Vol.2020-CLE-32 No.3
2020/11/27

デジタル教科書のログ分析効率化に向けたオープンソースライブラリ「OpneLA」の開発と適用事例の紹介

村田 隆介[1,a)]　島田 敬士[2,b)]　峰松 翼[2,c)]　谷口 倫一郎[2,d)]

概要：近年，デジタル教科書の操作ログを分析することで教員や学習者にフィードバックを返し，教育・学習を改善するための研究が多数行われている．デジタル教科書の操作ログは分析のために集計や加工といった事前処理が必要であり，これは様々な研究で共通する部分が多い．それにも関わらず，分析のための事前処理部分を各研究者が別個に開発しているのが実情である．そこで本研究では，事前処理部分の冗長な開発を軽減し先進技術の効率的な開発を支援するオープンソースライブラリ OpenLA を開発する．OpenLA は基本情報の取得，必要な情報の抽出，適した形へのデータ整形，データの可視化に関する関数群を実装している．本稿では OpenLA が提供する機能と，デジタル教科書のログ分析に関する既存研究への適用例を紹介する．

1. はじめに

近年の情報化社会の進展，加えて新型コロナウイルス感染症対策の影響により，教育支援システムの導入が加速している．教育支援システムは学習活動に関する大量のデータを収集できるので，教育ビッグデータの分析によって教育・学習の改善を目指すラーニング・アナリティクス (Learning Analytics，LA，学習分析) に一層の注目が集まっている．一般的な教育支援システムには学習管理システムやポートフォリオシステムがあるが，タブレット端末やノート PC などの普及によりデジタル教科書システムの活用事例も増えている．

デジタル教科書システムは基本的に，教材へのアクセスやページ遷移，各ページへのマーカー（ハイライト）やメモ，ブックマークといった操作とその時刻を記録する．このような学習活動の記録を利用した研究成果として，学習行動の分析 [1] や学習行動パターンの発見 [2]，成績予測 [3]，注意が必要な学生の発見 [4] などが報告されている．これらの研究に取り組むためには，デジタル教科書システムに記録された操作ログを学習時間や各ページへの注目度といった情報へ集計・加工する必要がある．こうした事前処理には多くの共通点が存在しているにもかかわらず，これまでのところ各研究者が独自に開発・実装を行っている．

そこで本研究では，事前処理でしばしば利用される計算処理の冗長な開発を軽減し，より先進的なラーニングアナリティクス技術開発を支援するためのオープンソースライブラリ「OpenLA: An Open-Source Library for e-Book Log Analytics」を開発する．OpenLA は Python 言語によって記述されており，Scikit-learn や Tensorflow といった Python ライブラリとの親和性がある．本稿では，ライブラリが提供する API 関数群の紹介ならびに実際の利用例について述べる．より詳細な情報は Web サイト[*1] に掲載しているのでそちらも参照されたい．

2. OpenLA ライブラリ

2.1 データセット

OpenLA の利用には，LAK Data Challenge Workshop で提供されているデータセットと同形式のものが必要である．公開データセットには以下の 4 種類のファイルが含まれている．

Course_#_EventStream.csv
　デジタル教科書の操作履歴が記録されたファイル．

Course_#_LectureMaterial.csv
　各デジタル教科書のページ数が記録されたファイル．

Course_#_LectureTime.csv
　各界の授業が実施された時間が記録されたファイル．

Course_#_LectureMaterial.csv
　各学生の最終成績が記録されたファイル．

[1] 九州大学 工学部 電気情報工学科
[2] 九州大学 大学院システム情報科学研究院
[a)] murata@limu.ait.kyushu-u.ac.jp
[b)] atsushi@ait.kyushu-u.ac.jp
[c)] minematsu@limu.ait.kyushu-u.ac.jp
[d)] rin@ait.kyushu-u.ac.jp

[*1] https://www.leds.ait.kyushu-u.ac.jp/achievements

ここで，#はコースの ID を表す．各ファイルの形式や詳細については LAK Data Challenge Workshop の Web サイト[*2]を確認されたい．

2.2 OpenLA の構成モジュール

OpenLA は Course Information, Data Conversion, Data Extraction，Data Visualization の 4 つのモジュールから構成される．基本的な処理の流れとしては，Course Information モジュールでコースの基本情報を取得し，Data Conversion モジュールで分析しやすい形式に整形，Data Extraction モジュールで必要な情報を抽出し，Data Visualization モジュールで情報の可視化を行う．OpenLA による処理の概要図を図 1 に示す．この章の残りの部分ではそれぞれのモジュールについて説明を行う．

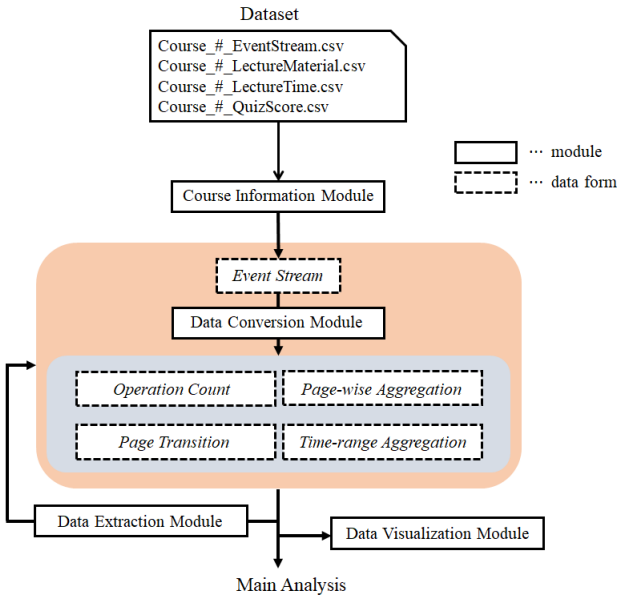

図 1 OpenLA の処理の概要図

2.2.1 Course Information Module

Course Information モジュールはデータセットファイルを格納したディレクトリへのパスを読み込み，クラス *CourseInformation* のインスタンスを返す．*CourseInformation* はクラスのメソッドによってコースの基本情報を取得する．以下に Course Information モジュールが提供している関数の一例を示す．関数の詳細やその他の関数については Web サイトを参照されたい[*1]．

CourseInformation.load_eventstream()
　デジタル教科書の操作ログを読み取る．

CourseInformation.user_ids()
　コースを履修している学習者の ID を取得する．

CourseInformation.lecture_start_time()
　授業の開始時刻を取得する．

[*2] https://sites.google.com/view/lak20datachallenge

関数 *load_eventstream()* はデジタル教科書の操作ログを Pandas.DataFrame 型で読み取り，それをメンバ変数として持つ Python クラス *EventStream* のインスタンスを返す．表 1 に操作ログの例を示す．表 1 が示すように，操作ログにはいつ誰がどの教材のどのページにどの操作を行ったかが記録されている．クラス *EventStream* は操作ログ中の各操作の回数や学習者の人数などを集計する関数をメソッドとして提供している．しかし，これらのメソッドは生の操作ログから基本的な集計を行う関数であるため，分析に必要な情報が取得できるとは限らない．そこで，OpenLA は *EventStream* をより分析に適した形に変換する Data Conversion モジュールや，必要な情報を抽出する Data Extraction モジュールを提供している．

表 1 操作ログの例

user id	contents id	operation name	page no.	event time	...
A	X	OPEN	1	2020/10/26 13:00	
A	X	NEXT	1	2020/10/26 13:05	
A	X	MARKER	2	2020/10/26 13:12	
...					

2.2.2 Data Conversion Module

Data Conversion モジュールは *EventStream* を 4 種類のクラス（*OperationCount*，*Page-wiseAggregation*，*PageTransition*，*Time-rangeAggregation*）に変換する関数群を提供している．これらのクラスは *EventStream* と同様に，変換後の操作ログをメンバ変数として持ち，その集計関数を提供する．

OperationCount はデジタル教科書に対する各操作の回数をコンテンツごとに集計した結果を表す．変換後の操作ログの例を表 2 に示す．デジタル教科書に関する多くの研究が操作ログの回数に基づいて行われており，学習者の学習傾向を大まかに把握するのに適している．

表 2 *OperationCount* 変換後の操作ログの例

user id	contents id	NEXT	PREV	MARKER	...
A	X	75	32	19	
A	Y	169	106	11	
B	X	60	18	2	
...					

Page-wiseAggregation はコンテンツの各ページにおける閲覧時間と操作の回数を集計した結果を表す．変換後の操作ログの例を表 3 に示す．この表現は学習者がどのページに注目し，メモやマーカーを残したかといった分析に役立つ．

PageTransition は *Page-wiseAggregation* と同様にコンテンツの各ページにおける閲覧時間と操作の回数を集計した結果を表す．しかし，*Page-wiseAggregation* が各ページ

表 3　*Page-wiseAggregation* 変換後の操作ログの例

user id	contents id	page no.	reading seconds	MARKER	...
A	X	1	109	0	
A	X	2	245	3	
A	X	3	195	1	
...					

の閲覧時間や操作回数の合計値を集計するのとは異なり，*PageTransition* ではページが遷移するごとに集計を行う．この表現は学習行動をより正確にトラッキングすることができ，学習パターン分析に適している．

Time-rangeAggregation は操作ログを一定の時間間隔ごとに分割し，各区間で最も長い時間閲覧されたページの番号と，その区間での各操作の回数を集計した結果を表す．集計する時間間隔は変換関数の引数で指定可能である．集計の間隔として 60 秒を指定した場合の例を表 4 に示す．この表現によって同じ時間帯の学習活動を比較することができ，学習者が教師の授業スピードに追い付いているかどうかの検出などに役立てることができる．

表 4　*Time-rangeAggregation* 変換後の操作ログの例

user id	contents id	elapsed seconds	page no.	MARKER	...
A	X	0	1	0	
A	X	60	4	3	
A	X	120	5	1	
...					

2.3 Data Extraction Module

Data Extraction Module は *EventStream* およびその変換クラスのインスタンスを受け取り，特定の学習者やコンテンツに関する情報を抽出する．例えば，成績に影響する学習活動を分析したい場合は，成績の良い学生と悪い学生の情報をそれぞれ抽出して比較すればよい．以下に Data Extraction モジュールが提供している関数の一例を示す．関数の詳細やその他の関数については Web サイトを参照されたい[*1]．

select_user()
　特定のユーザーに関する情報を抽出する．

select_contents()
　特定のコンテンツに関する情報を抽出する．

select_by_lecture_time()
　講義中/講義前/講義後の情報を抽出する

2.4 Data Visualization Module

Data Visualization モジュールは *EventStream* およびその変換クラスのインスタンスを受け取り，グラフによる可視化を行う．可視化をすることによってデータへの理解が深まり，また，学習者ごとのデータを容易に比較することができる．以下に Data Visualization モジュールが提供している関数の一例を示す．関数の詳細やその他の関数については Web サイトを参照されたい[*1]．

visualize_time_series_graph()
　データを時系列に沿って可視化する．

visualize_operation_count_bar()
　各操作の回数を棒グラフによって可視化する．

visualize_pages_in_time_range()
　一定時間間隔による集計から，各区間での閲覧ページの遷移を可視化する．

3. 使用例

OpenLA が冗長な前処理の削減において有効であることを示すために，講義中の off-task 検出 [5] と学習行動分析 [1] の前処理に対して OpenLA を適用する．

3.1 off-task 検出

off-task 検出の前処理では各学生の学習活動を 1 分間隔で分割し，各区間で 1 番長く読まれていたページを集計する．ただし，講義の終了前 15 分間は小テストが行われるので集計から取り除かれる．

図 2 に前処理の概略を示す．コードの全文は Web サイト[*1]に掲載する．一定間隔で操作ログを集計するため，関数 *convert_into_time_range()* を適用し *Time-rangeAggregation* クラスへの変換を行った．集計の間隔が 1 分という条件や講義終了前 15 分が集計から取り除かれるという条件は *convert_into_time_range()* の引数を指定することで満たすことができる．

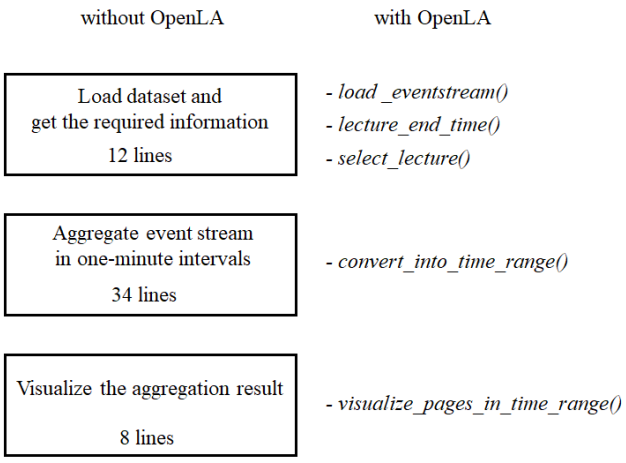

図 2　off-task 検出の前処理の概略

3.2 学習活動分析

学習活動分析の前処理では，各学生の学習活動を RP (Reading Pages)，PT (Preview Time)，RT(Reading

Time)，BRR (Backtrack Reading Rate) という 4 つの基準に沿って集計する．それぞれ，RP は重複を含めた閲覧ページ数，PT は予習時間，RT は閲覧時間，BRR はページを戻った回数を進んだ回数で割ることで計算される振り返りの割合を表す．

図 3 に前処理の概略を示す．こちらもコードの全文を Web サイト[*1]に掲載する．学習者の活動を詳細に分析するため，関数 *convert_into_page_transition()* を適用し *PageTransition* クラスへの変換を行った．予習時間 PT は，講義時間を基準に情報を抽出する *select_by_lecture_time()* を適用した後に *PageTransition* クラスのメソッド *reading_seconds()* を使用することで計算できる．重複を含む閲覧ページ数 RP と閲覧時間 RT は *PageTransition* クラスのメソッド *num_transition()* と *reading_seconds()* のみから集計できる．ページの振り返り割合 BRR はメソッド *operation_count()* によって次のページに進む操作と前のページに戻る操作の回数を取得し計算できる．

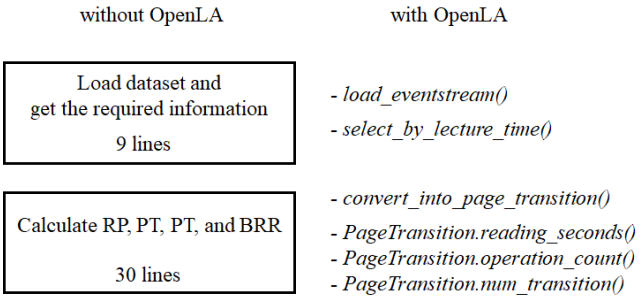

図 3 学習活動分析の前処理の概略

3.3 使用例に関する議論

使用例が示す通り，OpenLA は前処理の実装を簡略化することができる．OpenLA による実装の簡略化はコード行数を削減するだけでなく，入力と出力が明確な関数によってコードの可読性を向上させている．このように OpenLA は効率的な事前処理の実装が可能であるが，使用のために CSV ファイルのデータセットを作成するのは手間である．そこで現在，データベースからデータを直接読み取れるように改良を進めている．

4. まとめと今後の課題

本稿ではデジタル教科書のログ分析を支援するオープンソースライブラリ OpenLA を紹介した．OpenLA はデジタル教科書の操作ログ分析に役立つ 4 種類のモジュール，Course Information，Data Conversion，Data Extraction，Data Visualization を提供している． これにより，コースに関する基本的な情報の取得，操作ログの分析に適した形式への変換，必要な情報の抽出，および可視化を行うことができる．これらのモジュールによって，多くの学習分析研究で共通している前処理の冗長な再実装が削減され，研究者は先進的な分析に集中することができる．

今回は OpenLA の開発と既存研究への適用にのみ着目したため，今後はエンドユーザーによる客観的な評価やその評価に基づく改善を進めたい．

謝辞 本研究は，JST AIP 加速課題 JPMJCR19U1 と科研費基盤研究（A）JP18H04125 の支援を受けた．

参考文献

[1] Chengjiu Yin, Masanori Yamada, Misato Oi, Atsushi Shimada, Fumiya Okubo, Kentaro Kojima, and Hiroaki Ogata. Exploring the relationships between reading behavior patterns and learning outcomes based on log data from e-books: A human factor approach. *International Journal of Human–Computer Interaction*, Vol. 35, No. 4-5, pp. 313–322, 2019.

[2] Atsushi Shimada, Kousuke Mouri, Yuta Taniguchi, Hiroaki Ogata, Rin Ichiro Taniguchi, and Shin'ichi Konomi. Optimizing assignment of students to courses based on learning activity analytics. In *EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining*, pp. 178–187, January 2019.

[3] Fumiya Okubo, Takayoshi Yamashita, Atsushi Shimada, Yuta Taniguchi, and Konomi Shin'ichi. On the prediction of students' quiz score by recurrent neural network. *CEUR Workshop Proceedings*, Vol. 2163, , January 2018.

[4] Atsushi Shimada, Yuta Taniguchi, Fumiya Okubo, Shin'ichi Konomi, and Hiroaki Ogata. Online change detection for monitoring individual student behavior via clickstream data on e-book system. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pp. 446–450, March 2018.

[5] Gökhan Akçapınar, Mohammad Nehal Hasnine, Rwitajit Majumdar, Brendan Flanagan, and Hiroaki Ogata. Using learning analytics to detect off-task reading behaviors in class. 2019.