Aprendizaje Automático

- Las técnicas que hemos visto hasta ahora nos permiten crear sistemas que resuelven tareas que necesitan inteligencia
- La limitación de estos sistemas reside en que sólo resuelven los problemas ya previstos
- Sólo podemos considerar que un sistema es realmente inteligente si es capaz de observar su entorno y aprender de él
- La autentica inteligencia reside en adaptarse, tener capacidad de integrar nuevo conocimiento, resolver nuevos problemas, aprender de errores

Objetivos

- No se pretende modelar el aprendizaje humano
- Busca aumentar las capacidades de los programas de IA (SBC, planificación, TLN, búsqueda, ...):
  - Su límite está en el conocimiento que se les ha introducido
  - No resuelven problemas mas allá de esos límites
- Es imposible prever todos los problemas desde el principio
- Buscamos dar a programas la capacidad de adaptarse sin tener que ser reprogramados

¿Funciona?

Does Machine Learning Really Work? Tom Mitchell. AI Magazine 1997

¿Donde y para que se puede usar el aprendizaje automático?

- Tareas difíciles de programar (adquisición de conocimiento, reconocimiento de caras, voz, ...)
- Aplicaciones auto adaptables (interfaces inteligentes, spam filters, sistemas recomendadores, ...)
- Minería de datos/Descubrimiento de conocimiento (análisis de datos inteligente)

Tipos de aprendizaje

- **Aprendizaje inductivo:** Creamos modelos de conceptos a partir de *generalizar* ejemplos simples. Buscamos patrones comunes que expliquen los ejemplos.
- **Aprendizaje analítico o deductivo:** Aplicamos la deducción para obtener descripciones generales a partir de un ejemplo de concepto y su explicación.
- **Aprendizaje genético:** Aplica algoritmos inspirados en la teoría de la evolución para encontrar descripciones generales a conjuntos de ejemplos.
- **Aprendizaje conexionista:** Busca descripciones generales mediante el uso de la capacidad de adaptación de redes de neuronas artificiales

Aprendizaje inductivo

- Es el área con mas métodos
- **Objetivo:** Descubrir leyes generales o conceptos a partir de un número limitado de ejemplos (búsqueda de patrones comunes)
- Su funcionamiento reside en la observación de similaridades entre ejemplos
- Sus métodos se basan en el razonamiento inductivo

Razonamiento Inductivo vs Deductivo

- Razonamiento Inductivo
  - Obtiene conclusiones generales de información específica
  - El conocimiento obtenido es nuevo
  - No preserva la verdad (nuevo conocimiento puede invalidar lo obtenido)
  - No tiene una base teórica bien fundamentada
- Razonamiento Deductivo
  - Obtiene conocimiento mediante el uso de mecanismos bien establecidos
  - Este conocimiento no es nuevo (ya está presente implícitamente)
  - Nuevo conocimiento no invalida el ya obtenido
  - Se fundamenta en la lógica matemática

Aprendizaje inductivo

- Desde un punto de vista formal sus resultados no son válidos
- Suponemos que un número limitado de ejemplos representan las características de lo que queremos aprender
- Un solo contraejemplo puede invalidar el resultado
- ¡Gran parte del aprendizaje humano es inductivo!

Aprendizaje como búsqueda

- Podemos plantear el aprendizaje inductivo como una búsqueda heurística
- El objetivo es descubrir una función/descripción que resuma las características de un conjunto de ejemplos
- El espacio de búsqueda son todos los posibles conceptos que podemos construir
- Definiríamos el problema como:
  - **Espacio de búsqueda:** Lenguaje de descripción de conceptos $\Longrightarrow$ Conjunto de conceptos que podemos describir
  - **Operadores de búsqueda:** Operadores heurísticos que permiten explorar el espacio de conceptos posibles
  - **Función heurística:** Función de preferencia que guía en el proceso

Tipos de aprendizaje inductivo

- Aprendizaje Inductivo Supervisado
  - Para cada ejemplo se indica a que concepto pertenece
  - El aprendizaje se realiza por contraste entre conceptos
  - Un conjunto de heurísticas permitirán generar diferentes hipótesis
  - Existirá un criterio de preferencia (*sesgo*) que permitirá escoger la hipótesis mas *"adecuada"* a los ejemplos
  - **Resultado:** El concepto o conceptos que mejor describen a los ejemplos
- Aprendizaje Inductivo no Supervisado
  - No existe una clasificación de los ejemplos
  - Se busca descubrí la manera mas adecuada de particionar los ejemplos
  - El aprendizaje se guía por la similaridad/disimilaridad entre ejemplos
  - Existirán criterios heurísticos de preferencia que guíen la búsqueda
  - **Resultado:** Una partición de los ejemplos y una descripción de la partición

Árboles de decisión

- Podemos plantear el aprender un concepto como averiguar que preguntas hay que hacer para distinguirlo de otros
- Tomamos como representación un árbol que almacena esas preguntas
- Cada nodo del árbol es una pregunta sobre un atributo
- La búsqueda se realizará entre todos los posibles árboles
- Esta representación es equivalente a una FND (2^{2^n})

Árboles de decisión

- Para reducir el coste computacional hemos de imponer un sesgo (que tipo de conceptos preferimos)
- **Decisión:** árbol que represente la mínima descripción del concepto objetivo dadas las instancias ejemplo
- **Justificación:** Un árbol así será el que mejor podrá predecir nuevas instancias (reducimos la probabilidad de que haya condiciones innecesarias)
- **Navaja de Occam**: "Dadas dos teorías igualmente predictivas es preferible la mas simple"

Algoritmos de árboles de decisión

- El primer algoritmo de árboles de decisión fue **ID3** (Quinlan 1986)
- Se encuadra dentro de la familia de algoritmos Top Down Induction Decision Trees (TDIDT)
- ID3 realiza una búsqueda mediante Hill-Climbing en el espacio de árboles
- Se elige en cada nivel del árbol un atributo y se particiona el conjunto de ejemplos según sus valores, se repite recursivamente el proceso con cada partición
- La elección del atributo se hace mediante una función heurística

Teoría de la Información

- La teoría de la información estudia entre otras cosas los mecanismos de codificación de mensajes y el coste de su transmisión
- Si definimos un conjunto de mensajes $M = \{m_1, m_2, ..., m_n\}$, cada uno de ellos con una probabilidad $P(m_i)$, podemos definir la cantidad de información (I) contenida en un mensaje de M como:

$$I(M) = \sum_{i=1}^{n} -P(m_i)log(P(m_i))$$

- Este valor se puede interpretar como la información necesaria para distinguir entre los mensajes de M (Cuantos bits de información son necesarios para codificarlos)

Cantidad de Información como Heurística

- Podemos hacer la analogía con la codificación de mensajes suponiendo que las clases son los mensajes y la proporción de ejemplos de cada clase su probabilidad
- Podemos ver un árbol de decisión como la codificación que permite distinguir entre las diferentes clases
- Buscamos el mínimo código que distingue entre las clases
- Cada atributo se deberá evaluar para decidir si se le incluye en el código
- Un atributo será bueno si permite distinguir mejor entre las diferentes clases

Cantidad de Información como Heurística

- En cada nivel del árbol debemos evaluar que atributo permite minimizar el código (reduce el tamaño del árbol)
- Este atributo será el que haga que la cantidad de información que quede por cubrir sea la menor
- La elección de un atributo debería hacer que los subconjuntos que genera el atributo sean mayoritariamente de una clase
- Necesitamos una medida de la cantidad de información que no cubre un atributo (medida de Entropía, E)

Ganancia de información

- Cantidad de Información ($\mathcal{X}$ - ejemplos, $\mathcal{C}$ - clasificación)

$$I(\mathcal{X},\mathcal{C}) = \sum_{\forall c_i \in \mathcal{C}} -\frac{\sharp c_i}{\sharp \mathcal{X}} \; log(\frac{\sharp c_i}{\sharp \mathcal{X}})$$

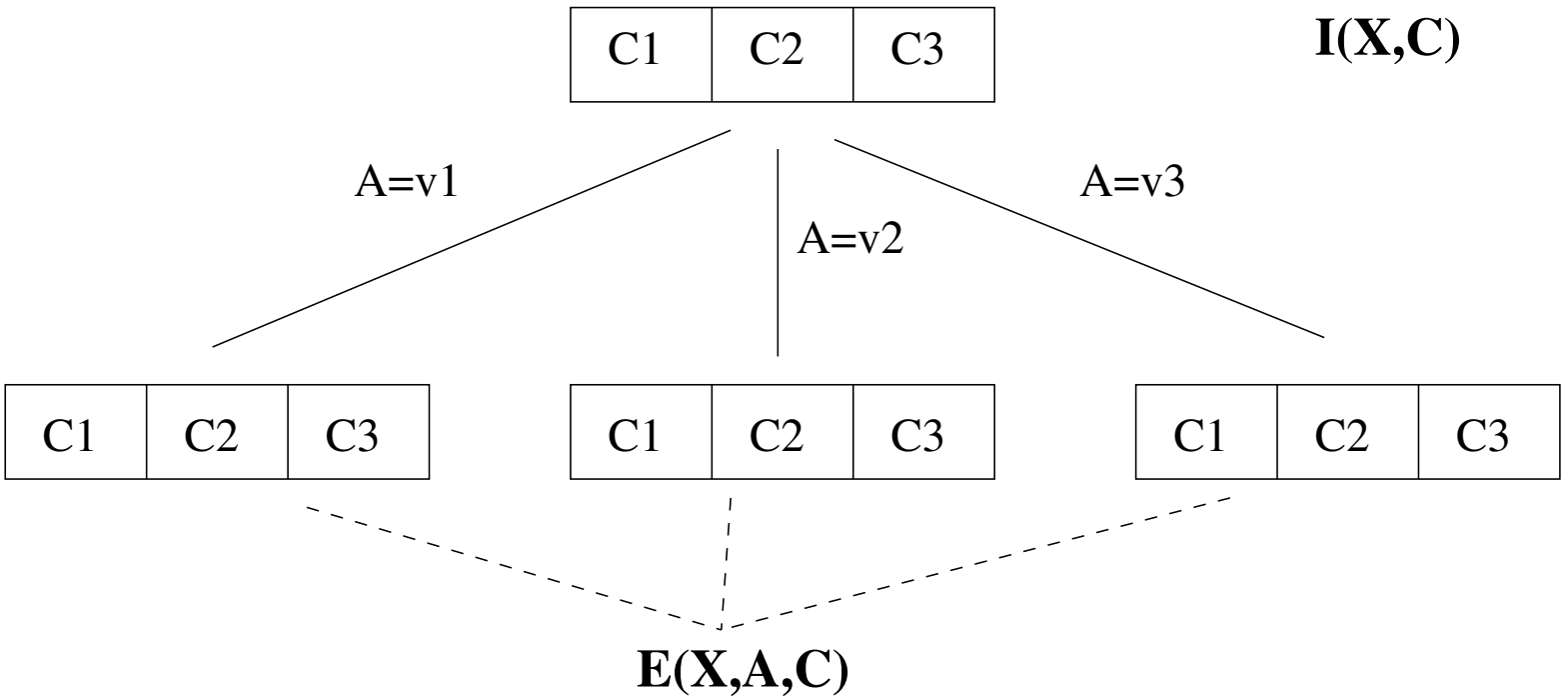
- Entropía (A - atributo, $[A(x) = v_i]$ - ejemplos con valor v_i)

$$E(\mathcal{X},A,\mathcal{C}) = \sum_{\forall v_i \in A} \frac{\sharp[A(x) = v_i]}{\sharp \mathcal{X}} \; I([A(x) = v_i],\mathcal{C})$$

- Ganancia de Información

$$G(\mathcal{X},A,\mathcal{C}) = I(\mathcal{X},\mathcal{C}) - E(\mathcal{X},A,\mathcal{C})$$

Ganancia de información



$$G(X,A,C) = I(X,C) - E(X,A,C)$$

Algoritmo ID3

Algoritmo: ID3 ($\mathcal{X}$: Ejemplos, $\mathcal{C}$: Classificación, $\mathcal{A}$: Atributos)

si *todos los ejemplos son de la misma clase*
entonces
  retorna *una hoja con el nombre de la clase*
sino
  Calcular la función de cantidad de información de los ejemplos (**I**)
  para cada *atributo en* $\mathcal{A}$ **hacer**
    Calcular la función de entropia (**E**) y la ganancia de información (**G**)
  Escoger el atributo que maximiza **G** (**a**)
  Eliminar **a** de la lista de atributos ($\mathcal{A}$)
  Generar un nodo raíz para el atributo **a**
  para cada *Partición generada por los valores* v_i *del atributo* **a hacer**
    Árbol$_i$=ID3($\mathcal{X}$(a=v$_i$), $\mathcal{C}$(a=v$_i$),$\mathcal{A}$-*a*)
    Generar una nueva rama con **a=v**$_i$ y Árbol$_i$
  retorna *El nodo raíz para* ***a***

Ejemplo (1)

Tomemos el siguiente conjunto de ejemplos

Ej.	Ojos	Cabello	Estatura	Clase
1	Azules	Rubio	Alto	+
2	Azules	Moreno	Medio	+
3	Marrones	Moreno	Medio	−
4	Verdes	Moreno	Medio	−
5	Verdes	Moreno	Alto	+
6	Marrones	Moreno	Bajo	−
7	Verdes	Rubio	Bajo	−
8	Azules	Moreno	Medio	+

Ejemplo (2)

$$
\begin{aligned}
I(X, C) &= -1/2 \cdot log(1/2) - 1/2 \cdot log(1/2) = 1 \\
E(X, ojos) &= (azul)\ 3/8 \cdot (-1 \cdot log(1) - 0 \cdot log(0)) \\
&+ (marrones)\ 2/8 \cdot (-1 \cdot log(1) - 0 \cdot log(0)) \\
&+ (verde)\ 3/8 \cdot (-1/3 \cdot log(1/3) - 2/3 \cdot log(2/3)) \\
&= 0{,}344 \\
E(X, cabello) &= (rubio)\ 2/8 \cdot (-1/2 \cdot log(1/2) - 1/2 \cdot log(1/2)) \\
&+ (moreno)\ 6/8 \cdot (-1/2 \cdot log(1/2) - 1/2 \cdot log(1/2)) \\
&= 1 \\
E(X, estatura) &= (alto)\ 2/8 \cdot (-1 \cdot log(1) - 0 \cdot log(0)) \\
&+ (medio)\ 4/8 \cdot (-1/2 \cdot log(1/2) - 1/2 \cdot log(1/2)) \\
&+ (bajo)\ 2/8 \cdot (0 \cdot log(0) - 1 \cdot log(1)) \\
&= 0{,}5
\end{aligned}
$$

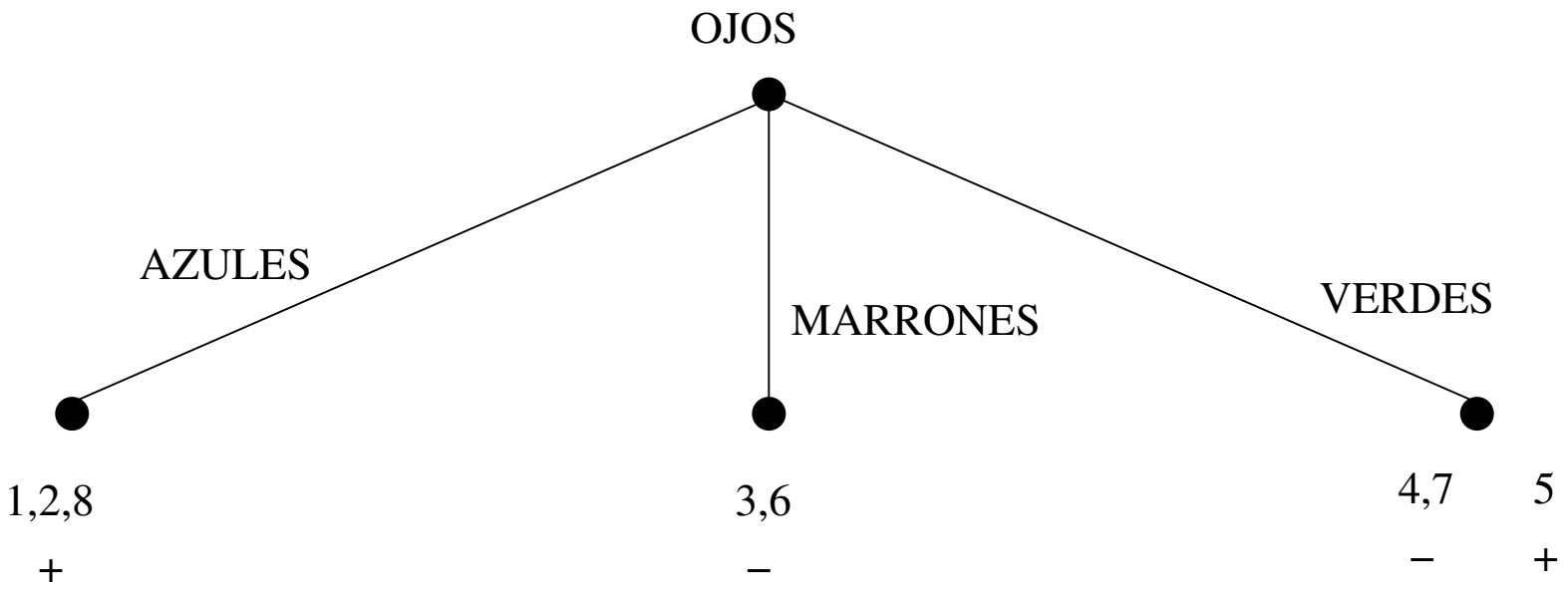
Ejemplo (3)

Como podemos comprobar, es el atributo **ojos** el que maximiza la función.

$$\begin{aligned} G(X, \textit{ojos}) &= 1 - 0{,}344 = 0{,}656 * \\ G(X, \textit{cabello}) &= 1 - 1 = 0 \\ G(X, \textit{estatura}) &= 1 - 0{,}5 = 0{,}5 \end{aligned}$$

Ejemplo (4)

Este atributo nos genera una partición que forma el primer nivel del árbol.



Ejemplo (5)

Ahora solo en el nodo correspondiente al valor **verdes** tenemos mezclados objetos de las dos clases, por lo que repetimos el proceso con esos objetos.

Ej.	Cabello	Estatura	Clase
4	Moreno	Medio	$-$
5	Moreno	Alto	$+$
7	Rubio	Bajo	$-$

Ejemplo (6)

$$
\begin{aligned}
I(X, C) &= -1/3 \cdot log(1/3) - 2/3 \cdot log(2/3) = 0{,}918 \\
E(X, cabello) &= (rubio)\ 1/3 \cdot (0 \cdot log(0) - 1 \cdot log(1)) \\
&+ (moreno)\ 2/3 \cdot (-1/2 \cdot log(1/2) - 1/2 \cdot log(1/2)) \\
&= 0{,}666 \\
E(X, estatura) &= (alto)\ 1/3 \cdot (0 log(0) - 1 \cdot log(1)) \\
&+ (medio)\ 1/3 \cdot (-1 \cdot log(1) - 0 \cdot log(0)) \\
&+ (bajo)\ 1/3 \cdot (0 \cdot log(0) - 1 \cdot log(1)) \\
&= 0
\end{aligned}
$$

Ejemplo (7)

Ahora el atributo que maximiza la función es **ojos**.

$$\begin{aligned} G(X, \mathit{cabello}) &= 0{,}918 - 0{,}666 = 0{,}252 \\ G(X, \mathit{estatura}) &= 0{,}918 - 0 = 0{,}918* \end{aligned}$$

Ejemplo (8)

El árbol resultante es ya totalmente discriminante.