



Universidad de Granada

decsai.ugr.es

Inteligencia Artificial en Telecomunicaciones

Máster en Ingeniería de Telecomunicaciones

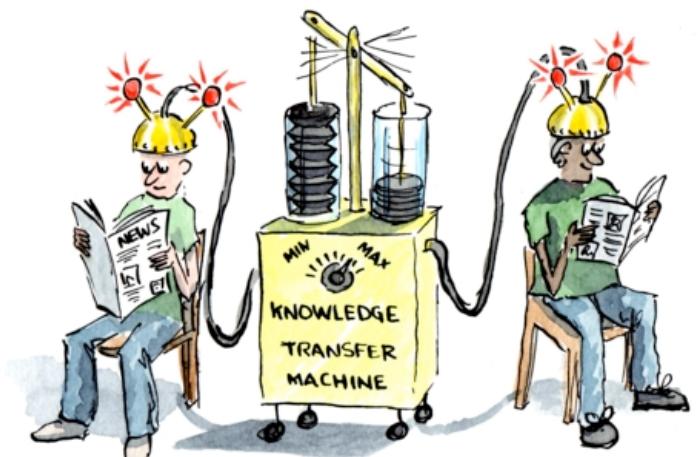
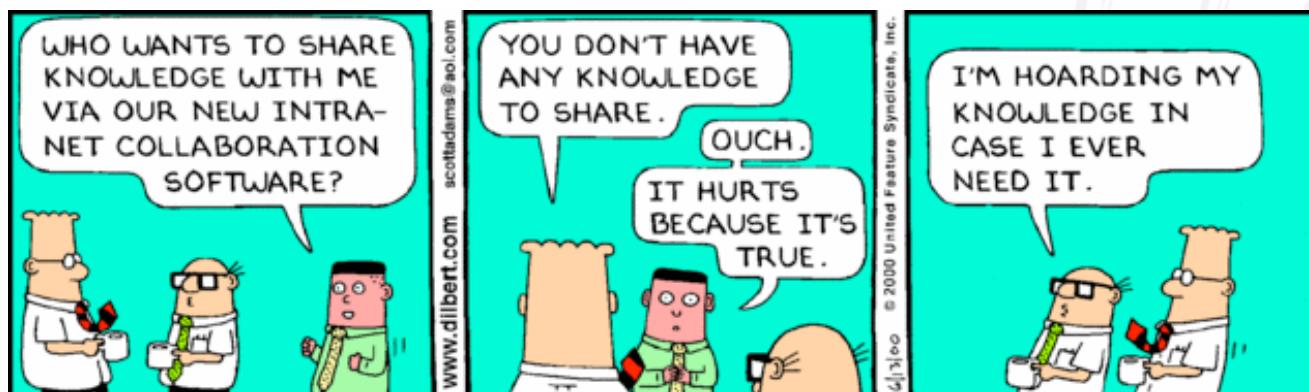
Tema 4: Aspectos Básicos del Aprendizaje Automático



**Departamento de Ciencias de la
Computación e Inteligencia Artificial**

La adquisición del Conocimiento

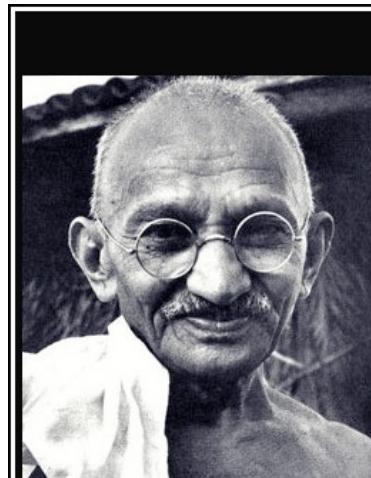
- ▶ En los **sistemas basados en conocimiento** es preciso realizar una tarea de **adquisición del conocimiento**, donde tenemos que obtener las reglas de funcionamiento del sistema a partir de los conocimientos de los expertos en un determinado área.
- ▶ La adquisición de conocimiento es una **tarea compleja** por diversos motivos, entre otros:
 - El conocimiento puede ser muy **difícil de formalizar**.
 - Los expertos pueden ser **poco colaboradores** o no colaborar en absoluto.
 - Puede que **no tengamos expertos** en el tema que estamos estudiando.



Aprendizaje

El aprendizaje es una capacidad fundamental de la inteligencia humana que nos permite:

- ▶ **Adaptarnos a cambios** de nuestro entorno.
- ▶ **Desarrollar habilidades** o mejorarlas.
- ▶ **Adquirir experiencia** en nuevos dominios.



Live as if you were to die tomorrow. Learn as if you were to live forever.

(Mahatma Gandhi)

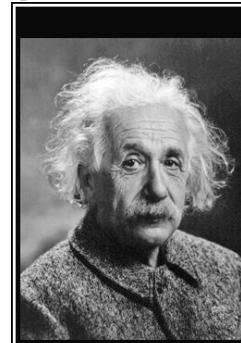
izquotes.com



"Once I learn how to use Google, isn't that all the education I really need?"

Aprendizaje Automático (Machine Learning)

- El Aprendizaje Automático son **programas que mejoran su comportamiento con la experiencia**.
- El aprendizaje **modifica el mecanismo de decisión** del agente para mejorar su comportamiento y permite **adquirir nuevo conocimiento**.
- Es **esencial en entornos desconocidos**.
- En los problemas de Búsquedas en espacio de estados y en los sistemas expertos tienen su **límite en el conocimiento** que se les ha proporcionado y, por tanto, **no resuelven problemas más allá de esos límites**.
- Normalmente se aprende a partir de datos.



No lo sé, procuro no cargar mi memoria con datos que puedo encontrar en cualquier manual, ya que el gran valor de la educación no consiste en atiborrarse de datos, sino en preparar al cerebro a pensar por su propia cuenta y así llegar a conocer algo que no figure en los libros.

(Albert Einstein)

akifrases.com

Ventajas Aprendizaje Automático

- ▶ No es preciso contar con **expertos** costosos o poco colaboradores.
- ▶ El proceso manual **se automatiza**, haciéndolo más simple, rápido y barato.
- ▶ Los algoritmos utilizados en una aplicación pueden ser reutilizados en otras. De este modo, es posible utilizar **algoritmos de aprendizaje** similares para la construcción de reglas de asesoría crediticia, diagnóstico médico, o reparación de locomotoras eléctricas. Es decir, los algoritmos no son dependientes del dominio de aplicación, aunque el conocimiento sí lo sea.
- ▶ Se **aprovecha el conocimiento implícito** disponible en forma de ejemplos resueltos manualmente por expertos a lo largo de muchos años.
- ▶ No obstante los sistemas de aprendizaje no suelen ser tan efectivos como los sistemas basados en conocimiento, al menos al principio, pero pueden evolucionar y mejorar con el tiempo, ej.: spam.



¿Qué aprender?

- ▶ **Tareas difíciles de programar** (reconocimiento de caras, voz, ...).
- ▶ **Aplicaciones auto adaptables** (interfaces inteligentes, spam killers, sistemas recomendadores, ...).
- ▶ **Minería de datos** (análisis de datos inteligente).



Definición Formal de Aprendizaje Automático

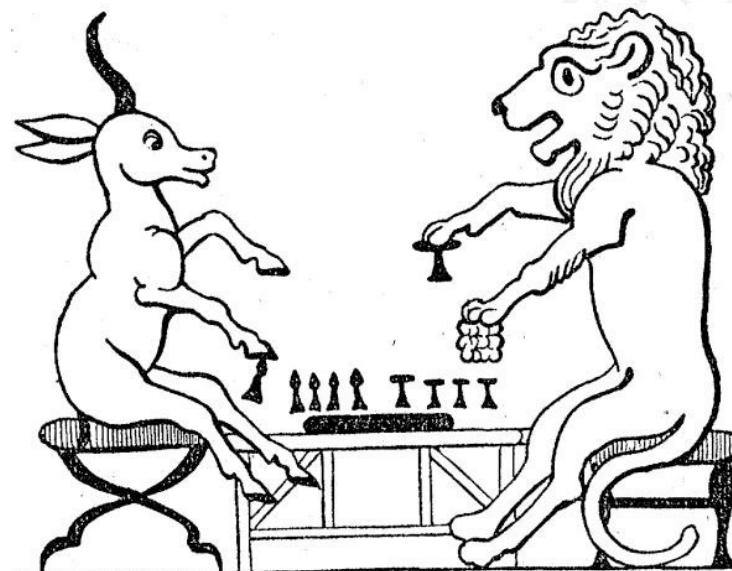
- Un programa de ordenador se dice que aprende de la experiencia E con respecto a alguna clase de tareas T y a alguna medida de comportamiento P, si su comportamiento en tareas de T, medido a través de P, mejora con la experiencia E.



Ejemplos

Aprendizaje de damas:

- ▶ T: jugar a las damas.
- ▶ P: % de juegos ganados.
- ▶ E: partidas jugadas contra una copia de si mismo.



Egyptian Draughts. (From a papyrus in the British Museum.)

Ejemplos

Aprendizaje de reconocimiento de caracteres de escritos a mano:

- ▶ T: reconocer y clasificar palabras escritas a mano a través de imágenes.
- ▶ P: % de palabras correctamente clasificadas.
- ▶ E: una base de datos de palabras escritas a mano con su correspondiente clasificación.

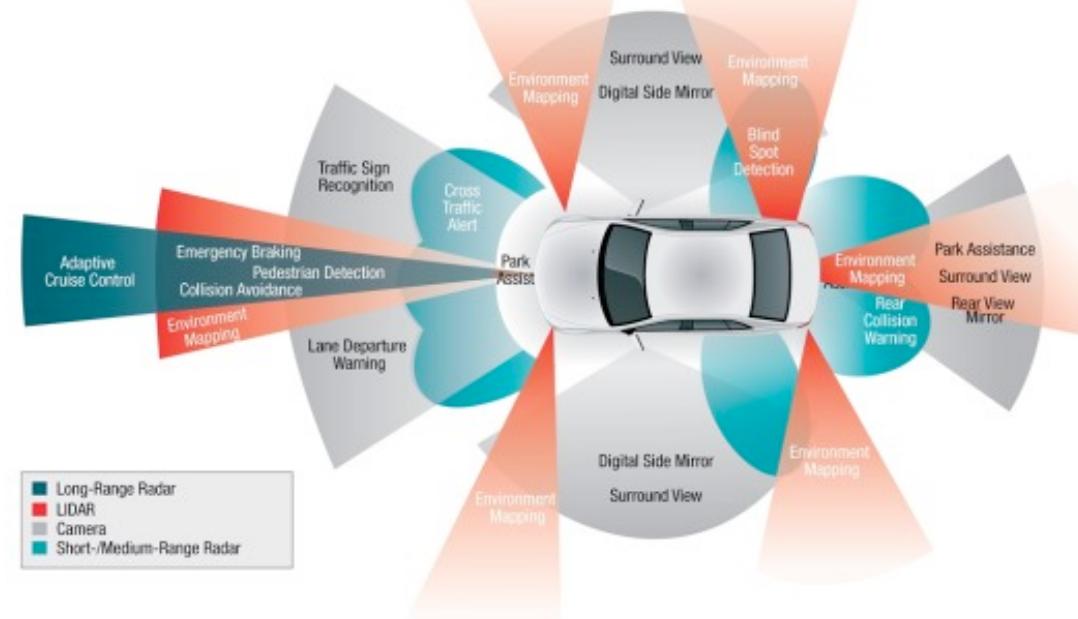
Actual screengrab of reCAPTCHA interface showing a street number to be translated:



Ejemplos

Aprendizaje de un sistema para conducir:

- ▶ T: conducir por una carretera usando sensores de visión.
- ▶ P: distancia promedio antes de que se produzca un error (juzgado por un humano).
- ▶ E: secuencia de imágenes y comandos de conducción registrados en la observación de un conductor humano.



Ejemplos

Monitorización y análisis del tráfico en una red

- ▶ T: Clasificación del tráfico de red.
 - ▶ P: Porcentaje del tráfico bien clasificado.
 - ▶ E: Conjunto de flujos de tráfico de red etiquetados: {descriptores del tráfico de red, aplicación}.
-
- ▶ T: Detección de ataques día cero.
 - ▶ P: Porcentajes de falsas alarmas y detecciones de ataques.
 - ▶ E: Conjunto de flujos de tráfico de red libres de ataques.
-
- ▶ T : Modelado y predicción de la QoE
 - ▶ P : Tasa de niveles de QoE correctamente predichos,
 - ▶ E : Conjunto de tests subjetivos: {descriptores de QoS/aplicaciones, nivel de QoE}

Ejemplos del sector financiero y seguros

- ▶ Estimación de Riesgos en la Concesión de Seguros de Crédito
- ▶ Detección y Control de Fraude en el Uso de Tarjetas de Crédito
- ▶ Segmentación de Clientes de Entidades Financieras.
- ▶ Trading algorítmico.
- ▶ Selección de start-ups para invertir.

Credit Score



Ejemplos del sector sanitario y farmacéutico

- ▶ Predicción de Ventas de Productos Farmacéuticos
- ▶ Diagnóstico de Accidentes Cerebro-Vasculares Agudos
- ▶ Diagnóstico de enfermedades o predicción de supervivencia usando datos de expresión genética
- ▶ Supervisión de Calidad del precultivo en el Cultivo Industrial de Antibióticos
- ▶ Supervisión de la Evolución de Cultivos en la Fabricación de Antibióticos



Ejemplos del sector industrial

- ▶ Optimización de Centrales Eléctricas
- ▶ Control de Trenes de Laminado en la Industria del Acero
- ▶ Optimización de Altos Hornos
- ▶ Optimización de la Producción de Cartón en la Industria Papelera
- ▶ Gestión de Alarmas en Plantas Petroquímicas
- ▶ Control de Calidad en la Fabricación de Electrodomésticos
- ▶ Optimización del Proceso de Producción de Cemento
- ▶ Control de Calidad de Materiales Fabricados Industrialmente
- ▶ Control Adaptativo para Optimización de Trayectorias de Robots Industriales
- ▶ Control de Calidad en la Fabricación de Cajas de Cambio en la Industria del Automóvil
- ▶ Diagnóstico Automático de Componentes de Sistemas hi-fi para Automóviles

Ejemplos en el sector de las telecomunicaciones

- ▶ Detección y predicción de fallos en las redes.
- ▶ Detección y predicción de intrusiones/ataques.
- ▶ Recomendación de servicios: Análisis de patrones de clientes que permitan detectar los productos mas rentables que satisfagan sus necesidades (Publicidad personalizada).
- ▶ Personalización de servicios a usuarios.
- ▶ Etiquetado automático de medios.
- ▶ Retención de clientes por medio de prevención temprana de insatisfacciones.
- ▶ Clasificación del tráfico de red.
- ▶ Estimación de QoE en servicios multimedia.

Fases del Descubrimiento del Conocimiento

► **Selección:** Consiste en ver qué datos son relevantes y no redundantes.

Por ejemplo, datos personales como la altura o la práctica de deportes pueden ser poco relevantes para un sistema de asesoría de crédito, pero pueden ser esenciales en un sistema de diagnóstico médico.

► **Procesamiento:** Los datos pueden estar almacenados en varios sitios o puede haber valores desconocidos/perdidos.

Por ejemplo, los datos de todos los miembros de una unidad familiar se pueden unir en una sola tabla a fin de valorar a toda la unidad de cara a la concesión del crédito.

► **Transformación:** Los datos que se tienen pueden no ser adecuadas para el aprendizaje, y entonces es necesario aplicar alguna transformación.

Por ejemplo, los múltiples créditos que tiene un solicitante se pueden colapsar en una sola entrada que combina dichos datos en un solo valor de deuda a amortizar anualmente.

► **Aprendizaje:** Se realiza la tarea de aprendizaje automático.

Por ejemplo, en esta fase se obtienen las reglas de concesión de préstamos a partir de la tabla de datos generada anteriormente.

► **Interpretación y Evaluación:** Es necesario evaluar el modelo aprendido de cara a su eficacia y eficiencia y también interpretar sus resultados.

Por ejemplo, se puede evaluar la eficacia del asesor crediticio sobre otros ejemplos reservados para dicha evaluación, no usados en el aprendizaje, contabilizando el porcentaje de aciertos.

Retos del aprendizaje automático

- ▶ **Tamaño creciente de las bases de datos.** Cada dos días, la humanidad crea tanta información como lo había hecho la civilización hasta 2003. Big Data. En el 2007, sólo 0.007% de la información estaba en papel.
- ▶ **Sobreajuste.** Un modelo puede tener poca capacidad de generalización.
- ▶ **Evolución constante de los datos.** Las poblaciones evolucionan en el tiempo por lo que se requiere que los algoritmos de aprendizaje sean capaces de adaptarse a esta evolución con poco o ningún esfuerzo, sin más que reentrenar el sistema periódicamente, o actualizar el clasificador ya generado.
- ▶ **Bases de datos incompletas, con datos erróneos o con ruido.** Por ejemplo, se estima que el 20% de los datos del censo de los EE.UU. son erróneos en algunos campos.
- ▶ **Modelos interpretables.** Es deseable que los modelos obtenidos sean comprensibles para el ser humano, que es usuario final de un sistema de aprendizaje.
- ▶ **Integrar conocimiento experto.** Si podemos acceder a un experto siempre es conveniente usar su conocimiento.

Estrategias de Aprendizaje

- ▶ Aprendizaje memorístico.
- ▶ Aprendizaje a través de consejos.
- ▶ Aprendizaje en la resolución de problemas.
- ▶ Aprendizaje a partir de ejemplos: inducción.
- ▶ Aprendizaje basado en explicaciones.
- ▶ Aprendizaje a través de descubrimiento.
- ▶ Aprendizaje por analogía.

Tipos de Aprendizaje

Uno de los puntos clave para el aprendizaje es el tipo de realimentación disponible en el proceso:

- ▶ **Aprendizaje supervisado:** Aprender una función a partir de ejemplos de sus entradas y salidas.

Por ejemplo, la concesión de créditos a los clientes. En este caso, los datos de entrada son las características del cliente en el momento de solicitar el préstamo: saldo, deudas, ingresos, garantías, etc. y la salida es la variable que se desea predecir: si el cliente devolvió o no el préstamo. En una primera etapa, se utilizarán los datos históricos de préstamos previos para que el sistema “aprenda” a decidir sobre la solvencia del cliente.

- ▶ **Aprendizaje no supervisado:** Aprender o extraer conclusiones a partir de patrones de entradas para los que no se especifican los valores de su salidas.

Por ejemplo, deducir a partir de los datos históricos que la mayoría de las llamadas telefónicas que hace un determinado cliente se dirigen a un número reducido de números o agrupar los clientes por el consumo.

- ▶ **Aprendizaje por refuerzo:** Aprender a partir del refuerzo que devuelve el entorno.

Por ejemplo, si un robot planea una trayectoria y choca con un objeto, recibe una penalización, y si no hay colisión recibe una gratificación, así el robot tendrá en cuenta estos estímulos al planear las trayectorias.

Ejemplo

- ▶ Supermercado: se desea clasificar los clientes entre buenos y malos clientes.
- ▶ Base de datos: información acerca de los clientes y forma de pago de los mismos.

Id	Casado	N-hijos	Sexo	Pago	Buen-cliente
1	sí	3	m	Tarjeta	sí
2	no	0	h	Tarjeta	sí
3	no	1	m	Efectivo	no
4	sí	4	m	Crédito	sí
5	sí	2	h	Efectivo	no
6	no	1	m	Tarjeta	no
7	no	0	h	Efectivo	sí
8	no	0	h	Crédito	sí
9	no	1	h	Tarjeta	no
10	sí	4	m	Crédito	sí

- ▶ Si $N\text{-hijos} > 2$ ENTONCES $\text{Buen-cliente} = \text{sí}$
- ▶ Si $\text{Casado} = \text{no}$ Y $\text{sexo} = \text{h}$ Y $N\text{-hijos} = 0$ ENTONCES $\text{Buen-Cliente} = \text{sí}$

Ejemplo

- ▶ Ejemplo: modelado de la probabilidad de fallo de una máquina.
- ▶ Clase: la máquina fallará / la máquina no fallará.
- ▶ Atributos: conjunto de medidas:
 - Temperatura.
 - Nivel de vibraciones.
 - Horas de funcionamiento.
 - Meses desde la última revisión.
- ▶ Instancias: ejemplos pasados (situaciones conocidas).
- ▶ Hipótesis: relación entre las medidas y la probabilidad de fallo:
- ▶ SI nivel_vibraciones = alto Y temperatura = alta ENTONCES fallará.

Terminología

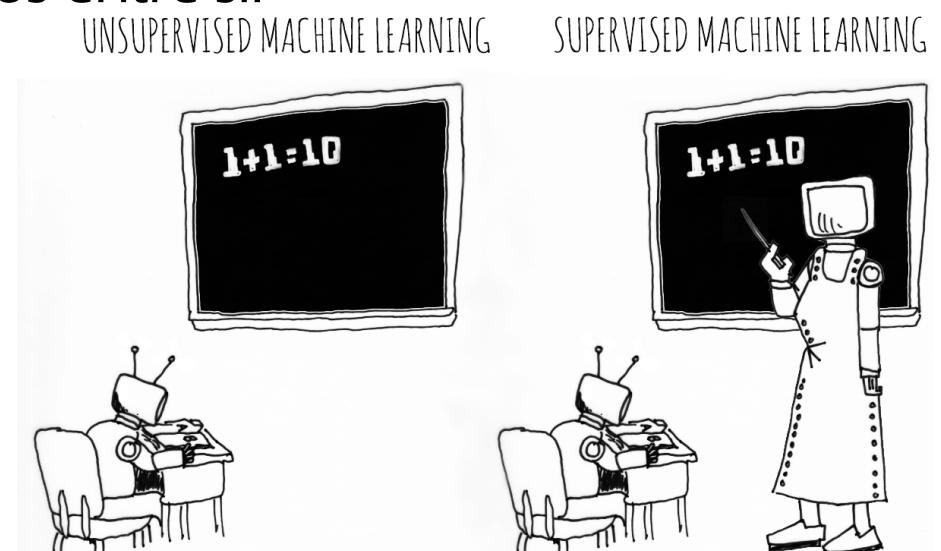
					Entrada(s)	Salida(s)
Ejemplo (o individuo)	Temp	Tiempo	Día	Ropa	Elección	
	26°C	Sol	S	Informal	Andar	
	2°C	Nieve	L	Informal	Conducir	
	17°C	Nublado	M	Informal	Andar	

Variable, atributo o dato

- ▶ **Clase** o variable a clasificar: Variable de salida.
- ▶ **Atributos**, variables predictoras, variables inductoras o características: Variables de entrada.
- ▶ Variable **nominal o categórica**: Variable discreta.
- ▶ Variable **numérica o continua**: Variable no discreta o con muchos estados (p.ej.: edad). Atributos: Instancias: ejemplos pasados (situaciones conocidas).
- ▶ Dependiendo si la clase es discreta o no tenemos:
 - Problemas de **regresión**: Si la clase es continua.
 - Problemas de **clasificación**: Si la clase es discreta.

Aprendizaje supervisado vs no supervisado

- ▶ En **clasificación supervisada** el conjunto de variables a considerar será $V = X \cup \{C\}$. Donde **C** es la clase y las variables **X** los atributos. El objetivo es describir la variable clase en función de los atributos.
- ▶ En **clasificación no supervisada** partiendo de un conjunto de variables $V = X$, lo que se suele hacer es asignar cada caso a un grupo (**clúster** o agrupamiento) **C**, es decir, el objetivo es descubrir una estructura de clases en los datos, de tal manera que los casos que pertenezcan a una misma clase o grupo presenten una gran homogeneidad, mientras que los casos que pertenezcan a distintos agrupamientos o clasificaciones deben ser muy heterogéneos entre sí.

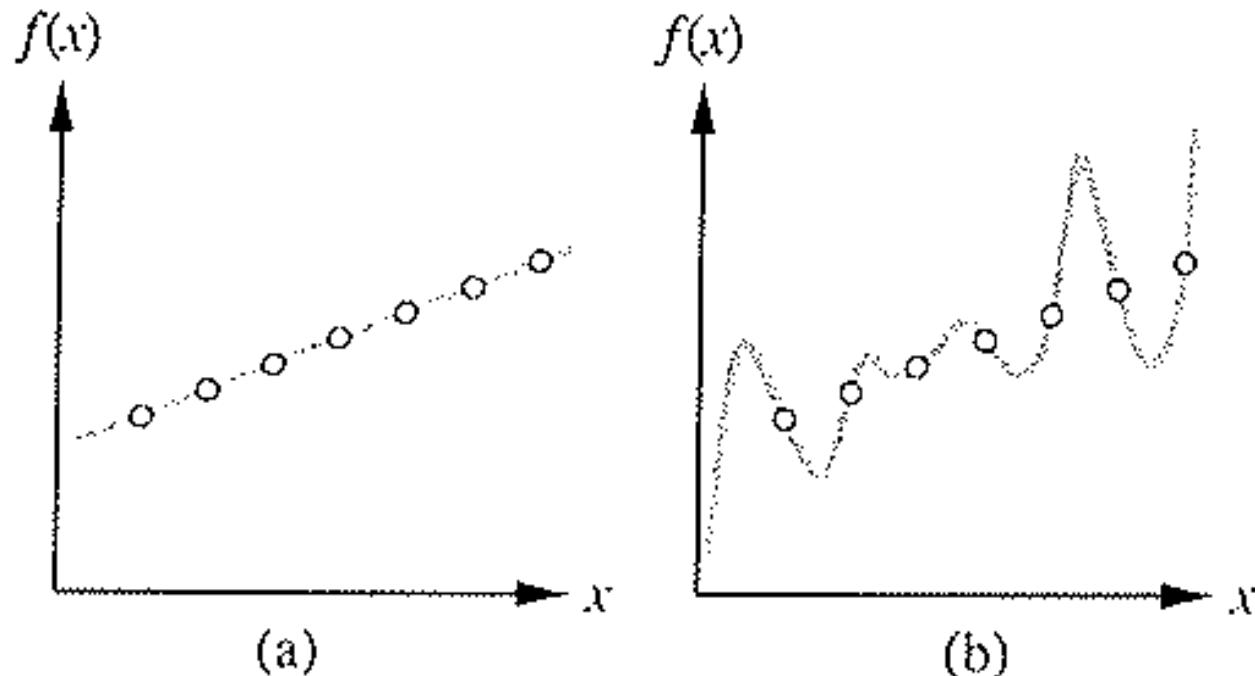


Aprendizaje Inductivo

- ▶ Aprender a partir de ejemplos en aprendizaje supervisado también se le denomina **aprendizaje inductivo**.
- ▶ Dentro del método científico, la el **razonamiento inductivo** es el conocimiento que pasa de lo particular a lo global. Basándose en el número de repeticiones o experimentos que se hace.
- ▶ En clasificación/regresión **inducimos hipótesis** a partir de los ejemplos.
- ▶ Una hipótesis la consideraremos **cierta si no encontramos algún contrajemplo**.
- ▶ Una **hipótesis estará bien generalizada** si puede predecir ejemplos que no se conocen.
- ▶ El aprendizaje supervisado podemos también verlo como que cada ejemplo es del tipo **($\mathbf{x}, f(\mathbf{x})$)** donde **\mathbf{x}** es el conjunto de atributos y **$f(\mathbf{x})$** es la clase. Por tanto nuestro objetivo es encontrar la función **f** o, mejor dicho, encontrar una hipótesis **h** tal que **$h=f$** para todo ejemplo del conjunto de datos.

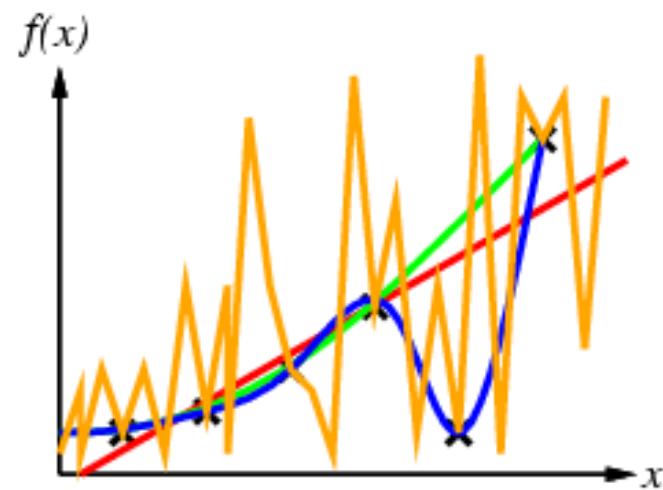
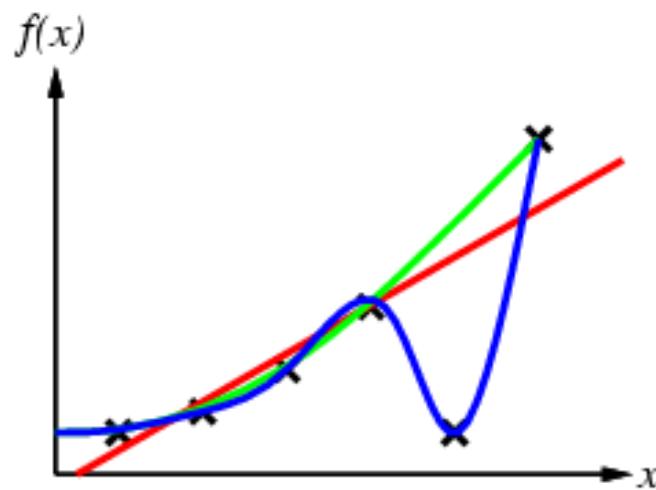
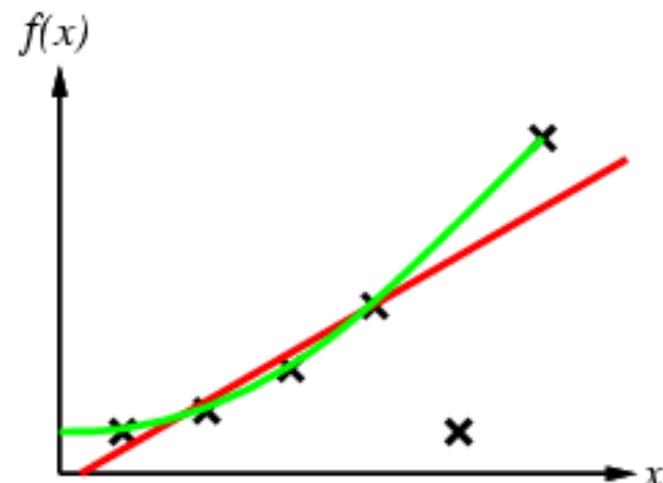
Múltiples hipótesis

- ▶ ¿Cómo elegir entre múltiples hipótesis que sean consistentes con los datos?.
- ▶ Una hipótesis se dice **consistente** si satisface todos los datos.



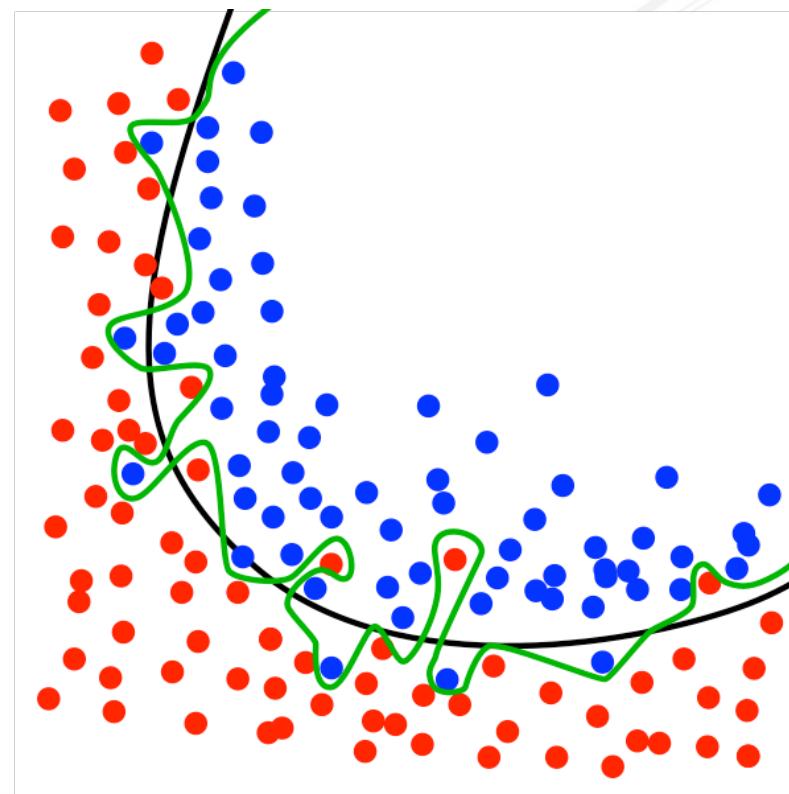
- ▶ **Navaja de Ockham:** "en igualdad de condiciones, la explicación más sencilla suele ser la más probable".
- ▶ Por tanto, se elegirá la hipótesis más simple consistente con los datos.
- ▶ Dicho de otra forma: KISS (keep it simple, stupid!).

Múltiples hipótesis



Sobreajuste

- ▶ El **sobreajuste** (en inglés, overfitting) es el resultado de un sobreaprendizaje, donde el método de aprendizaje se ha ajustado mucho a los datos. Es lo contrario a la **generalización**. Es un problema pues clasifica muy bien los datos usados para el aprendizaje pero muy mal los casos nuevos.
- ▶ Normalmente, soluciones más complejas se ajustan más a los datos.



Evaluación de Clasificadores

- ▶ A la hora de evaluar un clasificador se hará teniendo en cuenta distintos criterios. Por ejemplo, tiempo que tardamos en construirlo, la interpretabilidad del modelo obtenido, la sencillez del modelo (cuanto más sencillo, mayor capacidad de abstracción) o diferencias respecto al modelo original; pero al que más atención se le presta es a la **precisión del clasificador** (o, a la inversa, la tasa de error) que posee.
- ▶ La precisión de un clasificador (*accuracy*, en inglés) la podemos ver como el número de casos clasificados correctamente entre el número total de elementos.

$$\text{precisión} = \frac{\text{número de aciertos}}{\text{número de casos}}$$

Estimación de la precisión

- ▶ **Estimación por resustitución** (resubstitution estimate) o **error aparente**: este modo, el más simple, consiste en utilizar los mismos datos que se han utilizado para construir el clasificador para ver cuántos predice correctamente.
- ▶ **Holdout** o **entrenamiento y test**: se basa en partir el conjunto de datos aleatoriamente en dos grupos. El denominado conjunto de entrenamiento (normalmente 2/3 del número total de casos) con el cual se construye el clasificador y el conjunto de test o validación (construido con el 1/3 restante) usado para estimar la precisión del clasificador.
- ▶ **Remuestreo** (random subsampling): es una variación del sistema anterior donde se realizan diferentes particiones de los conjuntos de entrenamiento y test. Obteniéndose la precisión del clasificador a partir de la media obtenida en los distintos conjuntos de test.

Estimación de la precisión

- ▶ **Validación cruzada de k-hojas**(k-fold cross-validation): se puede ver como una generalización del criterio de remuestreo. Hacemos k particiones del conjunto de datos mutuamente excluyentes y de igual tamaño. $k - 1$ conjuntos se utilizan para construir el clasificador y se valida con el conjunto restante. Este paso se efectúa k veces y la estimación de la precisión del clasificador se obtiene como la media de las k mediciones realizadas.
- ▶ **Dejar-uno-fuera** (leave-one-out): es un caso particular de la validación cruzada, donde se hacen tantas particiones como casos tenga el conjunto de datos. De esta forma los conjuntos de validación tienen un sólo caso y los de entrenamiento todos los casos menos ese en particular. Al tener que construir el clasificador tantas veces como casos tenga el conjunto de datos se hace bastante costoso en tiempo
- ▶ **Bootstrapping**: En bootstrapping (o bootstrap) para una muestra de tamaño n se genera el conjunto de entrenamiento con n casos mediante **muestreo con reemplazamiento**, es decir, cogemos un caso de forma aleatoria del conjunto de datos para el conjunto de entrenamiento y lo volvemos a dejar en el conjunto de datos, de esta forma puede haber casos repetidos en el conjunto de entrenamiento. El conjunto de test se genera cogiendo aquellos casos que no estén en el de entrenamiento. Esto se repite muchas veces.

Evaluación del clasificador

- ▶ Algunas veces es interesante no sólo conocer la precisión del clasificador, sino que es importante saber el sentido en el que se equivoca. Por ejemplo, no es lo mismo dar por enfermo de cáncer a una persona sana (**falso positivo**) que dar por persona sana a un enfermo de cáncer (**falso negativo**).
- ▶ Cuando distinguir entre los distintos tipos de errores es importante, entonces se puede usar una **matriz de confusión** (también llamada tabla de contingencia) para mostrar los diferentes tipos de error.

	Enfermo	Sano
Enfermo	Verdadero positivo	Falso positivo
Sano	Falso negativo	Verdadero negativo

Clase estimada	Clase real	
	Clase referencia	Clase no referencia
Clase referencia	TP	FP
	FN	TN

Evaluación del clasificador

	Enfermo	Sano
Enfermo	Verdadero positivo	Falso positivo
Sano	Falso negativo	Verdadero negativo

- La **sensibilidad** es la capacidad del clasificador para detectar la clase positiva (la que más interesa, en el ejemplo, Enfermo)

$$\text{sensibilidad} = \frac{\text{verdaderos_positivos}}{\text{verdaderos_positivos} + \text{falsos_negativos}}$$

- La **especificidad** es la capacidad de clasificar correctamente a un individuo cuyo estado real sea negativo para la prueba que se hace (en nuestro ejemplo, que esté sano)

$$\text{especificidad} = \frac{\text{verdaderos_negativos}}{\text{verdadero_negativos} + \text{falsos_positivos}}$$

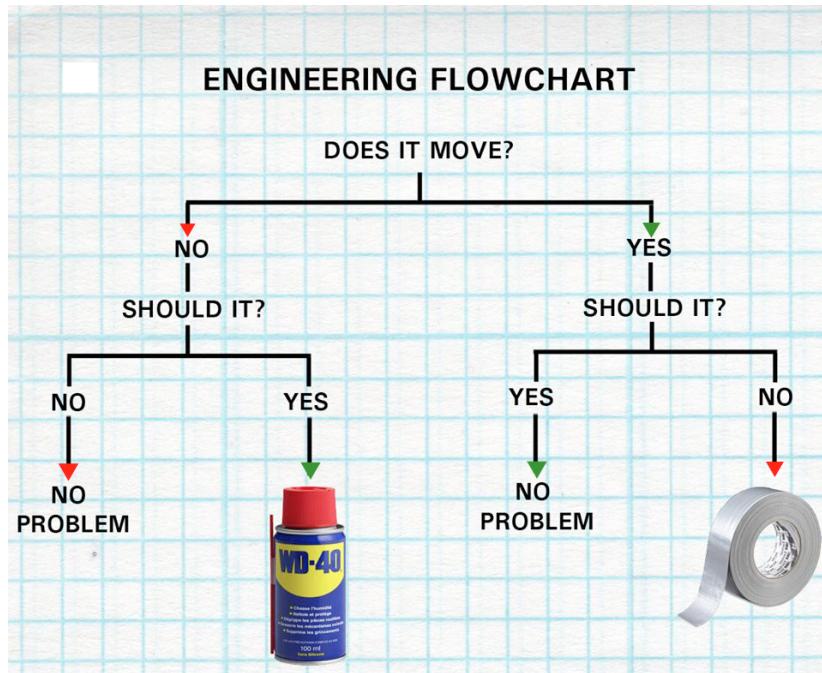
Algoritmos de Aprendizaje

- ▶ En el **aprendizaje supervisado**, el objetivo del aprendizaje es la construcción de un modelo o clasificador que pueda servir para predecir la clase de futuros ejemplos. Dicho clasificador debe construirse de manera que se maximice la efectividad del sistema, es decir, el número de aciertos sobre los datos de entrenamiento, con la esperanza de que sea igual de efectivo sobre los futuros ejemplos.
- ▶ Hay **muchos algoritmos** utilizados en aprendizaje supervisado como las redes neuronales, los árboles de decisión, SVM (support Vector Machines), las redes bayesianas, sistemas de obtención de reglas, los clasificadores bayesianos como las redes bayesianas, etc.
- ▶ Las tareas de aprendizaje supervisado son las más aplicadas en la práctica, especialmente en clases discretas => **Clasificación supervisada**.
- ▶ Dado que es imposible cubrir todos ellos, nos centraremos en algunos muy representativos.

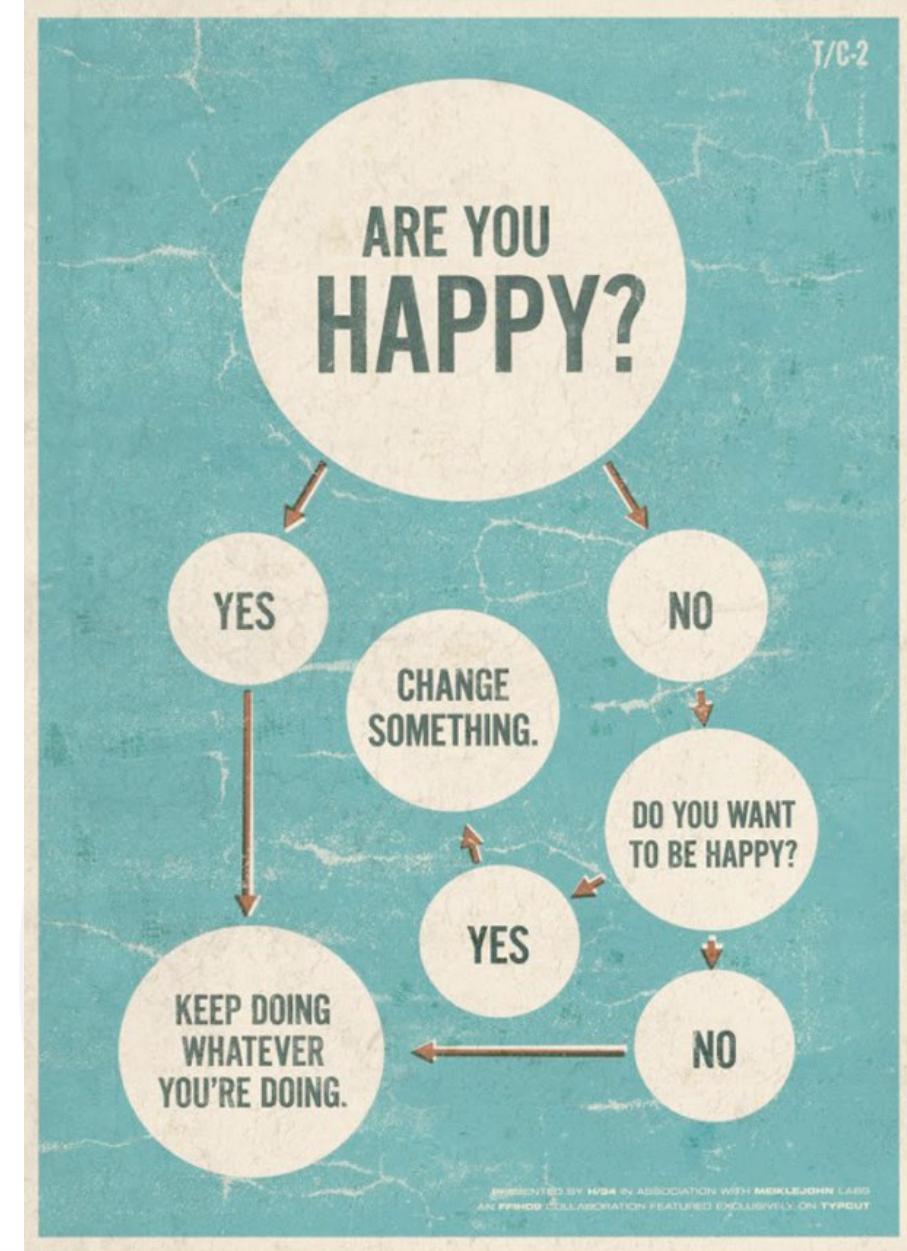
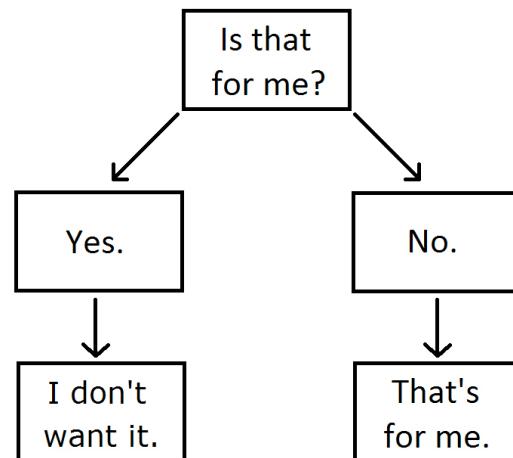
Árboles de Decisión

- ▶ Los árboles de decisión se basan en el **particionamiento recursivo** del espacio de valores de los atributos del problema.
- ▶ Se dividen en **árboles de clasificación** cuando la clase es una variable discreta o **árboles de regresión** cuando la salida es una variable continua.
- ▶ El **objetivo es ir dividiendo el conjunto de casos** en base a un criterio y usando una única variable en cada partición, hasta que al final, idealmente, en cada una de las distintas particiones realizadas no haya más que casos pertenecientes a una misma clase.
- ▶ La representación del conocimiento adquirido en un árbol de decisión se puede interpretar como un **conjunto de reglas compactadas en forma de árbol**.
- ▶ La inducción de árboles de decisión es uno de los métodos más **sencillos y con más éxito** para construir algoritmos de aprendizaje.
- ▶ La aparición del **algoritmo ID3 de Quinlan** es cuando cobraron importancia; posteriormente Quinlan presentó el **algoritmo C4.5** que es una mejora del anterior y que obtiene mejores resultados.

Árboles de decisión: ejemplos



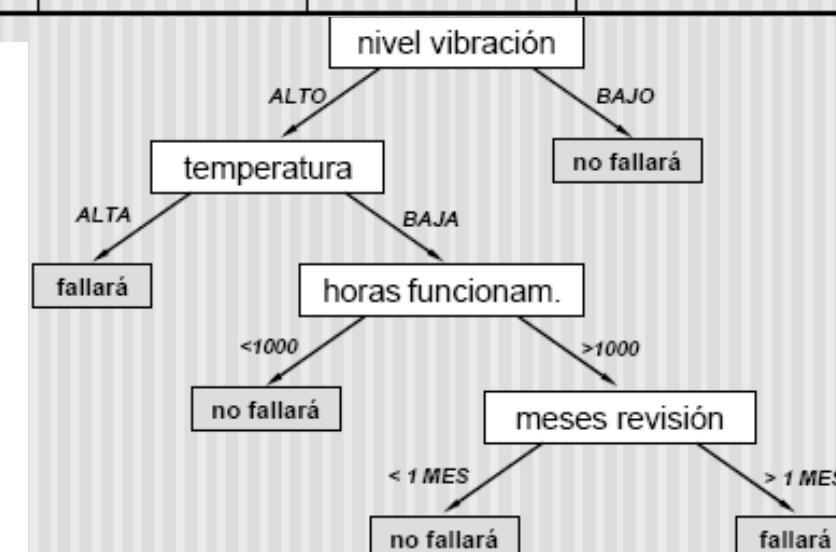
My Cat's Decision-Making Tree.



Árboles de Decisión: Ejemplo

► Modelado de la probabilidad de fallo de una máquina

Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará



Árboles de Decisión: Ejemplo

- Decidir si se debe de esperar por una mesa en un restaurante utilizando los siguientes atributos:
 1. Alternate: is there an alternative restaurant nearby?
 2. Bar: is there a comfortable bar area to wait in?
 3. Fri/Sat: is today Friday or Saturday?
 4. Hungry: are we hungry?
 5. Patrons: number of people in the restaurant (None, Some, Full)
 6. Price: price range (\$, \$\$, \$\$\$)
 7. Raining: is it raining outside?
 8. Reservation: have we made a reservation?
 9. Type: kind of restaurant (French, Italian, Thai, Burger)
 10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

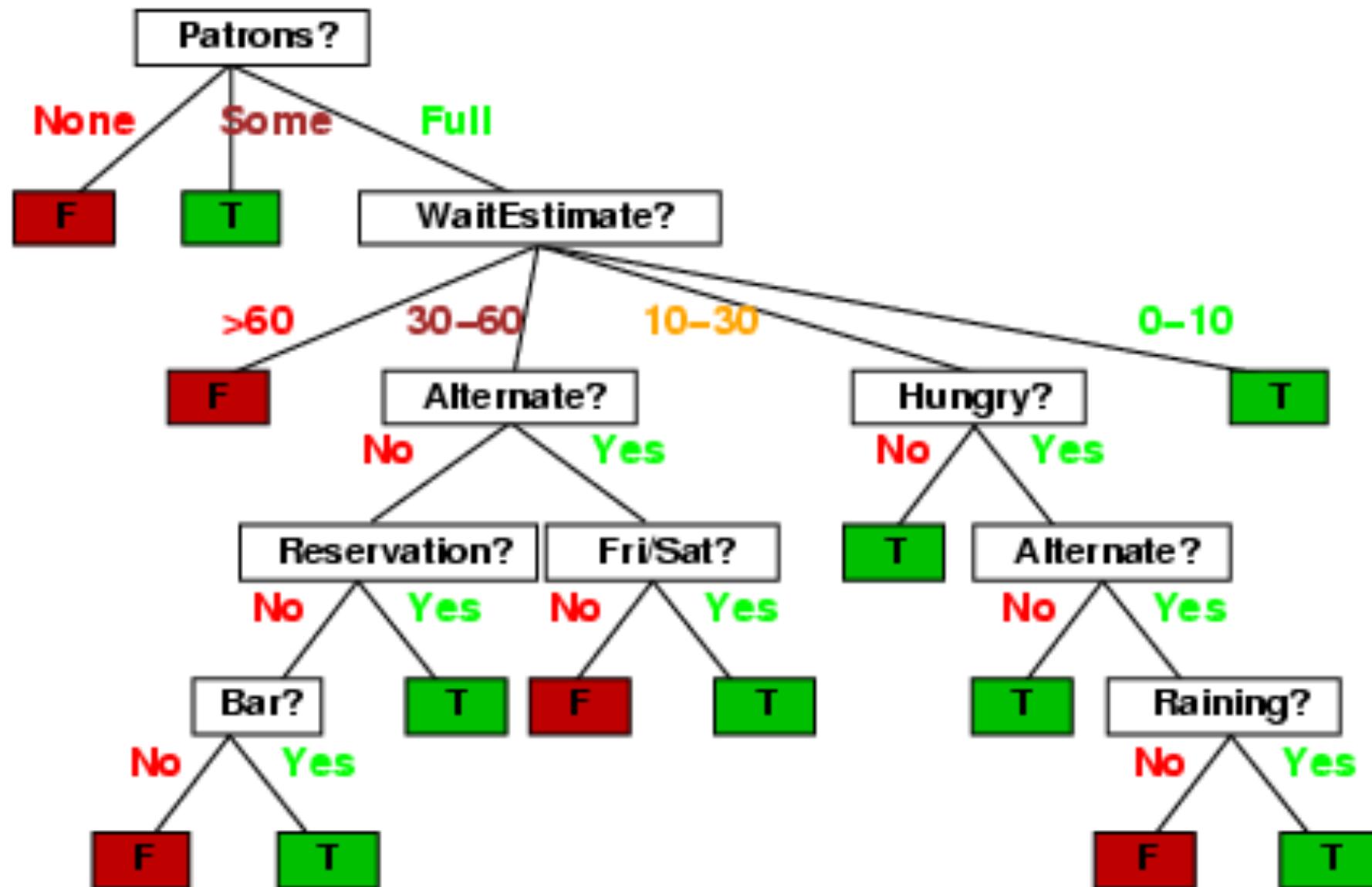
Árboles de Decisión: Ejemplo

► Los datos que tenemos son:

Example	Attributes										Target Wait
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

- T-> True (verdad, esperar)
- F->False (falso, no esperar)

Árboles de Decisión: Ejemplo



Inducción de Árboles de Decisión

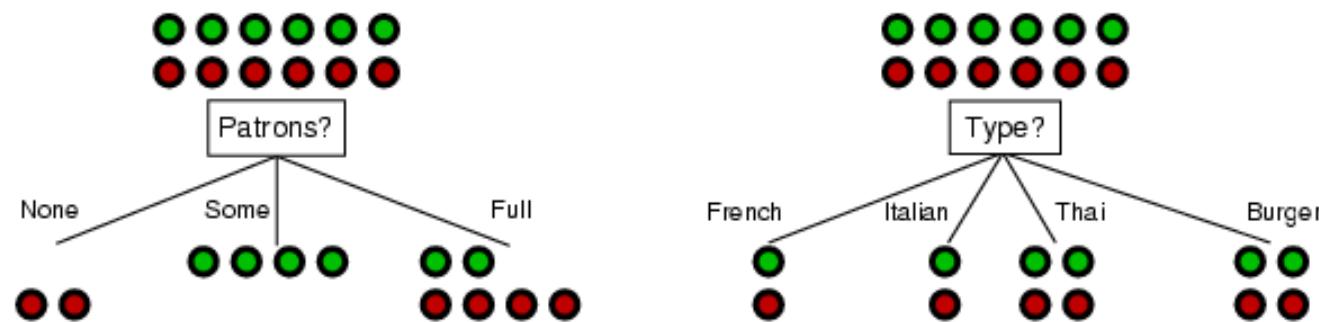
- ▶ Trivial: se crea una ruta del árbol por cada instancia de entrenamiento.
 - Árboles excesivamente grandes.
 - No funcionan bien con instancias nuevas.
- ▶ Óptimo: el árbol más pequeño posible compatible con todas las instancias (navaja de Ockham).
 - Inviabile computacionalmente.
- ▶ Pseudo-optimo (heurístico): selección del atributo en cada nivel del árbol en función de la calidad de la división que produce.
 - Los principales programas de generación de árboles utilizan procedimientos similares (ID3, C4.5, CART, etc).

Inducción de Árboles de Decisión

La creación del árbol de decisión se basa en los siguientes puntos:

- ▶ Determinación del procedimiento para **elegir un nodo** raíz del árbol actual (se empieza con el árbol vacío) y, en general, determinar que variable se elige para ramificar el árbol.

Idea: un buen atributo debería dividir el conjunto de ejemplos en subconjuntos que sean o “todos positivos” o “todos negativos”.



Patrons=Nº Clientes, parece una buena opción.

- ▶ Determinación del **criterio de parada**.
- ▶ Determinación del procedimiento de refinamiento (**poda**): Para evitar sobreajustes.

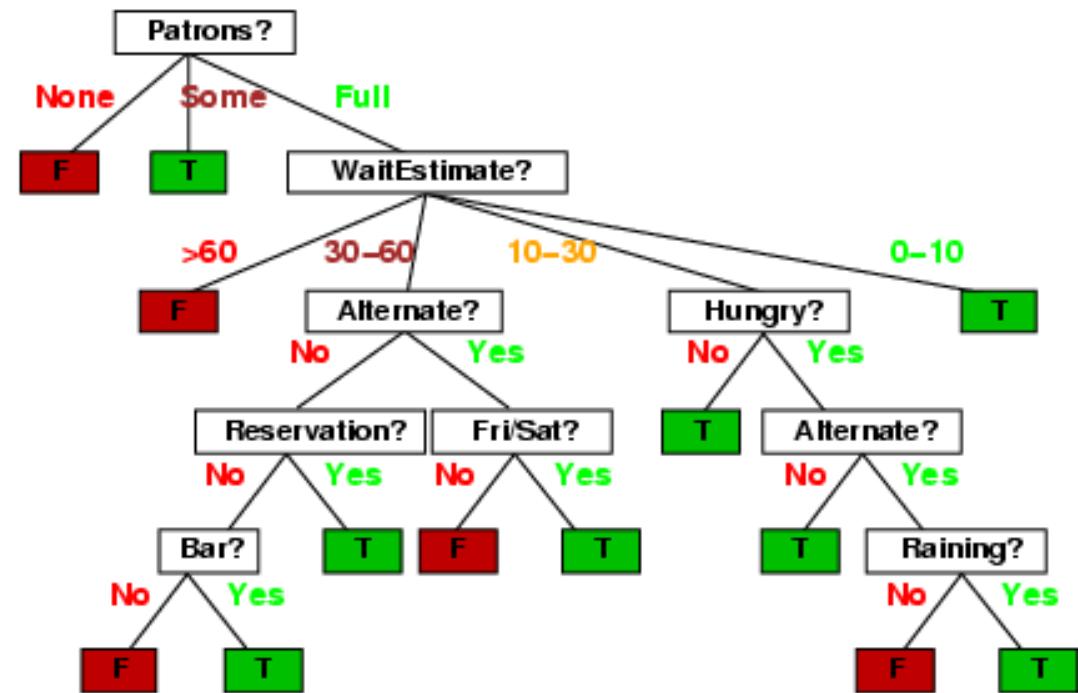
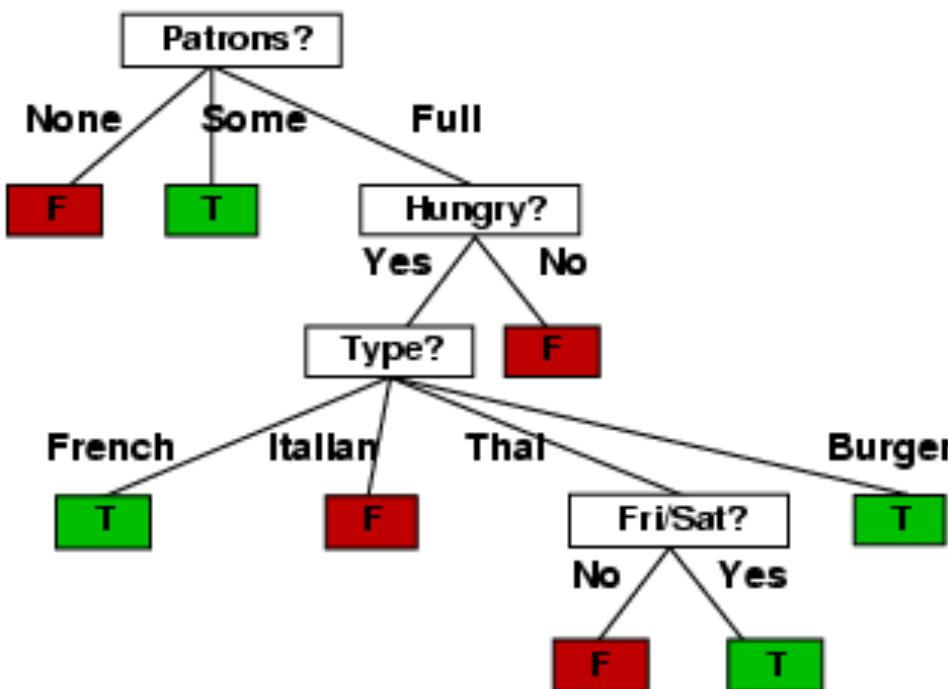
Inducción de Árboles de Decisión

Algoritmo usado para la creación de un árbol de decisión:

```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best  $\leftarrow$  CHOOSE-ATTRIBUTE(attributes, examples)
        tree  $\leftarrow$  a new decision tree with root test best
        for each value  $v_i$  of best do
            examplesi  $\leftarrow$  {elements of examples with best =  $v_i$ }
            subtree  $\leftarrow$  DTL(examplesi, attributes - best, MODE(examples))
            add a branch to tree with label  $v_i$  and subtree subtree
    return tree
```

Inducción de Árboles de Decisión

- Árbol de Decisión obtenido a partir de los datos. Mucho más simple que “el verdadero”

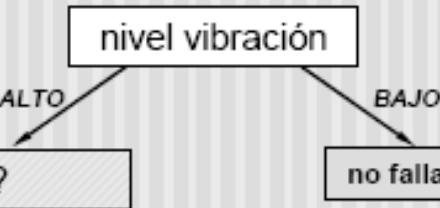


Árboles de Decisión: Ejemplo

► Modelado de la probabilidad de fallo de una máquina

- ¿Qué atributo elegir para el primer nodo?

ATRIBUTO	VALORES	CLASE	
		fallará	no fallará
Temperatura	Alto	2	2
	Bajo	1	2
Nivel de vibraciones	Alto	3	1
	Bajo	0	3
Horas defuncionamiento	< 1000	2	3
	> 1000	1	1
Meses desde revisión	> 1 mes	2	3
	< 1 mes	1	1



No fallará (1 instancia)
fallará (3 instancias)

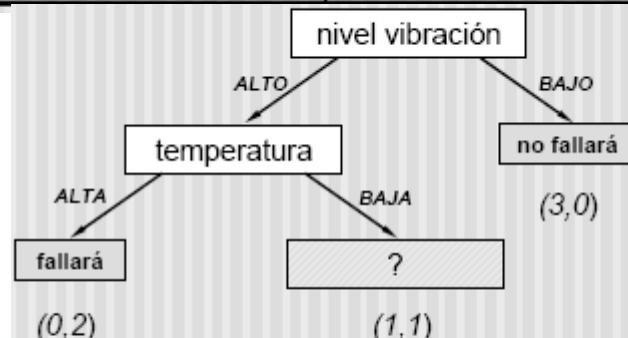
no fallará (3 instancias)
fallará (0 instancias)

Árboles de Decisión: Ejemplo

► Modelado de la probabilidad de fallo de una máquina

Sólo aquellos ejemplos de entrenamiento que llegan al nodo se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		<i>fallará</i>	<i>No fallará</i>
Temperatura	Alta	2	0
	Baja	1	1
Horas de funcionamiento	< 1000	2	1
	> 1000	1	0
Meses desde revisión	> 1 mes	2	1
	< 1 mes	1	0

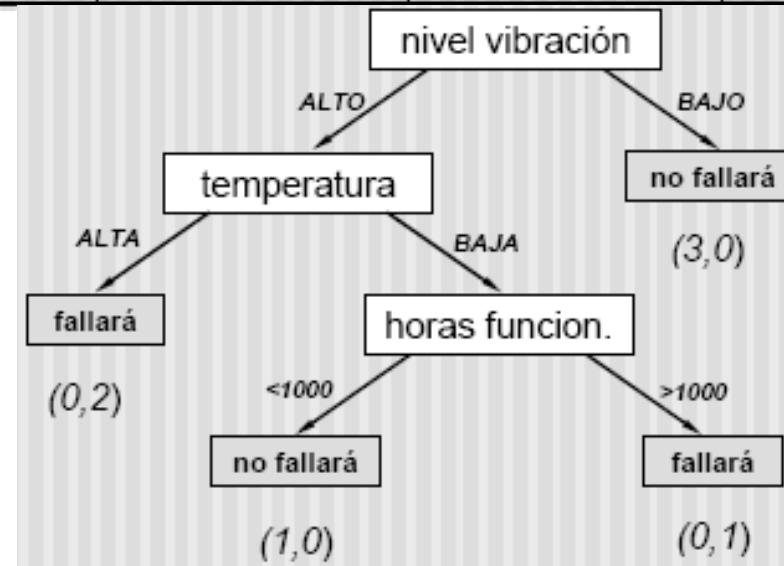


Árboles de Decisión: Ejemplo

► Modelado de la probabilidad de fallo de una máquina

De nuevo, sólo aquellos ejemplos de entrenamiento que llegan al nodo se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		<i>fails</i>	<i>works</i>
Horas de funcionamiento	< 1000	0	1
	>1000	1	0
Meses desde revisión	> 1 mes	1	1
	< 1 mes	0	0



Árboles de Decisión: Elección variables

- ▶ La **entropía** definida por Shannon , referida a la teoría de la información, hace referencia a la cantidad media de información que contiene una variable aleatoria.
- ▶ La entropía también se puede interpretar usando el concepto, más intuitivo, de **incertidumbre**.

$$H(S) = \sum_{i=1}^n p(s_i) \log_2 \frac{1}{p(s_i)}$$

- ▶ Shannon afirma que **cuanto menos información haya, mayor será su entropía**, es decir que la información reduce la incertidumbre o la entropía, así que guardan una relación directa entre sí, también se puede decir que cuando recibimos información sobre un tema determinado estamos disminuyendo la ignorancia, desorden o incertidumbre de este.
- ▶ Shannon demostró analíticamente que la entropía es el límite máximo al que se puede comprimir una fuente sin ninguna pérdida de información.

Árboles de Decisión: Elección variables

En ID3, el mejor atributo X para ramificar el árbol dado el conjunto de datos D, es **aquel que maximiza la ganancia de información**:

- ▶ Se calcula la **entropía de la clase C antes de ramificar**:

$$H_D(C) = \sum_{i=1}^n p_i \log_2 \frac{1}{p_i}$$

donde p_i representa la probabilidad (calculada por frecuencias relativas) de la clase i en D y n es el número de clases.

- ▶ Se calcula el valor medio de la **entropía en el nivel siguiente**, generado por el atributo X:

$$H_D(C|X) = \sum_{j=1}^k p_D(X = x_j) H_{D_j}(C)$$

- ▶ Se denomina ganancia de información de la variable X, a la diferencia de entropía de C menos la entropía media del siguiente nivel si ramificásemos por X:

$$\text{Ganancia}_D(C|X) = H_D(C) - H_D(C|X)$$

Árboles de Decisión: Elección variables

- ▶ En C4.5 además de la ganancia de información, se calcula la Información de ruptura y la razón de ganancia.

$$\text{RazónGanancia}(X|C) = \frac{\text{Ganancia}_D(C|X)}{\text{InfoRuptura}_D(X)}$$

- ▶ donde,

$$\text{InfoRuptura}_D(X) = \sum_{j=1}^k p_D(X = x_j) \log_2 \frac{1}{p_D(X=x_j)}$$

- ▶ El test basado en el criterio de máxima ganancia, tiende a escoger aquellas variables con un mayor número de valores. Esto es debido a que cuantas más particiones se hagan debido a los valores de la variable, la entropía de un nuevo nodo será, normalmente, menor. En C4.5 se utiliza la razón de ganancia para paliar esta tendencia.

Árboles de Decisión: Parada

Las condiciones de parada para dejar de ramificar un árbol se utilizan cuando su crecimiento no parece mejorar la capacidad predictiva del árbol:

- ▶ Todos los casos pertenecen a la **misma clase**.
- ▶ Todos los casos tienen los **mismos valores para las variables**, aunque no necesariamente coincidan en los valores de la variable a clasificar.
- ▶ Se ha llegado a un nivel superior a un **límite**.
- ▶ No hay rechazo de independencia entre una variable X y la variable a clasificar en un **test** de hipótesis (por ejemplo, en un test chi cuadrado χ^2). **Prepoda**.

Árboles de Decisión: Poda

Se introduce el procedimiento poda para evitar sobreajustes.

Algunos métodos:

- ▶ Reduced Error Pruning: Ascendente. ID3. Compara el error del conjunto de datos de un nodo con el del subárbol que cuelga de él, si el error del padre es menor, se poda el subárbol hijo. ID3.
- ▶ Pessimistic Error Pruning: Descendente. Introduce la corrección de continuidad para la distribución binomial en el calculo del error. Compara el error del árbol con el guardado en el nodo, si el del nodo es menor, poda.
- ▶ Error Based Pruning: Ascendente. C4.5. Se estima un intervalo de confianza, para modificar el error respecto a casos no vistos. Para cada nodo se tiene su error calculado de esta forma y el error que se obtendría si fuera podado, si este segundo error es menor, se realiza la poda.

Árboles de Decisión: Ejemplo

► Modelo psicológico de la felicidad de un estudiante.

Estudia	Vida social	Aprueba	Feliz
Poco	Poco	Poco	NO
Poco	Poco	Mucho	SI
Poco	Algo	Algo	NO
Poco	Algo	Mucho	SI
Poco	Mucho	Poco	SI
Algo	Poco	Poco	NO
Algo	Poco	Mucho	NO
Algo	Mucho	Poco	SI
Mucho	Poco	Poco	NO
Mucho	Poco	Algo	NO
Mucho	Algo	Mucho	NO
Mucho	Mucho	Algo	SI

Árboles de Decisión: Ejemplo

- Veamos cual es el atributo con mayor ganancia de información para que sea el nodo raíz

$$Et(S) = -5/12 \log_2(5/12) - 7/12 \log_2(7/12) = 0,979$$

$$Et(\text{Estudia}=Poco) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0,970$$

$$Et(\text{Estudia}=Algo) = -1/3 \log_2(1/3) - 2/3 \log_2(2/3) = 0,918$$

$$Et(\text{Estudia}=Mucho) = -1/4 \log_2(1/4) - 3/4 \log_2(3/4) = 0,811$$

$$\mathbf{G(S, Estudia)} = Et(S) - 5/12 \cdot 0,970 - 3/12 \cdot 0,918 - 4/12 \cdot 0,811 = 0,075$$

$$Et(\text{VidaSocial}=Poco) = 0,650$$

$$Et(\text{VidaSocial}=Algo) = 0,918$$

$$Et(\text{VidaSocial}=Mucho) = -3/3 \log_2(3/3) - 0/3 \log_2(0/3) = 0$$

$$\mathbf{G(S, VidaSocial)} = Et(S) - 6/12 \cdot 0,650 - 3/12 \cdot 0,918 - 3/12 \cdot 0 = 0,425$$

$$Et(\text{Aprueba}=Poco) = 0,970$$

$$Et(\text{Aprueba}=Algo) = 0,918$$

$$Et(\text{Aprueba}=Mucho) = 1,0$$

$$\mathbf{G(S, Aprueba)} = Et(S) - 5/12 \cdot 0,970 - 3/12 \cdot 0,918 - 4/12 \cdot 1,0 = 0,012$$

Árboles de Decisión: Ejemplo



Árboles de Decisión: Ejemplo

- Veamos cual es el atributo con mayor ganancia de información para la rama de la izquierda.

Estudia	Aprueba	Feliz
Poco	Poco	NO
Poco	Mucho	SI
Algo	Poco	NO
Algo	Mucho	NO
Mucho	Poco	NO
Mucho	Algo	NO

Poco
↓

$$Et(S) = -1/6 \log_2(1/6) - 5/6 \log_2(5/6) = 0,650$$

$$Et(\text{Estudia}=Poco) = 1,0$$

$$Et(\text{Estudia}=Algo) = 0$$

$$Et(\text{Estudia}=Mucho) = 0$$

$$G(S, \text{Estudia}) = Et(S) - 2/6 \cdot 1,0 - 2/6 \cdot 0 - 2/6 \cdot 0 = 0,316$$

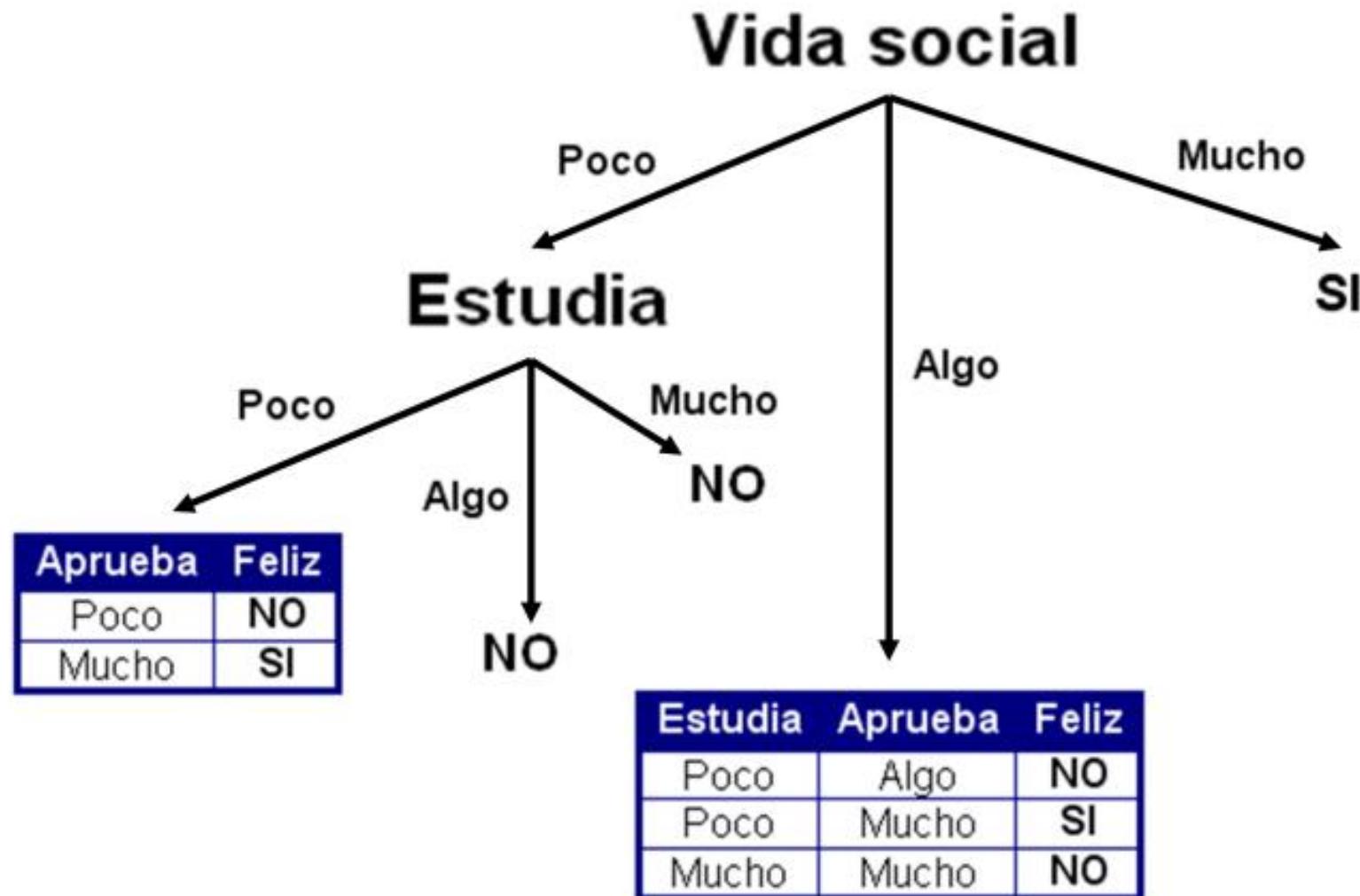
$$Et(\text{Aprueba}=Poco)=0$$

$$Et(\text{Aprueba}=Algo)=0$$

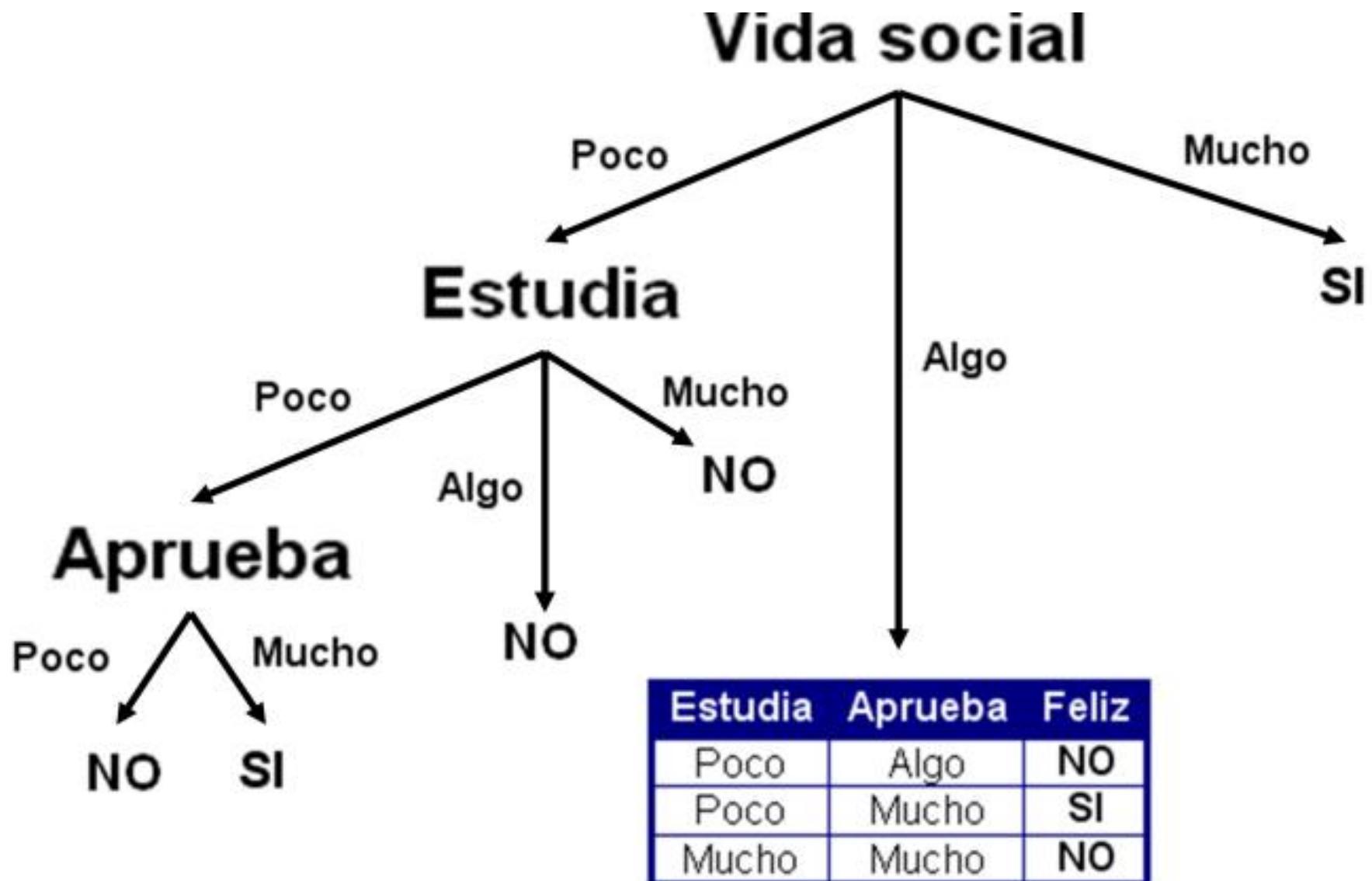
$$Et(\text{Aprueba}=Mucho)=1,0$$

$$G(S, \text{Aprueba}) = Et(S) - 3/6 \cdot 0 - 1/6 \cdot 0 - 2/6 \cdot 1,0 = 0,316$$

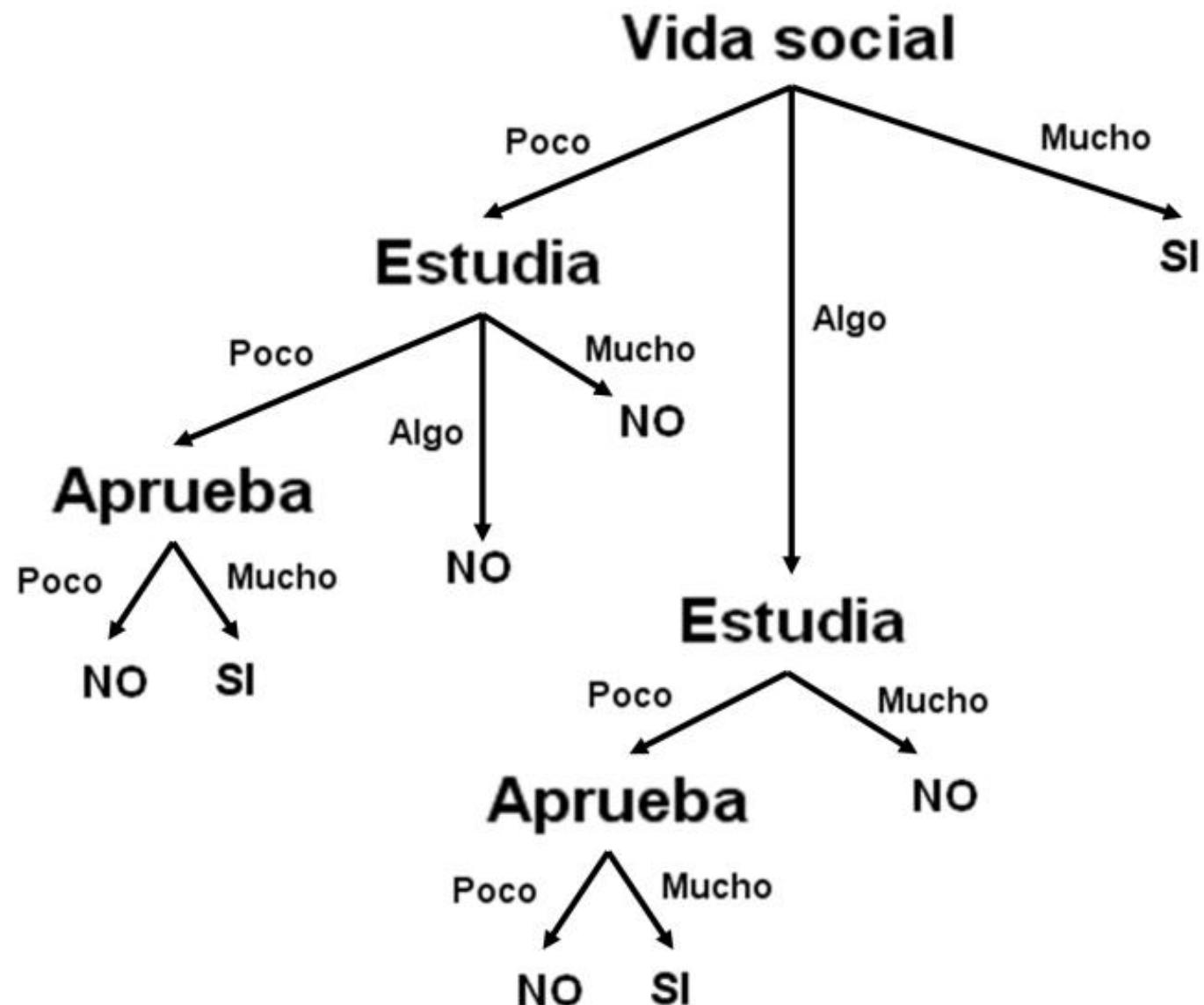
Árboles de Decisión: Ejemplo



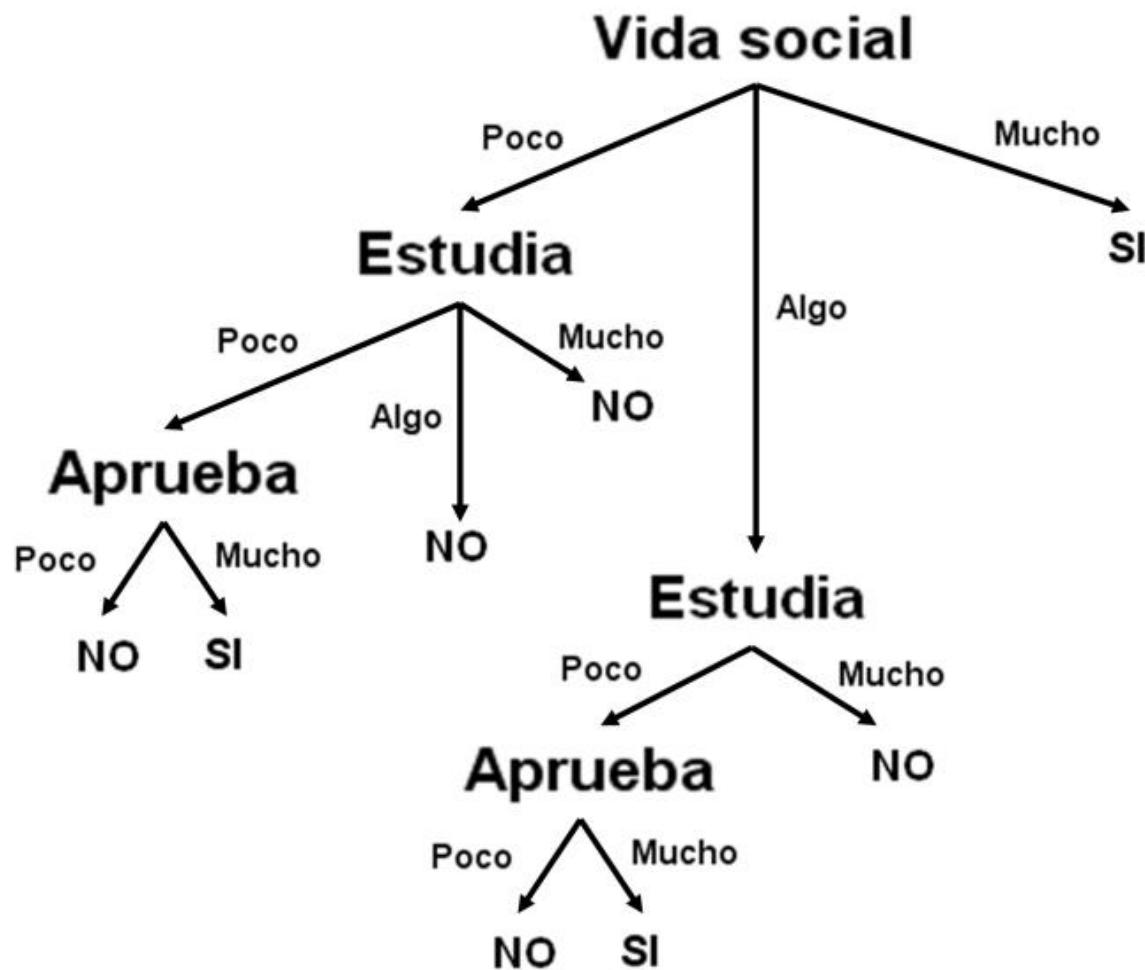
Árboles de Decisión: Ejemplo



Árboles de Decisión: Ejemplo



Árboles de Decisión: Ejemplo



Si $VS=M$ entonces $F=SI$

Si $VS=P$ y $E=P$ y $AP=P$ entonces $F=NO$

Si $VS=P$ y $E=P$ y $AP=M$ entonces $F=SI$

Si $VS=P$ y $E=A$ entonces $F=NO$

Si $VS=P$ y $E=M$ entonces $F=NO$

Si $VS=A$ y $E=P$ y $AP=A$ entonces $F=NO$

Si $VS=A$ y $E=P$ y $AP=M$ entonces $F=SI$

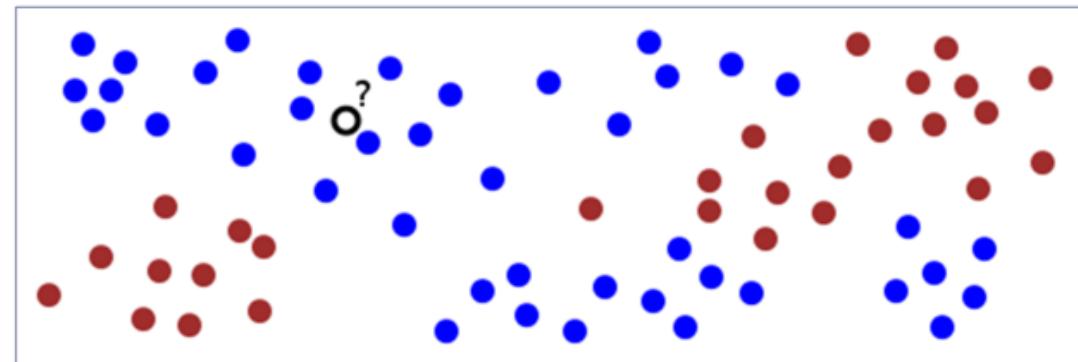
Si $VS=A$ y $E=M$ entonces $F=NO$

Si $VS=M$ entonces $F=SI$

En otro caso $F=NO$ (Regla por defecto)

Vecino más cercano (Nearest Neighbour)

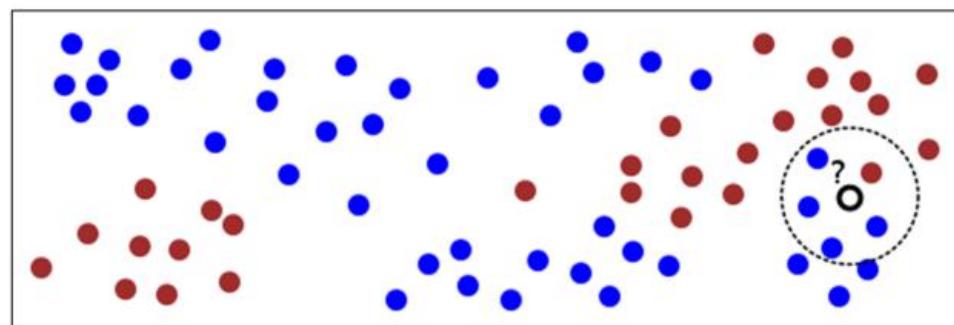
- ▶ Es la técnica más sencilla de aprendizaje supervisado.
- ▶ Consiste en memorizar todos los ejemplos conocidos.
- ▶ Para cada caso nuevo que llega, busca el ejemplo almacenado más parecido, es decir, aquél cuya distancia al nuevo es mínima, y devuelve su clase como respuesta.
- ▶ Como métrica de distancia suele usarse la distancia Euclídea.



- ▶ En el ejemplo el punto "?" se clasificaría como azul, puesto que el más próximo es de dicha clase.

KNN (K Nearest Neighbours)

- ▶ La técnica de los K vecinos más cercanos está basada en l método anterior intentando minimizar el error en valores frontera o muy distintos.
- ▶ En lugar de limitarse a buscar el ejemplo más próximo, busca los N casos más próximos.
- ▶ Para variables de salida discretas, el algoritmo k-NN devuelve como resultado la moda (el valor más frecuente).
- ▶ Para variables continuas, se devuelve como resultado la media.



- ▶ En el ejemplo el punto "?" con k=5 se clasificaría como azul, aunque el rojo sea el más cercano.

Redes Neuronales Artificiales

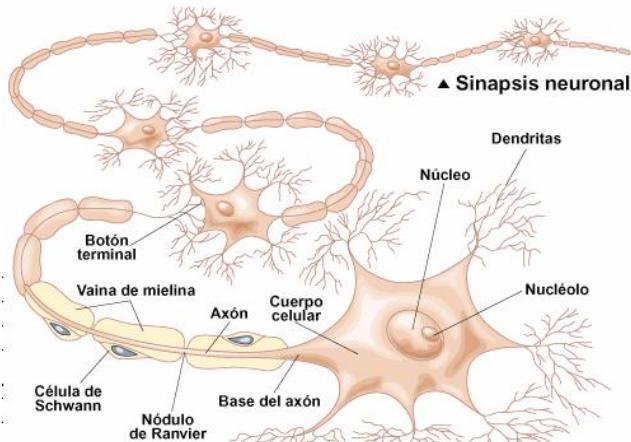
► El cerebro humano:

- Tiene distribuido el conocimiento en un gran número de neuronas.
- Las neuronas se comunican unas con otras.
- El cerebro puede aprender.

► Las redes neuronales artificiales se basan en los mismos conceptos que las redes neuronales naturales, que estructuralmente tienen como elemento atómico la neurona.

► Una neurona está compuesta de un cuerpo o soma que contiene el núcleo, un tallo o axón, y las prolongaciones del cuerpo y del axón, denominadas dendritas.

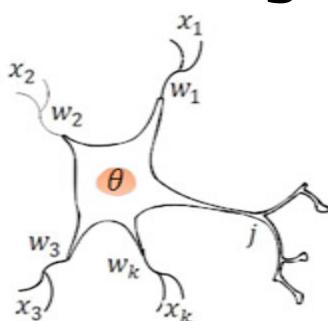
► Las dendritas de una neurona se conectan con las de otras, una conexión denominada sinapsis, que permite la transmisión de impulsos eléctricos.



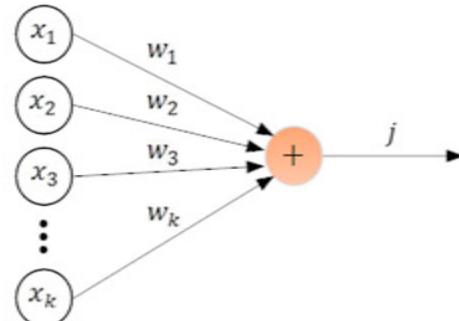
Redes Neuronales Artificiales

► Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por las siguientes funciones:

- **Función de propagación (o excitación)**: que consiste (normalmente) en la **suma ponderada de las señales de entrada**. Si el peso es positivo la conexión es **excitatoria** y si es negativo **inhibitoria**.
- **Función de activación y/o transferencia**: que en su forma más sencilla (la de la animación) es 1 si se supera un **umbral u**, y 0 en otro caso. Existen otras versiones de función de activación, como una función lineal, la de la tangente hiperbólica o, la más popular, la **sigmoidal**.



a) Neurona biológica



a) Neurona artificial

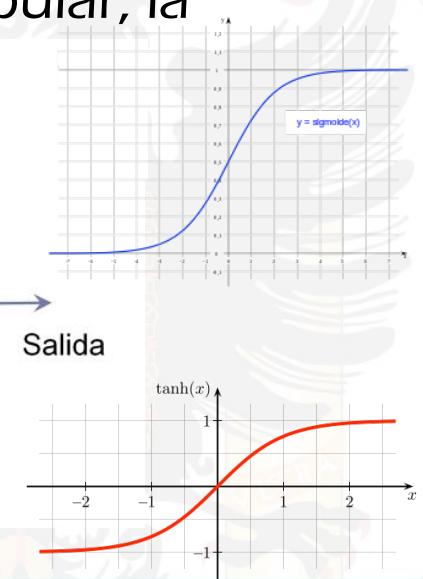
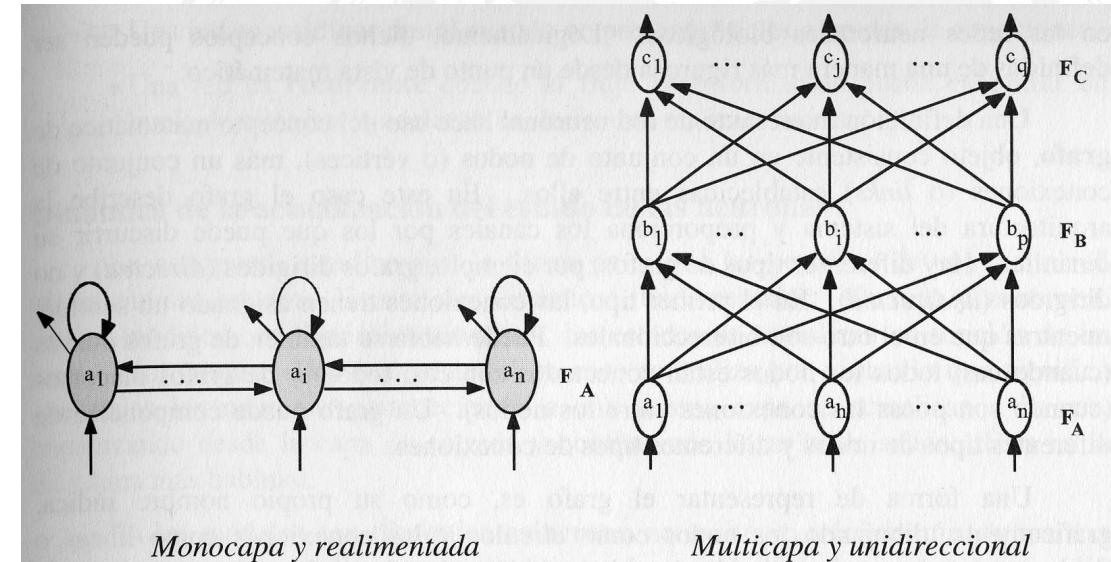
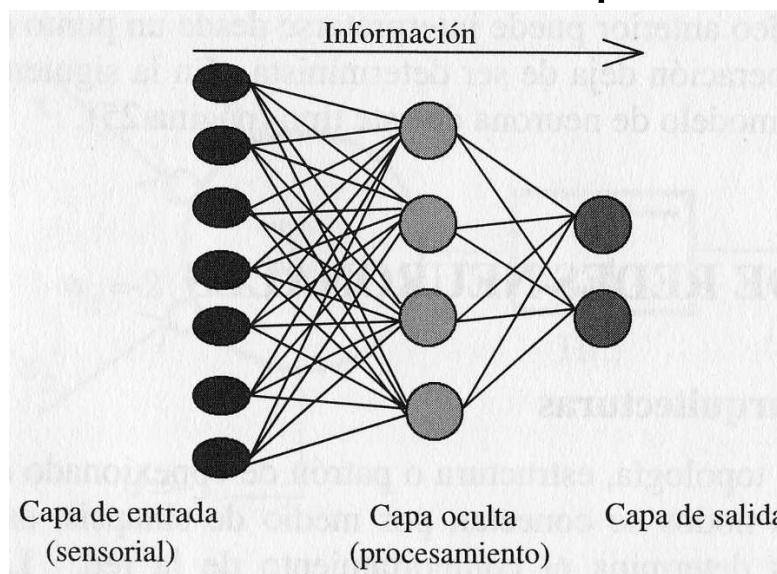


Figura 7. Neurona biológica versus artificial.

RNA: Arquitectura.

- ▶ Las redes neuronales artificiales se disponen por **capas de neuronas**. Se distinguen tres tipos de capas: **de entrada, de salida y ocultas**.
- ▶ Teniendo en cuenta el flujo de datos, podemos distinguir entre **redes unidireccionales** (feedforward) y **redes recurrentes o realimentadas** (feedback). Mientras que en las redes unidireccionales la información circula en un único sentido, en las redes recurrentes o realimentadas la información puede circular entre las distintas capas de neuronas en cualquier sentido, incluso en el de salida-entrada.



RNA: Tipos.

Algoritmos de aprendizaje más conocidos				
Paradigma	Regla de aprendizaje	Arquitectura	Algoritmo de aprendizaje	Tareas
Supervisado	Corrección del error	Perceptrón o perceptrón multicapa	Algoritmos de aprendizaje perceptrón, retropropagación del error, ADALINE, MADALINE	Clasificación de patrones, aproximación de funciones, predicción, control, ...
		Elman y Jordan recurrentes	Retropropagación del error	Síntesis de series temporales
	Boltzmann	Recurrente	Algoritmo de aprendizaje Boltzmann	Clasificación de patrones
	Competitivo	Competitivo	LVQ	Categorización intra-clase, compresión de datos
		Red ART	ARTMap	Clasificación de patrones, categorización intra-clase
No supervisado	Corrección del error	Red de Hopfield	Aprendizaje de memoria asociativa	Memoria asociativa
		Multicapa sin realimentación	Proyección de Sannon	Ánalisis de datos
	Competitiva	Competitiva	VQ	Categorización, compresión de datos
		SOM	Kohonen SOM	Categorización, análisis de datos
Por refuerzo	Hebbian	Redes ART	ART1, ART2	Categorización
		Multicapa sin realimentación	Análisis lineal de discriminante	Ánalisis de datos, clasificación de patrones
		Sin realimentación o competitiva	Análisis de componentes principales	Ánalisis de datos, compresión de datos

RNA: Tipos.

MODELOS DE REDES NEURONALES ARTIFICIALES



RNA: El Perceptrón

- ▶ El perceptrón: fue el primer tipo de red neuronal. Es similar al nodo de un grafo, que recibe varias posibles entradas y, en función de ellas, resulta estimulada o no.
- ▶ El algoritmo de aprendizaje es el mismo para todas las neuronas, todo lo que sigue se aplica a una sola neurona en el aislamiento. Se definen algunas variables primero:

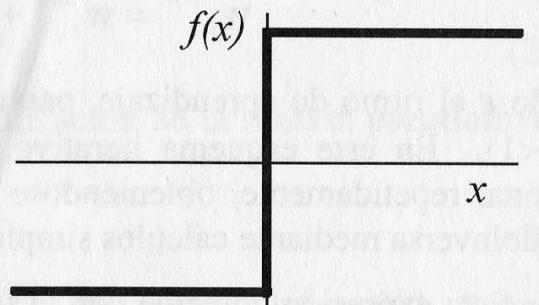
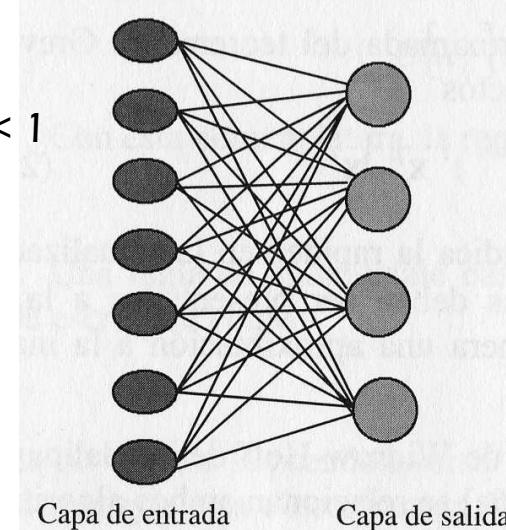
- $x(j)$ denota el elemento en la posición j en el vector de la entrada
- $w(j)$ el elemento en la posición j en el vector de peso
- C denota la salida esperada
- y es la salida de la neurona
- α es una constante tal que $0 < \alpha < 1$

- ▶ La regla de actualización de los pesos es un proceso iterativo y será

$$w(j)_{\text{Nuevo}} = w(j)_{\text{Viejo}} + C \cdot x(j)$$

- ▶ o bien, usando α tasa de aprendizaje:

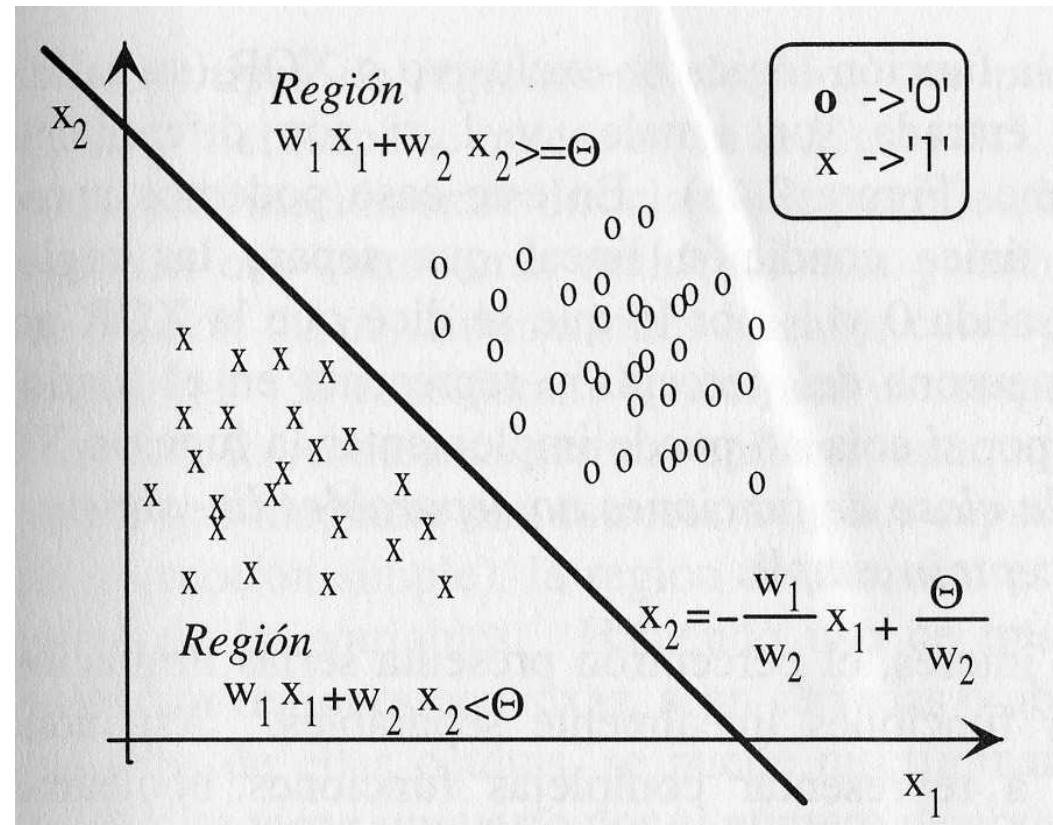
$$w(j)_{\text{Nuevo}} = w(j)_{\text{Viejo}} + \alpha(C-y) \cdot x(j)$$



Arquitectura (izquierda) y función de transferencia (derecha) de un perceptrón

RNA: El Perceptrón

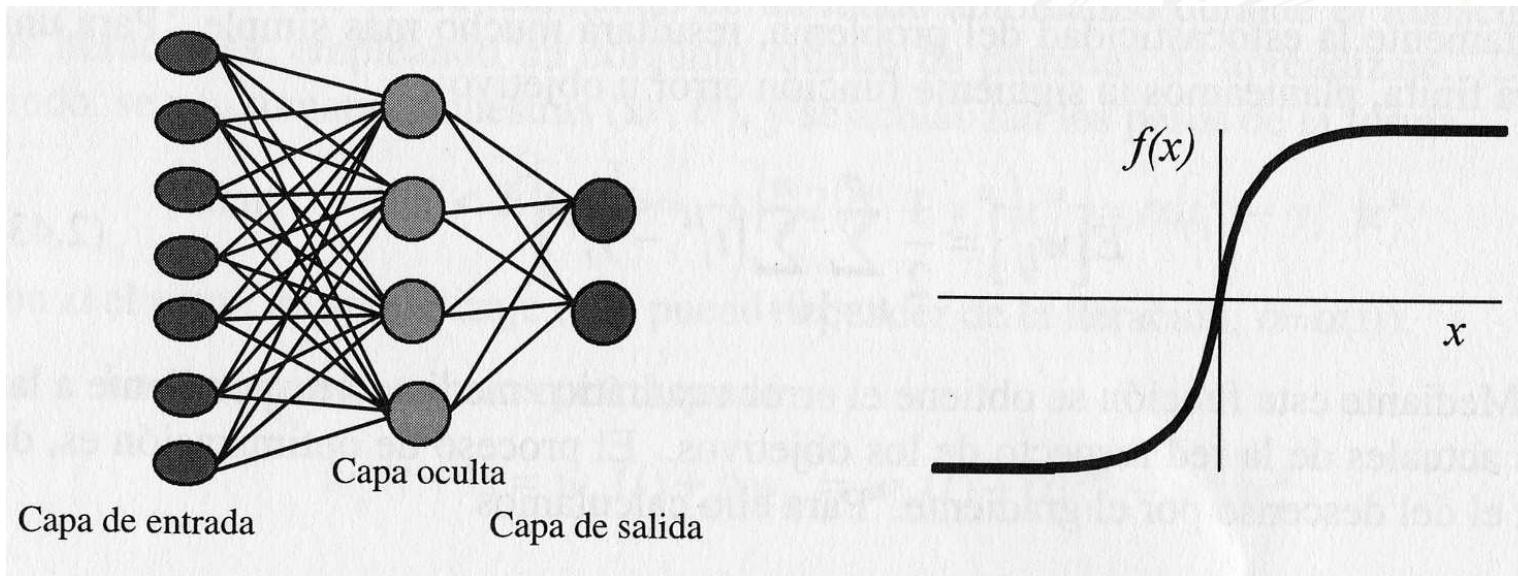
- ▶ El perceptrón simple presenta grandes limitaciones, ya que tan sólo es capaz de representar funciones linealmente separables.



Región de decisión correspondiente a un perceptrón simple con dos neuronas de entrada

RNA: El Perceptrón multicapa

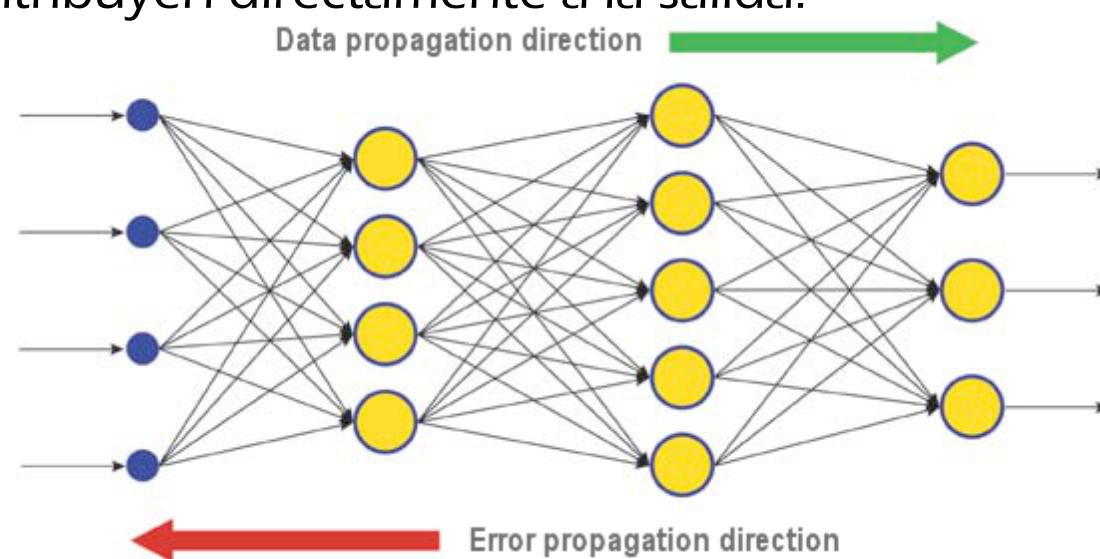
- ▶ El perceptrón multicapa está formado por múltiples capas, esto le permite resolver problemas que no son linealmente separables.
- ▶ El perceptrón multicapa puede ser totalmente o localmente conectado.



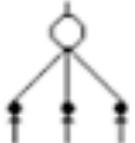
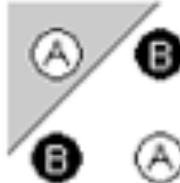
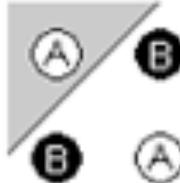
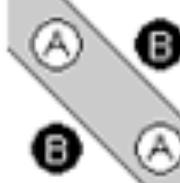
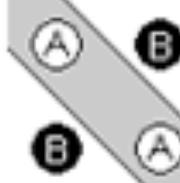
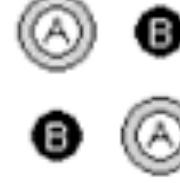
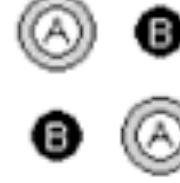
: Arquitectura (izquierda) y función de activación (derecha) para el perceptrón multicapa

RNA: El Perceptrón multicapa

- ▶ La propagación hacia atrás de errores o retropropagación (del inglés **backpropagation**) es el algoritmo de aprendizaje que se utiliza para el perceptrón multicapa.
- ▶ Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida.
- ▶ La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida.



RNA: Expresividad

Estructura	Regiones de Desición	Problema de la XOR	Clases con Regiones Mezcladas	Formas de Regiones más Generales
1 Capa 	Medio Plano Limitado por un Hiperplano 			
2 Capas 	Regiones Cerradas o Convexas 			
3 Capas 	Complejidad Arbitraria Limitada por el Número de Neuronas 			

RNA: El Perceptrón multicapa

► Las redes neuronales tienen las siguientes ventajas:

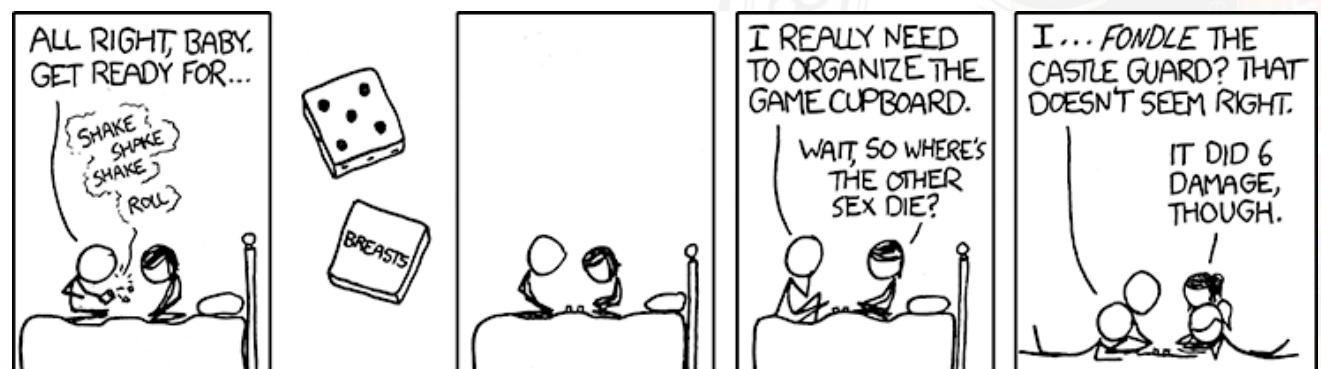
- Tolerancia a fallos.
- Fácilmente paralelizables.
- Son excelentes reconocedores de patrones. Especialmente buenas con datos continuos, como señales de audio o imágenes.
- Flexibilidad: Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada

► La principal desventaja es que no se puede extraer conocimiento de ellas.



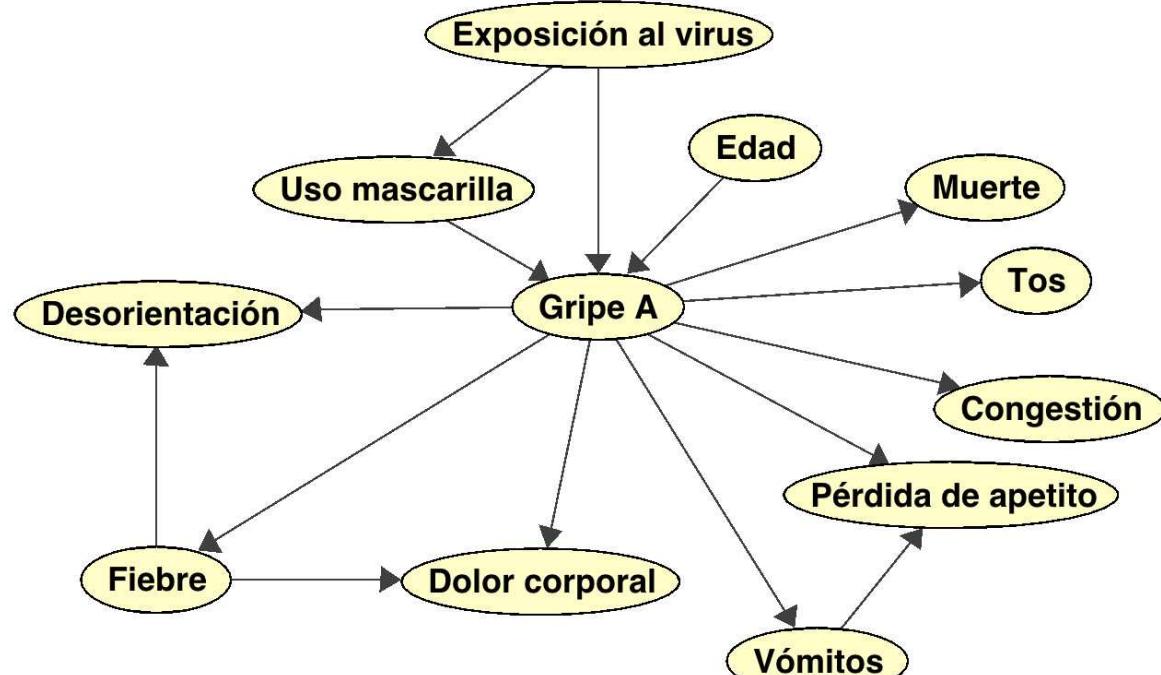
Redes Bayesianas

- ▶ El que dos hechos X e Y sean **independientes** nos viene a decir que el conocer que la primera variable X toma un determinado valor, no aporta ninguna información acerca del valor que puede tomar la segunda variable Y y viceversa. Ej.: 2 dados, síntoma-enfermedad.
- ▶ Las redes bayesianas están compuestas de una **parte gráfica** y una **parte cuantitativa**, más concretamente: un **grafo dirigido acíclico** y una colección de parámetros numéricos, normalmente **tablas de probabilidad condicionada**.



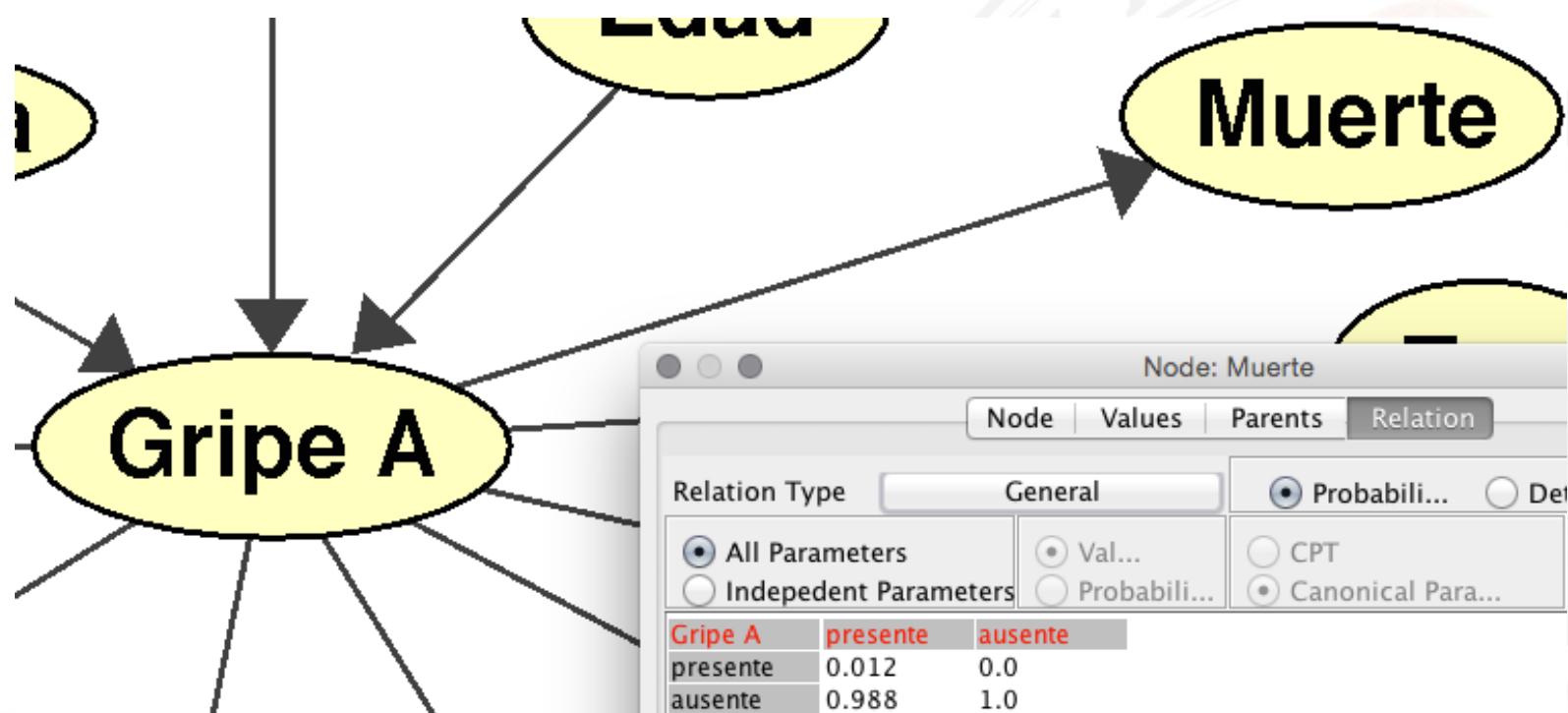
Redes Bayesianas

- ▶ La parte gráfica nos va a permitir representar relaciones de dependencia e independencia entre variables.
- ▶ En el grafo los nodos representan a las variables y la relación de dependencia entre dos variables se representa mediante la existencia de un camino o un arco entre ellas.
- ▶ Si dos variables X e Y están conectadas por un arco $X \rightarrow Y$ sabemos que las dos variables están relacionadas.



Redes Bayesianas

- ▶ La parte cuantitativa de nuestra red es la información numérica necesaria para determinar una distribución conjunta teniendo en cuenta las independencias representadas en la parte cualitativa.
- ▶ Nos va decir en qué medida nos creemos las relaciones de dependencia entre las variables, permitiéndonos así representar la incertidumbre.

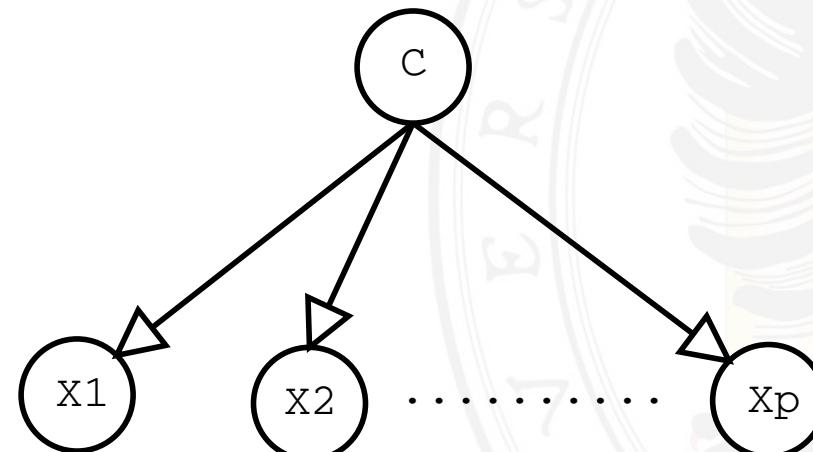


Redes Bayesianas: Construcción

- Las redes bayesianas se pueden construir con un **experto**.
- Aprendizaje estructural:
 - **Basadas en tests de independencia:** son métodos que utilizan criterios de independencia entre variables, para obtener la estructura que mejor representa el conjunto de independencias que se deducen de los datos.
 - **Métricas + búsqueda:** son paradigmas de aprendizaje que se basan en un criterio de la bondad del ajuste de una estructura a los datos. Utilizando dicho criterio se realiza un proceso de búsqueda entre las estructuras candidatas, dando como resultado aquella estructura que mejor se ajuste a los datos.
- Aprendizaje paramétrico: Calculando probabilidades condicionadas (Teorema de Bayes). Por máxima verosimilitud o estimador suavizado (Laplace).

Redes Bayesianas: Naïve Bayes

- Las redes bayesianas **aprenden/representan la distribución de probabilidad que hay en los datos**: se puede utilizar tanto para aprendizaje supervisado como no supervisado.
- El primer clasificador basado en una red bayesiana fue el **naïve bayes** o ingenuo bayes. Este clasificador se basa en dos supuestos:
 - Cada atributo es condicionalmente independiente de los otros atributos dada la clase (**¡ingenuo!**)
 - Todos los atributos tienen influencia sobre la clase.



Redes Bayesianas: Naïve Bayes

El **Teorema de Bayes** expresa la probabilidad condicional de un evento aleatorio A dado B .

► Sea $\{A_1, A_2, \dots, A_i, \dots, A_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos, y tales que la probabilidad de cada uno de ellos es distinta de cero (0). Sea B un suceso cualquiera del que se conocen las probabilidades condicionales $P(B | A_i)$. Entonces, la probabilidad $P(A_i | B)$ viene dada por la expresión:

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)}$$

► donde:

- $P(A_i)$ son las probabilidades a priori.
- $P(B | A_i)$ es la probabilidad de B en la hipótesis A_i
- $P(A_i | B)$ son las probabilidades a posteriori.

► Podemos ver también el Teorema de Bayes como la probabilidad que se dé una causa (p.e. gripe) condicionada a que se dé el efecto observado (p.e. fiebre)

$$P(\text{causa} | \text{efecto}) = \frac{P(\text{efecto} | \text{causa}) \times P(\text{causa})}{P(\text{efecto})}$$

Redes Bayesianas: Naïve Bayes

- Supongamos que un paciente tiene fiebre y queremos saber la probabilidad de que, al tener fiebre, el paciente tenga una gripe, es decir, queremos calcular $P(\text{gripe}|\text{fiebre})$, que es precisamente lo que hacen los médicos: a partir de unos síntomas da un diagnóstico.
- Supongamos que el doctor sabe que la gripe causa fiebre en el paciente el 70% de las veces. $P(\text{fiebre}|\text{gripe})=0,7$.
- Sabemos que el porcentaje de fiebre (independientemente de su causa) en la población es del 15%. $P(\text{fiebre})=0,15$.
- Y el porcentaje de la gripe para esto época alcanza el 5%. $P(\text{gripe})=0,05$
- Por tanto, la probabilidad de que la causa de la fiebre sea gripe es del 23,3%:

$$P(\text{gripe}|\text{fiebre}) = \frac{P(\text{fiebre}|\text{gripe}) \times P(\text{gripe})}{P(\text{fiebre})} = \frac{0.7 \times 0.05}{0.15} = 0.233$$

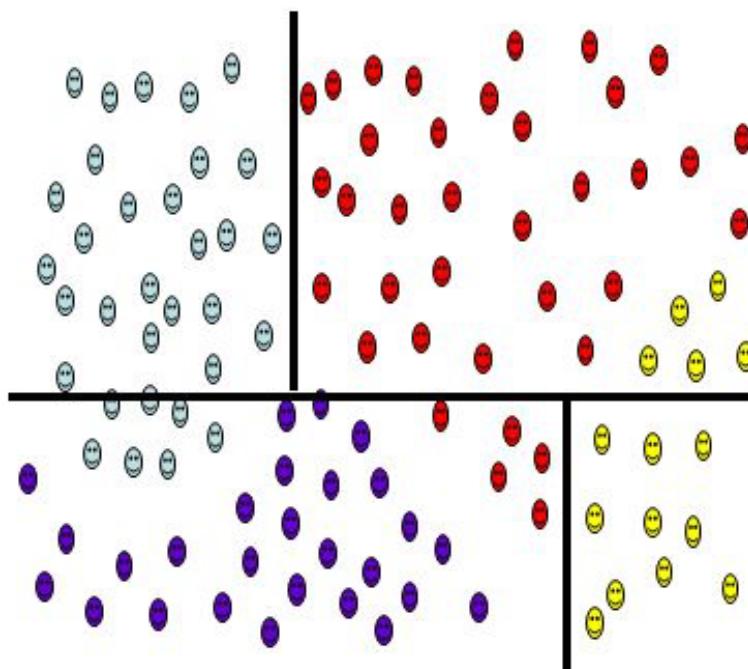
- En este ejemplo hemos trabajado con un sólo síntoma o efecto. Podemos extender el cálculo a múltiples causas:

$$P(\text{causa}|\text{efecto}_1, \text{efecto}_2, \dots, \text{efecto}_n) = P(\text{causa}) \times \prod_i P(\text{efecto}_i|\text{causa})$$

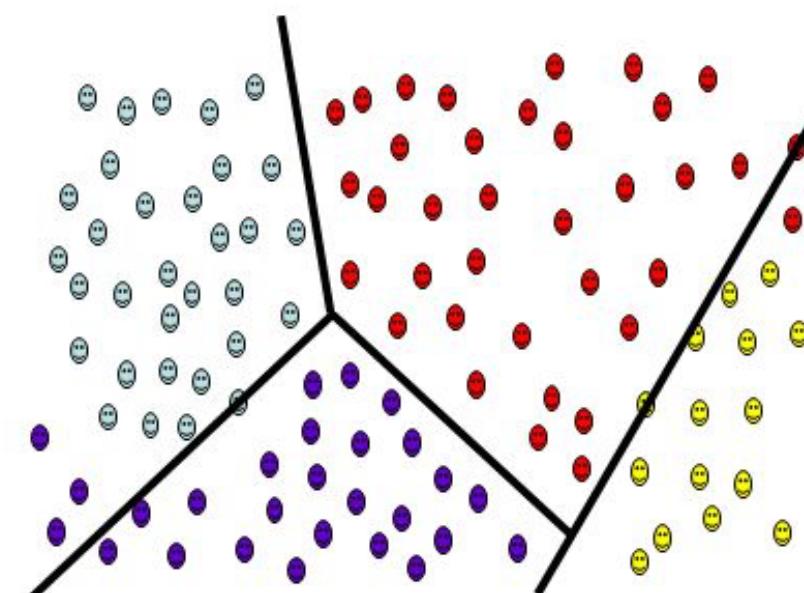
Redes Bayesianas: Naïve Bayes

- ▶ El clasificador Naïve Bayes es más expresivo que los árboles de clasificación.

Decision Trees

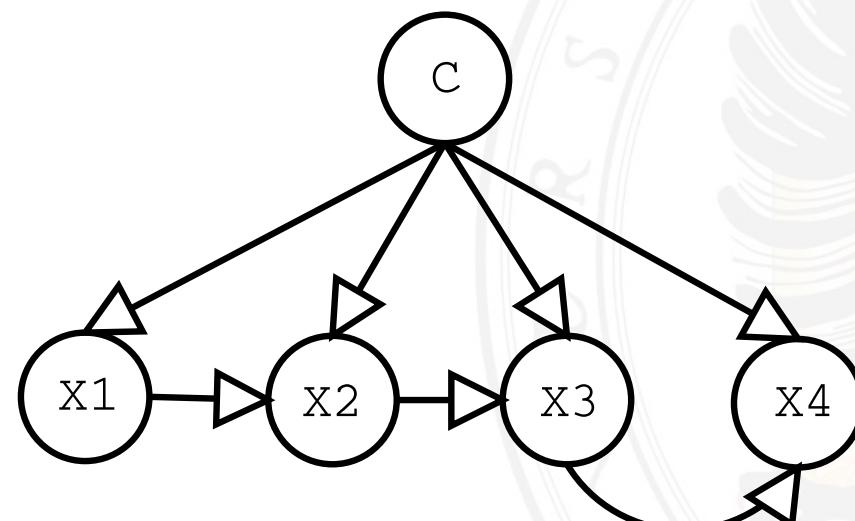


Naïve Bayes



Clasificadores Bayesianos: TAN

- ▶ Estructura de árbol aumentado (TAN)
- ▶ La restricción de que los atributos son independientes entre sí, es algo que no es cierto cuando los atributos están fuertemente correlacionados
- ▶ TAN (Tree Augmented Naïve Bayes), desarrolla una estructura de árbol entre los atributos del clasificador.
- ▶ Obtiene mejores resultados que los obtenidos por el naïve bayes.



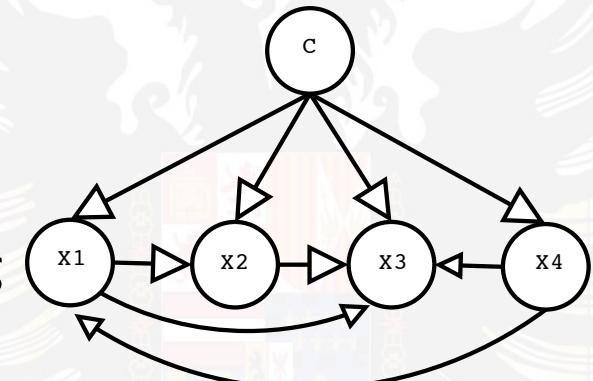
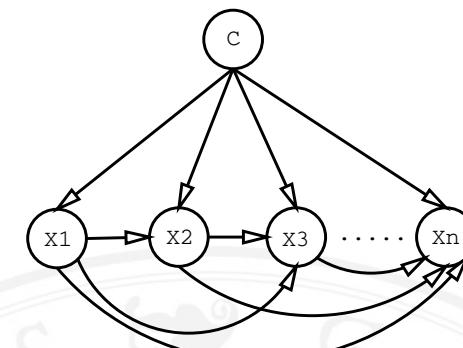
Clasificadores Bayesianos: KDB y BAN

► KDB: Clasificadores bayesianos k-dependientes.

- contiene la estructura de un clasificador naïve bayes y permite a cada atributo X_i tener un máximo de k atributos como nodos padre.
- el naïve bayes, se puede ver como un clasificador bayesiano 0-dependiente y el TAN como 1-dependiente.

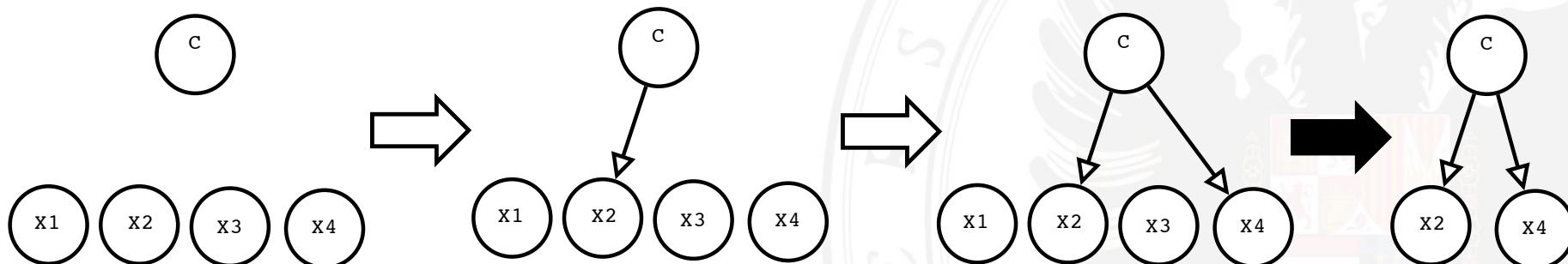
► BAN: naïve bayes aumentado a red bayesiana.

- Se construye una red bayesiana con los atributos se fuerza a que la variable clase sea padre de todos los atributos.



Clasificadores Bayesianos: Selective NB

- Naïve bayes selectivo.
- no todos los atributos tienen influencia sobre la clase.
- En el naïve bayes selectivo se construye el clasificador naïve bayes pero sólo utilizando aquellos atributos que son relevantes.
- Se eliminan aquellos atributos que sean irrelevantes y/o redundantes y que merman su eficacia.



Clasificadores Bayesianos: Redes Bayesianas

- ▶ Una red bayesiana es una representación gráfica de una distribución de probabilidad.
- ▶ Los anteriores clasificadores bayesianos se basan en el naïve bates imponiendo de alguna forma una restricción estructural.
- ▶ Podemos quitar cualquier tipo de restricción acerca de la estructura y poder usar cualquier red bayesiana como un clasificador.
- ▶ Si tenemos en cuenta que una variable es condicionalmente independiente del resto de variables dado su manto de Markov (variables padres, variables hijas y variables padre de las hijas).

