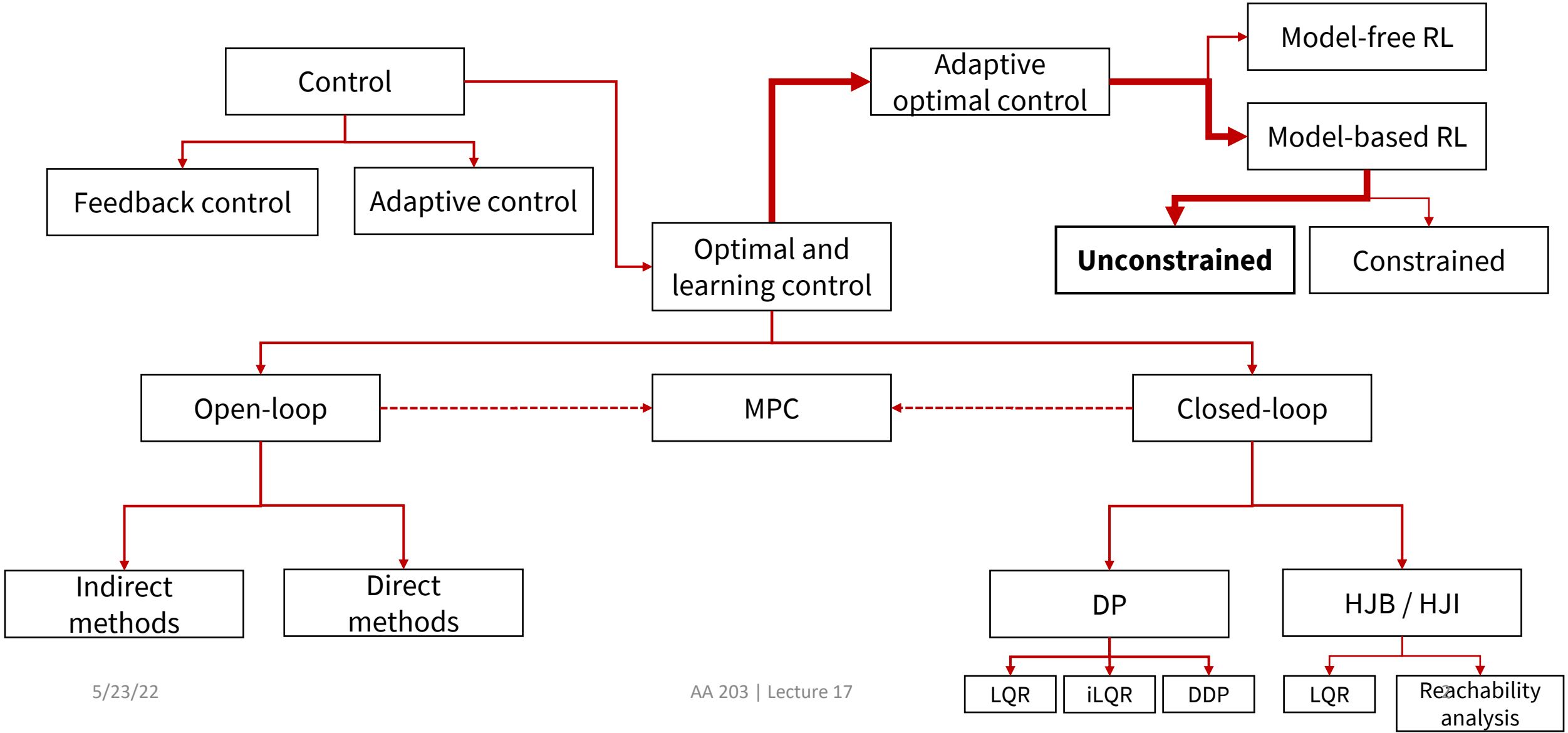
AA203
Optimal and Learning-based Control

Model-based reinforcement learning

Stanford University

ASL
Autonomous Systems Lab

Roadmap


Control
Feedback control
Adaptive control
Optimal and learning control
Adaptive optimal control
Model-free RL
Model-based RL
Unconstrained
Constrained
Open-loop
MPC
Closed-loop
Indirect methods
Direct methods
DP
HJB / HJI
LQR
iLQR
DDP
LQR
Reachability analysis

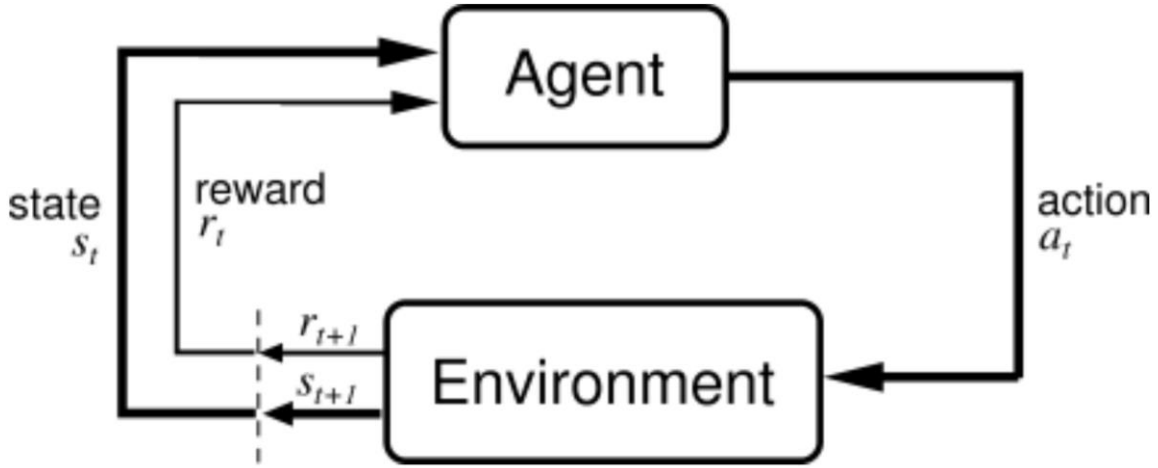
Agenda

- Reviewing the reinforcement learning problem statement
- Tabular model-based RL
- Continuous model-based RL

- Readings:
  - M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, *Bayesian Reinforcement Learning: A Survey,* Foundations and Trends in ML, 2016.
  - R. Sutton and A. Barto. *Reinforcement Learning: An Introduction,* 2018.
  - M. Kochenderfer. *Decision Making Under Uncertainty,* 2015.
  - T. M. Moerland, J. Broekens, C. M. Jonker. *Model-based Reinforcement Learning: A Survey*, 2020
  - K. Chua, R. Calandra, R. McAllister, and S. Levine. *Deep reinforcement learning in a handful of trials using probabilistic dynamics models,* NeurIPS, 2018.

Reinforcement learning assumptions and information patterns

- Previously:
  - Intro model-free RL: Q-learning, SARSA
    - Multiple episodes, interleave data collection and policy improvement
    - Tabular (discrete state space, discrete action space)
  - System Identification
    - Batch, offline data collection
    - Primarily linear dynamics
  - Adaptive Control
    - Intra-episode/online adaptation
    - Primarily linear dynamics



How are these different?

- Short answer: different historical developments, thus different standard assumptions (e.g., “episodes” are the standard way to approach learning game-playing agents)
- Currently, these fields are increasingly overlapping, thus it’s increasingly important to clearly state problem setting/assumptions, so that you can
  - connect similar ideas in different fields
  - know what works, and when
  - know what you should use for your problem

Core assumptions

- Linear vs. nonlinear dynamics
- Known vs. unknown cost function
- Episodic interaction vs. single episode (online) vs. batch offline data
  - Related: which policy was used to collect data (Offline vs online)?

Breaking down assumptions

	System Identification	Adaptive Control	Model-based RL	Model-free RL
Dynamics	Usually linear	Linear or nonlinear, usually control affine	Discrete or nonlinear continuous	Discrete or nonlinear continuous
Reward knowledge?	N/A	Designed (thus known)	Typically assumed known (not always)	Typically assumed unknown, provided by environment
Data collection/ episodic structure	Dataset provided	One episode	Typically, repeated episodes	Typically, repeated episodes
What do we learn?	Dynamics model	Usually policy (MRAC) or model (MIAC)	Dynamics model, sometimes reward model, sometimes policy	Policy (or Q function)

Caveat: there are exceptions to **all** of the above.

Unconstrained stochastic control problem

$$J_0^*(\mathbf{x}_0) = \min_{\boldsymbol{\pi}_0,\ldots,\boldsymbol{\pi}_{T-1}} \mathrm{E}[\, p(\mathbf{x}_T) + \sum_{k=0}^{T-1} c(\mathbf{x}_k, \boldsymbol{\pi}_k(\mathbf{x}_k)) \,]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \boldsymbol{\pi}_k(\mathbf{x}_k), \mathbf{w}_k, \boldsymbol{\theta}) \quad k = 0, \ldots, N-1$$

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$$

$$\mathbf{w}_k \sim p(\mathbf{w}) \text{ iid}, k = 0, \ldots, N-1$$

Notes:

- Possibly instantiated over multiple episodes
- Soft constraints may be encoded in *c*

Generalization and exploration

- Linear time-invariant (LTI) dynamics: if dataset generated with sufficient excitation, gives **global** knowledge
- Nonlinear dynamics: extrapolation is difficult and can be misleading
  - As AC/RL moves to more complex systems, have to consider uncertainty, exploration, and data collection process

Tabular model-based RL

- Discrete state/action space with stochastic transitions
- If model is known, can use value iteration/policy iteration/etc.
- Model unknown: want to build approximate model from observed transitions

Tabular MBRL outline

- Assume initial policy
- Loop forever (i.e., until loop end of episode, then loop over episodes):
  - Take some number of actions, resulting in transition/reward data
  - Improve dynamics model
  - Choose actions/policy

- Approaches for action selection:
  - Dynamic programming/VI/DP on approximate model
    - Expensive, gives optimal policy for model
  - Plan suboptimal sequence of actions via online control optimization

Dynamic programming for action selection

- Given an updated model, can perform value iteration/DP to yield new policy. Gives a global solution but…
  - Can be very expensive for large MDPs!
  - Effect of local model changes (often) has minor impact on far away states.

Local methods for action selection

- Tree search methods:
  - Similar idea to MPC: continuously generate short plans to approximate closed-loop policy.
  - For example, Monte Carlo tree search (MCTS); in its simplest form:
    - Sample random action sequences
    - Choose best sequence and execute first action

Combining local and global methods

- Many ways to combine local search (e.g., MCTS) with global/dynamic programming methods
  - Can use (possibly old) running value estimate as tail value in search
  - Forward search gives a TD update for value

Learning a tabular model from data

- States $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$
- Actions $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m)$
- Want to learn $p(\mathbf{x}_i | \mathbf{x}_j, \mathbf{u}_k)$ for all i, j, k

- We will discuss both max likelihood point estimation and fully Bayesian approaches

Max likelihood for tabular MBRL

- Categorical likelihood: $p\left(\mathbf{x}_i \middle| \mathbf{x}_j, \mathbf{u}_k, \boldsymbol{\theta}\right) = \boldsymbol{\theta}_{ijk}; \sum_i \boldsymbol{\theta}_{ijk} = 1$
- Assume data $D = \{(\mathbf{x}, \mathbf{u}, \mathbf{x}')\}_{i=1}^{d}$
- Max likelihood:

$$\max_{\theta \in \Theta} \sum_{D} \log p(\mathbf{x}' | \mathbf{x}, \mathbf{u}, \boldsymbol{\theta})$$

- Optimizing this gives the maximum likelihood estimate

$$\widehat{\boldsymbol{\theta}}_{ijk} = \frac{N\left(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i\right)}{N\left(\mathbf{x}_j, \mathbf{u}_k\right)}$$

  where $N(\cdot,\cdot)$ is the empirical count

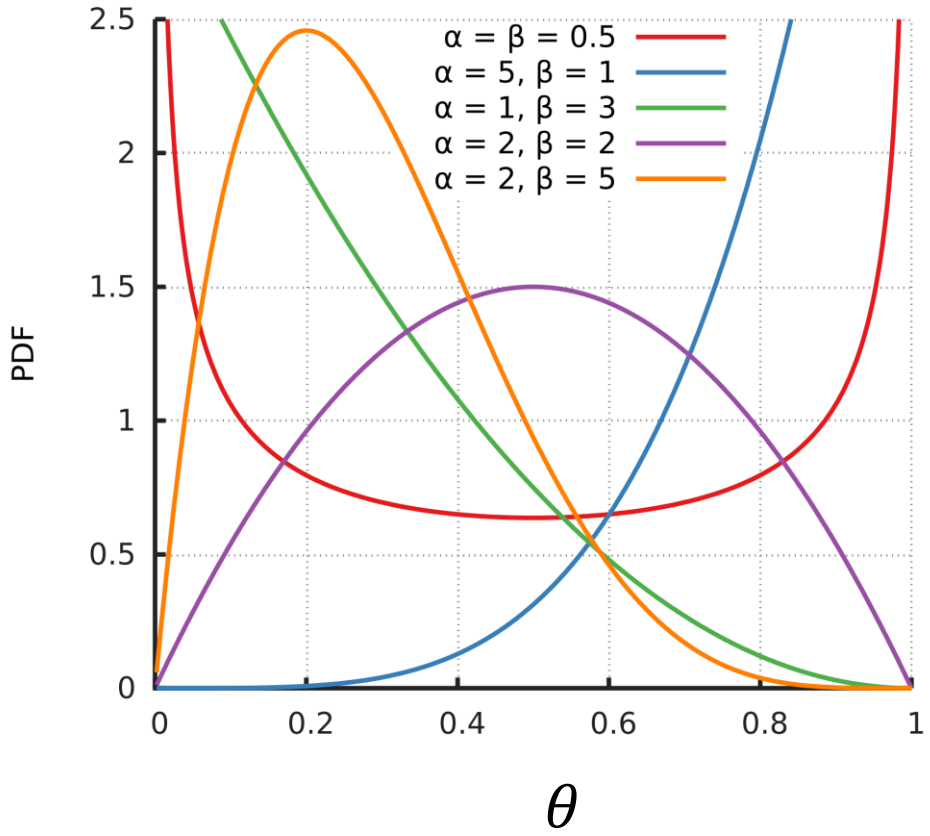
Max likelihood for tabular MBRL

- $\boldsymbol{\theta}_{ijk} = N(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i)/N(\mathbf{x}_j, \mathbf{u}_k)$
- Problem: what if $N(\mathbf{x}_j, \mathbf{u}_k) = 0$?
  - For example, if we are starting with zero information, this model estimation scheme breaks
- Simple solution: start all of our counts at 1, i.e.,
  - Store $N(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i)$; note that $N(\mathbf{x}_j, \mathbf{u}_k) = \sum_{\mathbf{x}_i} N(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i)$
  - Replace $N(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i)$ with $N(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i) + 1$
  - Gives $\boldsymbol{\theta}_{ijk} = (N(\mathbf{x}_j, \mathbf{u}_k, \mathbf{x}_i) + 1)/(N(\mathbf{x}_j, \mathbf{u}_k) + n)$

- We will see: corresponds to weak prior over transition probability mass function (pmf)

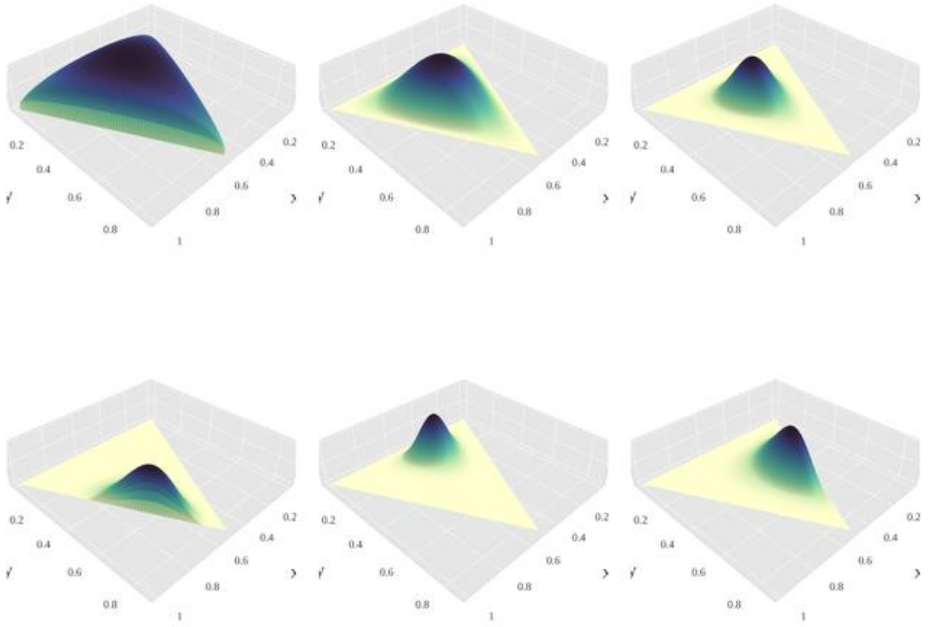
Bayesian inference for categorical distribution

- Bayesian inference:
  - Instead of a single "point estimate" $\widehat{\boldsymbol{\theta}}$, we want to reason about a distribution $p(\boldsymbol{\theta}|D)$
  - Computable using Bayes rule given prior $p(\boldsymbol{\theta})$ and observation model $p(D|\boldsymbol{\theta})$
- Example:
  - Beta distribution: interpretable as a "probability distribution on probabilities", conjugate prior of Bernoulli (coin toss)
  - $p(\theta) = \text{Beta}(\alpha, \beta)$
  - $p(D|\theta) = \text{Bernoulli}(\theta)$
  - ➔ $p(\theta|D) = \text{Beta}\left(\alpha + N_{\{D=1\}}, \beta + N_{\{D=0\}}\right)$
  - Parameters α, β interpretable as "pseudocounts"



Bayesian inference of transition probabilities

- Fix *Dirichlet distribution* prior
  - Corresponds to a probability distribution *over* discrete probability distributions
  - Write $\mathrm{Dir}(\boldsymbol{\alpha})$ with pdf $p(\mathrm{x}_1, \ldots, \mathrm{x}_n | \alpha_1, \ldots, \alpha_n)$
  - $\mathrm{E}[\mathrm{x}_i] = \alpha_i / \sum_j \alpha_j$
- Dirichlet is *conjugate* with categorical distribution:
  - Dirichlet prior with parameters $\alpha_1, \ldots, \alpha_n$, plus Categorical distribution gives Dirichlet posterior $\mathrm{Dir}(\boldsymbol{\alpha} + \boldsymbol{c})$
  - $\boldsymbol{c} = (c_1, \ldots, c_n)$ is counts of data



For details on derivation of posterior, see: *The Dirichlet-Multinomial and Dirichlet-Categorical models for Bayesian inference,* Stephen Tu (available online).

Bayesian posterior

- Prior parameter $\boldsymbol{\alpha}$ corresponds to number of prior observations
- For $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \sim Dir(\alpha)$, $\mathrm{E}[\mathbf{x}_i] = \alpha_i / \sum_j \alpha_j$
- Posterior predictive is $\mathrm{p}(\mathbf{x}'|\mathbf{x}, \mathbf{u}, \boldsymbol{\alpha}, D) = \frac{N(\mathbf{x},\mathbf{u},\mathbf{x}')+\alpha_{\mathbf{x},\mathbf{u},\mathbf{x}'}}{\sum_{\mathbf{x}'} N(\mathbf{x},\mathbf{u},\mathbf{x}')+\alpha_{\mathbf{x},\mathbf{u},\mathbf{x}'}}$
- Choosing $\boldsymbol{\alpha} = (1, \ldots, 1)$ gives our previous correction

- But, we have more than just the point estimate of our model. How can we use this?

Bayes-adaptive MDPs

- Have model parameters $\boldsymbol{\theta}$ (i.e., $\boldsymbol{\alpha}$), which summarizes "counts" for Dirichlet posterior for every state/action/next state combination.
- Recalling our brief discussion of dual control, we can consider a *Bayes-adaptive* MDP, with hyperstate $(\mathbf{x}, \boldsymbol{\theta})$
  - Transition dynamics $p(\mathbf{x}', \boldsymbol{\theta}' | \mathbf{x}, \mathbf{u}, \boldsymbol{\theta})$ can be factored as $\mathrm{p}(\mathbf{x}' | \mathbf{x}, \mathbf{u}, \boldsymbol{\theta})\ \mathrm{p}(\boldsymbol{\theta}' | \mathbf{x}', \mathbf{x}, \mathbf{u}, \boldsymbol{\theta})$
    - $\mathrm{p}(\mathbf{x}' | \mathbf{x}, \mathbf{u}, \boldsymbol{\theta}) = \frac{\theta_{\mathbf{x},\mathbf{u},\mathbf{x}'}}{\sum_{\mathbf{x}'} \theta_{\mathbf{x},\mathbf{u},\mathbf{x}'}} = \frac{N(\mathbf{x},\mathbf{u},\mathbf{x}') + \theta^{\text{prior}}_{\mathbf{x},\mathbf{u},\mathbf{x}'}}{\sum_{\mathbf{x}'} N(\mathbf{x},\mathbf{u},\mathbf{x}') + \theta^{\text{prior}}_{\mathbf{x},\mathbf{u},\mathbf{x}'}}$
    - Model parameter count increases by 1 for corresponding transition count

- Problem: state space grows infinitely, so cannot do dynamic programming
  - Approximate DP/local search is possible; a good reference on review of Bayes-adaptive RL and approximate methods: Michael Duff's PhD thesis, 2003.

Exploration heuristics: Thompson sampling

- Even in tabular and linear MDPs, dual control/Bayes-adaptive MDP (solving which would yield optimal exploration) is intractable
- So we turn to heuristics to explore, e.g., as previous mentioned
  - In the context of system identification: noise addition (persistent excitation)
  - In the context of RL: epsilon-greedy exploration
- Simple approach using posterior over models: Thompson sampling
  - Sample MDP from posterior
  - Act optimally w.r.t. this MDP for episode (allows for “deep exploration”)
  - Update model posterior and loop

Continuous MBRL

- Will now consider general non-LTI continuous models
- Many possible model choices:
  - Nonlinear features in linear regression
  - Time varying linear dynamics
  - Gaussian processes
  - Neural networks
- Many possible control choices:
  - MPC (often without persistent feasibility/stability guarantees)
    - Repeated direct methods/trajectory optimization (e.g., iLQR)
    - Many variants: gradient-based vs. sampling, with/without final cost
  - Directly optimize policy (next week)

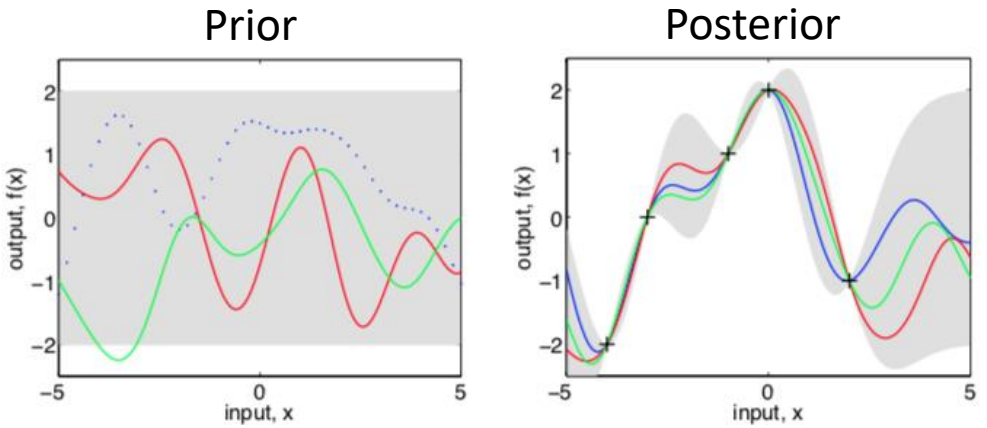
Abbeel et al., NeurIPS 2008

Gaussian process models

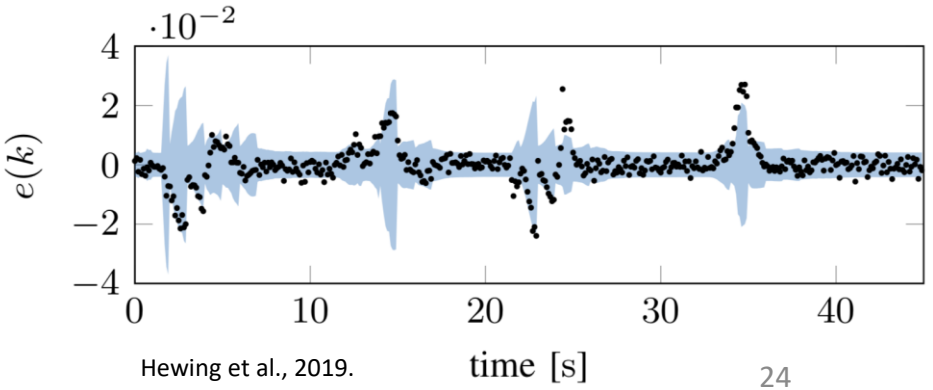
- Place prior over dynamics, $f \sim GP\big(m(\cdot), \mathrm{k}(\cdot,\cdot)\big)$
  - Corresponds to infinite dimensional gaussian distribution, prior over functions

- Strengths
  - Data efficient
  - Exact posterior
  - Predictable behavior via kernel choice

- Weaknesses
  - High computational complexity
  - Assume Gaussian measurement error
  - Can not learn expressive features



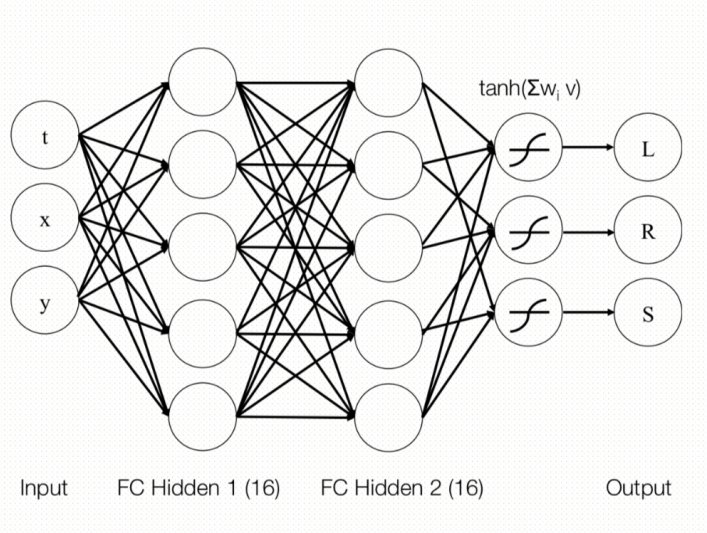
Rasmussen and Williams, 2006.

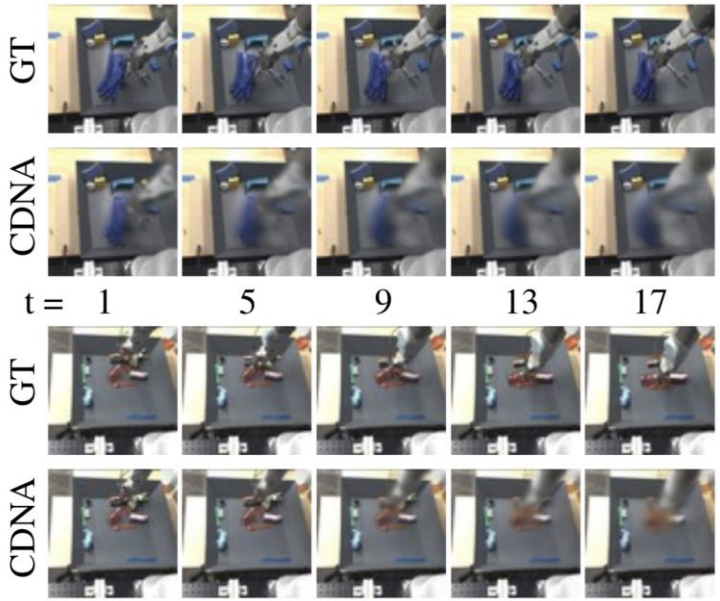


Hewing et al., 2019.

Neural network models

- Parameterize model using neural network
- Strengths
  - Can learn complex, expressive features
  - Can be paired with arbitrary loss functions
- Weaknesses
  - Data inefficient
  - Difficult to represent uncertainty
  - Unpredictable extrapolation





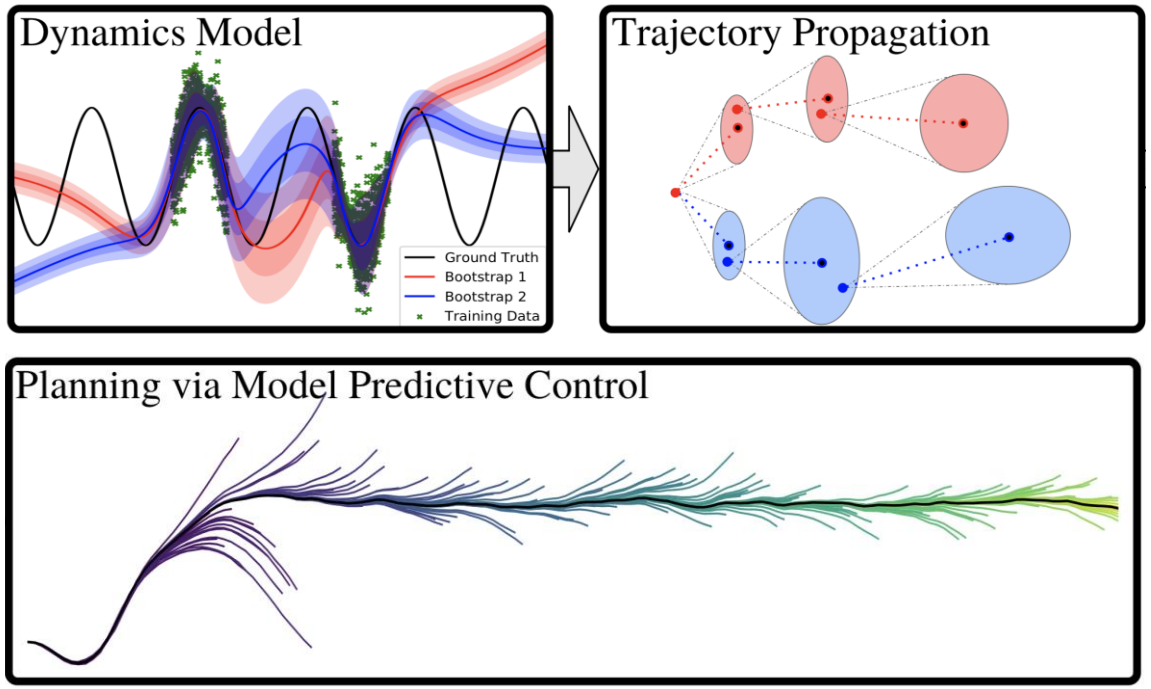
Finn et al., 2017.

Case study: PETS

- Probabilistic Ensembles with Trajectory Sampling
- Key idea:
  - Use *ensemble* (collection) of NNs to approximate posterior over model
  - Incorporate model uncertainty into control
- Ensembling:
  - Initialize several networks with different weights
  - Will agree where there is a lot of data, disagree elsewhere

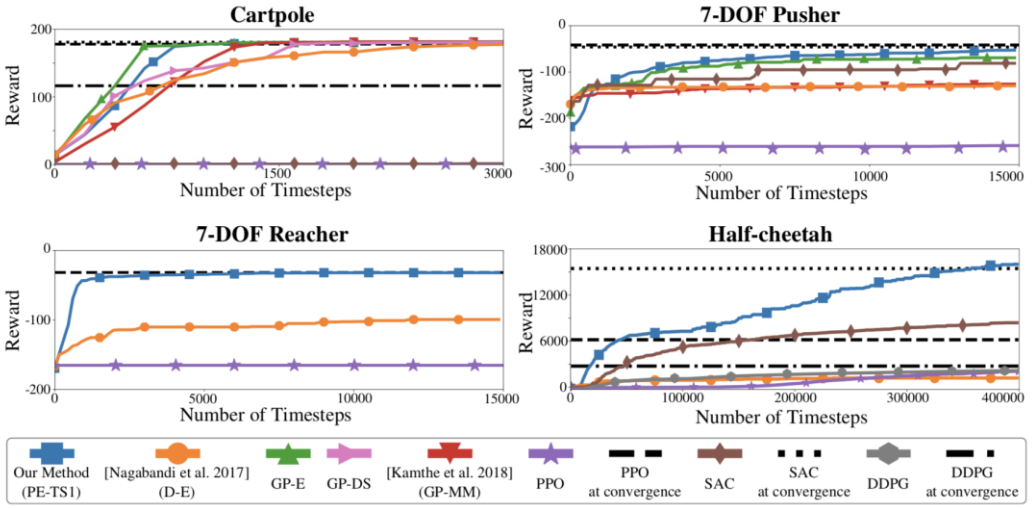
Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models

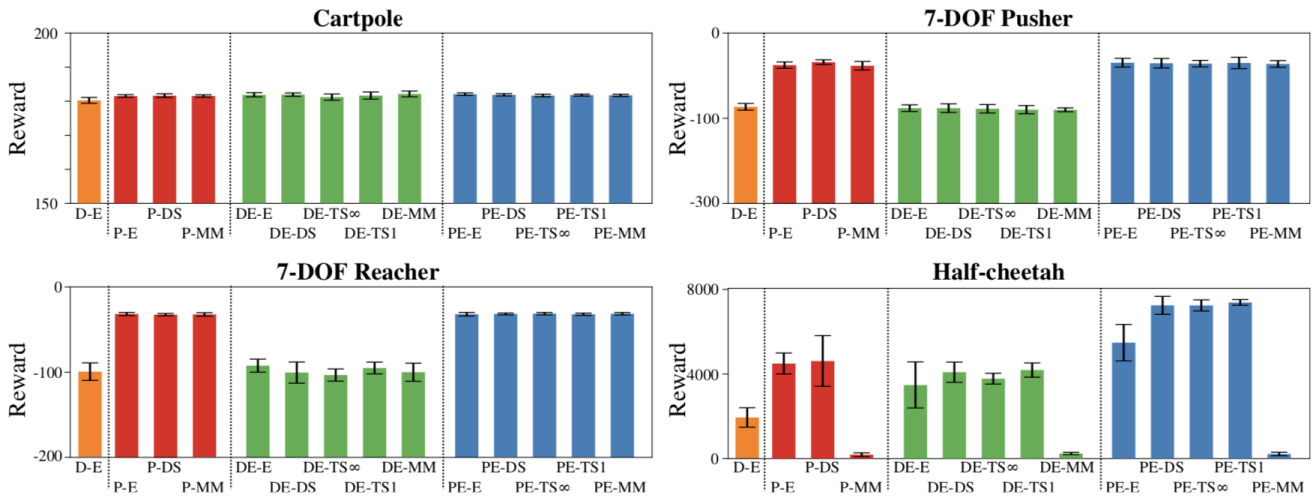
Kurtland Chua **Roberto Calandra** **Rowan McAllister** **Sergey Levine**
Berkeley Artificial Intelligence Research
University of California, Berkeley
{kchua, roberto.calandra, rmcallister, svlevine}@berkeley.edu



PETS findings

- Consider both probabilistic network (outputs mean + variance) and deterministic
- Use particle-based MPC controller (random action sampling)
- Either re-sample dynamics at each time, or keep fixed





Why model-based?

- Advantages
  - Transitions give strong signal
  - Data efficiency, improved multi-task performance, generalization

- Weaknesses
  - Optimizing the wrong objective (i.e., not your ultimate task of optimizing reward)
  - May be very difficult/intractable for systems with high dimensional observations/states

Next time

- Model-free RL: policy gradient, variance reduction, actor-critic.