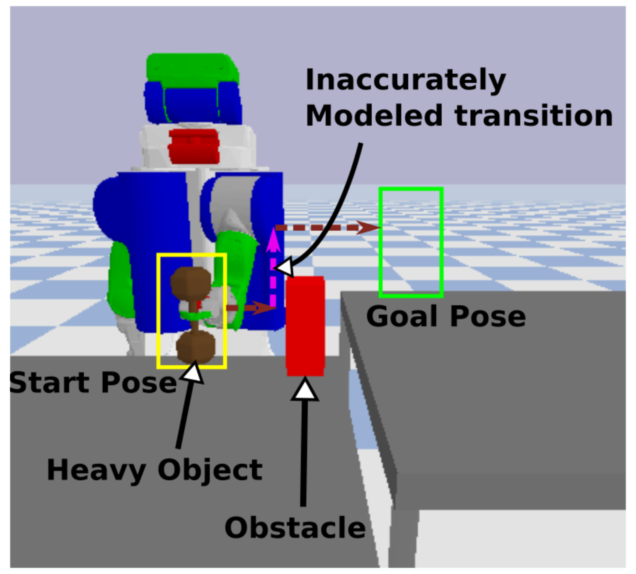
CMAX++ : Leveraging Experience in Planning and Execution using Inaccurate Models

Anirudh Vemula (vemula@cmu.edu), J. Andrew Bagnell and Maxim Likhachev
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

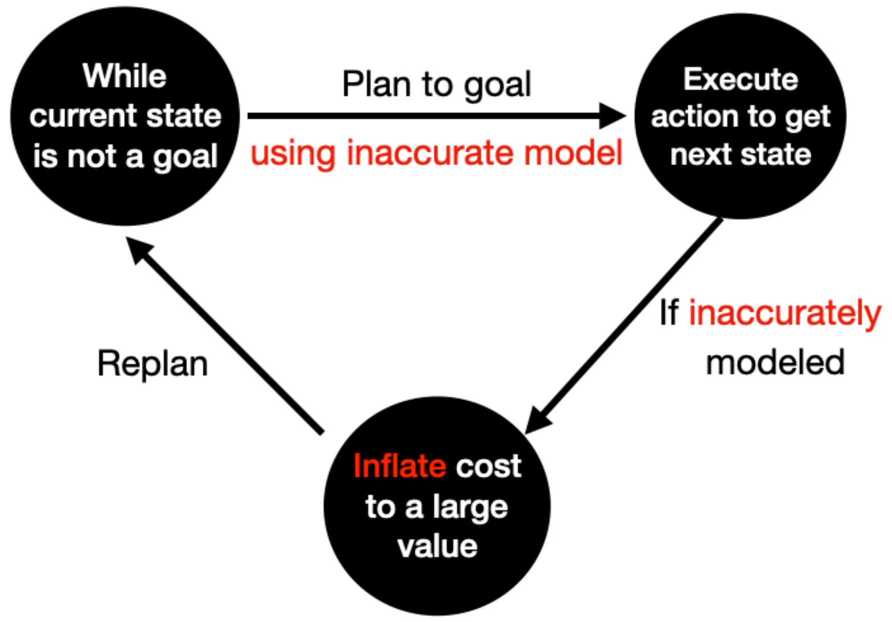
Motivation

- Success of robotic planning mostly in domains with accurate models of robot and environment dynamics
- Hard to model dynamics in the wild - *how do we use inaccurate models and provably complete task?*
- Naively using inaccurate models can result in task failure
- Our focus on *repetitive tasks*
- **Objectives**:
  - Provably complete task in each repetition *without any resets* despite using inaccurate model
  - Improve task performance across repetitions



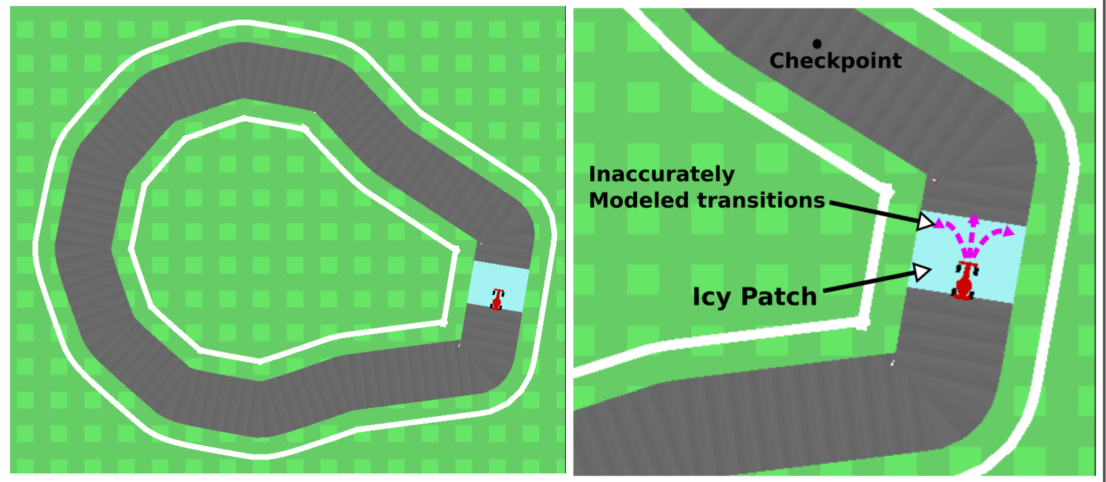
Prior Work

- Updating (residual) dynamical models from executions
  - Large number of samples, require access to resets, no perfect model in model class
- Model-based planning with model-free learning [2,3]
  - Fine-tuning in inaccurately modeled regions, relies on prior knowledge such as inaccuracies/demonstrations
- Updating behavior of planner - CMAX [1]
  - Does not require updating model, no resets required, provably task-complete

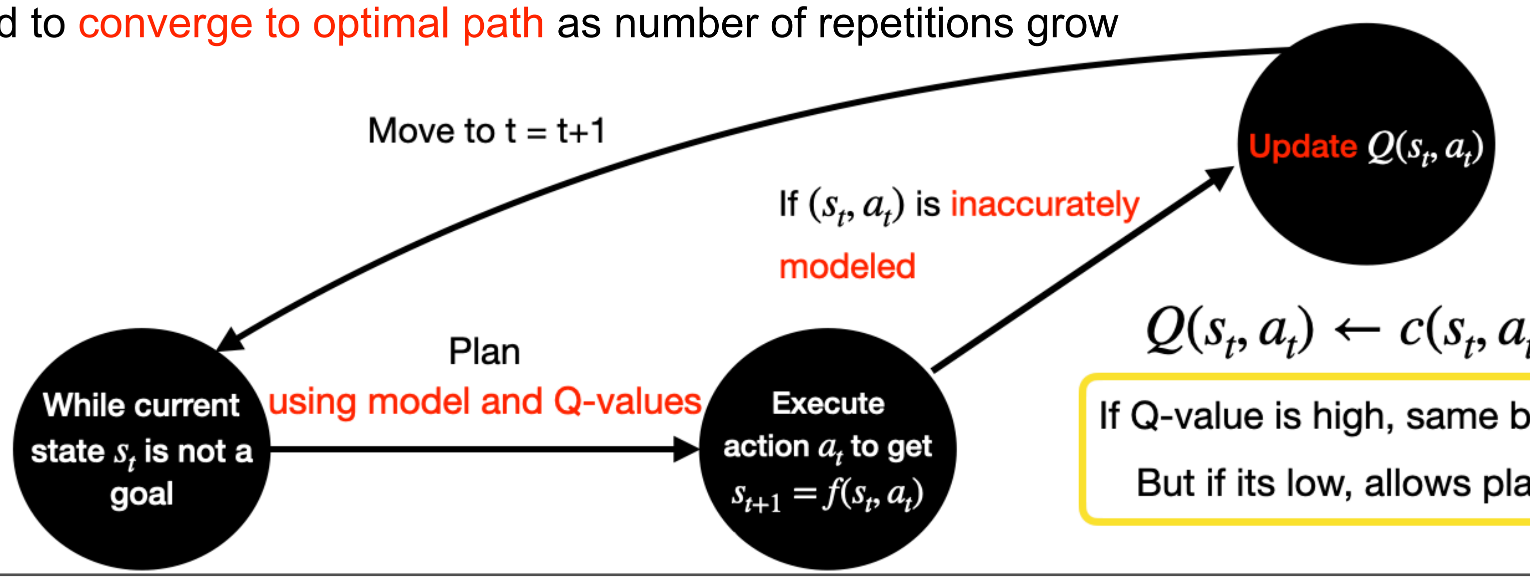


CMAX++

- CMAX fails to improve task performance across repetitions
- Requires strong assumptions on accuracy of the model
- **Key Idea:** *CMAX++ maintains model-free Q-value estimates of inaccurately modeled transitions and uses them in a model-based planning procedure*
- Does not require any updates to the model
- Requires weaker assumptions to guarantee task-completeness



- **Optimistic Model Assumption:** Optimal value function using approximate model dynamics *always underestimates* the optimal value under true dynamics *at all states*
- E.g. Free-space assumption in robot navigation - Robot is never "pleasantly surprised" during execution
- **Theoretical Guarantees:** CMAX++ is guaranteed to be task-complete in each repetition
- Guaranteed to converge to optimal path as number of repetitions grow

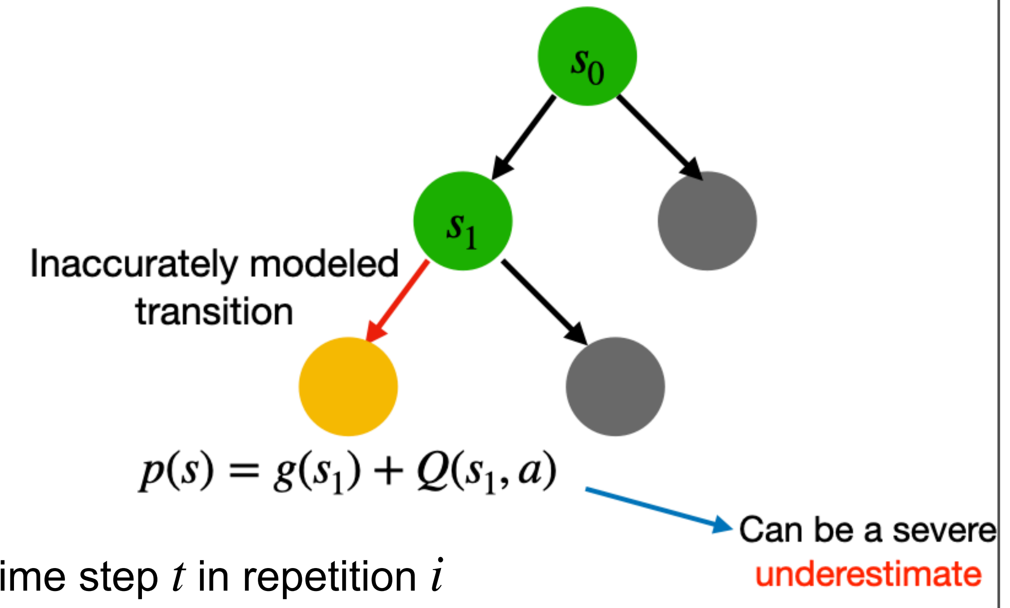


$$Q(s_t, a_t) \leftarrow c(s_t, a_t) + V(s_{t+1})$$

If Q-value is high, same behavior as CMAX
But if its low, allows planner to exploit

Adaptive-CMAX++

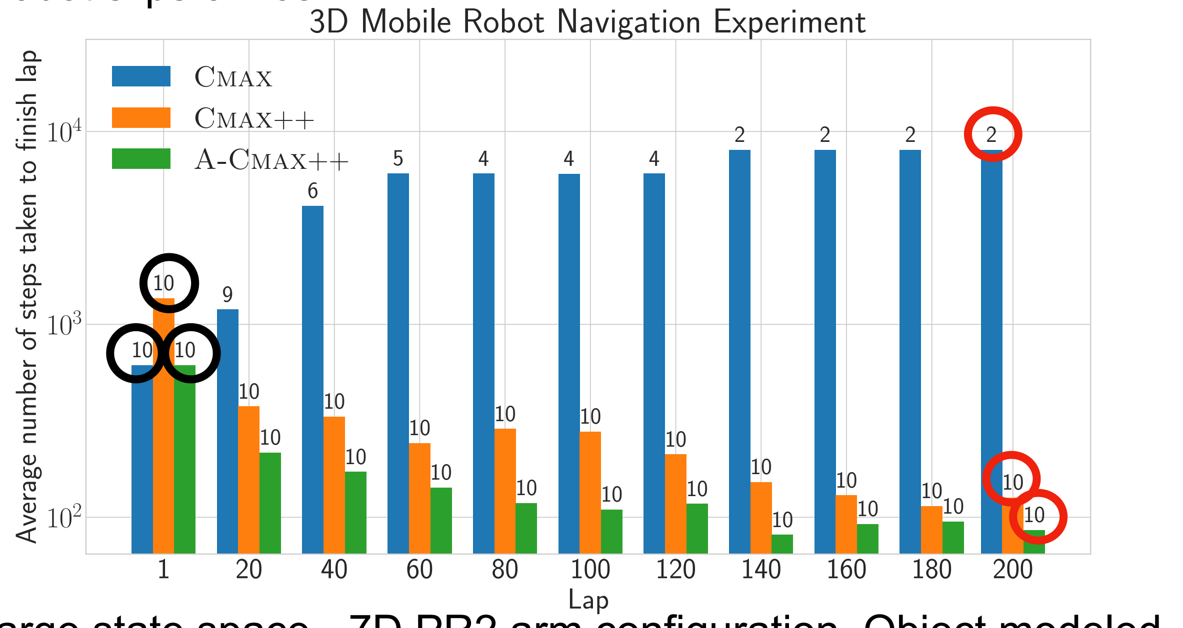
- CMAX++ wastes executions estimating Q-values and *lacks goal-driven behavior of CMAX* - typical of model-free methods
- **Key Idea:** *Adaptive-CMAX++ switches between CMAX and CMAX++ during execution to combine advantages of both*
- If value estimate following CMAX is not far from CMAX++, prefer CMAX - goal driven. Else, prefer CMAX++ - optimal
- Anytime-like: Goal-driven in early repetitions, Optimal in later repetitions
- Executions required to estimate Q-values spread across repetitions



- Given $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_N \geq 1$ where N is number of repetitions. At time step t in repetition i
  - If $V_{CMAX}(s_t) \leq \alpha_i V_{CMAX++}(s_t)$ - Execute CMAX action
  - Else - Execute CMAX++ action

Experiments

- Small state space - 3D (x, y, θ). Model has no icy patches and robot slips on ice



- Large state space - 7D PR2 arm configuration. Object modeled as light, arm can lift object only in certain configurations

Repetition→	1		5		20	
	Steps	*Success*	*Steps*	*Success*	*Steps*	*Success*
CMAX	**17.8 ± 3.4**	100%	13.6 ± 0.5	60%	15 ± 0	20%
CMAX++	**17 ± 4.9**	100%	14.2 ± 3.3	100%	**10.8 ± 0.1**	100%
A-CMAX++	**17.8 ± 3.4**	100%	**11.6 ± 0.7**	100%	**10.6 ± 0.4**	100%
Model KNN	40.6 ± 7.3	100%	12.8 ± 1.3	100%	12.4 ± 1.4	100%
Model NN	56 ± 16.2	100%	208.2 ± 92.1	80%	37.5 ± 20.1	40%
Q-learning	172.4 ± 75	100%	23.2 ± 10.3	80%	10.2 ± 0.6	80%

Advantages and Limitations

- Exploit inaccurately modeled transitions without learning true dynamics
- Useful in domains where modeling true dynamics is intractable
- Requires weaker assumptions when compared to CMAX
- Designing optimistic initial model requires domain knowledge
- Infeasible to relax assumption without resorting to undirected exploration methods

[1] Vemula, A.; Oza, Y.; Bagnell, J.; and Likhachev, M. 2020. Planning and Execution using Inaccurate Models with Provable Guarantees. In Proceedings of Robotics: Science and Systems. Corvalis, Oregon, USA. doi:10.15607/RSS.2020. XVI.001.

[2] Lee, M. A.; Florensa, C.; Tremblay, J.; Ratliff, N. D.; Garg, A.; Ramos, F.; and Fox, D. 2020. Guided Uncertainty-Aware Policy Optimization: Combining Learning and ModelBased Strategies for Sample-Efficient Policy Learning. CoRR abs/2005.10872. URL https://arxiv.org/abs/2005. 10872

[3] Lagrassa, A.; Lee, S.; and Kroemer, O. 2020. Learning skills to patch plans based on inaccurate models. In 2020 IEEE International Conference on Intelligent Robots and Systems (IROS).

Blog post: https://vvanirudh.github.io/blog/cmaxpp

Code: https://github.com/vvanirudh/cmaxpp