Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Industrial y de Sistemas
ICS3105 Optimización Dinámica (2018-2)
Profesor Mathias Klapp

Tarea 2

Ignacio Guridi (iguridi@uc.cl) - Raimundo Herrera (rjherrera@uc.cl)

1

1. Para este problema la modelación es muy similar a la del Secretary Problem. El problema visto como un MDP es el siguiente:

   - Etapas: $\{1,\ldots,n\}$ con n el número de lotes de estacionamiento
   - Estado: 1 si es la mejor posibilidad de estacionamiento hasta ahora 0 si no. Esto se traduce en que será la mejor posibilidad si está vacío o no. 1 lote vacío, 0 lote lleno.
   - Decisión: 1 si me estaciono, 0 si no.
   - Valor inmediato:
   $$r_t(s,x) = \begin{cases} r_t(0,0) = r_t(0,1) = r_t(1,0) = 0 \\ r_t(1,1) = t \end{cases}$$
   - Probabilidades: como solo depende de si el estacionamiento está vacío o no, se colapsa desde $p_t(s' \mid_{s,x})$ a $p_t(s')$
   $$p_t(s') = \begin{cases} p_t(0) = 1 - p_t \\ p_t(1) = p_t \end{cases}$$
   - Ecuación de Bellman y Value-to-go:
   $$V_t(s) = max\{r_t(s,1); r_t(s,0) + \mathbb{E}_{s'}(V_{t+1}(s'))\}$$
   Lo que se traduce en lo siguiente para el caso:
   $$V_t(s) = \begin{cases} V_t(0) = max\{0;\ 0 + P_t(0)\cdot V_{t+1}(0) + P_t(1)\cdot V_{t+1}(1)\} \\ V_t(1) = max\{t;\ 0 + P_t(0)\cdot V_{t+1}(0) + P_t(1)\cdot V_{t+1}(1)\} \end{cases}$$
   Lo que se reduce a:
   $$V_t(s) = \begin{cases} V_t(0) = (1-p_t)\cdot V_{t+1}(0) + p_t \cdot V_{t+1}(1) \\ V_t(1) = max\{t;\ V_t(0)\} \end{cases}$$
   - Valores terminales: considera el caso de que si está al final, ya no puede hacer nada más y se va.
   $$V_n(s) = \begin{cases} V_n(0) = 0 \\ V_n(1) = n \end{cases}$$

2. Para demostrar esto se procederá por inducción. La inducción será sobre τ. Para esto, se probará primero que si $\tau = 2$, es decir, se decide seguir para t = τ, entonces para todo $t \leq \tau$ también se decide seguir.

- Caso base: $\tau = 2$ Si $\tau = 2$ y se decide seguir, esto quiere decir, por la estructura del value to go, que $\tau = 2 \leq V_2(0)$.

  Dado eso, podemos analizar los casos anteriores, en este caso $t = 1$. Para $t = 1$, quedarse implica un *reward* de 1. Pero
  $$1 < 2 \leq V_2(0),$$
  Por lo que no puede ser óptimo ya que quedándose y decidiendo quedarse en el siguiente paso, el $\tau = 2$, habría obtenido una recompensa estrictamente mayor.
- Hipótesis de inducción: si $\tau = n$ entonces para todo $t < \tau$ es óptimo no estacionar.
- Paso inductivo: Dada la hipótesis de inducción, deducir el caso $\tau = n + 1$.

  Como en $n + 1$ no es óptimo estacionar, entonces quiere decir, por la estructura del value to go, que
  $$n + 1 \leq V_{n+1}(0),$$
  ya que de lo contrario se habría quedado (el máximo habría escogido $n + 1$).

  Dado lo anterior, en particular se sabe que $n < n + 1$ por lo tanto, en n tampoco debería parar porque si paro en n obtengo una recompensa inferior a $n + 1$, que ya se sabe que es inferior a la real, porque en el óptimo no se elige $n + 1$.

  Por hipótesis de inducción, todos los pasos menores a n no se eligen si n no se elige. Entonces, como n tampoco se elige si $n + 1$ no se elige, sumando ambas partes se completa el cuadro, es decir, por hipótesis de inducción se tiene que de 2 a $n - 1$ no se detiene si n no se detiene, y por lo explicado en el párrafo anterior, no se detiene en n, por lo que para todo $t < n + 1$ no se detiene.

■

3. Demostrar que para $N \geq 2$, se da que $\tau \geq 1$. Esto es, dejar pasar la primera opción siempre (la definición de τ incluye al número en lo que se va a dejar pasar, por eso el $\geq$).

   Por contradicción asumamos que no se cumple lo anterior, entonces se puede ver que para todo t $V_t(1) = t$. Por lo que reemplazando en el value to go se tiene:
   $$V_t^*(0) = (1 - p_t) \cdot V_{t+1}^*(0) + p_t \cdot (t + 1)$$
   Reemplazando V_{t+1} por $t + 1$.

   Desenvolviendo un par de resultados para $t + 1$ y $t + 2$, se observa la siguiente estructura:
   $$\begin{aligned}
   &V_{t+1}^*(0) = (1 - p_{t+1}) \cdot V_{t+2}^*(0) + p_{t+1} \cdot (t + 2)\\
   &V_{t+2}^*(0) = (1 - p_{t+2}) \cdot V_{t+3}^*(0) + p_{t+2} \cdot (t + 3)\\
   &\dots
   \end{aligned}$$
   Desarrollando al reemplazar se ve lo siguiente:
   $$\begin{aligned}
   V_{t+1}^*(0) &= (1 - p_{t+1})((1 - p_{t+2})V_{t+3}^*(0) + p_{t+2}(t + 3)) + p_{t+1} \cdot t + 2\\
   &= (1 - p_{t+1})(1 - p_{t+2})V_{t+3}^*(0) + (1 - p_{t+1})p_{t+2}(t + 3) + p_{t+1} \cdot t + 2
   \end{aligned}$$
   Si a lo anterior le agregamos el hecho de que $V_N^*(0) = 0$, se puede deducir la siguiente ecuación a partir de la recursión:
   $$V_t^*(0) = \sum_{k=t+1}^{N} k \cdot p_k \cdot \prod_{k'=1}^{k-1} (1 - p_{k'})$$
   Lo anterior tras observar la estructura de la recursión, y ver que quedaban elementos de la forma $(1 - p_t)(1 - p_{t+1}) \dots (1 - p_{t+k-1})p_{t+k} \cdot (t + k + 1)$ cada uno con menos $(1 - p_t)$ que el anterior y con un $(t + k + 1)$ mayor.

   Con lo anterior se puede deducir que para $t = 1$ $V_1^*(0)$ es mayor que $V_1^*(1)$ y como por la definición del value to go $V_1^*(1) \geq V_1^*(0)$, es una contradicción, y se demuestra que τ debe ser mayor que 1.

■

4. Si es óptimo estacionarse desde el lote $\tau+1$ en adelante, entonces para todos los lotes siguientes el óptimo va a ser t, ya que de lo contrario, estarían eligiendo seguir, y eso contradecería el hecho de que τ era el último en el que se decidía seguir. Por ende,

$$V_t^*(1) = t, \quad \forall t > \tau$$

Y por lo tanto, similar a la demostración anterior, se tiene que la fórmula es válida también para este caso:

$$V_\tau^*(0) = \sum_{k=\tau+1}^{N} k \cdot p_k \cdot \prod_{k'=1}^{k-1} (1 - p_{k'})$$

5. Para lo pedido primero hay que evaluar la expresión anterior usando p en vez de p_k, esto es:

$$V_\tau^*(0) = \sum_{k=\tau+1}^{N} k \cdot p \cdot \prod_{k'=1}^{k-1} (1 - p)$$

Lo que es equivalente por la no dependencia de los índices a:

$$V_\tau^*(0) = \sum_{k=\tau+1}^{N} k \cdot p \cdot (1-p)^{k-1}$$

De ahí que el algoritmo corresponde a encontrar el primer τ para el cual se cumple la siguiente desigualdad

$$\tau \geq \sum_{k=\tau+1}^{N} k \cdot p \cdot (1-p)^{k-1}$$

Con lo anterior para los casos enunciados se obtiene, resolviendo con WolfraAlpha, lo siguente:

- 20%: con $\tau = 4$
- 40%: con $\tau = 2$
- 60%: con $\tau = 1$

Lo cual señala que probablemente se cometió algún error en el cómputo de la fórmula para derivar el τ. Ya que los resultados son muy bajos y siguen disminuyendo a medida que se aumenta la probabilidad, de hecho debería ser al revés, que aumentara a medida que la probabilidad aumenta. Esto sugiere cambios necesarios en la fórmula derivada o en la formulación del proceso markoviano en cuestión.

En cualquier caso, es interesante observar como la estructura recursiva del problema y las propiedades del mismo permiten derivar fórmulas analíticas que se desprenden de la recursión, dejando todo en términos de las probabilidades, el τ en cuestión y la cantidad de lotes.

6. Dada la fórmula obtenida anteriormente, independiente de su correctitud, se puede observar que la fórmula en cuestión se vuelve la siguiente:

$$V_\tau^*(0) = \sum_{k=\tau+1}^{N} k \cdot \frac{1}{k} \cdot \prod_{k'=1}^{k-1} (1 - \frac{1}{k'})$$

Y resolviendo dicha expresión se obtiene que da un resultado igual a 0 para cualquier n, lo que sugiere la misma conclusión anterior sobre la correctitud de la expresión derivada. Sin embargo, es importante notar que la razón por la que se va a 0 es porque el índice del que comienza la pitatoria es 1, no obstante al aumentar ese índice en 1, se comprueba que la suma converge a la función $\psi(n) + \gamma$, siendo ambos la función *digamma* y la constante de Euler-Mascheroni respectivamente.

En cuanto al resultado para dicho $n = 100$ el τ obtenido corresponde a 4 nuevamente.

2

1. Dada la función $f : \mathbb{R}^n \to \mathbb{R}^n$, en donde f es una contracción. Se demuestra que el sistema de ecuaciones $f(x) = x$ para $x \in \mathbb{R}$ posee solución única por contradicción.

   Además, se tiene que f es una contracción, por lo que $||f(U) - f(V)|| \leq c \cdot ||U - V|| \quad \forall \quad U, V \in \mathbb{R}$, con un c escalar tal que $0 < c < 1$

   Suponiendo dos soluciones, A y B, donde $A \neq B$.

$$\begin{aligned} ||A - B|| &= ||f(A) - f(B)|| && \text{pues } A \text{ y } B \text{ son solución} \\ &\leq c \cdot ||A - B|| && \text{pues } f \text{ es una contracción} \\ 1 &\leq c && \text{pues } ||A - B|| > 0 \\ &\rightarrow\leftarrow && \text{pues } c < 1 \end{aligned}$$

   Por lo tanto, se demuestra que no existen 2 soluciones diferentes para la ecuación $f(x) = x$

2. (a) Se mostrará por contrucción. Por lo visto en clases, se tiene que

$$\begin{aligned} V^d &= \sum_{t=1}^{\infty} \lambda^{t-1} \cdot P_d^{t-1} \cdot r_d \\ &= \lambda^0 \cdot P_d^0 \cdot r_d + \sum_{t=2}^{\infty} \lambda^{t-1} \cdot P_d^{t-1} \cdot r_d \\ &= r_d + \lambda \cdot P_d \cdot \sum_{t=2}^{\infty} \lambda^{t-2} \cdot P_d^{t-2} \cdot r_d \\ &= r_d + \lambda \cdot P_d \cdot \sum_{t=1}^{\infty} \lambda^{t-1} \cdot P_d^{t-1} \cdot r_d \\ &= r_d + \lambda \cdot P_d \cdot V^d \\ V^d - \lambda \cdot P_d \cdot V^d &= r_d \\ V^d \cdot (1 - \lambda \cdot P_d) &= r_d \\ V^d &= (1 - \lambda \cdot P_d)^{-1} \cdot r_d \end{aligned}$$

   (b) Teniendo un vector $a \geq 0$, con $a \in \mathbb{R}^{|\mathbb{S}|}$, un factor temporal λ tal que $0 \leq \lambda \leq 1$ y una matrix de probabilidades de transición bajo la regla de desición d, $P_d = \{p(j|i, d(i))\}_{i,j \in}$, con $p : \mathbb{S} \to [0, 1]$
   Se demostrará por contradicción que para $a \geq 0$ se cumple que $(1 - \lambda P_d)^{-1} \cdot a \geq a$
   Asumimos que $(1 - \lambda P_d)^{-1} \cdot a < a$

$$\begin{aligned} (1 - \lambda P_d)^{-1} \cdot a &< a \\ a &< (1 - \lambda \cdot P_d) \cdot a \\ a &< a - \lambda P_d \cdot a \\ \lambda \cdot P_d \cdot a &< 0 \\ &\rightarrow\leftarrow && \text{pues } \lambda \geq 0, \\ & && P_d \geq 0, \text{ por ser matrix de probabilidades y} \\ & && a \geq 0 \text{ por definición} \end{aligned}$$

   Con esto se demuestra que $(1 - \lambda P_d)^{-1} \cdot a \geq a$, y como $a \geq 0$, queda demostrado también que $(1 - \lambda P_d)^{-1} \cdot a \geq 0$

3

1. MDP correspondiente

   - Notación

$$
\begin{aligned}
B: &\quad \text{cantidad total de quirófanos} \\
h(x): &\quad \text{beneficio por hacer } x \text{ operaciones} \\
c(x): &\quad \text{penalización por postergar } x \text{ operaciones}
\end{aligned}
$$

   - Etapas

$$t \in \{0, \dots, \infty\}$$

   - Estados

$$\mathbb{S} = \{s\}, \quad s \in \{0, 1, 2\}$$

   - Variables de decisión

$$\mathbb{X}_s = \begin{cases} \{\text{azul, negro}\} & \text{si } s = 0 \\ \{\text{verde, rojo}\} & \text{si } s = 1 \\ \{\text{gris }\} & \text{si } s = 2 \end{cases}$$

   - Probabilidades de transición

$$\mathbb{P}(s_{t+1} = k | s_t, a_t) = \begin{cases} 0.5 & \text{si } k = 0, s_t = 0, a_t = \text{azul} \\ 0.5 & \text{si } k = 1, s_t = 0, a_t = \text{azul} \\ 0.1 & \text{si } k = 0, s_t = 0, a_t = \text{negro} \\ 0.9 & \text{si } k = 2, s_t = 0, a_t = \text{negro} \\ 0.2 & \text{si } k = 0, s_t = 1, a_t = \text{rojo} \\ 0.8 & \text{si } k = 1, s_t = 1, a_t = \text{rojo} \\ 0.6 & \text{si } k = 1, s_t = 1, a_t = \text{verde} \\ 0.4 & \text{si } k = 2, s_t = 1, a_t = \text{verde} \\ 1 & \text{si } k = 2, s_t = 2, a_t = \text{gris} \\ 0 & \text{e.o.c} \end{cases}$$

   - Valor inmediato

$$r(s, a) = \begin{cases} -3.5 & \text{si } s = 0, a = \text{azul} \\ -1.7 & \text{si } s = 0, a = \text{negro} \\ -2.6 & \text{si } s = 1, a = \text{rojo} \\ -2.6 & \text{si } s = 1, a = \text{verde} \\ 1 & \text{si } s = 2, a = \text{gris} \end{cases}$$

   - Value-to-go

$$u_t^*(s_t) = \max_{a_t \in \mathbb{X}} \left\{ r(s_t, a_t) + \sum_{i=0}^{\infty} \mathbb{P}(s_{t+1} = i | s_t, a_t) \cdot u_{t+1}^*(i) \right\}$$

2. (a) Como podemos observar en los gráficos a continuación, el algoritmo termina en un menor número de etapas cuando el futuro importa menos (3 iteraciones vs. 80 iteraciones). Sobre el valor del *value-to-go* no se puede hacer conclusiones, pues están ponderados por factores distintos por cada etapa (0.1 vs 0.9). Para este ejercicio, la política óptima no cambia según se mire a corto o largo plazo. En ambas será (*negro, verde, gris*), para cada estado, respectivamente

   (b) Usando el algortimo de iteración por política para resolver el problema, se tiene que se termina a la primera iteración. Inicia con la política (*azul, verde, gris*) y ya en la segunda iteración llega a su política final (*negro, verde, gris*). Esto es independiente del factor de importancia del futuro. El resultado de la política óptima coincide con el llegado por iteración de valor.

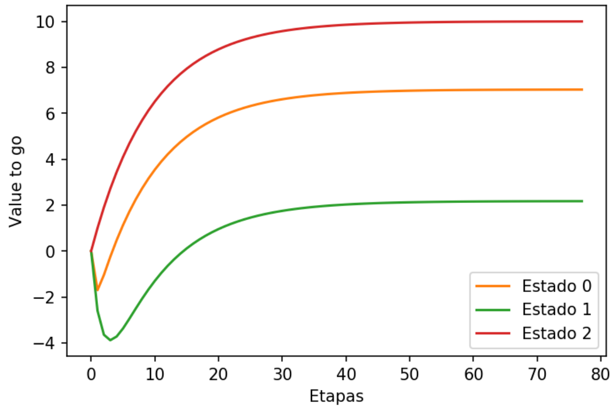

Figure 1: Gráfico que muestra la evolución del *value-to-go* de cada estado en función de la iteración para $\lambda = 0.9$

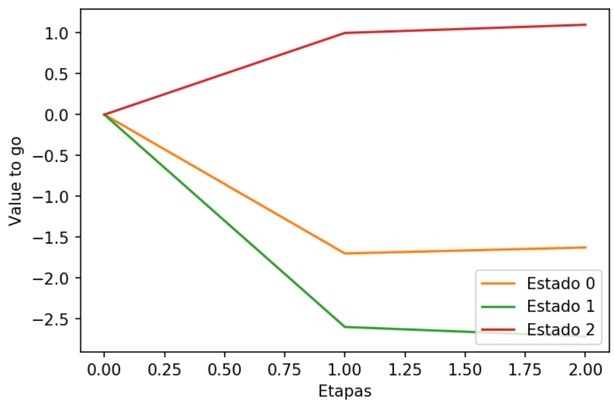

Figure 2: Gráfico que muestra la evolución del *value-to-go* de cada estado en función de la iteración para $\lambda = 0.1$

(c) Problema dual

Función objetivo

$$\max_{f \geq 0} \sum_{s \in \mathbb{S}} \sum_{x \in \mathbb{X}} r(s, x) \cdot f_{s,x}$$

Restricciones

$$\sum_{x \in \mathbb{X}} f_{s_1,x} - \sum_{s \in \mathbb{S}} \sum_{x \in \mathbb{X}} \lambda \cdot \mathbb{P}(s_1 | s, x) \cdot f_{s,x} = a_{s_1} \quad \forall s_1 \in \mathbb{S}$$

Se escoge $a_s = 1 \quad \forall s \in \mathbb{S}$

Se llegó a los resultados que se muestran a continuación. Se resolvió en 2 iteraciones, con un valor objetivo óptimo de 19.2069

$$\begin{aligned} f_{0,azul} &= 0 \\ f_{0,negro} &= 1.0989 \\ f_{1,verde} &= 2.17391 \\ f_{1,rojo} &= 0 \\ f_{2,gris} &= 26.7272 \end{aligned}$$

Se puede observar que la política óptima corresponde con los $f > 0$. Esta sería entonces (*negro, verde, gris*), igual a la oplítica obtenida cn los métodos de iteración y de valor

4

1. La modelación es la siguiente:
   - Etapas: $t \in \{1, \ldots, \infty\}$
   - Estados: $s \in \{0, 1, 2, 3\}$
   - Decisión: Para cada s existen las siguientes posibles acciones:

     $$X(s) = \{0, 3 - s\}$$

   - Probabilidades de transición:

     $$p(s_1 \mid s, X(s)) = \begin{cases} P(D \geq s + X(s)) & \text{si } s_1 = 0 \\ P(D = s + X(s) - s_1) & \text{e.o.c} \end{cases}$$

     Con $s_1 = s + X(s) - d$
   - Costo inmediato:

     $$r(s, X(s)) = x(s) \cdot h + P(D \geq s + X(s)) \cdot min(0, s + X(s) - D)$$