

Documento de apoyo

Matías López - Rolf Traeger - Antonia Inda

December 2020

1 Programación Dinámica

1.1 Definición

La programación dinámica es una técnica que se puede utilizar para resolver muchos problemas de optimización. En la mayoría de las aplicaciones, la programación dinámica obtiene soluciones trabajando hacia atrás desde el final de un problema hacia el principio, dividiendo así un problema grande y difícil de manejar en una serie de problemas más pequeños y manejables.

1.2 Componentes

1. **Estados:** Es la información que tenemos disponibles, cuando se toma la decisión.
2. **Etapas:** Los instantes en que se toman las decisiones
3. **transiciones:** Decisiones que se pueden tomar.
4. **Función de recurrencia:** La herramienta matemática, o optimizador, que conecta los estados de diferentes etapas con alguna transición.

1.3 Tipos de Problemas de Programación Dinámica

Se puede clasificar en 7 grupos, en base a los siguientes conceptos:

- **Naturaleza**
 - Determinista
 - Aleatorio
- **Horizonte**
 - Finito
 - Infinito
- **Herramienta de Cálculo de un problema dinámico**
 - Iteración en estados.
 - Iteración en políticas.

1.3.1 Diferencias entre las Herramientas de Cálculo

Diferencias entre las Herramientas de Cálculo: Suele aplicarse horizonte finito en este tipo de problemas y también en horizontes infinitos, cuando se poseen condiciones cíclicas.

Iteración en Políticas: Solo es aplicable en horizontes infinitos. Donde se posee un sets de estados y un sets de decisiones para cada estado suelen por muchos estados.

2 Markov decision processes

Los problemas de programación dinámica probabilísticos de horizontes infinito, son llamados también **Markov Decision Processes** (MDP)

El tomador de decisiones puede optar por maximizar la recompensa esperada obtenida por período. Entonces él o ella elegiría una decisión durante cada período en un intento de maximizar la recompensa promedio por período dada por

2.1 Componentes para formular un MDP

- **Espacio de Estados:** S

Al principio de cada período, el MDP está en algún estado i , $i \in S = 1, 2, \dots, N$. Donde S es el espacio de estados de un MDP

- **Conjunto de decisiones:** $D(i)$

Para cada estado i , hay un conjunto finito de decisiones permitidas, $D(i)$.

- **Probabilidades de transición:** $p(j|i, d)$

Suponga que un período comienza en el estado i y se elige una decisión $d \in D(i)$. Entonces, con probabilidad $p(j|i, d)$, el estado del siguiente período será j .

- **Recompensas esperadas:** r_{id}

Durante un período en el cual el estado es i , y una decisión $d \in D(i)$ fue tomada, un beneficio r_{id} será recibido.

2.2 Resolución

A través de un proceso iterativo, se busca encontrar la política óptima que resuelve el problema de programación dinámica.

2.3 ¿Qué es una política?

Una **política** es una regla que especifica cómo se elige la decisión de cada período.

Luego una política δ es una **política estacionaria** si siempre que el estado es i , la política δ elige (independientemente del período) la misma decisión (llame a esta decisión $\delta(i)$).

Finalmente se define una política óptima de la siguiente manera: si para todos los $i \in S$, una política δ presenta la siguiente propiedad:

$$V_\delta(i) = V_{\delta^*}(i) \quad (1)$$

Entonces δ^* es una **política óptima**

2.4 Value Determination Equation (VDE)

Parte del proceso iterativo incluye el cálculo de los valores $V_\delta(i)$ mencionados en el apartado anterior. Donde $V_\delta(i)$ es el beneficio esperado obtenido en un número infinito de períodos. Dado que en el primer período el estado era i y la política estacionaria era δ .

Para obtener $V_\delta(i)$, se debe construir un sistema de ecuaciones que nos permitirán obtener dichos valores. Este sistema de ecuaciones viene definido por la siguiente fórmula:

$$V_\delta(i) = r_{i,\delta(i)} + \beta \sum_{j=1}^{j=N} p(j|i, \delta(i)) V_\delta(j) \quad (2)$$

Donde:

$r_{i,\delta(i)}$: Es la utilidad esperada dado un estado inicial i y una política tomada δ . (Ganancias por ventas, menos los costos asociados a no suplir la demanda o el costo de inventariar producto).

β : Es la tasa de descuento (En este caso analizaremos el problema para un $\beta = 1$)

$\delta(i)$: Decisión tomada en el período t

Con este proceso inicia la resolución iterativa de el problema. Para iniciar, se puede elegir una política arbitraria (como por ejemplo: no comprar producto para ningún estado i posible $\delta(i) = [0, 0, 0]$).

Es importante considerar una constante g , debido a que sin ella el sistema de ecuaciones se indetermina. Por lo que se agrega al sistema (como una nueva variable, es decir como una nueva columna) de la siguiente manera:

$$V_\delta(i) + g = r_{i,\delta(i)} + \beta \sum_{j=1}^{j=N} p(j|i, \delta(i)) V_\delta(j) \quad (3)$$

Y para poder agregarla se debe adicionar una fila más al sistema de ecuaciones como:

$$V_\delta(i = S) = 0 \quad (4)$$

Donde S es el valor de i más elevado que se puede tomar.

2.5 Howard's Policy Iteration Method (HPIM)

Luego se calcula $T_\delta(i)$, que corresponde al máximo beneficio que se espera tener en un número infinito de periodos al compararse todas las decisiones que podrían tomarse dado un estado inicial i . En este punto se utilizarán los valores de $V_\delta(i)$ calculados en el apartado anterior (VDE).

$$T_\delta(i) = \max_{d \in D(i)} \left(r_{i,\delta(i)} + \beta \sum_{j=1}^{j=N} p(j|i, \delta(i)) V_\delta(j) \right) \quad (5)$$

2.6 Comparación de VDE con HPIM

En este paso se comparan los valores obtenidos de la primera iteración de Value Determination Equation con los de Howard's Policy Iteration Method, es decir se compara:

$$T_\delta(i) = V_\delta(i) \quad \forall i \quad (6)$$

Si la comparación es verdadera, entonces encontramos la política δ^* óptima.

En el caso contrario, debemos comenzar una nueva iteración, pero esta vez la política con la que se inicia la iteración es la que entrega el máximo en el apartado de HPIM, es decir, la política que genera los $T_\delta(i)$. Y se repite el procedimiento hasta que la comparación se cumpla.

3 Ejemplo del proceso iterativo "a mano"

(Ver siguientes páginas)

EJERCICIO DE JUAN "A MANO"

Datos: c : precio de compra = 300

v : precio de venta = 1500

K : costo de realizar un pedido = 1000

h : costo de almacenamiento = 200

a : demanda mínima = 0

b : demanda máxima = 3

s : capacidad de inventario = 4

i : nivel de inventario = {0, 1, 2, 3, 4}

$\delta(i)$: política de compra

probabilidades:

$P(j| \delta(i) + i)$

$i + \delta(i) \leq 4$

| j | 0 | 1 | 2 | 3 | 4 |
|------|------|------|------|------|------|
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0,75 | 0,75 | 0 | 0 | 0 | 0 |
| 0,5 | 0,25 | 0,25 | 0 | 0 | 0 |
| 0,25 | 0,25 | 0,25 | 0,25 | 0 | 0 |
| 0 | 0,25 | 0,25 | 0,25 | 0,25 | 0,25 |

beneficio: $r_{i+\delta(i)} = [1800, 175, 1425, 1950, 1750]$

VALUE DETERMINATION EQUATION

$$V_f(i) = r_{i+\delta(i)} + \sum_{j=0}^{j=N} P(j|i, \delta(i)) V_f(j)$$

$$T_f(i) = \max_{d \in D} \{ r_{id} + \sum_{j=0}^{j=N} P(j|i, d) V_f(j) \}$$

Iteración 1

$$\delta(0) = [0 \ 0 \ 0 \ 0 \ 0]$$

$$V_f(0) = -1800 + 1 \cdot V_f(0) + 0 \cdot V_f(1) + 0 \cdot V_f(2) + 0 \cdot V_f(3) + 0 \cdot V_f(4) - g$$

$$V_f(1) = 175 + 0,75 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0 \cdot V_f(2) + 0 \cdot V_f(3) + 0 \cdot V_f(4) - g$$

$$V_f(2) = 1425 + 0,5 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0 \cdot V_f(3) + 0 \cdot V_f(4) - g$$

$$V_f(3) = 1950 + 0,25 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0,25 \cdot V_f(3) + 0 \cdot V_f(4) - g$$

$$V_f(4) = 1750 + 0 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0,25 \cdot V_f(3) + 0,25 \cdot V_f(4) - g$$

$$V_f(4) = 0$$

$$g = -1800$$

$$(X_1 = V_f(0), X_2 = V_f(1), X_3 = V_f(2), X_4 = V_f(3), X_5 = V_f(4))$$

$$\text{solución: } V_f(0) = -9871,60493827 ; V_f(1) = -7238,27160494 ; V_f(2) = -4693,82716049 ;$$

$$V_f(3) = -2267,90123457 ; V_f(4) = 0 \quad y \quad g = -1800$$

$$d=0$$

$$d=1$$

$$d=2$$

$$T_f(0) = \max_{d \in D} \{ \underbrace{-1800 + V_f(0) + 1800}_{d=0} ; \underbrace{175 + 0,75 \cdot V_f(0) + 0,25 \cdot V_f(1) + 1800 - K - 1 \cdot C}_{d=1} ; \underbrace{1425 + 0,5 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 1800 - K - 2 \cdot C}_{d=2} ; \underbrace{1950 + 0,25 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0,25 \cdot V_f(3) + 1800 - K - 3 \cdot C}_{d=3} ; \underbrace{1750 + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0,25 \cdot V_f(3) + 0,25 \cdot V_f(4) + 1800 - K - 4 \cdot C}_{d=4} \}$$

$$= \max_{d \in D} \{ -9871,60493827 ; -8538,271605 ; -6293,82716 ; -4167,901235 ; -2200 \} = -2200$$

$$T_f(1) = \max_{d \in D} \{ \underbrace{175 + 0,75 \cdot V_f(0) + 0,25 \cdot V_f(1) + 1800}_{d=0} ; \underbrace{1425 + 0,5 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 1800 - K - 1 \cdot C}_{d=1} ; \underbrace{1950 + 0,25 \cdot V_f(0) + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0,25 \cdot V_f(3) + 1800 - K - 2 \cdot C}_{d=2} ; \underbrace{1750 + 0,25 \cdot V_f(1) + 0,25 \cdot V_f(2) + 0,25 \cdot V_f(3) + 0,25 \cdot V_f(4) + 1800 - K - 3 \cdot C}_{d=3} \}$$

$$= \max_{d \in D} \{ -7238,271605 ; -5993,82716 ; -3867,901235 ; -1900 \} = -1900$$

$$T_f(2) = \max_{d \in D} \left\{ \begin{array}{l} \underbrace{1425 + 0,5 V_f(0) + 0,25 V_f(1) + 0,25 V_f(2) + 1800}_{d=0}; \underbrace{1950 + 0,75 V_f(0) + 0,25 V_f(1) + 0,25 V_f(2) +}_{d=1} \\ \underbrace{0,25 V_f(3) + 1800 - K - 1C}_{d=1}; \underbrace{1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) + 1800 - K - 2C}_{d=2} \end{array} \right\}$$

$$= \max_{d \in D} \left\{ -4693,82716; -3567,901235; -1600 \right\} = -1600$$

$$T_f(3) = \max_{d \in D} \left\{ \underbrace{-1950 + 0,25 V_f(0) + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) + 1800}_{d=0}; \underbrace{1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3)}_{d=1} + 1800 - K - 1C \right\}$$

$$\max_{d \in D} \left\{ -2267,901235; -1300 \right\} = -1300$$

$$T_f(4) = \max_{d \in D} \left\{ \underbrace{1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) + 1800}_{d=0} \right\} = 0$$

$$T_f(i) = [-2200, -1900, -1975, -2675, 0]$$

Al realizar la comparación entre $T_f(i)$ y $V_f(i)$ y no resultar igual para todos sus i (sino que $T_f(i) > V_f(i) \forall i$), se debe realizar una nueva iteración, con política $d(i) = [4, 3, 2, 1, 0]$

Iteración 2

$$V_f(0) = 1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) + 0,25 V_f(4) - g - K - 4C$$

$$V_f(1) = 1750 + 0,25 V_f(0) + 0,25 V_f(2) + 0,25 V_f(3) + 0,25 V_f(4) - g - K - 3C$$

$$V_f(2) = 1750 + 0,25 V_f(1) + 0,25 V_f(3) + 0,25 V_f(4) - g - K - 2C$$

$$V_f(3) = 1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(4) - g - K - 1C$$

$$V_f(4) = 1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) - g$$

$$V_f(4) = 0$$

$$(x_1 = V_f(0); x_2 = V_f(1); x_3 = V_f(2); x_4 = V_f(3); x_5 = V_f(4))$$

Solución: $V_f(i) = [-2200, -1900, -1600, -1300, 0] \quad g = 550$

$$T_f(0) = \max_{d \in D} \left\{ \begin{array}{l} \underbrace{-1600 + V_f(0) - 550}_{d=0}; \underbrace{1750 + 0,75 V_f(0) + 0,25 V_f(1) - 550 - K - 1C}_{d=1}; \underbrace{1425 + 0,5 V_f(0) + 0,25 V_f(1) - 550 - K - 2C}_{d=2}; \\ \underbrace{1950 + 0,25 V_f(0) + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) - 550 - K - 3C}_{d=3}; \\ \underbrace{1750 + 0,25 V_f(1) + 0,25 V_f(2) + 0,25 V_f(3) - 500 - K - 4C}_{d=4} \end{array} \right\}$$

$$= \max_{d \in D} \left\{ -4550, -3800, -2700, -2250, -2200 \right\} = -2200$$

4 Recomendaciones

Les recomendamos del Libro Operations Research: Applications and Algorithms de Wayne L. Winston, hacer el ejemplo 11 del apartado 19.5 Markov Decision Processes.