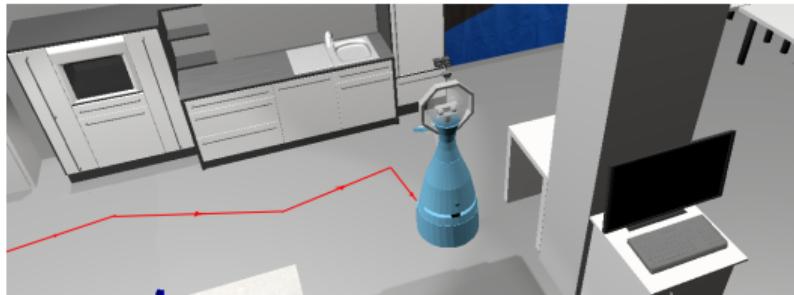


Learning Probabilistic Models for Planning under Uncertainty by Bayesian Optimization of Unsupervised Machine Learning Algorithms for State-Spaces



Rob van Bekkum

Delft University of Technology

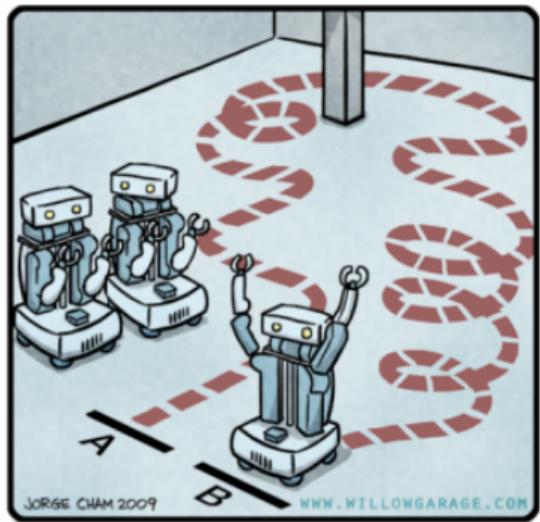
April 5, 2017

Decision-Theoretic Planning

DTP Problem Scope

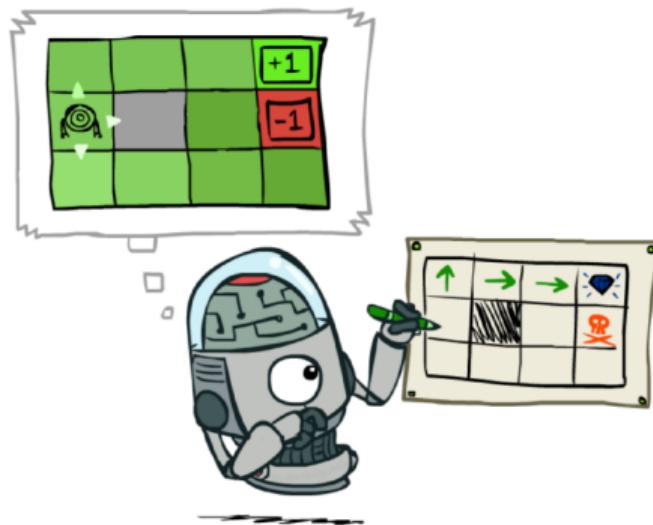
Systems:

- Dynamics are stochastic, i.e., involve uncertainty
 - ▶ Execution of actions (e.g., robot may slip)
 - ▶ Exogenous events (e.g., doors open/closed)
 - ▶ Observations (e.g., sensor noise)
- Controlled by one or more agents
- Sequential decisions of actions to execute

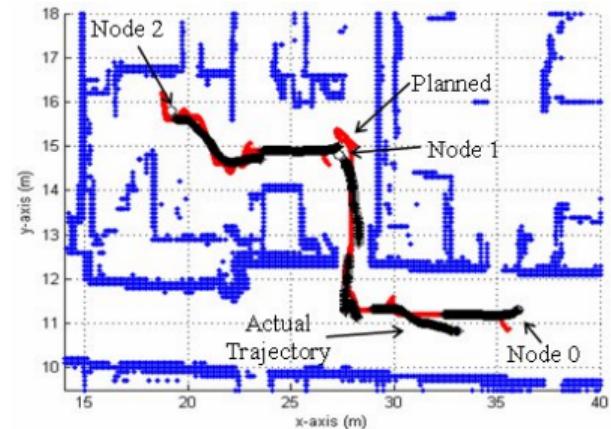


Decision-Theoretic Planning

Example Domain: Path Planning



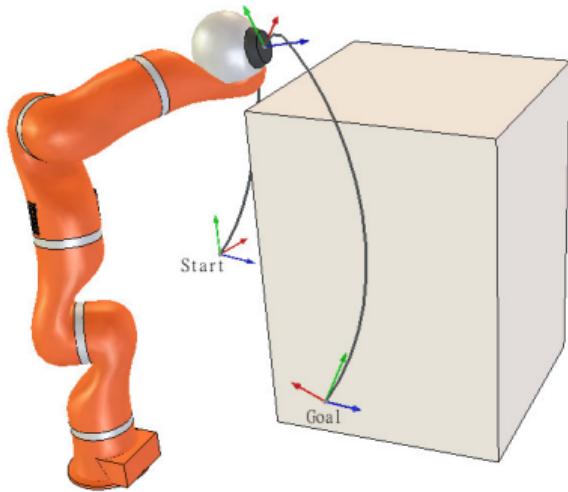
Source: Berkeley CS188 sheets



Source: http://www.cs.cmu.edu/~gunhee/r_psr.html

Decision-Theoretic Planning

Example Domain: Motion Planning



Source: <http://www.coppeliarobotics.com/helpFiles/en/motionPlanningModule.htm>

Decision-Theoretic Planning

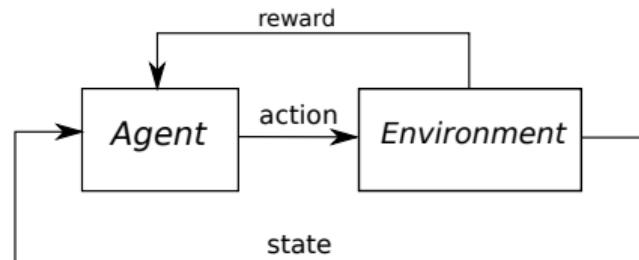
Markov Decision Processes (MDPs)

In DTP systems are modeled by probabilistic models, e.g. MDPs:

Definition

An MDP is a 5-tuple $\mathcal{M} = (\mathcal{S}, s_0, A, \delta, R)$:

- \mathcal{S} is the state-space, $s_0 \in \mathcal{S}$ the initial state
- A is the action-space
- δ is the transition function (N.B.: accounts for uncertainty)
- R is the reward function (N.B.: defines the goals)



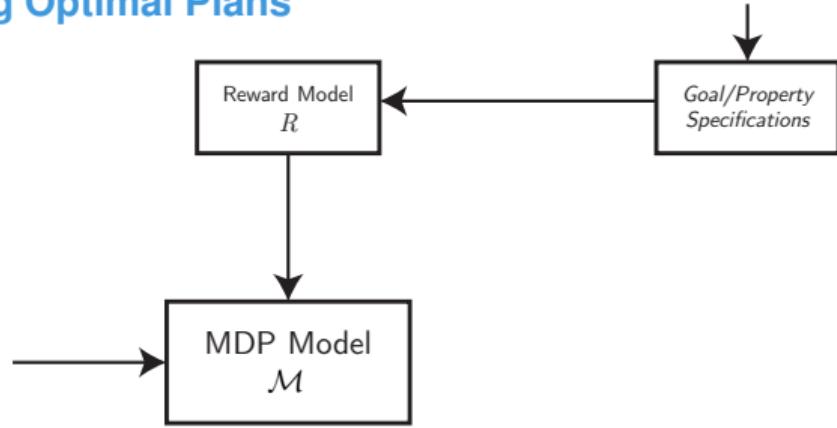
Decision-Theoretic Planning

Learning Optimal Plans



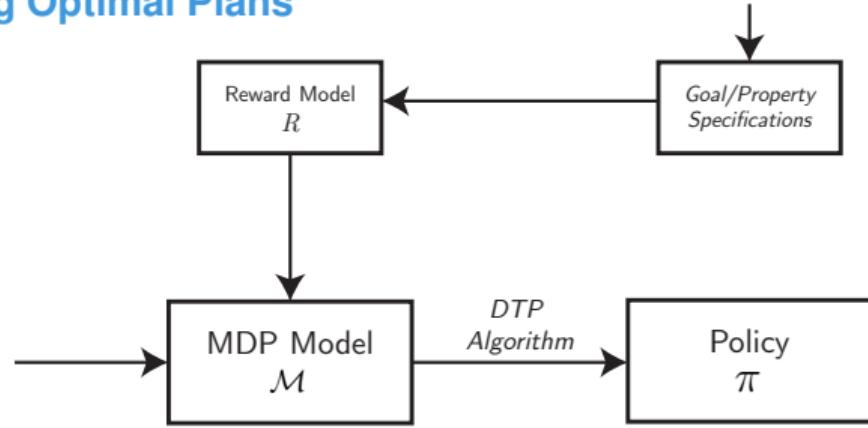
Decision-Theoretic Planning

Learning Optimal Plans



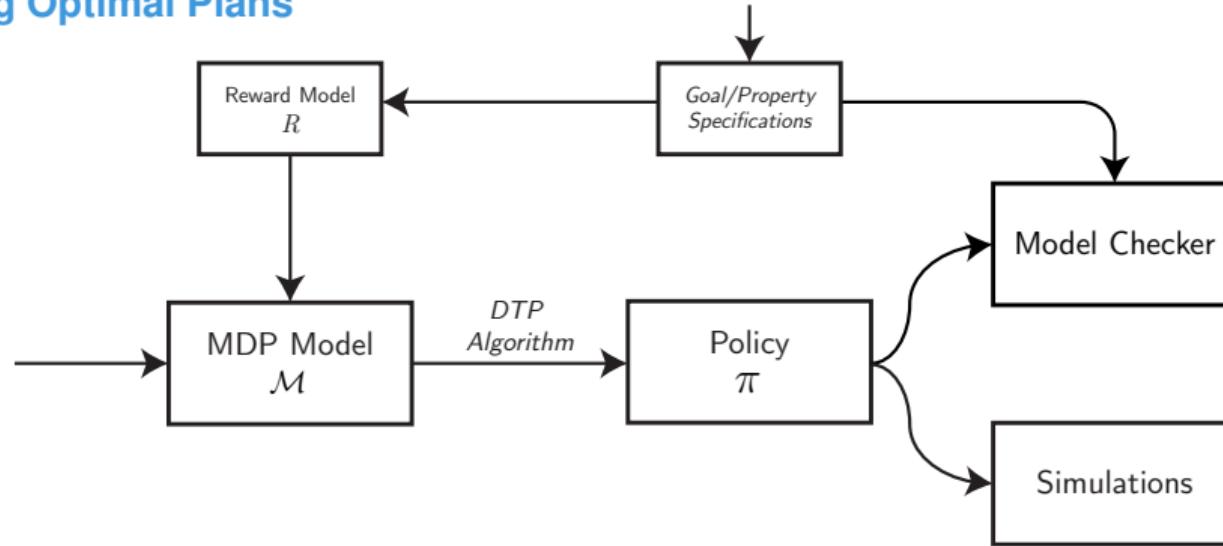
Decision-Theoretic Planning

Learning Optimal Plans



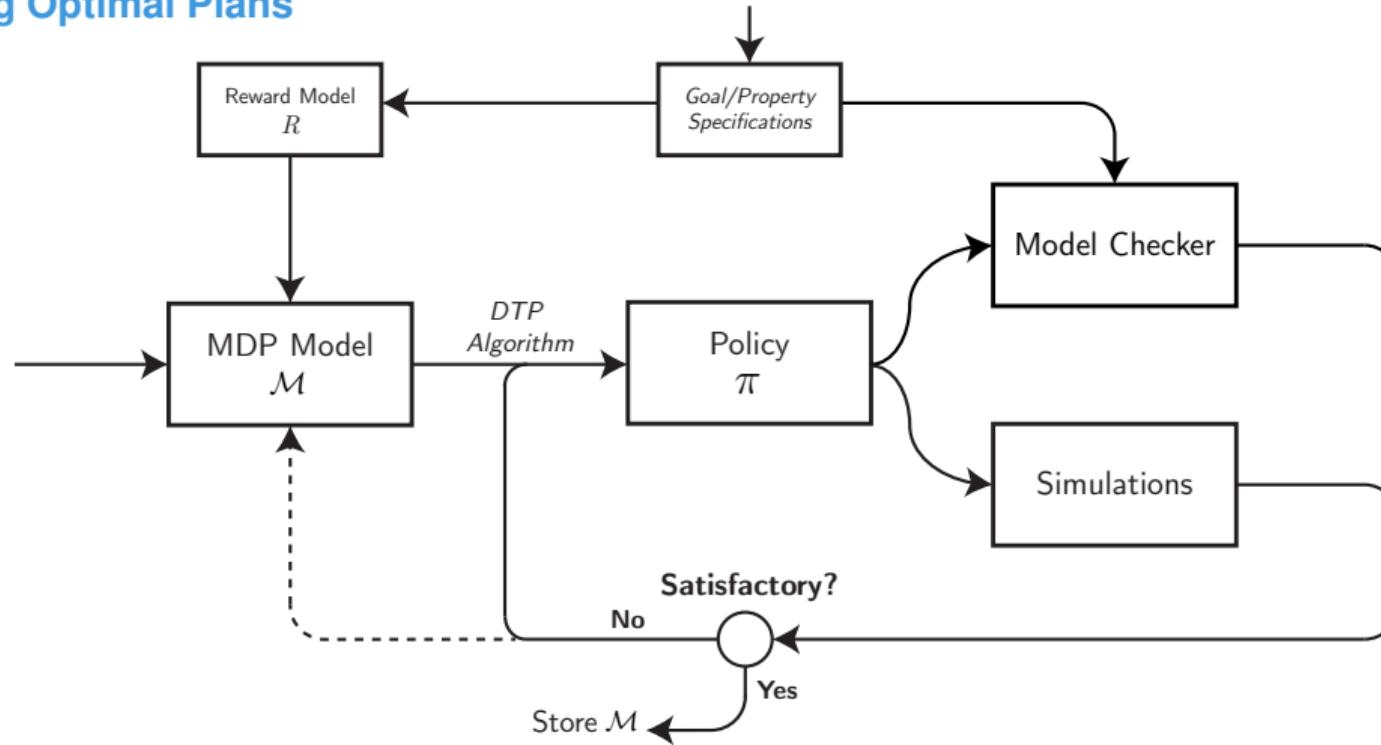
Decision-Theoretic Planning

Learning Optimal Plans



Decision-Theoretic Planning

Learning Optimal Plans



Decision-Theoretic Planning

Model Development

Problem: How to find a suitable MDP model?

Decision-Theoretic Planning

Model Development

Problem: How to find a suitable MDP model?

Classical approach: Model development by a *human designer*, however:

- Requires significant effort (e.g., trial-and-error)
- Typically demands knowledge/experience, accompanied by high costs

Decision-Theoretic Planning

Model Development

Problem: How to find a suitable MDP model?

Classical approach: Model development by a *human designer*, however:

- Requires significant effort (e.g., trial-and-error)
- Typically demands knowledge/experience, accompanied by high costs

Alternative: Use Reinforcement Learning instead of Planning, however:

- Requires direct interaction with environment
- One might require *reusable* models, applicable for multiple tasks

Decision-Theoretic Planning

Model Development

Problem: How to find a suitable MDP model?

Idea: Automate the model building process by learning algorithms

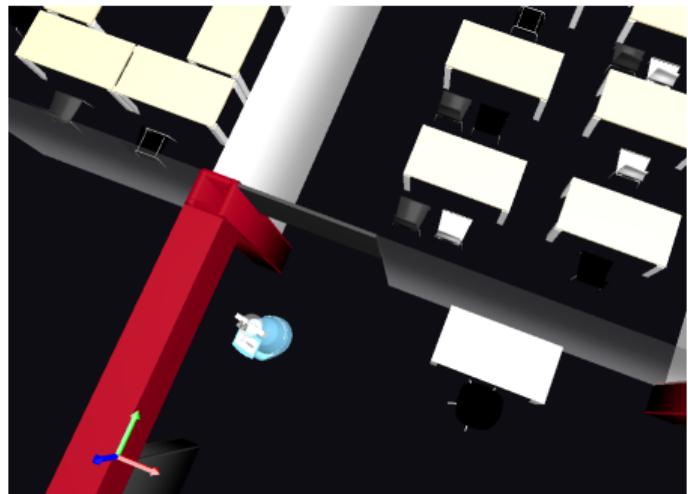
- Learn from (exploration) data about the environment
- Optimization for the best MDP model

Mobile Robot Navigation

Goal: Move robot to area while minimizing traveled distance

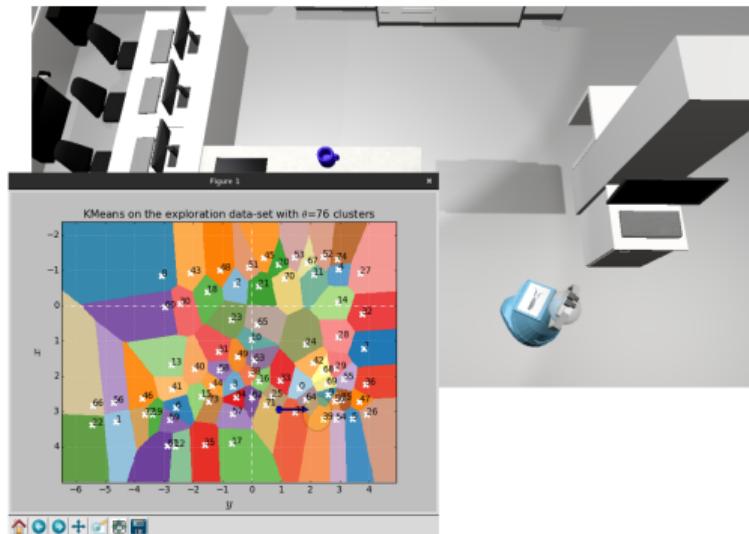
MDP Model \mathcal{M} should define:

- States \mapsto locations
- Actions \mapsto robot translations
- Rewards \mapsto goal areas
- Transition-function



Model-Learning for Mobile Robot Navigation

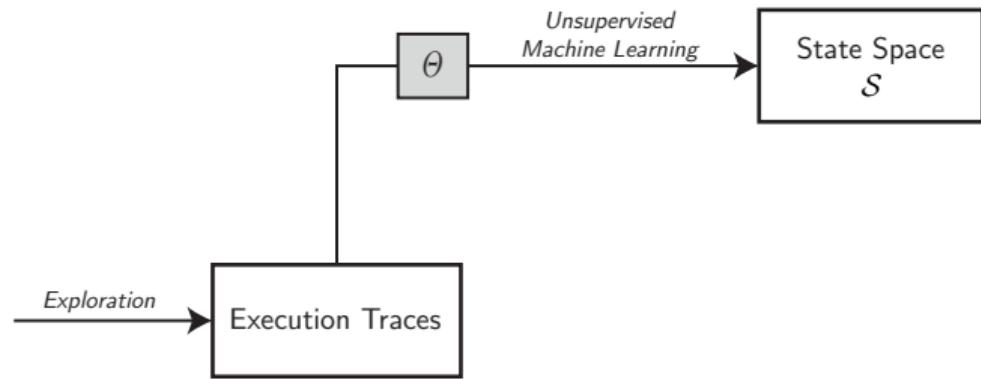
- Existing approaches for state-space acquisition:
 - ▶ Grid Discretization
 - ▶ Best-First Model Merging
 - ▶ State-Merging by Trajectory Clustering
- **TODO**
- Simulation in Morse with a Scitos A5 robot



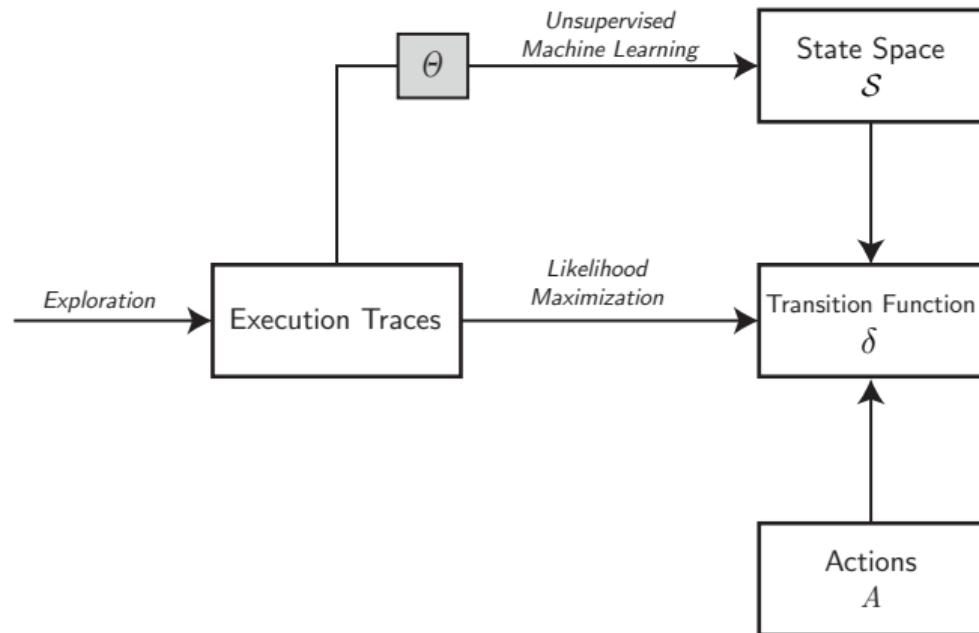
Model-Optimization Routine



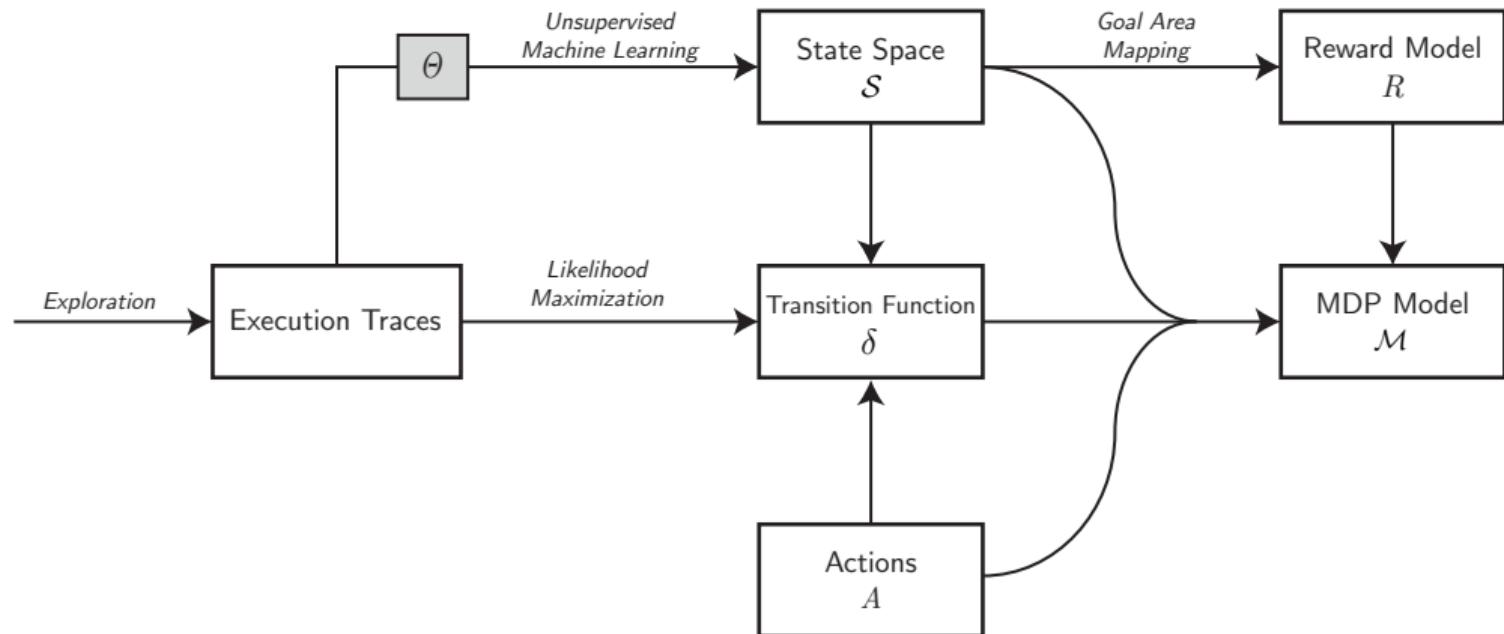
Model-Optimization Routine



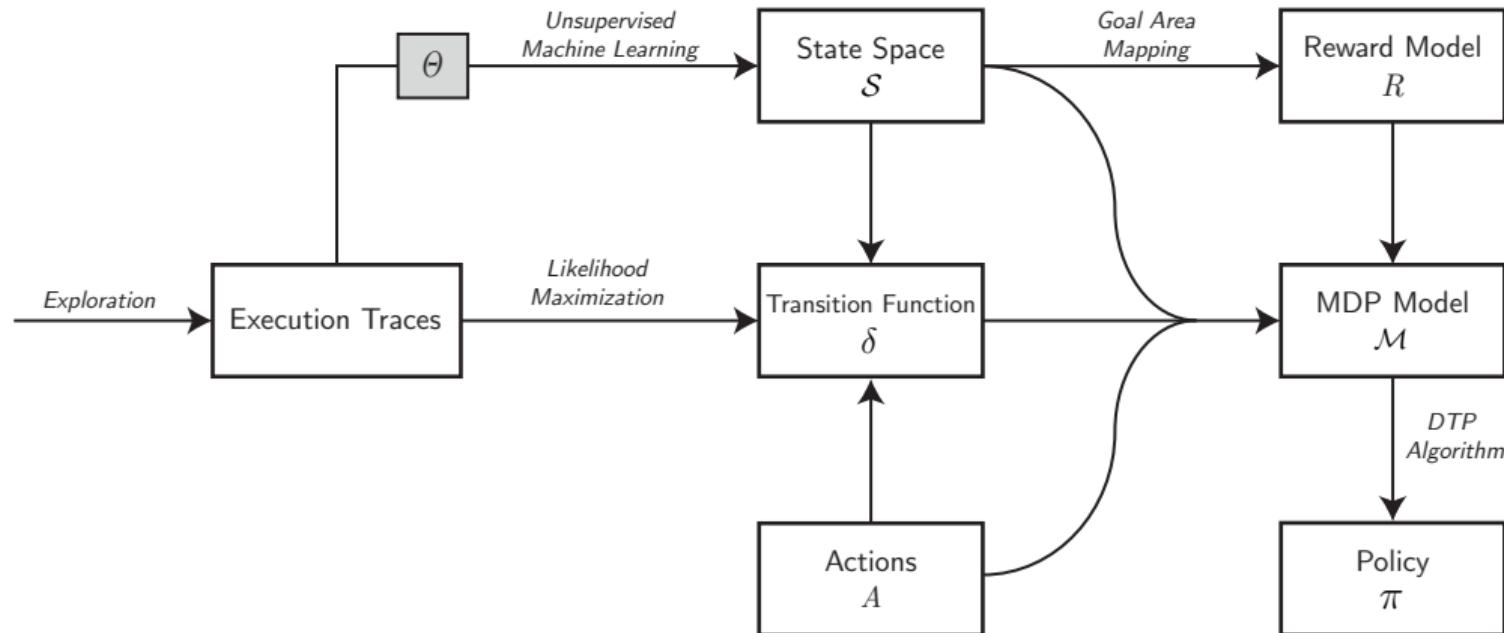
Model-Optimization Routine



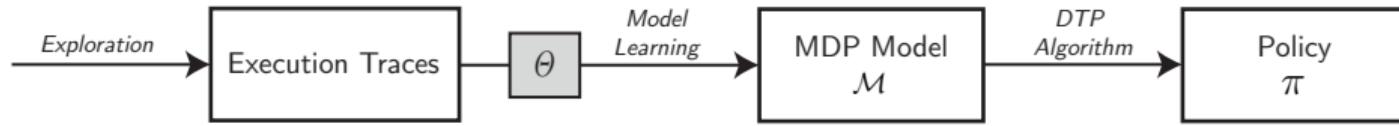
Model-Optimization Routine



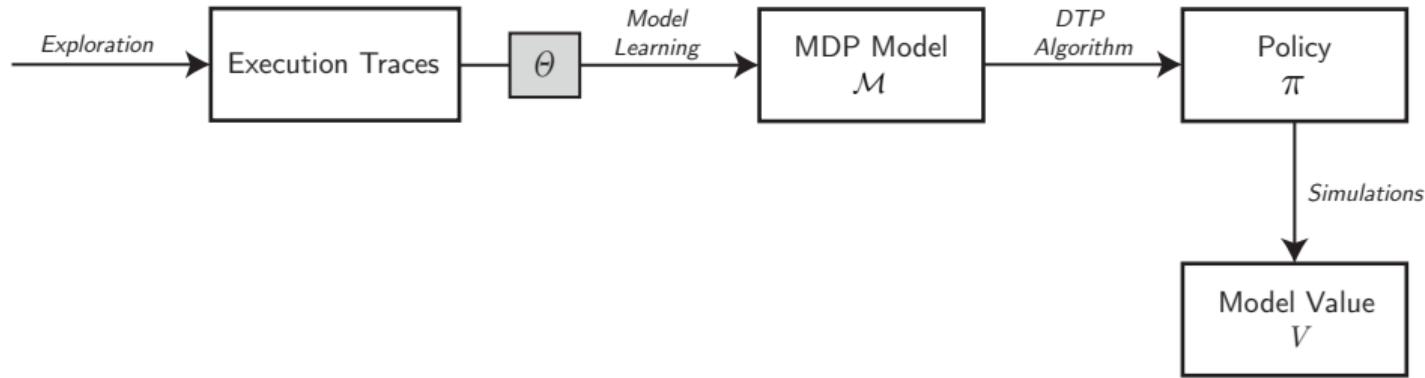
Model-Optimization Routine



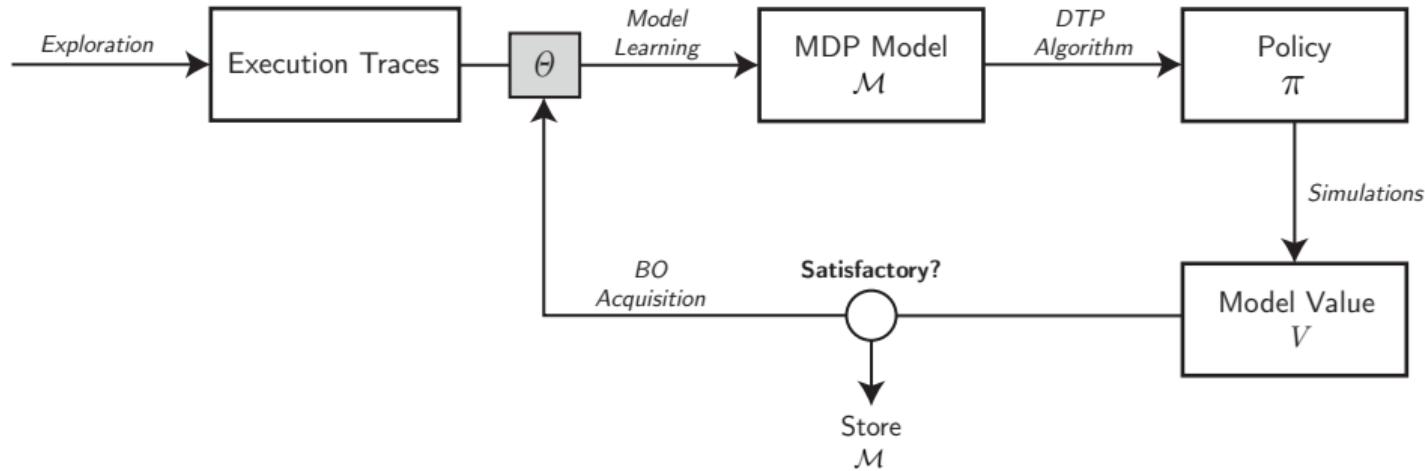
Model-Optimization Routine



Model-Optimization Routine



Model-Optimization Routine



Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)

Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x)$$

Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x)$$

- Bayesian Optimization components:

Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x)$$

- Bayesian Optimization components:
 - ▶ Prior over f : Gaussian Process $f \sim GP(\mu(\cdot), K(\cdot, \cdot))$

Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x)$$

- Bayesian Optimization components:
 - ▶ *Prior over f :* Gaussian Process $f \sim GP(\mu(\cdot), K(\cdot, \cdot))$
 - ▶ *Evidence set:* $\mathcal{D}_{1:t} = \{(x_i, y_i) \mid i = 1 \dots t\}$

Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

$$x^* = \arg \max_{x \in \mathcal{X}} f(x)$$

- Bayesian Optimization components:
 - ▶ *Prior over f :* Gaussian Process $f \sim GP(\mu(\cdot), K(\cdot, \cdot))$
 - ▶ *Evidence set:* $\mathcal{D}_{1:t} = \{(x_i, y_i) \mid i = 1 \dots t\}$
 - ▶ *Posterior or Surrogate function* ('estimate of f ')

Bayesian Optimization

- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

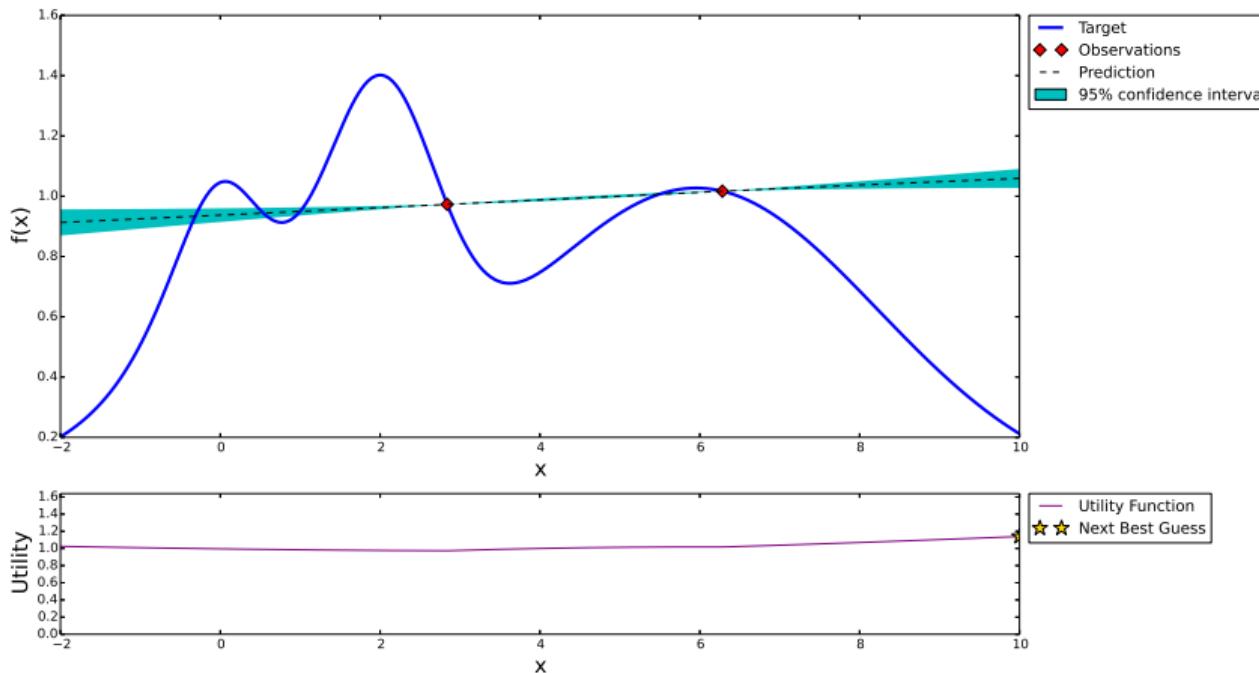
$$x^* = \arg \max_{x \in \mathcal{X}} f(x)$$

- Bayesian Optimization components:
 - ▶ *Prior over f :* Gaussian Process $f \sim GP(\mu(\cdot), K(\cdot, \cdot))$
 - ▶ *Evidence set:* $\mathcal{D}_{1:t} = \{(x_i, y_i) \mid i = 1 \dots t\}$
 - ▶ *Posterior or Surrogate function* ('estimate of f ')
 - ▶ *Acquisition or Utility function:* $u : \mathcal{X} \mapsto \mathbb{R}$

Bayesian Optimization

Example

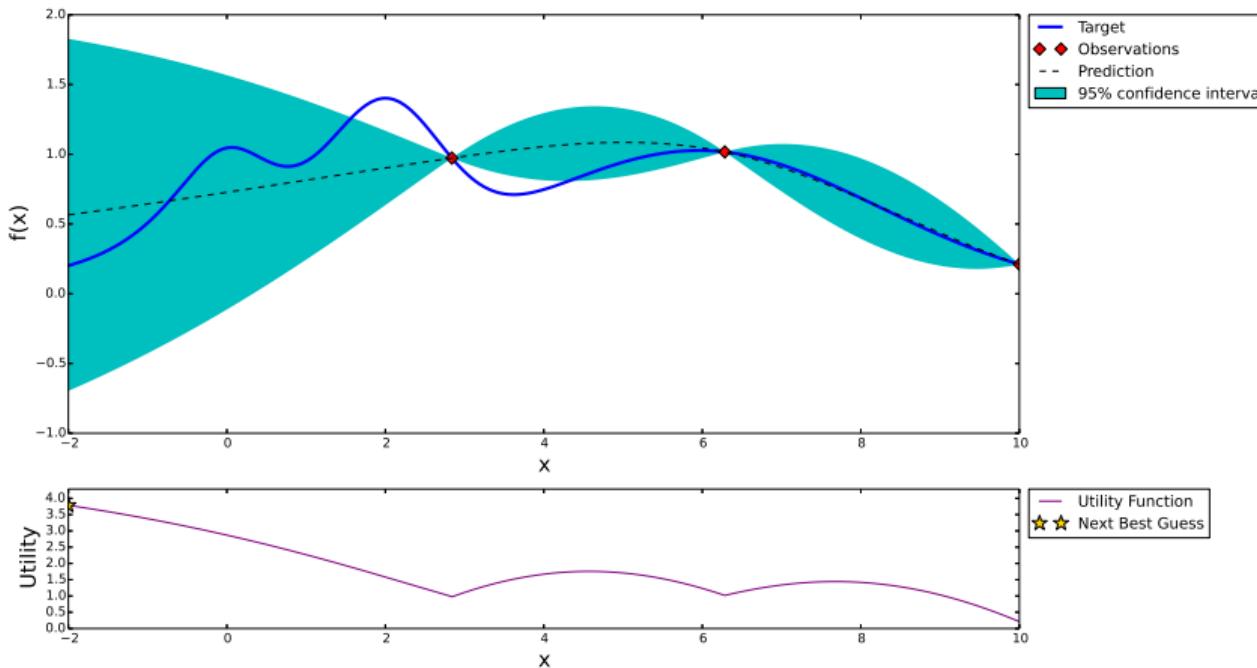
Gaussian Process and Utility Function After 2 Steps



Bayesian Optimization

Example

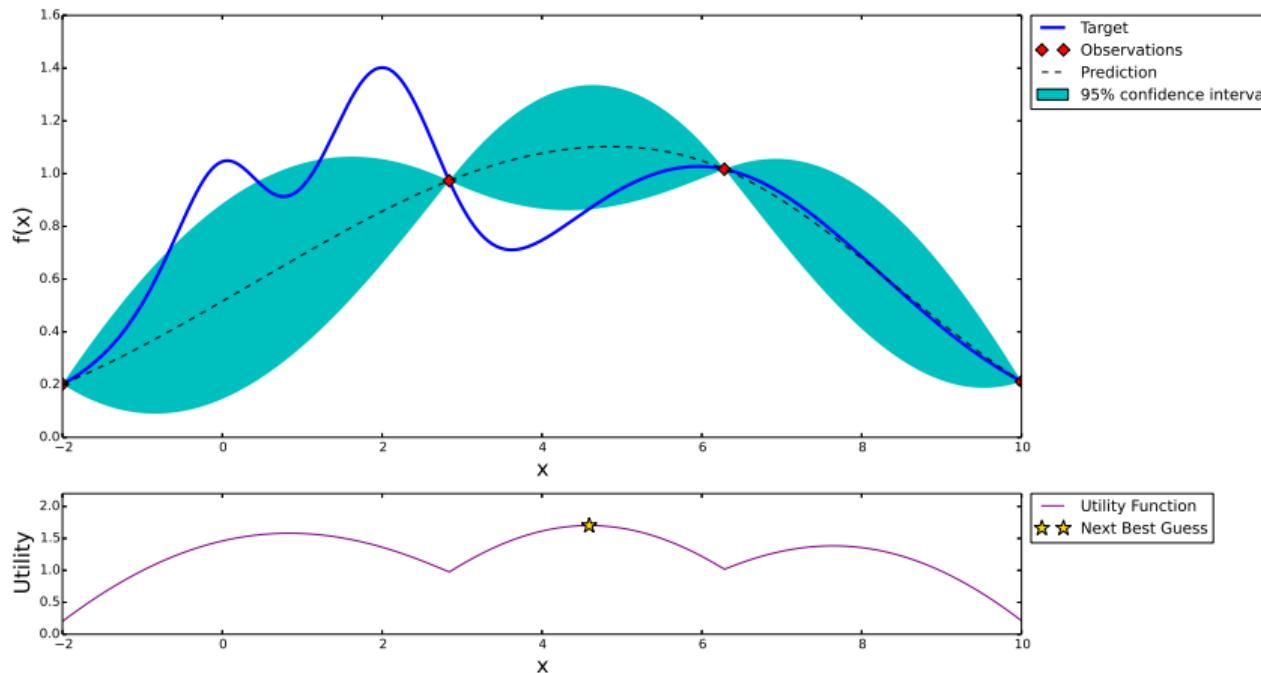
Gaussian Process and Utility Function After 3 Steps



Bayesian Optimization

Example

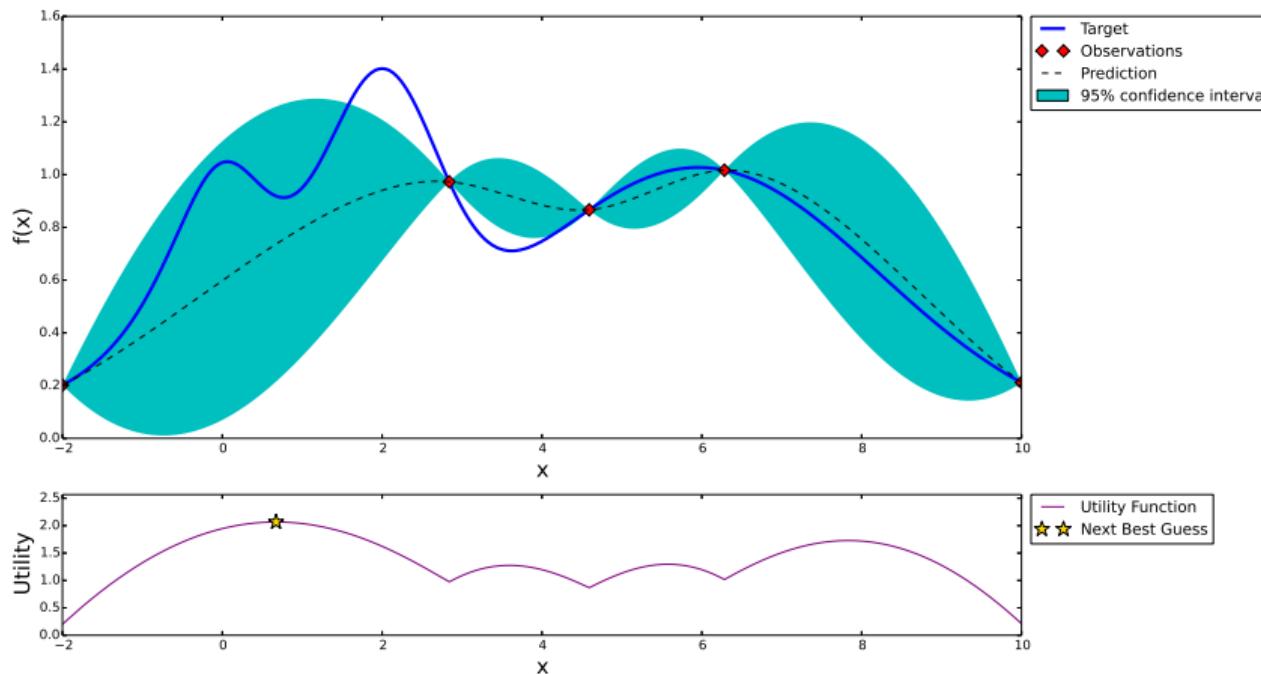
Gaussian Process and Utility Function After 4 Steps



Bayesian Optimization

Example

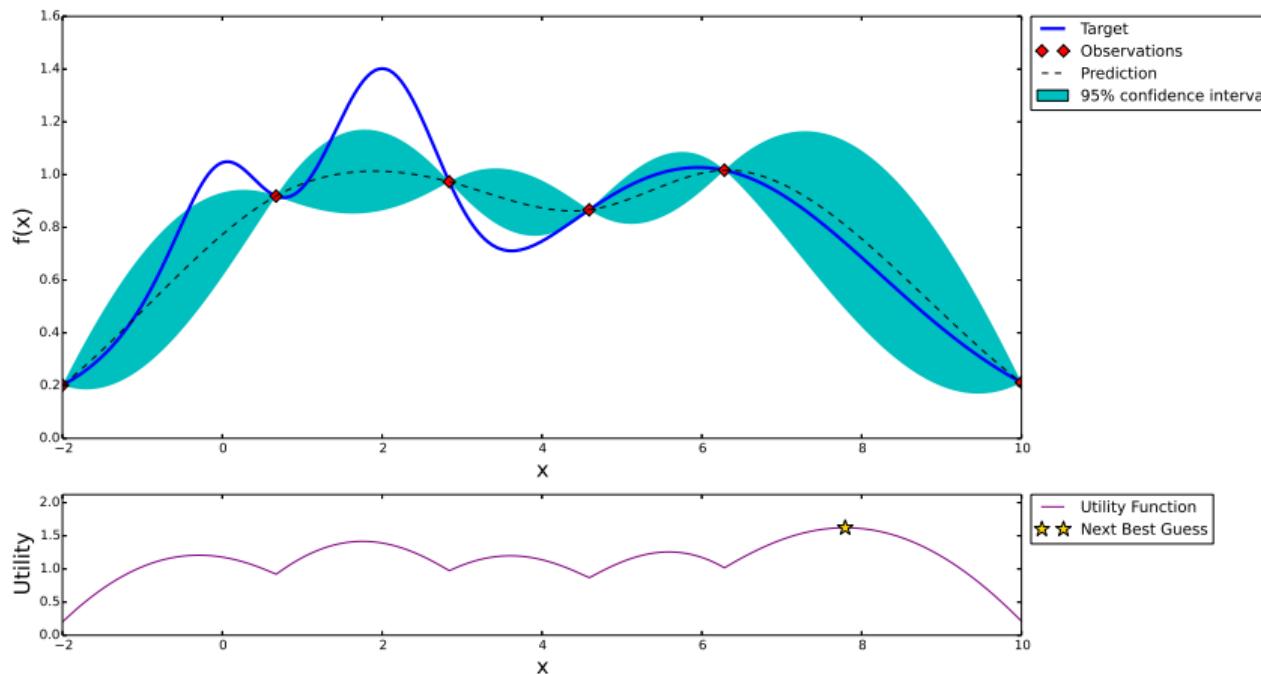
Gaussian Process and Utility Function After 5 Steps



Bayesian Optimization

Example

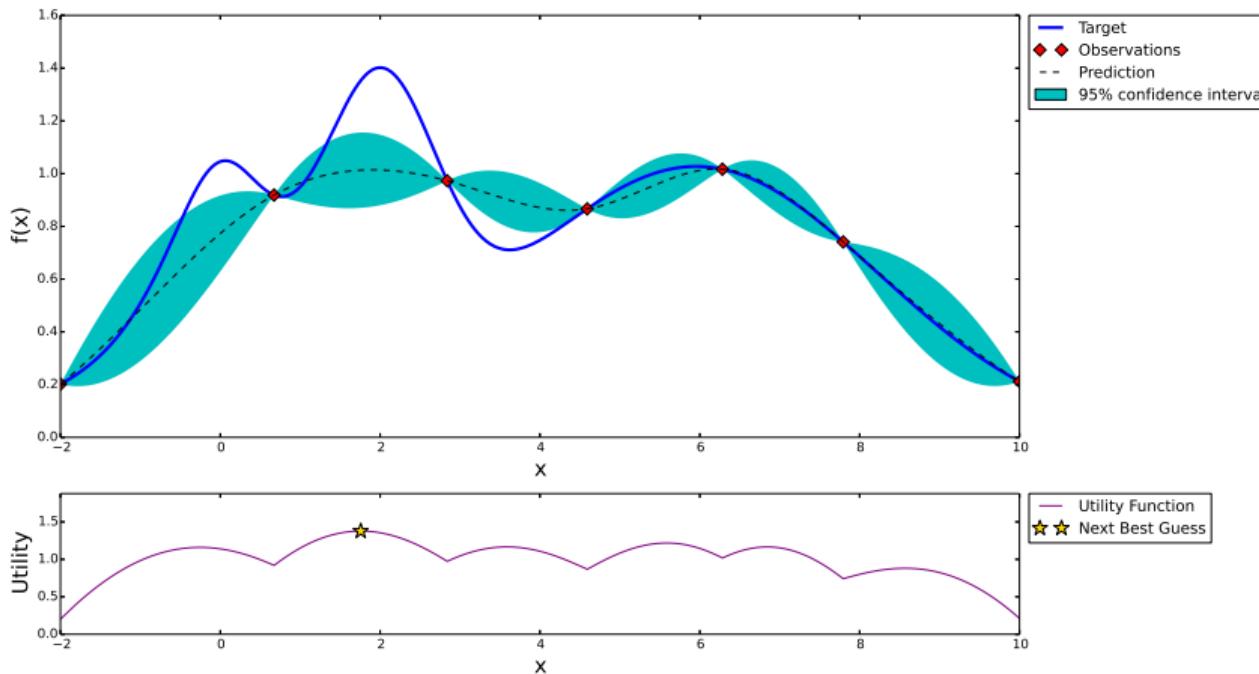
Gaussian Process and Utility Function After 6 Steps



Bayesian Optimization

Example

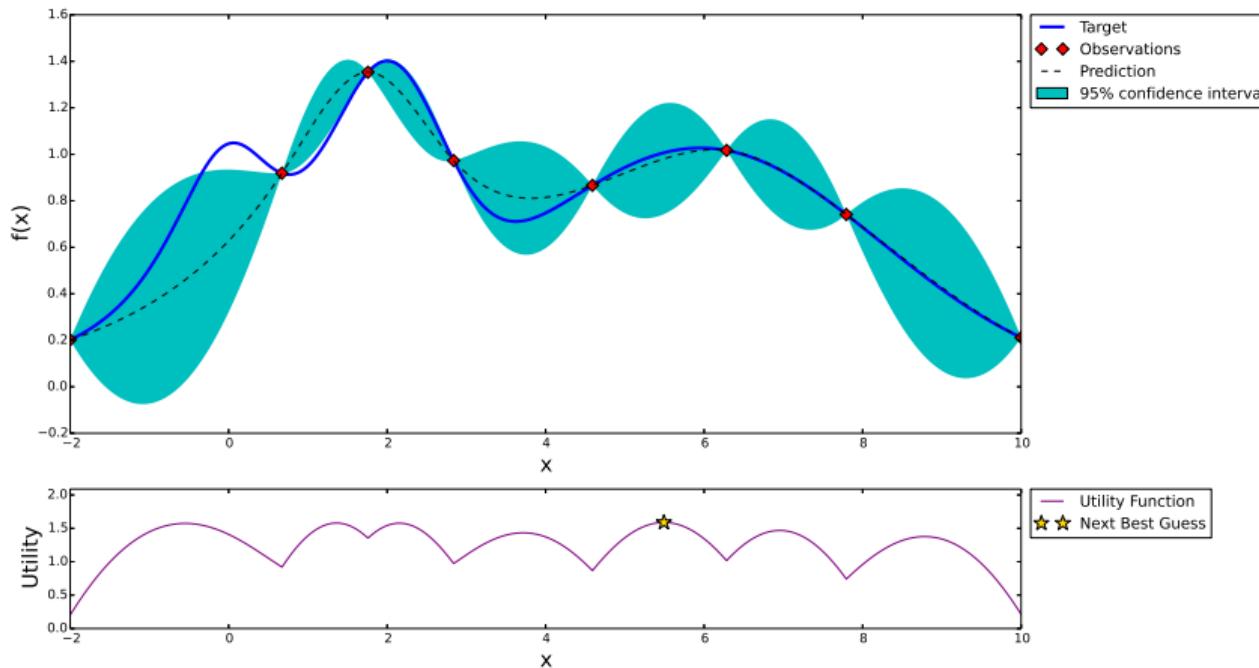
Gaussian Process and Utility Function After 7 Steps



Bayesian Optimization

Example

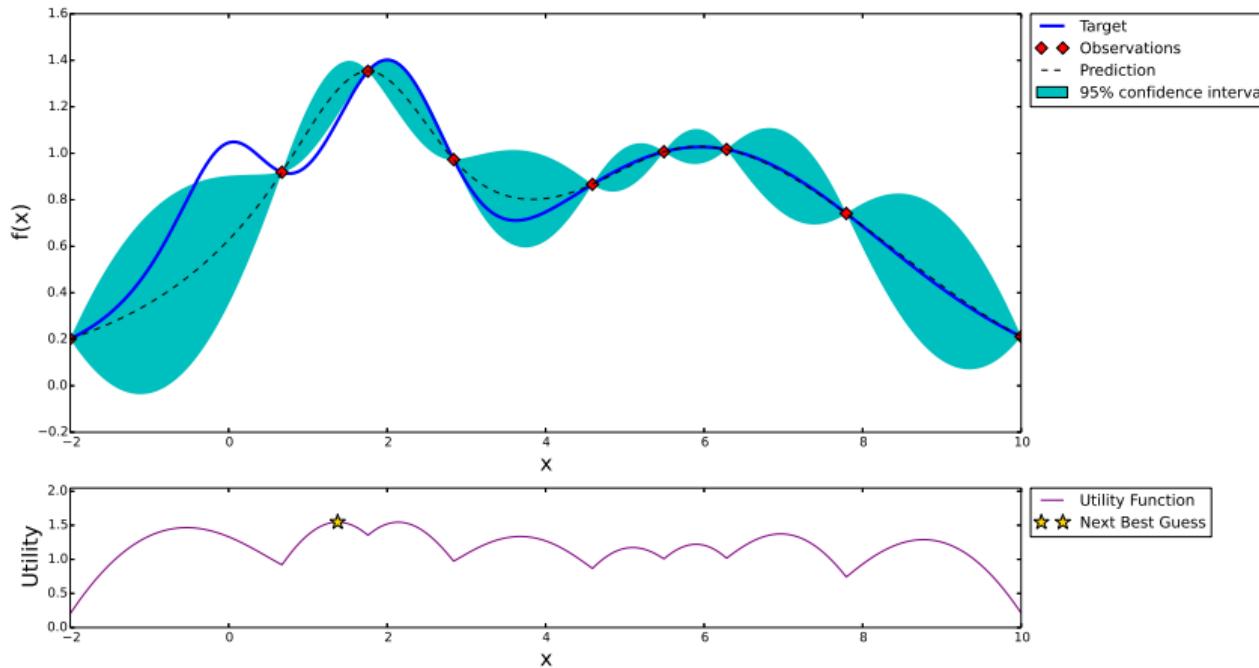
Gaussian Process and Utility Function After 8 Steps



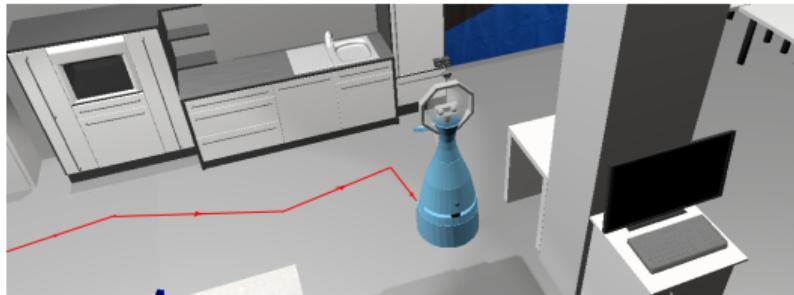
Bayesian Optimization

Example

Gaussian Process and Utility Function After 9 Steps



Learning Probabilistic Models for Planning under Uncertainty by Bayesian Optimization of Unsupervised Machine Learning Algorithms for State-Spaces



Rob van Bekkum

Delft University of Technology

April 5, 2017