Departamento de Informática
Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Aprendizaje Automático

Curso 2020-2021

Práctica 1: Clasificación y predicción

28 de febrero de 2021

Esta práctica consiste en la aplicación de técnicas de Aprendizaje Automático para llevar a cabo tareas de predicción y clasificación en el Pac-Man utilizando Weka. Además, con los modelos generados, se construirá un agente que controle de forma automática a Pacman.

1. Introducción

En la plataforma de Pac-Man que se utiliza en las prácticas de esta asignatura el objetivo es comerse a todos los fantasmas en el menor tiempo posible, maximizando la puntuación obtenida. Para el desarrollo de esta práctica se utilizará el simulador proporcionado que permite controlar a Pac-Man de manera automática y recoger información del entorno. Esta información (número de fantasmas, distancia a los fantasmas, distancia al punto de comida más cercano, etc) podrá ser utilizada para aplicar diferentes técnicas de Aprendizaje Automático y conseguir que Pac-Man sea capaz de resolver una tarea de forma automática.

Para ello se deberán generar un conjunto de ejemplos de entrenamiento (o experiencias de aprendizaje) que serán utilizadas como entrada para las técnicas de clasificación y de predicción (regresión). De manera más precisa, esta práctica consiste en:

- Generar un modelo que permita clasificar qué acción debe de realizar Pac-Man ante una situación determinada.
- Generar un modelo de regresión que permita, dada la información y la acción ejecutada en el tick actual, predecir la puntuación del tick siguiente.
- Emplear uno de los modelos generados anteriormente para construir un agente automático que juegue a Pac-man.

2. Tareas a realizar en la práctica

Para el desarrollo de esta práctica el alumno deberá desarrollar tres modelos que permitan a Pac-Man realizar ciertas acciones de manera automática. La práctica se divide en cuatro fases (Recogida de la información, Clasificación, Predicción, Construcción de un Agente Automático). A continuación se realiza una descripción detallada de cada una de las fases:

3. Fase 1: Recogida de información

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 en el tick siguiente.

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 de datos 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 del tick siguiente. Para la fase 3 de esta práctica, es necesario que una misma instancia contenga los atributos *score* (la puntuación en el turno actual), y *scoreSiguiente* (la puntuación del turno siguiente). La puntuación del turno siguiente 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. 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 5 mapas combinando partidas en las que los fantasmas se mueven de forma aleatoria y en los que no.
   - **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_othermaps_keyboard.arff**: Instancias de test del agente controlado por el teclado, usando solamente 5 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_othermaps_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 (posición del Pac-Man, número de fantasmas, distancia a la bola de comida más cercana, etc.). Serán los atributos de entrada de los algoritmos de aprendizaje. **Es trabajo del alumno seleccionar los atributos que se consideren para las tareas de clasificación y predicción que se plantean**.
- **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 del siguiente turno**: Atributos que solo se sabrán una vez se haya jugado el turno siguiente. Serán atributos de salida de los algoritmos de predicción (lo que se quiere predecir en la fase 3).

4. 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 e 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. 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).

Describir y analizar todos los resultados obtenidos de la experimentación, describiendo textualmente cada uno y comparando en **tablas y/o gráficos**. Para las comparaciones se pueden variar los ficheros utilizados para el entrenamiento y test, y los algoritmos (será necesario hacer comparativas con al menos un algoritmo de clasificación que no se haya visto en tutoriales anteriores).

5. 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 un modelo de predicción. Para ello será necesario seguir los siguientes.

1. Se generarán un modelo de predicción cuyo objetivo será predecir la puntuación que va a tener el agente en el turno siguiente.

2. Transformación de datos: Equivalente al proceso descrito para esta misma tarea en la fase 2.

3. Experimentación: Equivalente al proceso descrito para esta misma tarea en la fase 2, sólo que en este caso hay que seleccionar varios algoritmos (al menos dos) de predicción que permitan predecir la puntuación del tick siguiente.

4. Comparar la calidad de cada algoritmo de forma equivalente a la realizada en la fase 2 (dos tipos de ficheros de test, etc.).

Describir y analizar todos estos resultados de la experimentación de forma similar a la fase 2.

6. Fase 4: Construcción de un Agente Automático

En esta fase se pretende construir un agente automático que tome la decisión de qué acción ejecutar en cada instante a partir de alguno de los modelos generados anteriormente. **Será tarea del alumno seleccionar el modelo que considere adecuado y la forma en que se va a utilizar para construir este comportamiento automático**. Para ello es recomendable hacer uso del fichero *wekaI.py* que se suministra, y que permite utilizar en python los modelos generados en weka. Los pasos para poder utilizar este fichero correctamente son:

1. Asegurarse de tener instalado un compilador e intérprete de Java compatible, en caso contrario será necesario instalarlo:
   sudo apt install default-jdk

2. Abrir un terminal e instalar los paquetes *java-bridge* y *python-weka-wrapper* utilizando los comandos:

   pip3 install javabridge
   pip3 install python-weka-wrapper3

3. Importar la clase *Weka* que se encuentra en *wekaI.py* dentro de *bustersAgents.py* escribiendo en la cabecera:

   from wekaI import Weka

4. Arrancar la máquina virtual de java en el método *init* de la clase *BusterAgent* en *bustersAgents.py*:

   self.weka = Weka()
   self.weka.start_jvm()

5. Invocar al método *predict* de la clase *Weka* cada vez que se quiera obtener la clasificación/predicción de uno de los modelos anteriores que ayude a decidir qué acción debe ejecutar Pac-man. Por ejemplo, suponed que nuestro modelo es un árbol J48 que clasifica, dado un estado, la acción que debe ejecutar Pac-man. Previamente habremos creado este modelo con el *Explorer* de weka y lo habremos guardado con el nombre *j48.model*. En este caso invocaremos al método *predict* de la siguiente forma:

   x = [1.51299,14.4,1.74,'None',74.55,0,7.59,0,0]
   a = self.weka.predict("./j48.model", x, "./training_set.arff"))

   donde x es un array con los valores de la instancia que se trata de clasificar (y que variará dependiendo de la información seleccionada por el alumno), *training_set.arff* es el conjunto de entrenamiento utilizado para entrenar nuestro modelo J48, y a contiene la acción de salida que clasifica nuestro modelo J48 para esa instancia.

Una vez construido el agente, comparar sus resultados con el agente automático construido en el tutorial 1 y con el agente que juega por teclado.

7. 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. 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?
3. ¿Qué ventajas puede aportar predecir la puntuación respecto a la clasificación de la acción? Justifica tu respuesta.
4. ¿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?

8. 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.

- Fase 4: Explicación de qué modelo se ha seleccionado y por qué para construir el agente automático, y breve descripción del funcionamiento del mismo.
- Las respuestas a cada una de las preguntas que se formulan en el apartado 7.
- 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.

9. 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 Abril de 2021**. 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 8.
- 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.

Se valorará la claridad de la memoria, la justificación de las respuestas a la preguntas propuestas, así como las conclusiones aportadas. El peso de esta práctica sobre la nota final de la asignatura es de 1.5 puntos.