

Job Search Models and Code

Pranay Gundam

June 10, 2024

Contents

1	The Basic Job Search Question	1
2	Value Function Iteration (VFI)	2
2.1	Stability of VFI	3
2.2	VFI Pseudo-code	3
2.3	Speeding up VFI	3
3	Common Extensions of the Job Search Question	3
4	Code Documentation	3
5	What more to add?	3

1 The Basic Job Search Question

Consider a worker who lives infinitely and lives their life in discrete time periods $t = 0, 1, \dots$. They possess some temporal discount factor $0 < \beta < 1$ and also have an increasing concave utility function $u(\cdot)$. At each time period t , this worker can either be employed or unemployed. If they are employed then they will have been given some job offer in the past they have accepted which pays them a wage w . If they are unemployed, in the beginning of the time period they draw an employment offer w from some distribution $F(\cdot)$ and then have the choice to accept the job for life such that they earn that wage w for the rest of their lives or receive a unemployment benefit b in that period to remain unemployed in the beginning of the next period.

This is the basic setup of the problem (we will discuss other extensions of the problem as well later) from which point there are a few questions that are typically asked such as: at any point in time, if the worker is unemployed, what is the smallest wage from a job offer they would accept. This involves solving for the opportunity cost of taking the offer which we can set up as

a dynamic programming problem.

Specifically we can say that when the unemployed worker is faced with a job wage offer w they are to maximize the value function

$$v(w) = \max_{accept, reject} \left\{ \frac{w}{1-\beta}, b + \beta \int v(w') dF(w') \right\}.$$

The first term in the maximization represents the closed form of the geometric series of all the future income the worker will receive as a result of taking the job offer and the second term represents getting an unemployment payment in the current period and the expected value of the value function itself (which we get by changing the integral to be with respect to the probability distribution of the wage draws); this is the term that represents the opportunity cost.

For the basic setup, we are given the discount factor, unemployment benefit, and probability distribution for the wage draws so the only difficult concept to consider now is the recursive formulation of the value function. We can't solve for an explicit form since the value function is itself in part defined by an integral over it. One of the most common ways to proceed now is to use value function iteration, an iterative algorithm that starts with a guess for what the shape of our value function looks like and continue to refine.

2 Value Function Iteration (VFI)

Value function iteration is one of the analytical processes people use to learn the shape of certain functions and is used a lot in Macroeconomics. The benefit of value function iteration is that it is stable, meaning that we know it will eventually converge to the true solution. It is, however, also very slow and achieving the granularity that we often desire requires us to keep track of a very large statespace especially as we increase the dimensionality of the state we want to track.

The algorithm will look slightly different based on the notation and problem setup but the general idea is the same. For the problem defined above we can do VFI as such: determine the statespace S and initialize a $v_0(s)$ for all $s \in S$. Then determine v_1, v_2, \dots with the update rule:

$$v_k(s) = \max_{accept, reject}$$

One additional key point is the idea of a policy function. This goes hand-in-hand with the value function in that instead of taking a max over the two terms we specified above we take an argmax to identify the optimal action of the agent given a state.

2.1 Stability of VFI

Let v_k be our guess of value function at the k —th iteration of VFI. We can say that VFI is stable iff

$$\lim_{k \rightarrow \infty} v_k = v^*$$

where v^* is the true value function.

2.2 VFI Pseudo-code

2.3 Speeding up VFI

3 Common Extensions of the Job Search Question

4 Code Documentation

5 What more to add?