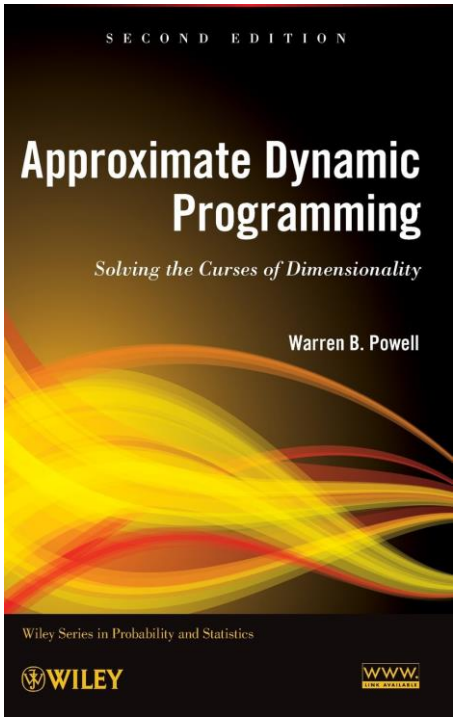
Cuarta Parte:
Clase 1 – Introducción a Approximate Dynamic Programming

Optimización Dinámica - ICS



Mathias Klapp

¿Qué hemos visto?

- DP determinístico como alternativa en la resolución de problemas combinatoriales.

- MDPs para resolver problemas dinámicos-estocásticos con horizontes finito e infinito.

- Métodos de solución exactos:
    1. Backward Dynamic Programming (Horizonte finito)
    2. Iteración de Valor (Horizonte infinito)
    3. Iteración de Política (Ambos)
    4. Métodos basados en LP (Ambos)

- ¿Y si el problema posee la ``maldición de la dimensionalidad''?

``unfortunately, in the vast majority of real applications we cannot solve Bellman's equations exactly''

Warren Powell.

- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

MDP

Bucamos política $\pi = (d_1, d_2, \dots d_T)$ que resuelve las ecuaciones de Bellman para cada $t \leq T$ y estado $s \in \mathbb{S}_t$ dado el estado inicial s_1.

Ecuación de Bellman:

$$V_t(s) = \max_{x \in \mathbb{X}_t(s)} \left\{ r_t(s,x) + \sum_{s' \in \mathbb{S}_{t+1}} p_t(s'|s,x) V_{t+1}(s') \right\}$$

Una política óptima cumple:

$$d_t^*(s) \leftarrow \operatorname*{argmax}_{x \in \mathbb{X}_t(s)} \left\{ r_t(s,x) + \sum_{j \in \mathbb{S}_{t+1}} p_t(j|s,x) V_{t+1}(j) \right\}$$

La triple maldición de la dimensionalidad

$\forall t \leq T, \forall s \in \mathbb{S}_t$:

$$V_t(s) = \max_{x \in \mathbb{X}_t(s)} \left\{ r_t(s,x) + \sum_{j \in \mathbb{S}_{t+1}} p_t(j|s,x) V_{t+1}(j) \right\}$$

1. El tamaño de $\mathbb{S}$ es grande.
   - Ejemplo: Problema de ruteo dinámico

2. Optimizar el problema sobre $\mathbb{X}_t(s)$ es difícil.
   - Ejemplo: Un problema NP-completo.

3. Gran cantidad de transiciones futuras impide evaluar la esperanza del *value-to-go*.
   - Ejemplo: Potenciales realizaciones de un subset $S \subset \{1, \dots, n\}$

Dificultad adicional:

Disponibilidad de información probabilística

- En problemas reales de decisión secuencial, es complejo calibrar el modelo de probabilidad:

$$p_t(s_{t+1}|s_t, x_t)$$

- Típicamente se cuenta con un ``**simulador**'' del proceso representativo de la distribución de probabilidades.
  - Historia como simulador
  - Simulador computacional (Arena, Simio).
  - Corridas de entrenamiento.

- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

Approximate Dynamic Programming

- Conjunto de estrategias heurísticas para resolver DP's y MDP's ``malditos''.

- Se busca una política heurística π^H Buena.
  - No necesariamente óptima, ojalá con garantía de calidad.

- No hay una forma y selección de la estrategia depende de la ``maldición'' específica del problema, de la disponibilidad de datos y de requerimientos de cómputo *online*.

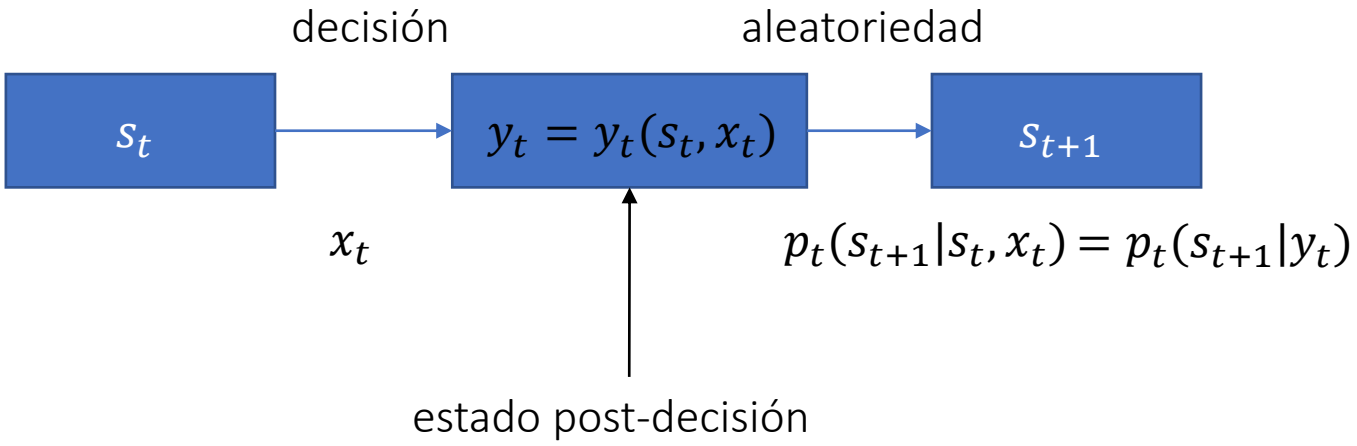
- Ejemplos:
  - Espacio de estados de gran tamaño: aproximación de función de valor
  - Espacio de decisiones complejo: heurísticas de decisión
  - Transición explosiva a futuro : simulación de Monte Carlo
  - Requerimientos de cómputo: max. tiempo disponible por decisión.

- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post-decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

Estado de post-decisión

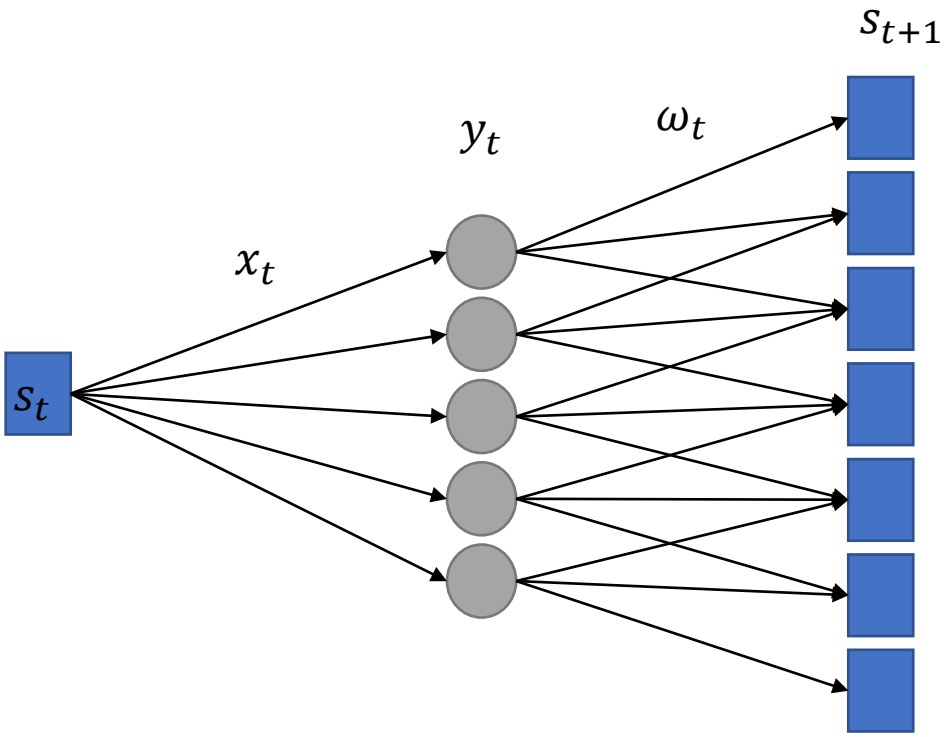
La dinámica del proceso estocástico de un MDP en la etapa t:

1. Sistema se encuentra en estado s_t (de **pre-decisión**)
2. Se toma una decisión x_t.
3. Proceso estocástico $s_{t+1} = f_t(s_t, x_t, \omega_t)$ genera transición al estado s_{t+1} con probabilidad $p_t(s_{t+1}|s_t, x_t)$.



Estado de post-decisión

- Separa la decisión determinística y la transición estocástica mediante estado intermedio $y_t = y_t(s_t, x_t)$.
- El estado de post-decisión y_t es determinístico dado s_t, x_t.



Reformulación de un MDP desde el estado de post-decisión

Tomemos la ecuación de Bellman:

$$V_t(s_t) = \max_{x_t \in \mathbb{X}_t(s_t)} \left\{ r_t(s_t, x_t) + \sum_{s_{t+1} \in \mathbb{S}_{t+1}} p_t(s_{t+1}|y_t(s_t, x_t)) V_{t+1}(s_{t+1}) \right\}$$

y definimos $Q_t(y_t)$: *value to-go* en estado de post-decisión $y_t(s_t, x_t)$:

$$Q_t(y_t) = \mathbb{E}_{s_{t+1}}(V_{t+1}(s_{t+1})|y_t)$$

Implica que el MDP se puede reformular como:

$$\boldsymbol{V_t(s_t) = \max_{x \in \mathbb{X}_t(s_t)} \{r_t(s_t, x) + Q_t(y_t(s_t, x))\}}$$

MDP desde el estado de post-decisión

$$\boldsymbol{V_t(s_t) = \max_{x_t \in \mathbb{X}_t(s_t)} \{r_t(s_t, x_t) + Q_t(y_t(s_t, x_t))\}}$$

Aplicando esperanza $\mathbb{E}_{s_t}(\cdot\,|y_{t-1})$ a ambos lados:

$$\mathbb{E}_{s_t}(V_t(s_t)|y_{t-1}) = \mathbb{E}_{s_t}\left(\max_{x_t \in \mathbb{X}_t(s_t)} \{r_t(s_t, x_t) + Q_t(y_t(s_t, x_t))\} \,\middle|\, y_{t-1}\right)$$

Ecuación de Belmann de post-decisión:

$$\boldsymbol{Q_{t-1}(y_{t-1}) = \sum_{s_t \in \mathbb{S}_t} p_t(s_t|y_{t-1}) \cdot \max_{x_t \in \mathbb{X}_t(s_t)} \{r_t(s_t, x_t) + Q(y_t(s_t, x_t))\}}$$

- Modelo equivalente al MDP planteado en función de $V_t(s)$

MDP desde el estado de post-decisión

Buscamos decisión en etapa t y estado s_t :

$$d_t^*(s_t) \in \underset{x \in \mathbb{X}_t(s_t)}{\operatorname{argmax}} \{r_t(s_t, x) + Q_t(y_t(s_t, x))\}$$

- Es un problema determinístico si conocemos $Q_t(y)$.
- ¿Cómo aprendemos Q?

Paradigmas para aprender Q:

- ***Online:*** Aproximar $Q_t(y)$ en línea después de observar estado s_t y conocer potenciales estados de post-decisión y_t factible:
  - Solo requiere aproximar Q en estados que el sistema visita.
  - Cómputo de Q online puede ser incompatible con necesidad de decisión *online*.
- ***Offline:*** Antes de ejecutar operación, pre-computar, *a.k.a.* ´´entrenar'' o ``aprender'', una aproximación de $Q_t(y)$ para todo t y todo $y \in \mathbb{Y}_t$. Luego, usarla *online*.
  - Exije aproximar Q para todo t y para todo estado y de post-decisión.
  - Cómputo de Q offline permite decisiones pseudo-inmediatas *online*.

- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post-decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

Resumen de técnicas ADP

1. Decisiones Miopes.

2. *Lookaheads, roll-outs* y horizontes rodantes (simple/estocástico).

3. Aproximación de función de valor.

4. Aproximación de política de decisión.

5. ALP (Approximate Linear Programming)

- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post-decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

Evaluación exacta de una política dada

Evaluar el objetivo $V_1^\pi(s_1)$ de una política π dada exige calcular recursivamente el *value-to-go* $V_t^\pi(s_t)$ para todo t y s_t.

- Es decir, $\forall s_t \in \mathbb{S}_t$ y $t \in \{1, \dots, T\}$:

$$V_t^\pi(s_t) = r_t\left(s_t, d_t^\pi(s_t)\right) + \sum_{j \in \mathbb{S}_{t+1}} p(j|s_t, d_t^\pi(s_t)) \cdot V_{t+1}^\pi(j)$$

- Si el procedimiento **sufre de una de las dos maldiciones** (estados y transiciones):

¿Cómo evaluar una política en ese caso?

Evaluación aproximada de una política π

- Alternativa: **Simulación de Monte Carlo**

- Simular la ejecución de la política π varias veces (réplicas) y estimar el costo esperado promedio.

- En cada réplica, ejecutar el proceso de decisión desde $t = 1$ hasta $t = T$ simulando una realización independiente de las variables aleatorias involucradas.

Ingrediente 1: Teorema Central del Límite (CLT)

I. Sea V una variable aleatoria con media $\mu < \infty$ y varianza $\sigma^2 < \infty$.

II. Sea una muestra aleatoria i.i.d. $\Omega = \{V_1, V_2, \dots, V_m\}$ de V.

III. Un estimador de μ es el **promedio muestral**:

$$\bar{V}_\Omega = \frac{1}{m} \sum_{\omega \in \Omega} V_\omega$$

Resultados básicos:

- Es insesgado: $\mathbb{E}(\bar{V}_\Omega) = \mu$
- Varianza tiende a cero: $\mathrm{Var}(\bar{V}_\Omega) = \frac{\sigma^2}{m}$
- CLT: $\lim_{|\Omega| \to \infty} \bar{V}_\Omega \sim \mathrm{Normal}\left(\mu, \frac{\sigma^2}{m}\right)$

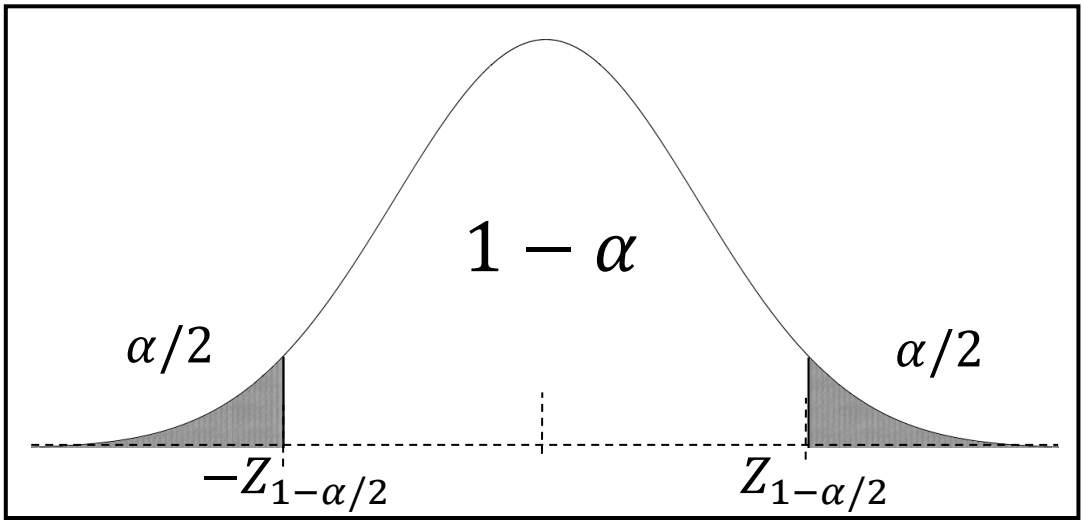
Ingrediente 2: Intervalo de Confianza (IdC)

Con probabilidad $1-\alpha$ se cumple que:

$$\mu \in \left[\bar{V}_\Omega - S_\Omega \frac{Z_{1-\frac{\alpha}{2}}}{\sqrt{\mathrm{m}}}; \bar{V}_\Omega + S_\Omega \frac{Z_{1-\frac{\alpha}{2}}}{\sqrt{\mathrm{m}}}\right]$$

, con $S_\Omega^2 = \frac{1}{\mathrm{m}-1}\sum_{\omega\in\Omega}(V_\omega - \bar{V}_\Omega)^2$, estimador de varianza de V.

- Precisión de estimación = $\frac{S_\Omega}{\bar{V}_\Omega} \cdot \frac{Z_{1-\frac{\alpha}{2}}}{\sqrt{\mathrm{m}}}$



$1-\alpha$	$\alpha/2$	$Z_{1-\alpha/2}$
70%	15%	1,04
80%	10%	1,28
90%	5%	1,64
95%	**2.5%**	**1,96**
97,5%	1,25%	2,24
99%	0,5%	2,58
99,9%	0,05%	3,29

Evaluación simulada de una política π

- **Input:** Política $\pi = (d_1, \dots, d_T)$, estado inicial s_1
- **Simular m ejecuciones de π** en una muestra Ω:

  Para cada corrida $\omega \in \Omega$:

  1. Inicializar: $s \leftarrow s_1, V_\omega^\pi \leftarrow 0$
  2. Para cada etapa $t = 1, \dots, T$:
     - Decidir: $x_t \leftarrow d_t^\pi(s)$
     - Actualizar valor: $V_\omega^\pi \leftarrow V_\omega^\pi + r_t(s, x_t)$
     - Actualizar estado: $s \leftarrow f_t(s, x_t, \omega_t)$
  3. Guardar indicador: V_ω^π

- **Estimar el valor de la política:**
  - $\bar{V}_\Omega^\pi := \frac{1}{m} \sum_{\omega \in \Omega} V_\omega^\pi$
  - $S_\Omega^\pi = \sqrt{\frac{1}{|\Omega|-1} \sum_{\omega \in \Omega} (V_\omega^\pi - \bar{V}_\Omega^\pi)^2}$
  - IdC para $V^\pi(s_1)$: $\left[\bar{V}_\Omega^\pi - S_\Omega^\pi \frac{Z_{1-\frac{\alpha}{2}}}{\sqrt{m}}; \bar{V}_\Omega^\pi + S_\Omega^\pi \frac{Z_{1-\frac{\alpha}{2}}}{\sqrt{m}} \right]$

- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post-decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

Comparación entre de políticas

Poder comparar dos políticas permite construir heurísticas de búsqueda local.

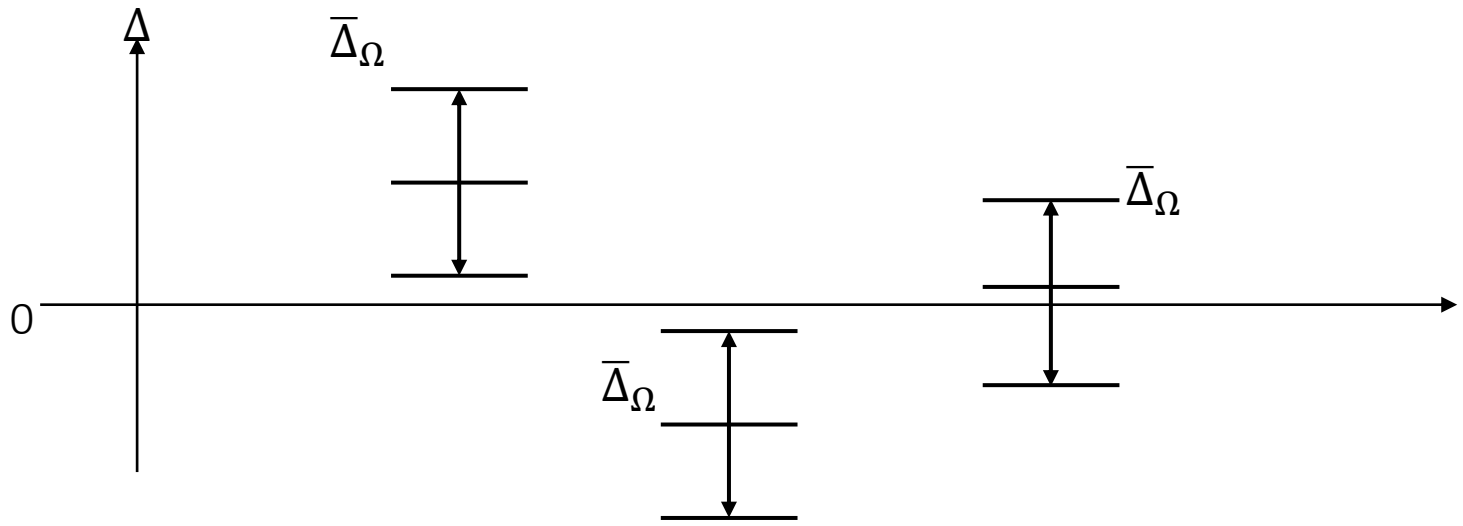
Deseamos comparar como es una política π^1 contra otra política π^1 iniciadas desde un estado inicial s.

En teoría π^1 es mejor que π^2 si $\Delta(s) := V_1^{\pi^1}(s) - V_1^{\pi^2}(s) > 0$

¿Cómo aproximar comparación de forma eficiente cuando se sufre de la maldición de dimensionalidad?

Comparación entre dos políticas

- Verificar mediante simulación si IdeC de $\overline{\Delta}_{\Omega}(s) = \frac{1}{m}\sum_{\omega\in\Omega}\left(V_{\omega}^{\pi^1} - V_{\omega}^{\pi^2}\right)$ está por sobre el 0.
    - Sincronizar realización de números aleatorios (reducción de varianza mediante variables antitéticas).



- La maldición de la dimensionalidad
- Approximate Dynamic Programming (ADP)
- El estado de post-decisión y función Q
- Resumen de técnicas de ADP
- Evaluación de una política
- Comparación de políticas
- Garantías de optimalidad

Relajación de Información perfecta:
Perfect Information Relaxation (PIR)

- Sobreestima el valor óptimo relajando totalmente el acceso a información futura de variables aleatorias.

- Hace trampa y ejecuta acciones 100% adaptadas al valor de las variables aleatorias en las etapas futuras $\{1, \dots, T\}$ antes de ejecutar el MDP.

- Una **PIR es super-óptima, pero infactible,** pues diseña decisión después de observas información.

- El valor de una PIR $V^{PIR}(s)$ dado un estado inicial s se define como el valor esperado del valor óptimo $V_\omega^{PIR}(s)$ sobre cada realización de la incertidumbre ω

$$V^{PIR} = \mathbb{E}_\omega(V_\omega^{PIR}) \geq V^*$$

En la práctica se estima mediante simulación computacional y se computa su IdC:

$$V^{PIR} \approx \frac{1}{m} \sum_{\omega \in \Omega} V_\omega^{PIR}$$

¿Por qué es una cota superior?

- Es **post-optimizar**, es decir, optimizar después de observar la realización de ω.
- **Relaja las leyes temporales de anticipación** de las soluciones óptima a la filtración de información futura.

Intuición en una etapa:

- Supongamos objetivo $f(x, \omega)$ a maximizar dependiente de decisión $x \in \mathbb{X}$ e incertidumbre $\omega \in \Omega$.
- Sea $f_\omega^* = \max_{x \in \mathbb{X}} f(x, \omega)$

Se cumple para todo $\omega \in \Omega$ y todo $x \in \mathbb{X}$ que:

$$f(x, \omega) \leq f_\omega^*$$

Tomando esperanzas, se cumple para todo$x \in \mathbb{X}$ que:

$$\mathbb{E}_\omega\big(f(x, \omega)\big) \leq \mathbb{E}_\omega(f_\omega^*) = f^{PIR}$$

Luego, el problema estocástico está acotado por la **PIR**:

$$\max_{x \in \mathbb{X}} \mathbb{E}_\omega\big(f(x, \omega)\big) \leq f^{\mathrm{PIR}}$$

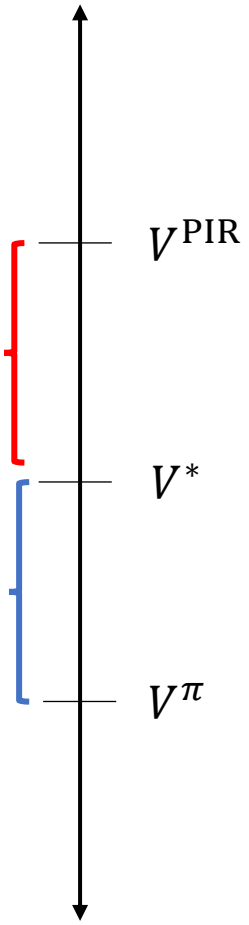
Garantía para una política π

- Buscamos estimar gap $V^* - V^\pi$ de una política heurística π al valor óptimo.

- Sabemos que:

$$V^* - V^\pi \leq V^{\text{PIR}} - V^\pi$$

  - $V^{PI} - V^\pi$: máxima distancia posible de política π al valor óptimo.
  - $V^* - V^\pi$: gap de optimalidad. Distancia recuperable de política π al valor óptimo del MDP.
  - $V^{\text{PIR}} - V^*$: gap de información incierta. Costo irrecuperable por tener que ejecutar decisiones dinámicas bajo incertidumbre.



Garantía para una política π

Para estimar $V^{\mathrm{PIR}} - V^{\pi}$, se debiese simular y estimar cantidad mediante promedio muestral:

$$\frac{1}{m}\sum_{\omega\in\Omega}\left(V_{\omega}^{\mathrm{PIR}} - V_{\omega}^{\pi}\right)$$

- Sincronizar números aleatorios.

- Notar que $V_{\omega}^{\mathrm{PIR}} \geq V_{\omega}^{\pi}$ para cada realización $\omega \in \Omega$ de la incertidumbre (¿Por qué?)

Relajación imperfecta de Información (IIR):

- Idea: Mejorar sobrestimación del valor óptimo relajando parcialmente el acceso a información futura.

- **Filtración normal de información:** en la etapa k se conoce la realización aleatoria hasta $t = k$.

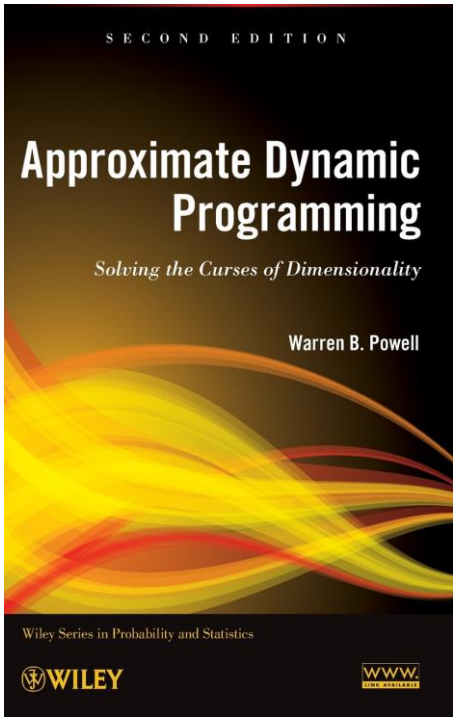
- **Ejemplos de relajación de información:**
  - Relajación Perfecta: Conoce hasta T al comienzo del horizonte.
  - Relajación Imperfecta de dos etapas: Conoce información inicial en $t = 1$, pero en $t = 2$ observa todo hasta la etapa T.
  - Relajación Imperfecta de tres etapas: Respeta la filtración de información hasta $t = 2$, luego se revela todo el futuro en $t = 3$.
  - Relajación de a dos, tres, cuatro periodos, etc….

- Para toda relajación de información IIR se cumple que:

$$V^* \leq V^{\mathrm{IIR}} \leq V^{\mathrm{PIR}}$$

Cuarta Parte:
Clase 1 – Introducción a Approximate Dynamic Programming

Optimización Dinámica - ICS



Mathias Klapp