



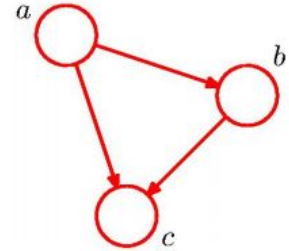
Probabilistic Graphical Models

aoteo@grupobme.es

¿Qué vamos a ver en la sesión de grafos probabilísticos?

- Bayesian Networks
- Markov Models
- Introducción a los Markov Decision Process

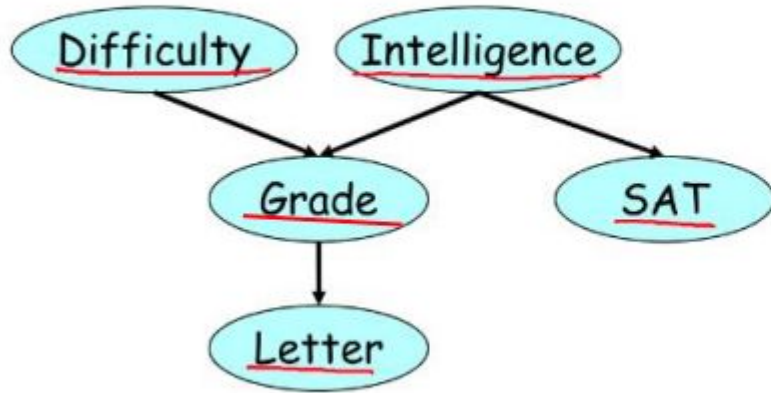
- Los **Grafos Probabilísticos** son representaciones **esquemáticas** de **distribuciones de probabilidad**.
- Estas representaciones esquemáticas se realizan a través de grafos. Estos grafos son estructuras formadas por **nodos** y **arcos**.
- Los grafos pueden ser **dirigidos** o **no dirigidos** haciendo referencia principalmente a las **Redes Bayesianas** y **Redes de Markov** respectivamente.



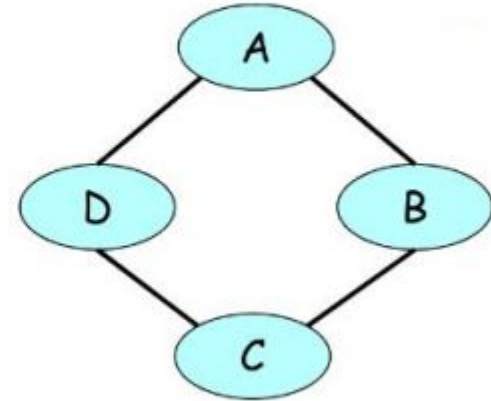
¿Por qué utilizar los modelos de grafos probabilísticos?

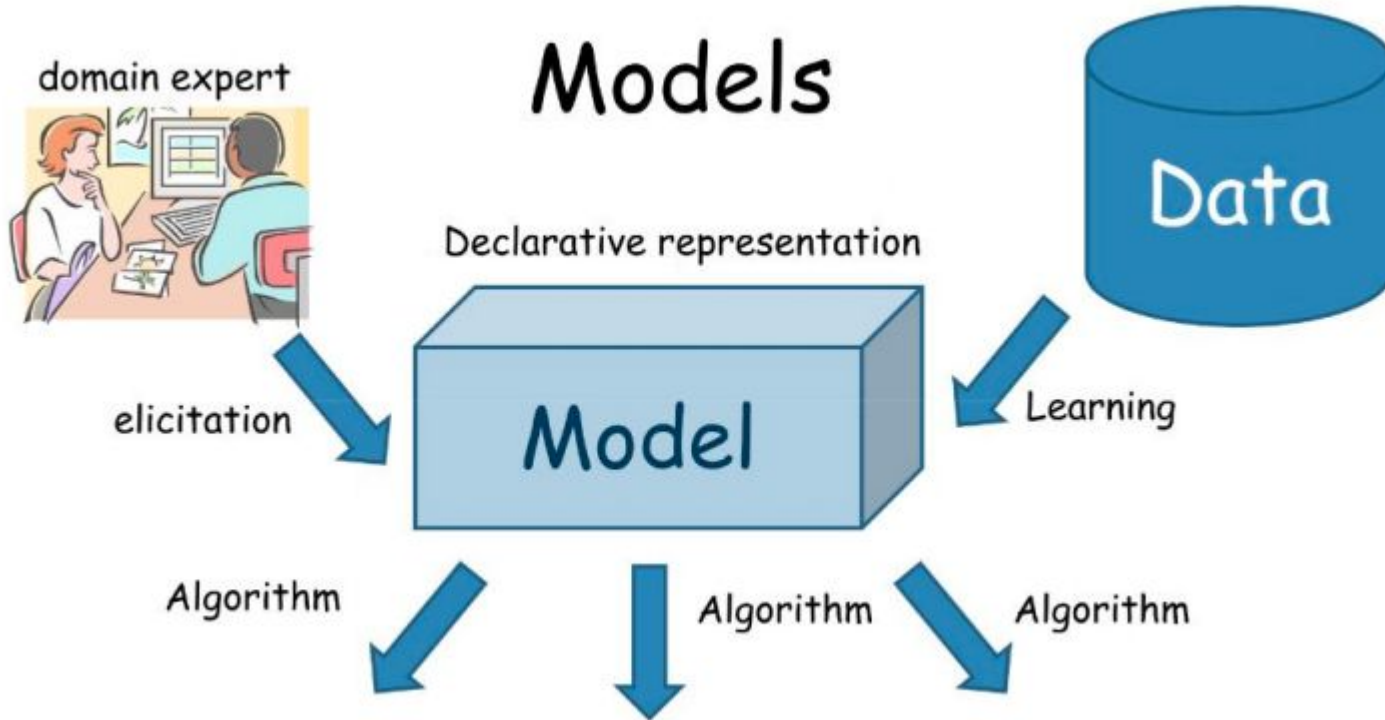
- Estos no dan una estructura de los datos **intuitiva** y **compacta** para capturar las distribuciones de probabilidad en todas las dimensiones del problema.
- Al mismo tiempo nos ofrece un conjunto de métodos para el **razonamiento eficiente** con algoritmos de propósito general que explotan la estructura de grafo.

Bayesian networks



Markov networks





- Representación:
 - ¿Cómo capturar/modelizar la incertidumbre de nuestros datos?
 - ¿Cómo codificar nuestro conocimiento/asumpciones/restricciones?
- Inferencia:
 - ¿Cómo responder a queries dados unos datos?
- Learning:
 - ¿Qué estructura es la más adecuada dados unos datos?

Recapitulación de conceptos básicos: Distribución conjunta

- **Intelligence (I)**
 - i^0 (low), i^1 (high),
- **Difficulty (D)**
 - d^0 (easy), d^1 (hard)
- **Grade (G)**
 - g^1 (A), g^2 (B), g^3 (C)

- ¿Qué es la distribución de probabilidad conjunta sobre múltiples variables?

$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

- ¿Cuántas configuraciones posibles? $\rightarrow 2^8$
- ¿Es necesario representar todas?
- ¿Necesitamos un conocimiento a priori?

Recapitulación de conceptos básicos: Distribución conjunta

- **Intelligence (I)**
 - i^0 (low), i^1 (high),
- **Difficulty (D)**
 - d^0 (easy), d^1 (hard)
- **Grade (G)**
 - g^1 (A), g^2 (B), g^3 (C)



I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024

- Dados dos o más eventos aleatorios, la distribución conjunta de estos es la distribución de probabilidad de que estos eventos ocurran juntos.

Recapitulación de conceptos básicos: Distribución conjunta

- **Intelligence (I)**
 - i^0 (low), i^1 (high),
- **Difficulty (D)**
 - d^0 (easy), d^1 (hard)
- **Grade (G)**
 - g^1 (A), g^2 (B), g^3 (C)



I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024

- Dados dos o más eventos aleatorios, la distribución conjunta de estos es la distribución de probabilidad de que estos eventos ocurran juntos.

¿ QUÉ OCURRE SI LAS VARIABLES ALEATORIAS SON INDEPENDIENTES?

- Condición sobre g_1 .

I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024

- Condición sobre g_1 .

I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024



I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^1	g^1	0.009
i^1	d^0	g^1	0.252
i^1	d^1	g^1	0.06

- Normalización de las probabilidades

I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^1	g^1	0.009
i^1	d^0	g^1	0.252
i^1	d^1	g^1	0.06

$P(I, D, g^1)$



- Normalización de las probabilidades

I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^1	g^1	0.009
i^1	d^0	g^1	0.252
i^1	d^1	g^1	0.06

$P(I, D, g^1)$

0.447



I	D	Prob.
i^0	d^0	0.282
i^0	d^1	0.02
i^1	d^0	0.564
i^1	d^1	0.134

$P(I, D | g^1)$

- Normalización de las probabilidades

I	D	Prob.
i^0	d^0	0.282
i^0	d^1	0.02
i^1	d^0	0.564
i^1	d^1	0.134

$P(I, D | g^1)$

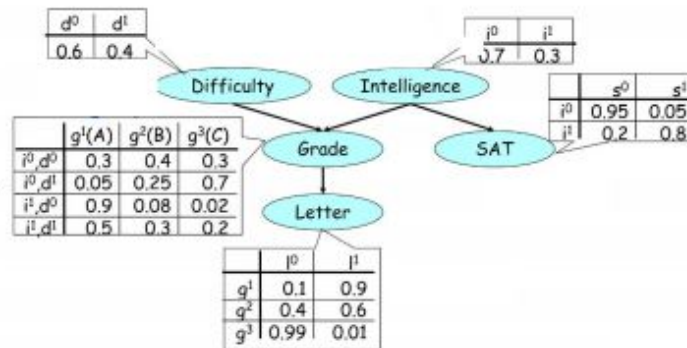


D	Prob.
d^0	0.846
d^1	0.154



Redes Bayesianas

- Las **redes bayesianas** son un tipo de grafos probabilísticos que utilizan inferencia bayesiana para el cálculo de las probabilidades.
- Su principal objetivo es representar las dependencias condicionales de las variables a través de un grafo acíclico dirigido (DAG). Por ejemplo, es capaz de representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas o algunos de ellos la red puede ser utilizada para computar la probabilidad de la presencia de la enfermedad.
- Formalmente es una distribución de probabilidad conjunta de un conjunto de variables aleatorias en el que el conjunto de independencias puede representarse utilizando un **grafo acíclico dirigido**.
- En este tipo de redes se pueden identificar 3 tipos de inferencias:
 - **Deducción de las variables no observadas.**
 - **Aprendizaje de parámetros.**
 - **Aprendizaje de estructuras.**



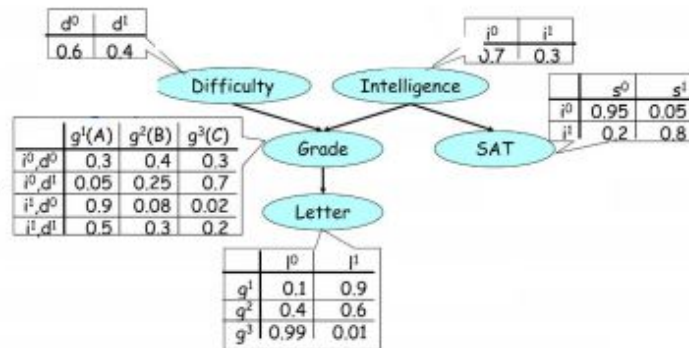
$$P(D, I, G, S, L) = P(D) P(I) P(G|I, D) P(S|I) P(L|G)$$

- Las **redes bayesianas** son un tipo de grafos probabilísticos que utilizan inferencia bayesiana para el cálculo de las probabilidades.
- Su principal objetivo es representar las dependencias condicionales de las variables a través de un grafo acíclico dirigido (DAG). Por ejemplo, es capaz de representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas o algunos de ellos la red puede ser utilizada para computar la probabilidad de la presencia de la enfermedad.
- Formalmente es una distribución de probabilidad conjunta de un conjunto de variables aleatorias en el que el conjunto de independencias puede representarse utilizando un **grafo acíclico dirigido**.
- En este tipo de redes se pueden identificar 3 tipos de inferencias:

- **Deducción de las variables no observadas.**
- **Aprendizaje de parámetros.**
- **Aprendizaje de estructuras.**

¿Es correcta esta descomposición de la FPC?

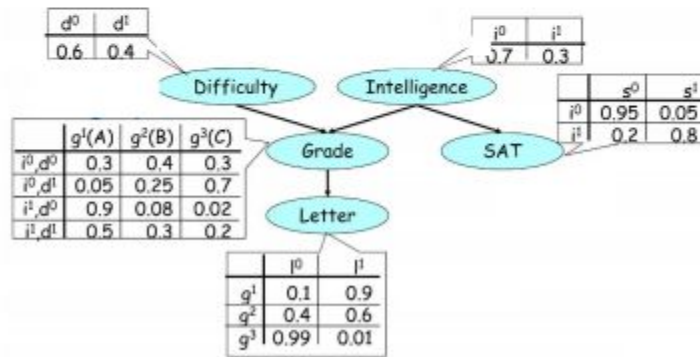
$$P(D, I, G, S, L) = P(D) P(I) P(G|I, D) P(S|I) P(L|G)$$



- Las **redes bayesianas** son un tipo de grafos probabilísticos que utilizan inferencia bayesiana para el cálculo de las probabilidades.
- Su principal objetivo es representar las dependencias condicionales de las variables a través de un grafo acíclico dirigido (DAG). Por ejemplo, es capaz de representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas o algunos de ellos la red puede ser utilizada para computar la probabilidad de la presencia de la enfermedad.
- Formalmente es una distribución de probabilidad conjunta de un conjunto de variables aleatorias en el que el conjunto de independencias puede representarse utilizando un **grafo acíclico dirigido**.
- Las BN representa la función de probabilidad conjunta mediante la regla de la cadena para redes bayesianas:

$$P(X_1, \dots, X_n) = \prod_i P(X_i | Par_G(X_i))$$

- En este tipo de redes se pueden identificar 3 tipos de inferencias:
 - Deducción de las variables no observadas.**
 - Aprendizaje de parámetros.**
 - Aprendizaje de estructuras.**

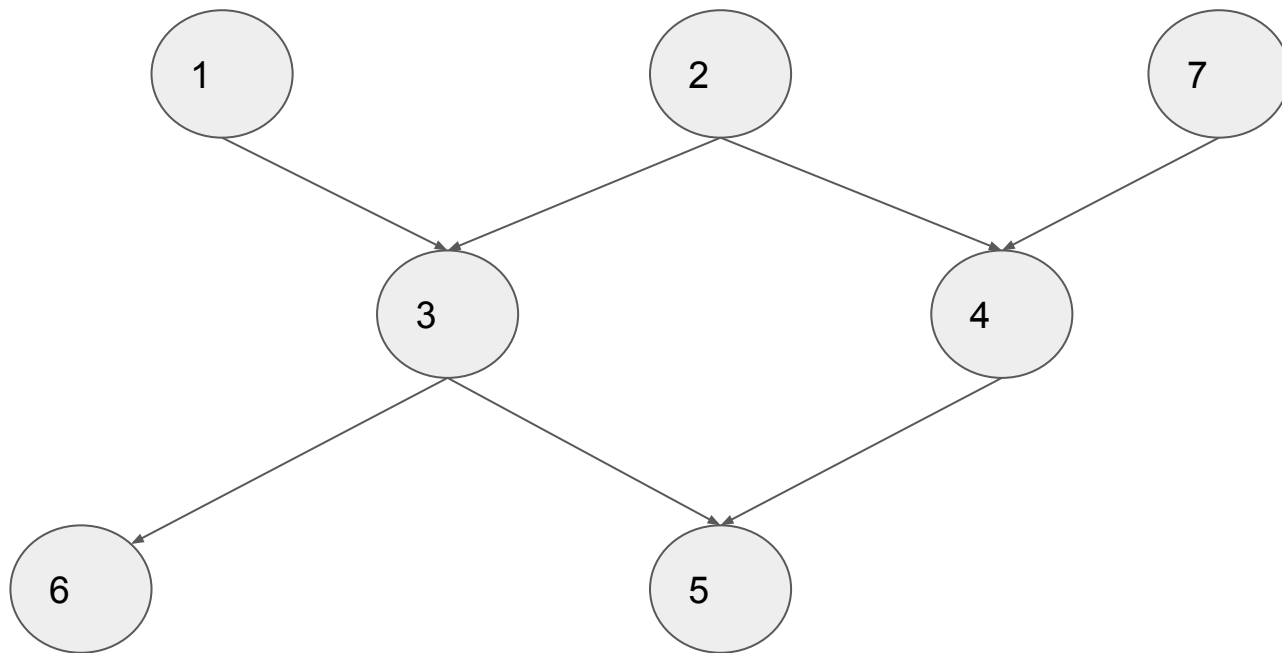


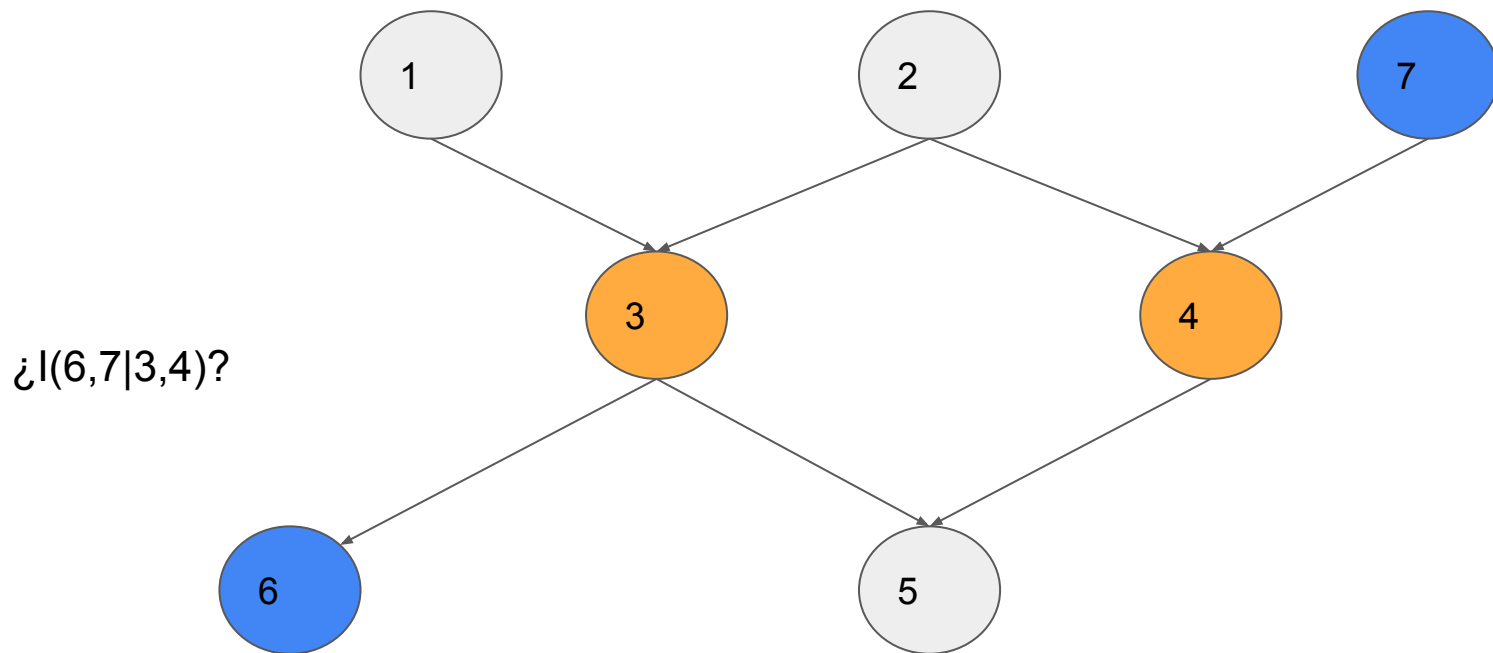
17 Instituto BME Redes Bayesianas: Como modelo de dependencia

- Modelo de dependencia definido gráficamente: Criterio de separación gráfica (d-separación)
- Nos interesan los modelos de dependencia ya que si un modelo de dependencia de un DAG (por tanto de su red bayesiana) es compatible con p entonces podemos representar de manera exacta la distribución de probabilidad con la red bayesiana.
- Si p factoriza conforme a G , es decir si una distribución se puede representar exactamente como un producto de probabilidades condicionadas que están representadas por el grafo G entonces todas las independencias del grafo G las verifica la distribución de probabilidad p $I_G \rightarrow I_p$

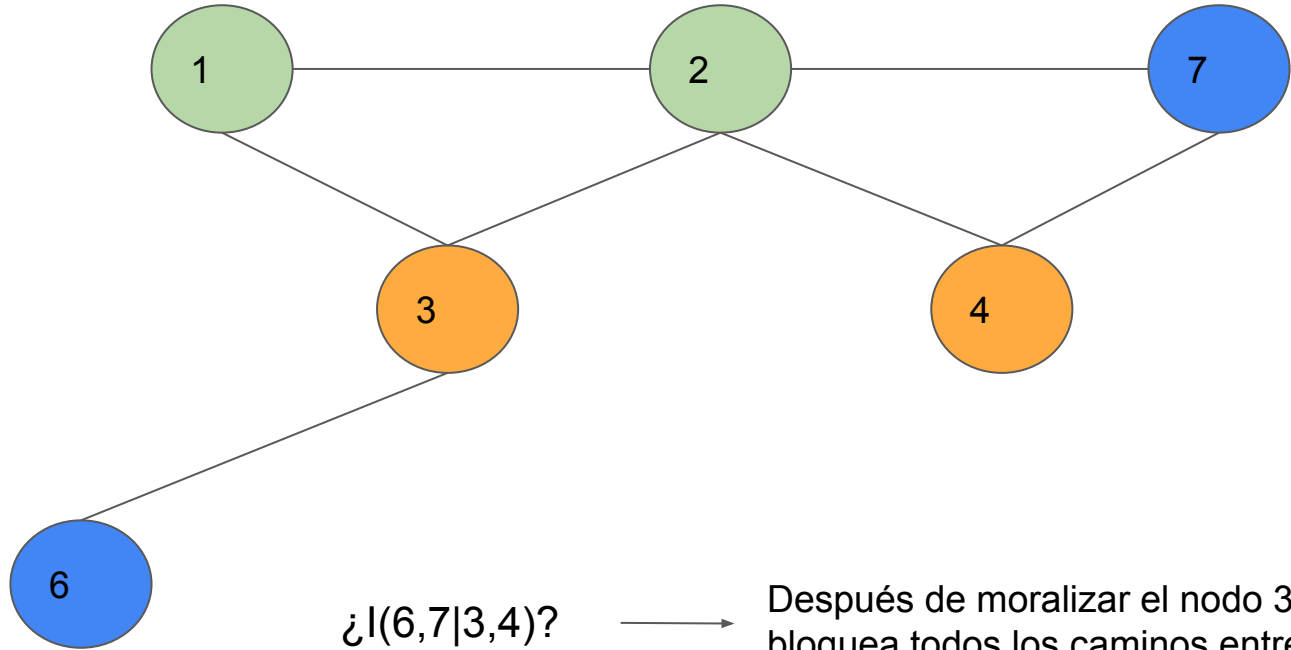
D-separación

Las variables X_a son independientes de X_b condicionadas a X_c si las variables X_c bloquean todos los caminos del subgrafo moral del menor subconjunto ancestral





Redes Bayesianas: Como modelo de dependencia



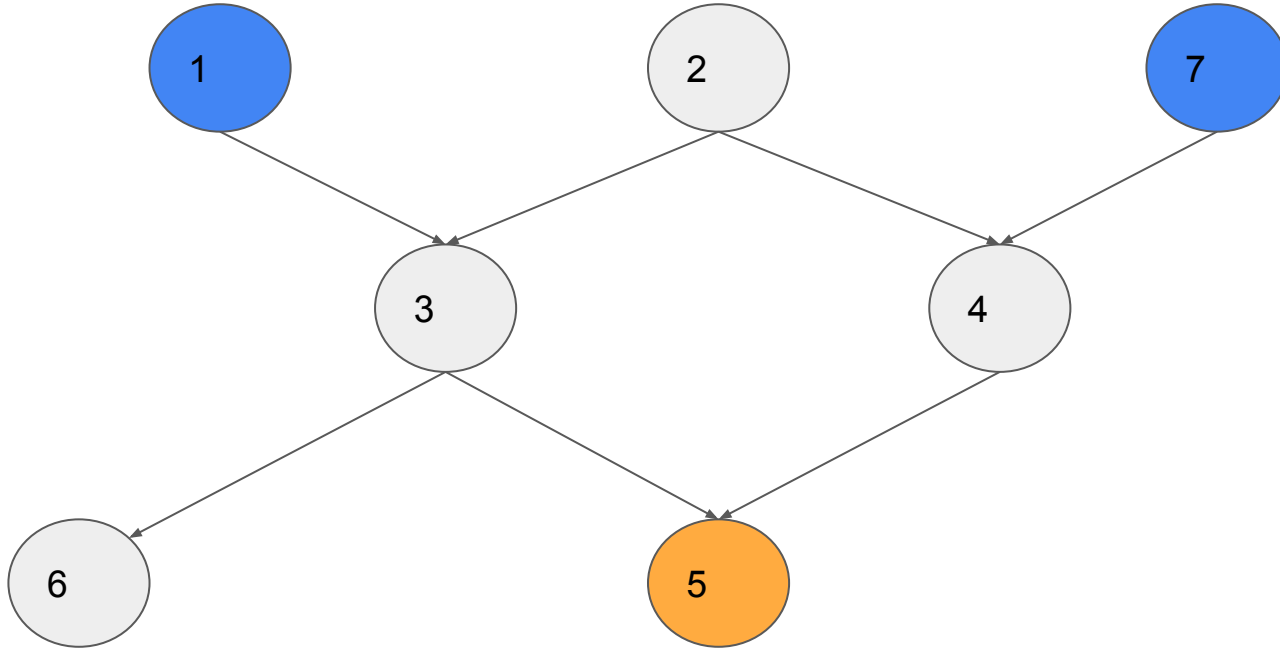
$\text{¿}I(6,7|3,4)\text{?}$

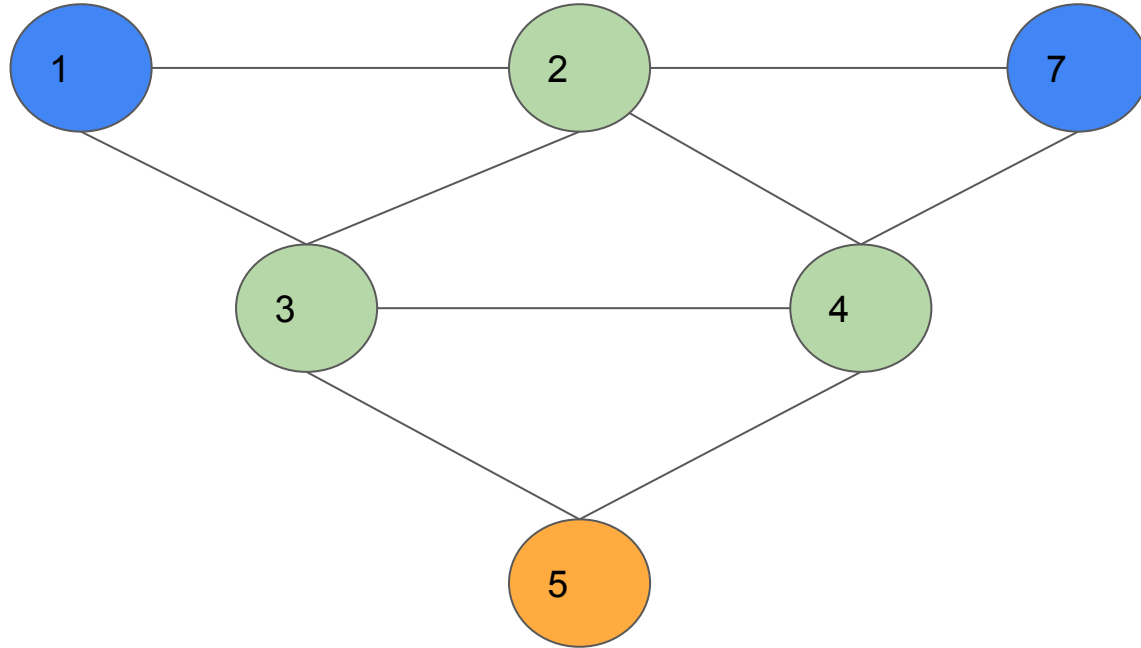


Después de moralizar el nodo 3
bloquea todos los caminos entre
7 y 6

Redes Bayesianas: Como modelo de dependencia

$\mathbb{I}(1,7|5)?$





¿ $I(1,7|5)$?



Existen múltiples
caminos entre 1 y 7

23

Instituto BME

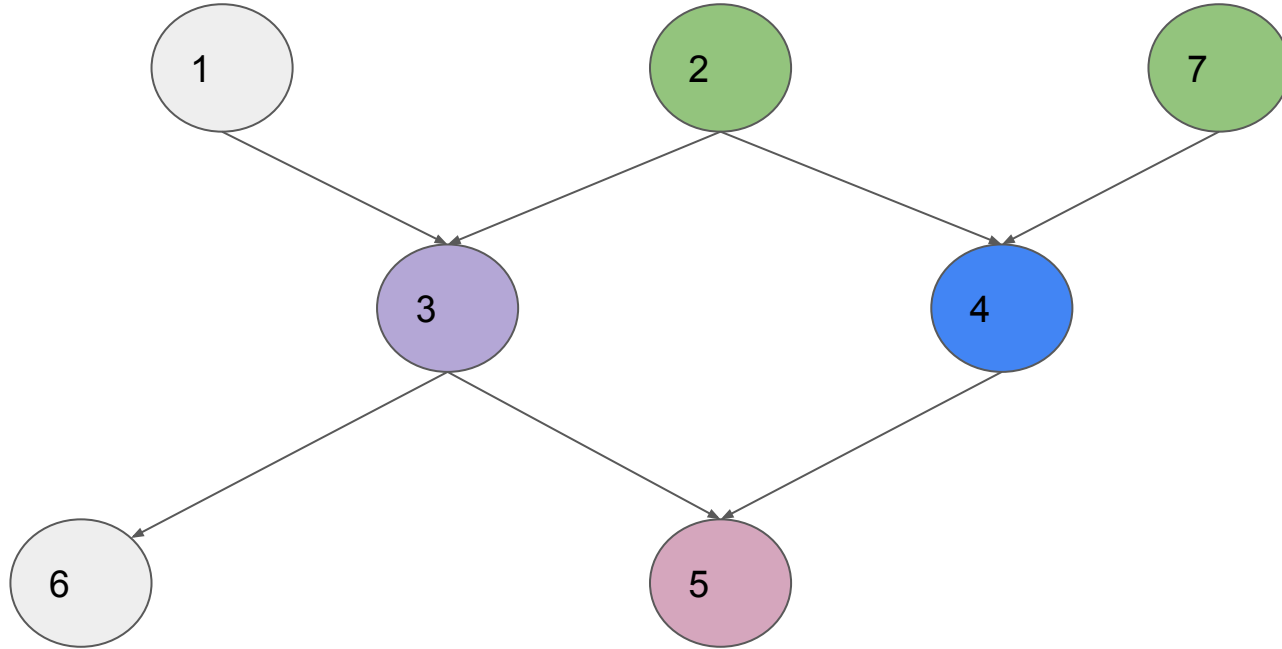
Redes Bayesianas: Como modelo de dependencia

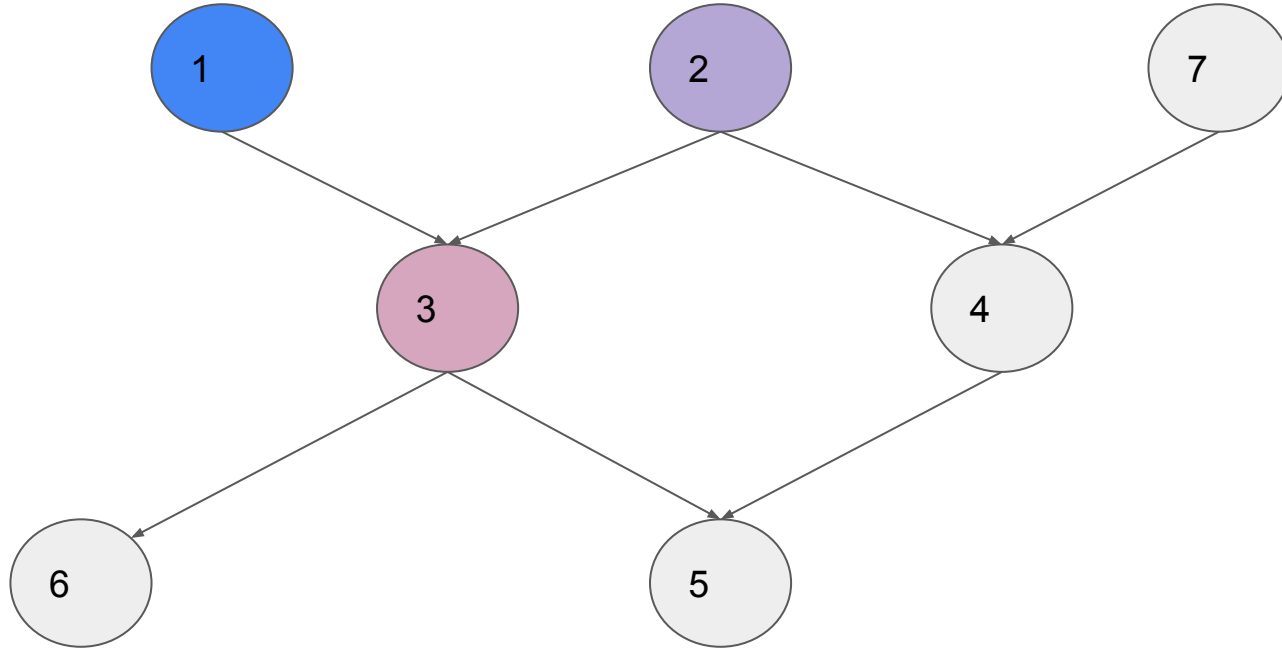
Manto de Markov

Un manto de Markov es el subgrafo definido para un nodo formado por los Padres, hijos y padres de los hijos

- $Mb(X_j)$ bloquea todos los caminos a X_j .
- Tiene importantes aplicaciones en el ámbito del aprendizaje de las redes bayesianas, en el aprendizaje de la estructura de las redes bayesianas o en la clasificación supervisada por parte de los modelos.

Redes Bayesianas: Manto de Markov Ejemplo I



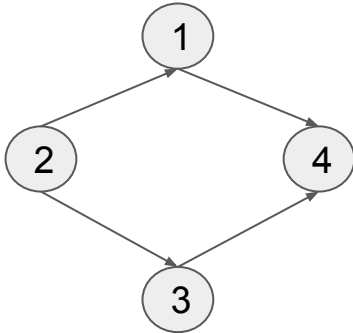


Mapa perfecto

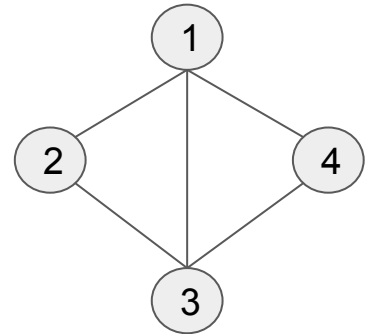
Modelo de dependencia (definido gráficamente) para el que $I_G \equiv I_p$, es decir, todas las independencias de la distribución de probabilidad se pueden leer en el grafo

- No siempre es posible construir modelos de dependencia perfectos. Por ejemplo:

$$I_p = \{i(1, 3|2, 3, 4); i(2, 3|1, 3)\}$$



Moralización de uno de los grafos más cercanos al modelo de independencia



Mapa de independencia

Modelo de dependencia (definido gráficamente) para el que $I_G \subseteq I_p$, es decir, todas las independencias del modelo de dependencia del grafo se pueden encontrar en la distribución.

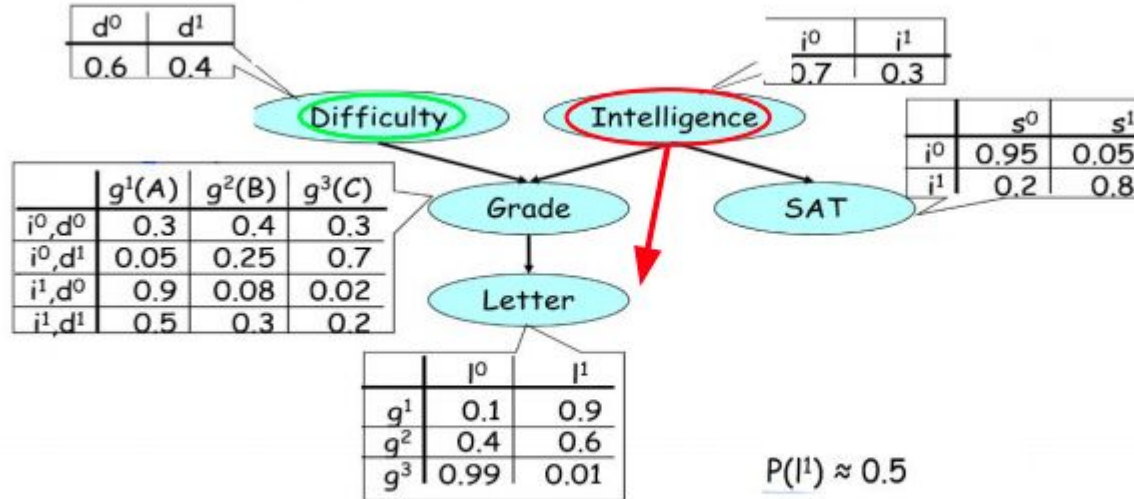
- Son todas las que están pero no todas las que son.
- Los I-mapas son modelos compatibles con p definidos gráficamente.
- El I-mapa trivial es el grafo completo: Demasiados parámetros (complejidad exponencial)

I-mapa minimal

I_G es un I-mapa minimal de I_p si es un I-mapa y eliminar un arco de G hace que pierda la condición de ser un I-mapa.

- Reflejan el máximo número de independencias de p
- Interesa construir grafos que sean I-mapas minimales: mínimo número de parámetros que permiten modelar p de forma exacta.
- Con las redes bayesianas vamos a poder construir I-mapas minimales de cualquier distribución de probabilidad.

Redes Bayesianas: Razonamiento Causal

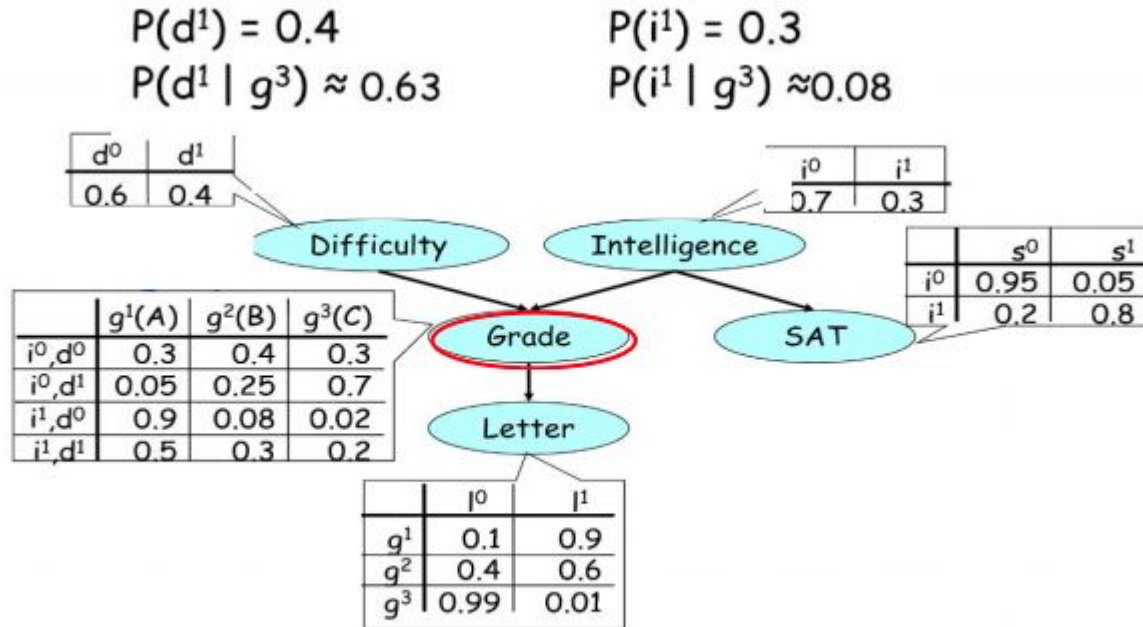


$$P(l^1) \approx 0.5$$

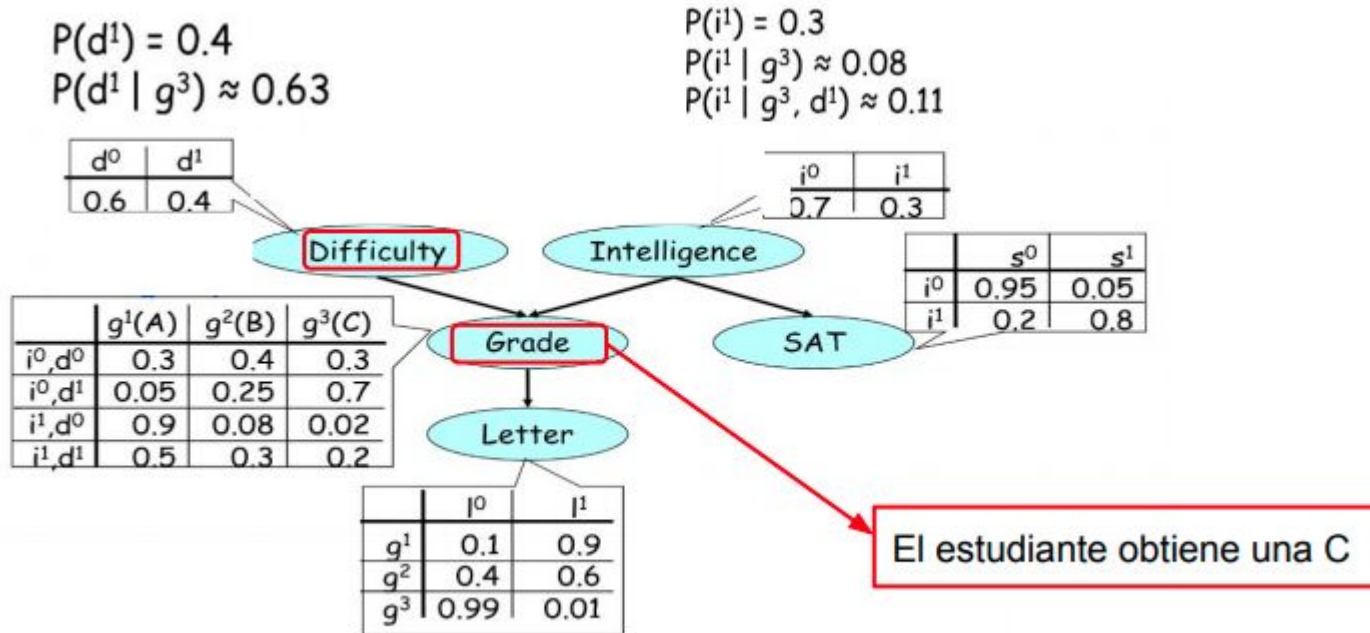
$$P(l^1 \mid i^0) \approx 0.39$$

$$P(l^1 \mid i^0, d^0) \approx 0.51$$

Redes Bayesianas: Razonamiento Evidencial



Redes Bayesianas: Razonamiento Intercausal





Redes Bayesianas

Estimación de los parámetros

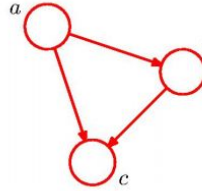
Objetivo

Aproximar p mediante una red Bayesiana aprendida a partir de un conjunto de datos D independiente e idénticamente distribuido (i.i.d) conforme a p .

- Maximizar la generalización.
- solo conocemos el ajuste.
- Hacer que el ajuste sea un buen estimador de la generalización.
- Encontrar un equilibrio entre el número de parámetros y de casos disponibles.

Redes Bayesianas: Estimación de parámetros

- Dada la estructura de la red Bayesiana: **Nodos, arcos y dependencias.**



- Dado un data set completo: **Datos con las variables mencionadas.**

X_1	X_2	X_3	X_4	X_5
0	0	1	1	0
1	0	0	1	0
0	1	0	0	1
0	0	1	1	1
\vdots	\vdots	\vdots	\vdots	\vdots

La estimación de los parámetros puede ser una tarea más difícil para un experto que la estimación de la arquitectura.

- Estimamos las distribuciones de probabilidad condicional: **¿Cuales son las CPD?**

$$P(X_1), P(X_2), P(X_3|X_1, X_2), P(X_4|X_1), P(X_5|X_1, X_3, X_4)$$

Existen diferentes aproximaciones para la estimación de los parámetros.

- Métodos basados en el **Maximum Likelihood Estimation**.
 - Uso de datos observados para estimar los parámetros de una distribución dada.
 - Por ejemplo, si tenemos una distribución normal de media y varianza desconocidas.
- Métodos basados en perspectivas **Bayesianas**.
 - Uso de un conocimiento a priori sobre las distribuciones que buscamos estimar.
 - Este conocimiento a priori viene de una distribución de probabilidad conocida.

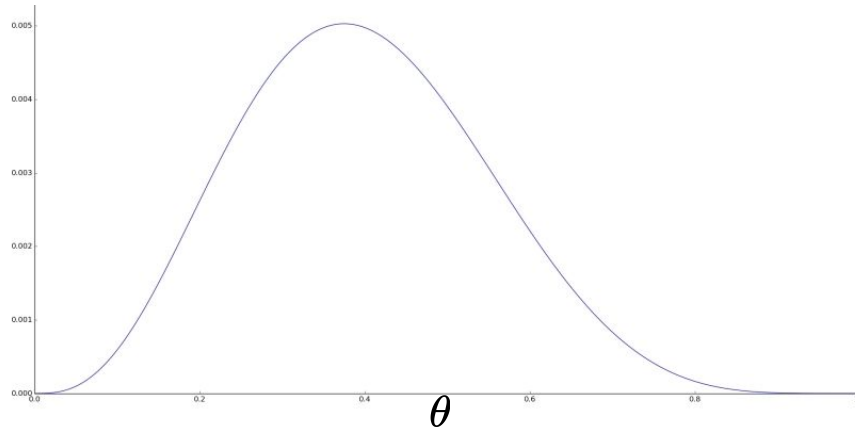
Redes Bayesianas: Maximum Likelihood Estimation Example I

- Asumimos que tenemos un conjunto de puntos $D = \{x_1, x_2, \dots, x_m\}$ que viene de 1000 tiradas de una moneda conde hemos obtenido $H=330$ y $T=670$.
- Basado en estas observaciones podemos definir un parámetro θ que represente la probabilidad de obtener HEAD ó TAIL. Por tanto, podemos considerar $P(H)=0.33$

- Ahora asumamos un conjunto $D = \{T, H, H, T, T, T, H, T\}$

$$P(D|\theta) = (1 - \theta)\theta\theta(1 - \theta)(1 - \theta)\theta(1 - \theta)$$

- Esta es la probabilidad de nuestros datos formada por nuestro parámetro θ , conocido como el likelihood.



- Podemos formalizar nuestro ejemplo de manera que:

$$P(D|\theta) = \theta^{M_h} (1 - \theta)^{M_t}$$

- Para resolver el problema podemos buscar una forma cerrada. Para ello utilizamos el likelihood. Aun así, es mucho más fácil trabajar con el log-likelihood:

$$\log(P(D|\theta)) = M_h \log \theta + M_t \log(1 - \theta)$$

- Para maximizar, derivamos e igualamos a 0 obteniendo los siguientes resultados:

$$\hat{\theta} = \frac{M_H}{M_H + M_T}$$

Redes Bayesianas: Estimación Bayesiana de Parámetros

- Los métodos MLE son plausibles pero pueden llegar a ser muy simplistas en cuanto a que asumen que toda la información está en los datos y sus cálculos se realizan únicamente sobre estas distribuciones.
- Los métodos bayesianos introducen un conocimiento referido como **Prior**. No buscan que los priors sean una guía absoluta para el modelo pero sí que supongan un punto razonable de partida.
- Se crean distribuciones de probabilidad representando nuestro conocimiento a priori.

Así pues, los métodos bayesianos de estimación de parámetros incluyen ambas características, el análisis de los datos y el conocimiento *a priori* (prior) acerca de los parámetros.

- Una de las críticas principales a los modelos bayesianos es la inclusión de este conocimiento prior y como se obtiene.

Redes Bayesianas: Estimación Bayesiana de Parámetros

- Recogiendo el ejemplo anterior del lanzamiento de una moneda, digamos que tenemos una probabilidad a priori $P(\theta)$: Recordamos también como definimos nuestro likelihood:

$$P(x[m]|\theta) = \begin{cases} \theta & \text{if } x[m] = x^1 \\ (1 - \theta) & \text{if } x[m] = x^0 \end{cases}$$

- Podemos utilizarlo por tanto para definir la distribución de probabilidad conjunta sobre los datos D y nuestro parámetro:

$$\begin{aligned} P(x[1], x[2], \dots, x[m], \theta) &= P(x[1], \dots, x[m]|\theta)P(\theta) \\ &= P(\theta)\theta^{M[1]}(1 - \theta)^{M[0]} \end{aligned}$$

- Utilizando la ecuación anterior podemos calcular la distribución de probabilidad a posteriori sobre el parámetro.

$$P(\theta | x[1], x[2], \dots, x[M]) = \frac{P(x[1], x[2], \dots, x[M] | \theta) P(\theta)}{P(x[1], x[2], \dots, x[M])} \longrightarrow \text{PRIOR}$$



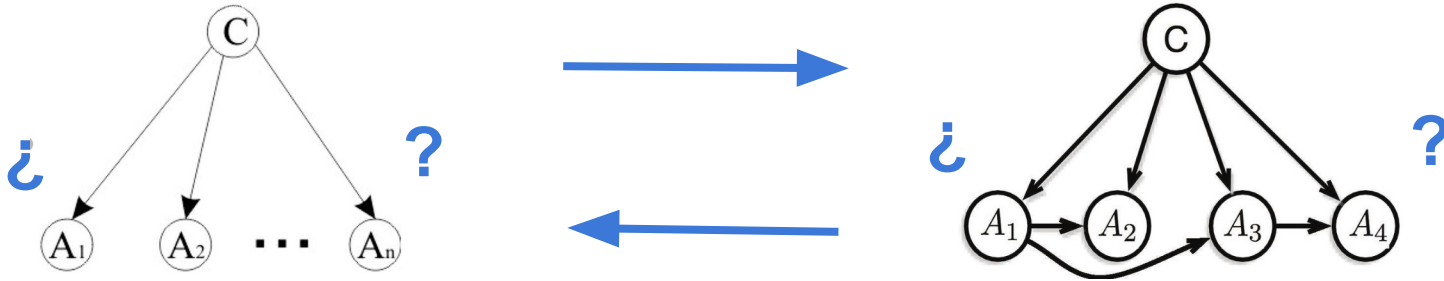
Redes Bayesianas

Estimación de la estructura

- Construir la estructura del modelo a partir de los datos puede ser una tarea muy difícil.
- Generalmente debemos realizar asunciones sobre las dependencias del modelo que dependen principalmente de nuestra tarea de aprendizaje.
 - Aprender las dependencias de las variables a partir del knowledge discovery.
 - Modelos de estimación de densidad. Basados en la predicción sobre nuevos valores en los datos.

Redes Bayesianas: Score-based

- Aunque la estimación de la estructura de una red bayesiana pueda ser una tarea asequible para un experto, esta también puede ser estimada mediante diferentes métodos.
 - Score-based methods: Hill Climbing, Tabu Search.
 - Constraint-based approach: Grow-Shrink, Semi-Interleaved Hiton-PC, etc



Redes Bayesianas: Estimación de la Estructura

- Aunque la estimación de la estructura de una red bayesiana pueda ser una tarea asequible para un experto, esta también puede ser estimada mediante diferentes métodos.
 - Scored-based methods:
 - Consideramos la red bayesiana como modelo estadístico. Luego definimos un espacio de hipótesis de posibles estructuras y una función de puntuación que nos dice qué tan cerca está nuestra estructura de la estructura subyacente. Como este método de aprendizaje considera el modelo completo a la vez, puede dar mejores resultados que el aprendizaje basado en restricciones. El problema con este modelo es que, dado que nuestro espacio de hipótesis puede ser muy grande, es difícil encontrar la estructura más óptima. Por lo tanto, generalmente recurrimos a técnicas de búsqueda heurísticas.
 - Constraint-based approach:
 - Basado en restricciones, funciona sobre la base de considerar una red bayesiana como un conjunto de condiciones de dependencia entre las variables aleatorias. Intentamos encontrar las condiciones de dependencia a partir de los datos que se nos proporcionan. Usando estas condiciones, luego intentamos construir una red.
 - Bayesian model averaging:
 - En este método, intentamos aplicar conceptos similares a los que vimos en secciones anteriores para aprender muchas estructuras, y luego usamos un conjunto de todas estas estructuras. Como la cantidad de estructuras de red puede ser enorme, a veces tenemos que usar algunos métodos aproximados para hacer esto.

Redes Bayesianas: Estimación de la Estructura -Scored based

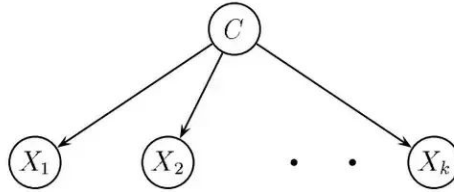
- Primero se define un criterio para evaluar como de bien se ajusta una red Bayesiana a los datos. Entonces se realiza una búsqueda por el espacio de DAGs para encontrar la estructura que obtiene el máximo score. Es esencialmente un problema de búsqueda que consiste en 2 partes:
 - Definición de la métrica
 - Definición del algoritmo de búsqueda
- Las funciones de scoring usualmente se miden como:

$$Score(G : D) = LL(G : D) - \phi(|D|)\|G\|$$

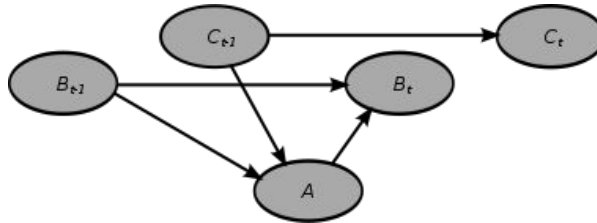
- Cuando phi es 1 tenemos el Akaike Information criterion, cuando phi es $\log(|D|)/2$ es el Bayesian Information Criterion.
- Un ejemplo de estos algoritmos es el Chow-Liu Algorithm.

- La elección más habitual en estos algoritmos son los de búsqueda local y los de búsqueda greedy.
- Para los de búsqueda local el algoritmo empieza con un grafo vacío o completo y en cada paso intenta cambiar el grafo con una única operación, ya sea añadiendo un eje, quitándolo o cambiando la dirección del mismo (Dichas operaciones han de preservar las propiedades del DAG). Si el score aumenta entonces adopta el cambio y lo realiza, si no, realiza otra búsqueda.
- Aunque estas aproximaciones son computacionalmente tratables, ninguna de ellas tiene la garantía de acabar en un grafo óptimo o incluso correcto. El espacio de grafos es altamente “no-convexo y los algoritmos pueden quedarse en regiones sub-óptimas del espacio.

- Existen casos especiales de Redes Bayesianas con un uso extenso en problemas de la vida real.
 - Modelo Naive Bayes Network



- Modelo Dynamic Bayesian Network



Redes Bayesianas: Naive Bayes Model

- Un clasificador bayesiano es un clasificador probabilístico que utiliza el teorema de Bayes para predecir una clase. Sea c una clase y un conjunto de características $X = \{x_1, x_2, \dots, x_n\}$. Entonces, la probabilidad de que las características pertenezcan a la clase c puede calcularse utilizando el teorema de Bayes como sigue:

$$P(c|X) = \frac{P(c) \cdot P(X|c)}{P(X)}$$

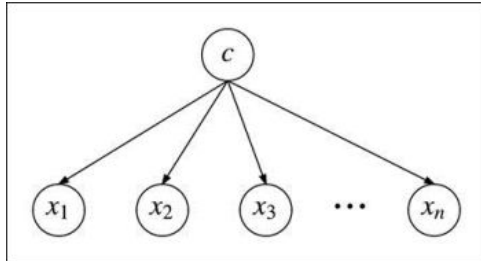
$$\hat{c} = \arg \max_{c \in C} P(c|X)$$

$$= \arg \max_{c \in C} \frac{P(c) \cdot P(X|c)}{P(X)}$$

$$= \arg \max_{c \in C} P(c) \cdot P(X|c)$$

$$P(X|c) = P(x_1|c) \prod_{i=2}^n P(x_i|x_{i-1}, \dots, x_1, c)$$

- El modelo Naive Bayes es uno de los algoritmos de aprendizaje más eficientes y eficaces, aunque es demasiado simplista, este modelo ha funcionado bastante bien.



$$\hat{c} = \arg \max_{c \in C} P(c|X)$$

$$= \arg \max_{c \in C} \frac{P(c) \cdot P(X|c)}{P(X)}$$

$$= \arg \max_{c \in C} P(c) \cdot P(X|c)$$

$$P(X|c) = \prod_{i=1}^n P(x_i|c)$$

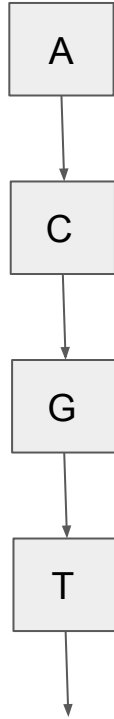
A person's hand is pointing at a tablet screen. The screen displays a bar chart and a network diagram. The entire image is overlaid with a blue tint. The text 'Redes de Markov' is centered in white.

Redes de Markov

- Un modelo de markov es un modelo en un espacio de estados estocásticos que involucra transiciones aleatorias entre estados donde la probabilidad de saltar de uno a otro solo depende del estado actual más allá de los estados anteriores.
- Estos modelos tienen la conocida como propiedad de markov.
- Estos modelos no tienen memoria.

Este tipo de modelos puede categorizarse en 4 clases dependiendo de si son autónomos o controlados y si la información es total o parcialmente observable.

	Fully Observable	Partially Observable
Autonomous	Markov Chain ^[5]	Hidden Markov Model ^[2]
Controlled	Markov Decision Process ^[3]	Partially Observable Markov Decision Process ^[4]



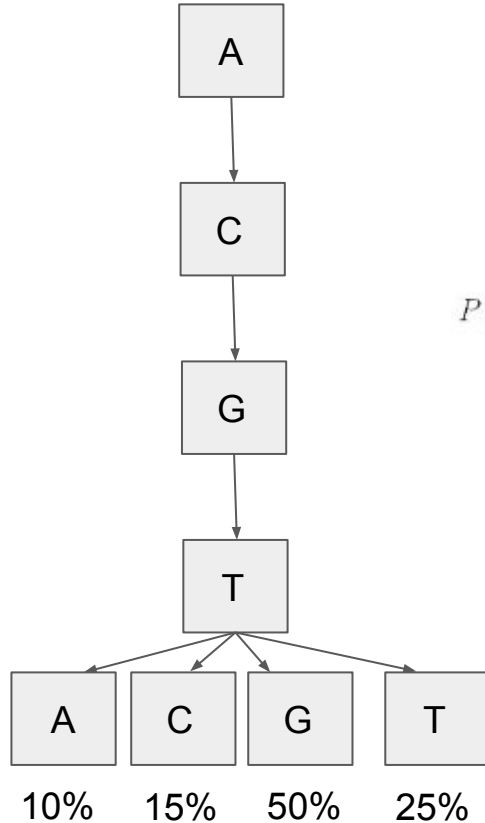
- Cada fila de nuestro dataset es dependiente y solo dependiente de la anterior. Cumplen la propiedad conocida como Propiedad de Markov.

$$P\{X_{t+1} = j \mid X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}, X_t = i\} = P\{X_{t+1} = j \mid X_t = i\},$$

for $t = 0, 1, \dots$ and every sequence $i, j, k_0, k_1, \dots, k_{t-1}$.

- No tenemos features. Solo tenemos la clase que queremos predecir.

NOTA: La propiedad de Markov restricción matemática de los modelos. Asume que cada fila solo depende de la fila inmediatamente anterior.



- Cada fila de nuestro dataset es dependiente y solo dependiente de la anterior. Cumplen la propiedad conocida como Propiedad de Markov.

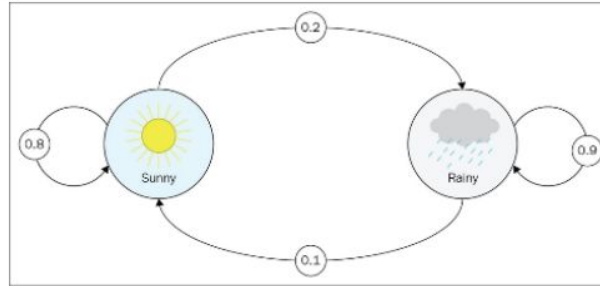
$$P\{X_{t+1} = j \mid X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}, X_t = i\} = P\{X_{t+1} = j \mid X_t = i\},$$

for $t = 0, 1, \dots$ and every sequence $i, j, k_0, k_1, \dots, k_{t-1}$.

- No tenemos features.

Modelos de Markov: Modelos de Markov Observables

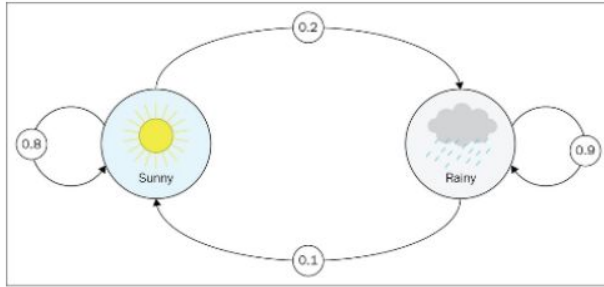
- El modelo más sencillo es la Cadena de Markov (Autónoma y totalmente observable)
- Los eventos climáticos son un buen ejemplo de cadena de markov. Ningún agente puede interactuar con ellos para modificarlos y son totalmente observables.







- Un algoritmo conocido es el Markov Chain Monte Carlo (MCMC) utilizado para obtener un muestreo a partir de la matriz de transición entre estados.

Modelos de Markov: Modelos de Markov Observables

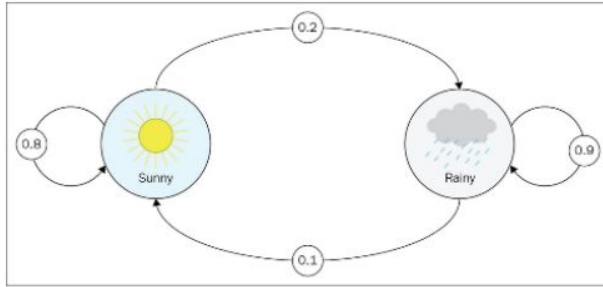
- El modelo más sencillo es la Cadena de Markov (Autónoma y totalmente observable)
- Los eventos climáticos son un buen ejemplo de cadena de markov. Ningún agente puede interactuar con ellos para modificarlos y son totalmente observables.



$A =$

		
	0.8	0.2
	0.1	0.9

Modelos de Markov: Cálculo de una secuencia.

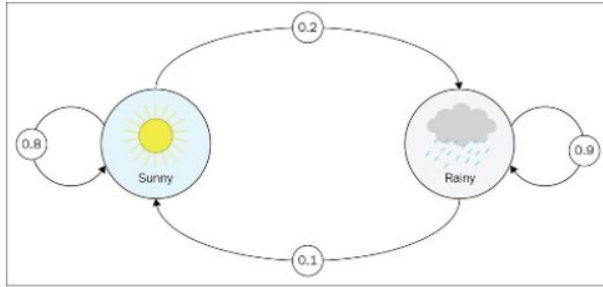


- Por la propiedad de Markov, la probabilidad de cualquier estado depende exclusivamente del estado anterior. De esta forma tenemos que:

$$\begin{aligned}
 P(s_{i1}, s_{i2}, \dots, s_{ik}) &= P(s_{ik} | s_{i1}, s_{i2}, \dots, s_{ik-1}) P(s_{i1}, s_{i2}, \dots, s_{ik-1}) \\
 &= P(s_{ik} | s_{ik-1}) P(s_{i1}, s_{i2}, \dots, s_{ik-1}) = \dots \\
 &= P(s_{ik} | s_{ik-1}) P(s_{ik-1} | s_{ik-2}) \dots P(s_{i2} | s_{i1}) P(s_{i1})
 \end{aligned}$$

- Supongamos que queremos calcular la probabilidad de la secuencia $O = \{S, S, R, R\}$. Esto es calcular $P(O|\text{Modelo})$. Asumimos que empezamos en el estado de sol, entonces tenemos:

$$P(\{S, S, R, R\} | \text{modelo}) = P(R|R) * P(R|S) * P(S|S) * P(S) = 0.9 * 0.2 * 0.8 * \boxed{0.6} \longrightarrow \text{Es conocido dada la frecuencia de aparición.}$$



- Por la propiedad de Markov, la probabilidad de cualquier estado depende exclusivamente del estado anterior. De esta forma tenemos que:

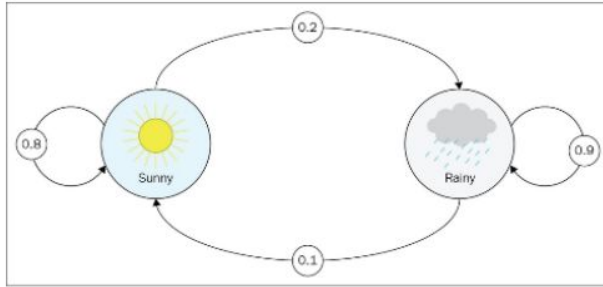
$$\begin{aligned} P(s_{i1}, s_{i2}, \dots, s_{ik}) &= P(s_{ik} | s_{i1}, s_{i2}, \dots, s_{ik-1}) P(s_{i1}, s_{i2}, \dots, s_{ik-1}) \\ &= P(s_{ik} | s_{ik-1}) P(s_{i1}, s_{i2}, \dots, s_{ik-1}) = \dots \\ &= P(s_{ik} | s_{ik-1}) P(s_{ik-1} | s_{ik-2}) \dots P(s_{i2} | s_{i1}) P(s_{i1}) \end{aligned}$$

- Supongamos que queremos calcular la probabilidad de la secuencia $O = \{S, S, R, R\}$. Asumimos que empezamos en el estado de sol, entonces tenemos:

$$P(\{S, S, R, R\} | \text{modelo}) = P(R|R) \cdot P(R|S) \cdot P(S|S) \cdot P(S) = 0.9 \cdot 0.2 \cdot 0.8 \cdot 0.6$$

¿La probabilidad que obtenemos es baja, normal o alta?

Modelos de Markov: Mantenerse en un estado en concreto



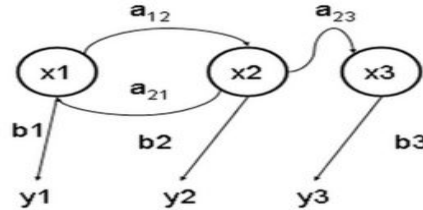
- Dado un modelo y un estado conocido, cual es la probabilidad de mantenerse en dicho estado D días?

$$P(O|Model, q_1 = S_i) = (a_{ii})^{d-1} \cdot (1 - a_{ii}) = p_i(D)$$

- **Ejercicio Opcional:**
 - ¿Cuál es la esperanza de un estado i en concreto? (+0.1 nota)
 - ¿Cuál sería la esperanza aproximada del estado sol y lluvia en nuestro modelo? (+0.4 nota)

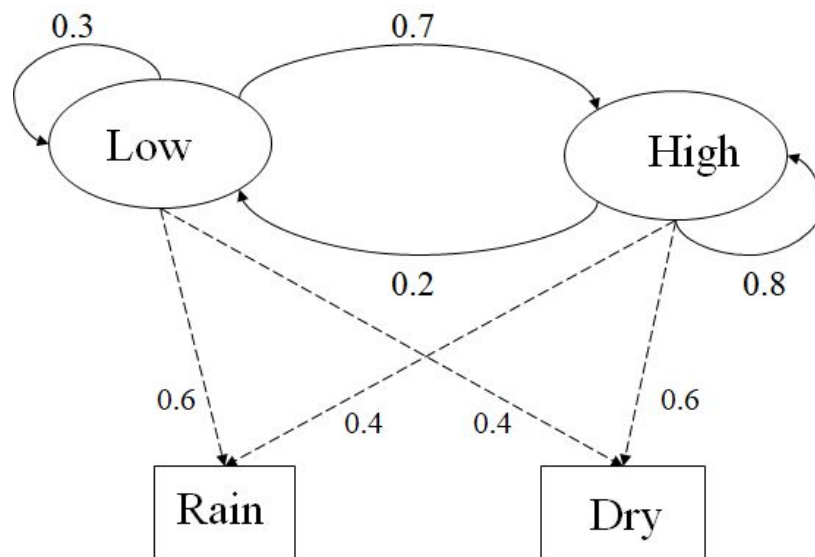
- Hay muchas situaciones en las que no podemos observar directamente el estado o los estados en los que estamos interesados.
- Solo podemos observar los efectos de estado sobre el sistema real.

- Los modelos de markov ocultos son cadenas de markov donde los estados no son directamente observables.
- En los modelos de markov normales los estados son visibles y por tanto las probabilidades de transición son los únicos parámetros. En un modelo oculto de markov solo las variables son visibles y cada estado tiene una distribución de probabilidad sobre estas variables de salida.

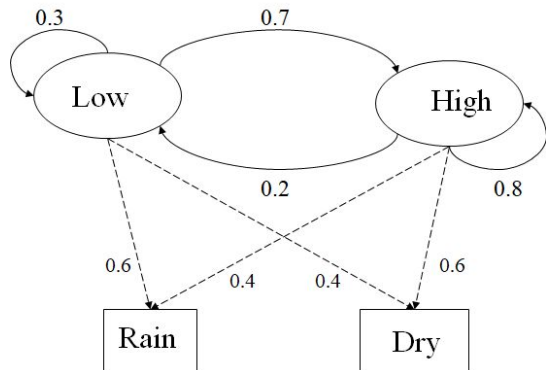


- Estos modelos son especialmente aplicados en el reconocimiento de formas temporales, reconocimiento del habla, escritura manual, reconocimiento de gestos o voz, etc.
- Para definir un HMM las siguientes probabilidades se tienen que definir; matriz de transición A con $a_{ij} = P(s_i|s_j)$, matriz de observaciones B con $b_i(v_m) = P(v_m|s_i)$ y un vector de probabilidades iniciales $\pi_i = P(s_i)$. Así pues, el modelo se representa por $M = (A, B, \pi)$.

Hidden Markov Models: Ejemplo I



Hidden Markov Models: Ejemplo I



- Queremos calcular la probabilidad de la secuencia dadas las observaciones {D,R}.

Considerando todas posibles secuencias de estados ocultos:

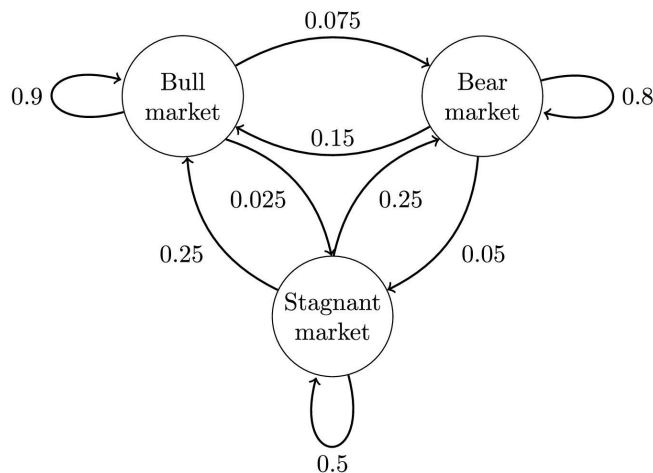
$$P(\{\text{'Dry'}, \text{'Rain'}\}) = P(\{\text{'Dry'}, \text{'Rain'}\}, \{\text{'Low'}, \text{'Low'}\}) + P(\{\text{'Dry'}, \text{'Rain'}\}, \{\text{'Low'}, \text{'High'}\}) + P(\{\text{'Dry'}, \text{'Rain'}\}, \{\text{'High'}, \text{'Low'}\}) + P(\{\text{'Dry'}, \text{'Rain'}\}, \{\text{'High'}, \text{'High'}\})$$

Donde el primer término es:



$$\begin{aligned} P(\{\text{'Dry'}, \text{'Rain'}\}, \{\text{'Low'}, \text{'Low'}\}) &= \\ P(\{\text{'Dry'}, \text{'Rain'}\} \mid \{\text{'Low'}, \text{'Low'}\}) P(\{\text{'Low'}, \text{'Low'}\}) &= \\ P(\text{'Dry'} \mid \text{'Low'}) P(\text{'Rain'} \mid \text{'Low'}) P(\text{'Low'}) P(\text{'Low'} \mid \text{'Low'}) &= \\ = 0.4 * 0.4 * 0.6 * 0.4 * 0.3 \end{aligned}$$

- Para los modelos ocultos de Markov es necesario crear un conjunto discreto de estados Z y un modelado de las observaciones con un modelo probabilístico $P(X_i | Z_i)$. Esto último refiere a la probabilidad de observar una muestra particular dado un estado determinado.
- Un ejemplo muy claro de una cadena de Markov oculta es la detección de regímenes, Bull, Bear, Stable, etc.



- La correspondiente función de densidad conjunta para las HMM viene dada por:

$$\begin{aligned} p(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) &= p(\mathbf{z}_{1:T})p(\mathbf{x}_{1:T} \mid \mathbf{z}_{1:T}) \\ &= \left[p(z_1) \prod_{t=2}^T p(z_t \mid z_{t-1}) \right] \left[\prod_{t=1}^T p(\mathbf{x}_t \mid z_t) \right] \end{aligned}$$

- En la primera línea se observa como la probabilidad conjunta de observar todos los estados ocultos y observaciones es igual a la probabilidad de observar los estados ocultos multiplicados por la probabilidad de observar las muestras condicionadas a los estados.
- En la segunda línea la función de transición viene dada por la probabilidad de un estado dado el anterior y las probabilidades de las observaciones dados los estados ocultos.
- Asumimos la invarianza temporal de las funciones de transición.
- Los estados también pueden ser continuos, véase las HMM para los assets returns.

Hidden Markov Models: Tres problemas a tratar

- **Evaluation Problem:** Dada la HMM $M = (A, B, \pi)$ y la secuencia de observaciones O , calcular la probabilidad de que el modelo M haya generado dicha secuencia O .
 - Se utilizan algoritmos **Forward-Backward HMM**. De manera iterativa se aproxima una variable que representa la probabilidad conjunta de la observación de O y el estado S_i en el instante K .
 - N^2K operaciones algoritmos complejos.
- **Decoding Problem:** Dada la HMM $M = (A, B, \pi)$ y la secuencia de observaciones O . Calcular la secuencia más probable de estados ocultos S_i que haya producido la secuencia de observaciones O .
 - Se busca la secuencia de estados Q que maximiza $P(Q|O)$.
 - Por fuerza bruta lleva un tiempo exponencial. Se utiliza el algoritmo de **Viterbi**.
- **Learning Problem:** Dados algunos datos de entrenamiento como secuencias observadas O y la estructura general del HMM. Determinar los parámetros $M=(A, B, \pi)$ de la HMM que ajustan mejor a los datos.
 - Se busca determinar M tal que maximiza $P(O|M)$
 - No hay un algoritmo que produzca los valores óptimos de los parámetros.
 - Se utilizan métodos iterativos de expectation-maximization para encontrar máximos locales de $P(O|M)$ - **Baum-Welch algorithm**

The background is a dark blue, semi-transparent image. It depicts a person's hands typing on a laptop keyboard. The laptop screen shows various data visualizations, including bar charts and line graphs. A pair of glasses is visible on the desk to the left of the laptop. The overall aesthetic is professional and tech-oriented.

Probabilistic Graphical Models

Relación con RL

aoteo@grupobme.es

Introducción

- La idea del aprendizaje cuando se interactúa con el entorno es probablemente el primer concepto de aprendizaje que nos encontramos en la naturaleza.
- Las **interacciones** con los **entornos** provocan **reacciones** y **consecuencias** con el objetivo de obtener **resultados** o las metas que nos proponemos.
- Durante las próximas clases vamos a aprender sobre el aprendizaje computacional a partir de las **interacciones** o lo que llamamos **aprendizaje por refuerzo**.



- Así pues, podemos entender el **aprendizaje por refuerzo** como un **sistema o modelo** que mapea situaciones a acciones para **maximizar una señal de recompensa**.
- Al contrario que en otros campos del machine learning **no nos encargaremos de indicar que acciones tomar o aprender las acciones a partir de unas etiquetas**.
- Los modelos de aprendizaje por refuerzo **han de aprender que acciones maximizan la señal de recompensa** teniendo en cuenta la situación y la acción a tomar.

Problema con datos etiquetados

Datos de entrada

Aprendizaje
Supervisado

Salida
(Mapeo)

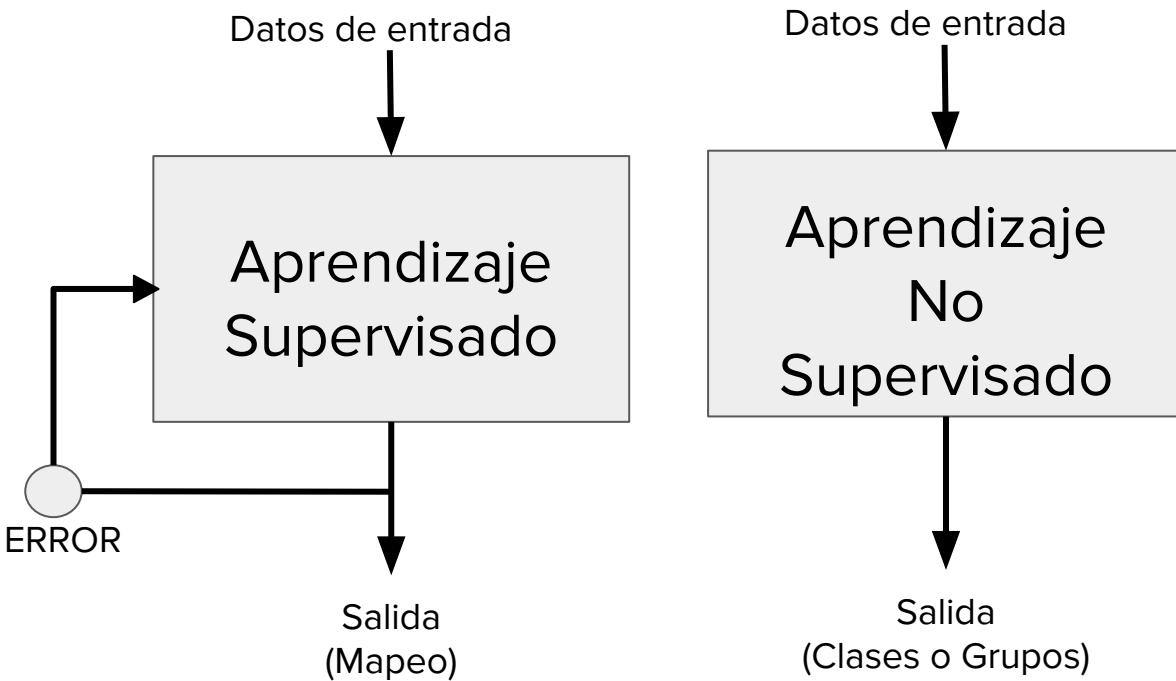
ERROR

Problema sin etiquetas

Datos de entrada

Aprendizaje
No
Supervisado

Salida
(Clases o Grupos)



Problema con datos etiquetados

Datos de entrada

Aprendizaje
Supervisado

Salida
(Mapeo)

ERROR

Problema sin etiquetas

Datos de entrada

Aprendizaje
No
Supervisado

Salida
(Clases o Grupos)

Problema sin etiquetas

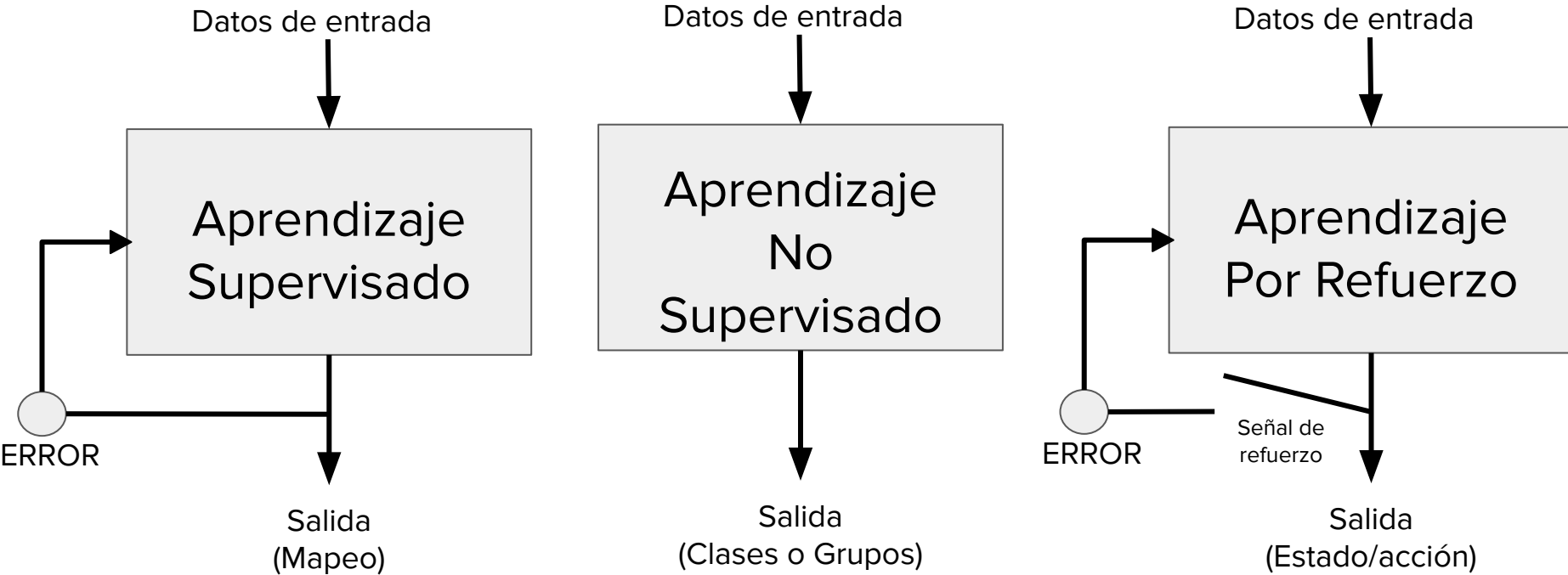
Datos de entrada

Aprendizaje
Por Refuerzo

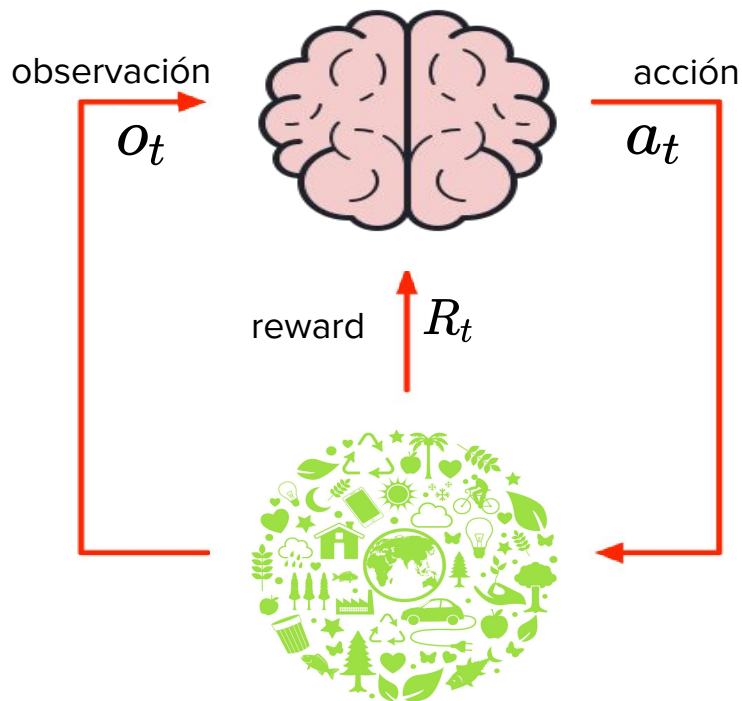
Salida
(Estado/acción)

ERROR

Señal de
refuerzo

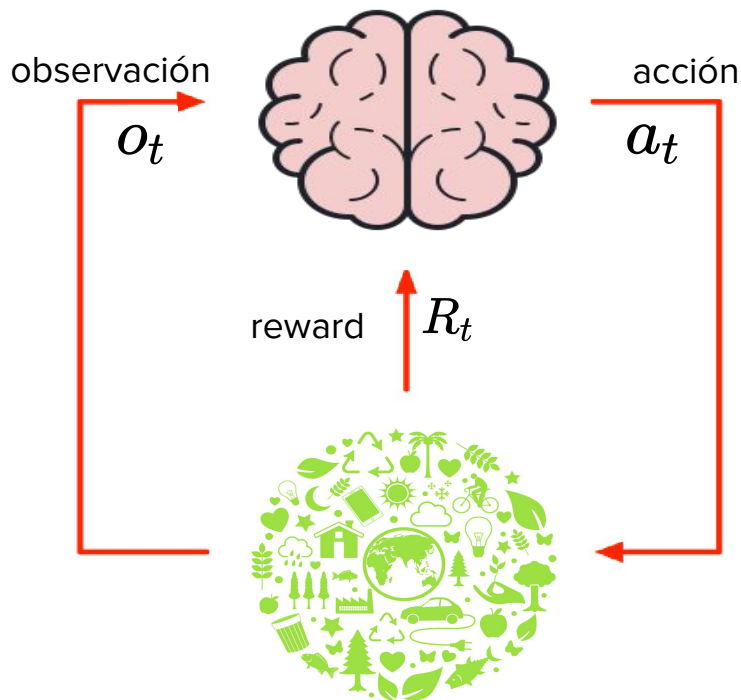


¿De que se compone un entorno de aprendizaje reforzado?



- Problema
- Acción
- Estado
- Recompensa

¿De que se compone un entorno de aprendizaje reforzado?



- El agente intenta una secuencia de acciones que determinaremos como a_t
- Se observa la salida o reacción y los resultados (denotadas como s_{t+1} y r_t) una vez tomadas las acciones.
- Estadísticamente se estiman las relaciones entre las acciones y las recompensas.

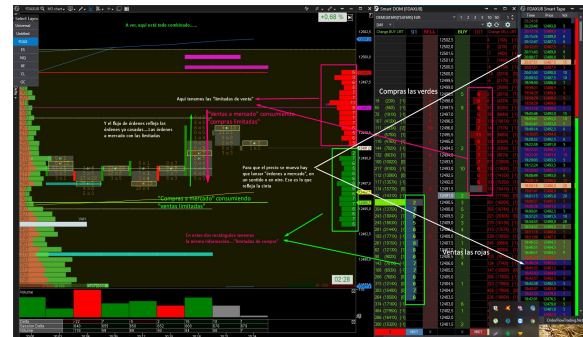
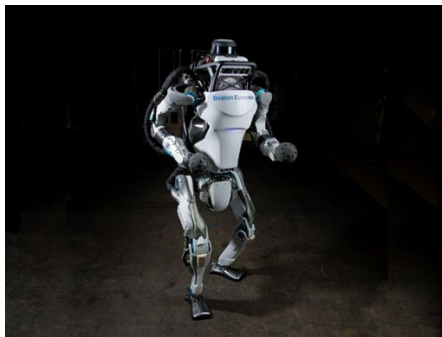
¿Cuándo utilizar el aprendizaje reforzado?

- Los datos están en forma de **trayectorias**.
 - Las muestras no son independientes entre ellas.
- Necesitas **tomar** una **secuencia de decisiones** que están relacionadas.
 - Para solucionar el problema se toman decisiones, acciones no independientes entre ellas.
- Se observa un **feedback parcial, total o con ruido** de los estados en la elección de las acciones.
 - Observamos una respuesta a las acciones que se toman.
- Cuando **existe una ganancia de la optimización de una acción sobre una porción de la trayectoria**.

Definiciones y Conceptos Básicos

Entorno o Environment

- El entorno es el conjunto de **estados** que forman el problema que buscamos resolver.
- Intuitivamente el entorno es un simulador que busca replicar la realidad del problema que estamos resolviendo. Es este simulador se reproducen todos los eventos y reacciones producidas por la interacción del usuario mediante sus acciones.
- En la práctica el entorno puede ser la el espacio de estados real.



Entorno o Environment

- Devuelve mi **estado** actual (s_t)
- Recibe una acción (a_t)
- Devuelve una recompensa (r_t) y un nuevo estado (s_{t+1})

¿En qué se traduce el estado (S)?



Vector de características

- Continuas
- Discretas



Estados

- Continuos
- Discretos



Nos podemos ayudar de la tecnología para construir las características ("**features**")

- Día 3, S_3

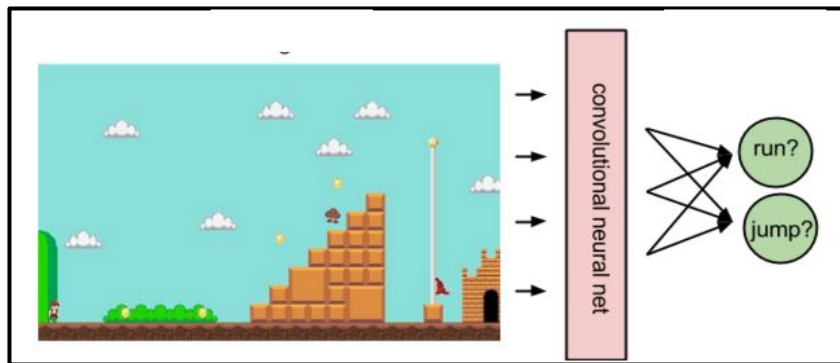
Verano	Nubes	Humedad	Viento
1	0	0.75	16.3

- Activo X en S_3

Return	Std	Beta	Posición
0.07	0.16	1.2	1

Agente

- El agente es aquella serie de **reglas** o **modelos** programados que dada una información del entorno toman una serie de decisiones.
- Puede ser un agente determinista como es el caso de un agente basado en reglas o un agente que se adapta como son aquellos que veremos a lo largo del curso.



Agente: Categorización

- **Value Based**
 - No Policy (Implicit)
 - Value Function
- **Policy Based**
 - Policy
 - No Value Function
- **Actor Critic**
 - Policy
 - Value Function
- **Model Free**
 - Policy and/or Value Function
 - No Model
- **Model Based**
 - Policy and/or Value Function
 - Model

Reward

- Es el valor que recibimos **periódicamente** del entorno.
- Su propósito es decirle al agente como de bien lo está haciendo.
- Es un **valor local** por lo que obtener un gran valor puntual no implica que es siguiente sea un buen valor.
- El agente trata de obtener la **mayor acumulación** de premios posible.

Acciones

- Las acciones son las cosas que el agente puede hacer.
- La complejidad de dichas acciones es variable y se pueden diferenciar 2 tipos principalmente: **continuas** y **discretas**.
 - Las **acciones discretas** son aquellas definidas por un **conjunto finito** de cosas mutuamente excluyentes que el agente puede hacer.
 - Las **acciones continuas** tienen un **valor** pegado a la acción como puede el ángulo de giro de un coche o la **aceleración del mismo**.

Observaciones y Estados

- Las observaciones son **piezas de información** que el entorno ofrece al agente con las cuales le dice que situación actual tiene.
- Dichas observaciones pueden incluso ofrecer información sobre los premios que se están acumulando.
- Es importante distinguir entre el **estado del entorno** y las **observaciones**. El estado contiene cada átomo del universo lo cual puede llegar a ser imposible de medir mientras que las **observaciones son porciones de información** que medimos y obtenemos.

Entornos para el Reinforcement Learning

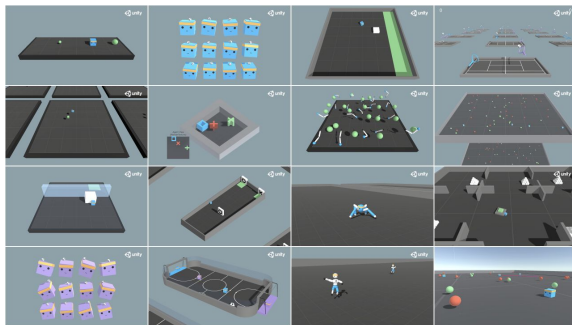
- **OpenAI Gym** es un conjunto de herramientas para desarrollar y comparar algoritmos de aprendizaje de refuerzo. Es compatible con cualquier biblioteca de computación numérica, como TensorFlow o Theano.

OpenAI Gym

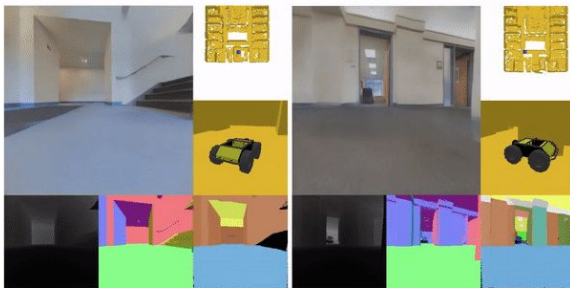
- La función **step** del entorno devuelve exactamente lo que necesitamos. Devuelve cuatro valores:
 - **Observation**: un objeto específico del entorno que representa su observación del entorno. Por ejemplo, los datos de píxeles de una cámara, los ángulos de unión y las velocidades de unión de un robot, o el estado del tablero en un juego de mesa.
 - **Reward**: Cantidad de recompensa conseguida por la acción anterior. La escala varía según el entorno, pero el objetivo siempre es aumentar la recompensa total.
 - **Done**: si es hora de restablecer el entorno de nuevo. La mayoría de las tareas (pero no todas) se dividen en episodios bien definidos, y al ser verdaderas, indica que el episodio ha terminado.
 - **Info**: información de diagnóstico útil para la depuración.

Otros Simuladores

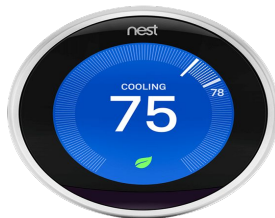
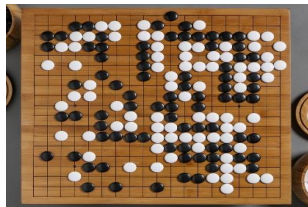
- Unity ML-Agents: Cuenta con una API en python que permite el entrenamiento de modelos sobre los entornos desarrollados en Unity.



- GibsonEnv: Entorno para agentes activos encarnados con percepción del mundo real.



Aplicaciones



Breakout: <https://www.youtube.com/watch?v=TmPfTpjtdgg>

Walking: <https://www.youtube.com/watch?v=gn4nRCC9TwQ>

Cars: <https://www.youtube.com/watch?v=Aut32pR5PQA>

Parking: https://www.youtube.com/watch?v=VMp6pq6_Qjl

Hide and Seek: <https://www.youtube.com/watch?v=Lu56xVIZ40M&t=3s>

Aplicaciones



<https://deepmind.com/blog/article/alphago-zero-starting-scratch>

Markov Decision Process

- Los **procesos de markov** se originan antes que el aprendizaje por refuerzo y son un punto de partida en los problemas de decisión mediante rewards

Proceso de Markov

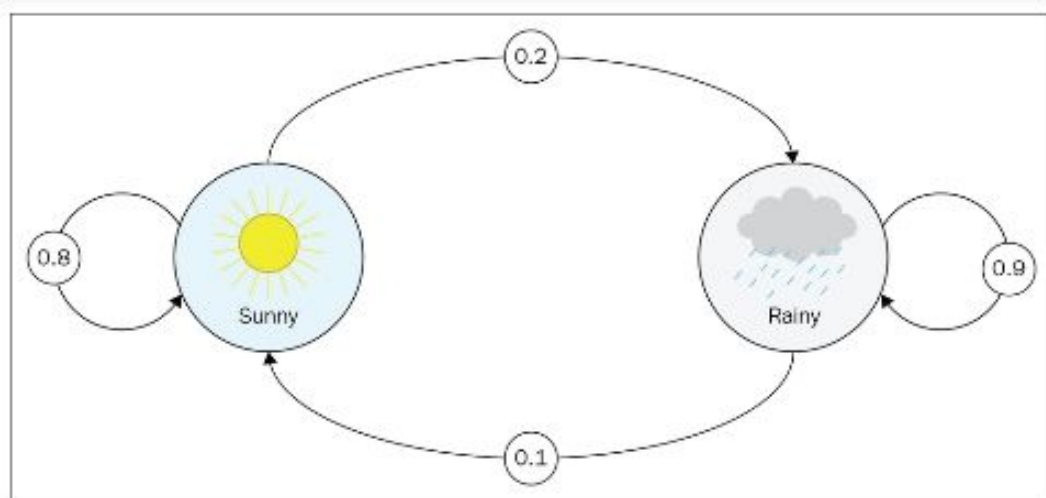
- Imagina que tienes un sistema delante de ti del cual solo puedes observar. Lo que observas son los llamados estados del sistema y el sistema puede cambiar entre ellos de acuerdo a ciertas dinámicas sin que puedas influir en ellas.
- Dichos procesos suponen la **propiedad de Markov**:

$$P(X_n = x_n \mid X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = P(X_n = x_n \mid X_{n-1} = x_{n-1}).$$

- De dichos sistemas se pueden capturar las probabilidades de transición entre los estados como matrices de transición NxN donde N es el número de estados en nuestro modelo.
- Un **MP** es un conjunto de estados que pueden representarse como un **matriz (T)** con las **probabilidades de transición** que definen el sistema dinámico.

Proceso de Markov

- El clima puede ser un claro ejemplo de proceso de Markov. El ejemplo más sencillo, en un sistema observable y autónomo se le llama **Cadena de Markov**.



	SOL	LLUVIA
SOL	0.8	0.2
LLUVIA	0.1	0.9

Markov Reward Process

- Para introducir los premios tenemos que extender el **proceso de Markov** para añadir valor a nuestra **matriz de transición** de un estado a otro. La manera habitual es tener una matriz adicional.
- También se introduce un término **gamma γ** que es un término de descuento. Así pues para cada episodio definimos el **retorno** como:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- En cada punto el **retorno** es la suma de **premios (R)** siguientes que se pueden obtener, donde los más distantes están multiplicados por un factor de descuento.
- **¿Qué ocurre con $\gamma=0$ y $\gamma=1$?**

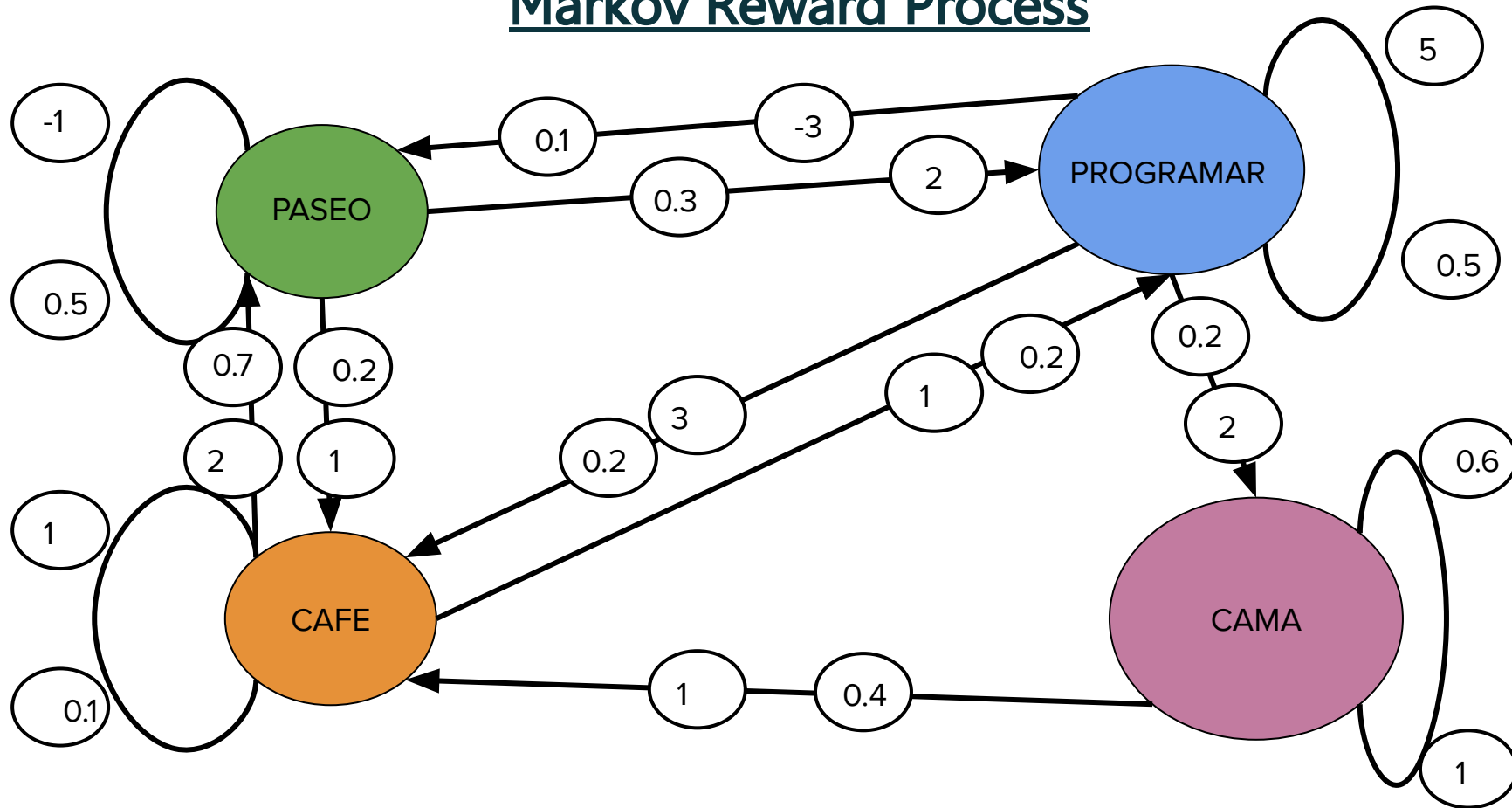
Markov Reward Process

- Debido a la aleatoriedad de los procesos el **retorno** puede variar incluso en el mismo estado por lo que se define el **valor de los estados** como:

$$V(s) = \mathbb{E}[G | S_t = s]$$

- Para cada estado s , el valor $V(s)$ es la media o valor esperado del retorno obtenido por el **MRP**.
- Veamos un ejemplo extraído de **Deep Reinforcement Learning Hands-On**.

Markov Reward Process



Markov Reward Process

- ¿Cómo calculamos el valor de los estados?

- Tomamos **gamma = 0** por simplicidad.

$$V(PASEO) = -1 * 0.5 + 2 * 0.3 + 1 * 0.2 = 0.3$$

$$V(CAMA) = 1 * 0.6 + 1 * 0.4 = 1$$

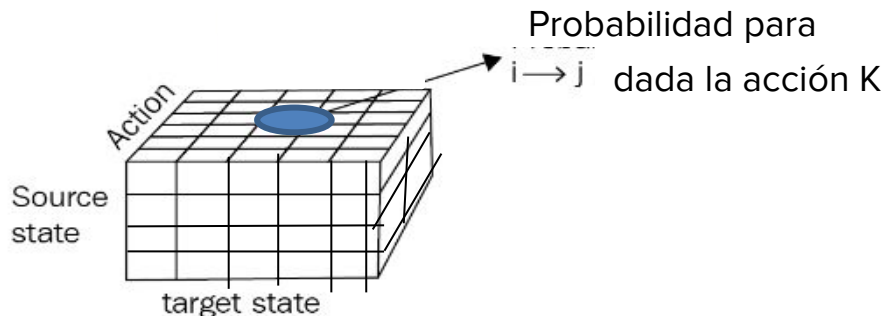
$$V(CAFE) = 2 * 0.7 + 1 * 0.1 + 3 * 0.2 = 2.1$$

$$V(PROGRAMAR) = 5 * 0.5 + (-3) * 0.1 + 1 * 0.2 + 2 * 0.2 = 2.8$$

- Entonces “PROGRAMAR” es el estado con más valor en el que nos podemos encontrar.

Markov Decision Process

- La extensión de los **MRP** se realiza mediante la adición de un conjunto de acciones A (action space) las cuales han de ser finitas.
- Se ha de condicionar nuestra matriz de transición y rewards con una dimensión más. Esto implica que el agente deja de ser pasivo y esto puede afectar a las probabilidades de los estados target.



Markov Decision Process

- La definición intuitiva de **policy** es un conjunto de reglas que controlan las acciones del agente. Se define de la siguiente manera:

$$\pi(a|s) = P(A_t = a|S_t = s)$$

Nota: La política (policy) puede entenderse como una matriz

- Al utilizar probabilidades introducimos la aleatoriedad en las elecciones del agente.
- ¿Qué ocurre si tenemos probabilidad igual a 1?
- ¿Qué ocurre si tenemos una política fija?
- Veamos como funciona una política determinista en un problema y trabajemos con un entorno. Abramos el notebook **[Implementacion_politica_determinista.ipynb](#)**.

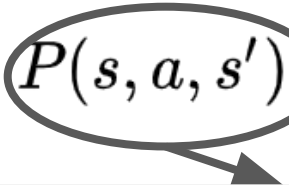
Markov Decision Process

- El objetivo de un **MDP** es maximizar el retorno acumulado mediante las decisiones tomadas. Para ello es necesario encontrar una **política óptima**.
- Intuitivamente, la **política óptima** es aquella que permite obtener el máximo retorno posible a través del sistema. Para obtenerla se necesitan determinar los siguientes elementos:
 - Una manera de determinar el valor de los estados.
 - Un valor estimado de tomar una acción en un determinado estado.
- La **Ecuación de Bellman de la optimalidad** nos da la manera de calcular el valor óptimo de cada estado $V^*(s)$ de la forma siguiente:

$$V^*(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') \cdot \gamma V(s')]$$

Markov Decision Process

- El objetivo de un **MDP** es maximizar el retorno acumulado mediante las decisiones tomadas. Para ello es necesario encontrar una **política óptima**.
- Intuitivamente, la **política óptima** es aquella que permite obtener el máximo retorno posible a través del sistema. Para obtenerla se necesitan determinar los siguientes elementos:
 - Una manera de determinar el valor de los estados.
 - Un valor estimado de tomar una acción en un determinado estado.
- La **Ecuación de Bellman de la optimalidad** nos da la manera de calcular el valor óptimo de cada estado $V^*(s)$ de la forma siguiente:

$$V^*(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') \cdot \gamma V(s')]$$


Es la probabilidad de transición del estado s al estado s' tomando la acción a . Puede ser notado como $T(s, a, s')$.

Markov Decision Process

- El objetivo de un **MDP** es maximizar el retorno acumulado mediante las decisiones tomadas. Para ello es necesario encontrar una **política óptima**.
- Intuitivamente, la **política óptima** es aquella que permite obtener el máximo retorno posible a través del sistema. Para obtenerla se necesitan determinar los siguientes elementos:
 - Una manera de determinar el valor de los estados.
 - Un valor estimado de tomar una acción en un determinado estado.
- La **Ecuación de Bellman de la optimalidad** nos da la manera de calcular el valor óptimo de cada estado $V^*(s)$ de la forma siguiente:

$$V^*(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') \cdot \gamma V(s')]$$

Es el reward inmediato que se recibe al pasar de s a s' tomando la acción a

Markov Decision Process

- El objetivo de un **MDP** es maximizar el retorno acumulado mediante las decisiones tomadas. Para ello es necesario encontrar una **política óptima**.
- Intuitivamente, la **política óptima** es aquella que permite obtener el máximo retorno posible a través del sistema. Para obtenerla se necesitan determinar los siguientes elementos:
 - Una manera de determinar el valor de los estados.
 - Un valor estimado de tomar una acción en un determinado estado.
- La **Ecuación de Bellman de la optimalidad** nos da la manera de calcular el valor óptimo de cada estado $V^*(s)$ de la forma siguiente:

$$V^*(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') \cdot \gamma V(s')]$$

Valor del estado s' con el descuento gamma

Markov Decision Process

- En la práctica se puede resolver el problema de manera iterativa de manera que se converga al mismo resultado que utilizando la fórmula de la optimalidad.

Value Iteration

$$V_{k+1}(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot V_k(s')]$$

Markov Decision Process

- En la práctica se puede resolver el problema de manera iterativa de manera que se converga al mismo resultado que utilizando la fórmula de la optimalidad.

Value Iteration

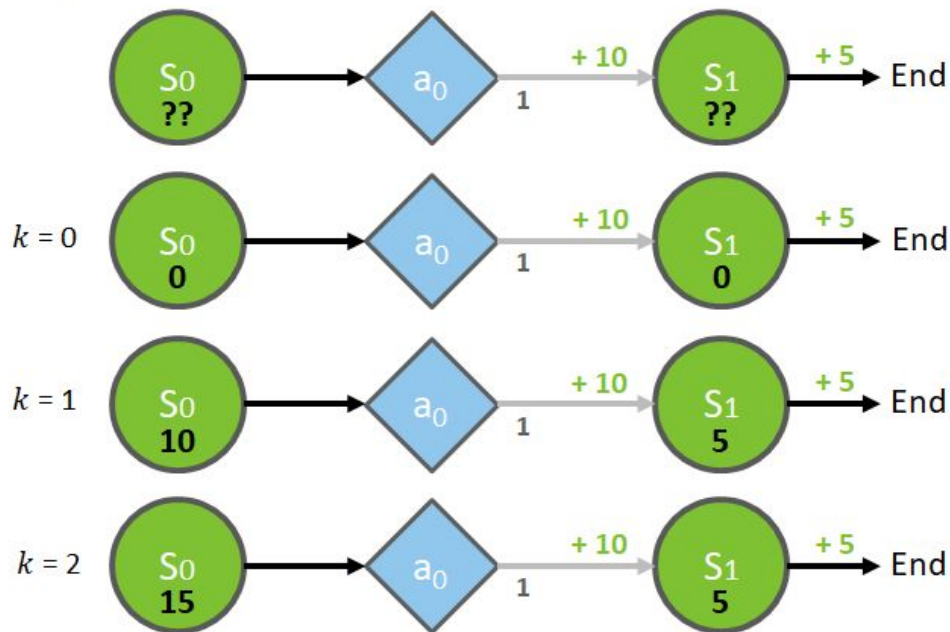
$$V_{k+1}(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot V_k(s')]$$

k: Número de la iteración

MDP: Value Iteration

$$V_{k+1}(s) = \max_a \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot V_k(s')]$$

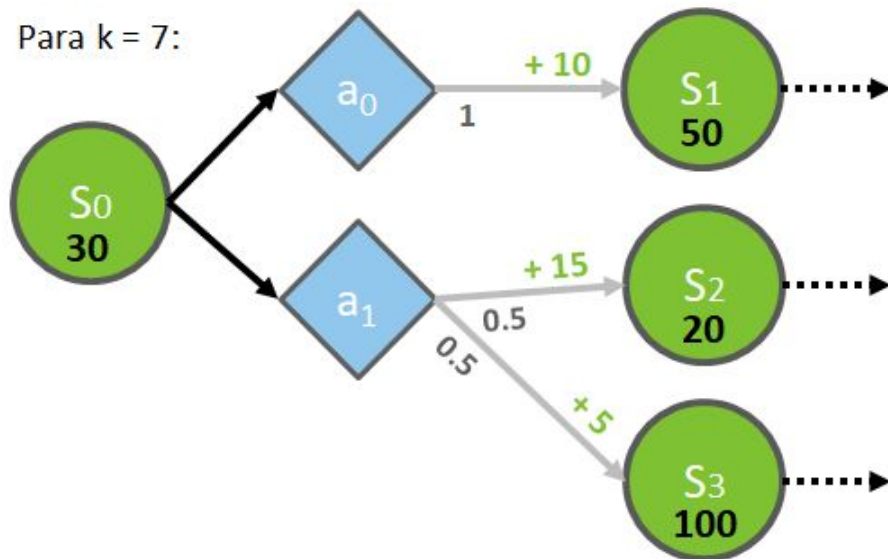
Dado $\gamma = 1$



MDP: Q-Value Iteration

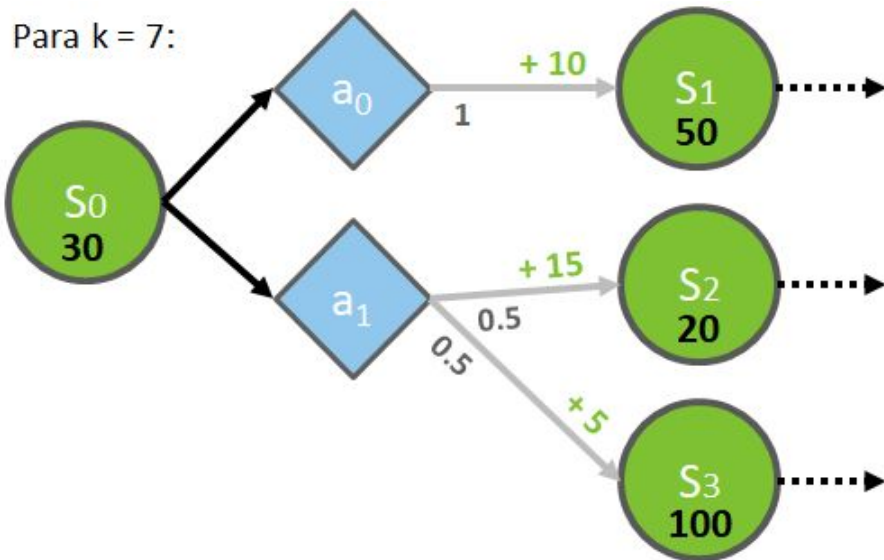
$$Q_{k+1}(s, a) = \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot \max_{a'} Q_k(s', a')]$$

Para $k = 7$:



MDP: Q-Value Iteration

$$Q_{k+1}(s, a) = \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot \max_{a'} Q_k(s', a')]$$



Dado $\gamma = 0.5$:

$$Q(s_0, a_0) = 1(10 + 0.5(50)) = 35$$

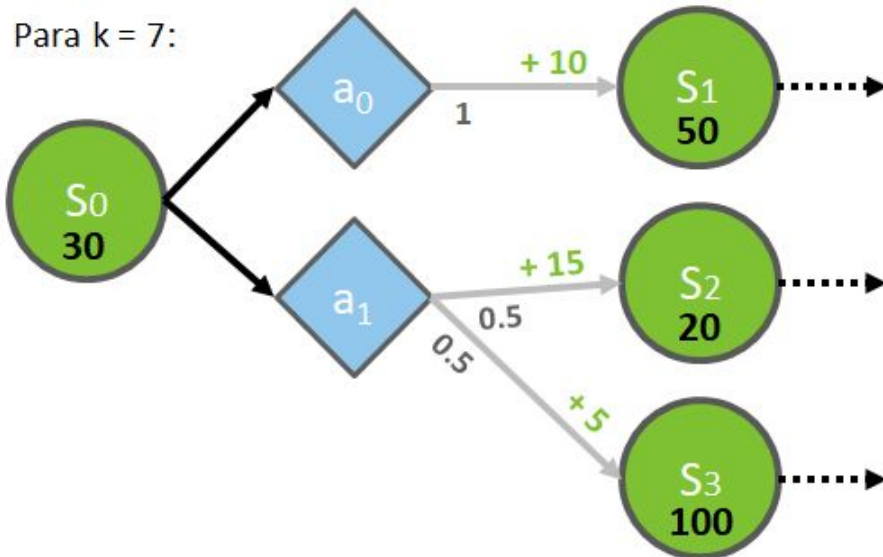
$$0.5 (15 + 0.5(20)) = 12.5$$

$$0.5 (5 + 0.5(100)) = 27.5$$

$$Q(s_0, a_1) = 40$$

MDP: Q-Value Iteration

$$Q_{k+1}(s, a) = \sum_{s'} P(s, a, s') [R(s, a, s') + \gamma \cdot \max_{a'} Q_k(s', a')]$$



Dado $\gamma = 0.5$:

$$Q(s_0, a_0) = 1(10 + 0.5(50)) = 35 \quad \text{✗}$$

$$0.5 (15 + 0.5(20)) = 12.5$$

$$0.5 (5 + 0.5(100)) = 27.5$$

$$Q(s_0, a_1) = 40 \quad \text{✓}$$

Entonces:

$a_1 \rightarrow$ Acción a tomar, que tiene un valor esperado de 40 para S_0

$V(s_0) = 40 \rightarrow$ ya no es 30



aoteo@grupobme.es