

Implementation of the recommendation model LIME in cognitive and visual interactive tutors from PSLC

Alberto Corbí, Daniel Burgos
Universidad Internacional de La Rioja (UNIR)

Abstract— This paper presents an application scenario of a rule-based recommender model (LIME) to a PSLC cognitive tutor analytics database. LIME is a recommendation model which provides a formative support to students, tutors and lecturers, thanks to a visual interaction between users and the elearning system. On the other side, cognitive computer *tutors*, kindly available from the Pittsburg Science Learning Centre (PSLC) and the Carnegie Mellon University (CMU) learning analytics repository, provide intelligent and visual interactive activities that help middle-school students improve their daily academic skills. They also offer on-demand step-by-step feedback during problem solving and report on student progress for teachers and students. Every single learner interaction with a tutor is logged and is freely available within specific databases expressed in the PSLC Logging Format (also known as Tutor Message format). This format is well defined in an XML Schema Definition (XSD file) and a Document Type Definition (DTD file). We begin our text by presenting all these technologies (LIME model, tutors and the PSLC Logging Format). We then forge some example configurations of our recommender model, and apply them to a database of over 1250 students, 2250 interaction hours and almost 1 Gb of logged data. The result of this process is a potential recommendation to each of these students after the use period of the interactive tutor software. Finally, we present the results of this practical implementation, and discuss the validity of our model in the related tutoring environments.

Keywords- Recommender model, cognitive interactive tutor, PSLC Datashop, learning, interaction, informal learning, simulation.

I. INTRODUCCIÓN

HISTÓRICAMENTE, los algoritmos de recomendación en entornos de aprendizaje en línea o *elearning* adolecen de insuficientes datos experimentales con los que ser, desde inicializados, hasta finalmente contrastados y verificados. La gran mayoría de literatura al respecto – resumida muy acertadamente en [7]– se queda *estancada* en un plano singularmente operativo y en la descripción de metodologías y procedimientos. En este contexto surge la

necesidad de buscar y usar fuentes de datos alternativas que puedan de alguna manera adaptarse al entorno de investigación de cada sistema recomendador. En este trabajo nos planteamos el uso de bases de datos del LearnLab del Pittsburg Science Learning Centre (PSLC). Este repositorio cuenta con ingentes cantidades de registros y cientos de actividades realizadas por miles de alumnos, así como cada uno de los pasos que cada uno de estos han dado para la consecución, exitosa o no, de las mismas. Actualmente se están implantando otras versiones inspiradas en el LearnLab del PSLC, como el LearnLab India [1, 2], que no solamente incorporan datos provenientes desde sesiones *de laboratorio*, sino también durante el transcurso normal de clases ordinarias.

La principal expresión del PSLC LearnLab es la herramienta *DataShop* que permite la descarga de bases de datos sobre *learning analytics* e incluso la realización de análisis preliminares en línea. *DataShop* es una aplicación web HTML5. Simultáneamente es también un sistema en red donde muchos programas *tutores* e interactivos pueden enviar de manera continua y en tiempo real información sobre las acciones y resultados de los estudiantes durante su interacción con los anteriores. Esta transmisión de datos se realiza en el formato *Tutor Message Format* (TMF), que también introduciremos brevemente en la sección II-C.

En este trabajo nos planteamos cómo atajar el problema de la deficiencia de bases de datos sobre monitorización de estudiantes integrando un sistema y modelo de recomendación concreto (LIME, que presentaremos también más adelante) con uno de estos ingentes repositorios de monitorización de actividades de estudiantes disponibles gracias al PSLC. En concreto haremos uso de una recopilación de la interacción de más de mil estudiantes con el software *MathTutor* y de varias configuraciones de nuestro modelo con el objetivo de verificar la idoneidad de nuestra propuesta. No es pretensión de este trabajo el estudiar cómo influyen las recomendaciones sugeridas en el posterior devenir académico del alumno, sino demostrar que los sistemas tutores desarrollados mediante tecnologías como CTAT y *MathTutor* y repositorios de sesiones de monitorización basados en TMF pueden utilizarse de manera exitosa para crear, refinar o poner a prueba sistemas recomendadores, con una fuerte componente visual que facilita una interacción eficaz usuario-sistema.

En una primera parte de este trabajo introduciremos al lector en las tecnologías de monitorización propuestas por el LearnLab y

el PSLC. Seguidamente presentaremos MathTutor y el modelo recomendación LIME. En la sección de metodología detallaremos la base de datos escogida y qué posibles adaptaciones han de aplicarse para su correcta incorporación al flujo de recomendación. Explicaremos también las configuraciones del modelo seleccionadas y cómo estas han sido aplicadas a la base de datos escogida. Por último presentaremos los resultados de todos estas evaluaciones y los discutiremos en la sección de conclusiones de este artículo.

II. TECNOLOGÍAS EMPLEADAS

Plantamos el uso específico del sistema tutor *MathTutor* [3] para nuestra investigación, pero existen muchas otras alternativas compatibles y cuyos datos están organizadamente presentes en la arquitectura Datashop. Aquí exponemos algunos otros ejemplos que podríamos haber utilizado de igual manera en nuestra investigación:

Andes Physics Tutor presenta ejercicios de física como los de la Figura 1 y guía a los estudiantes durante la resolución [4], monitorizándoles a lo largo de toda la ejecución.

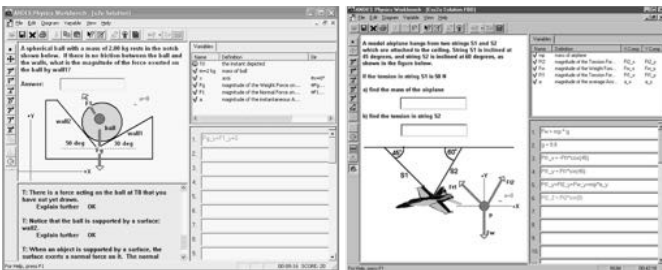


Figura 1. Varios ejemplos de uso del *Andes Physics Tutor*.

Chemistry Virtual Lab simula un entorno de laboratorio de química como el mostrado en la Figura 2 y persigue los mismos objetivos que el *Andes Physics Tutor*.

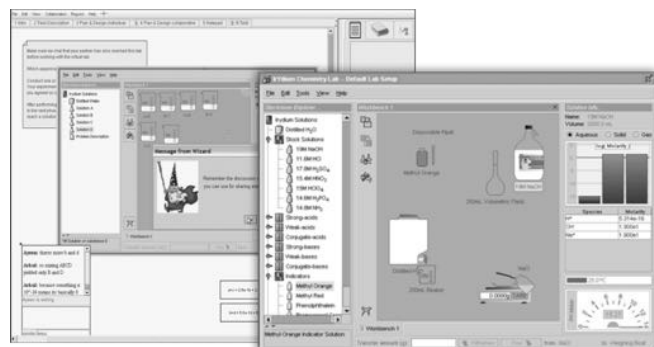


Figura 2. Varias escenas del *Chemistry Virtual Lab*.

Numbaland monitoriza al alumno mientras opera con pequeños juegos interactivos y conceptos numéricos variados [5]. Varias pantallas de esta herramienta aparecen en la figura 3.

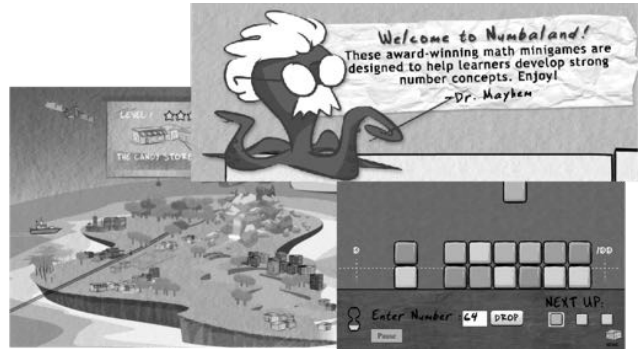


Figura 3. Varios juegos interactivos que forman parte de *Numbaland*.

Assistment es una plataforma online que permite a los profesores escribir preguntas sobre las que los estudiantes obtienen asesoramiento que pueda resultarles de utilidad [6].

Todos estos sistemas tutores están confeccionados sobre la tecnología *Cognitive Tutor Authoring (CTAT)*, la cual resumiremos en el apartado C de esta sección.

Por otro lado, el PSLC es una institución creada a su vez por la Fundación Nacional para la Ciencia (*National Science Foundation*) y está gestionada por directamente por la Universidad de Pittsburg y la Universidad Carnegie-Mellon (CMU). Simultáneamente forma parte del *Human Computer Interaction Institute (HCII)*. Gracias a su espacio web *DataShop* [10], es posible la descarga y consulta en línea de cientos de bases de datos que registran la actividad de estudiantes durante el desarrollo de ejercicios o durante la resolución de problemas mediante distintos sistemas tutores. De igual manera, el PSLC *DataShop* permite que, investigadores en el terreno del aprendizaje, contribuyan de manera altruista a este repositorio de información sobre *learning analytics*. *DataShop* se actualiza de manera constante con nuevas bases de datos y capacidades de análisis online. De igual manera, el nivel de interacción entre los gestores/administradores y usuarios investigadores del PSLC es remarcable, llegando a ser muy habitual recibir frecuente información sobre mejoras en la plataforma, actualizaciones o tiempos de inactividad por labores de mantenimiento.

Algunas de estas bases de datos son accesibles sin ningún tipo de restricción (Figura 4), mientras que para otras es necesario cumplimentar una solicitud de acceso. Una vez descargadas, pueden ser filtradas y analizadas con cualquier paquete de análisis numérico y estadístico (SPSS, R, Matlab, etc.).

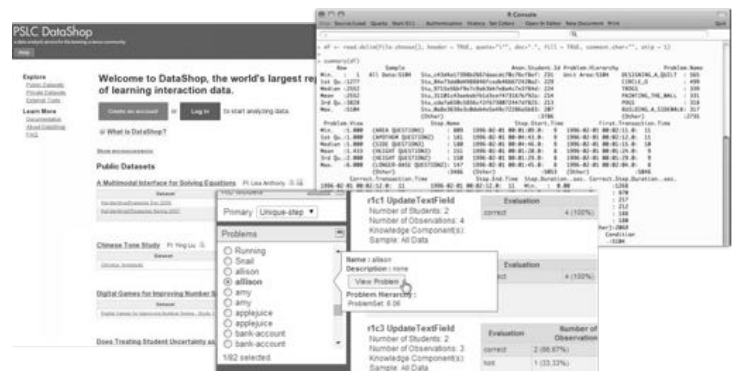


Figura 4. Varias de las secciones de la web del PSLC *DataShop* y procesado de datos con el paquete estadístico *R*.

Una de las bases de datos que el PSLC pone a disposición de la comunidad investigadora en tecnologías educativas es

precisamente la recolectada durante sesiones de estudio con alumnos reales entorno al software educativo MathTutor, que resumimos a continuación.

A. MathTutor

MathTutor es una herramienta de software confeccionada conjuntamente por el *Institute of Education Sciences* (dependiente del Departamento de Educación del Gobierno de Estados Unidos), el propio LearnLab de la CMU [12] y el *Human-Computer Interaction Institute* de la Universidad Carnegie-Mellon.

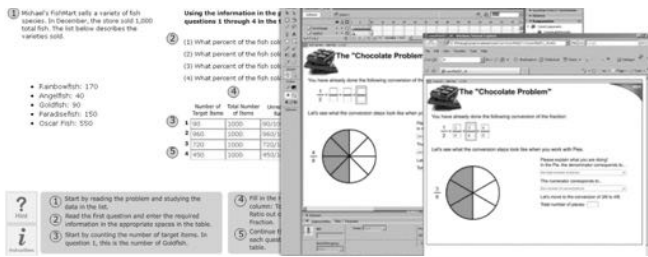


Figura 5. Una de las pantallas de MathTutor

MathTutor (figura 5) permite el diseño de conjuntos de problemas matemáticos que los alumnos deben resolver. Estos boletines de ejercicios van desde las más simples operaciones aritméticas hasta problemas cotidianos como la separación de los gastos tras una cena en sociedad. Durante la realización de los cálculos, los alumnos pueden pedir asesoramiento a un *tutor* virtual, el cual motiva de manera natural la interacción activa entre persona y computadora.

Actualmente, MathTutor se ejecuta como un programa escrito en tecnología Adobe® Flash. Asimismo, MathTutor está basado en la herramienta Cognitive Tutor Authoring, que presentamos a continuación.

B. Cognitive Tutor Authoring

El *framework* Cognitive Tutor Authoring (CTAT) [8, 9] es la base de muchos sistemas tutores. Gracias a CTAT es posible implementar de manera sencilla el enfoque *learn-by-doing* en cursos online, con el objetivo de potenciar el aprendizaje autónomo y la interacción persona-computadora. Estos tutores virtuales tienen su expresión en todo tipo de escenarios relacionados con la resolución de problemas y ejercitación de habilidades, desde las más simples hasta las más complejas. Los paquetes de software tutores programados sobre CTAT monitorizan el trabajo de los estudiantes a lo largo de la resolución de ejercicios y proveen de consejos y asesoramiento en tiempo real.

Los tutores basados en CTAT suelen estar desarrollados en tecnologías Java o Flash y por lo tanto pueden ser desplegados tanto en entornos web como escritorio. La Figura 6 muestra varios de estos entornos. Como ya hemos adelantado con el caso de MathTutor, estas herramientas tutores desarrolladas sobre CTAT se integran de manera transparente con el DataShop del PSLC. Igualmente, cualquiera de estos sistemas tutores es capaz de generar ficheros de seguimiento con las interacciones entre el sistema y el alumno. Esta secuencia de datos es expresada en el formato *Tutor Message Format*, que veremos en la siguiente sección. La fase de monitorización y registro no requiere

ningún esfuerzo extra por parte de los autores y/o investigadores: CTAT, en conjunción con DataShop se ocupa de todas las necesidades de carácter técnico: monitorización, serialización, transmisión de información a DataShop, control de calidad y, finalmente, persistencia e integración.

Existen actualmente numerosos tutores creados a partir de CTAT para enseñar todo tipo de materias, desde economía, lenguajes de programación, genética, protocolo y buenas maneras, termodinámica, reacciones químicas descritas mediante estequiometría, representación binaria, etc. Todos permiten, desde un enfoque HCI, adquirir destrezas y fomentar el aprendizaje de manera independiente. Presentamos más ejemplos de estos sistemas tutores en la siguiente Figura (genética, termodinámica y química):

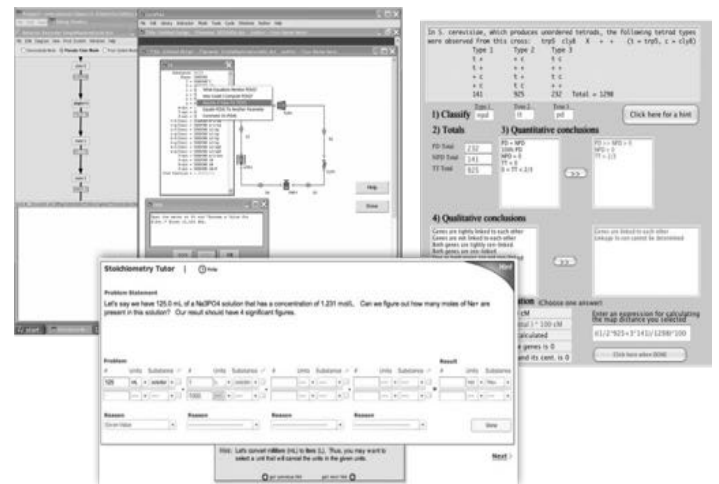


Figura 6. Otros sistemas tutores construidos sobre el *framework* CTAT.

C. Tutor Messaging Format (TMF)

Los sistemas tutores, como MathTutor, creados a partir del marco operativo CTAT, registran la actividad de los estudiantes mediante un esquema XML. Este flujo de información, conocido como *Tutor Messaging Format* [8, 11], puede posteriormente transmitirse al servicio DataShop del PSLC donde es integrado de manera organizada en su base de datos.

En este ejemplo se registra uno de los pasos –incorrecto, en este caso– de un estudiante y cómo a continuación el sistema tutor le hace llegar automáticamente una sugerencia. Finalmente se especifica qué habilidad pretende ser trabajada en esa etapa precisa del ejercicio para que quede también registrada, aunque sea de manera redundante.

```
<?xml version="1.0" encoding="UTF-8"?>
<tutor_message attempt_id="0CEF2E07-24DE-BFDE">
  <problem_name>ChemPT1</problem_name>
  <semantic_event id="0F957AB5-B828-4FC5-515F"
    semantic_event_id="0FB7AE5D-B8B9-3AAA"
    name="RESULT" />
  <event_descriptor>
    <selection>Numerator1Value</selection>
    <action>UpdateTextField</action>
    <input>1000</input>
  </event_descriptor>
  <action_evaluation>INCORRECT</action_evaluation>
  <tutor_advice>
    "1000" should be in this term's ratio, but look at the
    units you need to cancel in the first term. Will
    placing 1000 mg in the numerator cancel in the first
    term?
  </tutor_advice>
  <skill>
    Set-Numerator-Unit-of-Unit-Conversion
```

</skill>
</tutor_message>

Con un nivel mayor de detalle, estos son los campos presentes en un mensaje TMF y que pueden resultar de interés para cualquier proyecto de investigación relacionado con sistemas recomendadores en educación:

- 1) *Problema*: Se trata de una tarea que un estudiante debe resolver y comporta normalmente varios pasos o estadios. Puede ser desde hallar una secuencia genética o una simple simplificación de fracciones.
- 2) *Etapas*: Se corresponde a una parte observable de la solución de un problema por parte de un alumno. Existe cierta discusión sobre qué amplitud deben tener estas etapas tal y como plantea [13].
- 3) *Transacción*: Cualquier tipo de interacción entre el alumno y el sistema tutor.
- 4) *Componente de conocimiento*: Se trata de cada uno de los ítems y conceptos con los que un alumno puede construir la solución a un problema.
- 5) *Duración de la etapa*: Tiempo que emplea el alumno entre paso y paso de la resolución de un ejercicio.
- 6) *Tasa de error*: Porcentaje de estudiantes que solicitaron una pista o cometieron un error en la primera transacción.
- 7) *Tasa de asistencia*: Número de veces que un alumno solicita ayuda o pistas al programa tutor.
- 8) *Tiempo total de ejecución*: Se corresponde con el tiempo que dedica un alumno a resolver una actividad propuesta por el sistema tutor.

Como consideración final, el TMF forma parte del ya interesante conjunto de estándares de registro de eventos y actividades de usuarios y estudiantes basados en XML ó JSON: TinCan API (también conocido como xAPI), JSON Activity Streams [14], IMS Caliper [15] ó los IEEE 1484.11.1 e IEEE 1484.11.2 [16].

III. EL MODELO DE RECOMENDACIÓN LIME

LIME es un modelo de recomendación basado en reglas, que permite diseñar, aplicar y evaluar recomendaciones en función de la actividad del usuario, de una manera visual e interactiva [17].

Las siguientes Figuras 7 y 8 muestran un despliegue específico del modelo sobre un sistema LMS concreto: Apereo Sakai CLE. De cara al profesor, LIME ofrece un interfaz HCI limpio e intuitivo, donde el docente configura las reglas y los parámetros del modelo (detallados a continuación). Por otro lado, de cada al estudiante, LIME es capaz de presentar recomendaciones fruto de la ejecución de las reglas anteriormente mencionadas.

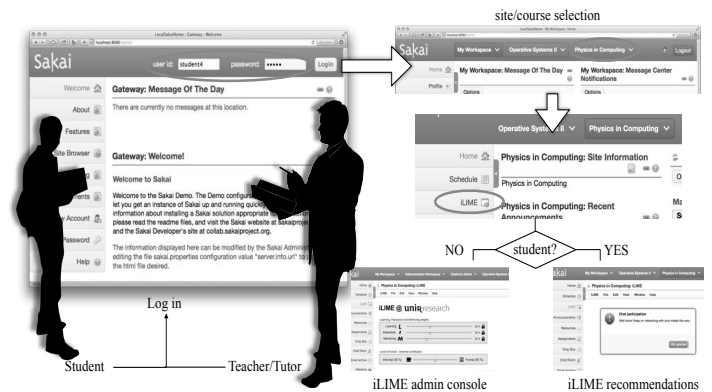


Figura 7. Implementación práctica de LIME en el módulo integrado en el LMS Apereo Sakai CLE (como herramienta externa IMS LTI).

Gracias a un interfaz dinámico, el profesor puede diseñar una experiencia de aprendizaje adaptada a cada usuario, en función de su rendimiento académico, interacción con el grupo, actividad online, y otros factores. El alumno, por su parte, recibe el resultado de aplicar esta configuración en forma de recomendaciones sobre su itinerario de aprendizaje y acciones concretas.

LIME se centra en cuatro vectores, según sus iniciales: *Learning, Interaction, Mentoring* y *Evaluation* (Aprendizaje, Interacción, Tutorización, y Evaluación). Cada vector permite incluir una serie de controles y reglas asociadas, teniendo en cuenta si se desarrollan en un entorno formal o informal de aprendizaje. De esta manera, se combina el entorno académico reglado (formal) con el entorno espontáneo externo (por ejemplo redes sociales o foros de discusión ajenos a los propuestos por la institución educativa), permitiendo que todo se integre en una experiencia de aprendizaje consolidada. Cada vector es ponderado y relacionado con los demás, con el objeto de proporcionar un apoyo unificado al alumno.

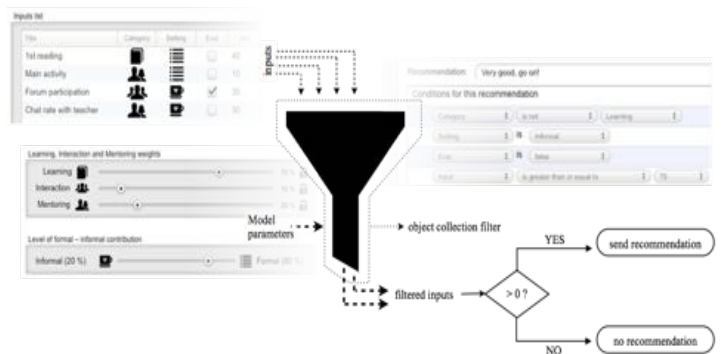


Figura 8. Interfaz de diseño de reglas y parámetros en LIME.

De manera más detallada, las entradas al modelo correspondientes a la categoría *Learning* son todas aquellas relacionadas con las acciones individuales de cada alumno y que revierten en su propio aprendizaje (leer un tema, entregar un boletín de ejercicios, etc.). Las categoría *Interaction* engloba aquellas acciones en las que el alumno ha contribuido a su propio aprendizaje en colaboración con otros alumnos (trabajos en grupo, chats, participación de foros, respuestas a mensajes, envío de correos a otros compañeros, etc.). La categoría *Mentoring* engloba el resto de elementos a los que el tutor da relevancia de cara al proceso de aprendizaje. Por último, la categoría transversal *Evaluation* agrega aquellas actividades con repercusión en las calificaciones del alumno.

El enfoque visual de LIME permite una interacción fluida y amigable entre el usuario (profesor y/o alumno receptor de la recomendación) y el modelo conceptual, basado en las características que acabamos de reflejar.

LIME contrasta con otros modelos de recomendación por el hecho principal de estar basado en reglas bajo el absoluto y continuo control del profesor. Los principales enfoques al proceso de recomendación son los basados en contenidos y los filtros colaborativos. En el caso del ámbito del aprendizaje online los contenidos son, evidentemente, recursos consumidos por los alumnos (videos, webs, archivos PDF, etc.) y que estos tienen que valorar de alguna manera. La información relativa a la colaboración se obtiene, por ejemplo, en la monitorización de la confluencia de los alumnos en espacios virtuales comunes (asistencia a la misma clase, coincidencia en grupos de trabajos) a lo largo del curso. Enfoques más recientes tienden a combinar ambas alternativas.

Otro elemento de diferenciación con respecto a otros sistemas recomendadores en aprendizaje es el espacio que LIME reserva a la adquisición de conocimientos y destrezas en escenarios informales (fuera del aula), permitiendo al profesor configurar el modelo y la aplicación de reglas teniendo en cuenta esta realidad cada día más presente en el mundo académico.

Un buen resumen del estado de la técnica en sistemas recomendadores en elearning puede consultarse en [7].

IV. METODOLOGÍA

A continuación describimos la base de datos DataShop escogida y qué adaptaciones han sido necesarias para garantizar su correcta integración con el modelo de recomendación (LIME) que acabamos de presentar.

A. Base de datos escogida y agregación de transacciones y actividades.

La base de datos PSLC que hemos seleccionado para la aplicación de nuestro modelo recomendador es la correspondiente a las experiencias sobre aprendizaje en matemáticas realizadas con estudiantes entre los años 2005 y 2010 por parte del *Instituto para la Investigación de la Interacción Hombre-Máquina* de la Escuela de Informática de la Universidad Carnegie-Mellon. Estas sesiones de laboratorio tuvieron lugar mediante el sistema tutor MathTutor (asentado también sobre el marco CTAT, discutido anteriormente).

Este conjunto de datos, en formato Tutor Message Format explicado anteriormente, agrupa las siguientes características:

- Más de 1200 alumnos.
- Casi dos millones de transacciones en total.
- Casi 2300 horas de interacción entre alumnos y el sistema tutor.
- 145 problemas matemáticos de nivel de educación primaria/secundaria que van desde operaciones aritméticas básicas, geometría y aplicaciones prácticas de conceptos matemáticos.

La Figura 10 muestra algunas de las pantallas del aplicativo que los alumnos usaron durante el mencionado ensayo.

Como se ha comentado previamente, el desglose de fuentes de datos en el sitio web DataShop es muy variado y

extenso. Incluso dentro del área de trabajo HCI en matemáticas, tenemos opción de seleccionar otros conjuntos de datos para su posterior tratamiento en el marco de proyectos de investigación en *learning analytics*, como puede verse por la figura 9:

Dataset	Domain/LearnLab	Dates	Status	Transactions
MathTutor Spring 2009	Other	May 19, 2009 - Jul 9, 2009	complete	4,088
MathTutor	Other	Dec 16, 2009 - Feb 18, 2010	complete	4,185
MathTutor through July 15, 2010	Other	Jun 30, 2005 - Jul 15, 2010	complete	1,857,091
MathTutor through January 31, 2013	Other	Jul 18, 2010 - Jan 31, 2013	complete	425,363
MathTutor	Other	Jan 2, 2004 - Sep 18, 2014	on-going	657,220
MathTutor Individual Students	Other	Feb 4, 2013 - Nov 7, 2013	on-going	183,842
MathTutor Problem Set 6.01	Math/Other	Sep 2, 2010 - Oct 2, 2013		24,671
MathTutor Problem Set 6.01 v2	Math/Other	Sep 15, 2010 - Oct 2, 2013		23,198
MathTutor Problem Set 6.01 from Fractions project	Math/Other	Nov 9, 2011 - Dec 10, 2011		181,589
POL Course Project: MathTutor Problem Set 6.01	Math/Algebra	-		181,589

Figura 9. Otras secuencias de datos capturadas mediante MathTutor y disponibles gracias al PSLC DataShop (resaltada la empleada en este trabajo).

La siguiente Figura muestra algunas de las pantallas que los alumnos visualizan durante sus sesiones de trabajo con MathTutor (concretamente con el conjunto de problemas que se integran en la base de datos empleada en este proyecto de investigación):

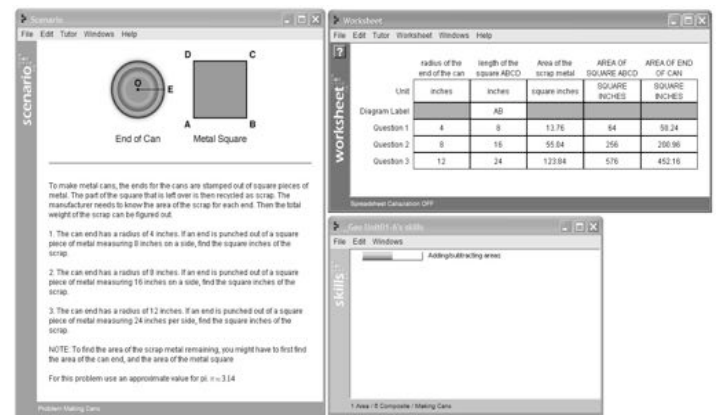


Figura 10. Algunas pantallas de MathTutor con las que los más de 1200 alumnos interactuaron entre 2005 y 2010.

Las transacciones han sido inicialmente agrupadas por valores estudiante-problema. Es decir, cada *actividad* (cada uno de los 145 problemas a resolver) ha sido considerada como una entrada discreta *mínima* para el modelo. El recomendador será aplicado sobre el conjunto de actividades que abstraeremos a un periodo académico típico de unos días o semanas y tras el cual los alumnos reciben una recomendación. Esta agregación del periodo académico queda reflejada en la Figura 11:

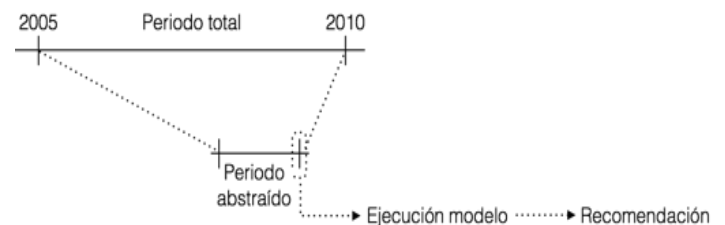


Figura 11. Adaptación del periodo de interacción total con MathTutor a un *periodo académico abstraído*.

Esta agregación de actividades se ha llevado a cabo por medio del paquete estadístico *opensource* y multiplataforma R (figura 12):



Figura 12. Fusión en el tiempo de actividades por medio del software de análisis estadístico R.

Cada problema/actividad ha sido clasificado según cada una de las categorías del modelo (*learning – interaction – mentoring*) y según su carácter pedagógico, teniendo en cuenta la definición específica de cada categoría. Asimismo se han compilado en escenarios formales o informales según si su contenido era susceptible de ser aprendido, practicado y asimilado en el aula o fuera de la misma. Cumplimentada esta agrupación, se asigna un peso porcentual indicativo de la relevancia que da el profesor-tutor a cada categoría y escenario. Los ítems evaluables (categoría *evaluation*) han sido escogidos aleatoriamente hasta un 20% del total (unas 30 actividades), configuración habitual para un periodo académico en un campus universitario online medio.

Un ejemplo de esta asignación de pesos específicos puede verse en la siguiente figura:

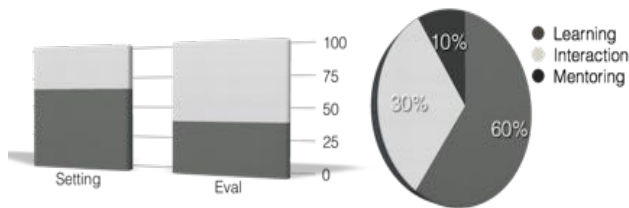


Figura 13. Ejemplo de configuración del modelo LIME.

B. Entradas para la categoría mentoring

La categoría *mentoring*, según ha quedado resaltado anteriormente, engloba a aquellas acciones del alumno para las que el profesor/tutor/pedagogo presta una especial atención. MathTutor (y en general, cualquier sistema tutor basado en CTAT) guarda registro de varios tipos de datos que pueden encajar en el espíritu de esta categoría del modelo LIME.

Además de la clasificación de cada uno de los problemas, se han considerado integrantes de la categoría *mentoring* las siguientes entradas:

- *Latencia* o tiempo entre acciones del estudiante en MathTutor.
- *Tasa de error* o pasos equivocados en la resolución de un problema.
- *Tasa de asistencia* o número de veces que el alumno precisa de asesoramiento por parte del sistema tutor.
- *Pasos* o número de pasos en que queda resuelto el ejercicio.
- *Tiempo hasta resolución correcta* o el tiempo medio entre el inicio de la resolución del problema o un paso previo hasta otro resuelto correctamente.

C. Aplicación del modelo LIME

Una vez equiparadas las actividades de MathTutor y las entradas del modelo y asignados pesos a cada una de las categorías, puede tener lugar la aplicación del recomendador LIME.

La aplicación del modelo se concreta en la ejecución de una serie de reglas sobre las entradas o *inputs* (acciones de los estudiantes). Las reglas filtran qué entradas satisfacen ciertas condiciones. Si tras el proceso de filtrado por parte de una regla se tiene un conjunto de entradas *no nulas*, se hace llegar una recomendación al estudiante. Esta recomendación, así como la confección de las reglas y configuración global del modelo, corresponde exclusivamente al profesor y/o responsable del grupo/clase.

Esta simulación se ha ejecutado en el lenguaje de programación Ruby gracias a sus estimables capacidades en el procesamiento de grandes segmentos y árboles XML y en la fluida agregación de datos. Por otro lado, también ha resultado muy apropiada la operatividad de este lenguaje en el tratamiento de información fruto de la interacción persona-computadora.

V. RESULTADOS

A continuación se presentan varias ejecuciones del modelo con distintas configuraciones y reglas. Para cada ejecución se muestra qué porcentaje de alumnos de la base de datos PSLC escogida recibe la recomendación. Como se ha comentado previamente, se ha adaptado el largo periodo de captura de datos (2005–2010) a un lapso de tiempo académico habitual (unas semanas). Tras este intervalo, y antes de que los alumnos prosigan con su actividad académica, son receptores de una recomendación por parte del sistema LIME, si procede.

En la siguiente tabla mostramos qué número de alumnos recibe una recomendación (que no se especifica) atendiendo a la configuración del modelo, el escenario (formal o informal) y a una regla de ejemplo.

LIME (%)	Setting (%)	Regla	# estudiantes
20-30-50	50-50	si los inputs pertenecen a la categoría <i>Interaction</i> y se encuentran entre el 50 y el 100 % del máximo	439
30-20-50	70-30	si los inputs pertenecen a la categoría <i>Mentoring</i> y se encuentran entre el 75 y el 100 % del máximo	531
10-40-50	30-70	si los inputs informales encuentran entre el 0 y el 40 % del máximo	875
70-20-10	90-10	si los inputs pertenecen a la categoría <i>Learning</i> o <i>Interaction</i> y se encuentran entre el 75 y el 100 % del máximo	197
70-30-0	80-20	si los inputs pertenecen a la categoría <i>Learning</i> y son evaluables y se encuentran entre el 85 y el 100 % del máximo	391

VI. CONCLUSIONES

En este trabajo hemos presentado la adaptación del modelo recomendador LIME a una base de datos con información sobre la monitorización de las actividades de más de mil estudiantes. Las actividades se corresponden con ejercicios de matemáticas del nivel apropiado para formar parte del currículum académico de los

primeros años de educación secundaria y final de primaria. La monitorización se ha realizado con el software tutor MathTutor basado en los entornos *tutor* del LearnLab del PSLC. Por su parte, LIME se nutre de entradas (acciones de los alumnos) que son previamente clasificadas en cuatro categorías (*learning, interaction, mentoring*, más una transversal, *evaluation*) y en escenarios formales e informales, combinados. A cada una de estas compartimentaciones se les asigna un peso por parte del profesor o responsable del grupo/clase. El profesor también es el encargado de definir las reglas lógicas a aplicar por parte del modelo y qué recomendación se remitirá al alumno en caso de que proceda. Unas 145 actividades en total llevadas a cabo con MathTutor han sido adaptadas a estas entradas y a su vez a cada una de las categorías y escenarios. Esta clasificación se ha efectuado según el contenido de la actividad y según dónde se hayan adquirido los conocimientos para realizarla con mayor probabilidad (en el aula –educación formal– o fuera de la misma –aprendizaje no formal–).

A modo de prueba de concepto, hemos aplicado varias reglas junto a varias configuraciones del modelo a una base de datos específica del PSLC (MathTutor 2015–2010). Hemos podido demostrar de esta manera, la idoneidad de estas ingentes fuentes de *learning analytics* –que a menudo pasan desapercibidas– como entradas a un sistema recomendador concreto.

AGRADECIMIENTOS

Esta actividad ha sido parcialmente financiada por UNIR Research (<http://research.unir.net>), Universidad Internacional de la Rioja (UNIR, <http://www.unir.net>), dentro del Plan Propio de Investigación, Desarrollo e Innovación [2013–2015], Grupo de Investigación TELSOCK.

Agradecemos también a la Universidad Carnegie–Mellon y al Pittsburg Science Learning Centre su generosidad por el mantenimiento y eficiente gestión del repositorio de *learning analytics* DataShop.

REFERENCES

[1] Chopra, N. and Lal, M. (2012). *Research Methodology for Educational Data Mining in India*, INDIACOM 2012.

[2] Rosé, C. P. and Kam, M. (2010). *LearnLab India: Towards in Vivo International Comparative Education Research*, p. 102–103.

[3] Anthony, L. (2008). *Developing handwriting-based Intelligent Tutors to enhance mathematics learning*, Carnegie Mellon University.

[4] Craig, S., VanLehn, K. and Chi, M. T. (2008). *Promoting learning by observing deep-level reasoning questions on quantitative physics problem solving with Andes*, p. 1065.

[5] Stamper, J. C., Lomas, D., Ching, D., Ritter, S., Koedinger, K. R. and Steinhart, J. (2012). *The Rise of the Super Experiment*, International Educational Data Mining Society.

[6] Song, S. (2010). *Automatic Reassessment and Relearning System*, Worcester Polytechnic Institute.

[7] Shishehchi, S., Banihashem, S., Zin, N. and Noah, S. (2011), *Review of personalized recommendation techniques for learners in e-learning systems*, p. 277–281.

[8] Ritter, S. and Koedinger, K. R. (1996), *An architecture for plug-in tutor agents*, Journal of Artificial Intelligence in Education 7, p. 315–348.

[9] Aleven, V., McLaren, B. M., Sewall, J. and Koedinger, K. R. (2006). *The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains*, p. 61–70.

[10] Koedinger, K. R., Baker, R., Cunningham, K., Skogsholm, A., Leber, B. and Stamper, J. (2010). *A data repository for the EDM community: The PSLC DataShop*, Handbook of Educational Data Mining, CRC Press.

[11] Jugo, I., Kovačić, B. and Slavuj, V. (2014). *Using Data Mining for Learning Path Recommendation and Visualization in an Intelligent Tutoring System*, 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).

[12] Aleven V., McLaren, B. and Sewall, J. (2009). *Scaling Up Programming by Demonstration for Intelligent Tutoring Systems Development: An Open-Access Web Site for Middle School Mathematics Learning*, Learning Technologies, IEEE Transactions on 2, 64–78.

[13] VanLehn, K., Koedinger, K., Skogsholm, A., Nwaigwe, A., Hausmann, R., Weinstein, A. and Billings, B. (2007). *What's in a Step? Towards General, Abstract Representations of Tutoring System Log Data*. In: Conati, C., McCoy, K. & Paliouras, G. (Ed.), *User Modeling 2007*, Springer Berlin Heidelberg.

[14] Snell, J., Atkins, M., Norris, W., Messina, C., Wilkinson, M. and Dolin, R., (2011), *JSON Activity Streams 1.0 specification*.

[15] Lukarov, V., Chatti, M. A., Thus, H., Kia, F. S., Muslim, A., Greven, C. and Schroeder, U. (2014). *Data Models in L.A.*

[16] Stracke, C. M. (2006). *Interoperability and Quality Development in e-Learning. Overview and Reference Model for e-Learning Standards*, Proceedings of the Asia-Europe e-Learning Colloquy, e-ASEM (Seoul).

[17] Burgos, D. (2013), *L.I.M.E. A recommendation model for informal and formal learning, engaged*, Journal of Artificial Intelligence and Interactive Multimedia, Vol. 2, No 2.



search (CSIC).

Alberto Corbi, M.Sc. works as a senior researcher at the Technology-enhanced Learning & Social Networks (TEL-SOCK) research group and as a assistant professor at the School of Engineering, both part of the International University of La Rioja. With a background in physics (M.Sc. in ocean-atmosphere interaction), computing and education, he is currently involved in research fields around recommender systems, eLearning standards and systems interoperability. Simultaneously, he is working on his PhD thesis around medical imaging at the Spanish Council for Scientific Re-



Prof. Dr. Daniel Burgos works as Vice-Chancellor for Research & Technology and UNESCO Chair on eLearning at the International University of La Rioja (www.unir.net, <http://research.unir.net>). Previously he worked as Director of Education Sector and Head of User Experience Lab in the Research & Innovation Department of Atos, Spain, and as an Assistant Professor at Open Universiteit Nederland before that. His research interests are mainly focused on Adaptive and Informal eLearning, Learning & Social Networks, eGames, and eLearning Specifications. He is or has been involved in a number of European-funded R&D projects, such as Intuitel, Hotel, Edumotion, Inspiring Science Education Stellar, Gala, IntelLEO, Go-MyLife, Grapple, Unfold, ProLearn, TenCompetence, EU4ALL, NiHao, Kaleidoscope, Sister, and ComeIn. Prof. Dr. Burgos holds degrees in Communication (PhD), Computer Science (Dr. Ing), Education (PhD), and Business Administration.

