



Universidad
Carlos III de Madrid

Programación Automática
MÁSTER EN CIENCIA Y TECNOLOGÍA
INFORMÁTICA

Ricardo Aler Mur

Fernando Fernández Rebollo

Raquel Fuentetaja Pizan

David Quintana Montero



La evaluación de la asignatura requiere la realización de un trabajo de investigación. Se distinguen trabajos teóricos (por ejemplo, estados de la cuestión de un tema concreto), trabajos prácticos aplicados (utilización de una herramienta sobre algún problema estándar) o trabajo de investigación (requiere investigación, programación y experimentación de alguna idea con algún grado de novedad).

1. TRABAJOS TEÓRICOS:

Estado de la cuestión sobre “Behavioral cloning”.

También conocido como “*Imitation learning*”, especialmente en videojuegos o robótica. La realización de un estado de la cuestión requiere de la localización de los artículos más relevantes sobre el tema, su descripción resumida y su organización siguiendo algún criterio. Se suele realizar una crítica de los trabajos y se sacan conclusiones globales del estado de la cuestión actual. Realizar un estado de la cuestión es una de las partes de una tesis doctoral y suele ser una tarea árdua. En esta asignatura se pretende que se realice algo más sencillo, siguiendo las líneas de lo que se puede encontrar en el apartado “*Related Work*” del artículo *Programming Robosoccer agents by modeling human behavior* (<http://e-archivo.uc3m.es/handle/10016/6032>). El trabajo debe de ser más extenso y recoger investigaciones posteriores al 2004. En las transparencias sobre el tema de behavioral cloning aparecen algunos de los trabajos originales sobre behavioral cloning.

Estado de la cuestión sobre programación automática de autómatas de estado finito. Preferiblemente, técnicas que usen técnicas evolutivas (algoritmos genéticos, programación genética, ...).

Estado de la cuestión/descripción de Evolino (aprendizaje de redes de neuronas recurrentes).

Este trabajo está más centrado en una única aplicación. Sería interesante que el alumno hiciera una presentación al final. Requiere cierto conocimiento sobre redes de neuronas. La información y artículos sobre Evolino se puede encontrar en:

<http://www.idsia.ch/~juergen/evolino.html>

Estado de la cuestión/descripción de MOSES (Meta-Optimizing Semantic Evolutionary Search), una variante de programación genética basada en estimación de distribuciones.

Este trabajo está más centrado en una única aplicación. Sería interesante que el alumno hiciera una presentación al final. La información sobre MOSES se puede encontrar en:

http://wiki.opencog.org/w/Meta-Optimizing_Semantic_Evolutionary_Search

Estado de la cuestión sobre “Probabilistic Policy Reuse”.

Probabilistic Policy Reuse surgió como una técnica de *Transfer Learning*, o transferencia de conocimiento aprendido. Sin embargo, en los últimos años se ha utilizado en muchos más escenarios, como el aprendizaje por demostración, el aprendizaje desde instructores, etc. Algunos de estos trabajos son los siguientes:

- Using Spatial Hints to Improve Policy Reuse in a Reinforcement Learning Agent. Bruno da Silva and Alan Mackworth
- Help an Agent Out: Student/Teacher Learning in Sequential Decision Tasks. Lisa Torrey and Matthew E. Taylor

El objetivo es, por tanto, estudiar los trabajos más recientes que utilizan o se basan en alguno de los algoritmos de Probabilistic Policy Reuse, cuya principal descripción se encuentra en <http://dx.doi.org/10.1007/s13748-012-0026-6>.

Estado de la cuestión sobre aplicaciones de programación genética a la optimización de estrategias inversión.

La programación genética es una técnica que se ha venido empleando desde hace años para la búsqueda de estrategias de inversión en bolsa. El trabajo que impulsó inicialmente la línea fue

- Allen, F., Karjalainen, R., 1999. Using Genetic Algorithms to Find Technical Trading Rules. *Journal of Financial Economics* 51, 245-271.

pero son muchos los esfuerzos posteriores que han venido a complementarlo. El trabajo consistiría en identificar los últimos trabajos relevantes y condensar las conclusiones sobre la pertinencia de su uso en el dominio.

2. TRABAJOS PRÁCTICOS DE DIFICULTAD MEDIA

Entender y utilizar la herramienta de programación automática ADATE (Automatic Design of Algorithms Through Evolution).

Se trataría de aprender a usar la herramienta y documentar su utilización con algún ejemplo sencillo planteado por el propio alumno (es decir, que no sea un ejemplo de los que ya vienen en la propia herramienta).

ADATE debe ser ejecutado en Linux en un PC (no sirve otro tipo de CPU). Es una herramienta escrita en el lenguaje funcional ML, por lo que hay que instalar previamente un compilador de ML. La herramienta se puede encontrar en la siguiente dirección, incluyendo un manual de instalación y uso:

<http://www-ia.hiof.no/~rolando/ML/ADATE/AdateManual.pdf>

Entender y utilizar la herramienta de programación automática MOSES (Meta-Optimizing Semantic Evolutionary Search).

Se trataría de aprender a usar la herramienta y documentar su utilización con algún ejemplo sencillo planteado por el propio alumno (es decir, que no sea un ejemplo de los que ya vienen en la propia herramienta).

MOSES está escrito en C++ y se puede encontrar en:

http://wiki.opencog.org/w/Meta-Optimizing_Semantic_Evolutionary_Search

Comparación de distintas herramientas de aprendizaje relacional.

Este trabajo consiste en seleccionar algún conjunto de datos relacional y experimentar con al menos dos técnicas (o herramientas) de aprendizaje relacional para estudiar cuál es más adecuada para el problema seleccionado.

El conjunto de datos puede ser generado por el propio alumno o se puede seleccionar de algún repositorio de datos relacionales, como UCI

(<http://archive.ics.uci.edu/ml/datasets.html>). Un conjunto de datos interesante es: <http://www.doc.ic.ac.uk/~shm/Software/Datasets/mutagenesis/progol/>

Una de las opciones de experimentación es proposicionalizar los datos y utilizar la herramienta WEKA.

Posibles herramientas para aprendizaje relacional son:

- Progol <http://www.doc.ic.ac.uk/~shm/progol.html>
- Aleph <http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph>
- Relational weka http://cui.unige.ch/~woznica/rel_weka/

- ACE (TILDE) <http://dtai.cs.kuleuven.be/ACE/>
- Treeliker <http://ida.felk.cvut.cz/treeliker/Treeliker.html>
- SUBDUE (graph-based relational learning) <http://www.subdue.org>

Uso de técnicas de aprendizaje relacional para aprender a construir el pasado de verbos en inglés de forma automática .

Este trabajo consiste en estudiar distintas formas de modelizar el problema y distintas aproximaciones para generar un software que aprenda de forma automática a construir el pasado de verbos en inglés. El trabajo de partida sería <http://www.cs.utexas.edu/~ai-lab/pub-view.php?PubID=51430>

3. TRABAJOS PRÁCTICOS

Utilización de la herramienta PIPE (se ve en clase) para resolver problemas de clasificación.

Es decir, en lugar de evolucionar programas, PIPE evolucionará clasificadores. Será necesario implementar una función de fitness en C, que calcule el porcentaje de aciertos del clasificador representado por el árbol de PIPE. Es necesario que el alumno tenga cierta familiaridad con alguna de las técnicas de clasificación típicas de aprendizaje automático (redes de neuronas, árboles de decisión, vecino más cercano, ...). PIPE está escrito en C y corre sobre Linux. La información sobre PIPE 1.0 se puede encontrar en:

ftp://ftp.idsia.ch/pub/rafal/PIPE_v1.0.tar.gz

<ftp://ftp.idsia.ch/pub/rafal/>

<http://citeseer.ist.psu.edu/cache/papers/cs/534/ftp:zSzzSzftp.idsia.chzSzpubzSzrafalzSzPIPE.pdf/salustowicz97probabilistic.pdf>

Utilización de técnicas de aprendizaje por refuerzo y reutilización de políticas para resolver el juego Mario Bros

El juego de Mario Bros. se ha establecido como un dominio de aplicación de técnicas de inteligencia artificial en general, así como de aprendizaje por refuerzo. Hoy en día, existe una competición cuyo objetivo es el desarrollo de jugadores con técnicas de IA (<http://www.marioai.org/>). El objetivo de este proyecto es desarrollar un agente Mario basándose en la aplicación de técnicas de reutilización de políticas (Probabilistic Policy Reuse:

<http://dx.doi.org/10.1007/s13748-012-0026-6>) para aprendizaje por demostración en aprendizaje por refuerzo.

Análisis de viabilidad para el uso de herramientas alternativas a TILDE para ROLLER.

Roller (<http://www.jair.org/papers/paper3231.html>) es un sistema de aprendizaje que genera árboles relacionales que posteriormente se utilizan para guiar a un planificador automático. Actualmente Roller utiliza la herramienta TILDE (<http://dtai.cs.kuleuven.be/ACE/>) para el aprendizaje de los árboles de decisión. El objetivo de este trabajo es estudiar desde un punto de vista práctico si es viable reemplazar TILDE por una herramienta de ILP como Aleph (<http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph>)

Evaluar el rendimiento de la programación genética en un caso de regresión simbólica.

El trabajo que se propone consiste en evaluar la programación genética en un caso de regresión simbólica estudiando aspectos como

- Sensibilidad de la calidad de las soluciones sobre entrenamiento y test, ante diferencias en el tamaño de la población y el número de generaciones.
- Influencia del número de generaciones sobre el tamaño promedio de los individuos de la población.
- Evolución del coste computacional con el tamaño de la población y el número de iteraciones.
- Sensibilidad de la solución a la ausencia de alguna de funciones necesarias para modelar la función objetivo.