Dynamic optimization

Felix Kubler[1]

[1]DBF, University of Zurich and Swiss Finance Institute

January 27, 2017

Dynamic optimization

- In economics we often have problems where one wants to maximize an objective function that is infinite dimensional but has a lot of "stationary structure"
- Problem can be Bellmanized.
- Best method depends a lot on the actual problem!
- First consider a very standard model, explain how to solve it
- On the way, do a bit of an overview and some weird methods
- Then consider a more complicated model and explain how to Bellmanize it.
- Look at infinite horizon problems, but method can also be used for finite horizon, obviously.

The Stochastic Ramsey Model

$$\max \mathbb{E}_0[\sum_{t=0}^{\infty} u(c_t)] \quad \text{s.t. } c_t + k_{t+1} \leq \underbrace{z(s_t)f(k_t) + (1-\delta(s_t))k_t}_{\equiv \bar{f}(s_t,k_t)},$$

$$c_t \geq 0, \; k_{t+1} \geq 0 \; \forall t \in \mathbb{N}_0$$

$$k_0, s_0 \text{ given}$$

where the expectation is over the sequence of stocks $\{s_t\}_{t=1}^{\infty}$ given s_0.

Special Cases:

- δ fixed and AR(1)-process for TFP shock
  $\ln z_{t+1} = \rho \ln z_t + \epsilon_{t+1}$ where $\epsilon_{t+1} \sim \mathcal{N}(0, \sigma_\epsilon)$
- δ fixed and discrete Markov process:
  $z_t \in \mathcal{Z} = \{1, \ldots, Z\}, \mathbb{P}(z_{t+1} = i | z_t = j) = \pi_{ij}$
  with transition matrix π

The Stochastic Ramsey Model

- How can we solve the stochastic model?
- Want consumption, investment as a function of the shock and the capital-stock
- In this simple case can use numerical dynamic programming...

Markov Control Processes

Fancy treatment of dynamic programming. See e.g. Hernandez-Lerma and Lasserre 1996 , “Discrete-Time Markov Control Processes”

or, at least, Stokey and Lucas w. Prescott.

Markov Control Processes

Suppose the state space is $X \subset \mathbb{R}^l$ with Borel σ-algebra. Actions are $A : X \rightrightarrows \mathbb{R}^k$. The objective is

$$V(\pi, x) = E_x^\pi \sum_{t=0}^{\infty} \beta^t c(x_t, a_t), x \in X, \pi \in \Pi$$

Want to find optimal policy π^* such that

$$V(\pi^*, x) = \inf_\pi V(\pi, x) =: V^*(x)$$

Typically do not want to allow for randomizations, i.e. can associate an optimal policy with (a_t).
A measurable function $v : X \to \mathbb{R}$ is said to be a solution to the Bellman-equation if it satisfies

$$v(x) = \min_{A(x)} \left[c(x, a) + \beta \int_X v(y) Q(dy|x, a) \right]$$

Markov Control Processes

There are three cases of interest:

1. State space and action space are finite
2. Infinite state and actions, but cost-function is bounded and continuous, probabilities are fixed
   Stokey and Lucas' textbook gives conditions to ensure existence of value function
3. Infinite state and actions, but cost-function not necessarily bounded or continuous

Numerical dynamic programming

- As for theoretical results, it is also crucial here whether we assume finite or infinite state/action space
- Also important whether functions can be expected to be smooth/concave etc
- We consider 2 cases: Finite states and actions (FDP) and smooth and concave problems (EDP) (E for easy)

FDP: Value function iteration

- State space is $X = \{1, \ldots, S\}$, action sets are $A(x)$ finite sets, $c(x, a)$ finite for all $x, a \in A(x)$. Transition probabilities are $\pi(x'|x, a)$. $\beta < 1$
- For any $v \in \mathbb{R}^S$ define $Tv \in \mathbb{R}^S$ by

$$Tv(x) = \min_{a \in A(x)} c(x, a) + \beta \sum_{x'} \pi(x'|x, a) v(x')$$

- Given and $v_0 \in \mathbb{R}^S$ value function iteration computes

$$v^* = \lim_{n \to \infty} T^n v_0$$

- The limit exists, is unique and satisfies $v^*(x) = Tv^*(x)$ for all x.

FDP: Value function iteration - comments

- Convergence of value function iteration follows from the contraction mapping theorem.
- This is also true in function spaces, but here it is obvious, since $\beta < 1$
- For β close to one, convergence might be very slow

FDP: Real time DP

The following converges to the true value function with probability one. Fix v_0, x_0.

- Given any v_t, x_t, compute

$$a_t = \arg\min c(x_t, a) + \beta \sum_{x'} \pi(x'|x_t, a) v_t(x')$$

- Update v_t

$$v_{t+1}(x) = \begin{cases} c(x_t, a_t) + \beta \sum_{x'} \pi(x'|x_t, a_t) v_t(x') & \text{if } x = x_t \\ v_t(x) & \text{otherwise.} \end{cases}$$

- Sample x_{t+1} from $\pi(.|x_t, a_t)$

Approximate FDP

For the case of finite actions, finite states applied mathematicians have spend a lot of effort trying to find efficienct approximate methods that might use randomization. Two popular ones are:

- Linear programming approach to approximate dynamic programming (based on De Farias and Van Roy (2003, Operations Research)).
- Temporal difference learning methods (e.g. Bertsekas and Tsitsiklis (1996, Book)).

LP-DP- basic idea 1

- Bellman equation can be formulated as a linear program
- Given and $b \in \mathbb{R}^S_{++}$ the optimal value v^* is the solution to

$$\max b^T v \text{ s.t. } Tv \geq v$$

- $Tv \geq v$ does not look very linear but with finitely many actions this can be written ad

$$c(x,a) + \beta \sum_{x'} \pi(x'|x,a) v(x') \geq v(x) \text{ for all } x \in X, a \in A(x)$$

- Huge linear program!

LP-DP- basic idea 2

- Pick some bases functions $\phi_j \in \mathbb{R}^S$, $j = 1, \ldots J$ and approximate $v(x) = \sum_j r_j \phi_j(x)$.
  Write $v = \Phi r$
- Approximate LP is
$$\max_r b^T \Phi r \text{ s.t.}$$
$$c(x, a) + \beta \sum_{x'} \pi(x'|x, a) \Phi r(x') \geq \Phi r(x) \text{ for all } x \in X, a \in A(x)$$
- Not many variables but huge number of constraints $\rightarrow$ sample constraints

LP-DP- basic idea 3

How to sample the constraints

- Take linear program

$$\max b^T x \text{ s.t. } Ax \leq k$$

  with A an $m \times n$ matrix and large $m \gg n$.
- Given a probability distribution μ over $\{1, \ldots, m\}$ sample $i_1, i_2, \ldots$ according to μ. Define $\hat{x}_N$ as the optimal solution to the linear program with only constraints $A_{i_1}, \ldots, A_{i_N}$.
- For any $\epsilon, \delta > 0$, if $N \geq \frac{n}{\epsilon\delta} - 1$

$$P(\mu\{i | A_i \hat{x}_N > k_i\} \leq \epsilon) \geq 1 - \delta$$

LP-DP- some comments

$$\max_r b^T \Phi r \text{ s.t.}$$

$$c(x,a) + \beta \sum_{x'} \pi(x'|x,a)\Phi r(x') \geq \Phi r(x) \text{ for all } x \in X, a \in A(x)$$

- Choice of b makes a big difference. Ideally should be the ergodic probability distribution over states under the optimal policy
- Choice of Φ very problem-dependent

EDP

- Many problems in economics have a lot of structure absent in the general formulation

EDP

- Many problems in economics have a lot of structure absent in the general formulation
- Suppose the state can be written as $x = (s, y)$ and conditional on s the future y' is a deterministic function of a. We have

$$v^*(x) = \min_{a \in A(x)} c(x, a) + \beta \sum_{s'} \pi(s'|s) v^*(x') \text{ s.t. } y' = p(x, a)$$

- Action space $A(x) \subset \mathbb{R}^l$ convex for all x
- Admissible endogenous states $y \in Y \subset \mathbb{R}^k$, s_t follows Markov chain
- Assume $c(.)$ is bounded and continuous

EDP: Value Function Iteration – The Idea I

For 'well behaved problems' we have that

- The Bellman operator has a unique fixed point v_0 in the space of bounded continuous value functions $C(X)$
- Starting from any function in $C(X)$ and repeatedly applying T brings us closer and closer to the fixed point
- Thus, we will finally converge (by any desired precision)

The Value Function: The Idea II

Thus, we can take any $v_0 \in C(X)$ and repeatedly apply the Bellman operator T until we reach the desired level of convergence.
$\Rightarrow$ Value function iteration is a robust and reliable method!

However, in theory, we would need ‘perfect approximations’ for the value function in each iteration. In practice, we will either:

- Discretize the state space, i.e. allow only discrete choices and no values in-between. What happens with very large, finite state-space?
- Interpolate the value function, i.e. the update will only be ‘perfect’ at the interpolation nodes.
  Alternatively, approximate value function

The Stochastic Ramsey Model reconsidered

$$\max \mathbb{E}_0[U(\{c_t\})] \quad \text{s.t. } c_t + k_{t+1} \leq \underbrace{z(s_t)f(k_t) + (1-\delta(s_t))k_t}_{\equiv \bar{f}(s_t,k_t)},$$

$$c_t \geq 0,\ k_{t+1} \geq 0\ \forall t \in \mathbb{N}_0$$

$$k_0, s_0 \text{ given}$$

where the expectation is over the sequence of stocks $\{s_t\}_{t=1}^{\infty}$ given s_0.

Assume for now that s_t follows a Markov chain with finite support. The only continuous state variable is capital. We have S value functions that are functions of capital alone.

Continuous versus discrete shocks

- For now we assume that the true stochastic process has finite support
- In applied work one typically assumes AR(1) or other continuous processes
- Our assumption can be viewed as a discretization of continuous shock (Tauchen and Hussey), but there are problems with this
- Will also look at continuous shocks but there we also need to approximate integrals $\rightarrow$ numerical integration

Value Function Iteration with Discrete Spate Space

0. Choose:
   a) Capital grid $k = \{k_1, k_2, \ldots, k_n\},\ k_j < k_{j+1}$
   b) Guess for value functions $V_s^{(0)} = \{V_s^{(0)}(k_1), V_s^{(0)}(k_2), \ldots, V_s^{(0)}(k_n)\}$, $s = 1, ..., S$
   c) Stopping rule parameter $\epsilon > 0$
1. Update the value function
   1. $\forall s, i, j$: if $\bar{f}({}_s k_i) - k_j > 0$, compute

$$T_{i,j} = u(\bar{f}_s(k_i) - k_j) + \beta \sum_{s'} \pi(s'|s) V_{s'}^{(0)}(k_j),$$

      else $T_{i,j} = -huge$
   2. $\forall i$: $V_s^{(1)}(k_i) = \max\{\{T_{i,j}\}_{j=1}^n\}$; store index of maximum, j_i^{max}, as identifier of optimal choice, $k_{j_i^{max}}$
2. Check stopping rule: if $\| V^{(1)} - V^{(0)} \|_\infty < \epsilon(1 - \beta)$, stop and report results; else $V^{(0)} = V^{(1)}$ and go to step 1.

Value Function Iteration with Interpolation

- Approximate value function with a polynomial
- Pick some small number of points and compute exact Bellman operator at these points
- Interpolate...
  - Occasionally binding constraints...
  - Shape preservation....
  - Many dimensions...
  - Discrete choices :(

Back to the Ramsey problem

- Suppose we use polynomials of degree d to approximate the value function
- Given some upper and lower bounds for capital $\underline{K} < \overline{K}$ define value function on $[\underline{K}, \overline{K}]$, translate Chebychev zeros to obtain $n = d + 1$ points $k_1, \ldots, k_n \in [\underline{K}, \overline{K}]$
- Fix some initial coefficients θ^0.
- Iterate, given $V(., \theta^i)$ solve for V^{i+1} at all points $k_1, ..., k_n$ and for all values of the shock $z \in \mathcal{Z}$.
  This can be done with Newton's method!
  Interpolate to find θ^{i+1}
- Hope that it converges (it should)

Easy dynamic programming - comments

- Easy to extent the method to higher dimensions and/or continuous shocks
- Need good function approximation in high dimensions (e.g. Smolyak)
- Need good solvers for Bellman equation
- Need good domains!
- Problems can arise due to non-differentiabilities
- How important is shape-preservation? see e.g. Cai and Judd (2010) and references therein.

An alternative way to solve the Ramsey problem

- Instead of doing slow value function iteration, can solve for the policy function.
- Can transform the problem into a functional equation
- $\rightarrow$ projection methods

Reducing EDP to a functional equation

- Suppose the state can be written as $x = (s, y)$ and conditional on s the future y' is a deterministic function of a. We have

$$v^*(x) = \min_{a \in A(x)} c(x, a) + \beta \sum_{s'} \pi(s'|s) v^*(x') \text{ s.t. } y' = p(x, a)$$

- In 'good cases' can consider the first order condition

$$\frac{\partial c(x,a)}{\partial a} + \beta \sum_{s'} \pi(s'|s) \frac{\partial c(x', a')}{\partial y} \frac{\partial p(x,a)}{\partial a} = 0$$

(envelope theorem !)

- Write this as function equation in $\alpha(x)$

An alternative way to solve the Ramsey problem

- Instead of doing slow value function iteration, can solve for the policy function.
- Can transform the problem into a functional equation
- $\rightarrow$ projection methods. Possibly quadratic convergence !

A problem with forward looking constraints

$$\max_{(a_t)_{t=0}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t r(x_t, a_t, s_t) \text{ subject to}$$

$$x_{t+1} = \zeta(x_t, a_t, s_t) \tag{1}$$

$$p(x_t, a_t, s_t) \geq 0 \tag{2}$$

$$g(x_t, a_t, s_t) + E_t \sum_{n=1}^{\infty} \beta^n g(x_{t+n}, a_{t+n}, s_{t+n}) \geq \underline{G}, \ \forall t \tag{3}$$

$$x_0 \text{ given .} \tag{4}$$

Assume that (s_t) follows a finite Markov chain.

Economic applications

- There is a single principal and an agent who contract at $t = 0$. The principal has within period payoff $r(x_t, a_t, s_t)$ which depends upon the state x_t, the action a_t and the shock $s_{t.}$ The agent's period payoff is $g(x_t, a_t, s_t)$. An incentive constraint has to hold every period.
- Agents trade in financial markets but trades must be incentive compatible in the sense that at each node agents walk away from their promise if autarky utility is higher than future expected utility from staying in the market (Kehoe-Levine Alvarez-Jermann). How can we determine (constrained) efficient allocations?

Recursive solutions

- The standard approach to making recursive contracting problems (or game theoretic models) is to use continuation utility as a state variable.
- Marcet and Marimon (1994) develop an alternative recursive saddle point method which is recursive in a cumulation of Lagrangian multipliers.
- Equivalence turns out to rely on concavity assumptions about the original problem which insure strictly concave Pareto frontier.
- Method can be extended to weakly concave Pareto frontiers using lotteries (Cole and Kubler (2012)).

Recursive in utility

The original problem is not recursive in the state x_t because of the forward-looking constraint. It can be recursive in the ex ante promised utility to the agent G (before s is realized) and the state x.

$$V(G, x, s_-) \quad = \quad \max_{a \in A, (G'_s) \in ?} r(x, a, s) + \beta \sum_{s' \in \mathcal{S}} \pi(s'|s) V(G'_s, x', s')$$

subject to

$$g(x, a, s) + \beta \sum_{s \in \mathcal{S}} \pi(s'|s) G'_s \geq G$$

$$g(x, a, s) + \beta \sum_{s \in \mathcal{S}} \pi(s'|s) G'_s \geq \underline{G}$$

$$x' = \zeta(x, a, s)$$

$$p(x, a, s) \geq 0.$$

Recursive in utility - comments

- Need to find some good space for the G...
- This can be computationally very costly
- Simpler to use mutlipliers/Negishi-weigths as state
  Pioneered by Marcet and Marimon, but they got a little confused about duality...

The Lagrange dual function

Consider a finite dimensional constrained optimization problem

$$p^* = \min_f(x) \text{ s.t. } g(x) \leq 0,$$

with $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$.
Can define the Lagrangian

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$$

and dual function

$$h(\lambda) = \min_x L(x, \lambda)$$

Easy to see that for any $\lambda \geq 0$, $h(\lambda) \leq p^*$

The dual problem

Consider now

$$d^* = \max_{\lambda \geq 0} h(\lambda)$$

Note that this is a convex problem and that by the above $d^* \leq p^*$.
We have that 'strong duality holds' if f and g are convex functions and if Slater's condition holds, i.e. there is some x with $g(x) \ll 0$.

A max-min interpretation

Note that

$$\sup_{\lambda \geq 0} L(x, \lambda) = \begin{cases} f(x) & \text{if } g(x) \leq 0 \\ \infty & \text{otherwise} \end{cases}$$

This obviously implies that

$$p^* = \inf_x \sup_\lambda L(x, \lambda)$$

So we have seen

$$\inf_x \sup_\lambda L(x, \lambda) \geq \sup_\lambda \inf_x L(x, \lambda)$$

and $=$ if there is no duality gap

Saddle point

A saddle point of a function $f(w, c)$ is w^*, c^* such that

$$f(w^*, c) \leq f(w^*, c^*) \leq f(w, c^*) \text{ for all } w, c$$

If $f(.)$ has a saddle point there is no duality gap and

$$\inf_w \sup_c f(w, c) = \sup_c \inf_w f(w, c)$$

So: if (x, λ) is a saddle point of the Lagrangian then there is no duality gap and (x, λ) are optimal solutions.

Lagrangian of forward looking problem

The Lagrangian for the dynamic problem (just using IC-constraints) is

$$L\left((\lambda_t),(a_t);x_0,s_0\right)=$$
$$E_0^{\psi}\sum_{t=0}^{\infty}\beta^t\left(r(x_t,a_t,s_t)+\lambda_t(\sum_{n=0}^{\infty}\beta^n g(x_{t+n},a_{t+n},s_{t+n})-\underline{G})\right)$$

Saddle point for forward looking problem

- If $[(\lambda_t)^*, (a_t)^*]$ is a saddlepoint, then

$$L[(\lambda_t)^*, (a_t)^*] = \min_{(\lambda_t)} \max_{(a_t)} L((\lambda_t), (a_t)) = \max_{(a_t)} \min_{(\lambda_t)} L((\lambda_t), (a_t)) .$$

- Existence of a saddle point with (λ_t) in a nice space is a bit of a complicated problem, but let's assume that there is one

Recursive formulation

We can construct *co-state* variables from the Lagrangian multipliers

$$\gamma' = \gamma + \lambda.$$

The co-state variable summarizes the impact of the past incentive and the initial participation constraints.

$$F(\gamma, x, s) = \sup_{a \in A} \inf_{\lambda \geq 0}$$
$$\left[r(x, a, s) + \gamma g(x, a, s) + \lambda (g(x, a, s) - \underline{G}) + \beta E_s F(\gamma + \lambda, x', s') \right]$$

subject to

$$x' = \zeta(x, a, s)$$
$$p(x, a, s) \geq 0.$$

Key is that this problem does not have the future in its constraints and hence is recursive

The role of strict concavity

Suppose we want to solve the following problem

$$V = \max_{(a_1,a_2)\in[0,1]^2} -a_1 - a_2 \text{ subject to}$$
$$a_1 + a_2 \geq 1$$

The value of the corresponding Lagrangian can be obtained recursively (or by backward induction). We can write

$$\begin{aligned} V_2(\lambda) &= \max_{a_2\in[0,1]} (-a_2 + \lambda a_2) \\ V_1 &= \max_{a_1\in[0,1]} \min_{\lambda\geq 0} (-a_1 + \lambda(a_1 - 1) + V_2(\lambda)) \end{aligned}$$

and get the correct value of the saddle point

$$V_1 = \max_{(a_1,a_2)\in[0,1]^2} \min_{\lambda\geq 0} (-a_1 - a_2) + \lambda(a_1 + a_2 - 1).$$

The role of strict concavity (2)

However, one cannot recover the correct policies from this if one takes the Lagrange multiplier as a state variable: The correspondence solving the problem in the second period is given by

$$a_2(\lambda) = \arg\max_{a_2 \in [0,1]} (-a_2 + \lambda a_2) = \begin{cases} 0 & \lambda < 1 \\ [0,1] & \lambda = 1 \\ 1 & \lambda > 1. \end{cases}$$

So clearly, one element of the recursive problem's argmax is $a_1 = 1/2$, $\lambda = 1$ and $a_2 = 0$. But this is not a feasible point. Another element of the recursive problem's solution is $a_1 = 1/2, \lambda = 1$ and $a_2 = 1$. This is a feasible point, but obviously suboptimal.