Autonomous Planning and Control with Bayesian Nonparametric Models

Marc Deisenroth
Technische Universität Darmstadt

Talk at NIPS Workshop on Bayesian Nonparametric Models for Reliable Planning and Decision-Making under Uncertainty, December 8, 2012

Joint work with
P. Englert, A. Paraschos, J. Peters, D. Fox, C.E. Rasmussen
Model Learning in Robotics/Control

- Models in robotics are important for planning and control
- Typically: rigid-body dynamics, parameter identification
- Robots violate these assumptions quite often
Model Learning in Robotics/Control

- Models in robotics are important for planning and control
- Typically: rigid-body dynamics, parameter identification
- Robots violate these assumptions quite often
- Nonparametric models can be very useful here:
  - Flexible model for data-driven extraction of relevant information
  - Make assumption at more intuitive level (e.g., smoothness instead of friction parameters)
Learning Nonparametric Transition Models

Learning a forward model: Find a transition function

\[ f : (x_{t-1}, u_{t-1}) \rightarrow x_t \]

Observed function values
Learning Nonparametric Transition Models

Learning a forward model: Find a transition function

\[ f : (x_{t-1}, u_{t-1}) \mapsto x_t \]

Plausible function approximator
Learning Nonparametric Transition Models

Learning a forward model: Find a transition function
\( f : (x_{t-1}, u_{t-1}) \mapsto x_t \)

Plausible function approximator

Predictions? Decision Making?
Learning Nonparametric Transition Models

Learning a forward model: Find a transition function
\[ f : (x_{t-1}, u_{t-1}) \mapsto x_t \]

Plausible function approximator

Predictions? Decision Making? Model Errors!
Learning Nonparametric Transition Models

Learning a forward model: Find a transition function

\[ f : (x_{t-1}, u_{t-1}) \rightarrow x_t \]

Distribution over plausible functions
Learning Nonparametric Transition Models

Learning a forward model: Find a transition function

\[ f : (x_{t-1}, u_{t-1}) \mapsto x_t \]

Distribution over plausible functions

- Express **uncertainty** about the underlying function
- **Probabilistic models**
Controller Learning

Objective (Control Learning/Policy Search)

Find policy parameters $\theta^*$ that minimize the expected long-term cost

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}[c(x_t)|\theta], \quad p(x_0) = \mathcal{N}(\mu_0, \Sigma_0).$$

- Markovian transition dynamics
  $$x_{t+1} = f(x_t, u_t) + w$$
- Controls $u_t = \pi(x_t, \theta)$
- Policy $\pi$
- Policy parameters $\theta$
- Cost function $c$, e.g., $\|x - x_{\text{target}}\|^2$
Model-based Policy Search

Objective:
Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$

Algorithm:
1. Probabilistic GP model for transition function $f$ to be robust to model errors

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Deisenroth et al. (submitted): Fast Interactive Learning in Nonparametric Models for Robotics and Control
Model-based Policy Search

Objective:
Minimize expected long-term cost $J(\theta) = \sum_t E[c(x_t)|\theta]$

Algorithm:
1. Probabilistic GP model for transition function $f$ to be robust to model errors
2. Compute long-term predictions $p(x_1), \ldots, p(x_T)$
Model-based Policy Search

Objective:
Minimize expected long-term cost $J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta]$

Algorithm:
1. Probabilistic GP model for transition function $f$ to be robust to model errors
2. Compute long-term predictions $p(x_1), \ldots, p(x_T)$
3. Policy Learning

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Deisenroth et al. (submitted): Fast Interactive Learning in Nonparametric Models for Robotics and Control
Model-based Policy Search

Objective:
Minimize expected long-term cost \( J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta] \)

Algorithm:
1. Probabilistic GP model for transition function \( f \) to be robust to model errors
2. Compute long-term predictions \( p(x_1), \ldots, p(x_T) \)
3. Policy Learning
   - Compute expected long-term cost \( J(\theta) \)

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Deisenroth et al. (submitted): Fast Interactive Learning in Nonparametric Models for Robotics and Control
Model-based Policy Search

Objective:
Minimize expected long-term cost $J(\theta) = \sum_t E[c(x_t)|\theta]$  

Algorithm:
1. Probabilistic GP model for transition function $f$ to be robust to model errors 
2. Compute long-term predictions $p(x_1), \ldots, p(x_T)$ 
3. Policy Learning 
   - Compute expected long-term cost $J(\theta)$ 
   - Find parameters $\theta$ that minimize $J(\theta)$

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search 
Deisenroth et al. (submitted): Fast Interactive Learning in Nonparametric Models for Robotics and Control
Model-based Policy Search

Objective:
Minimize expected long-term cost \( J(\theta) = \sum_t \mathbb{E}[c(x_t)|\theta] \)

Algorithm:
1. Probabilistic GP model for transition function \( f \) to be robust to model errors
2. Compute long-term predictions \( p(x_1), \ldots, p(x_T) \)
3. Policy Learning
   - Compute expected long-term cost \( J(\theta) \)
   - Find parameters \( \theta \) that minimize \( J(\theta) \)

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Deisenroth et al. (submitted): Fast Interactive Learning in Nonparametric Models for Robotics and Control
Long-Term Predictions

- Iteratively compute $p(x_1), \ldots, p(x_T)$
Long-Term Predictions

- Iteratively compute $p(x_1), \ldots, p(x_T)$

\[
p(x_t | x_{t-1}, u_{t-1}) \quad \text{GP pred. distribution} \quad p(x_{t-1}, u_{t-1}) \quad \mathcal{N}(\mu, \Sigma)
\]
Long-Term Predictions

Iteratively compute $p(x_1), \ldots, p(x_T)$

$$ p(x_t) = \int \int \int \underbrace{p(x_t|x_{t-1}, u_{t-1})}_{\text{GP pred. distribution}} \underbrace{p(x_{t-1}, u_{t-1})}_{\mathcal{N}(\mu, \Sigma)} \, df \, dx_{t-1} \, du_{t-1} $$
Long-Term Predictions

- Iteratively compute $p(x_1), \ldots, p(x_T)$

$$p(x_t) = \int \int \int p(x_t | x_{t-1}, u_{t-1}) \ p(x_{t-1}, u_{t-1}) \ dx_{t-1} du_{t-1}$$

GP pred. distribution $\mathcal{N}(\mu, \Sigma)$
Long-Term Predictions

- Iteratively compute $p(x_1), \ldots, p(x_T)$

$$p(x_t) = \int \int \int p(x_t | x_{t-1}, u_{t-1}) \cdot p(x_{t-1}, u_{t-1}) \, df \, dx_{t-1} \, du_{t-1}$$

**GP pred. distribution** $\mathcal{N}(\mu, \Sigma)$

- Approximate inference

- Moment matching (Quiñonero-Candela et al., 2003)
Standard Benchmark Problem: Cart-Pole Swing-up

- Swing up and balance a freely swinging pendulum on a cart
- Learn from scratch (random initialization)

http://www.youtube.com/user/PilcoLearner

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Standard Benchmark Problem: Cart-Pole Swing-up

- Swing up and balance a freely swinging pendulum on a cart
- Learn from scratch (random initialization)

http://www.youtube.com/user/PilcoLearner

Deisenroth & Rasmussen (ICML, 2011): *PILCO: A Model-based and Data-efficient Approach to Policy Search*
Swing up and balance a freely swinging pendulum on a cart

Learn from scratch (random initialization)

Unprecedented learning speed compared to state-of-the-art

http://www.youtube.com/user/PilcoLearner

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Are probabilistic (nonparametric) models really necessary?
Demo

Are probabilistic (nonparametric) models really necessary?

Successful learning:
Bayesian nonparametric dynamics (GP): 95%
Deterministic nonparametric dynamics: 0%
Other Applications

- Learning to ride a unicycle in less than 30 seconds
- Learning to control a noisy low-cost manipulator
- Imitation learning with learned probabilistic forward models

Deisenroth & Rasmussen (ICML, 2011): *PILCO: A Model-based and Data-efficient Approach to Policy Search*

Deisenroth, Rasmussen, Fox (RSS, 2011): *Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning*

Englert, Paraschos, Peters, Deisenroth (submitted): *Behavioral Cloning with Learned Probabilistic Forward Models*
Imitation Learning: Tendon-driven BioRob

- Find a robot controller that imitates demonstrated trajectory
- No good forward model available (yet) because of compliance, stiction, springs, etc.
  ▶ Imitation learning with learned GP forward models:
   Match demonstrated and predicted distributions over trajectories

Englert, Paraschos, Peters, Deisenroth (submitted): *Behavioral Cloning with Learned Probabilistic Forward Models*
Conclusion

- Bayesian nonparametric models are promising in robotics and control
- Bayesian averaging over \textit{model uncertainty} is crucial to be robust to model errors when making decisions
- In practice: we shouldn’t ignore engineers completely!

marc@ias.tu-darmstadt.de
References


