Machine Learning, Probabilistic Inference, System Identification and Control

Sheffield Gaussian Process Summer School, 2014

Carl Edward Rasmussen

Department of Engineering, University of Cambridge

September 15-17th, 2014

with thanks to my students: Marc Deisenroth, Roger Frigola, Joe Hall & Andrew McHutchon
Adaptive Systems Wish List

Good adaptive systems must be

• flexible
• automatic
• efficient
• able to handle noise and uncertainty
• practical
• optimal
• provably stable
Adaptive Systems Wish List – Realistic Version

Good adaptive systems must be

- flexible
- automatic
- efficient
- able to handle noise and uncertainty
- practical
- optimal
- provably stable
Probabilistic or Bayesian Inference

Probability theory is a framework ideally suited to make inferences under uncertainty

- able to handle noise and uncertainty
- automatic
- efficient,

but, can it be made

- flexible?
- practical?

Unfortunately, the Bayesian framework is often misunderstood.
Be flexible: Don’t Lie

Cromwell’s dictum:

*I beseech you, in the bowels of Christ, think it possible that you may be mistaken*

In modeling terms

- **don’t** assign zero probability to something that *might* be true
  Ex.: ’The friction is assumed purely viscous’.
- **don’t** condition on things which *might not* be true.
  Ex.: ’The Kalman filter is optimal for linear Gaussian systems’.

How can we achieve this? Non-parametric inference over functions
Learning Dynamics: Short and Long time Horizons

It is only tractable to capture short term dynamics . . .

. . . but you need to understand long term dynamics to control

⇒

• learn short term dynamics
• make probabilistic predictions (because we don’t know the exact dynamics)
• probabilistically cascade predictions to get long term behaviour
• propagating a Gaussian state distribution through a non-linear dynamics model is intractable: use moment matching
Learning Algorithm

Require:

- reward or loss function
- initial policy or control law (random)

Algorithm:

- execute current algorithm in the real world, collect data
- train the short term dynamics models on all available data
- evaluate different controllers by *simulating* their long term performance
- pick the best controller you can find (using gradient based minimization)
- repeat
Loss function geometric (not dynamic).

Short term (100ms) dynamics are captured by 4 separate GP models.

A non-linear parameterized state feedback policy is optimized based on the experience so far.

The system learns quickly and automatically, from essentially no prior knowledge:

- relevant time frames: sampling time 100 ms, horizon 2 s.
- dynamics are smooth
- dynamics are time-invariant
- scale for the loss function
Properties of the Solution

Some observations about the learner

- Dynamics are learnt only locally around successful trajectories
- learning algorithm is greedy
  - exploration vs exploitation
- characteristic behaviour of the uncertainty or error-bars
  - Initially, error bars are wide and expanding
  - after successful learning, error bars are collapsing
Learning to Unicycle
Equations of Motion

\[ A = \begin{bmatrix} C_t \end{bmatrix} \]

Rasmussen (Engineering, Cambridge) Learning, Inference and Control September 15-17th, 2014 11 / 12
Conclusions

It is possible to learn automatically and rapidly with essentially no prior knowledge.

Don’t lie about the complexity of the system: use non-parametric models

Don’t lie about uncertainties: faithfully keep track of error-bars.

The learning approach is robust and practical for real applications:

• calibration of sensors and actuators unnecessary
• inaccurate assumptions unnecessary