Volume 5(1), 55-74. http://dx.doi.org/10.18608/jla.2018.51.5

A Sequence Data Model for Analyzing Temporal Patterns of Student Data

Mohammad Javad Mahzoon[1]*, Mary Lou Maher[2], Omar Eltayeby[3], Wenwen Dou[4], Kazjon Grace[5]

Abstract

Data models built for analyzing student data often obfuscate temporal relationships for reasons of simplicity, or to aid in generalization. We present a model based on temporal relationships of heterogeneous data as the basis for building predictive models. We show how within- and between-semester temporal patterns can provide insight into the student experience. For example, in a within-semester model, the prediction of the final course grade can be based on weekly activities and submissions recorded in the LMS. In the between-semester model, the prediction of success or failure in a degree program can be based on sequence patterns of grades and activities across multiple semesters. The benefits of our sequence data model include temporal structure, segmentation, contextualization, and storytelling. To demonstrate these benefits, we have collected and analyzed 10 years of student data from the College of Computing at UNC Charlotte in a between-semester sequence model, and used data in an introductory course in computer science to build a within-semester sequence model. Our results for the two sequence models show that analytics based on the sequence data model can achieve higher predictive accuracy than non-temporal models with the same data.

Notes for Practice

- The sequence model for learning analytics represents temporal relationships in heterogeneous student data as a basis for predictive models.
- This paper contributes an approach to learning analytics that uses temporal models for predicting students at risk and shows how these models perform with higher accuracy than non-temporal models.
- The implications of the sequence model for student data is improved predictive accuracy and understanding of students at risk.

Keywords

Sequence data model, educational data mining, learning analytics, predictive modelling, knowledge discovery.

Submitted: 02/09/17 — **Accepted:** 12/10/17 — **Published:** 04/09/18

*Corresponding author [1]*Email: mmahzoon@uncc.edu Address:The University of North Carolina at Charlotte, Charlotte, NC, USA, 9201 University City Blvd, Charlotte, NC 28223-0001, ORCID: 0000-0003-4689-5842*

[2]*Email: m.maher@uncc.edu Address: The University of North Carolina at Charlotte, Charlotte, NC, USA, 9201 University City Blvd, Charlotte, NC 28223-0001*

[3]*Email: oeltayeb@uncc.edu Address: The University of North Carolina at Charlotte, Charlotte, NC, USA, 9201 University City Blvd, Charlotte, NC 28223-0001*

[4]*Email: wdou1@uncc.edu Address: The University of North Carolina at Charlotte, Charlotte, NC, USA, 9201 University City Blvd, Charlotte, NC 28223-0001*

[5]*Email: kazjon.grace@sydney.edu.au Address: The University of Sydney, Sydney, Australia, Wilkinson Bldg G04, 148 City Rd, Darlington, NSW 2008, Australia*

1. Introduction

In this paper, we present the sequence data model as a repository that explicitly represents the temporal aspects of student data. In this model, student data is grouped into nodes that are temporally ordered, integrating the passing of time into the structure of the model. Temporal features in student data capture important information about the time in which specific events occurred. Including temporal features in a feature vector model does not capture temporal relationships of the data items. We claim that the explicit representation of temporal relationships can facilitate developing more accurate predictive models for student risk and success. To demonstrate the benefits of the sequence data model, we represent temporal relationships and dependencies in within- and between-semester student data models. Within- and between-semester models provide insight for how students progress during a single semester and across multiple semesters during their academic career. The representation of temporal relationships gives rise to the following properties: contextualization, segmentation, and storytelling. The generality of the sequence model enables multiple analytic processes and facilitates pattern identification through a process of re-representation and analytic interpretation. In this paper, we present and elaborate on the structure of the sequence model, its properties, and the use of re-representation to enable multiple analytic models for student success and risk.

In contrast to our sequence data model, the more common approach in knowledge discovery and data mining is to construct a feature vector for each entity in the model as the basis for generating patterns, predictive, or probabilistic models. The feature vector model does not explicitly account for temporal information that is inherent in student records and learning activity logs. In this feature vector representation, each data point is represented by a vector with a fixed set of features (or dimensions). For example, in the learning analytics context, each data point can be a vector of a student's performance in a certain course or in a certain degree program. This representation may have features such as student background information, course information, and the student's achievements in the course, such as grades, assignments, and quizzes.

Many examples in the learning community use the feature vector representation as a student data model. For example, Romero and Ventura (2007) and Romero, Ventura, and García (2008) conducted surveys showing different approaches taken in the learning community to discern student behaviour using machine learning or statistical methods. Generally, the data mining approaches discussed in their survey used statistics or machine learning techniques operating on a feature vector representation of each student having data such as demographic information, course grades, and learning management system (LMS) logs. Several others, such as Mohamad and Tasir (2013) and Peña-Ayala (2014), review approaches that used different analytics with similar feature sets for their vector representations.

More recent projects such as Course Signals (Campbell, 2007; Campbell, DeBlois, & Oblinger, 2007; Arnold, 2010; Arnold & Pistilli, 2012) and Open Academic Analytics Initiative (OAAI; Jayaprakash, Moody, Lauría, Regan, & Baron, 2014) used the same vector representation, but added different features such as academic history or course partial grades. For example, Macfadyen and Dawson (2010) used the Blackboard Vista LMS to extract features correlating with the final grade of a fully online course. These features include the total number of discussion messages posted, total number of mail messages sent, total number of assessments completed, and other LMS tracking features. Macfadyen and Dawson (2010) used logistic regression to classify students as successful or at risk and could identify 80.9% of students who were actually at risk (failed the course). They acknowledge the fact that some of the LMS features such as "time spent on activities" do not have predictive power because of the complex composite behaviour of students. For example, students in the lower quartile of course grade tended to spend slightly more time, on average, than the highest quartile. This means that including temporal features such as "time spent on activities" in the feature vector model does not accurately model student behaviours and eventually does not increase predictive accuracy. In this paper, we model student behaviours over time by building temporal models to account for temporal relations and dependencies of data.

As another example, Wolff, Zdrahal, Nikolov, and Pantucek (2013) used click behaviours in the virtual learning environments (VLE) as the data source to identify students at risk using a decision tree model (C4.5). They include assignment scores and number of clicks in the VLE in specific time periods to predict final outcome and performance drop for students who were performing well. They also acknowledge the fact that number of clicks cannot predict successful behaviour. Based on Wolff et al. (2013): "There were students who clicked a lot and still failed, or those who clicked hardly (if) at all and yet passed." They created time frames for counting the number of clicks to break down the general feature of "number of clicks" into "number of clicks in a time window." This is similar to the concept of "nodes" in our sequential model (see section 3), which groups the information into certain time frames; however, we consider heterogeneous data sources in the information in nodes, as well as the time dependency of nodes on each other.

One of the advantages of the feature vector representation is that it makes strict assumptions that enable the application of multiple statistical and machine learning analyses. Vector representations assume that data items are not related to each other (independency of data items), and their features have no correlation with each other (independency of features). These assumptions of independence, as well as the fixed length of the vector representation, make analytics relatively easier.

However, these assumptions can be problematic as student data has dependencies. Students progress, improve, and learn over time in structures related to semesters and courses. Thus, new semester data can be highly correlated to that of the previous semester, and this dependency is not directly captured by approaches using vector representation. A typical example of a temporal correlation that is not considered by approaches using a feature vector representation is the correlation between the final grade and activity grades for the same course. Student activity grades can include assignments, quizzes, or midterm grades obtained during the semester by students. The order in which these grades occur provide important information for predicting success or risk.

OAAI (Jayaprakash et al., 2014) is one of the approaches that include student activity information in addition to the final grade in their vector data model. In OAAI, all the student activity information for each course is aggregated into one feature called partial contribution score. Even though OAAI adds to the overall information about a student in the vector data model by considering student activities, it relies on a predefined aggregate for all the data items that lead to the student's performance. A sequence data model allows the contributions to be ordered and disaggregated, and allows multiple analytic interpretations, providing more flexibility in finding better predictive patterns of risk or success.

Data modelling is a critical step in the general process of knowledge discovery as it captures the assumptions and choices made about the data and sometimes determines the analytics that can be applied to the model. Figure 1 illustrates the knowledge discovery process as an interactive learning analytics process. Data modelling includes decisions about what data is collected and the selection of features to be included in the data model. As shown in Figure 1, data modelling can influence how we view the knowledge discovery challenge and the options for building the predictive model. Therefore, what we choose as the data model narrows down our choices for the predictive model and how we can formulate a response to the challenge (e.g., retention or risk). We believe modelling student data as a feature vector representation misses the opportunity to explore temporal relationships within data.

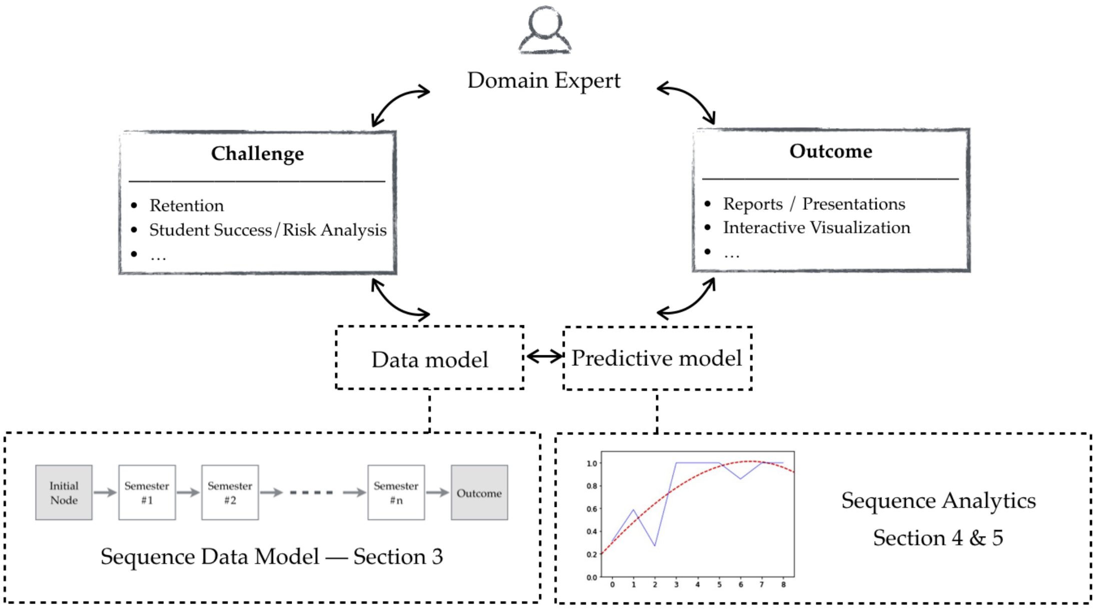

Figure 1. The knowledge discovery process in the context of interactive learning analytics.

In this paper, we present a sequence data model as an intermediate model for a repository of student data to explicitly include temporal dependencies. We discuss properties and potentials of the sequence data model and how it differs from well-known techniques in temporal data analysis, such as time series in section 2. In section 3, the structure of the sequence model is presented. We also present the concept of re-representation as the mapping from the sequence data model to patterns or signatures that enables different analytic interpretations of the sequences. We demonstrate the benefits of the sequence data model for two cases: a within-semester sequence model built on the student activity data from an Introduction to Computer Science course (section 4), and a between-semester sequence model using data from 10 years of students majoring in Computer Science (section 5).

In summary, our paper presents the following contributions:

- Temporal relationships in student data capture important sequences that facilitate developing predictive models for student risk and success.
- The sequence data model has the following four features: 1) capturing temporal dependencies in data, 2) identifying salient and context features for different analytic models, 3) allowing for different segment granularities, and 4) being expressive for storytelling.
- Demonstrating two sequence models and analyzing them on real data for between- and within-semester models. The between-semester sequence model was applied on 10 years of student data to identify students at risk of not graduating on time, and the within-semester sequence model was built on data from a large enrollment introductory course to predict student performance in the course.

2. Approaches to Representing Time in Data Models

During the last decade, increasing research in the data mining and machine learning communities has produced many approaches to analyzing time-related raw data, but most of these approaches have been developed in a sub-domain to address a specific problem. For example, time series is used primarily in business and marketing to identify market trends; and data stream mining was developed in the machine learning community to address concept change in data. Each of these approaches is tailored to their corresponding domain, and built with certain assumptions that are not necessarily applicable to student data. In this section, we discuss some of these approaches and explain how they are different from our sequence data model.

2.1. Time Series

Time series analysis aims to arrive at a mathematical or statistical model to describe series of observations over time, and it has applications to domains ranging from the stock market to weather forecasting. Various methods have been proposed in time series literature to solve prediction, classification, and regression problems. All these models were built on the same assumptions that 1) the data is in numerical format, and 2) a significant number of data samples are available.

Neither of these assumptions is necessarily true for student data:

1. Student data is highly heterogeneous, containing ordinal and categorical features in addition to numerical. Even though some data items such as grade and other performance features can be converted to numeric data, many features such as courses taken or transferred cannot be represented in numbers while preserving their meaning.
2. The data we have for each student is limited and uniquely different from other students. The data about a single student cannot be generalized to a format that reconciles it with the data on all students without significant information being lost. The amount of data available for each student is also unique and can vary widely.

Additionally, time series analysis usually looks for recurring patterns or regularities within a time period. By contrast student data is temporal but not periodic. Students progress in each semester as they acquire knowledge and prepare to meet the new requirements for the next semester. While time series can still be applied to student data to identify periodic patterns for numeric features, our sequence data model facilitates detecting trends and irregularities in sequences having heterogeneous and variable length data items.

2.2. Data Stream Mining

Data Stream Mining is a subdomain of data mining that presents methods to efficiently process continuous massive sequences of data items called streams. These methods can watch for “concept drift” (Widmer & Kubat, 1996): when the general statistical properties of the target prediction changes. Methods in data stream mining adapt to the changes in the stream to have a better prediction for new instances of data. For example, Hulten, Spencer, and Domingos (2001) present a model to maintain and update a decision tree for concept-drifting data streams. The model is always up-to-date with the latest instances of the stream, while discarding old concepts that were changed over time.

Adapting data stream mining ideas to the student data analytics faces several challenges. In student data analytics, we are not dealing with massive continuous data streams. Student sequences have a clear starting point and a duration of several years, making them neither continuous nor massive. Also, data stream algorithms do not keep track of changes in data since they discard the changed concepts to account for the newest ones. To interpret student behaviour and investigate what makes a student at risk, we need to capture changes in trends and identify unexpected patterns.

2.3. Sequence Pattern Mining

Another sub-domain of data mining that works with sequences is sequence pattern mining, which is used to identify frequent sets of items or patterns in data or strings (Agrawal, Imieliński, & Swami, 1993). This domain is generally used for identifying behaviour patterns of consumers in the business domain. One such approach detects frequent items bought together from the all transactions dataset. For example, Padmanabhan and Tuzhilin (1999) propose an interestingness measure to filter all frequent items to obtain interesting items that happen to be unexpected transactions contradicting beliefs.

We can make an analogy to transfer ideas from sequence pattern mining to student sequence data mining. If we treat each student sequence as a transaction, then the task becomes frequent events happening together in student sequences. However, there are certain assumptions in sequence pattern mining, which makes it hard to continue the analogy further. For instance, in sequence pattern mining, it is assumed that we know beforehand about all potential items in transactions (i.e., all items being sold in a store). This assumption holds in business and marketing since the number of items are finite and known. However, student data sequences have a wide range of possibilities such as courses taken, assignment grades, forum participation, and other academic and non-academic activities. It is a daunting task to generate all potential events for a student sequence.

3. Sequence Data Model

In this section, we present a sequence data model that uses time to sort heterogeneous sources of student data and form sequences of information for each student. The sequence data model allows the analytic models to make use of the dependency of events happening during a student's life. Sorting student data over time and aggregating heterogeneous data in a sequence format enables the analytics to account for time dependency. Also, the sequence format enables flexibility in defining the salient information in each data node, the contextual information within each node, the granularity of the nodes by changing time-related boundaries, and the ability to interpret sequences as stories. This section presents the basic representation of a student sequence data model, its properties and significance.

3.1. Structure and Properties of the Sequence Data Model

We define a student sequence as data items grouped into temporally ordered structures called "nodes." For example, a node may represent a semester, and may contain a student's data items related to that semester: courses taken, grades received, extra-curricular activities, and so on. This grouping gives context to the data items and allows for analysis at the level of both data items and nodes.

Figure 2 illustrates the structure of the sequence data model in which information about a student is grouped by semester. The sequence starts with an initial node that captures attributes outside of the node-based temporal sequence such as demographics and prior academic achievement. A node is then included for each semester the student is enrolled and finishes with an outcome node.

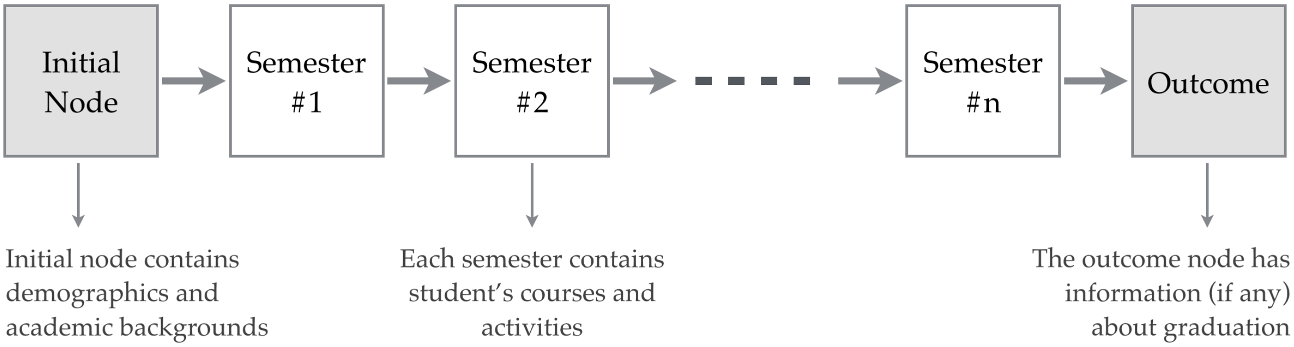
Initial Node
Semester #1
Semester #2
Semester #n
Outcome
Initial node contains demographics and academic backgrounds
Each semester contains student's courses and activities
The outcome node has information (if any) about graduation

Figure 2. The structure of the sequence data model, illustrated by the between-semester sequence.

The properties that characterize a sequence data model include time dependency, contextualization, segmentation, and storytelling.

Time dependency: The sequence data model explicitly represents that the later data items can depend on former data items. This allows the explicit representation of temporal dependencies such as the correlation between final grade and student assignment grades. In comparison, a vector representation assumes that data points are independent of each other, and features (independent variables) do not have correlation with each other.

Contextualization: The grouping of data items into nodes gives context to salient features that are selected for analysis. For example, if each node groups information for one semester, then data can be identified as a salient feature within each node, such as course grades, while other features such as student activities are the context of the salient feature.

Segmentation: The nodes in a sequence allow us to represent the data in segments. Different choices for the beginning and ending of each node define a principle for a window of time and allow the data model to capture a different granularity for the segments; for example, semesters versus weeks. Access to LMS data makes finer-grained node segmentation possible, which may lead to more timely assessments of academic risk.

Storytelling: A sequence of information expresses a student's life. This property enables us to view the nodes as events happening during a student's academic life. We can infer a narrative from the nodes to tell a story about a specific or typical student in order to hypothesize about success or risk.

To describe the advantages of our sequence data model, we collected data from a 10-year period about students enrolled in our College of Computing from multiple sources, including the university database of students and courses as well as data collected by the learning management system used during that period. We cleaned and extracted information about each student, and constructed sequences for each student. In Table 1 we show broadly defined categories of student data included in our sequence data models.

We constructed two models using different node segmentations: one at the level of semesters and the other at the level of weeks of a course. The between-semester sequence model groups data about each student for each semester they are enrolled. The within-semester sequence model groups data about each student for each week recorded by the LMS within a single course. In the sections below, we describe the sequence models and their properties for each of the two segmentations. We demonstrate the benefits of the sequence model through re-representation and analytic interpretations of the sequence models and the ability to identify patterns of student behaviour that can predict risk or success.

Table 1. Different categories of student data that can be included in the sequence data model

Categories of student data	Description	Occurrence
Demographics	Age, gender, employment status, citizenship type, marital status, etc.	One time only (stored in the initial node)
Academic background	Passed tests, previous degree/education, transferred courses	One time only (stored in the initial node)
Academic information	Major, advisor, funding, courses taken, course grades	Each semester
Course activities	Assignments, quizzes, and other activities logged in Learning Management Systems (LMS)	Daily, weekly, or monthly depending on the course
Extracurricular activities	Participation in student organizations, use of library and other facilities such as laboratories	Daily, weekly, or monthly depending on the activity
Outcome	Graduation status such as date of graduation, major, date of becoming inactive or withdrawing	One time only (stored in the outcome node)

We constructed two models using different node segmentations: one at the level of semesters and the other at the level of weeks of a course. The between-semester sequence model groups data about each student for each semester they are enrolled. The within-semester sequence model groups data about each student for each week recorded by the LMS within a single course. In the sections below, we describe the sequence models and their properties for each of the two segmentations. We demonstrate the benefits of the sequence model through re-representation and analytic interpretations of the sequence models and the ability to identify patterns of student behaviour that can predict risk or success.

4. A Within-Semester Sequence Model and Analysis

The within-semester sequence model permits the analysis of risk and success for students within a single course. In contrast to a between-semester sequence model that works with a high-level success measure, such as on-time graduation, a within-semester sequence model defines success in terms of passing or failing a course. This would be of value to an instructor during a course, or as a more detailed analysis by an advisor when a student is not doing well across several courses.

To demonstrate a within-semester sequence model, we collected activity information for 91 students enrolled in the Computer Science I course offered in Spring 2017. The student activities include lecture tests, quizzes, assignments, and participation in class activities that were logged from the Learning Management System (LMS). The course materials also included students' reflections: an informal survey taken by students regularly after major quizzes and assignments. Students are not required to take reflection surveys, but are encouraged to do so for extra credit. In this section, we discuss how to build a within-semester sequence model around a course (section 4.1), analyze sequences to identify students at risk of failing the course (section 4.2), and evaluate the within-semester sequence model (section 4.3).

4.1. Student Data in a Within-Semester Sequence Model

A within-semester sequence model starts with student background information and ends with the outcome node, which in this case is whether the student failed or passed the course as well as the course grade. The rest of the nodes in the within-semester sequence model aggregate the data for each week.

We created the within-semester sequence model for an introductory course in computer science. The course had 91 students, 12 of whom failed the course. While this is an imbalanced dataset in which the percentage of students who fail the course is much lower than those who pass, this is typical of student performance in all courses. Our sequence model consists of 19 nodes (one background node, 17 weekly nodes, and one outcome node). In total, we have 4 background features, and 110 gradable test features. The course also recorded students' reflections: an informal survey taken by students regularly after major quizzes and assignments. To produce numeric features from reflections we used the Linguistic Inquiry and Word Count (LIWC; Tausczik & Pennebaker, 2010) tool to generate linguistic sentiment features. From the sentiment features we picked 18 reflection features, which has less correlation among the others. In summary, we included 4 background features, 110 test features, and 18 reflection features in the sequence model. Figure 3 shows an example of a sequence for a successful student. The student started participating from the first week — Jan 9–Jan 15, 2017 — and passed the course after 17 weeks (student recess and reading weeks were excluded). Each node in Figure 3 groups the student's activities in one week. Not all weeks have the same activities, and therefore, each node can contain different set of activity logs.

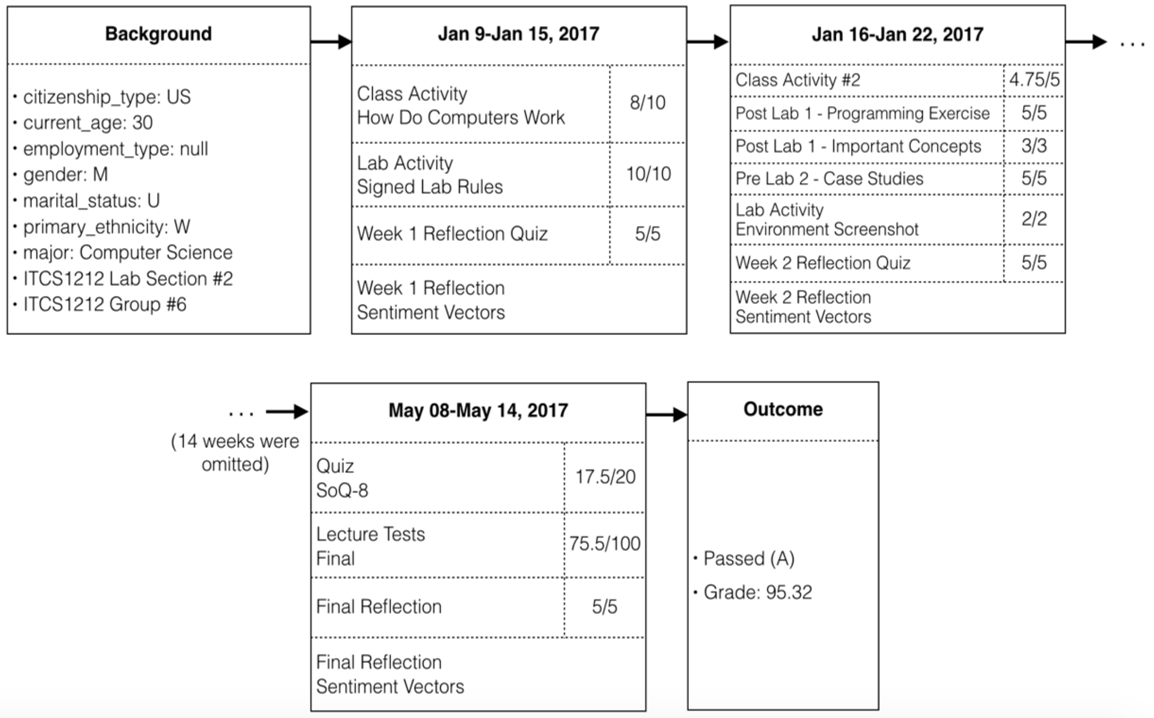

Figure 3. An example of the data in a within-semester sequence model for a successful student.

Based on the example in Figure 3, we can examine properties of the within-semester sequence data model:

Time dependency. Materials in a course are usually coherent, and ordered from basic to more complex. In general, later materials are dependent on former ones, and as a result, test grades obtained by students for each week are predictive grades of previous weeks. The within-semester sequence model affords analytics that consider time dependency between data items of each week.

Contextualization. We can define features that are "salient" for analytics. Salient features can be a subset of features in the sequence, such as test grades, which are used by the analytics to discover trends of student activities. The rest of the features act as the "context" for examining analytics results. Context features are used after the analytics to increase interpretability of the results. For example, in Figure 3, we can choose quizzes and assignments as the "salient" features and the rest of the features, such as class activities and reflections, as the "context" features. This choice will constrain the analytics to use only the salient features (i.e., quizzes and assignments), while context features (i.e., class activities and reflections) add more information to the nodes to interpret the analytic results.

Segmentation. In a within-semester sequence model, we define nodes to represent one week of a semester. Depending on the frequency of course content/assessment we could change the segment granularity from one week to every two weeks (if the number of assessments are low), or every day (if the number of assessments are high).

Storytelling. Storytelling provides context and insight when a specific student is not performing well in a course. For the within-semester sequence model, the story is about a student's engagement in a specific course.

4.2. Analyzing the Within-Semester Sequence Model

In this section, we analyze the within-semester sequence model to identify students who are at risk of failing the course. We use the analytic process shown in Figure 4 to map from the complex data in the sequence model to signatures that are easier to analyze (re-representation), and then extract metadata from the signatures for classification and clustering (analysis). The re-representation process is illustrated for a single student sequence model to a single student signature. The analytic process is illustrated by extracting the metadata from each signature to be entered as a row in the metadata table. This tabular data is used to classify and cluster students so that we can see groups of similar students as well as outliers.

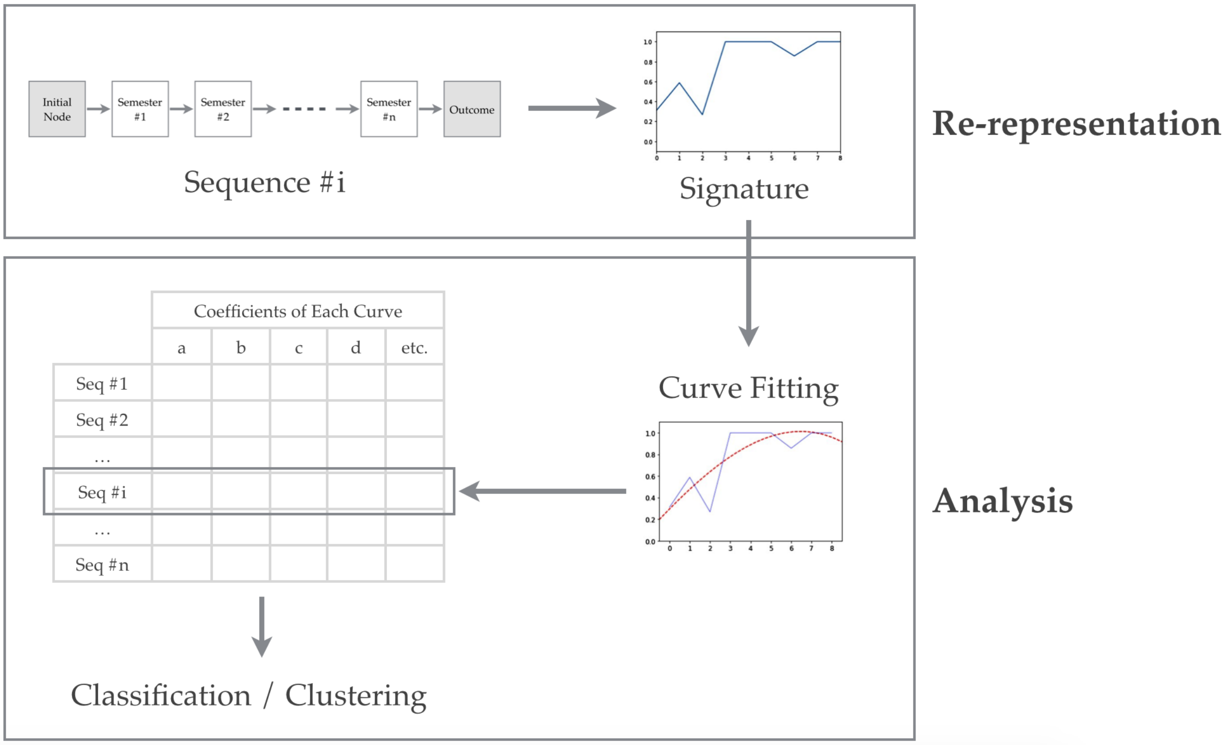

Figure 4. An analytic process to analyze sequence data for classification and clustering tasks.

This re-representation part of the process is similar to the one used in Maher and Mahzoon (2015) to identify patterns and unexpected data items in mobile device design where the data source provided a sequence of mobile devices ordered by date of appearance on the market. Maher and Mahzoon (2015) re-represented the sequences into "signatures" using self-organizing maps (SOM), which made it easier to identify when a new mobile phone is significantly different from the previous phones (outlier) and when that difference forms a new trend (harbinger).

To build signatures of a within-semester sequence, we proposed an approach called progressive classification, which builds signatures that reflect the confidence of predicting the student as being successful (i.e., passing the course) over time. This process is discussed in detail in the next section (4.2.1). After creating signatures, we used curve fitting to extract metadata from generated signatures and build a predictive model for the course performance (section 4.2.2).

4.2.1. Generating Signatures from the Sequence Model

To create signatures from the within-semester sequences, we propose an approach called progressive classification, which applies a classification algorithm over time on each node of student sequences. The classifier aims at predicting if each student will pass the course at each time-step, and records the confidence of its prediction. After the classification is done for all nodes, the student signatures are generated by plotting the confidence of the predictions over time. There are several ways to calculate confidence of a predictor depending on the prediction algorithm. In our case, we used Support Vector Machines (SVMs; Cortes & Vapnik, 1995) for classification. SVMs transform the data to a high dimension space such that it will be linearly separable in the new dimensions. In our case, the SVM uses a nonlinear transformation using a Radial Basis Function (RBF) kernel to transform the data to the higher dimension space, then classifies the data using a hyperplane. The hyperplane is used as the decision boundary to separate the classes. The distance of the data to the decision boundary will be the confidence of the classifier. The higher the absolute distance, the higher the confidence of the SVM. We use the absolute distance to capture the confidence of the classifier.

Algorithm 1 explains the progressive classification in pseudocode. The algorithm starts with all student sequences as the input. For each node i (excluding the outcome node), we run the SVM classifier to identify students failing the course based on features in node i. Then, for each student sequence S we record the distance (D) of S to the classifier's decision boundary. This will show the confidence of the SVM's prediction. We create a point P=(Px, Py) in the student's signature by recording the confidence level (Py=D) along with the node index (Px=i). Repeating this process for all students and all nodes will generate a signature plot for each student. Each signature plot will have n points where n=number of nodes (excluding the outcome node).

Algorithm 1. Progressive Classification Using SVM

```
Input: Sequences for all students
Output: Signature plots for all students
For $i = 0$ to number of nodes (excluding outcome node)
    Classify students based on salient features of node $i$ using SVM
    For each student $S$
        $D \leftarrow$ The distance between $S$ and the classifier's decision boundary
        $(P_x, P_y) \leftarrow (i, D)$
        Add point $P$ to the signature plot of $S$
```

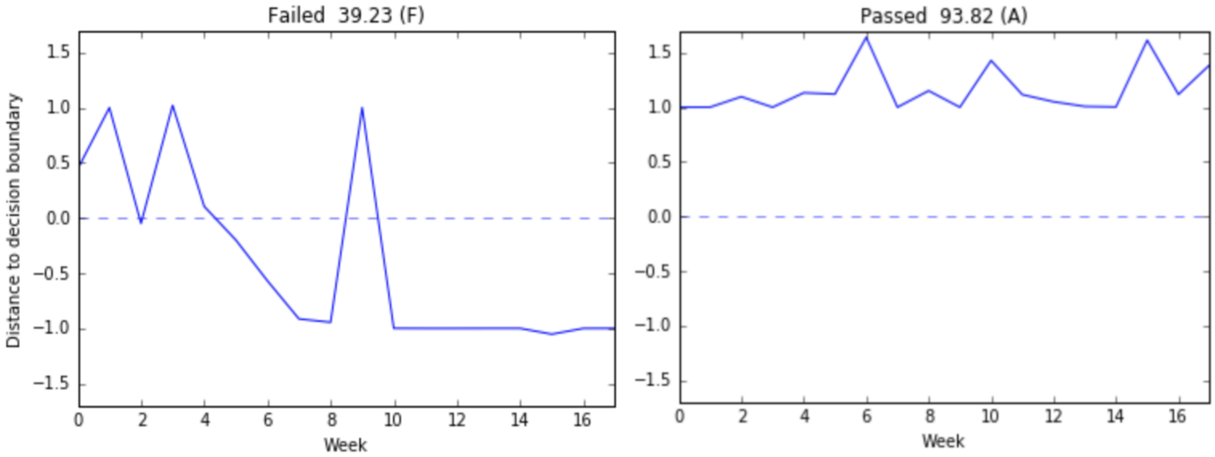

Figure 5. Signatures generated with a progressive classification algorithm. The left figure shows the signature for a student failing the course, and the right figure shows the signature of a successful student. Positive distance to decision boundary classifies a student as successful. Magnitude of the distance value represents the confidence of the classifier.

Figure 5 shows the signature of two students: one failing the course (left) and the other successfully passing the course with an A (right). In this figure, the X-axis is the node index. Since in our within-semester data model each node contains one week of data, the node index is the week number. Node index 0 refers to the background node. The Y-axis in Figure 5, shows the distance to the classifier's decision boundary. The dotted line (Y=0) shows the decision boundary, the area with positive Y (positive distance to the decision boundary) indicates prediction of success, and the area with negative Y belongs to the at-risk prediction.

To describe Figure 5, we start with the student on the left, who eventually failed the course. This student was classified as a successful student at the beginning of the semester using only the features of the background node (x=0, y=0.5). After week 4 (except for week 9), however, this student was classified as being unsuccessful (negative Y). On the other hand, the student on the right was classified as successful (positive Y) in all weeks with high confidence (average Y=1.25).

Signatures generated from a progressive classification algorithm (such as Figure 5), give us a new representation of the sequences that discriminate successful students from students at risk of failing the course. The next section uses this new representation as the basis to extract metadata (features) to automatically identify at-risk students.

4.2.2. Analyzing Signatures from the Within-Semester Sequence Model

Signatures produced from progressive classification can be used by domain experts to get insights about student behaviours. However, to automatically identify at-risk students having the signatures, we need to extract metadata (features) from the signatures. After extracting features from the signatures, we can convert them to a vector representation, and apply machine learning algorithms to classify or cluster signatures.

Curve Fitting. One approach to convert the signatures into a vector representation is curve fitting. We can use coefficients of a fitted curve as features of the vector representation. Curve fitting captures trends of the signature, while directly extracting features from the signature plot might not give us this information. We used polynomial curves to fit curves to signatures. Lower degree polynomials give simpler models but lose more information, whereas higher degree polynomials increase model complexity while adding insignificant information. We examined for different degrees of polynomial curves for our dataset and settled on 3 degrees as a sweet spot between model complexity and sufficient information.

4.3. Evaluating the Within-Semester Sequence Model

We evaluated the benefits of our within-semester sequence model by revisiting its four features:

Time Dependency. The within-semester sequence model affords analytics that consider the time dependency between data items. The progressive classification algorithm is an example of such analytics. It uses the sequence model to generate signatures. Fitting a curve on the signature and extracting metadata features (fitted curve's coefficients) from the signatures captures the trend of the data that accounts for time dependency.

To show the significance of such analytics we built a non-temporal feature vector model from our student data with only non-temporal features, such as demographics and statistic features, and compared its performance with a temporal model that includes temporal features (signature's metadata features) in addition to the non-temporal features. We evaluated the comparison in three separate groups based on the features included in the models.

Group 1: Including background features, excluding statistical features. The background features included in the models are: age, gender, major, and lab section. Both non-temporal and temporal models have the background features, but the temporal model adds four more features extracted from the student signatures. These four features extracted by fitting a 3-degree polynomial curve to the signatures and using the fitted curve's coefficients (see section 4.2.2 for more details). We call these four features temporal features.

After building temporal and non-temporal feature vector models, we used SVMs for both models to classify at-risk students. We tested the temporal model versus the non-temporal model for their accuracy over 10-fold cross-validation with different settings and summarized the results in Table 2. Based on Table 2, the temporal model outperforms the non-temporal model in all cases over 6.8% on average.

Table 2. Comparing a non-temporal model with a temporal (within-semester) model in group 1. Both models include background information, but exclude statistical features.

Model	Background Info	Statistical Features	Salient: Tests	Salient: Reflections	Average Accuracy
Non-Temporal Model	✔	✘	Not Applicable	Not Applicable	83.92
Temporal Model	✔	✘	✔	✘	87.53
			✘	✔	90.03
			✔	✔	**94.64**

Group 2: Excluding background features, including statistical features. The statistical features included in the models are as follows: average score for class activities, quizzes, lecture tests, lab tests, lab activities, assignments, and all tests. Both temporal and non-temporal models have the statistical features, but the temporal model adds four more temporal features as described in the previous group. We used the same process as to the previous group to compare the average accuracy of the models. Table 3 shows the performance results for this group. Based on the table, the temporal model outperforms the non-temporal model in all cases near 8% on average. Comparing results of this group to group 1 indicates that the background features extracted in the models are not as good as statistical features in discriminating between successful and at-risk students.

Table 3. Comparing a non-temporal model with a temporal (within-semester) model in group 2. Both models exclude background information, but include statistical features.

Model	Background Info	Statistical Features	Salient: Tests	Salient: Reflections	Average Accuracy
Non-Temporal Model	✘	✔	Not Applicable	Not Applicable	84.92
Temporal Model	✘	✔	✔	✘	89.53
			✘	✔	93.53
			✔	✔	**95.64**

Group 3: Including both background features and statistical features. We combined features from group 1 and 2 and compared the performance of temporal and non-temporal models in Table 4. As shown in the table, the accuracy of the temporal model is on average 5.7% better than the non-temporal model. Comparing group 3 with the previous two groups shows that including background features does not improve the model's performance.

Table 4. Comparing a non-temporal model with a temporal (within-semester) model in group 3. Both models include background information and statistical features**.**

Model	Background Info	Statistical Features	Salient: Tests	Salient: Reflections	Average Accuracy
Non-Temporal Model	✔	✔	Not Applicable	Not Applicable	84.92
Temporal Model	✔	✔	✔	✘	88.14
			✘	✔	90.53
			✔	✔	**93.39**

Based on the results shown in Tables 2–4, the temporal model has significantly better accuracy over the non-temporal model. Also, in cases where we include all features as the salient features, we obtain the maximum accuracy for the temporal model (94.64%, 95.64%, and 93.39%), which is on average 9.6% better than the best non-temporal model. However, we expect this gap between the accuracy of the non-temporal model and the temporal model (9.6%) decreases as we add more statistical or background features. While our results may have been affected by the small percentage of students being at risk, our comparison is based on the same imbalanced dataset and the temporal model outperforms the non-temporal model.

Contextualization. The sequence model can contextualize features by separating salient features versus context features. While salient features are those features used in the predictive or statistical analytics, the context features give us more information to interpret the results. In our evaluation, we tried different subsets of features as salient features. Based on Table 2, having reflections as the salient feature produces a model with better predictive power (~3% better accuracy on average), than having tests as the salient feature. While having all features as salient creates a model with even better accuracy (94.56% on average), the model's complexity is higher and more prone to overfitting.

Segmentation. The frequency of data items and test materials in our course data led us to use a weekly level sequence model. We can adjust the granularity level for other courses to include enough data in each node. For example, we can use nodes having two or three weeks of data for classes with fewer activities. The sequence data model provides a representation that makes it easier to change the granularity of nodes and aggregating data over time.

Storytelling. We can interpret the data about student engagement in course activities in the form of stories from a within-semester sequence model. For example, a successful student sequence (shown in Figure 3) can be interpreted as this story:

> This student is a 30-year-old white, male US citizen who is a Computer Science major. In the first week, he participated in all activities, and he continued to improve on his class activities in the second week. In the last week, he obtained a good grade relative to his classmates for the last quiz, even though he got a very low grade for the final lecture test. This low grade is fine since the final lecture test does not contribute much to the final grade. Finally, the student passed the course with an A.

The storytelling feature of the sequence model contributes to the interpretability of the analytics. For instance, the interpreted stories can be used in two scenarios when the goal of the analytics is to classify student as successful or at risk:

1. **Description:** After the analytics, we can build stories for certain students, such as the story provided earlier, to get more explanation for why a student is classified as being successful or at risk.

2. **Diagnosis:** During the analysis, we can inspect the misclassified student sequences and interpret them as stories. Such stories can identify limitations of the analytics in classifying students. To demonstrate this, we chose two student sequences that were misclassified, and interpreted their stories to identify why our temporal model could not classify such students.

Example 1: A successful student who is classified as being at risk:

> This student is a 25-year-old white, female US citizen who is a Computer Science major. In the first four weeks, she did not participate in most of the quizzes and assignments, and did not complete the reflection tests. However, after the fourth week, she participated in all major exams, and submitted all the assignments and reflection tests. She obtained relatively high grades for the exams and assignments. Also, the reflection tests show a positive change in sentiments towards the end of the semester. Finally, the student passed the course with B.

Based on this story, the student changed her behaviour after week four and proved her improvement in later exams and assignments. However, the analytics misclassified this student as being at risk. The possible explanations for this misclassification can be: 1) low number of samples (students) to learn different kinds of student behaviours in the class, or 2) error in extracting sentiment vectors from the reflection tests. In this case, accurate sentiment vectors extracted from the reflection tests could capture the positive change in sentiment, which indicates a positive change in student performance.

Example 2: An at-risk student who is classified as being successful:

> This student is a 25-year-old white, male US citizen who is a Computer Science major. He achieved on-average grades for his quizzes and assignments during almost all the weeks. However, he unexpectedly did not perform well on the final exam. Finally, the student failed the course.

This student's story indicates that unexpected events such as low performance on the final exam can cause misclassifications. The low number of samples (students) having unexpected conditions can explain why the temporal model could not correctly classify such students.

5. A Between-Semester Sequence Model and Analysis

In this section, we describe a between-semester sequence data model for identifying at-risk students in the computer science major. We explain the properties of a between-semester sequence and present an analytic process to analyze between-semester sequences for finding patterns of success and risk.

5.1. Student Data in a Between-Semester Sequence Model

The between-semester sequence model allows the analysis of risk and success by finding patterns from data about all students in a major. We created a between-semester sequence model for College of Computing student data. We limited our analysis to undergraduate students who spent eight or more semesters at UNC Charlotte and selected computer science as their major at some point in their academic career. We chose on-time graduation as the measure of success, and built predictive models using our sequence model to identify students being at risk of not graduating in four years. The total number of student sequences was 2574, of which 30% were at risk. Our sequence model consisted of a background node (containing age, gender, citizenship type, ethnicity, marital status, and some limited information about a student's previous college or high school), semester nodes for each registered semester (containing course information as well as academic information such as major and advisor), and an outcome node recording graduation information if any.

Figure 6 illustrates the data we have selected for a sequence data model for a student in the Computer Science major: the student entered the College in Fall 2004 and graduated in 2008 after being enrolled in courses and activities for eight semesters.

Based on the example in Figure 6, we can examine properties of the between-semester sequence:

Time dependency. In general, students enroll in courses depending on the grades they achieved in the previous semesters in addition to curriculum requirements. For instance, in the sample sequence shown in Figure 6, the student completed Calculus II (MATH1242) with a C in Spring 2005 after passing Calculus I (MATH1241) in Fall 2004 with a B. The C grade for Calculus II in the Spring semester can be partially dependent on the B grade achieved in the previous Fall semester.

Contextualization. In the student sequence, we can choose any set of features to be the salient feature for the analysis. Other features provide context for the analysis. For example, in Figure 6 we choose "course level" as the salient feature shown in bold face. In our data, each course has a four-digit number in which the first digit represents the level: 1000 level courses are introductory and 4000 level courses are advanced, with 2000 and 3000 level courses in between. Given the "course level" information we can analyze the progress of a student by tracking their course enrollment.

Segmentation. In the sample sequence shown in Figure 6, we define nodes as beginning and ending with the semester dates. This grouping of student data is significant for analytic models that support university administrators and advisors because they track a student through the degree and look for patterns for success or risk in the major.

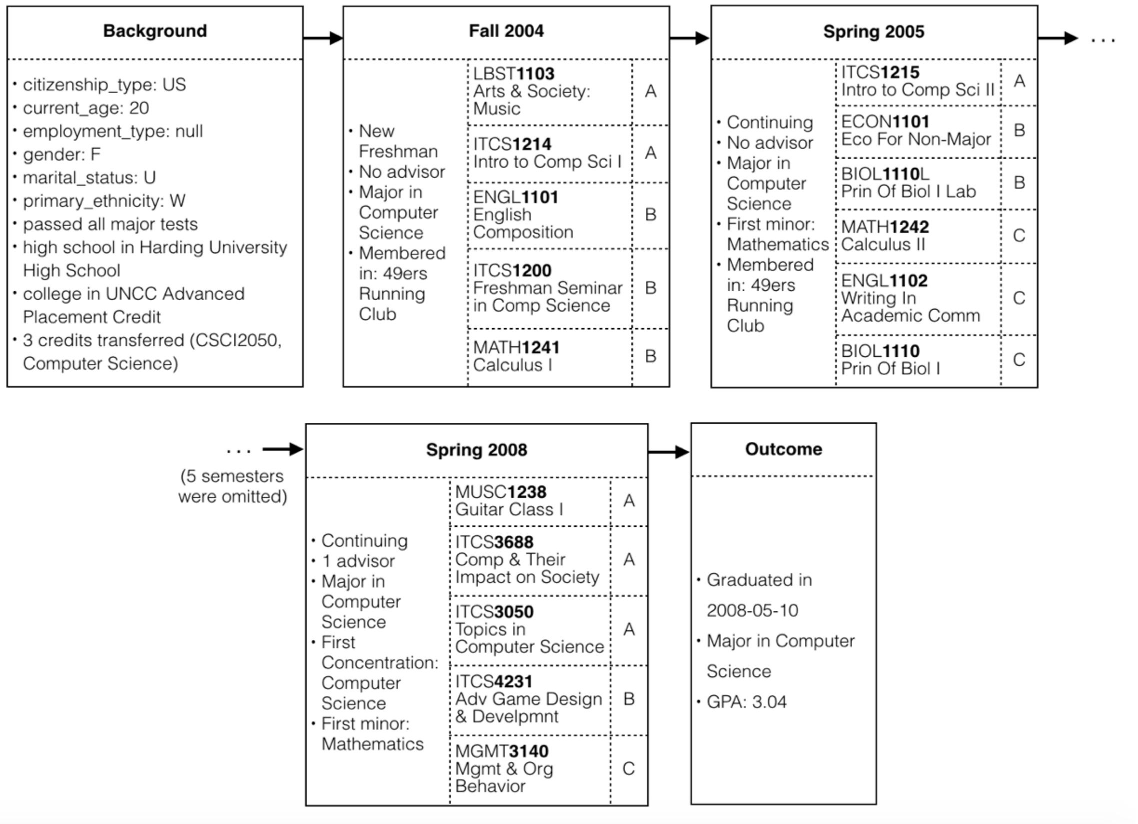
Background
• citizenship_type: US
• current_age: 20
• employment_type: null
• gender: F
• marital_status: U
• primary_ethnicity: W
• passed all major tests
• high school in Harding University High School
• college in UNCC Advanced Placement Credit
• 3 credits transferred (CSCI2050, Computer Science)
Fall 2004
• New Freshman
• No advisor
• Major in Computer Science
• Membered in: 49ers Running Club
LBST1103 Arts & Society: Music
A
ITCS1214 Intro to Comp Sci I
A
ENGL1101 English Composition
B
ITCS1200 Freshman Seminar in Comp Science
B
MATH1241 Calculus I
B
Spring 2005
• Continuing
• No advisor
• Major in Computer Science
• First minor: Mathematics
• Membered in: 49ers Running Club
ITCS1215 Intro to Comp Sci II
A
ECON1101 Eco For Non-Major
B
BIOL1110L Prin Of Biol I Lab
B
MATH1242 Calculus II
C
ENGL1102 Writing In Academic Comm
C
BIOL1110 Prin Of Biol I
C
. . .
. . .
(5 semesters were omitted)
Spring 2008
• Continuing
• 1 advisor
• Major in Computer Science
• First Concentration: Computer Science
• First minor: Mathematics
MUSC1238 Guitar Class I
A
ITCS3688 Comp & Their Impact on Society
A
ITCS3050 Topics in Computer Science
A
ITCS4231 Adv Game Design & Develpmnt
B
MGMT3140 Mgmt & Org Behavior
C
Outcome
• Graduated in 2008-05-10
• Major in Computer Science
• GPA: 3.04

Figure 6. An example of the data in a between-semester sequence model for a student with computer science as her major.

Storytelling. For the between-semester sequence model, the story is about a student's progression through the major. Such stories provide insight when a student has been identified as at risk or is an exemplar of a successful student.

5.2. Analyzing the Between-Semester Sequence Model

The data in the between-semester sequence model is the basis for the analysis of student progression through a major. We use the same process as the within-semester sequence model in Figure 4. We first re-represent the sequence data in signatures, and then analyze the signatures for classification and clustering by extracting metadata from the signatures. In this section, we present an approach to generate signatures from student data in a between-semester sequence data model, and in the next section we evaluate the model and the analytics using a dataset that includes 10 years of student data in the College of Computing at UNC Charlotte.

5.2.1. Generating Signatures from the Sequence Model

Re-representation is a process of mapping from one representation to another. In the re-representation step, we transform the data in the sequence model into signatures. Signatures can be inspected by human experts to better understand typical and unexpected student behaviour, or they can be used to extract metadata that provide the features for classification or clustering.

We generate signatures in the between-semester sequence model using a similar algorithm proposed in section 4.2, called progressive clustering. Progressive clustering applies the HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise; Campello, Moulavi, & Sander, 2013) clustering algorithm over time on each node of student sequences. The clustering algorithm groups students using the salient features of each node, and records the percentage of at-risk students in each group. After the clustering is done for all nodes, the student signatures are generated by plotting the percentage of at-risk students for each group.

Algorithm 2 explains the progressive clustering in pseudocode. The algorithm starts with all student sequences as the input. For each node *i* (excluding the outcome node), we run HDBSCAN clustering algorithm to group students based on the features in node *i*. Then, for each student sequence *S* we obtain the group (*C*) to which the student belongs and record the percentage of at-risk students in the group (P_y). This will show how much the student is similar to an at-risk population. We create a point $P=(P_x, P_y)$ in the student's signature by recording the percentage of at-risk students in *C* (i.e., P_y), along with the node index ($P_x=i$). Repeating this process for all students and all nodes will generate a signature plot for each student. Each signature plot will have *n* points where n=number of nodes (excluding the outcome node).

Algorithm 2: Progressive Clustering Using HDBSCAN

```
Input: Sequences for all students
Output: Signature plots for all students
For $i = 0$ to number of nodes (excluding outcome node)
        Cluster students based on salient features in node $i$ using HDBSCAN
        For each student $S$
                $C \leftarrow$ The cluster to which $S$ belongs
                $P_x \leftarrow i$
                $P_y \leftarrow$ Percentage of at-risk students in $C$
                Add point $(P_x, P_y)$ to the signature plot of $S$
```

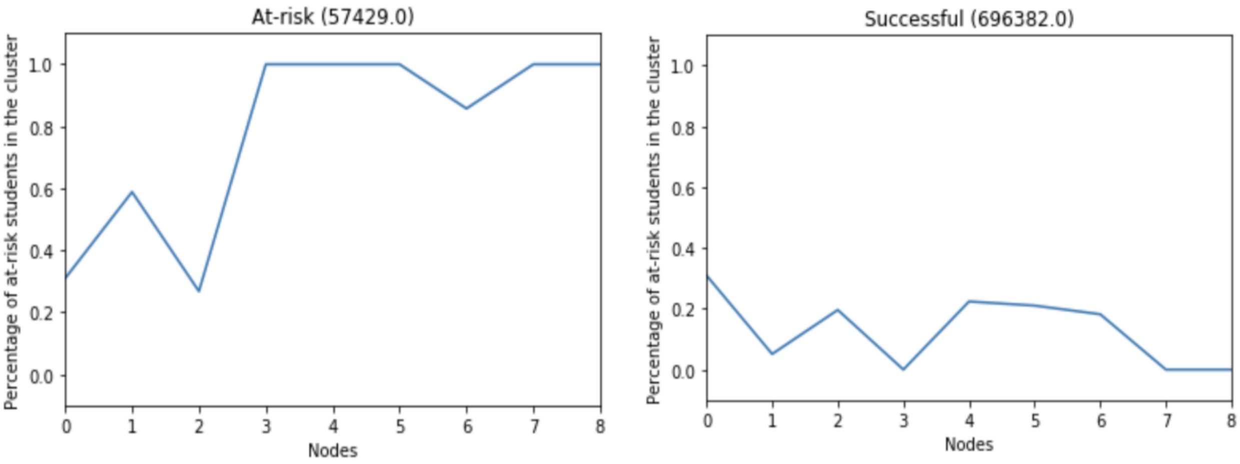

Figure 7. Signatures produced by the progressive clustering algorithm from a between-semester sequence model.

Figure 7 shows the signature of two students: one at risk of not graduating on time (left) and the other successfully graduating on time (right). In this figure, the X-axis is the node index. Since in our between-semester data model each node contains one semester of data, the node index is the semester number. Node index 0 refers to the background node. The Y-axis in Figure 7 shows the percentage of at-risk students in clusters.

To describe Figure 7, we start with the student who could not graduate on time (left). This student was clustered in a group of 30% of students who were at risk at the beginning of the semester, having only the features of the background node (x=0, y=0.3). After the second semester, however, the algorithm clustered this student with all at-risk students (x=3, y=1.0). On the other hand, the student on the right started in the same group as the at-risk student on the left (x=0, y=0.3), but remained in clusters with mostly successful students (on average 13% at risk).

Signatures generated from the progressive clustering algorithm (such as Figure 7) give us a new representation of the sequences that discriminates successful students from students at risk of not graduating on time. The next section uses this new representation as the basis to extract metadata (features) to automatically identify at-risk students.

5.2.2. Analyzing Signatures from the Between-Semester Sequence Model

Signatures by themselves can be inspected by domain experts to gain insight about a student sequence or identify unexpected behaviours. However, to be able to automatically process the signatures, detect similar structures to identify clusters, and learn to classify signatures we need to analyze the signatures by converting them into a vector representation. Then, the vector format can be input to a machine-learning algorithm to analyze the structure of signatures. For example, using support vector machines (SVM) we can classify successful students and students at risk of not graduating on time. Since each signature represents one student sequence, we can classify or cluster sequences given the signature's feature set. This section uses an approach similar to that discussed in section 4.2.2 for the within-semester sequence model of analyzing signatures.

Curve Fitting. Similar to section 4.2.2, we used curve fitting as an approach to extract features from signatures while considering the trend of the signature. Directly extracting features from the signature plot might not give us enough information about the signature data, while fitting a curve and extracting the fitted curve's coefficients ensures capturing the signature trend. As discussed in section 4.2.2, we chose a 3-degree polynomial as a sweet-spot between model complexity and sufficient information.

Figure 8 shows a 3-degree polynomial fit on the signatures generated, such as the one in Figure 7. A 3-degree polynomial curve produces four features (coefficients), which comprise the metadata of the signatures for our dataset of students. The metadata is used to construct a feature vector that characterizes the behaviour of each student in their course progression and can be used in machine learning algorithms, such as classification, clustering, or self-organizing maps (SOM).

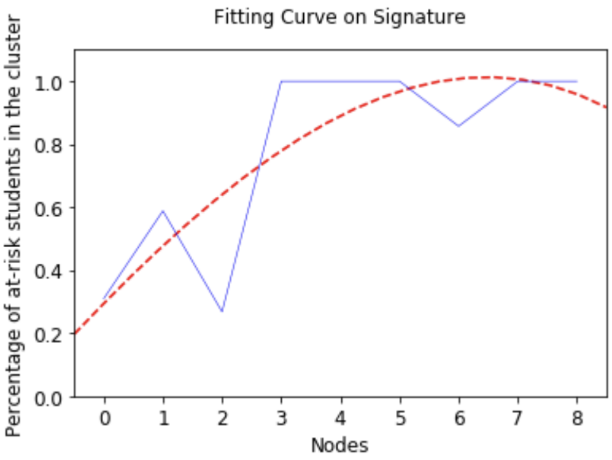

Figure 8. Fitting a 3-degree polynomial function to the signature created by the progressive clustering algorithm for a between-semester sequence model.

5.3. Evaluating the Between-Semester Sequence Model

We evaluated the benefits of the between-semester sequence model by revisiting its four features:

Time Dependency. The between-semester sequence model affords analytics that consider the time dependency between data items. The progressive clustering uses the sequence model to generate signatures that allow us to capture this temporal dependency. By fitting a curve on the signatures and extracting the metadata features (fitted curve's coefficients) we can get the trend of the data that embeds the temporal features.

To show the significance of such analytics, we built a non-temporal feature vector model from our student data with only non-temporal features, such as demographics and statistic features, and compared its performance with a temporal model that includes temporal features (signature's metadata features) in addition to the non-temporal features. Similar to evaluating a within-semester sequence model (section 4.3), we evaluated the comparison in three separate groups based on the features included in the models.

Group 1: Including background features, excluding statistical features. The background features included in the models are: age, gender, citizenship type, primary ethnicity, marital status, previous institution (college or high school), previous school GPA (if there is any), previous school rank and size. Both non-temporal and temporal models have the background features, but the temporal model adds four temporal features extracted from the student signatures as discussed in section 5.2.2. After building temporal and non-temporal feature vector models, we used support vector machines (SVM) for both models to classify at-risk students. We tested the temporal model versus the non-temporal model for their accuracy over 10-fold cross-validation with different settings and summarized the results in Table 5. Based on Table 5, the temporal model outperforms the non-temporal model in all cases over 16.6% on average.

Group 2: Excluding background features, including statistical features. The statistical features included in the models are as follows: percentage of courses with an A to D grade, percentage of F courses, average course level taken, percentage of IT courses with an A to D grade, percentage of IT courses with an F, and average IT course level taken. Both temporal and non-temporal models include the statistical features, but the temporal model adds four more temporal features. We used the same process as for the previous group to compare the average accuracy of the models. Table 6 shows the performance results for this group. Based on the table and comparison with the previous group, statistical features generally provide better results, and the temporal model outperforms the non-temporal by 8% on average. However, the non-temporal model marginally performs better (by 0.66%) compared to the temporal model that uses “course grade” as the only salient feature. This means that the temporal information in course grades and their progression in time does not provide enough information to classify at-risk students.

Table 5. Comparing a non-temporal model with a temporal (between-semester) model in group 1. Both models include background information, but exclude statistical features.

Model	Background Info	Statistical Features	Salient: Course Level	Salient: Course Grade	Average Accuracy
Non-Temporal Model	✔	✘	Not Applicable	Not Applicable	71.13
Temporal Model	✔	✘	✔	✘	**95.69**
			✘	✔	71.99
			✔	✔	95.65

Table 6. Comparing a non-temporal model with a temporal (between-semester) model in group 2. Both models exclude background information, but include statistical features.

Model	Background Info	Statistical Features	Salient: Course Level	Salient: Course Grade	Average Accuracy
Non-Temporal Model	✘	✔	Not Applicable	Not Applicable	84.77
Temporal Model	✘	✔	✔	✘	**97.20**
			✘	✔	84.11
			✔	✔	97.09

Group 3: Including both background features and statistical features. We combined features from groups 1 and 2 and compared the performance of temporal and non-temporal models in Table 7. As shown in the table, the accuracy of the temporal model is on average 7% better than the non-temporal model. Comparing group 3 with the previous two groups shows that including background features marginally improves the model’s performance. Similar to the previous groups, the results also confirm that “course grade” is not a predictive salient feature to be included in analysis.

Based on the results shown in Tables 5–7, the temporal model has better accuracy over the non-temporal model. Also, in cases where we include “course level” as the only salient feature, we obtain the maximum accuracy for the temporal model (95.69%, 97.20%, and 96.08%), which is on average 9.5% better than the best non-temporal model. We expect to see a decrease in this gap by adding more statistical or background features.

Table 7. Comparing a non-temporal model with a temporal (between-semester) model in group 3. Both models include background information and statistical features.

Model	Background Info	Statistical Features	Salient: Course Level	Salient: Course Grade	Average Accuracy
Non-Temporal Model	✔	✔	Not Applicable	Not Applicable	85.39
Temporal Model	✔	✔	✔	✘	**96.08**
			✘	✔	85.18
			✔	✔	95.92

Another observation from the results presented in Tables 5–7 is that the background features in our model do not play an important role in classifying students in our dataset. Most of the research in finding students at risk includes both time variant features — such as GPA, which changes over semesters — and time invariant features, such as student demographics. Lakkaraju et al. (2015) reports that in some cases "gender" was among the best predictors along with GPA and math proficiency for building an early warning system for high schools. However, Er (2012) argues that time invariant features have no effect on the classification task. Er (2012) applied an instance-based learning classifier on only time varying features, such as partial grades in three stages of each semester of an online MOOC, and compared the results to approaches that used both time varying and time invariant features. Based on Er (2012) there is no significant difference between approaches, which means that time invariant features have less influence on classifiers in finding at-risk students in the online course system.

Contextualization. The sequence model uses salient features for the analytics and context features to get insight and help in interpreting the results. In our evaluation of the between-semester sequence model, we chose different subsets of features (course level and course grade) as being salient. Based on Tables 5–7, having only course level as the salient feature boosts the model's accuracy in all cases to above 95%. However, course grade does not improve the accuracy at all.

Segmentation. Having the sequence model gives us more flexibility to choose the granularity of sequence nodes and data aggregation over time. In our between-semester model, sequence nodes aggregate "semester level" data, which makes sense for tracking student progress through the major.

Storytelling. We can interpret the data about a student's progression in the major in forms of stories from a between-semester sequence model. As an example, a successful student sequence (shown in Figure 6) can express the following story:

> This student is a 20-year-old white female US citizen who came from a relatively good high school and college. She passed all major proficiency tests and enrolled in Fall 2004. She did not have any employment records (within or outside the university) during her study. She started with a major in computer science, and successfully completed all core courses in this major by her fourth semester. During this time, her grades were C or above in all courses. She chose an advisor in the fourth semester. In the fifth semester, she dropped the course ITCS 2215 (design and analysis of algorithms), but achieved an A retaking the same course the next semester. Throughout her enrollment in this major, she maintained all her grades at C or above. She finally graduated after 8 semesters in May 2008.

Similar to the within-semester analytics (Section 4.3), we can use the storytelling feature to diagnose the analytics. To demonstrate this, we identified two student sequences that were misclassified, and interpreted their stories to identify why our temporal model could not classify such students.

Example 1*:* A successful student who is classified as being at risk:

> This student is a 24-year-old female US citizen who started with a major in mathematics. In her third semester, she started taking computer science major core courses. She passed the core courses with average grade of B. In her fourth semester, she changed her major to computer science. She transferred her courses from the previous semesters. After changing her major to computer science, her grades were C or above in all courses. She finally graduated after 8 semesters.

Since the analytics uses only the course-level and course grade as the salient features to classify students, the course taking behaviour of the student plays an important role in the classification. Students such as the one in the above example do not have a common course-taking behaviour. For instance, the above student starts taking core courses in the third semester, changes her major, and transfers many courses in the fourth semester, whereas the common course-taking behaviour is to take the core courses in the first semester and progress afterwards. Such examples indicate the limitations of the analytics when using only course-level and course grade as the salient features and also indicate the benefit of including information about student transfers in the analytics.

Example 2*:* An at-risk student who is classified as being successful:

> This student is a 26-year-old white male US citizen who came from a very good high school and college. He passed all major tests prior to his enrollment. He started with a Computer Science major and passed all core courses. His performance in courses in the major was above average for two years. After two years, he left the university and no further records of him exist in the system.

Another limitation of the analytics is its inability to distinguish sequences that end unexpectedly, such as the above example where the student unexpectedly leaves the university. He should therefore be labelled as being at risk, even though his performance is good and is similar to the successful students. The analytics classifies this student solely based on his similarity to successful students, and does not consider the possibility of attrition for successful students.

6. Conclusion

Temporal relationships among student data are the basis for understanding trends in student behaviours, and identifying behaviours leading to success or failure. This paper presents the concept of a sequence data model as a repository of heterogeneous student data that captures temporal relationships explicitly in meaningful temporal ranges we call nodes. We demonstrate the benefits in the sequence data model by showing how temporality improves the accuracy of predictive models and has the ability to identify trends and unexpected patterns in data.

We define sequences as a repository of student data ordered by time that groups heterogeneous data about students into nodes. We show how the sequence data model allows analytic models to include temporal dependencies and can be used to group student data in nodes with different time-based granularity. The sequence data model can enable an analysis around a salient feature such as “course level” while maintaining the representation of the contextual data. This model enables an iterative analytic process over different salient features to find predictors with the highest accuracy. Another feature of the sequence data model is that a student sequence more strongly affords narrative interpretation about the student. The storytelling feature of the sequence can be used after the analytics, for example to inspect for a student’s unexpected behaviour detected by the analytics.

We present an analytic process in which we first re-represent the sequences into simpler structures called signatures, and then extract Figure 7 shows the signature of two students: one at risk of not graduating on time (left) and the other successfully graduating on time (right). In this figure, the X-axis is the node index. Since in our between-semester data model each node contains one semester of data, the node index is the semester number. Node index 0 refers to the background node. The Y-axis in Figure 7 shows the percentage of at-risk students in clusters.

To describe Figure 7, we start with the student who could not graduate on time (left). This student was clustered in a group of 30% of students who were at risk at the beginning of the semester, having only the features of the background node (x=0, y=0.3). After the second semester, however, the algorithm clustered this student with all at-risk students (x=3, y=1.0). On the other hand, the student on the right started in the same group as the at-risk student on the left (x=0, y=0.3), but remained in clusters with mostly successful students (on average 13% at risk).

Signatures generated from the progressive clustering algorithm (such as Figure 7) give us a new representation of the sequences that discriminates successful students from students at risk of not graduating on time. The next section uses this new representation as the basis to extract metadata (features) to automatically identify at-risk students.the re-representation of sequences into signatures using progressive classification, and progressive clustering algorithms, for both models. The progressive clustering algorithm iteratively uses HDBSCAN to cluster student data using salient features for each node, and builds signatures for students by capturing the percentage of at-risk students in each cluster. The progressive classification iteratively uses SVM to classify students using salient features for each node, and generates signatures for students by recording the classifier’s confidence. We applied a 3-degree polynomial curve fitting to extract metadata from signatures, and used the coefficients of the fitted curves as the temporal features.

To evaluate the between- and within-semester sequence models, we compared the predictive accuracy with non-temporal models using the same data. Based on the comparison results, we show that temporal models generally outperform non-temporal models by about 9.5% in performance accuracy. For the between-semester model, "course level" was the best salient feature that produced over 97% accuracy; for the within-semester model, "reflections" and "tests" were salient features that gave more than 95% accuracy on average.

As future work, we will be evaluating the performance of our sequence model by selecting different salient features to understand their predictive power. This evaluation can be done incrementally over sequence nodes to assess how the temporal model's accuracy changes over time as we change the number of nodes available to the analytics. For example, we can explore the accuracy of the model's prediction for a between-semester sequence model with only the first four nodes. Also, analyzing different measures of accuracy such as recall, precision, and F-measure will benefit in diagnosing the sequence analytics.

Finally, we are exploring the design of an interactive system that enables a professor or advisor to select the features in a student sequence model and explore several analytic models. Currently, the process of creating the sequence model, selecting salient features, running the analytics, and evaluating the model is done by a data scientist. We plan an interactive system that makes this process accessible to the domain experts by designing an exploratory interactive framework similar to that shown in Figure 1. We will also expand our dataset to include a broader range of students for analytics, and extend the data for each student with new activity logs, such as library access records and dining plans. Another future research direction is to automate the process of building stories from sequences, and identify possible points of anomalies within the sequences and stories. We expect many challenges in this process, such as translating heterogeneous data items in nodes into intelligible sentences, and connecting sentences to build a coherent story.

Acknowledgements

We acknowledge Dr. Shannon Schlueter and Dr. Audrey Rorrer for their assistance in gaining access to the student data stored in the university databases. We thank Dr. Mohsen Dorodchi and his teaching assistants Devansh Desai and Aileen Benedict for helping collect/clean the data and their insights about the students in their introductory computer science class to enable the within-semester sequence model analysis. We had numerous discussions about our ideas for modelling student data with a broad range of faculty, of whom we especially thank Dr. Bojan Cukic, Dr. Xi (Sunshine) Niu, and Dr. Noseong Park.

Declaration of conflicting interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research was supported by the Charlotte Research Institute.

References

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM Sigmod Record*, *22*(2), 207–216. http://dx.doi.org/10.1145/170036.170072

Arnold, K. E. (2010). Signals: Applying academic analytics. *Educause Quarterly*, *33*(1), n1.

Arnold, K. E., & Pistilli, M. D. (2012). Course signals at Purdue: Using learning analytics to increase student success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (LAK 'LA), 29 April–2 May 2012, Vancouver, BC, Canada (pp. 267–270). New York: ACM. http://dx.doi.org/10.1145/2330601.2330666

Campbell, J. P. (2007). *Utilizing student data within the course management system to determine undergraduate student academic success: An exploratory study*. PhD dissertation. Purdue University. https://docs.lib.purdue.edu/dissertations/AAI3287222/

Campbell, J. P., DeBlois, P. B., & Oblinger, D. G. (2007). Academic analytics. *EDUCAUSE Review*, *42*(4), 40–57.

Campello, R. J., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, & G. Xu (Eds.), Advances in Knowledge Discovery and Data Mining: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2013), 14–17 April 2013, Goldcoast, Queensland, Australia. *Lecture Notes in Computer Science*, vol. 7819 (pp. 160–172). Springer, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-642-37456-2_14

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. 10.1007/BF00994018

Er, E. (2012). Identifying at-risk students using machine learning techniques: A case study with IS 100. *International Journal of Machine Learning and Computing*, *2*(4), 476–480. http://www.ijmlc.org/papers/171-L003.pdf

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '01), 26–29 August 2001, San Francisco, CA, USA (pp. 97–106). New York: ACM. http://dx.doi.org/10.1145/502512.502529

Jayaprakash, S. M., Moody, E. W., Lauría, E. J., Regan, J. R., & Baron, J. D. (2014). Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics*, *1*(1), 6–47. 10.18608/jla.2014.11.3

Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. L. (2015). A machine learning framework to identify students at risk of adverse academic outcomes. *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '15), 10–13 August 2015, Sydney, NSW, Australia (pp. 1909–1918). New York: ACM. http://dx.doi.org/10.1145/2783258.2788620

Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & Education, 54*(2), 588–599. http://dx.doi.org/10.1016/j.compedu.2009.09.008

Maher, M. L., & Mahzoon, M. J. (2015). Finding unexpected patterns in citizen science contributions using innovation analytics. *Proceedings of Collective Intelligence 2015*. 31 May–2 June 2015, Santa Clara, CA, USA https://sites.lsa.umich.edu/collectiveintelligence/wp-content/uploads/sites/176/2015/05/Maher-and-Mahzoon-CI-2015-Abstract.pdf

Mohamad, S. K., & Tasir, Z. (2013). Educational data mining: A review. In C. S. Teh et al. (Eds.), The 9th International Conference on Cognitive Science. *Procedia: Social and Behavioral Sciences*, 97, 320–324 (6 November 2013). http://dx.doi.org/10.1016/j.sbspro.2013.10.240

Padmanabhan, B., & Tuzhilin, A. (1999). Unexpectedness as a measure of interestingness in knowledge discovery. *Decision Support Systems*, *27*(3), 303–318. http://dx.doi.org/10.1016/S0167-9236(99)00053-6

Peña-Ayala, A. (2014). Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, *41*(4, Part 1), 1432–1462. http://dx.doi.org/10.1016/j.eswa.2013.08.042

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, *33*(1), 135–146. http://dx.doi.org/10.1016/j.eswa.2006.04.005

Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, *51*(1), 368–384. http://dx.doi.org/10.1016/j.compedu.2007.05.016

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology, 29*, 24–54. http://dx.doi.org/10.1177/0261927X09351676

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, *23*(1), 69–101. http://dx.doi.org/10.1023/A:1018046501280

Wolff, A., Zdrahal, Z., Nikolov, A., & Pantucek, M. (2013). Improving retention: Predicting at-risk students by analyzing clicking behaviour in a virtual learning environment. *Proceedings of the 3rd International Conference on Learning Analytics and Knowledge* (LAK '13), 8–12 April 2013, Leuven, Belgium (pp. 145–149). New York: ACM. http://dx.doi.org/10.1145/2460296.2460324