INGENIERIA INDUSTRIAL
UNIVERSIDAD DE CHILE

fcfm

**Pauta Control 1**
Primavera 2009

IN831 - Web Mining
*Juan D. Velásquez y Víctor Rebolledo Lorca*
Auxiliar: *Iván Videla*

NOMBRE: _____

**INDICACIONES**: *Guarde todos sus apuntes, deje su mochila adelante, sólo necesita lápiz y borrador. Las hojas de respuestas están adjuntas.*

# Pregunta 1

1. (2 puntos) Durante el curso se explicó que el proceso de sessionización no está libre de errores. Explique las 4 causas principales de ello y, para cada una de ellas, indique posibles soluciones para neutralizar su efecto.

**Respuesta:**

- **La Web es asincrona** y, por tanto, un usuario no puede ser identificados a priori. A menos que se implemente un sistema de registro en el sitio y el usuario se identifique con su password, es imposible conocer quién es la persona detrás de la sesión. Por ejemplo, resultaría imposible conocer si se trata del padre o el hijo cuando ambos comparten el mismo computador y conexión a Internet.
- **Proxies y Firewall:** Estos dispositivos enmascaran bajo una misma dirección IP a todos los usuarios que están detrás de ellos. Esto impide que dos usuarios con distinta IP y detrás de un proxie o firewall puedan ser identificados navegando por Internet, pues la IP del dispositivo será la registrada en los web logs. Por consecuencia, las acciones de ambos serán asignadas a la sesión de un usuario que no existe. Una posible solución a esto es incorporar un medio de indentificación invasivo como una cookie en el browser del navegador del visitante.
- **Web crawlers:** Corresponden a programas o spyder robots que recorren la Web guardando copias de las páginas de diversos sitios de forma automatizada. Son frecuentemente usados por motores de búsqueda como *Google* o *Yahoo!* para recopilar las últimas versiones de las páginas web e indexarlas en sus servidores,acelerando con ello las búsquedas. Su paso por los sitios deja los web logs con gran cantidad de registros de peticiones en pocos segundos. Esto puede ser solucionado filtrando las entradas mediante el campo "Agent". De hecho, existen bases de datos con los nombres de los crawlers más utilizados. De no encontrarse, se pueden filtrar de acuerdo al comportamiento de las sesiones identificadas. Por ejemplo: *es poco probable que un usuario haya recorrido 100 paginas en 2 segundos.*
- **La memoria caché:** Los web browser guardan en la memoria caché las páginas visitadas recientemente. Por ejemplo, si un usuario desea retornar a una página recientemente visitada mediante el Back Button del navegador, éste le devuelve la versión guardada y, por tanto, no hace petición al servidor, quedando sin registrar esa acción en los web logs. También existen cachés corporativas en las que un servidor central

almacena las páginas visitadas por todos los empleados de una corporación. Este problema puede subsanarse usando heuristicas que completen las sesiones de acuerdo a la estructura del sitio.

2. (2 puntos) Comente la siguiente afirmación: *"Haciendo uso de un web crawler es posible rescatar el contenido HTML y multimedia de las páginas web, además de la estructura de hyperlinks presente en ellas. En ese sentido, será posible identificar las páginas hub y autoritativas de la Web"*

   **Respuesta:** Esta afirmación no es del todo verdadera por la manera en la que el crawler recorre las páginas web. En efecto, el crawler recorre las páginas como si esta fuera un árbol. En ese sentido, es capaz de identificar claramente las *páginas hub* pues para cada página visita todos los hiperlinks presentes en ésta. Sin embargo, no logra determinar del todo la *autoratitividad* de las páginas, pues para ello debiera recorrer toda la Web, lo cual no es del todo posible. Precisamente, la autoridad de una página está dada por la cantidad de hyperlinks que la apuntan.

3. (2 puntos) Acerca del Vector Space Model VSM, responda:

   - (1 punto) ¿Qué rol juegan los metadatos?

     **Respuesta:** Los metadatos permiten caracterizar con contenido textual aquellos objetos multimedia que no pueden ser considerados mediante técnicas de minería de texto provenientes de Information Retrieval. En ese sentido, los metadatos permiten representar aquellos objetos como representaciones vectoriales de palabras haciéndolos comparables con el texto de las páginas web.

   - (1 punto) ¿Qué supuesto está subyacente con el VSM? En consecuencia: ¿Qué información se pierde bajo esta representación?

     **Respuesta:** El **Vector Space Model** considera que todas las palabras, una vez tokenizadas y lematizadas (llevadas a su raíz o lema), son independientes entre sí, es decir, que no tienen significado común o parecido. Por tanto, se pierde la semántica de las palabras. Por ejemplo, se da el mismo peso a sinónimos, antónimos, homónimos, etc.

INGENIERIA INDUSTRIAL
UNIVERSIDAD DE CHILE

fcfm

**Pauta Control 1**
Primavera 2009

IN831 - Web Mining
*Juan D. Velásquez y Víctor Rebolledo Lorca*
Auxiliar: *Iván Videla*

NOMBRE: _____

**INDICACIONES**: *Guarde todos sus apuntes, deje su mochila adelante, sólo necesita lápiz y borrador. Las hojas de respuestas están adjuntas.*

# Pregunta 2: Técnicas de Data Mining

1. (1 puntos) Explique el concepto de "sobreajuste" en el contexto de los algoritmos de machine learning. Para el caso de las redes neuronales, explique cuáles serían las posibles causas de este y cómo se puede subsanar?

   **Respuesta:** Se produce *sobreajuste* o *sobreaprendizaje* cuando el algoritmo memoriza las instancias del conjunto de entrenamiento perdiendo capacidad de *generalización* para clasificar las instancias del conjunto de testeo. En otras palabras, el algoritmo tiene 100% de eficacia en la clasificación del conjunto de entrenamiento, pero tiene un error no aceptable para clasificar el conjunto de testeo. En el caso de las redes neuronales, las posibles causas podrían ser las siguientes:

   - Un número excesivo de capas ocultas en la red.
   - Un número excesivo de neuronas en las distintas capas de la red
   - Un múmero excesivo de iteraciones o épocas.
   - Un mal muestreo en la selección de las instancias de entrenamiento y testeo

   Para solucionar lo anterior, habría que ajustar la red en el siguiente orden:

   - En primer lugar, reducir el número de capas ocultas en la red.
   - En segundo lugar, reducir el número de neuronas en la red siguiendo una caida de un 50% entre capas. Por ejemplo: 8 neuronas en la capa de entrada, 4 neuronas en la capa oculta y 2 neuronas en la capa de salida.
   - Estimar el número óptimo de épocas con las curvas de aprendizaje.
   - Por último, si lo demás no funciona, hacer un nueva selección de instancias de entrenamiento y testeo.

2. (1 puntos) Demuestre matemáticamente que el modelo de Naive-Bayes cumple con el principio de la "Navaja de Occam". En otras palabras, que la hipótesis más probable dada por el teorema es también la más simple.

   **Respuesta:** El teorema de Bayes nos permite actualizar la creencia que tenemos en un suceso a la luz de nuevos datos u observaciones. En otras palabras, permite pasar de la probabilidad a priori $P(suceso)$ a la probabilidad a posteriori $P(suceso|observaciones)$. La probabilidad a priori puede verse como la probabilidad que fijamos sin saber nada más, en cambio la

probabilidad a posteriori es aquella que obtenemos tras conocer cierta información, por tanto, es un refinamiento de nuestro conocimiento. Teniendo en cuenta esto, el teorema de Bayes dice lo siguiente:

$$P(h|D) = \frac{P(D|h) \times P(h)}{P(D)} \tag{1}$$

Donde tenemos las probabilidades a priori de la hipótesis h, $P(h)$ y de las observaciones D del conjunto de entrenamiento $P(D)$, además de las probabilidades condicionales $P(h|D)$ y $P(D|h)$. En el contexto de la clasificación, nos interesa obtener la hipótesis más probable dado un conjunto de observaciones, es decir, aquella que tiene máxima probabilidad a posteriori dados los atributos (**hipótesis máxima a posteriori o MAP**)

$$h_{MAP} = argmax_h \frac{P(D|h) \times P(h)}{P(D)} \tag{2}$$

Por otro lado, de acuerdo a la navaja de Occam "Ante dos hipótesis que expliquen un mismo suceso, simpre se debe elegir la más simple". La simpleza de una hipótesis puede ser definida mediante el principio de la longitud de descripción minima o **MDL (Minimum Description Length)**. Esta tería consiste en calcular el tamaño en bits (unidad de información) de la descripción de una hipótesis más la descripción de los ejemplos que no son cubiertos (excepciones). Formalmente, el principio MDL recomienda seleccionar la hipótesis h que minimice la siguiente expresión:

$$K(h) + K(D|h) \tag{3}$$

Donde $K(h)$ es la complejidad en bits de describir la hipótesis $h$ y $K(D|h)$ es la complejidad de describir la evidencia D a partir de la hipótesis $h$ (lo que incluye las excepciones). Por lo tanto, la hipótesis más simple será:

$$h = argmin_h K(h) + K(D|h) \tag{4}$$

Volviendo a la hipótesis más probable dada por Bayes, tenemos:

$$h_{MAP} = argmax_h \frac{P(D|h) \times P(h)}{P(D)} \tag{5}$$
$$h_{MAP} = argmax_h P(D|h) \times P(h) \tag{6}$$
$$h_{MAP} = argmax_h \log_2 P(D|h) + \log_2 P(h) \tag{7}$$
$$h_{MAP} = argmin_h - \log_2 P(D|h) - \log_2 P(h) \tag{8}$$
$$h_{MAP} = argmin_h K(D|h) + K(h) \tag{9}$$
$$h_{MAP} = argmin_h K(h) + K(D|h) \tag{10}$$

Donde $P(h) = 2^{-K(h)}$ y $P(D|h) = 2^{-K(D|h)}$. Nótese que la probabilidad a priori de las observaciones del conjunto de entrenamiento P(D) es una constante, por lo tanto, se puede eliminar la división desde la equación 5 a la 6.

Table 1: Conjunto de entrenamiento

| **Temperatura** | **Gusto** | **Tamaño** | **Sabroso** |
|---|---|---|---|
| Caliente | Salado | Pequeño | NO |
| Frio | Dulce | Grande | NO |
| Frio | Dulce | Grande | NO |
| Frio | Acido | Pequeño | YES |
| Ambiente | Acido | Pequeño | YES |
| Ambiente | Salado | Grande | NO |
| Ambiente | Acido | Grande | YES |
| Frio | Dulce | Pequeño | YES |
| Frio | Dulce | Pequeño | YES |
| Ambiente | Salado | Grande | NO |

3. (2 punto) Considere el conjunto de entrenamiento de la Tabla 1:

   Dibuje el árbol de decisión resultante para estos datos (no considere PODA)

   **Respuesta**: Dado un conjunto de entrenamiento $S$, la **entropia** o impureza de los datos estará dada por:

$$Entropy(S) = -(p_{pos}) \log_2(p_{pos}) - (p_{neg}) \log_2(p_{neg}) \tag{11}$$

   Donde $p_{pos}$ es la proporción de casos positivos y $p_{neg}$, la proporción de casos negativos. Por otro lado, la **ganancia de información** o reducción de entropia que tenemos cuando particionamos el conjunto $S$ usando los valores de un atributo $A$, estará dado por:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{12}$$

   Donde $|S|$ es la cardinalidad de la clase $S$ y $|S_v|$ es la cardinalidad de S dado el valor de $v$. Considerando lo anterior, calculemos la entropia de SABROSO

$$E(Sabroso) = -5/10 \log_2 5/10 - 5/10 \log_2 5/10 = 1 \tag{13}$$

   Asimismo, la entropía de Sabroso dados cada uno de los atributo.

   **Para Temperatura:**

$$E(Sabroso, Temp = Caliente) = -0/1 \log_2 0/1 - 1/1 \log_2 1/1 = 0 \tag{14}$$
$$|S_{Caliente}| = 1 \tag{15}$$
$$E(Sabroso, Temp = Frio) = -3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971 \tag{16}$$
$$|S_{Frio}| = 5 \tag{17}$$
$$E(Sabroso, Temp = Ambiente) = -2/4 \log_2 2/4 - 2/4 \log_2 2/4 = 1 \tag{18}$$
$$|S_{Ambiente}| = 4 \tag{19}$$

**Para Gusto:**

$$E(Sabroso, Gusto = Salado) = -0/3 \log_2 0/3 - 3/3 \log_2 3/3 = 0 \quad (20)$$
$$|S_{Salado}| = 3 \quad (21)$$
$$E(Sabroso, Gusto = Dulce) = -2/4 \log_2 2/4 - 2/4 \log_2 2/4 = 1 \quad (22)$$
$$|S_{Dulce}| = 4 \quad (23)$$
$$E(Sabroso, Gusto = Acido) = -3/3 \log_2 3/3 - 0/3 \log_2 0/3 = 0 \quad (24)$$
$$|S_{Acido}| = 3 \quad (25)$$

**Para Tamaño:**

$$E(Sabroso, Tam = Pequeno) = -4/5 \log_2 4/5 - 1/5 \log_2 1/5 = 0.867 \quad (26)$$
$$|S_{Pequeno}| = 5 \quad (27)$$
$$E(Sabroso, Tam = Grande) = -1/5 \log_2 1/5 - 4/5 \log_2 4/5 = 0.867 \quad (28)$$
$$|S_{Grande}| = 5 \quad (29)$$

Usando los resultados anteriores, la Ganancia de Información por atributo sería:

$$Gain(Sabroso, Temp) = 1 - (\frac{1}{10} \times 0 + \frac{5}{10} \times 0.971 + \frac{4}{10} \times 1) = 0.115 \quad (30)$$
$$Gain(Sabroso, Gusto) = 1 - (\frac{3}{10} \times 0 + \frac{4}{10} \times 1 + \frac{3}{10} \times 0) = 0.6 \quad (31)$$
$$Gain(Sabroso, Tam) = 1 - (\frac{5}{10} \times 0.867 + \frac{5}{10} \times 0.867) = 0.133 \quad (32)$$

Por lo tanto, el atributo que da más información es **Gusto**. En ese sentido, el árbol quedaría como el de la Figura 1:

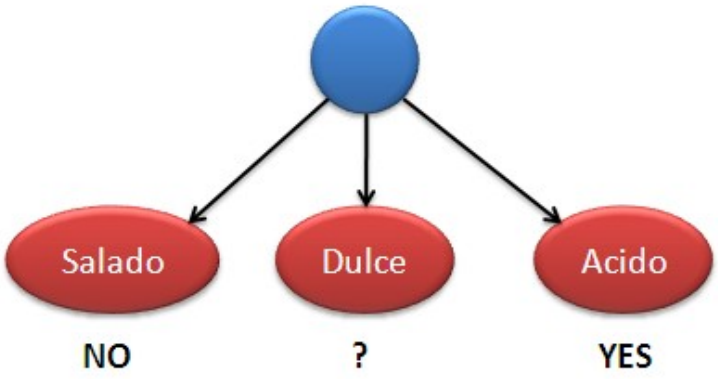


Figure 1: Arbol inicial

Donde las entropias de la categoría serían nulas cuando el atributo estuviera clasificado en "Salado" o "Acido". Basta revisar qué ocurre cuando la clasificación es dulce. En ese caso, el conjunto de entrenamiento sería como el de la tabla 2:

Table 2: Nuevo conjunto de entrenamiento

| Temperatura | Tamaño | Sabroso |
|---|---|---|
| Frio | Grande | NO |
| Frio | Grande | NO |
| Frio | Pequeño | YES |
| Frio | Pequeño | YES |

Es fácil notar que la ganancia de información con el atributo "Temperatura" es nulo, pues no discrimina en base a sus valores si la instancia es positiva o negativa (Para todos los valores de la clase, Temperatura es *Frio*). En tanto, el atributo "Tamaño" discrimina perfectamente el valor de la clase (Cuando es *Grande*, todos los casos son negativos y cuando es *Pequeño*, todos los casos son positivos)

Finalmente, el árbol resultante sería el siguiente como el de la Figura 2:


Figure 2: Arbol final

4. (2 punto) Usted ha sido contratado para realizar un estudio de web mining sobre el comportamiento de los usuarios en un sitio web. Para ello, cuenta con las sesiones de navegación ya identificadas y libres de errores. Por otro lado, usted no cuenta con información a priori sobre los usuarios del sitio. En ese sentido, responda:

   - (1 punto) ¿Qué técnicas de minería de datos utilizaría para realizar el estudio? Justifique su respuesta.

     **Respuesta:** Dado que no se cuenta con información a priori sobre los usuarios del sitio y su comportamiento de navegación, las técnicas de clustering resultan ser bastante útiles para la extracción de patrones. Por ejemplo: SOFM y K-means

   - (1 punto) ¿Cuáles serían los pro y los contra de cada una de las técnicas elegidas? ¿Se compensan entre si?

**Respuesta:** Por un lado, K-means tiene una alta performance, por tanto, es posible correr el algortimo múltiples veces con distintos parámetros. No obstante, K-means exige fijar apriori el número de clústers a obtener. Por otro lado, SOFM no exige fijar los parámetros a priori. En ese sentido, entrega una cantidad de clusters dada. Sin embargo, su performance es menor que K-means. Finalmente, se puede asegurar que ambos algoritmos son complementarios, pues una vez corroborados los patrones obtenidos con SOFM, estos pueden ser validados con los obtenidos mediante K-means prefijando el número de clusters a obtener.