Iteração da Função valor
Macroeconomia I

Yuri Passuelo – `yuripassuelo@usp.br`

16 de maio de 2025

Aqui vamos revisar e detalhar um pouco mais a respeito do algoritmo da iteração da função valor com complexidades a mais

---

Problema com Incerteza

Vamos considerar o problema em que a produtividade do modelo de crescimento neoclássico sofre um processo markoviana.

Dado o problema vamos ao passo a passo do algoritmo:

Passo 1: Criação dos Grids

Obs: O motivo de criar um Grid de capital se dá pois o algoritmo se trata de uma implementação que seria colocada dentro de um computador, como não conseguimos representar um continuo dentro de um computador, usamos essa definição de Grid que se trata basicamente da discretização de um intervalo continuo.

$$\mathcal{K} = \{k_1, k_2, ..., k_n\}$$

Aonde temos que:

$$\forall i (k_i \geq 0)$$

$$k_i < k_j, \text{ para } i < j$$

Ou seja, nosso vetor de capital se trata de uma lista ordenada de valores de capital, com a restrição de não negatividade desses valores de capital.

$$\mathcal{Z} = \{z_1, z_2, ..., z_m\}$$

Aonde temos que nosso vetor precisa ser ordenado:

$$z_i < z_j, \text{ para } i < j$$

Por último definimos a matriz de transição do nosso problema:

$$\pi_{z'|z} = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \dots & p_{mm} \end{bmatrix}$$

Passo 2: Parametrização da Economia

Dentro do nosso problema básico assim como visto acima trabalhamos com funções utilidade, funções de produção, e temos presente parâmetros em cada uma dessas funções assim como os parâmetros de depreciação do capital δ e de desconto intertemporal β.

Dado que escolhemos por exemplo uma função de utilidade CRRA:

$$u(c) = \frac{c^{1-\sigma}}{1-\sigma}$$

E uma função de produção *Cobb-Douglas*:

$$f(k) = zk^{\alpha}$$

Devemos declarar tanto as funções utilizadas como os parâmetros escolhidos, assim nessa etapa parametrizamos escolhendo δ, β, α e σ.

Passo 3: Criação dos objetivos de iteração

Aqui definiremos uma série de objetos que nos ajudaram na etapa de iteração, são basicamente vetores que vamos guardar os números.

- Matriz de iteração da função valor com dimensão $n \times m$

$$\text{v\_it} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

- Vetor de iteração da função política com dimensão $n \times m$

$$\text{g\_it} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

- Vetor de armazenamento da função valor com dimensão $n \times m$

$$V = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Por último vamos definir nosso parâmetro de tolerância para convergência dado aqui como:

$$\text{tol} \ = 1 \times 10^{-5}$$

Passo 4: Iteração da função Valor

Criar operador Tv de dimensão $1 \times n$ para operar a função valor

$$Tv = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$$

- Para cada $z \in \mathcal{Z}$
  - Para cada $k \in \mathcal{K}$
    * Para cada $k' \in \mathcal{K}$
      Calcularemos o Consumo como:

      $$c = zk^{\alpha} - k'$$

      Se $c \leq 0$:

      $$c = -99999$$

      Obs: Caso $c > 0$ não precisamos modificar em nada c

      Com c calculado, calcularemos:

      $$u(c) + \beta V[k']$$

Obs: Aqui $V[k']$ se refere ao elemento de índice k' do vetor V

Guardamos o valor calculado em Tv:

$$Tv[k'] = u(c) + \beta \mathbb{E}_{z'|z}\big[V[k', z']\big]$$

Obs: Aqui $Tv[k']$ se refere ao k'-ésimo elemento do vetor Tv Obs: Aqui a operação $\mathbb{E}_{z'|z}\big[V[k', z']\big]$ é o produto interno do vetor coluna da matriz $V[;, z]$ e a linha relativa a z da matriz π

Finalizado o *Loop* em relação à k' dado um k, pegamos aqui tanto o valor máximo do vetor Tv quanto seu argumento máximo e salvamo-os em duas variáveis:

$$\text{max\_k} = \text{arg max } (Tv)$$

$$\text{max\_V} = \text{max } (Tv)$$

Guardamos os valores nos vetores de iteração v_it e g_it:

$$\text{v\_it}[k, z] = \text{max\_v}$$

$$\text{g\_it}[k, z] = \text{max\_k}$$

Obs: Aqui v_it$[k, z]$ e g_it,z$[k, z]$ referem-se ao elemento de posição k, z da matriz

Passo 5: Checagem de Convergência

Terminado o processo de iteração vamos a checagem de convergência. Primeiro calcularemos a distância da matriz V para a matriz v_it

$$\text{dist} = \text{max } |V - \text{v\_it}|$$

Caso dist < tol então paramos o processo e temos que v_it será nossa função valor $v(k)$ e g_it será nossa função política $g(k)$.

Caso contrário voltamos ao passo 4, mas antes atualizamos V:

$$V = \text{v\_it}$$

Problema com Trabalho e Incerteza

Vamos adicionar agora mais um elemento que é a escolha de trabalho além da incerteza

Dado o problema vamos ao passo a passo do algoritmo:

Passo 1: Definição dos grids

Primeiro criaremos o grid de Capital.

$$\mathcal{K} = \{k_1, k_2, ..., k_n\}$$

Aonde temos que:

$$\forall i (k_i \geq 0)$$

$$k_i < k_j \text{, para } i < j$$

Ou seja, nosso vetor de capital se trata de uma lista ordenada de valores de capital, com a restrição de não negatividade desses valores de capital.

Agora criaremos o grid de Choques.

$$\mathcal{Z} = \{z_1, z_2, ..., z_m\}$$

Aonde temos que nosso vetor precisa ser ordenado:

$$z_i < z_j \text{, para } i < j$$

Definiremos a matriz de transição do nosso problema:

$$\pi_{z'|z} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix}$$

Por ultimo criaremos o vetor de trabalho

$$\mathcal{L} = \{l_1, l_2, ..., l_p\}$$

Assim como os demais vetores temos que:

$$\forall i (l_i \geq 0)$$

$$l_i < l_j \text{ para } i < j$$

Passo 2: Parametrização da Economia

Dentro do nosso problema básico assim como visto acima trabalhamos com funções utilidade, funções de produção, e temos presente parâmetros em cada uma dessas funções assim como os parâmetros de depreciação do capital δ e de desconto intertemporal β.

Dado que escolhemos por exemplo uma função de utilidade CRRA:

$$u(c) = \log\,(c) + \frac{l^{1+\phi}}{1+\phi}$$

E uma função de produção *Cobb-Douglas*:

$$f(k) = zk^{\alpha}l^{1-\alpha}$$

Devemos declarar tanto as funções utilizadas como os parâmetros escolhidos, assim nessa etapa parametrizamos escolhendo δ, β, α e ϕ.

Passo 3: Criação dos objetivos de iteração

Aqui definiremos uma série de objetos que nos ajudaram na etapa de iteração, são basicamente vetores que vamos guardar os números.

- Matriz de iteração da função valor com dimensão $n \times m$

$$\text{v\_it} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

- Vetor de iteração da função valor com dimensão $1 \times p$

$$\text{vn\_it} = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$$

- Matriz de iteração da função política com dimensão $n \times m$

$$\text{gk\_it} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

- Matriz de iteração da função política do trabalho com dimensão $n \times m$

$$\text{gn\_it} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

- Vetor de armazenamento da função valor com dimensão $n \times m$

$$V = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Por último vamos definir nosso parâmetro de tolerância para convergência dado aqui como:

$$\text{tol } = 1 \times 10^{-5}$$

Passo 4: Iteração da função Valor

Criar operador Tv de dimensão $1 \times n$ para operar a função valor

$$Tv = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$$

Criar vetor gn de dimensão $1 \times n$ para operar decisão de trabalho

$$\text{gn} = \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$$

- Para cada $z \in \mathcal{Z}$
  - Para cada $k \in \mathcal{K}$
    - Para cada $k' \in \mathcal{K}$
      - Para cada $l \in \mathcal{L}$
        Calcularemos o Consumo como:
        $$c = zk^{\alpha}l^{1-\alpha} - k'$$
        Se $c \leq 0$:
        $$c = -99999$$
        Obs: Caso $c > 0$ não precisamos modificar em nada c

Com c calculado, calcularemos:

$$u(c) + \beta V[k']$$

Obs: Aqui $V[k']$ *se refere ao elemento de índice* k' *do vetor* V

Guardamos o valor calculado em Tv:

$$\text{vn\_it}[l] = u(c) + \beta \mathbb{E}_{z'|z}\big[V[k', z']\big]$$

Obs: Aqui $Tv[k']$ *se refere ao* k'*-ésimo elemento do vetor* Tv *Obs: Aqui a operação* $\mathbb{E}_{z'|z}\big[V[k', z']\big]$ *é o produto interno do vetor coluna da matriz* $V[;, z]$ *e a linha relativa a* z *da matriz* π

Guardamos aqui o valor máximo de cada trabalho

$$Tv[k'] = \max(\text{vn\_it})$$

$$\text{gn}[k'] = \arg\max()$$

Finalizado o *Loop* em relação à k' dado um k, pegamos aqui tanto o valor máximo do vetor Tv quanto seu argumento máximo e salvamo-os em duas variáveis:

$$\text{max\_k} = \arg\max\ (Tv)$$

$$\text{max\_n} = \arg\max\ (\text{gn})$$

$$\text{max\_V} = \max\ (Tv)$$

Guardamos os valores nos vetores de iteração v_it e g_it:

$$\text{vk\_it}[k, z] = \text{max\_v}$$

$$\text{gk\_it}[k, z] = \text{max\_k}$$

$$\text{gn\_it}[k, z] = \text{max\_n}$$

Obs: Aqui v_it$[k, z]$*, gk_it,z*$[k, z]$ *e g_it,z*$[k, z]$ *referem-se ao elemento de posição* k*,* z *das respectivas matrizes*

Passo 5: Checagem de Convergência

Terminado o processo de iteração vamos a checagem de convergência. Primeiro calcularemos a distância da matriz V para a matriz v_it

$$\text{dist} = \max |V - \text{v\_it}|$$

Caso dist $<$ tol então paramos o processo e temos que v_it será nossa função valor $v(k)$ e g_it será nossa função política $g(k)$.

Caso contrário voltamos ao passo 4, mas antes atualizamos V:

$$V = \text{v\_it}$$