Contents in latter part

Linear Dynamical Systems
  What is different from HMM?
Kalman filter
  Its strength and limitation
Particle Filter
  Its simple algorithm and broad application
Dynamical Systems

State space model of Dynamical systems

Now latent variable is continuous rather than discrete

\[
\begin{align*}
    z_n &= F(z_{n-1}, w) & \text{State equation} \\
    x_n &= H(z_n, \nu) & \text{Observation equation}
\end{align*}
\]
Dynamical Systems

State space model of Dynamical systems

Now latent variable is continuous rather than discrete

\[
\begin{align*}
    z_n &= F(z_{n-1}, w) \\
    x_n &= H(z_n, v)
\end{align*}
\]

or

\[
\begin{align*}
    p(z_n | z_{n-1}) \\
    p(x_n | z_n)
\end{align*}
\]
Linear Dynamical Systems

Special case of dynamical systems

Gaussian assumption on distribution

\[
\begin{align*}
    z_n &= Az_{n-1} + w_n \\
    x_n &= Cz_n + v_n \\
    z_1 &= \mu_0 + u
\end{align*}
\]

\( w \sim \mathcal{N}(w|0, \Gamma) \)

\( v \sim \mathcal{N}(v|0, \Sigma) \)

\( u \sim \mathcal{N}(u|0, V_0) \).

Where
\( z \): Latent variable
\( x \): Observation
\( A \): Dynamics matrix
\( C \): Emission matrix
\( w, v, u \): Noise
Linear Dynamical Systems

Bayesian form of LDS

\[
p(z_n|z_{n-1}) = \mathcal{N}(z_n|Az_{n-1}, \Gamma)
\]

\[
p(x_n|z_n) = \mathcal{N}(x_n|Cz_n, \Sigma).
\]

\[
p(z_1) = \mathcal{N}(z_1|\mu_0, V_0)
\]

Since LDS is linear Gaussian model, joint distribution over all latent and observed variables is simply Gaussian.
Kalman Filter

• Kalman filter does exact inference in LDS in which all latent and observed variables are Gaussian (incl. multivariate Gaussian).

• Kalman filter handles multiple dimensions in a single set of calculations.

• Kalman filter has two distinct phases: Predict and Update
Application of Kalman Filter

Tracking moving object

Blue: True position
Green: Measurement
Red: Post. estimate
Two phases in Kalman Filter

Predict

Prediction of state estimate and estimate covariance

Update

Update of state estimate and estimate covariance with Kalman gain
**Estimation of Parameter in LDS**

Distribution of $Z_{n-1}$ is used as a prior for estimation of $Z_n$

Blue: $p(z_{n-1}|x_1, \ldots, x_{n-1})$

Red: $p(z_n|x_1, \ldots, x_{n-1})$

Green: $p(x_n|z_n)$

Blue: $p(z_n|x_1, \ldots, x_n)$
Derivation of Kalman Filter

• We use sum-product algorithm for efficient inference of latent variables.

• LDS is continuous case of HMM (sum -> integer)

\[
c_n \hat{\alpha}(z_n) = p(x_n|z_n) \sum_{z_{n-1}} \hat{\alpha}(z_{n-1}) p(z_n|z_{n-1})
\]

\[
c_n \hat{\alpha}(z_n) = p(x_n|z_n) \int \hat{\alpha}(z_{n-1}) p(z_n|z_{n-1}) \, dz_{n-1}
\]

where \( \hat{\alpha}(z_n) = \mathcal{N}(z_n | \mu_n, V_n) \)
Sum-product algorithm

\[ c_n \hat{\alpha}(z_n) = p(x_n|z_n) \int \hat{\alpha}(z_{n-1}) p(z_n|z_{n-1}) \, dz_{n-1} \]

where \( \hat{\alpha}(z_n) = \mathcal{N}(z_n|\mu_n, V_n) \)

\[ \begin{align*}
\mu_n &= A\mu_{n-1} + K_n (x_n - CA\mu_{n-1}) \\
V_n &= (I - K_n C) P_{n-1} \\
c_n &= \mathcal{N}(x_n|CA\mu_{n-1}, CP_{n-1}C^T + \Sigma) \\
\end{align*} \]

where \( K_n = P_{n-1}C^T (CP_{n-1}C^T + \Sigma)^{-1} \) (Kalman Gain Matrix)
What we have estimated?

Predict: \[ \mu_n = A \mu_{n-1} \]

Update: \[ \mu_n = A \mu_{n-1} + K_n(x_n - CA\mu_{n-1}) \]
What we have estimated?

Predict: \[ \mu_n = A \mu_{n-1} \]

Update: \[ \mu_n = A \mu_{n-1} + K_n (x_n - CA \mu_{n-1}) \]

Predicted mean of Zn

Predicted mean of Xn

Prediction error

Observed Xn
Limitation of Kalman Filter

• Due to assumption of Gaussian distribution in KF, KF can not estimate well in nonlinear/non-Gaussian problem.

• One simple extension is mixture of Gaussians
  • In mixture of K Gaussians, $\hat{\alpha}(z_1)$ is mixture of K Gaussians, and $\hat{\alpha}(z_n)$ will comprise mixture of $K^n$ Gaussians. -> Computationally intractable
Limitation of Kalman Filter

- To resolve nonlinear dynamical system problem, other methods are developed.
  - Extended Kalman filter: Gaussian approximation by linearizing around the mean of predicted distribution
  - Particle filter: Resampling method, see later
  - Switching state-space model: continuous type of switching HMM
Particle Filter

- In nonlinear/non-Gaussian dynamical systems, it is hard to estimate posterior distribution by KF.

- Apply the sampling-importance-resampling (SIR) to obtain a sequential Monte Carlo algorithm, particle filter.

- Advantages of Particle Filter
  - Simple algorithm -> Easy to implement
  - Good estimation in nonlinear/non-Gaussian problem
How to Represent Distribution

Original distribution (mixture of Gaussian)

Gaussian approximation

Approximation by PF (distribution of particle)

Approximation by PF (histogram)
Where’s a landmine?

Use metal detector to find a landmine (orange star).
Where’s a landmine?

Random survey in the field (red circle).

The filter draws a number of randomly distributed estimates, called particles. All particles are given the same likelihood
Where’s a landmine?

Get response from each point (strength: size).
Assign a likelihood to each particle such that the particular particle can explain the measurement.
Where’s a landmine?

Decide the place to survey in next step.

Scale the weight of particles to select the particle for resampling.
Where’s a landmine?

Intensive survey of possible place

Draw random particles based upon their likelihood (Resampling). High likelihood -> more particle; low likelihood -> less particle

All particle have equal likelihood again.
Operation of Particle Filter
Algorithm of Particle Filter

• Sample representation of the posterior distribution \( p(z_n|X_n) \) expressed as a samples \( \{z^{(l)}_n\} \) with corresponding weights \( \{w^{(l)}_n\} \).

  where \[
  w^{(l)}_n = \frac{p(x_n|z^{(l)}_n)}{\sum_{m=1}^{L} p(x_n|z^{(m)}_n)}
  \]

• Draw samples from mixture distribution \[
  \sum_{l=1}^{L} w^{(l)}_n f(z^{(l)}_n)
  \]
  where \( \sum_{l} w^{(l)}_n = 1 \quad 0 \leq w^{(l)}_n < 1 \)

• Use new observation \( x_{n+1} \) to evaluate the corresponding weights \( w^{(l)}_{n+1} \propto p(x_{n+1}|z^{(l)}_{n+1}) \).
Limitation of Particle Filter

- In high dimensional models, enormous particles are required to approximate posterior distribution.
- Repeated resampling cause degeneracy of algorithm.
  - All but one of the importance weights are close to zero
  - Avoided by proper choice of resampling method