TECNOLOGÍA EDUCATIVA ISSN: 2519-9463

ADAPTACIÓN DEL ALGORITMO BAYESIAN KNOWLEDGE TRACING PARA LA ESTIMACIÓN DEL CONOCIMIENTO LATENTE SOBRE DATOS EDUCACIONALES MASIVOS

ADAPTATION OF THE BAYESIAN KNOWLEDGE TRACING ALGORITHM FOR THE ESTIMATION OF LATENT KNOWLEDGE ON MASSIVE EDUCATIONAL DATA

M.Sc. Angel Alberto Vazquez Sánchez

aavazquez@uci.cu

Universidad de las Ciencias Informáticas, Cuba

M.Sc. Lisset Salazar Gómez

lsgomez@uci.cu

Universidad de las Ciencias Informáticas, Cuba

Dr.C. Roxana Cañizares González

rcanizares@uci.cu

Universidad de las Ciencias Informáticas, Cuba

Resumen

Ante la masividad de los datos que se generan en la educación, se han tenido que cambiar los métodos tradicionales para el descubrimiento de conocimientos. Uno de los algoritmos es el Bayesian Knowledge Tracing (BKT) que permite Estimar Conocimiento Latente (ECL). La ECL no es más que la forma de medir el conocimiento de un estudiante sobre habilidades y conceptos específicos, que es evaluada por sus patrones de corrección sobre esas habilidades. Dicho algoritmo está diseñado para ser utilizado en volúmenes de datos pequeños, afectándose su rendimiento ante la presencia de grandes volúmenes de datos. Para dar solución al problema se presentará como resultado la transformación del algoritmo BKT teniendo en cuenta la programación paralela y distribuida; además, se utilizaron herramientas de procesamiento en paralelo como el marco de trabajo Apache Spark en un entorno de minado. Se valida la propuesta de solución mediante pruebas para medir rendimiento y eficacia, usando métricas como speedup, eficiencia, error medio cuadrático del diferencial de probabilidades y error medio cuadrático del diferencial del área bajo la curva ROC; para las pruebas fueron empleadas bases de datos educacionales.

Palabras clave: Estimación del Conocimiento Latente (ECL); Minería de Datos Educacionales (MDE); Rastreo del Conocimiento Bayesiano (BKT)

Abstract

In view of the massive of data generated in education, traditional methods for the discovery of knowledge have had to be changed. One of the algorithms is the Bayesian Knowledge Tracing (BKT) that allows estimating latent knowledge (ECL). ECL is just the way to measure a student's knowledge about specific skills and concepts, which is evaluated by his or her correction patterns about those skills. This algorithm is designed to be used in small volumes of data, affecting its performance in the presence of large volumes of data. In order to solve the problem, the transformation of the BKT algorithm will be presented, taking into account parallel and distributed programming; in addition, parallel processing tools such as the Apache Spark framework were used in a mining environment. The solution proposal is validated through tests to measure performance and effectiveness, using metrics such as speedup, efficiency, probability differential root mean square error and area differential root mean square error under the ROC curve; educational databases were used for the tests.

Keywords: Bayesian Knowledge Tracking (BKT); Educational Data Mining (EDM); Latent Knowledge Estimation (LKEs)

1. Introducción

La proliferación y el constante desarrollo de plataformas educativas con la utilización de internet, han abierto nuevas posibilidades de análisis debido al gran volumen de datos que se generan y almacenan en servidores. Esta masividad de los datos se debe a las trazas que se van originando a partir de las inter-

acciones con las actividades que proponen las plataformas educativas. Esta información almacenada con datos potenciales de múltiples fuentes procedentes de diferentes formatos y a diferentes niveles de granularidad se caracteriza por ser abundante y accesible. Por lo que, se hace necesario un análisis previo que transforme los datos almacenados en conocimiento útil que permita mejorar el proceso de enseñanza - aprendizaje en dichas plataformas educativas.

La aplicación de la minería de datos en la educación es un campo de investigación interdisciplinario emergente, también conocido como minería de datos (Romero y Ventura, 2013).

Al análisis y exploración de grandes volúmenes de datos en el contexto educativo es llamado Minería de Datos Educacional (MDE) (Martínez, Gutiérrez, Toral y Barrero; 2014), que tiene como objetivo promover nuevos descubrimientos y avances en el terreno educativo mediante el uso de la información almacenadas en las plataformas educativas. Trata de desarrollar métodos para explorar los tipos únicos de datos que provienen de los entornos educativos. Su objetivo es comprender mejor cómo aprenden los estudiantes e identificar los entornos en los que aprenden para mejorar los resultados educativos y comprender y explicar los fenómenos educativos (Romero y Ventura, 2013).

La MDE es un proceso utilizado para extraer información útil y patrones de una enorme base de datos educativa. Esta información útil y los patrones pueden ser utilizados para predecir el desempeño de los estudiantes, lo que ayudaría a los educadores a proporcionar un enfoque de enseñanza eficaz. Los estudiantes podrían mejorar sus actividades de aprendizaje, permitiendo a la administración mejorar el rendimiento de los sistemas (Shahiri, Husain y otros, 2015).

El área de la ECL es de particular importancia dentro de la MDE, debido a que aumentar el conocimiento de los estudiantes es la meta primaria de la educación. Por tanto, si el conocimiento puede ser medido, puedes saber dónde los estás haciendo mejor, puedes informar a los instructores (o cualquier otro interesado en el proceso) sobre el mismo y además puedes realizar decisiones pedagógicas automáticas (Baker y Corbett, 2014), (Larusson y White, 2014)

El BKT determina en qué medida un estudiante conoce una determinada aptitud o habilidad a partir de su rendimiento pasado con esa habilidad. Proporciona un conocimiento sobre habilidades de un sistema y predice comportamientos futuros sobre dichas habilidades, en pos de mejorar los sistemas de enseñanza- aprendizaje (Román, Sánchez-Guzmán y García; 2014). Esta información resulta de gran utilidad para determinar en qué medida una plataforma educativa cumple con su objetivo, para informar a los profesores o incluso para realizar acciones correctoras pedagógicas de manera automática (Martínez, Gutiérrez, Toral y Barrero; 2014).

Dicho algoritmo está diseñado para ser utilizado en volúmenes de datos pequeños, afectándose su rendimiento ante la presencia de grandes volúmenes de datos (Vera, Morales y Soto, 2012); (Pardos, Gowda, Baker y Heffernan, 2012); (Pardos, Bergner, Seaton y Pritchard, 2013), teniendo como limitaciones que:

Los tiempos de cálculo para estimar conocimiento latente son elevados debido al gran cúmulo de información que generan las plataformas.

- El rendimiento del proceso de inferencia de conocimiento estaría condicionado por el volumen de estos datos, por lo que se requeriría de una gran cantidad de recursos de cómputo.
- Al ser el origen de los datos muy diversos, se hace engorrosa la aplicación del algoritmo debido a que hay que adecuar los datos a los parámetros que necesita el algoritmo ya que está pensado para tratar con datos dicotómicos, o sea, resultados absolutos.
- Al estar los datos dispersos en diferentes bases de datos y no tener el mismo formato trae consigo que no se ejecute correctamente el algoritmo, porque por cada formato por separado habrá que estructurar el algoritmo para su utilización.

Por lo que, la utilización de dicho algoritmo en el contexto de grandes volúmenes de datos en la educación constituye un proceso complejo y engorroso porque actualmente si se toma en cuenta que es necesario aplicar nuevos mecanismos que permitan procesar de forma distribuida un algoritmo que no está diseñado para ello. El objetivo de la investigación es adaptar el algoritmo Bayesian Knowledge Tracing utilizando técnicas de programación paralela y distribuida para disminuir los tiempos de ejecución manteniendo la eficacia en la estimación del conocimiento latente en datos educacionales masivos.

2. Materiales y métodos

Dentro de los métodos utilizados en la investigación se encuentra el Histórico - Lógico, donde el empleo de este método permitió realizar un estudio del funcionamiento del algoritmo BKT

Otros de los métodos es la revisión documental, donde se detecta, obtiene y consulta las bibliografías y otros materiales donde se hayan aplicado el algoritmo BKT para la estimación del conocimiento latente. Se extrae y recopila la información relevante y

necesaria que sirva para dar solución a la investigación. Esta revisión debe ser selectiva, puesto que cada año se publican en diversas partes del mundo miles de artículos de revistas, periódicos, libros y otras clases de materiales en las áreas del conocimiento. Con la revisión se logra clasificar las más importantes y recientes literaturas que sirven de referencia al trabajo.

Por último, se utilizó el método de modelado, donde se representa cada uno de los pasos de la propuesta de solución mediante un diagrama para dar una mejor visualización de la solución.

Se propone un modelo para ECL basada en el empleo de la programación distribuida y paralela a través del marco de trabajo Apache Spark para adaptar el algoritmo BKT al contexto de grandes volúmenes de información.

El algoritmo propuesto está compuesto por 5 pasos fundamentales siguiendo el proceso de KDD en la MDE, pero adaptado a la necesidad de cálculo del algoritmo. De forma general el algoritmo parte de la modificación del BKT adaptándolo a las condiciones propias del problema planteado. Todos los pasos del algoritmo sufren modificaciones respecto al algoritmo original. En la Figura se esbozan estos pasos.

Figura 1. Pasos lógicos para la construcción del algoritmo

1. Carga de datos: En este paso se realiza la carga de datos desde las diferentes fuentes que proporcionan datos.
2. Pre-procesar los datos: Se encarga de modificar el conjunto de datos para que estos estén listos para la ejecución del algoritmo propuesto.
3. Ajustar parámetros del algoritmo. El algoritmo propuesto utiliza un grupo de parámetros para su funcionamiento que necesitan ser ajustados antes de poder aplicarlos.
4. Ejecutar estimación en paralelo: Se realiza la estimación para todos los estudiantes en el conjunto de datos de forma paralela.
5. Visualizar resultados: Se visualizan y salvan los resultados obtenidos.

2.2. Descripción de los pasos del algoritmo adaptado

Cargar lo datos

Para comenzar el algoritmo es necesario cargar desde una de las fuentes de datos posibles disponibles para Spark SQL. (Figura 1)

Pre-procesar los datos

Para el correcto funcionamiento del algoritmo es necesario que el conjunto de datos posea la información referente al estudiante, el problema, la habilidad y el valor observado de respuesta a dicho problema. Por tanto, es preciso transformar el conjunto de datos obtenido en el paso anterior para obtener uno que tenga la estructura adecuada (Tabla 1). Para ello se seguirán unas series de pasos para el preprocesamiento de los datos como se muestra en la figura 3.

Quedando de la siguiente manera.

Observación	ID Estudiante	Problema	Habilidad
id_observacion	id_estudiante	id_problema	id_habilidad

Tabla 1. Estructura del conjunto de datos para BKT.

Ajustar parámetros del algoritmo

Para este proceso se hará el ajuste de parámetro con el par estudiante habilidad, donde se ejecutará en paralelo el proceso. (Figura 4)

Ejecutar estimación en paralelo

Una vez terminado el paso anterior del algoritmo, por cada par estudiante-habilidad se posee cuales pará-

metros se deben usar para calcular el conocimiento latente que posee el estudiante para cada habilidad. En el siguiente gráfico se muestra cómo se comportará el flujo de los datos en este método.

Este paso del algoritmo recibe un conjunto de datos con las transacciones de todos los estudiantes-habilidades (estudiante, habilidad, secuencia de observaciones, parámetros con sus valores ajustados). De forma paralela y distribuida se calcula para cada par estudiante - habilidad cuál es el valor de la probabilidad de que se domine dicha habilidad por el estudiante. (Figura 5)

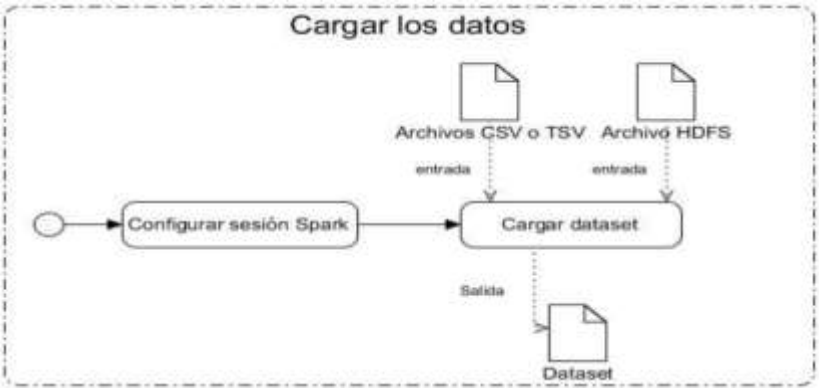

Figura 2. Pasos para cargar los datos

Visualizar resultados

Este paso se encarga de visualizar los resultados derivados del paso anterior. La información se representará en diferentes tipos de gráficos representando conocimientos diversos. Para la visualización se utilizará la biblioteca gráfica JFreeChart. Unos de los gráficos representados es el de habilidades a través de un gráfico de burbuja. En el mismo se visualizará por habilidad la cantidad de estudiantes, la cantidad de problemas y el promedio de dominio de la habilidad para todos los estudiantes (radio de la burbuja). Para poder realizar los gráficos es necesario procesar la información obtenida para adecuarla a su representación visual. En el caso del gráfico de habilidades se obtendrán los parámetros habilidad, cantidad de problemas, cantidad de estudiante y el promedio de habilidad. (Figura 6)

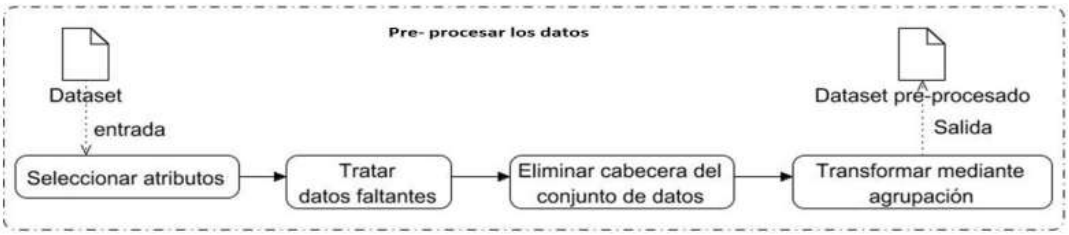

Figura 3. Pre - procesamiento de los datos.

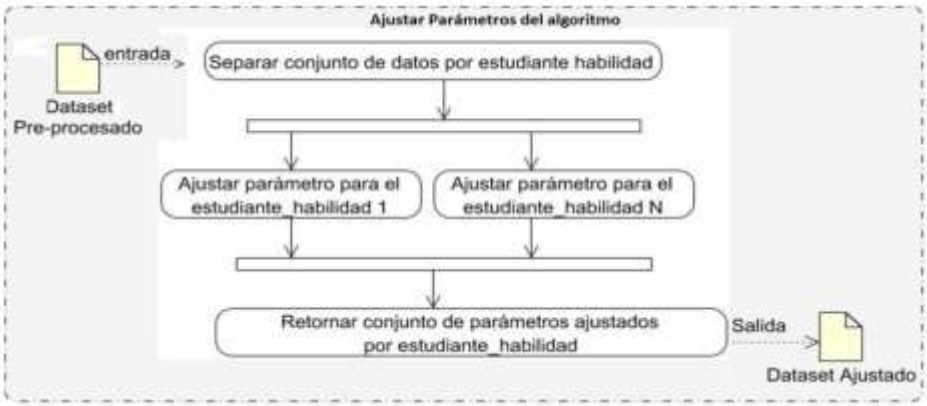

Figura 4. Ajuste de parámetro en paralelo. Fuente: Elaboración propia

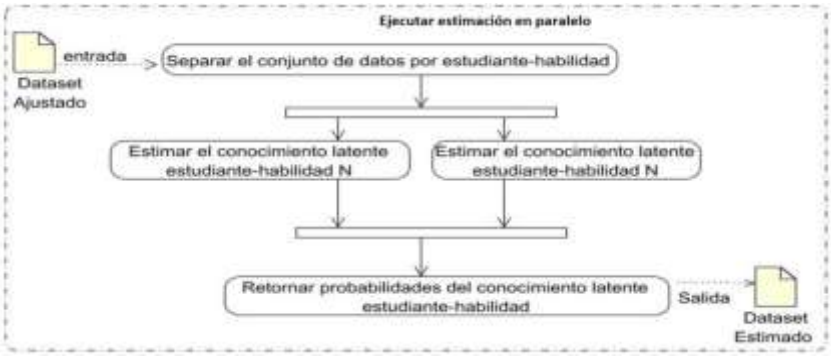

Figura 5. Estimación del conocimiento por estudiante. Fuente: Elaboración propia

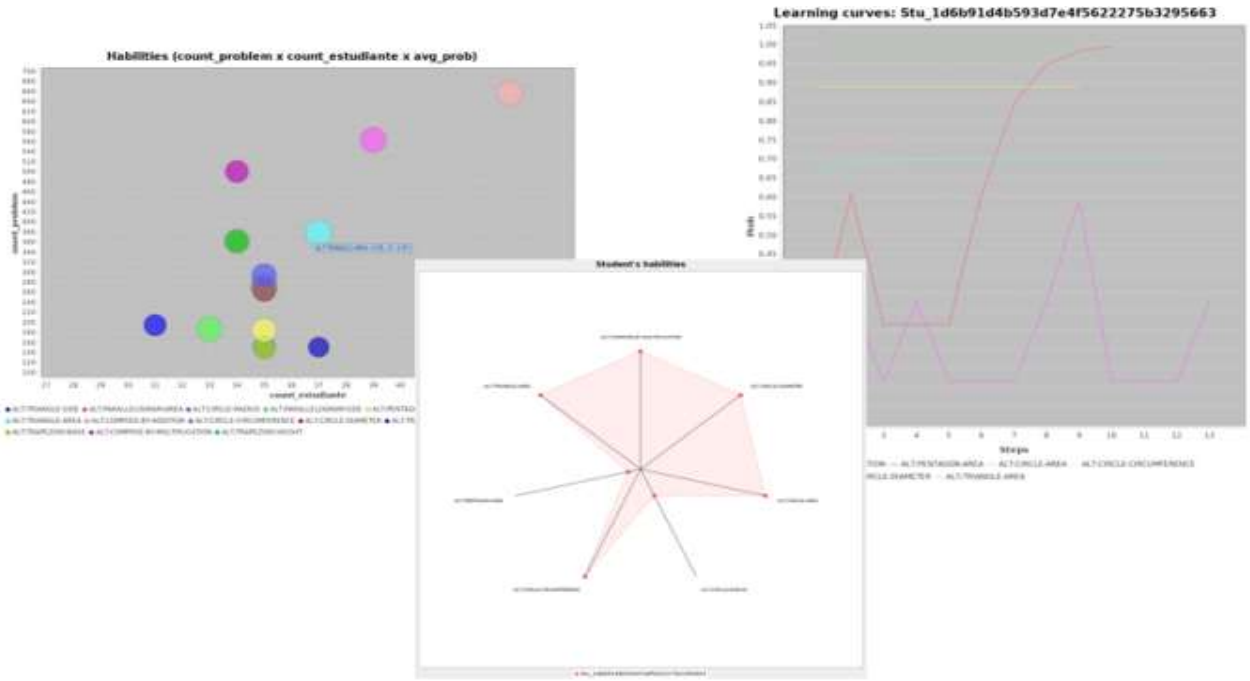

Figura 6. Visualización de los resultados.

De esta forma, se presenta una vista donde se muestra el algoritmo integrado en la capa de servicio REST. (Figura 7)

Una vez concluida la ejecución del algoritmo se obtendrá por cada par de estudiante – habilidad la probabilidad, según se muestra en la figura 8.

Para la validación del algoritmo se utilizaron diferentes conjuntos de datos públicos pertenecientes al repositorio de base de datos educacionales PSLC Datashop, disponible en http://pslcdatashop.org/. Los datos estarán organizados de forma tal que en cada columna se tenga la siguiente información:

First attemp – Valor del resultado del primer intento del estudiante (1 – intento correcto, 0 – intento incorrecto).

Anon Student Id – Identificador del estudiante.

Concatenación de los campos Problem Hierarchy – Problem Name – Step Name. Identifica el problema donde fue aplicada la habilidad.

Knowledge Component – Habilidad a estimar.

Los conjuntos de datos cuentan con las siguientes características descritas en la tabla 2.

Figura 7. Vista general de la propuesta de solución.

Figura 8. Salida del paso ejecutar estimación en el entorno minado (Representación de las primeras 20 filas).

Conjunto de los datos	No. de estudiantes	No. de componentes de conocimiento	No. de problemas	Cantidad de filas
OSU, Honors Physics, Fall 2011 (Dataset 1)	314	154	44	323.912
USNA Physics Fall 2006 (Dataset 2)	66	250	251	345.536
ElemChinese (Dataset 3)	221	22	868	812.329
Assistments Math 2006 – 2007 (5046 Students) (Dataset 4)	5046	318	1872	1451.003

Tabla 2. Características de los datasets.

Para la realización de las pruebas se desplegó la propuesta de solución en un clúster de computadoras usando la herramienta Apache Spark en su versión 2.2.0. Se aplicó el modo Standalone de despliegue, donde se utilizan las funciones nativas de la herramienta. El clúster estaba conformado por una estación de trabajo que actuó como máster y dos estaciones de trabajo utilizadas como nodos trabajadores. Se debe tener en cuenta que las pruebas fueron realizadas en un entorno no dedicado de nodos que pertenecen a la misma subred.

Para ejecutar el algoritmo secuencial se utilizó una computadora con las siguientes características de hardware y software.

Especificaciones de hardware y software para ejecutar el algoritmo secuencial: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz. 8 núcleos. 8Gb DDR3 1600.

Especificaciones de software y sistema operativo: Ubuntu 18.04 LTS 64 bits, versión del núcleo 4.15.0-43-generic. Java OpenJDK versión “8” Update 192. Spark 2.2.0.

Para las pruebas del algoritmo adaptado se utilizó un entorno de minado que consiste de las siguientes características.

Especificaciones de hardware del cluster de computadoras empleado para las pruebas.

Estación de trabajo master: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz. 8 núcleos. 8Gb DDR3 1600.

Estaciones de trabajo usadas como nodos trabajadores: Intel(R) Celeron(R) CPU G1830 @ 2.80GHz. 2 núcleos. 4Gb DDR3. Cantidad: 2. Ancho de banda de la red de datos: 100 Mb/s.

Especificaciones de software y sistema operativo del entorno distribuido: Ubuntu 18.04 LTS 64 bits, versión del núcleo 4.15.0-43-generic. Java OpenJDK versión “8” Update 192. Spark 2.2.0.

En la prueba de los algoritmos secuencial y paralelo se utilizarán 4 dataset de diferentes tamaños. Para el algoritmo secuencial se harán de 10 ejecuciones por cada dataset recogiéndose el tiempo de ejecución y obteniendo la probabilidad de dominio de la habilidad por cada estudiante y el AUC. Lo mismos se hará con el algoritmo paralelo, pero utilizando el entorno minado, con 10 ejecuciones por cada dataset, recogiendo el tiempo de ejecución, la probabilidad de dominio de la habilidad por cada habilidad y el AUC. A partir de esos valores entonces se procederá al cálculo de speedup y eficiencia.

Para la eficacia, se utilizará el Error Cuadrático Medio (ECM), para verificar que no exista diferencia significativa entre los resultados arrojados por el algoritmo secuencial y el paralelo.

Las pruebas realizadas para el algoritmo adaptado con los conjuntos de datos seleccionados en el entorno de minado configurado, los valores obtenidos de aceleración y eficiencia permiten afirmar que el algoritmo adaptado presenta una mejora importante de tiempo de ejecución respecto al algoritmo secuencial.

Figura 9. Resumen del speedup.

Figura 10. Resumen de la eficiencia.

Para comprobar la eficacia del algoritmo adaptado se tomará en cuenta dos métricas a medir que son el Error Cuadrático Medio (ECM) de los valores calculados de la probabilidad de dominio de las habilidades y la ECM del área bajo la curva ROC (AUC). Ambas métricas serán medidas inicialmente por el algoritmo secuencial y luego por el paralelo.

Una vez realizada las pruebas se comprueba que no existen diferencias significativas entre los resultados para los diferentes conjuntos de datos. Lo que demuestra que la eficacia de usar el algoritmo paralelo es similar a la obtenida al usar el algoritmo secuencial. Además, las diferencias entre la métrica de AUC para ambos algoritmos (secuencial y paralelo) nunca es mayor que 0,006598 lo que es un indicador de que la métrica se comporta de forma similar para ambos casos.

Luego de realizado todas las pruebas se puede afirmar que los tiempos de ejecución son mejores para el algoritmo paralelo y la eficacia se mantiene con valores similares para las métricas utilizadas.

3. Conclusiones

El algoritmo BKT adaptado cuenta con 5 pasos fundamentales: carga de los datos, pre-procesamiento, ajuste de parámetros, ejecutar el algoritmo y la visualización, el cual se ejecuta sobre un entorno minado distribuido sobre el marco de trabajo Apache Spark de forma paralela, obteniéndose las probabilidades por habilidades de cada estudiante a partir de la secuencia de observaciones por cada habilidad.

Con la ejecución de los algoritmos sobre los 4 dataset de diferentes tamaños se pudo afirmar que el algoritmo adaptado presenta una mejora importante de tiempo de ejecución respecto a la aceleración y eficiencia con el algoritmo secuencial. Así mismo, la eficacia mantiene valores similares para las métricas del ECM utilizadas entre las probabilidades de dominio de las habilidades y el AUC.

4. Referencias bibliográficas

Baker and Corbett. (2014) *Assessment of robust learning with educational data mining* Res. Pract. Assess., vol. 9.

JFreeChart (2017) *JFree Chart*. 2005-2017, [Online]. Available: http://www.jfree.org/jfreechart/index.html

Larusson and White (2014) *Learning Analytics*. Springer.

Martínez Torres, Gutiérrez Reina, Toral, and Barrero (2014) *Metodologías de análisis de los big data en las plataformas educativas*. pp. 79–83.

Pardos, Bergner, Seaton, and Pritchard (2013) *Adapting Bayesian Knowledge Tracing to a Massive Open Online Course in edX.* in EDM, pp. 137–144.

Pardos, Gowda, Baker, and Heffernan (2012) *The sum is greater than the parts:ensembling models of student knowledge in educational software*. ACM SIGKDD Explor. Newsl., vol. 13, no. 2, pp. 37–44.

Román, Sánchez-Guzmán, and García (2014) *Minería de datos educativa: Una herramienta para la investigación de patrones de aprendizaje sobre un contexto educativo*. Lat. Am. J. Phys. Educ. Vol, vol. 7, no. 4, p. 662.

Romero and Ventura (2013) *Data mining in education*. Wiley Interdiscip. Rev. Data Min. Knowl.Discov., vol. 3, no. 1, pp. 12–27.

Shahiri, Husain, and others (2015) *A review on predicting student's performance using datamining techniques*. Procedia Comput. Sci., vol. 72, pp. 414–422.

Vera, Morales, and Soto (2012) *Predicción del fracaso escolar mediante técnicas de minería de datos*. Rev. Iberoam. Tecnol. del/da Aprendizaje/Aprendizagem, vol. 109.

Fecha de recepción: 30 de septiembre de 2019

Fecha de aceptación: 14 de noviembre de 2019