Departamento de Informática
Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Aprendizaje Automático

Curso 2020-21

Tutorial 3: Experimentación múltiple

Interfaces KnowledgeFlow y Experimenter de Weka

25 de febrero de 2021

- El objetivo de este tutorial es familiarizarse con dos interfaces de Weka que facilitan el diseño y la ejecución de múltiples experimentos de una sola vez.
- La interfaz KnowledgeFlow es una alternativa al Explorer manejado en el tutorial anterior, aportando algunas capacidades adicionales. La interfaz Experimenter permite ejecutar de una sola vez distintos algoritmos de aprendizaje automático sobre distintos conjuntos de datos y comparar los resultados usando tests estadísticos.
- En la web http://www.cs.waikato.ac.nz/ml/weka/ se puede encontrar el manual de Weka así como más documentación, como tutoriales y ejemplos.

1. Ejercicio 1: KnowledgeFlow

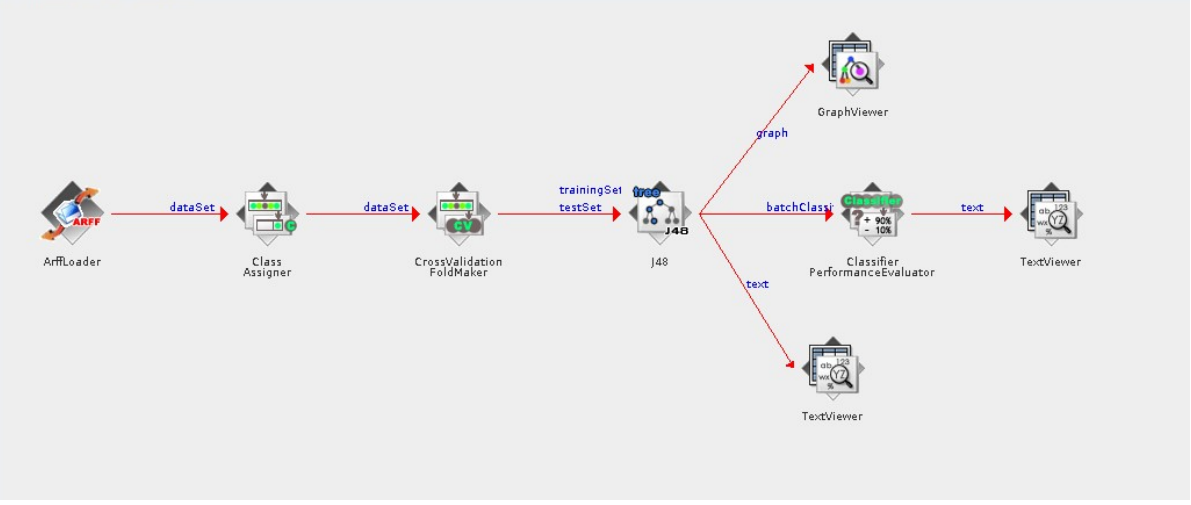

Figura 1: Flujo de conocimiento para analizar los datos en `adult-data.arff`.

1. Ejecuta WEKA con: `java -jar weka.jar -Xmx500m` para establecer la memoria en la ejecución.
2. Inicia el módulo KnowledgeFlow de Weka. Esta interfaz permite diseñar experimentos gráficamente. El usuario puede seleccionar los componentes de Weka de una barra de herramientas, situarlos en pantalla y conectarlos

para formar un flujo de conocimiento que recoja, procese y analice los datos. Los próximos apartados explican cómo construir el flujo de conocimiento que se muestra en la Figura 1.

3. Insertar un nodo del tipo `Arff Loader` (pestaña `DataSources`). Configurarlo para que lea el fichero `adult-data.arff` (botón derecho sobre el nodo opción `Configure`).

4. Insertar un nodo del tipo `Class Assigner` (pestaña `Evaluation`). Unirlo con el nodo anterior enviando el `dataSet` (botón derecho en el nodo anterior opción `dataSet`). Configurar el nodo para que la clase sea la variable llamada `salary` (debería ser la opción por omisión).

5. Insertar un nodo del tipo `Cross Validation FoldMaker` (pestaña `Evaluation`). Unirlo con el nodo anterior enviando el `dataSet` (botón derecho en el nodo anterior opción `dataSet`). En la configuración podremos elegir el número de `Folds` (por defecto son 10) y la semilla aleatoria (por defecto a 1).

6. Insertar un nodo del tipo J48 (pestaña `Classifier`). Unirlo también con el nodo anterior enviando `trainingSet` (botón derecho en el nodo anterior, opción `trainingSet`). Unirlo con el nodo anterior enviando `testSet`. Hay que enviar tanto el conjunto de entrenamiento como el conjunto de test, porque, si no, la validación cruzada no funcionará correctamente.

7. Insertar un nodo del tipo `Classifier PerformanceEvaluator` (pestaña Evaluation). Unirlo con el nodo anterior enviando `batchClassifier`. Dejar el resto de opciones por defecto.

8. Insertar un nodo del tipo `TextViewer` (pestaña `Visualization`). Unirlo con el nodo J48 enviando text.

9. Insertar un nodo del tipo `TextViewer` (pestaña `Visualization`). Unirlo con el nodo `Classifier Performance Evaluator`, enviando `text`.

10. Insertar un nodo del tipo `GraphViewer` (pestaña `Visualization`). Unirlo con el nodo `J48`, enviando `graph`.

11. Ejecutar el flujo de conocimiento. Seleccionar la opción `Show Results` en los nodos `TextViewer`. ¿Qué se muestra en cada uno de ellos? ¿Cuál es el porcentaje de instancias clasificadas correctamente?

12. Guarda el diagrama de flujo de conocimiento. Este fichero hay que incluirlo en la entrega.

13. ¿Cuál es la utilidad de crear flujos de conocimiento con esta interfaz de Weka?

2. Ejercicio 2: Experimenter

El desarrollo de esta parte del tutorial se centra en modelos que permitan clasificar la dirección de Pac-Man. Este ejercicio se apoya en la función básica de extracción de características programada en el tutorial 1.

2.1. Generación y preprocesado de datos de Pac-Man

1. Modificar la función de escritura de fichero realizada en el tutorial 1, de modo que genere un fichero .arff [1] legible por Weka (que contenga las cabeceras y los ejemplos). Cada línea se corresponde con una instancia del estado del juego seleccionada en el tutorial 1 y, además, la dirección que ha tomado Pac-Man.

2. Generar un fichero de entrenamiento de nombre "all_data_pacman.arff" con más de 500 instancias. Para ello, ejecutar el juego con los siguientes parámetros: *python busters.py -g RandomGhost -l openHunt*. Se muestra el mapa *oneHunt*, el control es por teclado y los fantasmas tienen un comportamiento aleatorio.

3. Crear dos nuevos ficheros a partir del generado anteriormente. En cada uno se debe seleccionar un subconjunto de atributos diferente a criterio del alumno. Guardar dichos ficheros con el nombre "filter_data_pacman_manual1.arff" y "filter_data_pacman_manual2.arff".

De modo que se tendrán 3 ficheros .arff distintos con más de 500 instancias cada uno.

[1] http://www.cs.waikato.ac.nz/ml/weka/arff.html

2.2. Diseño y ejecución del experimento

Ahora vamos a proceder a diseñar y ejecutar el experimento con Experimenter. Para ello hay que realizar los siguientes pasos:

1. Ejecutar Weka.
2. Abrir el *Experimenter*.
3. Pulsar el botón *New* para generar un nuevo experimento.
4. Seleccionar *Classification* en el tipo de experimento.
5. Seleccionar los tres conjuntos de datos anteriores.
6. Seleccionar los algoritmos J48, IbK para varios valores de k (al menos k=1,3,5), PART, ZeroR y NaiveBayes.
7. En el apartado *Results Destination* seleccionar ARFF y utilizar el botón *Browse* para elegir el fichero. Este fichero contendrá los datos y resultados del experimento y se podrá abrir como una hoja de cálculo cuando termine el *Experimenter*.
8. Guardar el experimento seleccionando pulsando *Save*, eligiendo el nombre que se prefiera para el experimento.
9. Pulsar la pestaña *Run* y después el botón *Start*.

2.3. Análisis de los resultados

1. Pulsar la pestaña *Analyse* para analizar los resultados del experimento.
2. Pulsar el botón *Experimenter* para seleccionar los resultados del experimento actual.
3. Seleccionar *Percent_correct* en el *Comparison field*, y después seleccionar *Perform_test*.
4. Los caracteres v y * indican si el resultado es significativamente mejor (v), peor (*) o igual ( ) que el esquema base, con el nivel de significación especificado (por omisión 0.05). Por otro lado, los números entre paréntesis (v/ /*) que aparecen debajo de cada esquema indican el número de veces que el esquema es mejor (v), igual y peor (*) que el esquema base.
5. ¿Hay algún conjunto de datos que parezca más adecuado?
6. ¿Qué algoritmo te parece más adecuado?
7. ¿Son los resultados del mejor algoritmo mucho mejores que los del resto?
8. Grabar en fichero tanto la configuración del experimento como los resultados del análisis.
9. ¿Por qué o para qué te parece adecuado el uso del *Experimenter* de Weka?

3. Directrices para la documentación

Se 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 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 acerca del tutorial.

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

4. 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 4 de Marzo de 2021**. El nombre del archivo comprimido debe contener los últimos 6 dígitos del NIA de los dos alumnos, `ej. tutorial3-123456-234567.zip`.

El archivo comprimido debe incluir lo siguiente:

- Una memoria en formato `PDF`, que deberá contener al menos los contenidos descritos en la sección 3.
- Los diferentes ficheros generados durante el tutorial, en el que debe estar la información generada por el simulador.
- Código empleado para la extracción de las características del simulador.
- El fichero del Experimenter de Weka y el fichero de resultados del Experimenter.
- Los ficheros generados por el KnowledgeFlow de Weka.