Minimum Consumption and Borrowing

Go back to LuuRobFan Project Main Page

Table of Contents

Natural Borrowing Constraint	1
Minimum Consumption	1
Graphical Illustration of the Natural Borrowing Constraint as Probability of Bad State Diminishes	2
Effects of Utility of a Minimum Consumption	4

Natural Borrowing Constraint

In the standard borrowing problem--natural borrowing constraint--with uncertainty in the future. Due to negative infinite utility when consumption is zero, households' borrowing is constrained by endowment in the worst state of future shocks. This leads to the following result:

Given a two period economy where endowment are:

$$e_1 = 0$$
, and, $e_{2,l} = 1$ or $e_{2,h} = 3$

where p_l is the probability of the low state

The optimal choice is:

$$b^*(p_l, e_{2,l}, e_{2,h}) = \arg\max_b \left[\log(b) + p_l \cdot \log(e_{2,l} + b) + (1 - p_l) \cdot \log(e_{2,h} + b)\right]$$

If $p_l = 0$, households would borrow exactly b = -1.5, but because $e_{2,l} = 1$, as p_l converges to 0, the limit of the optimal borrowing choice not -1.5, but -1.

$$b^*(p_l=0,e_{2,h}) = -\frac{e_{2,h}}{2} \neq \lim_{p_l \to 0} b^*(p_l,e_{2,l},e_{2,h}) = -e_{2,l}$$

This results happens when households are required to always repay their debts. Non-full-repayment will be avoided even the probability of that occuring is very small.

Minimum Consumption

We can allow for higher borrowing choices by allowing for minimum consumption. Minimum consumption allows for default. We can write the problem as:

$$\max_{b} \left[\log(\hat{c_1}) + p_l \cdot \log(\hat{c_{2,l}}) + (1 - p_l) \cdot \log(\hat{c_{2,h}}) \right]$$

where $c_1 - b = e_1$, $c_2 = e_{2,l} + b$, and $c_3 = e_{2,h} + b$

and **Minimum Consumption**: $\hat{c} = \begin{cases} c, \text{ if } c \ge c_{\min} \\ c_{\min}, \text{ if } c < c_{\min} \end{cases}$

Without additional constraints, the introduction of minimum consumption gives households the possibility of increasing utility by borrowing ever more (increasing current utility), or saving towards infinity (increasing future utility).

• **Borrowing Constraint**: To prevent households from borrowing infinitely, we impose a limit a borrowing that is non-binding:

(non-binding) borrowing constraint: $b \ge \overline{b}$

• **Debt Continuation Condition**: To prevent households from saving infinitely, we modify the second period budget constraints as below:

updated budget constraints: $c_2 = e_{2,l} + \hat{b}$, and $c_3 = e_{2,h} + \hat{b}$

where:
$$\hat{b} = \begin{cases} b, \text{ if } c \ge c_{\min} \\ b + (c_1 - c_{\min}), \text{ if } c < c_{\min} \end{cases}$$

The **Debt Continuation Condition** is important also for delaying payment when households are unable to fully repay debts due for models that have no-roll-over conditions. In finite horizon model here, this condition will allow for possible delay in repayments until the final period.

Graphical Illustration of the Natural Borrowing Constraint as Probability of Bad State Diminishes

The natural borrowing constraint. The graph below demonstrates the limitation on borrowing when there is no minimum consumption.

```
% Define Some Parameters
e1 = 0;
e^{2h} = 3;
e21 = 1;
beta = 1;
r = 0;
% Solve for Optimal Choices as the probability of the low state happening decreases
p_n = 1000;
p vector = linspace(0, 1, p n);
b_o_vector = zeros(size(p_vector));
% Define function
fu_bp = @(b, p, e2h) (log(e1 - b) + beta*(p*log(e2l + (1+r)*b) + (1-p)*log(e2h + (1+r)*b)));
for p_i=1:length(p_vector)
    % Grab p
    p = p vector(p i);
    % Function of b only
```

```
fu_b = @(b) (-1)*fu_bp(b, p, e2h);
    % Find Optimal Choice
    options = optimset('Display', 'off');
    b_o_vector(p_i) = fminsearch(fu_b, -0.1, options);
end
% Plot Optimal Choices
figure();
hold on;
plot(p_vector(2:p_n), b_o_vector(2:p_n));
scatter(p_vector(1), b_o_vector(1), 100, 'k', 'filled');
% Graph Bounds
b_o_max = max(b_o_vector);
b_o_min = min(b_o_vector);
b_o_margin = (b_o_max - b_o_min)/10;
b \circ graph max = b \circ max + b \circ margin;
b o_graph_min = b_o_min - b_o_margin;
p vector max = max(p vector);
p_vector_min = min(p_vector);
p_vector_margin = (p_vector_max - p_vector_min)/10;
p_vector_graph_max = p_vector_max + p_vector_margin;
p_vector_graph_min = p_vector_min - p_vector_margin;
xlim([p_vector_graph_min p_vector_graph_max]);
ylim([b o graph min b o graph max]);
% Grids
grid on;
grid minor;
% Titling
title({['Natual Borrowing Constraint'], ['Optimal Borrowing Choice']});
xlabel('Probability of the Bad State');
ylabel('Optimal Borrowing Level');
legend({['Optimal Borrowing as P_{low}] gets Small']...
       , ['Optimal Borrowing when P {low} = 0']}...
       ,'Location', 'southeast');
```



Effects of Utility of a Minimum Consumption

What happens to the utility objective when we allow for minimum consumption? The example below illustrates this graphically. Minimum consumption opens up new set of b choices that were previously not available to households. But now there are two local maximums. Searching for optimal choices is trickier in this context.

Visually, we can also see if borrowing and savings bounds are not considered, then households would find it optimal to borrow or save infinitely.

```
% r and beta
beta = 1;
r= 0;
c_min_v = 0.09;
% Define the conditional consumption function
fc = @(c, cmin) (c).*(c >= cmin) + (cmin).*(c < cmin);
% Utility Function
fu_bp_cmin = @(b, p, e2h, cmin) log(fc(e1-b, cmin))...
+ beta*(p*log(fc(e21+b*(1+r), cmin)))...
+ (1-p)*log(fc(e2h+b*(1+r), cmin)));
% Bad State has 50 percent chance of happening
p = 0.50;
```

```
% A vector of b choice grid, include very large borrowing choices
b \min = -4;
b max = 1;
b_margin = (b_max - b_min)/10;
b n = 1000;
b_vector = linspace(b_min, b_max, b_n);
% Evaluate Utility
util vector = zeros(size(b vector));
% Evaluate Function
util_cmin = fu_bp_cmin(b_vector, p, e2h, c_min_v);
[max val cmin, max idx cmin] = max(util cmin);
opti b cmin = b vector(max idx cmin);
util orig = fu bp(b vector, p, e2h);
util_orig_real_idx = arrayfun(@(x) ~any(imag(x)), util_orig);
% Graph Utility
figure();
hold on;
scatter(opti b cmin, max val cmin, 100, 'black', 'filled');
plot(b_vector, util_cmin, 'LineWidth', 2);
plot(b_vector(util_orig_real_idx), util_orig(util_orig_real_idx), 'b--', 'LineWidth', 2);
% Y and X limits
b grh max = b max + b margin;
b grh min = b min - b margin;
util_max = max(max([util_orig(util_orig_real_idx) util_cmin]));
util min = min(min([util orig(util orig real idx) util cmin]));
util_margin = (util_max - util_min)/5;
util_grh_max = util_max + util_margin;
util_grh_min = util_min - util_margin;
xlim([b grh min b grh max]);
ylim([util grh min util grh max]);
% Titling
title(['opti b with cmin, opti b=' num2str(opti_b_cmin)])
% Grids
grid on;
grid minor;
ylabel('Utility');
xlabel('B choices');
legend({['Utility with Cmin = 0']...
       ,['Utility with Cmin > 0']}...
       ,'Location', 'southwest');
```

