Algorithm for Optimal Growth Model in 2 periods

Dr. Guanyi Yang

St. Lawrence University

February 7, 2019

Plan

Target

Solving a One-sector two-period optimal growth model

Algorithm

- **discrete state space methods** restrict functions to a state space that is a finite grid of values
- Backwards induction: solving finite period model from the last period back

Optimal Growth Model

For the sake of simplicity, we assume HH's don't value leisure, hence everybody is working full time, $N = 1$.

$$
\begin{aligned}
V = \max_{c,c',K'} &\{u(c) + \beta u(c')\} \\
&\text{s.t.} \\
c + K' &= F(K,1) + (1-d)K \\
c' &= F(K',1) + (1-d)K' \\
&\text{K given}
\end{aligned}
\tag{1}
$$

Algorithm scratch

- Given the finite horizon, the economy is going to use up all resources in the second period. Hence, using up all resources is the optimal choice for the second period.
- Knowing the second period choice and utility value, we find the first period choice, and lifetime value.

End of period problem

Let's look at the last period problem:

$$\begin{aligned} V_2(K) &= \max_c \{u(c)\} \\ &\text{s.t.} \\ c &= F(K,1) + (1-d)K \end{aligned} \tag{2}$$

State variable

First thing: identifying state variable

State variable carries information from previous periods on.

State variable

First thing: identifying state variable

State variable carries information from previous periods on.

- labor-leisure variables L, N only concerns about today;
- consumption c doesn't depend on previous consumption;
- Capital K, however, is given (from previous period's investment and leftover capital).

Therefore, K is the state variable.

State variable determines the **state** of the economy now. All the rest of the variables are valued **given** the state variable.

State space

The state space for K is R_+. However, we cannot solve the model in an infinite space.

- Replace space for capital, R_+, with a finite set of values, $Kgrid$.

  $$Kgrid = \{k_1, k_2, ..., k_n\}$$

  The corresponding vector of indices is:

  $$IK = \{1, 2, ..., n\}$$

  , with $\#IK = n$
- Discrete state-space methods constrain $K_t \in Kgrid$ for each $\{t = 0, 1...\}$

State space approach for the end period

Let me assume the utility function $u(c) = log(c)$.
We have a functional equation in v with K as state variable:

$$V_2(K) = log(F(K,1) + (1-d)K) \tag{3}$$

Algorithm:

For each value of K, we find the value of $V_2(K)$.

State space approach for the end period

1. Initiate memory space for consumption c with the same size as the state variable
2. Initiate memory space for value V_2 with the same size as the state variable
3. For each $i \in IK$, find $c(i) = F(K(i), 1) + (1 - d)K(i)$.
4. For each $i \in IK$, find $V_2(i) = log(c(i))$.

State space approach for the first period

Now we have already found the value for the second period, the first period problem becomes (with $V_2(K')$ known):

$$V(k_i) = \max_{0 \leq K_j \leq F(K_i,1)+(1-d)K_i - K_j} \{log(F(K_i,1) + (1-d)K_i - K_j)$$
$$+\beta V_2(K_j)\} \tag{4}$$

Notice here I use i to represent today's value and j to represent tomorrow's value.

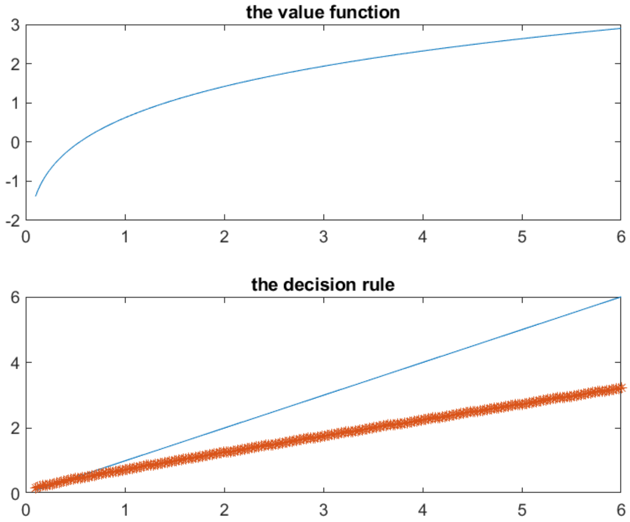
Algorithm:

For each value of K_i, we find any choice of K_j that will return the highest $V(K_i)$.

State space approach for the first period

1. Initiate memory space for consumption $c1$ with the size of current state and all possible future state choices $[Knum, Knum]$
2. Initiate memory space for all possible lifetime value V_1, same as $c1$
3. For each $i \in IK$, and for each $j \in IK$, find $c1(i,j) = max(F(K(i),1) + (1-d)K(i) - K(j), 0)$.
4. For each $i \in IK$, and for each $j \in IK$, find $V_1(i,j) = log(c(i,j)) + beta V_2(j)$.
5. To find the optimal K_j, given each K_i, we record the location (loc, an IK index) that makes $V_1(i,:)$ the largest.
6. Decision rule $gk(i) = Kgrid(loc)$ becomes the optimal choice of K' and $V(i) = V_1(i, loc)$ becomes the maximized lifetime value given inital capital.

Result



Production function: $F(K, 1) = AK^{\alpha}$

Parameters: $\alpha = 0.36; \beta = 0.96; d = 0.069; A = 1;$