Topic 6: Q-learning in decision problems

Jaroslav Borovička

Computational Dynamics (Spring 2023)

New York University

Content

Economic problem

- A Bayesian learner uses available information to update beliefs about unobserved quantities.
- The learner must understand the probabilistic structure of the problem. What if it is not easily available?
- Q-learning is an example of a so-called reinforcement learning algorithm in which agent learns ‘optimal’ actions by experimentation

Tools

- Monte Carlo methods
- Reinforcement learning

Textbook

- *Reinforcement learning*: Sutton and Barto (2018)
- *Non-Bayesian learning in economics*: Sargent (1993), Sargent (1999), Evans and Honkapohja (2001)

Applications and ideas in economics

- Hart and Mas-Colell (2001), Evans et al. (2005)

QuantEcon

- Quantitative Economics with Python: Topic 40 (Q-learning in worker search problem)

Monte-Carlo methods and Q-learning

Consider a random variable X and its expectation $E[X]$.

- we have seen a range of methods how to evaluate the expectation operator
- one method relied on creating a large sample of draws x^i, $i = 1, \ldots, N$ and approximating

$$E[X] \approx \frac{1}{I} \sum_{i=1}^{I} x^i$$

- a law of large numbers states assumptions that guarantee convergence, and central limit theorems characterize properties

This is an example of a Monte-Carlo algorithm

- methods that rely on repeated random sampling in numerical computation

The expectation can take the form of a present value conditoned on a state $s_0 \in \mathcal{S}$

$$V(s) = E\left[\sum_{t=0}^{\infty} \beta^t y_t | s_0 = s\right].$$

A Monte-Carlo algorithm would proceed analogously as in the static case

- draw long samples of paths y_t^i, $t = 0, 1, \ldots, T$, $i = 1, \ldots, I$
- these paths replicate the potential temporal dependence of the data
- truncation at T justified due to discounting
- then evaluate

$$V(s) \approx \sum_{i=1}^{I} \sum_{t=0}^{T} \beta^t y_t^i$$

We know that this problem satisfies the Bellman equation

$$V(s) = E\left[y_0 + \beta V(s_1) \,|s_0 = s\right].$$

- this problem can be solved by backward induction
- in every step, we evaluate the expectations operator

What if we replace the expectation operator with simulation?

- imagine that we draw next period state $s_1 = s'$
- if s' is drawn from the correct conditional distribution of $s_1|s_0$, then $y_0 + \beta V(s')$ is an unbiased (albeit very noisy) estimate of $E\left[y_0 + \beta V(s_1) \,|s_0 = s\right]$

Take the Bellman equation

$$V(s) = E\left[y_0 + \beta V(s_1) \,|s_0 = s\right].$$

and form

$$y_0 + \beta V\left(s'\right) - V(s)$$

- this quantity is known as temporal difference (difference between the simulated draw of the continuation value and current value)
- the quantity has zero conditional expectation for the correct value function V

The temporal difference learning algorithm updates the value $V(s)$ as follows

$$V(s) \leftarrow V(s) + \alpha\left[y_0 + \beta V\left(s'\right) - V(s)\right]$$

- $\alpha \in (0, 1)$ is the learning rate parameter
- idea: if the continuation policy draw indicates higher realization of the value than what the current value $V(s)$, then update current value upward, and vice versa

Temporal difference learning algorithm

$$V(s) \leftarrow V(s) + \alpha \underbrace{\left[y_0 + \beta V\left(s'\right) - V(s)\right]}_{\text{temporal difference}}$$

Idea of the algorithm: If the continuation policy draw indicates higher realization of the value that what the current value indicates,

$$y_0 + \beta V\left(s'\right) > V(s)$$

then, on average, $V(s)$ is too low, and needs to be updated upward.

- How to choose the learning rate? It should vanish over time for convergence.
- Convergence properties? Hard to evaluate for complex environments.
- How to implement when s is a continuous state variable? Discretization.

The algorithm finds the present value of a particular cash flow y_t, $t = 0, 1, \ldots$

- this cash flow can be interpreted as one obtained under a particular policy

How to implement optimal policy choice?

- a naive procedure could try to repeat the procedure for every alternative policy and then choose the 'best' one
- this is not useful
- need to incorporate updating of choices into the recursive procedure $\implies$ Q-learning

Imagine that in every state $s \in \mathcal{S}$, there is a set of available actions $a \in \mathcal{A}$

- distribution of next period state s' can now also depend on current action a

Define the state-action value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as

$$Q(s, a) = y(a) + \beta E\left[\max_{a' \in \mathcal{A}} Q\left(s', a'\right) | s, a\right] \tag{6.1}$$

The state-action value function is tightly related to the value function $V(s)$

$$V(s) = \max_{a \in \mathcal{A}} Q(s, a)$$

We can now combine the temporal dependence algorithm with 'optimal' choice

- start from some initial guess of the function $Q(s,a)$
- for a given current state s and action a, draw next period state s', and update using

$$Q(s,a) \leftarrow Q(s,a) + \alpha \underbrace{\left[y(a) + \beta \max_{a' \in \mathcal{A}} Q\left(s', a'\right) - Q(s,a) \right]}_{\text{temporal difference}}$$

- the temporal difference has again conditional mean zero for the correct function Q
- the idea is that the algorithm should 'stabilize' in the neighborhood of the true Q

We can also write this as the weighted average

$$Q(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha \left[y(a) + \beta \max_{a' \in \mathcal{A}} Q\left(s', a'\right) \right]$$

The algorithm

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[y(a) + \beta \max_{a' \in \mathcal{A}} Q\left(s', a'\right) - Q(s,a) \right]$$

Experimentation

- the above algorithm can get 'stuck' in local maxima
- problem known from Markov chain Monte-Carlo methods in econometrics
- experimentation: occasionally replace $\max_{a' \in \mathcal{A}} Q(s', a')$ with $Q\left(s', \widetilde{a}\right)$ with $\widetilde{a}$ randomly drawn

Discretization

- algorithm cannot pointwise update continuous state spaces $\mathcal{S}$ and $\mathcal{A}$
- discretization (tabular approach): replace $\mathcal{S} \times \mathcal{A}$ with a grid
- deep Q-learning: integrate updating with a projection method that approximates the continuous function, for example, using a neural netword (nonlinear regression)

Once algorithm converged, we can obtain optimal value and policy

$$a^*(s) = \arg\max_{a \in \mathcal{A}} Q(s, a) \qquad V(s) = Q(s, a^*(s))$$

The method is an example of reinforcement learning.

- actions that lead to high realized values (rewards) are 'reinforced' as good choices

The algorithm does not involve the formation of agent's beliefs.

- the agent is presented with draws from the conditional distribution of the state and learns how to take optimal action conditional on the state
- markedly different from forms of learning where agent uses information to update beliefs (in a Bayesian or non-Bayesian way)
- advantageous when the description of the probability distribution is too complex or in situations where uncertainty is not essential but actions are complex (strategic games)

Why did we use the simulation approach in the first place? Two approaches.

Approach 1: A method for solving rational expectations problems

- We are interested in solving problem (6.1) but evaluation of the expectations operator is hard.
- Once the problem has converged, we obtain optimal policy under the data-generating process.

Approach 2: Approximation of actual behavior.

- humans are not Bayesian, and reinforcement learning approximates the way they act
- a complex structure that needs to be disciplined (learning rate, experimentation,...).
- what are the (unique) testable implications to compare such a theory with data?

Once we have solved this problem, we still have many steps ahead of us.

- How to incorporate interactions between agents and the formation of equilibria? (Hart and Mas-Colell (2001))

Q-learning in the worker search problem

We now revisit the McCall (1970) model of a worker who samples wage offers.

- in the McCall (1970) model, worker understands the probabilistic structure of the model
- worker is able to form a belief (objective or subjective) over the distribution of offers next period

Instead assume that the worker does not have available the probability distribution.

- worker observes realized draws and is able to take accept/reject decisions
- uses the Q-learning algorithm to learn the value of actions, and deduce optimal action

Problem based on the QuantEcon lecture

https://python.quantecon.org/mccall_q.html

An infinite-horizon model of job search (McCall (1970))

- time is discrete and infinite, $t = 0, 1, 2, \ldots$
- every period t, an iid wage offer w from distribution $F(w)$ is drawn, with $F(0) = 0$, $F(B) = 1$ for some $B > 0$

A worker decides to accept or reject the offer, $a_t \in \{\text{accept}, \text{reject}\}$

- when accepts, the worker receives income $y_t = w$ forever
- when rejects, the worker receives unemployment benefit $y_t = c$ and moves to next period where a new offer is drawn
- time is discounted at rate $\beta \in [0, 1)$

The worker solves the sequence problem

$$V_0^* = \max_{\{a_t\}_{t=0}^{\infty}} E_0 \left[\sum_{t=0}^{\infty} \beta^t y_t \right] \tag{6.2}$$

where $a_t \in \{\text{accept, reject}\}$ if the worker has not yet accepted any earlier offer, and $a_t \in \{\}$ otherwise.

- V_0^* is the value function, assume V_0^* conditions on the initial offer w_0
- every decision a_t is made conditional on the time-t information set, which contains the history of all offers up to time t, $w^t = (w_0, \ldots, w_t)$
- $E[\cdot]$ is the mathematical expectations operator

$$E[w] = \int_0^B w dF(w) = \int_0^B w f(w)\, dw.$$

We deduced that the problem of a worker with current offer w at hand can be formulated recursively

$$V(w) = \max_{\{\text{accept, reject}\}} \left\{ V^a(w), c + \beta \int_0^B V(w') \, dF(w') \right\}$$

where $V^a(w)$ is the value of accepting the offer.

We assumed that once an offer w is accepted, the worker works at that wage forever

$$V^a(w) = \frac{w}{1-\beta}.$$

Let us modify the problem formulation slightly.

- assume that after working for one period at w, the worker can decide to accept or reject the same wage w
- if rejected, the worker becomes unemployed for the period and receives a new draw next period

$$V^a(w) = w + \beta \max_{\{\text{accept, reject}\}} \left\{ V^a(w), c + \beta \int_0^B V(w') \, dF(w') \right\}$$

- this is inconsequential for the solution since we know that an accepted offer in the stationary environment would never be rejected later

Rewrite the problem in the form of the state-action value function $Q(w, a)$:

$$\begin{aligned} Q(w, \text{accept}) &= w + \beta \max_{\{\text{accept, reject}\}} \{Q(w, \text{accept}), Q(w, \text{reject})\} \\ Q(w, \text{reject}) &= c + \beta \int_0^B \max_{\{\text{accept, reject}\}} \left\{Q\left(w', \text{accept}\right), Q\left(w', \text{reject}\right)\right\} dF\left(w'\right) \end{aligned}$$

- notice the distinction between w and w'

We now have

$$V^a(w) = Q(w, \text{accept}) \qquad V(w) = \max_{\{\text{accept, reject}\}} \{Q(w, \text{accept}), Q(w, \text{reject})\}.$$

We can now replace the expectations with sample draws and form temporal differences.

Start with an old iteration of the state-action value function $Q^{old}(w, a)$

$$\begin{aligned} TD(w, \text{accept}) &= w + \beta \max_{a' \in \mathcal{A}} Q^{old}\left(w, a'\right) - Q^{old}(w, \text{accept}) \\ TD(w, \text{reject}) &= c + \beta \max_{a' \in \mathcal{A}} Q^{old}\left(w', a\right) - Q^{old}(w, \text{reject}) \end{aligned}$$

for $\mathcal{A} = \{\text{accept}, \text{reject}\}$ and $w' \sim F(w')$.

Then the new iteration of the state-action value function is

$$Q^{new}(w, a) = Q^{old}(w, a) + \alpha TD(w, a)$$

Discretization

- action is already discrete, replace the state space $[0, B]$ with a discrete grid w^i, $i = 1, \ldots, I$
- replace continuous distribution $F(w)$ with a discrete counterpart $\hat{f}^i$ of mass points on grid w^i
- sample offers w' from $\hat{f}^i$, $i = 1, \ldots, I$
- alternatively, use a form of projection and a 'deep Q-learning' algorithm to update the projection coefficients

Experimentation: modify the algorithm to incorporate deviations from currently 'optimal' choice

- in every step, with probability ε, replace $\max_{a' \in \mathcal{A}}$ with $\min_{a' \in \mathcal{A}}$
- allow exploration of alternatives that may be omitted if algorithm gets stuck in a local maximum

Summary

Summary

Q-learning is a model-free (unstructured) learning algorithm that learns optimal actions using

- simulation (Monte-Carlo methods)
- implementation of the concept of dynamic programming

The method can be used for

- approximation of RE in problems with complicated structures
- modeling of non-Bayesian behavior

Challenges

- convergence properties of the decision problem
- embedding of the individual decision problem into models of strategic interaction and equilibria
- how distinct is this form of learning from other non-Bayesian approaches?
- What do we learn from solving these problems? We do not only want to characterize optimal decisions, we also want to understand them. Policy implications?

Summary

(Macro)economic theory has long explored deviations from Bayesian learning

- interview with Tom Sargent (Evans et al. (2005))
- Sargent (1993) Bounded Rationality in Macroeconomics
- Sargent (1999) The Conquest of American Inflation
- Evans and Honkapohja (2001) Learning and Expectations in Macroeconomics

Start here!

Appendix

Evans, George W. and Seppo Honkapohja (2001) *Learning and Expectations in Macroeconomics*: Princeton University Press, Princeton, NJ.

Evans, George W., Seppo Honkapohja, and Thomas J. Sargent (2005) "An Interview with Thomas J. Sargent," *Macroeconomic Dynamics*, 9 (4), 561–583.

Hart, Sergiu and Andreu Mas-Colell (2001) "A Reinforcement Procedure Leading to Correlated Equilibrium," in Debreu, Gérard, Wilhelm Neuefeind, and Walter Trockel eds. *Economics Essays: A Festschrift for Werner Hildenbrand*, 181–200: Springer-Verlag, Berlin, Heidelberg.

McCall, John (1970) "Economics of Information and Job Search," *Quarterly Journal of Economics*, 84 (1), 113–126.

Sargent, Thomas J. (1993) *Bounded Rationality in Macroeconomics*: Clarendon Press, Oxford.

——— (1999) *The Conquest of American Inflation*: Princeton University Press, Princeton, NJ.

Sutton, Richard S. and Andrew G. Barto (2018) *Reinforcement Learning: An Introduction*: MIT Press, Cambridge, MA, 2nd edition.