/UCIENCIA/.21
IV Conferencia Científica Internacional
Universidad de las Ciencias Informáticas

Temática: IV Taller internacional de Enseñanza de las Ciencias Informáticas.

Estimar conocimiento latente en grandes volúmenes de datos utilizando el algoritmo Bayesian Knowledge Tracing

Estimating latent knowledge in large volumes of data using the Bayesian Knowledge Tracing algorithm

Lisset Salazar Gómez [1*], Angel Alberto Vazquez Sánchez [2] , Roxana Cañizares González[3]

[1] Universidad de las Ciencias Informática. Km 2 ½ carretera de San Antonio, Torrens, La Habana. lsgomez@uci.cu
[2] Universidad de las Ciencias Informática. Km 2 ½ carretera de San Antonio, Torrens, La Habana. aavazquez@uci.cu
[3] Universidad de las Ciencias Informática. Km 2 ½ carretera de San Antonio, Torrens, La Habana. rcanizares@uci.cu

[*] Autor para correspondencia: lsgomez@uci.cu

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. 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
Given the massive amount of data generated in education, the traditional methods for knowledge discovery have had to be changed. One of the algorithms is the Bayesian Knowledge Tracing (BKT) that allows to Estimate Latent Knowledge (LKE). LKE is nothing more than a way of measuring a student's knowledge about specific skills and concepts, which is evaluated by his or her correction patterns on those skills. This algorithm is designed to be used on 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 as a result, taking into account parallel and distributed programming. Parallel processing tools such as the Apache Spark framework were used in a mining environment. The proposed solution is validated through tests to measure performance and effectiveness, using metrics such as speedup, efficiency, mean squared error of the differential of probabilities and mean squared error of the differential of the area under the ROC curve; educational databases were used for the tests.

Keywords: *Latent Knowledge Estimation (LKE), Educational Data Mining (EDM), Bayesian Knowledge Tracking (BKT).*

Introducción

La aplicación de la minería de datos en la educación es un campo de investigación interdisciplinario emergente, también conocido como minería de datos (Cristobal Romero & Ventura, 2013). Al análisis y exploración de grandes volúmenes de datos en el contexto educativo es llamado Minería de Datos Educacional (MDE)(Martinez Torres et al., 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(Cristobal Romero & 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 et al., 2015).

En (Cristobal Romero & Ventura, 2013),(Cristóbal Romero & Ventura, 2010), (R. S. J. D. Baker & Yacef, 2009) y (Cristobal Romero et al., 2010) la definen como una disciplina emergente, preocupada por el desarrollo de métodos para explorar los tipos únicos de datos que provienen de entornos educativos y utilizan esos métodos para comprender mejor a los estudiantes y los entornos en los que aprenden. Otros autores reconocidos en el área (Cristóbal Romero & Ventura, 2010) lo definen como un campo que explota algoritmos estadísticos, de aprendizaje de máquina y de minería de datos sobre los diferentes tipos de datos educativos. MDE busca utilizar estos repositorios de datos para entender mejor a los estudiantes y al aprendizaje, y a desarrollar enfoques computacionales que combinan datos y teoría para transformar la práctica en beneficio para los aprendices. Un análisis realizado en el área, arroja que el número de publicaciones en la MDE ha aumentado exponencialmente desde el 2005, siendo el mayor pico en el 2014

(Aristizábal Fúquene, 2017), manteniéndose los avances de investigación hasta la fecha. Específicamente se ha dedicado un gran número de las investigaciones hacia la predicción, el cual permite predecir el rendimiento de un estudiante(Cristóbal Romero & Ventura, 2010).

Dentro de la MDE, se utilizan diferentes técnicas de minería de datos, de ellas están las predictivas que tienen como objetivo predecir comportamiento de un aspecto de los datos (Larusson & White, 2014). Las técnicas predictivas se basan esencialmente en regresiones y clasificaciones supervisadas. Estas técnicas se han empleado con éxito para crear modelos de predicción del rendimiento de los estudiantes, en el que se han obtenido resultados prometedo res que demuestran cómo determinadas características sociológicas, económicas y educativas de los alumnos pueden afectar en el rendimiento académico (Carlos Márquez Vera, 2012). Normalmente estas técnicas se aplican para predecir el rendimiento académico de los alumnos. Por ejemplo, establecer modelos predictores sobre el índice de aprobados de un curso, sobre su nota media, o predecir el tiempo que un estudiante tardará en completar una tarea. También pueden utilizarse para predecir si el alumno posee una determinada competencia sobre una habilidad. A esto último es lo que se conoce como Estimación de Conocimiento Latente (ECL), llamado así porque el conocimiento no es una variable directamente observable (Martinez Torres et al., 2014).

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 (R. S. Baker & Corbett, 2014) (Larusson & White, 2014).

Inferir el conocimiento de los estudiantes puede ser útil para varios objetivos, por ejemplo, puede ser una entrada significativa para otros tipos de análisis (Aristizábal Fúquene, 2017). Puede ser útil para decidir cuándo avanzar un estudiante en el currículo o intervenir en otras vías (Roll et al., 2007) ,y además puede constituir información útil para los instructores (Feng & Heffernan, 2007).

Existen diferentes métodos que permiten ECL que surgen en el ambiente de la MDE (Feng & Heffernan, 2007), entre los que se encuentran Análisis de los Factores de Rendimiento (PFA, *Performance Factors Analysis*), Teoría de Respuesta al Ítem (IRT, *Item Response Theory*) y el Rastreo del Conocimiento Bayesiano (BKT, *Bayesian Knowledge Tracing*). Cada uno de los algoritmos trata la capacidad de estimar el conocimiento latente de diferentes maneras (Larusson & White, 2014). En el caso del IRT, predice la respuesta de una persona ante un ítem determinado. El PFA no es una expresión directa de la cantidad de habilidad latente, excepto por la probabilidad de responder

correctamente. Por otro lado, el BKT expresa la probabilidad de que el estudiante domine la habilidad latente, y además muestra la probabilidad de que el estudiante responda correctamente la próxima vez que enfrente un problema donde deba aplicar dicha habilidad. Por lo que se diferencian en la manera de ECL, siendo el BKT el único algoritmo que puede determinar la probabilidad de dominio sobre una habilidad.

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 (Feng & Heffernan, 2007). 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 (Martinez Torres et al., 2014).

Dicho algoritmo está diseñado para ser utilizado en volúmenes de datos pequeños, afectándose su rendimiento ante la presencia de grandes volúmenes de datos (Carlos Márquez Vera, 2012), (Ballesteros Román et al., 2013), (Pardos et al., 2013) . Por lo que se plantea como objetivo de la investigación 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.

Materiales y métodos

Algoritmo BKT

Para predecir $P(L_j)$ para un estudiante individual se puede emplear el algoritmo de seguimiento del conocimiento. En este caso se busca $P(L_j \vee O_j)$, la probabilidad de que el estudiante ha aprendido la habilidad justamente luego de completar el paso j dado el rendimiento del estudiante O_j en los pasos previos, donde $O_j = \{o_1, o_2, \ldots, o_j\}$ es el rendimiento del estudiante en las primeras j oportunidades y o_i puede ser correcto o incorrecto (van De Sande, 2013). Estas probabilidades condicionales responden a la recurrencia:

$$P(L_{j-1}|O_j) = \frac{P(L_{j-1}|O_{j-1})(1-P(S))}{P(L_{j-1}|O_{j-1})(1-P(S)) + \left[1-P(L_{j-1}|O_{j-1})\right]P(G)}, O_j = correcto \quad (1)$$

$$P(L_{j-1}|O_j) = \frac{P(L_{j-1}|O_{j-1})P(S)}{P(L_{j-1}|O_{j-1})P(S) + \left[1-P(L_{j-1}|O_{j-1})\right](1-P(G))}, O_j = incorrecto \quad (2)$$

$$P(L_j|O_j) = P(L_{j-1}|O_j) + \left[1-P(L_{j-1}|O_j)\right]P(T) \quad (3)$$

Existen una variedad de algoritmos que permiten realizar el ajuste de los parámetros necesarios para realizar la estimación del conocimiento latente del algoritmo BKT. Entre estos algoritmos se encuentran:

- Maximizar la Expectación (EM, por sus siglas en inglés)(Dellaert, 2002), (van De Sande, 2013).
- Fuerza Bruta (BF, por sus siglas en inglés) (d Baker et al., 2008), (Yudelson et al., 2013).
- Probabilidad Empírica (EP, por sus siglas en inglés) (Hawkins et al., 2014)
- Ajuste Aleatorio (RF, por sus siglas en inglés).
- Recocido Simulado (SAF, por sus siglas en inglés) (Miller et al., 2014).
- Baum-Welch (BW, por sus siglas en inglés) (Shen, 2008)

Características del algoritmo Bayesian Knowledge Tracing

El algoritmo BKT está regido por las expresiones 1 y 2. Estas expresiones en su forma funcional se pueden apreciar en la Figura 1. En la misma la línea puntuada representa la frontera entre soluciones crecientes y decrecientes. Desde que la curva "step j correct" está por encima de la línea puntuada, la ecuación 1 provoca que la secuencia converge al punto fijo en 1. De igual manera, desde que la curva "step j incorrect" está por debajo de la línea, la ecuación 1 causa que la secuencia $\{P(L_j \vee O_j)\}$ converge al punto fijo menor.

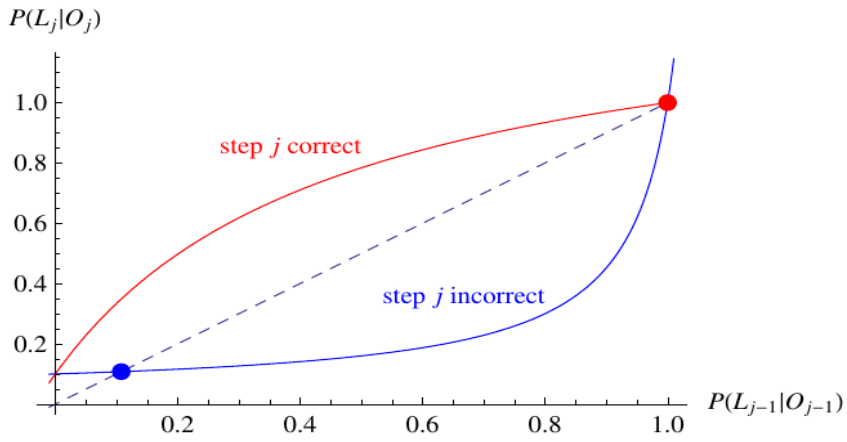

Figura 1 Gráfico de la relación de recurrencia para el algoritmo BKT, de las ecuaciones (1) y (2). Los parámetros de este modelo son P(S)=0.5, P(G)=0.3, P(T)=0.1 y P(L0) = 0.36. Fuente: (van De Sande, 2013).

Esta relación recurrencia (ecuaciones 1 y 2) no puede ser resuelta analíticamente, se puede aprender mucho de sus soluciones mediante la realización de un análisis de punto fijo (van De Sande, 2013). El objetivo de un análisis de punto fijo es determinar la calidad del comportamiento de la secuencia $\left\{P\left(L_j \vee O_j\right)\right\}_{j=0}^{n}$ como una función de j. Si $P\left(L_j|O_j\right)>P\left(L_{j-1} \vee O_{j-1}\right)$, entonces se puede decir que crece con j. Por otro lado, si $P\left(L_j|O_j\right)<P\left(L_{j-1} \vee O_{j-1}\right)$, entonces se puede decir que decrece con j. Por lo que, la frontera entre soluciones crecientes y decrecientes es $P\left(L_j|O_j\right)=P\left(L_{j-1} \vee O_{j-1}\right)$, mostrada como la línea punteada en la Figura 1. Un punto fijo es un valor de $P\left(L_j \vee O_j\right)$ tal que la relación de recurrencia obedece $P\left(L_j|O_j\right)=P\left(L_{j-1} \vee O_{j-1}\right)$.

En (van De Sande, 2013) también se definen dos tipos de puntos fijos :

- Punto fijo estable: Si $P\left(L_j|O_j\right)$ está cerca del punto fijo, entonces $P\left(L_j|O_j\right)$ converge al punto fijo mientras j se incrementa.
- Punto fijo inestable: Si $P\left(L_j|O_j\right)$ está cerca del punto fijo, entonces $P\left(L_j|O_j\right)$ se aleja del punto fijo al incrementarse j.

Aplicando estas ideas en las ecuaciones 1 y 2. En la ecuación 1 se encuentra un punto fijo estable en 1 y un punto fijo inestable en:

$$\frac{-P(G)P(T)}{1-P(G)-P(S)} \tag{4}$$

Similarmente, la ecuación 2 tiene un punto fijo inestable en 1 y un punto fijo estable en:

$$\frac{(1-P(G))P(T)}{1-P(S)-P(G)} \tag{5}$$

Para que $P\left(L_j|O_j\right)$ permanezca en el intervalo [0,1] para cualquier valor inicial de $P\left(L_0\right) \in [0,1]$ y cualquier secuencia de pasos O_j correctos/incorrectos, necesitamos que el punto fijo (5) esté en el intervalo [0,1] y el punto inestable (4) se mantenga negativo. Esto nos da las siguientes restricciones en los valores permitidos para el modelo de los parámetros (Beck et al., 2008):

$$P(G)+P(S)<1 \tag{6}$$

$$0<P(T)<1-\frac{P(S)}{1-P(G)} \tag{7}$$

Otras restricciones propuestas para este modelo son:

- $P(S)<0.5$ y $P(G)<0.5$ (d Baker et al., 2008)
- $P(S)<0.1$ y $P(G)<0.3$ (Corbett & Anderson, 1994)

La idea conceptual detrás de este algoritmo es:

- Dominar una habilidad generalmente conlleva a un rendimiento correcto.
- Un buen rendimiento implica que un estudiante domina la habilidad relevante.

Por tanto, mediante la búsqueda de dónde el estudiante muestra un buen rendimiento se puede inferir que domina la habilidad (S. Baker, 2015).

Para el experimento se utilizarán 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:

Tabla 1 Características de los datasets.

Conjunto de datos	Número de estudiantes	Número de componentes de conocimiento	Número de problemas	Cantidad de filas
OSU, Honors Physics: Mechanics, Fall 2011 (dataset 1)	314	154	44	323,912
USNA Physics Fall	66	250	251	345,536

2006 (dataset 2)				
ElemChinese (dataset 3)	221	22	868	812,329
Assistments Math 2006-2007 (5046 Students) (dataset 4)	5,046	318	1872	1,451,003

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 independiente (*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.

Tabla 2 Especificaciones de hardware y software para ejecutar el algoritmo secuencial.

Estación de trabajo usada para ejecutar el algoritmo secuencial		
Tipo de procesador	Cantidad de núcleos	Memoria principal
Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz	8	8 Gb DDR3 1600

Tabla 3 Especificaciones de software y sistema operativo.

Sistema operativo
Ubuntu 18.04 LTS 64 bits, versión del núcleo: 4.15.0-43-generic
Software necesarios instalados
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:

Tabla 4 Especificaciones de hardware del clúster de computadoras empleado para las pruebas.

Estación de trabajo máster		
Tipo de procesador	Cantidad de núcleo	Memoria principal
Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz	8	8 Gb DDR3 1600
Estaciones de trabajo usadas como nodos trabajadores		
Tipo de procesador	Cantidad de núcleo	Memoria principal
Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	2	4 Gb DDR3
Intel(R) Celeron(R) CPU G1830 @ 2.80GHz	2	4 Gb DDR3
Características de la red de datos		
100 Mb/s		

Tabla 5 Especificaciones de software y sistema operativo del entorno distribuido.

Sistema operativo
Ubuntu 18.04 LTS 64 bits, versión del núcleo: 4.15.0-43-generic
Software necesarios instalados
Java OpenJDK versión "8" Update 192
Spark 2.2.0

En la prueba de los algoritmos secuencial y paralelo se utilizarán 4 conjunto de datos (*dataset*) de diferentes tamaños. Para el algoritmo secuencial se harán de 10 ejecuciones por cada conjunto de datos recogiéndose el tiempo de ejecución yobteniendo la probabilidad de dominio de la habilidad por cada estudiante y el AUC (Área Bajo la Curva). Lo mismos se hará con el algoritmo paralelo, pero utilizando el entorno minado, con 10 ejecuciones por cada conjunto de datos, 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 aceleración (en lo adelante *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.

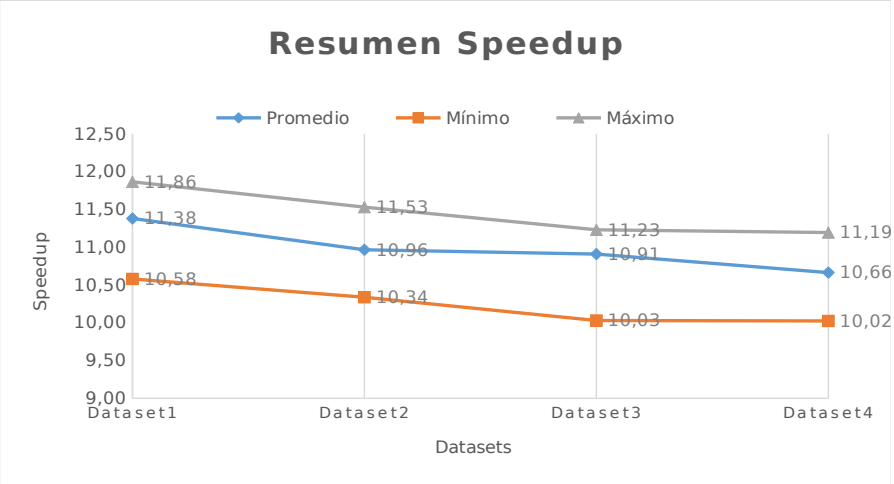

Figura 2 Resumen del speedup. Fuente: Elaboración propia.

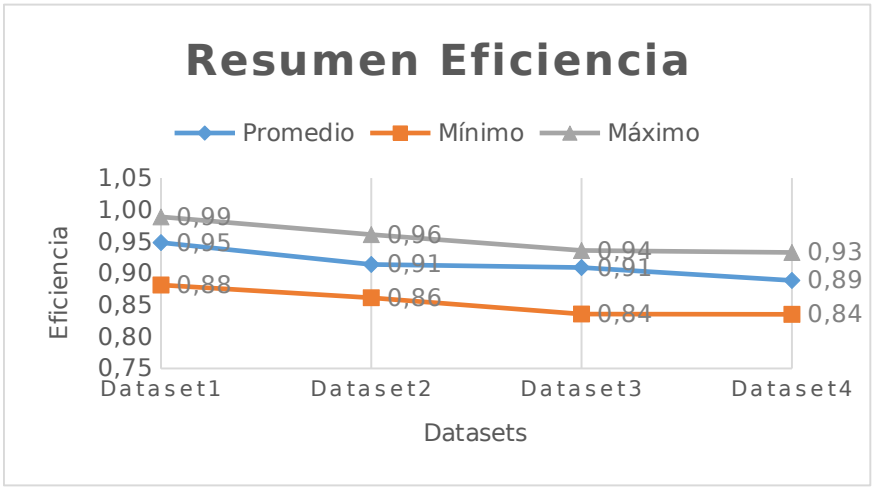

Figura 3 Resumen de la eficiencia. Fuente: Elaboración propia.

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.

Resultados y discusión

El algoritmo propuesto utiliza los siguientes elementos para su funcionamiento:

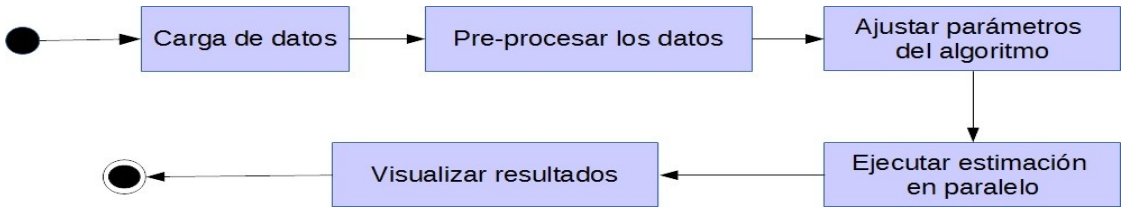

Figura 4 Pasos lógicos para la construcción del algoritmo. Fuente: Elaboración propia.

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.

Carga de datos

En el primer paso, se tiene como opciones de orígenes de datos a:

- Archivos CSV o TSV.
- Hadoop Distributed File System (HDFS).

Por tanto, este procedimiento recibe como entrada el tipo de origen de los datos y los datos necesarios para cargarlo y consta de los siguientes pasos (ver Figura 5):

1. Configuración del SparkSession a través del objeto SparkConf.
2. A partir del SparkSession y del origen de datos, se obtiene el conjunto de datos (dataset) a utilizar.
3. El conjunto de datos obtenido es devuelto a la ejecución central del algoritmo.

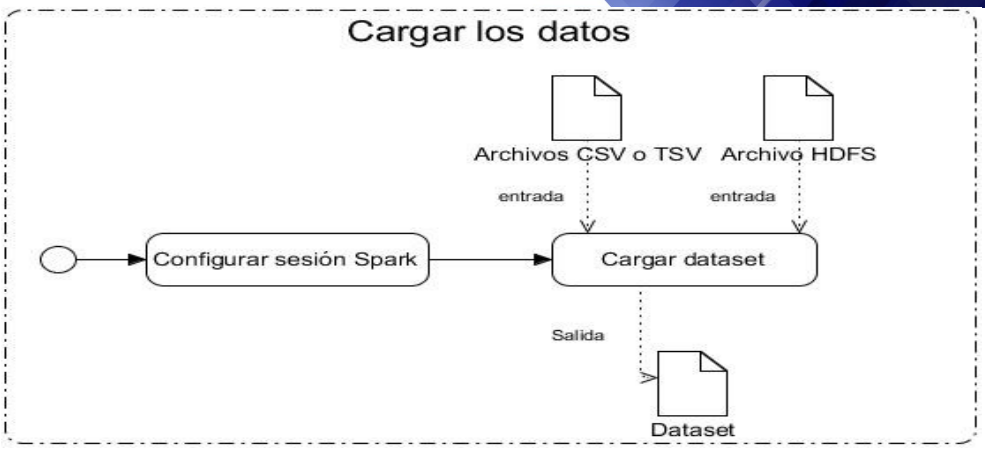

Figura 5 Paso cargar los datos. Fuente: Elaboración propia.

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 6). Para ello se seguirán una series de pasos para el pre-procesamiento de los datos (ver Figura 6).

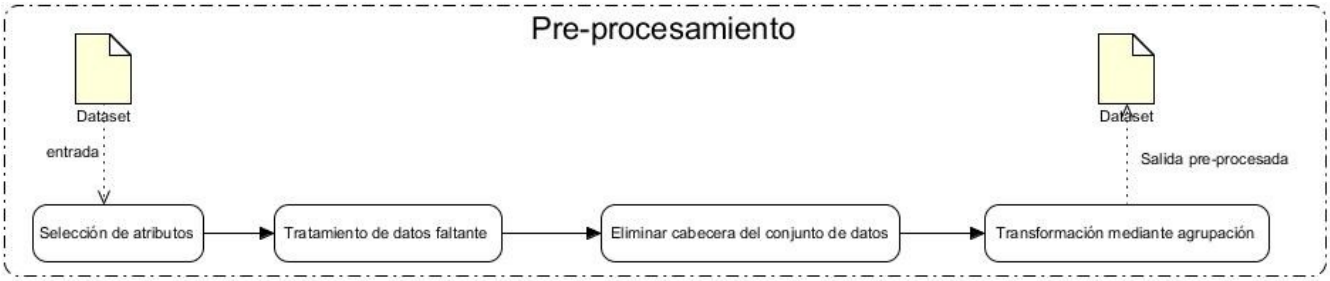

Figura 6 Pre-procesamiento de los datos. Fuente: Elaboración propia.

Este proceso de transformación el conjunto de datos debe tener el siguiente formato(Yudelson et al., 2013):

Tabla 6 Estructura conjunto de datos para BKT.

Observación	ID Estudiante	Problema	Habilidad
id_observación	id_estudiante	id_problema	id_habilidad

Donde

- id_observación – Observación de la respuesta (correcto, incorrecto).
- id_estudiante – Identificación del estudiante.

- id_problema – Concatenación de jerarquía del problema – nombre del problema – nombre del paso.
- id_habilidad – Habilidad o componente de conocimiento a medir.

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 (ver Figura 7).

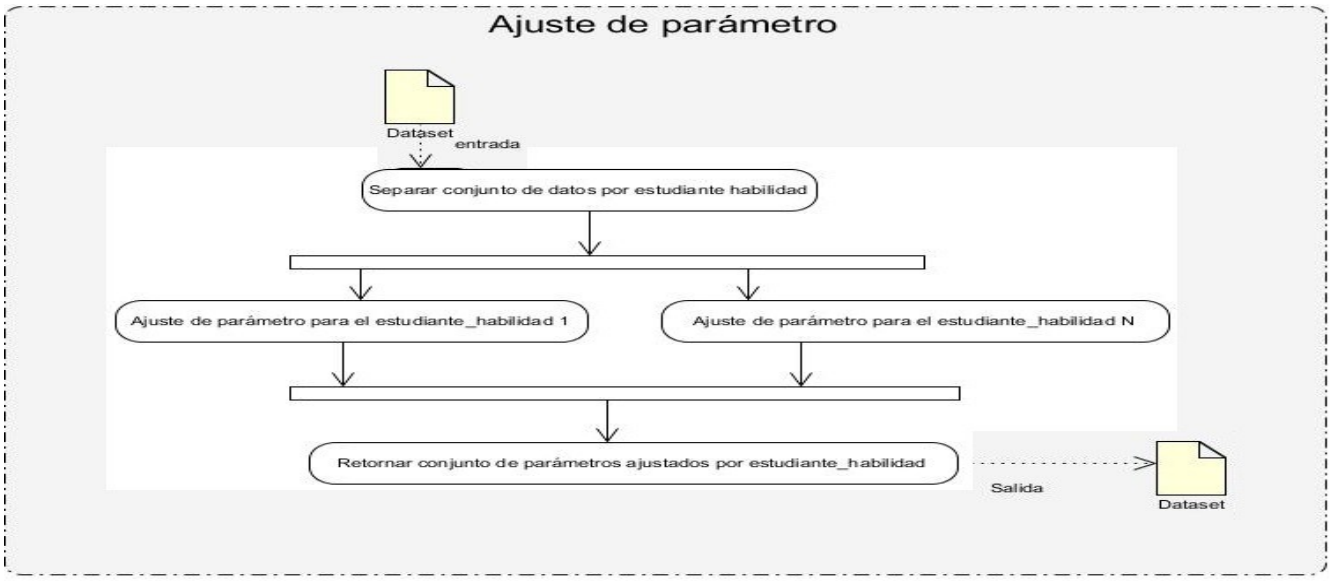

Figura 7 Ajuste de parámetro en paralelo. Fuente: Elaboración propia.

Este algoritmo es un proceso de dos pasos que involucra anotar los datos del rendimiento con conocimiento y usar esta información para computar los parámetros de BKT (Hawkins et al., 2014).

Ejecutar estimación en paralelo

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

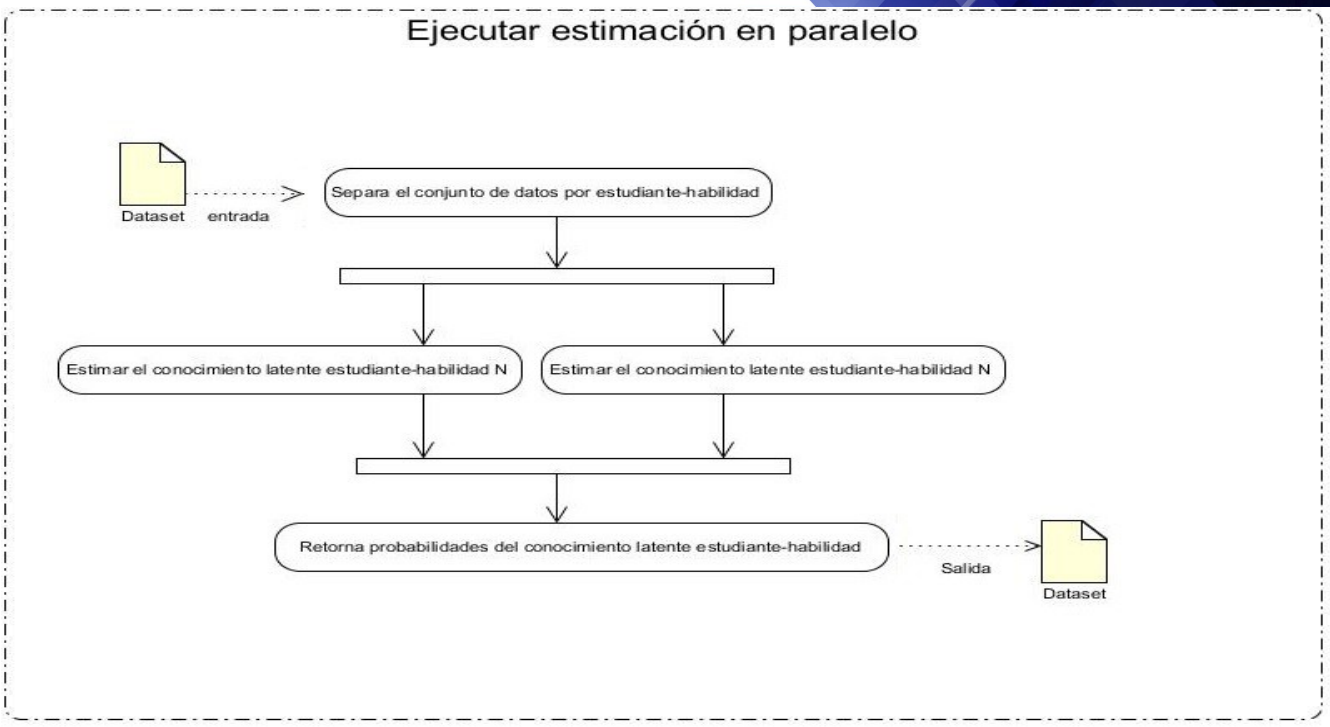

Figura 8 Estimación del conocimiento por estudiante. Fuente: Elaboración propia.

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[1].

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 (ver Figura 9, solamente se representan en la imagen los primeros 20 resultados).

1 http://www.jfree.org/jfreechart/

habilidad	count_problem	count_estudiante	avg_prob
ALT:TRIANGLE-SIDE	194	31	0.8336255768469941
ALT:PARALLELOGRAM...	806	45	0.9799971295481011
ALT:CIRCLE-RADIUS	293	35	0.9181172627394458
ALT:PARALLELOGRAM...	186	33	0.9393325876365023
ALT:PENTAGON-AREA	184	35	0.904085610627396
ALT:CIRCLE-AREA	563	39	0.9708077508660576
ALT:TRIANGLE-AREA	378	37	0.969936064282016
ALT:COMPOSE-BY-AD...	656	44	0.9445138639237194
ALT:CIRCLE-CIRCUM...	271	35	0.9777442011276605
ALT:CIRCLE-DIAMETER	264	35	0.8790610871881608
ALT:TRAPEZOID-AREA	150	37	0.7791070824432661
ALT:PENTAGON-SIDE	361	34	0.9083701625477026
ALT:TRAPEZOID-BASE	147	35	0.7918728262895418
ALT:COMPOSE-BY-MU...	500	34	0.8614983992246082
ALT:TRAPEZOID-HEIGHT	151	35	0.8370217666886987

Figura 9 Datos para representar en el gráfico de burbuja. Fuente: Elaboración propia.

Una vez obtenido los datos a representar, entonces se visualiza el gráfico de burbuja. En el mismo se muestra en cada burbuja la etiqueta que representa la información sobre la cantidad de problema, la cantidad de estudiantes y el promedio de probabilidad (ver Figura 10).

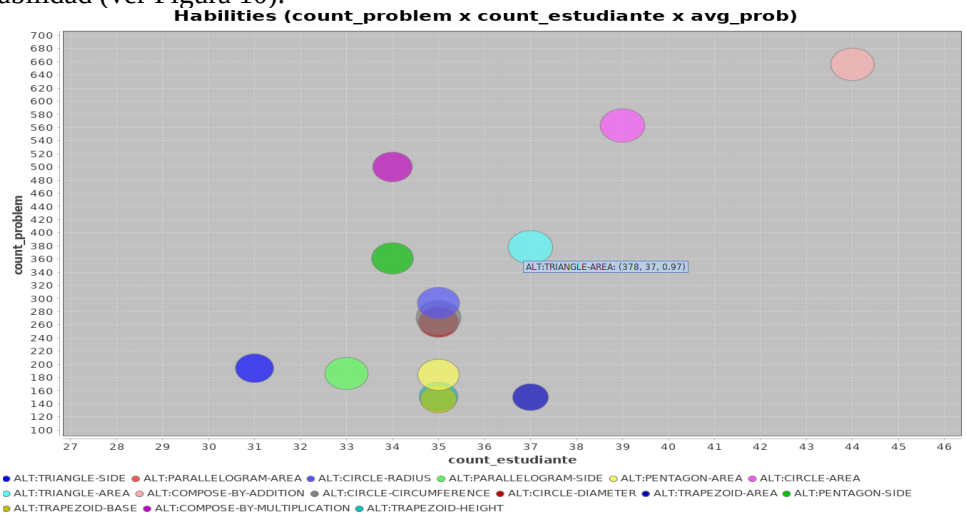

Figura 10 Ejemplo de un gráfico de burbuja de las habilidades. Fuente: Elaboración propia.

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 importantede 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.

Referencias

Aristizábal Fúquene, J. A. (2017). *Diseño y aportes de un modelo para miner{\'\i}a de datos educativos en aulas de educación media de carácter presencial*.

Baker, R. S., & Corbett, A. T. (2014). Assessment of Robust Learning with Educational Data Mining. *Research & Practice in Assessment*, *9*, 38–50.

Baker, R. S. J. D., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM| Journal of Educational Data Mining*, *1*(1), 3–17.

Ballesteros Román, A., Sánchez-Guzmán, D., & Garc\'\ia Salcedo, R. (2013). Miner{\'\i}a de datos educativa: Una herramienta para la investigación de patroness de aprendizaje sobre un contexto educativo. *Latin-American Journal of Physics Education*, *7*(4).

Beck, J. E., Chang, K., Mostow, J., & Corbett, A. (2008). Does help help? Introducing the Bayesian Evaluation and Assessment methodology. *International Conference on Intelligent Tutoring Systems*, 383–394.

Carlos Márquez Vera, C. R. (2012). *Predicción del Fracaso Escolar Mediante Técnicas de Miner{\'\i}a de Datos*. Obtenido de http://rita. det. uvigo. es/201208/uploads/IEEE-RITA.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278.

d Baker, R. S. J., Corbett, A. T., & Aleven, V. (2008). More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. *International Conference on Intelligent Tutoring Systems*, 406–415.

Dellaert, F. (2002). *The expectation maximization algorithm*.

Feng, M., & Heffernan, N. T. (2007). Towards live informing and automatic analyzing of student learning: Reporting in ASSISTment system. *Journal of Interactive Learning Research*, *18*(2), 207–230.

Hawkins, W. J., Heffernan, N. T., & Baker, R. S. J. D. (2014). Learning Bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. *International Conference on Intelligent Tutoring Systems*, 150–155.

Larusson, J. A., & White, B. (2014). *Learning analytics: From research to practice* (Vol. 13). Springer.

Martinez Torres, M. del R., Gutiérrez Reina, D., Toral, S. L., & Barrero, F. (2014). Metodologias de análisis de los big data en las plataformas educativas. *TAEE 2014: XI Congreso de Tecnolog{\'\i}as, Aprendizaje y Enseñanza de La Electrónica XI Congreso de Tecnolog{\'\i}as, Aprendizaje y Enseñanza de La Electrónica: Libro de Actas, Bilbao 11-12 y 13 de Junio de 2014 (2014), p 79-83*.

Miller, W. L., Baker, R. S., & Rossi, L. M. (2014). Unifying computer-based assessment across conceptual instruction, problem-solving, and digital games. *Technology, Knowledge and Learning*, *19*(1–2), 165–181.

Pardos, Z. A., Bergner, Y., Seaton, D. T., & Pritchard, D. E. (2013). Adapting Bayesian Knowledge Tracing to a Massive Open Online Course in edX. *EDM*, *13*, 137–144.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2007). Can Help Seeking Be Tutored? Searching for the Secret Sauce of Metacognitive Tutoring. *AIED*, *2007*, 203–210.

Romero, Cristobal, & Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *3*(1), 12–27.

Romero, Cristóbal, & Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(6), 601–618.

Romero, Cristobal, Ventura, S., Pechenizkiy, M., & Baker, R. Sj. (2010). *Handbook of educational data mining*. CRC press.

S. Baker, R. (2015). *Big Data and Education*. Teachers College, Columbia University.

Shahiri, A. M., Husain, W., & others. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, *72*, 414–422.

Shen, D. (2008). Some mathematics for HMM. *Massachusetts Institute of Technology*.

van De Sande, B. (2013). Properties of the Bayesian Knowledge Tracing Model. *Journal of Educational Data Mining*, *5*(2), 1–10.

Yudelson, M. V, Koedinger, K. R., & Gordon, G. J. (2013). Individualized bayesian knowledge tracing models. *International Conference on Artificial Intelligence in Education*, 171–180.