Minería de Datos

Práctica 3. Primeros pasos con los clasificadores

Ángel Ríos San Nicolás

19 de marzo de 2021

1. Abre con Weka el fichero "contact-lense.arff"

2. Ejecuta el algoritmo ID3 con "cross-validation (10 folds)". Describe el árbol de decisión obtenido y responde:

   El árbol de decisión que devuelve ID3 es el siguiente:

```
tear-prod-rate = reduced:  none
tear-prod-rate = normal
| astigmatism = no
| | age = young:  soft
| | age = pre-presbyopic:  soft
| | age = presbyopic
| | | spectacle-prescrip = myope:  none
| | | spectacle-prescrip = hypermetrope:  soft
| astigmatism = yes
| | spectacle-prescrip = myope:  hard
| | spectacle-prescrip = hypermetrope
| | | age = young:  hard
| | | age = pre-presbyopic:  none
| | | age = presbyopic:  none
```

   a. ¿Qué atributo tiene una mayor ganancia de información? Calcula la ganancia de información de los diferentes atributos y comprueba que efectivamente el que tiene una mayor ganancia es el escogido por Id3.

      Calculamos la ganancia de información de cada atributo en R con la librería RWeka.

```r
library("RWeka")  # Importamos la librería RWeka.
datos <- read.csv("contact-lense.csv")  # Cargamos los datos.
# Por defecto la clase es la primera columna.
datos <- datos[, c(5,1,2,3,4)]  # Permutamos para fijar la clase.
InfoGainAttributeEval(datos)  # Ganancia de información
```

```
        age  spectacle.prescript  astigmatism  tear.prod.rate
 0.03939650           0.03951084   0.37700523      0.54879494
```

      Efectivamente, el atributo *tear-prod-rate* es el que tiene mayor ganancia de información, que es el escogido por Id3.

   b. ¿Cuál es el segundo atributo seleccionado por el algoritmo? Vuelve a hacer los cálculos de ganancia de los atributos restantes y comprueba que efectivamente Id3 ha escogido el esperado.

      Partimos del código en R anterior y volvemos a hacer los cálculos sin considerar el atributo *tear-prod-rate*.

```
datos <- datos[1:4]  # Eliminamos la columna tear.prod.rate.
datos <- unique(datos)  # Eliminamos las filas repetidas.
InfoGainAttributeEval(datos)  # Ganancia de información.
```

```
        age  spectacle.prescrip  astigmatism
 0.02134362          0.04122722   0.42693510
```

Efectivamente, el atributo con mayor ganacia de información es *astigmatism*, el segundo escogido por Id3.

c. Describe el rendimiento del clasificador en términos de Accuracy, TP rate de cada valor de clase, Precision de cada clase y Recall de cada clase por separado... ¿diría que el resultado del clasificador es bueno? ¿por qué?

La matriz de confusión es

soft	*hard*	*none*
4	0	1
0	1	3
1	2	12

con Accuracy del 70.8333%.

Las medidas TPrate, Precision y Recall que devuelve Weka para cada clase son:

Clase	TPRate	Precision	Recall
soft	0.800	0.800	0.800
hard	0.250	0.333	0.250
none	0.800	0.750	0.800

El resultado del clasificador no es bueno porque aunque clasifica relativamente bien las instancias de las clases *soft* y *none* no clasifica bien las de la clase *hard* ya que los TPrate, Precision y Recall son bajas, lo que queda reflejado en el bajo Accuracy. De hecho, en la matriz de confusión se ve que de tres instancias de la clase *hard*, solo clasifica bien una.

d. Vuelve a ejecutar el algoritmo ID3 con la opción "Percentage Split 66%". ¿Es el rendimiento el mismo que en el punto c.?

El rendimiento es mucho peor con la opción "Percentage Split 66%". La matriz de confusión es

soft	*hard*	*none*
0	3	1
0	0	1
0	0	3

con Accuracy del 37.5%.

Las medidas TPrate, Precision y Recall que devuelve Weka para cada clase son:

Clase	TPrate	Precision	Recall
soft	0.000	?	0.000
hard	0.000	0.000	0.000
none	1.000	0.600	1.000

Observamos que el método no clasifica ninguna instancia en la clase *soft*, por lo que en particular las medidas TPrate y Recall son 0 para esa clase y no se puede calcular la precisión por la ausencia de instancias. La única instancia escogida de la clase *hard* se clasifica como *none* con lo que también clasifca mal esa clase. La pésima clasifcación se aprecia claramente en la matriz de confusión y es consecuencia de usar un método que divide los datos cuando tenemos muy poca cantidad.

e. Vuelve a ejecutar el algoritmo ID3 con la opción "Use Training Set". ¿Es el rendimiento el mismo que en el punto c. o d.? ¿Qué resultados consideras más fiables y por qué?

Con la opción "Use Training Set" el método clasifica bien todas las instancias en su clase, en particular la medida de Accuracy es 100%, TPrate, Precision y Recall de cada clase son 1.000 y la matriz de confusión es diagonal. Esto es consecuencia de usar como conjunto de entrenamiento todos los datos y validar sobre ellos mismos. Como la cantidad de datos es muy pequeña, el sobreajuste es total.

El clasificador más fiable en el sentido de ajustarse a la realidad y generalizar mejor es el del apartado c. porque tiene Accuracy mayor que el del apartado d. pero sin sobreajustar como con el clasificador anterior. Además, el del apartado d. no clasifica ninguna instancia como *soft* y falla todas las instancias *hard*.

f. Ejecuta otra vez el algoritmo ID3 con "cross-validation" pero esta vez usando "5 folds". ¿Cambian los resultados? ¿Por qué sucede?

Al ejecutar el algoritmo con "cross-validation", pero usando "5 folds" cambian ligeramente los resultados. La matriz de confusión es

soft	*hard*	*none*
4	0	1
0	1	2
1	1	12

con Accuracy del 70.8333%.

Las medidas TPrate, Precision y Recall que devuelve Weka para cada clase son:

Clase	TPrate	Precision	Recall
soft	0.800	0.800	0.800
hard	0.333	0.500	0.333
none	0.857	0.800	0.857

La medida de Accuracy no varía, sin embargo en 5 rondas quedan dos instancias sin clasificar lo que hace variar ligeramente el resto de medidas. La primera clase *soft* se clasifica de la misma manera, con lo que tienen las mismas medidas. El hecho de procesar menos instancias provoca que, en este caso, se produzcan dos errores menos en la clasificación, con lo que las medidas de evaluación aumentan.

3. Ejecuta el algoritmo J48. Responde a las siguientes cuestiones:

a. ¿Es el árbol generado el mismo que con el algoritmo ID3? ¿En qué se diferencian?

El árbol que devuelve J48 es el siguiente.

```
tear-prod-rate = reduced:  none (12.0)
tear-prod-rate = normal
| astigmatism = no:  soft (6.0/1.0)
| astigmatism = yes
| | spectacle-prescrip = myope:  hard (3.0)
| | spectacle-prescrip = hypermetrope:  none (3.0/1.0)
```

El árbol es distinto al generado con ID3. Observamos que es un árbol podado más pequeño que el árbol de ID3, que no está podado. En el árbol de J48 no hay ninguna ramificación por el atributo *age* mientras que ID3 utiliza, por definición, todos los atributos.

b. ¿La matriz de confusión y/o las diferentes medidas (Accuracy, etc) son iguales que las obtenidas con el algoritmo ID3? ¿Cuál es la diferencia?

La matriz de confusión de J48 es

soft	*hard*	*none*
5	0	0
0	3	1
1	2	12

y las diferentes medidas de evaluación son un Accuracy del 83.3333% y

Clase	TPrate	Precision	Recall
soft	1.000	0.833	1.000
hard	0.750	0.600	0.750
none	0.800	0.923	0.800

Observamos que la medida de Accuracy es mayor en J48 frente al 70.8333% de ID3. También mejoran el resto de medidas TPRate, Precision y Recall. Esto se refleja en la matriz de confusión, en la que vemos que el algoritmo clasifica igual respecto a *none*, pero clasfica mejor *hard* y *soft*, del que acierta todas las instancias.

4. Ejecuta el algoritmo RandomTree. Responde a las siguientes cuestiones:

   a. ¿Es el árbol generado el mismo que con los algoritmos ID3 o J48? ¿En qué se diferencian?

   El árbol que devuelve RandomTreee es

```
astigmatism = no
| tear-prod-rate = reduced :  none (6/0)
| tear-prod-rate = normal
| | age = young :  soft (2/0)
| | age = pre-presbyopic :  soft (2/0)
| | age = presbyopic
| | | spectacle-prescrip = myope :  none (1/0)
| | | spectacle-prescrip = hypermetrope :  soft (1/0)
astigmatism = yes
| spectacle-prescrip = myope
| | tear-prod-rate = reduced :  none (3/0)
| | tear-prod-rate = normal :  hard (3/0)
| spectacle-prescrip = hypermetrope
| | age = young
| | | tear-prod-rate = reduced :  none (1/0)
| | | tear-prod-rate = normal :  hard (1/0)
| | age = pre-presbyopic :  none (2/0)
| | age = presbyopic :  none (2/0)
```

   Es un árbol más grande que los generados por ID3 y por J48. Discrimina primero por *astigmatism* en lugar de por *tear-prod-rate* como hacen ID3 y J48.

   b. ¿La matriz de confusión y/o las diferentes medidas (Accuracy, etc) son iguales que las obtenidas con los algoritmos ID3 o J48? ¿Cuál es la diferencia?

   La matriz de confusión de RandomTree es

soft	*hard*	*none*
4	1	0
0	1	3
1	2	12

   La medida de Accuracy es 70.8333% y el resto de medidas de evaluación son

Clase	TPrate	Precision	Recall
soft	0.800	0.800	0.800
hard	0.250	0.150	0.250
none	0.800	0.800	0.800

   Observamos que la medida de Accuracy y el resto de medidas coinciden exactamente con las de ID3 salvo que la medida Precision es menor para los atributos *hard* y *none*. En particular, todas las medidas de evaluación son menores que las de J48.

5. ¿A la vista de los resultados anteriores, con qué clasificador te quedarías para este conjunto de datos? Razona la respuesta.

Me quedaría con el clasificador J48 porque tiene un Accuracy superior al resto sin llegar a sobreajustar los datos. Además, el resto de medidas de evaluación del clasificador son todas mejores en cada atributo y el árbol de decisión es más pequeño, lo que facilita su visualización, interpretación y uso.