



DEPARTAMENTO DE INFORMÁTICA
UNIVERSIDAD CARLOS III DE MADRID

Grado en Ingeniería Informática

Aprendizaje Automático
Curso 2015-2016

Práctica 1: Clasificación y predicción

Creación de un agente automático de Pac-Man con Weka

25 de febrero de 2016

Esta práctica consiste en la aplicación de técnicas de Aprendizaje Automático para llevar a cabo tareas de predicción y clasificación en el Pac-Man. Para ello se continuará trabajando sobre los cambios ya realizados para el tutorial 1 en la plataforma facilitada.

1. Introducción

El Pac-Man original, conocido coloquialmente como Comecocos en España, es un videojuego arcade lanzado en 1980. El protagonista del juego, Pac-Man, es un círculo amarillo al que le falta un sector a modo de boca. Su objetivo es comerse todos los puntos que hay sobre los pasillos de un laberinto y huir de unos fantasmas. Cuando se come un punto grande, los fantasmas se vuelven vulnerables por un tiempo y Pac-Man puede comérselos para obtener más puntuación.

En la plataforma de Pac-Man que se utiliza en las prácticas de esta asignatura, el juego se ha simplificado para focalizar la tarea de aprendizaje. El objetivo es comerse al menos una vez a todos los fantasmas en el menor tiempo posible, teniendo en cuenta que los fantasmas aquí siempre son vulnerables.

Cuando se come un fantasma, éste reaparece en el tablero a menos que se definan en él unas “celdas” para cada uno, donde los fantasmas se quedan encerrados. Allí se colocan automáticamente cuando son comidos.

Para esta práctica se van a desarrollar 2 tipos de agentes para Pac-Man:

1. **Extractores de ejemplos de entrenamiento:** Estos agentes guardan en un fichero .arff su estado del mundo en cada turno. Se utilizará como base la función realizada en el tutorial 1 para grabar instancias en un fichero. Esta función se deberá copiar en el agente controlado por teclado y en el agente automático manual que se creó para el tutorial 1.
2. **Agentes automáticos:** Tras analizar los ejemplos de aprendizaje recogidos por los agentes extractores de características, con Weka se generará un modelo de clasificación que, una vez implementado, permitirá jugar bien a un nuevo agente automático.

Adicionalmente, también se creará un modelo de predicción con Weka utilizando técnicas de regresión, que podría ser aprovechado por el agente automático que el alumno debe desarrollar.

Esta práctica se distribuye en 3 fases. Gran parte hay que dedicarla a la experimentación. Será necesario probar distintos algoritmos, atributos de entrada, formas de capturar ejemplos, y formas de evaluación de cada agente. Toda esta información deberá ir convenientemente resumida y comparada en tablas y gráficas, según se considere oportuno.

2. Fase 1: Instancias de entrenamiento y test

En la primera fase será necesario modificar la función de extracción de características del tutorial 1 para que cree un fichero de datos .arff legible por Weka. También deberá incluir un mecanismo que guarde, en una misma instancia, atributos que solo se sabrán a N turnos vista (al menos con N igual a 2 y 5).

Esta función de extracción servirá como base para esta práctica y todas las demás, por lo que es de vital importancia que esté bien programada lo antes posible. Para ello se deberán seguir los siguientes pasos.

1. Modificar la función de extracción de características realizada en el tutorial 1, de modo que genere un fichero .arff legible directamente por Weka (que contenga las cabeceras con la definición correcta de cada atributo, etc.).
2. Cada línea con datos que almacenaba originalmente será ahora una instancia interpretable por Weka. Cada instancia tendrá toda la información del estado y, al final, la acción que se ha ejecutado (arriba, abajo, izquierda o derecha). Esta acción será la clase que se clasificará en la fase 2.
3. También será necesario incorporar un mecanismo a esta función que sea capaz de guardar en una misma línea cierta información futura a N turnos vista. Para la fase 3 de esta práctica, es necesario que una misma instancia contenga el atributo *score* (la puntuación en el turno actual), y los atributos *scoreN* (la puntuación dentro de N turnos). Debe haber atributos al menos para N igual a 2 y 5. La puntuación será lo que se intentará predecir con los algoritmos de regresión de la fase 3.
4. Si procede, añadir a la función nuevos atributos derivados que se consideren que podrían ser útiles para el proceso de aprendizaje.
5. Implementar esta función en el agente controlable por teclado y en el programado por el alumno en el tutorial 1 para que jugase automáticamente.
6. Jugar con el agente manual (BustersKeyboardAgent) y el automático desarrollado en el tutorial 1 para generar datos de entrenamiento y test. De entrenamiento se pueden usar unas 300-500 instancias y de test unas 100-200. Las instancias de test no pueden ser una copia de las de entrenamiento, deben capturarse nuevas para este conjunto. Se debe obtener una colección de al menos estos 6 ficheros:
 - **training_keyboard.arff**: Instancias de entrenamiento del agente controlado por el teclado con varios mapas.
 - **test_samemaps_keyboard.arff**: Instancias de test del agente controlado por el teclado, usando solamente los mismos mapas que se usaron para el fichero de entrenamiento.
 - **test_oothermaps_keyboard.arff**: Instancias de test del agente controlado por el teclado, usando solamente mapas distintos de los que se usaron para el fichero de entrenamiento.
 - **training_tutorial1.arff**: Igual que en el caso del teclado, pero para el agente automático implementado por el alumno en el tutorial 1.
 - **test_samemaps_tutorial1.arff**: Como el anterior.
 - **test_oothermaps_tutorial1.arff**: Como el anterior.

Por tanto, una misma instancia de entrenamiento va a tener tres tipos de información:

- **Estado actual**: Atributos acerca del estado (turno) actual del juego. Serán los atributos de entrada de los algoritmos de aprendizaje.
- **Acción realizada**: La acción que ha realizado el agente después de evaluar el estado actual. Será el atributo de salida del algoritmo de clasificación (lo que se quiere clasificar en la fase 2).
- **Información futura a N turnos vista**: Atributos que solo se sabrán una vez se haya jugado dentro de N turnos. Serán atributos de salida de los algoritmos de predicción (lo que se quiere predecir en la fase 3).

3. Fase 2: Clasificación

Una vez obtenida la información a partir de una serie de partidas jugadas por los diferentes agentes, se deberá construir un modelo de clasificación que permita a Pac-Man moverse hacia los fantasmas y comérselos.

1. El objetivo del modelo de clasificación es que Pac-Man pueda decidir, dado un cierto estado del mundo, la acción a realizar de entre todas las acciones disponibles, o dicho de otro modo, la “tecla” que debe pulsar en un turno concreto: arriba, abajo, izquierda, derecha o ninguna.
2. **Transformación de datos:** Los ficheros de datos generados en la fase 1 contienen toda la información de todos los atributos. Conviene probar los conjuntos de datos con varios filtros diferentes, haciendo un análisis en profundidad de los atributos seleccionados y eliminando aquellos cuya información no sea generalizable para otros mapas diferentes. Considera al menos alguno de los filtros o procesos que se han utilizado/explicado en los tutoriales anteriores. También se podrán filtrar las instancias que no se consideren adecuadas para aprender.
3. Los atributos correspondientes a la información futura NO se pueden utilizar como atributos de entrada para clasificar, ya que forman parte del futuro y nunca estarán disponibles a tiempo para decidir qué acción hay que realizar en el instante actual. Se recomienda guardar el fichero con los atributos eliminados para facilitar la experimentación.
4. **Experimentación:** Se deberá experimentar con distintos algoritmos de clasificación, utilizando los ficheros de datos generados en la fase 1. Será necesario comentar la calidad de cada uno y compararlos mediante algún tipo de tabla y/o gráfico. Considera los algoritmos utilizados en los tutoriales previos y alguno/s que se encuentren en la herramienta pero que aún no se hayan trabajado. Para poder aplicar algunos algoritmos, puede ser necesario aplicar diversas transformaciones (normalización, discretización, balanceo, etc.), que deben estar bien documentadas en la memoria para cada algoritmo.
5. Para evaluar los modelos será necesario utilizar dos conjuntos de test diferentes de los de entrenamiento. Uno con los mismos mapas que se usaron en el entrenamiento y otro conjunto de test que solo tenga mapas distintos (no entrenados).
6. **Implementación:** Una vez que se haya hecho la experimentación será necesario elegir uno de los modelos obtenidos para implementarlo en un nuevo agente y comprobar su comportamiento. Para ello se debe programar manualmente el modelo que muestra Weka en la parte de “*Classifier output*” usando *training set* o *cross-validation*.

Es probable que durante esta fase (y la siguiente), aparezca la necesidad de añadir algún nuevo atributo que no se había pensado inicialmente. Una vez se tienen los ejemplos capturados, este nuevo atributo se puede calcular con Weka o programar en la función de extracción de características. Programarlo en la función puede implicar que haya que volver a tomar todas las instancias de entrenamiento y test, pero puede ahorrar mucho tiempo realizando experimentos.

Describir y analizar todos estos resultados de la experimentación, describiendo textualmente cada uno y comparando en tablas y/o gráficos los resultados. Para las pruebas que se hagan se deben variar los siguientes elementos para comparar los resultados de cada configuración:

- Instancias de dos agentes: manual por teclado y el automático programado en el tutorial 1 (de manera opcional se puede repetir el análisis con una mezcla de ambos).
- Dos ficheros de test distintos de los de entrenamiento: uno solo con mapas ya entrenados y otro solo con mapas sin entrenar.
- Varios algoritmos de clasificación diferentes, con al menos uno que no se haya visto en otros tutoriales.

4. Fase 3: Predicción

Esta fase tiene como objetivo explorar los algoritmos de predicción de Weka. El procedimiento de experimentación será equivalente al de la fase 2, al igual que la documentación del mismo que deberá realizarse en la memoria.

Una vez obtenida la información a partir de una serie de partidas jugadas por los diferentes agentes seleccionados se deberá construir unos modelos de predicción. Para ello será necesario seguir estos pasos.

1. El objetivo de los modelos de predicción será estimar la puntuación que va a tener el agente dentro de N turnos (atributo *scoreN*, donde N al menos tendrá que tener los valores 2 y 5). Se hará un modelo de predicción por cada valor de N.
2. **Transformación de datos:** Se deberá realizar un análisis de la estructura de los datos del conjunto de entrenamiento para adaptarlos a cada uno de los algoritmos predictivos que se vayan a utilizar, en caso de que sea necesario.
3. **Experimentación:** Seleccionar varios algoritmos (al menos dos) que permitan realizar predicción a partir del conjunto de entrenamiento obtenido.
4. Comparar la calidad de cada algoritmo para cada modelo predictivo de forma equivalente a la realizada en la fase 1 (dos tipos de ficheros de test, etc.).
5. **Implementación:** Seleccionar un modelo predictivo e implementarlo dentro de la función de extracción de características. El nuevo atributo predicho formará parte de la parte del “estado actual” de la instancia, en lugar de la parte de la información futura.

Describir y analizar todos estos resultados de la experimentación de forma similar a la fase 1, teniendo en cuenta ahora se piden al menos 2 modelos predictivos.

5. Preguntas

Responder de forma clara a cada una de estas preguntas:

1. ¿Qué diferencias hay a la hora de aprender esos modelos con instancias provenientes de un agente controlado por un humano y uno automático?
2. ¿Crees que los resultados del modelo de regresión a 5 turnos vista guardan relación con los de 2 turnos? ¿Por qué?
3. Si quisieras transformar la tarea de regresión en clasificación ¿Qué tendrías que hacer? ¿Cuál crees que podría ser la aplicación práctica de predecir la puntuación?
4. ¿Qué ventajas puede aportar predecir la puntuación respecto a la clasificación de la acción? Justifica tu respuesta.
5. ¿Crees que se podría conseguir alguna mejora en la clasificación incorporando un atributo que indicase si la puntuación en el instante actual ha descendido o ha bajado?

6. Directrices para la documentación

El alumno deberá entregar una memoria en formato **PDF** que debe contener al menos los siguientes contenidos:

- Portada.
- Breve descripción explicando los contenidos del documento.
- Fase 1: Explicación de la función de extracción de características que se ha programado, así como del mecanismo para incluir datos futuros en una misma instancia.
- Fase 2: Explicación de la experimentación tal como se explica en el enunciado. Debe incluir la justificación de los algoritmos seleccionados, de los atributos seleccionados y de cualquier tratamiento sobre los datos que se haya llevado a cabo. Se concluirá con un análisis de los resultados producidos por los algoritmos elegidos y justificación de la elección del modelo final.
- Fase 3: Igual que en la fase 2, pero con cada modelo de regresión.
- Las respuestas a cada una de las preguntas que se formulan en el apartado 5.
- 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:
 - Conclusiones técnicas sobre la tarea que se ha realizado.
 - Apreciaciones más generales como: para qué puede ser útil el modelo obtenido, si al realizar la práctica se os han ocurrido otros dominios en que se pueda aplicar aprendizaje automático, etc.
 - Descripción de los problemas encontrados a la hora de realizar esta práctica.
 - Comentarios personales. Opinión acerca de la práctica. Dificultades encontradas, críticas, etc.

7. Normas de entrega

La práctica 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 9 de Marzo de 2016**. El nombre del archivo comprimido debe contener los últimos 6 dígitos del NIA de los dos alumnos, ej. `practica1-123456-234567.zip`. El archivo comprimido debe incluir lo siguiente:

- Una memoria en formato **PDF**, que deberá contener al menos los contenidos descritos en la sección 6.
- El código fuente del agente por teclado y del agente automático hecho por el alumno que incluyan la función de extracción de características.
- Los diferentes ficheros de ejemplos utilizados para la generación y evaluación de los modelos.
- Los diferentes modelos de clasificación y predicción generados por Weka.
- El agente implementado de la fase 2, así como la salida de Weka del modelo en que se ha basado este agente.

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

El peso de esta práctica sobre la nota final de la asignatura es de 1 punto.