Reinforcement Learning and Automated Planning

Class 5: Model-Based Reinforcement Learning

Ronen I. Brafman

Department of Computer Science

אוניברסיטת בן-גוריון בנגב
جامعة بن غوريون في النقب
Ben-Gurion University of the Negev

Model-Based RL

We know how to solve a given model. What if we don't have one?

- Two general approaches:
  - Model-Based RL: Learn a model and solve it
  - Model-Free RL: Learn a policy directly

- Model-Free RL is more popular because:
  - Large models can be complicated to learn and store
  - Large models can be very difficult to solve
- Model-Based RL usually requires fewer samples
  - This is very important in real-world problems (why?)
- Model-Based RL can *sometimes* generalize better, be more robust, and more explainable

Markov Decision Process Definition

Markov Decision Process

$M = \langle S, A, Tr, R \rangle$

- S is a set of states
- A is a set of actions
- $Tr : S \times A \rightarrow \Pi(S)$ is a stochastic transition function
- $R : S \times A \rightarrow \mathbb{R}$ is the reward function

- We assume we know S and A
- We need to learn Tr and R

Learning R and TR

Learning R

Learning a deterministic function is easy:

If you receive r when applying a in s then $R(s,a) = r$

Learning Tr

- We need to assess $Tr(s,a,s')$ for every s,a,s'
- $Tr(s,a)$ is a stochastic function – each time we execute a in s we can get a different outcome
- According to the frequentist definition of probability $Tr(s,a,s') = \lim_{n(s,a)\to\infty} \frac{n(s,a,s')}{\sum_{s'\in S} n(s,a,s')} = \lim_{n(s,a)\to\infty} \frac{n(s,a,s')}{n(s,a)}$
- $\frac{n(s,a,s')}{n(s,a)}$ is the *maximum likelihood* estimate (MLE) of $Tr(s,a,s')$

How Accurate is the Model?

- Assessment of R is accurate (assuming it is deterministic)
- Assessment of Tr is noisy:
  - If we twice execute a in s n times, we are likely to get different results
  - Think of it as tossing a biased coin n
  - More accurately: throwing a biased dice with $|S|$ sides

How Accurate is the Model?

- Suppose $Tr(s, a, s') = p$
- Consider a random variable X: its value is 1 when applying a in s results in s', and 0 otherwise
- $Pr(X = 1) = p, Pr(X = 0) = 1 - p$ and $E(X) = p$
- We can associate n attempts to apply a in s with n random variables $X_1, \ldots, X_n$ that are independent and identically distributed (IID) as X
- We write $S_n = X_1 + \cdots + X_n$
- So $E(S_n) = E(X_1) + \cdots + E(X_n) = np$

How Accurate is the Model?

- We can estimate p using its MLE: $\hat{p} = \frac{S_n}{n} = \frac{n(s,a,s')}{n}$
- Our estimation error is $|\frac{S_n - E(S_n)}{n}| = |p - \hat{p}|$
- Hoeffding's Inequality tells us that:

$$Pr(|S_n - E(S_n)| \geq \epsilon) \leq 2\exp -\frac{2\epsilon^2}{n}$$

- So

$$Pr(|\frac{S_n - E(S_n)}{n}| \geq \epsilon) = Pr(|S_n - E(S_n)| \geq n\epsilon) \leq 2\exp\left(-2n\epsilon^2\right)$$

Error Bounds

$$Pr\left(\left|\frac{S_n - E(S_n)}{n}\right| \geq \epsilon\right) = Pr(|S_n - E(S_n)| \geq n\epsilon) \leq 2\exp\left(-2n\epsilon^2\right)$$

- For an error of less than ϵ with probability at least $1 - \delta$
  We need $n = n(s, a)$ to be large enough so that:

$$2\exp\left(-2n(s, a)\epsilon^2\right) \leq \delta$$

- This gives us:

$$n(s, a) \geq \frac{\log(2/\delta)}{2\epsilon^2}$$

Is This Enough?

- To get a good model, we just need to try $n(s, a)$ enough times
- Is that it?

Is This Enough?

- To get a good model, we just need to try $n(s, a)$ enough times
- This assumes we can get to every state s as often as we want
    - But we don't usually get to be in whatever state we want
    - It may take many trials to get to some states
- This assumes we want an accurate model on all states
    - But usually, what we want is a good policy
    - And quickly – without performing too many actions
- The number of trials required to get a good policy with an algorithm is called its *sample complexity*
- We usually want algorithms with low sample complexity

Getting a (Near) Optimal Policy Quickly

- In the limit, if we can visit every state infinitely many times, we will have an accurate model and therefore an optimal policy
- We say policy π is ϵ-optimal if for every state s (or for s_0):

$$v^*(s) - v^\pi(s) \leq \epsilon$$

- Can we always get an ϵ-optimal policy "quickly"?
  - There is always a small probability that we will make very rare observations that will cause us to learn a bad model
- We want an ϵ-optimal policy "quickly" with high probability
- This is called *PAC – probably approximately correct*, or in our case probably approximately optimal

1. Estimating Means
2. R_{max}
3. Estimating Expectations

R_{max}

R_{max} returns a policy that is ϵ-optimal with probability $1-\delta$ in time polynomial in $\epsilon, 1/\delta$, and $|M|$.

R_{max}

Require: States S, Actions A

M $=$ InitializeModelAndCounts
loop
    $\pi =$ SolveModel(M)
    while $known(s,a)$ unchanged for all s, a **do**
        Execute one step of policy π obtaining (s, a, s', r)
        $n(s,a)++; n(s,a,s')++, R(s,a) = r$
        If $n(s,a)$ reached k : set $known(s,a) = true$
    end while
    $M =$ Update(M)
end loop

R_{max}

Define: $r_{max} = \max_{s,a} R(s,a)$

Initialize Model

1: $M = \langle S \cup \{\hat{s}\}, A, Tr, R \rangle$
2: $\forall s, a:\ \ Tr(s,a,\hat{s}) = 1, R(\hat{s},a) = r_{max};$
3: $\forall s \neq \hat{s}, a:\ \ R(s,a) = 0$
4: $\forall s, a, s':\ \ n(s,a) = n(s,a,s') = 0$
5: $\forall s, a:\ \ known(s,a) = false$

Update Model

1: **for all** s, a such that $known(s,a) = true$ **do**
2: $\quad Tr(s,a,s') = \frac{n(s,a)}{n(s,a,s'}$
3: **end for**

1. Estimating Means
2. R_{max}
3. Estimating Expectations

Monte Carlo Expectations

- Often, we'll want to estimate the expected value of some function f

$$y = \mathbb{E}_{x \sim \pi}[f(x)]$$

- Monte-Carlo Estimate of y:
  - Sample $x_1, \ldots, x_m$ from π
  - Compute $\hat{y}$ of y:

$$\hat{y} = \frac{1}{m} \sum_{i=1}^{m} f(x_i)$$

- Incremental computation:
  - $\hat{y}_1 = f(x_1)$
  - $\hat{y}_k = \frac{(k-1)}{k} \cdot f(x_{k-1}) + \frac{f(x_k)}{k}$

Bias and Variance

- $\hat{y}$ is a random variable: it depends on $x_1, \ldots, x_m$ which is a random sample
- *Bias* of an estimator y' of y is $||\mathbb{E}[y'] - y||$
- The Monte-Carlo estimator $\hat{y}$ of y is unbiased
- The variance of an estimator $\hat{y}$ is

$$\mathbb{E}[(\hat{y} - \mathbb{E}\hat{y})^2]$$

Importance Sampling

- Sometimes we cannot sample from distribution π
- We can estimate $\mathbb{E}_{x\sim\pi}[f(x)]$ using a different distribution π'

$$y = \mathbb{E}_{x\sim\pi}[f(x)] = \int \pi(x) f(x) dx$$

$$= \int \frac{\pi'(x)}{\pi'(x)} \pi(x) f(x) dx = \int \pi'(x) \frac{\pi(x)}{\pi'(x)} f(x) dx = \mathbb{E}_{x\sim\pi'}[\frac{\pi(x)}{\pi'(x)} f(x)]$$

- We can estimate y by sampling $x_1, \ldots, x_m$ from π' and computing

$$\hat{y} = \frac{1}{m} \sum_{i=1}^{m} \frac{\pi(x_i)}{\pi'(x_i)} f(x_i)$$

- π' is called the *proposal* distribution