Práctica 6: Minería de datos mediante la herramienta Weka

Sistemas de Información

Grado de Informática
Dpto. de Informática e Ingeniería de Sistemas,
Universidad de Zaragoza
Escuela de Ingeniería y Arquitectura

9 de enero de 2020

1. Objetivos

Hoy en día numerosos sistemas de información emplean técnicas de Data Mining y Machine Learning para obtener información y conocimiento de grandes volúmenes de datos. En esta práctica se presenta la herramienta Weka, la cual permite realizar diferentes análisis de datos almacenados en bases de datos relacionales (JDBC), ficheros CSV y ficheros ARFF. En mayor detalle, los objetivos de esta práctica son:

- Descargar e instalar la herramienta Weka.
- Familiarizarse con la herramienta Weka probando los ejemplos proporcionados.
- Crear un árbol de decisión que nos permita realizar predicciones sobre la calificación final de un determinado estudiante (opcional)

2. Contenidos

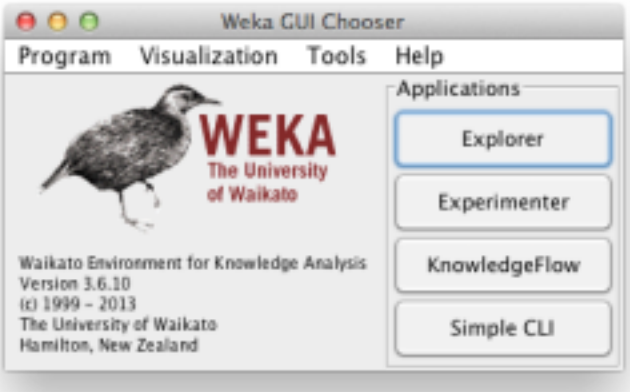
La práctica consta de dos bloques. El primer bloque presenta ejemplos guiados y requiere la realización de una serie de pruebas con los datos proporcionados. El segundo bloque (opcional) se centra en el análisis de datos de rendimiento del alumnado de una asignatura de la titulación Grado en Ingeniería Eléctrica. En primer lugar se requiere que se adapten los datos proporcionados para poder llevar a cabo un análisis sobre ellos. A continuación, se requiere que se creen una serie de modelos predictivos a partir de los datos proporcionados.

2.1 Instalación de la herramienta Weka

En primer lugar, es necesario descargar la herramienta. Para ello se puede emplear la siguiente dirección http://www.cs.waikato.ac.nz/ml/weka/index.html. Una vez descargada la herramienta para ejecutarla basta lanzar la siguiente orden:

```
java -jar weka.jar
```

De este modo se accede a la pantalla principal de la herramienta, la cual define cuatro entornos de trabajo: Simple CLI, Explorer, Experimenter y KnowledgeFlow. A lo largo de esta práctica emplearemos el entorno visual Explorer.



El entorno Explorer agrupa seis módulos: pre-procesamiento, clasificación, clusterización, asociación, selección de atributos y visualización.

2.2 Utilización de la herramienta Weka: Primeros pasos

En esta primera parte, vamos a analizar un fichero en el que se han registrado datos de distintas características físicas de 150 flores de la especie Iris, indicando a qué subespecie pertenece cada una (setosa, versicolor, virginica). El objetivo de la práctica es analizar el fichero, y tratar de configurar un clasificador que, a partir de los datos recogidos en el fichero, y conociendo las características de una nueva flor, nos diga a qué subespecie pertenece esta nueva flor. Es decir, queremos que el programa, a partir de un conjunto de ejemplos, sea capaz de inferir un algoritmo de clasificación de flores de la especie Iris.

1. Click en el botón “Explorer” para lanzar el explorador de datos de Weka. Esta opción te permite cargar los datasets y ejecutar algoritmos de clasificación. También permite otras opciones como filtrado de datos, clustering, extracción de reglas de asociación y visualización.

2. Click en el botón “Open file” para abrir el dataset y doble clic en el directorio “data” de Weka. Weka proporciona un conjunto de datasets pequeños para realizar pruebas y practicar. Uno de ellos es el dataset iris.arff, que es con el que vamos a trabajar. Seleccionar el fichero iris.arff para cargar el dataset.

   Este dataset contiene datos de la flor Iris. Contiene 150 instancias (filas) con 4 atributos cada una (columnas) y un atributo clase que indica el tipo (especie) de flor de Iris (setosa, versicolor y virgínica).

3. Seleccionar y ejecutar un algoritmo de minería de datos. Una vez cargado el dataset es el momento de elegir el algoritmo de minería de datos para modelar el problema y poder hacer análisis y/o predicciones.

   Haz click en la pestaña “Classify”. Esta pestaña contiene el área de ejecución de los algoritmos contra el dataset cargado en Weka. En ella, por defecto se encuentra seleccionado el algoritmo “ZeroR”. Para ejecutarlo simplemente haz click en el botón “Start”.

El algoritmo ZeroR selecciona como clase probable para una instancia la clase mayoritaria en el dataset. Debido a que las tres especies de Iris están igualmente presentes en el dataset, se elige la primera: (setosa) y usa eso para hacer todas las predicciones. Esta es una línea base para el conjunto de datos y la medida por la cual todos los algoritmos pueden ser comparados. El resultado es 33%, como se esperaba (3 clases, cada uno igualmente representado, asignando uno de los tres a cada predicción resulta en un 33% de precisión de clasificación)

Observa también que en las opciones de test aparece por defecto Cross Validation con 10. Esto significa que el dataset se divide en 10 partes: las primeras 9 partes se usan para entrenar el algoritmo y la décima se usa para evaluar al algoritmo.

4. Seleccionar y ejecutar otro algoritmo de minería de datos. Haz clic en el botón “Choose” en la sección “Classifier” y posteriormente haz click en “trees” y en el algoritmo “J48”. Para finalizar haz click en el botón “Start” para ejecutar nuevo algoritmo.

5. Analizar los resultados. Después de ejecutar el algoritmo J48, puedes ver los resultados en la sección “Classifier output”. Al igual que el algoritmo anterior, este algoritmo se ejecutó usando 10-fold cross-validation

   Se puede observar que el modelo establecido obtiene el siguiente resultado: clasifica correctamente 144 de las 150 entradas (96%), lo cual es bastante superior a la línea base (33%) del algoritmo anterior. En la matriz de confusión se puede ver una tabla en donde se comparan los datos reales con los datos predichos. En la primera fila de la matriz vemos que hay un error y una flor setosa fue clasificada/predicha como una flor versicolor, mientras que en la última fila vemos que 2 flores del tipo virgínica fueron también clasificadas como versicolor.

2.3 Clasificadores y reglas de asociación

En este caso, vamos a utilizar un segundo conjunto de datos, que recoge datos meteorológicos de varios días, junto con la indicación de si ese día se pudo o no jugar al tenis. El objetivo es poder predecir, a partir de la climatología de un día concreto, si ese día se podrá o no jugar al tenis. En este caso, compararemos los resultados del algoritmo de clasificación con un sistema de deducción de reglas de asociación

1. Ejecución de algoritmo de clasificación con el conjunto de datos (dataset) weather.nominal.arff. Este fichero contiene, como hemos dicho, datos sobre los días que se ha podido jugar al tenis en función de diversos aspectos meteorológicos, y se utilizará para tratar de predecir si hoy se podrá jugar al tenis o no.

   a. Para poder trabajar con el fichero es necesario cargar los datos en el área de trabajo. Para ello, se hace click en el botón Open file del entorno preprocess y se selecciona el fichero correspondiente.

   b. Cuando se carga un documento se muestra información descriptiva de su contenido. Por ejemplo, en este caso, se puede observar que este fichero contiene 14 instancias (registros). Cada una de las instancias consta de 5 valores para los siguientes atributos (outlook, temperature, humidity, windy, y play). Si se selecciona cada uno de los atributos, en la parte derecha, se obtiene información acerca de sus valores

en el conjunto de instancias.

c. Se solicita la creación de un árbol de decisión partiendo de los datos descriptivos. Para ello se pulsa en la pestaña Classify y se selecciona un método de clasificación mediante el botón Choose. En primer lugar, seleccionamos el algoritmo clásico de aprendizaje de árboles de decisión J48 e indicamos que se usará como opción de testeo Use training set. Finalmente se hace click en el botón Start y se realiza el aprendizaje del modelo predictivo de forma automática.

d. Mostrar el árbol obtenido. Se debe hacer click con el botón derecho sobre el texto trees. J48 de la caja Result-list, situada en la esquina inferior izquierda, y seleccionar la opción Visualize Tree.

e. Analizad el árbol generado y redactad las conclusiones del análisis.

2. Vamos a probar otros algoritmos con el conjunto de datos (dataset) anterior. A continuación, se indican los pasos a ejecutar en Weka para tratar de descubrir reglas de asociación.

   a. En primer lugar, se debe seleccionar la ventana de *Associate*.

   b. En segundo lugar, indicar cuál es el algoritmo que vamos a usar para el aprendizaje de las reglas y los parámetros de este. En este caso vamos a considerar WEKA.associations.Apriori. Este algoritmo se puede configurar con varias opciones, por ejemplo, con la opción LowerBoundMinSupport se indica el límite inferior de soporte (o cobertura) requerido para aceptar una regla; con la opción minMetric se indica la confianza mínima para mostrar una regla de asociación; y con la opción numRules se establece el número de reglas que se mostrarán.

   c. Haz click en el botón Start. Para cada regla obtenida, se muestra la cobertura de la parte izquierda y de la regla, así como la confianza de la regla. Examina cada una de las reglas y analiza los resultados obtenidos. Redactad las conclusiones de dicho análisis.

2.4 Algoritmos de minería de datos aplicados a lenguaje natural

Probaremos ahora algoritmos de minería de datos aplicándolos a textos escritos en lenguaje natural (minería de textos). En este caso, tomaremos datos de opiniones de películas obtenidos de la base de datos IMDb, clasificadas en dos grupos, según la valoración global de la película por parte del espectador/opinador haya sido positiva o negativa. El objetivo será ser capaces, a partir del texto de una opinión, de decidir si dicha opinión ha sido globalmente positiva o negativa.

1. Descargar el conjunto de datos de revisiones de películas disponible en: http://www.cs.cornell.edu/people/pabo/movie%2Dreview%2Ddata/review_polarity.tar.gz.

2. A través del explorador de Weka, convierte el conjunto de datos en un fichero .arff. Para ello, utiliza "Open file" y selecciona el directorio que tiene dos subcarpetas ("pos" y "neg"). Obtendrás un mensaje de error porque Weka no sabe cómo abrir el directorio. Pulsa aceptar en la ventana de error. En ese momento se abre una ventana de diálogo en la que mediante el botón "Choose" se puede seleccionar la opción "TextDirectoryLoader" para cargar

directorios. A continuación, pulsa "OK". Por último, guarda el conjunto de datos (botón "Save").

3. Convierte el atributo "text" en un vector de palabras. Para ello, en la caja de "Filter" selecciona (botón "Choose"), la operación "StringToWordVector" (weka -> unsupervised -> attribute -> StringToWordVector).
    a. Selecciona el atributo text para aplicar este filtro sólo al primer atributo (text) y no al atributo @@class@@.
    b. Haz click en "Apply" para ejecutar el filtro.

4. Aplica un algoritmo de clasificación. Para ello, en la pestaña "Classify" selecciona (botón "Choose") el algoritmo "NaiveBayes". En el desplegable que hay en la parte izquierda, selecciona el atributo que se desea predecir (en este caso, "@@classd@@", que indica si la opinión de la película es positiva o negativa). Ejecuta el algoritmo (clic en "Start"). Anota los valores obtenidos correspondientes a los siguientes resultados:
    i. Precisión al detectar las opiniones positivas.
    ii. Precisión al detectar las opiniones negativas.
    iii. Recall al detectar las opiniones positivas.
    iv. Recall al detectar las opiniones negativas.
    v. Precisión promedio.
    vi. Recall promedio.

5. Ahora vamos a repetir la misma operación, pero pre-procesando un poco más los datos de entrada: elimina del vector de palabras aquellas que sean claramente irrelevantes (números, signos como * y +, etc.); para localizar estas palabras, puedes ayudarte de patrones (botón "Pattern"). Compara la precisión y recall globales obtenidos tras eliminar estas palabras: ¿mejora o empeora?).

6. Opcionalmente, puedes repetir la operación seleccionando parámetros diferentes a los de por defecto al aplicar la conversión a vector de palabras. Haciendo click en la caja que hay junto al botón "Choose" podemos cambiar los parámetros por defecto. ¿Consigues de algún modo mejorar la precisión y recall globales? (si es así, indica qué parámetros tienen más influencia).

7. Se debe elaborar un documento con las respuestas a los diferentes puntos y con comentarios y explicaciones.

2.5 Profundizando en la comprensión de los algoritmos

Por último, vamos a analizar con un poco más de detalle los algoritmos de clasificación. En este caso se trata de predecir el tipo de fármaco (drug) que se debe administrar a un paciente afectado de rinitis alérgica según distintos parámetros/variables. Las variables que se recogen en los historiales clínicos de cada paciente son:

- Age: Edad
- Sex: Sexo
- BP (Blood Pressure): Tensión sanguínea.
- Cholesterol: nivel de colesterol.
- Na: Nivel de sodio en la sangre.
- K: Nivel de potasio en la sangre.

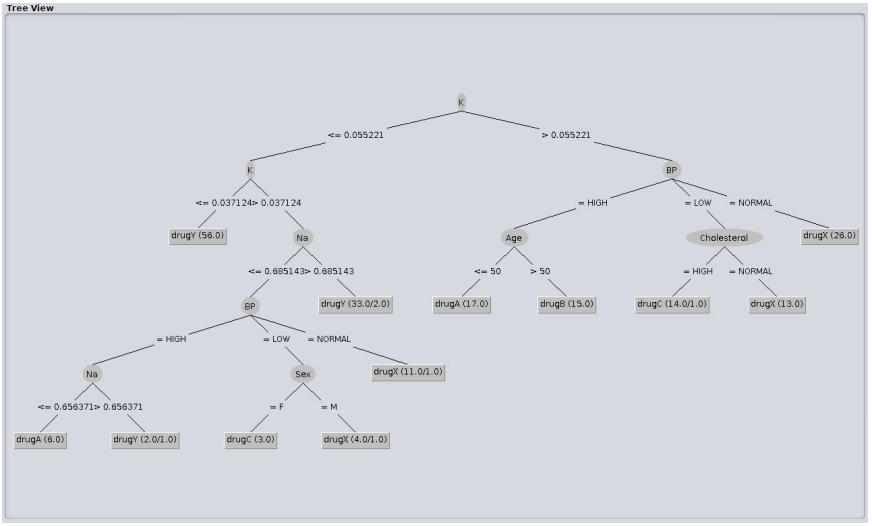
Hay cinco fármacos posibles: DrugA, DrugB, DrugC, DrugX, DrugY. Se han recogido los datos

del medicamento idóneo para muchos pacientes en cuatro hospitales. Se pretende, para nuevos pacientes, determinar el mejor medicamento a probar. Ésta sería la base de un sistema de ayuda a la prescripción farmacéutica.

1. Desde el Explorer, cargamos el fichero “drug1n.arff” facilitado en moodle.

2. La primera pregunta que nos podemos hacer es ver qué fármacos son más comunes en general, para ver si todos suelen ser igualmente efectivos en términos generales. Para ello seleccionamos el atributo drug, y de esta manera vemos la distribución por clases. Resulta evidente que el fármaco Y es el que tiene éxito en más pacientes. Este será el fármaco que nos recomiende un algoritmo como el ZeroR, que tendrá una precisión del 45%. Utilizar este algoritmo como recomendador, es lo mismo que proponer al paciente el fármaco Y (el más “eficaz”), si no funciona, le daríamos el X (el segundo más eficaz)... y así sucesivamente. Es una aproximación muy utilizada en la práctica médica habitual, pero seguramente seremos capaces de afinar un poco más, y poder determinar con mayor probabilidad qué fármaco va a ser el más eficaz para un paciente concreto, si analizamos más datos.

3. Si utilizamos el algoritmo de clasificación J48 (que ya hemos visto anteriormente), utilizando el conjunto de datos como “training set”, encontraremos un árbol de decisión bastante más complicado, con una precisión del 97%



Si probamos el mismo algoritmo utilizando una validación cruzada con 10 grupos, la precisión es algo menor, del 92.5%. Eso es normal. La precisión es menor, pero probablemente el sistema se comporte mejor en un caso de pacientes “reales”. Es muy probable que las reglas obtenidas utilizando los datos como “training set” estén un poco sobre-ajustados.

Podríamos probar otros algoritmos más complejos y sofisticados, pero vamos a realizar un análisis previo, que quizá nos ayude a mejorar la capacidad predictiva, incluso con el propio algoritmo J48.

4. Hay veces que algunas de las variables que utilizamos en el análisis, tienen una alta correlación entre sí. En este caso, el resultado final viene desvirtuado por esta correlación, y es preferible utilizar otra variable auxiliar que simplifique el conjunto de datos, y nos dé una mejor precisión. Si vamos a la pestaña “Visualize” del Explorer, podemos ver la relación que hay entre pares de variables. En color, vemos la variable nominal que estamos intentando “predecir”, en este caso, el fármaco recomendado. Vamos a perder un poco de tiempo

mirando las distintas gráficas que hay. De entre todas ellas, muestran una cierta correlación el nivel de potasio y la edad... pero mejor aún, los niveles de sodio y potasio. Esa gráfica nos permite casi de inmediato trazar una línea recta, por debajo de la cual siempre se recomienda Y, y por encima de la misma, ese fármaco no es recomendable en absoluto. Esto indica una altísima correlación entre ambas variables (K y Na), que podemos simplificar inmediatamente.

5. Re-procesaremos el fichero, sustituyendo las variables K y Na por el cociente entre ambas, K/Na. Para ello, creamos un nuevo atributo derivado, mediante el uso de filtros de preprocesado. Desde la pestaña "preprocess", pulsamos "choose" en Filter, y seleccionamos el filtro *Unsupervised.Attribute.Addexpresión*. Este filtro nos permite añadir un nuevo atributo a partir de una expresión basada en los atributos previos. Para configurar el filtro recién creado, pulsamos en el cuadro de texto, y retocamos los parámetros:
   Expresión: a6/a5
   Name: Na_to_K

   a6/a5 significa atributo 6/atributo 5, es decir, K / Na
   Pulsamos "apply" para crear el nuevo atributo. Una vez creado, podemos eliminar los atributos Na y K, marcando la casilla de selección y pulsando "Remove".

6. Ya tenemos un nuevo conjunto de datos retocado. Volvemos a ir a "Classify", y volvemos a probar el clasificador J48. Veremos que hemos mejorado la precisión considerablemente, con un árbol de decisión bastante más sencillo.

3. Entrega de la práctica

La práctica se realizará en grupos de tres personas. Cuando se finalice la práctica se debe entregar en un fichero denominado práctica5_NIA1_NIA2_NIA3.tar o práctica5_NIA1_NIA2_NIA3.zip (donde NIA1, NIA2 y NIA3 son los NIAs de los autores de la práctica) con el siguiente contenido:

1. Un fichero de texto denominado autores.txt que contendrá el NIA, los apellidos y el nombre de los autores de la práctica en las primeras líneas del fichero. Por ejemplo:

   NIA Apellidos Nombre

   ——————————————————————————————-

   100001 Apellido1 Apellido2 Nombre

   100002 Apellido1 Apellido2 Nombre

   100003 Apellido1 Apellido2 Nombre

2. Un fichero denominado memoriaPractica6.pdf cuya extensión no excederá de las 10 páginas donde se realicen los análisis y reflexiones requeridos en el enunciado. Opcionalmente en este documento se podrán añadir comentarios del modo de funcionamiento de la herramienta argumentando los comentarios con pruebas. Además, se debe indicar la metodología de trabajo empleada para el desarrollo de la práctica (recursos, herramientas utilizadas, distribución del trabajo, horas de trabajo, etc.), las dificultades encontradas durante la realización de la práctica, etc.) y notas a considerar.

Para la realización de esta práctica se han planificado 1 sesión. Por tanto, la fecha límite de

entrega es una semana después de la realización de la práctica en el laboratorio. Para la entrega del fichero .tar, se utilizará el enlace correspondiente disponible en la página de la asignatura en el Anillo Digital Docente (ADD) de la Universidad de Zaragoza en la plataforma Moodle 2 (http://moodle2.unizar.es). En caso de que no sea posible realizar la entrega a través del ADD se enviará dicho tar al profesorado de la asignatura mediante correo electrónico.

Procedimiento de corrección y recomendaciones

No se realizará ninguna defensa específica de la práctica, salvo las comprobaciones y revisiones que se realizarán el mismo día de su realización en el laboratorio.

- *Requisitos a cumplir (10 puntos)*
    - Realización de todos los apartados indicados en la práctica. Recogida y exposición de los resultados obtenidos con la herramienta utilizada (5 puntos).
    - Memoria técnica de la realización del proyecto. Comentarios, explicación e interpretación de los resultados (5 puntos).