Bayes Filter ,Kalman Filter ,Extended Kalman Filter， Particle Filter 总结

贝叶斯滤波（BF），卡尔曼滤波（KF），扩展卡尔曼滤波（EKF）等均属于动态模型（Dynamic Model）中的连续分布问题， 动态模型离散模型有隐马尔科夫链(HMM)。所有动态模型基于两个假设：1，当前时刻状态只与上一时刻有关。2，当前时刻观测只与当前的状态有关。
HMM 可应用于自然语言处理(NLP)，在 HMM 的基础上发展出循环神经网络（RNN），多用于语音语义识别，以及经典隐马尔科夫混合高斯-梅尔频率倒谱系数（HMM-GMM-MFCC）用于语音语义识别（初代 Siri）
在计算机视觉（CV）方向，常用 KF，EKF 进行多目标追踪。
在 SLAM 方向，最早的 SLAM 是基于滤波，例如 gmapping。现在 SLAM 多基于图优化。
在多传感器融合方向，最初 KF 提出需要解决的问题就是阿波罗项目的多传感器融合定位问题。

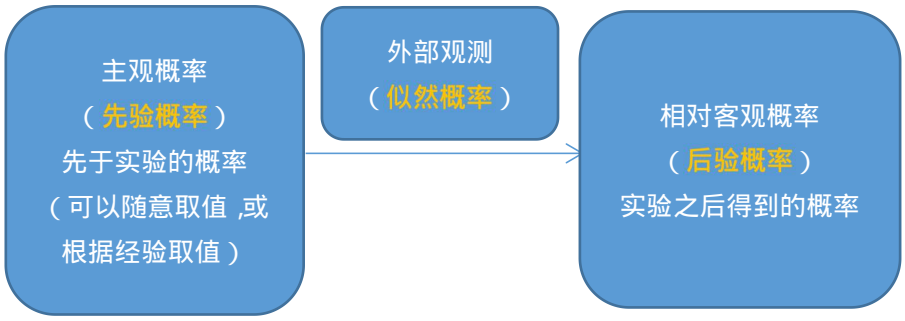
该部分为机器人状态估计的核心内容

建议先读《概率论与数理统计》

1. 贝叶斯滤波 Bayes Filter

贝叶斯滤波是基于贝叶斯估计的一种对随机信号进行处理，减少其不确定性的一种方法。

贝叶斯估计引入外部观测（证据，观测）对主观概率进行修正，得到一个相对客观的概率



先验概率，似然概率（传感器的精度），以及后验概率是贝叶斯滤波的三个主要概率

贝叶斯估计：

离散情况下

其公式为 P（X|Y）=$\frac{P(Y|X)P(X)}{P(Y)}$（贝叶斯公式），其中 P（X|Y）为后验概率， P（Y|X）为似然概率， P（X）为先验概率。其中 P（Y）为在所有 X 的情况下 Y 发生的概率，即为 $\sum_X P(Y|X)P(X)$，在模型已定的情况下为一定值。

eg1：一个温度的例子，预测今天多少度

首先，先验概率分布（先验可以随便猜测）P（T=10）=0.8，（温度等于 10 度的概率为 0.8），P（T=11）=0.2（温度等于 11 度的概率为 0.2）。

其次，温度计测量值为 Tm=10.3（温度计测量是有误差的）。

后验概率可以写为

$$P(T=10|Tm=10.3) = \frac{P(Tm=10.3|T=10)P(T=10)}{P(Tm=10.3)}$$和

$$P(T=11|Tm=10.3) = \frac{P(Tm=10.3|T=11)P(T=11)}{P(Tm=10.3)}$$

其中 P（Tm=10.3）=P（Tm=10.3|T=10）P(T=10)+P（Tm=10.3|T=11）P(T=11)

重要一点 P（Tm=10.3）跟 T 的取值无关，跟 T 的分布率有关

最终的温度的预测结果为 T=10*P(T = 10|Tm = 10.3)+11*P(T = 11|Tm = 10.3)。

离散的贝叶斯通用公式可写为
后验概率=η * 似然概率 * 先验概率
其中η =1/（$\sum$ *似然概率* ∗ *先验概率*）。

连续情况下

P（X<x|Y=y）=$\frac{P(Y=y|X<x)P(X<x)}{P(Y=y)}$，此时 P 是概率，X<x 等同于概率密度函数（pdf）从零进行积分到 x，此时，P（Y=y）为零（原因跟数轴取一个有理数的概率为零的原因是一样的）。求 P（X<x|Y=y）就要用到化积分为求和，尽量离散化。P（X<x|Y=y）=$\sum_{u=-\infty}^{x} P(X=u|y)$，然后类似与积分，加小变量进行积分，就可以积分得到$\int_{-\infty}^{x} \frac{f_{Y|X}(y|x)fx(x)}{f_Y(y)}dx$,（积分过程略），f 为概率密度函数，其中 P（X<x|Y=y） = $\int_{-\infty}^{x} f_{X|Y}(x|y)dx$，概率密度函数积分得概率，则有$\int_{-\infty}^{x} f_{X|Y}(x|y)dx$=$\int_{-\infty}^{x} \frac{f_{Y|X}(y|x)fx(x)}{f_Y(y)}dx$，即为连续贝叶斯公式
其中$f_Y(y)$=η ∗ $f_{Y|X}(y|x)f(x)$（类比离散情况下）η =1/($\int_{-\infty}^{+\infty} f_{Y|X}(y|x)f(x)dx$)
（积分推导过程略，公式比较难打）

eg2：还是温度的例子，前面是二项分布为离散，现在是假设为高斯分布 f（x）~N（10，1^2）均值为 10，方差为 1 的高斯分布
假设温度计精度为±0.2 度 ,假设观测值为 9 ,则似然概率的高斯分布$f_{X|Y}$（y|x）~N（9，0.2^2）则根据上面积分过程可得后验概率分布$f_{X|Y}$~N（9.0385， 0.038^2）,计算过程设计多维高斯分布模型。从结果来看后验分布的方差明显降低，这就是贝叶斯滤波的效果，方差降低，置信度升高。现在的预测温度就是后验分布的期望。

以上就是为什么要用滤波，让两个方差较大的分布，融合成一个方差较小的分布

贝叶斯滤波

上面描述了什么是贝叶斯估计，基于贝叶斯估计滤波的方法为贝叶斯滤波

假设一个随机机过程

$$x_0 -> x_1 -> x_2 -> \dots x_k$$

y1 是对 x1 的观测，y2 是对 x2 的观测等

如果每一个先验信息都是靠猜测的话，就会过于依赖观测，放弃了预测信息。所以就利用所有的预测信息，只有 x0 是猜测的，x1，x2....等通过递推公式来计算，这样就是贝叶斯滤波。就需要马尔科夫的假设，即 1，当前时刻状态只与上一时刻有关。2，当前时刻观测只与当前的状态有关。
状态方程 Xk=f(X(k-1))+Qk (反映状态变化的方程，Qk 为噪声分布)
观测方程 Yk=h(Xk)+Rk (反映观测的方程，Rk 为噪声分布)(Xk 本应该是真值，会更精准，但是实际执行中没有真值，当前时刻的最有可能的真值是你的预测值，所以 Xk 也可以是预测值，即为状态方程推出的当前时刻的状态)

所以贝叶斯滤波分为两步
预测步，根据状态方程，上一时刻的后验->这一时刻的先验
更新步，根据观测方程，这一时刻的先验->这一时刻的后验
预测步需积分，跟贝叶斯估计类似，积分过程略，或参考《概率论与数理统计》

结果为 f（x）- = $\int_{-\infty}^{+\infty} f_Q[u-f(v)]f(v)dv$, v 为上一时刻的后验

更新步 f（x）+ = $\eta f_R[y-f(x)]f(x)-$
f（x）- 为先验，f（x）+ 为后验
$\eta= 1/\int_{-\infty}^{+\infty} f_R[y-f(x)]f(x)-dx$
期望（即结果）x=$\int_{-\infty}^{+\infty} x$f（x）+ dx

贝叶斯滤波到此结束，两个公式，但是涉及三个无穷积分，大多时候无解析解

所以在贝叶斯滤波的基础上添加限制条件 ,状态方程和观测方程都为线性 ,且 Q ,R 为正态分布，即为卡尔曼滤波（KF），状态方程和观测方程都为非线性，且 Q，R 为正态分布，扩展卡尔曼

滤波（EKF）和无迹卡尔曼滤波（UKF）。直接对无穷积分有两种方法，一是基于蒙特卡罗积分，即采样，以样本来反映整体分布，即粒子滤波（particle filter）。一种是基于直方图形式，类似微分相加，即为直方图滤波（Histogram filter）

2. 卡尔曼滤波

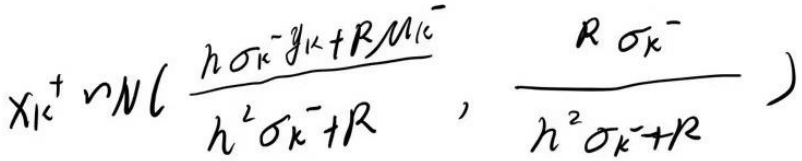
卡尔曼滤波的限制条件状态方程和观测方程都为线性，且 Q，R 为正态分布
观测方程 f（xk）=F*f（xk-1），观测方程 h（xk）=H*xk
Q~N（0，Q），R~N（0，R）

预测方程根据 f（x）- = $\int_{-\infty}^{+\infty} f_Q[u-f(v)]f(v)dv$，进行积分，可得先验分布 N~（Fxk-1，F^2σk-1 + Q）

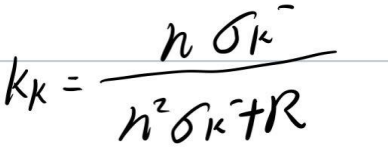
积分过程过于复杂，不在打出来，提供三种方法计算

- 通过复变函数，用留数定理计算（大学课程复变函数有讲）
- 通过傅里叶变换和卷积计算（大学课程信号处理和系统控制都有讲）
- Matlab 计算（简单粗暴）

更新步根据 f（x）+ = η$f_R[y-f(x)]f(x)$ −和η= $1/\int_{-\infty}^{+\infty} f_R[y-f(x)]f(x)-dx$
计算可得后验分布 N~（ ），不好打，看图片



其中可以把可以提出均值和方差的公因式，即为卡尔曼增益

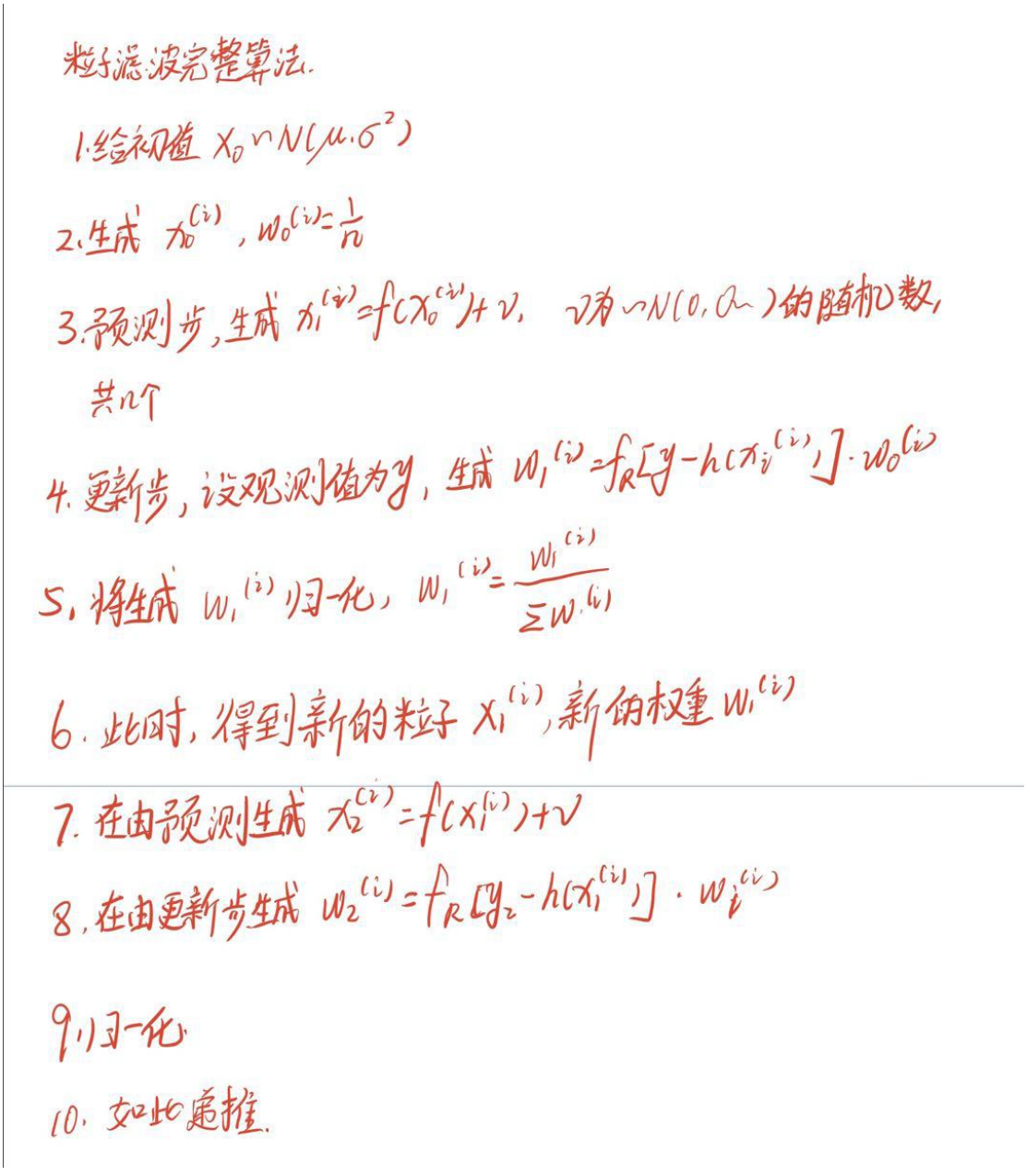


所以卡尔曼的五个公式，本来只有四个，即先验的均值，以及协方差，后验的均值，以及协方差。然后从后验均值和协方差里面提取一个公因式即第五个公式，卡尔曼增益

3. 粒子滤波

粒子滤波的主要思想，是通过粒子的位置，和粒子的权重来反映样本的分布。

同样要从 f (x) + = η$f_R[y-f(x)]f(x)$ −和η= $1/\int_{-\infty}^{+\infty} f_R[y-f(x)]f(x)-dx$这两个函数，来推导。其中要用到狄拉克函数（即脉冲信号）（信号处理和数理统计中有讲）
积分过程中，可以推到出权重的递推公式。过程更加复杂，算了，不打了。



最后粒子的位置乘以权重，然后相加，就的结果。

粒子滤波到此算法完结。

但是，此时粒子滤波有缺陷，就是位置好的粒子权重接近于 1，而其他的粒子权重接近于 0。就会导致粒子多样性减少，类似于遗传算法（GA）的多样性减少，处理办法遗传算法（GA）中轮盘重采样法（智能计算与应用中有涉及）

第一次采样的方法，也可以用轮盘采样法，亦可以用接受-拒绝采样法（查资料，不在过多解释）

预测方程的离散化，可以采用欧拉法，改进欧拉法，以及龙格库塔法（详细，以后在说，不在本次滤波范围内）

4. 扩展卡尔曼滤波

在卡尔曼滤波的基础上，把非线性的预测方程，和观测方程泰勒展开，变成线性，带入计算积分即可，计算过程（略）

5. Code Demo

```
%卡尔曼滤波
%X(K) = F*X(K-1)+Q
%Y(K) = H*X(K)+R

%生成一段时间 t
t=0.1:0.01:1;
L=length(t);
%生成真实信号 x，以及观测 y
%首次初始化
x=zeros(1,L);
y=x;
for i=1:L
    x(i)=t(i)^2;
    y(i)=x(i)+normrnd(0,0.1);
    y(i)=x(i)+normrnd(0,0.1);
end
%生成信号完成

%滤波算法

%观测方程 Y(K)=X(K)+R R~N(0,1)
%预测方程
%模型一，
%X(K)=X(K-1)+Q
%Y(K)=X(K)+R
%Q~N(0,1)
F1 = 1;
H1 = 1;
Q1 = 0.1;
R1 = 1;
%初始化 X(k)+
Xplus1=zeros(1,L);
```

```
%设置一个初始值，假设 Xplus1(1)~N(0.01, 0.01^2)
Xplus(1)=1;
Pplus1=0.01^2;
%%%%卡尔曼滤波
%X(K)minus=F*X(K-1)plus
%P(K)minus=F*P(K-1)plus*F'+Q
%K=P(K)minus*H'*inv(H*P(K)minus*H'+R)
%X(K)plus=X(K)minus+K(y(k)-H*X(K)minus)
%P(K)plus=(1-K*H)*P(K)minus

for i = 2:L
    %%%预测步
    Xminus1=F1*Xplus1(i-1);
    Pminus1=F1*Pplus1*F1'+Q1;
    %%%更新步
    K1=(Pminus1*H1')/(H1*Pminus1*H1'+R1);
    Xplus1(i)=Xminus1+K1*(y(i)-H1*Xminus1);
    Pplus1=(1-K1*H1)*Pminus1;
end

%plot(t,x,'r',t,y,'g',t,Xplus1,'b','LineWidth',2);

%模型二，
%X(K)=X(K-1)+X'(K-1)*dt+X''(K-1)*dt^2*(1/2!)+Q2
%Y(K)=X(K)+R R~N(0,1)
%Q~N(0,1)
%此时状态变量 X=[X(K) X'(K) X''(K)]T(列向量)
%F = 1  dt  0.5dt^2
%    0, 1   dt
%    0, 0,  1
%H=[1, 0, 0]
%Q = Q2, 0, 0
%    0, Q3, 0
%    0, 0, Q4
dt=t(2)-t(1);
F2 = [1,dt,0.5*dt^2;0,1,dt;0,0,1];
H2 = [1, 0, 0];
Q2 = [0.2, 0, 0;0, 0.1, 0;0, 0, 0.01];
R2 = 5;

Xplus2=zeros(3, L);
Xplus2(1,1)=0.1^2;
Xplus2(2,1)=0;
Xplus2(3,1)=0;
Pplus2=[0.01, 0, 0;0, 0.01, 0; 0, 0, 0.001];

for i=2:L
    Xminus2=F2*Xplus2(:,i-1);
```

```
    Pminus2=F2*Pplus2*F2'+Q2;

    K2=(Pminus2*H2')/(H2*Pminus2*H2'+R2);
    Xplus2(:,i)=Xminus2+K2*(y(i)-H2*Xminus2);
    Pplus2=(eye(3)-K2*H2)*Pminus2;
end


%粒子滤波

%预测方程 x(i)=sin(x(i-1))+5*x(i-1)/(x(i-1)^2+1)+Q
%观测方程 y(i)=x(i)^2+R

t=0.01:0.01:1;
x=zeros(1,100);
y=zeros(1,100);

x(1)=0.1;
y(1)=0.01^2;

for i=2:100
    x(i)=sin(x(i-1))+5*x(i-1)/(x(i-1)^2+1);
    y(i)=x(i)^2+normrnd(0,1);
end
%plot(t,x,t,y,'LineWidth',2)

n=100;
xold=zeros(1,n);
xnew=zeros(1,n);
xplus=zeros(1,n);%xplus 用于存放滤波值，就是每一次后验概率的期望
w=zeros(1,n);
%设置 x0(i),可以直接在正态分布采样，如果对初值有信心，也可以让所有粒子都相同
for i=1:n
    xold(i)=0.1;
    w(i)=1/n;
end

for i=2:100

    %预测步
    for j = 1:n
        xold(j)=sin(xold(j))+5*xold(j)/(xold(j)^2+1)+normrnd(0,0.1) %Q;
    end

    %更新步
    for j = 1:n
        %w(j)=w(j)*fR(y-观测)
        %正态分布
        %fR=(2*pi*R)^(-0.5)*exp(-((y(i)-xold(j)^2)^2/(2*R)))
```

```
        %w(j)=(2*pi*R)^(-0.5)*exp(-((y(i)-xold(j)^2)^2/(2*R)))*w(j);
        w(j)=exp(-((y(i)-xold(j)^2)^2/(2*0.1))); %R
    end

    %归一化
    w=w/sum(w);
    %w/sum（w）与 k*w/sum（k*w）结果一模一样
    %(2*pi*R)^(-0.5)是常数
    %w(j),如果每次都重采样，每次 w(j)都会被设为 1/n,也是常数
    %所以可以将他们去掉

    %重采样
    %N<1/sum(wi^2),若不是每次都重采样的化，那么权重更新就要把 w(j)乘上去
    %生成数组 c
    c=zeros(1,n);
    c(1)=w(1);
    for j=2:n
        c(j)=c(j-1)+w(j);
    end

    %首先要重采样 n 个粒子，粒子数要跟之前相同
    for j = 1:n
        a = unifrnd(0,1);
        for k=1:n
            if(a<c(k))
                xnew(j)=xold(k);
                break;
            end
        end
    end

    xold = xnew;
    %权重都重新设为 1/n，只有重采样得时候有这一步

    for j=1:n
        w(j)=1/n;
    end
    %把每一步的后验概率期望赋值跟 xplus
    xplus(i)=sum(xnew)/n;

end

plot(t,x,'r',t, xplus, 'b', 'LineWidth',2)

%y=x^2+R 似然概率是一个多峰分布，y=4，x 可能为 2，或者-2
%如果问题本身的性质就是强烈的非线性，比如多峰分布这种，粒子滤波也不行
%粒子滤波的计算速度有问题，而且占很多内存
```

```
%扩展卡尔曼滤波
%x(k)=sin(3*x(k-1))
%y(k)=x(k)^2
%多峰分布，非常强的非线性

t=0.01:0.01:1;
n=length(t);
x=zeros(1,n);
y=zeros(1,n);
x(1)=0.1;
y(1)=0.1;
for i=2:n
    x(i)=sin(3*x(i-1));
    y(i)=x(i)^2+normrnd(0, 0.2);
end

Xplus=zeros(1,n);
Pplus=0.1;
Xplus(1)=0.1;
Q=0.0001;
R=1;

for i=2:n
    %预测步
    A=3*cos(3*Xplus(i-1));
    Xminus=sin(3*Xplus(i-1));
    Pminus=A*Pplus*A'+Q;
    %更新步
    C=2*Xminus;
    K=(Pminus*C)/(C*Pminus*C'+R);
    Xplus(i)=Xminus+K*(y(i)-Xminus^2);
    Pplus=(eye(1)-K*C)*Pminus;
end

plot(t,x,'r',t,Xplus,'b','LineWidth',4)
```

6. reference

1. 《概率论与数理统计》 茆诗松
2. 《捷联惯导算法与组合导航原理》严恭敏
3. 《State Estimation for Robotics》 Timothy D. Barfoot
4. Multi-sensor optimal information fusion Kalman filter Shu-Li Sun