MOT, 01 Single Object Tracking

Yao Ge, alexgecontrol@qq.com

Jan. 2022

1 Introduction

In this document **the Bayesian algorithm workflow for single object tracking** under:

- possible miss detection
- possible clutter detection

is presented.

2 Assumptions

Assumptions below are made for single object tracking:

- there is always **precisely one object present** at all times
- **at most one detection** per object per measurement

3 Extensions

To begin with, first let's have a look at **extensions of standard sequential estimation**, which are needed by single object tracking.

3.1 Miss Detection

The first extension is the modeling of potential miss detection. Here **Bernoulli random finite set**, which is based on **Bernoulli distribution**, is used to model the potential miss detection.

For **point object tracking**, the object detection under potential miss detection can be modelled as a **Bernoulli random finite set**, which contains **either none or one detection**:

$$p(O_k|x_k) = \begin{cases} 1 - p^D(x_k), & \text{if} O_k = [] \\ p^D(x_k)g(o_k|x_k), & \text{if} O_k = o_k \end{cases} \tag{1}$$

With this extension, the update step for single-object tracking can be generalized as follows:

$$p(x_k|O_{1:k}) = \begin{cases} (1 - p^D(x_k))p(x_k|O_{1:k-1}), & \text{if } O_k = [] \\ p^D(x_k)g(o_k|x_k)p(x_k|O_{1:k-1}), & \text{if } O_k = o_k \end{cases} \tag{2}$$

Which means:

- if the object is detected, perform standard update
- otherwise use predicted distribution as posterior density

3.2 Clutter Detection

The second extension is the modeling of potential clutter detection. Here **Poisson point process**, which is based on **Poisson distribution**, is used to model the potential clutter detection.

Inside sensor **field of view, FOV**, assume **clutter intensity function** is defined as $\lambda_c(v)$, then we have:

$$\text{expected num. of clutters } \overline{\lambda_c} = \int \lambda_c(v)\, dv$$

$$\text{clutter detection distribution } f_c(v) = \frac{\lambda_c(v)}{\overline{\lambda_c}}$$

With this extension, assume at current timestamp m_k^c clutter detections are generated $C_k = [c_k^1, ..., c_k^{m_k^c}]$. The likelihood of this measurement is:

$$
\begin{aligned}
P(C_k) &= \frac{exp(-\overline{\lambda_c})\overline{\lambda_c}^{m_k^c}}{m_k^c!} \prod_{i=1}^{m_k^c} f_c(c_k^i) \\
&= \frac{exp(-\overline{\lambda_c})\overline{\lambda_c}^{m_k^c}}{m_k^c!} \prod_{i=1}^{m_k^c} \frac{\lambda_c(c_k^i)}{\overline{\lambda_c}} \\
&= \frac{exp(-\overline{\lambda_c})}{m_k^c!} \prod_{i=1}^{m_k^c} \lambda_c(c_k^i)
\end{aligned} \tag{3}
$$

3.3 Data Association

The final extension is the modeling of potential data association. Assume at current timestamp there are m_k^z measurements $Z_k = [z_k^1, ..., z_k^{m_k^z}]$. Here introduce another **indicator variable** for potential data association:

$$
\theta_k = \begin{cases} i > 0, & \text{if } z_k^i \text{ is the object detection} \\ 0, & \text{if the object is not detected} \end{cases} \tag{4}
$$

With this extension, define $m = |Z_k|$ the measurement model can be written as:

$$
P(Z_k|x_k) = \sum_{i=0}^{m} p(Z_k|\theta_i, m, x_k)p(\theta_i, m|x_k) \tag{5}
$$

This can be further decomposed into the two cases below:

Case 01, the object is not detected and there are m clutter detections. In this case we have:

$$
p(\theta_k = 0, m|x_k)p(Z_k|\theta_k = 0, m, x_k) = (1 - p^D(x_k)\frac{exp(-\overline{\lambda_c})}{m_k^z!} \prod_{i=1}^{m_k^z} \lambda_c(z_k^i)
$$

Case 02, the object is detected and there are $m-1$ clutter detections. In this case we have:

$$
p(\theta_k = i, m|x_k)p(Z_k|\theta_k = i, m, x_k) = p^D(x_k)\frac{g(z_k^i|x_k)}{\lambda_c(z_k^i)} \frac{exp(-\overline{\lambda_c})}{m_k^z!} \prod_{j=1}^{m_k^z} \lambda_c(z_k^j)
$$

So the total probability is:

$$P(Z_k|x_k) = ((1 - p^D(x_k) + (p^D(x_k) \sum_{i=1}^{m_k^z} \frac{g(z_k^i|x_k)}{\lambda_c(z_k^i)}) \frac{exp(-\overline{\lambda_c})}{m_k^z!} \prod_{i=1}^{m_k^z} \lambda_c(z_k^i) \tag{6}$$

4 Exact Solutions

4.1 Challenge

Assume the number of measurements till timestamp k are $m_1, ..., m_k$. So the total number of possible data association sequences at timestamp k is:

$$\prod_{i=1}^{k}(m_i + 1) \tag{7}$$

Which grows exponentially as timestamp goes by.

4.2 Normalize a Mixture

For $p(x) \propto \sum_{i=1}^{k} g_i(x)$, the general normalization procedure for creating a PDF is as follows:

- first normalize each component as $p_i(x) = \frac{g_i(x)}{\int g_i(x)\, dx}$
- then get the combination weight of each component as $w_i = \frac{\int g_i(x)\, dx}{\sum_{i=1}^{k} \int g_i(x)\, dx}$

4.3 General Single Object Tracking

For **single object tracking** the above procedure can be specialized into follows:

- first each mixture component is **conditional posterior PDF**, that is:
  - either **predicted PDF due to missing object detection**, the result of Kalman predict step
  - or **updated PDF from associated object detection**, the result of Kalman update step
- then the component weight is corresponding data association probability, that is:
  - either the prob. of miss detection, $\int (1 - p^D(x_k)) p(x_{k|k-1})\, dx_{k|k-1}$
  - or the prob. of D.A. with measurement i, $\int p^D(x_k) \frac{g(z_k^i|x_k)}{\lambda_c(z_k^i)} p(x_{k|k-1})\, dx_{k|k-1}$

4.4 Single Object Tracking, Linear and Gaussian

For **the further specialized case**, that is:

- The motion and the measurement models are both linear and Gaussian
- Object detection probability is constant inside FOV, $p^D(x_k) = p^D$

$$
\begin{aligned}
\textbf{Prior}: \; & x_{k-1|k-1} = \mathcal{N}(x_{k-1}, \hat{x_{k-1|k-1}}, \hat{P_{k-1|k-1}}) \\
\text{Motion}: \; & x_k = F_{k-1} x_{k-1} + q_{k-1} \\
& q_{k-1} \sim \mathcal{N}(q_{k-1}, 0, Q_{k-1}) \\
\textbf{Predict:} \; & x_{k|k-1} \sim \mathcal{N}(x_{k|k-1}, \check{x_{k|k-1}}, \check{P_{k|k-1}}) \\
& x_{k|k-1} \sim \mathcal{N}(x_{k|k-1}, F_{k-1}\hat{x_{k-1|k-1}}, F_{k-1}\hat{P_{k-1|k-1}}F_{k-1}^T + Q_{k-1}) \\
\text{Measurement:} \; & x_k = H_k x_k + r_k \\
& r_k \sim \mathcal{N}(r_k, 0, R_k) \\
\textbf{Measurement Likelihood:} \; & z_{k|k-1} \sim \mathcal{N}(z_{k|k-1}, \check{z_{k|k-1}}, \check{S_{k|k-1}}) \\
& z_{k|k-1} \sim \mathcal{N}(z_{k|k-1}, H_k \check{x_{k|k-1}}, H_k \check{P_{k|k-1}} H_k^T + R_k) \\
\textbf{Posterior Weight:} \; & w_{k|k}^{\tilde{\theta}_i} = \begin{cases} 1 - p^D, & \text{if } i = 0 \text{, miss detection} \\ p^D \frac{\mathcal{N}(z_k^{\theta_i}, \check{z_{k|k-1}}, \check{S_{k|k-1}})}{\lambda_C(z_k^{\theta_i})}, & \text{if the object is not detected} \end{cases} \\
\textbf{Posterior Component:} \; & x_{k|k}^{\theta_i} = \mathcal{N}(x_{k|k}^{\theta_i}, x_{k|k}^{\hat{\theta}_i}, P_{k|k}^{\hat{\theta}_i}) \\
& K_k = \check{P_{k|k-1}} H_k^T \check{S_{k|k-1}}^{-1} \\
& x_{k|k}^{\hat{\theta}_i} = \check{x_{k|k-1}} + K_k(z_k^{\theta_i} - \check{z_{k|k-1}}) \\
& P_{k|k}^{\hat{\theta}_i} = \check{P_{k|k-1}} - K_k H_k \check{P_{k|k-1}}
\end{aligned}
$$

4.5 Single Object Tracking, Nonlinear

Just approximate the above Kalman update workflow with:

- Extended Kalman Filter
- Iterative Extended Kalman Filter
- Unscented Kalman Filter
- ...

5 Practical Solutions

5.1 Engineering Challenge

Assume the number of measurements till timestamp k are $m_1, ..., m_k$. The total number of possible data association sequences at timestamp k is:

$$\prod_{i=1}^{k}(m_i + 1) \tag{8}$$

Which grows exponentially as timestamp goes by. To implement this in actual system, both:

- total **time** needed to perform Kalman update on each candidate component inside the mixture
- total **space** needed to maintain all active hypotheses

will grow exponentially as timestamp goes by, which is **NOT ACCEPTABLE** for practical engineering system. So we need **approximation algorithms** to tackle this challenge.

5.2 Approximation Strategy Overview

Here we always try to approximate the exact posterior distribution with Gaussian through a combination of:

- **data association reduction** through **gating**
- **active hypothesis reduction** through **pruning** and **merging**

5.3 Data Association Reduction

If the measurement likelihood can be approximated as

$$\mathcal{N}(z_k^{\theta_i}, z_{k|\check{k}-1}, S_{k|\check{k}-1}) \tag{9}$$

then the measurement's **squared Mahalanobis distance from predicted mean**

$$d^2_{h_{k-1},\theta_i} \sim \chi^2(n_Z) \tag{10}$$

in which n_Z is the dimensionality of measurement Z. With this, we can choose the gating threshold as:

$$\Pr(\text{discard object detection}) = Prd^2_{h_{k-1},\theta_i} > G|h_{k-1}, \theta_i, n_Z \tag{11}$$

which can be calculated with Chi-Squared Cumulative Distribution Table

5.4 Active Hypothesis Reduction

Here three representative algorithms are presented:

- **Nearest Neighbor (NN)** for **pruning**
- **Probabilistic Data Association (PDA)** for **merging**
- **Gaussian Sum Filter (GSF)** for **benchmark algorithm**

5.4.1 Nearest Neighbor (NN)

The algorithm can be summed up as follows:

- **Num. of Active Priors**: **1**
- **Hypo. Reduction**: Only keep the component with highest weight $w^{\tilde{\theta}_i}_{k|k}$
- **Num. of Active Posteriors**: **1**

5.4.2 Probabilistic Data Association (PDA)

- **Num. of Active Priors**: **1**
- **Hypo. Reduction**:
  - compute the mixture mean
$$\hat{x}_{k|k} = \sum_{\theta_i} w^{\theta_i}_{k|k} x^{\hat{\theta}_i}_{k|k} \tag{12}$$
  - compute the mixture covariance
$$\hat{P}_{k|k} = \sum_{\theta_i} w^{\theta_i}_{k|k} (P^{\hat{\theta}_i}_{k|k} + (x^{\hat{\theta}_i}_{k|k} - x_{k|k}))(x^{\hat{\theta}_i}_{k|k} - x_{k|k}))^T \tag{13}$$
  - approximate the mixture using single Gaussian as
- **Num. of Active Posteriors**: **1**

5.4.3 Gaussian Sum Filter (GSF)

- **Num. of Active Priors**: H_{k-1}
- **Hypo. Reduction**:
  - could be **pruning** like **NN**, that is, only keep candidate whose weight is about the threshold
  - could be **merging** like **PDA**, that is, merge selected candidates into one new candidate
  - or mixture of both
- **Num. of Active Posteriors**: H_k