EL2805, Exercise 5 (Learning MDPs II)

Robert Mattila

Previously, we saw that one approach to learn to control an MDP was to estimate the q-function. Now, we'll directly try to find $\pi^*(s)$.

Today's central concepts:

- Randomized policy (in order to explore):

$$\pi(s,a) = \Pr\{a_t = a \mid s_t = s\}$$

(If $\mathcal{A}$ discrete, otherwise pdf.)

- Parametrized policy $\pi_\theta(s,a)$

  - Common choice:

$$\pi_\theta(s,a) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}, \qquad \text{(soft max)}$$

where $h(s,a,\theta)$ are action-preferences.

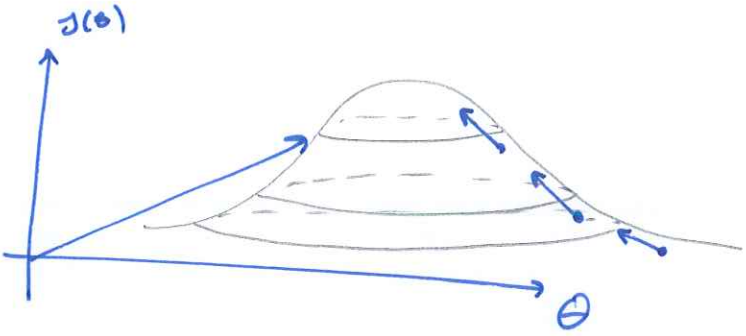
the higher, the more we prefer action a in state s.

- We aim to find parameters θ that maximizes an objective. For example,

$$J(\theta) = \underbrace{\mathbb{E}_{\pi_\theta}}_{\mathbb{E}\text{ under policy } \pi_\theta}\Big\{ \underbrace{\sum_{t=1}^{T} r(s_t,a_t)}_{\text{total reward}} \;\Big|\; \underbrace{s_1}_{\text{assume we know where we start in an episode}} \Big\}$$

in a finite-horizon MDP.

- <u>Idea</u>: Use gradient ascent to maximize $J(\theta)$ w.r.t. θ.



For this, we need the gradient $\nabla J(\theta)$:

- <u>Policy gradient theorem</u>: (Finite-horizon MDP)

$$\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left\{\left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=1}^{T} r(s_t, a_t)\right)\right\}$$

$$= \{\text{see exercise } 5.6\}$$

$$= \mathbb{E}_{\pi_\theta}\left\{\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t) \sum_{u=t}^{T} r(s_u, a_u)\right\}$$

- <u>Problem (?)</u>: In the RL setting the transition probabilities and rewards are unknown, so we cannot compute $\nabla J(\theta)$ with this expression.
- <u>Solution</u>: Sample a trajectory and use

$$\hat{\nabla} J(\theta) = \sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t) \sum_{u=t}^{T} r(s_u, a_u)$$

as an unbiased estimator of $\nabla J(\theta)$.

$\Longrightarrow$ A stochastic gradient ascent algorithm to maximize $J(\theta)$, <u>REINFORCE</u>.

- One step (parameter update) in REINFORCE is:

$$\underbrace{\theta}_{\text{new parameters}} \leftarrow \underbrace{\theta}_{\text{old parameters}} + \underbrace{\alpha}_{\text{step-size}} \underbrace{\sum_{t=1}^{T}}_{\substack{\text{sum over}\\ \text{all steps in}\\ \text{the episode}}} \left\{ \underbrace{\nabla \log \pi_\theta(s_t, a_t) \sum_{u=t}^{T} r(s_u, a_u)}_{\substack{\text{correction}\\ \text{vector}}} \right\}$$

- Interpretation of the correction vector:

$$\nabla \log \pi_\theta(s_t, a_t) \sum_{u=t}^{T} r(s_u, a_u) =$$

$$\frac{\nabla \pi_\theta(s_t, a_t)}{\pi_\theta(s_t, a_t)} \sum_{u=t}^{T} r(s_u, a_u)$$

- $\nabla \pi_\theta(s_t, a_t)$: the update is in the direction in parameter space that most increases the probability of playing action a_t in state s_t.

- $\sum_{u=t}^{T} r(s_u, a_u)$: the size of the update is proportional to the reward-to-go we observed from this state-action pair.
  (The higher reward, the more we will increase the probability of this action in this state.)

- $\frac{1}{\pi_\theta(s_t, a_t)}$: we penalize actions with high probability since they are played more often (and hence the parameters are updated more often).
  in direction of being played more often.

Note:

- If we did not scale by $1/\pi_\theta(s,a)$, then "common" action could "win" despite having lower reward solely by being played often.

- This is a Monte-Carlo method since we need to wait for an episode to finish before we update the parameters (the policy). We need the $\underbrace{\text{rewards-to-go}}_{\text{future}}$ from each state: $\sum_{u=t}^{T} r(s_u, a_u)$.

(- log denotes the natural logarithm (base e))

Ex. 5.3

Consider a soft-max policy parametrization

$$\pi_\theta(s,a) = \frac{e^{h(s,a,\theta)}}{\sum_b' e^{h(s,b,\theta)}}$$

with linear action preferences

$$h(s,a,\theta) = \theta^T x(s,a),$$

where $x(s,a)$ is a feature vector.

Find the eligibility vector $\nabla \log \pi_\theta(s,a)$.

Solution:

we have that:

$$\nabla \log \pi_\theta(s,a) = \nabla \log \left\{ \frac{e^{h(s,a,\theta)}}{\sum_b' e^{h(s,b,\theta)}} \right\} =$$

$$= \nabla \log e^{h(s,a,\theta)} - \nabla \log \left\{ \sum_b' e^{h(s,b,\theta)} \right\} =$$

$$= \nabla h(s,a,\theta) - \frac{1}{\sum_b' e^{h(s,b,\theta)}} \nabla \left\{ \sum_c' e^{h(s,c,\theta)} \right\} =$$

$$= \underbrace{\nabla \theta^T x(s,a)}_{= x(s,a)} - \frac{1}{\sum_b' e^{h(s,b,\theta)}} \sum_c' \nabla e^{h(s,c,\theta)} = \left\{ \text{Recall: } \frac{\partial e^{f(\theta)}}{\partial \theta_i} = \frac{\partial e^{f(\theta)}}{\partial f} \frac{\partial f(\theta)}{\partial \theta_i} \right\}$$

$$= x(s,a) - \frac{1}{\sum_b' e^{h(s,b,\theta)}} \sum_c' e^{h(s,c,\theta)} \underbrace{\nabla h(s,c,\theta)}_{\substack{= \nabla \theta^T x(s,c) \\ = x(s,c)}}$$

$$= x(s,a) - \frac{1}{\sum_b' e^{h(s,b,\theta)}} \sum_c' e^{h(s,c,\theta)} \, x(s,c)$$

$$= x(s,a) - \sum_c' \left\{ \underbrace{\frac{e^{h(s,c,\theta)}}{\sum_b' e^{h(s,b,\theta)}}}_{\overset{\text{def.}}{=} \pi_\theta(s,c)} \right\} x(s,c)$$

$$= x(s,a) - \sum_c' \pi_\theta(s,c) \, x(s,c) .$$

Ex 5.5

Consider rock-paper-scissor, where opponent plays iid according to the distribution $\mu = (\mu_R, \mu_P, \mu_S)$.

a, Assume μ is known and model as MDP.

Now assume μ is unknown.

b, Propose a policy parametrization.

c, Explain how we can learn to play with REINFORCE.

Solution:

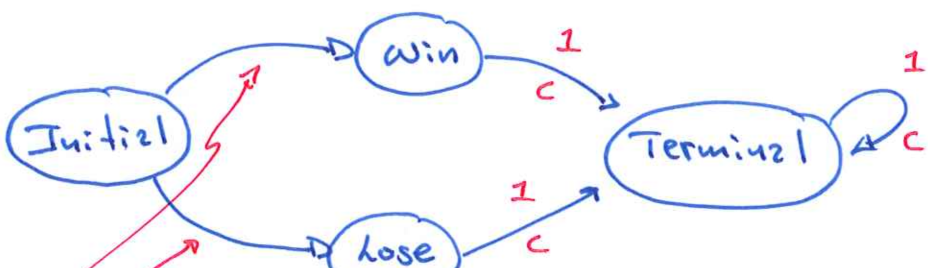
State-space: $\mathcal{S} = \{\text{Initial}, \text{Win}, \text{Lose}, \text{Terminal}\}$

Actions:
- $\mathcal{A}(\text{Initial}) = \{\text{Rock}, \text{Paper}, \text{Scissors}\}$
$= \{R, P, S\}$
- $\mathcal{A}(\text{Win}, \text{Lose}, \text{Terminal}) = \{\text{Continue}\} = \{C\}$

Rewards:
- $r(s = \text{Win}, a = C) = 1$
- all other zero

only reward when we win

Transitions:



probability depends on our action and what the opponent played

this state represents "not win", i.e., lose or draw

- $p(s' = \text{Win} \mid s = \text{Initial}, a = R) = \mu_S$ ← We play Rock and opponent scissors
- $p(s' = \text{Lose} \mid s = \text{Initial}, a = R) = 1 - \mu_S$ ← We play Rock and opponent Rock or Paper

- $p(s' = \text{Win} \mid s = \text{Initial}, a = S) = \mu_P$
- $p(s' = \text{Lose} \mid s = \text{Initial}, a = S) = 1 - \mu_P$

- $p(s' = \text{Win} \mid s = \text{Initial}, a = P) = \mu_R$
- $p(s' = \text{Lose} \mid s = \text{Initial}, a = P) = 1 - \mu_R$

- $p(s' = \text{Terminal} \mid s \in \{\text{Win}, \text{Lose}, \text{Terminal}\}, a = C) = 1$

Time-horizon and objective:

Finite-horizon, $T = 3$:

$$\mathbb{E}\left\{ \sum_{t=1}^{T} r(s_t, a_t) \;\middle|\; s_1 = \text{Initial} \right\}$$

The system always starts in Initial

b) In the discrete setting, it is common to use a soft-max parametrization:

$$\pi_\theta(s,a) = \begin{cases} \dfrac{e^{h(s,a,\theta)}}{\sum_{b\in\mathcal{A}(s)} e^{h(s,b,\theta)}} & \text{if } s = \text{Initial}, \\ & a \in \mathcal{A}(\text{Initial}), \\ 1 & \text{otherwise}, \end{cases}$$

always chose C_1 in the other states

where $h(s,a,\theta)$ are action preferences.

Since we only have one action, C_1, to chose from in Win, Lose and Terminal, we don't need to parametrize the policy there.

Note that the soft-max ensures that:

Recall: $\pi_\theta(s,a) = \Pr\{a_t = a \mid s_t = s\}$

i) $\pi_\theta(s,a) \geq 0$

ii) $\sum_a \pi_\theta(s,a) = 1$

i.e., that $\pi_\theta(s,a)$ represent probabilities (pmf)

iii) actions with higher preference have higher probabilities of being chosen.

(Note: Actually i) is $\pi_\theta(s,a) > 0$: we always explore!)

Next, we need to define the action preferences.

The simplest choice is linear:

$$h(s, a, \theta) = \theta^T x(s, a),$$

where $x(s, a)$ is a feature vector.

<u>Note:</u>

Many other choices are available, for example neural networks — see chap. 9 of Sutton's book.

We try to find the element $\pi_{\theta^*}(s, a)$ in the class of parametrized functions $\pi_\theta(s, a)$ that is closest to $\pi^*(s)$.

The choice of parametrization is a trade-off between $\underbrace{\text{computational complexity}}_{\text{of tuning the parameters } \theta.}$ and $\underbrace{\text{expressibility}}_{\text{being able to approximate } \pi^*(s)}$.

(there can be problems computing gradients and/or with local optima, etc.)

The feature vector $x(s,a)$ is a numerical encoding of the (potentially abstract) state-action pair, and/or its properties.

We only need to define it for $s=$ Initial and $a \in \{R, S, P\}$. A simple encoding of a categorical variable is a one-hot encoding:

$$x(s=\text{Initial}, a) = \begin{bmatrix} \mathbb{I}\{a=R\} \\ \mathbb{I}\{a=S\} \\ \mathbb{I}\{a=P\} \end{bmatrix},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

Note that $\theta \in \mathbb{R}^3$ in this case.

c) In REINFORCE, we first select initial parameters $\theta^{(0)}$ arbitrarily.

We then:

i) Play the game (i.e., generate a trajectory/episode) under the policy induced by $\theta^{(k)}$.

ii) Update the policy (parameters):

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \sum_{t=1}^{T} \left\{ \underbrace{\nabla \log \pi_\theta(s_{t,k}, a_{t,k})}_{(*)} \left(\underbrace{\sum_{u=t}^{T} r(s_{u,k}, a_{u,k})}_{\text{observed reward-to-go.}} \right) \right\},$$

where $s_{t,k}$ is the state at time t in episode k (the current).

iii) Repeat to i).

(*): Note that we computed the expression for the eligibility vector for this choice (linear) in Ex. 5.3:

$$\nabla \log \pi_\theta(s_{t,k}, a_{t,k}) = x(s_{t,k}, a_{t,k}) - \sum_{b} \pi_\theta(s_{t,k}, a_{t,k})\, x(s_{t,k}, b),$$

when $s_{t,k}$ = Initial, and that

$$\nabla \log \pi_\theta(s_{t,k}, a_{t,k}) = 0$$

when $s_{t,k} \in \{\text{Win, Lose, Terminal}\}$.

$\nabla \log \pi_\theta(\text{Lose}, C) =$
$\nabla \log 1 = 0,$
etc.

Note:

The setup in this exercise is fundamentally different from Ex. 5.1 (and the example in the lectures):

There, we observed what the opponent played in order to compute $\hat{\mu}_R$ etc., and then use these to find $\hat{\nabla J}(\theta)$.

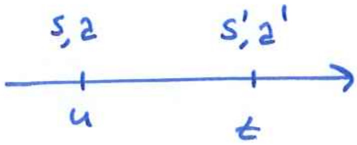
Here, we do not need to see what the opponent played: only whether we won or not.

Ex 5.6 Assume $u < t$ and show that

$$\mathbb{E}_{\pi_\theta}\{\nabla \log \pi_\theta(s_t, a_t)\, r(s_u, a_u)\} = 0.$$

Solution (proof):

we have that:

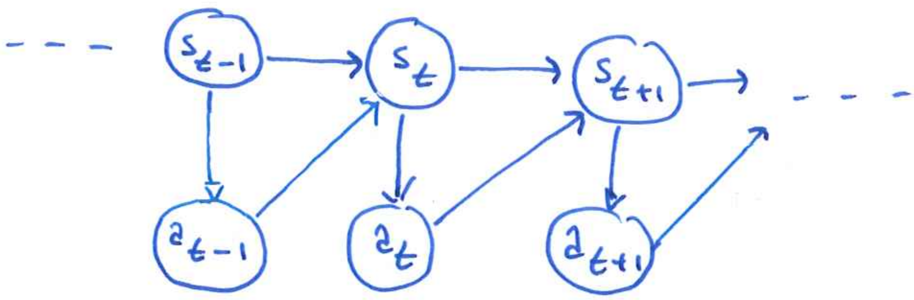


$$\mathbb{E}_{\pi_\theta}\{\nabla \log \pi_\theta(s_t, a_t)\, r(s_u, a_u)\} =$$

$$\sum_{s,a}\sum_{s',a'} \Pr\{s_t = s', a_t = a', s_u = s, a_u = a\}\, r(s,a)\, \nabla \log \pi_\theta(s', a') =$$

$$\sum_{s,a}\sum_{s',a'} \Pr\{a_t = a' \mid s_t = s', s_u = s, a_u = a\}\, \Pr\{s_t = s', s_u = s, a_u = a\}$$
$$\times\, r(s,a)\, \nabla \log \pi_\theta(s', a') =$$

Recall that both the system and the policy are Markovian:



Given the current state, the action we will apply does not depend on previous states or actions. That is:

$$\Pr\{a_t = a' \mid s_t = s', \underbrace{s_u = s, a_u = a}_{\text{old state and action since } u < t}\} = \Pr\{a_t = a' \mid s_t = s'\}$$

$$\sum_{s,a}{}' \sum_{s',a'}{}' \underbrace{\Pr\{a_t = a' \mid s_t = s'\}}_{\overset{\text{def.}}{=} \pi_\theta(s', a')} \Pr\{s_t = s', s_u = s, a_u = a\} r(s,a) \nabla \log \pi_\theta(s', a') =$$

$$= \{\text{shift terms around}\} =$$

$$\sum_{s,a}{}' \sum_{s'}{}' \Pr\{s_t = s', s_u = s, a_u = a\} r(s,a) \underbrace{\sum_{a'}{}' \pi_\theta(s', a') \nabla \log \pi_\theta(s', a')}_{= 0} = 0.$$

why is the last factor zero?

$$\sum_{a'}{}' \pi_\theta(s', a') \nabla \log \pi_\theta(s', a') =$$

$$\sum_{a'}{}' \cancel{\pi_\theta(s', a')} \frac{\nabla \pi_\theta(s', a')}{\cancel{\pi_\theta(s', a')}} =$$

$$\sum_{a'}{}' \nabla \pi_\theta(s', a') =$$

$$\nabla \underbrace{\sum_{a'}{}' \pi_\theta(s', a')}_{= 1} = \left\{ \begin{array}{l} \text{Recall that} \\ \pi_\theta(s', a') = \Pr\{a_t = a' \mid s_t = s'\} \end{array} \right\} =$$

$$\nabla 1 =$$

$$0.$$

Ex 5.7c | Policy gradient theorem, discounted episodic

Notation as in Sutton's book. See p. 328, exercise 13.2 for details.

Proof:

For any $s \in \mathcal{S}$ (not including the terminal state):

$$\nabla_\theta v_{\pi_\theta}(s) = \nabla_\theta \Big[\underbrace{\sum_a{}' \pi_\theta(a|s)\, q_{\pi_\theta}(s,a)}_{\cancel{\mathbb{E}}\{ q_{\pi_\theta}(s,A) \mid A \sim \pi_\theta \}} \Big]$$

$$= \sum_a{}' \Big[\nabla_\theta \pi_\theta(a|s)\, q_{\pi_\theta}(s,a) + \pi_\theta(a|s)\, \nabla_\theta q_{\pi_\theta}(s,a) \Big]$$

$$= \sum_a{}' \Big[\nabla_\theta \pi_\theta(a|s)\, q_{\pi_\theta}(s,a) + \pi_\theta(a|s)\, \nabla_\theta \Big\{ r(s,a) + \lambda \sum_{s'}{}' p(s'|s,a)\, v_{\pi_\theta}(s') \Big\} \Big]$$

$$/ = \nabla_\theta r(s,a) = 0 \; /$$

$$= \sum_a{}' \Big[\nabla_\theta \pi_\theta(a|s)\, q_{\pi_\theta}(s,a) + \pi_\theta(a|s)\, \lambda \sum_{s'}{}' p(s'|s,a)\, \nabla_\theta v_{\pi_\theta}(s') \Big]$$

start unrolling

$$= \sum_a{}' \Big[\nabla_\theta \pi_\theta(a|s)\, q_{\pi_\theta}(s,a) + \pi_\theta(a|s)\, \lambda \sum_{s'}{}' p(s'|s,a) \times \qquad (*)$$

$$\sum_{a'}{}' \Big\{ \nabla_\theta \pi_\theta(a'|s')\, q_{\pi_\theta}(s',a') + \pi_\theta(a'|s')\, \lambda \sum_{s''}{}' p(s''|s',a')\, \nabla_\theta v_{\pi_\theta}(s'') \Big\}$$

Regroup the terms

$$= \underbrace{1}_{= \sum_{s'}{}' \lambda^0 \Pr\{s \to s', 0, \pi_\theta\}} \cdot \Big\{ \sum_a{}' \nabla_\theta \pi_\theta(a|s)\, q_{\pi_\theta}(s,a) \Big\} +$$

$$\sum_{s'}{}' \lambda \underbrace{\sum_a{}' \pi_\theta(a|s)\, p(s'|s,a)}_{= \Pr\{s \to s', 1, \pi_\theta\}} \Big\{ \sum_{a'}{}' \nabla_\theta \pi_\theta(a'|s')\, q_{\pi_\theta}(s',a') \Big\} +$$

$$\sum_{s''}{}' \lambda^2 \underbrace{\sum_{s'}{}' \sum_{a'}{}' \pi_\theta(a|s)\, p(s'|s,a)\, \pi_\theta(a'|s')\, p(s''|s',a')}_{= \Pr\{s \to s'', 2, \pi_\theta\}} \nabla_\theta v_{\pi_\theta}(s'')$$

Change variables in the sums. Should probably use x instead.

the sum is only non-zero for $s' = s$

$$= \sum_{s'}{}' \lambda^0 \Pr\{s \to s', 0, \pi_\theta\} \Big[\sum_a{}' \nabla_\theta \pi_\theta(a|s')\, q_{\pi_\theta}(s',a) \Big] +$$

$$\sum_{s'}{}' \lambda^1 \Pr\{s \to s', 1, \pi_\theta\} \Big[\sum_a{}' \nabla_\theta \pi_\theta(a|s')\, q_{\pi_\theta}(s',a) \Big] +$$

$$\sum_{s'}{}' \lambda^2 \Pr\{s \to s', 2, \pi_\theta\}\, \nabla_\theta v_{\pi_\theta}(s')$$

Now, the first term when expanding $\nabla_\theta v_{\pi_\theta}(s')$ will be $\{\sum_a' \nabla_\theta \pi_\theta(a|s') q_{\pi_\theta}(s',a)\}$ by (*).

In general, if we continue unrolling, we obtain

$$= \sum_x' \gamma^0 \Pr\{s \to x, 0, \pi_\theta\} \left[\sum_a' \nabla_\theta \pi_\theta(a|x) q_{\pi_\theta}(x,a)\right] +$$

$$\sum_x' \gamma^1 \Pr\{s \to x, 1, \pi_\theta\} [\text{———} \; '' \; \text{———}] +$$

$$\sum_x' \gamma^2 \Pr\{s \to x, 2, \pi_\theta\} [\text{———} \; '' \; \text{———}] +$$

$$\vdots$$

$$\sum_x' \gamma^t \Pr\{s \to x, t, \pi_\theta\} [\text{———} \; '' \; \text{———}] +$$

$$\vdots$$

$$= \sum_x' \sum_{k=0}^{\infty}{}' \gamma^k \Pr\{s \to x, k, \pi_\theta\} \left[\sum_a' \nabla_\theta \pi_\theta(a|x) q_{\pi_\theta}(x,a)\right]$$

Note: we sum over S', not S'^{+} (with terminal state), so for some N all $k > N$ terms will be zero. Or at least they will tend to zero. This is the probability of transitioning between s and x in one episode. After the episode is over, this probability will be zero.

not terminal

the T when this happens can be random however.

Define the <u>discounted state distribution</u> $\mu_\theta^\lambda(s)$:

$$\mu_\theta^\lambda(s) = (1-\lambda)\sum_{k=0}^{\infty}\lambda^k \Pr\{s_0 \to s, k, \pi_\theta\}$$
$$= (1-\lambda)\sum_{k=0}^{\infty}\lambda^k \Pr\{S_k = s \mid S_0 = s_0, \pi_\theta\}.$$

Assume every episode starts in s_0.
(Can always add an initial state)

Then:

$$\nabla J(\theta) = \nabla v_\pi(s_0)$$
$$= \sum_x \left\{\sum_{k=0}^{\infty}\lambda^k \Pr\{s \to x, k, \pi_\theta\}\right\}\left[\sum_a \nabla_\theta \pi_\theta(a|x)\, q_{\pi_\theta}(x,a)\right]$$
$$= \sum_x \frac{1}{1-\lambda}\mu_\theta^\lambda(x)\left[\sum_a \nabla_\theta \pi_\theta(a|x)\, q_{\pi_\theta}(x,a)\right]$$
$$\propto \sum_x \mu_\theta^\lambda(x)\left[\sum_a \nabla_\theta \pi_\theta(a|x)\, q_{\pi_\theta}(x,a)\right]$$

<u>Question:</u>

How to obtain state samples from $\mu_\theta^\lambda(x)$?

<u>Answer:</u>

Thomas (2014):

a, Convert the discounted MDP to an undiscounted by w.p. λ terminating each trajectory (and compute undiscounted reward-sums).
May disregard a lot of data.

b, Use the algorithm proposed in Thomas (2014).