

信号大作业选题——粒子滤波

1. 粒子滤波

粒子滤波：通过寻找一组在状态空间中传播的随机样本来近似的表示概率密度函数，用样本均值代替积分运算，进而获得系统状态的最小方差估计的过程，这些样本被形象的称为“粒子”，故而叫粒子滤波。

粒子滤波(PF: Particle Filter)的思想基于蒙特卡洛方法(Monte Carlo methods)，它是利用粒子集来表示概率，可以用在任何形式的状态空间模型上。其核心思想是通过从后验概率中抽取的随机状态粒子来表达其分布，是一种顺序重要性采样法(Sequential Importance Sampling)。简单来说，粒子滤波法是指通过寻找一组在状态空间传播的随机样本对概率密度函数进行近似，以样本均值代替积分运算，从而获得状态最小方差分布的过程。这里的样本即指粒子，当样本数量 $N \rightarrow \infty$ 可以逼近任何形式的概率密度分布。

2. 粒子滤波的matlab实现

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，这些采样就称之为粒子。那么这些采样在状态空间中的分布实际上就是 $x(t-1)$ 的概率分布了。好，接下来依据状态转移方程加上控制量可以对每一粒子得到一个预测粒子。

校正阶段：观测值 y 到达后，利用观测方程即条件概率 $P(y|x_i)$ ，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。如此这般下来，对所有粒子都进行这么一个评价，那么越有可能获得观测 y 的粒子，当然获得的权重越高。

重采样算法：去除低权值的粒子，复制高权值的粒子。所得到的当然就是我们说需要的真实状态 $x(t)$ 了，而这些重采样后的粒子，就代表了真实状态的概率分布了。下一轮滤波，再将重采样过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。

初始状态的问题：由于开始对 $x(0)$ 一无所知，所有我们可以认为 $x(0)$ 在全状态空间内平均分布。于是初始的采样就平均分布在整个状态空间中。然后将所有采样输入状态转移方程，得到预测粒子。然后再评价下所有预测粒子的权重，当然我们在整个状态空间中只有部分粒子能够获的高权值。最后进行重采样，去除低权值的，将下一轮滤波的考虑重点缩小到了高权值粒子附近。

具体过程：

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，程序计算跟踪目标的特征，比如可以采用目标的颜色特征。具体到Rob Hess的代码，开始时需要人工用鼠标拖动出一个跟踪区域，然后程序自动计算该区域色调(Hue)空间的直方图，即为目标的特征。直方图可以用一个向量来表示，所以目标特征就是一个 $N \times 1$ 的向量 V 。

2) 搜索阶段-放狗

好，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particle。狗有很多种放法。比如，a)均匀的放：即在整个图像平面均匀的撒粒子(uniform distribution)；b)在上一帧得到的目标附近按照高斯分布来放，可以理解成，靠近目标的地方多放，远离目标的地方少放。Rob Hess的代码用的是后一种方法。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 $\text{sum}(\text{abs}(V_i - V))$ 。每条狗算出相似度后再做一次归一化，使得所有的狗得到的相似度加起来等于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 $X = \text{sum}(X_n W_n)$, $Y = \text{sum}(Y_n W_n)$ 。

4) 重采样阶段Resampling

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Sampling Importance Resampling，根据重要性重采样(更具重要性重新放狗)。

(2)->(3)->(4)->(2)如是反复循环，即完成了目标的动态跟踪。

```
clc;
clear all;
close all;
x = 0; %初始值
R = 1;
Q = 1;
tf = 100; %跟踪时长
N = 100; %粒子个数
P = 2;
xhatPart = x;
for i = 1 : N
    xpart(i) = x + sqrt(P) * randn; %初始状态服从0均值，方差为sqrt(P)的高斯分布
end
xArr = [x];
yArr = [x^2 / 20 + sqrt(R) * randn];
xhatArr = [x];
PArr = [P];
xhatPartArr = [xhatPart];
for k = 1 : tf
    x = 0.5 * x + 25 * x / (1 + x^2) + 8 * cos(1.2*(k-1)) + sqrt(Q) * randn;
    %k时刻真实值
    y = x^2 / 20 + sqrt(R) * randn; %k时刻观测值
    for i = 1 : N
        xpartminus(i) = 0.5 * xpart(i) + 25 * xpart(i) / (1 + xpart(i)^2) ...
            + 8 * cos(1.2*(k-1)) + sqrt(Q) * randn; %采样获得N个粒子
        ypart = xpartminus(i)^2 / 20; %每个粒子对应观测值
        vhat = y - ypart; %与真实观测之间的似然
        q(i) = (1 / sqrt(R) / sqrt(2*pi)) * exp(-vhat^2 / 2 / R);
        %每个粒子的似然即相似度
    end
    qsum = sum(q);
    for i = 1 : N
        q(i) = q(i) / qsum; %权值归一化
    end
    for i = 1 : N %根据权值重新采样
        u = rand;
```

```

qtempsum = 0;
for j = 1 : N
    qtempsum = qtempsum + q(j);
    if qtempsum >= u
        xpart(i) = xpartminus(j);
        break;
    end
end
end
xhatPart = mean(xpart);
%最后的状态估计值即为N个粒子的平均值，这里经过重新采样后各个粒子的权值相同
xArr = [xArr x];
yArr = [yArr y];
% xhatArr = [xhatArr xhat];
PArr = [PArr P];
xhatPartArr = [xhatPartArr xhatPart];
end
t = 0 : tf;
figure;
plot(t, xArr, 'b-.', t, xhatPartArr, 'k-');
legend('Real Value','Estimated Value');
set(gca,'FontSize',10);
xlabel('time step');
ylabel('state');
title('Particle filter')
xhatRMS = sqrt((norm(xArr - xhatArr))^2 / tf);
xhatPartRMS = sqrt((norm(xArr - xhatPartArr))^2 / tf);
figure;
plot(t,abs(xArr-xhatPartArr),'b');
title('The error of PF');

```