Bayesian Estimation for Tracking Multiple Objects: Sequential Monte Carlo Methods

Bahman Moraffah

June 2019
ASU-EECE-06-2019

School of Electrical, Computer, and Energy Engineering
Arizona State University
Tempe, AZ 85287

Abstract

This report surveys the main methods for estimation of dynamic parameters that are embedded in stochastic processes using popular methods such as the linear Kalman filter and suboptimal nonlinear extensions such as the extended and unscented Kalman filter. We will also examine the sequential importance sampling, resampling, and auxiliary particle filter. We survey the hidden Markov chain and outline its properties. A worked out example for each topic is discussed and the results are analyzed for performance.

Contents

List of Figures

1 Introduction

In many engineering applications, problem of dynamic system that is embedded in noise appears. In Bayesian statistics, to analyze a dynamic system we construct a probability density function of the state variables based on the physical model and noise. Then we estimate the state according to a tail recursion. To recursively do Bayesian inference, estimation of the state follows two core steps: (A) the prediction step in which we predict the state distribution given the observations up to that time and (B) Update step in which we update our prediction upon receiving new observations. Suppose $K(x_{k-1}, x_k)$ is a probability transition kernel following the physical system and the likelihood probability $p(z_k|x_k)$ which determines the probability of generating the measurement z_k conditioned on knowing the state x_k. Given the set of measurements collected up to time $(k-1)$, $Z_{k-1} = \{z_1, z_2, ..., z_{k-1}\}$, the Chapman-Kolmogorov equation[referecne] provides

$$p(x_k|Z_{k-1}) = \int_{x_{k-1}} p(x_{k-1}|Z_{k-1})K(x_{k-1}, x_k)dx_{k-1} \tag{1}$$

Equation 1 is called prediction equation. Upon receiving the measurement z_k at time k, Bayes rule leads to the update equation:

$$p(x_k|Z_k) \propto p(z_k|x_k)p(x_k|Z_{k-1}) \tag{2}$$

where the proportionality constant equals

$$\int_{x_k} p(z_k|x_k)p(x_k|Z_{k-1})dx_k. \tag{3}$$

However, analytical computation of the integral in equation 3 is almost always intractable. In this report, we investigate special cases that analytically computing this integral is possible. Furthermore, we generalize to methods such that this integral is numerically approximated. We provide simulations to support the introduced methods.

2 Recursive Bayesian Estimation

2.1 Kalman Filter

Suppose we have a linear dynamic system with Gaussian noise. It is shown [1, 2] that in this case thew Bayesian inference introduces in section 1 is tractable and can analytically be computed. This dynamic system first studied by Kalman in 1973. Here we detail this model,

Let x_k and $\hat{x}_k$ be the state and its estimate, respectively. It turns out there a recursive algorithm known as Kalman Filter to estimate $\hat{x}_k$. As any Bayesian inference modeling, this method contains two main steps; prediction and update. To do this end, the following linear dynamic system is considered

$$x_k = A_{k-1}x_{k-1} + u_k \tag{4}$$

$$z_k = B_k x_k + v_k \tag{5}$$

where equations 4 and 5 are known as motion and measurement equations, respectively. An i.i.d. Gaussian noise is assumed, that is, $u_k \sim \mathcal{N}(\mu_{u,k}, \Sigma_{u,k})$ and $v_k \sim \mathcal{N}(\mu_{v,k}, \Sigma_{v,k})$. For simplicity, we assume that the matrices A_k, B_k do not vary with time, that is, $A_k = A$ and $B_k = B$. Without loss of generality, we assume $\mu_{u,k} = 0$ and $\mu_{v,k} = 0$, and assume that covariance matrices are stationary in time meaning $\Sigma_{u,k} = \Sigma_u$ and $\Sigma_{v,k} = \Sigma_v$. The prediction and update distributions follow

$$\text{Predictive distribution:} \qquad x_k|Z_{k-1} \sim \mathcal{N}(x_k; \hat{x}_{k|k-1}, \hat{\Sigma}_{k|k-1}) \tag{6}$$

$$\text{Update distribution:} \qquad x_k|Z_k \sim \mathcal{N}(x_k; \hat{x}_{k|k}, \hat{\Sigma}_{k|k}) \tag{7}$$

where $\hat{x}_{k|k-1}$ and $\hat{x}_{k|k}$ are the prediction estimate and update estimate upon receiving the measurements at time k and equal

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} \\ \hat{\Sigma}_{k|k-1} &= \sigma_u + A\hat{\Sigma}_{k-1|k-1}A^T \\ \hat{x}_{k|k} &= W_k(z_k - B\hat{x}_{k|k-1}) \\ \hat{\Sigma}_{k|k} &= \hat{\Sigma}_{k|k-1} - W_k B\hat{\Sigma}_{k|k-1} \end{aligned} \tag{8}$$

where $W_k = \hat{\Sigma}_{k|k-1}B^T(B\hat{\Sigma}_{k|k-1}B^T + \Sigma_v)^{-1}$.

It is shown [1, 2] that the Kalman filter is the optimal linear estimator meaning the Kalman filter minimizes the mean square error of the estimated parameters under when the linear state-space assumption. The Kalman filter is popular becasue:

- Kalman filter produces good results in practice due to optimality and structure.
- The process is real time.
- It is easy to implement and measurement equation does not need to be inverted.

- It has found many applications in radar tracking, robot localization, map building, and many more applications.

The main issues with Kalman filter is that what if the transitioning does not follow a linear dynamic system or what if the noise is not Gaussian. Nonlinear estimators such as extended Kalman filter, unscented Kalman filter are generalizations of the Kalman filter which are studied in detail in [3, 2]. Particle filter and Markov chain Monte Carlo methods are other methods of approximating the posterior distribution. We discuss Particle filter in detain in the next section.

2.1.1 Simulations: Kalman Filter Target Tracking

Consider the linear dynamic system in equations 4, 5 for a moving target with constant velocity and altitude. Let $x_k = [x, y, \dot{x}, \dot{y}]^T \in \mathbb{R}^4$ be the state vector at time k where indicates the x-position, y-position, x-velocity, y-velocity, respectively. Let $z_k \in \mathbb{R}^4$ be the noisy measurement at time k. Assume that motion equation follows

$$\begin{pmatrix} x_k \\ y_k \\ \hat{x}_k \\ \hat{y}_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \hat{x}_{k-1} \\ \hat{y}_{k-1} \end{pmatrix} + u_k \tag{9}$$

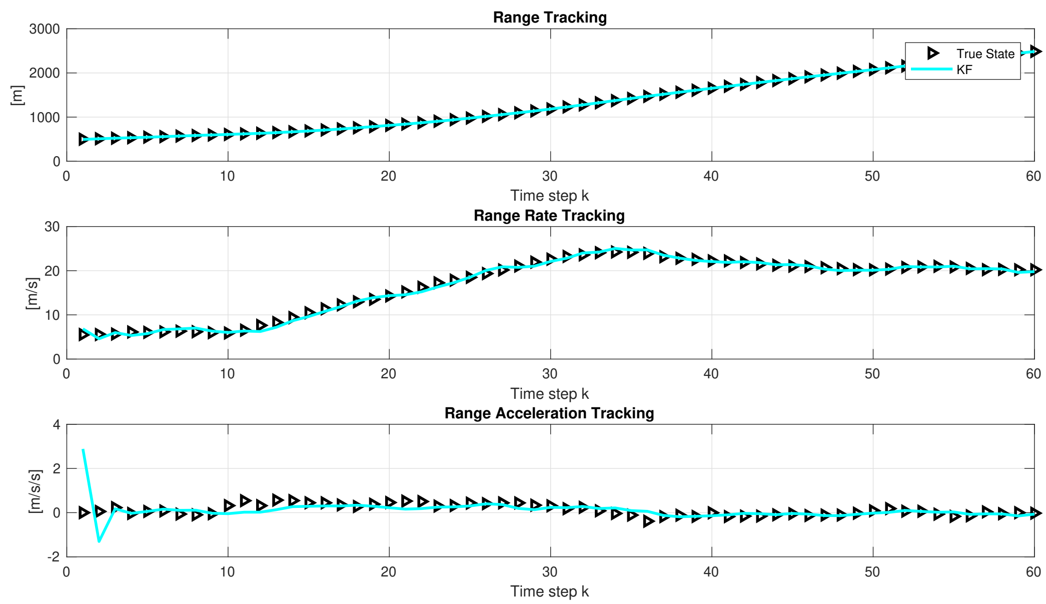

Figure 1: Estimating the location and velocity of the target using Kalman filter.

and the measurements equal

$$\begin{pmatrix} x_k \\ y_k \\ \hat{x}_k \\ \hat{y}_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \hat{x}_{k-1} \\ \hat{y}_{k-1} \end{pmatrix} + v_k \tag{10}$$

where $u_k \sim \mathcal{N}(0_{4\times 1}, \Sigma_u)$ and $v_k \sim \mathcal{N}(0_{4\times 1}, \Sigma_v)$ for $\Sigma_u = \mathrm{I}_4$ and

$$\Sigma_v = S \begin{pmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^2}{2} & T_s \end{pmatrix} \otimes \mathrm{I}_2$$

where S is the noise intensity, T_s is the sampling period, and $\otimes$ indicates the Kronecker product. Figure 1 shows the estimate using Kalman filter.

2.2 Point Mass Filter

Dynamic systems are often non-linear and hence the Kalman method does not work. In Bayesian inference, we need to compute the posterior distribution. However, computing the posterior can be intractable. Particle filter introduces a method to approximate the posterior distributions. The fundamental idea is to draw sufficiently large number of particles and approximate the probability density p by the point mass,

$$\hat{p}(x) = \sum_{i=1}^{N} w^i \delta_{x^i}(x) \tag{11}$$

where x^i is drawn from the proposal distribution q and $w^i = p(.)/q(.)$ is the corresponding weight. Note that δ is the Dirac function. We approximate the posterior distribution $p(x_k|Z_k)$ and $\hat{x}_k$ using the point mass method.

Assuming that

$$p(x_{k-1}|Z_{k-1}) = \sum_{i=1}^{N} w^i_{k-1|k-1} \delta_{x^i_{k-1}}(x_{k-1}). \tag{12}$$

Intuitively speaking, $w^i_{k-1|k-1} \approx p(x^i_{k-1}|z_k)$. Consider the dynamic system

$$\begin{aligned} x_k &= f(x_{k-1}) + u_k \\ z_k &= g(x_k) + v_k \end{aligned} \tag{13}$$

with f and g being two possibly nonlinear functions. Equation 13 gives the probability transition kernel $K(x_{k-1}, x_k) = p(x_k|x_{k-1})$ and the likelihood $p(z_k|x_k)$. the prediction equation 1 can be approximated by

$$\begin{aligned} p(x_k|Z_{k-1}) &\approx \int_{x_{k-1}} \sum_{i=1}^{N} w^i_{k-1|k-1} \delta_{x^i_{k-1}}(x_{k-1}) p(x_k|x_{k-1}) dx_{k-1} \\ &= \sum_{i=1}^{N} w^i_{k-1|k-1} p(x_k|x^i_{k-1}) \end{aligned} \tag{14}$$

and hence the prediction probability is approximated by

$$p(x_k|Z_{k-1}) = \sum_{i=1}^{N} w^i_{k|k-1}\delta_{x^i_k}(x_k) \tag{15}$$

for $i = 1, \dots N$ and

$$w^i_{k|k-1} = \sum_{i=1}^{N} w^i_{k-1|k-1}p(x_k|x^i_{k-1}).$$

The update equation 2, therefore, equals

$$p(x_k|Z_k) \propto p(z_k|x_k)\sum_{i=1}^{N} w^i_{k|k-1}\delta_{x^i_k}(x_k) \tag{16}$$

and the proportional constant $p(z_k|Z_{k-1})$ can be computed as

$$\begin{aligned} p(z_k|Z_{k-1}) &= \int p(z_k|x_k)p(x_k|Z_{k-1})dx_{k-1} \\ &\approx \int p(z_k|x_k)\sum_{i=1}^{N} w^i_{k|k-1}\delta_{x^i_k}(x_k)dx_{k-1} \\ &= \sum_{i=1}^{N} w^i_{k|k-1}p(z_k|x^i_k). \end{aligned} \tag{17}$$

Combining 16 and 17, we have

$$p(x_k|Z_k) = \sum_{i=1}^{N} w^i_{k|k}\delta_{x^i_k}(x_k) \tag{18}$$

where

$$w^i_{k|k} = \frac{w^i_{k|k-1}p(z_k|x^i_k)}{\sum_{j=1}^{N} w^j_{k|k-1}p(z_k|x^j_k)}.$$

The posterior mean gives us the $\hat{x}_k$,

$$\hat{x}_k = \mathbb{E}\big[p(x_k|Z_k)\big] \approx \sum_{i=1}^{N} w^i_{k|k}x^i_k. \tag{19}$$

2.3 Particle Filter

Using the point mass filter method we can write

$$p(x_k|Z_k) = \sum_{i=1}^{N} w^i_k\delta_{x^i_k}(x_k) \tag{20}$$

where

$$w_k^i = \frac{p(x_k^i|Z_k)}{q(x_k^i|Z_k)}.$$

We can recursively compute the weights as

$$w_k^i = w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} \tag{21}$$

and $w_0 = 1/N$ uniformly chosen. While $u_k \sim \mathcal{N}(\mu_u, \Sigma_u)$ and $v_k \sim \mathcal{N}(\mu_v, \Sigma_v)$, it is shown that the optimal proposal density gives

$$w_k^i = w_{k-1}^i p(z_k|x_k^i) \tag{22}$$

where the likelihood function is

$$p(z_k|x_k^i) \propto \exp\Big[-\frac{1}{2}(z_k - g(x_k^i))^T \Sigma_v^{-1}(z_k - g(x_k^i))\Big] \tag{23}$$

This is called bootstrap filter [4]. This algorithm is summarized in Algorithm 1. The

Algorithm 1 Bootstrap Algorithm

```
Input: z_k, {x_{k-1}^i, w_{k-1}^i}_{i=1}^N
Output: x_k, {x_k^i, w_k^i}_{i=1}^N
for i = 1, 2, ..., N do
    Draw x_k^i ~ p(x_k | x_{k-1}^i)
    Update w̃_k^i = w_{k-1}^i p(z_k | x_k^i)
L = Σ_{j=1}^N w̃_k^i
for i = 1, 2, ..., N do
    Normalize the weights w_k^i = L^{-1} w̃_k^i
Estimate x̂_k using 19
Return
```

variance of the estimate increases with N [5, 4]. A solution to this issue is resampling method. Intuitively speaking, assuming $\hat{p}$ to be the approximate of p. This approximation is based on the samples drawn from the proposal distribution q. However, these samples are not from the actual p. To make these samples as close as possible to the samples from p, we sample from $\hat{p}$ which is equivalently obtained by sampling with replacement from these samples. The resampling algorithm is summarized in Algorithm 2.

Combining the particle filter algorithm 1 and the resampling algorithm 2, the sequential importance resampling algorithm is obtained and summarized in Algorithm 3.

Algorithm 2 Resampling Algorithm

Input: $\{x_k^i, w_k^i\}_{i=1}^N$
Output: $\{\tilde{x}_k^i, \tilde{w}_k^i\}_{i=1}^N$
$\lambda_1 = w_k^1$
for $i = 2, \ldots, N$ **do**
    $\lambda_i = \lambda_{i-1} + w_k^i$
$\ell = 1$
Draw $u \sim Unif(0, 1/N)$
for $i = 1, 2, \ldots, N$ **do**
    $u_i = u_1 + \frac{(i-1)}{N}$
    while $u_i > \lambda_\ell$ **do**
        $\ell = \ell + 1$
    $\tilde{x}_k^i = x_k^\ell$
    $\tilde{w}_k^i = w_k^\ell$
Return

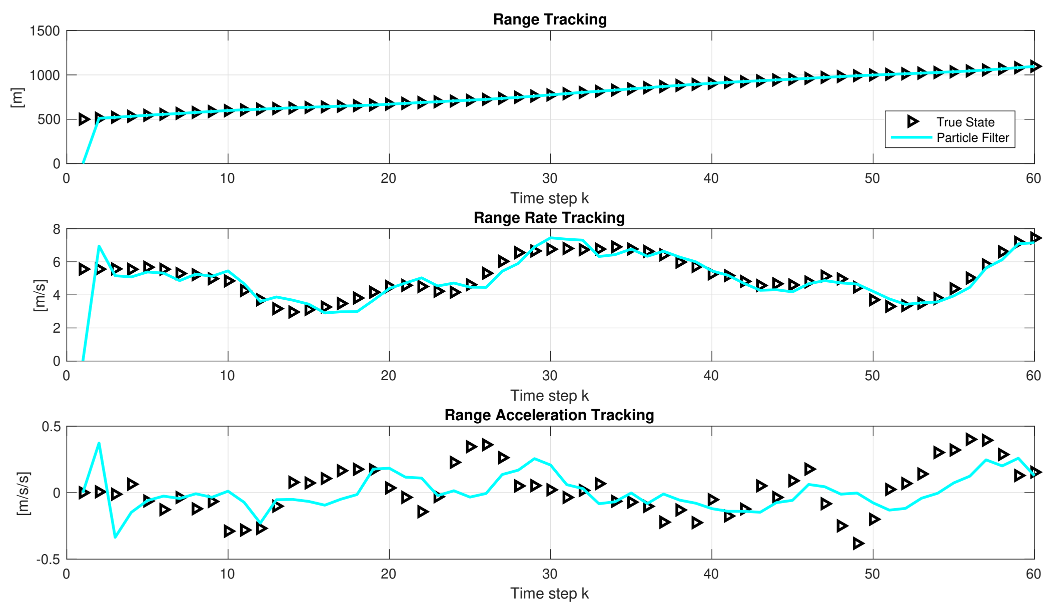

Figure 2: Estimating the location and velocity of the target using Particle filter method.

2.3.1 Simulations: Particle Filter Target Tracking

Applying the Particle algorithm in Algorithm 3 to the problem described in section 2.1.1. Figure 2 depicts this method.

The RMSE comparison between Kalman filter and Particle filter is provided and depicted in Figure 3.

Algorithm 3 Sequential Importance Resampling Algorithm

Input: $z_k, \{x_{k-1}^i\}_{i=1}^N$
Output: $x_k, \{x_k^i\}_{i=1}^N$
for $i = 1, 2, \ldots, N$ do
    Draw $x_k^i \sim p(x_k | x_{k-1}^i)$
    Update $\tilde{w}_k^i = w_{k-1}^i p(z_k | x_k^i)$
$L = \sum_{j=1}^{N} \tilde{w}_k^i$
for $i = 1, 2, \ldots, N$ do
    Normalize the weights $w_k^i = L^{-1} \tilde{w}_k^i$
Resample using Algorithm 2 to obtain $\{\tilde{x}_k^i, \tilde{w}_k^i\}_{i=1}^N$
Estimate $\hat{x}_k = \sum_{i=1}^{N} \tilde{w}_k^i \tilde{x}_k^i$
Return

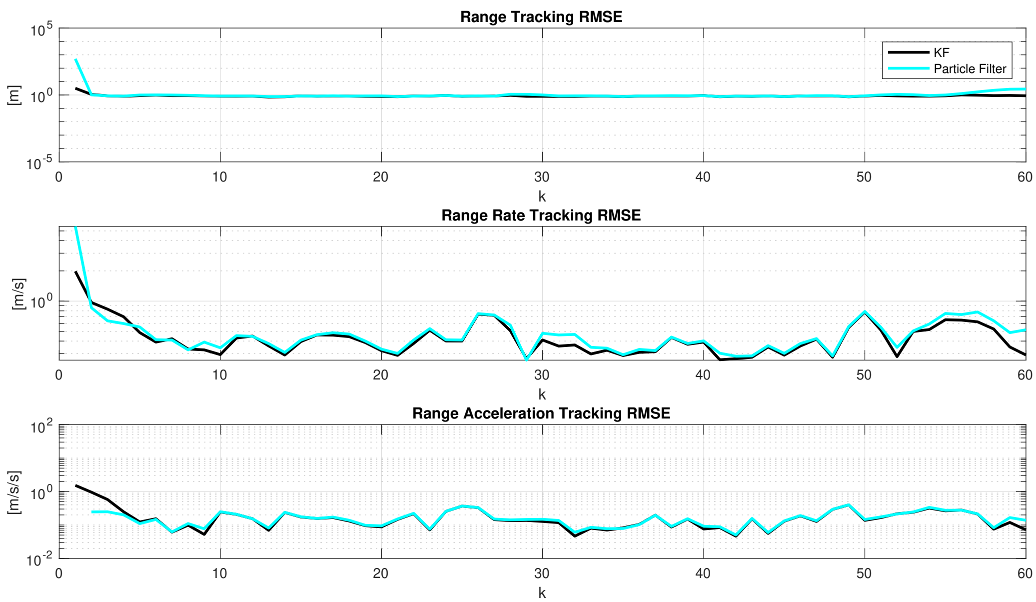

Figure 3: RMSE comparison between Kalman filter and Particle filter methods.

3 Hidden Markov Model

Hidden Markov models (HMMs) are widely used in many applications including pattern recognition, video processing, and tracking. In this section, we survey the HMM and their applications in machine learning and object tracking. We discuss the advantages and disadvantages of HMM [6, 7].
Definition: A Markov chain is a sequence of random variables such that the probability

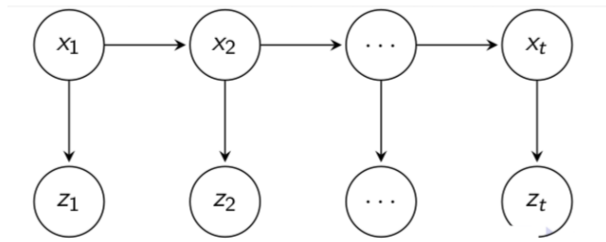

Figure 4: Hidden Markov model for observations $z_1, \ldots, z_t$ for latent variable $x_1, \ldots, x_t$.

distribution at time $(t+1)$ only depends on t, that is,

$$Pr(X_{t+1}|X_t, \ldots, X_1) = Pr(X_{t+1}|X_k) = K(X_t, X_{t+1}) \tag{24}$$

where $Pr(X_{t+1} = x_{t+1}|X_t = x_t) = K(x_t, x_{t+1})$ is the transition kernel, where

$$\sum_{x_{t+1}} K(x_t, x_{t+1}) = 1.$$

Definition: A hidden Markov model (HMM) is a sequence of random variables, $Z_1, \ldots, Z_t$ such that the distribution of Z_t depends only on the hidden state x_t of an associated Markov chain.

Let $\mathcal{X}$ and $\mathcal{Z}$ be the finite set of states and the fintie set of observations, respectively. For transition kernel $K : \mathcal{X} \times \mathcal{X} \to [0, 1]$ and $P : \mathcal{X} \times \mathcal{Z} \to [0, 1]$ is the observation probability. Assume that $\pi : \mathcal{X} \to [0, 1]$ is the prior distribution on the initial state x_0.

Given the Markovity of the states and observations depicted in Figure 4, we may write that joint distribution is of the form

$$p(x_0, x_1, \ldots, x_t, z_1, \ldots, z_t) = \pi(x_0)K(x_0, x_1) \prod_{i=1}^{t} K(x_i, x_{i+1})P(z_i|x_i) \tag{25}$$

Given Equation 25, we can do filtering and smoothing. In other words, we can compute how likely it is to observe measurements $z_1, \ldots, z_t$, i.e., $p(z_1, \ldots, z_t)$, or given a set of observations, how likely the current state is, i.e., $p(X_t = x_t|z_1, \ldots, z_t)$. Assuming we have total observations of T,

$$\begin{aligned} p(z_1, \ldots, z_t) &= \sum_{x_0, \ldots, x_T} p(x_0, x_1, \ldots, x_T, z_1, \ldots, z_T) \\ &= \sum_{x_0, \ldots, x_T} \pi(x_0)K(x_0, x_1) \prod_{i=1}^{t} K(x_i, x_{i+1})P(z_i|x_i). \end{aligned} \tag{26}$$

We also can do filtering by calculating $p(X_t = x_t|z_1, \ldots, z_T)$,

$$\begin{aligned} p(x_t|z_1, \ldots, z_T) &\propto p(x_t, z_1, \ldots, z_T) \\ &= p(z_{t+1}, \ldots, z_T|x_t)p(x_t, z_1, \ldots, z_t) \end{aligned} \tag{27}$$

However, with the total law of probability the forward algorithm can be derived as follows

$$p(x_t, z_1, \dots, z_t) = \sum_{x_{t-1}} P(z_t|x_t) K(x_{t-1}, x_t) p(x_{t-1}, z_1, \dots, z_{t-1}) = \alpha_t(x_t) \tag{28}$$

and the backward algorithm is derived as

$$p(z_{t+1}, \dots, z_T|x_t) = \sum_{x_{t+1}} K(x_t, x_{t+1}) P(z_{t+1}|x_{t+1}, x_t) p(z_{t+2}, \dots, z_T|x_t, x_{t+1}, z_{t+1}) = \beta_t(x_t) \tag{29}$$

and therefore,

$$\beta_t(x_t) = \sum_{x_{t+1}} K(x_t, x_{t+1}) P(z_{t+1}|x_{t+1}, x_t) \beta_{t+1}(x_{t+1}). \tag{30}$$

4 Multiple Object Tracking

Multiple object tracking(MOT) has drawn attention in may fields of study. One of methods to go about MOT problem is through random finite set (RFS). The problem of MOT is the joint estimate of number of objects and trajectories upon receiving measurements. Each object may stay or leave the field of view. Number of new objects may also come to the scene. Let N_{k-1} be the number of objects at time $(k-1)$. Suppose $\{x_{1,k-1}, \dots, x_{N_{k-1},k-1}\} \in \mathcal{X}^{N_{k-1}}$ be the set of objects at time $(k-1)$. Each object may leave with probability $1 - P_{k|k-1}$ or may stay with probability $P_{k|k-1}$ and transitions to the next step through transition kernel probability $p(x_k|x_{k-1})$. Measurements $\{z_{1,k}, \dots, z_{M_k,k}\} \in \mathcal{Z}^{M_k}$ are received at time k. Given the state x_k, the measurement z_k is drawned from $p(z_k|x_k)$. The goal is to jointly estimate the number of objects and their states from measurements. In this survey, we outline two main RFS-based filters for multiple object tracking. We also study the main properties of RFS [8, 9, 10].

4.1 Random Finite Set

An RFS X is a random variable taking values in $\sigma(X)$, where $\sigma(X)$ does not have the properties a regular sigma-field. Detailed properties of RFS $\sigma(X)$ are discussed [11]. We discuss some of these properties in this report upon which we build the next sections.

An RFS X has nonnegative density of π on $\sigma(X)$ such that for any $\mathcal{B} \subseteq \mathcal{X}$

$$Pr(X \subseteq \mathcal{B}) = \int_{\mathcal{B}} \pi(X) \delta(X) \tag{31}$$

where the integral is a set integral defined as [10]

$$\int_{\mathcal{B}} \pi(X) \delta(X) = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{\mathcal{B}^i} \pi(x_1, \dots, x_i) d(x_1, \dots, x_i). \tag{32}$$

Note that $\pi(X)$ is not a probability density in a common sense. However, if $\mathcal{U}$ is the unit hyper-measure of $\mathcal{X}$, then $\pi(X)\mathcal{U}^{|X|}$ is a probability density.

In a multi-object tracking using RFS, the posterior distribution has the same form as the regular posterior distribution, i.e.,

$$\pi(X|Z) = \frac{\pi(Z|X)\pi(X)}{\int \pi(Z|X)\pi(X)\delta X}. \tag{33}$$

Where $\pi(Z|X)$ is the likelihood function. Maximizing the posterior distribution gives the estimate

$$\hat{X} = \underset{\{X:|X|=\hat{n}\}}{\arg\max}\, \pi(X|Z) \tag{34}$$

where $\hat{n} = \arg\max_n \rho(n|Z)$.

Example 1: Bernoulli RFS X

An RFS X is Bernoulli RFS for which it has probability $(1-r)$ of being empty and probability r of being a singleton $\{x\}$ whose density is $p(x)$, that is,

$$\pi(X) = \begin{cases} 1-r & X = \emptyset \\ rp(x) & X = \{x\} \end{cases}. \tag{35}$$

Example 2: Multi-Bernoulli RFS X

Multi-Bernoulli RFS X is a union of n independent Bernoulli RFS. The cardinality distribution of an RFS is

$$\rho(n) = Pr(|X| = n). \tag{36}$$

And hence the RFS characterization follows

$$\pi(\{x_1, \ldots, x_n\}) = n!\rho(n) \prod_{i=1}^{n} p(x_i). \tag{37}$$

4.1.1 Multi-Object Tracking Modeling through RFS

In a multi-object tracking setup, each object $x_{k-1} \in X_{k-1}$ generates a Bernoulli RFS $S_{k|k-1}(x_{k-1})$ at time k. New objects at time k are modeled by a birth RFS Γ_k. The multi-object state transition equation

$$X_k = \bigcup_{x_{k-1} \in X_{k-1}} S_{k|k-1}(x_{k-1}) \cup \Gamma_k. \tag{38}$$

The set X_{k-1} can evolve to X_k according to a transition density $\phi(X_k|X_{k-1})$ which captures the births and deaths.

An RFS based model for the observations is, for any $x_k \in X_k$, generate a Bernoulli RFS $\epsilon_K(x_k)$, and generate Z_k from X_k via the measurement equation. And an RFS η_k to model the false detections, i.e.,

$$Z_k = \bigcup_{x_k \in X_k} \epsilon_K(x_k) \cup \eta_k. \tag{39}$$

In general, the measurements are generated according to the likelihood function $\psi(Z_k|X_k)$.

4.1.2 Recursive Bayesian Inference

With using RFS, the posterior distribution $p(X_k|Z^k)$ for multi-object tracking can be computed the same way as a single object tracking method. We compute the prediction and update recursively. The Bayesian prediction equation is

$$p(X|Z^{k-1}) = \int \phi(X|Y)p(Y|Z^{k-1})\delta Y \tag{40}$$

and the update equation

$$p(X|Z^k) = \frac{\psi(Z_k|X)p(X|Z^{k-1})}{\int \psi(Z_k|Y)p(Y|Z^{k-1})\delta Y}. \tag{41}$$

4.2 Probability Hypothesis Density Filter

The first moment of an RFS is probability hypothesis density (PHD) which is known as density function [12, 13]. PHD is a function $\nu \geq 0$ such that

$$\mathbb{E}\big[|X \cap \mathcal{B}|\big] = \int_{\mathcal{B}} \nu(x)dx \tag{42}$$

for any region $\mathcal{B} \subseteq \mathcal{X}$. It is shown in [13] that

$$\nu(X) = \int \pi(\{x\} \cup X)\delta X. \tag{43}$$

and $\hat{n} = \arg\max_n \rho(n)$ or $\hat{n} = \mathbb{E}[|X|]$.

The probability hypothesis density (PHD) filter is a Bayes approximate of the multi-object filter which is based on point processes [13].In this method, first moment of $p(\cdot|Z^k)$ is propagated, $\nu(\cdot|Z^k)$. In this filter, a Poisson RFS is assumed. The PHD filter prediction equals

$$\nu_{k|k-1}(x) = \langle P_{S,k|k-1}\phi(x|\cdot), \nu_{k-1}\rangle + \gamma_k(x) \tag{44}$$

where $\langle \cdot, \cdot \rangle$ represents the inner product, $P_{S,k|k-1}$ is the survival probability, and $\gamma_k(x)$ represents the birth, for instance, a Gaussian mixture birth PHD is

$$\gamma_k(x) = \sum_{i=1}^{N_k} w^i_{\Gamma_k}\mathcal{N}(x : \mu^i_{\Gamma_k}, \Sigma^i_{\Gamma_k}). \tag{45}$$

Algorithm 4 Particle PHD Filter Recursion at Time k

```
Input: $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^{L_{k-1}}$ from time $(k-1)$, $Z_k$ measurements
for $i = 1, \dots, L_{k-1}$ do
    Sample $\tilde{x}_k^i \sim q(\cdot|x_{k-1}^i, Z_k)$
    Set $\tilde{w}_{k|k-1}^i = \frac{\phi(\tilde{x}_k^i|x_{k-1}^i)P_{S,k|k-1}(x_{k-1}^i)}{q(\tilde{x}_k^i|x_{k-1}^i, Z_k)} w_{k-1}^i$
for $i = L_{k-1}+1, \dots, N_k$ do
    Sample $\tilde{x}_k^i \sim r_k(\cdot|Z_k)$
    Set $\tilde{w}_{k|k-1}^i = \frac{1}{N_k - L_{k-1}} \frac{\gamma_k(\tilde{x}_k^i)}{r_k(\tilde{x}_k^i|Z_k)}$
where $r_k(\cdot|Z_k)$ is the importance density such that $\gamma_k(\tilde{x}_k^i) > 0$
$P_{S,k|k-1}(x_{k-1}^i)$ is the probability of survival
$N_k - L_{k-1}$ is the number of new born objects
for $z \in Z_k$ do
    $C_k(z) = \sum_{i=1}^{N_k} \zeta_k(z, \tilde{x}_k^i)\tilde{w}_{k|k-1}^i$
where $\zeta_k(z,x) = \frac{P_{D,k}(x)\psi_k(z|x)}{p_{F,k}(z)}$ is the detection to false alarm ratio of $z$ given $x$.
for $i = 1, \dots, N_k$ do
    Update the weights
    $\tilde{w}_k^i = \left[1 - P_{D,k}(\tilde{x}_k^i) + \sum_{z \in Z_k} \frac{\zeta_k(z, \tilde{x}_k^i)}{\epsilon_{F,k} + C_k(z)}\right] \tilde{w}_{k|k-1}^i$
where $\epsilon_{F,k}$ is the expected number of false alarms
Compute $\hat{W}_{k|k} = \sum_{i=1} N_k \tilde{w}_k^i$
Resample $\{\tilde{x}_k^i, \frac{w_k^i}{\hat{W}_{k|k}}\}_{i=1}^{N_k}$ to get $\{x_k^i, w_k^i\}_{i=1}^{L_k}$
```

The update equation is

$$\nu_k(x) = [1 - P_{D,k}]\nu_{k|k-1}(x) + \sum_{z \in Z_k} \frac{\zeta_k(z,x)\nu_{k|k-1}(x)}{\lambda_{F,k} + \langle \zeta_k(z,\cdot), \nu_{k|k-1} \rangle} \tag{46}$$

where $P_{D,k}$ is the probability of detection and $\zeta_k(z,x) = \frac{P_{D,k}(x)\psi_k(z|x)}{p_{F,k}(z)}$. Note that $p_{F,k}$ is the density of false alarm and $\lambda_{F,k}$ is the expected number of false alarms at time k. We apply the particle PHD filter algorithm summarized in Algorithm 4 to a problem of tracking with two objects with Gaussian noise. Figure 5 demonstrate the PHD filter tracking method.

4.3 Labeled Multi-Bernoulli Filtering

The labeled multi-Bernoulli filter is another approximation of the Bayesian multi-object tracking [14]. Each object is labeled by an ordered pair $\ell = (k, i)$ for k as the time of birth and i as the unique index to differentiate among objects born at time same time. In addition

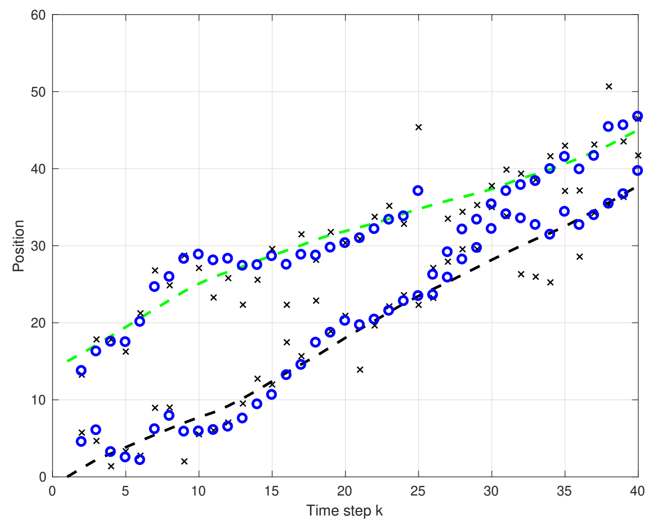

Figure 5: Particle PHD filter to track two objects.

to the aforementioned spaces, the label space for new born objects is defined as $\mathbb{L}_k$. An object at time k has state (x, ℓ) for $x \in \mathcal{X}$ and $\ell \in \mathbb{L}^k$. Set $\mathcal{L}(X)$ to denote the set of labels for X. We assume X and $\mathcal{L}$ have the same cardinality. We define $\Delta(X) = \delta_{|X|}[|X|]$ and $\xi : \mathbb{L}^k \to \{0, 1, \ldots, |Z|\}$ to be the distinct label indicator and association map with $\Xi^k = \{\xi\}$, respectively.

The prediction and update equations can be computed as follows [15, 14, 16]. The prediction equation is

$$p(X|Z^{k-1}) = \Delta(X) \sum_{\eta \in \Xi^{k-1}} w_{k-1}^{\eta}(\mathcal{L}(X))[p_{k-1}^{\eta}]^X \tag{47}$$

where p_{k-1}^{η} is a probability estimation, $\eta = (\xi_1, \ldots, \xi_{k-1}) \in \Xi^k$, and each weight $w_{k-1}^{\eta} > 0$ and

$$\sum_L \sum_\eta w_{k-1}^{\eta}(L) = 1.$$

Also the cardinality distribution equals

$$\rho_{k-1}(n) = \sum_L \sum_\eta \delta_n(|L|) w_{k-1}^{\eta}(L). \tag{48}$$

We demonstrate the importance of the labeled multi-Bernoulli method through simulations. Figure 6 displays the cardinality estimation for 5 objects using the labeled multi-Bernoulli filter. We also demonstrate the performance of this multi-object filtering method for 10,000 Monte Carlo simulations. The optimal sub-pattern assignment (OSPA) [17] for order $p = 1$ and cut-off $c = 100$ is depicted in Figure 7.

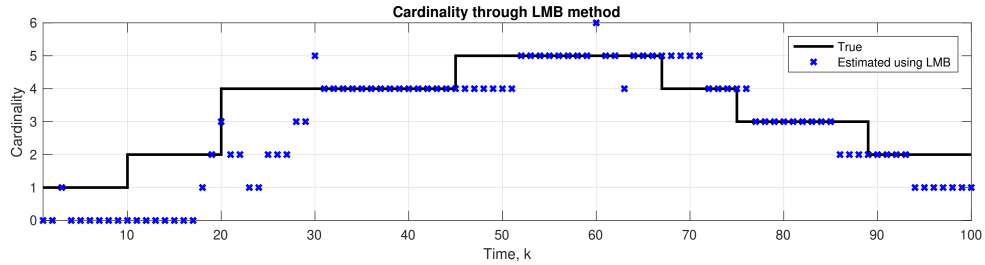

Figure 6: Cardinality estimation using labeled multi-Bernoulli filtering.

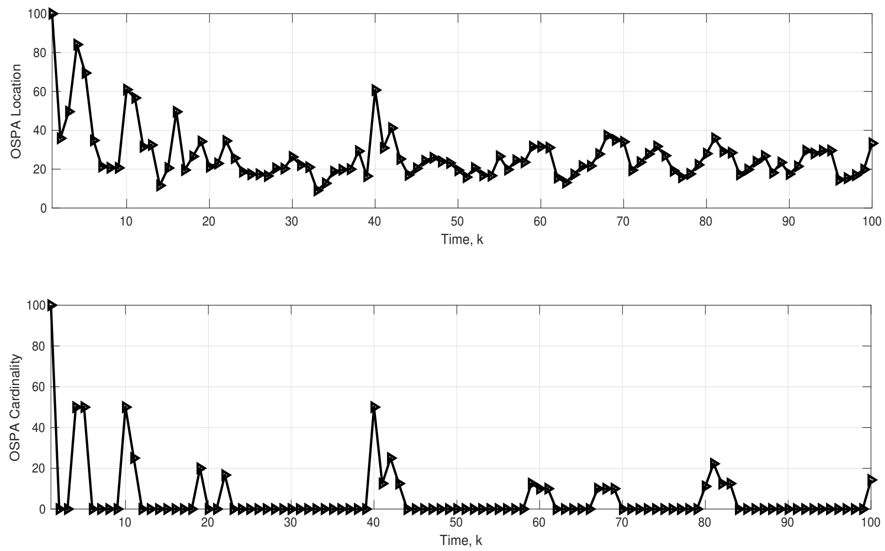

Figure 7: Labeled multi-Bernoulli OSPA performance for order $p = 1$ and cut-off $c = 100$ for 10000 MCMC simulations.

Recently, nonparametric models have attracted a lot of attention. A Dirichlet process is employed to address issues in a linear dynamic model [18]. A hierarchical modeling as a prior on the modes on the maneuvering problem is studied [19]. More advanced multiple object tracking models are introduced by using the dependent Dirichlet processes and Pitman-Yor processes [20, 21, 22, 23, 24, 25, 26]. A comprehensive review of the Dirichlet process may be found in [27].

5 Conclusion

We have simulated the some useful stochastic filtering methods to various linear and non-linear state space models. We also compared their results appropriately and discussed their performance. We have also discussed some of the fundamentals of particle filter algorithm with worked out examples. We discuss the hidden Markov chain and its properties. we studied some newer filtering methods for multiple object tracking. It is obvious that the application of a particular filtering method obviously depends on the problem presented.

References

[1] Christopher M Bishop. Pattern recognition and machine learning. springer, 2006.

[2] Adrian Smith. Sequential Monte Carlo methods in practice. Springer Science & Business Media, 2013.

[3] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In Signal processing, sensor fusion, and target recognition VI, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.

[4] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on signal processing, 50(2):174–188, 2002.

[5] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of nonlinear filtering, 2009.

[6] Kevin P Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.

[7] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. In Hidden Markov models: applications in computer vision, pages 9–41. World Scientific, 2001.

[8] Trevor M Wood. Random finite sets for multitarget tracking with applications. PhD thesis, Oxford University, UK, 2011.

[9] Ronald PS Mahler. Statistical multisource-multitarget information fusion, volume 685. Artech House Norwood, MA, 2007.

[10] Ronald PS Mahler. Advances in statistical multisource-multitarget information fusion. Artech House, 2014.

[11] Ba-ngu Vo, Mahendra Mallick, Yaakov Bar-shalom, Stefano Coraluppi, Richard Osborne III, Ronald Mahler, and Ba-tuong Vo. Multitarget tracking. Wiley Encyclopedia of Electrical and Electronics Engineering, pages 1–15, 1999.

[12] Ronald Mahler. Phd filters of higher order in target number. IEEE Transactions on Aerospace and Electronic systems, 43(4):1523–1543, 2007.

[13] Ronald PS Mahler. Multitarget bayes filtering via first-order multitarget moments. IEEE Transactions on Aerospace and Electronic systems, 39(4):1152–1178, 2003.

[14] Ba-Tuong Vo, Ba-Ngu Vo, and Antonio Cantoni. The cardinality balanced multi-target multi-bernoulli filter and its implementations. IEEE Transactions on Signal Processing, 57(2):409–423, 2008.

[15] Stephan Reuter, Ba-Tuong Vo, Ba-Ngu Vo, and Klaus Dietmayer. The labeled multi-bernoulli filter. IEEE Transactions on Signal Processing, 62(12):3246–3260, 2014.

[16] Ba-Tuong Vo, Ba-Ngu Vo, Reza Hoseinnezhad, and Ronald PS Mahler. Robust multi-bernoulli filtering. IEEE Journal of Selected Topics in Signal Processing, 7(3):399–409, 2013.

[17] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A consistent metric for performance evaluation of multi-object filters. IEEE transactions on signal processing, 56(8):3447–3457, 2008.

[18] Francois Caron, Manuel Davy, Arnaud Doucet, Emmanuel Duflos, and Philippe Vanheeghe. Bayesian inference for linear dynamic models with Dirichlet process mixtures. IEEE Transactions on Signal Processing, 56(1):71–84, 2007.

[19] Emily B Fox, Erik B Sudderth, and Alan S Willsky. Hierarchical dirichlet processes for tracking maneuvering targets. In 2007 10th International Conference on Information Fusion, pages 1–8. IEEE, 2007.

[20] Bahman Moraffah and Antonia Papandreou-Suppappola. Dependent Dirichlet process modeling and identity learning for multiple object tracking. In Asilomar Conference on Signals, Systems, and Computers, 2018.

[21] Bahman Moraffah, Cesar Brito, Bindya Venkatesh, and Antonia Papandreou-Suppappola. Tracking multiple objects with multimodal dependent measurements: Bayesian nonparametric modeling. In Asilomar Conference on Signals, Systems, and Computers, 2019.

[22] Bahman Moraffah, Muralidhar Rangaswamy, and Antonia Papandreou-Suppappola. Nonparametric Bayesian methods and the dependent Pitman-Yor process for modeling evolution in multiple state priors. In 22nd International Conference on Information Fusion, 2019.

[23] Bahman Moraffah, Cesar Brito, Bindya Venkatesh, and Antonia Papandreou-Suppappola. Use of hierarchical Dirichlet processes to integrate dependent observations from multiple disparate sensors for tracking. In 22nd International Conference on Information Fusion, 2019.

[24] Bahman Moraffah. On the use of dependent nonparametric processes in multiple object tracking. Technical report, Arizona State University, June 2019.

[25] Bahman Moraffah and Antonia Papandreou-Suppappola. Random infinite tree and dependent Poisson diffusion process for nonparametric Bayesian modeling in multiple object tracking. In International Conference on Acoustics, Speech, and Signal Processing, 2019.

[26] Bahman Moraffah. Inference for multiple object tracking: A Bayesian nonparametric approach. arXiv preprint arXiv:1909.06984, 2019.

[27] Bahman Moraffah. Introduction to machine learning: A review. arXiv preprint arXiv:1007.0296, 2019.