采样，重采样后的粒子集为 $\{x_k^{(i)},1/N\}$；

③ 输出：计算 k 时刻的状态估计值：$\hat{x}_k=\sum_{i=1}^{N}\tilde{x}_k^{(i)}\tilde{w}_k^{(i)}$。

7 SIR 粒子滤波器

在 SIR 中，选取：

$$q(x_k^{(i)}|x_{k-1}^{(i)},y_k)=p(x_k^{(i)}|x_{k-1}^{(i)})$$

代入：

$$W_k^{(i)}\propto W_{k-1}^{(i)}\frac{p\left(y_k\mid x^{(i)}{}_k\right)p\left(x^{(i)}{}_k\mid x^{(i)}{}_{k-1}\right)}{q(x^{(i)}{}_k\mid x^{(i)}{}_{k-1},y_k)}$$

得到：

$$W_k^{(i)}\propto W_{k-1}^{(i)}p\left(y_k\mid x^{(i)}{}_k\right)$$

由之前的重采样我们知道，实际上每次重采样以后，有 $W_{k-1}^{(i)}=\dfrac{1}{N}$，所以：

$$W_k^{(i)}\propto p\left(y_k\mid x^{(i)}{}_k\right)$$

这其实很好理解，粒子 x_i 得到观测值 y 的几率越大，则 x_i 越接近真实状态，则其权重越大

8 粒子滤波的仿真

8.1 算法伪代码流程图

$$\left[\left\{x_k^{(i)}, w_k^{(i)}\right\}_{i=1}^{N}\right] = SIR\left[\left\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_{i=1}^{N}, y_k\right]$$

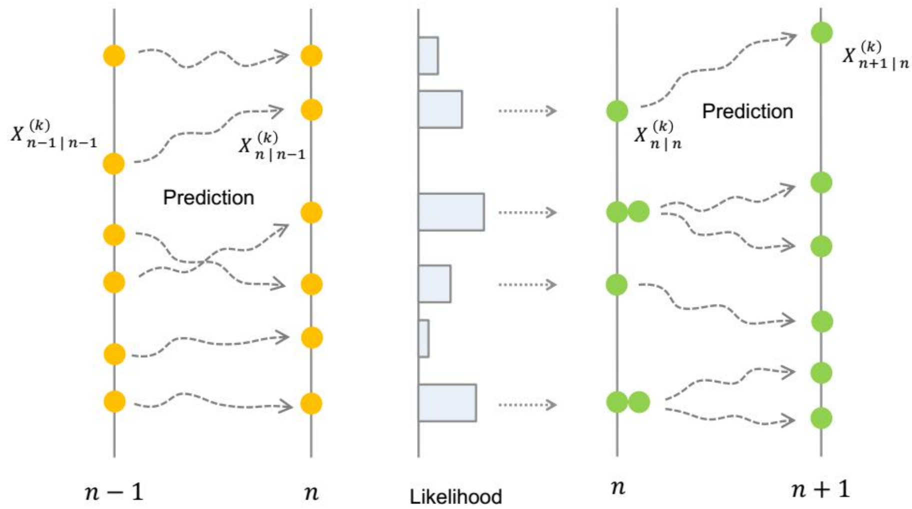
- FOR i = 1:N

  (1)采样粒子： $x_k^{(i)} \sim p\left(x_k | x_{k-1}^{(i)}\right)$

  (2)计算粒子的权重： $w_k^{(i)} = p\left(y_k | x_k^{(i)}\right)$

- END FOR
- 计算粒子权重和，t=sum(w)
- 对每个粒子，用上面的权重和进行归一化,w = w/t
- 粒子有了，每个粒子权重有了，进行状态估计 $\sum_{i=1}^{N} \tilde{W}_k(x_k^{(i)}) f(x_k^{(i)})$
- 重采样

流程图：



8.2 仿真代码

以一维情况为例，分析 SIR 滤波特性：

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8cos(1.2(k-1)) + v_k$$

$$y_k = \frac{x_k^2}{20} + n_k$$

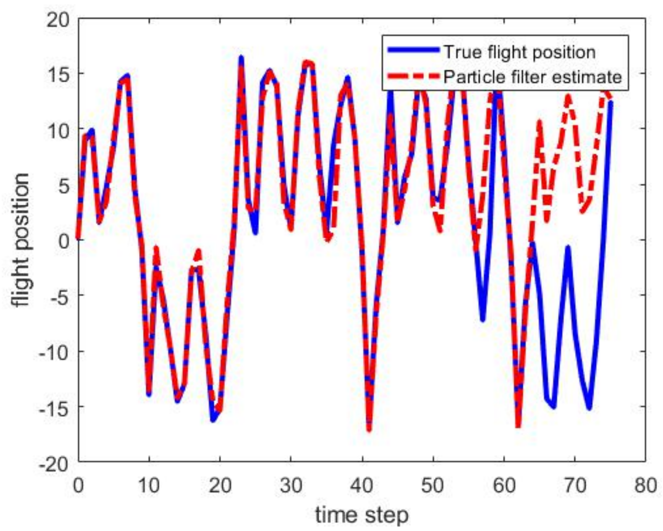
```
%% SIR 粒子滤波的应用
clear all
close all
clc
%% initialize the variables
x = 0.1; % 初始分布下的粒子状态
x_N = 1; % 过程噪声的协方差  (由于是一维的，这里就是方差)
x_R = 1; % 测量造成的协方差  (由于是一维的，这里就是方差)
T = 75;  % 共进行 75 次重复
N = 100; % 粒子数，越大效果越好，计算量也越大


V = 2; %初始分布的方差
x_P = []; % 粒子
% 用一个高斯分布随机的产生初始的粒子
for i = 1:N
    x_P(i) = x + sqrt(V) * randn;
end

z_out = [x^2 / 20 + sqrt(x_R) * randn];  %实际测量值向量
x_out = [x];  %实际状态值向量
x_est = [x]; % time by time output of the particle filters estimate
x_est_out = [x_est]; % the vector of particle filter estimates.

for t = 1:T
    x = 0.5*x + 25*x/(1 + x^2) + 8*cos(1.2*(t-1)) +  sqrt(x_N)*randn;
    z = x^2/20 + sqrt(x_R)*randn;
    for i = 1:N
        %从先验 p(x(k)|x(k-1))中采样
        x_P_update(i) = 0.5*x_P(i) + 25*x_P(i)/(1 + x_P(i)^2) +
8*cos(1.2*(t-1)) + sqrt(x_N)*randn;
        %计算采样粒子的值，为后面根据似然去计算权重做铺垫
        z_update(i) = x_P_update(i)^2/20;
        %对每个粒子计算其权重，这里假设量测噪声是高斯分布。所以 w = p(y|x)对应下
面的计算公式
        P_w(i) = (1/sqrt(2*pi*x_R)) * exp(-(z - z_update(i))^2/(2*x_R));
    end
    % 归一化.
    P_w = P_w./sum(P_w);

    for i = 1 : N
        x_P(i) = x_P_update(find(rand <= cumsum(P_w),1));   % 粒子权重大的
将多得到后代
    end                                                   % find( ,1)返回第一个
```

符合前面条件的数的下标

```
    %状态估计，重采样以后，每个粒子的权重都变成了 1/N
    x_est = mean(x_P);

    % 将数据存储在向量中
    x_out = [x_out x];
    z_out = [z_out z];
    x_est_out = [x_est_out x_est];

end

t = 0:T;
figure(1);
clf
plot(t, x_out, '.-b', t, x_est_out, '-.r','linewidth',3);
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('time step'); ylabel('flight position');
legend('True flight position', 'Particle filter estimate');
```

8.3 仿真结果

10 个粒子仿真（N=10）：



100 个粒子仿真结果（N=100）：