Macroeconomía Internacional

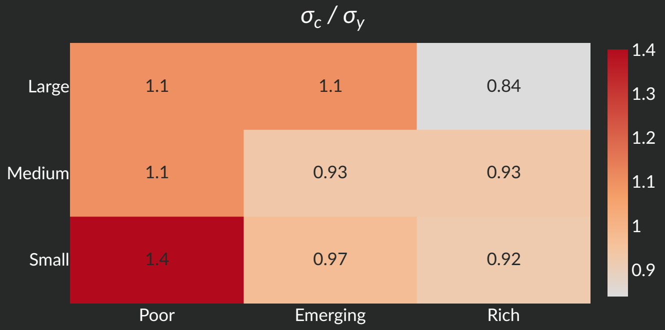
Francisco Roldán
IMF

September 2025

1. Economías emergentes $\neq$ avanzadas

1. Economías emergentes $\neq$ avanzadas

$$u'(c_t) = \beta(1+r)\mathbb{E}\left[u'(c_{t+1})\right]$$

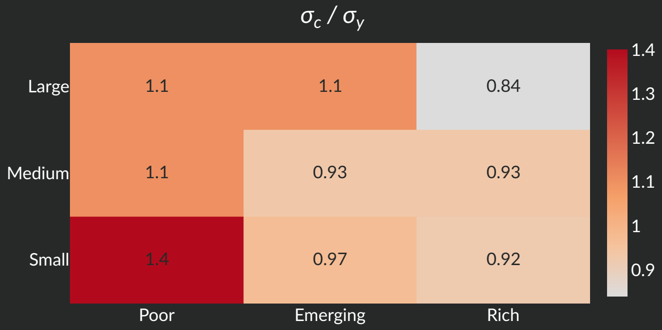


Fuente: Schmitt-Grohé and Uribe (2020)

Macro Internacional

1. Economías emergentes $\neq$ avanzadas

$$u'(c_t) = \beta(1+r)\mathbb{E}\left[u'(c_{t+1})\right]$$



Fuente: Schmitt-Grohé and Uribe (2020)

2. Excusa para métodos

$$v = u + \beta v$$

Plan

Tres modelos

1. RBC con default (Arellano, 2008)
   ... La deuda se paga con el valor presente del superávit, pero **cuándo** se paga la deuda?

2. Modelos con rigideces nominales (Schmitt-Grohé y Uribe, 2016)
   ... Tipo de cambio, externalidades de demanda

3. Deflación fisheriana y sudden stops (Bianchi, 2011)
   ... Cómo el precio del colateral amplifica la salida de capitales

4. Consistencia temporal (Chang, 1999)
   ... Cómo escribir problemas de control óptimo de forma recursiva

Plan

Tres modelos

1. RBC con default (Arellano, 2008)

   . . . La deuda se paga con el valor presente del superávit, pero **cuándo** se paga la deuda?

2. Modelos con rigideces nominales (Schmitt-Grohé y Uribe, 2016)

   . . . Tipo de cambio, externalidades de demanda

3. Deflación fisheriana y sudden stops (Bianchi, 2011)

   . . . Cómo el precio del colateral amplifica la salida de capitales

4. Consistencia temporal (Chang, 1999)

   . . . Cómo escribir problemas de control óptimo de forma recursiva

Plan

Tres modelos

1. RBC con default (Arellano, 2008)

   ... La deuda se paga con el valor presente del superávit, pero **cuándo** se paga la deuda?

2. Modelos con rigideces nominales (Schmitt-Grohé y Uribe, 2016)

   ... Tipo de cambio, externalidades de demanda

3. Deflación fisheriana y sudden stops (Bianchi, 2011)

   ... Cómo el precio del colateral amplifica la salida de capitales

4. Consistencia temporal (Chang, 1999)

   ... Cómo escribir problemas de control óptimo de forma recursiva

Plan

Tres modelos

1. RBC con default (Arellano, 2008)
   ... La deuda se paga con el valor presente del superávit, pero **cuándo** se paga la deuda?
2. Modelos con rigideces nominales (Schmitt-Grohé y Uribe, 2016)
   ... Tipo de cambio, externalidades de demanda
3. Deflación fisheriana y sudden stops (Bianchi, 2011)
   ... Cómo el precio del colateral amplifica la salida de capitales
4. Consistencia temporal (Chang, 1999)
   ... Cómo escribir problemas de control óptimo de forma recursiva

Objetivos

1. Discusión no exhaustiva de la mecánica de los modelos
2. Foco en aplicación cuantitativa
   ... Códigos para resolver, simular, calibrar, graficar

   Por qué?

3. Julia
   - Iteración en la función de valor

Objetivos

1. Discusión no exhaustiva de la mecánica de los modelos

2. Foco en aplicación cuantitativa

   . . . Códigos para resolver, simular, calibrar, graficar

Por qué?

3. Julia

   - Iteración en la función de valor

Objetivos

1. Discusión no exhaustiva de la mecánica de los modelos

2. Foco en aplicación cuantitativa

   . . . Códigos para resolver, simular, calibrar, graficar

   Por qué?

3. Julia

   - Iteración en la función de valor

Objetivos

1. Discusión no exhaustiva de la mecánica de los modelos

2. Foco en aplicación cuantitativa

   ... Códigos para resolver, simular, calibrar, graficar

Por qué?

3. Julia

   - Iteración en la función de valor

Organización

Nosotros

- Teóricas
  - Modelos, algoritmos
- Prácticas
  - Implementación en la compu

Ustedes

[No representation without taxation]

- Presentaciones cortas
- Guías de ejercicios

Organización

Nosotros

- Teóricas
    - Modelos, algoritmos
- Prácticas
    - Implementación en la compu

Ustedes
[No representation without taxation]

- Presentaciones cortas
- Guías de ejercicios

Hoy

- Repaso de programación dinámica

  ... McCall (1970)

  ... Problema de la torta

- Estructura de implementación numérica

$$v = u + \beta v$$

- "qué hacer?"

  es más difícil que

- "qué hacer hoy?"

  entendiendo cómo vas a reaccionar después

Hoy

- Repaso de programación dinámica

  ... McCall (1970)

  ... Problema de la torta

- Estructura de implementación numérica

$$v = u + \beta v$$

- "qué hacer?"

  es más difícil que

- "qué hacer hoy?"

  entendiendo cómo vas a reaccionar después

Programación Dinámica: Búsqueda

McCall (1970)

- Un agente busca trabajo.
- Preferencias standard: utilidad u, descuento β.
- Los trabajos son heterogéneos y sólo difieren en el salario que pagan.
- Cada período llega una oferta de trabajo $w \overset{iid}{\sim} F(\cdot)$
- Sólo se puede aceptar un trabajo. El agente recibe b mientras busca

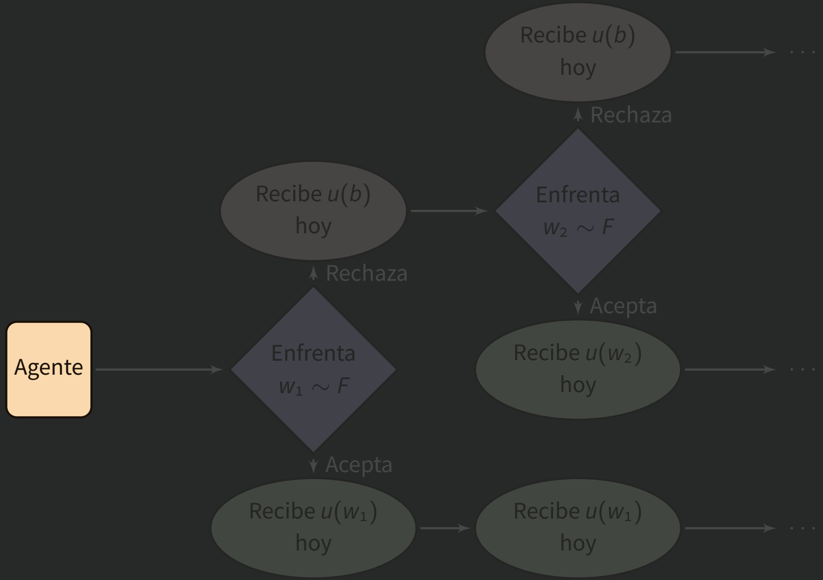
- Cómo decide el agente qué trabajo aceptar?

McCall (1970)

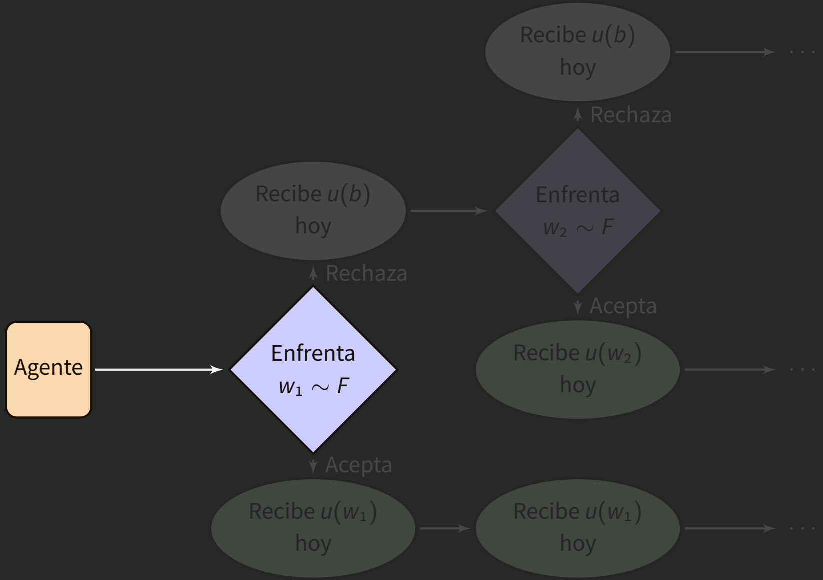
- Un agente busca trabajo.
- Preferencias standard: utilidad u, descuento β.
- Los trabajos son heterogéneos y sólo difieren en el salario que pagan.
- Cada período llega una oferta de trabajo $w \overset{iid}{\sim} F(\cdot)$
- Sólo se puede aceptar un trabajo. El agente recibe b mientras busca

- **Cómo decide el agente qué trabajo aceptar?**

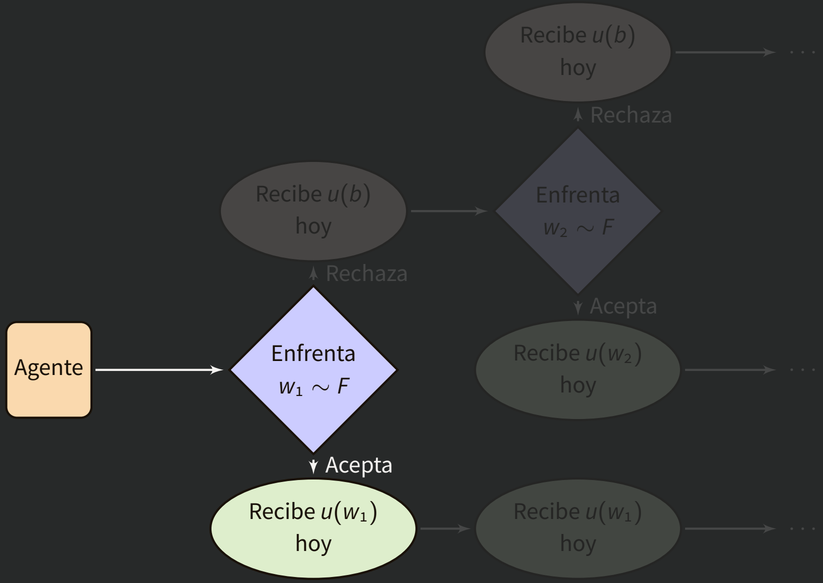
McCall (1970) con un árbol difícil



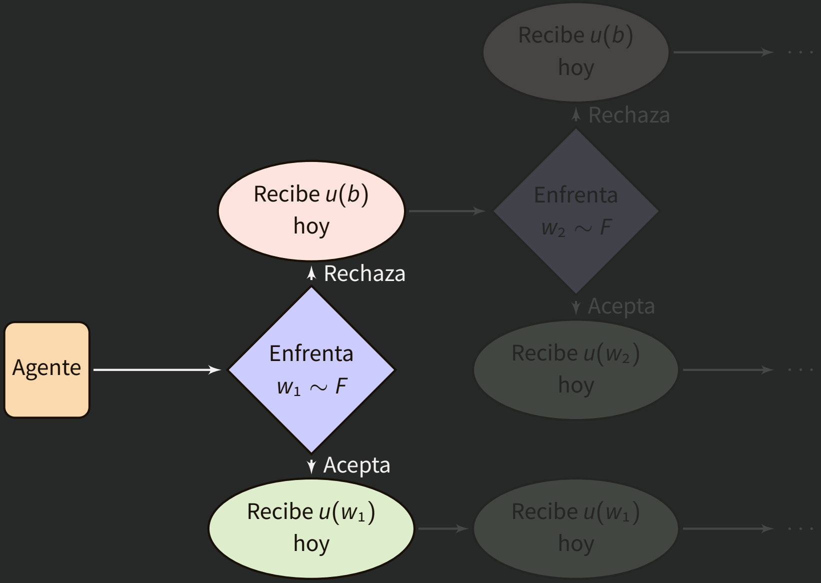
McCall (1970) con un árbol difícil



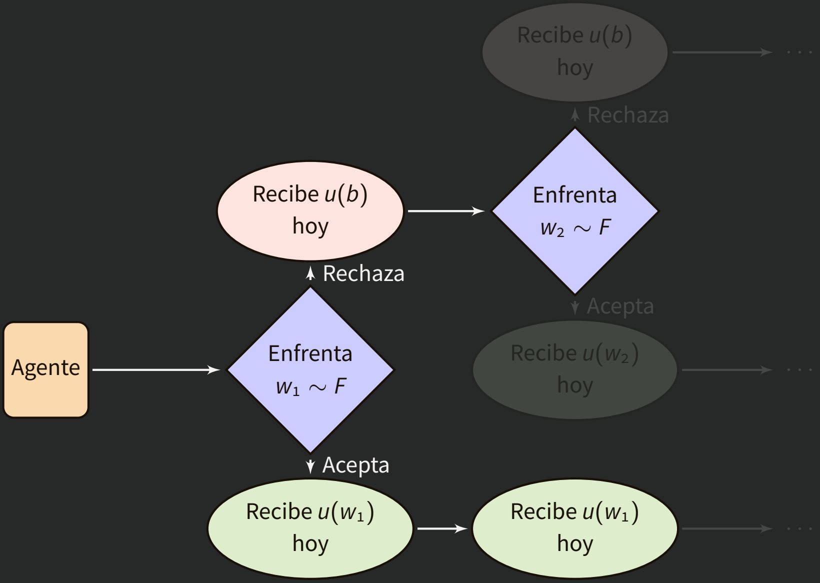
McCall (1970) con un árbol difícil



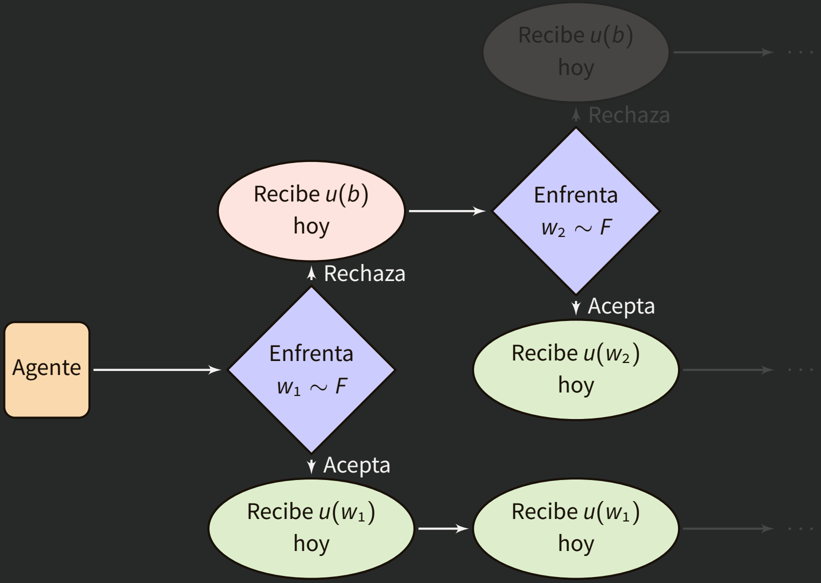
McCall (1970) con un árbol difícil



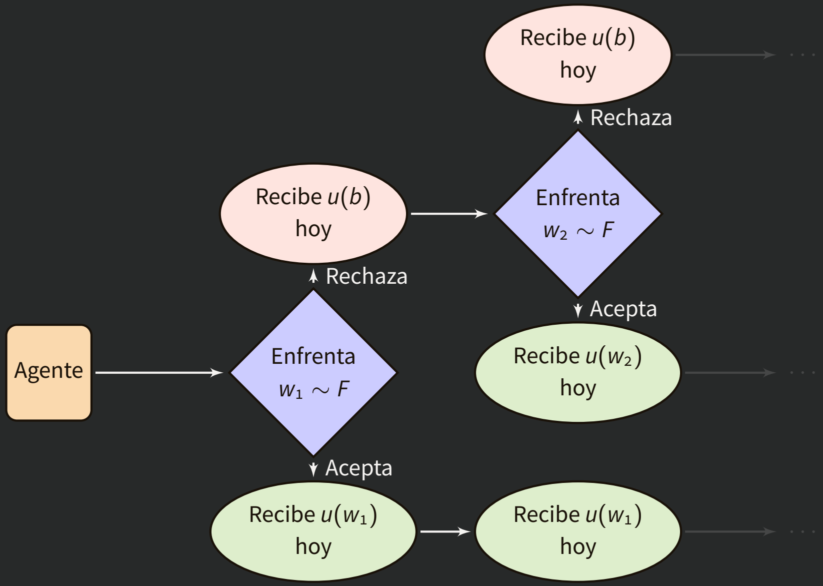
McCall (1970) con un árbol difícil



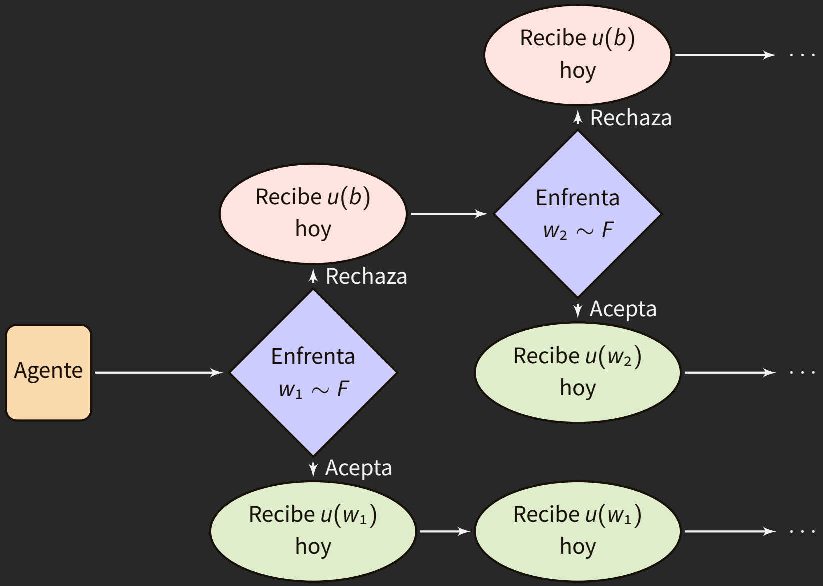
McCall (1970) con un árbol difícil



McCall (1970) con un árbol difícil



McCall (1970) con un árbol difícil



McCall (1970) escrito difícil

Problema del agente:

$$V = \max_{T} \mathbb{E}_0 \left[\sum_{t=0}^{T-1} \beta^t u(b) + \sum_{t=T}^{\infty} \beta^t u(w_T) \right]$$

sujeto a $\quad w_t \overset{iid}{\sim} F(\cdot)$

T debe ser adaptado a $\mathcal{F}(\{w_t\})$

- T es una función de los salarios w_t sacados antes de T.
- Cómo elijo T? En qué conjunto vive T? Cuál es la CPO?

McCall (1970) escrito difícil

Problema del agente:

$$V = \max_{T} \mathbb{E}_0 \left[\sum_{t=0}^{T-1} \beta^t u(b) + \sum_{t=T}^{\infty} \beta^t u(w_T) \right]$$

$$\begin{aligned} \text{sujeto a} \quad & w_t \overset{iid}{\sim} F(\cdot) \\ & T \text{ debe ser adaptado a } \mathcal{F}(\{w_t\}) \end{aligned}$$

- T es una función de los salarios w_t sacados antes de T.
- Cómo elijo T? En qué conjunto vive T? Cuál es la CPO?

McCall (1970) escrito difícil

Problema del agente:

$$V = \max_{T} \mathbb{E}_0 \left[\sum_{t=0}^{T-1} \beta^t u(b) + \sum_{t=T}^{\infty} \beta^t u(w_T) \right]$$

$$\text{sujeto a} \quad w_t \overset{iid}{\sim} F(\cdot)$$

$$T \text{ debe ser adaptado a } \mathcal{F}(\{w_t\})$$

- T es una función de los salarios w_t sacados antes de T.
- Cómo elijo T? En qué conjunto vive T? Cuál es la CPO?


COME TO THE
DARK
SIDE
WE HAVE COOKIES

El Lado Oscuro

- En t, si todavía no acepté una oferta, después de ver w_t

$$V_t = \begin{cases} \sum_{j=0}^{\infty} \beta^j u(w_t) & \text{si acepto} \\ u(b) + \beta \max_T \mathbb{E}\left[\sum_{j=0}^{T-1} \beta^j u(b) + \sum_{j=T}^{\infty} \beta^j u(w_T)\right] & \text{si rechazo} \end{cases}$$

- Así que

$$V_t = \max\left\{u(b) + \beta \mathbb{E}\left[V_{t+1}\right], R(w_t)\right\}$$

- MAGIA: V_t no depende de t dado w_t

El Lado Oscuro

- En t, si todavía no acepté una oferta, después de ver w_t

$$V_t = \begin{cases} \sum_{j=0}^{\infty} \beta^j u(w_t) = R(w_t) & \text{si acepto} \\ u(b) + \beta \max_T \mathbb{E}\left[\sum_{j=0}^{T-1} \beta^j u(b) + \sum_{j=T}^{\infty} \beta^j u(w_T)\right] & \text{si rechazo} \end{cases}$$

- Así que

$$V_t = \max\left\{u(b) + \beta \mathbb{E}\left[V_{t+1}\right], R(w_t)\right\}$$

- MAGIA: V_t no depende de t dado w_t

El Lado Oscuro

- En t, si todavía no acepté una oferta, después de ver w_t

$$V_t = \begin{cases} \sum_{j=0}^{\infty} \beta^j u(w_t) = R(w_t) & \text{si acepto} \\ u(b) + \beta \max_T \mathbb{E}\left[\sum_{j=0}^{T-1} \beta^j u(b) + \sum_{j=T}^{\infty} \beta^j u(w_T)\right] & \text{si rechazo} \end{cases}$$

- Así que

$$V_t = \max\left\{u(b) + \beta \mathbb{E}\left[V_{t+1}\right], R(w_t)\right\}$$

- MAGIA: V_t no depende de t dado w_t

El Lado Oscuro

- En t, si todavía no acepté una oferta, después de ver w_t

$$V_t = \begin{cases} \sum_{j=0}^{\infty} \beta^j u(w_t) = R(w_t) & \text{si acepto} \\ u(b) + \beta \max_T \mathbb{E}\left[\sum_{j=0}^{T-1} \beta^j u(b) + \sum_{j=T}^{\infty} \beta^j u(w_T)\right] & \text{si rechazo} \end{cases}$$

- Así que

$$V_t = \max \left\{ u(b) + \beta \mathbb{E}\left[V_{t+1}\right], R(w_t) \right\}$$

- MAGIA: V_t no depende de t dado w_t

El Lado Oscuro

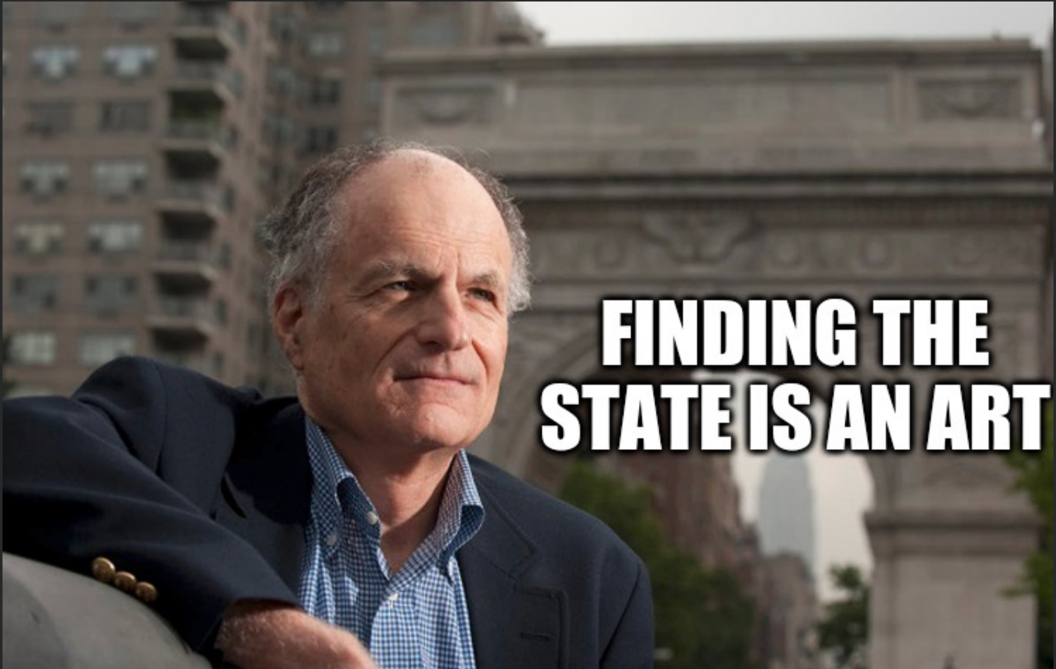
- En t, si todavía no acepté una oferta, después de ver w_t

$$V_t = \begin{cases} \sum_{j=0}^{\infty} \beta^j u(w_t) = R(w_t) & \text{si acepto} \\ u(b) + \beta \max_T \mathbb{E} \left[\sum_{j=0}^{T-1} \beta^j u(b) + \sum_{j=T}^{\infty} \beta^j u(w_T) \right] & \text{si rechazo} \end{cases}$$

- Así que

$$V_t = \max \left\{ u(b) + \beta \mathbb{E}\left[V_{t+1}\right], R(w_t) \right\}$$

- MAGIA: V_t no depende de t dado w_t


FINDING THE
STATE IS AN ART

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w)$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w)$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w) = \beta^0 u(w) + \sum_{j=1}^{\infty} \beta^j u(w)$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1}) \right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w) = \beta^0 u(w) + \sum_{j=1}^{\infty} \beta^j u(w)$$

$$= u(w) + \beta \sum_{j=0}^{\infty} \beta^j u(w)$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w) = \beta^0 u(w) + \sum_{j=1}^{\infty} \beta^j u(w)$$

$$= u(w) + \beta \sum_{j=0}^{\infty} \beta^j u(w)$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w) = \beta^0 u(w) + \sum_{j=1}^{\infty} \beta^j u(w)$$

$$= u(w) + \beta \sum_{j=0}^{\infty} \beta^j u(w) = u(w) + \beta R(w)$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w) = \beta^0 u(w) + \sum_{j=1}^{\infty} \beta^j u(w)$$

$$= u(w) + \beta \sum_{j=0}^{\infty} \beta^j u(w) = u(w) + \beta R(w)$$

$$= \frac{u(w)}{1-\beta}$$

McCall (1970) escrito recursivo

$$V(w_t) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w_{t+1})\right], R(w_t) \right\}$$

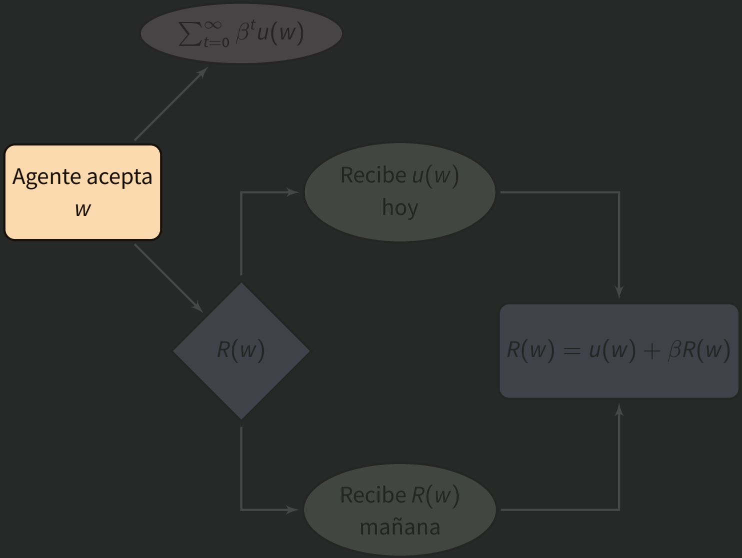
- Programación dinámica a mano

$$R(w) = \sum_{j=0}^{\infty} \beta^j u(w) = \beta^0 u(w) + \sum_{j=1}^{\infty} \beta^j u(w)$$

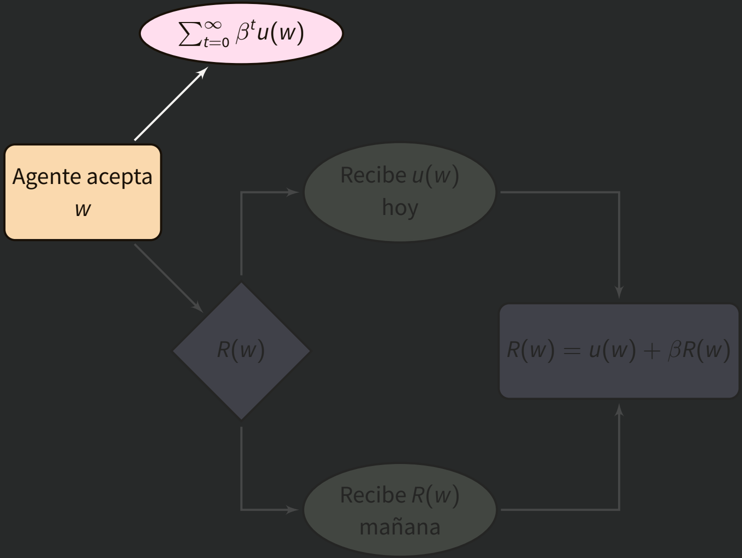
$$= u(w) + \beta \sum_{j=0}^{\infty} \beta^j u(w) = u(w) + \beta R(w)$$

$$= \frac{u(w)}{1-\beta}$$

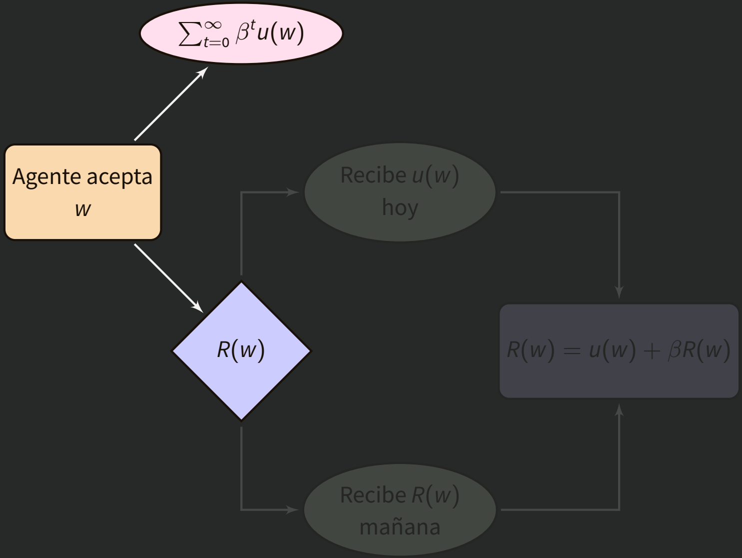
Aceptar una oferta: árbol recursivo



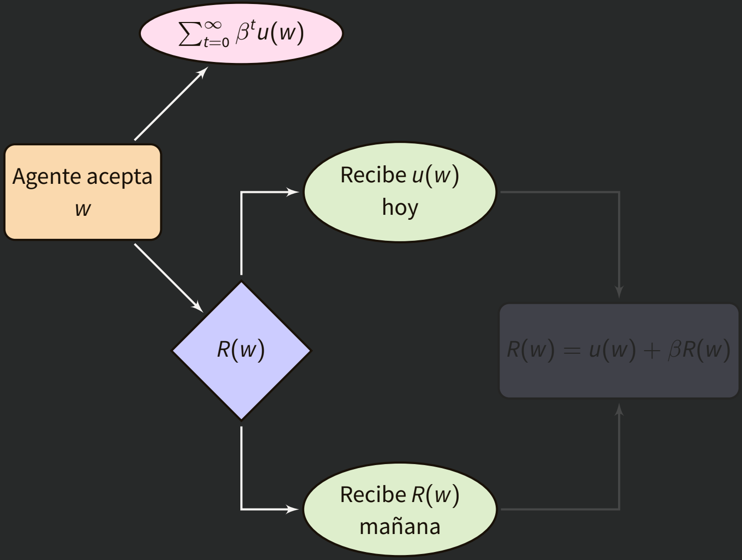
Aceptar una oferta: árbol recursivo



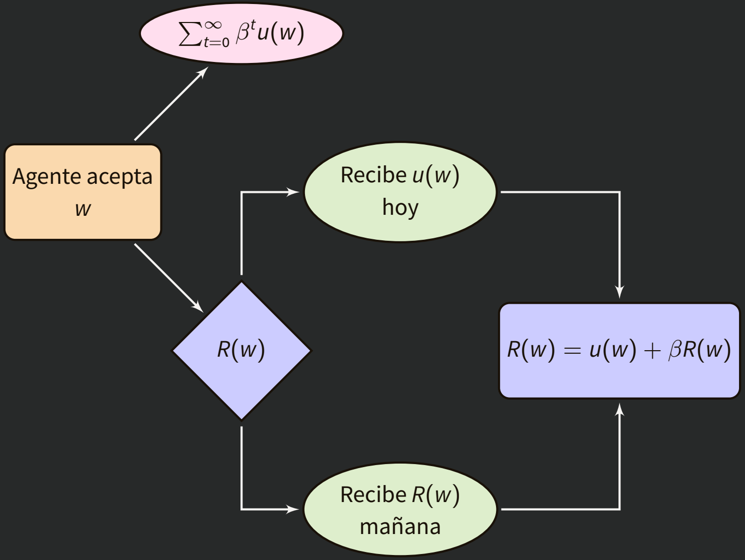
Aceptar una oferta: árbol recursivo



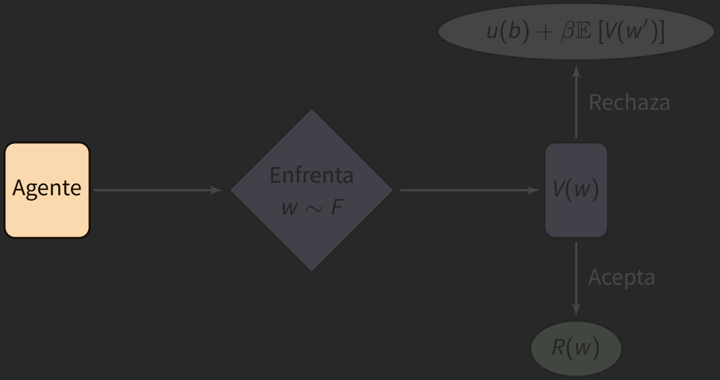
Aceptar una oferta: árbol recursivo



Aceptar una oferta: árbol recursivo



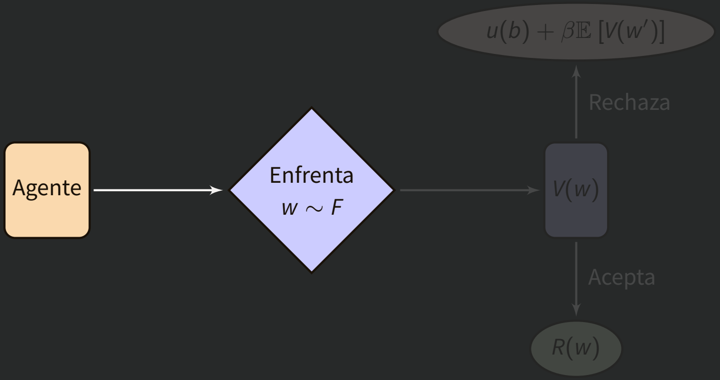
McCall (1970) con un árbol recursivo



Por lo tanto

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}\left[V(w')\right], R(w)\right\}$$

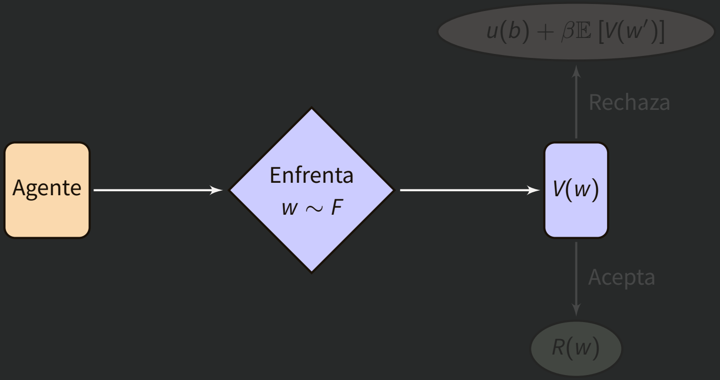
McCall (1970) con un árbol recursivo



Por lo tanto

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}\left[V(w')\right], R(w)\right\}$$

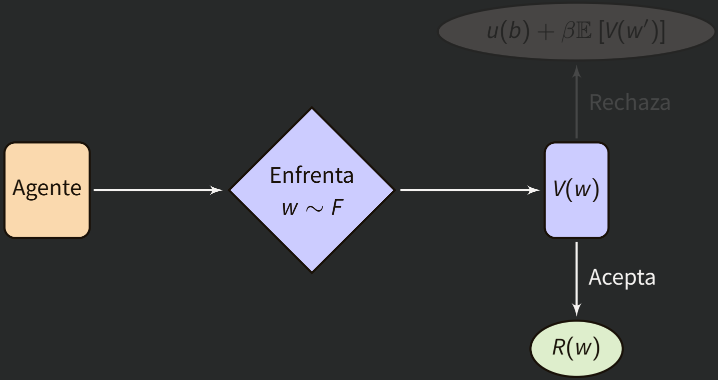
McCall (1970) con un árbol recursivo



Por lo tanto

$$V(w) = \max \left\{ u(b) + \beta \mathbb{E}[V(w')], R(w) \right\}$$

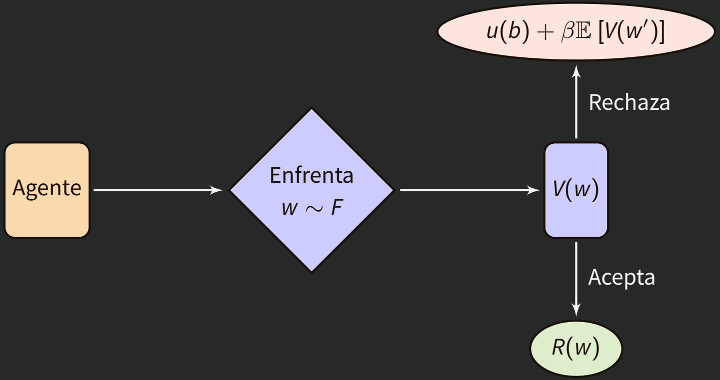
McCall (1970) con un árbol recursivo



Por lo tanto

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}\left[V(w')\right], R(w)\right\}$$

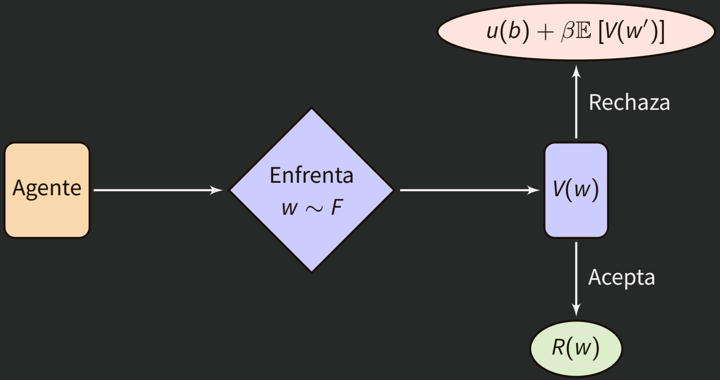
McCall (1970) con un árbol recursivo



Por lo tanto

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}\left[V(w')\right], R(w)\right\}$$

McCall (1970) con un árbol recursivo

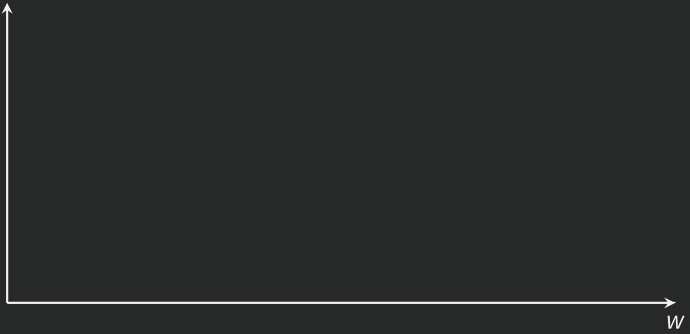


Por lo tanto

$$V(w) = \max \left\{ u(b) + \beta \mathbb{E}\left[V(w')\right], R(w) \right\}$$

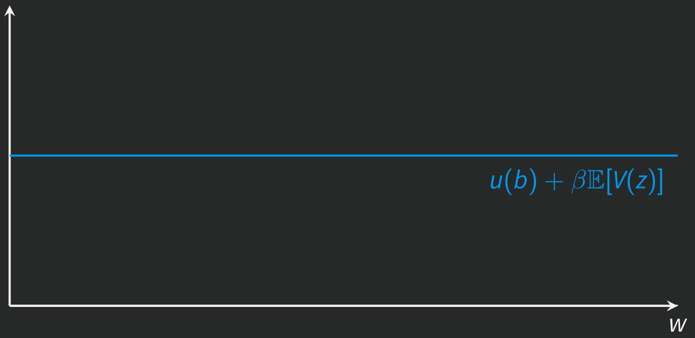
McCall (1970) escrito recursivo

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$



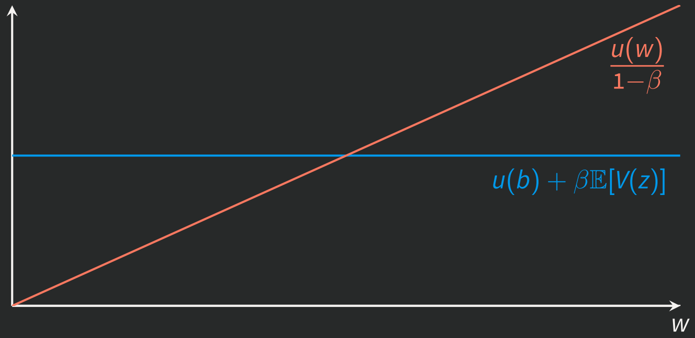
McCall (1970) escrito recursivo

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$



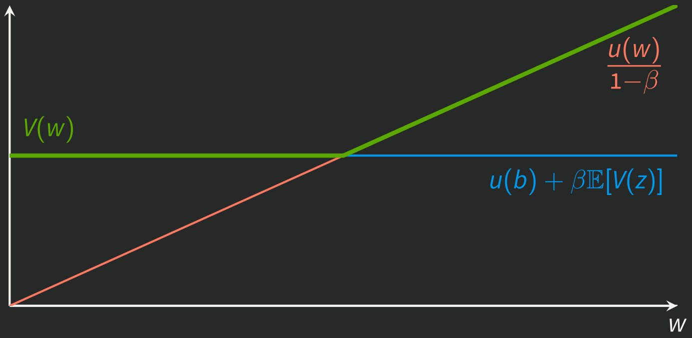
McCall (1970) escrito recursivo

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$



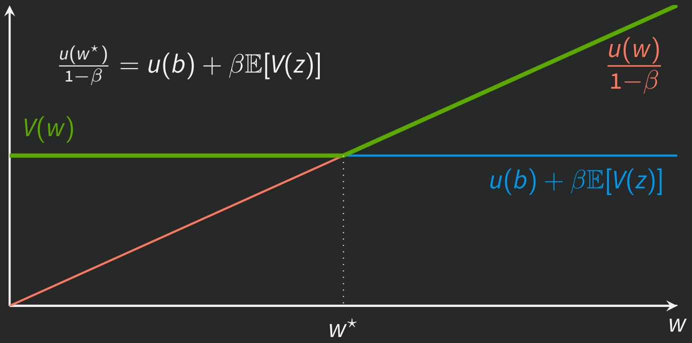
McCall (1970) escrito recursivo

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$



McCall (1970) escrito recursivo

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$



McCall (1970) con programación dinámica

$$V(w) = \max \left\{ u(b) + \beta \mathbb{E}[V(z)], \frac{u(w)}{1-\beta} \right\}$$

Algoritmo

1. Inicializar: $V^0(w) = 0$
2. Usar V^0 del lado derecho, obtener $V^1(w) = \max \left\{ u(b) + 0, \frac{u(w)}{1-\beta} \right\}$
3. Usar V^1 del lado derecho, obtener V^2

... Iterar hasta que $|V^n - V^{(n-1)}| \leq \epsilon$ (distancia entre funciones)

McCall (1970) con programación dinámica

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$

Algoritmo

1. Inicializar: $V^0(w) = 0$
2. Usar V^0 del lado derecho, obtener $V^1(w) = \max\left\{u(b) + 0, \frac{u(w)}{1-\beta}\right\}$
3. Usar V^1 del lado derecho, obtener V^2

... Iterar hasta que $|V^n - V^{(n-1)}| \leq \epsilon$ (distancia entre funciones)

McCall (1970) con programación dinámica

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$

Algoritmo

1. Inicializar: $V^0(w) = 0$
2. Usar V^0 del lado derecho, obtener $V^1(w) = \max\left\{u(b) + 0, \frac{u(w)}{1-\beta}\right\}$
3. Usar V^1 del lado derecho, obtener V^2

... Iterar hasta que $|V^n - V^{(n-1)}| \leq \epsilon$ (distancia entre funciones)

McCall (1970) con programación dinámica

$$V(w) = \max \left\{ u(b) + \beta \mathbb{E}[V(z)], \frac{u(w)}{1-\beta} \right\}$$

Algoritmo

1. Inicializar: $V^0(w) = 0$
2. Usar V^0 del lado derecho, obtener $V^1(w) = \max \left\{ u(b) + 0, \frac{u(w)}{1-\beta} \right\}$
3. Usar V^1 del lado derecho, obtener V^2

... Iterar hasta que $|V^n - V^{(n-1)}| \leq \epsilon$ (distancia entre funciones)

McCall (1970) con programación dinámica

$$V(w) = \max\left\{u(b) + \beta\mathbb{E}[V(z)], \frac{u(w)}{1-\beta}\right\}$$

Algoritmo

1. Inicializar: $V^0(w) = 0$
2. Usar V^0 del lado derecho, obtener $V^1(w) = \max\left\{u(b) + 0, \frac{u(w)}{1-\beta}\right\}$
3. Usar V^1 del lado derecho, obtener V^2

... Iterar hasta que $|V^n - V^{(n-1)}| \leq \epsilon$ (distancia entre funciones)

Programación Dinámica: Consumo/ahorro

Problema de la torta

- Un agente tiene una torta de tamaño K
- Preferencias standard: utilidad u, descuento β

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$\text{sujeto a} \quad c_t + k_{t+1} = k_t$$

$$k_{t+1} \geq 0$$

- Y hagamos que $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$

Problema de la torta

- Un agente tiene una torta de tamaño K
- Preferencias standard: utilidad u, descuento β

$$\begin{aligned}
\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t u(c_t) \\
\text{sujeto a} \quad & c_t + k_{t+1} = k_t \\
& k_{t+1} \geq 0
\end{aligned}$$

- Y hagamos que $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$

Problema de la torta

CPOs

- Derivando contra c_t y k_{t+1}

$$\begin{aligned} \beta^t u'(c_t) &= \lambda_t \\ \lambda_t &= \lambda_{t+1} \end{aligned}$$

- Así que

$$\begin{aligned} u'(c_t) &= \beta u'(c_{t+1}) \\ \implies c_{t+1} &= \beta^{\frac{1}{\gamma}} c_t \\ \implies c_t &= c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t \end{aligned}$$

Problema de la torta

CPOs

- Derivando contra c_t y k_{t+1}

$$\beta^t u'(c_t) = \lambda_t$$
$$\lambda_t = \lambda_{t+1}$$

- Así que

$$u'(c_t) = \beta u'(c_{t+1})$$
$$\implies c_{t+1} = \beta^{\frac{1}{\gamma}} c_t$$
$$\implies c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta

CPOs

- Derivando contra c_t y k_{t+1}

$$\beta^t u'(c_t) = \lambda_t$$
$$\lambda_t = \lambda_{t+1}$$

- Así que

$$c_t^{-\gamma} = \beta c_{t+1}^{-\gamma}$$
$$\implies c_{t+1} = \beta^{\frac{1}{\gamma}} c_t$$
$$\implies c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta

CPOs

- Derivando contra c_t y k_{t+1}

$$\beta^t u'(c_t) = \lambda_t$$
$$\lambda_t = \lambda_{t+1}$$

- Así que

$$c_t^{-\gamma} = \beta c_{t+1}^{-\gamma}$$
$$\implies c_{t+1} = \beta^{\frac{1}{\gamma}} c_t$$
$$\implies c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta

CPOs

- Derivando contra c_t y k_{t+1}

$$\beta^t u'(c_t) = \lambda_t$$
$$\lambda_t = \lambda_{t+1}$$

- Así que

$$\begin{aligned} c_t^{-\gamma} &= \beta c_{t+1}^{-\gamma} \\ \implies \quad c_{t+1} &= \beta^{\frac{1}{\gamma}} c_t \\ \implies \quad c_t &= c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t \end{aligned}$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1}$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t\to\infty} k_t$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}} \right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t \to \infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t \to \infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t = \sum_{t=0}^{\infty} c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t \to \infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t = \sum_{t=0}^{\infty} c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \sum_{t=0}^{\infty} \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t\to\infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t = \sum_{t=0}^{\infty} c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \sum_{t=0}^{\infty} \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \frac{1}{1-\beta^{\frac{1}{\gamma}}}$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t\to\infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t = \sum_{t=0}^{\infty} c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \sum_{t=0}^{\infty} \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \frac{1}{1-\beta^{\frac{1}{\gamma}}}$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t\to\infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t = \sum_{t=0}^{\infty} c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \sum_{t=0}^{\infty} \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \frac{1}{1-\beta^{\frac{1}{\gamma}}}$$

Problema de la torta

Tenemos

$$c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t$$

y

$$c_t + k_{t+1} = k_t \implies k_0 = c_0 + k_1 \implies k_0 = \sum_{s=0}^{t} c_s + k_{t+1} \implies k_0 = \sum_{t=0}^{\infty} c_t + \lim_{t\to\infty} k_t$$

así que

$$k_0 = \sum_{t=0}^{\infty} c_t = \sum_{t=0}^{\infty} c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \sum_{t=0}^{\infty} \left(\beta^{\frac{1}{\gamma}}\right)^t = c_0 \frac{1}{1-\beta^{\frac{1}{\gamma}}}$$

Al final,

$$\begin{cases} c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t \\ k_0 = \frac{c_0}{1-\beta^{\frac{1}{\gamma}}} \end{cases} \implies c_t = k_0 \left(1 - \beta^{\frac{1}{\gamma}}\right) \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta: a little knowledge of geometric sums. . .

Guess and Verify

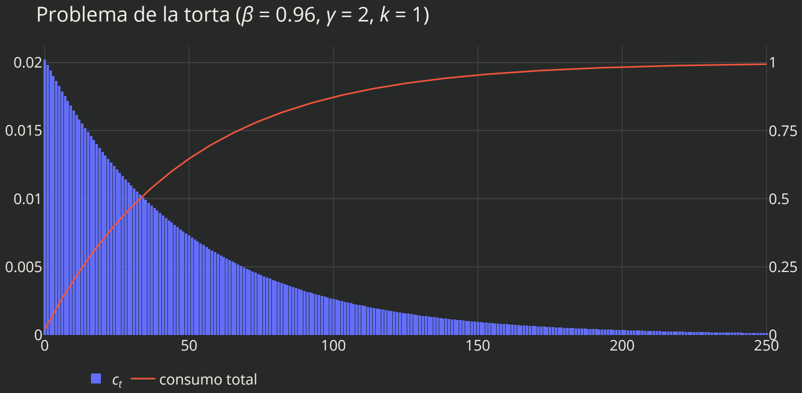
Al final,

$$\begin{cases} c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t \\ k_0 = \frac{c_0}{1-\beta^{\frac{1}{\gamma}}} \end{cases} \implies c_t = k_0 \left(1 - \beta^{\frac{1}{\gamma}}\right) \left(\beta^{\frac{1}{\gamma}}\right)^t$$

Problema de la torta: a little knowledge of geometric sums. . .

Al final,

$$\begin{cases} c_t = c_0 \left(\beta^{\frac{1}{\gamma}}\right)^t \\ k_0 = \frac{c_0}{1-\beta^{\frac{1}{\gamma}}} \end{cases} \implies c_t = k_0 \left(1 - \beta^{\frac{1}{\gamma}}\right) \left(\beta^{\frac{1}{\gamma}}\right)^t$$



Extensiones que necesitamos

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1 + r) \qquad \text{si } r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1 + r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada...

Extensiones que necesitamos

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1+r)$ si $r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1+r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada. . .

Extensiones que necesitamos

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1 + r)$ si $r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1 + r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada...

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1 + r)$ si $r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1 + r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada...

Extensiones que necesitamos

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1 + r)$ si $r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1 + r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada...

Extensiones que necesitamos

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1+r)$ si $r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1+r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada...

Extensiones que necesitamos

Problema de la torta con

- La torta se va pudriendo: $k_{t+1} = (k_t - c_t)(1 + r)$ si $r < 0$
- Llega nueva torta: $k_{t+1} = k_t - c_t + y_{t+1}$
- La nueva torta es aleatoria según una cadena de Markov $F(y'|y)$
- Puedo pedir torta prestada: $k_{t+1} \geq \bar{k}$

$$\max_{c_t, k_{t+1}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t u(c_t) \right]$$

$$\text{sujeto a } k_{t+1} + c_t = k_t(1 + r) + y_t$$

$$k_{t+1} \geq \bar{k}$$

- En general no tiene forma cerrada. . .

Problema de la torta recursivo

- Otra vez llamemos V_t al valor del problema en t

$$V_t = \max_{c_{t+s}, k_{t+s}} \mathbb{E}_t \left[\sum_{s=0}^{\infty} \beta^s u(c_{t+s}) \right] = \max_{c_t, k_{t+1}} u(c_t) + \beta \mathbb{E}_t \left[V_{t+1} \right]$$

$$\begin{aligned} \text{sujeto a } & k_{t+1} + c_t = k_t(1+r) + y_t \\ & k_{t+2} + c_{t+1} = k_{t+1}(1+r) + y_{t+1} \\ & \dots \\ & k_{t+s+1} + c_{t+s} = k_{t+s}(1+r) + y_{t+s} \\ & k_{t+s+1} \geq \bar{k} \end{aligned}$$

Problema de la torta recursivo

- Otra vez llamemos V_t al valor del problema en t

$$V_t = \max_{c_{t+s}, k_{t+s}} \mathbb{E}_t \left[\sum_{s=0}^{\infty} \beta^s u(c_{t+s}) \right] = \max_{c_t, k_{t+1}} u(c_t) + \beta \mathbb{E}_t \left[V_{t+1} \right]$$

$$\begin{aligned} \text{sujeto a } & k_{t+1} + c_t = k_t(1+r) + y_t \\ & k_{t+2} + c_{t+1} = k_{t+1}(1+r) + y_{t+1} \\ & \dots \\ & k_{t+s+1} + c_{t+s} = k_{t+s}(1+r) + y_{t+s} \\ & k_{t+s+1} \geq \bar{k} \end{aligned}$$

Problema de la torta recursivo

- Otra vez llamemos V_t al valor del problema en t

$$V_t = \max_{c_{t+s}, k_{t+s}} \mathbb{E}_t \left[\sum_{s=0}^{\infty} \beta^s u(c_{t+s}) \right] = \max_{c_t, k_{t+1}} u(c_t) + \beta \mathbb{E}_t \left[V_{t+1} \right]$$

$$\begin{aligned} \text{sujeto a } & k_{t+1} + c_t = k_t(1+r) + y_t \\ & k_{t+2} + c_{t+1} = k_{t+1}(1+r) + y_{t+1} \\ & \dots \\ & k_{t+s+1} + c_{t+s} = k_{t+s}(1+r) + y_{t+s} \\ & k_{t+s+1} \geq \bar{k} \end{aligned}$$

Problema de la torta recursivo

- Otra vez llamemos V_t al valor del problema en t

$$V_t = \max_{c_{t+s}, k_{t+s}} \mathbb{E}_t \left[\sum_{s=0}^{\infty} \beta^s u(c_{t+s}) \right] = \max_{c_t, k_{t+1}} u(c_t) + \beta \mathbb{E}_t \left[V_{t+1} \right]$$

$$\begin{aligned} \text{sujeto a } & k_{t+1} + c_t = k_t(1+r) + y_t \\ & k_{t+2} + c_{t+1} = k_{t+1}(1+r) + y_{t+1} \\ & \dots \\ & k_{t+s+1} + c_{t+s} = k_{t+s}(1+r) + y_{t+s} \\ & k_{t+s+1} \geq \bar{k} \end{aligned}$$

Problema de la torta recursivo

$$v(k, y) = \max_{c, k'} u(c) + \beta \mathbb{E}\left[v(k', y') | y\right]$$

$$\text{sujeto a } c + k' = y + k(1 + r)$$

$$k' \geq \bar{k}$$

Problema de la torta recursivo

$$v(k, y) = \max_{c, k'} u(c) + \beta \mathbb{E}\left[v(k', y') | y\right]$$

$$\text{sujeto a } c + k' = y + k(1 + r)$$

$$k' \geq \bar{k}$$

- La función v es desconocida
- Podemos
  1. meter una f cualquiera del lado derecho (reemplazar v por f)
  2. usar la ec. de Bellman para encontrar una nueva f
  3. comparar la f que entró con la f que salió
  4. usar la f que salió del lado derecho

Problema de la torta recursivo

$$v(k,y) = \max_{c,k'} u(c) + \beta\mathbb{E}\left[v(k',y')|y\right]$$

$$\text{sujeto a } c + k' = y + k(1+r)$$

$$k' \geq \bar{k}$$

- La función v es desconocida
- Podemos
  1. meter una f cualquiera del lado derecho (reemplazar v por f)
  2. usar la ec. de Bellman para encontrar una nueva f
  3. comparar la f que entró con la f que salió
  4. usar la f que salió del lado derecho

Problema de la torta recursivo

$$v(k, y) = \max_{c,k'} u(c) + \beta \mathbb{E}\left[v(k', y')|y\right]$$

$$\text{sujeto a } c + k' = y + k(1 + r)$$

$$k' \geq \bar{k}$$

- La función v es desconocida
- Podemos
  1. meter una f cualquiera del lado derecho (reemplazar v por f)
  2. usar la ec. de Bellman para encontrar una nueva f
  3. comparar la f que entró con la f que salió
  4. usar la f que salió del lado derecho

Problema de la torta recursivo

$$v(k, y) = \max_{c, k'} u(c) + \beta \mathbb{E}\left[v(k', y') | y\right]$$

$$\text{sujeto a } c + k' = y + k(1 + r)$$

$$k' \geq \bar{k}$$

- La función v es desconocida
- Podemos
  1. meter una f cualquiera del lado derecho (reemplazar v por f)
  2. usar la ec. de Bellman para encontrar una nueva f
  3. comparar la f que entró con la f que salió
  4. usar la f que salió del lado derecho

Problema de la torta recursivo

$$v(k, y) = \max_{c, k'} u(c) + \beta \mathbb{E}\left[v(k', y') | y\right]$$

$$\text{sujeto a } c + k' = y + k(1 + r)$$

$$k' \geq \bar{k}$$

- La función v es desconocida
- Podemos
    1. meter una f cualquiera del lado derecho (reemplazar v por f)
    2. usar la ec. de Bellman para encontrar una nueva f
    3. comparar la f que entró con la f que salió
    4. usar la f que salió del lado derecho

Cierre

Cierre

Vimos

- Programación dinámica
    - Finding the state is an art
    - Iterar sobre la ecuación de Bellman es 90% de un algoritmo
    - Por qué funciona?
- El modelo de búsqueda más sencillo posible
- El problema de la torta

. . . si te hizo acordar a otra cosa, vamos bien

El arma secreta: Guess and Verify

Volver

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\qquad \xi = \frac{1}{1-\beta^{1/\gamma}}$

  … Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$

  … Y entonces $c_1 = \xi k_1 \dots$
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = u(\xi k) + \beta v(k')$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma} v(1) = k^{1-\gamma} u(\xi) + \beta (1-\xi)^{1-\gamma} k^{1-\gamma} v(1)$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

El arma secreta: Guess and Verify

Volver

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\qquad \xi = \frac{1}{1-\beta^{1/\gamma}}$

  … Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$

  … Y entonces $c_1 = \xi k_1$…
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = u(\xi k) + \beta v(k')$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma} v(1) = k^{1-\gamma} u(\xi) + \beta (1-\xi)^{1-\gamma} k^{1-\gamma} v(1)$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

El arma secreta: Guess and Verify

Volver

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\qquad \xi = \frac{1}{1-\beta^{1/\gamma}}$

  … Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$

  … Y entonces $c_1 = \xi k_1 \ldots$
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = u(\xi k) + \beta v((1-\xi)k)$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma}v(1) = k^{1-\gamma}u(\xi) + \beta(1-\xi)^{1-\gamma}k^{1-\gamma}v(1)$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

El arma secreta: Guess and Verify

Volver

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\qquad \xi = \frac{1}{1-\beta^{1/\gamma}}$

  ... Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$

  ... Y entonces $c_1 = \xi k_1 \ldots$
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = k^{1-\gamma} u(\xi) + \beta v((1-\xi)k)$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma} v(1) = k^{1-\gamma} u(\xi) + \beta (1-\xi)^{1-\gamma} k^{1-\gamma} v(1)$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

El arma secreta: Guess and Verify

Volver

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\quad \xi = \frac{1}{1-\beta^{1/\gamma}}$

  … Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$

  … Y entonces $c_1 = \xi k_1 \ldots$
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = k^{1-\gamma}u(\xi) + \beta v((1-\xi)k)$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma}v(1) = k^{1-\gamma}u(\xi) + \beta(1-\xi)^{1-\gamma}k^{1-\gamma}v(1)$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

El arma secreta: Guess and Verify

Volver

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\quad \xi = \frac{1}{1-\beta^{1/\gamma}}$
  … Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$
  … Y entonces $c_1 = \xi k_1 \ldots$
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = k^{1-\gamma}u(\xi) + \beta v((1-\xi)k)$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma}\tilde{v} = k^{1-\gamma}u(\xi) + \beta(1-\xi)^{1-\gamma}k^{1-\gamma}\tilde{v}$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

El arma secreta: Guess and Verify

- En el problema de la torta resultó que $c_0 = \xi k_0$ $\quad \xi = \frac{1}{1-\beta^{1/\gamma}}$

  … Con lo cual como $c_t + k_{t+1} = k_t$, pasa que $k_1 = (1-\xi)k_0$

  … Y entonces $c_1 = \xi k_1 \ldots$
- Podemos adivinar que $c = \xi k$. Si fuera cierto, entonces

$$v(k) = k^{1-\gamma} u(\xi) + \beta v((1-\xi)k)$$

- Como el término $k^{1-\gamma}$ sale de la u, podemos pensar que también va a salir de las v

$$k^{1-\gamma}\tilde{v} = k^{1-\gamma} u(\xi) + \beta(1-\xi)^{1-\gamma} k^{1-\gamma}\tilde{v}$$

- Guess-and-verify: en el problema original intentemos ver si $v(k) = k^{1-\gamma}\tilde{v}$ es una solución

$$\underbrace{k^{1-\gamma}\tilde{v}}_{=v(k)} = \max_{k'} u(k-k') + \beta \underbrace{(k')^{1-\gamma}\tilde{v}}_{=v(k')}$$

Guess and Verify

Volver

- Tenemos

$$k^{1-\gamma}\tilde{v} = \max_{k'} u(k-k') + \beta(k')^{1-\gamma}\tilde{v}$$

- Entonces eligiendo $\xi = k'/k$, lo que realmente hay que ver es que ξ^* no dependa de k

$$k^{1-\gamma}\tilde{v} = \max_{\xi\in[0,1]} u(k-\xi k) + \beta\tilde{v}\,(\xi k)^{1-\gamma}$$

- Si la u es CRRA, $u(x) \propto x^{1-\gamma}$, así que

$$\tilde{v}k^{1-\gamma} = \max_{\xi\in[0,1]} u(1-\xi)k^{1-\gamma} + \beta\tilde{v}\xi^{1-\gamma}k^{1-\gamma}$$

- Así que ξ no puede depender de k porque k ni aparece

$$\tilde{v} = \max_{\xi\in[0,1]} u(1-\xi) + \beta\tilde{v}\xi^{1-\gamma}$$

Guess and Verify

- Tenemos

$$k^{1-\gamma}\tilde{v} = \max_{k'} u(k-k') + \beta(k')^{1-\gamma}\tilde{v}$$

- Entonces eligiendo $\xi = k'/k$, lo que realmente hay que ver es que $\xi^\star$ no dependa de k

$$k^{1-\gamma}\tilde{v} = \max_{\xi\in[0,1]} u(k-\xi k) + \beta\tilde{v}\,(\xi k)^{1-\gamma}$$

- Si la u es CRRA, $u(x) \propto x^{1-\gamma}$, así que

$$\tilde{v}k^{1-\gamma} = \max_{\xi\in[0,1]} u(1-\xi)k^{1-\gamma} + \beta\tilde{v}\xi^{1-\gamma}k^{1-\gamma}$$

- Así que ξ no puede depender de k porque k ni aparece

$$\tilde{v} = \max_{\xi\in[0,1]} u(1-\xi) + \beta\tilde{v}\xi^{1-\gamma}$$

Guess and Verify

Volver

- Tenemos

$$k^{1-\gamma}\tilde{v} = \max_{k'} u(k-k') + \beta(k')^{1-\gamma}\tilde{v}$$

- Entonces eligiendo $\xi = k'/k$, lo que realmente hay que ver es que $\xi^\star$ no dependa de k

$$k^{1-\gamma}\tilde{v} = \max_{\xi\in[0,1]} u(k-\xi k) + \beta\tilde{v}\,(\xi k)^{1-\gamma}$$

- Si la u es CRRA, $u(x) \propto x^{1-\gamma}$, así que

$$\tilde{v}k^{1-\gamma} = \max_{\xi\in[0,1]} u(1-\xi)k^{1-\gamma} + \beta\tilde{v}\xi^{1-\gamma}k^{1-\gamma}$$

- Así que ξ no puede depender de k porque k ni aparece

$$\tilde{v} = \max_{\xi\in[0,1]} u(1-\xi) + \beta\tilde{v}\xi^{1-\gamma}$$

Guess and Verify

Volver

- Tenemos

$$k^{1-\gamma}\tilde{v} = \max_{k'} u(k-k') + \beta(k')^{1-\gamma}\tilde{v}$$

- Entonces eligiendo $\xi = k'/k$, lo que realmente hay que ver es que $\xi^\star$ no dependa de k

$$k^{1-\gamma}\tilde{v} = \max_{\xi\in[0,1]} u(k-\xi k) + \beta\tilde{v}\,(\xi k)^{1-\gamma}$$

- Si la u es CRRA, $u(x) \propto x^{1-\gamma}$, así que

$$\tilde{v}k^{1-\gamma} = \max_{\xi\in[0,1]} u(1-\xi)k^{1-\gamma} + \beta\tilde{v}\xi^{1-\gamma}k^{1-\gamma}$$

- Así que ξ no puede depender de k porque k ni aparece

$$\tilde{v} = \max_{\xi\in[0,1]} u(1-\xi) + \beta\tilde{v}\xi^{1-\gamma}$$