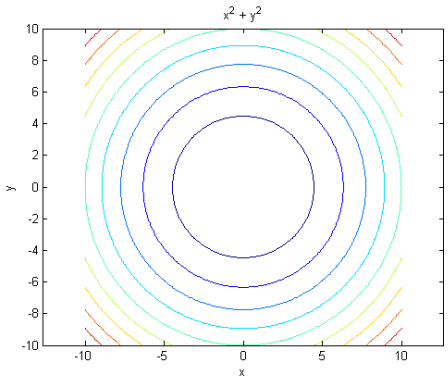
CS 287: Advanced Robotics
Fall 2009

Lecture 20:
Natural gradient
Reward shaping
Approximate LP with function approximation
POMDP
Hierarchical RL

Pieter Abbeel
UC Berkeley EECS

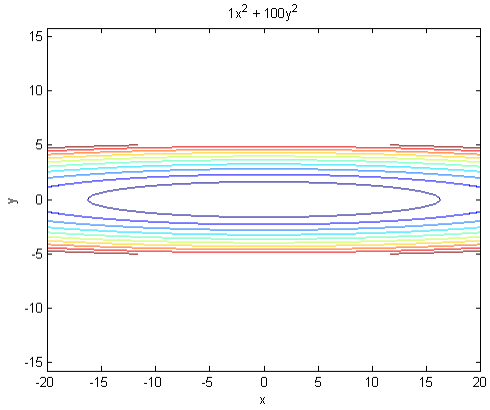
Natural gradient

- Is the gradient the correct direction?



Natural gradient

- Is the gradient the correct direction?



Gradient and linear transformations

Consider the optimization of the function $f(\theta)$ through gradient descent. In iteration k we would perform an update of the following form:

$$\theta^{(k+1)} = \theta^{(k)} + \alpha \nabla_\theta f(\theta^{(k)}). \tag{1}$$

Consider a new coordinate system $x = A^{-1}\theta$. We could work in the new coordinate system instead, and optimize $f(Ax)$ over the variable x. A gradient descent step is given by:

$$x^{(k+1)} = x^{(k)} + \alpha \nabla_x f(Ax^{(k)}) = x^{(k)} + \alpha A^\top \nabla_\theta f(Ax^{(k)}) \tag{2}$$

If $x^{(k)} = A^{-1}\theta^{(k)}$, do we have $x^{(k+1)} = \theta^{(k+1)}$?
No!
The value in θ coordinates that corresponds to x_{k+1} is given by

$$Ax^{(k+1)} = Ax^{(k)} + \alpha AA^\top \nabla_\theta f(Ax^{(k)}) = \theta^{(k)} + \alpha AA^\top \nabla_\theta f(\theta^{(k)}) \neq \theta^{(k+1)}$$

Newton's direction

Newton's method approximates the function $f(\theta)$ by a quadratic function through a Taylor expansion around the current point θ_k:

$$f(\theta) \approx f(\theta_k) + \nabla_\theta f(\theta^{(k)})^\top (\theta - \theta^{(k)}) + \frac{1}{2}(\theta - \theta^{(k)})^\top H(\theta^{(k)})(\theta - \theta^{(k)})$$

Here $H_{ij}(\theta^{(k)}) = \frac{\partial^2 f}{\partial \theta_i \theta_j}(\theta^{(k)})$ is a matrix with the 2nd derivatives of f evaluated at $\theta^{(k)}$.

The local optimum of the 2nd order approximation is found by setting its gradient equal to zero, which gives:

$$\text{Newton step direction} = (\theta - \theta^{(k)}) = -H^{-1}(\theta^{(k)})\nabla_\theta f(\theta^{(k)})$$

The Newton step direction is affine invariant

Newton's step direction for $f(\theta)$ is given by:

$$H^{-1}(\theta^{(k)})\nabla_\theta f(\theta^{(k)}). \tag{1}$$

For $f(Ax)$, with $x = A^{-1}\theta$, we have

$$\begin{aligned} \text{Hessian} &= A^\top H A \\ \text{gradient} &= A^\top \nabla_\theta f \end{aligned}$$

Hence we have for the Newton step direction in the x coordinates:

$$\left(A^\top H A\right)^{-1} A^\top \nabla_\theta f(Ax^{(k)}) = A^{-1}\nabla_\theta f(Ax^{(k)}) \tag{2}$$

Translating this into θ coordinates gives us $AA^{-1}\nabla_\theta f(Ax^{(k)}) = \nabla_\theta f(Ax^{(k)})$, which is identical to the step direction directly computed in θ coordinates.

Natural gradient

- Gradient depends on choice of coordinate system.
- Newton's method is invariant to affine coordinate transformations, but not to general coordinate transformations.
- Can we achieve more invariance than simply affine invariance?

Natural gradient

- Let's first re-interpret the gradient:
  - The gradient is the direction of steepest ascent:

Natural gradient

- Let's first re-interpret the gradient:
  - The gradient is the direction of steepest ascent:

    For small ϵ we have:

$$\begin{aligned}\arg\max_{\delta\theta:\|\delta\theta\|_2\leq\epsilon} f(\theta+\delta\theta) &\approx \arg\max_{\delta\theta:\|\delta\theta\|_2\leq\epsilon} f(\theta)+\nabla_\theta f(\theta)^\top\delta\theta \\ &= \arg\max_{\delta\theta:\|\delta\theta\|_2\leq\epsilon} \nabla_\theta f(\theta)^\top\delta\theta \\ &= \frac{\nabla_\theta f(\theta)}{\|\nabla_\theta f(\theta)\|_2}\epsilon\end{aligned}$$

  - When expressing our problem in a different coordinate system f remains the same, but the || . || <= 1 constraint means something different in different coordinate systems.
- → Can we find a norm constraint that is independent of the coordinate system????

A distance which is independent of the parameterization of the policy class

- **Kullback-Leibler divergence** between distributions over paths induced by the policies:

$$KL(P(\tau;\theta_1)||P(\tau;\theta_2)) = \sum_\tau P(\tau;\theta_1)\log\frac{P(\tau;\theta_1)}{P(\tau;\theta_2)}$$

- E.g., 2 Bernoulli distributions:
  - Prob(heads)=p and Prob(heads)=q

$$\mathrm{KL}(P(\cdot;p)||P(\cdot;q)) = p\log\frac{p}{q}+(1-p)\log\frac{1-p}{1-q}$$

  - Alternative parameterization: $\mathrm{Prob(heads)}=\frac{\exp(\theta)}{1+\exp(\theta)}$ and $\mathrm{Prob(heads)}=\frac{\exp(\psi)}{1+\exp(\psi)}$

$$\begin{aligned}\mathrm{KL}(P(\cdot;\theta)\|P(\cdot;\psi)) &= \frac{\exp(\theta)}{1+\exp(\theta)}\log\frac{\frac{\exp(\theta)}{1+\exp(\theta)}}{\frac{\exp(\psi)}{1+\exp(\psi)}}+\frac{1}{1+\exp(\theta)}\log\frac{\frac{1}{1+\exp(\theta)}}{\frac{1}{1+\exp(\psi)}} \\ &= \mathrm{KL}(P(\cdot;p)\|P(\cdot;q)) \quad \text{if } p=\frac{\exp(\theta)}{1+\exp(\theta)}, q=\frac{\exp(\psi)}{1+\exp(\psi)}\end{aligned}$$

A distance which is independent of the parameterization of the policy class

- **Kullback-Leibler divergence** between distributions over paths induced by the policies:

$$KL(P(\tau;\theta_1)||P(\tau;\theta_2)) = \sum_{\tau} P(\tau;\theta_1) \log \frac{P(\tau;\theta_1)}{P(\tau;\theta_2)}$$

- Second-order Taylor approximation

$$\begin{aligned} KL(P(\tau;\theta)||P(\tau;\theta+\delta\theta)) &\approx \sum_{\tau} P(\tau;\theta)\delta\theta^\top \nabla_\theta \log P(\tau;\theta) \nabla_\theta \log P(\tau;\theta)^\top \delta\theta \\ &= \delta\theta^\top \left(\sum_{\tau} P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) \nabla_\theta \log P(\tau;\theta)^\top \right) \delta\theta \\ &= \delta\theta^\top G(\theta) \delta\theta \end{aligned}$$

- G(θ) = ***Fisher information matrix***, independent of the choice of parameterization of the class of distributions.

2nd order Taylor expansion of KL divergence

$$\begin{aligned}
& \mathrm{KL}(P(X;\theta)\|P(X;\theta+\delta\theta) \\
= & \sum_x P(x;\theta) \log \frac{P(x;\theta)}{P(x;\theta+\delta\theta)} \\
\approx & \sum_x P(x;\theta) \left(\log \frac{P(x;\theta)}{P(x;\theta)} - \frac{d}{d\theta} \log P(x;\theta)^\top \delta\theta - \frac{1}{2}\delta\theta^\top \frac{d^2}{d\theta^2} \log P(x;\theta) \delta\theta \right) \\
= & -\sum_x P(x;\theta) \frac{d}{d\theta} \log P(x;\theta)^\top \delta\theta - \frac{1}{2}\delta\theta^\top \frac{d^2}{d\theta^2} \log P(x;\theta) \delta\theta \\
= & -\sum_x P(x;\theta) \left(\frac{\frac{d}{d\theta}P(x;\theta)}{P(x;\theta)} \right)^\top \delta\theta - \frac{1}{2}\delta\theta \sum_x P(x;\theta) \frac{P(x;\theta)\frac{d^2}{d\theta^2}P(x;\theta) - \left(\frac{dP(x;\theta)}{d\theta}\right)\left(\frac{dP(x;\theta)}{d\theta}\right)^\top}{P(x;\theta)^2} \delta\theta \\
= & -\sum_x \frac{d}{d\theta} P(x;\theta)^\top \delta\theta - \frac{1}{2}\delta\theta \sum_x \frac{d^2}{d\theta^2} P(x;\theta) \delta\theta + \frac{1}{2}\delta\theta \sum_x P(x;\theta) \left(\frac{\frac{dP(x;\theta)}{d\theta}}{P(x;\theta)} \right) \left(\frac{\frac{dP(x;\theta)}{d\theta}}{P(x;\theta)} \right)^\top \delta\theta \\
= & -\left(\frac{d}{d\theta} \sum_x P(x;\theta) \right)^\top \delta\theta - \frac{1}{2}\delta\theta^\top \left(\frac{d^2}{d\theta^2} \sum_x P(x;\theta) \right) \delta\theta \\
& + \frac{1}{2}\delta\theta^\top \left(\sum_x P(x;\theta) \left(\frac{d}{d\theta} \log P(x;\theta) \right) \left(\frac{d}{d\theta} \log P(x;\theta) \right)^\top \right) \delta\theta \\
= & -\left(\frac{d}{d\theta} 1 \right)^\top \delta\theta - \frac{1}{2}\delta\theta^\top \left(\frac{d^2}{d\theta^2} 1 \right) \delta\theta \\
& + \frac{1}{2}\delta\theta^\top \left(\sum_x P(x;\theta) \left(\frac{d}{d\theta} \log P(x;\theta) \right) \left(\frac{d}{d\theta} \log P(x;\theta) \right)^\top \right) \delta\theta \\
= & -0 - 0 + \frac{1}{2}\delta\theta^\top \left(\sum_x P(x;\theta) \left(\frac{d}{d\theta} \log P(x;\theta) \right) \left(\frac{d}{d\theta} \log P(x;\theta) \right)^\top \right) \delta\theta \\
= & \frac{1}{2}\delta\theta^\top G(\theta) \delta\theta
\end{aligned}$$

Natural gradient g_N

Natural gradient: general setting

Natural gradient in policy search

Natural gradient g_N

- = the direction with highest increase in the objective per change in KL divergence

$$
\begin{aligned}
g_N &= \arg \max_{\delta\theta: KL(P(\tau;\theta)||P(\tau;\theta+\delta\theta)\leq\epsilon} f(\theta+\delta\theta) \\
&\approx \arg \max_{\delta\theta: \frac{1}{2}\delta\theta^\top G(\theta)\delta\theta\leq\epsilon} f(\theta) + \nabla_\theta f(\theta)^\top \delta\theta \\
&= \arg \max_{\delta\theta: \frac{1}{2}\delta\theta^\top G(\theta)\delta\theta\leq\epsilon} \nabla_\theta f(\theta)^\top \delta\theta \\
&= G(\theta)^{-1}\nabla_\theta f(\theta)
\end{aligned}
$$

Natural gradient: general setting

Problem setting: optimize an objective which depends on a probability distribution P_θ

$$\max_\theta f(P_\theta)$$

Rather than following the gradient, which depends on the choice of parameterization for the set of probability distributions that we are searching over, follow the natural gradient g_N:

$$g_N = G(\theta)^{-1} \nabla_\theta f(P_\theta)$$

Here $G(\theta)$ is the Fisher information matrix, and can be computed as follows:

$$G(\theta) = \sum_{x \in X} P_\theta(x) \nabla_\theta \log P_\theta(x) \nabla_\theta \log P_\theta(x)^\top$$

Natural gradient in policy search

Objective:

$$U(\theta) = \sum_\tau P(\tau;\theta) R(\tau)$$

Natural gradient g_N:

$$g_N = G(\theta)^{-1} \nabla_\theta U(\theta)$$

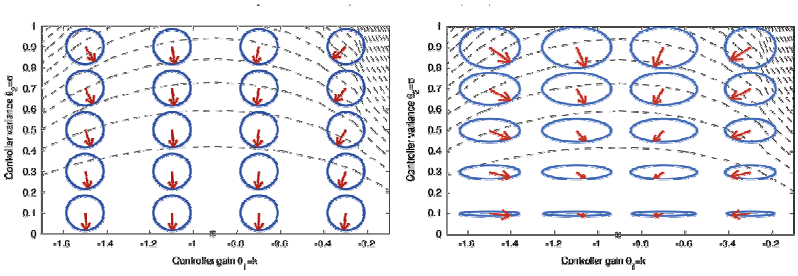
Both the Fisher information matrix G and the gradient need to be estimate from samples. We have seen many ways to estimate the gradient from samples. Remains to show how to estimate G.

$$\begin{aligned} G(\theta) &= \sum_\tau P(\tau) \nabla_\theta \log P(\tau;\theta) \nabla_\theta \log P(\tau;\theta)^\top \\ &\approx \frac{1}{m} \sum_{i=1}^m \nabla_\theta \log P(\tau^{(i)};\theta) \nabla_\theta \log P(\tau^{(i)};\theta)^\top \end{aligned}$$

As we have seen earlier, we can compute the expression $\nabla_\theta \log P(\tau^{(i)})$ even without access to the dynamics model:

$$\begin{aligned} \nabla_\theta \log P(\tau^{(i)};\theta) &= \nabla_\theta \sum_{t=0}^{H-1} \log T(s_t, u_t, s_{t+1}) + \log \pi_\theta(u_t|s_t) \\ &= \nabla_\theta \sum_{t=0}^{H-1} \log \pi_\theta(u_t|s_t) \end{aligned}$$

Example



(a) Vanilla policy gradient (b) Natural policy gradient.

Kober and Peters, NIPS 2009

Announcements

- Project milestone due tomorrow 23:59pm
  - = 1 page progress update.
  - Format: pdf
- Assignment #2: out tomorrow night, due in 2 weeks
  - Topic: RL
  - Start early!
- Late day policy: 7 days total; -20pts (out of 100pts of the thing you are submitting late) per day beyond that
- Assignment #3 will be released in 2 weeks and will be very small compared to #1 and #2.

Thomas Daniel, University of Washington

A Tale of Two (or Three?) Gyroscopes: Inertial measurement units (IMUs) in flying insects

Date: Thursday, November 5, 2009
Time: 4:00 PM
Place: 2040 Valley Life Sciences Building

Animals use a combination of sensory modalities to control their movement including visual, mechanosensory and chemosensory information. Mechanosensory systems that can detect inertial forces are capable of responding much more rapidly than visual systems and, as such, are thought to play a critical role in rapid course correction during flight. This talk focusses on two gryoscopic organs: halteres of flies and antennae of moths. Both have mechanical and neural components play critical roles in encoding relatively tiny Coriolis forces associated with body rotations, both of which will be reviewed along with new data that suggests each have complex circuits that connect visual systems to mechanosensory systems. But, insects are bristling with mechanosensory structures, including the wings themselves. It is not clear whether these too could serve an IMU function in addition to their obvious aerodynamic roles.

"MODULARITY, POLYRHYTHMS, AND WHAT ROBOTICS AND CONTROL MAY YET LEARN FROM THE BRAIN"

Jean-Jacques Slotine, Nonlinear Systems Laboratory, MIT

Thursday, Nov 5th, 4:00 p.m., 3110 Etcheverry Hall

ABSTRACT

Although neurons as computational elements are 7 orders of magnitude slower than their artificial counterparts, the primate brain grossly outperforms robotic algorithms in all but the most structured tasks. Parallelism alone is a poor explanation, and much recent functional modelling of the central nervous system focuses on its modular, heavily feedback-based computational architecture, the result of accumulation of subsystems throughout evolution. We discuss this architecture from a global functionality point of view, and show why evolution is likely to favor certain types of aggregate stability. We then study synchronization as a model of computations at different scales in the brain, such as pattern matching, restoration, priming, temporal binding of sensory data, and mirror neuron response. We derive a simple condition for a general dynamical system to globally converge to a regime where diverse groups of fully synchronized elements coexist, and show accordingly how patterns can be transiently selected and controlled by a very small number of inputs or connections. We also quantify how synchronization mechanisms can protect general nonlinear systems from noise. Applications to some classical questions in robotics, control, and systems neuroscience are discussed.

The development makes extensive use of nonlinear contraction theory, a comparatively recent analysis tool whose main features will be briefly reviewed.

Reinforcement learning: remaining topics

- approximate LP, pomdp's, reward shaping, exploration vs. exploitation, hierarchical methods

Approximate LP

- Exact Bellman LP

$$\begin{aligned} \min_V \quad & \sum_s c(s)V(s) \\ \text{s.t.} \quad & V(s) \geq \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma V(s')\right) \quad \forall s, \forall a \end{aligned}$$

- Approximate LP

$$\begin{aligned} \min_\theta \quad & \sum_{s \in S'} c(s)\theta^\top \phi(s) \\ \text{s.t.} \quad & \theta^\top \phi(s) \geq \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma \theta^\top \phi(s')\right) \quad \forall s \in S', \forall a \end{aligned}$$

Approximate LP guarantees

- When retaining all constraints, yet introducing function approximation.

$$\begin{aligned} \min_\theta \quad & \sum_{s \in S} c(s)\theta^\top \phi(s) \\ \text{s.t.} \quad & \theta^\top \phi(s) \geq \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma \theta^\top \phi(s')\right) \quad \forall s \in S, \forall a \end{aligned}$$

Theorem. [**de Farias and Van Roy**] If one of the basis function satisfies $\phi_i(s) = 1$ for all $s \in S$, then the LP has a feasible solution and the optimal solution $\hat{\theta}$ satisfies:

$$\|V^* - \Phi\theta\|_{1,c} \leq \frac{2}{1-\alpha} \min_\theta \|V^* - \Phi\theta\|_\infty$$

Constraint sampling

Consider a convex optimization problem with a very large number of constraints:

$$\begin{aligned} \min \quad & c^\top x \\ s.t. \quad & g_i(x) \leq b \quad i = 1, 2, \ldots, m \end{aligned}$$

where $x \in \Re^n$, g_i convex, and $m >> n$.

We obtain the sampled approximation by sampling the sequence $\{i_1, i_2, \ldots, i_N\}$ IID according to some measure over the constraints μ. This gives us:

$$\begin{aligned} \min \quad & c^\top x \\ s.t. \quad & g_j(x) \leq b \quad j = i_1, i_2, \ldots, i_N \end{aligned}$$

Let $\hat{x}_N$ be the optimal solution to the sampled convex problem.

Theorem. [**Calafiore and Campi, 2005**] For arbitrary $\epsilon > 0, \delta > 0$, if $N \geq \frac{n}{\epsilon\delta} - 1$, then

$$\text{Prob}\left(\mu\left(\{i | g_i(\hat{x}_N) > b_i\}\right) \leq \epsilon\right) \geq 1 - \delta$$

where the probability is taken over the random sampling of constraints.

This result can be leveraged to show that the solution to the sampled approximate LP is close to V^* with high probablity. (de Farias and Van Roy, 2001)

Reward shaping

- What freedom do we have in specifying the reward function? Can we choose it such that learning is faster?
- Examples:
  - + Tetris: set the reward equal to the distance between highest filled square and the top of the board vs. a reward of 1 for placing a block
  - - Bicycle control task: provide a positive reward for motion towards the goal
  - - Soccer task: provide a positive reward for touching the ball

Potential based shaping

- Let $F(s,a,s') = \gamma \phi(s') - \phi(s)$
- Shaped reward = $R + F$

- **Theorem** [Ng, Harada & Russell, 1999]

  Potential based reward shaping is a necessary and sufficient condition to guarantee that the optimal policy in the shaped MDP $M' = (S, A, T, \gamma, R+F)$ is also an optimal policy in the original MDP $M = (S, A, T, \gamma, R)$

  [In fact even stronger: all policies retain their relative value.]

Intuition of proof

- In the new MDP, for a trace $s_0, a_0, s_1, a_1, \ldots$ we obtain:

$$R(s_0, a_0, s_1) + \gamma \phi(s_1) - \phi(s_0)$$
$$+ \gamma \left(R(s_1, a_1, s_2) + \gamma \phi(s_2) - \phi(s_1)\right)$$
$$+ \gamma^2 \left(R(s_2, a_2, s_3) + \gamma \phi(s_3) - \phi(s_2)\right)$$
$$+ \ldots$$

$$= -\phi(s_0) + R(s_0, a_0, s_1) + \gamma R(s_1, a_1, s_2) + \gamma^2 R(s_2, a_2, s_3) + \ldots$$

→ For any policy π we have: (M: original MDP, M': MDP w/shaped reward)

$$V^{\pi}_{M'}(s_0) = V^{\pi}_{M}(s_0) - \phi(s_0)$$

A good potential?

- Let ϕ = V*
- Then in one update we have:

$$\begin{aligned} V(s) &\leftarrow \max_a \sum_{s'} T(s,a,s') \left(R(s,a,s') + F(s,a,s') + \gamma V(s')\right) \\ &= \max_a \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma V^*(s') - V(s) + \gamma V(s')\right) \\ &= -V^*(s) + \max_a \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma V^*(s') + \gamma V(s')\right) \end{aligned}$$

- If we initialize V = 0, we obtain:

$$\begin{aligned} V(s) &\leftarrow -V^*(s) + \max_a \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma V^*(s')\right) \\ &= -V^*(s) + V^*(s) \\ &= 0 \end{aligned}$$

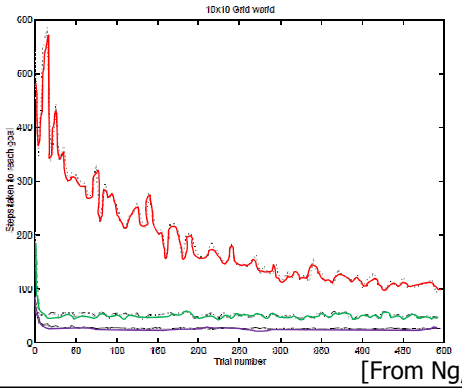
→ V=0 satisfies the Bellman equation; → this particular choice of potential function / reward shaping, we can find the solution to the shaped MDP very quickly

Example

10x10 grid world, 1 goal state = absorbing, other states R=-1;
Prob(action successful) = 80%

→ Shaping function: ϕ_0(s) = - (manhattan distance to goal) / 0.8

Plot shows performance of Sarsa(0) with epsilon=0.1 greedy, learning rate = .2
(a) no shaping vs. (b) ϕ = 0.5 * ϕ_0 vs. (c) $\phi = \phi_0$



[From Ng, Harada & Russell 1999]

Relationship to value function intialization

- Potential based shaping can be shown to be equivalent to initializing the value function to the shaping potential.

[see course website for the technical note describing this]

→ If restricting ourselves to potential based shaping, can implement it in 2 ways.

Partial observability

- = no direct observation of the state
- Instead: might have noisy measurements
- → Partially Observable Markov Decision Process (POMDP)

- Main ideas:
  - Based on the noisy measurements and knowledge about the dynamics, keep track of a probability distribution for the current state
  - Define a new MDP for which the probability distribution over current state is considered the state

Exploration vs. exploitation

- One of the milestone results: Kearns and Singh, Explicit Exploration and Exploitation (E^3), 2002
- Question/Problem:
  - Given an MDP with unknown transition model.
  - Can we
    - (a) explicitly decide to take actions that will assist us in building a transition model of the MDP, and
    - (b) detect when we have a sufficiently accurate model and start exploiting?

Basic idea of E^3 algorithm

- Repeat forever
  - Based on all data seen so far, partition the state space is a set of "known" states and a set of "unknown" states. [A state is known when each action in that state has been observed sufficiently often.]
  - If currently in a "known" state:
    - Lump all unknown states together in one absorbing meta-state. Give the meta-state a reward of 1. Give all known states a reward of zero. Find the optimal policy in this new MDP.
      - If the optimal policy has a value of zero (or low enough): exit. [No need for exploration anymore.]
      - Otherwise: Execute the optimal action for the current state.
  - If currently in a "unknown" state:
    - Take the action that has been taken least often in this state.

Technical aspects underneath E^3

- Simulation lemma: if the transition models and reward models of two MDPs are sufficiently close, then the optimal policy in one will also be close to optimal in the other
- After having seen a state-action pair sufficiently often, with high probability the data based transition model estimate will be accurate

→ Their analysis provides a finite time result (as opposed to asymptotic, such as for Q learning, sarsa, etc.)

- Various extensions since:
  - Brafman and Tenneholtz, Rmax
  - Kakade + al, Metric E3
  - Kearns and Koller, E3 in MDP w/transition model ~ temporal Bayes net

Hierarchical RL

- Main idea: use hierarchical domain knowledge to speed up RL
- I posted one representative paper onto the course website, should be a reasonable starting point if you wanted to find out more.

RL summary

- Exact methods: VI, PI, GPI, LP
- Model-free methods: TD, Q, sarsa
  - Batch versions: LSTD (recursive version: RLSTD), LSPI
- Function approximation:
  - Contractions – infinity norm, 2norm weighted by state visitation frequency
  - Approximate LP
- Policy gradient methods:
  - analytical, finite difference, likelihood ratio
  - Gradient <-> natural gradient
- Imitation learning:
  - Behavioral cloning <-> inverse RL