Abstraction

Following from Near Optimal Behavior via Approximate State Abstraction, I am interested in using their framework where abstractions that approximately

$$\phi(\cdot) = \phi(\cdot) \implies \forall_\cdot \mid f(\cdot) - f(\cdot) \mid \leq \epsilon$$

What does the above say? *We want our abstraction to contain approximate symmetries $\forall x \mid f(x) - f(T(x)) \mid \approx 0$.*

What can we pick as f? 1. $\mid Q(\cdot) - Q(\cdot) \mid \leq \epsilon$. 2. $\sum_{\cdot \in N} \mid \tau(\cdot) - \tau(\cdot) \mid \leq \epsilon$ AND $\mid r(\cdot) - r(\cdot) \mid \leq \epsilon$. The transition and reward) functions are approximately the same. 3. ???

Note: two states having similar f are not guaranteed to have similar abstraction!

Given an abstraction of the form above, we want to find an abstraction that allows us to find a policy in the abstracted MDP that achieves bounded error wrt the original MDP. We want to find a tight η_f s.t.

$$\forall_{s \in S_G, a \in A_G} \mid Q_G^{\pi^*}(s, a) - Q_G^{\pi^*_{GA}}(s, a) \mid \leq 2\epsilon\eta_f$$

> **Claim 1:** 1. will yield the most compression, but it is also task specific, this it will not transfer / generalise well.

Extension to other types of abstraction

State abstraction groups together states that are similar. For example, sprinting 100m is equivalent regardless of which track lane you are in.

Action abstraction groups together actions that are similar. For example, X and Y both yeild the state change in state,

$$\phi(s_1) = \phi(s_2) \implies \forall_a \mid Q(s_1, a) - Q(s_2, a) \mid \leq \epsilon \qquad \text{(State abstraction)}$$
$$\phi(a_1) = \phi(a_2) \implies \forall_s \mid Q(s, a_1) - Q(s, a_2) \mid \leq \epsilon \qquad \text{(Action abstraction)}$$
$$\phi(s_1, a_1) = \phi(s_2, a_2) \implies \mid Q(s_1, a_1) - Q(s_2, a_2) \mid \leq \epsilon \qquad \text{(State-action abstraction)}$$
$$\text{(1)}$$

Generalised symmetries

What about other types of symmetry, other than mirror?

- $\exists f \in X : \forall_{s,a} r(s,a) = r(f(s),a)$. Where $X = GL_N \vee S_N \vee \ldots$

> **Claim 2:** The state-action abstraction is the most powerful because it allows the compression of the most symmetries. (want to prove!)

Motivating example: Symmetric maze

Imagine you are in a mirror symmetric maze. It should not matter to you which side of mirror you are on.

This reduces the state-action space by half! $\frac{1}{2} \mid S \mid \times \mid A \mid$. Note: just using state abstraction it is not possible to achieve this reduction. Mirrored states are not equivalent as the actions are inverted.

Transition based abstraction for transfer

Abstraction for model based RL!? Learning abstract models of the transition function.

As noted above. Using the Q function or the reward function to construct an abstraction makes the abstraction task specific.

$$\forall_{s_A \in S_A} \mid \sum_{s'_G \in G(s_A)} \tau(s_1, a, s'_G) - \tau_G(T(s_2), T(a), T(s'_G)) \mid \leq \epsilon$$

Temporal abstraction

$$Q(s_{1:m}, a) = \sum_1^{m-1} \gamma^t r(s_t) + \gamma^m Q(s_m, a) \tag{2}$$

$$\phi(s_{1:m}) = \phi(x_{1:n}) \implies \mid Q(s_{1:m}, a) - Q(x_{1:n}, a) \mid \leq \epsilon \qquad \text{(Temporal state abstraction)}$$
$$\tag{3}$$

$$Q(s, a_{1:m}) = \sum_1^{m-1} \gamma^t r(s_t, a_t) + \gamma^m Q(s_m, a_m) \tag{4}$$

$$\phi(a_{1:m}) = \phi(b_{1:n}) \implies \mid Q(s, a_{1:m}) - Q(s, b_{1:n}) \mid \leq \epsilon \qquad \text{(Temporal action abstraction)}$$
$$\tag{5}$$

Can be framed as an unusual type of action abstraction, where the meaning of abstraction is being stretched… Initially we had $Q(s,a) \quad s \in S, a \in A$ but we have transformed this (in the case of temporal-action abstraction) to $Q(s,a) \quad s \in S, a \in A^*$.

Notes

Struggling with the direction of implication, $\phi(s_1) = \phi(s_2) \implies \forall_a \mid Q(s_1, a) - Q(s_2, a) \mid \leq \epsilon$, what about $\phi(s_1) = \phi(s_2) \impliedby \forall_a \mid Q(s_1, a) - Q(s_2, a) \mid \leq \epsilon$?

But can we guarantee that these abstractions do not make it harder to find the optimal policy? Is that even possible?

So this is about messing with what information the value function has. Given more information about the future, we should expect the acuracy (or speed of learning) of the estimate to go up!?