



MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning[Course](#)[Progress](#)[Dates](#)[Discussion](#)[Resources](#)[Course](#) / [Unit 5. Reinforcement Learning \(2 we...](#) / [Lecture 17. Reinforcement L](#)[< Previous](#)

6. Bellman Equations

[Bookmark this page](#)

Exercises due May 3, 2023 08:59 -03 Completed

Bellman Equations



Video

[!\[\]\(de95854c7ee024cfadc48187bbb781b2_img.jpg\) Download video file](#)

Transcripts

[!\[\]\(6059a5aa8b4ca7bb793408023d6c6e42_img.jpg\) Download SubRip \(.srt\) file](#)[!\[\]\(c50c8b7b2cc2cf9ff925edec0ee94c0d_img.jpg\) Download Text \(.txt\) file](#)

Recall from lecture the **Bellman Equations** are

$$V^*(s) = \max_a Q^*(s, a)$$

$$Q^*(s, a) = \sum_{s'} T(s, a, s') (R(s, a, s') + \gamma V^*(s'))$$

where

- the **value function** $V^*(s)$ is the expected reward from starting at state s and acting optimally afterwards.
- the **Q-function** $Q^*(s, a)$ is the expected reward from starting at state s , then acting optimally afterwards.

[Submit](#)

You have used 1 of 2 attempts

Bellman Equation for Q Function

1/1 point (graded)

As above, let there be n possible actions, a_1, \dots, a_n , from a given state s with

Let s' be a state that can be reached from s by taking the action a_i . Let

[< Previous](#)[Next >](#)

edX

[About](#)[Affiliates](#)[edX for Business](#)[Open edX](#)[Careers](#)[News](#)

Legal

[Terms of Service & Honor Code](#)[Privacy Policy](#)[Accessibility Policy](#)[Trademark Policy](#)[Sitemap](#)[Cookie Policy](#)



© 2023 edX LLC. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)