Personalización y Aprendizaje

Aprendizaje en Servicios y Agentes

Javier Béjar

ECSDI - 2021/2022 2Q

CS-GEI-FIB 

Adaptacion y perfilado

- Dos características interesantes de los agentes son la adaptación y la proactividad
  - Adaptación es la capacidad de cambiar su comportamiento según sus experiencias y las percepciones del entorno
  - Proactividad es la capacidad de adelantarse a las necesidades a partir de las consecuencias de su modelo del mundo

- Estas dos capacidades pasan porque el agente genere un modelo de su entorno
- Este modelo se basará en la integración de sus experiencias en una representación
- En inteligencia artificial estos modelos se construyen mediante técnicas de aprendizaje automático

- Existen múltiples escenarios en los que se puede aplicar aprendizaje automático en agentes:
  - **Perfilado de usuarios, recomendación**: Aprender las características y necesidades de un usuario asociadas a una tarea/dominio
  - **Automatización de tareas**: Aprender los pasos para realizar una tarea a través de la interacción con un usuario o el entorno
- Se pueden aplicar diferentes técnicas de aprendizaje

Recomendación/Perfilado

- Un ámbito interesante del uso de aprendizaje en agentes/servicios es el perfilado de usuarios y la recomendación
- Se aprenden las preferencias del usuario para poder personalizar el funcionamiento del agente/servicio
- Las aplicaciones más comunes son en agentes que sirven para el acceso a algún tipo de contenido (productos, información)
- Puede realizarse a partir de la información que se obtiene de manera individual por usuario o de manera colectiva

- **Perfil individual estático**: Se recolecta una vez la información de un usuario y se utiliza como modelo para sus preferencias
- **Perfil individual dinámico**: Se observa o se obtiene feedback del usuario y se adapta el perfil inicial a las observaciones
- **Perfil colaborativo**: Se organizan a los usuarios en perfiles de acuerdo con sus características y se utiliza el modelo obtenido del colectivo para determinar sus preferencias

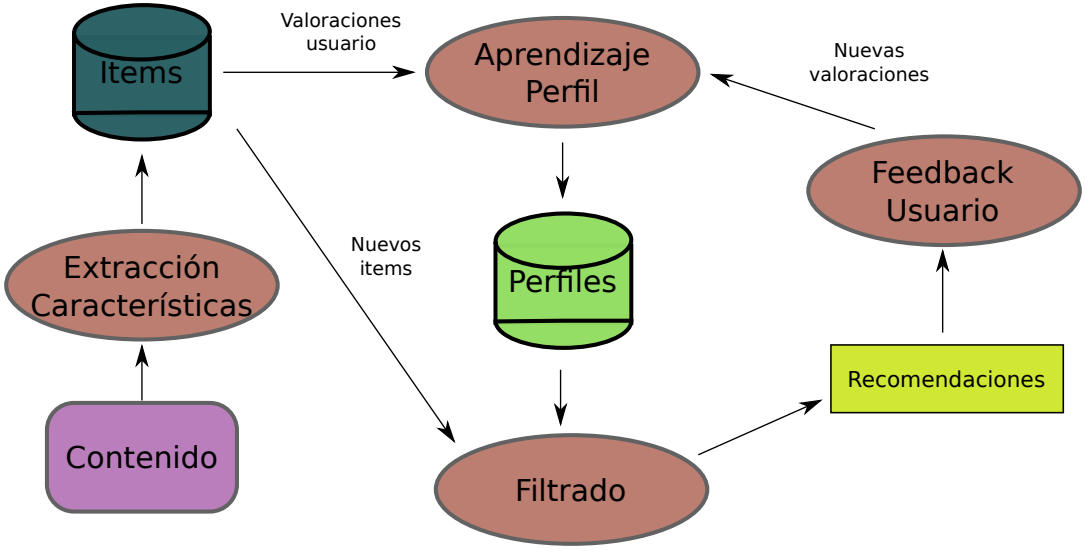
- **Filtrado del correo no deseado**
- El contenido son los mensajes de correo que recibe un usuario
- El comportamiento observable es el marcado como correo no deseado de mensajes
- El perfil individual son las características que diferencian los correos no deseados para el usuario
- El perfil se aplica para etiquetar nuevos correos

- **Recomendación de productos en una tienda**
- El contenido son los productos de la tienda
- El comportamiento observable es el historial de compra y navegación de los usuarios
- El perfil colectivo se puede obtener explícita (eg: partición de usuarios) o implícitamente (eg: usuarios con compras similares)
- El perfil se aplica para recomendar al usuario compras realizadas por usuarios con un comportamiento similar

- Las características en un perfil dependen del dominio de aplicación
- Pueden incluir:
  - Información propia del usuario (eg: info. demográfica)
  - Información que representa el contenido que se va a recomendar (eg: descripción de productos, texto...)
- La representación depende de la estructura del contenido
  - No estructurada (vector de atributos)
  - Estructurada (grafos de relaciones, secuencias...)

- Recomendación basada en contenido
  - Se elabora un perfil explícito (modelo) de los usuarios a partir de sus características y las del contenido
  - Los nuevos elementos se recomiendan o no dependiendo de la respuesta del modelo
- Filtrado colaborativo
  - No se elabora un perfil explícito del usuario
  - Se determina la adecuación de nuevo contenido a partir de la similaridad del usuario con otros usuarios o contenidos y sus evaluaciones de estos

- Basada en el análisis de las descripciones de contenido valorado por el usuario
- El perfil será una representación de las preferencias del usuario a partir de esas descripciones
- Este perfil permite obtener la predicción de la valoración del usuario para nuevo contenido
- Esta valoración permite tomar la decisión sobre su recomendación


Items
Valoraciones usuario
Aprendizaje Perfil
Nuevas valoraciones
Feedback Usuario
Extracción Características
Nuevos items
Perfiles
Recomendaciones
Contenido
Filtrado

- Ventajas:
  - Las recomendaciones de un usuario son independientes del resto
  - Se puede obtener una explicación de la recomendación a partir del perfil
  - Se pueden recomendar contenidos nuevos que no han sido valorados por nadie

- Inconvenientes:
  - La calidad del perfil depende de las características usadas
  - Sobreespecialización en las recomendaciones, no podemos recomendar contenidos muy diferentes de los que ya conocemos
  - Hacen falta un numero inicial de valoraciones por parte del usuario para poder recomendar eficazmente (*cold start*)

- El contenido puede tener una representación explícita (estructurada)
  - Una ontología define la descripción del contenido
  - eg: Productos en una tienda (libros, discos...)
- El contenido puede tener una representación no estructurada
  - Se han de utilizar técnicas especializadas de extracción de características
  - ej.: Texto de noticias $\rightarrow$ técnicas de recuperación de la información (Information Retrieval)

- El feedback puede ser explícito:
  - A partir de una valoración cualitativa (e.g.: gusta, no gusta)
  - A partir de una valoración cuantitativa (e.g.: una puntuación)
  - A partir de una valoración no estructurada (e.g.: el texto de una opinión)
- El feedback puede ser implícito:
  - Acciones del usuario que podemos transformar en una valoración (eg: tiempo de lectura de una noticia)

Aprendizaje y recomendación

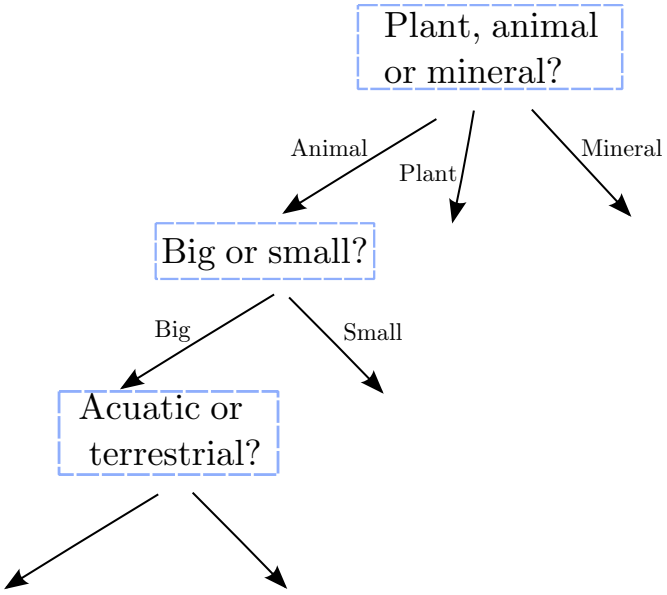
- Los métodos más utilizados para la recomendación basada en contenidos provienen del aprendizaje inductivo supervisado
- **Inducción**: Pasamos de lo específico a lo general
- **Supervisión**: Conocemos el concepto al que pertenece cada ejemplo

- A partir de 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 conceptos

- Métodos usados en recomendación basada en contenido
  - 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 definir el problema como:
  - Clasificación: predecir un conjunto finito de conceptos
  - Regresión: predecir una función continua

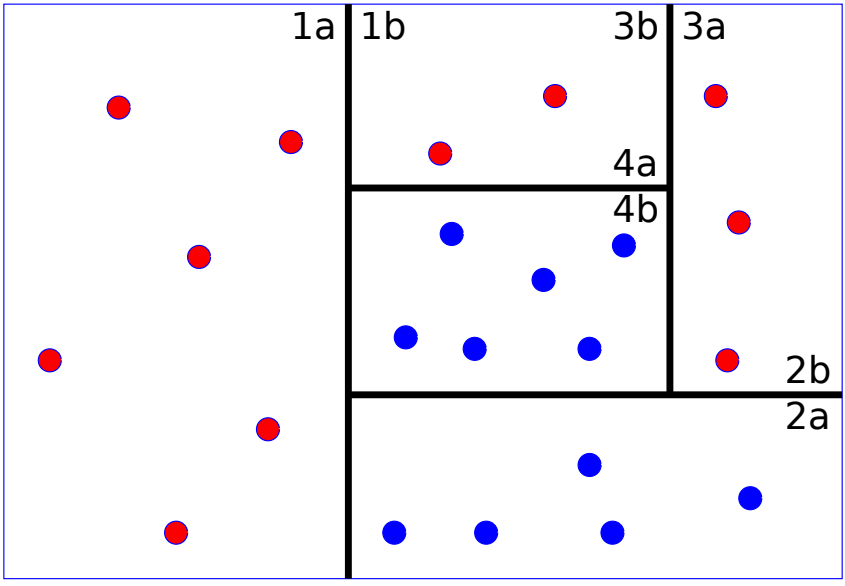
Árboles de decisión

- Aprender un concepto como la búsqueda del conjunto de preguntas que lo distingue de otros
- Estas preguntas se pueden organizar de manera jerárquica (árbol)
- El árbol de preguntas nos sirve de lenguaje de representación, cada nodo es un test sobre un atributo
- Representacion es equivalente a una FND (2^{2^n} conceptos posibles)
- Búsqueda en el espacio de árboles de preguntas


Plant, animal or mineral?
Animal
Plant
Mineral
Big or small?
Big
Small
Acuatic or terrestrial?

- Buscar en el espacio de todas las FND es muy costoso
- Para reducir el coste computacional imponemos un sesgo
- **Restricción:** Queremos el árbol que represente la mínima descripción del concepto objetivo dados los ejemplos
- **Justificacion:** 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*"

- El algoritmo **ID3** realiza una búsqueda por ascenso en el espacio de árboles
  - Para cada nuevo nodo de decisión se elige un atributo 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


1a
1b
3b
3a
4a
4b
2b
2a

- 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
  - ...

- La teoría de la información estudia los mecanismos de codificación de mensajes y el coste de su transmisión
- Dados un conjunto de mensajes $M = \{m_1, m_2, ..., m_n\}$, 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 interpreta como la información necesaria para distinguir entre los mensajes de M (Cuantos bits de información son necesarios para codificarlos)

- Podemos hacer la analogía con codificación de mensajes:
  - las clases son los mensajes
  - la proporción de ejemplos de cada clase su probabilidad
- Árbol de decisión = codificación que distingue las clases

  (Aprender un árbol de decisión $\Longleftrightarrow$ Aprender un código)
- Buscamos el mínimo código que distingue entre las clases
- Cada atributo se evalúa para decidir si se le incluye

- La elección se realiza en cada punto de decisión
- Elegir un atributo si hace que la cantidad de información que quede por cubrir sea la menor (bits restantes por codificar)
- La elección resulta en una decisión donde los ejemplos para cada posibilidad están sesgados hacia una clase
- Una heurística calcula la cantidad de información que no cubre un atributo (Entropia, E)

- Bits necesarios para codificar los ejemplos $\mathcal{X}$ y siendo $\mathcal{C}$ su clasificación, sin ninguna información adicional

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

- Bits necesarios para codificar los ejemplos dado un atributo A y siendo $[A(x) = v_i]$ los ejemplos con valor v_i

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

(Suma ponderada de la cantidad de I para cada partición)

Algorithm: ID3 ($\mathcal{X}$: Ejemplos, $\mathcal{C}$: Clasificación, $\mathcal{A}$: Atributos)

if *todos los ejemplos son de la misma clase*
then
    return *una hoja con el nombre de la clase*
else
    Calcular la cantidad de información de los ejemplos (**I**)
    foreach *atribute en* $\mathcal{A}$ **do**
        Calcular la entropia (**E**) y la ganancia de información (**G**)
    Escoger el atributo que maximiza **G** (**a**)
    Borrar **a** de la lista de atributos ($\mathcal{A}$)
    Generar el nodo raíz para el atributo **a**
    foreach *partición generada por los valores del atributo* **a do**
        Árbol$_i$=ID3($\mathcal{X}[a] = v_i$, $\mathcal{C}$,$\mathcal{A} - a$)
        generar una nueva rama con **a**=**v**$_i$ y Árbol$_i$
    return *El nodo raíz para* ***a***

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	+

Década	N ejemplos	Gusta	no Gusta
70	3/8	3/3	0/3
80	2/8	0/2	2/2
90	3/8	1/3	2/3

G(X,década)= 1 - 0.34= 0.65

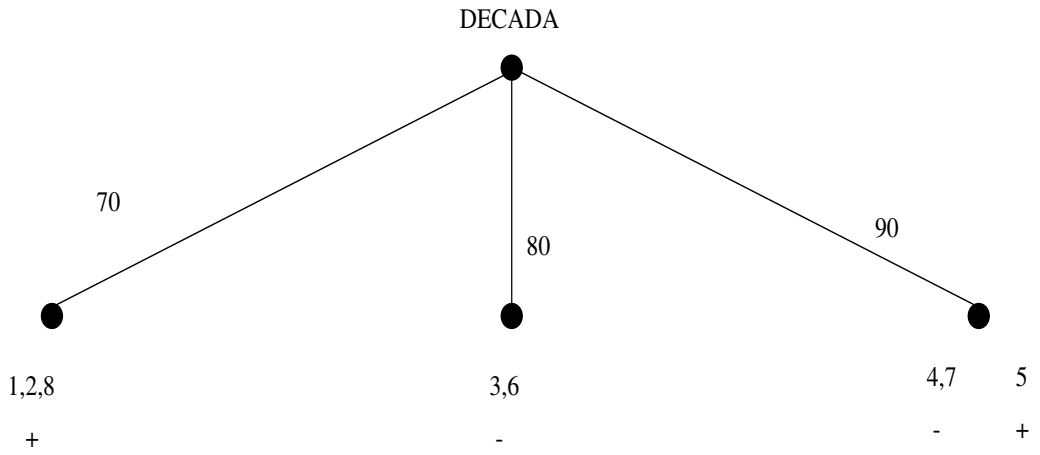
País	N ejemplos	Gusta	no Gusta
USA	2/8	1/2	1/2
no USA	6/8	3/6	3/6

G(X,país)= 1 - 1 = 0

Género	N ejemplos	Gusta	no Gusta
comedia	2/8	2/2	0/2
drama	4/8	2/4	2/4
acción	2/8	0/2	2/2

G(X,genero)= 1 - 0.5 = 0.5

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



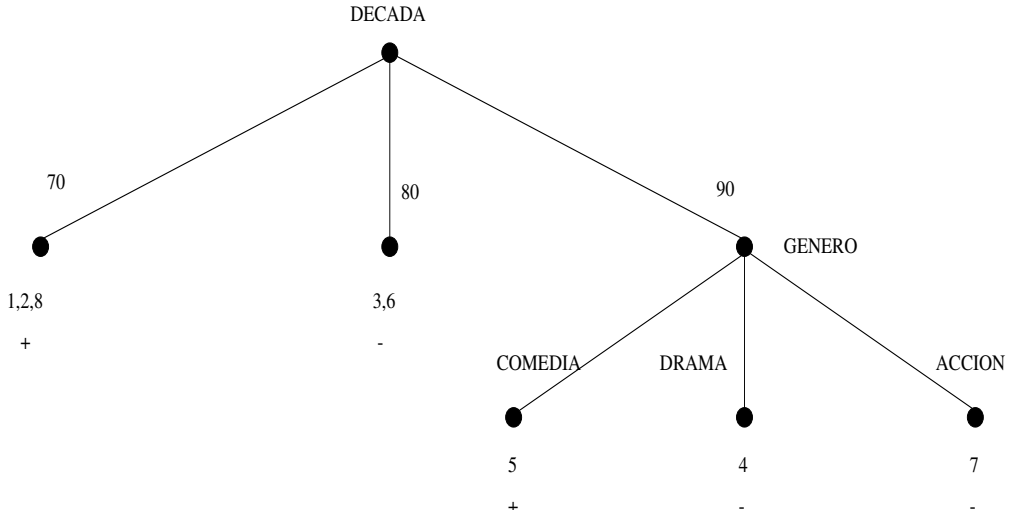
Ahora solo en el nodo correspondiente al valor **90s** tenemos mezclados objetos de las dos clases, por lo que repetimos el proceso con esos objetos.

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

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

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

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



Aprendizaje bayesiano

- Los modelos basados en árboles de decisión o reglas asumen una división perfecta entre conceptos
- Para algunos problemas es más interesante tener decisiones difusas (soft)
- Esto se puede modelar usando distribuciones de probabilidad para representar los conceptos
- Asume que con una muestra de datos problema podemos obtener un modelo para evaluar nuestras hipótesis (la clasificación de nuestros datos) estimando probabilidades

- El mecanismo de razonamiento del aprendizaje bayesiano es el **teorema de Bayes**
- Este teorema liga hipótesis con observaciones

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

- Obtiene de un modelo probabilístico de un problema una estimación de la probabilidad de una decisión

- Supongamos que queremos recomendar una novela a un amigo
- Podemos basar esta decisión en:
  - Nuestra opinión sobre la novela (evidencia)
  - Probabilidad a priori de los gustos de nuestro amigo
  - Como se parecen nuestros gustos a los de nuestro amigo modelado como la probabilidad conjunta de los gustos de ambos (asumimos que correlación es causalidad)

- Sabemos que la probabilidad de que a nuestro amigo le guste una novela es del 60 % ($p(h)$, probabilidad a priori)
- Sabemos que nuestro amigo tiene un gusto similar al nuestro ($P(\mathcal{E}|h)$, probabilidad condicionada)
- Suponemos que parámetros estimados del $P(\mathcal{E}|h)$ son:
  - Probabilidad de tener nosotros una opinión positiva si nuestro amigo tiene una opinión positiva es del 90 %
  - La probabilidad de tener nosotros una opinión negativa si 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?) ($P(h|\mathcal{E})$)

Si enumeramos todas las probabilidades relevantes:

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

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:

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

Es muy probable que a nuestro amigo le vaya a gustar la novela

- El objetivo de aprendizaje es estimar la función de densidad de probabilidad (FDP) de los datos
- Estimar una FDP necesita 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...)

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

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

- La estimación del modelo para cada atributo se puede hacer por separado $P(\mathcal{E}_i|h)$ a partir de los datos
- La probabilidad de un conjunto de hipótesis se expresa:

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

Algorithm: Naive Bayes

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

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

foreach $h \in \mathcal{H}$ **do**

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

  foreach $a \in \mathcal{A}$ **do**

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

- Predecir nuevos ejemplos implica calcular la probabilidad de las hipótesis (aplicar el teorema de Bayes)

- Atributos discretos: $P(\mathcal{E}_i|h)$ se estima a partir de la frecuencia de los valores del atributo en los de datos para cada clase (distribución multinomial)

- Atributos continuos: $P(\mathcal{E}_i|h)$ se estima asumiendo una distribución modelo continua (e.g. gausiana) y estimando sus parámetros con los datos

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	+

Década			País		Género			$P(\mathcal{H})$
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)

$$\underset{h\in\{+,-\}}{\text{argmax}}\langle 0{,}5 \times 0{,}33 \times 0{,}5 \times 0{,}5; 0{,}5 \times 0{,}66 \times 0{,}5 \times 0{,}5\rangle =$$

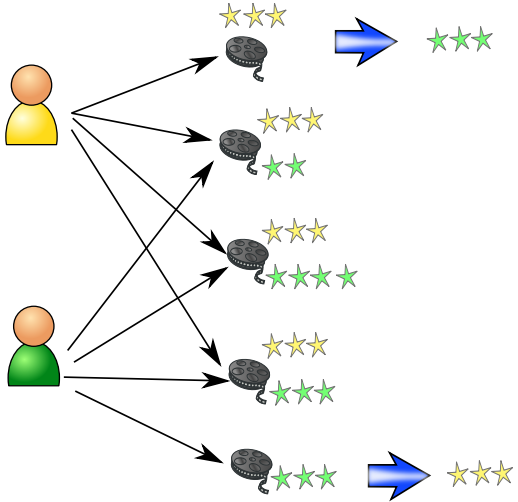
$$\underset{h\in\{+,-\}}{\text{argmax}}\langle 0{,}33; 0{,}66\rangle = 0{,}66 \Rightarrow -$$

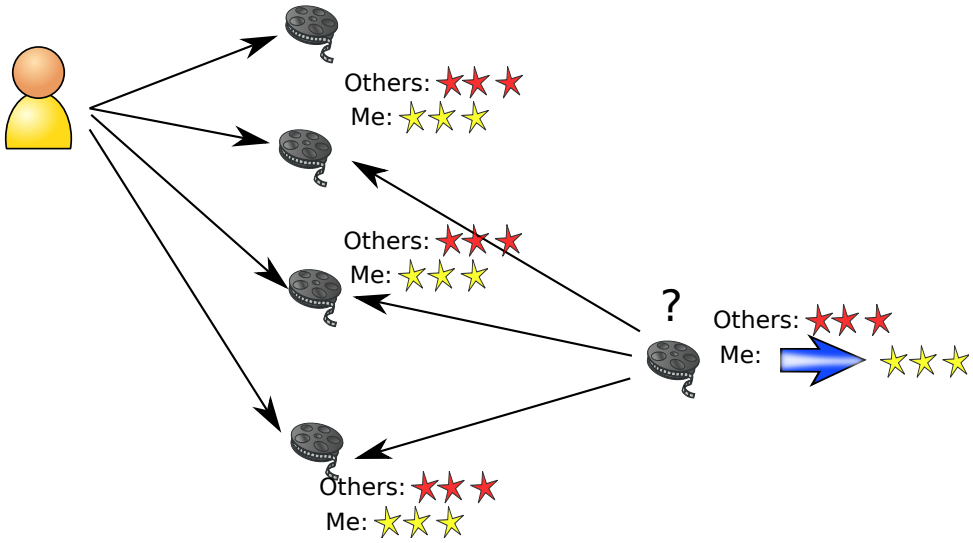
- El modelo bayesiano es utilizado frecuentemente en la clasificación de texto
- El filtrado de correo no deseado se puede ver como un proceso de recomendación:
  - **Contenido**: Correos electrónicos
  - **Recomendación**: Leerlo o no
- En el caso del texto/documentos el contenido se representa por la frecuencia de sus palabras (*Bag of words*)

Filtrado colaborativo

- Se basa no solo en la información del usuario, sino también en la de los usuarios que más se le parecen
- Este método permite superar el problema de no tener una buena descripción a partir del contenido
  - Asumimos que a personas similares les gustan contenidos similares
- Esto permite recomendar contenidos que no se parezcan a los que el usuario ya ha calificado (*serendipia*)
  - Si ha sido calificado por personas similares a él, seguramente se le puede recomendar

- Basado en las valoraciones de los ítems del contenido
- La recomendación utiliza una combinación de las valoraciones de los k usuarios/ítems más parecidos
- La recomendación se puede interpretar a partir de las calificaciones de los vecinos
- La valoración de un nuevo ítem se puede obtener
  - **Usuario con usuario**: Buscamos usuarios que hayan calificado el item y que tengan un conjunto de ítems calificados similares, con valoraciones similares
  - **Ítem con ítem**: Buscamos los ítems del usuario que son más similares al que queremos recomendar




Others:
Me:
Others:
Me:
?
Others:
Me:
Others:
Me:

- Si las valoraciones son un valor continuo podemos calcular la valoración como una suma ponderada de las valoraciones de los vecinos
- Si las valoraciones son valores discretos se puede aplicar una estrategia de voto ponderado
- La ponderación se obtiene de la similaridad/distancia entre los elementos que estamos combinando (usuarios/ítems)

	item 1	item 2	item 3	item 4	item 5	item 6	item 7	item Recom
Usuario Recom	G	NV	NG	G	G	G	NV	?
Usuario 1	G	NG	NV	NV	G	G	NV	G
Usuario 2	NG	NG	NG	G	G	NG	G	NG
Usuario 3	G	G	NG	NG	G	G	NV	G
Usuario 4	NG	G	G	NV	NG	G	NV	G
Usuario 5	G	NV	NG	NV	NG	NG	G	G

G= Gustado/NG=No Gustado/NV=No Valorado

Asumiremos que tenemos una función de similaridad que asigna un valor 1 si las valoraciones coinciden, un -1 si no coinciden y un 0 si alguno de los usuarios no ha valorado el ítem.

	item 1	item 2	item 3	item 4	item 5	item 6	item 7	SIM
Usuario 1	1	0	0	0	1	1	0	3
Usuario 2	-1	0	1	1	1	-1	0	1
Usuario 3	1	0	1	-1	1	1	0	3
Usuario 4	-1	0	-1	0	-1	1	0	-2
Usuario 5	1	0	1	0	-1	-1	0	0

3 Vecinos: $(3 \times G + 3 \times G, 1 \times NG) \Rightarrow G$

	item 1	item 2	item 3	item 4	item 5	item 6	item 7	item Recom
Usuario Recom	3	?	1	4	5	3	?	?
Usuario 1	4	1	?	?	4	4	?	4
Usuario 2	1	2	1	3	5	1	4	2
Usuario 3	4	3	1	2	4	5	?	5
Usuario 4	1	4	5	?	1	4	?	5
Usuario 5	5	?	2	?	2	1	4	4

Puntuación = [1...5]

Distancia = suma del valor absoluto de la diferencia entre las valoraciones

La distancia cuando alguna valoración es desconocida es un valor desconocido, por lo que reescalamos la distancia multiplicando por el número de ítems dividido por el número de ítems en común

	it 1	it 2	it 3	it 4	it 5	it 6	it 7	DIST
Usuario 1	1	?	?	?	1	1	?	3 * (7/3) = 7
Usuario 2	2	?	0	1	0	2	?	3 * (7/5) = 4.2
Usuario 3	1	?	0	2	1	2	?	6 * (7/5) = 8.4
Usuario 4	2	?	4	?	4	1	?	11 * (7/4) = 19.25
Usuario 5	2	?	1	?	3	2	?	8 * (7/4)= 14

Ahora podemos calcular la valoración eligiendo de nuevo los tres usuarios más similares y calculándola como la suma ponderada por el inverso de la distancia de las tres puntuaciones

$$rec = \frac{\sum_{i=1..,3} p_i/d_i}{\sum_{i=1..,3} 1/d_i} = \frac{\frac{4}{7} + \frac{2}{4,2} + \frac{5}{8,4}}{\frac{1}{7} + \frac{1}{4,2} + \frac{1}{8,4}} = 3,27$$

Si queremos tener un valor entero podemos simplemente truncar al entero más cercano, lo que nos daría una puntuación de 3