Outline

Value Aggregation

Does Q-learning converge on aggregated problems?

Learning the aggregation
Value Aggregation
Histories

In general reinforcement learning, agent takes action $a_t$ at time $t$, receives observation $o_{t+1}$, reward $r_{t+1}$. History $h_t \in \mathcal{H}_t$ is the full sequence up to time $t$.

$$h_t = o_1 r_1 a_1 ... a_{t-1} o_t r_t$$

History is generated stochastically by environment $P$ and agent’s policy $\Pi$.

$$P : \mathcal{H} \times \mathcal{A} \xrightarrow{} \mathcal{O} \times \mathcal{R}$$

$$\Pi : \mathcal{H} \xrightarrow{} \mathcal{A}$$

$$P(o_{t+1} r_{t+1} | h_t a_t) \quad \Pi(a_t | h_t)$$
Feature map $\phi$

Can define a feature map $\phi : H \rightarrow S$ that maps histories to a finite set of states $S$.

E.g. if $P$ is a finite state fully-observable MDP, can use

$$\phi(h_t) = s_t = o_t$$

E.g. if $P$ is a $k$-order MDP

$$\phi(h_t) = s_t = o_{t-k}r_{t-k}a_{t-k}...o_t$$

$\phi$ aggregates histories. For appropriate $P$, $\phi$, can define reduced process $p$

$$p : S \times A \rightsquigarrow S \times R \quad p(s_{t+1}r_{t+1}|s_t)$$
The Value Function

For histories, pseudo-recursive equations

\[ Q^\Pi(h_t, a_t) = E_P[r_{t+1} + \gamma V^\Pi(h_{t+1})|h_t a_t] \]
\[ V^\Pi(h_t) = Q^\Pi(h_t, \Pi(h_t)) \]

Note: \( h_t \) is never encountered again.

For finite state MDP, equations are recursive

\[ q^\pi(s_t, a_t) = E_p[r_{t+1} + \gamma v^\pi(s_{t+1}|s_t a_t)] \]
\[ v^\pi(s_t) = q^\pi(s_t, a_t) \]
Why aggregate?

A good reinforcement learner learns the optimal policy as fast as possible.

If $P$ is a $k$-order MDP, observations $O$ are $D$-dimensional with $n$ possible values per dimension, state space is size $O(n^kD)$; will need many samples to learn value function. Want a map $\phi$ to reduce this while still preserving ability to learn optimal policy.

If $P$ is general non-Markovian, histories never repeats so cannot generally estimate $P$ from experience. Want $\phi$ to reduce $P$ to a finite state process $p$, while still learning the optimal policy.
When can we aggregate?

Hutter (2014)[1]: Given $P$, $\phi$ such that

$$
\Pi^*(h) = \Pi^*(\tilde{h})
$$

$$
V^*(h, a) = V^*(\tilde{h}, a)
$$

for all $\phi(h) = \phi(\tilde{h})$

then

$$
V^*(h, a) = v^*(s, a)
$$

$$
\Pi^*(h) = \pi^*(\phi(h))
$$

A similar condition holds for $Q^*$. This is “exact aggregation”.
When can we aggregate?

For approximate aggregation, conditions are

\[ \Pi^*(h) = \Pi^*(\tilde{h}) \]
\[ |Q^*(h, a) - Q^*(\tilde{h}, a)| \leq \epsilon \]

for all \( \phi(h) = \phi(\tilde{h}) \)

then

(i) \( |Q^*(h, a) - q^*(\pi(h), a)| \leq \frac{\epsilon}{1 - \gamma} \) and \( |V^*(h, a) - v^*(\pi(h), a)| \leq \frac{\epsilon}{1 - \gamma} \)

(ii) \( 0 \leq V^*(h) - V^{\tilde{\Pi}}(h) \leq \frac{2\epsilon}{(1 - \gamma)^2} \) where \( \tilde{\Pi}(h) = \pi^*(\phi(h)) \)

(iii) If \( \epsilon = 0 \) then \( \Pi^*(h) = \pi^*(\phi(h)) \)
Questions

• For approximate $V^*$ aggregation, does (ii) hold?

• Does Q-learning converge to $Q^*$ for exact/approximate aggregation?

• How do we find/learn appropriate $\phi$?
Approximate $V^*$ aggregation

Approximate $V^*$ aggregation does not, in general, admit a bound on $V^*(h) - V^{\tilde{\pi}}(h)$. Jan Leike:
\( V^* \) aggregation

Stochastic inverse \( B(F|\phi(F)) = 1, B(H|\phi(H)) = 1/2 \) and \( B(C|\phi(C)) = 0 \).

\[
\begin{align*}
V^*(E) &= \frac{1}{1-\gamma} + \epsilon \\
V^*(G) &= \frac{1}{1-\gamma} \\
V^*(F) &= \frac{1}{1-\gamma} \\
V^*(H) &= \frac{1}{1-\gamma} + \epsilon \\
V^*(C) &= V^*(D) = \frac{\gamma}{1-\gamma} + \gamma\epsilon
\end{align*}
\]
Second counterexample

\[ \phi(C), \phi(A), \phi(E) \]

Action = \( \alpha \) if \( T_{AD}, T_{BD} = 1 \)

Action = \( \beta \) if \( T_{AF}, T_{BF} = 1 \)
Does Q-learning converge on aggregated problems?
Random Aggregateable MDPs

Want an MDP \( (T^a_{ss'}, R^a_{ss'}, \gamma) \) such that a \( \phi \) exists that is a \( Q^* \)-aggregation.

For \( |A| = 1 \): Specify \( T \) arbitrarily, and specify \( v(s) \) so that desired aggregation is possible. Constrain \( R \) such that reward only depends on the final state of any transition \( R_{s's'} = R_{ss'} \) for all \( s \in S \). Then, we can find \( R \) by solving

\[
Tr = (I - \gamma T)v
\]

Can choose \( v \) so that aggregation is \( Q^* \) or \( V^* \), exact or approximate.

Adding more actions: make sure they’re all sub-optimal
Convergence Example

\[ \Delta = \sum_{a,s} |\hat{q}(a, s) - q^*(a, s)| \]
Comparing raw and aggregated problems

\[ R = \frac{\Delta^r_n}{\Delta^a_n} \text{ where } \Delta_n \text{ is } \Delta \text{ after } n \text{ timesteps.} \]
Approximate vs exact

\[ V(s) = v(\phi(s)) + \epsilon \]

\[ \epsilon \sim \text{Uniform}[0, \epsilon_{\text{max}}] \]
State aggregation for the mountain car task

Used k-means to define 16 q-value clusters
Performance of aggregated mountain car agents
Learning the aggregation
Learning the aggregation

How do we know if a representation is a $Q^*$-aggregation?

Easy if you know the q-value function, hard if you don’t.

Adapt state representation as agent learns $q(s, a)$
U-Tree

McCallum (1996)[2]: Algorithm to learn an adaptive state representation using utile (value) distinctions.
U-Tree

4 states, 2 aggregated states
\( o_t = (\phi(s_t), s_t) \)

\( t = 0 \) dim = 0

\( t = 0 \) dim = 1

\( \hat{v}^* = 1.85 \)

\( \hat{v}^* = 1.99 \)

\( \hat{v}^* = 1.97 \)

\( \hat{v}^* = 1.86 \)
Kolmogorov-Smirnov statistic

U-Tree uses Kolmogorov-Smirnov test to decide when to split nodes. Alternative tests may work better.
source: wikipedia
MERL

Exploration algorithm by Lattimore (2013) [3]. Given a finite set of models $\mathcal{M}$ of size $|\mathcal{M}| = N$, containing true environment $\mu$, confidence $\delta$ and precision $\epsilon$, the agent will converge to within $\epsilon$ of the true value with probability $1 - \delta$ after

$$\tilde{O}\left(\frac{N}{\epsilon^2(1 - \gamma)^3 \log^2 \frac{N}{\delta \epsilon(1 - \gamma)}}\right)$$

timesteps.

General agent: no assumption that $\mu$ is MDP.

Question: If we weaken assumption to $\phi(\mu) \in \mathcal{M}$ where $\phi(\mu)$ is an exact $Q^*$-aggregation of $\mu$, does it still work?
MERL

MERL uses expected $d$-step return to approximate the value function:

$$V_\mu^\pi (h_t; d) = \mathbb{E} \left[ \sum_{k=t}^{t+d} \gamma^{k-t} r_k | h_t \right]$$

$V_\mu^\pi (h_t; d)$ may not be the same as $V_{\phi(\mu)}^\pi (h_t; d)$. 
Figure 1:
Open Questions

• Does Q-learning converge for all approximately $Q^*$ aggregated representations?

• Can aggregated representations fill the role of the true environment in other algorithms (e.g. UCRL($\gamma$))?

• Are there any ways to determine if a representation is a $Q^*$-aggregation besides checking value functions?

• Is there an agent that heuristically often learns minimal $Q^*$ aggregated representations?
References

