Bayesian Machine Learning for Controlling Autonomous Systems

Marc Deisenroth
Department of Computing
Imperial College London

Department of Computer Science
TU Darmstadt

m.deisenroth@imperial.ac.uk

Talk at University of Oxford
September 30, 2013
Motivation

- Three key challenges in autonomous systems:
  - Modeling.
  - Predicting.
  - Decision making.
Motivation

- Three key challenges in autonomous systems: **Modeling. Predicting. Decision making.**

- Noisy signals and processes
Motivation

- Three key challenges in autonomous systems:
  - **Modeling. Predicting. Decision making.**
- Noisy signals and processes

Increase autonomy: deal with uncertainty

▶ **Bayesian machine learning**
Motivation

- Three key challenges in autonomous systems: **Modeling. Predicting. Decision making.**

- Noisy signals and processes

Increase autonomy: deal with uncertainty

▶ **Bayesian machine learning**
Motivation

- Three key challenges in autonomous systems:
  - Modeling. Predicting. Decision making.
- Noisy signals and processes

Increase autonomy: deal with uncertainty
  ▶ Bayesian machine learning
Motivation

- Three key challenges in autonomous systems:
  
  Modeling. Predicting. Decision making.

- Noisy signals and processes

Increase autonomy: deal with uncertainty  

▶ Bayesian machine learning
Outline

Controller Learning

Reinforcement Learning

Bayesian Optimization
Reinforcement Learning Set-up

\[
x_{t+1} = f(x_t, u_t) + w, \quad u_t = \pi(x_t, \theta)
\]

State \quad Control \quad Policy \quad Policy parameters
Reinforcement Learning Set-up

\[ x_{t+1} = f(x_t, u_t) + w, \quad u_t = \pi(x_t, \theta) \]

State \quad Control \quad Policy \quad Policy parameters

Objective

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

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

Instantaneous cost \( c(x_t) \), e.g., \( \|x_t - x_{\text{target}}\|^2 \)

Typical objective in optimal control and reinforcement learning
(Bertsekas, 2005; Sutton & Barto, 1998)
Model-based Policy Search

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

High-Level Steps:
1. Probabilistic model for transition function $f$ to be robust to model errors
Model-based Policy Search

Objective

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

High-Level Steps:

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

2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$

---

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

Objective

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

High-Level Steps:

1. Probabilistic model for transition function \( f \) to be robust to model errors
2. Compute long-term predictions \( p(x_1|\theta), \ldots, p(x_T|\theta) \)
3. Policy improvement

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

**Objective**

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

**High-Level Steps:**

1. Probabilistic model for transition function $f$ to be robust to model errors
2. Compute long-term predictions $p(x_1 | \theta), \ldots, p(x_T | \theta)$
3. Policy improvement
4. Apply controller

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

Objective

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

High-Level Steps:

1. Probabilistic model for transition function $f$ to be robust to model errors
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$
3. Policy improvement
4. Apply controller

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

Model learning problem: Find a function $f : x \mapsto f(x) = y$
Model Learning

Model learning problem: Find a function \( f : x \mapsto f(x) = y \)

Plausible function approximators
Model Learning

Model learning problem: Find a function $f : x \mapsto f(x) = y$

Plausible function approximators

Predictions? Decision Making?
Model Learning

Model learning problem: Find a function $f : x \mapsto f(x) = y$

Plausible function approximators

Predictions? Decision Making? Model Errors!
Model Learning

Model learning problem: Find a function $f : x \mapsto f(x) = y$

Distribution over plausible functions
Model Learning

Model learning problem: Find a function $f : x \mapsto f(x) = y$

Distribution over plausible functions

- Express **uncertainty** about the underlying function
- **Gaussian process** for model learning (Rasmussen & Williams, 2006)
Introduction to Gaussian Processes

- State-of-the-art nonparametric Bayesian regression method
- Probability distribution over functions
- Fully specified by
  - Mean function $m$ (average function)
  - Covariance function $k$ (assumptions on structure)

$$\text{Cov}[f(x_p), f(x_q)] = k(x_p, x_q)$$

Posterior predictive distribution at $x_\hat{}$ is Gaussian:

$$p(f(x_\hat{}) | x_\hat{}, X, y) \sim \mathcal{N}(m(x_\hat{}), \sigma^2)$$
Introduction to Gaussian Processes

- State-of-the-art nonparametric Bayesian regression method
- Probability distribution over functions
- Fully specified by
  - Mean function $m$ (average function)
  - Covariance function $k$ (assumptions on structure)

\[
\text{Cov}[f(x_p), f(x_q)] = k(x_p, x_q)
\]

- Posterior predictive distribution at $x_*$ is Gaussian:

\[
p(f(x_*)| x_*, X, y) = \mathcal{N}(f(x_*) | m(x_*), \sigma^2(x_*))
\]

Test input  Training data
Intuitive Introduction to Gaussian Processes

Prior belief about the function

Predictive (marginal) mean and variance:

\[ E[f(x_*)|\emptyset] = m(x_*) = 0 \]
\[ V[f(x_*)|\emptyset] = \sigma^2(x_*) = \text{Cov}[f(x_*), f(x_*)] = k(x_*, x_*) \]
Prior belief about the function

Predictive (marginal) mean and variance:

\[
E[f(x_*)|\emptyset] = m(x_*) = 0
\]

\[
V[f(x_*)|\emptyset] = \sigma^2(x_*) = \text{Cov}[f(x_*), f(x_*)] = k(x_*, x_*)
\]
Intuitive Introduction to Gaussian Processes

\[
\begin{align*}
\mathbb{E}[f(x_*)|X, y] &= m(x_*) = k(X, x_*)^\top k(X, X)^{-1} y \\
\mathbb{V}[f(x_*)|X, y] &= \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1} k(X, x_*)
\end{align*}
\]

Posterior belief about the function
Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(x_*)|X, y] &= m(x_*) = k(X, x_*)^\top k(X, X)^{-1} y \\
\mathbb{V}[f(x_*)|X, y] &= \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1} k(X, x_*)
\end{align*}
\]
Intuitive Introduction to Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*)|X, y] = m(x_*) = k(X, x_*)^\top k(X, X)^{-1}y
\]

\[
\mathbb{V}[f(x_*)|X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1}k(X, x_*)
\]
Intuitive Introduction to Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*)|X, y] = m(x_*) = k(X, x_*)^\top k(X, X)^{-1}y \\
\text{Var}[f(x_*)|X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1}k(X, x_*)
\]
Intuitive Introduction to Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(x_*)|X,y] &= m(x_*) = k(X,x_*)^\top k(X,X)^{-1}y \\
\mathbb{V}[f(x_*)|X,y] &= \sigma^2(x_*) = k(x_*,x_*) - k(X,x_*)^\top k(X,X)^{-1}k(X,x_*)
\end{align*}
\]
Intuitive Introduction to Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*) | X, y] = m(x_*) = k(X, x_*)^\top k(X, X)^{-1} y \\
\mathbb{V}[f(x_*) | X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1} k(X, x_*)
\]
Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*)|X, y] = m(x_*) = k(X, x_*)^\top k(X, X)^{-1} y
\]

\[
\mathbb{V}[f(x_*)|X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1} k(X, x_*)
\]
Intuitive Introduction to Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[ E[f(x_*)|X, y] = m(x_*) = k(X, x_*)^\top k(X, X)^{-1} y \]

\[ \text{Var}[f(x_*)|X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1} k(X, x_*) \]
Intuitive Introduction to Gaussian Processes

**Posterior** belief about the function

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*)|X, y] = m(x_*) = k(X, x_*)^\top k(X, X)^{-1}y
\]

\[
\mathbb{V}[f(x_*)|X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top k(X, X)^{-1}k(X, x_*)
\]
Model-based Policy Search

Objective

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

High-Level Steps:

1. Probabilistic model for transition function $f$ to be robust to model errors
2. Compute long-term predictions $p(x_1 | \theta), \ldots, p(x_T | \theta)$
3. Policy improvement
4. Apply controller

Deisenroth & Rasmussen (ICML, 2011): PILCO: A Model-based and Data-efficient Approach to Policy Search
Iteratively compute $p(x_1|\theta), \ldots, p(x_T|\theta)$
Long-Term Predictions

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

$p(x_{t+1}|x_t, u_t)$

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

GP prediction

Approximate inference

Moment matching (Quiñonero-Candela et al., 2003)
Long-Term Predictions

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

\[
p(x_{t+1}|\theta) = \int \int \int p(x_{t+1}|x_t, u_t) \, p(x_t, u_t|\theta) \, df \, dx_t \, du_t
\]

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

\[ p(x_{t+1} \mid \theta) = \int \int \int p(x_{t+1} \mid x_t, u_t) \mathcal{N}(\mu, \Sigma) \, df \, dx_t \, du_t \]

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

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

$$p(x_{t+1}|\theta) = \int \int \int p(x_{t+1}|x_t,u_t) p(x_t,u_t|\theta) \, df \, dx_t \, du_t$$

- Approximate inference

- Moment matching (Quiñonero-Candela et al., 2003)
Model-based Policy Search

Objective

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

High-Level Steps:

1. Probabilistic model for transition function $f$ to be robust to model errors
2. Compute long-term predictions $p(x_1|\theta), \ldots, p(x_T|\theta)$
3. **Policy improvement**
   - Compute expected long-term cost $J(\theta)$
   - Find parameters $\theta$ that minimize $J(\theta)$
4. Apply controller

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

**Objective**

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

- Know how to predict $p(x_1|\theta), \ldots, p(x_T|\theta)$

---

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

Objective

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

- Know how to predict \( p(x_1|\theta), \ldots, p(x_T|\theta) \)
- Compute

\[
\mathbb{E}[c(x_t)|\theta] = \int c(x_t)N(x_t | \mu_t, \Sigma_t)dx_t, \quad t = 1, \ldots, T,
\]

and sum them up to obtain \( J(\theta) \)

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

Objective

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

- Know how to predict \( p(x_1|\theta), \ldots, p(x_T|\theta) \)
- Compute

\[
\mathbb{E}[c(x_t)|\theta] = \int c(x_t) \mathcal{N}(x_t | \mu_t, \Sigma_t) \, dx_t, \quad t = 1, \ldots, T,
\]

and sum them up to obtain \( J(\theta) \)
- Analytically compute gradient \( dJ(\theta)/d\theta \)
- Standard gradient-based optimizer (e.g., BFGS) to find \( \theta^* \)

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

Objective

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

- Know how to predict $p(x_1|\theta), \ldots, p(x_T|\theta)$
- Compute

$$\mathbb{E}[c(x_t)|\theta] = \int c(x_t) \mathcal{N}(x_t | \mu_t, \Sigma_t) \, dx_t, \quad t = 1, \ldots, T,$$

and sum them up to obtain $J(\theta)$

- Analytically compute gradient $dJ(\theta)/d\theta$
- Standard gradient-based optimizer (e.g., BFGS) to find $\theta^*$

PILCO framework for controller learning

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

Cost function \( c(x) = -\exp(-\|x - x_{\text{target}}\|^2) \)


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
- Cost function $c(x) = -\exp\left(-\|x - x_{\text{target}}\|^2\right)$


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
- Cost function $c(x) = -\exp(-\|x - x_{\text{target}}\|^2)$
- **Unprecedented learning speed** compared to state-of-the-art

Deisenroth & Rasmussen (ICML, 2011): **PILCO: A Model-based and Data-efficient Approach to Policy Search**
Learning to Control an Off-the-Shelf Robot

- Autonomously learn block-stacking with a low-cost robot
- Robot very noisy
- Learn forward model and controller from scratch

Deisenroth et al. (RSS, 2011): *Learning to Control a Low-Cost Manipulator using Data-efficient Reinforcement Learning*
Controlling Throttle Valves in Combustion Engines

Bischoff et al., ECML 2013

More videos at http://www.youtube.com/user/PilcoLearner
Summary (1)

Practical Framework for Autonomous Learning

- Key: Explicit incorporation of model uncertainty into long-term predictions and decision making
- Applied to real systems
Outline

Controller Learning

- Reinforcement Learning
- Bayesian Optimization
Bayesian Optimization for Learning Controllers

- Learning forward models is not always easy
- Legged locomotion: ground contacts

Objective

Find parameters $\theta$ of controller $\pi(\theta)$
Bayesian Optimization for Learning Controllers

- Learning forward models is not always easy
- Legged locomotion: ground contacts

**Objective**

Find parameters $\theta$ of controller $\pi(\theta)$

**Challenges:**

- No forward model
- No analytic cost function, no demonstrations
- Still need to be data efficient (fragile robot)
- Manual parameter search is tedious

**Bayesian optimization** (e.g., Jones 1998; Osborne et al., 2009)
Bayesian Optimization

Objective
Minimize an objective function $g$, which is very expensive to evaluate
Bayesian Optimization

Objective

Minimize an objective function \( g \), which is very expensive to evaluate

Key Idea:

1. Build a model \( \tilde{g} \) of the objective function
2. Find \( \theta^* = \arg\min_\theta \tilde{g}(\theta) \)
3. Evaluate true objective \( g \) at \( \theta^* \)
4. Update the model \( \tilde{g} \)
Bayesian Optimization

Objective
Minimize an objective function $g$, which is very expensive to evaluate

Key Idea:
1. Build a model $\tilde{g}$ of the objective function
2. Find $\theta^* \in \text{arg min}_\theta \tilde{g}(\theta)$
3. Evaluate true objective $g$ at $\theta^*$
4. Update the model $\tilde{g}$
   - Standard model $\tilde{g}$ is a Gaussian process
   - Standard assumption:
     Computations are cheap compared to evaluating true objective $g$
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB)** criterion to select next point

\[ \theta^* = \arg \min_\theta \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right) \]
- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[
\theta^* = \arg\min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound** (UCB) criterion to select next point

\[ \theta^* = \arg \min_{\theta} \left( E[\tilde{g}(\theta)] - 2\sqrt{V[\tilde{g}(\theta)]} \right) \]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion** to select next point

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right)
\]
- **Upper-Confidence-Bound (UCB)** criterion to select next point

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{V}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- Upper-Confidence-Bound (UCB) criterion to select next point

\[ \theta^* = \arg \min_\theta \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{Var}[\tilde{g}(\theta)]} \right) \]
• **Upper-Confidence-Bound** (UCB) criterion to select next point

\[ \theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right) \]
Bayesian Machine Learning for Controlling Autonomous Systems

Marc Deisenroth @University of Oxford, September 30, 2013

21

Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[
\theta^* = \arg\min_\theta \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[ \theta^* = \arg \min_\theta \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{V}[\tilde{g}(\theta)]} \right) \]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB)** criterion to select next point

\[ \theta^* = \arg\min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{Var}[\tilde{g}(\theta)]} \right) \]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion to select next point**

  \[
  \theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{Var}[\tilde{g}(\theta)]} \right)
  \]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{V}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion** to select next point

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right)
\]
- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[
\theta^* = \arg \min_{\theta} \left( E[\tilde{g}(\theta)] - 2\sqrt{V[\tilde{g}(\theta)]} \right)
\]
• **Upper-Confidence-Bound** (UCB) criterion to select next point

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{Var}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

Upper-Confidence-Bound (UCB) criterion to select next point

\[ \theta^* = \arg\min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right) \]
Bayesian Optimization: Illustration

- Upper-Confidence-Bound (UCB) criterion to select next point

\[ \theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right) \]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion** to select next point

\[
\theta^* = \arg \min_\theta \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\mathbb{V}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- Upper-Confidence-Bound (UCB) criterion to select next point

\[ \theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{V}[\tilde{g}(\theta)]} \right) \]
Bayesian Optimization: Illustration

- **Upper-Confidence-Bound (UCB) criterion to select next point**

\[
\theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{Var}[\tilde{g}(\theta)]} \right)
\]
Bayesian Optimization: Illustration

- Upper-Confidence-Bound (UCB) criterion to select next point

\[ \theta^* = \arg \min_{\theta} \left( \mathbb{E}[\tilde{g}(\theta)] - 2\sqrt{\text{Var}[\tilde{g}(\theta)]} \right) \]

- Global minimum found after 10 function evaluations
Bayesian Gait Optimization for Legged Locomotion

- Fragile biped
  - Only few experiments feasible
- Maximize robustness and walking speed
- 4 motors:
  - 2 actuated hips + 2 actuated knees
- Controller implemented as a finite-state-machine (8 parameters)
- Good parameters found after 100 experiments
Summary (2)

Bayesian Gait Optimization

- Bayesian optimization for learning controllers in a few experiments
- General framework
  (no assumptions on dynamics, no explicit cost required)
- Limited to few parameters ($\approx 10–20$)
Wrap-up

- Controller learning for autonomous systems (from scratch)
  - Reinforcement learning
  - Bayesian optimization
- Key to success: Probabilistic modeling and Bayesian inference

m.deisenroth@imperial.ac.uk

Thank you for your attention
References


