Árboles de decisión con Weka

Sistemas Inteligentes 2022-2023

El objetivo de esta práctica es tomar contacto con el entorno de trabajo Weka. Weka es un conjunto de librerías Java para la extracción de conocimientos en bases de datos. Weka puede encontrarse en este enlace y está disponible para varias plataformas (*Windows, Linux y Mac*). La última versión estable (3.8) se puede descargar en este enlace (hay varias opciones dependiendo de su plataforma y de si dispone o no de la versión adecuada de Java instalada previamente en sus sistema).

En primer lugar, nos familiarizaremos con el tipo de fichero en el que vamos a suministrar los datos. Para ello, abriremos con un editor de texto el fichero ***clima.arff*** (con extensión ***arff***, *Attribute-Relation File Format*). Como se puede comprobar tiene las definiciones de ***relation*** (nombre del conjunto de datos), ***attribute*** (lista de atributos, cada uno con su lista de valores posibles, o bien con su tipo) y ***data*** (lista de ejemplos, conteniendo cada uno los datos para cada atributo del ejemplo, separados por comas). Un fichero de extensión ***arff*** se puede crear y/o editar con cualquier editor de textos. Si abre el fichero ***clima-num.arff*** verá que los atributos numéricos no tienen asignada la lista de valores, sino el tipo ***numeric***. Para más información se puede consultar el manual de usuario.

Pasemos ahora a Weka. Al ejecutar el programa, aparecen 5 entornos de trabajo: ***Explorer***, ***Experimenter***, ***KnowledgeFlow***, ***Workbench*** y ***Simple CLI***. En esta práctica trabajaremos en el entorno ***Explorer***. En este entorno abra el fichero ***clima.arff*** con *Open File*. La pestaña de preprocesado (pestaña inicial al entrar en este entorno Explorer) nos da mucha información sobre los datos. Por ejemplo el atributo **cielo** tiene tres valores asociados ***soleado***, ***nublado*** y ***lluvioso***, con 5, 4 y 5 ejemplos respectivamente. La distribución de colores nos permite ver qué proporción de estos ejemplos son ***positivos*** y ***negativos***. Podemos ir pinchando sobre el nombre de cada uno de los atributos para ver su distribución. Weka trae su propio editor interno, de manera que podemos modificar el fichero desde dentro de Weka. Pulse el botón *Edit...* (en la ventana, arriba a la derecha) y modifica, en el primer ejemplo, el valor del atributo ***cielo*** de ***soleado*** a ***lluvioso***. Pulse el botón *OK*, y verá cómo cambia el histograma. Recuerde volver a poner el valor ***soleado*** de nuevo antes de seguir, o de modo equivalente emplear la opción *Undo* (deshacer).

Weka nos facilita muchos filtros para preprocesar la información, que en general se pueden elegir a partir del botón *Choose* en el apartado *Filter*. El más simple es el filtro que permite eliminar algunos atributos, pero para ello no hace falta entrar en el apartado de filtros (basta con marcar en la ventana principal los atributos que queramos eliminar y pulsar *Remove*). Una vez

eliminado, podemos salvar (*Save*) el nuevo conjunto de entrenamiento o deshacer (*Undo*) los cambios. Los botones están en la misma línea que *Open file*.

Weka también dispone de opciones adicionales para la visualización de los datos. Esto se ilustra mejor con un fichero numérico. Abra el fichero ***iris.arff***, primero mediante algún editor de texto para explorar su contenido y luego en Weka. Este fichero contiene 150 ejemplos de plantas. De cada planta se han considerado 4 atributos: la longitud y anchura del pétalo y del sépalo. La clasificación consta de tres valores: ***Iris-setosa***, ***Iris-versicolor*** e ***Iris-virgínica***. Si pulsa en la pestaña *Visualize* (junto a *Select Attributes*) aparece la representación por pares de valores. Por ejemplo, pulsando en el par *sepalwidth* - *petalwidth* podemos ver con detalle la distribución de valores (representados por colores) de los 150 ejemplos.

Pasamos ahora a ver cómo podemos usar Weka para encontrar un árbol de decisión a partir de un conjunto de ejemplos. Para ello, volvemos a cargar el fichero ***clima.arff***. Una vez cargado el fichero, seleccionamos la pestaña *Classify*. Con el botón *Choose* podemos elegir el clasificador entre muchas opciones. Pulsamos en dicho botón, y seleccionamos el J48 del menú *trees*. J48 es una implementación del algoritmo ID3 con pequeñas modificaciones. Entre las *Test options*, seleccionamos *Use training set* y observamos que aparece como atributo de clasificación *jugar_tenis*, que es el último que aparece en la lista. Con el botón *Start* generamos el árbol. Aparece en modo texto, con información complementaria. En particular podemos destacar que ha encontrado un árbol con 8 nodos (size), de los cuales 5 son hojas (leaves) y que ha clasificado correctamente los 14 ejemplos de la base de datos (el 100 %). Es importante también la *Matriz de confusión* que aparece al final. En ella nos dice que de los 9 ejemplos con clasificación ***si*** el árbol clasifica 9 como ***si*** y 0 como ***no***. Análogamente, de los 5 ejemplos con clasificación ***no***, el árbol clasifica 5 como no y 0 como ***si***. Con el botón secundario sobre la operación (en el panel *Result list* que aparece a la izquierda, con formato *hora* - *técnica empleada*), podemos hacer uso de la opción *Visualize tree* para ver el árbol como un grafo.

Para ver cómo se tratan los valores numéricos, veamos el ejemplo contenido en el fichero ***num.arff***.

```
@relation clima2

@attribute temperatura real
@attribute outlook {soleado, nublado}
@attribute clasificacion {SI, NO}

@data
32,soleado,SI
33,nublado,SI
31,soleado,SI
20,nublado,NO
22,nublado,NO
```

Como puede observar, el atributo ***temperatura*** se ha declarado como real.

Cargue el fichero en Weka en la pestaña de preprocesado. A continuación, vaya a la pestaña de clasificación y elija de nuevo el algoritmo ***J48*** (en caso de que no lo tenga ya seleccionado). Mantenga como atributo objetivo por defecto ***clasificación***, y lance el algoritmo (*Start*). ¿Qué resultado observa? ¿Qué nodo se ha generado en el árbol obtenido? ¿Qué diferencia hay respecto al manejo de atributos continuos que vimos en la clase anterior de teoría?

Volvamos a cargar ahora el fichero ***iris.arff***. Vamos a aplicar J48 para encontrar un árbol de clasificación. Para ello, seleccionamos *Use training set*. Además, pulsamos sobre **J48** -C 0.25 -M 2 (sobre el campo junto al botón *Choose*) con el botón izquierdo (o bien con el derecho, y posteriormente le decimos que nos muestre las propiedades - *Show properties*). Indicamos que no pode el árbol obtenido (***Unpruned* = *True***) y que admita hojas con un único ejemplo (***minNumObj* = *1***). Al pulsar OK y volver a la ventana principal de clasificación, observará que la descripción ha cambiado a **J48** -U -M 1. Antes de lanzar el algoritmo, ¿cree que esta decisión puede ocasionar algún problema en el árbol obtenido? ¿Qué ventajas e inconvenientes puede tener respecto al lanzamiento anterior con las opciones por defecto?

Pulse Start, de modo que se genere el árbol. ¿Cuántos nodos contiene? ¿Qué porcentaje de ejemplos del conjunto de entrenamiento clasifica correctamente?

Observe el árbol. ¿Cuántas ramas se han desarrollado para albergar 1 único ejemplo? ¿Y conteniendo dos ejemplos? Reflexione de nuevo: ¿le parece que puede ser un buen árbol para usarlo como herramienta de clasificación o piensa que puede producir algún efecto no deseado? Justifique su respuesta, y en su caso dé alguna posible solución.

Cargue ahora el fichero ***clima-num.arff***. ¿Cuántos ejemplos tiene el conjunto de entrenamiento? Genere el árbol con las opciones de J48 por defecto. ¿Cuántos nodos tiene el árbol? ¿Cuántas hojas? ¿Qué medida de rendimiento obtiene con respecto al conjunto de entrenamiento? Pruebe también a generarlo con la opción de no podar y admitir hojas con un único ejemplo. Realice para cada uno de ellos validación cruzada (***cross-validation***), con un valor adecuado, y compare el rendimiento obtenido por ambas opciones. ¿Qué opción proporciona mejor rendimiento realizando *cross-validation*? ¿Significa eso que es la mejor opción de cara a usarlo como clasificador?

Weka nos permite también usar los árboles generados para clasificar nuevas instancias. Observe el fichero ***test.arff***:

```
@relation deporte-test

@attribute cielo {soleado, nublado, lluvioso}
@attribute temperatura numeric
@attribute humedad numeric
@attribute hace_viento {si, no}
@attribute hacer_deporte {si, no}

@data
soleado,40,25,no,?
soleado,12,22,no,?
nublado,23,54,si,?
lluvioso,20,82,no,?
```

Como verá, aparecen una serie de signos de cierre de interrogación, **?**, al final de cada línea representando un ejemplo. Como habrá podido deducir, se trata de ejemplos de los que se desconoce su salida, son instancias sin clasificar, se trata de un *conjunto de prueba*.

Volvamos a generar el árbol con ***clima-num.arff***. Para conocer la clasificación que ofrece el árbol a los elementos de prueba, en el entorno *Classify* debemos proporcionar el nuevo fichero ***test.arff*** como *Supplied test set*. Además, en *More options* debemos marcar *Output predictions*, por ejemplo en texto plano. ¿Qué respuesta nos da el árbol para cada una de las instancias?

Otra cosa que podemos hacer es ver si podemos mejorar el árbol generado. Para ello, vamos a emplear la opción ***Visualize*** sobre los datos. En dicha pestaña, podemos ver gráficamente la distribución de los elementos del conjunto de entrenamiento representados por pares de atributos. Esta visualización puede ser útil. Por ejemplo, en el cuadro ***Humedad*** vs. ***Temperatura***, vemos que hay una separación aparentemente lineal de los datos (es decir, podemos trazar una línea

recta separando los ejemplos positivos de los negativos). Podemos usar esa separación lineal para tener una representación más compacta.

Para ello, vamos a introducir un atributo nuevo al árbol. En *Preprocess*, elegimos con *Choose* un nuevo filtro. En este caso, dentro de los Atributos en la carpeta de *Unsupervised - Attribute*, elegimos *AddExpression*. En la definición del filtro, podemos escribir expresiones que hagan uso de los atributos ya existentes en base a su posición; así, nos referiremos a ***cielo*** como ***a1*** o a hace ***viento*** como ***a4***. Sabiendo que el operador / realiza la división, proporcione una expresión para el cociente entre la temperatura y la humedad. Pulsemos *OK*, y volvamos a la ventana principal. A continuación, pulsamos el botón *Apply* (en el filtro, a la derecha del todo). ¿Qué ha ocurrido con los atributos? Para ver el resultado sobre nuestro conjunto de datos, recordemos que podemos visualizar el mismo con el botón *Edit*.Además del cambio en los atributos, ¿han sufrido algún cambio los colores que aparecen en los histogramas de la parte inferior derecha de la pantalla? Corrija dicha circunstancia para recuperar los colores originales.

Generemos ahora el árbol. ¿De cuántos nodos consta? ¿De cuántas hojas? ¿Qué medida de rendimiento obtiene empleando validación cruzada?

Finalmente, se proporciona un conjunto de bases de datos pública ***arrythmia*** en formato ***arff***. Contiene un conjunto de 452 ejemplos y 279 atributos. El objetivo es aprender el tipo de arritmia a partir de los datos de electrocardiogramas (ver fichero). Esta base de datos se usó en el artículo de investigación http://www.cs.bilkent.edu.tr/tech-reports/1998/BU-CEIS-9802.pdf Prueba a generar el árbol de decisión.

Ejercicio: En este ejercicio se espera que el alumno investigue nuevas capacidades de Weka no vistas en clase, de modo que pueda desenvolverse en un entorno cambiante como es la ciencia de datos. Para esto, se proporciona un conjunto de datos (*arrythmia*) más parecido a los que nos encontraremos en problemas reales y que, por tanto, ofrecen más posibilidades.
A continuación se enumeran las tareas típicas de un proyecto de ciencia de datos, en el que el objetivo es obtener un modelo con la mayor capacidad predictiva posible:

- Exploración/visualización
- Imputación de valores perdidos (esto será necesario para algunos conjuntos de datos que contienen valores faltantes, denotados con el carácter **?** en Weka)
- Normalización, estandarización, discretización, binarización de categóricas (codificación OneHot) etc.
- Creación de nuevos atributos como se vio en la práctica
- Selección de atributos
- Selección de los mejores parámetros para un algoritmo
- Selección del mejor algoritmo

Para comparar modelos, debemos usar una metodología de evaluación que proporcione medidas de rendimiento lo más honestas posible.