lecture 5

Previously…

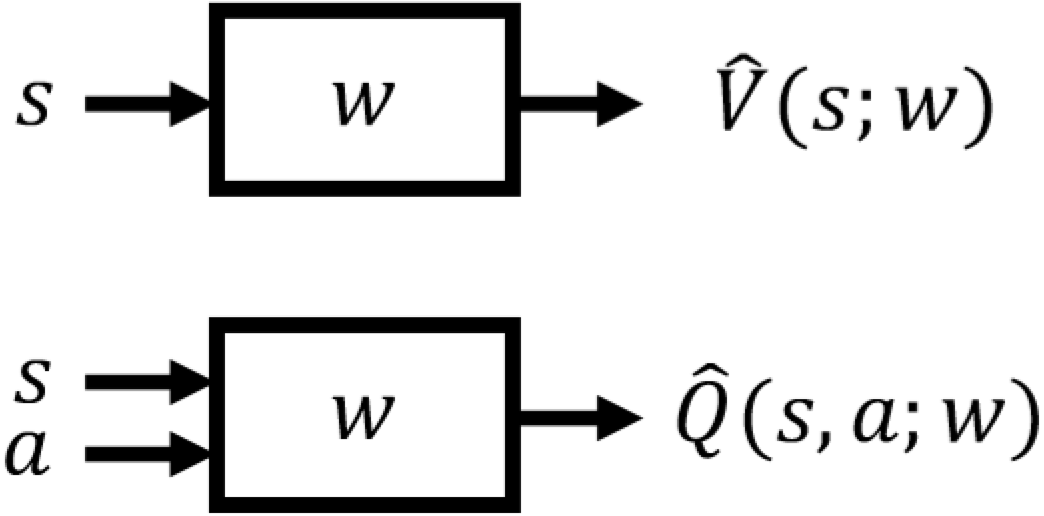
Q. Difference between tabular representation vs functional approximation?

Previously…

Q. Difference between tabular representation vs functional approximation?

A. Tabular representation is a table that holds probabilities/likelihood of every possible states as a result of current state + action, whilst functional approximation gives a more compact representation using parameters to represent the tabular representations.

Value Functional Approximation



Tabular Representation (lec2)

$$\begin{pmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{pmatrix} = \begin{pmatrix} R(s_1) \\ \vdots \\ R(s_N) \end{pmatrix} + \gamma \begin{pmatrix} P(s_1|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \ddots & \vdots \\ P(s_1|s_N) & \cdots & P(s_N|s_N) \end{pmatrix} \begin{pmatrix} V(s_1) \\ \vdots \\ V(s_N) \end{pmatrix}$$

Value Functional Approximation

=

Generalization

1. Reduce memory required
2. Reduce computation
3. Reduce experience…

Q1. I understand that it can be also described as reducing data required.. But how does it reduce experience..?

A1. 근사치를 구하니까 필요한 데이터도 줄어 수 있다.

lecture 5

From just policy evaluation → To Value Functional Approximation Prediction

From: Having a look up table of value estimates and then updating the value estimates each episode or steps

To: reapproximating function when every time new data is given (every step/run)

Feature vectors

$$\mathbf{x}(s) = \begin{pmatrix} x_1(s) \\ x_2(s) \\ \dots \\ x_n(s) \end{pmatrix} \qquad \mathbf{x}(s,a) = \begin{pmatrix} x_1(s,a) \\ x_2(s,a) \\ \dots \\ x_n(s,a) \end{pmatrix}$$

lecture 5

Update Linear VFA for Prediction with...

Oracle

$$\hat{V}(s;\boldsymbol{w}) = \sum_{j=1}^{n} x_j(s) w_j = \boldsymbol{x}(s)^T \boldsymbol{w}$$

$$J(\boldsymbol{w}) = \mathbb{E}_\pi[(V^\pi(s) - \hat{V}(s;\boldsymbol{w}))^2]$$

$$\Delta \boldsymbol{w} = -\frac{1}{2}\alpha \nabla_{\boldsymbol{w}} J(\boldsymbol{w})$$

$$\triangle\, \mathbf{w} = -\frac{1}{2}\alpha\left(2\left(V^\pi(s) - \hat{V}(s;\mathbf{w})\right)\right)\mathbf{x}(s)$$

Update = step-size x prediction error x feature value

lecture 5

Update Linear VFA for Prediction with…

Oracle

$$\triangle \mathbf{w} = -\frac{1}{2}\alpha\left(2\left(V^{\pi}(s) - \hat{V}(s;\mathbf{w})\right)\right)\mathbf{x}(s)$$

$$\triangle \mathbf{w} = \alpha\left(V^{\pi}(s) - \hat{V}(s;\mathbf{w})\right)\mathbf{x}(s)$$

Monte Carlo

$$\begin{aligned}\Delta \boldsymbol{w} &= \alpha(G_t - \hat{V}(s_t;\boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{V}(s_t;\boldsymbol{w}) \\ &= \alpha(G_t - \hat{V}(s_t;\boldsymbol{w}))\boldsymbol{x}(s_t) \\ &= \alpha(\boxed{G_t} - \boldsymbol{x}(s_t)^T\boldsymbol{w})\boldsymbol{x}(s_t)\end{aligned}$$

During algorithm for MC linear VFA policy evaluation

$G_t(s) = \sum_{j=t}^{L_k} r_{k,j}$

Gamma is set a 1, and this is no problem because MC itself is episodic = bound to terminate = return is bounded

lecture 5

Update Linear VFA for Prediction with…

Oracle

$$\triangle \mathbf{w} = -\frac{1}{2}\alpha\left(2\left(V^{\pi}(s) - \hat{V}(s;\mathbf{w})\right)\right)\mathbf{x}(s)$$

$$\triangle \mathbf{w} = \alpha\left(V^{\pi}(s) - \hat{V}(s;\mathbf{w})\right)\mathbf{x}(s)$$

Monte Carlo

$$\Delta \boldsymbol{w} = \alpha(G_t - \boldsymbol{x}(s_t)^T \boldsymbol{w})\boldsymbol{x}(s_t)$$

Temporal Difference

$$\begin{aligned}\Delta \boldsymbol{w} &= \alpha\underbrace{\boxed{(r + \gamma \hat{V}^{\pi}(s';\boldsymbol{w})}}_{\text{TD target}} - \hat{V}^{\pi}(s;\boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{V}^{\pi}(s;\boldsymbol{w}) \\ &= \alpha(r + \gamma \hat{V}^{\pi}(s';\boldsymbol{w}) - \hat{V}^{\pi}(s;\boldsymbol{w}))\boldsymbol{x}(s) \\ &= \alpha(r + \gamma \boxed{\boldsymbol{x}(s')}^T \boldsymbol{w} - \boldsymbol{x}(s)^T \boldsymbol{w})\boldsymbol{x}(s)\end{aligned}$$

Q2. Can I point at this and say that bootstrapping is used?

Convergence Guarantees

“The Markov Chain defined by a MDP with a particular policy will eventually converge to a probability distribution over states d(s)”

Q3. 엥? 무슨 뜻?

Convergence Guarantees

"The Markov Chain defined by a MDP with a particular policy will eventually converge to a probability distribution over states d(s)"

Q4. 엥? 무슨 뜻?

$$d(s) = \sum_{s'} \sum_{a} \overset{1}{\pi(s'|a)} \overset{2}{p(s'|s,a)} \overset{3}{d(s')}$$

모든 action과 state을 거쳐서 합한 게 d(s)이고 이는 1이다. 다 1로 합해지는 것은 당연. 혹시 다른 의미가 있을까

Convergence Guarantees

Stationary distribution

Q5. what does it mean by "stationary" in stationary distribution?

$$MSVE(\boldsymbol{w}) = \sum_{s \in S} d(s)(V^{\pi}(s) - \hat{V}^{\pi}(s; \boldsymbol{w}))^2$$

$$MSVE(\boldsymbol{w}_{MC}) = \boxed{\min_{\boldsymbol{w}} \sum_{s \in S} d(s)(V^{\pi}(s) - \hat{V}^{\pi}(s; \boldsymbol{w}))^2}$$

same

$$MSVE(\boldsymbol{w}_{TD}) \leq \boxed{\frac{1}{1-\gamma}} \boxed{\min_{\boldsymbol{w}} \sum_{s \in S} d(s)(V^{\pi}(s) - \hat{V}^{\pi}(s; \boldsymbol{w}))^2}$$

Error from bootstrapping

Control using VFA

Interleave

1. Policy evaluation
2. E-greedy policy improvement

But unstable

Deadly Triad

- Functional approximation
- Bootstrapping
- Off-policy learning

Control using VFA

Incremental model-free approach

MC

$$\Delta \boldsymbol{w} = \alpha(G_t - \hat{Q}(s_t, a_t; \boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{Q}(s_t, a_t; \boldsymbol{w})$$

SARSA

$$\Delta \boldsymbol{w} = \alpha(r + \gamma \hat{Q}(s', a'; \boldsymbol{w}) - \hat{Q}(s, a; \boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{Q}(s, a; \boldsymbol{w})$$

Q-learning

$$\Delta \boldsymbol{w} = \alpha(r + \gamma \max_{a'} \hat{Q}(s', a'; \boldsymbol{w}) - \hat{Q}(s, a; \boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{Q}(s, a; \boldsymbol{w})$$

lecture 5

Control using VFA

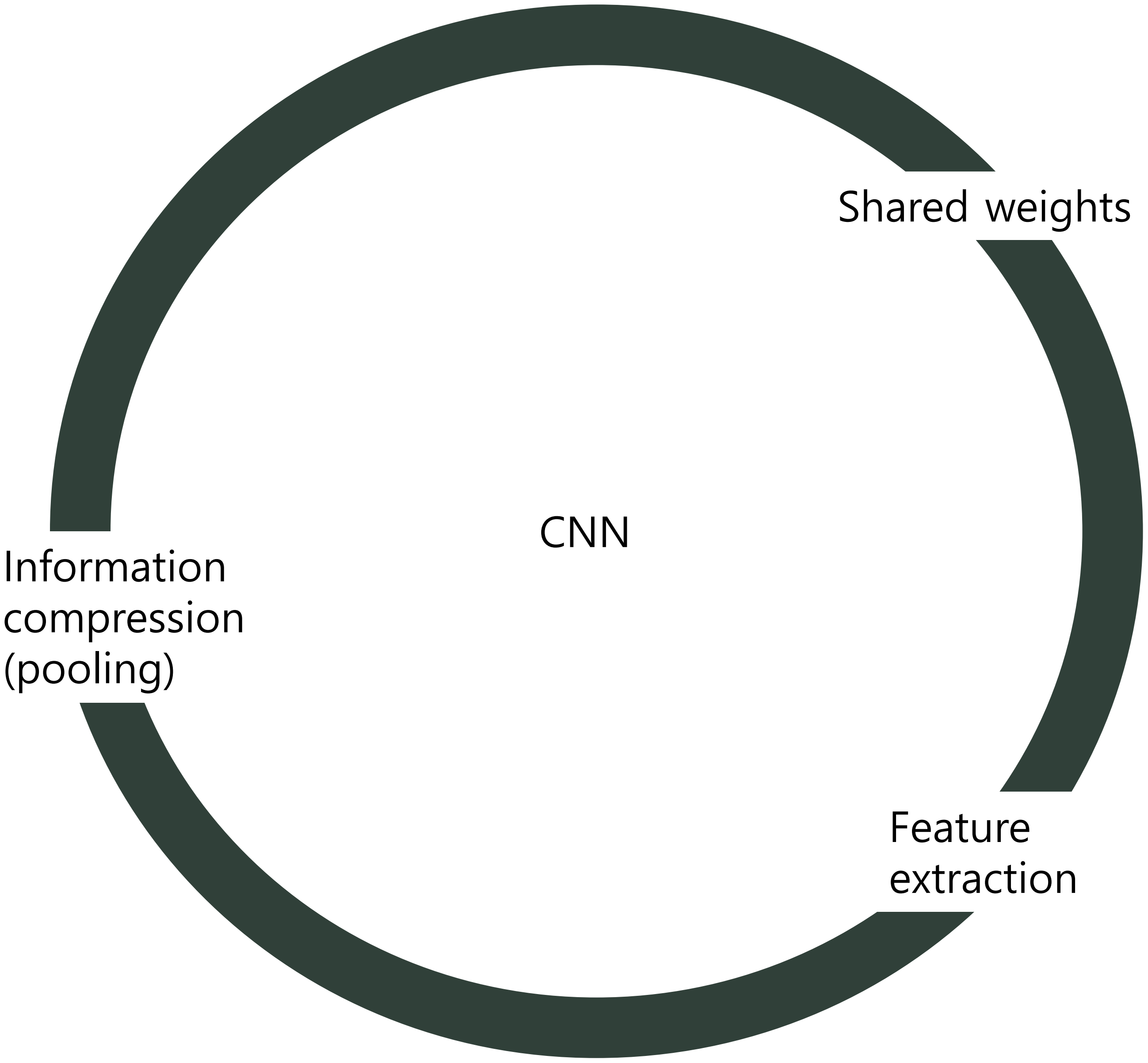
Q6. What is the difference between linear and nonlinear VFA?

Algorithm	Tabular	Linear VFA	Nonlinear VFA
Monte-Carlo Control	converges	Converges but might have some oscillation	nope
SARSA	converges	Converges but might have some oscillation	Nope
Q-learning	converges	Nope	Nope

lecture 6

DNN

CNN


Shared weights
CNN
Information compression (pooling)
Feature extraction

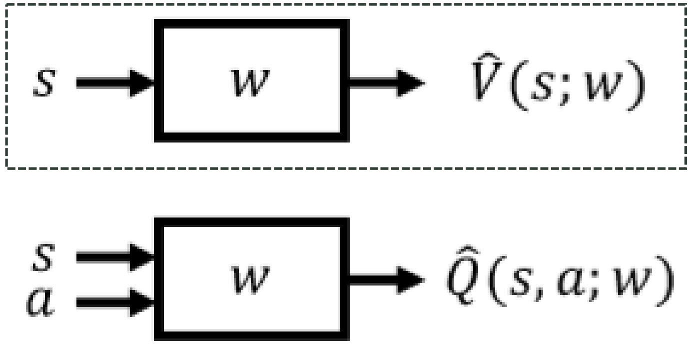
lecture 6

Deep Q-Learning

Deep Reinforcement Learning

Deep Q-Networks

Q7. Don't need the Value function in the DQN's only Q function (parameterized) required. Isn't it?



From lecture 5

Control using VFA

Incremental model-free approach

MC

$$\Delta \boldsymbol{w} = \alpha(G_t - \hat{Q}(s_t, a_t; \boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{Q}(s_t, a_t; \boldsymbol{w})$$

SARSA

$$\Delta \boldsymbol{w} = \alpha(r + \gamma\hat{Q}(s', a'; \boldsymbol{w}) - \hat{Q}(s, a; \boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{Q}(s, a; \boldsymbol{w})$$

Q-learning

$$\Delta \boldsymbol{w} = \alpha(r + \gamma \max_{a'} \hat{Q}(s', a'; \boldsymbol{w}) - \hat{Q}(s, a; \boldsymbol{w}))\nabla_{\boldsymbol{w}}\hat{Q}(s, a; \boldsymbol{w})$$

lecture 6

DQN

Issues
- Correlation between samples
- Non-stationary targets

solutions
- Experience replay
- Fixed Q-targets

lecture 6

Double DQN

$$\Delta \boldsymbol{w} = \alpha\Big(r + \gamma \overbrace{\hat{Q}(\underbrace{\arg\max_{a'} \hat{Q}(s', a'; \boldsymbol{w})}_{\text{Action selection: } \boldsymbol{w}}; \boldsymbol{w}^{-})}^{\text{Action evaluation: } \boldsymbol{w}^{-}} - \hat{Q}(s, a; \boldsymbol{w})\Big)$$

Prioritized order replay

$$p_i = \left| r + \gamma \max_{a'} Q(s_{i+1}, a'; \boldsymbol{w}^{-}) - Q(s_i, a_i; \boldsymbol{w}) \right|$$

$$P(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}}$$

Dueling DQN

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$$

Double DQN

$$\Delta \boldsymbol{w} = \alpha(r + \gamma \overbrace{\hat{Q}(\underbrace{\arg\max_{a'} \hat{Q}(s', a'; \boldsymbol{w})}_{\text{Action selection: } \boldsymbol{w}}; \boldsymbol{w}^{-})}^{\text{Action evaluation: } \boldsymbol{w}^{-}} - \hat{Q}(s, a; \boldsymbol{w}))$$

Q7. Difference between fixed Q-target.

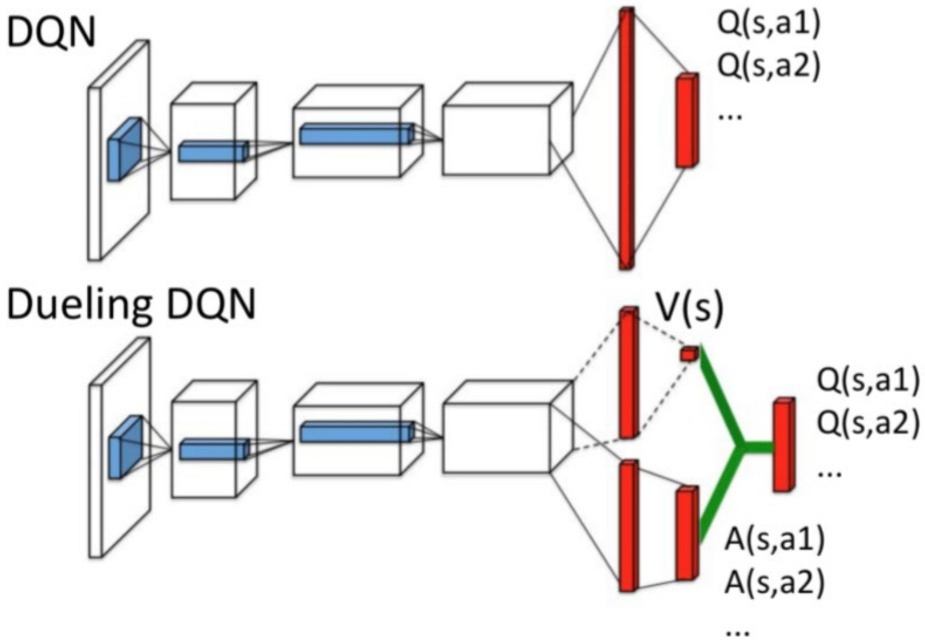
lecture 6

TD Error - Priority of a tuple is proportional to DQN error

Prioritized order replay

$$p_i = \boxed{\left| r + \gamma \max_{a'} Q(s_{i+1}, a'; \boldsymbol{w}^-) - Q(s_i, a_i; \boldsymbol{w}) \right|}$$

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$



How much better or worse taking a particular action versus following the current policy

Identifiability : Whether there exists a unique Q for A and V given pi (policy)

Dueling DQN $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$

lecture 7

DQN may require too large number of samples to learn a good policy

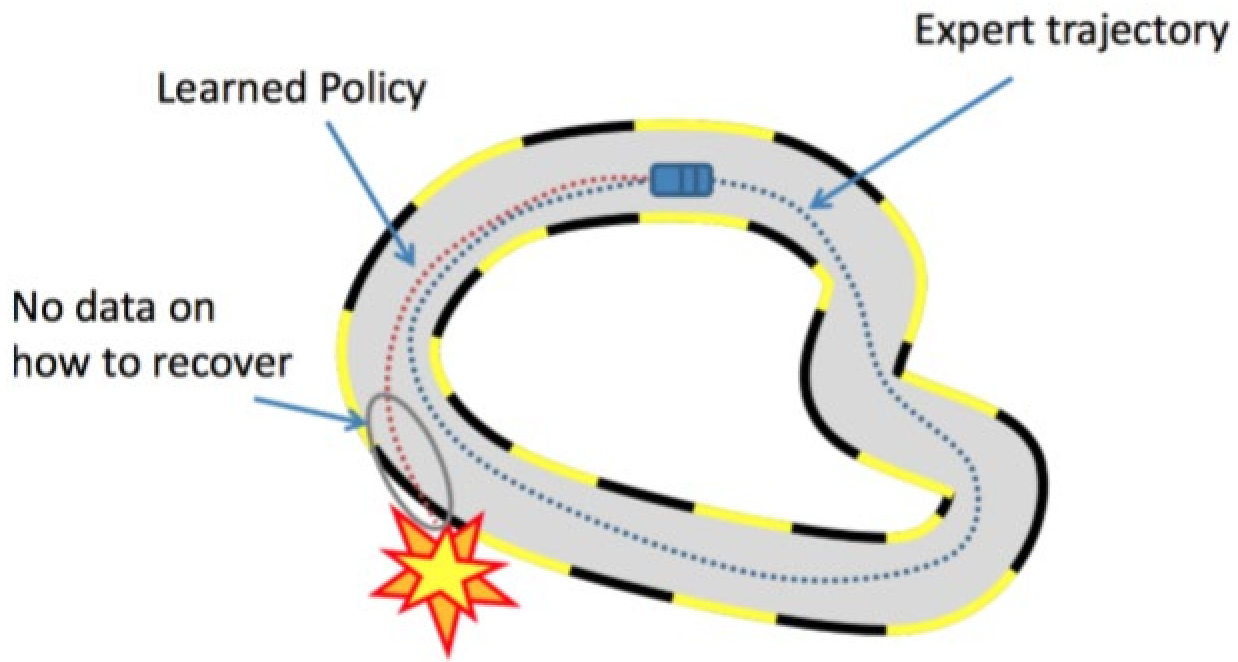
Or even a large number of samples used to learn might not promise a good policy improvement

→ Imitation learning

→ Imitation learning

→ Behavioral Cloning : Estimate policy from training examples

→ Problem : Compounding Error



→ Solution : DAGGER (or is it?)

lecture 7

→ Imitation learning

→ Apprenticeship learning via Inverse RL

Inverse RL

1. Identify weight vector given demonstrations (from teacher)
2. Use that to calculate function of $\boxed{\text{policy pi}}$ ⟷ Distribution of states
3. Mu : discounted sum of state features under policy pi

= Q8. Can I understand it as Indirectly gaining information on the policy?

Q9. So are we calculating anything at the 'just' Inverse RL stage? Or is it a method to combine with apprenticeship learning?

lecture 7

→ Imitation learning

→ Apprenticeship learning via Inverse RL

Possibly wrong

Inverse RL

1. Identify weight vector given demonstrations (from teacher)
2. Use that to calculate function of policy pi
3. Mu : discounted sum of state features under policy pi

Maybe room for some discussion?

→ Imitation learning

→ Apprenticeship learning via Inverse RL

Find w such that…

$$w^{*T}\mu(\pi^*) \geq w^{*T}\mu(\pi), \forall\pi \neq \pi^*$$

$$\arg\max_{\boldsymbol{w}} \max_{\gamma} s.t. w^T\mu(\pi^*) \geq w^T\mu(\pi) + \gamma \quad \forall\pi \in \{\pi_0, \pi_1, \ldots, \pi_{i-1}\}$$

Q10. So… how do we obtain mu? ~~Mu is the demonstration itself~~? Only calculation of weight w is sufficient in finding out the reward function?

Ambiguity

- There is an infinite number of reward functions with the same optimal policy.
- There are infinitely many stochastic policies that can match feature counts
- Which one should be chosen?

Maybe room for some discussion?