Reinforcement learning

CMSC 723 / LING 723 / INST 725

Hal Daumé III [he/him]

31 Oct 2019

Announcements, logistics

- Homework 4/5:
  - A few fixes on written homework (sorry!), you get an extra attempt as an apology
  - HW4-programming will be up early next week
- Project meetups:
  - If you haven't signed up, please do so by tomorrow!
  - Think of this as an "ask me anything" and come with questions (we'll try to answer)
- Grading:
  - HW3 grades out soon
  - We're working on regrades of early exam
- Late exam:
  - If you have a big deadline/conference +/- 2 days of late exam, you can take it on the following Monday; please mark availability *today!* (See ELMS)

Last time

- Dependency parsing transition systems
- Learning from imitation
    - Supervised learning: assume all past decisions are correct
    - Imitation learning: allow for errors in past
    - Key ideas: Optimal action, minimal achievable loss

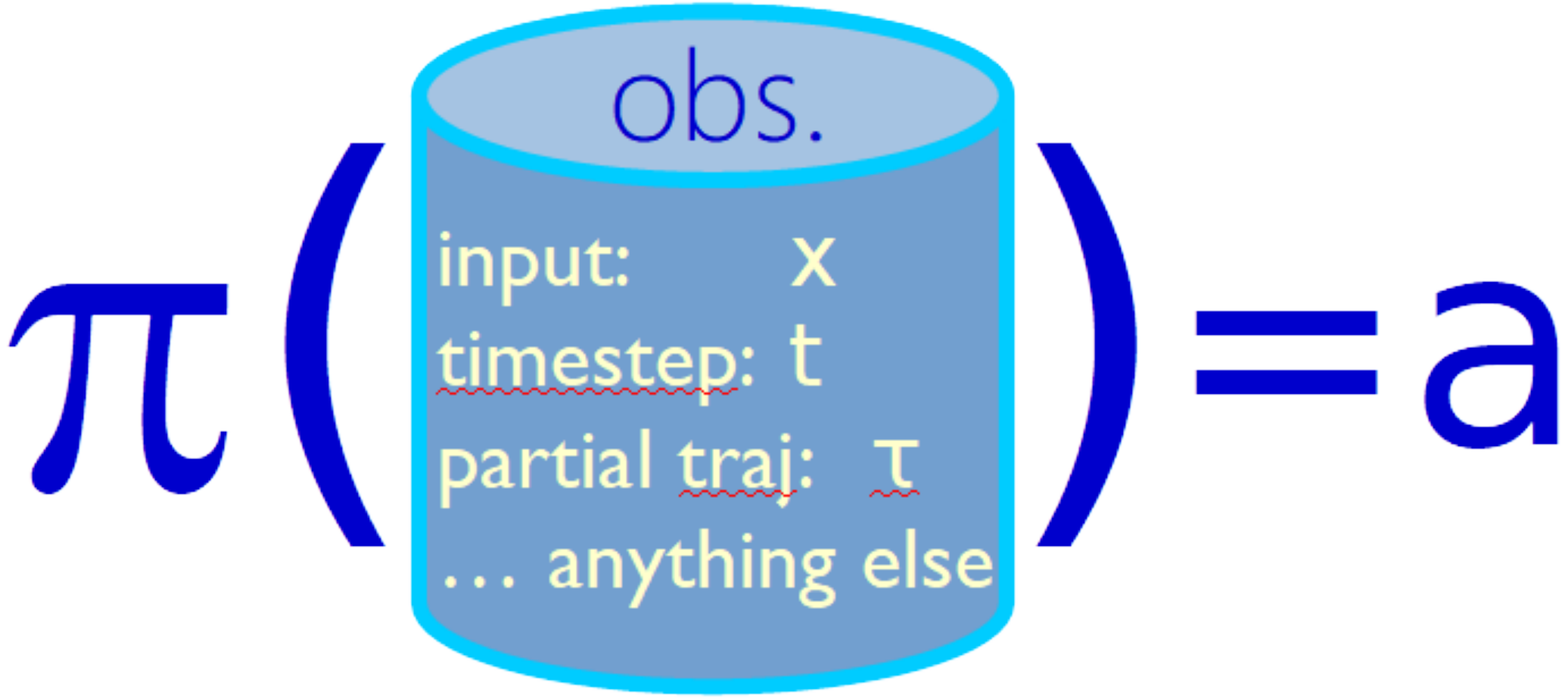
Today

- Learning from “external” reward signal
- Reinforcement learning paradigm
- Example algorithm: Policy gradient

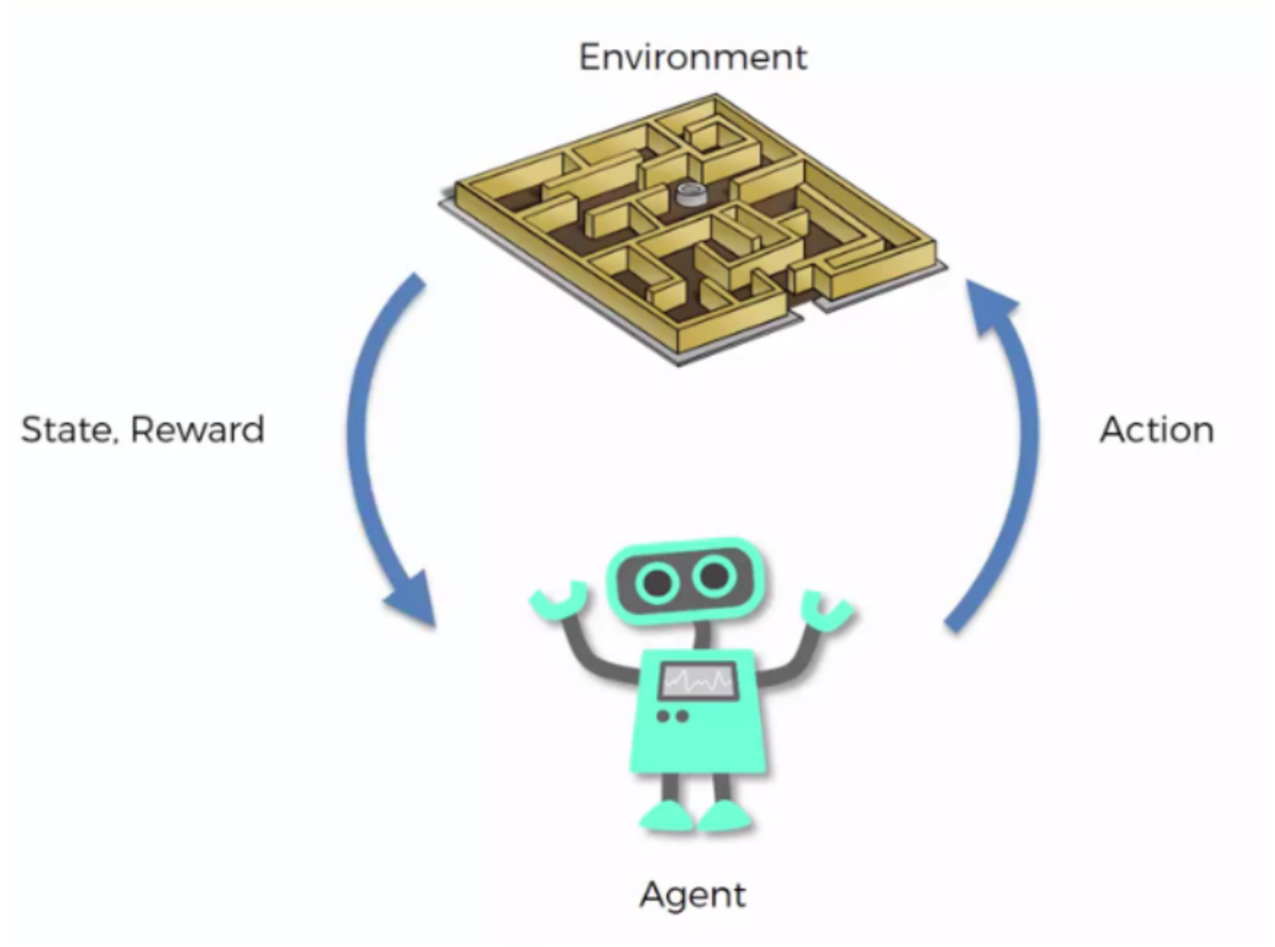
  (If time allows: Q-learning also)
- Connection(s) to imitation learning
- When (and when not) to RL

Policies

- A policy maps observations to actions



Sequential Decision-Making



Three problems of increasing complexity

Bandits

- for t = 1..T:
  - Choose $a_t \sim \pi$
  - Observe $r[a_t]$

Contextual Bandits

- for t = 1..T:
  - Observe o
  - Choose $a_t \sim \pi(o)$
  - Observe $r[a_t]$

Reinforcement Learning:

- for t = 1..T:
  - For h = 1..H:
    - Observe o_h
    - Choose $a_h \sim \pi(o_h)$
    - Observe $r[a_h]$
    - Transition to new state given a_h

Goal: Minimize regret

$$max_{\pi^*} \sum_t E_{a \sim \pi^*} r(a) \ - \sum_t r\left(a_t\right)$$

Challenges

- Bandits, CB, RL: exploration/exploitation
  (aka learning vs earning)

- RL: Credit assignment

Policy gradient for CB

- Assume pi has Boltzmann (“softmax”/Gibbs) form:
  pi(a|x) = (1/Z) exp[g(x,a)]
- CB:
  - Max_pi E_{x, r} E_{a ~ pi} r[a]
    = max_pi E_{x,r} sum_a pi(a) r[a]
    grad: trick…. d log(f(x))/dx = df/dx (1/f(x)) ➔ df/dx = f(x) d log(f(x)) / dx
    grad E_{x,r} sum_a pi(a) r[a]
    = E_{x,r} sum_a r[a] grad pi(a)
    = E_{x,r} sum_a r[a] pi(a) grad log pi(a)
    = E_{x,r} E_{a~pi} r[a] grad g(x,a)
  - Monte Carlo estimate:
    draw x ~ world
    draw a ~ pi(x)
    g <- g + 0.1 r(a) grad g

Policy gradient for CB, intuition

- g <- g + 0.1 r(a) grad g(x,a)
- If r is high, move g toward (x,a)
- If r is negative, move g away from (x,a)

… but what if r is always in [0,1]?

Policy gradient for CB with baseline

- Let r'(a) = r(a) – b for some fixed constant b
- Max_pi E_{x,r,a} r'(a)
  = Max_pi E_{x,r,a} [r(a) – b]
  = Max_pi E_{x,r,a} r(a) – const. → constant shift does not change optimum

- Useful heuristic: set b = average reward over past 100 updates
  (or some exponentially decaying running average)

- "Useful" == absolutely 100% necessary

Policy gradient for CB with baseline, intution

- g <- g + 0.1 (r(a) – avg_reward) grad g(x,a)

- If a is much better than you've been doing, move toward (x,a)
- If a is much worse than you've been doing, move away from (x,a)

Downside of policy gradient

- Very naïve exploration

- When pi(x) puts all probability on a single action, no exploration => no learning
- Under reasonable assumptions, optimal pi puts all probability on a single a

➔ As you get closer to the optimum, you get less and less learning signal

Some argue that it’s better to explore in “action space” than in probability space

Policy gradient for RL

- Max_pi E_{s1,a1,r1,s2,…,rH} sum_h rh
  = E_{s1,a1,r1,s2,…,rH} (sum_h rh) sum_h grad g(sh,ah)

- But *causality* implies rh only depends on previous actions, so…
  = E_{s1,a1,r1,s2,…,rH} sum_h (sum_{h'=h}^H rh) grad g(sh,ah)

- Monte Carlo estimate:

```
draw trajectory ~ world, policy
for h = 1 .. H:
    g <- g + 0.1 (tail reward from h) grad g(sh,ah)
```

- And then also add a baseline

Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning

Karthik Narasimhan
CSAIL, MIT
karthikn@mit.edu

Adam Yala
CSAIL, MIT
adamyala@mit.edu

Regina Barzilay
CSAIL, MIT
regina@csail.mit.edu

ShooterName: Scott Westerhuis
NumKilled: 6

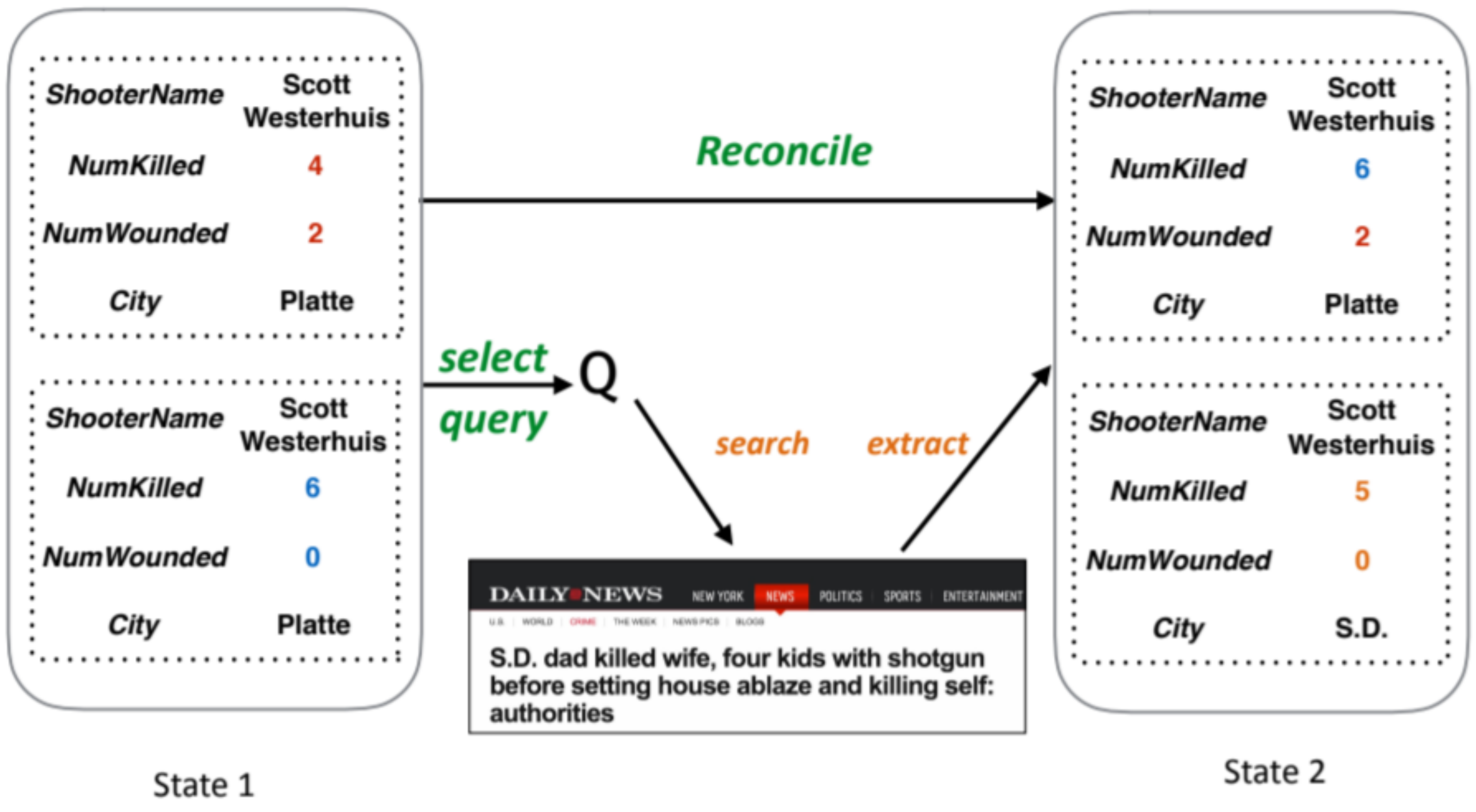
A couple and four children found dead in their burning South Dakota home had been shot in an apparent murder-suicide, officials said Monday.
...
Scott Westerhuis's cause of death was "shotgun wound with manner of death as suspected suicide," it added in a statement.

Figure 1: Sample news article on a shooting case. Note how the article contains both the name of the shooter and the number of people killed but both pieces of information require complex extraction schemes.

The **six** members of a South Dakota family found dead in the ruins of their burned home were fatally shot, with one death believed to be a suicide, authorities said Monday.

AG Jackley says all evidence supports the story he told based on preliminary findings back in September: **Scott Westerhuis** shot his wife and children with a shotgun, lit his house on fire with an accelerant, then shot himself with his shotgun.

Figure 2: Two other articles on the same shooting case. The first article clearly mentions that six people were killed. The second one portrays the shooter in an easily extractable form.



S.D. dad killed wife, four kids with shotgun before setting house ablaze and killing self: authorities

Current Values:
ShooterName $\rightarrow$ Scott Westerhuis
NumKilled $\rightarrow$ 4
NumWounded $\rightarrow$ 2
City $\rightarrow$ Platte

New Values:
ShooterName $\rightarrow$ Scott Westerhuis
NumKilled $\rightarrow$ 6
NumWounded $\rightarrow$ 0
City $\rightarrow$ Platte

State:

$\langle 0.3, 0.2, 0.5, 0.1,$	$\leftarrow$ currentConf
$0.4, 0.6, 0.2, 0.4,$	$\leftarrow$ newConf
$1, 0, 0, 1, 0, 1, 1, 0,$	$\leftarrow$ matches
$0.2, 0.3, ..., 0.1, 0.5,$	$\leftarrow$ contextWords
$0.65\rangle$	$\leftarrow$ document tf-idf similarity

Figure 3: Left: Illustration of a transition in the MDP – the top box in each state shows the current entities and the bottom one consists of the new entities extracted from a downloaded article on the same event. **Right:** Sample state representation (bottom) in the MDP based on current and new values of entities (top). *currentConf*: confidence scores of current entities, *newConf*: confidence scores of new entities, *contextWords*: tf-idf counts of context words.

States The state s in our MDP consists of the extractor's confidence in predicted entity values, the context from which the values are extracted and the similarity between the new document and the original one. We represent the state as a continuous real-valued vector (Figure 3) incorporating these pieces of information:

1. Confidence scores of current and newly extracted entity values.
2. One-hot encoding of matches between current and new values.
3. Unigram/tf-idf counts[2] of context words. These are words that occur in the neighborhood of the entity values in a document (e.g. the words *which*, *left*, *people* and *wounded* in the phrase "*which left 5 people wounded*").
4. *tf-idf* similarity between the original article and the new article.

Actions At each step, the agent is required to take two actions - a reconciliation decision d and a query choice q. The decision d on the newly extracted values can be one of the following types: (1) accept a specific entity's value (one action per entity)[3], (2) accept all entity values, (3) reject all values or (4) stop. In cases 1-3, the agent continues to inspect more articles, while the episode ends if a stop action (4) is chosen. The current values and confidence scores are simply updated with the accepted values and the corresponding confidences.[4] The choice q is used to choose the next query from a set of automatically generated alternatives (details below) in order to retrieve the next article.

$$R(s,a) = \sum_{\text{entity } j} \text{Acc}(e^j_{cur}) - \text{Acc}(e^j_{prev})$$

There is a negative reward per step to penalize the agent for longer episodes.

Queries The queries are based on automatically generated templates, created using the title of an article along with words[5] most likely to co-occur with each entity type in the training data. Table 1 provides some examples – for instance, the second template contains words such as *arrested* and *identified* which often appear around the name of the shooter.

⟨*title*⟩
⟨*title*⟩ + (police ∣ identified ∣ arrested ∣ charged)
⟨*title*⟩ + (killed ∣ shooting ∣ injured ∣ dead ∣ people)
⟨*title*⟩ + (injured ∣ wounded ∣ victim)
⟨*title*⟩ + (city ∣ county ∣ area)

Table 1: Examples of different query templates for web search for articles on mass shootings. The ∣ symbol represents logical OR. The last 4 queries contain context words around values for entity types ShooterName, NumKilled, NumWounded and City, respectively. At query time, ⟨*title*⟩ is replaced by the source article's title.

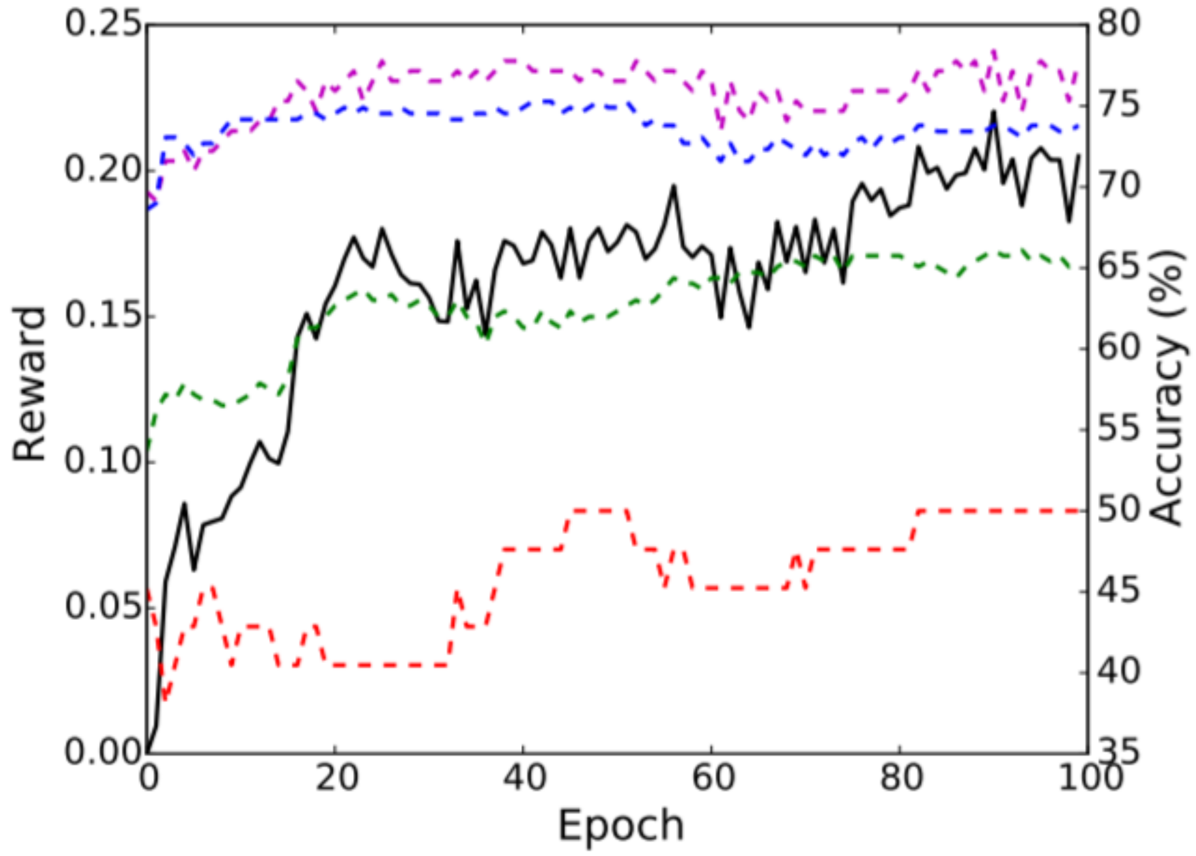


Figure 4: Evolution of average reward (solid black) and accuracy on various entities (dashed lines; red=*ShooterName*, magenta=*NumKilled*, blue=*NumWounded*, green=*City*) on the *test* set of the *Shootings* domain.

Reward shaping

$$R(s,a) = \sum_{\text{entity } j} \text{Acc}(e^j_{cur}) - \text{Acc}(e^j_{prev})$$

- Could alternatively have said:
  R(s,a) = 0 *if s is not a terminal state*
  = sum_j Acc(ejcur) *if s is terminal*

- But this would make learning really really hard!

- Proposition: optimal policy is unchanged if rewards are shifted by a "potential" function of states:

  r'(s,a,s') := r(s,a,s') + γF(s') - F(s')

- Optimal potential: F(s) := V(s) --- optimal cumulative reward from s

Encouraging more exploration

- Temperature on softmax; “temp” hyperparameter

  pi(alx) = (1/Z) exp[(1/temp) g(x,a)]

- Entropy regularization (aka “Soft Actor Critic”)

  Add regularization term to loss of form:

$$\sum_h H\Big(\pi(\,\cdot \mid s_h)\Big) = -\sum_h \sum_a \pi(a \mid s_h)\log\pi(a \mid s_h)$$

Different types of RL problems

- Is the environment known == do you have a simulator?
- Is the problem YOLO?
- Can I reset to given states?

Image credit: SnorgTees

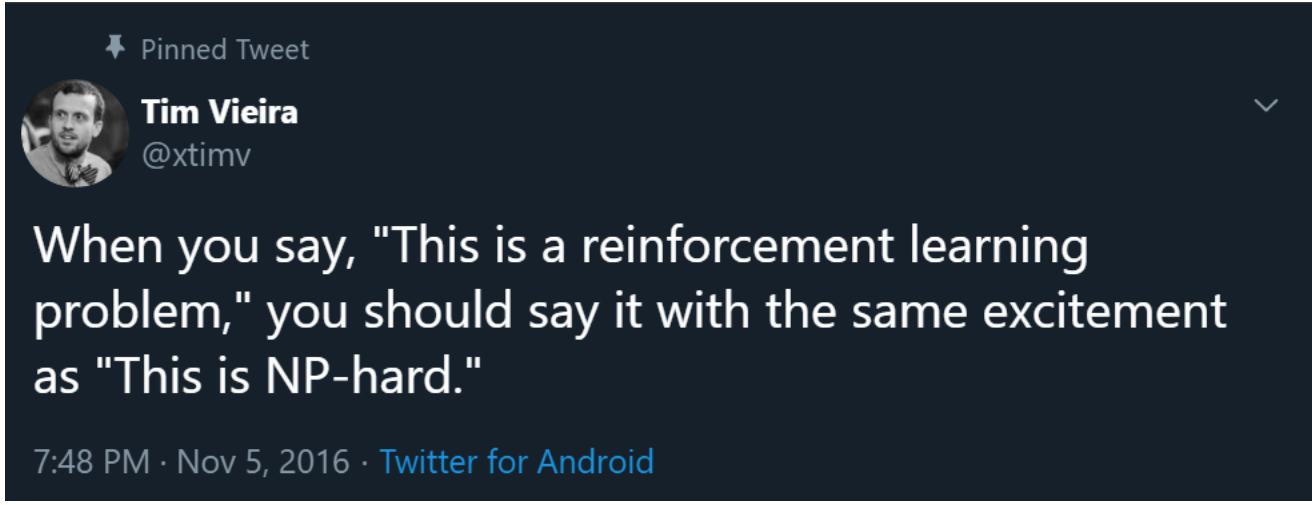
General advice

If you must RL:

- Do whatever you can to push the reward as close to the “causal” action as possible

- If you can push it entirely (or 90%), but still must explore, do CB

- If you’re in a simulator, or other non-YOLO setting, consider imitation


Pinned Tweet

Tim Vieira
@xtimv

When you say, "This is a reinforcement learning problem," you should say it with the same excitement as "This is NP-hard."

7:48 PM · Nov 5, 2016 · Twitter for Android

Today

- Learning from “external” reward signal
- Reinforcement learning paradigm
- Example algorithm: Policy gradient
  (If time allows: Q-learning also)
- Connection(s) to imitation learning
- When (and when not) to RL