Learning and Optimizing Probabilistic Models for Planning under Uncertainty

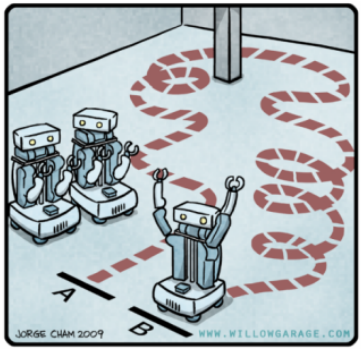
R. van Bekkum

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

September 27, 2017

Planning under Uncertainty

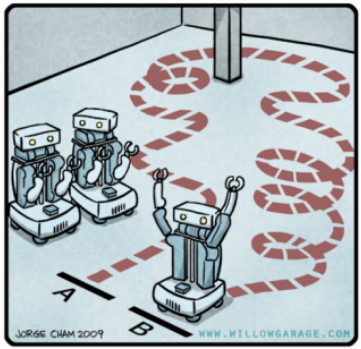
Domain Characteristics



Planning under Uncertainty

Domain Characteristics

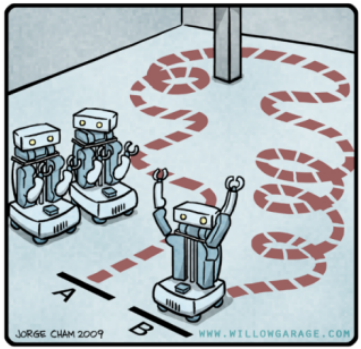
- A system is controlled by one or more *agents*



Planning under Uncertainty

Domain Characteristics

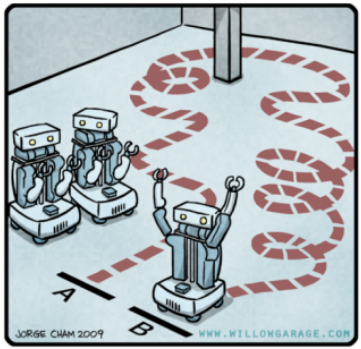
- A system is controlled by one or more *agents*
- *Uncertain domain dynamics*, i.e. uncertainty may be present in:
  - Execution of actions (e.g., robot may slip)
  - Exogenous factors (e.g., doors open/closed)
  - Percepts (e.g., sensor noise)



Planning under Uncertainty

Domain Characteristics

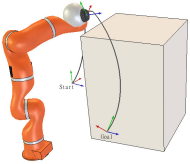
- A system is controlled by one or more *agents*
- *Uncertain domain dynamics*, i.e. uncertainty may be present in:
  - Execution of actions (e.g., robot may slip)
  - Exogenous factors (e.g., doors open/closed)
  - Percepts (e.g., sensor noise)
- *Sequential decision making*
  - Non-myopic agents
  - Selection of actions with high future pay-off



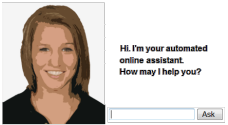
Planning under Uncertainty

Example Domains

Path planning [7, 8]



Motion planning



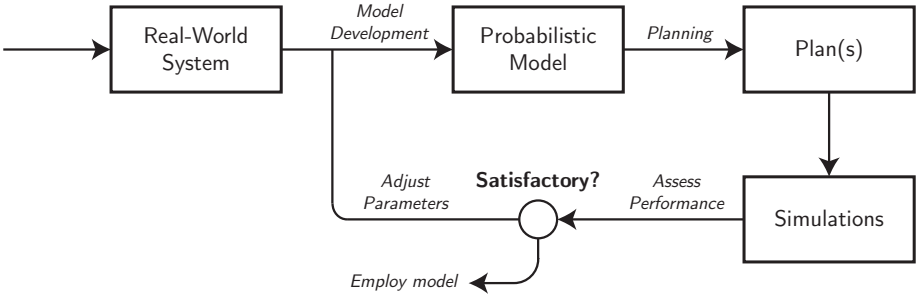
Dialog Management [2]

Operations planning [1, 13]

Image credits: Hawes et al. [7], Iocchi et al. [8], V-REP Manual, Bemidji State University

Planning under Uncertainty

Typical Development Routine



Planning under Uncertainty

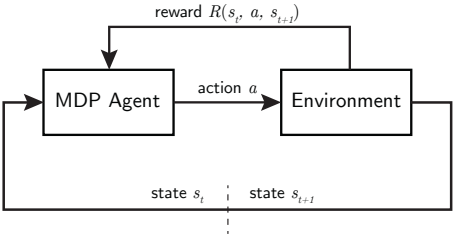
Markov Decision Processes (MDPs)

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
- $\delta : \mathcal{S} \times A \times \mathcal{S} \mapsto [0, 1]$ is the transition function
- $R : \mathcal{S} \times A \times \mathcal{S} \mapsto \mathbb{R}$ is the reward function

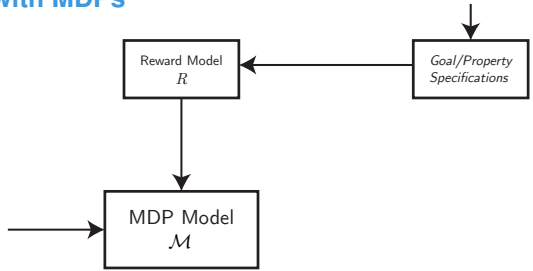


Planning under Uncertainty

Planning with MDPs

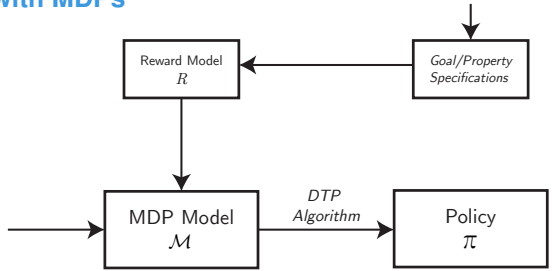
Planning under Uncertainty

Planning with MDPs



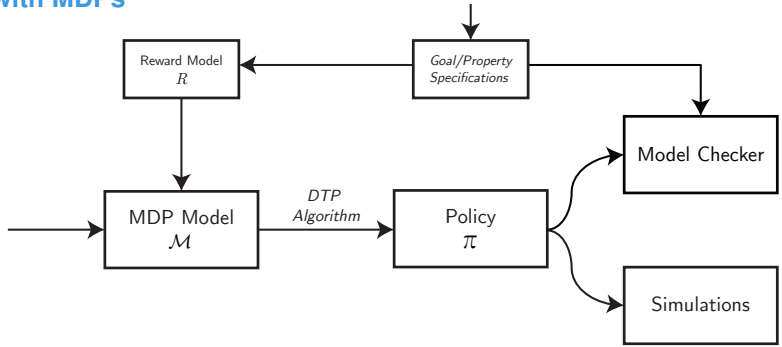
Planning under Uncertainty

Planning with MDPs



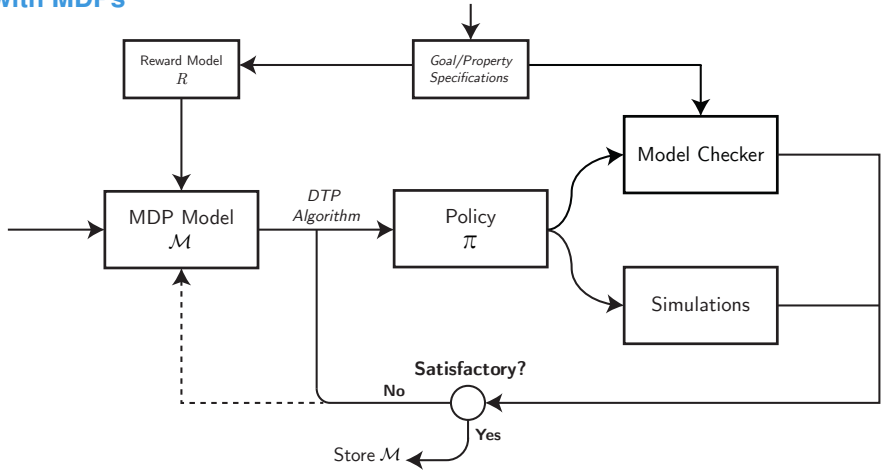
Planning under Uncertainty

Planning with MDPs



Planning under Uncertainty

Planning with MDPs



Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

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

Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

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

Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

Idea: Automate the development process through learning algorithms

Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

Idea: Automate the development process through learning algorithms

- Making use of a dataset describing the system's dynamics

Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

Idea: Automate the development process through learning algorithms

- Making use of a dataset describing the system's dynamics
- Learn the components of MDPs:
  - State space (e.g., clustering, time-state merging)
  - Transition probabilities (e.g., maximum likelihood, Bayesian inference)
  - Emission probabilities in POMDPs (e.g., Baum-Welch, gradient ascent)

Planning under Uncertainty

Model Development

Problem: How to obtain a suitable MDP for offline planning?

Idea: Automate the development process through learning algorithms

- Making use of a dataset describing the system's dynamics
- Learn the components of MDPs:
  - State space (e.g., clustering, time-state merging)
  - Transition probabilities (e.g., maximum likelihood, Bayesian inference)
  - Emission probabilities in POMDPs (e.g., Baum-Welch, gradient ascent)

Next Question: How to set the hyperparameters for these algorithms?

Problem Description

Research Questions [1/2]

Problem Description

Research Questions [1/2]

Main Research Question

How can the task of obtaining a (discrete) MDP that maximizes the yielded performance of executing plans that are derived from it, given a dataset about the system under consideration, be automated?

Problem Description

Research Questions [1/2]

Main Research Question

How can the task of obtaining a (discrete) MDP that maximizes the yielded performance of executing plans that are derived from it, given a dataset about the system under consideration, be automated?

RQ1 Which learning algorithms exist that can be employed for learning MDPs from data for systems involving uncertainty that require plans for automated control?

Problem Description

Research Questions [1/2]

Main Research Question

How can the task of obtaining a (discrete) MDP that maximizes the yielded performance of executing plans that are derived from it, given a dataset about the system under consideration, be automated?

RQ1 Which learning algorithms exist that can be employed for learning MDPs from data for systems involving uncertainty that require plans for automated control?

RQ2 How should a performance measure be defined which can be used to fairly compare the value of different MDPs?

Problem Description

Research Questions [2/2]

Main Research Question

How can the task of obtaining a (discrete) MDP that maximizes the yielded performance of executing plans that are derived from it, given a dataset about the system under consideration, be automated?

RQ3 How can the parameter space of model learning algorithms cost-effectively be explored towards a global maximizer with only limited knowledge about the system under consideration?

Problem Description

Research Questions [2/2]

Main Research Question

How can the task of obtaining a (discrete) MDP that maximizes the yielded performance of executing plans that are derived from it, given a dataset about the system under consideration, be automated?

RQ3 How can the parameter space of model learning algorithms cost-effectively be explored towards a global maximizer with only limited knowledge about the system under consideration?

RQ4 How can the hierarchy of different abstraction levels be exploited to find a performance-maximizing MDP in a more cost-effective way?

Problem Description

Problem Statement

Problem Statement

Given a set E of execution traces describing the dynamics of a system that involves uncertainty,

Problem Description

Problem Statement

Problem Statement

Given a set E of execution traces describing the dynamics of a system that involves uncertainty, an MDP $\mathcal{M}$ needs to be developed, utilizing learning algorithms parameterized by a set of hyperparameters θ

Problem Description

Problem Statement

Problem Statement

Given a set E of execution traces describing the dynamics of a system that involves uncertainty, an MDP $\mathcal{M}$ needs to be developed, utilizing learning algorithms parameterized by a set of hyperparameters θ configured in such way to maximize the performance yielded by following the policies computed from this model in a real-world setting.

Problem Description

Related Work

Problem Description

Related Work

- Model Learning Algorithms
  - Baum-Welch for HMMs/POMDPs [15], e.g., Shatkay et al. [12]
  - Best-First Model Merging
  - State-Merging by Trajectory Clustering [10]

Problem Description

Related Work

- Model Learning Algorithms
  - Baum-Welch for HMMs/POMDPs [15], e.g., Shatkay et al. [12]
  - Best-First Model Merging
  - State-Merging by Trajectory Clustering [10]
- Reinforcement Learning
  - Model-based RL and Integrated Architectures, e.g. DYNA [6]
  - Active RL [4] - Uses provided MDP as blueprint for exploration
  - Bayesian RL [11] - Resolves exploration-exploitation dilemma by planning in the belief space

Problem Description

Related Work

- Model Learning Algorithms
  - Baum-Welch for HMMs/POMDPs [15], e.g., Shatkay et al. [12]
  - Best-First Model Merging
  - State-Merging by Trajectory Clustering [10]
- Reinforcement Learning
  - Model-based RL and Integrated Architectures, e.g. DYNA [6]
  - Active RL [4] - Uses provided MDP as blueprint for exploration
  - Bayesian RL [11] - Resolves exploration-exploitation dilemma by planning in the belief space
- MDPs with Transition Probability Uncertainty
  - MDP-IPs [3]
  - BMDPs [5]

Proposed Solution

General Idea

- Explore the hyperparameter space of model learning algorithms

Proposed Solution

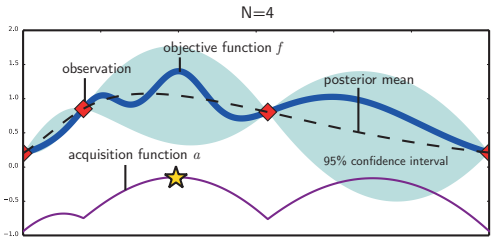
General Idea

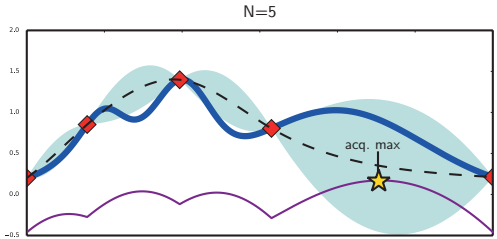
- Explore the hyperparameter space of model learning algorithms
- Use *Bayesian Optimization* to sequentially sample hyperparameter settings θ towards a global maximizer of a performance measure of learned MDPs

Proposed Solution

General Idea

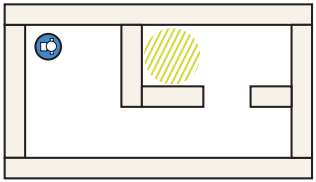
- Explore the hyperparameter space of model learning algorithms
- Use *Bayesian Optimization* to sequentially sample hyperparameter settings θ towards a global maximizer of a performance measure of learned MDPs





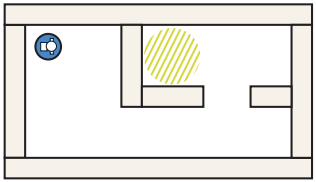
Proposed Solution

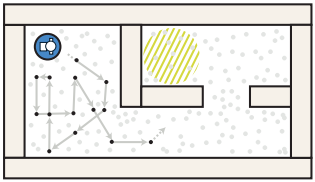
Running Example: Path Planning for Mobile Robot Navigation



Proposed Solution

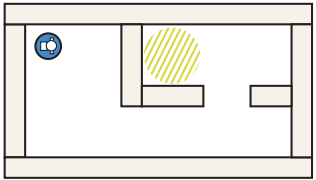
Running Example: Path Planning for Mobile Robot Navigation

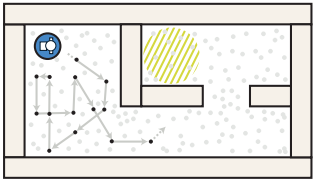


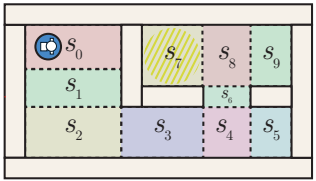


Proposed Solution

Running Example: Path Planning for Mobile Robot Navigation

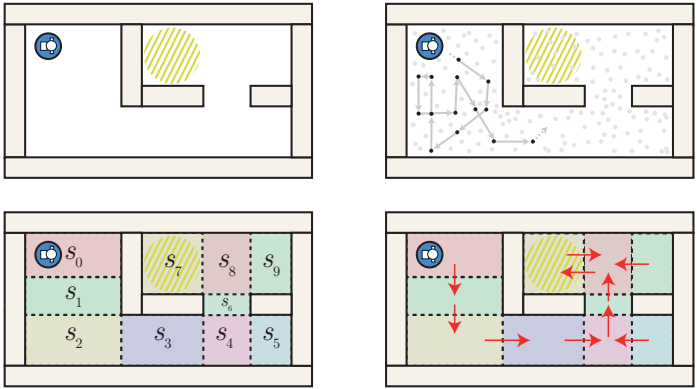






Proposed Solution

Running Example: Path Planning for Mobile Robot Navigation



Proposed Solution

Performance Measure

- Performance is assessed over multiple tasks T (each mappable to a pair (s_0, R))

Proposed Solution

Performance Measure

- Performance is assessed over multiple tasks T (each mappable to a pair (s_0, R))
- Based on Value Functions from DTP algorithm (e.g., VI, PI):

$$V_{DTP,(s_0,R)} = V[s_0]$$

Proposed Solution

Performance Measure

- Performance is assessed over multiple tasks T (each mappable to a pair (s_0, R))
- Based on Value Functions from DTP algorithm (e.g., VI, PI):

$$V_{DTP,(s_0,R)} = V[s_0]$$

- Based on Simulations:

$$V_{SIM,(s_0,R)} = \gamma^n \cdot R[i]$$

Proposed Solution

Performance Measure

- Performance is assessed over multiple tasks T (each mappable to a pair (s_0, R))
- Based on Value Functions from DTP algorithm (e.g., VI, PI):

$$V_{DTP,(s_0,R)} = V[s_0]$$

- Based on Simulations:

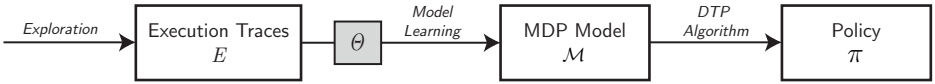
$$V_{SIM,(s_0,R)} = \gamma^n \cdot R[i]$$

- Combined:

$$V_{\mathcal{M}} = \frac{\sum_{t \in T_{\mathcal{M}}} \beta \cdot V_{DTP,t} + (1-\beta) \cdot V_{SIM,t}}{|T_{\mathcal{M}}|}$$

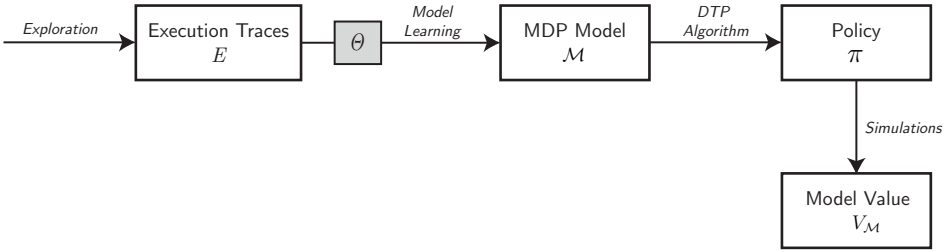
Proposed Solution

Base Framework



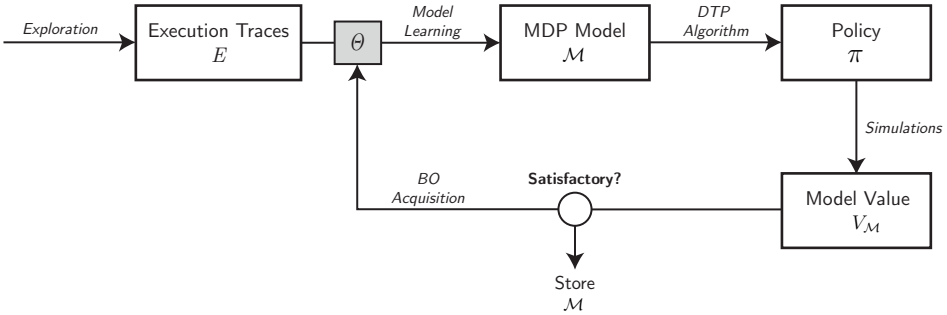
Proposed Solution

Base Framework



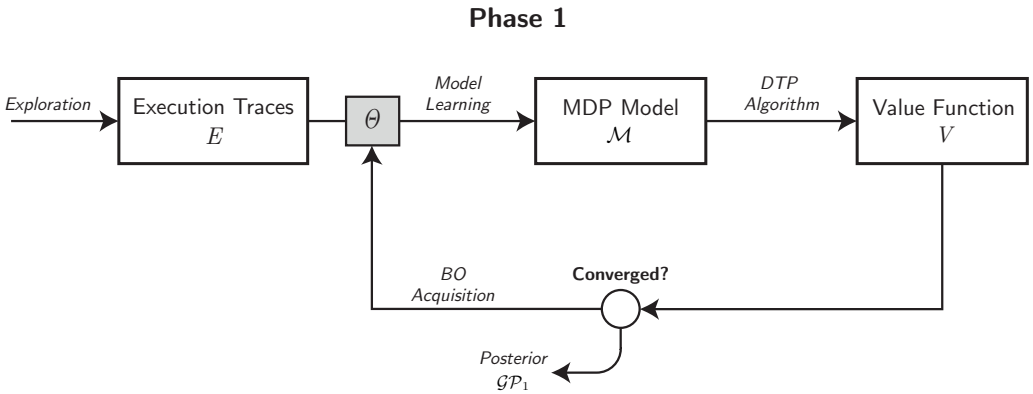
Proposed Solution

Base Framework



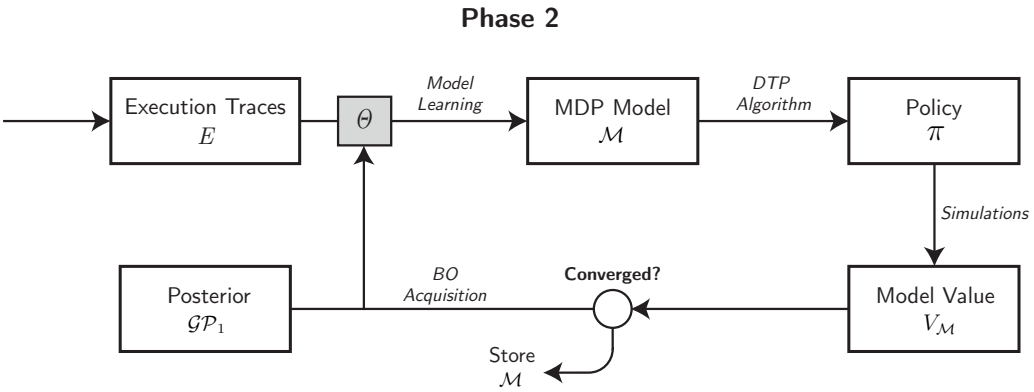
Proposed Solution

Multi-Phase Framework: Phase 1



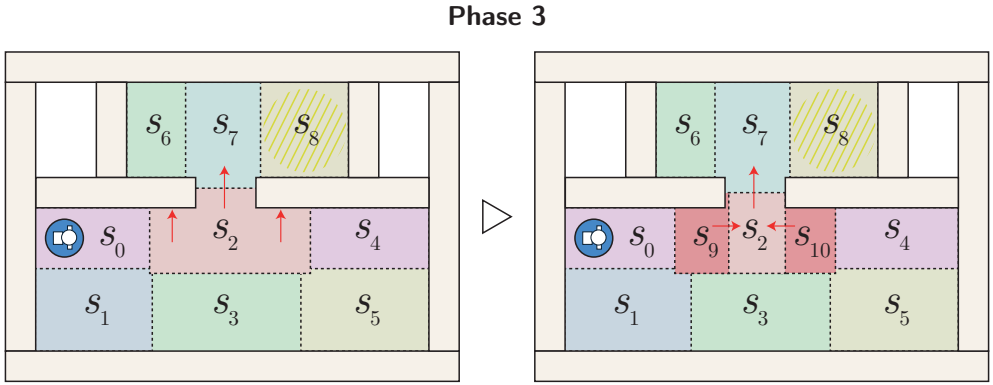
Proposed Solution

Multi-Phase Framework: Phase 2



Proposed Solution

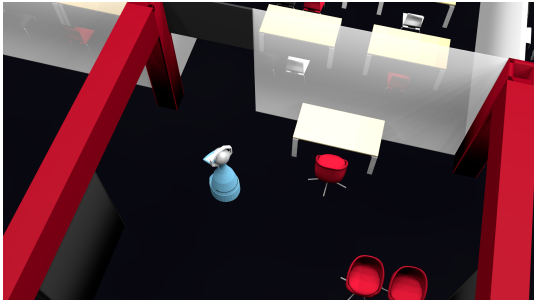
Multi-Phase Framework: Phase 3



Experimental Results

Implementation

- **Domain**: Mobile Robot Navigation

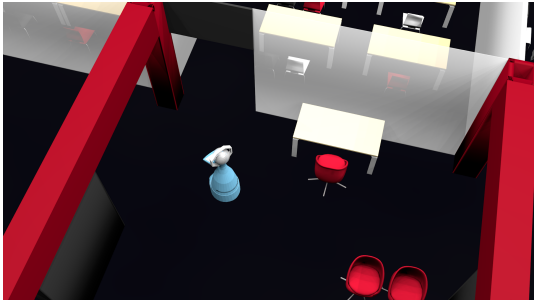


SCITOS-A5 mobile robot in a Morse simulation of the `uol_bl` *environment*

Experimental Results

Implementation

- **Domain**: Mobile Robot Navigation
- Execution traces with odometric readings
- State space discretized by unsupervised machine learning (k-Means, GMM)
- Transition probabilities of MDPs computed by maximum likelihood

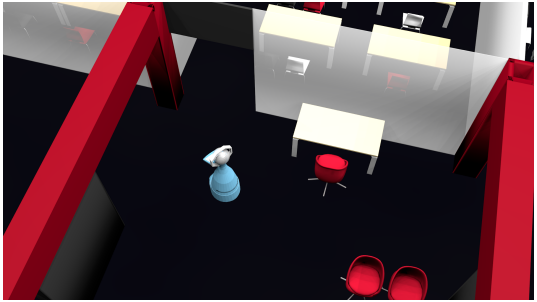


SCITOS-A5 mobile robot in a Morse simulation of the `uol_bl` *environment*

Experimental Results

Implementation

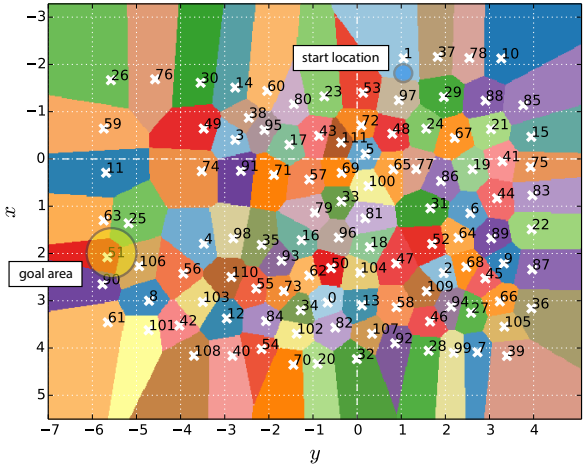
- **Domain**: Mobile Robot Navigation
- Execution traces with odometric readings
- State space discretized by unsupervised machine learning (k-Means, GMM)
- Transition probabilities of MDPs computed by maximum likelihood
- Performance assessed through simulations
- Tested on small and large environment
- **Software**: Python, ROS, *Morse* simulator



SCITOS-A5 mobile robot in a Morse simulation of the `uol_bl` *environment*

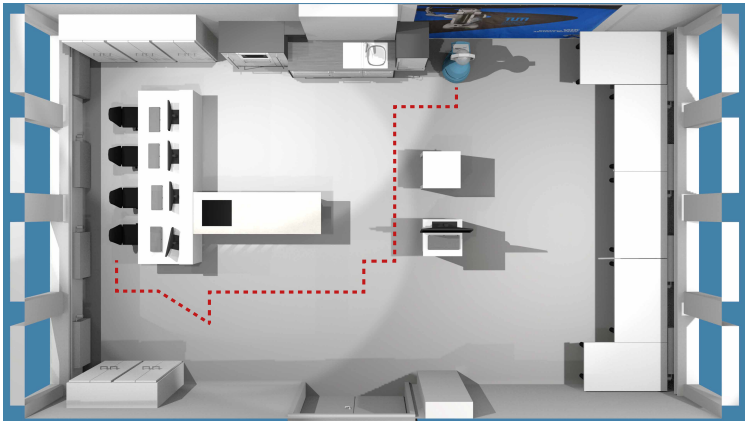
Experimental Results

Demonstration [1/2]



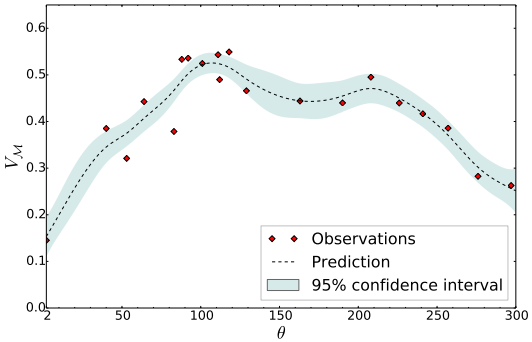
Experimental Results

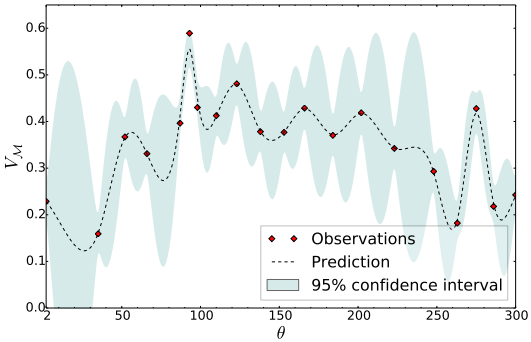
Demonstration [2/2]



Experimental Results

Base Framework - Small Environment - Dataset Size

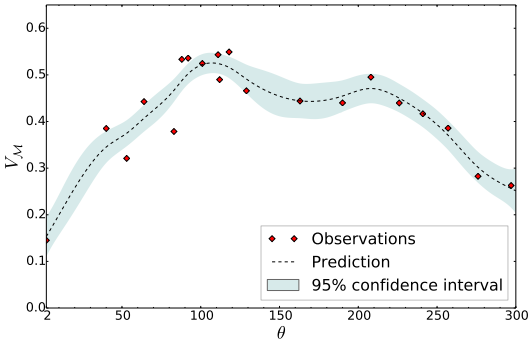


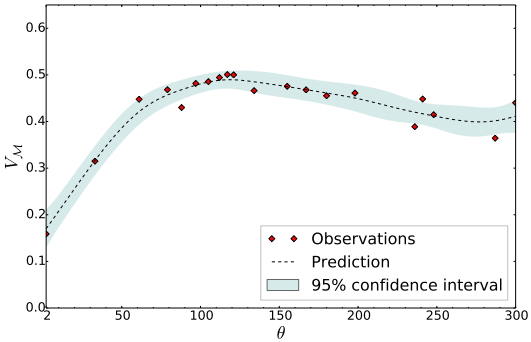


Resulting GP approximation with 100% and 50% of our dataset used
(k-Means, `tum_kitchen` environment, EI acquisition, 20 iterations)

Experimental Results

Base Framework - Small Environment - β factor



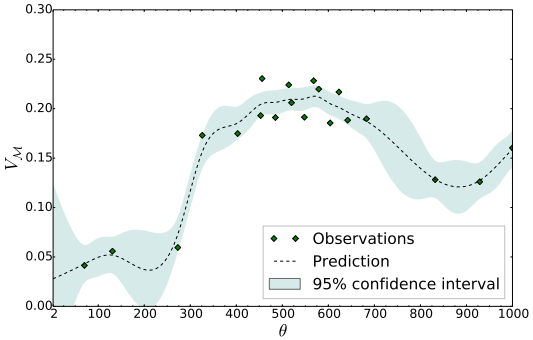


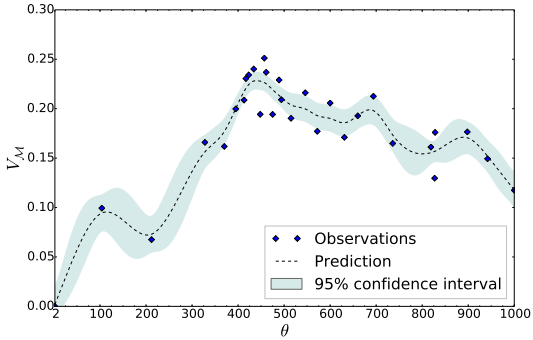
Resulting GP approximation with $\beta = 0.0$ and $\beta = 0.5$

(k-Means, `tum_kitchen` environment, EI acquisition, 20 iterations, 100% dataset used)

Experimental Results

Base Framework - Large Environment

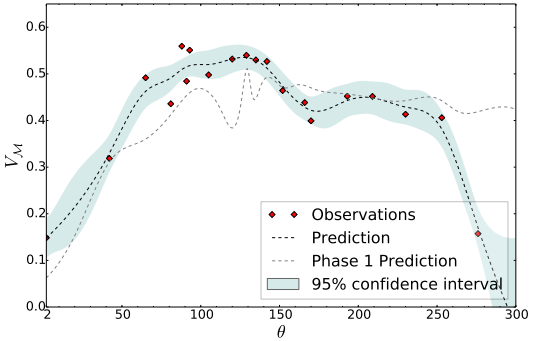


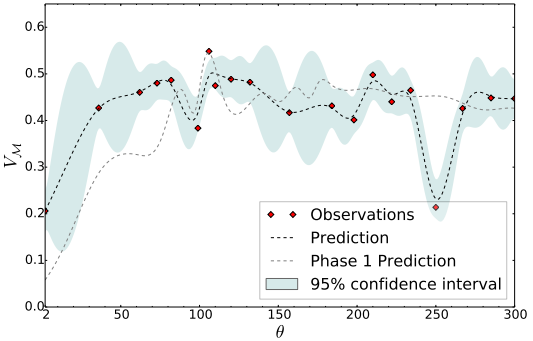


Resulting GP approximation with GP-UCB and EIPS acquisition functions used
(k-Means, `uol_bl` environment, 100% dataset used)

Experimental Results

Multi-Phase Framework - Small Environment

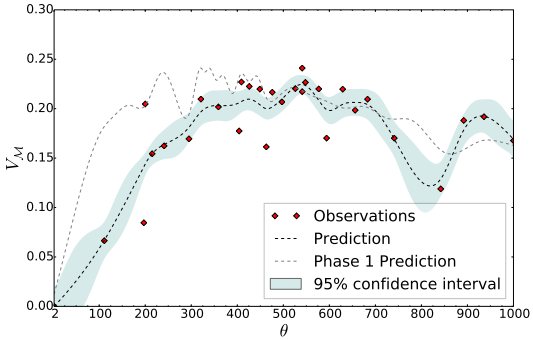


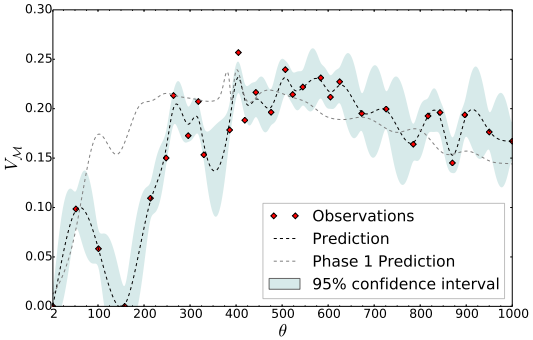


Resulting GP approximation with EI and GP-UCB acquisition functions used in Phase 1 (k-Means, `tum_kitchen` environment, 20 iterations, 100% dataset used)

Experimental Results

Multi-Phase Framework - Large Environment





Resulting GP approximation with GP-UCB and EIPS acquisition functions used
(k-Means, `uol_bl` environment, 30 iterations, 100% dataset used)

Discussion

Conclusions:

- Handcrafting MDPs for planning under uncertainty is a daunting task
  - Given a dataset, learning algorithms can ease the development

Discussion

Conclusions:

- Handcrafting MDPs for planning under uncertainty is a daunting task
  - Given a dataset, learning algorithms can ease the development
- Evaluating all hyperparameter settings is a cost-expensive endeavor
  - Determining appropriate settings more effectively approached by BO

Discussion

Conclusions:

- Handcrafting MDPs for planning under uncertainty is a daunting task
  - Given a dataset, learning algorithms can ease the development
- Evaluating all hyperparameter settings is a cost-expensive endeavor
  - Determining appropriate settings more effectively approached by BO
- The multi-phase approach can help to find a global maximizer of the performance faster
  - With the first phase abstracting more, but being significantly cheaper

Discussion

Future work suggestions:

- Investigate other application domains, e.g., learning POMDPs for dialogue management [2]

Discussion

Future work suggestions:

- Investigate other application domains, e.g., learning POMDPs for dialogue management [2]
- Automated Model Checking [9]

Discussion

Future work suggestions:

- Investigate other application domains, e.g., learning POMDPs for dialogue management [2]
- Automated Model Checking [9]

- Combination with model-based RL techniques (using offline learned MDPs as blueprints for exploration)

Discussion

Future work suggestions:

- Investigate other application domains, e.g., learning POMDPs for dialogue management [2]
- Automated Model Checking [9]

- Combination with model-based RL techniques (using offline learned MDPs as blueprints for exploration)
- Coverage and size of datasets for model learning

Learning and Optimizing Probabilistic Models for Planning under Uncertainty

R. van Bekkum

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

September 27, 2017

References I

D. Aberdeen, S. Thiébaux, and L. Zhang.
Decision-Theoretic Military Operations Planning.
In *Proceedings of the 14th International Conference on Automated Planning and Scheduling*, ICAPS'04, 2004.

H. R. Chinaei and B. Chaib-draa.
Learning Dialogue POMDP Models from Data, pages 86–91.
CAI'11. Springer Berlin Heidelberg, Berlin, Heidelberg, May 2011.

K. V. Delgado, S. Sanner, and L. N. De Barros.
Efficient Solutions to Factored MDPs with Imprecise Transition Probabilities.
Artificial Intelligence, 175(9-10):1498–1527, 2011.

A. Epshteyn, A. Vogel, and G. DeJong.
Active Reinforcement Learning.
In *Proceedings of the 25th International Conference on Machine Learning*, ICML'08, pages 296–303, New York, NY, USA, 2008. ACM.

References II

R. Givan, S. Leach, and T. Dean.
Bounded-parameter Markov Decision Processes.
Artificial Intelligence, 122(1-2):71–109, 2000.

A. Guez, D. Silver, and P. Dayan.
Efficient Bayes-Adaptive Reinforcement Learning using Sample-Based Search.
In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 1025–1033. Curran Associates Inc., 2012.

N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrová, J. Young, J. Wyatt, D. Hebesberger, T. Körtner, et al.
The STRANDS Project: Long-Term Autonomy in Everyday Environments.
IEEE Robotics Automation Magazine, 24(3):146–156, September 2017.

L. Iocchi, L. Jeanpierre, M. T. Lázaro, and A.-I. Mouaddib.
A Practical Framework for Robust Decision-theoretic Planning and Execution for Service Robots.
In *Proceedings of the 26th International Conference on Automated Planning and Scheduling*, ICAPS'16, pages 486–494. AAAI Press, 2016.

References III


B. Lacerda, D. Parker, and N. Hawes.
Optimal Policy Generation for Partially Satisfiable Co-Safe LTL Specifications.
In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1587–1593, 2015.


D. N. Nikovski and I. R. Nourbakhsh.
Learning Probabilistic Models for Decision-Theoretic Navigation of Mobile Robots.
In *Proceedings of the 17th International Conference on Machine Learning*, ICML'00, pages 671–678. Morgan Kaufmann Publishers Inc., 2000.


P. Poupart.
Bayesian Reinforcement Learning, pages 90–93.
Springer US, Boston, MA, USA, 2010.


H. Shatkay and L. P. Kaelbling.
Learning Hidden Markov Models with Geometric Information.
Technical report, Providence, RI, USA, 1997.


O. Sigaud and O. Buffet.
Markov Decision Processes in Artificial Intelligence.
Wiley-IEEE Press, 2010.

References IV

A. Stolcke and S. M. Omohundro.
Best-First Model Merging for Hidden Markov Model Induction.
Technical report, Berkeley, CA, USA, 1994.

L. R. Welch.
Hidden Markov Models and the Baum-Welch Algorithm.
IEEE Information Theory Society Newsletter, 53(4):10–13, 2003.

Bayesian Optimization

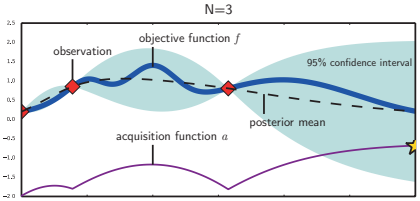
- Given an expensive, unknown objective function $f : \mathcal{X} \mapsto \mathbb{R}$ ($\mathcal{X} \subset \mathbb{R}^m$)
- Find global maximizer $x^* \in \mathcal{X}$ s.t.:

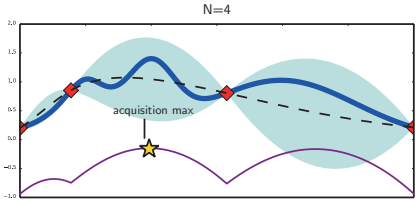
$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

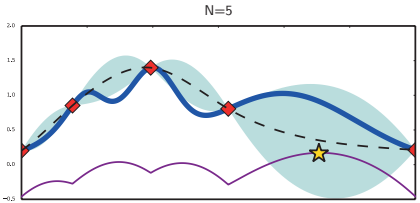
- Bayesian Optimization components:
  - *Prior over* f: Gaussian Process $f \sim GP(\mu(\cdot), K(\cdot, \cdot))$
  - *Evidence set*: $\mathcal{D}_{1:t} = \{(x_i, y_i) \mid i = 1 \dots t\}$
  - *Posterior* or *Surrogate function* ('estimate of f')
  - *Acquisition* or *Utility function*: $u : \mathcal{X}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

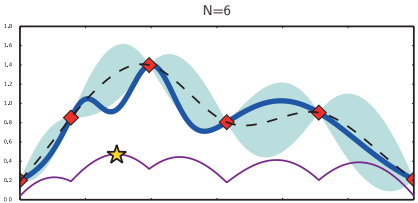
Bayesian Optimization

Toy Example









Bayesian Optimization

Acquisition Functions

Let $f(x^+)$ be the best observation, then the acquisition functions sample the maximizer x of the following quantities:

Probability of Improvement (PI)	$P(f(x) \geq f(x^+) + \xi)$
Expected Improvement (EI)	$\mathbb{E}[\max(0, \mu_f(x) - f(x^+) - \xi)]$
GP Upper Confidence Bound (GP-UCB)	$\mu_f(x) + \beta\sigma_f(x)$
Expected Improvement Per Second (EIPS)	$\mathrm{EI}(x)/\mu_s(x)$

where $\xi \geq 0$ is an exploration-exploitation trade-off parameter, $\beta \geq 0$ specifies number of standard deviations, μ_s is the mean of a timing GP.

Automated Model Checking

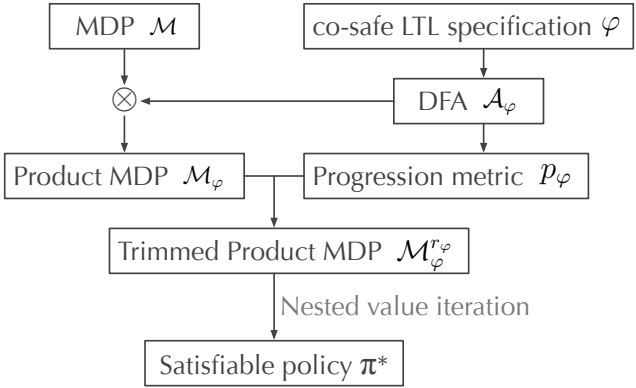
Objective

Goal: Compose algorithms that construct probabilistic models which (partially) satisfy temporal goals.

- Temporal goals expressed through LTL formulas
- Example: $\varphi = (a \,\mathcal{U}\, b) \wedge \mathcal{X} c$
  "a should hold until b is true after which c should become true"

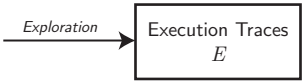
Automated Model Checking

Example Lacerda et al. [9]



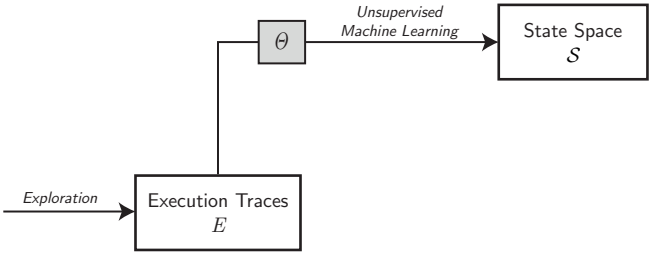
Proposed Solution

Base Framework



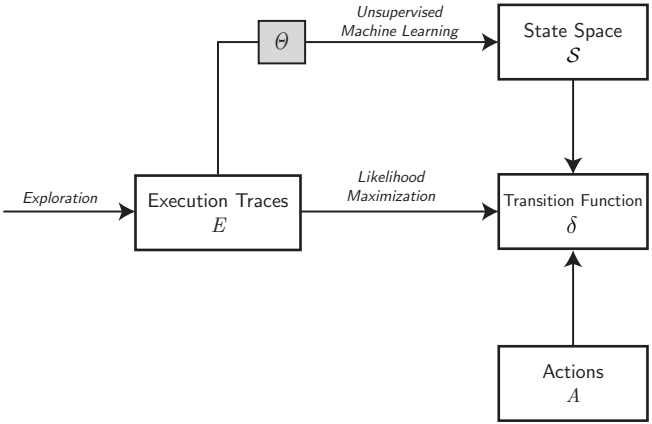
Proposed Solution

Base Framework



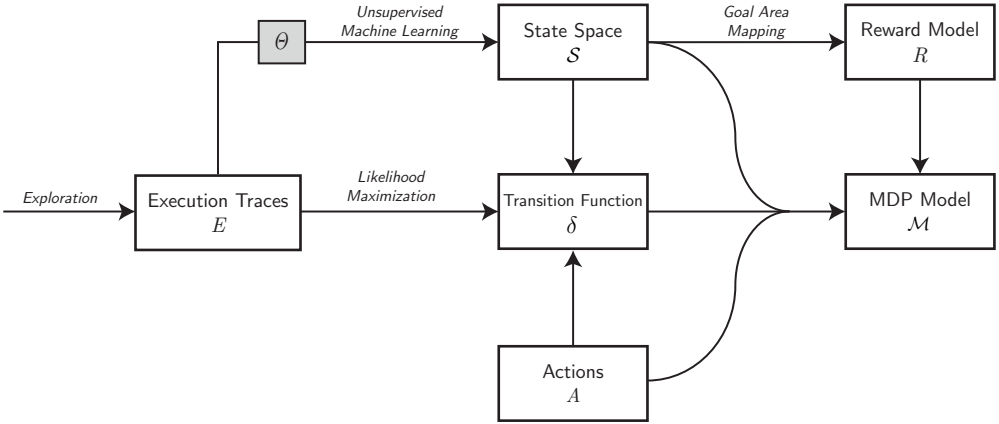
Proposed Solution

Base Framework



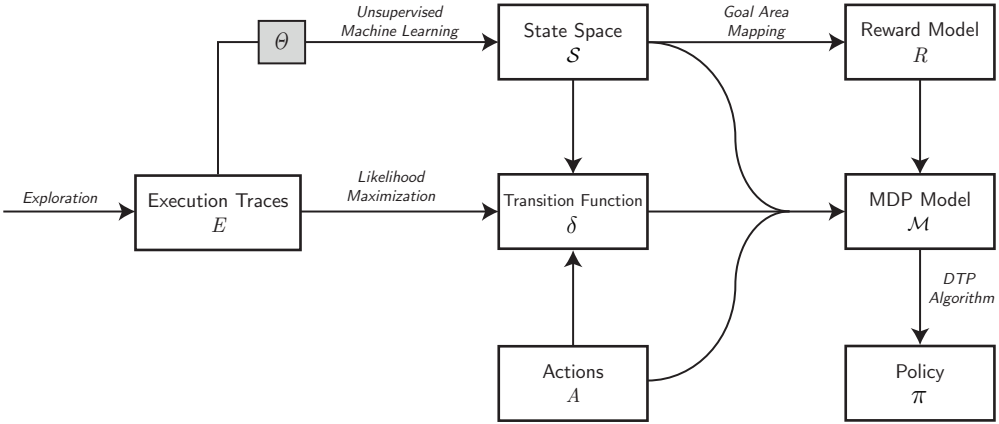
Proposed Solution

Base Framework



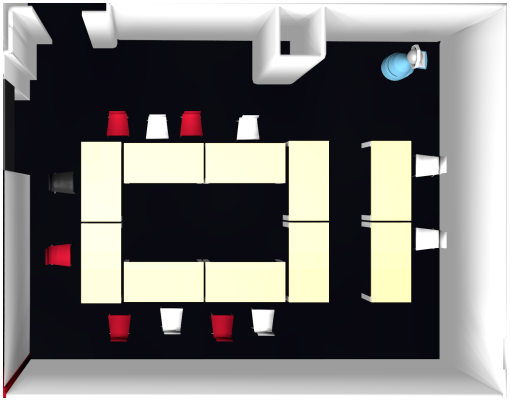
Proposed Solution

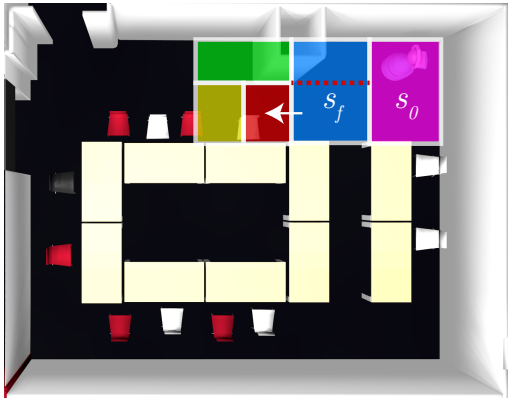
Base Framework



Model Optimization Framework

Multi-Phase Framework: Fine-Tuning Example





Model Learning Algorithms

Baum-Welch for HMMs/POMDPs

Goal: Maximize $P(\lambda|O)$ with λ the model parameters and O the observation sequences

High-level steps:

1. Provide initial probability estimates
2. Repeat the following until convergence:
   1. *E-step*: Compute expectations of how often transitions and emissions are used
   2. *M-step*: Update parameters based on these expectations

Model Learning Algorithms

Baum-Welch for HMMs/POMDPs

Goal: Maximize $P(\lambda|O)$ with λ the model parameters and O the observation sequences

Detailed Steps [15]:

- Compute the following quantities using the *forward* and *backward* procedures given observation sequences O of length T:

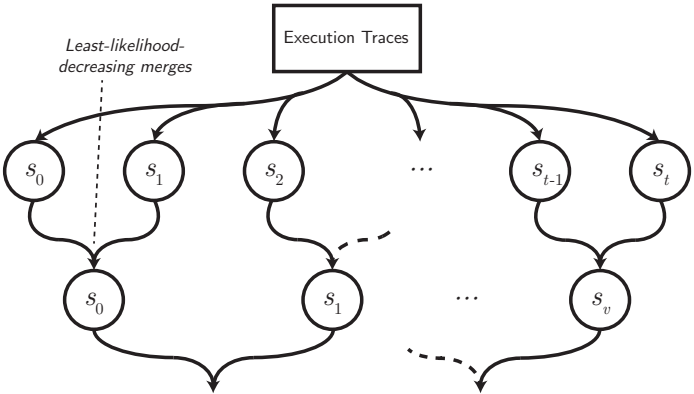
$$\xi_{ij}(t) = P(q_t = s_i, q_{t+1} = s_j) \qquad \gamma_i(t) = \sum_{j=1}^{|\mathcal{S}|} \xi_{ij}(t)$$

- Update model parameters as follows:

$$a_{ij}(t) = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \qquad b_{jk}(t) = \frac{\sum_{o_t = v_k} \gamma_j(t)}{\sum_{t=1}^{T} \gamma_j(t)} \qquad \pi_i = \gamma_i(1)$$

Model Learning Algorithms

Time-State Merging / Model Merging

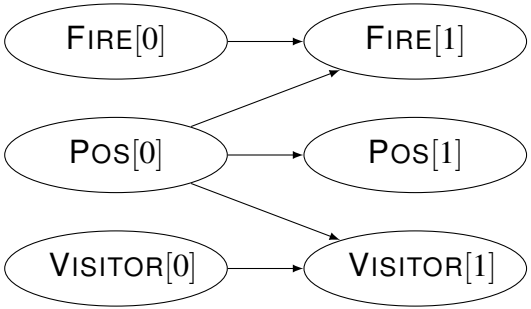


Factored MDPs

Definition

A Factored Markov Decision Process is a tuple $\mathcal{M} = (\mathcal{X}, s_0, A, T, R)$ in which:

- $\mathcal{X} = \{X_1, \ldots, X_n\}$ is a set of state variables (or *features*) with $s_0 \in \mathcal{X}$
- A is the action space
- $T : \mathcal{X} \times A \times \mathcal{X}$ is a transition function
- $R : \mathcal{X} \mapsto \mathbb{R}$ a reward function



Decision epoch t Decision epoch $t + 1$