

## Práctica 1

### Clasificación y predicción

Experimentación y desarrollo de un agente automático de MarioAI  
mediante técnicas atributo-valor.

Esta práctica consiste en la aplicación de técnicas de Aprendizaje Automático para llevar a cabo tareas de predicción y clasificación en el dominio de MarioAI. Para ello se utilizará la plataforma de la competición MarioAI Championship y la herramienta Weka.

### Introducción

Para el desarrollo de esta práctica se utilizarán los agentes T3BotAgent y T3HumanAgent que se han ido desarrollando en los tutoriales anteriores. La idea básica es generar un agente P1BotAgent que intente pasarse niveles de MarioAI y cuyo comportamiento sea más eficiente que el de BaselineAgent. Su comportamiento debe ser producido únicamente por algún modelo (o modelos) de los que haya generado Weka.

La generación de ejemplos de entrenamiento (también llamadas instancias de entrenamiento o experiencias de aprendizaje) ya debería estar prácticamente desarrollada tras la realización de los tutoriales anteriores. Toda su información (estado de Mario, número de monedas, localización de los enemigos, etc.) debe ser seleccionada, filtrada y posteriormente utilizada en Weka como entradas para las técnicas de predicción (regresión) y clasificación.

En esencia, las tareas de aprendizaje para las cuales se busca hacer una experimentación razonada son estas:

1. Elegir uno o varios modelos de clasificación complementarios que permitan determinar si Mario debería avanzar y/o saltar ante una situación determinada.
2. Elegir un modelo de regresión que permita predecir la recompensa intermedia que obtendrá Mario en el *tick*  $n + 24$  en función de la información y de la acción ejecutada en el *tick*  $n$ .

En la programación del agente P1BotAgent se utilizarán uno o varios de los modelos seleccionados y se experimentará para ver si la evaluación proporcionada por Weka coincide con la efectividad real observada en el agente programado.

Para la entrega de esta práctica, será necesario escribir una memoria con unas secciones adecuadas (consultar la sección “Directices para la documentación”) en las que se describan las pruebas realizadas, los resultados obtenidos, las conclusiones extraídas de ellos y la elección del algoritmo final que se ha programado. Junto a esta memoria, se entregará el agente P1BotAgent. Este agente podría ser candidato para participar en la Competición AA2018 de esta asignatura al final del curso, siempre y cuando su comportamiento esté íntegramente basado en técnicas de aprendizaje automático.

### 1. Recogida de información

A la hora de aplicar un algoritmo de aprendizaje automático es necesario obtener un conjunto de ejemplos de entrenamiento lo suficientemente grande para poder obtener información útil que suministrar a los diferentes algoritmos.

- Utilizar los agentes T3HumanAgent y T3BotAgent para recoger instancias de entrenamiento. Cada agente se deberá generar una serie de archivos .arff que después deberán ser utilizados en Weka. Estos ficheros almacenarán la información de cada instante (*tick*) del juego por línea. Los ficheros generados serán utilizados como entrada para las diferentes técnicas de Aprendizaje Automático que ofrece Weka. La información almacenada en el fichero debe ser obtenida a través de la plataforma MarioAI (escenario, enemigos, estado de Mario y otros atributos más avanzados desarrollados por los propios alumnos).
- Al terminar la práctica, la estructura de una instancia de aprendizaje sería similar a la expuesta en el Tutorial 3, con la peculiaridad de que las reglas de regresión pueden ser implementadas y el resultado predicho incluido como nuevo atributo del tick.
- Los modelos de clasificación utilizarán como clase la acción a ejecutar. Para esta práctica se pide que determine si hay que saltar y/o avanzar. Se puede crear un modelo con clase booleana para saltar y otro modelo aparte con clase booleana para avanzar. También se puede crear un único modelo con una clase de 4 valores posibles (parado, salta, avanza y salta+avanza). Si quiere usarse el Experimenter, la clase debe estar en la última posición de la línea.
- Los elementos de entrada pueden ser los que se consideren oportunos, teniendo en cuenta cuestiones de rendimiento, evitar que el modelo “memorice” un mapa concreto en vez de aprender a jugar en cualquiera, evitar usar atributos imposibles de obtener en un tick concreto para determinar después la acción a ejecutar, etc.
- Los modelos de regresión utilizarán los diferentes datos del futuro como atributo de salida (pero no de entrada). Al igual que con clasificación, si quiere usarse el Experimenter con regresión, el atributo de salida debe estar en el último lugar de la línea.
- La experimentación que se espera en la memoria exige hacer un análisis en profundidad de los atributos seleccionados en cada experimento, eliminando aquellos cuya información no sea generalizable para otros mapas diferentes. Igualmente la elección sobre qué instancias componen los .arff (de uno o varios agentes, humano o mezcla de ellos) puede variar en cada caso y su elección deberá ser razonada.

## 2. Clasificación

Para encontrar los modelos de clasificación requeridos será necesario hacer experimentos con varios ficheros .arff y con varios algoritmos de clasificación. Se puede utilizar cualquier interfaz de Weka de las que hemos visto hasta ahora en el curso, Explorer, KnowledgeFlow y Experimenter.

- Se deberán seleccionar varios algoritmos que permitan realizar clasificación a partir del conjunto de entrenamiento obtenido y valorar el algoritmo que mejores resultados tenga.
- Se deberá realizar un análisis de la estructura de los datos del conjunto de entrenamiento con el fin de poder aplicar las transformaciones necesarias (discretización, balanceo de clase, etc.) para poder utilizar los algoritmos previamente seleccionados. También se podrán filtrar las instancias generadas en la fase anterior para eliminar, por ejemplo, aquellas instancias o ejemplos de entrenamiento que no se consideren adecuados para aprender los modelos.
- La tarea a clasificar será si Mario puede avanzar y/o si tiene que saltar, para lo que se podrán generar uno o más modelos.
- La experimentación debe contener varios ficheros .arff con distintas instancias, así como pruebas con varias selecciones de atributos. También será importante utilizar varios mapas diferentes según el caso para ver si la solución es generalizable.
- En cada experimento realizado debe indicarse qué conjunto de entrenamiento se ha utilizado, qué filtros se han aplicado, qué tipo de algoritmo de clasificación se usa y qué tipo de evaluación del algoritmo se hace.

### 2.1. Predicción

Aparte de la clasificación será necesario hacer una predicción de la recompensa intermedia obtenida en el tick  $n+24$ . El tipo de experimentación para este punto debe ser similar a la de clasificación, pero aplicado a la tarea de predicción.

La experimentación debe incluir pruebas con las escalas de tiempo +6, +12 y +24 para ver si es capaz de ir hacia adelante.

## 2.2. Implementación

Implementar en un agente automático (puede tener el nombre que queráis) uno de los modelos aprendidos en la clasificación para hacer que Mario salte o no automáticamente en cada tick. Para ello se debe usar el modelo que muestra Weka en la parte de “*Classifier output*”.

En este agente automático y en P1HumanAgent y P1BotAgent será necesario implementar uno de los modelos de predicción, de forma que en cada tick se prediga el valor en +6, +12 y +24. Una vez implementado, el tick contendrá tanto los valores predichos para estos ticks futuros como los valores reales. Podrían hacerse pruebas utilizando los valores predichos como atributos de entrada en el apartado de clasificación por si estos valores son capaces de variar el comportamiento de Mario.

Para esta última parte recomendamos algoritmos basados en reglas o árboles de decisión.

## 3. Directrices para la documentación

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

- Portada e introducción explicando los contenidos del documento.
- Justificación de los algoritmos seleccionados.
- Descripción de los atributos seleccionados y de cualquier tratamiento sobre los datos que se lleve a cabo junto con todos los pasos realizados.
- Descripción de las instancias de aprendizaje seleccionadas en cada .arff
- Análisis de los resultados producidos por los algoritmos elegidos y justificación de la elección de los modelos finales.
- Breve descripción de las implementaciones realizadas.
- Conclusiones:
  - Conclusiones sobre qué se ha conseguido en la práctica que resuman los resultados finales.
  - 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.
- Comentarios personales. Opinión acerca de la práctica. Dificultades encontradas, críticas, etc.

## Normativa de entrega

1. Se debe entregar antes de la fecha límite indicada en Aula Global:
  - a) Memoria en formato .pdf (no se admite .doc, etc.) que contenga las respuestas a las preguntas y subpreguntas que se presentan en los ejercicios.
  - b) Agentes P1HumanAgent, P1BotAgent y el agente programado con la parte de clasificación.
  - c) Todos los ficheros externos que se mencionan en la experimentación, especialmente los .arff de los cuales se aprende después, sin filtrar, filtrados y seleccionados, convenientemente nombrados y organizados para que el profesor pueda consultarlos.
  - d) Cualquier otro fichero de apoyo que el alumno considere importante.
2. Es obligatorio que la entrega se haga en grupos de 2 personas. No se admiten grupos de 1 ni de más de 2.
3. Solo es necesario que entregue uno de los miembros del grupo.
4. La entrega debe comprimirse en un fichero .zip (no se admite .rar, .7z, etc.) y entregarse por Aula Global. El nombre del fichero debe tener un formato equivalente al del siguiente ejemplo: p1-387633-209339.zip. Donde los números son los 6 últimos dígitos del NIA de los alumnos.
5. No se admiten entregas fuera de plazo ni por email.
6. Se valorará la claridad de la memoria, el uso de tablas y en especial la justificación y conclusiones aportadas.
7. Se penalizará el uso de capturas de pantalla que contengan resultados de texto de la interfaz de Weka, así como no justificar respuestas, especialmente cuando se solicite de forma expresa.