

Convergencia del método de Nelder-Mead simplex en dimensiones bajas y aplicaciones en economía

Marcelo Gallardo

PUCP

Diciembre 2024

- 1 Introducción
- 2 El algoritmo Nelder-Mead
- 3 Enfoque matricial
- 4 Funciones estrictamente convexas
- 5 Aplicaciones del algoritmo NMS en bajas dimensiones
- 6 Programación dinámica
- 7 Resumen del modelo RBC clásico

Introducción

Introducción

- ❶ En 1965, se publicó el artículo que introdujo el algoritmo Nelder-Mead simplex (desde ahora NMS), [[Nelder and Mead, 1965](#)].
- ❷ Proporciona un método de búsqueda de mínimos en problemas de optimización sin restricciones en múltiples dimensiones.
- ❸ A pesar de su popularidad, hasta la fecha (1998) no se habían presentado resultados teóricos que respalden su desempeño.
- ❹ El trabajo de [[Jeffrey Lagarias and Wright, 1998](#)] aborda esta brecha, probando resultados de convergencia en casos unidimensionales y bidimensionales.
- ❺ Desde su publicación, el algoritmo NMS se ha consolidado como una herramienta común para resolver problemas de optimización irrestrictos.

- 1 No confundir con el algoritmo simplex de Dantzig, el cual también utiliza simplexes, pero en el contexto de programación lineal:

$$\max c^T x \quad \text{s.a.} \quad Ax \leq b, \quad x \in \mathbb{R}_+^n.$$

- 2 La relevancia del algoritmo NMS se refuerza con obras como [[Press et al., 1988](#)], donde se enfatizan tanto el algoritmo como algunas de sus variantes y adaptaciones.

Descripción general del algoritmo

- ❶ El algoritmo NMS busca minimizar una función real valuada con n variables y sin restricciones:

$$\min f(x), \text{ s.a : } x \in \mathbb{R}^n.$$

A diferencia de métodos clásicos de optimización, no utiliza derivadas (ni gradiente, ni Jacobiano, ni Hessiana) en ningún momento.

- ❷ Por esta razón, el algoritmo NMS pertenece a la familia de métodos de **búsqueda directa** [Wright, 1996, Torczon, 1997].
- ❸ En cada paso, el algoritmo mantiene un simplex no degenerado (envolvente convexa de $n + 1$ puntos).
- ❹ En cada iteración, partiendo de un simplex inicial, se realizan evaluaciones de la función objetivo, generando un nuevo simplex para la siguiente etapa.
- ❺ Este proceso lo convierte en un método parsimonioso¹, ya que realiza pocas evaluaciones de la función objetivo $f(\cdot)$.

¹ Lo opuesto a sobre-ajustado. Por ejemplo, en las regresiones lineales, incluir «varias» regresoras.

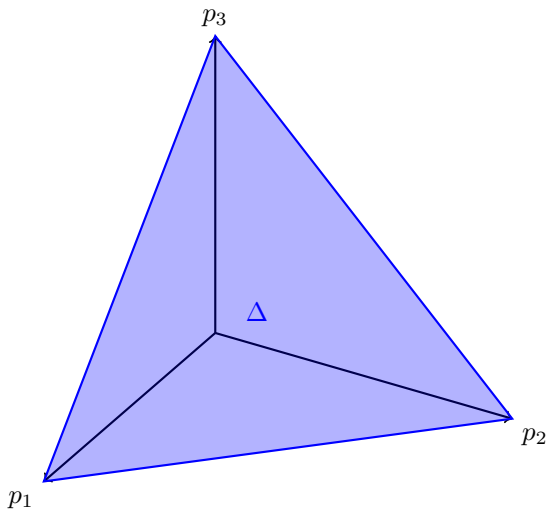


Figura Simplex en \mathbb{R}^3 .

Limitaciones

- ❶ Desempeño en altas dimensiones.
- ❷ [McKinnon, 1998] presentó un ejemplo bidimensional con funciones estrictamente convexas donde el algoritmo converge a un punto no estacionario.

$$f(x, y) = x^2 - y(y - 2), \quad \text{vértices iniciales: } \{(1, 0), (0, -3), (0, 3)\}.$$

- ❸ Ver código Matlab, «code simplex 1». $P^* = (0, -3)$.
- ❹ La falta de convergencia en este caso específico resalta que el algoritmo NMS no garantiza ciertas propiedades como:
 - ❶ que las aristas del simplex permanezcan uniformemente linealmente independientes,
 - ❷ una condición de «descenso fuerte» en el valor de la función objetivo en cada iteración.
- ❺ La familia que construye McKinnon es

$$\begin{aligned} f(x, y) &= \theta \phi |x|^\tau + y + y^2, \quad x \leq 0 \\ &= \theta x^\tau + y + y^2, \quad x \geq 0, \end{aligned}$$

donde $\theta, \phi > 0$. **Convergencia a un punto no estacionario.**

A pesar de las complicaciones y limitaciones mencionadas, el algoritmo NMS posee ventajas que lo hacen atractivo en ciertas aplicaciones:

- 1 En muchas aplicaciones prácticas, no es necesario encontrar el mínimo global, sino simplemente mejorar el rendimiento de un proceso. El algoritmo NMS logra mejoras significativas en pocas iteraciones.
- 2 No requiere el cálculo de derivadas y, además, realiza pocas evaluaciones de la función objetivo en cada iteración.

Resultados principales

El artículo [[Jeffrey Lagarias and Wright, 1998](#)] presenta resultados fundamentales para entender las propiedades teóricas del algoritmo NMS:

- ❶ En dimensión 1, el algoritmo NMS converge al mínimo global (teorema 4.1).
- ❷ En dimensión 2, los valores de la función objetivo en los vértices del simplex convergen al mismo valor (teorema 5.2).
- ❸ En dimensión 2, los diámetros de los simplexes convergen a cero.

El algoritmo Nelder-Mead

El algoritmo Nelder-Mead

El algoritmo NMS se propuso para resolver problemas de la forma:

$$\min_{x \in \mathbb{R}^n} f(x).$$

Para configurar el algoritmo, se definen cuatro parámetros escalares:

- ρ - reflexión (reflection),
- χ - expansión (expansion),
- γ - contracción (contraction),
- σ - regulación (shrinkage).

De acuerdo con [Nelder and Mead, 1965], se cumplen las siguientes relaciones:

$$\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1, \quad \text{y} \quad 0 < \sigma < 1.$$

Usualmente, los valores recomendados son:

$$\rho = 1, \quad \chi = 2, \quad \gamma = 1/2, \quad \text{y} \quad \sigma = 1/2.$$

Simulation [here](#).

Estructura de la iteraciones

En la iteración k , se tiene un simplex no degenerado² Δ_k con $n + 1$ vértices, etiquetados como $x_1^{(k)}, \dots, x_{n+1}^{(k)}$, ordenados de forma que:

$$f(x_1^{(k)}) \leq f(x_2^{(k)}) \leq \dots \leq f(x_{n+1}^{(k)}).$$

El vértice $x_1^{(k)}$ es el mejor (valor más bajo de f), mientras que $x_{n+1}^{(k)}$ es el peor. El objetivo de cada iteración es generar un nuevo simplex Δ_{k+1} distinto de Δ_k . Dependiendo de las evaluaciones de la función objetivo, se descarta el peor vértice y se introduce un nuevo vértice.

Observación

Definiremos con detalle más adelante qué pasa con los empates. Esto permite tener una clara definición del ordenamiento.

²Ej: en dimensión 2, los vértices del triángulo (un 2-simplex) están en una misma línea recta. En dimensión 3, los vértices del tetraedro (un 3-simplex) están en un mismo plano.

Algoritmo Nelder-Mead Simplex (Parte 1)

Algorithm 1: Algoritmo Nelder-Mead Simplex (Parte 1)

Input : Función objetivo $f(x)$, vértices iniciales del simplex $\{x_1, x_2, \dots, x_{n+1}\}$, parámetros $\rho, \chi, \gamma, \sigma$, tolerancia ϵ , número máximo de iteraciones max_iter .

Output: Vértice x^* que aproxima el mínimo de $f(x)$.

```
1 while criterio de convergencia no satisfecho y iteraciones < max_iter do
2   Ordenar vértices  $n+1$ :  $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$ ;
3   Calcular el centroide excluyendo  $x_{n+1}$ :  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ ;
   // Reflejar
4    $x_r = \bar{x} + \rho(\bar{x} - x_{n+1})$ ;
5   Evaluar  $f_r = f(x_r)$ ;
6   if  $f_1 \leq f_r < f_n$  then
7     Aceptar  $x_r$  y continuar con la siguiente iteración;
8   else
9     if  $f_r < f_1$  then
10      Calcular el punto de expansión:  $x_e = \bar{x} + \chi(x_r - \bar{x})$ ;
11      Evaluar  $f_e = f(x_e)$ ;
12      if  $f_e < f_r$  then
13        Aceptar  $x_e$ ;
14      else
15        Aceptar  $x_r$ ;
```

Algoritmo Nelder-Mead Simplex (Parte 2)

Algorithm 2: Algoritmo Nelder-Mead Simplex (Parte 2)

```
1 if  $f_r \geq f_n$  then
2   if  $f_n \leq f_r < f_{n+1}$  then
3     Calcular contracción exterior:  $x_c = \bar{x} + \gamma(x_r - \bar{x})$ ;
4     Evaluar  $f_c = f(x_c)$ ;
5     if  $f_c \leq f_r$  then
6       Aceptar  $x_c$ ;
7     else
8       Proceder a regulación;
9   else
10    Calcular contracción interior:  $x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1})$ ;
11    Evaluar  $f_{cc} = f(x_{cc})$ ;
12    if  $f_{cc} < f_{n+1}$  then
13      Aceptar  $x_{cc}$ ;
14    else
15      Proceder a regulación;
16 Reducir el simplex:  $v_i = x_1 + \sigma(x_i - x_1)$ ,  $i = 2, \dots, n+1$ ;
17 Actualizar vértices  $x_1, v_2, \dots, v_{n+1}$ ;
18 Retornar  $x_1$  como el vértice que aproxima el mínimo.
```

● Inicialización:

- ▶ Los $n + 1$ vértices iniciales forman un simplex, un polígono que abarca n -dimensiones.
- ▶ Cada vértice es evaluado con la función objetivo $f(x)$ para identificar puntos buenos (bajos) y malos (altos).
- ▶ *Intuición:* «establecemos un punto de partida con un conjunto simple que representa nuestro espacio de búsqueda.»

● Reflexión:

- ▶ El peor vértice x_{n+1} es reemplazado probando un punto reflejado: $x_r = \bar{x} + \rho(\bar{x} - x_{n+1})$, donde \bar{x} es el centroide de los mejores n vértices.
- ▶ *Intuición:* si x_{n+1} es el peor, intentemos movernos en la dirección opuesta para mejorar.

● Expansión:

- ▶ Si el punto reflejado x_r es mejor que el mejor vértice actual x_1 , se prueba una expansión en esa dirección: $x_e = \bar{x} + \chi(x_r - \bar{x})$, donde $\chi > 1$ es el parámetro de expansión.
- ▶ *Intuición:* Si nos movemos hacia algo realmente bueno, sigamos avanzando en esa dirección para maximizar la mejora.

● Contracción:

- ▶ Si el reflejo no mejora mucho o empeora, se ajusta el simplex hacia el centroide:
 - ★ Contracción exterior: si x_r no es el peor pero tampoco es bueno: $x_c = \bar{x} + \gamma(x_r - \bar{x})$, donde $\gamma \in (0, 1)$.
 - ★ Contracción interior: si x_r es peor que el peor vértice: $x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1})$.
- ▶ *Intuición:* Si no estamos mejorando mucho, reduzcamos la escala del ajuste para enfocarnos en áreas más prometedoras.

● Regulación:

- ▶ Si tanto la reflexión como la contracción fallan, se reduce el tamaño del simplex hacia el mejor vértice x_1 : $v_i = x_1 + \sigma(x_i - x_1)$, $i = 2, \dots, n + 1$, donde $\sigma \in (0, 1)$ es el parámetro de regulación.
- ▶ *Intuición:* si estamos estancados, hagamos un ajuste drástico reduciendo el área de búsqueda hacia el mejor punto conocido x_1

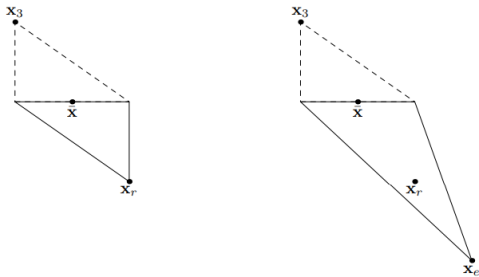


Figura Reflejar y expandir.

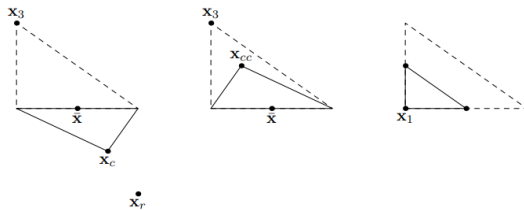


Figura Expandir, contraer y regular.

Sobre el ordenamiento con empates

El artículo de 1965 no especifica como ordenar los puntos en caso de empate. En [Jeffrey Lagarias and Wright, 1998] se adopta el siguiente criterio; se le asigna el nuevo vértice el mayor índice posible consistente con la relación

$$f(x_1^{(k+1)}) \leq f(x_2^{(k+1)}) \leq \dots \leq f(x_{n+1}^{(k+1)}).$$

Cuando un paso **sin regulación** ocurre, el vértice, $x_{n+1}^{(k)}$, es descartado. El nuevo punto aceptado creado en la iteración k y denotado $v^{(k)}$ se convierte en el nuevo vértice y asume la posición $j + 1$ con

$$j = \max_{0 \leq \ell \leq n} \left\{ \ell \mid f(v^{(k)}) < f(x_{\ell+1}^{(k)}) \right\};$$

mientras que todos los demás vértices mantiene su orden relativo a la iteración k . Por otro lado, cuando un paso **con regulación** ocurre, el único vértice que se mantiene es $x_1^{(k)}$. En este caso, solo una regla de desempate adicional se aplica: en caso

$$\min \left\{ f(v_2^{(k)}), \dots, f(v_{n+1}^{(k)}) \right\} = f(x_1^{(k)}),$$

entonces $x_1^{(k+1)} = x_1^{(k)}$.

Definition

Definimos un cambio de índice k^* de la iteración k como el menor índice de un vértice que difiere de la iteración k a la $k+1$:

$$k^* = \min \left\{ i \mid x_i^{(k)} \neq x_i^{(k+1)} \right\}.$$

Observación

Para el caso en el que no hay regulación:

$$f_j^{(k+1)} = f_j^{(k)} \quad \text{y} \quad x_j^{(k+1)} = x_j^{(k)}, \quad j < k^* \quad (1)$$

$$f_{k^*}^{(k+1)} < f_{k^*}^{(k)} \quad \text{y} \quad x_{k^*}^{(k+1)} \neq x_{k^*}^{(k)} \quad (2)$$

$$f_j^{(k+1)} = f_{j-1}^{(k)} \quad \text{y} \quad x_j^{(k+1)} = x_{j-1}^{(k)}, \quad j > k^*. \quad (3)$$

Por ello, el vector $(f_1^{(k)}, \dots, f_{n+1}^{(k)})$ es estrictamente decreciente en el orden lexicográfico.

Notación matricial

- Es conveniente usar notación matricial para describir las iteraciones del algoritmo NMS.
- El simplex Δ_k se representa como una matriz $n \times (n + 1)$, donde cada columna corresponde a un vértice:

$$\Delta_k = \begin{pmatrix} x_1^{(k)} & \cdots & x_{n+1}^{(k)} \end{pmatrix}.$$

- Para cualquier simplex $\Delta_k \subset \mathbb{R}^n$, se define la matriz $n \times n$, M_k , cuyas columnas representan las aristas del simplex que conectan los vértices $x_1^{(k)}, \dots, x_n^{(k)}$ con $x_{n+1}^{(k)}$:

$$M_k = \begin{pmatrix} x_1^{(k)} - x_{n+1}^{(k)} & x_2^{(k)} - x_{n+1}^{(k)} & \cdots & x_n^{(k)} - x_{n+1}^{(k)} \end{pmatrix} \doteq B_k - x_{n+1}^{(k)} \mathbf{1}^T. \quad (4)$$

- El volumen del simplex se calcula como:

$$\text{vol}(\Delta_k) = \frac{|M_k|}{n!}.$$

- El simplex Δ_k es **no degenerado** si M_k es no singular, es decir, si $\text{vol}(\Delta_k) > 0$. Además, se define el diámetro del simplex como:

$$\text{diam}(\Delta_k) = \max_{i \neq j} \|x_i^{(k)} - x_j^{(k)}\|_2.$$

Volumen del simplex

❶ x_0, \dots, x_n son $n + 1$ puntos afín-li si $x_1 - x_0, \dots, x_n - x_0$ son li.

❷ Si tomamos $v_i = x_i - x_0$

$$\Delta(x_0, \dots, x_n) = \left\{ x \in \mathbb{R}^n : x = \sum_{i=1}^n \alpha_i v_i, \sum_{i=1}^n \alpha_i \leq 1, \alpha_i \geq 0 \right\}.$$

❸ $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$ dada por $h(x) = Ax + x_0$, con $A = [x_1 - x_0 | \dots | x_n - x_0]$.

❹ Notamos que

$$h(\Delta(x_0, \dots, x_n)) = E = \left\{ x \in \mathbb{R}^n : x = \sum_{i=1}^n x_i v_i, \sum_{i=1}^n x_i \leq 1, x_i \geq 0 \right\}.$$

❺ Luego,

$$\text{vol}(E) = \int_0^1 \int_0^{1-x_n} \dots \int_0^{1-\sum_{i=2}^n x_i} 1 \prod_{i=1}^n dx_i = \frac{1}{n!}$$

❻ Finalmente,

$$\text{vol}(\Delta) = \int_{\Delta} 1 dx = \int_E |A| = \frac{1}{n!} |A|.$$

Observación

Durante cada iteración, la función objetivo f se evalúa en puntos de prueba (*test points*) de la forma:

$$z^{(k)}(\tau) = \bar{x}^{(k)} + \tau(\bar{x}^{(k)} - x_{n+1}^{(k)}) = (1 + \tau)\bar{x}^{(k)} - \tau x_{n+1}^{(k)}, \quad (5)$$

donde τ toma uno de los siguientes valores, dependiendo del paso del algoritmo:

$\tau = \rho$	(reflexión),
$\tau = \rho\chi$	(expansión),
$\tau = \rho\gamma$	(contracción exterior),
$\tau = -\gamma$	(contracción interior).

Observación

Quando un paso sin regulación se aplica, el único punto aceptado es uno de los puntos de prueba. Denotando por τ_k el coeficiente asociado al punto aceptado en la iteración k , el nuevo vértice $v^{(k)}$, que reemplaza $x_{n+1}^{(k)}$, se obtiene como:

$$v^{(k)} = z^{(k)}(\tau_k).$$

Observación

Cada punto de prueba $z^{(k)}(\tau)$ puede expresarse matricialmente como:

$$z^{(k)}(\tau) = \Delta_k t(\tau), \quad t(\tau) = \left(\frac{1+\tau}{n} \quad \dots \quad \frac{1+\tau}{n} \quad -\tau \right)^T.$$

En efecto, recordemos que

$$\Delta_k = \begin{pmatrix} x_1^{(k)} & \dots & x_{n+1}^{(k)} \end{pmatrix}.$$

Siguiendo el algoritmo NMS, los vértices (potencialmente desordenados) del nuevo simplex Δ_{k+1} se obtienen como columnas de $\Delta_k S_k$, donde $S_k \in \mathcal{M}_{(n+1) \times (n+1)}$ representa una transformación específica según el paso realizado:

- En pasos de reflexión, expansión o contracción, S_k toma la forma:

$$S_k = \begin{pmatrix} I_n & \frac{1+\tau_k}{n} \mathbf{1} \\ 0^T & -\tau_k \end{pmatrix},$$

donde \mathbf{e} es un vector de unos.

- En pasos de regulación, S_k se define como:

$$S_k = \begin{pmatrix} 1 & (1-\sigma)\mathbf{1}^T \\ 0 & \sigma I_n \end{pmatrix}.$$

Finalmente, para mantener la consistencia del orden de los vértices, se aplica una matriz de permutación P_k que respeta las reglas de desempate descritas anteriormente. De esta forma, el nuevo simplex se expresa como:

$$\Delta_{k+1} = \Delta_k S_k P_k = \Delta_k T_k.$$

Propiedades del algoritmo Nelder-Mead

Observación

De la definición del algoritmo, se sigue que una iteración requiere (i) una evaluación cuando la iteración termina en el paso 2, (ii) dos evaluaciones cuando la iteración termina en el paso 3 o 4, (iii) $n + 2$ evaluaciones si ocurre una regulación.

Observación

Es posible que $f_i^{(k+1)} > f_i^{(k)}$ para $2 \leq i \leq n + 1$.

Lemma

No degeneración de los volúmenes de los simplexes.

- 1 Si el simplex inicial Δ_0 es no-degenerado, entonces todos los simplexes posteriores tampoco lo son.
- 2 Luego de una iteración sin regulación, $\text{vol}(\Delta_{k+1}) = |\tau| \text{vol}(\Delta_k)$.
- 3 Luego de una iteración con regulación, $\text{vol}(\Delta_{k+1}) = \sigma^n \text{vol}(\Delta_k)$.

Simplexes no degenerados

Proof.

Un simplex Δ no degenerado si tiene volumen no nulo. Ahora bien, ciertamente el ítem (1) se deduce de (2) y (3) pues $\tau, \sigma \neq 0$.

- Para cada iteración, sin regulación supongamos sin pérdida de generalidad que $x_{n+1}^{(k)} = 0$.
- En dicho caso, el nuevo vértice es

$$v^{(k)} = M_k w = \begin{pmatrix} x_1^{(k)} & \dots & x_n^{(k)} \end{pmatrix} \begin{pmatrix} \frac{1+\tau}{n} \\ \vdots \\ \frac{1+\tau}{n} \end{pmatrix}.$$

Por ende, los vértices de Δ_{k+1} quedan formados por las columnas de M_k y $M_k w$. Dado que el volumen no depende del ordenamiento, sin pérdida de generalidad podemos asumir que el nuevo vértice es el peor, con lo cual, según (4)

$$|\det(M_{k+1})| = |\det(M_k - M_k w 1^T)| = |\det(M_k)| \cdot |\det(I - w 1^T)|.$$



Simplexes no degenerados

Proof.

Por un lado, $\{1\}^\perp$ tiene dimensión $n - 1$. Así, sean v_1, \dots, v_{n-1} los vectores asociados a $\{1\}^\perp$. Se cumple que

$$(I - w1^T)v_i = v_i - \underbrace{w1^T v_i}_{=0} = v_i.$$

O sea, $\lambda_1 = 1$ es un valor propio con multiplicidad geométrica $n - 1$. Finalmente,

$$1 - 1^T w = 1 - (1 + \tau) = -\tau$$

es valor propio y tiene multiplicidad 1. En efecto,

$$(I - w1^T)w = w - w(1 + \tau) = -\tau w.$$

Por ello,

$$|\det(I - w1^T)| = \left| \prod_{j=1}^n \lambda_j \right| = |-\tau| = \tau > 0.$$

Para el caso del paso con regulación, como M_{k+1} es una permutación de σM_k ,

$$|\det(M_{k+1})| = \left| (-1)^{\text{signo}} \det(\sigma M_k) \right| = \sigma^n |\det(M_k)|.$$



Lemma

Invarianza afín. El método de Nelder-Mead es invariante bajo transformaciones lineales afines de \mathbb{R}^n , es decir, bajo transformaciones del tipo $\phi(x) = Ax + b$, donde A es una matriz invertible. Esta invarianza se cumple en el siguiente sentido:

Cuando se minimiza una función $f(x)$ comenzando desde un simplex inicial Δ_0 , la secuencia completa de pasos generados por el algoritmo de Nelder-Mead y los valores de la función evaluada coinciden exactamente con los pasos y valores obtenidos al aplicar el algoritmo a la función transformada $\tilde{f}(z) = f(\phi(z))$, partiendo del simplex inicial transformado

$$\underbrace{\tilde{\Delta}_0 = \phi^{-1}(\Delta_0) = A^{-1}\Delta_0 - A^{-1}b}_{\text{traslación del simplex}}.$$

Proof.

En los vértices de $\tilde{\Delta}_0$, se cumple que $\tilde{f}(\tilde{x}_i^{(0)}) = f(x_i^{(0)})$. En efecto,

$$\begin{aligned}\tilde{f}(\tilde{x}_i^{(0)}) &= f(\phi(\tilde{x}_i^{(0)})) \\ &= f(A(\tilde{x}_i^{(0)}) + b) \\ &= f(A(A^{-1}\tilde{x}_i^{(0)} - A^{-1}b) + b) \\ &= f(x_i^{(0)}).\end{aligned}$$

La prueba se sigue por inducción. Supongamos que, dado

$$\tilde{\Delta}_k = A^{-1}\Delta_k - A^{-1}b$$

se tiene $\tilde{f}(\tilde{x}_i^{(k)}) = f(x_i^{(k)})$ para $1 \leq i \leq n+1$. Ahora, recordemos que cada punto de test puede escribirse

$$z^{(k)}(\tau) = \Delta_k t(\tau), \quad t(\tau) = \left(\frac{1+\tau}{n} \quad \dots \quad \frac{1+\tau}{n} \quad -\tau \right)^T.$$



Proof.

Por ende, se tiene que

$$\tilde{z}(\tau)^{(k)} = A^{-1}z(\tau)^{(k)} - A^{-1}b \implies \tilde{f}\left(\tilde{z}(\tau)^{(k)}\right) = f\left(z(\tau)^{(k)}\right).$$

Por ese motivo, dado que el ordenamiento no se ve afectado, la matriz T_k asociada a Δ_k es la misma que la de $\tilde{\Delta}_k$. De este modo, como

$$\tilde{\Delta}_{k+1} = A^{-1}\Delta_{k+1} - A^{-1}b,$$

se sigue que $\tilde{f}\left(\tilde{z}(\tau)^{(k+1)}\right) = f\left(z(\tau)^{(k+1)}\right)$. □

Observación

Usando el lema 3, el estudio del algoritmo de Nelder–Mead para una función cuadrática estrictamente convexa general en \mathbb{R}^n se puede reducir al estudio de

$$f(x) = \|x\|^2 = x_1^2 + \cdots + x_n^2.$$

Lemma

Sea f una función acotada inferiormente en \mathbb{R}^n . Cuando se aplica el algoritmo de Nelder–Mead para minimizar f , comenzando con un simplex no degenerado Δ_0 , se cumplen las siguientes propiedades:

- ❶ La sucesión $\{f_1^{(k)}\}$ siempre converge.
- ❷ En cada iteración sin regulación k , se cumple que $f_i^{(k+1)} \leq f_i^{(k)}$ para $1 \leq i \leq n+1$, con desigualdad estricta para al menos un valor de i .
- ❸ Si hay un número finito de iteraciones de regulación, entonces:
 - ❶ Cada sucesión $\{f_i^{(k)}\}$ converge cuando $k \rightarrow \infty$ para $1 \leq i \leq n+1$.
 - ❷ Se cumple $f_i^* \leq f_i^{(k)}$ para $1 \leq i \leq n+1$ y todo k , donde $f_i^* = \lim_{k \rightarrow \infty} f_i^{(k)}$.
 - ❸ Se cumple $f_1^* \leq f_2^* \leq \dots \leq f_{n+1}^*$.
- ❹ Si hay un número finito de iteraciones sin regulación, entonces todos los vértices del simplex convergen a un único punto.

Proof.

La prueba se basa en resultados de análisis real clásicos y el hecho que existe $c \in \mathbb{R}$ tal que $c \leq f(x)$ para todo $x \in \mathbb{R}^n$:

- ❶ $f_1^{(k)}$ es decreciente para todo k pues es el menor índice.
- ❷ Se sigue directamente de (1), (2) y (3).
- ❸ Sea K la iteración a partir de la cual ya no hay pasos con regulación. Entonces, por (2), para $k > K$, las sucesiones $f_i^{(k)}$ son decreciente y acotada inferiormente: convergen. Por estabilidad del límite, tendremos que $f_i^* \leq f_i^{(k)}$ para todo i y para todo k (en las regulaciones decrece o bien crece pero esto no altera que en la siguiente decrezca si no es regulación). Finalmente, nuevamente por estabilidad del límite $f_1^* \leq \dots f_i^* \leq f_{i+1}^* \leq \dots \leq f_{n+1}^*$.
- ❹ Si solo hay un número finito de pasos en los que no hay regulación (por ende, todos son de regulación para $k \geq N$, donde $N \in \mathbb{N}$), el volumen del simplex converge a cero pues $\text{vol}(\Delta_{k+1}) = \sigma^n \text{vol}(\Delta_k)$ y $\sigma \in (0, 1)$.



A continuación se analiza el algoritmo NMS para el caso en el cual no ocurren pasos de regulación. De hecho, [Torczon, 1997] observa que los pasos de regulación casi nunca ocurren: 33 de 2.9 millones. De hecho, en [Jeffrey Lagarias and Wright, 1998] demuestran que no ocurren pasos de regulación cuando la función objetivo es estrictamente convexa.

Lemma

Convergencia interrumpida. Supóngase que la función f está acotada inferiormente en \mathbb{R}^n , que el algoritmo de Nelder–Mead se aplica a f comenzando con un simplex inicial no degenerado Δ_0 , y que no ocurren pasos de regulación. Si existe un entero j , $1 \leq j \leq n$, para el cual

$$f_j^* < f_{j+1}^*, \quad \text{donde} \quad f_j^* = \lim_{k \rightarrow \infty} f_j^{(k)}, \quad (6)$$

entonces existe un índice de iteración K tal que para todo $k \geq K$, el índice de cambio satisface

$$k^* > j,$$

es decir, los primeros j vértices de todos los simplexes permanecen fijos después de la iteración K . La propiedad (6) se conoce como convergencia rota para el vértice j .

Proof.

La prueba es por contradicción. Por la hipótesis (6), existe $\delta > 0$ tal que $f_j^* + \delta = f_{j+1}^*$.

Tomemos $\epsilon > 0$ tal que $\delta - \epsilon > 0$. Dado que $f_j^* = \lim_{k \rightarrow \infty} f_j^{(k)}$, existe $K > 0$ tal que, para $k \geq K$, $f_j^{(k)} - \epsilon \leq f_j^*$. Pero entonces, para todo $k \geq K$,

$$f_j^{(k)} < f_j^{(k)} - \epsilon + \delta \leq f_j^* + \delta = f_{j+1}^*.$$

Pero, del Lema 4 (3) sabemos que $f_{j+1}^* \leq f_{j+1}^{(\ell)}$ para todo índice ℓ . Entonces, para todo $k \geq K$ y para todo ℓ ,

$$f_j^{(k)} < f_{j+1}^{(\ell)}. \quad (7)$$

Entonces, si $k^* \leq j$, para algún $k \geq K$, entonces de (3), debemos tener $f_{j+1}^{(k+1)} = f_j^{(k)}$, lo cual contradice (7). Así $k^* > j$ para $k \geq K$. □

Corollary

Supóngase que la función f está acotada inferiormente en \mathbb{R}^n , que el algoritmo de Nelder–Mead se aplica a f comenzando con un simplex inicial no degenerado Δ_0 , y que no ocurren pasos de contracción. Si el índice de cambio es 1 infinitas veces, es decir, el mejor punto cambia infinitas veces, entonces $f_1^ = \dots = f_{n+1}^*$.*

Proof.

Del lema 5, sabemos que si no se cumple que $f_1^* = \dots = f_{n+1}^*$, habría K tal que los primeros se mantiene fijos, lo cual contradice la hipótesis. □

Resultados para funciones estrictamente convexas

Definición

Decimos que $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$, con S convexo, es estrictamente convexa si para todo par de puntos y, z distintos, y $\lambda \in (0, 1)$,

$$f(\lambda y + (1 - \lambda)z) < \lambda f(y) + (1 - \lambda)f(z).$$

Más aún, (Jensen), si definimos

$$c = \sum_{i=1}^n \lambda_i z_i, \quad 0 < \lambda_i < 1, \quad \text{y} \quad \sum_{i=1}^n \lambda_i = 1$$

entonces,

$$f(c) < \sum_{i=1}^n \lambda_i f(z_i).$$

Esto implica en particular que

$$f(c) < \max\{f(z_1), \dots, f(z_n)\}.$$

Lema

Suponga que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es una función estrictamente convexa y que el algoritmo NMS se aplica a un simplex no degenerado Δ_0 . Entonces, no habrán pasos de regulación.

Proof.

Los pasos de regulación solo ocurren si el algoritmo llega a $f_r \geq f_n$ y no se acepta el punto de contracción.

- 1 Cuando $n = 1$, $f(\bar{x}) = f_n$.
- 2 Cuando $n > 1$, por la convexidad estricta de f ,

$$f(\bar{x}) = f\left(\frac{1}{n} \sum_{i=1}^n x_i\right) < f_n.$$

Consideremos primero una **contracción exterior**. Tenemos primero $f_n \leq f_r < f_{n+1}$. Luego,

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) = \gamma x_r + (1 - \gamma)\bar{x}.$$

Por ende,

$$f(x_c) < \max\{f(\bar{x}), f_r\} = f_r$$

pues $f(\bar{x}) \leq f_n$. Así, $f(x_c) < f_r$ y x_c será aceptado. En cuanto a la **contracción interior**, la situación es análoga pues

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1 - \gamma)\bar{x} + x_{n+1}.$$

Se sigue que

$$f(x_{cc}) < \max\{f(\bar{x}), f(x_{n+1})\}.$$



Observación

En el lema 1, la convexidad estricta es menester. En efecto, caso contrario, en el paso de la contracción interna, no tendríamos la desigualdad estricta.

Lemma

Suponga que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es estrictamente convexa y acotada inferiormente. Si $\rho\gamma < 1^a$, entonces

❶ $f_n^* = f_{n+1}^*$.

❷ hay infinitas iteraciones en las que $x_n^{(k+1)} \neq x_n^{(k)}$.

Se empieza desde un simplex no degenerado.

^aRecordemos que $\rho > 0$ $0 < \gamma < 1$.

Proof.

La prueba es por contradicción. Suponga que $f_n^* < f_{n+1}^*$. Dado que f es estrictamente convexa (no hay pasos de regulación) y se empieza desde un simplex no degenerado, del lema 5 sabemos que existe un entero positivo K de forma que el índice de cambio es $k^* = n + 1$ para todo $k \geq K$. S.p.d.g. tomemos $K = 0$. Se sigue que los primeros n vértices permanecen constantes. Así, $\bar{x} = \bar{x}^{(k)}$ para toda iteración y $f(x_n) = f_n^*$. La convexidad estricta de f implica que

$$f(\bar{x}) \leq f(x_n) = f_n^*.$$

Luego, el índice de cambio será $n + 1$ en cada iteración solo si un punto de contracción es aceptado y se convierte en el peor (en los otros casos, el nuevo punto es mejor que x_n). Por lo tanto,

$$x_{n+1}^{(k+1)} = \begin{cases} (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}^{(k)} & \text{si la contracción es exterior} \\ (1 - \gamma)\bar{x} + \gamma x_{n+1}^{(k)} & \text{si la contracción es interior.} \end{cases}$$



Proof.

Usando (16), podemos concluir rápidamente que

$$\lim_{k \rightarrow \infty} x_{n+1}^{(k)} = x_{n+1}^* = \bar{x}, \quad \text{con} \quad f_{n+1}^* = f(\bar{x}).$$

Del inicio teníamos que $f(\bar{x}) \leq f_n^*$, lo que implica que $f_{n+1}^* \leq f_n^*$. Pero entonces, del lema 4 se deduce que $f_n^* = f_{n+1}^*$, contradicción. La parte (2) es directa pues ya establecimos $x_1^{(k)}, \dots, x_n^{(k)}$ permanecen constantes para $k \geq K$. □

Observación

En el análisis de la convergencia, sabemos a partir del lema 5 que, si ocurre una *convergencia rota* (*broken convergence*), existe un índice j tal que todos los vértices $\{x_i^{(k)}\}$, para $1 \leq i \leq j$, permanecen constantes a partir de un cierto punto. Si esto sucede, el mejor punto $x_1^{(k)}$ no cambiará, y, en consecuencia, no podrán ocurrir pasos de expansión. Tampoco podrán ocurrir pasos de reflexión en los que se encuentre una mejora estricta sobre f_1 . Por esta razón, resulta interesante considerar una variante restringida del algoritmo de Nelder–Mead en la que no se realicen pasos de expansión. El análisis de este algoritmo restringido es más sencillo porque tanto $\text{vol}(\Delta^{(k)})$ (el volumen del simplex $\Delta^{(k)}$) como $\text{diam}(\Delta^{(k)})$ (el diámetro del simplex $\Delta^{(k)}$) son no crecientes si $\rho \leq 1$. No se discute el algoritmo restringido en mayor detalle en este [Jeffrey Lagarias and Wright, 1998], pero se puede consultar [Jeffrey Lagarias and Wright, 1998] para más información.

Resultados adicionales: caso unidimensional

En el resto de [Jeffrey Lagarias and Wright, 1998], se consideran funciones f estrictamente convexas con *conjuntos de nivel acotados*. El conjunto de nivel $\Gamma_\mu(f)$ se define como:

$$\Gamma_\mu(f) = \{x : f(x) \leq \mu\}.$$

Se dice que una función f tiene conjuntos de nivel acotados si $\Gamma_\mu(f)$ es acotado para cada μ . Esta restricción excluye funciones estrictamente convexas como e^{-x} , que tienen conjuntos de nivel no acotados. El propósito de esta restricción es que una función estrictamente convexa con conjuntos de nivel acotados tiene un único minimizador, denotado como x_{\min} .

Teorema

Convergencia unidimensional del algoritmo NMS. Sea $f : \mathbb{R} \rightarrow \mathbb{R}$ estrictamente convexa con conjuntos de nivel acotados. Suponga que el algoritmo NMS se aplica a f con una configuración de parámetros que satisface $\rho > 0, \chi > 1, \chi > \rho, \rho\chi \geq 1$ y $0 < \gamma < 1$. Suponga también que el algoritmo empieza desde un simplex no degenerado Δ_0 . Entonces,

$$x_i^{(k)} \rightarrow x_{\min}, \quad i = 1, 2$$

donde $x_{\min} \in \operatorname{argmin} \{f(x) : x \in \mathbb{R}\}$.

Theorem

Convergencia del algoritmo NMS para $\rho = 1$. Sea f una función estrictamente convexa sobre \mathbb{R} con conjuntos de nivel acotados. Suponga que el algoritmo NMS es aplicado con la configuración de parámetros: $\rho = 1, \chi > 1, 0 < \gamma < 1$ y suponga que se parte desde un simplex no degenerado Δ_0 . Entonces, existe un entero $M = M(\chi, \gamma) > 0$, tal que

$$\text{diam}(\Delta_{k+M}) \leq \frac{1}{2} \text{diam}(\Delta_k), \quad \forall k \geq K,$$

siendo $K \leq \left\lfloor \frac{|x_{\min} - x_1^{(0)}|}{\text{diam}(\Delta_0)} \right\rfloor + 1$.

Resultados adicionales: caso bidimensional

Teorema

Sea $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ una función estrictamente convexa con conjuntos de nivel acotados. Asuma que el algoritmo NMS se aplica con la siguiente configuración de parámetros: $\rho = 1$ y $\gamma = 1/2^a$. Suponga que se inicia con un simplex Δ_0 no degenerado. Entonces,

$$f_1^* = f_2^* = f_3^*.$$

^aAparte de los supuestos básicos.

Teorema

Convergencia de los diámetros a cero. Sea $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ una función estrictamente convexa con conjuntos de nivel acotados. Asuma que el algoritmo NMS se aplica con la siguiente configuración de parámetros: $\rho = 1$, $\gamma = 1/2$ y $\chi = 2$. Suponga que se inicia con un simplex Δ_0 no degenerado. Entonces,

$$\lim_{k \rightarrow \infty} \text{diam}(\Delta_k) = 0.$$

Literatura posterior

Convergence properties of the Nelder--Mead simplex method in low dimensions

[PDF] kent.edu

JC Lagarias, JA Reeds, MH Wright, PE Wright

SIAM Journal on optimization, 1998 · SIAM

The Nelder--Mead simplex algorithm, first published in 1965, is an enormously popular direct search method for multidimensional unconstrained minimization. Despite its widespread use, essentially no theoretical results have been proved explicitly for the Nelder--Mead algorithm. This paper presents convergence properties of the Nelder--Mead algorithm applied to strictly convex functions in dimensions 1 and 2. We prove convergence to a minimizer for dimension 1, and various limited convergence results for

MOSTRAR MÁS ▾

☆ Guardar 📄 Citar Citado por 9832 Artículos relacionados Las 28 versiones

Mejor resultado para esta búsqueda. [Ver todos los resultados](#)

Figura Citas (1).

Google Académico

Artículos

Cualquier momento

Desde 2024

Desde 2023

Desde 2020

Intervalo específico...

Ordenar por relevancia

Ordenar por fecha

Cualquier idioma

Buscar solo páginas en español

Crear alerta

Aproximadamente 399 resultados (0.09 s)

Mi perfil

Mi biblioteca

Convergence properties of the Nelder–Mead simplex method in low dimensions

Buscar en artículos que citan

Biological clocks, rhythms, and oscillations: the theory of biological timekeeping

DB Forger - 2024 - books.google.com

An introduction to the mathematical, computational, and analytical techniques used for modeling biological rhythms, presenting tools from many disciplines and example ...

☆ Guardar

🔗 Citar

Citado por 82

Artículos relacionados

Las 6 versiones

Can error mitigation improve trainability of noisy variational quantum algorithms?

S Wang, P Czamek, A Arasmith, M Cerezo ... - Quantum, 2024 - quantum-journal.org

Abstract Variational Quantum Algorithms (VQAs) are often viewed as the best hope for near-term quantum advantage. However, recent studies have shown that noise can severely limit ...

☆ Guardar

🔗 Citar

Citado por 76

Artículos relacionados

Las 7 versiones

🔗

quantum-journal.org

Probability density estimation of polynomial chaos and its application in structural reliability analysis

YY Weng, T Liu, ZY Zhang, YG Zhao - Reliability Engineering & System ..., 2025 - Elsevier

Polynomial chaos expansion (PCE) is a widely used approach for establishing the surrogate model of a time-consuming performance function for the convenience of uncertainty ...

☆ Guardar

🔗 Citar

Citado por 1

Artículos relacionados

Las 3 versiones

Comprehensive Task Optimization Architecture for Urban UAV-Based Intelligent Transportation System

M Rinaldi, S Primatesia - Drones, 2024 - mdpi.com

mdpi.com

Figura Citas (2).

Marcelo Gallardo (PUCP)

Convergencia del método de Nelder-Mead sin

Diciembre 2024

56 / 86

Aplicaciones

Nelder-Mead en bajas dimensiones

- ❶ Problema de minimización: $\min_{x \in \mathbb{R}^2} x_1^2 + x_2^2$.
- ❷ Problema del monopolista:

$$\max_{x \in \mathbb{R}} \pi(x; a, b, c) = (a - bx)x - cx, \quad a, b, c > 0.$$

De la CPO se llega a

$$x^* = \max \left\{ \frac{a - c}{2b}, 0 \right\}.$$

Programación Dinámica

Cake-eating

- ❶ $t = 1, \dots, T$
- ❷ x_t cantidad del pastel disponible en t .
- ❸ $x_{t+1} = x_t - c_t$.
- ❹ Buscamos resolver

$$\begin{aligned} \max_{\{c_t\}_{t=1}^T} \quad & \sum_{t=1}^T \beta^{t-1} u(c_t) \\ \text{s.a} \quad & x_{t+1} = x_t - c_t \\ & = x_1 \text{ dado.} \end{aligned}$$

Debe cumplirse que

$$\sum_{t=1}^T c_t + x_{T+1} = x_1.$$

Cake-eating

❶ $u(\cdot)$ función de utilidad que cumple supuestos clásicos: condiciones de Inada.

❷ Lagrangiano:

$$L = \sum_{t=1}^T \beta^{t-1} u(c_t) + \lambda \left[x_1 - x_{t+1} - \sum_{t=1}^T c_t \right] + \mu x_{t+1}.$$

Las CPO proveen

$$\begin{aligned} \beta^{t-1} u'(c_t) &= \lambda, \quad \forall t = 1, 2, \dots, T \\ \lambda &= \mu. \end{aligned}$$

Se deduce que

$$\underbrace{u'(c_t) = \beta u'(c_{t+1})}_{\text{condición de Euler}}.$$

Observación: la condición de Euler es una condición necesaria más no suficiente.

Del teorema de la envolvente se sigue que

$$V'_T(x_1) = \lambda = \beta^{t-1} u'(c_t), \quad \forall t = 1, \dots, T, \quad (8)$$

donde V_T es la función valor del problema con T periodo.

El enfoque de la programación dinámica permite transformar un problema de optimización de varios estados en uno de solo dos estados. Concretamente, introduciendo por comodidad $t = 0$. Consideremos el problema

$$\max_{c_0} u(C_0) + \beta V_T(x_1), \text{ s.a. : } x_1 = x_0 - c_0. \quad (9)$$

El problema (9) busca determinar el mejor consumo en el periodo $t = 0$, en vez de seleccionar la secuencia de valores. Una vez c_0 y por ende x_1 determinado, se obtiene $V_T(x_1)$. Esta función resume completamente el comportamiento óptimo de $t = 1$ en adelante.

Nota: para el enfoque de la programación dinámica no importa cómo será consumidor el paster luego del tiempo inicial. Este es el principio de optimalidad de Bellman.

Asumiendo diferenciabilidad por parte de V_T , la CPO provee

$$u'(c_0) = \beta V'_T(x_1). \quad (10)$$

Por otro lado, de (8) se sigue que

$$V'_T(x_1) = u'(c_1) = \beta^t u'(c_{t+1}). \quad (11)$$

Así, combinando (10) con (11), se deduce que

$$u'(c_t) = \beta u'(c_{t+1}).$$

Es decir, se recupera la condición de Euler.

Ejemplo concreto

Considere $u(c) = \ln c$, $T = 1$. Por lo tanto, $V_1(x_1) = \ln x_1$. Luego, para $T = 2$,

$$\frac{1}{c_1} = \frac{\beta}{c_2}, \quad x_1 = c_1 + c_2.$$

Se sigue que

$$c_1 = \frac{x_1}{1 + \beta}, \quad c_2 = \frac{\beta x_1}{1 + \beta}.$$

A partir de esto,

$$V_2(x_1) = \ln c_1 + \beta \ln c_2 = A_2 + B_2 \ln x_1$$

donde

$$A_2 = \ln \left(\frac{1}{1 + \beta} \right) + \beta \ln \left(\frac{\beta}{1 + \beta} \right), \quad B_2 = 1 + \beta.$$

Ahora para $T = 3$,

$$V_3(x_1) = \max_{x_2} \ln(x_1 - x_2) + \beta V_2(x_2).$$

La CPO provee

$$\frac{1}{c_1} = \beta V_2'(x_2).$$

Se sigue que

$$\frac{1}{c_1} = \beta \left(\frac{B_2}{x_2} \right) = \frac{\beta}{c_2}.$$

Dado que (por Euler)

$$\frac{1}{c_2} = \frac{\beta}{c_3}.$$

Esto conlleva

$$c_1 = \frac{x_1}{1 + \beta + \beta^2}, \quad c_2 = \frac{\beta x_1}{1 + \beta + \beta^2}, \quad c_3 = \frac{\beta^2 x_1}{1 + \beta + \beta^2}.$$

De este modo,

$$V_3(x_1) = A_3 + B_3 \ln x_1,$$

donde

$$A_3 = \ln \left(\frac{1}{1 + \beta + \beta^2} \right) + \beta \ln \left(\frac{\beta}{1 + \beta + \beta^2} \right) + \beta^2 \ln \left(\frac{\beta^2}{1 + \beta + \beta^2} \right) \\ B_3 = 1 + \beta + \beta^2.$$

Podemos continuar secuencialmente. Se observa desde ya un patrón.

Horizonte de tiempo infinito

$$\begin{aligned} \max_{\{c_t\}_{t=1, \dots, \infty}, \{x_t\}_{t=2, \dots, \infty}} \quad & \sum_{t=1}^{\infty} \beta^t u(c_t) \\ \text{s.a} \quad & x_{t+1} = x_t - c_t \\ & x_1 \text{ dado.} \end{aligned}$$

El enfoque de programación dinámica conlleva

$$V(x) = \max_{c \in [0, x]} u(c) + \beta V(x - c), \quad \forall x.$$

Note que si x' es la cantidad disponible en el siguiente periodo, como $x' = x - c$, se sigue que el problema es equivalente a resolver

$$V(x) = \max_{x' \in [0, x]} u(x - x') + \beta V(x'), \quad \forall x. \quad (12)$$

La ecuación (12) se conoce como ecuación de Bellman (en honor a Richard Bellman) y el objetivo es hallar la función $V(x)$.

Una función de política (policy function) es

$$c = \phi(x), \quad x' = \varphi(x) = x - \phi(x).$$

La ecuación de Euler, que se deriva de la CPO conlleva entonces a

$$u(\phi(x)) = \beta u'(\phi(x - \phi(x))), \quad \forall \quad x.$$

Shocks y programación dinámica estocástica

- 1 $u(c) \rightarrow \varepsilon u(c)$ con $\varepsilon \sim F$.
- 2 Simplificación: $\varepsilon \in \{\varepsilon_H, \varepsilon_L\}$ con $\varepsilon_H > \varepsilon_L > 0$.
- 3 Proceso de Markov de primer orden:

$$\pi_{LH} = \mathbb{P}(\varepsilon' = \varepsilon_H | \varepsilon = \varepsilon_L).$$

- 4 Como $\pi_{iH} + \pi_{iL} = 1$, se forma una matriz estocástica 2×2 , conocida como matriz de transición.
- 5 La ecuación de Bellman se torna

$$V(x, \varepsilon) = \max_{x'} \varepsilon u(x - x') + \beta \mathbb{E}_{\varepsilon' | \varepsilon} [V(x', \varepsilon')], \quad \forall (x, \varepsilon),$$

donde $x' = x - c$ como antes. La CPO se resume en

$$\varepsilon u'(x - x') = \beta \mathbb{E}_{\varepsilon' | \varepsilon} [V_1(x', \varepsilon')]$$

donde V_1 es la derivada parcial con respecto al primer argumento.

Formulación general

- ① s_t variable de estado.
- ② c_t variable de control.
- ③ $\tilde{\sigma}(s_t, c_t)$ función de pago en el periodo t .
- ④ $s_{t+1} = \tau(s_t, c_t)$, ecuación de transición.
- ⑤ $c \in C$ (eventualmente $C(s)$) y $s \in S$.
- ⑥ β factor de descuento

El programa de optimización es:

$$\max \sum_{t=0}^{\infty} \beta^t \tilde{\sigma}(s_t, c_t).$$

La ecuación de Bellman asociada es

$$V(s) = \max_{c \in C(s)} \tilde{\sigma}(s, c) + \beta V(s'), \quad \forall s \in S, s' = \tau(s, c).$$

No incorporar sub-índices alude a la estacionariedad. Una formulación alternativa es

$$V(s) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta V(s'). \quad (13)$$

Siguiendo a [?], (13) puede re-escribirse como sigue:

$$V(s) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta V(s'). \quad (14)$$

Theorem

Suponga que $\sigma(s, s')$ es una función real valuada, continua y acotada, $\beta \in (0, 1)$, y la correspondencia $\Gamma(s)$ no vacío, compacto y continua. Entonces, existe una única función valor óptima $V(s)$ que resuelve (14).

Proof.

Véase [?].



- ❶ Defina $T(F)(s) = \max_{s' \in \Gamma(s)} \sigma(s, s') + \beta F(s')$.
- ❷ $V(s) = T(V)(s)$.
- ❸ $T(F)$ resulta ser una contracción (condiciones de Blackwell (1) monotonía³, (2) descuento⁴).
- ❹ Por el teorema del punto fijo de Banach se concluye.
- ❺ Iteración de la función valor.

³ $F_1(s) \geq F_2(s)$ para todo s implica $T(F_1)(s) \geq T(F_2)(s)$ para todo $s \in S$.

⁴ $T(F + k)(s) \leq T(F)(s) + \beta k$ con $\beta \in [0, 1]$.

Theorem

Suponga que $\sigma(s, s')$ es una función real valuada, continua, cóncava y acotada, $\beta \in (0, 1)$, $S \subset \mathbb{R}^n$ convex y $\Gamma(s)$ compacta, no vacía, convexa y continua. Entonces, la solución al problema (14) es única.

Caso estocástico general

- 1 Consideramos $\varepsilon \in \Psi$ con $|\Psi| < \infty$.
- 2 La ecuación funcional de Bellman se torna en este caso

$$V(s, \varepsilon) = \max_{s' \in \Gamma(s, \varepsilon)} \sigma(s, s', \varepsilon) + \beta \mathbb{E}_{\varepsilon' | \varepsilon} V(s', \varepsilon'), \quad \forall (s, \varepsilon) \in S \times \Psi.$$

- 3 $\pi_{ij} = \mathbb{P}\{\varepsilon' = \varepsilon_j | \varepsilon = \varepsilon_i\}$, $\sum_j \pi_{ij} = 1$.
- 4 CPO:

$$\sigma_{s'}(s, s', \varepsilon) + \beta \mathbb{E}_{\varepsilon' | \varepsilon} V_{s'}(s', \varepsilon') = 0.$$

1 Parámetros:

- ▶ β : Factor de descuento, que refleja cuánto valora el agente el consumo futuro en comparación con el consumo presente.
- ▶ R : Tasa de retorno sobre el capital o recursos remanentes, que describe cómo evolucionan los recursos no consumidos.
- ▶ γ : Coeficiente de aversión al riesgo en la función de utilidad.
- ▶ y_L, y_H : Valores posibles de la dotación estocástica (ingresos adicionales).
- ▶ p_L, p_H : Probabilidades asociadas a las dotaciones y_L y y_H .
- ▶ X_L, X_H : Límite inferior y superior del espacio de estados (tamaño inicial del pastel).
- ▶ n_s : Número de puntos en la grilla del espacio de estados.
- ▶ n_c : Número de puntos en la grilla del espacio de control (niveles de consumo).

2 Definir:

- ▶ La función de utilidad $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$ si $\gamma \neq 1$, y $\ln(c)$ si $\gamma = 1$.

3 Inicializar:

- ▶ La función valor $V(X) \leftarrow 0$ para todo $X \in \{X_L, \dots, X_H\}$.

4 Repetir hasta convergencia:

1 Para cada $X \in \{X_L, \dots, X_H\}$:

- 1 Evaluar diferentes niveles de consumo $c \in \{X_L, \dots, X\}$, discretizados en n_c pasos.
- 2 Calcular el tamaño futuro del pastel: $X' = R(X - c) + y$, donde $y \in \{y_L, y_H\}$ es la dotación estocástica.
- 3 Interpolare $V(X')$ usando el valor más cercano en la grilla (espacio de estados discretizado).
- 4 Calcular la esperanza matemática de la función valor futura: $\mathbb{E}[V(X')] = p_L \cdot V(X'_L) + p_H \cdot V(X'_H)$.
- 5 Evaluar la utilidad inmediata y futura: $u(c) + \beta \mathbb{E}[V(X')]$.

- 2 Actualizar la función valor: $V(X) \leftarrow \max_c \{u(c) + \beta \mathbb{E}[V(X')]\}$.

5 Condición de parada:

- ▶ $|V_{j+1}(X) - V_j(X)| < \epsilon$ para todo X , donde ϵ es un umbral de tolerancia.

Real business cycles

Resumen del modelo RBC clásico

El modelo de Ciclos Económicos Reales (RBC, por sus siglas en inglés) se basa en las siguientes ecuaciones fundamentales:

1 Función de utilidad:

$$U = \sum_{t=0}^{\infty} \beta^t u(C_t), \quad u(C_t) = \frac{C_t^{1-\gamma}}{1-\gamma}, \gamma > 0,$$

donde C_t es el consumo en el período t , β es el factor de descuento intertemporal, y γ mide la aversión al riesgo.

2 Restricción presupuestaria intertemporal:

$$K_{t+1} = (1 - \delta)K_t + A_t F(K_t, L_t) - C_t, \quad 0 < \delta < 1,$$

donde K_t es el capital, δ es la tasa de depreciación, A_t es la productividad total de los factores (TFP), $F(\cdot)$ es la función de producción, y L_t es el trabajo.

3 Función de producción:

$$F(K_t, L_t) = K_t^\alpha L_t^{1-\alpha}, \quad 0 < \alpha < 1.$$

4 Choque tecnológico:

$$\ln(A_{t+1}) = \rho \ln(A_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2),$$

donde ρ mide la persistencia del choque y σ es la desviación estándar del choque.

Incorporación del problema de Bellman

El problema de maximización del agente representativo se puede escribir en forma recursiva usando la ecuación de Bellman:

$$V(K_t, A_t) = \max_{C_t, K_{t+1}} \{u(C_t) + \beta \mathbb{E}[V(K_{t+1}, A_{t+1})]\},$$

sujeito a:

- La restricción presupuestaria: $C_t + K_{t+1} = (1 - \delta)K_t + A_t F(K_t, L_t)$.
- Las condiciones de no negatividad: $C_t \geq 0, K_{t+1} \geq 0$.

Pasos para resolver el modelo usando iteración de Bellman

- ❶ **Discretizar los estados:** Crear una grilla para el capital K y el TFP A .
- ❷ **Inicializar:** Definir una función valor inicial $V_0(K, A)$, por ejemplo, $V_0 = 0$.
- ❸ **Iterar sobre Bellman:** Para cada punto en la grilla:
 - ❶ Evaluar todos los valores factibles de consumo C_t y capital futuro K_{t+1} .
 - ❷ Calcular la utilidad inmediata $u(C_t)$.
 - ❸ Interpolare la función valor futura $V(K_{t+1}, A_{t+1})$.
 - ❹ Maximizar la suma $u(C_t) + \beta \mathbb{E}[V(K_{t+1}, A_{t+1})]$.
- ❹ **Actualizar:** Definir la nueva función valor $V_{j+1}(K, A)$.
- ❺ **Condición de convergencia:** Detener cuando $\|V_{j+1} - V_j\| < \epsilon$.

Introducción de la desutilidad del trabajo

En el modelo RBC, además de la utilidad del consumo, se incluye una desutilidad del trabajo $v(L_t)$ para capturar el costo del esfuerzo laboral. La utilidad total del agente representativo se define como:

$$U = \sum_{t=0}^{\infty} \beta^t [u(C_t) - v(L_t)],$$

donde:

- $u(C_t) = \frac{C_t^{1-\gamma}}{1-\gamma}$, con $\gamma > 0$, mide la utilidad del consumo.
- $v(L_t)$ representa la desutilidad del trabajo, que aumenta con L_t .

La inclusión de $v(L_t)$ afecta las decisiones de trabajo y consumo, introduciendo una condición de optimalidad adicional relacionada con el esfuerzo laboral.

Propuesta de una función de desutilidad del trabajo

Una forma común de modelar la desutilidad del trabajo es:

$$v(L_t) = \frac{\phi L_t^{1+\eta}}{1+\eta},$$

donde:

- $\phi > 0$: Parámetro que escala la desutilidad del trabajo.
- $\eta \geq 0$: Elasticidad de la desutilidad del trabajo con respecto al esfuerzo laboral.

Propiedades de esta función:

- 1 Es creciente en L_t : A mayor trabajo, mayor desutilidad.
- 2 Es convexa: La tasa de desutilidad marginal del trabajo aumenta con L_t .

Esto permite capturar el costo incremental del trabajo adicional en términos de bienestar del agente.

Condiciones de primer orden (CPO)

Para resolver el problema de optimización dinámico, las condiciones de primer orden se derivan de la maximización de la utilidad sujeta a las restricciones del modelo:

1 Restricción presupuestaria:

$$C_t + K_{t+1} = (1 - \delta)K_t + A_t F(K_t, L_t).$$

2 Condición de optimalidad del consumo:

$$u'(C_t) = \beta \mathbb{E}_t [u'(C_{t+1}) ((1 - \delta) + A_{t+1} F_K(K_{t+1}, L_{t+1}))].$$

3 Condición de optimalidad del trabajo:

$$-v'(L_t) = A_t F_L(K_t, L_t) u'(C_t),$$

donde $v(L_t)$ representa la desutilidad del trabajo.

4 Ecuación de Euler:

$$\frac{1}{C_t} = \beta \mathbb{E}_t \left[\frac{1}{C_{t+1}} \left((1 - \delta) + A_{t+1} \alpha K_{t+1}^{\alpha-1} L_{t+1}^{1-\alpha} \right) \right].$$

5 Condiciones de no negatividad:

$$C_t \geq 0, \quad K_{t+1} \geq 0, \quad 0 \leq L_t \leq 1.$$

Apéndice

Ecuaciones en diferencias lineales de primer orden

Considere

$$x(t+1) = ax(t) + f(t), \quad t \in \mathbb{Z}_0^+, \quad x(t), f(t) \in \mathbb{R}.$$

Luego,

$$\begin{aligned}x(1) &= ax_0 + f(0) \\x(2) &= ax(1) + f(1) \\&= a(ax_0 + f(0)) + f(1) \\&= a^2x_0 + af(0) + f(1) \\x(3) &= ax(2) + f(2) \\&= a(a^2x_0 + af(0) + f(1)) + f(2) \\&= a^3x_0 + a^2f(0) + af(1) + f(2).\end{aligned}$$

Por inducción,

$$x(t) = a^t x_0 + \sum_{k=1}^t a^{t-k} f(k-1); \quad t = 1, 2, \dots \quad (15)$$

Usando, verificamos que (15)

$$\begin{aligned}x(t+1) &= a^{t+1}x_0 + \sum_{k=1}^{t+1} a^{t+1-k}f(k-1) \\&= a^{t+1}x_0 + \sum_{k=1}^t a^{t+1-k}f(k-1) + f(t) \\&= a \left(a^t x_0 + \sum_{k=1}^t a^{t-k}f(k-1) \right) + f(t) \\&= ax(t) + f(t).\end{aligned}$$

El caso particular $f(t) = b$ para todo b conlleva a

$$x(t) = a^t x_0 + \sum_{k=1}^t a^{t-k} b \quad (16)$$

$$= a^t x_0 (1 + a + a^2 + \dots + a^{t-1}) b \quad (17)$$

$$= a^t x_0 + \left(\frac{1 - a^t}{1 - a} \right) b. \quad (18)$$

para $a \neq 1$. Reordenando,

$$x(t) = a^t \left(x_0 - \frac{b}{1 - a} \right) + \frac{b}{1 - a}$$

Gracias



Jeffrey Lagarias, James Reeds, M. W. and Wright, P. (1998).

Convergence properties of the nelder-mead simplex method in low dimensions.
[SIAM Journal on Optimization](#), 9(1):112–147.



McKinnon, K. I. M. (1998).

Convergence of the nelder-mead simplex method to a nonstationary point.
[SIAM Journal on Optimization](#), 9(1):148–158.



Nelder, A. and Mead, R. (1965).

A simplex method for function minimization.
[Computer Journal](#), 7:308–313.



Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988).

[Numerical Recipes in C](#).
Cambridge University Press, Cambridge, UK.



Torczon, V. (1997).

On the convergence of pattern search algorithms.
[SIAM Journal on Optimization](#), 7:1–25.



Wright, M. H. (1996).

Direct search methods: Once scorned, now respectable.

In Griffiths, D. F. and Watson, G. A., editors, [Numerical Analysis 1995: Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis](#), pages 191–208. Addison Wesley Longman, Harlow, UK.