

Aprendizaje Automático



LSI - FIB

IA - Curso 2014/2015

- 1 Introducción
- 2 Aprendizaje inductivo
- 3 Árboles de decisión
- 4 Aprendizaje Bayesiano

- 1 **Introducción**
- 2 Aprendizaje inductivo
- 3 Árboles de decisión
- 4 Aprendizaje Bayesiano

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

- 1 Introducción
- 2 **Aprendizaje inductivo**
- 3 Árboles de decisión
- 4 Aprendizaje Bayesiano

Aprendizaje Inductivo

- Los métodos más utilizados en aplicaciones provienen del aprendizaje inductivo supervisado
- **Inducción:** Pasamos de lo específico a lo general
- **Supervisión:** Conocemos el concepto al que pertenece cada ejemplo
- A partir de un conjunto de ejemplos etiquetados obtenemos un modelo
- El modelo generaliza los ejemplos, representando los conceptos que definen las etiquetas
- Obtenemos lo que es común entre los ejemplos de un concepto que les diferencia de los otros

Aprendizaje Inductivo

- Desde el punto de vista formal lo que se obtiene mediante aprendizaje inductivo no es válido
- Asumimos que un número limitado de ejemplos representa las características del concepto que queremos aprender
- Solo hace falta un contraejemplo para invalidar el resultado
- Pero, !una gran parte del aprendizaje humano es inductivo!



$$F = G \frac{m_1 \cdot m_2}{r^2}$$



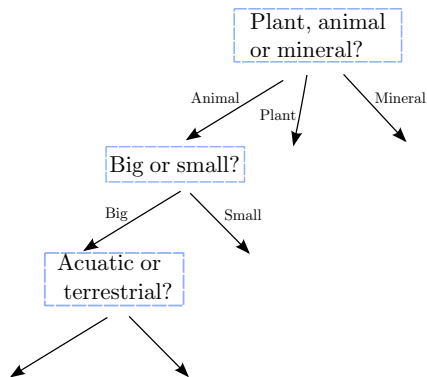
Aprendizaje Inductivo en recomendación

- Tipos de métodos de aprendizaje inductivo supervisado
 - Modelos caja blanca (podemos inspeccionar el modelo)
 - Árboles de decisión/reglas de inducción
 - Modelos probabilísticos
 - Modelos caja negra
 - Redes de neuronas artificiales
 - Máquinas de soporte vectorial
- Podemos plantear el problema como:
 - *Clasificación*: un conjunto finito de conceptos
 - *Regresión*: una función continua

- 1 Introducción
- 2 Aprendizaje inductivo
- 3 Árboles de decisión**
- 4 Aprendizaje Bayesiano

Árboles de decisión

- Podemos aprender un concepto como la aplicación de un algoritmo que busca el conjunto de preguntas que lo distingue de otros
- El árbol de preguntas nos sirve de lenguaje de representación, cada nodo es un test sobre un atributo
- Esta representación es equivalente a una FND (2^{2^n} conceptos)
- Para aprender hemos de realizar una búsqueda en el espacio de árboles de preguntas



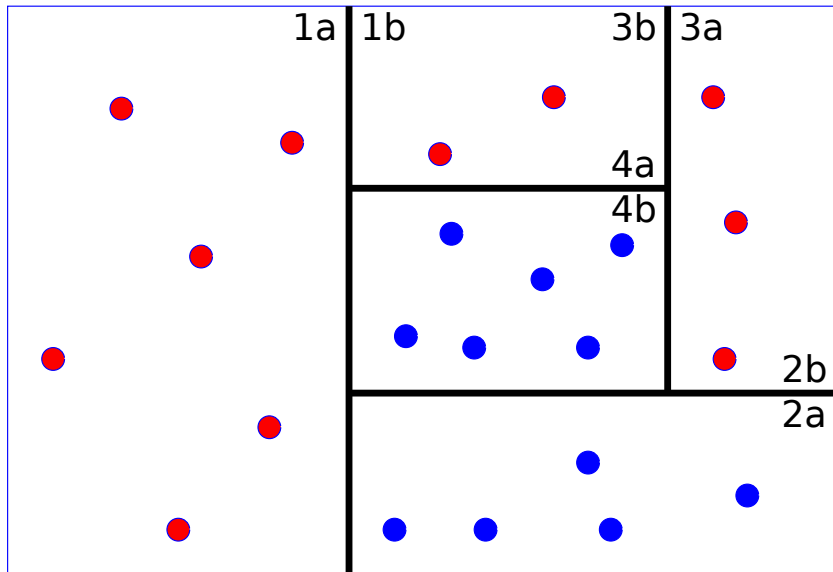
Árboles de decisión

- Buscar en el espacio de todas las FND es demasiado costoso
- Para reducir el coste computacional imponemos un *sesgo* (las descripciones que se prefieren)
- **Restricción:** Queremos el árbol que represente la mínima descripción del concepto objetivo dados los ejemplos
- **Justificación:** Este árbol será el mejor clasificando nuevos ejemplos (la probabilidad de tener preguntas innecesarias se reduce)
- **Navaja de Occam:** *“En igualdad de condiciones, la explicación más sencilla suele ser la correcta”*

Algoritmos de árboles de decisión

- Existen muchos algoritmos para aprender árboles de decisión
- El más simple es el denominado **ID3**
- Este algoritmo realiza una búsqueda por ascenso en el espacio de árboles
 - Para cada nuevo nodo de decisión un atributo es elegido y los ejemplos son distribuidos según sus valores
 - Este procedimiento es repetido recursivamente hasta que todos los ejemplos son del mismo concepto
- La selección de cada atributo es decidida mediante una función heurística que tiene preferencia a formar árboles mínimos

Algoritmos de árboles de decisión



Heurísticas para árboles de decisión

- Una heurística es un método aproximado para la solución de un problema
- Las heurísticas para árboles de decisión miden lo adecuado que es un atributo para formar un árbol mínimo
- Esta decisión se realiza de manera local (en cada nodo del árbol) aproximando el problema global
- Heurísticas utilizadas:
 - Entropía, entropía normalizada
 - GINI index
 - Rough sets
 - ...

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 $M = \{m_1, m_2, \dots, m_n\}$, 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 (**Cuántos 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
(Aprender un árbol de decisión \iff Aprender un código)
- Buscamos el mínimo código que distingue entre las clases
- Cada atributo se deberá evaluar para decidir si se le incluye en el código

Cantidad de Información como Heurística

- En cada nivel del árbol debemos evaluar qué atributo permite minimizar el código
- Este atributo será el que haga que la cantidad de información que quede por cubrir sea la menor (bits restantes por codificar)
- La elección de un atributo debería resultar en una partición donde los ejemplos en cada una de ellas estén sesgados hacia una clase
- Necesitamos una medida de la cantidad de información que no cubre un atributo (Entropía, E)

Cantidad de información

- Dado un conjunto de ejemplos \mathcal{X} y siendo \mathcal{C} su clasificación

$$I(\mathcal{X}, \mathcal{C}) = \sum_{\forall c_i \in \mathcal{C}} -\frac{\#c_i}{\#\mathcal{X}} \log\left(\frac{\#c_i}{\#\mathcal{X}}\right)$$

- Bits necesarios para codificar los ejemplos sin ninguna información adicional

Entropía

- Dado un atributo A y siendo $[A(x) = v_i]$ los ejemplos con valor v_i en el atributo

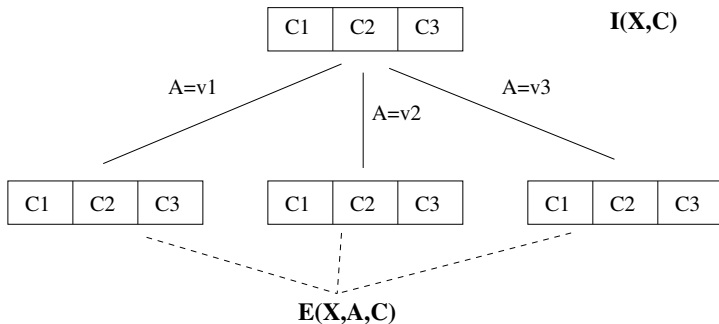
$$E(\mathcal{X}, A, \mathcal{C}) = \sum_{\forall v_i \in A} \frac{\#[A(x) = v_i]}{\#\mathcal{X}} I([A(x) = v_i], \mathcal{C})$$

- Bits necesarios para codificar los ejemplos dado un atributo
(Simplemente la suma ponderada de la cantidad de información de cada partición)

Ganancia de información

- Bits que aún han de ser codificados

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



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

ID3 algoritmo

Algoritmo: ID3 (\mathcal{X} : Ejemplos, \mathcal{C} : Clasificación, \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 cantidad de información de los ejemplos (**I**)

| **para cada** *attribute en* \mathcal{A} **hacer**

| | Calcular la entropía (**E**) y la ganancia de información (**G**)

| **fin**

| Escoger el atributo que maximiza **G** (**a**)

| Borrar **a** de la lista de atributos (\mathcal{A})

| Generar el nodo raíz para el atributo **a**

| **para cada** *partición generada por los valores del atributo a* **hacer**

| | $\text{Árbol}_i = \text{ID3}(\text{ejemplos de } \mathcal{X} \text{ con } \mathbf{a} = \mathbf{v}_i, \text{ clasificación, resto de atributos})$

| | generar una nueva rama con $\mathbf{a} = \mathbf{v}_i$ y Árbol_i

| **fin**

| **retorna** *El nodo raíz para a*

fin

Ejemplo (1)

Tomemos el siguiente conjunto de ejemplos de películas

Ej.	Década	País	Género	Gusta
1	70	USA	Drama	+
2	70	no USA	Comedia	+
3	80	no USA	Drama	−
4	90	no USA	Drama	−
5	90	no USA	Comedia	+
6	80	no USA	Acción	−
7	90	USA	Acción	−
8	70	no USA	Drama	+

Ejemplo (2)

$$I(X, C) = -1/2 \cdot \log(1/2) - 1/2 \cdot \log(1/2) = 1$$

$$\begin{aligned} E(X, decada) &= (70) \frac{3}{8} \cdot (-1 \cdot \log(1) - 0 \cdot \log(0)) \\ &+ (80) \frac{2}{8} \cdot (-1 \cdot \log(1) - 0 \cdot \log(0)) \\ &+ (90) \frac{3}{8} \cdot (-1/3 \cdot \log(1/3) - 2/3 \cdot \log(2/3)) \\ &= 0,344 \end{aligned}$$

$$\begin{aligned} E(X, pais) &= (USA) \frac{2}{8} \cdot (-1/2 \cdot \log(1/2) - 1/2 \cdot \log(1/2)) \\ &+ (noUSA) \frac{6}{8} \cdot (-1/2 \cdot \log(1/2) - 1/2 \cdot \log(1/2)) \\ &= 1 \end{aligned}$$

$$\begin{aligned} E(X, genero) &= (comedia) \frac{2}{8} \cdot (-1 \cdot \log(1) - 0 \cdot \log(0)) \\ &+ (drama) \frac{4}{8} \cdot (-1/2 \cdot \log(1/2) - 1/2 \cdot \log(1/2)) \\ &+ (accion) \frac{2}{8} \cdot (0 \cdot \log(0) - 1 \cdot \log(1)) \\ &= 0,5 \end{aligned}$$

Ejemplo (3)

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

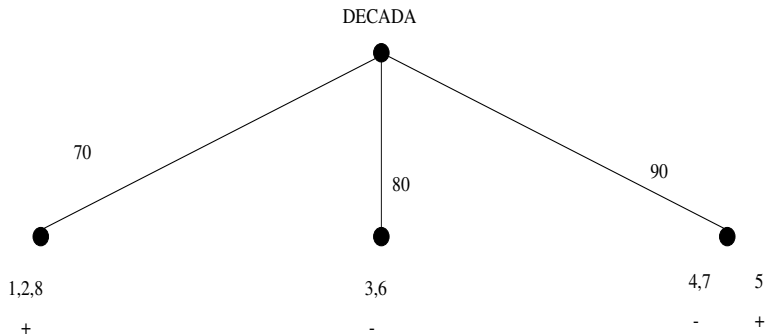
$$G(X, \text{decada}) = 1 - 0,344 = 0,656 *$$

$$G(X, \text{pais}) = 1 - 1 = 0$$

$$G(X, \text{genero}) = 1 - 0,5 = 0,5$$

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 **90s** tenemos mezclados objetos de las dos clases, por lo que repetimos el proceso con esos objetos.

Ej.	País	Género	Gusta
4	no USA	drama	—
5	no USA	comedia	+
7	USA	acción	—

Ejemplo (6)

$$I(X, C) = -1/3 \cdot \log(1/3) - 2/3 \cdot \log(2/3) = 0,918$$

$$\begin{aligned} E(X, \text{pais}) &= (\text{USA}) 1/3 \cdot (0 \cdot \log(0) - 1 \cdot \log(1)) \\ &+ (\text{noUSA}) 2/3 \cdot (-1/2 \cdot \log(1/2) - 1/2 \cdot \log(1/2)) \\ &= 0,666 \end{aligned}$$

$$\begin{aligned} E(X, \text{genero}) &= (\text{comedia}) 1/3 \cdot (0 \log(0) - 1 \cdot \log(1)) \\ &+ (\text{drama}) 1/3 \cdot (-1 \cdot \log(1) - 0 \cdot \log(0)) \\ &+ (\text{accion}) 1/3 \cdot (0 \cdot \log(0) - 1 \cdot \log(1)) \\ &= 0 \end{aligned}$$

Ejemplo (7)

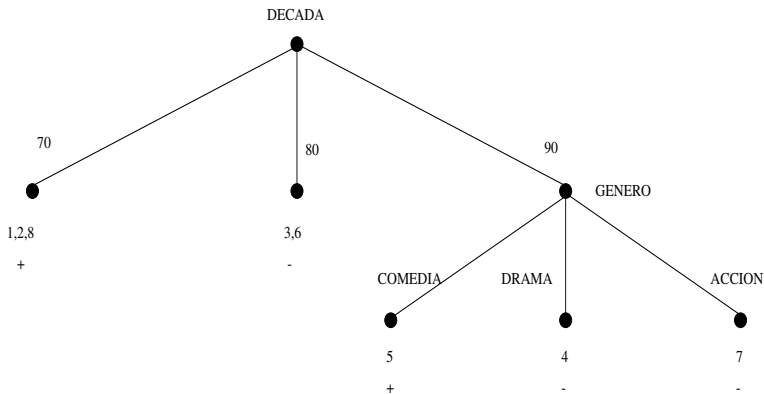
Ahora el atributo que maximiza la función es **género**.

$$G(X, pais) = 0,918 - 0,666 = 0,252$$

$$G(X, genero) = 0,918 - 0 = 0,918^*$$

Ejemplo (8)

El árbol resultante es ya totalmente discriminante y podemos usarlo como descripción del perfil



- 1 Introducción
- 2 Aprendizaje inductivo
- 3 Árboles de decisión
- 4 Aprendizaje Bayesiano**

Aprendizaje bayesiano

- Los modelos basados en árboles de decisión o reglas asumen que hay una división clara entre conceptos (solo una respuesta correcta)
- Para algunos problemas es más interesante tener decisiones difusas (soft)
- Esto se puede modelar usando distribuciones de probabilidad para representar los conceptos

Inferencia Bayesiana

- El elemento principal del aprendizaje bayesiano es el teorema de Bayes
- Este teorema liga hipótesis con observaciones

$$P(h|\mathcal{E}) = \frac{P(h) \cdot P(\mathcal{E}|h)}{P(\mathcal{E})}$$

- Esto significa que si tenemos una muestra de nuestro problema podemos evaluar nuestra hipótesis (la clasificación de nuestros datos) calculando un conjunto de probabilidades

Inferencia Bayesiana - Ejemplo

- Supongamos que queremos recomendar una novela a un amigo, esta decisión se basará en nuestra opinión acerca de la novela
- Sabemos que la probabilidad a priori de que a nuestro amigo le guste una novela es del 60 % ($p(h)$)
- Sabemos que nuestro amigo tiene un gusto similar al nuestro ($P(\mathcal{E}|h)$)
- Esa suposición es nuestro modelo de la tarea, con los siguientes parámetros estimados:
 - La probabilidad de tener nosotros una opinión positiva cuando nuestro amigo tiene una opinión positiva es del 90 %
 - La probabilidad de tener nosotros una opinión negativa cuando nuestro amigo tiene una opinión negativa es del 95 %
- A nosotros nos ha gustado la novela ¿deberíamos recomendársela a nuestro amigo? (¿cual sería su predicción?) ($P(h|\mathcal{E})$)

Inferencia Bayesiana - Ejemplo

Si enumeramos todas las probabilidades:

- $P(\text{Amigo}) = \langle 0,60; 0,40 \rangle$ (pos/neg)
- $P(\text{Nuestra} \mid \text{Amigo}=\text{pos}) = \langle 0,9; 0,1 \rangle$ (pos/neg)
- $P(\text{Nuestra} \mid \text{Amigo}=\text{neg}) = \langle 0,05; 0,95 \rangle$ (pos/neg)

Inferencia Bayesiana - Ejemplo

El teorema de bayes nos dice:

$$P(Amigo|Nuestra) = \frac{P(Amigo) \cdot P(Nuestra|Amigo)}{P(Nuestra)}$$

Dado que nuestra opinión es positiva (los datos) y dado que el resultado ha de sumar 1 para ser una probabilidad:

$$\begin{aligned} P(Amigo|Nuestra = pos) &= \langle P(A = pos) \cdot P(N = pos|A = pos), \\ &\quad P(A = neg) \cdot P(N = pos|A = neg) \rangle \\ &= \langle 0,6 \times 0,9; 0,4 \times 0,05 \rangle \\ &= \langle 0,94; 0,06 \rangle \quad (normalizada) \end{aligned}$$

Esto quiere decir que es muy probable que a nuestro amigo le vaya a gustar la novela

Aprendizaje Bayesiano

- Podemos aplicar el aprendizaje bayesiano para hacer clasificación de diferentes maneras
- Considerando solo la clase con la mayor probabilidad a posteriori (*maximum a posteriori hypothesis*, h_{MAP})

$$P(h|\mathcal{E}) = \operatorname{argmax}_{h \in \mathcal{H}} P(h) \cdot P(\mathcal{E}|h)$$

- Es exactamente lo mismo que hacemos en la estimación de las redes bayesianas

Aprendizaje Bayesiano

- El objetivo de aprendizaje es estimar la *función de densidad de probabilidad* (FDP) de los datos
- Para estimar una FDP debemos hacer ciertas suposiciones sobre:
 - El modelo de distribución que describe los atributos (continuos, discretos)
 - El modelo de distribución que describe las hipótesis
 - La dependencia entre las variables (todas independientes, algunas independientes, ...)

Aprendizaje Bayesiano - Naive Bayes

- La aproximación más simple es asumir que todos los atributos son independientes (no es cierto en general)
- La FDP para los atributos de los datos se puede expresar como:

$$P(\mathcal{E}|h) = \prod_{\forall i \in attr} P(\mathcal{E}_i|h)$$

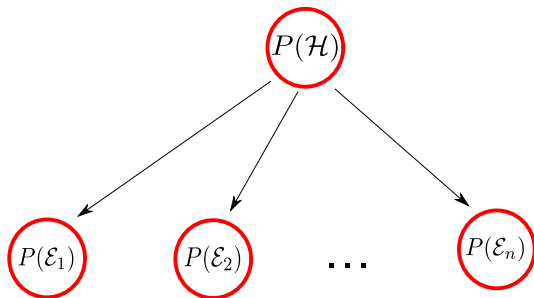
- La hipótesis maximum a posteriori puede transformarse en:

$$P(h|\mathcal{E}) = \operatorname{argmax}_{h \in \mathcal{H}} \left[P(h) \times \prod_{\forall i \in attr} P(\mathcal{E}_i|h) \right]$$

- Podemos estimar separadamente cada $P(\mathcal{E}_i|h)$ a partir de los datos

Aprendizaje Bayesiano - Naive Bayes

Este modelo es equivalente a la red bayesiana:



$$P(\mathcal{H}|\mathcal{E}) = P(\mathcal{H}) \times P(\mathcal{E}_1|\mathcal{H}) \times P(\mathcal{E}_2|\mathcal{H}) \cdots \times P(\mathcal{E}_n|\mathcal{H})$$

Naive Bayes - Algoritmo

Algoritmo: Naive Bayes

Entrada: \mathcal{E} ejemplos, \mathcal{A} atributos, \mathcal{H} hipótesis/clases

Salida : $P(\mathcal{H}), P(\mathcal{E}_{\mathcal{A}}|\mathcal{H})$

para cada $h \in \mathcal{H}$ **hacer**

$P(h) \leftarrow$ Estimar la probabilidad a priori de la clase (\mathcal{E}, h)

para cada $a \in \mathcal{A}$ **hacer**

$P(\mathcal{E}_a|h) \leftarrow$ Estimar la FDP del atributo de la clase (\mathcal{E}, h, a)

fin

fin

- Para predecir nuevos ejemplos solo hemos de calcular la probabilidad de la hipótesis
- Sorprendentemente esta aproximación funciona en algunos dominios mejor que métodos más complejos

Aprendizaje Bayesiano - Estimación en Naive Bayes

- Para atributos discretos podemos estimar $P(\mathcal{E}_i|h)$ como la frecuencia de los valores del atributo del conjunto de datos para cada clase (distribución multinomial)
- Para atributos continuos podemos estimar $P(\mathcal{E}_i|h)$ asumiendo que los datos provienen de una distribución continua (por ejemplo gaussiana) y estimar los parámetros de la distribución a partir de los datos
- Existen técnicas más avanzadas de estimación de probabilidad que no asumen una distribución de probabilidad específica (no paramétricas) y pueden evaluar una probabilidad a partir los datos mediante una estimación local

Aprendizaje Bayesiano - Estimación en Naive Bayes

Ejemplo

Ej.	Década	País	Género	Gusta
1	70	USA	Drama	+
2	70	no USA	Comedia	+
3	80	no USA	Drama	-
4	90	no USA	Drama	-
5	90	no USA	Comedia	+
6	80	no USA	Acción	-
7	90	USA	Acción	-
8	70	no USA	Drama	+

Aprendizaje Bayesiano - Estimación en Naive Bayes

Ejemplo

Década			País		Género			
70	80	90	USA	noUSA	Comedia	Drama	Acción	Gusta
1	0	.33	.5	.5	1	.5	0	+ (.5)
0	1	.66	.5	.5	0	.5	1	- (.5)

Ej: (90, USA, Drama)

$$\operatorname{argmax}_{h \in \{+, -\}} \langle 0,5 \times 0,33 \times 0,5 \times 0,5, 0,5 \times 0,66 \times 0,5 \times 0,5 \rangle =$$

$$\operatorname{argmax}_{h \in \{+, -\}} \langle 0,33, 0,66 \rangle = 0,66 \Rightarrow -$$