Bellman equation

A **Bellman equation**, named after its discoverer, Richard Bellman, also known as a *dynamic programming equation*, is a necessary condition for optimality associated with the mathematical optimization method known as dynamic programming. It writes the value of a decision problem at a certain point in time in terms of the payoff from some initial choices and the value of the remaining decision problem that results from those initial choices. This breaks a dynamic optimization problem into simpler subproblems, as Bellman's *principle of optimality* prescribes.

The Bellman equation was first applied to engineering control theory and to other topics in applied mathematics, and subsequently became an important tool in economic theory; though the basic concepts of dynamic programming are prefigured in John von Neumann and Oskar Morgenstern's *Theory of Games and Economic Behavior* and Abraham Wald's *sequential analysis*.

Almost any problem which can be solved using optimal control theory can also be solved by analyzing the appropriate Bellman equation. However, the term 'Bellman equation' usually refers to the dynamic programming equation associated with discrete-time optimization problems. In continuous-time optimization problems, the analogous equation is a partial differential equation which is usually called the Hamilton–Jacobi–Bellman equation.

Contents

Analytical concepts in dynamic programming

To understand the Bellman equation, several underlying concepts must be understood. First, any optimization problem has some objective: minimizing travel time, minimizing cost, maximizing profits, maximizing utility, et cetera. The mathematical function that describes this objective is called the *objective function*.

Dynamic programming breaks a multi-period planning problem into simpler steps at different points in time. Therefore, it requires keeping track of how the decision situation is evolving over time. The information about the current situation which is needed to make a correct decision is called the "state".[1][2] For example, to decide how much to consume and spend at each point in time, people would need to know (among other things) their initial wealth. Therefore, wealth (W) would be one of their *state variables*, but there would probably be others.

The variables chosen at any given point in time are often called the *control variables*. For example, given their current wealth, people might decide how much to consume now. Choosing the control variables now may be equivalent to choosing the next state; more generally, the next state is affected by other factors in addition to the current control. For example, in the simplest case, today's wealth (the state) and consumption (the control) might exactly determine tomorrow's wealth (the new state), though typically other factors will affect tomorrow's wealth too.

The dynamic programming approach describes the optimal plan by finding a rule that tells what the controls should be, given any possible value of the state. For example, if consumption (*c*) depends *only* on wealth (*W*), we would seek a rule $c(W)$ that gives consumption as a function of wealth. Such a rule, determining the controls as a function of the states, is called a *policy function* (See Bellman, 1957, Ch. III.2).[1]

Finally, by definition, the optimal decision rule is the one that achieves the best possible value of the objective. For example, if someone chooses consumption, given wealth, in order to maximize happiness (assuming happiness *H* can be represented by a mathematical function, such as a utility function), then each level of wealth will be associated with some highest possible level of happiness, $H(W)$. The best possible value of the objective, written as a function of the state, is called the *value function*.

Richard Bellman showed that a dynamic optimization problem in discrete time can be stated in a recursive, step-by-step form known as backward induction by writing down the relationship between the value function in one period and the value function in the next period. The relationship between these two value functions is called the "Bellman equation". In this approach, the optimal policy in the last time period is specified in advance as a function of the state variable's value at that time, and the resulting optimal value of the objective function is thus expressed in terms of that value of the state variable. Next, the next-to-last period's optimization involves maximizing the sum of that period's period-specific objective function and the optimal value of the future objective function, giving that period's optimal policy contingent upon the value of the state variable as of the next-to-last period decision. This logic continues recursively back in time, until the first period decision rule is derived, as a function of the initial state variable value, by optimizing the sum of the first-period-specific objective function and the value of the second period's value function, which gives the value for all the future periods. Thus, each period's decision is made by explicitly acknowledging that all future decisions will be optimally made.

Derivation

A dynamic decision problem

Let the state at time t be x_t. For a decision that begins at time 0, we take as given the initial state x_0. At any time, the set of possible actions depends on the current state; we can write this as $a_t \in \Gamma(x_t)$, where the action a_t represents one or more control variables. We also assume that the state changes from x to a new state $T(x, a)$ when action a is taken, and that the current payoff from taking action a in state x is $F(x, a)$. Finally, we assume impatience, represented by a

discount factor $0 < \beta < 1$.

Under these assumptions, an infinite-horizon decision problem takes the following form:

$$V(x_0) \ = \ \max_{\{a_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t F(x_t, a_t),$$

subject to the constraints

$$a_t \in \Gamma(x_t), \ x_{t+1} = T(x_t, a_t), \ \forall t = 0, 1, 2, \dots$$

Notice that we have defined notation $V(x_0)$ to denote the optimal value that can be obtained by maximizing this objective function subject to the assumed constraints. This function is the *value function*. It is a function of the initial state variable x_0, since the best value obtainable depends on the initial situation.

Bellman's Principle of Optimality

The dynamic programming method breaks this decision problem into smaller subproblems. Richard Bellman's *principle of optimality* describes how to do this:

> Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. (See Bellman, 1957, Chap. III.3.)[1][2][3]

In computer science, a problem that can be broken apart like this is said to have optimal substructure. In the context of dynamic game theory, this principle is analogous to the concept of subgame perfect equilibrium, although what constitutes an optimal policy in this case is conditioned on the decision-maker's opponents choosing similarly optimal policies from their points of view.

As suggested by the *principle of optimality*, we will consider the first decision separately, setting aside all future decisions (we will start afresh from time 1 with the new state x_1). Collecting the future decisions in brackets on the right, the previous problem is equivalent to:

$$\max_{a_0} \left\{ F(x_0, a_0) + \beta \left[\max_{\{a_t\}_{t=1}^{\infty}} \sum_{t=1}^{\infty} \beta^{t-1} F(x_t, a_t) : a_t \in \Gamma(x_t), \ x_{t+1} = T(x_t, a_t), \ \forall t \geq 1 \right] \right\}$$

subject to the constraints

$$a_0 \in \Gamma(x_0), \ x_1 = T(x_0, a_0).$$

Here we are choosing a_0, knowing that our choice will cause the time 1 state to be $x_1 = T(x_0, a_0)$. That new state will then affect the decision problem from time 1 on. The whole future decision problem appears inside the square brackets on the right.

The Bellman equation

So far it seems we have only made the problem uglier by separating today's decision from future decisions. But we can simplify by noticing that what is inside the square brackets on the right is *the value* of the time 1 decision problem, starting from state $x_1 = T(x_0, a_0)$.

Therefore, we can rewrite the problem as a recursive definition of the value function:

$$V(x_0) = \max_{a_0} \{F(x_0, a_0) + \beta V(x_1)\}$$, subject to the constraints: $a_0 \in \Gamma(x_0),\ x_1 = T(x_0, a_0).$

This is the Bellman equation. It can be simplified even further if we drop time subscripts and plug in the value of the next state:

$$V(x) = \max_{a \in \Gamma(x)} \{F(x, a) + \beta V(T(x, a))\}.$$

The Bellman equation is classified as a functional equation, because solving it means finding the unknown function *V*, which is the *value function*. Recall that the value function describes the best possible value of the objective, as a function of the state *x*. By calculating the value function, we will also find the function *a*(*x*) that describes the optimal action as a function of the state; this is called the *policy function*.

In a stochastic problem

In the deterministic setting, other techniques besides dynamic programming can be used to tackle the above optimal control problem. Although the agent has to account for the stochasticity, this approach becomes convenient for certain problems.

For a specific example from economics, consider an infinitely-lived consumer with initial wealth endowment a_0 at period 0. He has an instantaneous utility function $u(c)$ where c denotes consumption and discounts the next period utility at a rate of $0 < \beta < 1$. Assume what is not consumed in period t carries over next period with interest rate r. Then the consumer's utility maximization problem is to choose a consumption plan $\{c_t\}$ that solves

$$\max \sum_{t=0}^{\infty} \beta^t u(c_t)$$

subject to

$$a_{t+1} = (1 + r)(a_t - c_t),\ c_t \geq 0,$$

and

$$\lim_{t\to\infty} a_t \geq 0.$$

The first constraint is the capital accumulation/law of motion specified by the problem, while the second constraint is a transversality condition that the consumer does not carry debt at the end of his life. The Bellman equation is

$$V(a) = \max_{0\leq c\leq a}\{u(c) + \beta V((1+r)(a-c))\},$$

Alternatively, one can treat the sequence problem directly using, for example, the Hamiltonian equations.

Now, if the interest rate varies from period to period, the consumer is faced with a stochastic optimization problem. Let the interest *r* follow a Markov process with probability transition function $Q(r, d\mu_r)$ where $d\mu_r$ denotes the probability measure governing the distribution of interest rate next period if current interest rate is r. The timing of the model is that the consumer decides his current period consumption after the current period interest rate is announced.

Rather than simply choosing a single sequence $\{c_t\}$, the consumer now must choose a sequence $\{c_t\}$ for each possible realization of a $\{r_t\}$ in such a way that his lifetime expected utility is maximized:

$$\max \mathbb{E}\left(\sum_{t=0}^{\infty} \beta^t u(c_t)\right).$$

The expectation $\mathbb{E}$ is taken with respect to the appropriate probability measure given by *Q* on the sequences of *r* 's. Because *r* is governed by a Markov process, dynamic programming simplifies the problem significantly. Then Bellmann equation is simply

$$V(a, r) = \max_{0\leq c\leq a}\{u(c) + \beta \int V((1+r)(a-c), r')Q(r, d\mu_r)\}.$$

Under some reasonable assumption, the resulting optimal policy function *g*(*a*,*r*) is measurable.

For a general stochastic sequential optimization problem with Markovian shocks and where the agent is faced with his decision *ex-post*, the Bellmann equation takes a very similar form

$$V(x, z) = \max_{c\in\Gamma(x,z)} F(x, c, z) + \beta \int V(T(x, c), z')d\mu_z(z').$$

Solution methods

- The method of undetermined coefficients, also known as 'guess and verify', can be used to solve some infinite-horizon, autonomous Bellman equations.
- The Bellman equation can be solved by backwards induction, either analytically in a few special cases, or numerically on a computer. Numerical backwards induction is applicable to a wide variety of problems, but may be infeasible when there are many state variables, due to the curse of dimensionality. Approximate dynamic programming has been introduced by D. P. Bertsekas and J. N. Tsitsiklis with the use of artificial neural networks (multilayer perceptrons) for approximating the Bellman function.[4] This is an effective mitigation strategy for reducing the impact of dimensionality by replacing the memorization of the complete function mapping for the whole space domain with the memorization of the sole neural network parameters.
- By calculating the first-order conditions associated with the Bellman equation, and then using the envelope theorem to eliminate the derivatives of the value function, it is possible to obtain a system of difference equations or differential equations called the 'Euler equations'. Standard techniques for the solution of difference or differential equations can then be used to calculate the dynamics of the state variables and the control variables of the optimization problem.

Applications in economics

The first known application of a Bellman equation in economics is due to Martin Beckmann and Richard Muth.[5] Martin Beckmann also wrote extensively on consumption theory using the Bellman equation in 1959. His work influenced Edmund S. Phelps, among others.

A celebrated economic application of a Bellman equation is Robert C. Merton's seminal 1973 article on the intertemporal capital asset pricing model.[6] (See also Merton's portfolio problem).The solution to Merton's theoretical model, one in which investors chose between income today and future income or capital gains, is a form of Bellman's equation. Because economic applications of dynamic programming usually result in a Bellman equation that is a difference equation, economists refer to dynamic programming as a "recursive method" and a subfield of recursive economics is now recognized within economics.

Nancy Stokey, Robert E. Lucas, and Edward Prescott describe stochastic and nonstochastic dynamic programming in considerable detail, and develop theorems for the existence of solutions to problems meeting certain conditions. They also describe many examples of modeling theoretical problems in economics using recursive methods.[7] This book led to dynamic programming being employed to solve a wide range of theoretical problems in economics, including optimal economic growth, resource extraction, principal–agent problems, public finance, business investment, asset pricing, factor supply, and industrial organization. Lars Ljungqvist and Thomas Sargent apply dynamic programming to study a variety of theoretical questions in monetary policy, fiscal policy, taxation, economic growth, search theory, and labor economics.[8] Avinash Dixit and Robert Pindyck showed the value of the method for thinking about capital budgeting.[9] Anderson adapted the technique to business valuation, including privately held businesses.[10]

Using dynamic programming to solve concrete problems is complicated by informational difficulties, such as choosing the unobservable discount rate. There are also computational issues, the main one being the curse of dimensionality arising from the vast number of possible actions and potential state variables that must be considered before an optimal strategy can be selected. For an extensive discussion of computational issues, see Miranda and Fackler,[11] and Meyn 2007.[12]

Example

In Markov decision processes, a Bellman equation is a recursion for expected rewards. For example, the expected reward for being in a particular state *s* and following some fixed policy π has the Bellman equation:

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} P(s'|s, \pi(s)) V^{\pi}(s').$$

This equation describes the expected reward for taking the action prescribed by some policy π.

The equation for the optimal policy is referred to as the *Bellman optimality equation*:

$$V^{*}(s) = \max_{a} \{R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^{*}(s')\}.$$

It describes the reward for taking the action giving the highest expected return.

See also

- Bellman pseudospectral method
- Dynamic programming
- Hamilton–Jacobi–Bellman equation
- Markov decision process
- Optimal control theory
- Optimal substructure
- Recursive competitive equilibrium
- Stochastic dynamic programming

References

1. Bellman, R.E. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ. Republished 2003: Dover, ISBN 0-486-42809-5.
2. S. Dreyfus (2002), 'Richard Bellman on the birth of dynamic programming' (http://www.wu-wien.ac.at/usr/h99c/h9951826/bellman_dynprog.pdf) Archived (https://web.archive.org/web/20050110161049/http://www.wu-wien.ac.at/usr/h99c/h9951826/bellman_dynprog.pdf) January 10, 2005, at the Wayback Machine. *Operations Research* 50 (1), pp. 48–51.
3. R Bellman, *On the Theory of Dynamic Programming*, Proceedings of the National Academy of Sciences, 1952
4. Bertsekas, D. P., Tsitsiklis, J. N., *Neuro-dynamic programming*. Athena Scientific, 1996
5. Beckmann, Martin; Muth, Richard (1954). "On the Solution to the 'Fundamental Equation' of inventory theory" (http://cowles.yale.edu/sites/default/files/files/pub/cdp/e-2116.pdf) (PDF). *Cowles Commission Discussion Paper 2116*.
6. Merton, Robert C. (1973). "An Intertemporal Capital Asset Pricing Model". *Econometrica*. **41** (5): 867–887. JSTOR 1913811 (https://www.jstor.org/stable/1913811).
7. Stokey, Nancy; Lucas, Robert E.; Prescott, Edward (1989). *Recursive Methods in Economic Dynamics*. Harvard Univ. Press. ISBN 0-674-75096-9.
8. Ljungqvist, Lars; Sargent, Thomas (2012). *Recursive Macroeconomic Theory* (Third ed.). MIT Press. ISBN 978-0-262-01874-6.

9. Dixit, Avinash; Pindyck, Robert (1994). *Investment Under Uncertainty*. Princeton Univ. Press. ISBN 0-691-03410-9.
10. Anderson, Patrick L., Business Economics & Finance, CRC Press, 2004 (chapter 10), ISBN 1-58488-348-0; The Value of Private Businesses in the United States, *Business Economics* (2009) 44, 87–108. doi:10.1057/be.2009.4 (https://doi.org/10.1057%2Fbe.2009.4). *Economics of Business Valuation*, Stanford University Press (2013); ISBN 9780804758307. Stanford Press (http://www.sup.org/book.cgi?id=11400)
11. Miranda, M., & Fackler, P., 2002. *Applied Computational Economics and Finance*. MIT Press
12. S. P. Meyn, 2007. Control Techniques for Complex Networks (http://decision.csl.uiuc.edu/~meyn/pages/CTCN/CTCN.html), Cambridge University Press, 2007. Appendix contains abridged Meyn & Tweedie (http://decision.csl.uiuc.edu/~meyn/pages/book.html) Archived (https://web.archive.org/web/20071012194420/http://decision.csl.uiuc.edu/~meyn/pages/book.html) 2007-10-12 at the Wayback Machine..