Recap: Particle Filtering

- Many variations, one general concept:
  - Represent the posterior pdf by a set of randomly chosen weighted samples (particles)
  - Randomly chosen = Monte Carlo (MC)
  - As the number of samples become very large – the characterization becomes an equivalent representation of the true pdf.

Recap: Sequential Importance Sampling

\[
\text{function } \{x_i^t, w_i^t\}_{i=1}^N = \text{SIS } \{x_{i-1}, w_{i-1}\}_{i=1}^N, y_t
\]

\[
\eta = 0 \quad \text{Initialize}
\]

for \(i = 1 : N\)

\[
x_i^t \sim q(x_i | x_{i-1}, y_t)
\]

Sample from proposal pdf

\[
w_i^t = w_{i-1}^t \frac{p(y_t | x_t) p(x_t | x_{i-1})}{q(x_i | x_{i-1}, y_t)}
\]

Update weights

\[
\eta = \eta + w_i^t
\]

Update norm. factor

end

for \(i = 1 : N\)

\[
w_i^t = w_i^t/\eta
\]

Normalize weights

Recap: SIS Algorithm with Transitional Prior

\[
\text{function } \{x_i^t, w_i^t\}_{i=1}^N = \text{SIS } \{x_{i-1}, w_{i-1}\}_{i=1}^N, y_t
\]

\[
\eta = 0 \quad \text{Initialize}
\]

for \(i = 1 : N\)

\[
x_i^t \sim p(x_t | x_{i-1})
\]

Sample from proposal pdf

\[
w_i^t = w_{i-1}^t \frac{p(y_t | x_t) p(x_t | x_{i-1})}{q(x_i | x_{i-1}, y_t)}
\]

Update weights

\[
\eta = \eta + w_i^t
\]

Update norm. factor

end

for \(i = 1 : N\)

\[
w_i^t = w_i^t/\eta
\]

Normalize weights

For a concrete algorithm, we need to define the importance density \(q(. | .)\).
Recap: Resampling

- Degeneracy problem with SIS
  - After a few iterations, most particles have negligible weights.
  - Large computational effort for updating particles with very small contribution to \( p(x_t | y_1:t) \).

- Idea: Resampling
  - Eliminate particles with low importance weights and increase the number of particles with high importance weight.
  - \( \{ x_i^t, w_i^t \}_{i=1}^N \rightarrow \{ x_i^t, \frac{1}{N} \}_{i=1}^N \) – The new set is generated by sampling with replacement from the discrete representation of \( p(x_t | y_1:t) \) such that \( \sum x_i^t = x_i^t \) and \( \sum w_i^t = w_i^t \).

Recap: Efficient Resampling Approach

- From Arulampalam paper:
  - Basic idea: choose one initial small random number; deterministically sample the rest by “crawling” up the cdf.
  - This is \( O(N) \)!

Recap: Sampling-Importance-Resampling (SIR)

\[
\text{function}\ [X_i] = \text{SIR} \left[ X_{i-1}, y_1 \right] \\
\hat{X}_i - X_i = 0 \\
\text{for } i = 1:N \\
\text{Sample } x_i^t \sim p(x_i | x_{i-1}^t) \\
w_i^t = p(y_i | x_i^t) \\
\text{end} \\
\text{for } i = 1:N \\
\text{Draw } i \text{ with probability } \propto w_i^t \\
\text{Add } x_i^t \text{ to } X_i \\
\text{end}
\]

Important property:
- Particles are distributed according to pdf from previous time step.
- Particles are distributed according to posterior from this time step.

Today: Multi-Object Tracking

[Ess, Leibe, Schindler, Van Gool, CVPR’08; ICRA’09; PAMI’09]
Topics of This Lecture

- Multi-Object Tracking
  - Motivation
  - Ambiguities
- Simple Approaches
  - Gating
  - Mahalanobis distance
  - Nearest-Neighbor Filter
- Track-Splitting Filter
  - Derivation
  - Properties
- Outlook

**Elements of Tracking**

- Detection
  - Where are candidate objects?
  - Lectures 2-6
- Data association
  - Which detection corresponds to which object?
  - Today’s topic
- Prediction
  - Where will the tracked object be in the next time step?
  - Lectures 7-9

Motion Correspondence

- Motion correspondence problem
  - Do two measurements at different times originate from the same object?
- Why is it hard?
  - First make predictions for the expected locations of the current set of objects
  - Match predictions to actual measurements
  - This is where ambiguities may arise...

Motion Correspondence Ambiguities

1. Predictions may not be supported by measurements
   - Have the objects ceased to exist, or are they simply occluded?
2. There may be unexpected measurements
   - Newly visible objects, or just noise?
3. More than one measurement may match a prediction
   - Which measurement is the correct one (what about the others)?
4. A measurement may match to multiple predictions
   - Which object shall the measurement be assigned to?

Let’s Formalize This

- Multi-Object Tracking problem
  - We represent a track by a state vector $x$, e.g.,
    
    $x = [x_1, y_1, v_x, v_y]^T$
  - As the track evolves, we denote its state by the time index $k$:
    
    $x^{(k)} = \begin{bmatrix} x_1^{(k)} & y_1^{(k)} & v_x^{(k)} & v_y^{(k)} \end{bmatrix}^T$
  - At each time step, we get a set of observations (measurements)
    
    $y^{(k)} = \begin{bmatrix} y_1^{(k)} & \ldots & y_N^{(k)} \end{bmatrix}$
  - We now need to make the data association between tracks
    
    $\{x_1^{(k)}, \ldots, x_N^{(k)}\}$ and observations $\{y_1^{(k)}, \ldots, y_N^{(k)}\}$:
    
    $i_j = j$ if $y_j^{(k)}$ is associated with $x_i^{(k)}$
Reducing Ambiguities: Simple Approaches

• Gating
  – Only consider measurements within a certain area around the predicted location.
  ⇒ Large gain in efficiency, since only a small region needs to be searched

• Nearest-Neighbor Filter
  – Among the candidates in the gating region, only take the one closest to the prediction \( x_o \).
  \[ z_j^{(k)} = \underset{z_j}{\text{arg min}} |x_j^{(k)} - y_j^{(k)}| \]
  – Better: the one most likely under a Gaussian prediction model
  \[ z_j^{(k)} = \underset{z_j}{\text{arg max}} \mathcal{N}(y_j^{(k)}, x_j, \Sigma_p) \]
  which is equivalent to taking the Mahalanobis distance

Gating with Mahalanobis Distance

• Recall: Kalman filter
  – Provides exactly the quantities necessary to perform this
  – Predicted mean location \( x_p \)
  – Prediction covariance \( \Sigma_p \)
  – The Kalman filter prediction covariance also defines a useful gating area.
  ⇒ E.g., choose the gating area such that 95% of the probability mass is covered.

• Side note
  – The Mahalanobis distance is \( \chi^2 \) distributed with the number of degrees of freedom \( n_z \) equal to the dimension of \( x \).
  – For a given probability bound, the corresponding threshold on the Mahalanobis distance can be obtained from \( \chi^2 \) distribution tables.

Mahalanobis Distance

• Additional notation
  – Our KF state of track \( x_l \) is given by the prediction \( x_l^{(k)} \) and covariance \( \Sigma_l^{(k)} \).
  – We define the innovation that measurement \( y_j \) brings to track \( x_l \) at time \( k \) as
  \[ v_j^{(k)} = (y_j^{(k)} - x_l^{(k)}) \]
  – With this, we can write the observation likelihood shortly as
  \[ p(y_j^{(k)} | x_l^{(k)}) = \exp \left\{ \frac{1}{2} \sum_{i,j} v_j^{(k)} \Sigma_{ji}^{-1} v_j^{(k)} \right\} \]
  – We define the ellipsoidal gating or validation volume as
  \[ v^{(k)}(\gamma) = \left\{ y | (y - x_l^{(k)})^T \Sigma_l^{(k)-1} (y - x_l^{(k)}) \leq \gamma \right\} \]

Problems with NN Assignment

• Limitations
  – For NN assignments, there is always a finite chance that the association is incorrect, which can lead to serious effects.
  ⇒ If a Kalman filter is used, a misassigned measurement may lead the filter to lose track of its target.
  – The NN filter makes assignment decisions only based on the current frame.
  ▪ More information is available by examining subsequent images.
  ⇒ Let’s make use of this information by postponing the decision process until a future frame will resolve the ambiguity...

Topics of This Lecture

• Multi-Object Tracking
  – Motivation
  – Ambiguities

• Simple Approaches
  – Gating
  – Mahalanobis distance
  – Nearest-Neighbor Filter

• Track-Splitting Filter
  – Derivation
  – Properties

• Outlook

Track-Splitting Filter

• Idea
  – Problem with NN filter was hard assignment.
  – Rather than arbitrarily assigning the closest measurement, form a tree.
  – Branches denote alternate assignments.
  – No assignment decision is made at this stage!
  ⇒ Decisions are postponed until additional measurements have been gathered...

• Potential problems?
  – Track trees can quickly become very large due to combinatorial explosion.
  ⇒ We need some measure of the likelihood of a track, so that we can prune the tree!
Track Likelihoods

• Expressing track likelihoods
  – Given a track $l$, denote by $\theta_l$ the event that the sequence of assignments
    \[ Z_{k,t} = \{ z_{1,t}, \ldots, z_{N_{k},t} \} \]
    from time 1 to $k$ originate from the same object.
  – The likelihood of $\theta_l$ is the joint probability over all observations in the track
    \[ L(\theta_{l,t}) = \prod_{j=1}^{k} \frac{1}{(2\pi)^{\frac{N_j}{2}}} \exp\left( -\frac{1}{2} \sum_{j=1}^{k} v_{j,t}^T \Sigma_{j,t}^{-1} v_{j,t} \right) \]
    \[ \mathcal{N}(0, \Sigma) \]
  – If we assume Gaussian observation likelihoods, this becomes
    \[ L(\theta_{l,t}) = \prod_{j=1}^{k} \frac{1}{(2\pi)^{\frac{N_j}{2}}} \exp\left( -\frac{1}{2} \sum_{j=1}^{k} v_{j,t}^T \Sigma_{j,t}^{-1} v_{j,t} \right) \]

Track Likelihoods (2)

• Starting from the likelihood
    \[ L(\theta_{l,t}) = \prod_{j=1}^{k} \frac{1}{(2\pi)^{\frac{N_j}{2}}} \exp\left( -\frac{1}{2} \sum_{j=1}^{k} v_{j,t}^T \Sigma_{j,t}^{-1} v_{j,t} \right) \]
  – Define the modified log-likelihood $\lambda_l$ for track $l$ as
    \[ \lambda_l(k) = -2 \log \left( \prod_{j=1}^{k} \frac{1}{(2\pi)^{\frac{N_j}{2}}} \exp\left( -\frac{1}{2} \sum_{j=1}^{k} v_{j,t}^T \Sigma_{j,t}^{-1} v_{j,t} \right) \right) \]
    \[ = \lambda_l(k) = -2 \log \left( \prod_{j=1}^{k} \frac{1}{(2\pi)^{\frac{N_j}{2}}} \exp\left( -\frac{1}{2} \sum_{j=1}^{k} v_{j,t}^T \Sigma_{j,t}^{-1} v_{j,t} \right) \right) \]
  \[ \Rightarrow \text{Recursive calculation, sum of Mahalanobis distances of all the measurements assigned to track } l. \]

Track-Splitting Filter

• Effect
  – Instead of assigning the measurement that is currently closest, as in the NN algorithm, we can select the sequence of measurements that minimizes the total Mahalanobis distance over some interval!
  – Modified log-likelihood provides the merit of a particular node in the track tree.
  – Cost of calculating this is low, since most terms are needed anyway for the Kalman filter.
• Problem
  – The track tree grows exponentially, may generate a very large number of possible tracks that need to be maintained.

Pruning Strategies

• In order to keep this feasible, need to apply pruning
  – Deleting unlikely tracks
    • May be accomplished by comparing the modified log likelihood $\lambda_l(k)$, which has a chi-square distribution with $kN$ degrees of freedom, with a threshold $\alpha$ (set according to chi-square distribution tables).
    • Problem for long tracks: modified log-likelihood gets dominated by old terms and responds very slowly to new ones.
  \[ \Rightarrow \text{Use sliding window or exponential decay term.} \]
  – Merging track nodes
    • If the state estimates of two track nodes are similar, merge them.
    • E.g., if both tracks validate identical subsequent measurements.
  – Only keeping the most likely $N$ tracks
    • Rank tracks based on their modified log-likelihood.

Summary: Track-Splitting Filter

• Properties
  – Very old algorithm
    • Improvement over NN assignment.
    • Assignment decisions are delayed until more information is available.
• Many problems remain
  – Exponential complexity, heuristic pruning needed.
  – Merging of track nodes is necessary, because tracks may share measurements, which is physically unrealistic.
  \[ \Rightarrow \text{Would need to add exclusion constraints such that each measurement may only belong to a single track.} \]
  \[ \Rightarrow \text{Impossible in this framework...} \]

Topics of This Lecture

• Multi-Object Tracking
  – Motivation
  – Ambiguities
• Simple Approaches
  – Gating
  – Mahalanobis distance
  – Nearest Neighbor Filter
• Track-Splitting Filter
  – Derivation
  – Properties
• Outlook
Outlook for the Next Lectures

• More powerful approaches
  – Multi-Hypothesis Tracking (MHT)
    • Well-suited for KF, EKF approaches [Reid, 1979]
  – Joint Probabilistic Data Association Filters (JPDAF)
    • Well-suited for Particle Filter based approaches [Fortmann, 1983]

• Data association as convex optimization problem
  – Bipartite Graph Matching (Hungarian algorithm)
  – Network Flow Optimization
    ⇒ Efficient, globally optimal solutions for subclass of problems.

References and Further Reading

• A good tutorial on Data Association