High-Dimensional Inferencing for Multi-Object Dynamical Systems

Ba-Ngu Vo
Dept. Electrical & Computer Engineering
Curtin University
Perth
Australia
http://ba-ngu.vo-au.com

Bayesian Computation for High-Dimensional Statistical Models
Singapore, 29 August 2018
Hidden Markov Model (of a dynamical system)

$x_k = \Phi(x_{k-1}, u_{k-1}, n_k)$

$z_k = \Psi(x_k, u_{k-1}, v_k)$

$f_{k|k-1}(x_k | x_{k-1}, u_{k-1})$

$g_k(z_k | x_k, u_{k-1})$

Observation likelihood

Markov transition density

Filtering

System ID

Control
Finite Sets (of points, usually vectors) aka Point Patterns
Introduction

Multi-object HMM

- Target Tracking: $X_k = \text{set of target tracks}$
- Computer Vision: $X_k = \text{set of object trajectories}$
- Simultaneous Localization & Mapping: $X_k = \text{set of landmarks}$
- Multiple Instance (MI) Learning: $X_k = \text{set of features/instances}$
Outline

- Multi-object System
- Inference for Multi-object HMM
  - Example applications
- High-Dimensional Multi-Object Estimation
  - Gibbs Sampling
- Large Scale Multi-Object Filtering
  - “Deep” Multi-Object Smoothing
- Concluding Remarks
Multi-Object System

- Multi-object state: Point Pattern or Finite set
- Multi-object system: Finite-set-valued dynamical system

Fundamental difference from classical dynamical system:
- Random number of state and measurement vectors
- False negatives, false positives, association uncertainty

Recall: Fundamental (single-object) system problems

**Estimation:** estimate (system) trajectory
- filtering: $\hat{x}_0, \ldots, \hat{x}_k$
- smoothing: $\hat{x}_{0:k}$

**System Identification:** estimate system parameters

**Control:** manipulate (system) state/trajectory
For single-object system: \textbf{trajectory} = \textit{history of states}

\[ x_{0:k} = [x_0, \ldots, x_k] \]

Multi-object trajectory \equiv \textit{set of trajectories}

Multi-object trajectory = \textit{history of multi-object states}
**Multi-Object System**

**multi-object trajectory** = history of labeled multi-object states

labeled multi-object state $X = \{x_1, \ldots, x_j, \ldots, x_n\}$

labeled state vector $(x_j, l_j) \in X \times L$

kinematics label (distinct from other objects)

labeled trajectory = history of labelled state vectors with the same label

State space

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_4$</th>
<th>$X_k$</th>
</tr>
</thead>
</table>

0 1 2 3 4 ... $k$

labeled trajectories

$X_{0:k}$

time
labeled multi-object sequence

set of labeled trajectories (in \(\{j:k\}\))

trajectory with label \(\ell\):

\[
X_{j:k} = \left\{ x_{s(\ell):t(\ell)}^{(\ell)} : \ell \in \bigcup_{i=j}^{k} \mathcal{L}(X_i) \right\}
\]

starting time: terminating time

Multi-Object System

\(X_0:k\)

labeled trajectories

State space

0 1 2 3 4 ...

\(X_1\) \(X_4\) \(X_k\)

time
Multi-Object System

Fundamental multi-object system problems

- **Estimation**: estimate multi-object trajectory
  - filtering with labeled multi-object states: $\hat{X}_0, \ldots, \hat{X}_k$
  - smoothing with labeled multi-object states: $\hat{X}_{0:k}$

- **System Identification**: estimate multi-object system parameters

- **Control**: manipulate multi-object trajectory
Stochastic Multi-Object Model

- The **number of points** is random
- The **points** have no ordering and are random
- Statistical Model: (simple finite) **point process**

\[
N_X(S) = |X \cap S|
\]

A Simple Multi-Object State Transition Model [Mahler 03]

Transition for each element $x$ of a given multi-object state $X_{k-1}$

State transition equation

$$X_k = S_{k|k-1}(X_{k-1}) \cup B_{k|k-1}(X_{k-1}) \cup \Gamma_k$$

State transition kernel

$$f_{k|k-1}(X_k | X_{k-1})$$
A Simple Multi-Object Observation Model [Mahler 03]

Observation equation:
\[ Z_k = \Theta_k(X_k) \cup K_k \]

Observation likelihood:
\[ g_k(Z_k|X_k) \]

Observation for each element \( x \) of a given multi-object state \( X_k \)

State space \( \emptyset \)

Observation space

Likelihood

Misdetection \( \emptyset \)

Clutter
Inference for Multi-Object HMM

Posterior Recursion (smoothing-while-filtering, expensive)

\[ \pi_{0:k}(X_{0:k}|Z_{0:k}) \propto g_k(Z_k|X_k) f_k|_{k-1}(X_k|X_{k-1}) \pi_{0:k-1}(X_{0:k-1}|Z_{0:k-1}) \]

Solutions: MCMC, PMMH

Filtering Recursion [Mahler 03]

\[ \pi_k|_{k-1}(X_k) = \int f_{k|k-1}(X_k|X) \pi_{k-1}(X) \delta X, \]
\[ \pi_k(X_k) = \frac{g_k(Z_k|X_k) \pi_k|_{k-1}(X_k)}{\int g_k(Z_k|X) \pi_{k-1}(X) \delta X}, \]

Solutions: SMC, Analytic Approximation
Inference for Multi-Object HMM

Multi-object Exponential Notation:

\[ [h] X_{j:k} \triangleq [h] \left\{ x_{s(\ell):t(\ell)}^{(\ell)} : \ell \in \cup_{i=j}^{k} \mathcal{L}(X_i) \right\} = \prod_{\ell \in \cup_{i=j}^{k} \mathcal{L}(X_i)} h(x_{s(\ell):t(\ell)}^{(\ell)}) \]
Inference for Multi-Object HMM

Filtering Recursion Solution (GLMB) [VV 13]

\[
\pi_k(X) = \Delta(X) \sum_{\xi \in \Xi} w_k^{(\xi)} (\mathcal{L}(X)) \left[ p_k^{(\xi)} \right]^X
\]

distinct label indicator \quad \text{labels of } X \quad \text{multi-object exponential}

Closed under prediction & Bayes update

Posterior Recursion Solution (replace \(X\) with \(X_{0:k}\)) [VV 18]

\[
\pi_{0:k}(X_{0:k}) = \Delta(X_{0:k}) \sum_{\xi \in \Xi} w_{0:k}^{(\xi)} (\mathcal{L}(X_{0:k})) \left[ p_{0:k}^{(\xi)} \right]^{X_{0:k}}
\]

joint distinct label indicator \quad \text{labels of } X_{0:k} \quad \text{multi-object exponential}
Inference for Multi-Object HMM

GLMB filtering recursion [VV 13]

\[ w_k^{(\xi, \theta_k)}(I_{k-1}) = 1 \mathcal{F}(\mathbb{B}_k \cup I_{k-1})(\mathcal{D}(\theta_k)) \left[ w_{k|k-1}^{(\xi, \theta_k)} \right] w_{k-1}^{(\xi)}(I_{k-1}) \]

\[ p_k^{(\xi, \theta_k)}(x, \ell) = \begin{cases} \frac{\Lambda_{B,k}^{(\theta_\ell,k)}(x,\ell)}{\Lambda_B^{(\theta_\ell,k)}(\ell)}, & s(\ell) = k \\ \frac{\Lambda_{S,k|k-1}^{(\theta_\ell)}(x_{s(\ell)} \in \ell) p_{k-1}(\xi)}{\Lambda_{S,k|k-1}^{(\xi)}(\ell)}, & t(\ell) = k > s(\ell) \end{cases} \]

GLMB Posterior Recursion [VV 18]

\[ w_{0:k}^{(\xi, \theta_k)}(I_{0:k-1}) = 1 \mathcal{F}(\mathbb{B}_k \cup I_{k-1})(\mathcal{D}(\theta_k)) \left[ w_{k|k-1}^{(\xi, \theta_k)} \right] w_{0:k-1}^{(\xi)}(I_{0:k-1}) \]

\[ p_{0:k}^{(\xi, \theta_k)}(x_{s(\ell)} \in \ell; t(\ell); \ell) = \begin{cases} \frac{\Lambda_{B,k}^{(\theta_\ell,k)}(x_{t(\ell)},\ell)}{\Lambda_B^{(\theta_\ell,k)}(\ell)}, & s(\ell) = k \\ \frac{\Lambda_{S,k|k-1}^{(\theta_\ell)}(x_{k-1} \in \ell) p_{0:k-1}(x_{s(\ell)} \in \ell)}{\Lambda_{S,k|k-1}^{(\xi)}(\ell)}, & t(\ell) = k > s(\ell) \\ \frac{\Lambda_{S,k|k-1}^{(\xi)}(\ell)}{Q_{S,k-1}^{(\xi)}(\ell)}, & t(\ell) = k - 1 \\ p_{0:k-1}^{(\xi)}(x_{s(\ell)} \in \ell; t(\ell); \ell), & t(\ell) < k - 1 \end{cases} \]
Growing no. components: requires component reduction

Truncating smallest components minimizes L1-norm error [VVP 14]

Filter Implementation:
- K-shortest path prediction & ranked assignment update [VVP 14]
- Quadratic in no. components, Cubic/Quartic in no. measurements

Smoother Implementation:
- Multi-dimensional assignment problem: NP-Hard
Example Applications

Autonomous Driving: SLAM (Simultaneous Localisation & Mapping)

Objective: Jointly estimate robot pose & map (set of landmarks)

RFS-SLAM  [Mullane et. al. 11]
Autonomous Driving: **SLAM + multi-object filtering**

Prototype system with E-Class Mercedes-Benz

[Reuter et. al 15, 17]
Example Applications

Autonomous Driving: **SLAM + multi-object filtering**

[Reuter et. al 15, 17]
Example Applications

- POMPD for sensor scheduling
- **Aim:** track an unknown and time varying number of targets
- Tracks closer targets better
- Performance can be improved by maneuvering sensor
- How to “best” maneuver the sensor?
- Challenges: False positives/negatives, unknown data association

- POMDP with information state = filtering distribution
- Model Predictive Control: Objective = Expected Information gain (between prediction & filtering distributions)
**Objective function:** Information gain [Beard et. al. 2017]

**Constraints:** Can’t get too close to targets

How to formulate this?
Void probability constraint

- $\text{VoidPr}(T) = \Pr(T \text{ contains no points of } X)$
- Necessary & Sufficient stats (Renyi)
- Evaluated in close form for for GLMBs
Example Applications

- **Objective function**: Information gain [Beard et. al. 2017]
- **Constraints**: Can’t get too close to targets [Beard et. al. 2017]
  
  Void probability constraint of 0.956 within 500m of ownship
Multi-object estimation: very high-dimensional inference problem

- current techniques: at best hundreds of objects per frame

**Problem size** = no. objects/observations per frame

- easy to track arbitrarily large no. in scenarios with 1-2 objects per frame

Large-scale multi-object filtering: transcomputational problem

Assume no misdetections:

- 310 observations per frame: \( \sim 10^{93} \) possible combinations
- 1000 observations per frame: \( \sim 10^{301} \) possible combinations

\( 10^{80} = \text{total no. atoms in the known universe} \)
\( 10^{93} = \text{no. bits processes by a hypothetical computer the size of the Earth over a period equal to estimated age of the Earth.} \)

**Cubic/Quartic** complexity of Murty’s algorithm not adequate
Gibbs Sampling

Joint Prediction & Update

\[ \pi_k(X) \propto \Delta(X) \sum_{I, \xi, J, \theta} \omega_{k-1}^{(1, \xi)} \omega_k^{(1, \xi, J, \theta)}(Z_k) \delta_J(L(X)) \left[ p_k^{(\xi, \theta)}(\cdot | Z_k) \right]^X \]

- **Truncation by sampling** \( \{(I^{(i)}, \xi^{(i)}, J^{(i)}, \theta^{(i)})\}_{i=1}^{H_{k,\text{max}}} \) from some distribution \( \pi \)
- **Want significant components to have higher selection chance**
  - Natural choice: set \( \pi(I, \xi) \propto \omega_{k-1}^{(1, \xi)} \), \( \pi(J, \theta | I, \xi) \propto \omega_k^{(1, \xi, J, \theta)}(Z_k) \) to give
    \[ \pi(I, \xi, J, \theta) \propto \omega_{k-1}^{(1, \xi)} \omega_k^{(1, \xi, J, \theta)}(Z_k) \]
  - To draw \( H_{k,\text{max}} \) samples from \( \pi \):
    1. \( \{(I^{(h)}, \xi^{(h)})\}_{h=1}^{H_{k,\text{max}}} \sim \pi(I, \xi) \propto \omega_{k-1}^{(1, \xi)} \)
    2. for each sample \( (I^{(h)}, \xi^{(h)}) \)
       \[ (J^{(h)}, \theta^{(h)}) \sim \pi(J, \theta | I^{(h)}, \xi^{(h)}) \propto \omega_k^{(I^{(h)}, \xi^{(h)}, J, \theta)}(Z_k) \]
Fix \((I, \xi)\) and sampling from

\[
\pi(I, \xi) = \omega^{(I, \xi)}(Z_k) \propto \prod_{i=1}^{\left| I_{\cup B_k} \right|} \eta_i(\gamma_i)
\]

\((\gamma_1, \ldots, \gamma_i, \ldots, \gamma_{I_{\cup B_k}})\) positive 1-1 indicator (1-1 when positive)

\[
\gamma_i = j \iff j \text{ is assigned to } i\text{-th label (}j = 0 \iff \text{misdeletion}, j = -1 \iff \text{death})
\]

Analytic conditionals:

\[
\pi(\gamma_i = j \mid \gamma_{i-1}, \gamma_{i+1: I_{\cup B_k}}, I, \xi) \propto \begin{cases} 
\eta_i(j) & \text{if } j \text{ is non-positive or has not been taken up} \\
0 & \text{otherwise}
\end{cases}
\]

Generate significant GLMB components via Gibbs sampling: all iterates are positive 1-1

Burn-ins do not affect the resulting filtering density.
Gibbs Sampling

Sampling from a (unnormalized) distribution $\pi$ by constructing a Markov Chain

$$\gamma^{(1)} \rightarrow \gamma^{(2)} \rightarrow \cdots \rightarrow \gamma^{(t)} \rightarrow \gamma^{(t+1)}$$

Given $\gamma^{(t-1)} = (\gamma_1^{(t-1)}, \ldots, \gamma_P^{(t-1)})$, construct $\gamma^{(t)} = (\gamma_1^{(t)}, \ldots, \gamma_P^{(t)})$ by:

$$\gamma_1^{(t)} \sim \pi(\cdot \mid \gamma_{2:P}^{(t-1)}) = \frac{\pi(\cdot, \gamma_{2:P}^{(t-1)})}{\sum_j \pi(j, \gamma_{2:P}^{(t-1)})}$$

$$\gamma_2^{(t)} \sim \pi(\cdot \mid \gamma_1^{(t)}, \gamma_{3:P}^{(t-1)}) = \frac{\pi(\gamma_1^{(t)}, \cdot, \gamma_{3:P}^{(t-1)})}{\sum_j \pi(\gamma_1^{(t)}, j, \gamma_{3:P}^{(t-1)})}$$

$$\gamma_i^{(t)} \sim \pi(\cdot \mid \gamma_{1:i-1}^{(t)}, \gamma_{i+1:P}^{(t-1)}) = \frac{\pi(\gamma_{1:i-1}^{(t)}, \cdot, \gamma_{i+1:P}^{(t-1)})}{\sum_j \pi(\gamma_{1:i-1}^{(t)}, j, \gamma_{i+1:P}^{(t-1)})}$$

$$\gamma_P^{(t)} \sim \pi(\cdot \mid \gamma_{1:P-1}^{(t)}) = \frac{\pi(\gamma_{1:P-1}^{(t)}, \cdot)}{\sum_j \pi(\gamma_{1:P-1}^{(t)}, j)}$$
Gibbs Sampling

- Tempering for sample diversity: introduce more uncertainty

\[
\pi(\gamma \mid I, \xi) \propto 1_\Gamma(\gamma) \prod_{i=1}^{\left|I \cup B_k\right|} \tilde{\eta}_i(\gamma_i)
\]

- Modify model parameters to increase uncertainty or increase the temperature
  - Cheap: MCMC with acceptance probability = 1
  - All iterates are positive 1-1
  - Exponential convergence
  - Linear complexity in \( |Z_k| \)
  - Can be used to solve ranked assignment problems

- Burn-ins do not affect the resulting filtering density

- Quadratic in no. components, linear in no. measurements
  - Still not enough for large scale problems
Large Scale Multi-Object Filtering

- Factorization: for a given partition of the label space

\[ \pi_k(X) = \pi_k(X^{(1)} \hat{\&} X^{(2)} \hat{\&} \ldots \hat{\&} X^{(N)}) \]
\[ \approx \pi_k^{(1)}(X^{(1)}) \pi_k^{(2)}(X^{(2)}) \ldots \pi_k^{(N)}(X^{(N)}) \] Minimizes KLD

- Parallelize: compute the marginals in parallel

- Best partition for a given cap on the size of constituent subsets? Infeasible!

- Exploit weak correlations between spatially separated regions

Clustering:
- segment-tree data structure
- finding intersecting bounding regions
- \( O(M \log_d M + K) \)
large scale multi-object filtering

- surveillance region: 64km by 36km (inset 2km by 1km)
- objects appear anywhere in the surveillance region, rate 3000/frame (unknown to filter)
- clutter density 200 km$^{-2}$ (460800 per frame) uniform, detection probability = 0.95
- linear Gaussian state dynamic, noise sigma = 0.2ms$^{-2}$
- linear Gaussian observation, noise sigma = 5m
Large Scale Multi-Object Filtering

- Over 1 million objects per frame
- Peak cardinality: 1,217,531 objects per frame (at time 700)
- (1 million observations: \( \sim 10^{3010^{30}} \) combinations
- Peak object density: 520 km\(^{-2}\)
- Duration 1000 instances
  \( \sim 1 \) billion data points

[Beard et. al. 18]
**“Deep” Multi-Object Smoothing**

**Multi-Object** smoother: solve multi-dimensional assignment problem

\[ \max_{\gamma_{0:k}} w_{0:k}^{(\gamma_{0:k})} \]

\[ w_{0:k}^{(\gamma_{0:k})} = \prod_{i=1}^{k} 1_{\Gamma_i} (\gamma_i) \eta_i^{(\gamma_{0:i}(-))} (\mathcal{L}(\gamma_i)) \left[ \mathcal{B}_i \cup \mathcal{L}(\gamma_{i-1}) \right] \]

\( \Gamma_i = \) space of maps \( \gamma_i : L_i \rightarrow \{-1, 0, \ldots, M_i\} \) that are 1-1 on \( \{1, \ldots, M_i\} \)

- Exponential in \( k \) & Factorial in \( M_i \) \[ \text{[Pasiliao 2010]} \]
- Efficient, Scalable, State-of-the-Arts \[ \text{[Nguyen et. al. 14]} \]

*Large-scale problem: 5-D, 20 elements per dimension*

*Equivalent to a 0-1 LP with 3.2M variables & 100 constraints.*

- Extend Gibbs Sampler (for 2-D) to multi-dimensional assignment:
  - Same algorithm (but different cost matrix) + 2 checks:
  - Looking backward 1 step: dead object \((-1)\) cannot be alive \((\geq 0)\) now
  - Looking forward 1 step: live object \((\geq 0)\) cannot be dead \((-1)\) now.
Example: standard linear Gaussian multi-object model with

- $P_D = 0.77$
- average of 66 false alarms per scan
- 100 scan (100-D assignment problem, ~ 60 variables per dimension)
- Much slower than filtering
Conclusions

- Discussed smoothing/filtering for multi-object HMM
  - Both problems are very high dimensional
  - Used Gibbs sampling to solve multi-dimensional assignment problem
  - Solve 100-D assignment problem for smoothing
  - Solve large-scale 2-D assignment problem for filtering

- More efficient simulation-based methods
  - Not interested in the actual distribution of the samples
  - Interested in the best (high-weigh) samples
  - Don’t want the Gibbs sampler (or MC) to revisit previous states
  - Is there a way to do this?

Thank You!
Large Scale Multi-Object Filtering

- surveillance region: 64km by 36km
- objects appear anywhere in the surveillance region, rate 3000/frame (unknown to filter)
- clutter density 200 km\(^{-2}\) (460800 per frame), detection probability = 0.95
- linear Gaussian state dynamic, noise sigma = 0.2ms\(^{-2}\)
- linear Gaussian observation, noise sigma = 5m
Large Scale Multi-Object Filtering

- surveillance region: 64km by 36km
- objects appear anywhere in the surveillance region, rate 3000/frame (unknown to filter)
- clutter density 200 km\(^{-2}\) (460800 per frame) uniform, detection probability = 0.95
- linear Gaussian state dynamic, noise sigma = 0.2ms\(^{-2}\)
- linear Gaussian observation, noise sigma = 5m
Labelling convention (ensures objects have distinct labels)

Object label: \( \ell = (k, i) \in \mathbb{L}_k = \{k\} \times \mathbb{N} \)

(space of labels born at time \( k \))

time of birth  object number/name

Space of (all) labels at time \( k \):

\[ \mathbb{L} = \mathbb{L}_{0:k} = \mathbb{L}_{0:k-1} \cup \mathbb{L}_k \]

State space

\( \begin{align*}
X_1 & \quad (1,2) \\
X_4 & \quad (4,1) \\
X_k & \quad \text{…} \\
\end{align*} \)

\( X_{0:k} \)

labeled trajectories

\( \text{time} \)