Sensor Signal Processing for Defence Conference

12th and 13th September 2023

Royal College of Physicians Conference Centre
Consensus-based Distributed Variational Multi-object Tracker in Multi-Sensor Network

Qing Li
Runze Gan, Simon Godsill
Cambridge University
Outline

- Introduction
  - Scalable variational tracker: single sensor case

- Distributed fusion and tracking in multi-sensor network
  - Centralized Variational Tracker
  - Consensus-based distributed variational multi-object tracker

- Summary and future directions
Non-homogeneous Poisson process (NHPP) measurement model

Example: measurements of 2 targets and clutter process, generated by the NHPP model:

At time step n: target state $X_n$, measurements $Z_n$, target number $K$

Measurements from each object ($i=1,...,K$) and clutter ($i=0$) follow a NHPP with intensity $\lambda_i(Z_n|X_n,i)$

Total measurements follow an NHPP with intensity $\lambda(Z_n|X_n) = \sum_{i=0}^{K} \lambda_i(Z_n|X_n,i)$

Likelihood function: $\frac{e^{-\Lambda} \prod_{j=1}^{M_n} \sum_{i=0}^{K} \lambda_i(Z_{n,j}|X_n,i)}{M_n!}$

Association prior $p(\theta_{n,j}|\Lambda) = \frac{\sum_{k=0}^{K} \Lambda_k \delta[\theta_{n,j} = k]}{\Lambda_s}$, categorical distribution

Data Association $\theta_n = [\theta_{n,1}, \ldots, \theta_{n,M_n}]$ $\theta_{n,j} = i$ $\begin{cases} i = 0 & \text{From clutter} \\ i \in \{1, \ldots, K\} & \text{From target i} \end{cases}$

Variational multi-object tracker: single sensor case

cordinate ascent variational filtering [2]:

**target posterior distribution:** \( \hat{p}_n(X_n, \theta_n | Y_n) \)

**mean-field factorisation**

\[
q_n(X_n, \theta_n) = q_n(X_n)q_n(\theta_n) \quad \text{minimise the KL divergence}
\]

\[
\text{KL}(q_n(X_n)q_n(\theta_n) || \hat{p}_n(X_n, \theta_n | Y_n))
\]

**Implementation:**

1) **update for** \( q_n(X_n) \)
   
   \[
   q_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu_{n|n}^k, \Sigma_{n|n}^k), \quad \text{Kalman filter}
   \]

2) **update for** \( q_n(\theta_n) \)
   
   \[
   q_n(\theta_{n,j}) \propto \frac{\Lambda_0}{V} \delta[\theta_{n,j} = 0] + \sum_{k=1}^{K} \Lambda_k l_k \delta[\theta_{n,j} = k], \quad \text{categorical distribution}
   \]

Scalable! All can be updated independently

Why we choose variational tracker

Tracking 20 targets with heavy clutter:
Target rate: 10
Clutter density: $10^{-4}$

Fast and accurate!

TABLE I: Tracking performance comparisons

<table>
<thead>
<tr>
<th>dataset</th>
<th>$K$</th>
<th>PF-NHPP</th>
<th>Gibbs-AbNHPP</th>
<th>ET-JPDA</th>
<th>VB-AbNHPP$^{(1)}$</th>
<th>VB-AbNHPP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE (mean ±1σ)</td>
<td>track loss percentage (%)</td>
<td>CPU time (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>7.69±0.64</td>
<td>0.00</td>
<td>4.05</td>
<td>5.50±0.34</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>N/A</td>
<td>51.8</td>
<td>15.9</td>
<td>5.63±0.13</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>—</td>
<td>6.06±0.25</td>
<td>0.10</td>
<td>0.67</td>
<td>9.41±1.99</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>—</td>
<td>6.25±0.26</td>
<td>0.65</td>
<td>2.31</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Distributed tracking in multi-sensor network

Problem settings:
- A network of sensors tracking **a large number of targets** in clutter
- **Decentralized processing:**
  1) No central processing unit
  2) local communication with neighbours *(constraints of bandwidth)*
- Time-varying sensor network *(communication link failure)*

Defence Impacts: e.g., border surveillance, and maritime operations
- Fast and precise tracking
- Resilient for adversarial disruptions and communication constraints

Proposed methods:
- Centralised fusion
  - Variational trackers in multi-sensor network
  - Optimal distributed fusion based on consensus algorithm
  - Comparison: suboptimal distributed fusion
  - Fuse local posteriors using arithmetic average
  - Aim to achieve an equivalent result
Measurement and association model for a sensor network

Consider a sensor network with $N_s$ sensors

For each sensor $s$ ($s=1, 2, ..., N_s$):

- **Local measurements $Y^s_n$:**
  - independent NHPP model with Poisson rate $\Lambda^s$
  - **Likelihood function:**
    \[
    p(Y^s_n|\theta^s_n, X_n) = \prod_{j=1}^{M^s_n} \ell^s(Y^s_{n,j}|X_n, \theta^s_{n,j}),
    \]
  - **Association prior**
    \[
    p(\theta^s_{n,j}) = \frac{\sum_{k=0}^{K} \Lambda^s_{k} \delta[\theta^s_{n,j} = k]}{\Lambda^s_{sum}}
    \]
    Categorical distribution

For all $N_s$ sensors at the central unit:

- **Joint likelihood**
  \[
  p(Y_n|\theta_n, X_n) = \prod_{s=1}^{N_s} p(Y^s_n|\theta^s_n, X_n).
  \]

- **Joint association prior**
  \[
  p(\theta_n|M_n) = \prod_{s=1}^{N_s} p(\theta^s_n|M^s_n).
  \]
Centralized variational multi-object tracker

Coordinate ascent update: \[ q_n(X_n, \theta_n) = q_n(X_n)q_n(\theta_n) \]
Iteratively update until convergence

1. Update for \( q_n(X_n) \)
\[
q_n(X_n) = \hat{p}_n(X_n) \prod_{k=1}^{K} \mathcal{N}(Y^k_n; HX_{n,k}; R^k_n) \]

\[ Y^k_n = R^k_n \sum_{s=1}^{N_s} \Omega^s_{k,2} \] \[ \Omega^s_{k,2} = R^s_{k-1} \sum_{j=1}^{M_n} q_n(\theta^s_{n,j} = k)Y^s_{n,j} \] \[ R^k_n = \left( \sum_{s=1}^{N_s} \Omega^s_{k,1} \right)^{-1} \] \[ \Omega^s_{k,1} = R^s_{k-1} \sum_{j=1}^{M_n} q_n(\theta^s_{n,j} = k) \]
Independently update for each target k
- prediction \( \hat{p}_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu^k_{n|n-1}, \Sigma^k_{n|n-1}) \)
- Kalman filter update \( q_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu^k_{n|n}, \Sigma^k_{n|n}) \)

2. Update for \( q_n(\theta_n) \)
\[
q_n(\theta_n) = \prod_{s=1}^{N_s} q_n(\theta^s_n) \]
Independently update for each sensor \( s \), each association \( j \)
\[
q_n(\theta^s_{n,j}) \propto \frac{\Lambda^s_0}{V^s} \delta[\theta^s_{n,j} = 0] + \sum_{k=1}^{K} \Lambda^s_k l^s_k \delta[\theta^s_{n,j} = k] \]
Categorical distribution
How to decentralise it?

Coordinate ascent update: $q_n(X_n, \theta_n) = q_n(X_n)q_n(\theta_n)$
Iteratively update until convergence

1. Update for $q_n(X_n)$

$\Omega_{k,2}^s = R_k^{-1} \sum_{j=1}^{M_n} q_n(\theta_{n,j}^s = k) Y_{n,j}^s$.

$\Omega_{k,1}^s = R_k^{-1} \sum_{j=1}^{M_n} q_n(\theta_{n,j}^s = k)$.

Compute at each sensor

Independently update for each target $k$
- Kalman filter update $q_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu_{n|n}^k, \Sigma_{n|n}^k)$

2. Update for $q_n(\theta_n)$

Independently update for each sensor $s$, each association $j$

$q_n(\theta_{n,j}^s) \propto \frac{\Lambda^s_{n,j} \delta[\theta_{n,j}^s = 0]}{V_s} + \sum_{k=1}^{K} \Lambda^s_{k} l^s_k \delta[\theta_{n,j}^s = k]$. 

Consensus:
When converge, each sensor has:

$\bar{Y}_n^k = \bar{R}_n^k \sum_{s=1}^{N_s} \Omega_{k,2}^s$,

$\bar{R}_n^k = \left( \sum_{s=1}^{N_s} \Omega_{k,1}^s \right)^{-1}$.
A demo of average consensus algorithm

- **Initial iteration**: Every node starts with an initial value $\Omega^S_{k,1}$.
- **For each iteration**: each node communicates with its neighbors to collect their values, and update value:
  \[
  \hat{\Omega}^{(s,m+1)}_{k,1} = W_{ss}^{(m)}\hat{\Omega}^{(s,m)}_{k,1} + \sum_{j \in N_s(m)} W_{sj}^{(m)}\hat{\Omega}^{(j,m)}_{k,1}
  \]
- **Output**: When converge, each node should have same value

Number of sensors: 20  
Iteration: 0

Initial value: range from 1-50  
Output: average value 31.66
Works in time-varying sensor network!

- Each sensor node has $\Omega^S_{k,1}$ locally
- When converge, each sensor has the same value $\frac{1}{N_s} \sum_{s=1}^{N_s} \Omega^S_{k,1}$
Consensus-based distributed variational multi-object tracker

Objective: with only local communications, each sensor has the same estimate as the centralized sensor fusion that have all data

Implementation: At each sensor \( s = 1, 2, \ldots, N_s \)

1. Update for \( q_n(X_n) \)
   
   Step 1. compute \( \Omega_{k,1}^s, \Omega_{k,2}^s \) locally
   
   Step 2. perform average consensus to get
   
   \[
   \hat{\Omega}_{k,1} = \frac{1}{N_s} \sum_{s=1}^{N_s} \Omega_{k,1}^s \]
   
   \[
   \hat{\Omega}_{k,2} = \frac{1}{N_s} \sum_{s=1}^{N_s} \Omega_{k,2}^s .
   \]

   Step 3. for each target \( k \)
   
   Kalman filter update \( q_n(X_{n,k}) = \mathcal{N}(X_{n,k}; \mu_{n|n}^k, \Sigma_{n|n}^k) \)
   
   using \( \bar{Y}_n^k = \bar{R}_n^k(N_s\hat{\Omega}_{k,2}) \)
   
   \( \bar{R}_n^k = (N_s\hat{\Omega}_{k,1})^{-1} \)

2. Update for \( q_n(\theta_n) \)

   for each association \( j \)
   
   \[
   q_n(\theta_{n,j}^s) \propto \frac{L_0^s}{V_s} \delta[\theta_{n,j}^s = 0] + \sum_{k=1}^{K} \Lambda_k l_k^s \delta[\theta_{n,j}^s = k] .
   \]
Results

Settings:
Number of sensors: 20
Number of targets: 50
Target rate: 1
Clutter rate: 500

Measurements from all sensors (grey dots)
Ground truth tracks (black lines)
Target initial positions (green circles)
Results - an example from one single run

Optimal distributed fusion

Suboptimal arithmetic average distributed fusion

Ground truth tracks: black lines
Estimate position: dotted colored line
95% confidence ellipse: shaded circles
Results

Mean OSPA of different fusion methods (average consensus iteration is 20)

- **Centralised fusion**
- **Optimal distributed fusion**
- **AA distributed fusion**

Mean OSPA of the optimal distributed fusion over different iterations

- **Optimal distributed fusion**
- **Centralised fusion**
Summary and extensions

**Summary:** A consensus-based distributed variational tracker

1. **Scalable to target and measurement number**
2. **Achieve optimal fusion with distributed implementation**
3. **Reliable solution: work in time-varying communication links**

**Extensions**

1. **A more flexible scheme that allows each sensor operate independently without waiting for consensus;**
2. **Solutions for a more general heterogeneous sensor network**
   - sensor network with different coverage
   - measurement function can be nonlinear (range, bearing)
3. **More robust and versatile tracking**
   - Variational tracker with missed objects relocation for heavy clutter cases [3]

---