

ENTREGA 7

Ejercicios y prácticas Métodos Bayesianos, aprendizaje de parámetros

Jhon Steeven Cabanilla Alvarado

1. Utilizando el conjunto de datos weather.nominal.practica que se proporciona, determinar la clasificación Naive Bayes de las siguientes instancias, utilizando la estimación de máxima verosimilitud (frecuencial) y sin utilizar ninguna herramienta de minería de datos:

- a. $X_1 = \langle \text{sunny, cool, normal, false} \rangle$
- b. $X_2 = \langle \text{overcast, hot, high, true} \rangle$

Comenzamos clasificando la instancia X_1 , por lo que tenemos que calcular:

$$v_{NB} = \arg \max p'(v_j) \prod p'(a_i / v_j) =$$

$$= \arg \max p'(v_j) p'(\text{outlook} = \text{sunny} / v_j) p'(\text{temperature} = \text{cool} / v_j) p'(\text{humidity} = \text{normal} / v_j) p'(\text{windy} = \text{false} / v_j)$$

A continuación, procedemos a estimar las probabilidades por las frecuencias en entrenamiento mediante la **estimación de máxima verosimilitud**.

$$p'(\text{play} = \text{yes}) = 9/14 = 0.64$$

$$p'(\text{play} = \text{no}) = 5/14 = 0.36$$

$$p'(\text{outlook} = \text{sunny} / \text{play} = \text{yes}) = 2/9 = 0.22$$

$$p'(\text{outlook} = \text{sunny} / \text{play} = \text{no}) = 3/5 = 0.6$$

$$p'(\text{temperature} = \text{cool} / \text{play} = \text{yes}) = 5/9 = 0.55$$

$$p'(\text{temperature} = \text{cool} / \text{play} = \text{no}) = 1/5 = 0.2$$

$$p'(\text{humidity} = \text{normal} / \text{play} = \text{yes}) = 6/9 = 0.66$$

$$p'(\text{humidity} = \text{normal} / \text{play} = \text{no}) = 1/5 = 0.2$$

$$p'(\text{windy} = \text{false} / \text{play} = \text{yes}) = 5/9 = 0.55$$

$$p'(\text{windy} = \text{false} / \text{play} = \text{no}) = 2/5 = 0.4$$

Clasificación:

- $p'(\text{play} = \text{yes} | \langle \text{sunny, cool, normal, false} \rangle)$:

$$p'(\text{yes}) p'(\text{sunny} / \text{yes}) p'(\text{cool} / \text{yes}) p'(\text{normal} / \text{yes}) p'(\text{false} / \text{yes}) = 0.0294$$

- $p'(\text{play} = \text{no} | \langle \text{sunny, cool, normal, false} \rangle)$:

$$p'(\text{no}) p'(\text{sunny} / \text{no}) p'(\text{cool} / \text{no}) p'(\text{normal} / \text{no}) p'(\text{false} / \text{no}) = 0.0034$$

- **Por tanto $v_{NB} = \text{yes}$**

Volvemos a realizar todo lo anterior pero ahora para la instancia X_2 .

$$v_{NB} = \arg \max p'(v_j) \prod p'(a_i / v_j) =$$

$$= \arg \max p'(v_j) p'(\text{outlook} = \text{overcast} / v_j) p'(\text{temperature} = \text{hot} / v_j) p'(\text{humidity} = \text{high} / v_j) p'(\text{windy} = \text{true} / v_j)$$

Estimación de máxima verosimilitud:

$$p'(\text{play} = \text{yes}) = 9/14 = 0.64$$

$$p'(\text{play} = \text{no}) = 5/14 = 0.36$$

$$p'(\text{outlook} = \text{overcast} / \text{play} = \text{yes}) = 2/9 = 0.22$$

$$p'(\text{outlook} = \text{overcast} / \text{play} = \text{no}) = 0/5 = 0$$

$$p'(\text{temperature} = \text{hot} / \text{play} = \text{yes}) = 0/9 = 0$$

$$p'(\text{temperature} = \text{hot} / \text{play} = \text{no}) = 2/5 = 0.4$$

$$p'(\text{humidity} = \text{high} / \text{play} = \text{yes}) = 3/9 = 0.33$$

$$p'(\text{humidity} = \text{high} / \text{play} = \text{no}) = 4/5 = 0.8$$

$$p'(\text{windy} = \text{true} / \text{play} = \text{yes}) = 4/9 = 0.4$$

$$p'(\text{windy} = \text{true} / \text{play} = \text{no}) = 3/5 = 0.6$$

Clasificación:

- $p'(\text{play} = \text{yes} \mid \langle \text{overcast}, \text{hot}, \text{high}, \text{true} \rangle)$:

$$p'(\text{yes}) p'(\text{overcast} / \text{yes}) p'(\text{hot} / \text{yes}) p'(\text{high} / \text{yes}) p'(\text{true} / \text{yes}) = 0$$

- $p'(\text{play} = \text{no} \mid \langle \text{overcast}, \text{hot}, \text{high}, \text{true} \rangle)$:

$$p'(\text{no}) p'(\text{overcast} / \text{no}) p'(\text{hot} / \text{no}) p'(\text{high} / \text{no}) p'(\text{true} / \text{no}) = 0$$

- La estimación de la probabilidad para los 2 valores de la clase no da 0, ya que no tenemos ningún ejemplo para el atributo *outlook* con la clase *no* ni para el atributo *temperature* con la clase *yes*.

En el caso de realizar una **corrección de muestreo** y dejar a un lado la probabilidad nula de esos atributos con el valor correspondiente de clase, podríamos clasificar la instancia como **no**, debido a que tendría una mayor estimación de la probabilidad ($\text{yes} = 0.0205$, $\text{no} = 0.0685$).

2. Entrenar, con Weka, un clasificador Naive Bayes para el conjunto de datos weather.nominal.practica

- a. Estimar la tasa de error cometida por el clasificador utilizando validación cruzada de 10 particiones.
- b. Examinar la salida proporcionada por el Explorer y determinar cómo está estimando esta implementación de Naive Bayes los parámetros del clasificador.
- c. ¿Coincide con la clasificación calculada en el ejercicio anterior?

a. La tasa de error que comete el clasificador Naive Bayes utilizando validación cruzada de 10 particiones es de 0.428571 (42.8571%).

b. El algoritmo *Naive Bayes Simple* implementado en WEKA funciona de la misma manera que la implementación manual realizada en este documento. Primero se calcula la probabilidad a priori de cada uno de los valores que puede tomar la clase de destino sobre el conjunto de datos de entrenamiento. Seguidamente calcula las probabilidades condicionadas de cada uno de los valores de los atributos del conjunto de datos respecto de las distintas clases de destino.

Para clasificar las instancias supone independencia entre atributos y calcula la probabilidad conjunto de todos los atributos de la instancia condicionada a las distintas clases para después clasificar dicha instancia en la clase que maximice el valor de probabilidad de la misma.

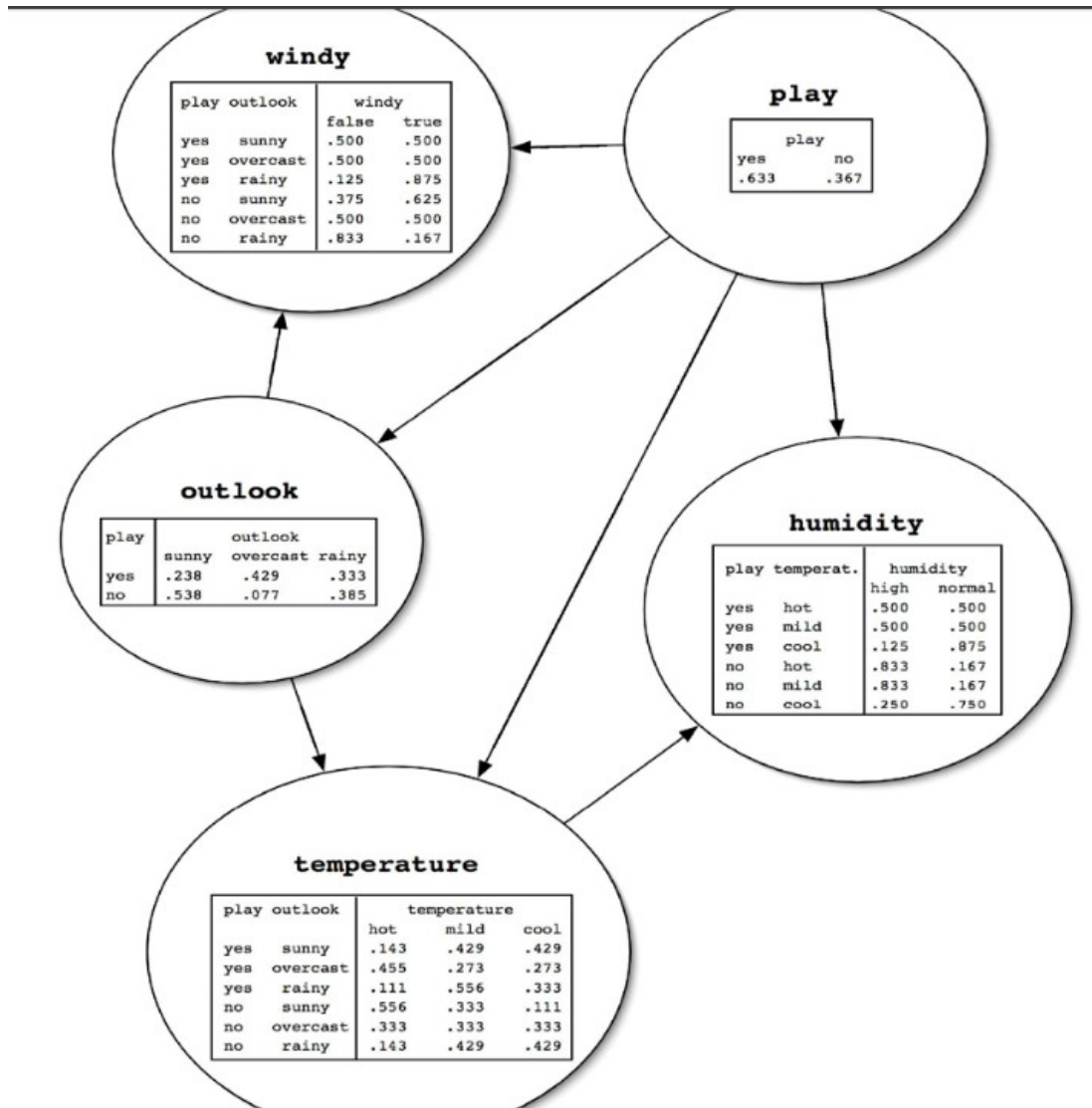
c. Para ello, creamos un clasificador Naive Bayes y entrenamos con las instancias del archivo weather.nominal.practica. Obtenemos los siguientes resultados para las instancias del archivo weather.nominal.prueba que utilizamos para realizar el test:

< sunny,cool,normal,FALSE,yes >
< overcast,hot,high,TRUE,no >

=== Summary ===

Correctly Classified Instances	2	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.2027		
Root mean squared error	0.2046		
Relative absolute error	40.5438	%	
Root relative squared error	39.7076	%	
Total Number of Instances	2		

3. Considerar el conjunto de datos weather.nominal.practica proporcionado y la siguiente red TAN:



Obtener los parámetros de la red para el nodo Windy utilizando la estimación de máxima verosimilitud y la corrección de Laplace con $mm = 0,5$ (caso particular de la estimación Bayesiana). Hacer el ejercicio sin utilizar software de aprendizaje.

Procedemos a realizar la Estimación Bayesiana la cual permite incluir conocimiento a priori de los parámetros. Además evita obtener estimas nulas:

$$\hat{\theta}_{x_i^j|u} = \frac{M[u, x_i^j] + mp}{M[u] + m}$$

$p = 0.5$ porque Windy unicamente puede tomar 2 valores: *True* y *False*.
 $m = 0.5$ tal y como se indica en el enunciado

		(sunny,yes)	(overcast,yes)	(rainy,yes)	(sunny,no)	(overcast,no)	(rainy,no)
P(W O,P)	TRUE	0.5	0.9	0.2272	0.3571	0.5	0.9
	FALSE	0.5	0.1	0.7727	0.6428	0.5	0.1

Cálculo:

1) (Sunny, yes) → Total de instancias: 2 (M[u])

De esas 2 instancias, Windy toma el valor TRUE en 1 y FALSE en 1.

Para TRUE → $(1 + 0.5 * 0.5) / (2 + 0.5) = 0.5$

Para FALSE → $(1 + 0.5 * 0.5) / (2 + 0.5) = 0.5$

2) (Overcast, yes) → Total de instancias: 2 (M[u])

De esas 2 instancias, Windy toma el valor TRUE en 2 y FALSE en 0.

Para TRUE → $(2 + 0.5 * 0.5) / (2 + 0.5) = 0.9$

Para FALSE → $(0 + 0.5 * 0.5) / (2 + 0.5) = 0.1$

3) (Rainy, yes) → Total de instancias: 5 (M[u])

De esas 5 instancias, Windy toma el valor TRUE en 1 y FALSE en 4.

Para TRUE → $(1 + 0.5 * 0.5) / (5 + 0.5) = 0.2272$

Para FALSE → $(4 + 0.5 * 0.5) / (5 + 0.5) = 0.7727$

4) (Sunny, no) → Total de instancias: 3 (M[u])

De esas 3 instancias, Windy toma el valor TRUE en 1 y FALSE en 2.

Para TRUE → $(1 + 0.5 * 0.5) / (3 + 0.5) = 0.3571$

Para FALSE → $(2 + 0.5 * 0.5) / (3 + 0.5) = 0.6428$

5) (Overcast, no) → Total de instancias: 0 (M[u])

Para TRUE → $(0 + 0.5 * 0.5) / (0 + 0.5) = 0.5$

Para FALSE → $(0 + 0.5 * 0.5) / (0 + 0.5) = 0.5$

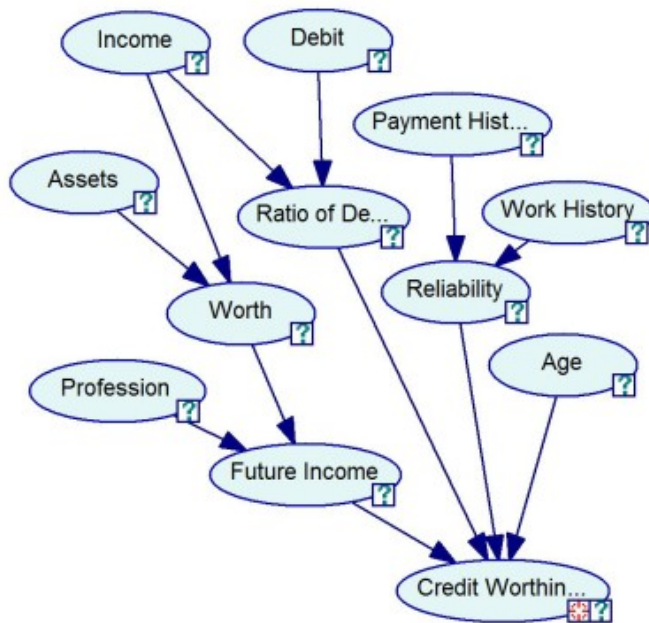
6) (Rainy, no) → Total de instancias: 2 (M[u])

De esas 2 instancias, Windy toma el valor TRUE en 2 y FALSE en 0.

Para TRUE → $(2 + 0.5 * 0.5) / (2 + 0.5) = 0.9$

Para FALSE → $(0 + 0.5 * 0.5) / (2 + 0.5) = 0.1$

4. Considerar la estructura la red Bayesiana Credit, en la que todas las variables son discretas:



A simple network for assessing credit worthiness of an individual, developed by Gerardina Hernandez as a class homework at the University of Pittsburgh. Note that all parentless nodes are described by uniform distributions. This is a weakness of the model, although it is offset by the fact that all these nodes will usually be observed and the network will compute the probability distribution over credit worthiness correctly. Another element of this model is that only the node CreditWorthiness is of interest to the user and is designated as a target.

Ejemplo Credit, proporcionado por la herramienta GeNIe2.0, cuya versión de educación ya no está disponible.

Se proporcionan tres conjuntos de datos, Datos_Credit_100.csv, Test_Credit_1000.csv y Datos_Credit_10000.csv, que contienen 100, 1000, y 10000 instancias, respectivamente, todas ellas muestreadas de forma aleatoria e independiente a partir de la red. Por tanto, todos los ejemplos son muestras independientes de la distribución de probabilidad modelada por la red, P credit.

Se desea examinar el comportamiento de las redes Naive Bayes y TAN sobre los conjunto de datos Datos_Credit_100.csv y Datos_Credit_10000.csv.

Utilizar Weka para entrenar los dos tipos de redes con los dos conjuntos de datos. En todos los casos, estimar los parámetros con la estimación de máxima verosimilitud con corrección de Laplace ($\alpha = 0,5$) y estimar la tasa de error de cada modelo mediante validación cruzada de diez particiones.

Una vez generado cada modelo y estimada su tasa de error mediante validación cruzada sobre los conjuntos proporcionados, guardar el modelo y estimar su tasa de error sobre el conjunto Test_credit_1000, que contiene otras mil instancias muestreadas aleatoriamente de la red.

Comparar el comportamiento de las redes en base a la estimación de la tasa de error sobre Test_Credit_1000. Comparar esta estimación con la que proporciona la estimación mediante validación cruzada.

Clasificación mediante Redes Bayesianas					
Datos	Naive Bayes		Tasa de Error		TAN
	100	10000	100	10000	
Entrenamiento	36%	29.69%	35%	28.27%	
Test_Credit_1000	30%	27.5%	33.3%	25%	

A continuación se describen los resultados a nivel de tasas de error obtenidas en los experimentos. Los resultados obtenidos en la fase de entrenamiento se corresponden con una metodología de *Validación Cruzada* de 10 particiones, mientras que los obtenidos en la fase de *Test_Credit_1000* son relativos a las clasificación de las instancias sobre el conjunto Test_credit_1000, que contiene otras mil instancias muestreadas aleatoriamente de la red.

La tendencia habitual de los 2 algoritmos de clasificación es de reducir su tasa de error conforme el tamaño del conjunto de datos de entrenamiento aumenta.

A nivel de resultados, los 2 algoritmos ofrecen tasas de error similares. Debido a las pequeñas variaciones a nivel de tasas de error entre ellos, no se puede asumir con exactitud que ninguno ofrezca mejores resultados que el otro sobre los conjuntos de datos utilizados.