Reinforcement Learning: Exploration

Joelle Pineau

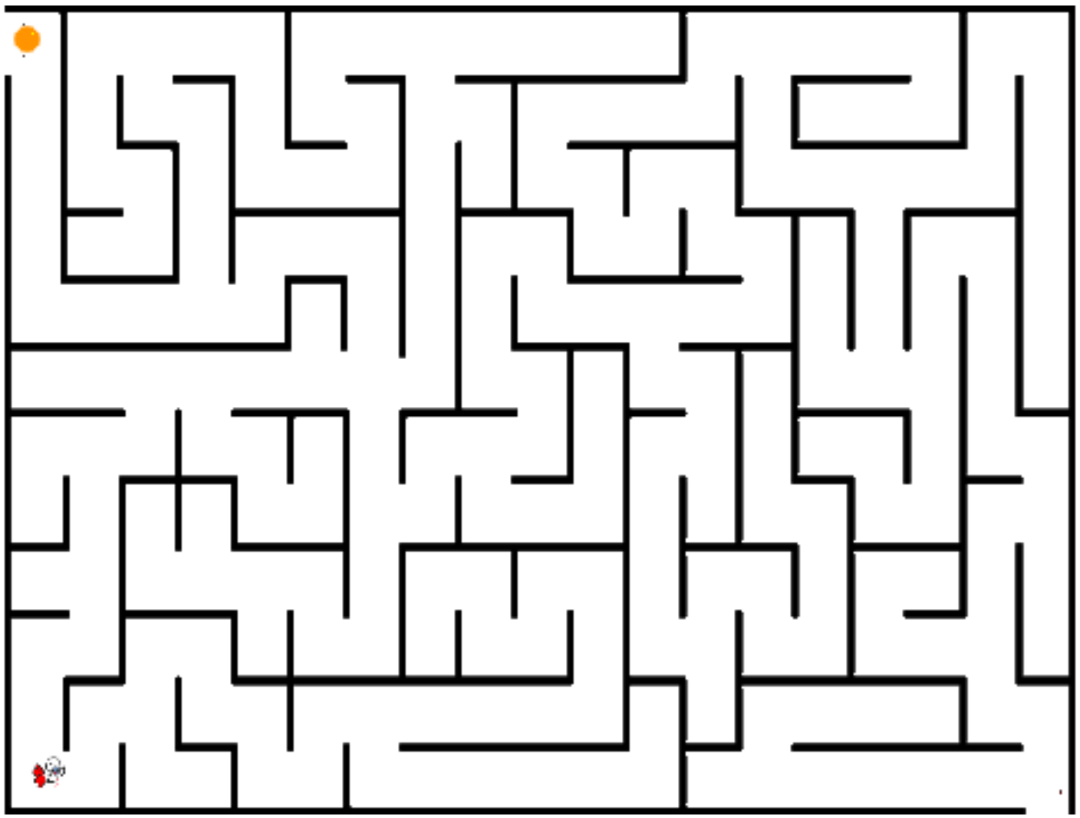
Reasoning and Learning Lab / *Center for Intelligent Machines*

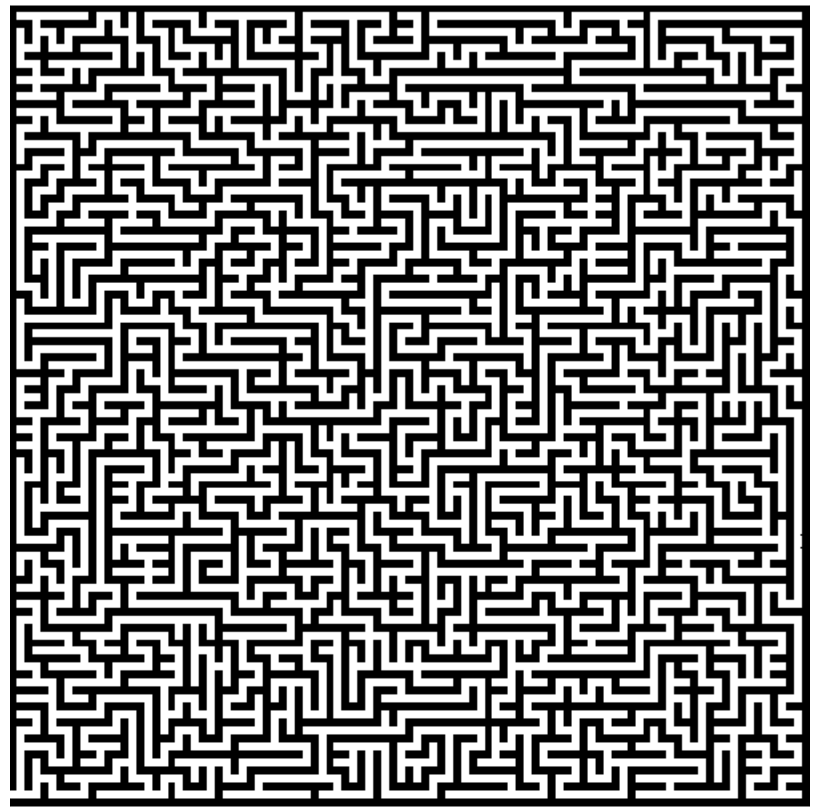
School of Computer Science, McGill University

Deep Learning Summer School

August 2016

Exploration





Exploration / Exploitation

Exploration: Increase knowledge for long-term gain, possibly at the expense of short-term gain

Exploitation: Leverage current knowledge to maximize short-term gain

Application #1: Internet advertising

- A large Internet company is interested in selling advertising on their website.
- It receives money when a company places an ad on the website and that ad gets clicked by a visitor to the website.
- What are the bandit's arms?

Application #1: Internet advertising

- A large Internet company is interested in selling advertising on their website.
- It receives money when a company places an ad on the website and that ad gets clicked by a visitor to the website.
- On a webpage, you can choose to display any of *n* possible ads.
  - Each ad is as an action, with an unknown probability of click rate.
  - If the add is clicked, there is a reward (utility), otherwise none.

Application #1: Internet advertising

- A large Internet company is interested in selling advertising on their website.
- It receives money when a company places an ad on the website and that ad gets clicked by a visitor to the website.
- On a webpage, you can choose to display any of *n* possible ads.
  - Each ad is as an action, with an unknown probability of click rate.
  - If the add is clicked, there is a reward (utility), otherwise none.
- Q: What is the best advertisement strategy to maximize return?
  - Note that this does not require knowledge of the user, the ad content, the webpage content, etc.

Application #2: Network server selection

- Suppose you can choose to send a job from a user to be processed on one of several servers.
- The servers have different processing speed (e.g. due to geographic location, load, etc.)

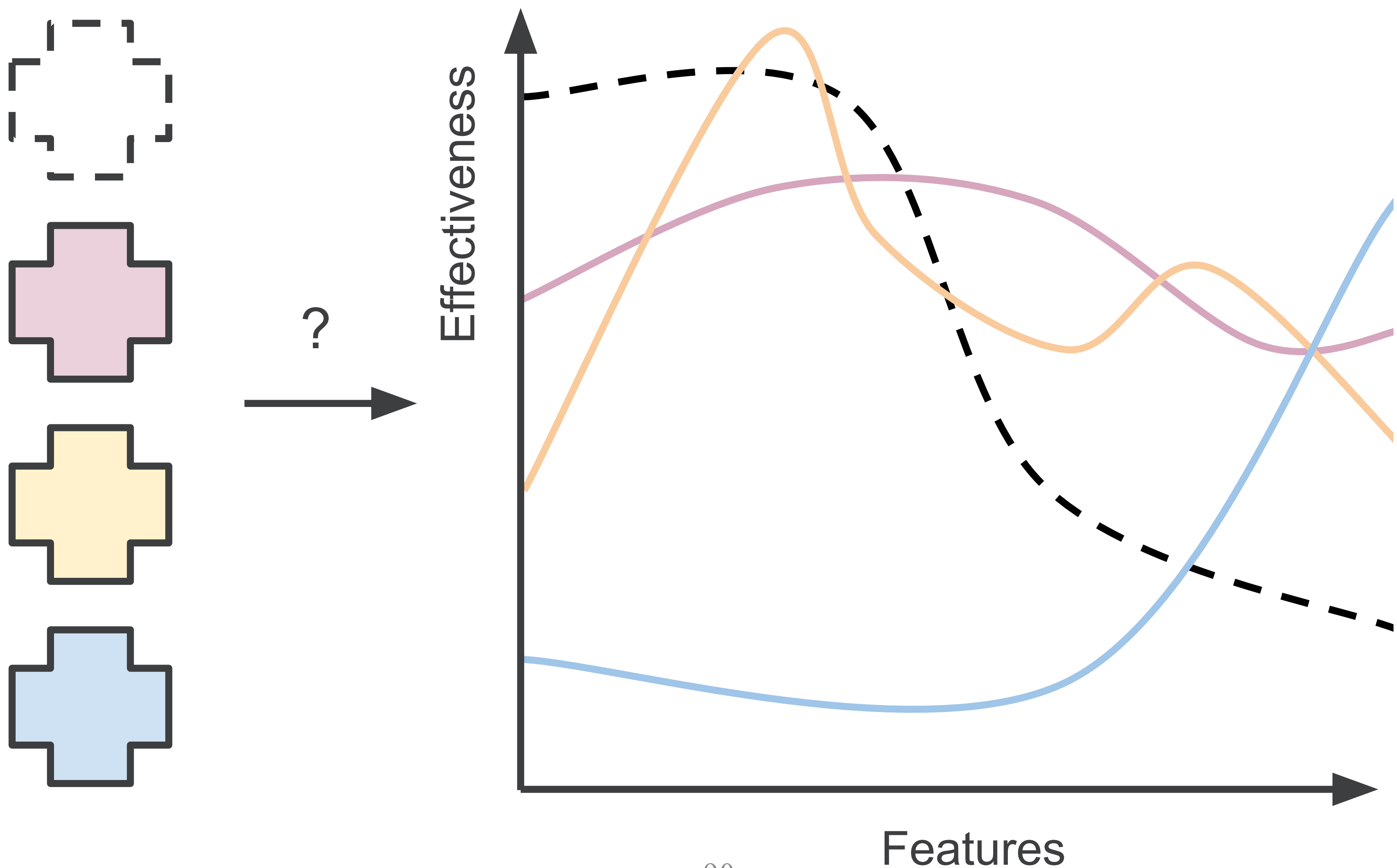
- What are the bandit's arms?

Application #2: Network server selection

- Suppose you can choose to send a job from a user to be processed on one of several servers.
- The servers have different processing speed (e.g. due to geographic location, load, etc.)
- What are the bandit's arms?
  - Each server can be viewed as an action (arm).
- Over time, you want to learn what is the best action to select.
- Used in routing, DNS server selection, cloud computing, etc.

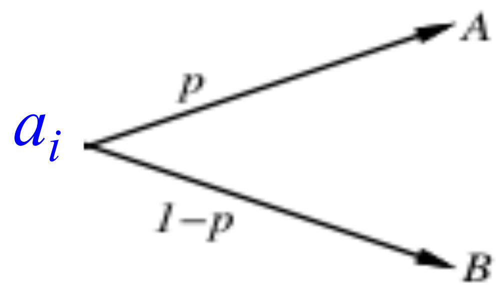
Personalized medical treatments

Which treatment should be assigned to a patient given the features of the person and/or the disease?



The multi-arm bandit

- A *K*-armed bandit is a collection of *K* actions (arms), each associated with a set of probabilistic outcomes:



https://vwo.com/blog/multi-armed-bandit-algorithm/

- Agent knows the number of arms, but not the outcomes or probabilities.

- The value of action *a* is its expected utility: $\mu_a = E[r \mid a]$.

The multi-arm bandit

- **Objective**: Choose a sequence of actions that maximizes the outcomes obtained in the long run.
  - Regret:

$$\mathfrak{R}_t = \mathbb{E}\left[\sum_{t=1}^{T}(\mu_\star - \mu_{a_t})\right]$$

- Bandit can use an adaptive strategy to change how it choses actions over time.

https://vwo.com/blog/multi-armed-bandit-algorithm/

The multi-arm bandit

- **Objective**: Choose a sequence of actions that maximizes the outcomes obtained in the long run.
  - Regret:

$$\mathfrak{R}_t = \mathbb{E}\left[\sum_{t=1}^{T}(\mu_\star - \mu_{a_t})\right]$$

https://vwo.com/blog/multi-armed-bandit-algorithm/

- Bandit can use an adaptive strategy to change how it choses actions over time.

- Simple adaptive strategy:
  - 10% of time, choose a random action.
  - 90% of time, choose action with best expected utility.

ε-greedy action selection

- Pick *ε∈(0,1)* (a constant), usually small (e.g. *ε=0.1*).
- On every play:
  - With probability *ε* you pull a random arm (explore).
  - With probability *1-ε*, pull the best arm according to current estimates (exploit).
- You can make *ε* depend on time (e.g. *1/t, 1/√*t, …)

- Advantage: Very simple! Easy to understand, easy to implement. Easy to extend to large state spaces.

- Disadvantage: Weak theoretical properties.

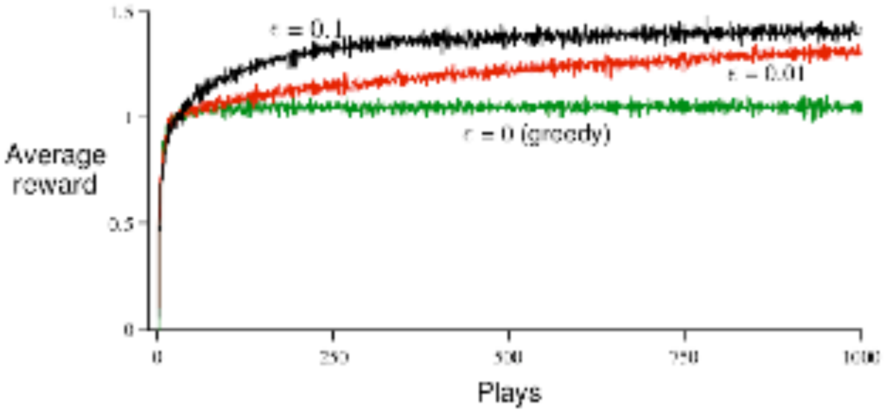
Example: 10-armed bandit problem

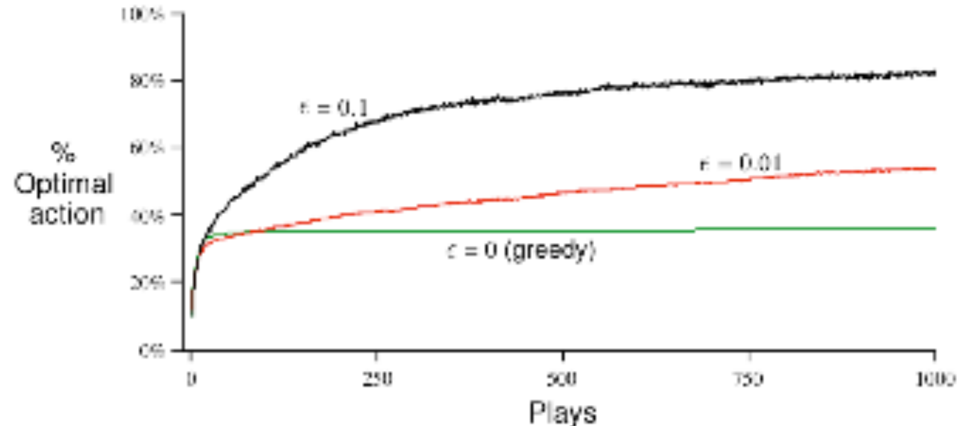
- The mean reward for each arm is chosen from a normal distribution with mean 0 and standard deviation 1

- Rewards are generated from a normal distribution around the true mean, with st.dev.1.

- We average 2000 different independent runs, each starts from $Q(a)=0$ and does 1000 pulls using the ε–greedy strategy.

Example: 10-armed bandit problem





- If ε is too high (not pictured here) rewards received during learning may be too low, and have high variance.

Softmax action selection

- **Key idea**: make the action probabilities a function of the current action values, $Q_t(a)$.

- At time t, we choose action a with probability proportional to:

$$Pr(a_t) = e^{Qt(a)/\tau}$$

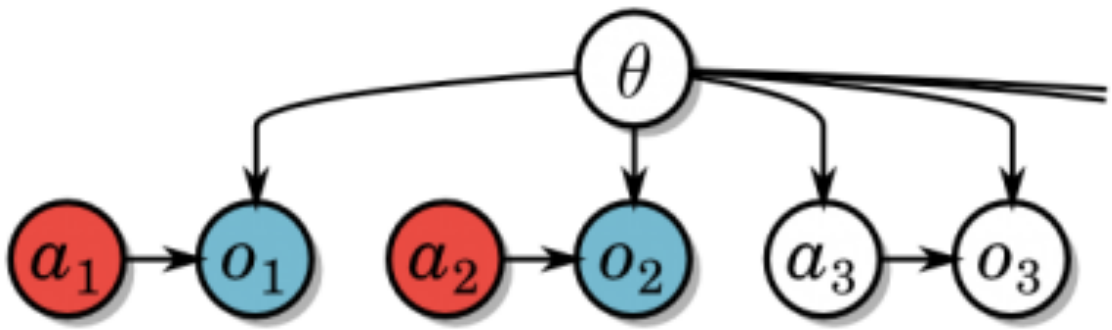
  - Normalize probabilities so they sum to 1 over the actions.
  - τ is the temperature parameter.

How does τ influence the exploration strategy?

Similar to simulated annealing.

Thompson sampling (1933)

- Bayesian adaptive strategy:
  - Maintain a posterior over the model, P(θ).
  - Sample a model θ from the posterior.
  - Select the action with highest expected utility.

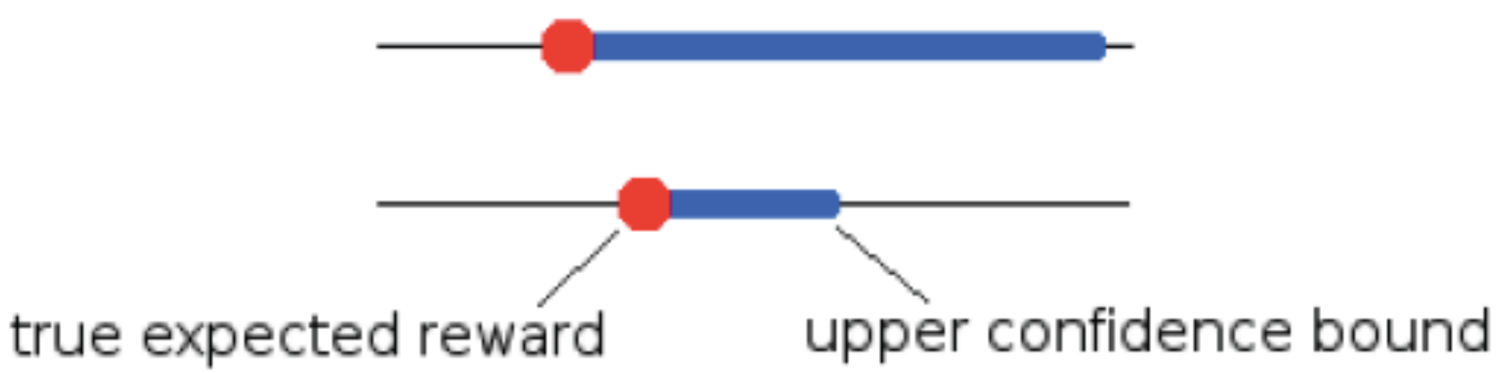


http://www.adaptiveagents.org/bayesian_control_rule

Strategy used in adaptive Bayesian trial designs.

Upper Confidence Bound (UCB) *[Auer, 2002]*

- Maintain an estimate of μ_a
- Estimate confidence on μ_a using # times action has been tried.
- Choose arm that maximizes μ_a + confidence bound



Contextual bandits

- Standard **multi-armed bandit** => **no notion of state**
  - » Ignores context, e.g. user information, words on the page.

- **Contextual bandits** incorporate state, in a feature vector *s*.
  - » The environment chooses the state.
  - » The agent chooses the action.

Contextual bandits

- Standard **multi-armed bandit => no notion of state**
  - » Ignores context, e.g. user information, words on the page.

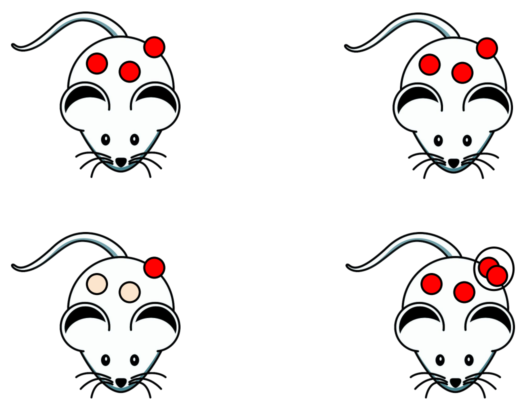
- **Contextual bandits** incorporate state, in a feature vector *s.*
  - » The environment chooses the state.
  - » The agent chooses the action.

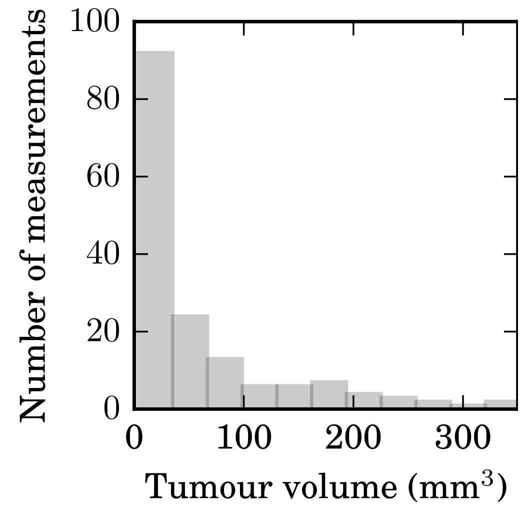
- The action's value depends on the state, e.g.: $Q(\mathbf{s}, a) = \mathbf{w}_a^T \mathbf{s}$

- Extensions of exploration methods: greedy, Boltzmann, UCB.
  - – Efficient choice of actions to learn *Q(s,a)*.

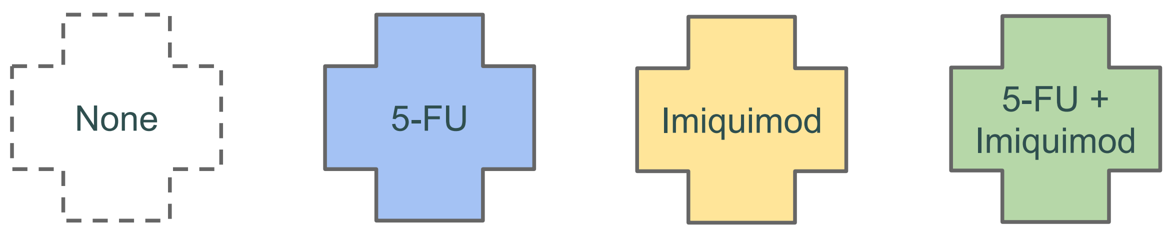
Contextual bandit problem: An example

- State: *x(t)* is the current tumour volume





- Action: *a(t)* is the treatment given at episode *t*



- Reward: *r(t)* is the tumour volume reduction
- Challenge: Estimate reward over continuous context.

Exploration in MDPs

- Straight-forward application of ε-greedy, softmax.
- Extension of Thompson sampling is conceptually simple, but computational expensive.
- For large action spaces: Start with imitation learning, then explore locally.
- Bayesian Reinforcement Learning

Bayesian reinforcement learning

- General idea:
  - Define prior distributions over all unknown parameters, *P(M)*.
  - Update posterior via Bayes' rule:

$$P(M \mid Y) = \frac{P(Y \mid M)P(M)}{P(Y)}$$

=> Optimize action choice w.r.t. posterior distribution over model.

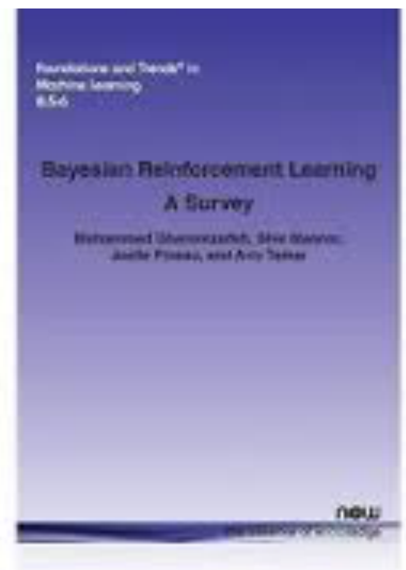
Bayesian reinforcement learning

- General idea:
  - Define prior distributions over all unknown parameters, *P(M)*.
  - Update posterior via Bayes' rule:

$$P(M \mid Y) = \frac{P(Y \mid M)P(M)}{P(Y)}$$

=> Optimize action choice w.r.t. posterior distribution over model.

- **Many advantages**:
  - Optimal trade-off of exploration and exploitatior
  - Explicit prior knowledge.
  - Learn enough about dynamics to accomplish the task (but not necessarily exhaustively).



Final comments

- If you **can't control the data** collection:
  - Don't worry about exploration!
  - Worry about off-policy correction, e.g. $\rho_t = \frac{\pi(s_t, a_t)}{b(s_t, a_t)}$

Final comments

- If you **can't control the data** collection:
  - Don't worry about exploration!
  - Worry about off-policy correction, e.g. $\rho_t = \frac{\pi(s_t, a_t)}{b(s_t, a_t)}$
- If you can afford to **collect lots of data**:
  - Use ε-greedy exploration with optimistic initialization of *Q()*.
  - Ensures coverage of all policies.

Final comments

- If you **can't control the data** collection:
  - Don't worry about exploration!
  - Worry about off-policy correction, e.g. $\rho_t = \frac{\pi(s_t, a_t)}{b(s_t, a_t)}$
- If you can afford to **collect lots of data**:
  - Use ε-greedy exploration with optimistic initialization of *Q()*.
  - Ensures coverage of all policies.
- If you can collect only **limited amounts of data**:
  - Bayesian methods tend to perform best.
  - Consider parameter-free method (BESA: Baranski et al. 2014).

Questions?