

# Aprendizaje por refuerzo

---

Clase 18: RL Inverso





# Antes de empezar...

- Dudas de
  - Tarea 4
  - Proyecto
  - Examen



## Para el día de hoy...

- RL inverso



# El contexto

- RL no necesita instrucciones detalladas
- Generar una señal de recompensa no depende del conocimiento de cuales deberían ser las acciones correctas
- El éxito depende de que tan bien las señales de recompensa marquen la tarea a realizar
- Y... que tan bien evalúen el progreso a la meta

# Diseñando una recompensa

- Diseñar una parte del ambiente que calcule  $R_t$  en cada tiempo  $t$

$$r(s, a)$$

# Algunos retos

- Encontrar una señal de recompensa tal que el agente aprenda el comportamiento que se desea
- El agente puede encontrar formas inesperadas de encontrar recompensa, pueden ser no deseadas o peligrosas
- Las recompensas pueden ser espaciadas

# Una primera aproximación

- Prueba y error
  - Diseñamos una señal de recompensa
  - Probamos
  - Si el agente no aprende, es muy lento o aprende mal, cambiamos la recompensa e iteramos

# Reward Shaping

- Cambiar la señal de recompensa mientras se aprende
- Iniciar con una señal de recompensa no esparza
- Gradualmente modificarla hacía la recompensa del problema original

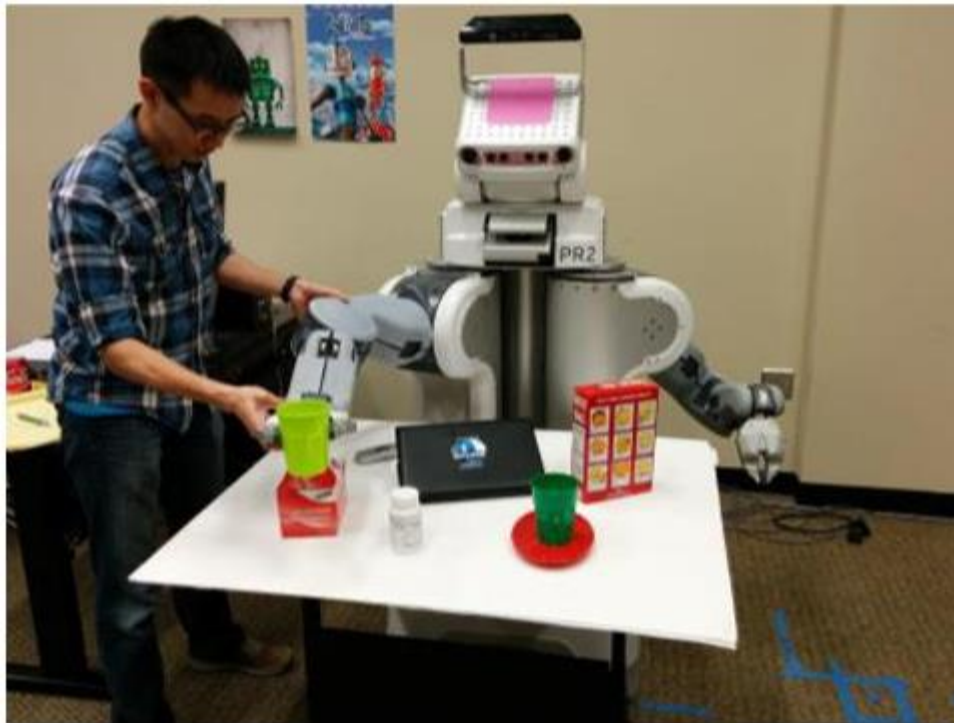


# ¿Y si no tenemos idea de una señal?

- Podemos utilizar
  - Aprendizaje por imitación
  - Aprendizaje por demostración
  - Aprendizaje inverso
- La idea
  - Beneficiarse de un agente experto que haya aprendido con aprendizaje supervisado o extrayendo la señal de recompensa

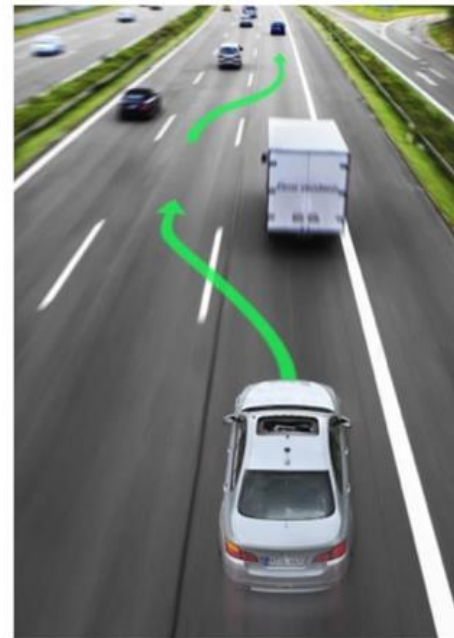
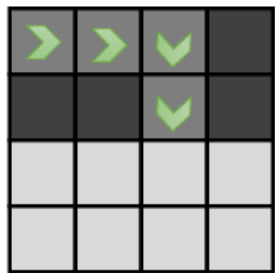
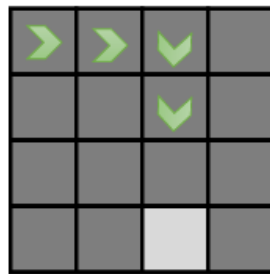
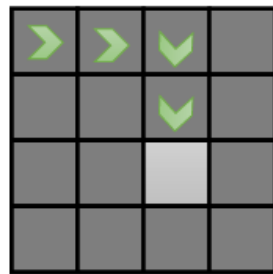
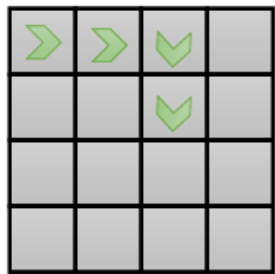
# La perspectiva de imitación

- Aprendizaje por imitación
  - Copiar las acciones realizadas por los expertos
  - No hay razonamiento acerca de las acciones
- Imitación de humanos
  - Copiar la intención del experto
  - Se pueden tomar diferentes acciones



# Aprendizaje por refuerzo inverso

Inferir la función de recompensa por medio de recompensas

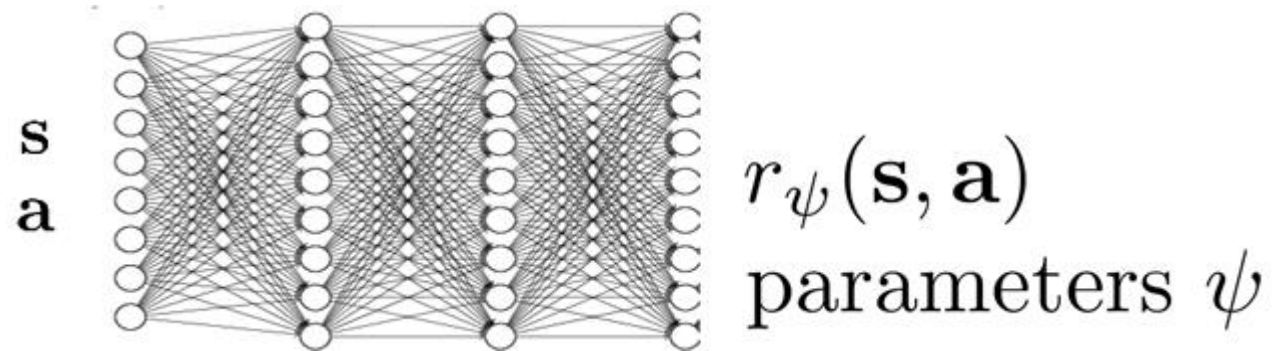


→  $r(\mathbf{s}, \mathbf{a})$

# Formalmente

- Aprendizaje por refuerzo
  - $s \in S, a \in A$
  - Transiciones  $p(s'|s, a)$
  - Función de recompensa  $r(s, a)$
  - Aprender  $\pi^*(a|s)$
- Aprendizaje por refuerzo inverso
  - $s \in S, a \in A$
  - Transiciones  $p(s'|s, a)$
  - Muestras  $\{\tau_i\}$  muestreadas de  $\pi^*(\tau)$
  - Aprender  $r_\psi(s, a)$  y después  $\pi^*(a|s)$

$$r_\psi(\mathbf{s}, \mathbf{a}) = \sum_i \psi_i f_i(\mathbf{s}, \mathbf{a}) = \psi^T \mathbf{f}(\mathbf{s}, \mathbf{a})$$



# Pareo de características

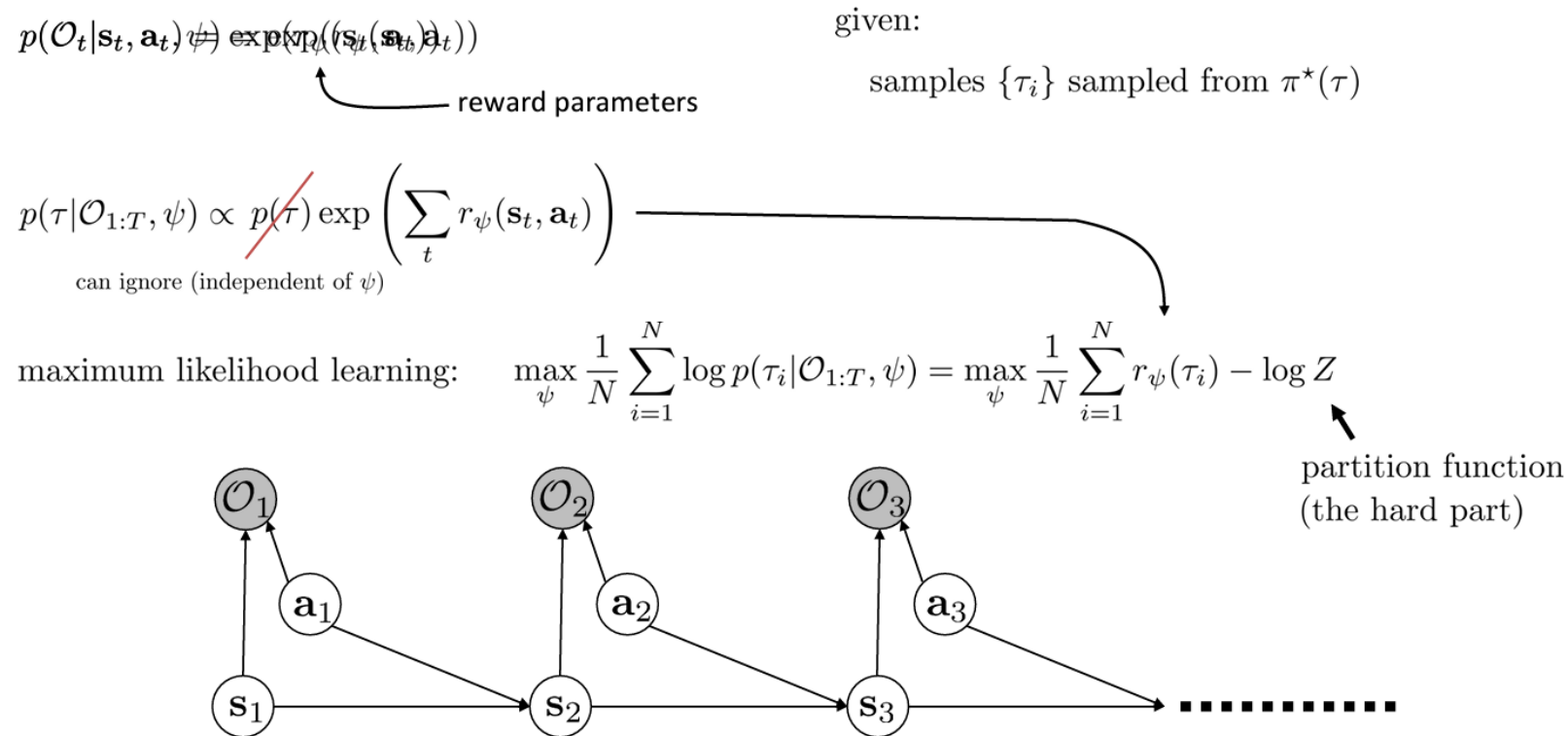
- Función de recompensa lineal
  - $r_\psi(s, a) = \sum_i \psi_i f_i(s, a) = \psi^T f(s, a)$
- Si las características  $f$  son importantes, parear sus esperanzas
  - Sea  $\pi^{r_\psi}$  la política óptima para  $r_\psi$
  - Elegir  $\psi$  tal que  $\mathbb{E}_{\pi^{r_\psi}}[f(s, a)] = \mathbb{E}_{\pi^*}[f(s, a)]$
- Principio de máximo margen
  - $\max_{\psi, m} m$
  - Tal que
    - $\psi^T \mathbb{E}_{\pi^*}[f(s, a)] \geq \max_{\pi \in \Pi} \psi^T \mathbb{E}_\pi[f(s, a)] + m$



# Una manipulación

- Transformar
  - $\max_{\psi, m}$
  - Tal que
    - $\psi^T \mathbb{E}_{\pi^*}[f(s, a)] \geq \max_{\pi \in \Pi} \psi^T \mathbb{E}_{\pi}[f(s, a)] + m$
- En
  - $\max_{\psi} \frac{1}{2} \|\psi\|^2$
  - Tal que
    - $\psi^T \mathbb{E}_{\pi^*}[f(s, a)] \geq \max_{\pi \in \Pi} \psi^T \mathbb{E}_{\pi}[f(s, a)] + D(\pi, \pi^*)$

# Aprendiendo la variable de optimalidad




# La función de partición


$$\max_{\psi} \frac{1}{N} \sum_{i=1}^N r_{\psi}(\tau_i) - \log Z$$

$$Z = \int p(\tau) \exp(r_{\psi}(\tau)) d\tau$$

$$\nabla_{\psi} \mathcal{L} = \frac{1}{N} \sum_{i=1}^N \nabla_{\psi} r_{\psi}(\tau_i) - \underbrace{\frac{1}{Z} \int p(\tau) \exp(r_{\psi}(\tau)) \nabla_{\psi} r_{\psi}(\tau) d\tau}_{p(\tau | \mathcal{O}_{1:T}, \psi)}$$

$$\nabla_{\psi} \mathcal{L} = E_{\tau \sim \pi^*(\tau)} [\nabla_{\psi} r_{\psi}(\tau_i)] - E_{\tau \sim p(\tau | \mathcal{O}_{1:T}, \psi)} [\nabla_{\psi} r_{\psi}(\tau)]$$

  
estimate with expert samples

  
soft optimal policy under current reward

# Un algoritmo: MaxEnt IRL

1. Given  $\psi$ , compute backward message  $\beta(\mathbf{s}_t, \mathbf{a}_t)$
2. Given  $\psi$ , compute forward message  $\alpha(\mathbf{s}_t)$
3. Compute  $\mu_t(\mathbf{s}_t, \mathbf{a}_t) \propto \beta(\mathbf{s}_t, \mathbf{a}_t)\alpha(\mathbf{s}_t)$
4. Evaluate  $\nabla_{\psi} \mathcal{L} = \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\psi} r_{\psi}(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) - \sum_{t=1}^T \int \int \mu_t(\mathbf{s}_t, \mathbf{a}_t) \nabla_{\psi} r_{\psi}(\mathbf{s}_t, \mathbf{a}_t) d\mathbf{s}_t d\mathbf{a}_t$
5.  $\psi \leftarrow \psi + \eta \nabla_{\psi} \mathcal{L}$

# Notas

---

Maximizar el margen es arbitrario



No modela claramente la sub optimalidad del experto



Es un problema complejo de resolver





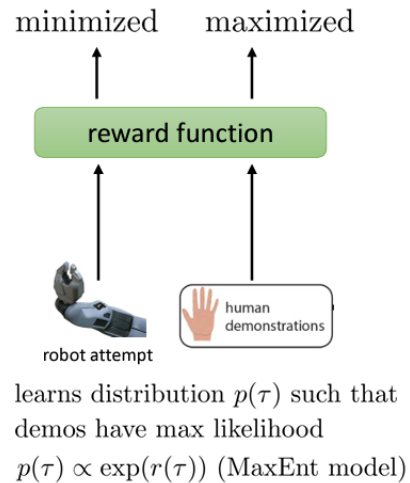
# Otros problemas

- Aplicar a problemas grandes y espacios continuos de estados y acciones
- Trabajar con estados obtenidos solo por muestreo
- Dinámica desconocida

# Otros enfoques

- Utilizar Redes generativas antagónicas (GANs)

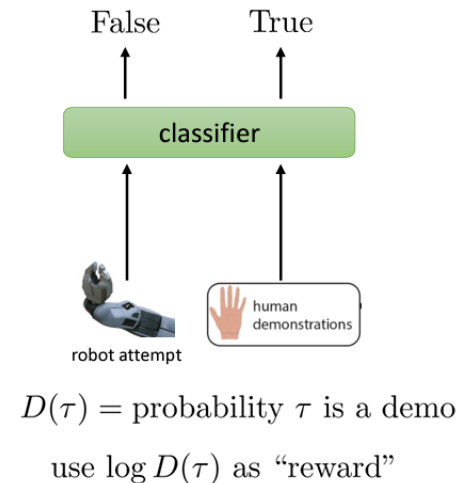
Guided Cost Learning  
Finn et al., ICML 2016



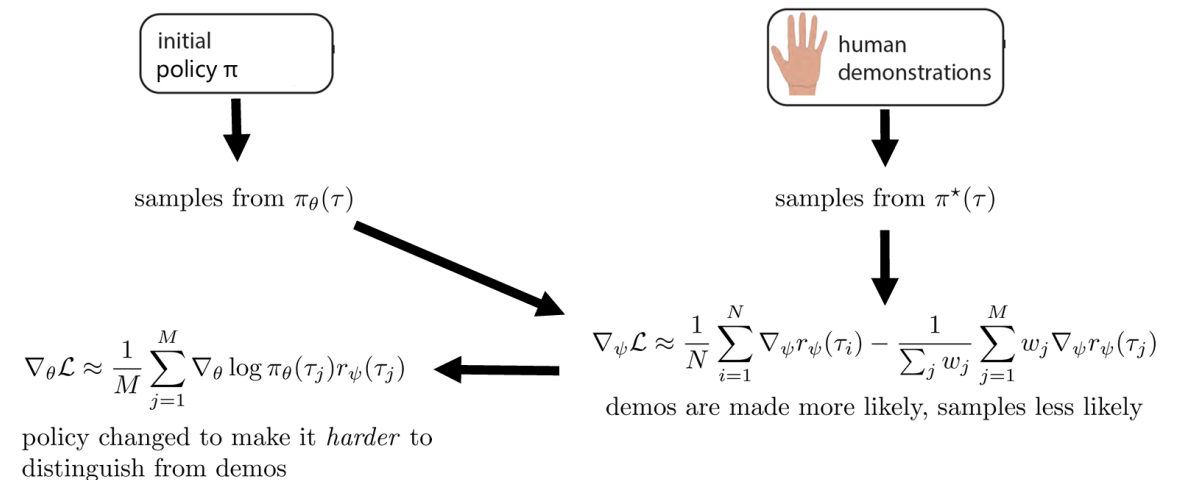
$$D(\tau) = \frac{\frac{1}{Z} \exp(r(\tau))}{\frac{1}{Z} \exp(r(\tau)) + \pi(\tau)}$$

actually the same thing!

Generative Adversarial Imitation Learning  
Ho & Ermon, NIPS 2016



$D(\tau)$  = some classifier



# Para saber más

- Classic Papers :
  - Abbeel & Ng ICML '04 . Apprenticeship Learning via Inverse Reinforcement Learning. Good introduction to inverse reinforcement learning
  - Ziebart et al. AAAI '08. Maximum Entropy Inverse Reinforcement Learning. A probabilistic method for inverse reinforcement learning
- Modern Papers :
  - Finn et al. ICML '16. et al. Guided Cost Learning. Sampling based method for handles unknown dynamics and deep reward functions
  - Wulfmeier arXiv '16 . Introduction MaxEnt IRL that Deep Maximum Entropy Inverse Reinforcement Learning. MaxEnt
  - Ho & inverse RL using deep reward functions Ermon NIPS '16. Generative Adversarial Imitation Learning. using generative adversarial networks Inverse RL method
  - Fu, Luo, Levine ICLR '18. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning

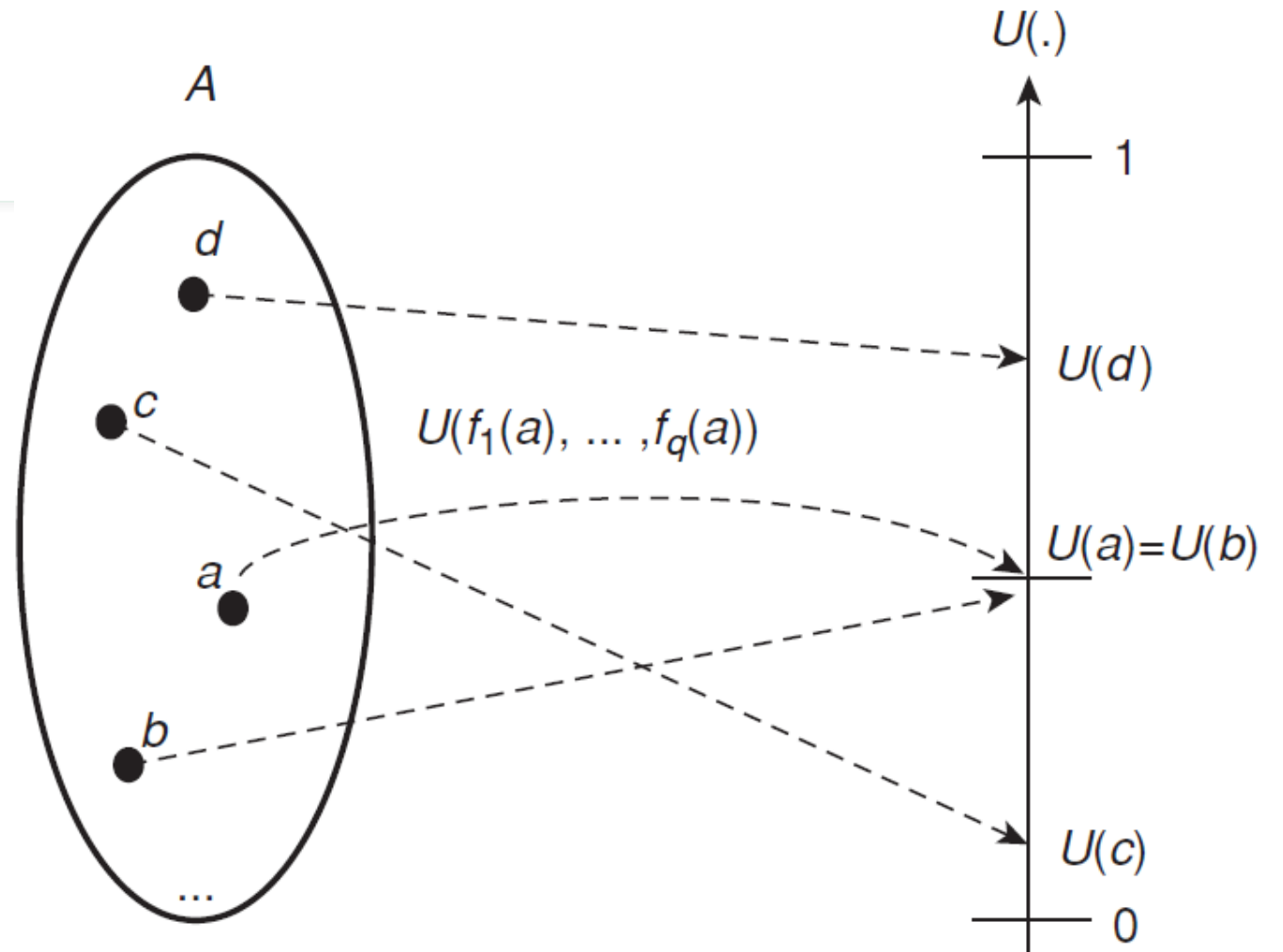
# Un ejemplo

Raw data	Price (€)	Customer review	Screen size (in)	Storage size (Gb)
SP1	429	4	4.65	32
SP2	649	4	3.5	64
SP3	459	5	4.3	32
SP4	419	3.5	4.3	16
SP5	519	4.8	4.7	16

- Supongamos que queremos comprar un celular. La decisión se realizará de acuerdo a precio, opiniones, tamaño de la pantalla y capacidad en memoria

# Teoría de utilidad multi-atributo

- La intención del enfoque es construir una forma en la que cada alternativa  $a$  esté asociada con un número real  $V(a)$
- Esto supone:
  - Las preferencias son completas:  $a \succ b, b \succ a, a \sim b$
  - Las preferencias e indiferencias son transitivas





# Los axiomas de racionalidad

- Ordenabilidad
  - $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
- Transitividad
  - $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
- Continuidad
  - $A \succ B \succ C \Rightarrow \exists p[p, A; 1 - p, C] \sim B$
- Sustituibilidad
  - $A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$
- Monotonicidad
  - $A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1 - p, B] \succcurlyeq [q, A; 1 - q, B])$

# Teoría de utilidad multi-atributo

- La forma más simple y más utilizada es el modelo aditivo

$$V(a) = \sum_{i=1}^m w_i v_i(a)$$

Donde

- $V(a)$  es el valor de la alternativa  $a$
- $v_i(a)$  es el puntaje que refleja la alternativa  $a$  en el criterio  $i$
- $w_i$  es el peso asignado que refleja la importancia del criterio  $i$



# Para la otra vez...

- Examen

The End.

