Métodos de aprendizaje por refuerzo aplicado a un escenario de e-learning

J. M. Marquez[1], F. Velasco[2], L. Gonzalez-Abril[2], J. A. Ortega[3], C.Chamizo[2]

[1] Telvent Global Services, R. Tamarguillo 29, Sevilla. jose.marquez@telvent.com

[2] Dep. de Economía Aplicada, Univ. de Sevilla. {velasco,luisgon}@us.es

[3] Dep. de Lenguajes y Sistemas Informáticos, Univ. de Sevilla, ortega@lsi.us.es

Abstract

En este artículo se presentan dos métodos de aprendizaje por refuerzo, y su posible aplicación un escenario e-learning. La principal diferencia con el tradicional esquema de aprendizaje por refuerzo que se describe en la introducción, es que en el escenario que se plantea en este trabajo el agente no es único, sino que múltiples agentes comparten la información de estado proporcionada por el sistema. Además estos agentes no son autómatas, sino que se corresponden con las acciones de los usuarios. Finalmente se propone una nueva aproximación basada en las dos técnicas presentadas en el trabajo.

1 Introducción

Habitualmente los sistemas de enseñanza se basan en modelos formales que contienen información acerca del perfil y mecanismos de aprendizaje del estudiante (modelo del estudiante), de la materia de estudio (modelo del dominio), de las estrategias pedagógicas a seguir (modelo pedagógico) y de la interacción entre los participantes (modelo del interfaz). Los sistemas de educación inteligentes utilizan estos modelos para representar internamente el entorno y obtener la información necesaria para emular el comportamiento de un profesor, con la ventaja de que la interacción se puede reducir a un único alumno.

La definición y desarrollo de estos modelos ha sido la base para la posterior aplicación de técnicas de Inteligencia Artificial en la enseñanza on-line.

Diferentes estudios previos se han centrado en la definición del modelo del dominio y su relación con el modelo pedagógico, basándose en técnicas tan variadas como *reglas* LISP [Anderson y Reiser, 1985], *técnicas de planificación* [André et al., 1993], *redes bayesianas* [VanLehn y Zhendong, 2001], sistemas que mezclan *reglas* con *redes neuronales* [Prentzas et al., 2002] o incluso *grafos conceptuales* [Bontcheva y Dimitrova, 2004].

Sin embargo, motivada por la definición de una especificación abierta que asegure la compatibilidad de cursos para diferentes plataformas, otras especificaciones han simplificado este modelo y su relación con el modelo pedagógico y se han impuesto al resto. Algunas de estas especificaciones como IMS-CP (para el agrupamiento de recursos educativos en paquetes) o IMS-SS (para la secuenciación de los recursos de un mismo paquete) se han convertido en estándares *de facto* y han proporcionado la base para otras más ambiciosas como SCORM, la especificación dominante actualmente en la industria de la enseñanza on-line.

SCORM es una especificación centrada en los paquetes de contenidos: cómo describirlos, como reproducirlos, como secuenciar sus contenidos internos, cómo evaluarlo y coordinar todas estas tareas con un entorno gestor del aprendizaje (LMS). Por tanto se ocupa no sólo del modelo del dominio sino también, parcialmente, del modelo pedagógico, al permitir definir la secuencia de contenidos de un paquete educativo. El secuenciamiento de contenidos es fijo, por lo que constituye un aprendizaje guiado, facilitando metodologías conductistas frente a constructivistas.

En el modelo de interfaz diferentes trabajos de investigación se han ocupado de adaptar la interacción a las necesidades de los usuarios, aplicando técnicas propias de la Inteligencia Artificial para facilitar búsquedas en el modelo del dominio [Brusilovsky et al., 1996] o conseguir interfaces guiadas por voz mediante el procesamiento del lenguaje natural [Aleven y Koedinger, 2000].

Continuamente se ha buscado aumentar el grado de adaptabilidad de un sistema inteligente a las necesidades de un usuario, tanto desde el punto de vista de interacción (modelo de interfaz) como desde el punto de vista del proceso de enseñanza-aprendizaje (modelo pedagógico). En este sentido el modelado del estudiante juega un papel crucial y han sido muy numerosas las estrategias seguidas, en función de como se entienda el proceso de aprendizaje: inducción de conocimiento, compilación de conocimientos existentes, aprendizaje supervisado o no supervisado, aislado o en grupo, etc.

Algunos de los enfoques más interesantes últimamente utilizan *redes bayesianas* [VanLehn y Zhendong, 2001] y el *aprendizaje por refuerzo* [Beck, 2001] para representar el conocimiento del estudiante y el impacto de sus acciones en el proceso de aprendizaje tanto del propio alumno como de sus iguales.

Actualmente el objetivo se centra no sólo en adaptar la presentación o los contenidos al perfil del estudiante, sino también la secuenciación de los mismos teniendo en cuenta no sólo el perfil del estudiante sino su rendimiento pasado y

el perfil del resto de estudiantes del mismo nivel participantes en el mismo curso.

En cambio, sólo una pequeña parte de los sistemas de enseñanza on-line se adaptan de forma inteligente a los estudiantes. La mayoría de estos sistemas adaptativos, lo hacen usando un conjunto de reglas pedagógicas del tipo *if-then*. Pero la diversidad de perfiles de estudiantes, hace que sea necesario codificar un conjunto muy grade de reglas, lo que hace muy costoso estos sistemas.

La aplicación de modelos de aprendizaje por refuerzo podría reducir el coste de estos sistemas pues se evitaría la definición de reglas para cada estudiante y cada situación.

2. Descripción del problema

Se pretende con este trabajo explicar como es posible aplicar métodos de aprendizaje por refuerzo para conseguir que sistemas de enseñanza on-line puedan ser adaptables a las necesidades pedagógicas de los alumnos, facilitando el itinerario formativo más adecuado (insertando más o menos cursos de refuerzo, seleccionando cursos con enfoques más prácticos o más teóricos según convenga, etc).

Algunos autores han afrontado el problema desde el punto de vista del modelo del estudiante [Beck, 2001], definiendo el estado del sistema
como todas las características de aprendizaje del estudiante, incluyendo el
conocimiento que tiene el estudiante sobre los temas principales descritos
en el módulo del dominio entre otras muchas características del usuario. Beck intenta adaptar el sistema a cada uno de ellos dependiendo de todas sus características
de aprendizaje
. El inconveniente es que estas características son muy numerosas, por lo que el sistema aprende muy lentamente pues en este tipo de sistemas depende de la interacción con el estudiante. Otros autores [Rey-López et al., 2006] separan el modelo del estudiante del modelo pedagógico, para dotar al sistema de la capacidad de personalizar el contenido en función de las preferencias y/o necesidades del estudiante pero sin afectar a la integración con los estándares de la industria.

Desde nuestro punto de vista, creemos que lo más indicado sería aplicar el aprendizaje por refuerzo, en lugar de al modelo de características del estudiante al modelo pedagógico, lo que reduciría el número de variables y simplificaría el problema, y aplicar reglas para la personalización al modelo del estudiante.

El modelo pedagógico puede representarse de diversas formas. En [Márquez et al., 2008] se utilizan Modelos de Características [Czarnecki et al., 2004] para ello, pero pueden utilizarse otras formas de representación (matricial, por ejemplo). Para ello puede utilizarse cualquier representación que facilite el desarrollo de una herramienta CASE para modelado visual de itinerarios pedagógicos de aprendizaje. En este sentido no hay muchos ejemplos en la literatura al respecto.

Algunos problemas de optimización combinatoria pueden ser eficientemente resueltos con algoritmos de aprendizaje por refuerzo, como por ejemplo encontrar el camino Hamiltoniano más corto en un grafo ponderado completo (también conocido como el *problema de los viajantes*) [Gambardella y Dorigo, 1995]. Estos problemas pueden representarse como un grafo en el que los nodos representan los estados y las aristas las transiciones entre dos estados. Igual para caminos no Hamiltonianos, como por ejemplo encontrar el camino más corto entre dos nodos. Este será el enfoque que adoptemos para el escenario que presentamos a continuación y que describe claramente el problema.

2.1 Escenario

Supongamos que una empresa quiere preparar a sus empleados para adaptarlos a un puesto diferente, de mayor responsabilidad, que queda vacante. Esta vacante requiere una serie de competencias que pueden ser adquiridas por los candidatos mediante la realización de algunos cursos específicos. Estos cursos pueden ser organizados (secuenciados) mediante un grafo de describa las transiciones entre cursos, definiendo así todos los caminos posibles del itinerario formativo necesario para adquirir las competencias requeridas. Teniendo en cuenta las necesidades de cada candidato individualmente, es posible personalizar el proceso de aprendizaje (el camino en el grafo) seguido por cada uno ellos al ritmo que mejor se adecue a las características de cada uno, maximizando la probabilidad de éxito y el nivel de adquisición de los conocimientos y habilidades requeridas para el puesto vacante, objetivo esencial de la formación.

El grafo dirigido de la Figura 1 modela un conjunto de itinerarios formativos encaminados a conseguir las competencias necesarias.

Figura 1: Grafo de itinerarios formativos por competencias

En el grafo de la figura se pueden diferenciar los siguientes elementos:

- **Nodos**: Representan los cursos. El nodo 1, es el único cuyo grado de entrada es 0, y representa el inicio del itinerario. Los nodos 5, 8, 10 y 12 tienen grado de salida 0 y representan los posibles estados finales. Sólo uno de ellos será el estado final para cada estudiante, por lo que como mínimo habrá 4 caminos diferentes.
- **Aristas**: Representan las transiciones de un curso a otro. Se produce una transición cuando el alumno ha sido evaluado de los contenidos del nodo.
- **Arcos**: Representan condiciones de recorrido. Los arcos OR (+) indican que las transiciones que salen de un nodo son excluyentes y que una vez evaluado el alumno, se deberá escoger una de las posibles de entre todas las aristas salientes del nodo que están rodeadas por el arco. Los arcos AND (&) representan obligatoriedad e indican que todos las transiciones rodeadas por el arco han de recorrerse. Para simplificar, el recorrido se realiza en profundidad y en el caso de condiciones AND se evalúan de izda a derecha, por lo que una condición AND que rodee a N aristas da lugar a N sub-itinerarios o ramas obligatorias. Normalmente cada una de estas ramas va asociada a la consecución de una competencia, mientras que las condiciones OR son útiles para definir caminos alternativos (itinerarios de refuerzo, de profundización, etc).

Una vez procesadas las condiciones, el grafo resultante equivalente al de la figura 1 es el representado en la figura 2:

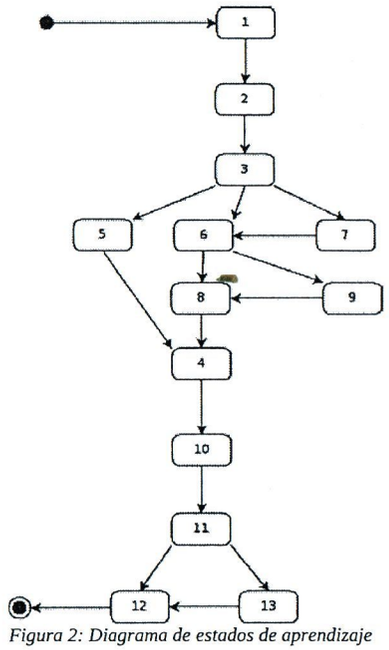

Figura 2: Diagrama de estados de aprendizaje

El objetivo es predecir el mejor camino en el grafo, desde el nodo raíz al nodo terminal recorriendo todas las ramas obligatorias (no todos los nodos), de entre todos los posibles para cada alumno, teniendo en cuenta su evolución (calificaciones durante las evaluaciones en cada curso) así como las del resto de alumnos de su mismo nivel que están realizando el mismo itinerario.

3. Aprendizaje por refuerzo

El objetivo del aprendizaje por refuerzo es maximizar la recompensa obtenida tras cada acción para aprender una determinada tarea, la cual permitirá tomar decisiones en el futuro de qué acción tomar a partir de una percepción del entorno. El aprendiz o tomador de las decisiones es denominado el *agente*. Las cosas con las que interactúa, comprendiendo todo aquello externo al agente, se denomina *entorno*. La interacción es continua, el agente seleccionando *acciones* y el entorno respondiendo a las mismas, pudiendo sus respuestas presentar nuevas situaciones al agente. En el esquema básico de aprendizaje por refuerzo, el entorno responde a las acciones del agente con una recompensa que será mayor o menor, positiva o negativa, dependiendo de la acción llevada a cabo por el agente.

Este método de aprendizaje tiene su raíz en una rama de la psicología experimental, que pueden remontarse a las experiencias de Pávlov [Pávlov, 1927] con el refuerzo condicionado, y es heredero por otro lado de los métodos de optimización que se originan a partir de los trabajos de Bellman [Bellman, 1957]. Dicho de forma breve, el aprendizaje por refuerzo es el problema de conseguir que un agente actúe en un entorno de manera que maximice la recompensa que obtiene por sus acciones. Este tipo de aprendizaje se encuadra en los denominados Aprendizaje Supervisados.

De acuerdo con la Figura 3, el agente y el entorno interactúan entre sí en una secuencia de pasos discretos, t=0,1,2,3,...n. En cada paso t, el agente recibe alguna representación del estado del entorno, s_t ∈ S, con S el conjunto de estados posibles. En base a ea información el agente selecciona una acción, a_t ∈ $A(s_t)$, dónde $A(s_t)$ es el conjunto de acciones disponibles para el estado s_t. En el instante o paso siguiente, en parte como consecuencia de su acción, el agente recibe una recompensa, r_{t+1} ∈ R, y busca por sí mismo un nuevo estado, s_{t+1}.

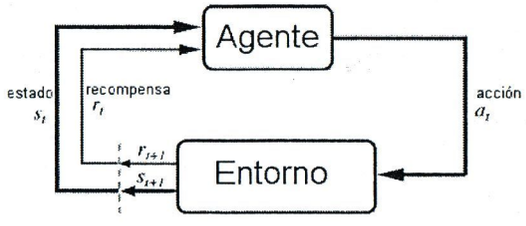

Figura 3: Modelo de interacción entre agente y entorno en el aprendizaje por refuerzo

En cada paso, el agente implementa un mapeo entre estados y probabilidades de seleccionar cada posible acción. A este mapeo se le conoce como la política del agente y es generalmente representado como π_t, dónde $\pi_t(s,a)$ es la probabilidad de que $a_t = a$ si $s_t = s$. Los métodos de aprendizaje por refuerzo especifican como el agente cambia su política como resultado de su experiencia. La señal de refuerzo puede ser inmediata o retardada. Inmediata es cuando se obtiene una crítica para cada acción efectuada justo después de su realización. La información aportada por el refuerzo en este caso es local a cada acción tomada. Por el contrario, en el caso del refuerzo retardado se dará cuando éste no se obtiene inmediatamente después de la realización de cada acción, sino al completar una secuencia de acciones empleadas para resolver el problema. En este caso, el refuerzo obtenido es una estimación global del comportamiento.

Si no importa qué acciones se hayan llevado a cabo para alcanzar el estado actual, pero éste es suficiente para determinar cuáles pueden ser las acciones futuras, decimos que tenemos un conjunto markoviano de señales de estado, puesto que estas señales poseen la propiedad de Markov [Bellman, 1957; Howard, 1960; Puterman, 1994]:

$$\Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t\} =$$

$$\Pr\{s_{t+1} = s', r_{t+1} = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r_1, s_0, a_0\} \quad (1)$$

Según dicha ecuación, la probabilidad de que el agente se encuentre en un estado y que reciba un determinado refuerzo sólo depende del estado en el que se encontraba y la acción ejecutada en el instante inmediatamente anterior.

Se denomina proceso de decisión de Markov a todo problema de aprendizaje que satisface esta propiedad. Además, si el número de estados y acciones definidos en el problema son finitos, se le denomina Problema de Decisión de Markov finito.

La mayoría de los algoritmos de aprendizaje por refuerzo se basan en funciones que estiman lo bueno que es encontrarse en un determinado estado y lo bueno que es ejecutar una acción desde ese estado. La función de *valor-estado*, $V^{\pi}(s)$ es el refuerzo que se espera obtener si el agente se deja guiar por una política de acción π, desde el estado s, hasta el infinito, donde γ es el parámetro de descuento de las futuras acciones.

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\} \quad (2)$$

donde Eπ{} denota el valor esperado si el agente sigue la política π.

Otro tipo de funciones para este tipo de algoritmos es la función valor-acción, $Q^{\pi}(s,a)$, que estima el refuerzo esperado si el agente se deja guiar por una política π, desde el estado s hasta el infinito, comenzando por ejecutar la acción a.

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} =$$

$$= E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right\} \quad (3)$$

Estas funciones de valor nos proporcionan un buen método de comparación de
calidad de políticas. Por ejemplo, se podría decir que la política más adecuada
es la que proporciona una ganancia mayor, tal y como se define en la ecuación (4).

$$\pi > \pi', \text{ si y sólo si, } V^{\pi}(s) > V^{\pi'}, \forall s \in S \quad (4)$$

siendo la política óptima, π*, aquella que proporciona la mejor función de valor estado (5)

$$V^*(s) = max_{\pi} V^{\pi}(s), \forall s \in S \quad (5)$$

siendo, de forma equivalente, la función de valor-acción óptima Q*(s,a):

$$Q^*(s,a) = max_{\pi} Q^{\pi}(s,a), \forall s \in S, \forall a \in A \quad (6)$$

La política óptima consiste en el conjunto de acciones que en cada caso maximiza la función (6).

3.1 Q-learning

Uno de los principales avances en aprendizaje por refuerzo fue el desarrollo del algoritmo Q-learning [Watkins, 1989] utilizando técnicas de aprendizaje por diferencia temporal (TD: Temporal-Difference learning).

En entornos deterministas, el algoritmo de Q-learning se basa en la actualización iterativa de la función Q(s,a).

$$Q(s,a) = R(s,a) + \gamma max_{a'} Q(s',a') \quad (7)$$

En entornos no deterministas, no es posible aplicar la función anterior que ha de ser redefinida para tener en cuenta las probabilidades de cambiar o no de estado, llevando al agente a estados distintos con una probabilidad α :

$$Q(s_t,a_t) = Q(s_t,a_t) +$$

$$\alpha\left[r_{t+1} + \gamma max Q(s_{t+1},a) - Q(s_t,a_t)\right] \quad (8)$$

En este caso la función recursiva de estado-acción, Q, facilita la implementación de algoritmos que aproximan Q de forma iterativa, como por ejemplo el algoritmo Q-learning.

La función (8) actualiza un par estado-acción, por lo que el primer nodo, la raíz, debe ser un nodo pequeño, de acción, lo que se ajusta perfectamente al nodo raíz del grafo de la figura 1. En esta función, γ es una constante tal que $0 \leq \gamma < 1$, siendo α el parámetro de aprendizaje del algoritmo, con valores igualmente comprendidos entre 0 y 1.

Una de las principales características de este algoritmo es que no necesita que el agente ejecute secuencias óptimas para converger a una política optima [Mitchell, 1997], pudiendo aprender la función Q ejecutando en cada paso acciones elegidas de forma aleatoria, siempre que los pares estado-acción sean visitados un número suficiente de veces.

Algoritmo Q-learning

Inicializar Q(s,a) arbitrariamente
Repetir (para cada episodio):
  Inicializar s
  Repetir (para cada paso del episodio):
    Seleccionar una acción *a* a partir de *s* usando una política π derivada de Q
    Ejecutar la acción *a* observando el refuerzo inmediato recibido, *r*, y el siguiente estado, s
    Actualizar Q(s,a) aplicando (8)
    s ← s';
  hasta llegar a un estado terminal

De forma similar a como se ha aplicado Q-learning para hallar el camino más corto.

Uno de los principales inconvenientes del método Q-learning es que para converger, es necesario visitar todos los nodos un número suficiente de veces, lo que en algunos dominios no puede garantizarse. Este aspecto en cambio no afecta negativamente al mecanismo de optimización por colonia de hormigas que puede adaptarse dinámicamente a los cambios de preferencias de los usuarios.

3.2 Optimización por colonia de hormigas

El método de optimización por colonias de hormigas [Dorigo, 1992] se utiliza para solucionar el problema del cálculo del camino más corto mediante la cooperación entre los usuarios del sistema, que se encargan de su retroalimentación.

En nuestro sistema, un alumno representaría una hormiga moviéndose libremente entre los nodos del árbol de la Figura 2, liberando una cantidad variable de feromonas cada vez que supera un curso y se dispone a seleccionar el siguiente. La función que determina la cantidad de feromonas a depositar es una representación proporcional del éxito obtenido por el alumno en el curso recién evaluado, siendo por tanto esta cantidad de feromonas una medida del éxito obtenido. Así, si tenemos en cuenta una escala de calificación de 0 a 10, el equipo pedagógico puede determinar como aceptable para cada curso una calificación determinada *m*, con $0<m<10$, y definir la cantidad de feromonas a depositar en el camino seguido hasta este curso en función de *m* como:

$$\Phi_{ij}(\varepsilon,m) = \begin{cases} -(1-\varepsilon)\varphi - \omega_3\left(\frac{m}{M} - \varepsilon\right), si \varepsilon < \frac{m}{M} \\ \varepsilon\varphi + \omega_3\left(\varepsilon - \frac{m}{M}\right), si \varepsilon \geq \frac{m}{M} \end{cases} \quad (9)$$

dónde φ es una constante que representa el valor unitario de feromonas y ε es la calificación obtenida por el alumno para los contenidos del último nodo superado. El rango de esta calificación se traslada a [0,1] dónde 0 representa la mínima calificación posible y 1 la máxima. Para potenciar el efecto de las últimas feromonas depositadas sobre el camino seguido por dicho alumno, las tres últimas aristas seguidas por este también verán incrementada la cantidad de feromonas existente en ellas, recibiendo la penúltima la mitad de feromonas que la última, la antepenúltima un tercio y la anterior una cuarta parte, tal y como otros autores proponen en este tipo de sistemas [Gutiérrez et al., 2007]. ω_3 es una constante de calibración del sistema que ayudará a estudiar y controlar el impacto de la calificación mínima requerida en la cantidad de feromonas depositadas. Por tanto, la cantidad de feromonas de una arista depende de la calificación que cada alumno haya obtenido en el curso origen de esa arista y puede influir positivamente o negativamente. Evidentemente, si ningún alumno a escogido esa arista su cantidad de feromonas será nula. En un nodo con grado de salida $g_{out} > 1$, al haber más de una posibilidad de nodo destino, la elección del curso (nodo) siguiente podría no ser la acertada. En ese caso (evaluación inferior a la deseada) el itinerario seguido (tres últimas aristas seguidas por el alumno) se vería afectado por un depósito negativo de feromonas, lo que ayudaría al sistema a asignar otra de las aristas posibles para futuros alumnos.

La cantidad de feromonas puede ser calculada teniendo en cuenta la calificación obtenida en la evaluación (ε) del nodo i, la mínima calificación deseada (m) y el tiempo (t) tenido en cuenta como iteraciones puntuales o acciones (la acción de superar un curso sería la unidad de medida). Teniendo en cuenta esto, la cantidad de feromonas a depositar en función del tiempo puede expresarse como:

$$\Phi_{ii}^{t}(\varepsilon,m) = \rho\Phi_{ii}^{t-1} + \lambda(i,j)\Phi_{ii}(\varepsilon,m) \quad (10)$$

En la fórmula anterior, ρ es la tasa de evaporación de las feromonas, que afecta a la cantidad de feromonas existentes en la arista ij antes de calcular su nuevo valor en el instante t. Para evitar un impacto desmedido de las feromonas en la toma de decisiones del sistema, las feromonas, al igual que en la vida real, se evaporarán parcialmente con el paso del tiempo, perdiendo así intensidad y por tanto minimizando su impacto en el sistema. Puesto que los usuarios del sistema (los alumnos) pueden cerrar su sesión y abandonar el curso cuando deseen y reanudarlo cuando deseen desde el mismo punto en que lo dejaron, el tiempo, debe entenderse tal y como hemos comentado anteriormente como una sucesión de acciones o eventos que se producen al finalizar un curso. La tasa de evaporación es simplemente un factor positivo y menor que uno, que se aplicará a todas las aristas cada vez que un alumno seleccione un nuevo curso, disminuyendo el número de feromonas depositadas en ellas. Un valor de $\rho=0,9$ es una tasa de evaporación típica. La función lambda devuelve 1 si el nodo fue el último nodo evaluado por el usuario y cero en caso contrario y sirve para asegurar

que un usuario sólo deposita feromonas directamente[1] en el último curso realizado.

Al igual que en el problema de los viajantes el diagrama de estados de la figura 2 puede ponderarse. En este caso el peso dado a cada arista representaría la complejidad pedagógica de la transición de un curso a otro, la importancia del curso destino, el esfuerzo o número de horas en relación con el total estimado del curso. Estos pesos son valores estáticos muy útiles para definir la función de ajuste que proporcionará una medida representativa de la idoneidad de la arista *ij* para el alumno, según sus necesidades.

$$f_{ii} = \omega_1 \cdot d_{ii} \cdot P_{ii} + \omega_2 \cdot \Phi_{ii} \quad (11)$$

Esta función será usada por el sistema para decidir el siguiente curso (nodo del árbol) que el alumno seguirá, construyendo dinámicamente y en tiempo de ejecución el camino de itinerario de aprendizaje del alumno.

En esta función de ajuste d_{ij} representa la distancia pedagógica asociada a la arista ij. La distancia pedagógica es un indicador de la dificultad de aprender los conocimientos del curso j partiendo de los conocimientos adquiridos en el curso i y anteriores. Es decir, mide la separación entre la zona de desarrollo real y la de desarrollo potencial. Es decir, la zona de desarrollo próximo del alumno. Esta distancia puede expresarse en número de conceptos del curso j, por ejemplo. Po otra parte, Φ_{ij} es la cantidad de feromonas depositadas en la arista ij, y ω_1 y ω_2 son constantes de calibración del sistema. P_{ij} es el factor de idoneidad de la arista ij, calculado *a priori* a partir de los resultados previos, actuando como multiplicador del peso pedagógico.

En [Márquez et al., 2008] se explica como aplicar Redes Bayesianas al cálculo del factor de idoneidad, con lo que se consigue que el sistema se adapte en parte a las necesidades pedagógicas de cada usuario (tomando como estimador únicamente las calificaciones obtenidas en cursos anteriores).

3.3 Q-Ant

En este trabajo proponemos la integración del método de optimización por colonias de hormigas, como mecanismo del cálculo de la función de refuerzo R(s,a) del algoritmo Q-learning, lo que automatizaría el proceso de selección del mejor camino (arista) en un nodo con grado de salida $g_{out} > 0$. Hemos llamado a este enfoque Q-Ant.

El grafo de la figura 2 puede representarse también en forma matricial, indicando la distancia pedagógica entre cada par de cursos. Una distancia negativa, $d_{ij} = -1$, indicaría que el nodo j no es alcanzable desde el nodo i. Igualmente el factor de idoneidad Pij para dos nodos no adyacentes o adyacentes pero no navegables en el sentido i-j, sería 0, al igual que el número de feromonas depositadas en la arista ij (no existente). Por tanto en estos casos el valor de la función de ajuste (11) sería siempre cero.

[1] Recuérdese que el sistema se encarga de depositar otras cantidades menores en los tres nodos anteriores para reforzar el camino. Estas cantidades son proporcionales a la calculada.

La aplicación del algoritmo Q-learning se llevaría a cabo para escoger la arista más adecuada para el alumno, que sería aquella en la que se maximizara la función estado-acción.

$$Q(s,a) = Q(s,a) + \alpha R \quad (12)$$

dónde $R = \alpha[f(a) + \gamma \max Q(s',a') - Q(s,a)]$

siendo f(a) la cantidad de feromonas depositadas en la arista asociada a la acción a. La acción será simplemente escoger una u otra arista, lo que finalmente determinará el número de feromonas a tener en cuenta.

Aplicando el algoritmo Q-learning podría decidirse qué arista es la más adecuada para cada alumno, teniendo en cuenta que la función de ajuste f(a) está influenciada tanto por la distancia pedagógica entre dos cursos como por el factor de idoneidad de la transición para el alumno.

4 Conclusiones y trabajo futuro

Este trabajo presenta la aplicación de dos métodos inteligencia artificial tradicionalmente aplicados al razonamiento automático y la robótica, al motor de inteligencia artificial de un sistema de enseñanza online con el objetivo de predecir el mejor itinerario formativo para un alumno.

Se propone la integración de Q-learning y ACO (Ant Colony Optimization) para predecir el mejor camino en el itinerario de aprendizaje. Se hace referencia a un trabajo anterior que explica la utilización de Redes Bayesianas para el cálculo del factor de idoneidad de cada transición entre cursos, para dotar al sistema de la capacidad de personalización del itinerario, algo que no es muy común en los sistemas de enseñanza on-line actuales.

Como trabajo futuro resta estudiar la eficiencia del algoritmo propuesto en cuanto a tiempo de respuesta cuando el número de posibilidades es amplia, así como el número de alumnos. Igualmente es necesario investigar el ritmo de aprendizaje de Q-ant con respecto a otros algoritmos similares.

Referencias

[Aleven y Koedinger, 2000] V. Aleven and K.R. Koedinger. The need for tutorial dialog to support self-explanation. In Building Dialogue Systems for Tutorial Applications, Papers of the 2000 AAAI Fall Symposium, pages 65–73, 2000.

[Anderson y Reiser, 1985] J. Anderson and B. Reiser. The lisp tutor. In Byte, volume 10:4, pages 159–175, 1985.

[André et al., 1993] E. André , W. Finkler, W. Graf, A. Schauder, and W. Wahister. Intelligent Multimedia Presentations. WIP: The Automatic Synthesis of Multimodal Presentations. Mark T. Maybury (ed.), MIT Press, 1993.

[Beck, 2001] J. Beck. ADVISOR: A machine learning architecture for intelligent tutor construction. PhD thesis, University of Massachusetts Amherst, 2001.

[Bellman, 1957] Richard E. Bellman. Dynamic Programming. 1957. Princeton University Press, Princeton, NJ. Reeditado en 2003.

[Bontcheva y Dimitrova, 2004] K. Bontcheva and V. Dimitrova. Examining the Use of Conceptual Graphs in Adaptive Web-Based Systems that Aid Terminology Learning. In Proceedings of International Journal on Artificial Intelligence Tools (IJAIT, 2004)

[Brusilovsky et al., 1996] P. Brusilovsky, E. Schwarz, and G. Weber. Elm-art: An intelligent tutoring system on world wide web. In Claude Frasson, Gilles Gauthier, and Alan Lesgold, editors, Intelligent Tutoring Systems, volume 1086, pages 261–269. Springer, 1996.

[Czarnecki et al., 2004] Krzysztof Czarnecki, Simon Helsen y Ulrich Eisenecker. Staged Configuration Using Feature Models. *Software Product Lines, 2004*, 266-283.

[Dorigo, 1992] M. Dorigo. Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italie, 1992.

[Gambardella y Dorigo, 1995] L. M. Gamardella y M. Dorigo. Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. Proceedings of ML-95, Twelfth International Conference on Machine Learning, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, 1995, pp. 252–260.

[Gutiérrez et al., 2007] S. Gutiérrez, G. Valigiani, Y. Jamont, P. Collet, C. Delgado Kloos. A Swarm Approach for Automatic Auditing of Pedagogical Planning. *In Proceedings of Seventh IEEE International Conference on Advanced Learning Technologies*, 2007. ICALT 2007. 18-20 July 2007 Page(s):136 – 138

[Howard, 1960] R. A. Howard. Dynamic Programming and Markov Processes. Cambridge: MIT Press.

[Márquez et al., 2008] Modelado de características para itinerarios formativos adaptativos. J. M. Márquez, C. Cetina, F. Velasco, L. Gonzalez-Abril, J. A. Ortega. X Jornadas de ARCA. Sistemas Cualitativos y Diagnosis, Robótica, Sistemas Domóticos y Computación Ubicua. JARCA 2008. Juan Antonio Ortega y Natividad Martínez Madrid (Editores). Pags 33-39.

[Mitchell, 1997] Tom Mitchell. Machine Learning. McGraw Hill, 1997.

[Pávlov, 1927] Iván Pávlov. Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex. *Traducido y editado por G. V. Anrep*. Londres. Oxford University Press. http://psychclassics.yorku.ca/Pavlov/

[Prentzas et al., 2002] J. Prentzas, I. Hatzilygeroudis, and J. Garofalakis. A web-based intelligent tutoring system using hybrid rules as its representational basis. Stefano A. Cerri, Guy Gouarderes, and Fábio Paraguacu, editors, Proceedings of the 6th International Conference, ITS 2002, volume 1, pages 119–128. Lecture Notes in Computer Science. Springer Verlag, 2002.

[Puterman, 1964] M. L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. New York. Wiley.

[Rey-López et al., 2006] Marta Rey-López, Ana Fernández-Vilas, Rebeca Díaz-Redondo y José Pazos-Arias. Providing SCORM with adaptivity. *In Proceedings of the 15th International Conference on World Wide Web* (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 981-982

[VanLehn y Zhendong, 2001] K. VanLehn and N. Zhendong. Bayesian student modelling, user interfaces and feedback: a sensitivity analysis. International Journal of Artificial Intelligent in Education, 2:155–184, 2001.

[Watkins, 1989] C. J. C. H. Watkins. Learning from Delayed Rewards. PhD thesis, King's College, Cambridge, UK, 1989