

INTRODUCCIÓN A LAS TÉCNICAS DE APRENDIZAJE AUTOMÁTICO (PARTE II): ENTORNO DE COMPETICIÓN

Xavier Llorà, Ester Bernadó, Elisabet Golobardes, Josep M. Garrell, Maria Salamó,
Joan Camps y David Vernet

*Grupo de Investigación en Sistemas Inteligentes, Departamento de Informática,
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Passeig Bonanova 8,
08022 - Barcelona*

e-mail: {xevil, esterb, elisabet, josepmg, mariasal, joanc, dave} @salleURL.edu

Resumen: El presente artículo describe la parte práctica de la asignatura de formación en Aprendizaje Automático, en el entorno de nuestro grupo de investigación. Se estructura como una competición entre distintos grupos de trabajo, lo cual sienta las bases para motivar al alumno y establecer un marco común de comparación de distintas metodologías de Aprendizaje Automático.

1.- INTRODUCCIÓN

La asignatura de introducción a las técnicas de aprendizaje automático es un punto de unión entre investigación y docencia. Permite dar a conocer a los alumnos la investigación que desarrolla el grupo de Sistemas Inteligentes, mediante una metodología docente flexible y muy participativa.

La asignatura consiste en una parte formativa, estructurada en seminarios, y una parte eminentemente práctica cuya finalidad es estimular la curiosidad e interés del alumno. La práctica se basa en dos actividades complementarias. La primera es el estudio bibliográfico y profundización en un tema concreto (relacionado con los temas impartidos en los seminarios). En segundo lugar, los alumnos, divididos en grupos de trabajo, desarrollan un pequeño sistema inteligente aplicado sobre un problema concreto.

Este artículo describe el diseño del problema a resolver por los alumnos, el cual se ha definido sobre dos bases: aplicar las metodologías impartidas en los seminarios y motivar la participación del alumno. En la sección 2 se detallan los aspectos más relevantes de este diseño. A continuación, se presenta el problema que se ha definido así como la plataforma desarrollada. Finalmente, se muestran los resultados y conclusiones de la experiencia.

2.- DISEÑO DE LAS PRÁCTICAS

El diseño de la parte práctica de la introducción a los Sistemas Inteligentes pretende cubrir tres objetivos principales: (1) aplicación práctica de las metodologías tratadas en los seminarios, (2)

incentivar la motivación del alumno y (3) divulgar los resultados a través de una plataforma Web. A continuación se detallan cada uno de estos objetivos.

2.1.- Aplicación práctica de las metodologías tratadas

La principal función de la parte práctica es que los alumnos tomen contacto con la teoría expuesta en los seminarios. Debido a la variedad de técnicas y metodologías tratadas, hay ciertas consideraciones a tener en cuenta.

En primer lugar, la práctica debe permitir la aplicación de todas las metodologías. Recordemos que las metodologías tratadas en los seminarios son muy heterogéneas. Éstas son: redes neuronales (NN), algoritmos genéticos (GA), programación genética (GP), sistemas clasificadores (CS) y razonamiento basado en casos (CBR). Por tanto, la primera labor del alumno es analizar la aplicabilidad de dichas técnicas en el problema definido. A partir de ahí, debe escoger la metodología más apropiada según su criterio y adaptarla convenientemente.

Es interesante, a la vez, que los distintos grupos de trabajo escojan metodologías distintas para la resolución del problema. De esta forma, será posible establecer una comparativa muy didáctica entre ellas.

Otro punto muy importante a tener en cuenta es la complejidad del problema. Por una parte, el problema o entorno que se defina debe tener cierta complejidad, para justificar la aplicación de técnicas de aprendizaje automático. No obstante, debe definirse de forma acotada, acorde al perfil de los alumnos que cursan la asignatura.

2.2.- Incentivar la motivación del alumno

La aplicación práctica de las técnicas vistas en clase es, por sí misma, una motivación para el alumno, puesto que ve en profundidad una técnica concreta, aumenta su comprensión de la misma y a la vez verifica su funcionamiento. Pero, además, creemos importante incentivar su participación en las prácticas si definimos un problema que les resulte atractivo, divertido o simplemente, nuevo o distinto a los que normalmente ha realizado en otras asignaturas. En segundo lugar, podemos aumentar el interés del alumno si definimos un entorno de **competición** entre los distintos grupos. De esta forma, si los diferentes sistemas desarrollados pueden entrar en competición directa, cada grupo se esforzará en hacer de su sistema *el mejor*.

2.3.- Plataforma Web

Un objetivo colateral de esta parte práctica es la difusión del entorno de competición que se defina, así como la difusión de los resultados que se obtengan. Por lo que al entorno se refiere, su difusión en una página Web puede ser útil desde distintos puntos de vista: (1) se puede usar como entorno de test para metodologías de aprendizaje automático y (2) dar a conocer la asignatura para los futuros alumnos de la asignatura.

3.- DESCRIPCIÓN DEL PROBLEMA

El problema diseñado se basa en el aprendizaje artificial de un agente (hormiga) el cual tiene que moverse en un mapa bidimensional con el objetivo de buscar comida.

Como primera prueba piloto hemos acotado el problema de la siguiente forma. Las posiciones del mapa pueden estar en tres estados distintos: vacías, con comida o ocupadas con una hormiga. Cada hormiga tiene 8 sensores, correspondientes a las casillas colindantes. La hormiga se mueve en ciclos discretos y tiene que decidir qué movimiento realizar, en función de la percepción de los sensores. Este movimiento está limitado también a una de las ocho casillas que la rodean. El objetivo de la hormiga es comer el máximo posible durante la vida de la misma, la cual se define por un número máximo de movimientos.

Todas las hormigas participantes compiten en un mismo mapa a la vez. En principio, cada grupo compete con una única hormiga, aunque también cabe la posibilidad que cada grupo participe en la competición con más de una hormiga. De este modo, podrían definirse comportamientos de competición entre las hormigas de distintos grupos, así como de colaboración entre hormigas de un mismo participante, pudiendo o no existir comunicación explícita entre ellas. Este último comportamiento se deja al libre criterio de cada grupo.

El marco de este problema sugiere muchas variantes posibles (algunas de ellas se pueden encontrar en la bibliografía). Por ejemplo, el entorno Woods usado por Wilson [6] se caracteriza por mapas con obstáculos (además de comida) y el objetivo de la hormiga es alcanzar la comida lo más rápido posible. En este caso, no hay competición ni colaboración entre hormigas. Tan sólo hay una hormiga por mapa. Otros entornos, como el descrito por Potter [5], dos robots compiten por la comida que deben alcanzar para sobrevivir y reemplazar así la pérdida de energía debida al movimiento.

En nuestro caso, el entorno que hemos definido es suficientemente amplio para que los alumnos puedan desarrollar sistemas relativamente simples, así como sistemas mucho más complejos que incorporen por ejemplo, comportamientos cooperantes, memoria, etcétera. Además, la implementación del entorno contempla la posible modificación o ampliación del mismo, introduciendo así algunas de las variantes anteriormente comentadas.

3.1.- El trabajo de los alumnos

El trabajo realizado por los alumnos se ve reflejado en la implementación de su hormiga, que les permitirá tomar parte en la correspondiente competición. Debido a la finalidad divulgativa del trabajo a través de una plataforma Web, se han fijado ciertas restricciones.

Concretamente nos referimos a que la hormiga se ha modelado como una *interfaz* (API) la cual modela el comportamiento de la hormiga anteriormente descrito. Básicamente, los estudiantes tienen que implementar la siguiente *interfaz* Java [1,2]:

```
public interface Ant extends Remote {  
    void initialize () throws RemoteException;  
    void newMap () throws RemoteException;  
    void born () throws RemoteException;  
    int move ( char [] baEnviroment ) throws RemoteException;  
    void cellContent ( char cCell ) throws RemoteException;  
    void die () throws RemoteException;  
    void endMap () throws RemoteException;  
    void endSimulation () throws RemoteException;  
}
```

Dicha *interfaz* modela completamente el comportamiento de la hormiga en el entorno, permitiendo una integración directa del trabajo de los alumnos en la plataforma.

3.2.- Plataforma

Una vez los alumnos han diseñado e implementado su hormiga, es el momento de tomar parte en la competición. Dicha competición se realiza sobre una plataforma específicamente diseñada para dicho objetivo. El funcionamiento básico de la misma se puede resumir en las siguientes fases:

- Configuración del entorno en el cual las hormigas vivirán (*carga de mapas*)
- Incorporación de las hormigas realizadas por los alumnos
- Selección del modo de competición
- Simulación (competición propiamente dicha)

Configuración del entorno

Consiste en la carga de los mapas que se utilizarán para realizar las simulaciones. Por cada mapa se indica el número de simulaciones, el número máximo de pasos de las hormigas y el peso asociado a la complejidad del mapa.

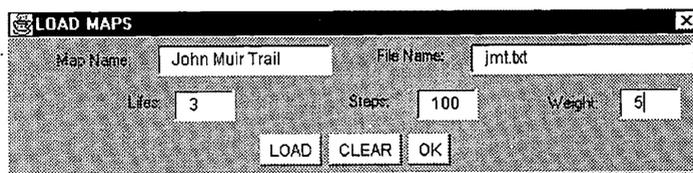


Figura 1 Carga interactiva de mapas

En la figura 1 se carga el mapa John Muir Trail [4], donde se definen tres vidas, cien pasos y un peso asociado de valor 5.

Carga de las hormigas

Cuando el entorno ya se ha configurado, se procede a la carga de las hormigas. Del conjunto de hormigas disponibles, se pueden seleccionar cuáles de ellas participan en la competición, así como el número de las mismas.

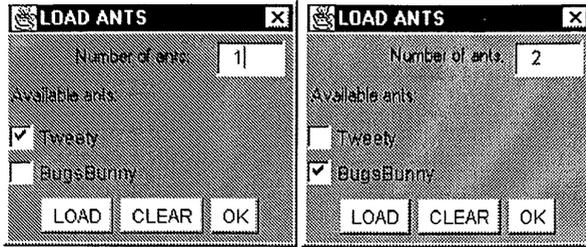


Figura 2 Carga de hormigas

En el ejemplo anterior se cargan tres hormigas: un *Tweety* y dos *BugsBunny*. Como se puede apreciar, la plataforma ofrece la máxima flexibilidad en la configuración del cuadro de participantes, permitiendo así distintos escenarios de competición.

Selección del modo de competición

Una vez seleccionadas las hormigas y el número de participantes de cada una de ellas, se pasa a la elección de la modalidad de competición. Hay dos modalidades. La primera consiste en que cada hormiga participe aisladamente, es decir en un mapa propio. De esta forma se puede apreciar el comportamiento individual de cada hormiga. La segunda modalidad ubica todas las hormigas en un único entorno –mapa–. Dicha variante busca la competencia entre hormigas contrarias –en el ejemplo anterior, entre *Tweety* y *BugsBunny*– y la cooperación entre hormigas homónimas –por ejemplo entre las dos *BugsBunny*–.

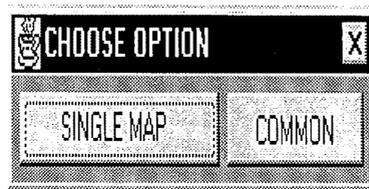


Figura 3 Selección del modo de

Simulación

Llegados a este punto la competición ya está lista para ser realizada. En la figura siguiente se presenta el aspecto de la competición.

La figura 4 ilustra el mapa seleccionado con las tres hormigas participantes, así como los gráficos de rendimiento alimenticio de cada hormiga. En la parte central de la figura se muestran los controles de la simulación.

5.- CONCLUSIONES Y LÍNEAS DE FUTURO

En este artículo se ha presentado el diseño de la parte práctica de una asignatura cuyo objetivo es introducir al alumno en el aprendizaje automático. La implementación de un sistema inteligente por parte de los alumnos permite una mayor consolidación de los conocimientos transmitidos en los seminarios. Así mismo, su estructuración en un esquema de competición estimula el interés del alumno hacia la inteligencia artificial y en particular, hacia las actividades de nuestro grupo de investigación en sistemas inteligentes.

Esta experiencia docente fue realizada por primera vez el curso 1998-1999 y durante este segundo curso estamos consolidando la plataforma de competición. Por tanto, esperamos en un futuro próximo poder divulgar la plataforma en una página Web, abriendo así la competición a toda persona interesada.

6.- AGRADECIMIENTOS

Nos gustaría agradecer a Ingeniería i Arquitectura La Salle el soporte que ofrece a la actividad del grupo de Sistemas Inteligentes. Así mismo, muchas gracias a Mireia Nadal y Vicenç Chamorro por su colaboración en el desarrollo y consolidación de la plataforma Web.

7.- BIBLIOGRAFÍA

- [1] Ken Arnold and James Gosling, *The Java Programming Language*, Addison Wesley, 1998.
- [2] Gary Cornell and Cay S. Horstmann, *Core Java Vol I Fundamentals*, Prentice-Hall, 1999.
- [3] David E. Goldberg, *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesley Publishing Company, Inc., 1989.
- [4] Steven Levy, *Artificial life: a report from the frontier where computers meet biology*, Vintage Books, 1992.
- [5] Mitchell A. Potter, Kenneth A. DeJong and John J. Grefenstette, *A Coevolutionary approach to Learning Sequential Decision Rules*, Proceedings of the sixth International Conference (ICGA95), L.J. Eshelman (eds.), San Francisco, CA, Morgan Kaufmann.
- [6] Stewart W. Wilson, *Generalization in the XCS Classifier System*, Genetic Programming 1998: Proceedings of the third annual conference, J. Koza et al. (eds.), San Francisco, CA, Morgan Kaufmann.