6. Q-learning with linear function approximation

Extension Note: Project 5 due date has been extended by 1 **more** day to **September 6 23:59UTC** .

Since the state displayed to the agent is described in text, we have to choose a mechanism that maps text descriptions into vector representations. A naive way is to create one unique index for each text description, as we have done in previous part. However, such approach becomes infeasible when the state space becomes huge. To tackle this challenge, we can design some representation generator that does not scale as the original textual state space. In particular, a representation generator $\psi_R(\cdot)$ reads raw text displayed to the agent and converts it to a vector representation $v_s = \psi_R(s)$. One approach is to use a bag-of-words representation derived from the text description.

In large games, it is often impractical to maintain the Q-value for all possible state-action pairs. One solution to this problem is to approximate $Q(s, c)$ using a parametrized function $Q(s, c; \theta)$.

In this section we consider a linear parametric architecture:

$$Q\left(s,c;\theta\right)=\phi(s,c)^{T}\theta=\sum_{i=1}^{d}\phi_{i}\left(s,c\right)\theta_{i},$$

where $\phi\left(s,c\right)$ is a fixed feature vector in $\mathbb{R}^{d}$ for state-action pair $\left(s,c\right)$ with i-th component given by $\phi_{i}\left(s,c\right)$, and $\theta\in\mathbb{R}^{d}$ is a parameter vector that is shared across state-action pairs. The key challenge here is to design the feature vectors $\phi\left(s,c\right)$. Note that given a textual state s, we first translate it to a vector representation v_{s} using $\psi_{R}\left(s\right)$. So the question here is how to design a mapping function convert $\left(\psi_{R}\left(s\right),c\right)$ into a vector representation in $\mathbb{R}^{d}$. Assume that the size of action space is d_{C}, and the dimension of the vector space for state representation is d_{R}.

Feature engineering

1/1 point (graded)

Exercise: Consider the following feature engineering. Define a function $\psi_{C}:\mathcal{C}\rightarrow\mathbb{R}^{d_{C}}$ where the j-th component $\psi_{C,j}\left(c\right)$ is given as follows:

$$\psi_{C,j}\left(c\right)=\begin{cases}1 & \text{if } j=c\\ 0 & \text{else}\end{cases}$$

The feature vector is defined as

$$\phi(s,c) = \begin{bmatrix} \psi_R(s) \\ \psi_C(c) \end{bmatrix}.$$

Will it work?

- Yes
- No ✔

Solution:

No. Let us fix the parameters θ. By the definition, for each state s:

$$\begin{aligned} \max_c Q(s,c,\theta) &= \max_c \sum_{i=1}^{d} \phi_i(s,c)\,\theta_i = \max_c \Big\{ \sum_{i=1}^{d_R} \psi_{R,i}(s)\,\theta_i + \sum_{j=1}^{d_C} \psi_{C,j}(c)\,\theta_{j+d_R} \Big\} \\ &= \max_c \Big\{ \sum_{j=1}^{d_C} \psi_{C,j}(c)\,\theta_{j+d_R} \Big\} \\ &= \max_c \theta_{c+d_R}. \end{aligned}$$

Note that the action that maximizes the Q-function for each state does not depend on the state s. In particular, for each learned parameter θ, the optimal action for all states are the same. Such policy is not optimal in our setting. Therefore, the above feature engineering is not sufficient to learn a good approximation of the optimal Q-function.

Submit You have used 2 of 4 attempts

Answers are displayed within the problem

Alternatively, consider the following feature map: $\phi(s,c) \in \mathbb{R}^{d_C \cdot d_R}$, where $\phi_i(s,c) = 0$ for all $i \notin [(c-1)\cdot d_R + 1,\ c \cdot d_R]$, and for $i \in [(c-1)\cdot d_R + 1,\ c \cdot d_R]$, $\phi_i(s,c) = \psi_{R,i-(c-1)d_R}(s)$. That is,

$$\phi(s,c) = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \psi_R(s) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

You will implement this feature map in the next tab.

Computing theta update rule

1.0/1 point (graded)

The Q-learning approximation algorithm starts with an initial parameter estimate of θ. As the tabular Q-learning, upon observing a data tuple $(s, c, R(s,c), s')$, the target value y for the Q-value of (s, c) is defined as the sampled version of the Bellman operator,

$$y = R(s,c) + \gamma \max_{c'} Q(s', c', \theta).$$

Then the parameter θ is simply updated by taking a gradient step with respect to the squared loss

$$L(\theta) = \frac{1}{2}(y - Q(s,c,\theta))^2.$$

The negative gradient can be computed as follows:

(Enter your answer in terms of `y`, `Q(s, c, theta)`, and `phi(s, c)`.)

$g(\theta) = -\frac{\partial}{\partial \theta} L(\theta) =$ (y-Q(s,c,theta))*phi(s,c)

✔ **Answer:** (y - Q(s, c, theta))*phi(s, c)

Solution:

The negative gradient can be computed as follows:

$$g(\theta) = -\frac{\partial}{\partial \theta} L(\theta) = (y - Q(s,c,\theta)) \cdot \frac{\partial}{\partial \theta} Q(s,c,\theta) = (y - Q(s,c,\theta))\, \phi(s,c)$$

Submit You have used 1 of 6 attempts

Answers are displayed within the problem

Hence the update rule for θ is :

$$\theta \leftarrow \theta + \alpha g(\theta) = \theta + \alpha \left[R(s,c) + \gamma \max_{c'} Q(s', c', \theta) - Q(s, c, \theta) \right] \phi(s, c),$$

where α is the learning rate.

Discussion

Topic: Unit 5 Reinforcement Learning (2 weeks) :Project 5: Text-Based Game / 6. Q-learning with linear function approximation

Hide Discussion

Add a Post

Show all posts ▼ by recent activity ▼

[Staff] Is "maxq_next" provided variable correct in deep_q_learning function? 2
Is "maxq_next" provided variable correct in deep_q_learning function? or It should be maxq_next = 0 if terminal else ...

[Staff] Feature engineering: 4 attempts for a yes/no question. 2
Feature engineering question: there are 4 attempts for a yes/no question. Kind regards

? Green check but where did phi come from?

I did get the answer by taking the derivative and then deducing the missing part using the update rule. After thinking about it I understand the e... 2

Feedback: I found this too abstract and I haven't understood 3
So I managed to get the first question right by elimination, and the second by blindly answering it (i.e. I just did what I was told and differentiated...

Clarification on not working feature vectors 2
Could someone clarify why the first way of feature engineering would not work? thanks

? How does feature engineering work in this case? Isn't phi(s, c) is a table of numbers? 2

? Theta update rule: why not use phi(s,c) transposed? 8

[STAFF] Please, help with some comments/explanations 2

Regarding feature vector 1 new 5
Community TA

Feature engineering 4 new
It's not very clear intuitively when a feature vector will work and when it will not (is it only the dimension reduction)? does anyone have clarity on ...
Following

How to compute a derivative of a max? 3

? Engineering a proper feature map 2

? [Staff] What variables and how to write them? 5 new 10
Really not seeing how to translate the answer into language for submission.