

UNIVERSIDAD CARLOS III DE MADRID

APRENDIZAJE AUTOMÁTICO

COMPUTER SCIENCE ENGINEERING

Práctica 1: Clasificación y Predicción

Authors:

Daniel MEDINA GARCÍA

Alejandro RODRÍGUEZ SALAMANCA

March 8, 2016

Contents

1	Fase 1	3
2	Fase 2	4
2.1	Elección de los datos de entrada para la creación del modelo	4
2.2	Elección del algoritmo generador de nuestro modelo	5
2.3	Elección de las modificaciones sobre los datos	5
2.4	Creación de nuevos atributos	6
2.5	Modificación de filtros	6
3	Fase 3	7
4	Preguntas	8
5	Conclusiones	9

Introducción

Abstract

1 Fase 1

Para extraer los datos del juego, se partió la función utilizada en el tutorial anterior. Esa base fue ampliada con experimentación, añadiendo y quitando atributos según mejoraba o empeoraba el éxito de clasificación del árbol resultante con distintos algoritmos, según explicaremos en la *Fase 2*.

La inclusión de datos futuros en las líneas del fichero ha sido implementada siguiendo un método de escritura retardada. Así, no se escribe una nueva línea en el fichero hasta que no se conocen todos los datos que se han de escribir, almacenando mientras tanto los datos intermedios. Esto se hizo creando listas globales, que almacenan las líneas incompletas (a falta de datos futuros) en `future_lines` y las puntuaciones (para completar las líneas anteriores cuando llegue su turno) en `future_score`. Cuando la primera línea esté completa (en este caso, en el sexto turno), se imprimirán en el archivo los datos que ya se tenían junto a los recién llegados, creando así una línea completa.

2 Fase 2

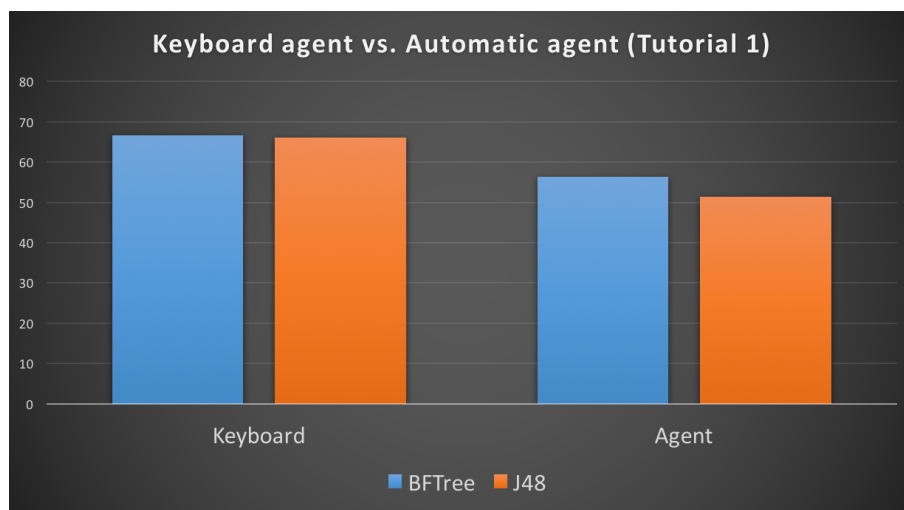
Explicación de la experimentación tal como se explica en el enunciado. Debe incluir la justificación de los algoritmos seleccionados, de los atributos seleccionados y de cualquier tratamiento sobre los datos que se haya llevado a cabo. Se concluirá con un análisis de los resultados producidos por los algoritmos elegidos y justificación de la elección del modelo final.

En esta fase el equipo dedicó su esfuerzo a realizar tantas pruebas como fuesen posibles, probando diferentes combinaciones de algoritmos, filtros y atributos para generar el modelo basándonos en conocimiento y razonamientos *a priori*.

Para crear un modelo exitoso, hay que tener en cuenta varios factores, a continuación analizaremos cada uno de ellos:

2.1 Elección de los datos de entrada para la creación del modelo

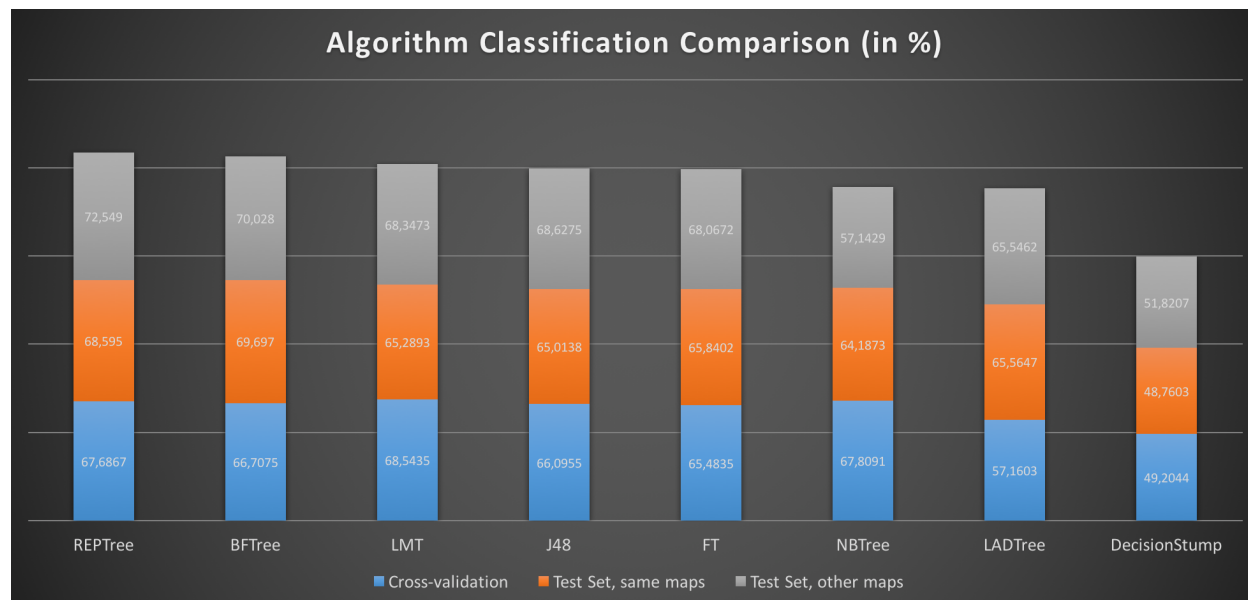
Para la elección de la entrada para el algoritmo, partimos de dos bases: los datos de ejecución del agente automático (que, para jugar, se informa tan sólo de los sensores que le indican las distancias a cada uno de los fantasmas) y los de las partidas que jugamos los miembros del equipo, viendo los fantasmas. No es difícil llegar a la conclusión de que el agente humano gana las partidas mucho más rápido que el agente implementado en el *Tutorial 1*, pero eso no tenía por qué indicar que necesariamente los datos del agente de teclado generasen un modelo con un mayor éxito de clasificación. Sin embargo, con posteriores pruebas encontramos un aplastante 10-15% de diferencia en éxito de clasificación a favor del agente de teclado entre los modelos generados para ambos archivos con *J48* y *BFTree* y evaluados con cross-validation, dos de los algoritmos que conocíamos de tutoriales anteriores y sabíamos generaban árboles bastante buenos. La siguiente tabla ilustra los resultados obtenidos:



Viendo estos datos, nos decantamos por `training_keyboard_edited.arff`, que es el fichero que contiene los datos legales (sin los datos de puntuaciones futuras), para generar el modelo en *Weka*.

2.2 Elección del algoritmo generador de nuestro modelo

Una vez escogido el archivo que se usará para generar el modelo, comenzamos la comparación entre algoritmos para realizar dicha tarea. Para ello, comparamos varios algoritmos de creación de árboles de decisión evaluándolos tanto con validación cruzada como con dos sets de instancias generadas en partidas jugadas, uno en los mismos mapas que en el entrenamiento y otro en mapas distintos. El éxito de clasificación de cada uno de los algoritmos utilizados se muestra en la siguiente tabla:



Nos sorprendieron los resultados de *REPTree*, un algoritmo que no habíamos utilizado con anterioridad y, sin embargo, obtuvo los mejores resultados. Si bien este algoritmo suele utilizarse para clasificaciones multi-clase, esto no le impide funcionar para una sola clase también, y queda demostrado que con efectividad. Esta información se tendrá muy en cuenta a la hora de elegir el algoritmo generador del modelo en el cual se basará nuestro nuevo agente automático para tomar las decisiones pertinentes, y nos sirve para descartar algunos algoritmos, agilizando los cálculos posteriores.

2.3 Elección de las modificaciones sobre los datos

Es este el factor donde el razonamiento sobre los datos puede dar lugar a más variantes. Ya que anteriormente no se han comentado, mencionamos aquí la base de atributos de los que partimos:

- **score** : puntuación actual de PacMan.
- **ghost<N>-living** : True si el fantasma <N> sigue vivo, False si ya ha sido comido.
- **distance-ghost<N>** : Distancia ruidosa de PacMan al fantasma <N>.
- **posX y posY** : Coordenadas de PacMan sobre el tablero.
- **direction** : Hacia dónde mira PacMan (último movimiento realizado).
- **wall-<direction>** : True si hay un muro en <direction>, False si vía libre.
- **move** : Movimiento tomado, es la **clase**.

La experimentación ha sido la protagonista, con numerosas pruebas descartadas ante la ausencia de resultados positivos, como son:

2.4 Creación de nuevos atributos

Tratamos de agregar los datos que teníamos de forma que resultasen en algún campo que fuese útil para el modelo.

Probamos convirtiendo los atributos `ghost<N>-living` en uno sólo que fuese la suma de los fantasmas vivos (aplicando primero el filtro *NominalToBinary* sobre los cuatro atributos, y luego el filtro *AddExpression* sumándolos todos), obteniendo un decremento de éxito de entre el 2.3% y el 3% dependiendo del método utilizado para la evaluación.

2.5 Modificación de filtros

Intentamos extraer el máximo de información de los datos que ya teníamos, sesgándolos mediante la aplicación de filtros que pudiesen acarrear un mejor funcionamiento de los algoritmos de prueba.

Como el movimiento **STOP** no aporta valor al juego, editamos nuestro fichero de datos eliminando las instancias que hubiesen realizado ese movimiento (i.e. aquellas transcurridas entre que empieza el juego y el jugador de teclado pulsa la primera tecla, ya que era el único momento en el que se producía este hecho) por considerarlas ruidosas. Una vez completada esta primera “purga”, procedimos a balancear las instancias de cada clase para tener un número similar de instancias que realizasen cada movimiento. Estos fueron los resultados obtenidos en cuanto a éxito de clasificación según los modelos generados:



Si bien observamos una clara mejora (de entre el 10 y el 20%) cuando evaluamos los modelos con validación cruzada, los resultados son desfavorables al evaluar los modelos con una pila de instancias adicional, en otros mapas, decrementando entre un 1 y un 11% el éxito en clasificación cuando lo comparamos con los datos en bruto.

3 Fase 3

Igual que en la fase 2, pero con cada modelo de regresión.

4 Preguntas

¿Qué diferencias hay a la hora de aprender esos modelos con instancias provenientes de un agente controlado por un humano y uno automático?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

¿Crees que los resultados del modelo de regresión a 5 turnos vista guardan relación con los de 2 turnos? ¿Por qué?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

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?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

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

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

¿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?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5 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.