Homework 3
ECON 8050: Macroeconomics II
Svetlana Pashchenko

Problem 1. Dynamic programming

Consider the following model with disability shock. There are three sources of uncertainty: out-of-pocket medical shock that evolves according to transition matrix $\Psi(x_t|x_{t-1})$; productivity that evolves according to $T(z_t|z_{t-1})$ and disability shock. Denote the invariant distribution of productivity shocks as $T^{inv}(z)$.

The timing of events is as follows. In the beginning of the period, an individual (with savings k_t) learns his productivity z_t and medical shock x_t. Then he decides whether to work or not ($l_t = 0$ or $l_t > 0$). If an individual works, his labor income is wz_tl_t where w is wage. Then an individual decides about his consumption c_t and savings for the next period k_{t+1}.

In the end of the period, the disability shock is realized. Disability shock arrives with probability d. Individuals who become disabled permanently stay disabled. They do not work, receive constant benefits DI and make only consumption/savings decisions. Medical spending of disabled individuals is fully covered by public insurance.

1. Write down the dynamic programming problem of a non-disabled individual. Denote the value function of a non-disabled person as V_t and the value function of a disabled person as V_t^d. To get full credit, you have to correctly specify all the state variables and budget constraints.

2. Write down the dynamic programming problem of a disabled individual

3. Assume now that disabled individuals can recover from disability with probability f realized in the very end of the period. A recovered individual draws new productivity realizations from the invariant distribution. Write down the modified dynamic programming problem for the disabled. (Hint: now you have to keep track of x_t as a state variable. Think why this is the case).

4. Continue with the setup of part 3, but with the following modification. Assume that non-disabled individuals can pretend to be disabled, i.e., to falsely claim disability benefits. The choice has to be made in the beginning of the period. An individual who falsely claims disability has to decide in the beginning of each period whether to continue claiming benefits or to go back to work. Also, false claimants can be hit by disability shock and become truly disabled; while truly disabled can recover but continue to pretend to be disabled. Write down the new dynamic programming setup.

Hint1: you have to distinguish between truly disabled (denote value function V_t^d), and falsely disabled (V_t^{fd}).

Hint2: note that falsely disabled individuals who go back to work do not draw their productivity realization from the invariant distribution but from the conditional distribution.

Hint3: a non-disabled individual should decide in the beginning of the period between falsely claiming disability (V_t^{fd}) or not (V_t^w). Thus V_t should be a maximum of these two value functions.

Problem 2.

Consider a consumer who has infinite lives, maximizes quadratic utility $\left(u(c_t) = -\frac{1}{2}(c_t - \bar{c})^2\right)$, discounts the future at the rate β and can freely borrow/save at the interest rate r. Denote savings made at period t as a_{t+1}. Assume $\beta(1+r) = 1$.

Suppose the endowment of the consumer y_t is i.i.d.; y_t can take two values: y_H and y_L with probabilities p_H and p_L, respectively.

Note, the budget constraint of the consumer at time t can be written as $c_t = a_t(1+r) + y_t - a_{t+1}$.

1. Solve for the consumption function and for the saving function. Provide intuition: when savings are positive/negative?

2. Now assume the consumer faces borrowing constraint, $a_{t+1} \geq 0$. Set up the consumer's optimization problem in recursive form and solve it on the computer (e.g., using Matlab). Set the following parameters' values:

$$r = 0.02,\ \beta = \frac{1}{1+r},\ \bar{c} = 100,\ y_L = 0.05,\ y_H = 0.5,\ p_L = 0.6,\ p_H = 0.4$$

.

Hint: Follow the steps

(a) Create a grid for assets, $agrid = \{a_1, ..., a_n\}$. Download the file Agrid.txt from the course website.

(b) Guess matrix V^0 dimension $(n \times 2)$, $V^0(i,m)$ corresponds to the value function when $a = a_i$ and $y = y_m$ (you can guess $V^0 = 0$). Initialize matrices for optimal consumptions (copt) and assets (aopt), both have dimension $(n \times 2)$: $aopt = 0,\ copt = 0$.

(c) Set tolerance criteria, say, $tol = 10^{-10}$.

(d) Start iteration on a value function. Loop over state variables

$$(i = 1 \dots n,\ m = L, H)$$

$$V^1(i,m) = \max_j \{u(a_i(1+r) + y_m - a_j) + \beta p_L V^0(j,L) + \beta p_H V^0(j,H)\}$$

Compare V^1 and V^0. If $\mid V^1 - V^0 \mid <\ tol \Rightarrow$ you are done.

Else, set $V^0 = V^1$ and continue.

Specifically, you should look at the largest difference element by element (use MAX command in Matlab).

Keep track of j^*:

$$j^* = argmax\{u(a_i(1+r) + y_m - a_j) + \beta p_L V^0(j,L) + \beta p_H V^0(j,H)\}$$

Create a matrix:

$$jopt(i,m) = j^*$$

$$aopt(i,m) = a_{j^*}$$

$$copt(i,m) = a_i(1+r) + y_m - a_{j^*}$$

3. Plot policy functions a_{t+1} and c_t as a function of current assets a_t . Do this for the case with borrowing constraint (part(2)) and for the case without borrowing constraint (part(1)). Do it for $y_t = y_L$ and $y_t = y_H$. Organize your graphs as follows:

Top left: a_{t+1} (with and without borrowing constraint) for $y_t = y_L$.

Bottom left: same for $y_t = y_H$.

Top right: c_t (with and without borrowing constraint) for $y_t = y_L$.

Bottom right: same for $y_t = y_H$.

4. Simulate the income process and the consumer optimal decision rules for $T = 100$ period. Start with $a = 0$.

On the same graph, draw a_{t+1} for the case of no borrowing constraint (that you found in part (1)) and for the case of borrowing constraint (from part (2)).

On another graph, compare c_t for the cases with and without borrowing constraint. For the latter case, report (as a %) number of periods, when the borrowing constraint is binding (i.e., $a_{t+1} = 0$).

Hint: to simulate $\{y_t\}_{t=1}^{100}$, draw a uniform random number (rand in Matlab). If this number is $< p_L \Rightarrow y = y_L$, else $y = y_H$. Also, keep track of j^*, i.e., create vector jtime(T), jtime(t)=j^*.