Departamento de Informática
Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Aprendizaje Automático

Curso 2019-2020

Tutorial 2: Introducción a Weka

Preprocesado y clasificación de datos con la interfaz Explorer de Weka.

6 de febrero de 2020

- El principal objetivo de estos ejercicios es familiarizarse con el entorno de Weka Explorer.
- En la página web de Weka *http://www.cs.waikato.ac.nz/ml/weka/* podéis encontrar el manual de Weka así como más documentación (tutoriales, ejemplos, etc).
- Es importante seguir el orden establecido para realizar los ejercicios correctamente.

1. Introducción

Weka (Waikato Environment for Knowledge Analysis) es una plataforma de software libre para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka contiene una colección de algoritmos para el análisis y el modelado predictivo.

Figura 1: Weka es un ave endémica que habita en Nueva Zelanda.

1.1. Aplicaciones de WEKA

La figura 2 muestra la interfaz principal de WEKA con sus cuatro aplicaciones principales:

- **Explorer**: es un banco de trabajo que dispone de un conjunto de componentes que permiten: aplicar operaciones de filtrado a los datos importados ("Preprocess"), utilizar algoritmos de clasificación estadística y regresión ("Classify"), identificación de las reglas interrelacionales entre los atributos de los datos ("Associate"), uso de técnicas de clustering ("Cluster"), identificación de los atributos de mayor relevancia en la predicción ("Selected attributes") y visualización de la matriz de dispersión para analizar la entropía entre atributos ("Visualize").
- **Experimenter**: permite hacer una evaluación sistemática entre varias configuraciones de algoritmos diferentes sobre la colección de datos.
- **KnowledgeFlow**: se trata de una interfaz que cuenta con las mismas herramientas que "Explorer" pero permite diseñar en modo gráfico ("drag and drop") el flujo de conocimiento.

- **Simple CLI**: se trata de una consola de comandos que permite acceder a todas las funciones de WEKA.

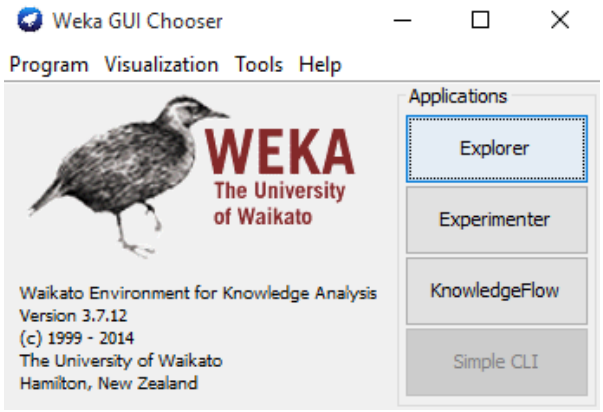

Figura 2: Interfaz principal de WEKA.

2. Instrucciones

Cada ejercicio tiene una serie de pasos que han de hacerse y/o contestarse en orden. Es necesario contestar claramente a cada una de las preguntas que se formulan en estos ejercicios, indicando el ejercicio y el paso al que corresponde cada respuesta. **No debe contener capturas de pantalla de código ni capturas con resultados de texto de la interfaz de Weka**. Se recomienda encarecidamente hacer tablas que sinteticen varios apartados, siempre y cuando estos apartados sean de comparación de algoritmos, comparación del mismo algoritmo con distintos ficheros o distintas evaluaciones del mismo algoritmo. También es necesario guardar separadamente los ficheros que se modifiquen en cada ejercicio para entregarlos junto con el documento.

3. Ejercicios

Ej. 1: Los ficheros de datos

- Descargar el fichero de datos `badges_plain.arff`.
- Abrir el fichero con cualquier editor y estudiar su contenido.

1. ¿Cuántos atributos de **entrada** tiene el fichero de datos? ¿De qué tipo son?
2. Podría un algoritmo de aprendizaje automático identificar una función capaz de predecir dicha clase con los datos que hay en el fichero. ¿Por qué?

Ej. 2: Clasificar con ID3

- Lanzar la herramienta WEKA
- Lanzar el Explorer
- Abrir el fichero `badges_plain.arff`

1. En la pestaña *Classify*, seleccionar el clasificador *trees/ID3* [1]. En las *Test options* seleccionar *Use training set* y pulsar el botón de *Start* para que se genere el modelo. ¿Cómo de buenos son los resultados?

[1] En caso que no encuentres instalado este clasificador, puedes incluirlo a través del *tools/package manager* instalando el paquete *simpleEducationalLearningSchemes*

Ej. 3: Generando nuevos atributos

En Aprendizaje Automático, los datos de entrenamiento a menudo pueden mejorarse gracias a la selección o extracción de características de los datos sin procesar recopilados inicialmente. Es importante utilizar atributos adecuados para cada tarea a realizar, por lo que a veces es necesario crear características más informadas que puedan mejorar el aprendizaje.

Partiendo de lo visto en el ejercicio anterior:

1. Imagina al menos 6 atributos que te parezca que podrían ser relevantes para este problema. Estos atributos inventados se deberían poder extraer tratando adecuadamente el único atributo de entrada del ejercicio anterior (*name*). Anótalos y describe en qué consiste cada uno. ¿Por qué has elegido esos atributos?

2. Abre el fichero de datos `badges1.arff` con Weka. ¿Cuántos atributos de **entrada** tiene el fichero de datos? ¿De qué tipo son?

3. ¿Qué otro tipo de información estadística se muestra sobre los atributos? Pulsa el botón "Visualize All" ¿Qué se muestra?

4. Tratar de generar un clasificador con *tree/ID3*. ¿Qué es lo que ocurre? ¿Qué se podría hacer para evitar este problema con ID3?

Ej. 4: Clasificar con ID3: resolviendo problemas

- Utilizando el mismo fichero de datos, volver a la ventana de preproceso.
- Seleccionar el atributo *name* y borrarlo.

1. Seleccionar el filtro *Filter/unsupervised/attribute/Discretize*, fijar el numero de *bins* a 5 y aplicar al conjunto de datos. ¿Qué efecto tiene este filtro?

- En la pestaña *Classifiers* elige ID3 y marca *Use training set* en las *Test options*. Vuelve a generar el clasificador.

2. ¿Cuántas instancias del conjunto de entrenamiento clasifica bien? ¿Qué porcentaje clasifica bien?

3. ¿Qué crees que indica la matriz de confusión?

4. ¿Cuántas instancias de cada tipo se han clasificado mal?

5. Pulsar el botón de *More Options* y seleccionar la opción *Output Predictions (PlainText)*. Volver a clasificar y revisar los resultados. ¿Cuál es la primera instancia del conjunto de entrenamiento que se clasifica mal? ¿Por qué?

6. ¿Cómo se clasificaría la instancia "Donald Trump"? ¿Cuáles son los atributos de este nombre? ¿Qué ocurre con los valores de esta instancia si utilizas el filtro usado anteriormente?

Ej. 5: Clasificar con ZeroR

- En la pestaña *Classifiers* elige *rules/ZeroR* y marca *Use training set* en las *Test options*. Vuelve a generar el clasificador.

1. ¿Qué modelo genera el clasificador ZeroR?

2. ¿Cuál es el porcentaje de éxito de este modelo?

3. ¿Cómo se clasificaría la instancia "Donald Trump"?

Ej. 6: Clasificar con J48 (C4.5)

- Volver a la pestaña de preproceso y cargar en Weka el fichero de datos `badges1.arff` de nuevo.
- En la pestaña *Classifiers* elige J48 y marca *Use training set* en las *Test options*. Vuelve a generar el clasificador.

1. ¿Cuántas hojas tiene el árbol generado con J48?
2. ¿Cuántas instancias del conjunto de entrenamiento clasifica bien?
3. ¿Qué porcentaje de instancias clasifica bien?
4. ¿Cuántas instancias de cada tipo se han clasificado mal?
5. ¿Cómo se clasificaría la instancia "Donald Trump"?
6. ¿Elegirías este modelo o el generado por ID3? ¿Por qué?
7. ¿Hemos encontrado la función exacta para generar las etiquetas? ¿Por qué lo sabes?

Ej. 7: Utilizando más atributos con J48 (C4.5)

1. Volver a la pestaña de preproceso y seleccionar el filtro *Filter/unsupervised/attribute/AddExpression* para generar un nuevo atributo que calcule el número de vocales.
2. Grabar el conjunto de datos como `badges-Ej7.arff`.
3. ¿Podrías decir cuál es el rango de vocales más común en el fichero proporcionado?
4. Volver a construir un clasificador con J48 con el conjunto de datos del punto anterior. En este caso tienes que seleccionar la clase "*class*" en el desplegable de la pestaña *Classify*.
5. Anota el porcentaje de instancias bien clasificadas y la matriz de confusión.
6. Haz click con el botón derecho del ratón en el modelo generado que aparece en *Result list*. Visualiza el árbol generado con *Visualize Tree*. ¿Qué indican los números que aparecen en las hojas?
7. Ir a la pestaña *Visualize*.
8. Pinchar en la gráfica que relaciona el atributo creado con la clase.
9. Aumentar el valor de *Jitter*. ¿Qué efecto tiene?
10. Tras todos estos resultados, ¿qué características o cualidades crees que deben tener los atributos para maximizar el éxito de los algoritmos de aprendizaje automático?

Ej. 8: Balanceado de datos, selección de características y otros filtros

1. Cargar en Weka el fichero de datos `adult-data.arff`.
2. ¿Cuántos atributos de entrada tiene este fichero? ¿Cuántas instancias de entrenamiento?
3. Ejecuta el clasificador J48. Selecciona en Test Options la opción "Cross-validation" ¿Qué resultados aparecen? Explica el resultado.
4. Ahora vamos a evaluar el clasificador solamente con las instancias que figuren en el fichero `adult-test.arff`. Para ello selecciona en Test Options la opción "Supplied test set". ¿Qué resultados aparecen? ¿Estos resultados son comparables a los anteriores? ¿Por qué?
5. Vuelve a la pestaña Preprocess y haz click en el atributo de salida (la clase). ¿Qué proporción de datos hay de cada clase? ¿Crees que este porcentaje es apropiado para que un algoritmo de aprendizaje automático aprenda bien?
6. Vamos a modificar las instancias de entrenamiento para que tengan un porcentaje similar entre las dos clases. Para ello selecciona **supervised/instance/Resample** cambiando el parámetro **biasToUniformClass** a 1,0 ¿Qué ocurre con el atributo de salida? ¿Ha descendido el número de ejemplos de entrenamiento?

7. Tras aplicar este filtro, evalúa de nuevo con *cross-validation* y *supplied test set* el algoritmo J48. ¿Qué resultado ofrece ahora el algoritmo? ¿Ha mejorado o empeorado?

8. Por último aplica el filtro de normalización **unsupervised/instance/Normalize** para los atributos numéricos. ¿Qué resultados se obtienen?

9. Después del procesamiento de datos que has realizado en este apartado, ¿crees que esto ayuda al proceso de aprendizaje? ¿Por qué?

10. ¿Cuál es el mejor resultado obtenido? Justifícalo.

11. Grabar el conjunto de datos como `badges-Ej8.arff`.

4. Directrices para la documentación

El alumno deberá entregar una memoria en formato **PDF** que podrá tener una extensión máxima de 10 páginas, incluyendo la portada y el índice. La memoria debe contener al menos los siguientes contenidos:

- Breve descripción explicando los contenidos del documento.
- Las respuestas a cada una de las preguntas y subpreguntas que se formulan en los ejercicios del tutorial.
- No debe contener capturas de pantalla de código ni capturas con resultados de texto de la interfaz de Weka. Estos resultados se deberán mostrar adecuadamente en tablas siempre que sea posible.
- Conclusiones y dificultades encontradas.

5. Normas de entrega

El tutorial se debe realizar **obligatoriamente** en grupos de 2 personas y se entregará a través del entregador que se publicará en Aula Global **hasta las 23:55 horas del día 16 de Febrero de 2020**. El nombre del archivo comprimido debe contener los últimos 6 dígitos del NIA de los dos alumnos, `ej. tutorial2-123456-234567.zip`.
El archivo comprimido debe incluir lo siguiente:

- Una memoria en formato `PDF`, que deberá contener los contenidos descritos en la sección 4.
- Los ficheros guardados durante la realización de este tutorial: badges-Ej7.arff y badges-Ej8.arff

Se valorará la claridad de la memoria, la justificación de las respuestas a la preguntas propuestas, así como las conclusiones aportadas.