

Sistema de recomendación basado en k-nn para condiciones de incertidumbre en un Sistema Tutor Inteligente

Enrique José Altuna Castillo
Lisandra Guibert Estrada
Vivian Estrada

El desarrollo de Sistemas Tutores Inteligentes tiene como objetivo crear herramientas que enseñan un dominio de conocimiento en forma similar a como lo haría un tutor humano experimentado. Para lograrlo el sistema debe ser capaz de modelar el estado de conocimiento y sugerir los materiales e intervenciones más adecuadas para cada estudiante. Sin embargo, el conocimiento asociado a la forma en que aprenden las personas, las vías más eficaces para transmitir el conocimiento y medirlo es incompleto y presenta errores. Por estas razones al modelar el estudiante se debe tener en cuenta la incertidumbre inherente al proceso. En el presente trabajo se propone una técnica para el desarrollo de un módulo del estudiante con el uso de redes bayesianas y filtrado colaborativo, vinculando las fases a través de una medida de efectividad generada durante la evaluación, llamada coeficiente de efectividad.

Palabras clave: Sistemas Tutores Inteligentes; Filtrado colaborativo; Redes bayesianas; Incertidumbre, Modelado del estudiante.

RESUMEN

ABSTRACT

The development of Intelligent Tutoring Systems aims to create tools to teach a knowledge domain in a similar way as would an experienced human tutor. To achieve it the system should be able to model the state of knowledge and recommend the best suited materials and interventions for each student. However, the knowledge associated with how people learn, the most effective ways to transmit knowledge and to measure it is incomplete and error prone. For these reasons, student modeling must take into account the uncertainty inherent in the process. This paper proposes a technique for developing a student module with the use of Bayesian networks and collaborative filtering, linking the stages through a measure of effectiveness generated during the assessment, called coefficient of effectiveness.

Keywords: Intelligent Tutoring Systems; Collaborative filtering; Bayesian networks; Uncertainty, Student modeling.

Introducción

En la sociedad se han producido profundos cambios en los últimos cincuenta años, que condicionan las necesidades de superación para las personas que deseen desempeñarse de forma adecuada en el ámbito profesional y empresarial. Este aumento en la demanda de instrucción ha provocado que el reto,

en las organizaciones encargadas, de brindar una educación más flexible sin pérdida en los estándares de calidad necesarios.

El desarrollo acelerado de las Tecnologías de la Información y de las Comunicaciones propició el auge de la educación a distancia,

que permite que la instrucción llegue a más personas debido a la eliminación de las restricciones en espacio y tiempo, así como la disminución de los costes. La eficacia de estos entornos de educación a distancia se encuentra muchas veces afectada por dificultad del estudiante para recibir soporte de parte del profesor

al realizar actividades de aprendizaje, lo que provoca que el uso de estos sistemas sea limitado teniendo en cuenta su potencial.

En este ámbito surgen los Sistemas Tutores Inteligentes (STI), aplicaciones informáticas que se adaptan a las características y necesidades de aprendizaje de los estudiantes y se encargan de enseñar cierto dominio del conocimiento de forma similar a como lo haría un tutor humano experimentado. El rasgo característico fundamental de los STI, que los diferencia de otros similares como los sistemas de instrucción asistida por computadoras, es que contienen un modelo del estudiante que recoge las principales características del alumno y le permite al sistema seleccionar la acción más adecuada a realizar para incrementar las posibilidades de aprendizaje. Al construir el modelo del estudiante sus creadores necesitan diseñar una estructura para almacenar los datos sobre el aprendizaje y los procedimientos para realizar consultas y actualizaciones sobre esta. La estructura seleccionada debe permitir que a través de una abstracción de las características del estudiante que se mantenga un balance entre su utilidad desde el punto de vista pedagógico y el costo computacional para mantenerla (Sison & Shimura, 1998). En el diseño de estas estructuras se debe tener en cuenta las condiciones de incertidumbre que rodean el conocimiento asociado a la forma en que aprenden las personas, las interacciones más adecuadas y las formas más precisas para medir el aprendizaje; conocimiento que presenta errores y es incompleto en la mayor parte de los dominios de conocimientos (Dean & Kanazawa, 1989; Woolf, 2009). Es imposible tener certeza sobre el nivel de conocimiento de un estudiante en un momento determinado, incluso si existe evidencia directa que lo avala puede ser producto de un desliz o suerte, especialmente para algunos tipos de actividades (Reye, 2004).

En el presente trabajo se expone el diseño de un modelo del estudiante para la representación del estado de conocimiento del estudiante y la toma de decisiones sobre que intervenciones y materiales deben ser usados de forma que se obtenga una ganancia en el aprendizaje. La novedad del mismo se encuentra en la aplicación de redes bayesianas dinámicas en conjunto con técnicas de recomendación basadas en filtrado colaborativo, de forma que se pueda tomar al estudiante como una fuente no

confiable de información al tener en cuenta la incertidumbre, lo que permite que se tomen decisiones más acertadas. Al mismo tiempo, se plantea el uso de una medida de efectividad basada en un marco probabilista que permite la interacción de los procesos de evaluación del aprendizaje y la recomendación de recursos.

Las siguientes secciones se organizan como se menciona a continuación. En la siguiente se muestran los trabajos que sirvieron como base teórica para el desarrollo de la presente investigación, se muestran además trabajos que han tratado la temática de modelado del estudiante y la recomendación usando un enfoque similar. Esto es seguido por la descripción de los pasos seguidos para la construcción de un modelo del estudiante para el sistema COJ-TUTOR con el uso Redes Bayesianas y filtrado colaborativo. Finalmente, se termina con una discusión de los resultados alcanzados y líneas de trabajo futuro.

Trabajos relacionados

Los trabajos de (J. Pearl, 1988; J. Pearl, 2000; Piech, Sahami, Koller, Cooper, & Blikstein, 2012) proveen un marco probabilista que sirve como base para modelar las interdependencias en el conocimiento mediante redes bayesianas, de forma que se realiza una representación más realista y se pueden incluir relaciones entre elementos del dominio que no habían sido manejadas de un modo consistente en otras técnicas anteriormente usadas basadas en la lógica formal.

Uno de los inconvenientes del uso de modelos del estudiante basados en redes bayesianas está dado por el alto costo computacional que trae consigo la actualización de los valores en redes que tienen un gran tamaño, debido a que contienen nodos que representan el estimado del conocimiento del estudiante sobre una parte del dominio y nodos de evidencia obtenidos de las acciones de los estudiantes al interactuar con el sistema. Este se minimiza a través de las aportaciones de (Reye, 2004), donde se divide la red en clústeres. Elementos dentro del dominio de conocimiento se representan mediante pequeñas redes que conectan mediante un canal principal. Para cada uno de los nodos de conocimiento se tiene una subred que

enlaza este con los diferentes nodos que sirven como evidencia para este conocimiento. Sólo los nodos de estimado de conocimiento están conectados al canal principal, limitando la inferencia que se genera entre estimado y evidencia a los elementos que componen el clúster.

Las aportaciones comentadas hasta el momento a pesar de que ofrecen un marco de trabajo para modelar el conocimiento del estudiante en condiciones de incertidumbre a través de redes bayesianas, fallan en no tomar en cuenta que el conocimiento del estudiante es dinámico, varía en el tiempo a causa de la interacción con el sistema o con cualquier otra fuente de conocimiento. Entre varias interacciones con el sistema pueden pasar periodos de tiempo variables, en los que el estudiante puede haber aprendido u olvidado. El trabajo de (Reye, 1998) propone el uso de una red bayesiana dinámica (Murphy, 2002; J. Pearl, 1988) y se propone un proceso en dos fases para la actualización del modelo del estudiante en condiciones de incertidumbre y con cambios a través del tiempo. Hasta el momento, las aportaciones que han sido comentadas, ofrecen un marco de trabajo para a través de redes bayesianas, modelar el conocimiento del estudiante en condiciones de incertidumbre, fallan en no tomar en cuenta que el conocimiento del estudiante es dinámico, varía en el tiempo a causa de la interacción con el sistema o con cualquier otra fuente de conocimiento. Entre varias interacciones con el sistema pueden pasar periodos de tiempo variables, en los que el estudiante puede haber aprendido u olvidado. El trabajo de (Reye, 1998) propone el uso de una red bayesiana dinámica (Murphy, 2002; J. Pearl, 1988) y se propone un proceso en dos fases para la actualización del modelo del estudiante en condiciones de incertidumbre y con cambios a través del tiempo.

Los sistemas de recomendación se identifican como técnicas y herramientas de software que proporcionan como sugerencias un subconjunto de elementos pertenecientes al universo de alternativas que se consideran más apropiados para el usuario (Mahmood & Ricci, 2009; Marlin, Adams, Sadasivam, & Houston, 2013). La

aplicación de estos sistemas de forma exitosa a múltiples dominios está recogida en (Ricci, Rokach, & Shapira, 2011), donde la educación ocupa un lugar importante. En el marco de los sistemas de recomendación, los algoritmos de filtrado colaborativo se usan para recomendar a un usuario elementos que han resultado de utilidad para usuarios con características similares (Herlocker, Konstan, & Riedl, 2002; Schafer, Frankowski, Herlocker, & Sen, 2007).

El uso del filtrado colaborativo para la recomendación de recursos e intervenciones ha sido ampliamente estudiado (Anderson et al., 2003; Drachsler, Hummel, & Koper, 2008; Dron, Mitchell, Siviter, & Boyne, 2000; Lemire, Boley, McGrath, & Ball, 2005). Sin embargo, en estos trabajos no se logra una integración de los procesos de evaluación del aprendizaje y recomendación que permita operar en condiciones de incertidumbre de forma efectiva.

Con el objetivo de incidir sobre deficiencias detectadas en modelos anteriores, se propone el uso de la integración de un modelo que integra la evaluación y recomendación. Las características de modelo propuesto favorecen su aplicación bajo condiciones de incertidumbre.

Materiales y métodos

En el presente trabajo se expone el diseño de un modelo del estudiante para la representación del estado de conocimiento del estudiante y la toma de decisiones sobre que intervenciones y materiales deben ser usados de forma que se obtenga una ganancia en el aprendizaje. La novedad del mismo se encuentra en la aplicación de redes bayesianas dinámicas en conjunto con técnicas de recomendación basadas en filtrado colaborativo, de forma que se pueda tomar al estudiante como una fuente no confiable de información al tener en cuenta la incertidumbre, lo que permite que se tomen decisiones más acertadas. Al mismo tiempo, se plantea el uso de una medida de efectividad basada en un marco probabilista que permite la interacción de los procesos de evaluación del aprendizaje y la recomendación de recursos.

Para realizar la evaluación de la propuesta

se implementó la misma por medio de un componente computacional integrado en un sistema informático. Se desarrolló un modelo del estudiante para el sistema COJ-TUTOR 1.0, un Sistema Tutor Inteligente para la enseñanza de la Programación para competencias estilo ACM-ICPC¹. El sistema selecciona los contenidos y actividades más adecuados para una mejor preparación, en función de sus estimados acerca del nivel del estudiante. Los ejercicios consisten en problemas que deben ser solucionados a través de un lenguaje de programación y que son evaluados mediante casos de entrada y salida diseñados por el profesor. La arquitectura del sistema COJ-TUTOR 1.0 se diseñó teniendo en cuenta los lineamientos propuestos por (Carbonell, 1970), donde se propone separar el funcionamiento del sistema en tres módulos que se intercambian información: módulo del estudiante, módulo del tutor y módulo del dominio. Esto permite a los módulos abstraerse de la forma en funciona el resto del sistema, de esta manera los cambios en la estructura interna quedan aislados a través de la interfaz que brinda el módulo al resto del sistema (Cataldi & Lage, 2009). Como exponen una de las debilidades que han presentado los sistemas desarrollados con arquitecturas similares a seleccionada radica en el hecho de que algunas funciones se encuentran partidas en varios módulos, lo que incide de negativamente en el acoplamiento del sistema. En la opinión de los autores del presente trabajo este hecho tiene una mayor incidencia cuando se ubican las funciones en semejanza a las funciones que ocupan los roles en el proceso de enseñanza aprendizaje, a pesar de violar las buenas prácticas de la programación orientada a objetos. Es por este motivo que en el sistema actual se decidió ubicar la responsabilidad de recomendar los ejercicios más adecuados en el módulo del estudiante, atendiendo al patrón de asignación de responsabilidades experto (Zhu, 2005), ya que es este módulo el que cuenta con mayor cantidad de la información necesaria para llevar a cabo la acción.

El diseño experimental se dividió en dos métodos. En el primero se evalúa la corrección de los algoritmos propuestos. La capacidad de la propuesta para hacer

converger evaluaciones correctas se hace a través del procesamiento de interacciones simuladas de estudiantes ficticios. Con los de estudiantes simulados se comprobó de manera general que el funcionamiento del sistema se comportará de forma estable ante un modelo que es una abstracción del proceso de aprendizaje.

En el segundo enfoque se paso a evaluar la experiencia de los usuarios con respecto al uso del sistema. Si hicieron estudios pilotos y al final de los mismos se aplicó un cuestionario para obtener su opinión con respecto a la calidad del sistema y el cumplimiento de los objetivos inicialmente trazados.

Las siguientes secciones se organizan como se menciona a continuación. En las siguientes se muestra la descripción de los pasos seguidos para la construcción de un modelo del estudiante para el sistema COJ-TUTOR con el uso Redes Bayesianas y filtrado colaborativo. Finalmente, se realiza una discusión de los resultados alcanzados y líneas de trabajo futuro.

Evaluación y actualización del estado de conocimiento

Para modelar el estado de conocimiento se decidió utilizar una red bayesiana que posibilita manejar la información proveniente del usuario. Para su construcción se realizaron encuestas y entrevistas a 47 entrenadores de equipos en competencias de ACM-ICPC a nivel nacional y regional para seleccionar los temas, sus relaciones de precedencia y definir los valores numéricos que indican la fuerza de cada relación. La red formada por los temas seleccionados y sus relaciones constituye el canal principal de la red como se puede observar en la figura 1.

Inicialmente se asigna a cada nodo de la red un valor de probabilidad de 0.5 que el usuario conozca el tema y 0.5 de que no lo conozca, que indica que el sistema no tiene evidencia de su estado actual de conocimiento. Se tiene asociado a cada tema el contenido, un conjunto de ejercicios a resolver, un listado con los errores que con mayor frecuencia se observan y las acciones orientadas a remediarlos. En

¹ Competencia internacional de Programación realizada de forma anual para estudiantes universitarios. Pueden encontrarse las reglas en <http://icpc.baylor.edu/icpc/finals/about.htm>

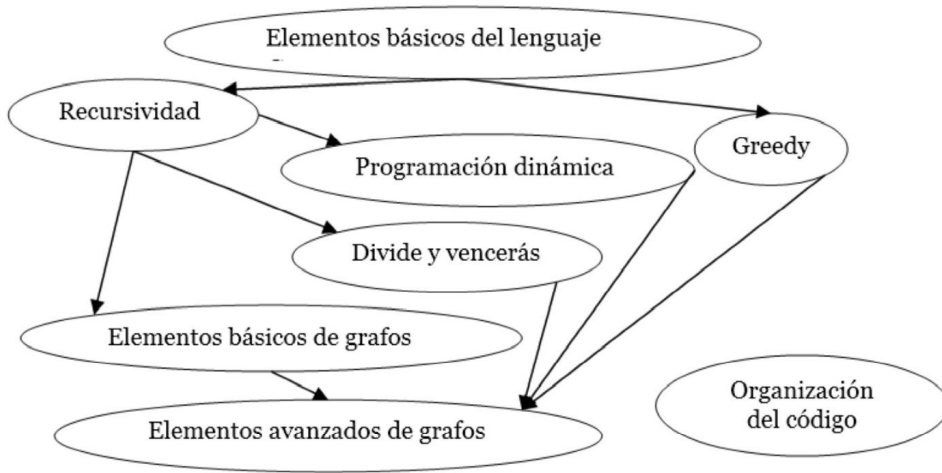


Figura 1. Papel de la Cultura informacional. Fuente: Pichs (2013)

función de la cantidad y el tipo de intervenciones que haya realizado el tutor, así como el resultado final del ejercicio se actualiza el nivel de conocimiento (como último recurso el tutor mostrará la solución correcta al estudiante y en este caso se considerará como no resuelto).

El proceso de actualización se realiza de forma iterativa en dos fases, en la primera se obtiene información sobre el conocimiento antes de la interacción y en la segunda se introducen los cambios esperados producto de la interacción. La red bayesiana dinámica diseñada permite representar los cambios en el estado de conocimiento antes, durante y después de las interacciones; se tiene en cuenta que estos cambios pueden ocurrir entre dos interacciones por motivos externos o a causa de una de ellas. El modelo gráfico que representa esta red se puede ver en la figura 2.

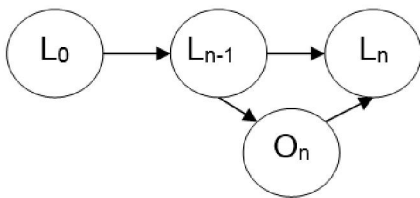


Figura 2. Red bayesiana dinámica para la actualización del estudiante en dos fases.

A continuación se usarán los siguientes términos para referirnos a elementos del problema:

O_n : Un resultado dentro del conjunto de resultados para una interacción.

$p(L_{n-1})$: Probabilidad que el estudiante conozca el tema antes de la interacción,

producto de las interacciones anteriores con el sistema.

$p(O_n|L_{n-1})$: Probabilidad que el estudiante obtenga el resultado O_n dado que conocía el tema antes de comenzar la interacción.

$p(O_n|\neg L_{n-1})$: Probabilidad que el estudiante obtenga el resultado O_n sin tener el conocimiento antes de la interacción.

$p(L_{n-1}|O_n)$: Probabilidad que el estudiante conocía el tema antes de la interacción teniendo en cuenta que su resultado fue O_n .

$p(L_n|L_{n-1}, O_n)$: Probabilidad de que si el estudiante ya conoce el tema se mantenga conociéndolo si el resultado fue O_n .

$p(L_n|\neg L_{n-1}, O_n)$: Probabilidad que un estudiante que se consideraba no conocía un tema pase a considerarse que si lo conoce luego que el resultado fue O_n .

$p(L_n|O_n)$: Probabilidad que el estudiante conozca el tema dado que el resultado fue O_n .

Para la primera fase podemos calcular el valor de $p(L_{n-1}|O_n)$ con la aplicación de la regla

de Bayes:

$$p(L_{n-1}|O_n) = \frac{p(O_n|L_{n-1}) * p(L_{n-1})}{p(O_n|L_{n-1}) * p(L_{n-1}) + p(O_n|\neg L_{n-1}) * p(\neg L_{n-1})}$$

Se calcula $p(L_n|O_n)$ de forma similar:

$$p(L_n|O_n) = \frac{p(L_n|L_{n-1}, O_n) * p(L_{n-1}|O_n) + p(L_n|\neg L_{n-1}, O_n) * p(\neg L_{n-1}|O_n)}{p(L_n|L_{n-1}, O_n) * p(L_{n-1}|O_n) + p(L_n|\neg L_{n-1}, O_n) * p(\neg L_{n-1}|O_n)}$$

Para el cálculo de las ecuaciones anteriormente expuestas son necesarios los valores de probabilidad para $p(O_n|L_{n-1})$, $p(O_n|\neg L_{n-1})$, $p(L_n|L_{n-1}, O_n)$, $p(L_n|\neg L_{n-1}, O_n)$ que son asignados por los expertos y en el caso del presente sistema se asignan atendiendo a la cantidad de pistas ofrecidas antes que el estudiante solucione el sistema, en la forma que se muestra en la tabla 1.

Al finalizar la interacción se habrá actualizado la creencia del sistema sobre el conocimiento del estudiante sobre los temas del curso. Es importante destacar en este punto que en caso de que el valor de probabilidad para el tema programación dinámica sea 0.75 no significa que el estudiante conoce el 75 por ciento del tema, sino que en función de los resultados obtenidos se estima que el estudiante conoce el tema con un 75 por ciento de confianza.

Recomendación de problemas

Al considerar los resultados de las dos fases del proceso de actualización podemos tener una medida E acerca de la efectividad de la interacción para el estudiante que llamaremos coeficiente de efectividad, que se calcula en función de la variación de la probabilidad que el estudiante conozca el tema antes y después de la

Tabla 1: Valores de probabilidad asociados a los resultados de las interacciones.

O_n	$p(O_n L_{n-1})$	$p(O_n \neg L_{n-1})$	$p(L_n L_{n-1}, O_n)$	$p(L_n \neg L_{n-1}, O_n)$
0	0.90	0.10	0.95	0.90
1	0.80	0.20	0.90	0.85
2	0.70	0.30	0.85	0.80
3	0.60	0.40	0.80	0.75
+3	0.55	0.45	0.55	0.55
NO	0.10	0.90	0.10	0.10

interacción teniendo en cuenta el resultado obtenido.

$$E = (p(L_n|O_n) - p(L_{n-1}|O_n)) * 100 + 100$$

Calculado E, entero y con valores entre 0 y 200, se almacena un registro de la interacción que es usada por el sistema para que el usuario se enfrente con los problemas más recomendables para su estado de conocimiento.

El proceso de recomendación que se mencionó anteriormente retorna aquel problema que ha demostrado tener mayor utilidad a estudiantes con estados de conocimiento similares al usuario actual, usando técnicas de filtrado colaborativo. El universo de alternativas está formado por los datos almacenados de las interacciones, de cada uno se tiene las probabilidades finales para los ejercicios enfrentados y el coeficiente de efectividad E de la presente interacción. Para la recomendación se escogió el algoritmo de los vecinos más cercanos, donde se seleccionan los elementos más cercanos en un espacio vectorial atendiendo a una medida de similitud, en este caso el coeficiente de correlación de Pearson. Según el estudio de (Lathia, Hailes, & Capra, 2008) la capacidad de recomendación del sistema no se afecta de significativa con la selección de cualquiera de las medidas de similitud más conocidas. El valor del coeficiente de correlación de Pearson para un par de elementos a y b se calcula mediante:

$$w(a,b) = \frac{\sum_k (V_{a,k} - V_a) (V_{b,k} - V_b)}{\sqrt{\sum_k (V_{a,k} - V_a)^2 \sum_k (V_{b,k} - V_b)^2}}$$

De esta forma una interacción i está representada por el vector $\langle V_{i,0}, V_{i,1}, \dots, V_{i,n} \rangle$, para los n problemas del tema (en caso de no realizado toma el valor 0). De esta forma se compara el perfil del usuario para el tema a con las interacciones b previamente almacenadas. Entonces $V_{a,k}$ representa el valor del perfil de usuario para el ejercicio k y $V_{b,k}$ el valor para el ejercicio k en la interacción b. V_a es la media de todos los ejercicios para a y V_b para b. Para evitar errores en números con coma flotante se decidió que todos los valores de probabilidad de los temas antes de ser usados en este procedimiento son multiplicados por 100, para convertirlos en valores enteros.

Al ser usado el de forma extensiva el sistema por los estudiantes se genera gran cantidad de interacciones con ejercicios. Al serle aplicado el coeficiente de correlación de Pearson algunas de ellas generan valores altos y no son usables dado que pertenecen a ejercicios resueltos anteriormente por él, causando así que se invierta tiempo computacional de forma innecesaria en el cálculo de estos valores. Por tanto, se tomó la decisión de solo analizar aquellas pertenecientes a ejercicios no resueltos por el estudiante.

Calculados los valores del coeficiente de correlación de Pearson entre el perfil del usuario actual y todas las interacciones que apliquen según los criterios anteriormente expuestos se seleccionarán los k mejores valores para el próximo paso del algoritmo. El valor más adecuado para k varía de un sistema a otro y en el caso del actual se obtuvo luego de experimentar con diferentes valores y observar el resultado. El valor con el que se trabaja por defecto es 14, aunque el mismo es modificable desde el panel de administración de la aplicación. Los datos de las k interacciones seleccionadas son ordenados atendiendo al coeficiente de efectividad E, de ellos se indicará al usuario el ejercicio que mayor valor de E si este tiene un valor mayor que 110. En caso que no exista un ejercicio que haya sido significativamente efectivo se seleccionará una de las interacciones posibles de forma aleatoria. En el primer caso, cuando existe al menos un elemento con valor de E mayor que 110, una cada 20 ocasiones se selecciona un ejercicio de forma aleatoria. El procedimiento de seleccionar un ejercicio de forma aleatoria en ocasiones, está dirigido a evitar el fenómeno de la sobre-especialización, que provocaría se recomendaran los mismos ejercicios sin tener en cuenta que otros puedan tener mayor efectividad.

Resultados y discusión

Para la simulación que se realizó se usaron 16 estudiantes simulados, donde de cada uno se definió la velocidad de aprendizaje (VA) como un valor entero en la escala de 0 a 0.4, que representa el incremento en el conocimiento por actividad del estudiante. Se crearon 8 pares de estudiantes, con igual valor en ambos, partiendo desde el valor 0.05 hasta 0.4 con incrementos de 0.05 por par. Se definió el valor para cometer un error despreocupado en 0.005.

Se usaron además los valores de 50 ejercicios con dificultades en el rango de 0 a 0.8. Al realizar un ejercicio si la diferencia entre el conocimiento del estudiante y la dificultad del ejercicio es mayor de 0.15 el estudiante se considera resuelve el problema al primer intento, si es mayor o igual que 0.05 después de un intento, mayor que -0.05 después de dos intentos, mayor que -0.10 después de tres menor o igual que -0.10 no podrá resolverlo. La simulación se realizó de forma equivalente al aprendizaje de un tema y permitió comprobar que los estudiantes con mayor velocidad de aprendizaje realizan menos ejercicios y avanzan a enfrentar los más complejos rápidamente. De igual forma se comprobó que los pares de estudiantes con igual velocidad de aprendizaje a pesar que en ocasiones tomaban caminos diferentes en cuanto a los ejercicios que realizaban, necesitan realizar como promedio la misma cantidad de ejercicios para demostrar que conocen el contenido. También resultó interesante el descubrimiento sobre la mejora en la calidad del funcionamiento del sistema al ser ejecutado sobre datos existentes sobre interacciones con los ejercicios, almacenados en ejecuciones anteriores del sistema.

El piloto se realizó en los cursos 2011-2012 y 2012-2013 en la Universidad de Ciencias Informáticas, con la asignatura Análisis y diseño de algoritmos. En estos cursos se aplicó la propuesta en 23 y 26 estudiantes respectivamente. Luego del final de cada curso se aplicó una encuesta a los estudiantes para conocer su grado de satisfacción sobre el uso del sistema y 46 por ciento de los encuestados indicó que el sistema influía positivamente en su aprendizaje, el 12 por ciento lo calificó como determinante, el 23 por ciento piensa que debe ser mejorado y el 19 por ciento restante piensa que el sistema aporta muy poco o nada en su preparación. Así mismo se presentó a un grupo de 8 profesores de la universidad un resumen de los ejercicios propuestos a los estudiantes con la información existente en el momento de la decisión: ejercicios que había resuelto y después de cuantos intentos. Los expertos coincidieron que las decisiones fueron acertadas en un 93 por ciento. Al analizar las que fueron marcadas como incorrectas se comprobó que se correspondían a ejercicios recientes con pocas interacciones o usuarios principiantes, donde se pone de manifiesto un problema conocido para

la técnica de filtrado colaborativo como arranque en frío.

Conclusiones

El desarrollo del módulo del estudiante para el sistema COJ-TUTOR 1.0 permitió modelar al estudiante como una fuente no confiable de información. Al tener en cuenta la incertidumbre se lograron recomendaciones más acertadas, de acuerdo con el criterio de los entrenadores consultados. El análisis de los datos obtenidos a partir del uso del sistema en la etapa del piloto y las opiniones de los expertos consultados arrojó que para obtener mejores resultados en evaluación del conocimiento deben individualizarse los valores de probabilidad mostrados en la tabla 1 para cada tema e incluso para cada ejercicio. Para futuras versiones del software se propone que se realice esta individualización a través de técnicas de minería de datos y aprendizaje automático.

Bibliografía

- Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D., & McGrath, S. (2003). A Rule-Applying Collaborative Filtering System. Paper presented at the Proceedings of the IEEE/WIC COLA 2003, Canada.
- Carbonell, J. R. (1970). AI in CAI: an artificial intelligence approach to computer assisted instruction. *IEEE transaction on Man Machine System*, 11(4), 190-202.
- Cataldi, Z., & Lage, F. (2009). Sistemas Tutores Inteligentes: Procedimientos, métodos, técnicas y herramientas para su creación. Paper presented at the Memorias de Virtual Educa.
- Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, 5(1), 142-150.
- Drachsler, H., Hummel, H. G. K., & Koper, R. (2008). Personal recommender systems for learners in lifelong learning: requirements, techniques and model. *International Journal of Learning Technology*, 3(4), 291-324.
- Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. Paper presented at the Proceedings of the ACM conference on Hypertext and hypermedia.
- Marlin, B. M., Adams, R. J., Sadasivam, R., & Houston, T. K. (2013). Towards Collaborative Filtering Recommender Systems for Tailored Health Communications. Paper presented at the Proceedings of the Annual Symposium of American Medical Informatics Association.
- Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. (Tesis Doctoral), UC Berkeley.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*: Morgan Kaufmann Publishers Inc.
- Pearl, J. . (2000). *Causality: models, reasoning and inference*. New York: Cambridge University Press.
- Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. . (2012). Modeling how students learn to program. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education.
- Reye, J. (1998). Two-phase updating of student models based on dynamic belief networks. Paper presented at the Proceedings of the 4th International Conference on Intelligent Tutoring Systems, San Antonio, USA.
- Reye, J. (2004). Student Modelling based on Belief Networks. *International Journal of Artificial Intelligence in Education*, 14(1), 63-96.
- Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems Handbook* Recommender Systems Handbook: Springer.
- Schafer, J.B., Frankowski, D., Herlocker, J., & Sen, S. . (2007). Collaborative filtering Collaborative filtering recommender systems The Adaptive Web (pp. 291-324): Springer.
- Sison, R., & Shimura, M. (1998). Student Modeling and Machine Learning. *International Journal of Artificial Intelligence in Education*, 9(2), 128-158.
- Woolf, B. P. (2009). *Building Intelligent Interactive Tutors*: Morgan Kaufmann.
- Zhu, Hong. (2005). *Software Design Methodology*: Safari.

Recibido: 3 de junio de 2014.
Aprobado en su forma definitiva:
26 de agosto de 2014

Enrique José Altuna Castillo
Centro FORTES, Universidad de las Ciencias
Informáticas. La Habana. CUBA
Correo electrónico:
ejaltuna@uci.cu

Lisandra Guibert Estrada
Centro FORTES, Universidad de las Ciencias
Informáticas. La Habana. CUBA
Correoelectrónico:
lguibert@uci.cu

Vivian Estrada
Centro Internacional de Postgrado,
Universidad de las Ciencias Informáticas.
La Habana. CUBA
Correo-electrónico:
miglesia@ceis.cujae.edu.cu
