CSE-571
Probabilistic Robotics

Gaussian Processes for Bayesian Filtering

Dieter Fox
University of Washington

Overview

• Gaussian Processes and Bayes Filters
  • GP-BayesFilters
• Filtering and Control
• System Identification with GP-BayesFilters
• Predictive State Representations
• Conclusions

GP-BayesFilters

• Learn GP:
  – Input: Sequence of ground truth states along with controls and observations: \( <s, u, z> \)
  – Learn GPs for dynamics and observation models
• Filters
  – Particle filter: sample from dynamics GP, weigh by GP observation function
  – EKF: GP for mean state, GP derivative for linearization
  – UKF: GP for sigma points

Amazon Dieter Fox: GP-BayesFilters

[Jo-F: RSS-08, ARJ-09]
Learning GP Dynamics and Observation Models

- Ground truth training sequence: 
  \[ S = \{x_1, x_2, \ldots, x_n\}, Z = \{z_1, z_2, \ldots, z_n\}, U = \{u_1, u_2, \ldots, u_n\} \]
- Learn observation and dynamics GPs:
  \[ \text{GP dynamics model} \]
  \[ \Delta s_k = s_{k+1} - s_k \]
  \[ \text{GP observation model} \]
  \[ r_k = \Delta s_k - f(s_k, u_k) \]
- Learn separate GP for each output dimension
- Diagonal noise matrix

[Deisenroth et al.] introduced GP-ADFs and EP for smoothing in GP dynamical systems

GP-PF Propagation

- Propagate each particle using GP prediction
- Sample from GP uncertainty
- One GP mean and variance prediction per particle

GP-EKF Propagation

- Propagate mean using GP prediction
- Use gradient of GP to propagate covariance

GP-UKF Propagation

- Propagate each sigma point using GP prediction
- 2d+1 sigma points -> 2d+1 GP mean predictions
Overview

- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- Filtering and Control
  - System Identification with GP-BayesFilters
  - Predictive State Representations
  - Conclusions

WiFi-Based Location Estimation

Similar to [Schwaighofer-etal: NIPS-03]
**Blimp Testbed**

- Task: Track a blimp with two webcams
- Baseline: Parametric model that takes drag, thrust, gravity, etc. into account
- GP-BayesFilters and parametric model trained on ground truth data obtained with Vicon motion capture system

**Tracking Results**

<table>
<thead>
<tr>
<th>Method</th>
<th>GP</th>
<th>EGP</th>
<th>hetGP</th>
<th>sparseGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKF</td>
<td>30.75 ± 1.41</td>
<td>34.10 ± 1.76</td>
<td>35.76 ± 1.61</td>
<td>32.05 ± 2.02</td>
</tr>
<tr>
<td>EKF</td>
<td>27.66 ± 1.04</td>
<td>31.44 ± 2.43</td>
<td>33.70 ± 2.09</td>
<td>29.72 ± 1.90</td>
</tr>
<tr>
<td>PF</td>
<td>33.93 ± 7.24</td>
<td>35.95 ± 6.91</td>
<td>na</td>
<td>38.92 ± 2.17</td>
</tr>
</tbody>
</table>

- Percentage reduction in RMS over parametric baseline
- Parametric model takes drag, thrust, gravity, etc. into account
- Cross validation with 900 timesteps for training
- hetGP: Heteroscedastic GP with variable noise [Kersting et al.: ICML-07]
- sparseGP: sparsified to 50 active points [Swinson-Ghahramani: NIPS-06]

**GP-UKF Tracking Example**

- Blue ellipses: sigma points projected into observation space
- Green ellipse: Mean state estimate

**Dealing with Training Data Sparsity**

- Training data for right turns removed
1. Investigation of muscle-joint kinematical relationship
2. How to control joints with muscles?

ACT Hand Tendon Arrangements

- Tendon hood structure for extensors
  - Critical for preserving hand functionality
  - Slides over the bones and joints
- We have non-linear, non-constant relationships between muscles and joints

GP-Based Control

Overview

- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- Filtering and Control
  - System Identification with GP-BayesFilters
- Predictive State Representations
- Conclusions
GP Latent Variable Models

- Sometimes ground truth states are not or only partially available
- Instead of optimizing over GP hyperparameters only, optimize over latent states $S$ as well

Latent variable models

- Learn latent states and GPs in one optimization
- Can take noisy labels into account

Slotcar Testbed

- Track contains banked curves, elevation changes
- Custom IMU with gyro and accelerometer built by Intel Research Seattle
- Observations very noisy, perceptual aliasing

Simple Trajectory Replay

- Learning
  - Human demonstrates control
  - Learn latent states using GPBF-Learn
  - Learn mapping from state to control
- Replay
  - Track state using GP-BayesFilter
  - Use control given by control GP

Ko-F: RSS-09, ARJ-10
Overview

- Gaussian Processes and Bayes Filters
- GP-BayesFilters
- Filtering and Control
- System Identification with GP-BayesFilters
  - Predictive State Representations
- Conclusions

In Hand Manipulation

- [Mordatch-Popovic-Todorov: SCA-12]

Learning Models for Manipulation

- Soon manipulators / hands / robots will be equipped with a variety of complex sensors (e.g. touch sensitive skin)
- Are accurate physics-based models the most appropriate representation for controlling such complex systems?
- Rather than imposing a model on the dynamical system, learn a state space that’s suitable for prediction and control
- Question: Can we learn expressive models from raw, high-dimensional sensor data?
Predictive State Representations (PSRs)

- Expressive dynamical system model

- Test: ordered sequence of action observation pairs
  \[ \tau = a_1 o_1 \ldots a_t o_t \]

- Prediction of a test:
  \[ P \left[ \tau^O \mid \text{do} \left( \tau^A \right), h_i \right] \]

- PSR state is a prediction over a set of core tests (future observable quantities)

Test Case

[Boots-Byravan-F: ICRA-14]

Summary

- GPs provide flexible modeling framework
- Take data noise and uncertainty due to data sparsity into account
- Seamless integration into Bayes filters
- Combination with parametric models increases accuracy and reduces amount of training data
- Subspace identification via latent variable models
- Computational complexity of GPs is a key problem
- Predictive state representations: scale to high-dimensional systems
WAM Trajectory Replay

- System: Barrett Whole Arm Manipulator
  - Four joints/degrees of freedom
  - 4D control (change in joint angles)
  - Significant control noise

- Observations:
  - 3D position of end effector

- User demonstration:
  - Manipulate to trace out circular trajectory of end effector

Control Experiment

- Learn 3D latent states for system
- Replay assuming noisy encoders
- Both time-based and simple control model fail

Simple Fix

- Want controls which decrease prediction uncertainty
- Prediction uncertainty obtained from GP
- Learn control model using only desired state-control pairs

Advanced Control Experiment

- Learn 3D latent states for system
- Replay assuming noisy encoders
- Both time-based and simple control model fail
- Advanced control model achieves proper replay