Approximate Q-Learning

Dan Weld / University of Washington

[Many slides taken from Dan Klein and Pieter Abbeel / CS188 Intro to AI at UC Berkeley – materials available at http://ai.berkeley.edu.]

Q Learning

For all s, a
Initialize Q(s, a) = 0

Repeat Forever
Where are you? s.
Choose some action a
Execute it in real world: (s, a, r, s')
Do update:

\[
Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') \right]
\]

difference = \left[ r + \gamma \max_{a'} Q(s', a') \right] - Q(s, a)

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \text{[difference]}
\]
Q Learning

**For all s, a**

- Initialize $Q(s, a) = 0$

**Repeat Forever**

Where are you? s.

Choose some action a

Execute it in real world: $(s, a, r, s')$

Do update:

\[
\text{difference} = \left[ r + \gamma \max_{a'} Q(s', a') \right] - Q(s, a)
\]

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \text{ [difference]}
\]
Generalizing Across States

- Basic Q-Learning updates a table of all q-values
- In realistic situations, we can’t possibly learn about every single state!
  - Too many states to visit them all in training
  - Too many states to hold the q-table in memory
- Instead, we want to generalize:
  - Learn about some small number of \(<s,a>\) from experience
  - Generalize experience to new, similar situations
  - Fundamental idea in machine learning, we’ll see it over and over again

Example: Pacman

Let’s say we discover through experience that this state is bad:

![Pacman game state](image1)

In naïve q-learning, we know nothing about this state:

![Pacman game state](image2)
Example: Pacman

Let's say we discover through experience that this state is bad:

Or even this one!

Feature-Based Representations

Soln: describe states w/ vector of features (aka “properties”)

- Features = functions from states to R (often 0/1) capturing important properties of the state

- Examples:
  - Distance to closest ghost or dot
  - Number of ghosts
  - $1 / (\text{dist to dot})^2$
  - Is Pacman in a tunnel? (0/1)
  - Is state the exact state on this slide?
  - etc.

- Can also describe a q-state $(s, a)$ with features (e.g. action moves closer to food)
How to use features?

Using features we can represent $V$ and/or $Q$ as follows:

$$V(s) = g(f_1(s), f_2(s), ..., f_n(s))$$

$$Q(s,a) = g(f_1(s,a), f_2(s,a), ..., f_n(s,a))$$

What should we use for $g$ (and $f$)?

Linear Combination

- Using a feature representation, we can write a q function (or value function) for any state using a few weights:

$$V(s) = w_1f_1(s) + w_2f_2(s) + ... + w_nf_n(s)$$

$$Q(s,a) = w_1f_1(s,a) + w_2f_2(s,a) + ... + w_nf_n(s,a)$$

- **Advantage**: our experience is summed up in a few powerful numbers

- **Disadvantage**: states sharing features may actually have very different values!
Approximate Q-Learning

Approximate Q-Learning

\[ Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a) \]

- Q-learning with linear Q-functions:

  \[ \text{transition } = (s,a,r,s') \]

  \[ \text{difference } = [r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \]

  \[ Q(s,a) \leftarrow Q(s,a) + \alpha \text{ [difference]} \]

  \[ w_i \leftarrow w_i + \alpha \text{ [difference]} f_i(s,a) \]

- Intuitive interpretation:
  - Adjust weights of active features
  - E.g., if something unexpectedly bad happens, blame the features that were on: **disprefer all states with that state's features**

- Formal justification: in a few slides!

Example: Pacman Features

\[ Q(s,a) = w_1 f_{DOT}(s,a) + w_2 f_{GST}(s,a) \]

\[ f_{DOT}(s,a) = \frac{1}{\text{distance to closest food after taking } a} \]

\[ f_{DOT}(s,NORTH) = 0.5 \]

\[ f_{GST}(s,a) = \text{distance to closest ghost after taking } a \]

\[ f_{GST}(s,NORTH) = 1.0 \]
Example: Q-Pacman

\[ Q(s, a) = 4.0 f_{DOT}(s, a) - 1.0 f_{GST}(s, a) \]

\[ f_{DOT}(s, \text{NORTH}) = 0.5 \]
\[ f_{GST}(s, \text{NORTH}) = 1.0 \]

\[ r + \gamma \max_{a'} Q(s', a') = -500 + 0 \]

\[ Q(s', \cdot) = 0 \]

\[ \alpha = 0.004 \]

\[ \text{difference} = -501 \]

\[ w_{DOT} \leftarrow 4.0 + \alpha [-501] \times 0.5 \]
\[ w_{GST} \leftarrow -1.0 + \alpha [-501] \times 1.0 \]

\[ Q(s, a) = 3.0 f_{DOT}(s, a) - 3.0 f_{GST}(s, a) \]