Differentiable Bayes Filters
Announcement

- Feedback for Project proposal latest on Wednesday night
- Kevin Zakka started course notes (see Piazza) - bonus points for contributing
What will you take home today?

Recap Bayesian Filters

- Kalman Filter
- Extended Kalman Filter
- Particle Filter

Differentiable Filters

- Backpropagation through a Kalman Filter
- Backpropagation through a Particle Filter
Recap Bayes Filters

Latent State

\[
p(x_t | z_{1:t}, u_{1:t}, x_0) = \frac{\eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_t) p(x_{t-1} | z_{1:t-1}, u_{1:t-1}) dx_{t-1}}{\int \int \int dx_{t-1} dx_{t-2} \cdots dx_0}
\]
Kalman Filter

\[ x_t = f(x_{t-1}, u_{t-1}, w_t) \]
\[ = \Phi x_{t-1} + B u_t + \omega_t \]
\[ x \sim N(x, P) \]
\[ \sim N(\mu, \Sigma) \]

Observation model
\[ z_t = h(x_t, v_t) \]
\[ = H x_t + v_t \]
\[ w \sim N(0, Q) \]
\[ v \sim N(0, R) \]
Kalman Filter

Predictive Step

\[
\hat{x}_t = \Phi \hat{x}_{t-1} + B u_t
\]
\[
\hat{P}_t = \Phi \hat{P}_{t-1} \Phi^T + Q_t
\]

Update Step:

\[
\hat{z}_t = z_t - H \hat{x}_t \quad \text{(Innovation)}
\]
\[
K_t = \frac{\hat{P}_t H^T}{H \hat{P}_t H^T + R_t}
\]
\[
\text{measured/ noise uncertainty above predicted state}
\]
\[
\hat{x}_t = \hat{x}_t + K_t \hat{z}_t / \hat{P}_t \left( I_n - K_t H \right) \hat{P}_t
\]
Extended Kalman Filter

**Prediction Step:**

\[ \hat{x}_t = Ax_{t-1} + Bu_t \]  
\[ \hat{P}_t = AP_{t-1}A^T + Q_t \]  
\[ i_t = z_t - H\hat{x}_t \]  
\[ K_t = \frac{\hat{P}_t H^T}{H\hat{P}_t H^T + R_t} \]  
\[ x_t = \hat{x}_t + K_t i_t \]  
\[ P_t = (I_n - K_t H)\hat{P}_t \]

**Update Step:**

\[ \hat{z}_t = h(\hat{x}_t, \theta) \]

\[ [\hat{x}_t] \]

\[ [H] \]

- \( \theta \) due nonlinear models

KF assumes linear models

\( \Rightarrow \) Jacobians for nonlinear models
Particle Filter

\[
p(x_t | z_{1:t}, u_{1:t}, x_0) \]

\[
\begin{align*}
\chi_t &= \{x^0_t, x^1_t, \ldots, x^n_t\} \\
\omega_t &= \{w^0_t, w^1_t, \ldots, w^n_t\}
\end{align*}
\]
Particle Filter

\( X_0 \)

\( X_t = f(X_{t-1}, u_{t-1}, w_t) \)  

\( Z_t \rightarrow \text{sensor observation} \)

\( p(Z_t | x_t) \rightarrow \text{how likely has each particle produced } Z_t \)

\( x^i_t, w^i_t = w_{t-1} p(Z_t | x^i_t) \)

\( \rightarrow \text{Resample particles} \)

\( \rightarrow \text{Roberto's wheel of death} \)
When to use what? How to choose hyperparameters?
Priors and Hyperparameters

A lot of hardcoded knowledge!

1. State Representation

2. Models
   - Forward Model
     - State to next state
     - Action to next state
   - Measurement Model

3. Probabilistic Properties
   - Process Noise
   - Measurement Noise
Differentiable filters

Can we learn model & the noise variance from data?

Approach: Embed algorithmic structure of Bayesian Filtering into a recurrent neural network

→ prevents overfitting through regularization

→ avoid manual tuning & modeling

→ Explanability because maintains explicit state representations
A note on variables

In Robotics and Control:
\[ (\mu, \Sigma) \]

In Machine Learning:
- \( z \): state
- \( x \): observation
- \( u \): control input

In Artificial Intelligence:
- \( s \): state
- \( o \): observation
- \( a \): action

- Differentiable version of the Kalman Filter
- Uses Images as observations; learns a sensors that outputs state directly
Differentiable Kalman Filter - Structure
Differentiable Kalman Filter - Structure

$x \sim \mathcal{N}(\mu, \Sigma)$

$z = \text{observation}$

$\hat{z}_t = C\hat{\mu}_t$

$C = I$

$\text{MSE loss over each batch and sequence}$
Differentiable Kalman Filter – Loss Function

\[
L(\mathbf{x}_0: T, \mu_0: T, \Sigma_0: T, \omega) = \\
\sum_{t=0}^{T} \frac{1}{2} ((\mathbf{x}_t - \mu_t)\mathbf{S}_t^{-1}(\mathbf{x}_t - \mu_t) + \log(|\mathbf{S}_t|)) \\
+ \lambda_2 \sum_{t=0}^{T} \|\mathbf{x}_t - \mu_t\|_2^2 \\
+ \lambda_3 \mathbf{w} \|\mathbf{w}\|_2
\]

\[
\rightarrow \text{negative loglikelihood of the true state given } \mu, \Sigma
\]

\[
\rightarrow \text{Euclidean error between } \mathbf{x} \& \mu
\]

\[
\rightarrow \text{Regularizers}
\]

Baseline: piecewise KF \(\Rightarrow\) directly train the CNN to predict \(\mathbf{x}\)
Table 1: Benchmark Results

<table>
<thead>
<tr>
<th>State Estimation Model</th>
<th># Parameters</th>
<th>RMS test error ±σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>feedforward model</td>
<td>7394</td>
<td>0.2322 ± 0.1316</td>
</tr>
<tr>
<td>piecewise KF</td>
<td>7397</td>
<td>0.1160 ± 0.0330</td>
</tr>
<tr>
<td>LSTM model (64 units)</td>
<td>33506</td>
<td>0.1407 ± 0.1154</td>
</tr>
<tr>
<td>LSTM model (128 units)</td>
<td>92450</td>
<td>0.1423 ± 0.1352</td>
</tr>
<tr>
<td>BKF (ours)</td>
<td>7493</td>
<td>0.0537 ± 0.1235</td>
</tr>
</tbody>
</table>
Differentiable Kalman Filter – Experiments and Baselines

• Kitti – Visual Odometry Dataset
• 22 stereo sequences with LIDAR
  • 11 sequences with ground truth (GPS/IMU data)
  • 11 sequences without ground truth (for evaluation)
Differentiable Kalman Filter – Experiments and Baselines

Results reproduced by Claire Chen
Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. Jonschkowski et al. RSS 2018.
Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. Jonschkowski et al. RSS 2018.
Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. Jonschkowski et al. RSS 2018.
Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. Jonschkowski et al. RSS 2018.
Differentiable Particle Filter – Loss Function

\[ \theta^* = \arg \min_\theta - \log E_t[\text{bel}(s^*_t; \theta)]. \]
Differentiable Particle Filter – Experiments and Baselines

Fig. 5: **Learned motion model.** (a) shows predictions (cyan) of the state (red) from the previous state (black). (b) compares prediction uncertainty in x to true odometry noise (dotted line).

Fig. 6: **Learned measurement model.** Observations, corresponding model output, and true state (red). To remove clutter, the observation likelihood only shows above average states.
Differentiable Particle Filter – Experiments and Baselines

Fig. 9: Generalization between policies in maze 2. A: heuristic exploration policy, B: shortest path policy. Methods were trained using 1000 trajectories from A, B, or an equal mix of A and B, and then tested with policy A or B.
Differentiable Particle Filter – Experiments and Baselines

(a) Visual input (image and difference image) at time steps 100, 200, and 300 (indicated in (b) by black circles)

(b) Trajectory 9; starts at (0,0)

Fig. 10: Visual odometry with DPFs. Example test trajectory