Problem Set 1 – Patent Renewal

Brian Higgins Ciaran Rogers

Introduction

In this problem set we will cover a concept called **dynamic programming**. To do it, you should be familiar with MATLAB syntax and basic concepts of recursion and loops. If you are not – please review the script and slides from the first MATLAB session and email us if you have any questions.

Dynamic programming is a fancy way of saying "solving a complex problem by reducing it to many simple ones". It was invented by applied mathematicians in the early 50's that wanted to solve complex problems using computers (most notably Richard Bellman).

In economics, the problems that lend themselves to this type of solution are usually intertemporal problems, e.g. how much to consume each period, whether to renew a patent, or whether to invest in a new technology. The basic common property of these problems is that having the solution for the problem at the next period makes it way simpler to solve for this period. **In the fall quarter you will study dynamic programming in the context of macroeconomics.** But this technique is not confined to macro – it has also been successfully applied in labor, IO and finance. The problem set is meant to be solved in groups (up to 4 students), though it can also be solved individually. In particular, if you feel less comfortable with programming, make sure you team up with someone that is more skilled and work together on this problem. As in everything in the first year, the most important thing is to build new skills.

Patent Renewal

Consider the following problem from IO, based on ideas of Pakes (1986). A firm owns a patent at period 0. At each period 1:T the patent may turn out to be "realized" with probability p, and generate a cash flow of $150 from that period on. The patent expires at the end of period T, or at the end of the period in which the firm decides not to renew it. Once expired, it can never be renewed. Renewal cost is $100 per period. The firm discounts cash flows at a constant period rate of r. This means that the present value for a given cash flow x_t is the sum PV = x_0 + x_1/(1+r) + x_2/(1+r)^2 +...

It is intuitive that the firm will want to renew if p is high enough (in the limiting cases this is easy to show – it generates a net cash flow of $50 each period when p = 1, and -$100 when p = 0). But how high exactly should p be for the firm to renew at period 0? Equivalently, given p, what is the value of the patent for the firm? One way to answer these questions would be to consider all possible strategies, and find the present value of their cash flows. The value would be the maximal present value. But any mapping from states to decision can be considered as a valid strategy and this may turn out to be a very large set to cover!

Dynamic programming can be used to efficiently go over this set. Instead of asking "what is the optimal strategy?" we ask: given the value of an "unrealized" and a "realized" patent at period t+1, what is the value of the patent in period t?

The answer to that question is simpler. If the patent was "realized" then its value must be

V(realized, t) = \$150 - \$100 + V(realized,t+1)/(1+r).

That is, the value of this period's earnings minus renewal fee, plus the discounted present value next period. If the patent was not realized, its value must be the maximum of \$0 (if you choose to scrap the patent) and the discounted expectations over the value next period net of the renewal fee (patent cannot be worth less than zero! Why?). That is:

V(unrealized,t) =max{\$0,-\$100 + [p*V(realized,t+1) + (1-p)*V(unrealized,t+1)]/(1+r)}.

Where we multiply the different possible values by the probability that they will actually happen. But wait, how do we know what V(.,t+1) is? Simply go to the last period, and set V(.,T+1) = 0. That's it! Now we can move backward to find values in any period t.

In the following three questions you are going to be asked to solve variations of this problem and to analyze the results. Good luck with your first dynamic program.

Question 1: Recursion

Recursion is a programming technique where you call a function from within that same function. If written correctly, it can give you the most readable code. This simple example can help you understand its power as well as its limitations. If this is the first time you see this concept, please go over the script from the session on basic MATLAB.

Please write code for the following function, using recursion:

```
function [v,d] = patent_value(p,r,realized,k,K,t,T)
```

Where input argument p is the probability of realization, r the discount rate, realized is an indicator that equals 1 when the patent is realized and 0 otherwise, k is the cost of renewal, K is the cash flow if realized, t is current period, and T is last period before the patent expires. Output v is the value and d is the choice to renew (1) or not (0).

Hints:

1. What is the patent value if t > T? Does it depend on realization?
2. Use the value returned from `patent_value` at t+1 to compute the value at t. No other computation is required.
3. If you call the function
   `v = patent_value(p,r,realized,k,K,t,T)`
   then MATLAB returns the first output variable only.

Answer the following questions using the function and any supporting code:

1. Given parameter values from the introduction, and T=20, p = 0.05, r = 0.05, what is the value of an unrealized patent at period 0? What is the value when p = 0.5?
2. Plot the value of the patent for p in between 0 and 1. Make sure that your plot shows discontinuities, if they exist. On the same graph, plot the present value of the patent when the renewal strategy is “always renew”. What is the source of the difference (in words)?

Question 2: Loops

Write a function called patent_value0, that has all the same input arguments except realized and t, and computes the value for unrealized and realized patent for all periods 1,…,T.

Hints:

1. Start by initializing a 2 by (T+1) matrix where you will store all values of V.
2. Write a main loop that goes through t from T back to 0. That way, at each period t, values for any period > t will already be stored in memory. Remember that MATLAB indexes start at 1.
3. Within the main loop, write a loop that goes through realization states. This is called a nested loop. It is not necessary for this case, but a good practice. Consider that instead of a binary state space, you may have N distinct cases.

Answer the following questions using the function and any supporting code:

1. Pick interesting parameter values, explain why you picked them and plot a graph of value function against time.
2. Can you describe the firm strategy using the results? When does it renew and when not?
3. What do you think was the advantage of writing the program as a loop?

Question 3: Large State Space

We change the problem in the following way. The cash flow $x of the patent is uncertain and can grow or shrink over time. Instead of having two states ("realized" or "unrealized"), consider a patent that generates cash flow $x at period t, and at period t+1 x*q with probability p and x/q with probability 1-p. All the other details stay the same: the patent expires after period T and there is a renewal fee that has to be paid each period, otherwise the patent rights are lost.

Write code that computes the value of the patent across states and across cash flow realization.

Hints:

1. Use either recursion or loop (or better yet – use both). Then use tic/toc commands to time the execution. Why do you think one is faster than the other?
2. State space is now bigger than 2. For a given initial value of x at time 0, how many possible values are there at time 2? Time 3? Time t? Construct a matrix that can contain all new states.
3. Instead of using "if" conditions, use the vector version of "max" and the index output. That way you can take full advantage of matrix notation in MATLAB.

Answer the following questions using the function and any supporting code:

1. Using mesh command, present the value function in a 3d graph. Try arguments p = 0.8, q = 1.03, r=1.05, k = 100, 60<x<100, T = 50. Use colors to show the firm decision at each state.
2. How would this analysis change if renewal fees were eliminated?