JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Volume 5(3), 136—153. http://dx.doi.org/10.18608/jla.2018.53.9

Topic Dependency Models: Graph-Based Visual Analytics for Communicating Assessment Data

Hassan Khosravi,[1] Kendra M.L. Cooper[2]*

Abstract

Educational environments continue to evolve rapidly to address the needs of diverse, growing student populations while embracing advances in pedagogy and technology. In this changing landscape, ensuring consistency among the assessments for different offerings of a course (within or across terms), providing meaningful feedback about student achievements, and tracking student progress over time are all challenging tasks, particularly at scale. Here, a collection of visual Topic Dependency Models (TDMs) is proposed to help address these challenges. It uses statistical models to determine and visualize student achievements on one or more topics and their dependencies at a course level reference TDM (e.g., CS 100) as well as assessment data at the classroom level (e.g., students in CS 100 Term 1 2016 Section 001), both at one point in time (static) and over time (dynamic). The collection of TDMs share a common two-weighted graph foundation. Exemplar algorithms are presented for the creation of the course reference and selected class (static and dynamic) TDMs; the algorithms are illustrated using a common symbolic example. Studies on the application of the TDM collection on datasets from two university courses are presented; these case studies utilize the open-source, proof of concept tool under development.

Notes for Practice

- The development of visualization techniques tailored specifically for the design, delivery, and analysis of assessment material seems, as yet, to be under-developed and under-explored.
- This paper presents a collection of Topic Dependency Models (TDMs) that use a common graph foundation to address some of the challenges stakeholders (students, instructors, administration, researchers) face relating to the design, delivery, and analysis of assessment.
- Two case studies with a focus on computer science education are presented that suggest that instructors can benefit from the TDMs.
- TDMs may be hard to comprehend for stakeholders without formal training in algorithmic literacy. Future work will investigate the use of TDMs among different stakeholders and disciplines with less formal training in algorithmic literacy; modifications to make TDMs more comprehensible for these audiences will also be explored.

Keywords

Visual Analytics, Learner Models, Graph Algorithms, Student Assessment

Submitted: 27.05.2018 — **Accepted:** 30.08.2018 — **Published:** 11.12.2018

[1] *Institute for Teaching and Learning Innovation, University of Queensland, St Lucia QLD 4072, Australia. Email: h.khosravi@uq.edu.au*
[2] *Independent Scholar, Kelowna, V1X 7L8, Canada. Email: kendra.m.cooper@gmail.com*

1. Introduction

Educational environments continue to evolve rapidly to address the needs of diverse growing student populations while embracing advances in pedagogy and technology. This evolution has resulted in numerous environments such as traditional, e-learning, flipped classrooms, and MOOCs, supported by a wide range of tools and techniques (Englund, Olofsson, & Price, 2016). As the environments change, assessment remains a core educational activity: the design and delivery of high quality assessments, in particular providing rich and timely feedback at scale, becomes even more challenging.

A wide range of educational stakeholders are involved in the design, delivery, and analysis of assessment data including those in the classroom (students, instructors), outside the classroom (course coordinators, course designers, program administrators), and educational researchers (Carless, 2006; Riordan T., 2009). Inside a classroom, instructors may find it challenging to evaluate

the quality of assessment material in multiple forms with respect to a well-defined description of the required course content (topics and the topic dependency relationships covered). In addition, effectively communicating personalized feedback at scale is recognized as a difficult problem (Khan & Pardo, 2016; Khosravi & Cooper, 2017). Students may find it challenging to infer their strengths and weaknesses with respect to the topics and their relationships, which can impede their studies. Outside a classroom program administrators, course designers, course coordinators, and researchers also face challenges. Administrators find it challenging to compare the content and difficulty of formal assessments as well as student outcomes across different offerings of a course. Course designers and coordinators find it challenging to ensure the required topics and their relationships (e.g., questions with a combination of topics a, b and c) have been assessed. Educational researchers are often required to compare the achievement results between control and experimental groups.

Here, three core issues are explored. The first is the need for course level models to communicate the scope and coverage of assessment material, providing a common foundation for all stakeholders. The second is the need for classroom level models to visualize assessment data within a class and support comparisons of assessment data between classes at one point in time. The third is the need for models that visualize assessment data trends over time. The solution is a collection of graph-based Topic Dependency Models (TDMs); the results presented extend the preliminary findings reported in Cooper and Khosravi (2018).

The proposed TDMs draw upon the rich literature available on learner models that represent students in the learning process. A number of studies provide strong evidence that opening the model to learners, leading to the notion of Open Learner Models (Bull & Kay, 2010), can be effective in helping students learn (Bodily et al., 2018). OLMs commonly use a set of individual topics as their underlying structure for modelling the learner knowledge state, which ignores relationships among topics. TDMs provide an extension by using graphs as their underlying structure for modelling the learner knowledge state, which also allows for considering relationships among topics. This research explores how an OLM can be defined to formally represent and visualize a wide variety of assessment data for one or more topics (i.e., topics and their dependency relationship) to meet the needs of diverse stakeholders.

The collection of TDMs shares a common graph foundation (two-weighted graph). Stakeholders can select the model and the assessment data to use. The algorithms to create the graph-based models and their visualizations are illustrated using a small, symbolic example created by the authors. Two case studies are presented to demonstrate the application of the models on historical data. These case studies utilize an open-source implementation of the TDMs (GitHub, 2017).

The remainder of this paper is organized as follows. Related work is presented in Section 2. The two-weighted graph foundation for the models is presented in Section 3. An overview covering the stakeholder and scenario analysis, high level description of the TDM collection, and the illustrative example used in the more detailed discussions of the models are in Section 4. The TDMs (reference, static, and dynamic models) are presented in more detail in Sections 5, 6, and 7 respectively. The studies exploring the application of the models to two cases studies are in Section 8. Conclusions and future work are in Section 9.

2. Related Work

The development of graph-based visual analytics for communicating assessment data draws upon the literature from two main related research domains: 1) development of statistical learner models that can capture student performance and 2) techniques that have the capacity to visualize these models for different stakeholders.

Learner models have been heavily studied in the educational data mining community with applications in intelligent tutoring systems (e.g., (D'Mello et al., 2010)), adaptive learning environments (e.g., (Oxman, Wong, & Innovations, 2014)), and recommender systems (e.g., (Khosravi, Cooper, & Kitto, 2017)). The Bayesian Knowledge Tracing (BKT) algorithm (Corbett & Anderson, 1994) is one of the most prominent methods used for modelling learners; this line of research has focused on individual topics. BKT uses Hidden Markov Models (HMMs) to capture the student knowledge states as a set of binary variables that represent whether or not a concept has been mastered. BKT uses Slip and Guess parameters where slipping refers to the situation where a student has the required skill for answering a question but mistakenly provides the wrong answer; guessing refers to the situation where a student provides the right answer despite not having the required skill for solving the problem. Later on, Pardos and Heffernan (2011) effectively extended BKT to capture item difficulty, which led to improved prediction accuracy. More recently, Yudelson, Koedinger, and Gordon (2013) further improved BKT by introducing a new set of parameters that captured the prior knowledge of individual learners. Other algorithms with comparable or superior predictive power to BKT have also been proposed for modelling learners. Cen, Koedinger, and Junker (2006) introduced the Learning Factors Analysis Framework, and Pavlik, Cen, and Koedinger (2009) introduced the Performance Factor Analysis Framework. More recently, Piech et al. (2015) and Sha and Hong (2017) used recurrent neural networks for deep knowledge tracing.

Visualizing student performance has been studied in two different communities. In the educational data mining community, understanding and visualizing the machine representation of the learner models has led to the notion of Open Learner Models (OLMs) (Bull & Kay, 2010). In the learning analytics community, helping different stakeholders make sense of educational datasets has led to the work on visualizing learning dashboards (Schwendimann et al., 2017). A recent literature review of

ISSN 1929-7750 (online).

OLMs and learning dashboards by Bodily et al. (2018) concluded that the two share many similarities. A key difference, however, is that research on OLMs has been mostly grounded in user modelling and artificial intelligence in education, whereas the research on dashboards has been more broadly grounded in data-driven decision making (often outside of the context of education). Benefits of both OLMs and learning dashboards have been framed mostly around promoting meta-cognitive activities such as reflection, planning, and self-regulation (Bull & Kay, 2010; Beheshitha, Hatala, Gašević, & Joksimović, 2016). OLMs and learning dashboards rely predominantly on commonly used visualization techniques which visualize student competencies on independent topics. For example, the use of bar charts (Dawson, Macfadyen, Risko, Foulsham, & Kingstone, 2012), box plots (Albert, Nussbaumer, & Steiner, 2010), radar graphs (May, George, & Prevot, 2011), and skill meters (Bull, Ginon, Boscolo, & Johnson, 2016) has been popular in both OLMs and learning dashboards.

Searching the literature related to capturing and visualizing student performance reveals that the issues around combining knowledge on two or more topics have received very little attention. Furthermore, the development of visualization techniques tailored specifically for the design and delivery of assessment material seems, as yet, to be under-developed and under-explored (Ellis, n.d.) offering many research opportunities. The new models and their visualizations presented in this work, TDMs, contribute to addressing this gap.

3. Two-Weighted Graph Foundation of the TDMs

The aim of the TDMs is to capture and communicate the coverage and achievements of a variety of forms of assessments among a range of stakeholders. Graphs and their visualizations are widely used to demonstrate the structure of complex data in a formal way. They can summarize a large of amount of data in a compact form that is straightforward to understand by a broad audience, which makes them ideal for representing TDMs. In this work, the two-weighted graph is adopted as the foundation. These graphs support the visualization of coverage and achievements on the edges using, for example, line colour and width.

Definition 1 *The topic dependency model (TDM) is represented using a two-weighted, undirected graph $G = (V, E)$, where V is the set of vertices representing the topics and E is the set of edges representing information on learning objects (e.g., questions) that are tagged with both topics. An edge $e \in E$ is represented as $e = (v, w, cov, achv)$, where v and w are vertices (topics) being connected, cov represents the number of learning objects that are tagged with both topics v and w (e.g., coverage), and $achv$ represents the performance on learning objects that are tagged with both topics v and w (e.g., achievements). An edge connecting a node v to itself (loop) contains information on learning objects that are only tagged with topic v.*

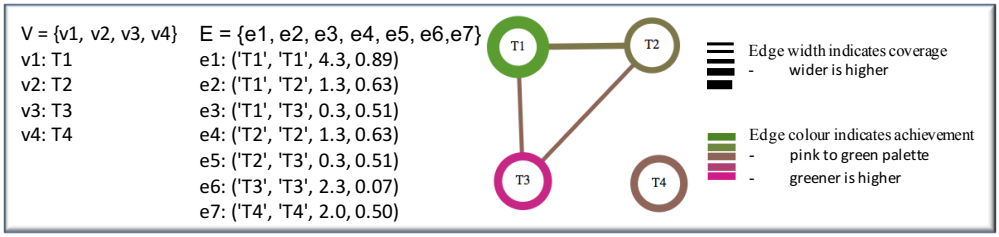


Figure 1. Symbolic example of a Topic Dependency Model (two-weighted graph) using colour and width to visualize edge weight values.

The use of two-weighted, undirected graphs as the formal underlying definition of the TDM allows for visualizations that have straightforward interpretations. As with any visualization, however, training is needed to assist those not familiar with graphs to understand the representations. The adopted visualization (*Cov*) is mapped to the thickness of an edge to represent coverage and (*Achv*) is mapped to a colour of an edge to represent achievement. If achievement data are not available, for example when a question is not answered by any student, a default edge colour is used.

Figure 1 illustrates an example of a TDM. Generally, these models can be generated manually or automatically from assessment scores. This example is demonstrating an assessment that spans four different topics. In this model the edge thickness indicates coverage based on the number of questions, where a thicker line indicates a higher coverage. The colour of the edges span a gradient from pink to green indicating achievements, where a greener line indicates a higher achievement. This gradient has been selected from established colour-blind-friendly palette options (Kirk, 2012).

For example, the edge between T1 and T2 indicates that the combination of these topics has received a relatively high coverage in the assessment and the average class score on these questions is relatively high. To simplify the visualization of the graph, the self-loops are integrated into the presentation of the vertices using colour and line width. The structure of the graph reveals that T4, an isolated vertex, is a topic that is not covered in combination with any other topic.

ISSN 1929-7750 (online).

Visualizing the coverage of questions tagged with one or two topics is straightforward using two-weighted graphs. However, in practice, questions may cover three or more topics (e.g., *a*, *b*, and *c*). There are several options that can be considered for visualizing these situations. The first is to distribute the coverage among all possible combinations, for example edges spanning (a,b), (a,c), and (b,c) using a weighting function. This option offers a complete visualization of the data; however, the visualization may become difficult to understand as the model includes edges that represent fractional contributions of questions. A second option is to treat the list of topic tags as a prioritized list and only consider the first two tags. This option relies on the data being correctly collected (i.e., prioritized tags), which may be difficult to implement in practice; this option does not support a complete visualization of the data. A third option is to visualize the coverage of questions that have up to two tags, and analyze the remaining questions separately; this option also does not support a complete visualization of the data. Here, the first option is used; the trade-offs among these options will be explored in future work with user studies.

4. Overview

4.1 Stakeholder and Scenario Analysis

A wide range of educational stakeholders including those in the classroom (e.g., students, instructors), outside the classroom (e.g., course coordinator, course designer, program administrator), or conducting educational research interact with assessment data. These stakeholders have distinct assessment goals; a collection of example scenarios is summarized in Table 1. These scenarios are the result of considering classic questions (who? what? where? when? why? how?) to identify stakeholder needs.

Within a classroom, students and instructors need to interact with assessment data in several ways. First, there is a need to share a common understanding of the educational goals for a class in terms of the required course content (topics and the dependency relationships covered). This establishes a core, common foundation of reference for all classes offered in a course.

Second, there is a need to communicate the assessment achievements for both individual students and a cohort in a class; one or more assessments can be selected that span formative and summative material. For example, achievements on an examination or several assignments and a collection of questions from a peer-learning repository can be explored. Comparisons across classes are useful to evaluate, for example, the coverage and achievements of an assessment item (e.g., homework, quiz, lab, examination) with respect to one used in a previous offering of the class. This is valuable for instructors who do not have previous experience teaching the course, as the experience of previous instructors is embodied in their assessment material.

Third, outside the classroom, program administrators and course designers need to establish and monitor the required content for each course, as part of a program curricula. Summaries of assessment data are needed to periodically analyze the coverage of the required course content in program and course level reviews (e.g., a program accreditation audit). Class coordinators may be responsible for monitoring multiple, large sections running concurrently and/or over multiple terms. The classes need to employ large teams of teaching assistants (undergraduate and graduate students) who change from term to term; different instructors are also assigned over time. A course coordinator can monitor the assessment data to help ensure the consistency of many sections running concurrently; in addition they can periodically analyze the coverage of the required course content, perhaps on a term or annual basis. From this broader perspective, the coordinator can identify, investigate, and address on-going problem areas in a course such as poor performance on particular topics. Here again, static and dynamic visualizations of assessment data would be valuable.

Fourth, educational researchers need to explore the results between, for example, control and experimental groups. Static and dynamic models can provide valuable insights, for example, with respect to pre- and post-test analysis (Campbell & Stanley, 1966) on the identification of trends.

4.2 The Topic Dependency Model Collection

Based on these stakeholder needs, a collection of TDMs for exploring assessment data is proposed that spans course and class levels. An overview of the collection is illustrated in Figure 2. The stakeholders are shown at the top, the TDMs in the middle, and the assessment data input sources to the left.

A reference model (further discussed in Section 5) is proposed as part of the collection, which establishes a standard for a course by defining the course assessment material in terms of topics, dependency relationships, and the historical level of achievement obtained. The model specifies what needs to be assessed in a course, but does not constrain how this is accomplished in the classroom. For example, instructors have the flexibility to design their assessments that achieve the course requirements by using their preferred combination of summative and formative material (homework, labs, quizzes, examinations, and/or repository of multiple choice questions in peer-based learning environments). This model provides a common assessment foundation for all classes on a course.

Both static TDMs (further discussed in Section 6) and dynamic TDMs (further discussed in Section 7) are needed by the stakeholders at the class level, which can utilize a wide variety of assessment data to explore data within one class and comparisons across two classes (e.g., homework assignments for individual students or an entire class). From a user's perspective, the static class level TDMs support visualizations at one point in time; dynamic class level TDM supports

ISSN 1929-7750 (online).

Table 1. Examples of Stakeholders and Scenarios of Use

Stakeholders	Goal (question)
Engaging within the classroom	
Students	What topics do I need to know? How are the topics related? Why are the topics important? Where is material on the topics available? What topics are covered on the upcoming assessment (e.g., examination)? How well have I mastered the topics? How well am I performing in comparison to to my classmates? What topics do I need to improve my knowledge on? How much have I improved over time on the topics? ...
Instructors	What topics and their relationships do I need to assess? How can I communicate the importance of the topics? How can I communicate the topic relationships? What material do I need to provide? What topics will be covered each of the assessments? How well are the students performing on the topics? How well are the students performing compared to other (current or previous) classes? What topics do I need to improve my material or presentation on? Who in the class may be at risk of failing the course? How have all the topics and their relationships been assessed? How much has the class improved over time on the topics? ...
Administering the classroom (hierarchy)	
Course Level	What topics and their relationships are assessed across the classes for a course? Where are the inconsistencies across classes? How well are the students performing across (current or previous) classes? What are the topics students consistently perform poorly on? When the class is complete (end of quarter/term/year), have all the topics and their relationships been assessed? How relevant are the topics and dependencies for the course? What trends are emerging over time across the classes? ...
Program Level	What are the trends across courses with respect to student performance? What are the trends across courses with respect to the assessment coverage of topics and their relationships? What are the trends across instructors with respect to student performance? What are the trends across instructors with respect to the assessment coverage of topics and their relationships? How relevant are the topics and their relationships that are covered across courses? What trends are emerging over time in the program? ...
Extending the research foundation	
Research Team	For specific questions under investigation (e.g., new models, visualizations) researchers explore: What are the assessment trends using a new approach (self-evaluation)? How do assessments using a new approach behave in comparison to an established approach? (cross-evaluation)? What are the trends across groups of students? What are the assessment trends across instructors? What are the assessment trends across different courses in the same program (e.g., computer science: introduction to programming, database)? What are the assessment trends across different courses in different programs (e.g., computer science, anthropology)? ...

the visualizations to help identify trends over time. From a model definition perspective, the static TDM visualizes the dataset selected using the entire time interval in the dataset (based on time-stamps); the dynamic TDM visualizes the dataset progressively using sub-intervals of time. A stakeholder can select the TDM and the assessment data to use, which provides a flexible solution. The Course Reference TDM is included as a grayed out backdrop in visualized class models; this helps stakeholders conceptualize the bigger picture for the course in terms of prior, current, and future topics.

4.3 Illustrative Example: Input Data

An example based on commonly available course assessment data – containing three students (s1, s2, s3), five questions (q1, q2, q3, q4, q5), four topics (T1, T2, T3, T4), 12 student responses, and 10 question tags – was prepared by the authors for illustration purposes. These data are represented in the form of two input files: 1) A student-response file that contains the student identifier, question identifier, score (correct/incorrect), and the time-stamp and 2) a question-topic file that contains the question identifier and topics (i.e., tags) it addresses. The example has been thoughtfully designed to present the expected output of TDMs under a range of conditions. The constraints and a sample dataset meeting these constraints are presented in Table 2.

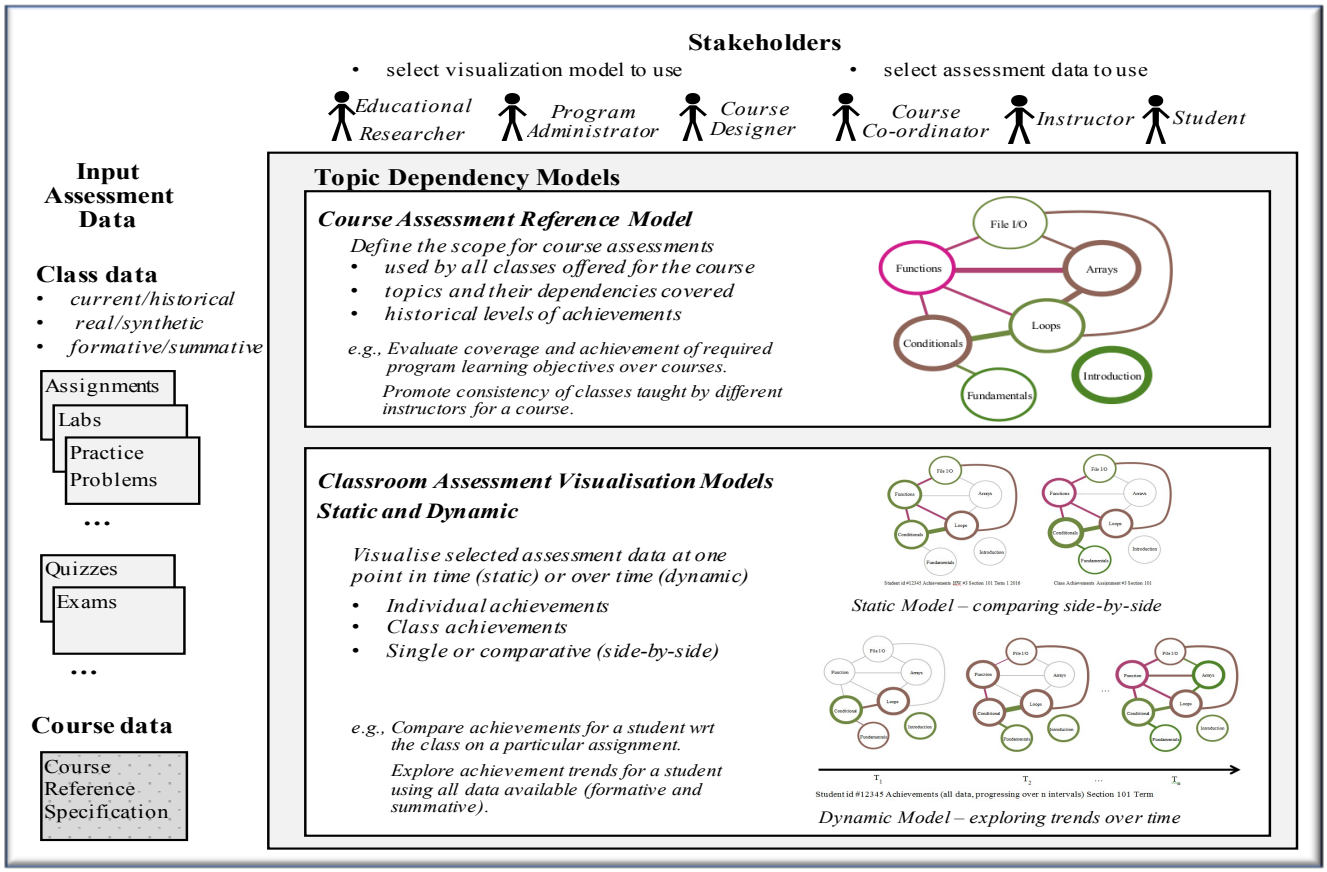


Figure 2. High level view of the Topic Dependency Models

5. Reference Topic Dependency Model

Establishing a standard for a course in terms of the topics covered (either alone or in combination) helps ensure consistency among different offerings of a course and a common, shared understanding of its scope. For example, an instructor can use the reference model to guide their development of assessment material and share the expectations around the topics and their dependencies covered in the course with the students. Students can use the reference model to help gauge their progress in the course, in terms of which topics and their dependencies have or will be covered. Administrators can use the reference model in accreditation efforts to identify any gaps in a program (mapping required program learning objectives to courses).

From an instructor's perspective, the example reference TDM illustrated in Figure 1 has an isolated topic T4, which indicates this topic needs to be assessed on its own. The brown colour of this edge indicates classes have historically performed reasonably well on this topic and the heavier weight of the vertex indicates a substantial number of questions have been answered by students. The connected sub-graph containing T1, T2, and T3 indicates these topics need to be assessed using questions that combine the topics. The bright green colour and heavy weight of T1 indicates excellent performance on a high number of questions has been achieved on this topic; colours close to pink indicate poor performance, for example T3 or the edge between T2 and T3. The pink colour highlights on-going issues in the course, which could be explored by course designers, class coordinators, and instructors.

The reference model for each course can be maintained in a specification file. In the actual implementation of TDMs, the specification is represented in JSON (Bray, 2017), which is a lightweight, formal data-interchange format that is straightforward for stakeholders to read and update. Other formal notations (e.g., XML) are also good candidates and could be adopted if preferred. Algorithm 1 shows at a high-level how a course Reference TDM can be generated using a 'config.JSON' file. Initially, the vertices and edges are extracted from the config file. A Reference TDM is then created and visualized using the obtained vertices and edges. Figure 3 presents an example of a reference TDM and its corresponding specification file using JSON.

ISSN 1929-7750 (online).

Constraints
Question(s) answered with only correct answers, only incorrect answers, or a mix of correct and incorrect answers.
Question(s) tagged with just one topic, two topics, and more than two topics.
Topic(s) assessed in isolation and in combination with other topics.
Topic(s) demonstrating achievements at different levels from very poor to very strong
Topic(s) that have questions over a time-span of multiple days.

Student-Question Responses

sid	qid	score	date	time
s1	q1	1	2018-05-05	13:15:30
s1	q2	1	2018-05-05	14:11:34
s2	q1	1	2018-05-05	18:35:10
s2	q2	1	2018-05-05	18:35:10
s3	q1	1	2018-05-06	09:44:17
s3	q2	0	2018-05-07	11:12:15
s1	q3	0	2018-05-07	16:43:33
s3	q3	0	2018-05-08	11:09:31
s1	q4	1	2018-05-08	11:15:15
s2	q4	0	2018-05-09	17:05:14
s2	q5	0	2018-05-09	19:15:24
s3	q5	1	2018-05-10	08:45:34

Question-Topic Tags

qid	tid
s1	q1
q1	T1
q2	T1
q2	T2
q3	T3
q4	T1
q4	T3
q5	T1
q5	T2
q5	T3

Table 2. Constraints and sample data for the illustrative example

Algorithm 1 Generating a Reference TDM

Require: *config.JSON*

Extract the TDM Graph Elements: Vertices and Edges

1: $VList \leftarrow ExtractVertices(Config.JSON)$
2: $EList \leftarrow ExtractEdges(Config.JSON)$

Create and visualise the TDM Graph

3: $TDMReference \leftarrow CreateTDM(VList, EList)$
4: $Visualise(TDMReference)$

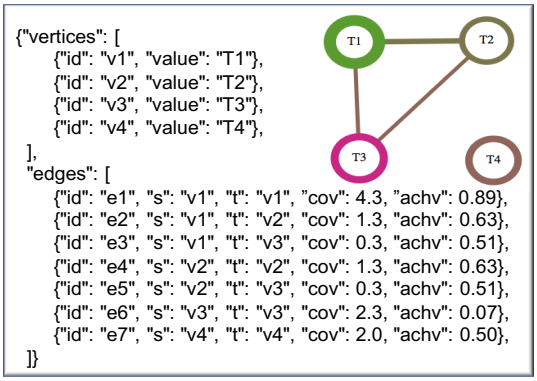


Figure 3. Example Reference TDM (mock-up design)

6. Static Topic Dependency Model

A Static TDM provides a visualization of classroom assessment data at one point in time, spanning the topics and their dependencies required by the course. It can be used to visualize the achievements of one student currently taking a class, an entire classroom of students currently taking a class, or exploring the achievements of a class from a previous year or in another section at the same time. Static TDMs can also be used to explore and compare achievement data by visualizing them side-by-side. For example, an instructor may be interested in comparing the results of their class on a midterm examination to a class offered the previous term. A student may be interested in comparing their individual achievements on a question repository with respect to the class to identify their strengths and areas to improve. Administrators can use Static TDMs to identify topics and their relationships that present on-going challenges to instructors and students.

The remainder of this section demonstrates how commonly available input data (student achievements/grades for specific questions and the mapping from the questions to the course topics) are transformed into a TDM (static achievement model for the cohort). High-level code and notation are presented in Algorithm 2. The algorithm consists of three high level steps: create working dictionaries and matrices; define the TDM graph elements (vertices and edges); and visualize (i.e., plot) the TDM graph. The implementation of this algorithm is in the file topicDependency.py in (GitHub, 2017).

In the given algorithm, *QDict*, *SDict*, *TDict* are data structures of type dictionary for mapping array and matrix indices to question, student, and topic identifiers. The information on topics assigned to each question is represented in a matrix T, in which $t_{ij} = 0$ indicates that question i is not tagged with topic j and $t_{ij} = 1$ indicates that question i is tagged with topic j. The correctness of the answers provided by the users are represented in matrix A, where $a_{ui} = 1$ indicates that user u has answered question i correctly, $a_{ui} = 0$ indicates that user u has not answered question i correctly and $a_{ui} = NULL$ indicates that user u has not attempted question i. Matrix D is used to keep track of attempted questions, where $d_{ui} = 1$ if user u has attempted question i and zero otherwise. *VList* stores the list of the vertices of the TDM graph and *EList* stores the list of the edges of the TDM graph. The coverage and competency associated with an edge are both computed using T, A and D within the *ComputeE* function. The coverage (*Cov*) associated with an edge between two vertices v_j and $v_{j'}$ is computed using the

ISSN 1929-7750 (online).

Algorithm 2 Generating a Static Achievement Model for a Class

Require: $SQA.CSV$, $QT.CSV$

Create dictionaries and matrices for efficient indexing

$QDict \leftarrow CreateQDict(SQA.CSV)$
$SDict \leftarrow CreateSDict(SQA.CSV)$
$TDict \leftarrow CreateTDict(QT.CSV)$
$T \leftarrow CreateT(QT.CSV, QDict, TDict)$
$A \leftarrow CreateA(SQA.CSV, SDict, QDict)$
$D \leftarrow CreateD(SQA.CSV, SDict, QDict)$

Compute the Graph Elements: Vertices and Edges

$VList \leftarrow ComputeV(TDict)$
$EList \leftarrow ComputeE(T, A, D, TDict)$

Create and visualize the TDM Graph

$TDMStatic \leftarrow CreateTDM(VList, EList)$
$Visualize(TDMStatic)$

following formula:

$$Cov(v_j, v_{j'}) = \sum_{i \in QDict} \frac{t_{ij} \times t_{ij'}}{C_1^{g_i} + C_2^{g_i}} \sum_{u \in SDict} d_{ui} \tag{1}$$

where g_i represents the number of topics associated with question i, and C_k^n represents the number of k-combinations from a given set of n elements. The outer summation loops through all of the questions. A question i will contribute to $cov(v_j, v'_j)$ only if it is tagged with both v_j and v'_j (i.e., $t_{ij} \times t_{ij'} = 1$). The extent of the contribution is determined by the multiplication of the following two factors: 1) the number of learners who have attempted the question, which is computed by $\sum_{u \in SDict} d_{ui}$ and 2) the summation of topic ($C_1^{g_i}$) and topic pair combinations ($C_2^{g_i}$) associated with the question. For example, a question tagged with just one topic (T_1) attempted by nine learners will contribute $\frac{9}{1+0} = 9$ to the coverage of $Cov(T_1, T_1)$, and a question tagged with two topics (T_1 and T_2) attempted by nine learners will contribute $\frac{9}{2+1} = 3$ to the coverage of each of $Cov(T_1, T_1)$, $c_1(T_1, T_2)$ and $c_1(T_2, T_2)$.

The achievement ($Achv$) associated with an edge between two vertices v_j and $v_{j'}$ is computed using the following formula:

$$Achv(v_j, v'_j) = \frac{\sum_{i \in QDict} \frac{t_{ij} \times t_{ij'}}{C_1^g + C_2^g} \sum_{u \in SDict} a_{ui}}{Cov(v_j, v_{j'})} \tag{2}$$

A question i will contribute to $Achv(v_j, v'_j)$ only if it is tagged with both v_j and v'_j (i.e., $t_{ij} \times t_{ij'} = 1$). The extent of the contribution is determined by the multiplication of the following two factors: 1) the number of learners who have correctly answered the question, which is computed by $\sum_{u \in SDict} a_{ui}$ and 2) the summation of topic and topic pair combinations associated with the question, which is computed by $C_1^{g_i} + C_2^{g_i}$. The numerator of this formula computes the weighted total of questions answered correctly on topics v_j and v'_j. Dividing this number by $Cov(v_j, v'_j)$ produces the probability of correctly answering questions on topics v_j and v'_j. Note that Cov and $Achv$ of self-loops can also be computed via the same two formulas by using $v_j = v'_j$.

Figure 4 illustrates the main steps taken for creating a Static TDM for the dataset provided in Table 2 using Algorithm 2. The left-hand side of the figure represents the data which is stored in dictionaries $QDict$, $SDict$ and $TDict$ and matrices T, A and D. The right-hand side of the figure illustrates the resulting Static TDM. It has four vertices as the questions in this dataset are tagged with four topics: T_1, T_2, T_3 and T_4. The coverage of the edges are computed using Formula 1. For example, the edge with id $e1$ has coverage of 4.3, which consists of contributions of $\frac{1\times3}{1+0} = 3$ from question 1, $\frac{1\times2}{2+1} = 0.667$ from question 2, $\frac{0\times2}{0+1} = 0$ from question 3, $\frac{1\times2}{3+3} = 0.33$ from question 4 and $\frac{0\times2}{1+0} = 0$ from question 5. The achievement of the edges are computed using Formula 2. For example the achievement of the edge with id $e1$ is computed as $\frac{\frac{1\times3}{1+0}+\frac{1\times2}{2+1}+\frac{0\times2}{0+1}+\frac{1\times1}{3+3}+\frac{0\times1}{1+0}}{4.3} = 0.89$

7. Dynamic Topic Dependency Model

A Dynamic TDM provides a visualization of classroom assessment data as a progression over time, spanning the topics and their dependencies required by the course. It can be used to visualize the progressive achievements of one student currently

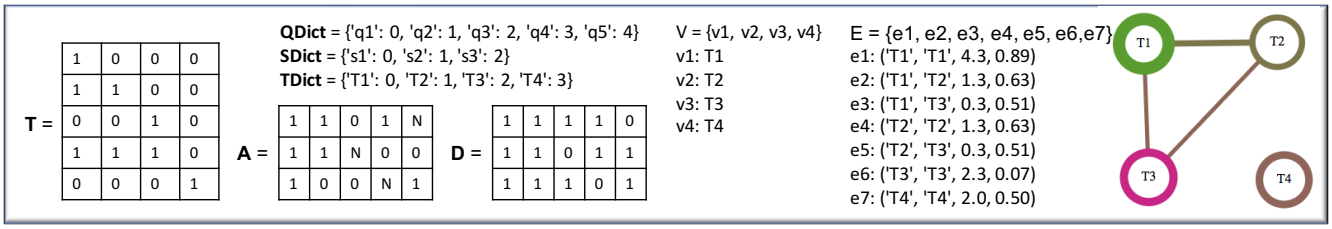


Figure 4. Creating a Static TDM for dataset provided in Table 2 using Algorithm 2.

taking a class, an entire classroom of students currently taking a class, or exploring the achievements of a class from a previous year or in another section at the same time. Dynamic TDMs can also be used to explore and compare achievement data by visualizing them side-by-side. For example, an instructor may be interested in comparing the results of their class on midterm examinations (e.g., midterm 1, midterm 2) to a class offered the previous term. A student may be interested in comparing their individual achievements on a question repository with respect to the class to identify the progression of their strengths and areas to improve. Administrators can use Dynamic TDMs to identify topics and their relationships that present on-going challenges to instructors and students as a class proceeds.

The algorithm provided in this section transforms commonly available input data (student achievements/grades for specific questions and the mapping from the questions to the course topics) into a Dynamic TDM. This algorithm has many similarities to Algorithm 2. It consists of three high level steps: create working dictionaries and matrices; define the TDM graph elements (vertices and edges); and visualize (i.e., plot) the TDM graph. The main difference is that here the algorithm takes in an addition parameter, k, and produces an ordered set of k TDMs that demonstrate the learning progression in k steps. In this algorithm, data from learners' responses to questions are split into k datasets with an equal number of rows. For generating the ith graph of the Dynamic Model, data from splits 1 to i are used in matrix A[i]. High-level code and notation are presented in Algorithm 3.

Algorithm 3 Generating the ith Graph of a Dynamic Achievement Model for a Class

Require: $SQA.CSV$, $QT.CSV$, K

Create dictionaries and matrices for efficient indexing

1: $QDict \leftarrow CreateQDict(SQA.CSV)$
2: $SDict \leftarrow CreateSDict(SQA.CSV)$
3: $TDict \leftarrow CreateTDict(QT.CSV)$
4: $T \leftarrow CreateT(QT.CSV, QDict, TDict)$
5: $A[k] \leftarrow CreateAs(SQA.CSV, SDict, QDict, k)$
6: $D[k] \leftarrow CreateIAs(SQA.CSV, SDict, QDict, k)$

Compute the Graph Elements: Vertices and Edges

7: $VList \leftarrow ComputeV(TDict)$
8: $EList[k] \leftarrow ComputeEs(T, A[k], TDict)$

Create and visualize the TDM Graphs

9: $TDMDynamic[k] \leftarrow CreateTDMs(VList, EList[k])$
10: $Visualize(TDMDynamic[k])$

Figure 5 illustrates the main steps taken for creating a Dynamic TDM with $k = 3$ for the dataset provided in Table 2 using Algorithm 3. Dictionaries $QDict$, $SDict$ and $TDict$, list V and matrix T have been omitted from this figure as they would contain the exactly the same data as shown in Figure 4. For each of the three TDMs, coverage and achievement values are computed via Formula 1 and Formula 2 associated with values from A, D and T. Only the first 4 rows of Table Responses are used from generating the first TDM. The first eight rows of Table References are used for generating the second TDM. Finally, all 12 rows of Table References are used for generating the third TDM.

8. Case Studies: Exploring the Application of TDMs on Historical Data

This section presents two case studies that demonstrate the applications of TDMs. Section 8.1 outlines the experimental environment setup used for presenting the case studies. Section 8.2 applies TDMs for analyzing a mid-sized, third-year undergraduate level course on relational databases. Finally, Section 8.3 applies TDMs for analyzing a large-sized, first-year undergraduate level course on programming and engineering design.

ISSN 1929-7750 (online).

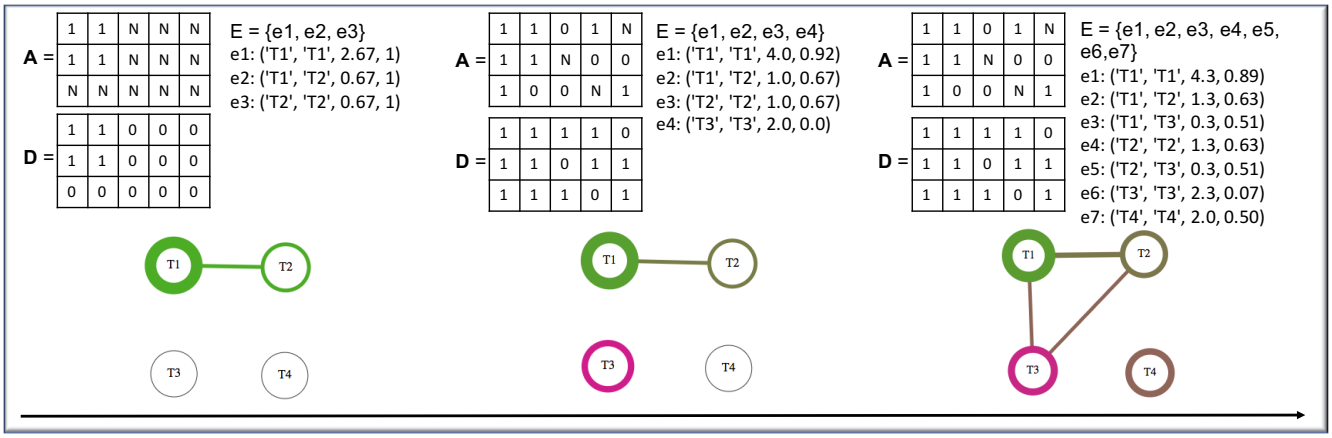


Figure 5. Creating a Dynamic TDM for dataset provided in Table 2 using Algorithm 3.

8.1 Experimental Environment Setup

8.1.1 Implementation

An open-source tool that visualizes TDMs has been developed (GitHub, 2017). The current version of the tool provides capabilities to visualize static achievement data from existing data or synthetic data created on demand with the tool. It is a web-based application, developed using javascript and python; well-established libraries such as Matplotlib (Hunter, 2007) and Data-Driven Documents (D3) (Bostock, Ogievetsky, & Heer, 2011) are utilized to help ensure high quality.

Synthetic data The tool enables users to create synthetic datasets for inspecting the tool and the provided algorithms. To create a synthetic dataset, the user can set the following characteristics around the student population: the number of students, their diversity in backgrounds regarding the course material; and the overall level of competency among the students, reflecting how well-prepared the class is. With respect to the course material, the number and difficulty of the questions can be set in addition to the number of topics spanned by the questions. The tool engine uses Dirichlet, discrete, and normal distributions for generating the datasets. The full implementation is available within dataGenerator.py in (GitHub, 2017). The user interface is straightforward for this feature. Figure 6 presents a screenshot of the interface for generating synthetic datasets and visualizing the results.

Real data The tool also provides the ability for users to visualize their own datasets. The tool takes in two CSV files as input. One of the CSVs has two fields, capturing the question-topic relationships. The other CSV has file three fields, capturing the student-question-grade relationships. Sample input files are available in (GitHub, 2017). Figure 7 presents a screenshot of the interface for loading historical datasets and visualizing the results.

8.1.2 Input Data

Assessments In each of the case studies TDMs visualizing one summative assessment and one formative assessment are presented. For the summative assessments, data from the final exams captured via the Gradescope platform (Singh, Karayev, Gutowski, & Abbeel, 2017), a system for the online assessment of handwritten exams, are used. For the formative assessments, data captured via the PeerWise platform (Denny, Hamer, Luxton-Reilly, & Purchase, 2008) are used. PeerWise is a free web-based platform in which students can answer, rate, and discuss multiple-choice questions created by their peers. To encourage participation, students received grades for their use of the PeerWise environment: 1) They were required to author at least three questions and to correctly answer at least 45 questions (worth 1.5% of final mark) and 2) a grade was calculated from the "Answer Score" (AS) and "Reputation Score" (RS), which were computed by the PeerWise system, using the following formula: $\frac{min(AS,RS)\times 1.5}{500}$, (worth 1.5% of final mark).

Datasets For each of the case studies the course code, name, date of offering, number of students (S), and number of high-level modules (M) covered in the offering are presented. The high-level modules are extracted from the formal course descriptions; they are similar across the different offerings of a course. For the summative assessments, the number of independent sub-questions (Q_S), number of concepts used for tagging the questions (C_S), and the total number of tags assigned to the questions (A_S) are presented. The concepts are offering- and assessment-dependent; they are created and assigned to the questions by the instructor of the course. For the formative assessments the number of independent sub-questions (Q_F) the number of concepts used for tagging the questions (C_F) and the total number of tags assigned to the questions (A_F) are presented. The concepts are offering- and assessment-dependent again; but in the formative assessment they are created and assigned to the questions by the students enrolled in the offering. An additional parameter, An_F, is used to present the total

ISSN 1929-7750 (online).

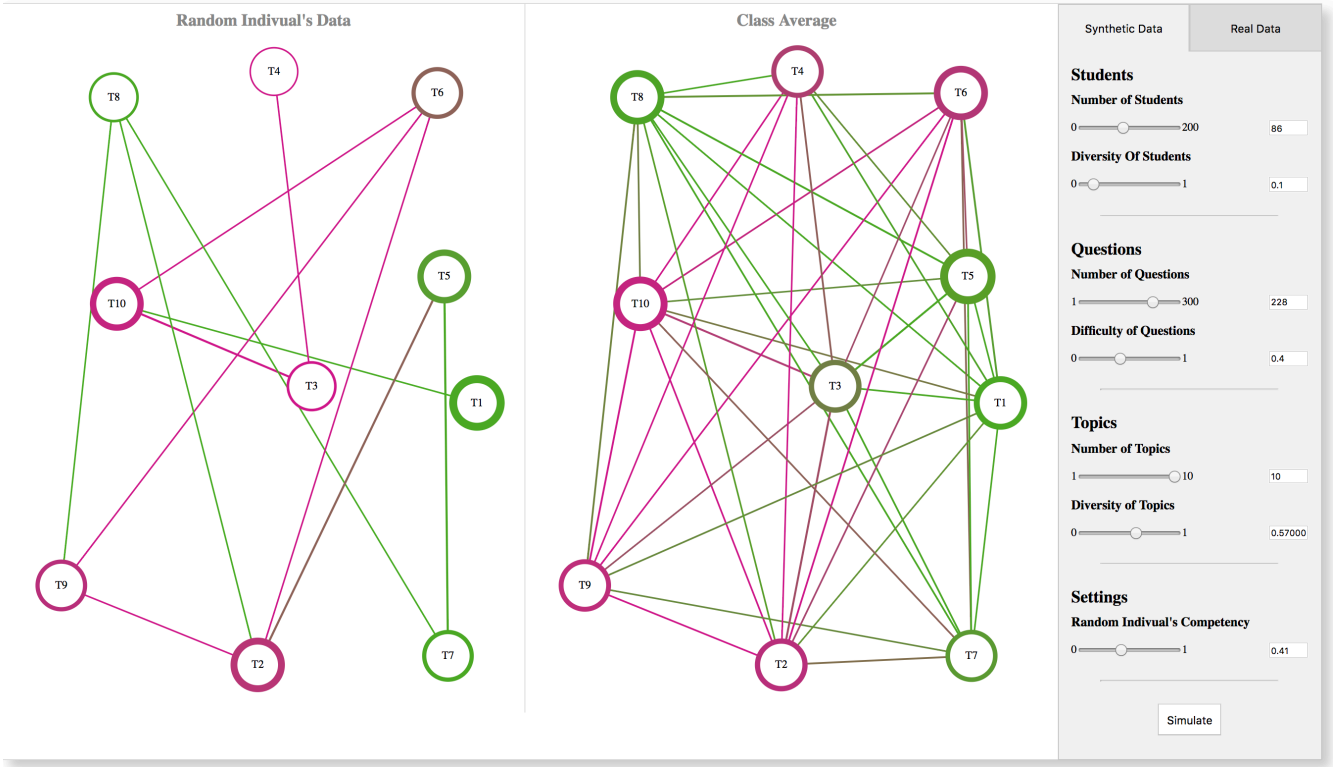


Figure 6. Screenshot of the TDM Visualization Tool: Application to Synthetic Data.

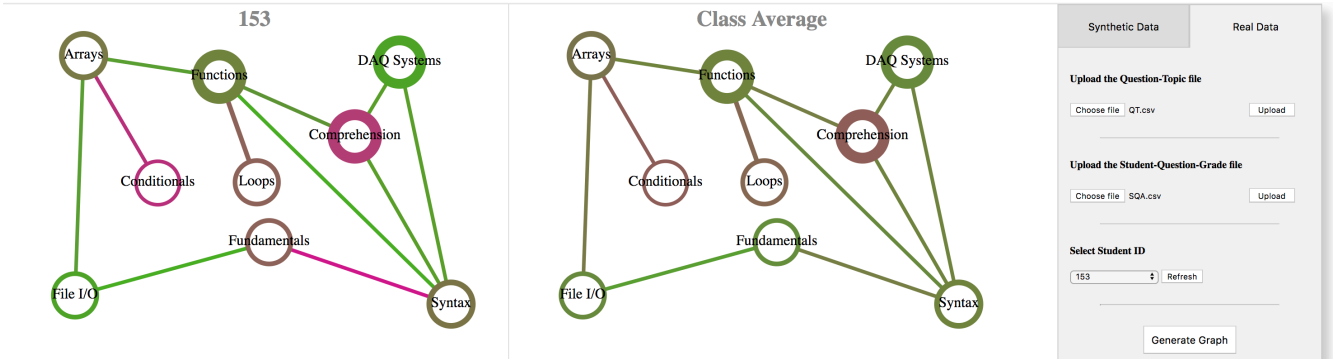


Figure 7. Screenshot of the TDM Visualization Tool: Application to Historical Data.

number of answers provided for the questions in the formative assessment. The number of questions answered by each student can vary significantly as they self-select their engagement level with the PeerWise platform. Note that this value is easier to approximate in summative assessments as all of the students are expected to answer all of the questions on the assessment. Table 3 provides an overview of the datasets used in the presented case studies.

The significant difference in the class sizes of the two cases enable us to investigate the effectiveness of using TDMs in both mid- and large-sized classes. In addition the availability of two sets of tags, at the module-level and the concept level, provides the ability to analyze the results at two levels of granularity. Question-tags at the module-level can be achieved by simply re-labelling concept tags to the module that best represents them. This re-labelling can lead to redundant tags, i.e., the same question being tagged with the same module more than once. Redundant tags may be removed after the relabelling is complete.

8.2 Case Study 1: A Mid-Sized Third-Year Undergraduate Course in Relational Databases

This case study is based on data collected from a third-year undergraduate level offering of a course on relational databases at The University of British Columbia. This offering of the course had 103 students and was held during the Summer of 2016. The course covers many topics that are generally included in an introductory course on relational databases including conceptual database design using ER diagrams, relational models, functional dependencies, normalization, relational algebra, Structured Query Language (SQL), Datalog and deductive databases, data warehousing and semi-structured data, and XML. The SQL

ISSN 1929-7750 (online).

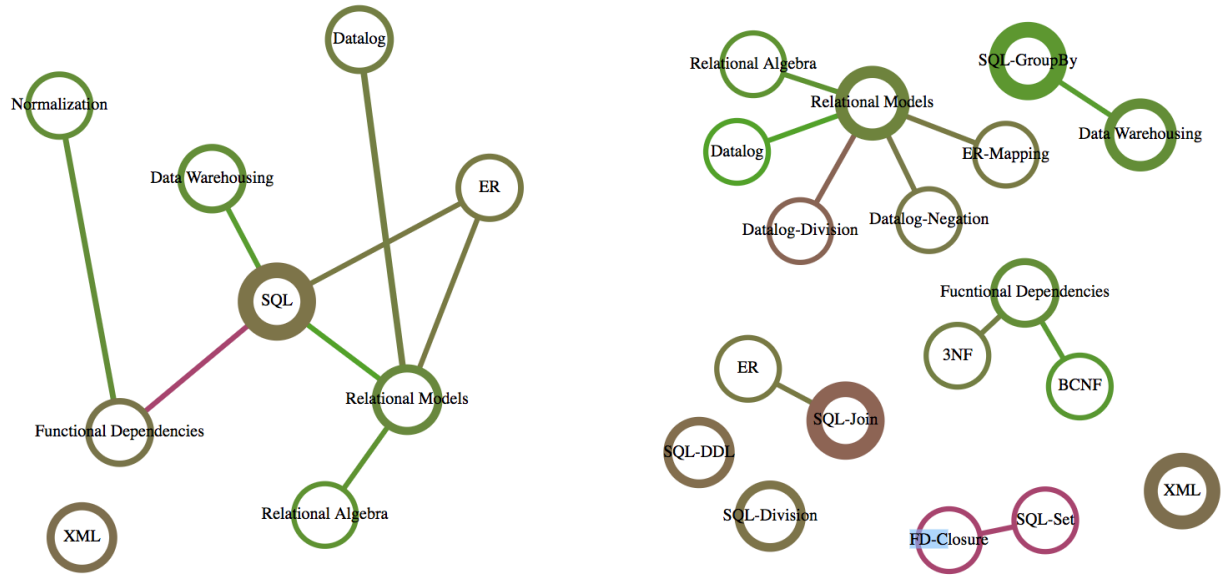
Code	Name	Offering	S	M	Summative			Formative			
					Q_S	C_S	A_S	Q_F	C_F	A_F	An_F
CPSC304	Introduction to Relational Databases	Summer 2016	103	9	23	18	28	387	27	630	8537
APSC160	Introduction to Computation in Engineering Design	Fall 2016	377	9	17	21	37	1111	30	2128	21434

Table 3. datasets used in the presented case studies

Module received roughly three weeks of lecture time; all of the other modules received roughly one week of lecture time.

Summative Assessment The final exam of this offering had 23 independent sub-questions which formed a total of eight main questions. These sub-questions were assigned a total of 28 tags using 17 instructor-defined concepts. Figure 8 shows the results of applying TDMs to this dataset using module-level and concept level tags, visualizing the results at two levels of granularity. Figure 8a shows the module-level TDM for this exam. This TDM shows that the exam adequately covered all of the modules of the course. The SQL Module has received more assessment items than other modules. This is expected as SQL was the biggest module of the course and it plays a central role in connecting many concepts of the course. It also shows that on average students have done well in this exam with the exception of questions that were co-tagged with functional dependencies and SQL. Interestingly, all of the modules other than XML were assessed jointly in combination with at least one other module.

Figure 8a shows the concept-level TDM for this exam, allowing for a finer level of granularity. For example, the SQL Module is further decomposed into five concepts: SQL-DDL, SQL-Set, SQL-GroupBy, SQL-Join, and SQL-Division. This further decomposition shows that SQL-GroupBy and SQL-Join have received more assessments items than the other SQL sub-modules. It also shows that students did well in questions on SQL-GroupBy, reasonably well in questions on SQL-DDL, SQL-Join and SQL-Division but quite poorly on questions on SQL-Set. As another example, further decomposition of items from the Datalog Module shows that students were able to do well in simple Datalog questions, but were finding questions on Datalog-Division and Datalog-Negation harder to answer correctly.



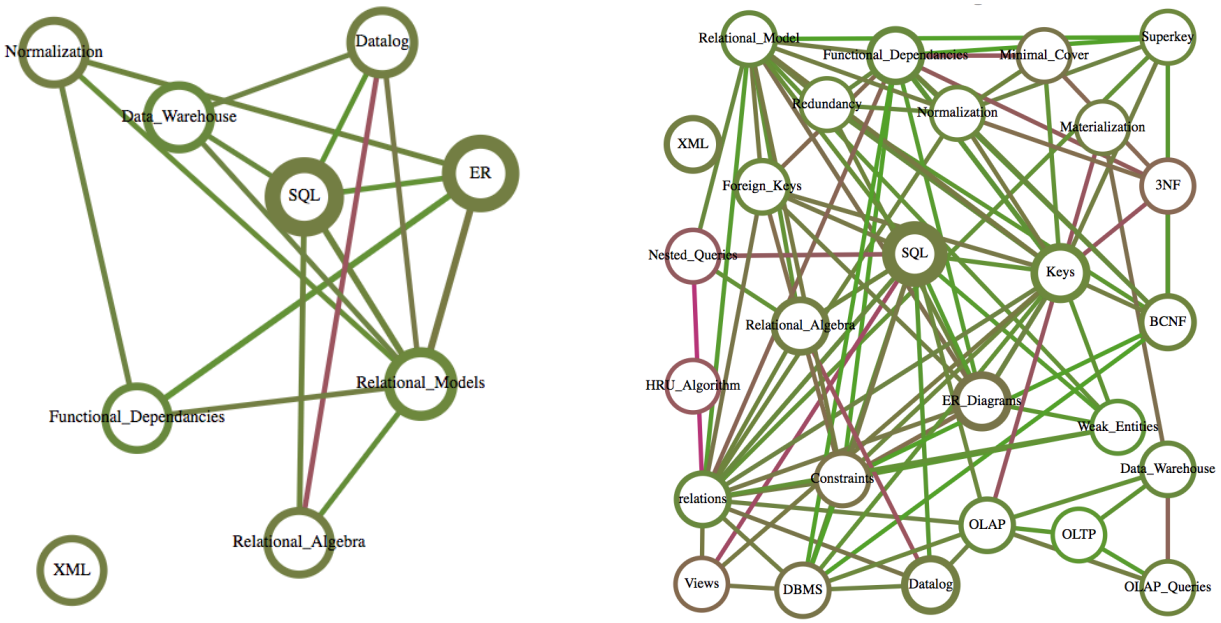
(a) TDM at the module level **(b)** TDM at the concept level

Figure 8. TDMs for the final exam of an offering of a third-year course on relational databases.

Formative Assessment Using the PeerWise platform, students created 27 concepts to author 387 questions, assigned 630 tags to questions and answered 8537 questions. Figure 9 shows the results of applying TDMs to this dataset using module-level and concept-level tags, visualizing the results at two levels of granularity. Figure 9a shows the module-level TDM for this summative assessment. This TDM shows that the questions on the PeerWise platform adequately covered all of the modules of the course. Again, the SQL Module has received more assessment items than other modules. It also shows that on average students have done well on the questions with the exception of questions that were co-tagged with Datalog and Relational Algebra. The XML Module has not been assessed in combination with any of the other modules.

ISSN 1929-7750 (online).

Figure 9b presents the concept-level TDM for this dataset. A tabular version of the dataset used for this visualization is presented in Appendix 1. As shown in this figure, the larger set of the questions and concepts, developed in partnership with students, provides the ability for the content from many modules to be assessed in combination with concepts within the same module as well as concepts from other modules. For example, questions tagged with SQL were co-tagged with other concepts such as Views and Nested Queries that are covered within the same module. They are also co-tagged with other concepts such as Relational Algebra, Foreign Keys, and Normalization, which are covered in other modules. It is worth noting that with the exception of Nested Queries and HRU Algorithms, the performance on most individual-level concepts have been reasonably high; however, questions from some concept-pairings such as Datalog and Relational Algebra, 3NF and Keys, SQL and Views, Relations, and Functional Dependencies seem to be not as well received by the students.



(a) TDM at the module level **(b)** TDM at the concept level

Figure 9. TDMs representing data collected from PeerWise in a third-year course on relational databases.

Take-away Points The following reflections are provided from an instructor's perspective (one of the authors has taught this course); they are organized around a subset of stakeholder needs outlined in Section 4.1. Based on the TDMs developed for this case study, the following observations can be made:

- What topics and their relationships do I need to assess?
  - In both assessments, the relative coverage of the modules seem to be proportional to the lecture time dedicated to the module (e.g., SQL is getting roughly three times as many questions as the other modules).
  - Most of the modules of the course are highly connected to one another. In particular the SQL module plays a central role in connecting many modules of the course. The XML Module, however, seems to be disconnected from the rest of this course.
- How well are the students performing on the topics?
  - The performance of students on questions targeting an individual modules seem to be adequate; however, their performance sometimes drops on questions covering content from across modules. Therefore, more emphasis on connecting concepts from across modules can potentially help students do better in their assessments.
- What topics do I need to improve my material or presentation on?
  - Updating lecture material on SQL-sets, Functional Dependency Closure, Nested SQL Queries, and the HRU Algorithm may be appropriate as these are the concepts that students are performing most poorly on.
- How can I communicate the importance of the topics and their relationships?
  - The visualizations at the module level in this study, containing nine modules, are relatively straightforward to comprehend. It could be improved by reducing the number of crossing edges (e.g., providing features in the tool

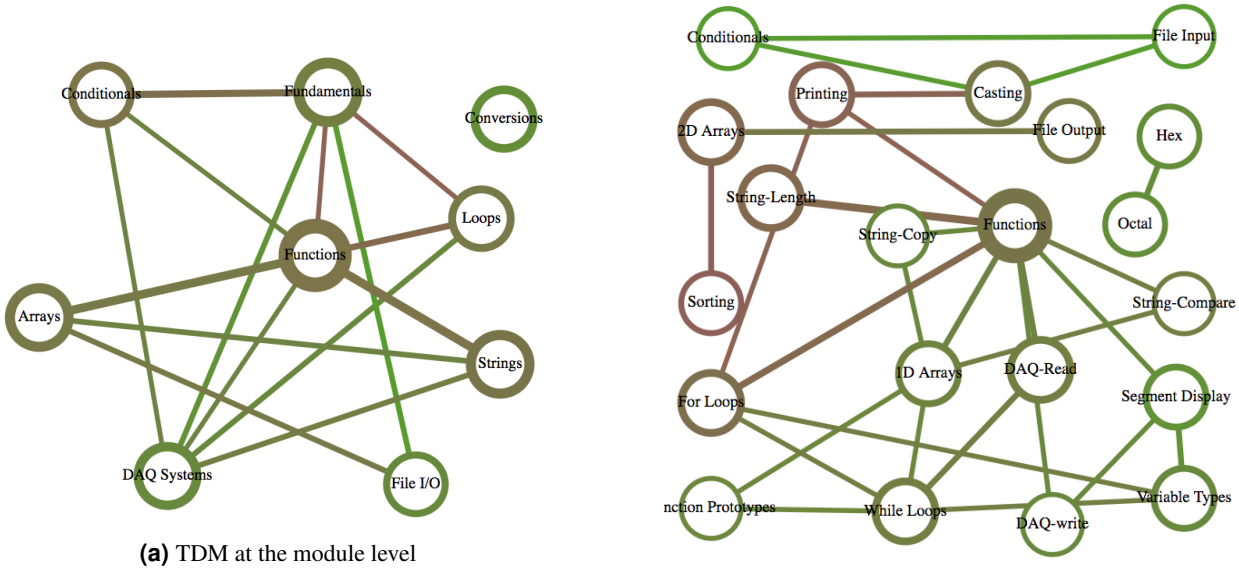
support to manually or automatically reshape an edge from a straight line to a curve and improve the layout). The visualization at the concept level in this study containing 18 concepts is also straightforward; however, the visualization including 27 concepts is difficult to comprehend. These could be improved with supporting variable levels of detail (e.g., providing a feature to abstract/decompose subgraphs of interest).

8.3 Case Study 2: A Large, First-Year Undergraduate Course in Programming and Engineering Design

This case study is based on data collected from a first-year undergraduate level offering of a course on programming and engineering design at The University of British Columbia. This offering had 377 students and was held during the Fall of 2016. The course covers many topics that are generally included in an introductory course on programming and engineering design, including number conversions, programming fundamentals, conditionals, loops, file I/O, functions, arrays, strings, and DAQ systems. Functions, strings, and DAQ systems received two weeks of lecture time; all of the other modules received roughly one week of lecture time.

Summative Assessment The final exam of this offering had 17 independent sub-questions that formed a total of eight main questions. These sub-questions were assigned a total of 37 tags using 21 learner-defined concepts. Figure 10 shows the results of applying TDMs to this dataset using module-level and concept-level tags, visualizing the results at two levels of granularity. Figure 10a shows the module-level TDM for this exam. This TDM shows that the exam adequately covered all of the modules of the course and that all modules except Conversions were highly connected to one another. It also shows that on average students performed better on conversions, fundamentals, file I/O, and DAQ Systems compared to the other modules.

Figure 10b shows the concept-level TDM for this exam, allowing for a finer level of granularity. For example, the Strings Module is further decomposed into three topics: String-Length, String-Copy, and String-Compare. This further decomposition shows that students did well on questions on String-Copy and String-Compare but not so well on questions on String-Length. As another example, further decomposition of items from the Arrays Module shows that students were able to do well in 1D-Arrays, but did quite poorly on questions on 2D-Arrays. The high level of connections among all modules, except the Conversion Module, is observable with the use of concepts as well. The Conversions Module has been decomposed into Hex and Octal, which remain disconnected from the rest of the graph.



(a) TDM at the module level

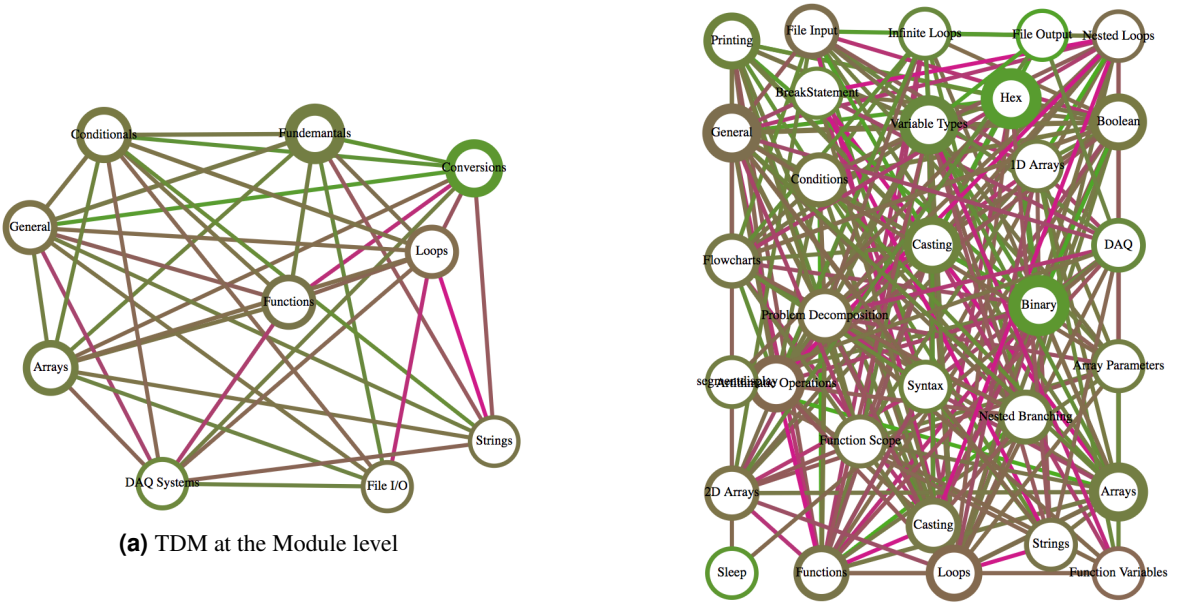
(b) TDM at the sub-module level

Figure 10. TDMs for the final exam of an offering of a first-year course on programming and engineering design.

Formative Assessment Using the PeerWise platform, students created 30 concepts to author 1111 questions, assigned 2128 tags to questions, and answered 21,434 questions. Figure 9 shows the results of applying TDMs to this dataset using module-level and concept-level tags, visualizing the results at two levels of granularity. Figure 11a models this data using the main 9 modules of the course plus an additional node titled "General," which has been used for relabelling concepts such as syntax that do not belong to a particular module. This figure demonstrates that all of the modules of this course are tightly coupled. This is somewhat expected in a programming course as the content of later modules of the course rely on the content that had been covered in the previous modules. Figure 11b presents the concept-level TDM for this dataset. As shown in

ISSN 1929-7750 (online).

this figure, the larger set of the questions and concepts, developed in partnership with students, provides the ability for the content from many modules to be assessed in combination with concepts within the same module as well as concepts from other modules.



(a) TDM at the Module level

(b) TDM at the Concept level

Figure 11. TDMs representing data collected from PeerWise in a first-year course on Programming and Engineering Design.

Take-away Points The following reflections are provided from an instructor's perspective (both authors have taught this course); they are organized around a subset of stakeholder needs outlined in Section 4.1. Based on the TDMs developed for this case study, the following observations can be made:

- What topics and their relationships do I need to assess?
  - The modules covered in this course are tightly coupled; therefore, it is very important to have a mechanism (such as frequent use of assessments) to ensure that students have adequately learned the content of a module before moving on.
- How well are the students performing on the topics?
  - Students have performed significantly better in answering questions from multiple concepts in the summative assessment compared to similar questions in the formative assessment. This suggests that first-year students have the ability to cognitively connect different concepts, but they may need a rich collection of questions covering one or more concepts to attempt and learn from their mistakes.
- What topics do I need to improve my material or presentation on?
  - Introduce additional material to help students with assessment items that require knowledge of multiple topics.
- How can I communicate the importance of the topics and their relationships?
  - The visualizations at the module level in this study, containing nine modules, are relatively straightforward to comprehend. These could be improved by reducing the number of crossing edges (e.g., providing features in the tool support to manually or automatically reshape an edge from a straight line to a curve and improve the layout). The visualization at the concept level in this study, containing 21 concepts, is difficult to comprehend; the visualization with 30 concepts is practically unreadable. They could be improved with supporting variable levels of detail (e.g., providing a feature to abstract/decompose subgraphs of interest).

9. Conclusions and Future Work

The rapidly changing landscape of educational environments presents new challenges on the design and delivery of high-quality assessments at scale. In particular ensuring that there is consistency among assessments of different offerings of a course

ISSN 1929-7750 (online).

(within and among terms), providing meaningful feedback about student achievement, and tracking student progress over time are all challenging tasks. In this work, a collection of Topic Dependency Models (TDMs) is introduced to address three facets of the problem. The first is the need for course level models to communicate the required topics and their relationships. The second is the need for classroom level models to visualize assessment data within a class and support comparisons of assessment data between classes (static). The third is the need for classroom level models that visualize assessment data trends over time. Stakeholders can select the TDM and the assessment data of interest to use, which may be available from a wide variety of sources. The algorithms to create exemplar TDMs in addition to two cases studies using historical data from computer science courses are also reported.

There are several limitations in the current work that restrict the generalizability of the results. One of the significant limitations of this study is that the TDMs have only been applied to courses from the computer science domain. It is important to acknowledge that TDMs may be hard to comprehend for stakeholders without formal training in algorithmic literacy. Future work aims to integrate TDMs into an educational learning dashboard called RiPPLE (Khosravi, 2017) to support the investigation of the usability of TDMs across disciplines with and without formal training in algorithmic literacy. This, in turn, may lead to the scaffolding of the model or development of training material (e.g., short videos) for stakeholders so that they can use TDM representations more effectively. A second limitation of the current work is that the case studies only explore the benefits of using TDMs from an instructor's perspective. A comprehensive stakeholder and scenario analysis study is planned for future work that will provide a framework for the usability study. A third limitation is the current graph foundation (the two-weighted graph) does not support a hierarchical decomposition for the TDMs. As an initial step, alternative graph definitions will be investigated (e.g., two-weighted hierarchical graphs) to support abstraction/decomposition at multiple levels. This will enable users to drive their own inquiry to allow them to zoom into a particular topic (across questions) or examine if certain topics drive particular assessment question types.

10. Conflict of Interest

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

This study was not financially supported by external organizations.

References

Albert, D., Nussbaumer, A., & Steiner, C. (2010). Towards generic visualisation tools and techniques for adaptive e-learning. In *Proc. 18th int. conf. computers in education* (pp. 61–65).

Beheshitha, S. S., Hatala, M., Gašević, D., & Joksimović, S. (2016). The role of achievement goal orientations when studying effect of learning analytics visualizations. In *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 54–63). New York, NY, USA: ACM. https://dx.doi.org/10.1145/2883851.2883904

Bodily, R., Kay, J., Aleven, V., Jivet, I., Davis, D., Xhakaj, F., & Verbert, K. (2018). Open learner models and learning analytics dashboards: A systematic review. In *Proceedings of the 8th international conference on learning analytics and knowledge* (pp. 41–50). New York, NY, USA: ACM. https://dx.doi.org/10.1145/3170358.3170409

Bostock, M., Ogievetsky, V., & Heer, J. (2011, December). D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309. https://dx.doi.org/10.1109/TVCG.2011.185

Bray, T. (2017). *The javascript object notation (json) data interchange format* (Tech. Rep.). https://dx.doi.org/10.17487/RFC8259

Bull, S., Ginon, B., Boscolo, C., & Johnson, M. (2016). Introduction of learning visualisations and metacognitive support in a persuadable open learner model. In *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 30–39). New York, NY, USA: ACM. https://dx.doi.org/10.1145/2883851.2883853

Bull, S., & Kay, J. (2010). Open learner models. In R. Nkambou, J. Bourdeau, & R. Mizoguchi (Eds.), *Advances in intelligent tutoring systems* (pp. 301–322). Berlin, Heidelberg: Springer Berlin Heidelberg. https://dx.doi.org/10.1007/978-3-642-14363-2$_1$5

Campbell, D., & Stanley, J. (1966). Experimental and quasi-experimental designs for research.

Carless, D. (2006). Differing perceptions in the feedback process. *Studies in Higher Education*, *31*(2), 219-233. https://dx.doi.org/10.1080/03075070600572132

Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis-a general method for cognitive model evaluation and improvement. In *Intelligent tutoring systems* (Vol. 4053, pp. 164–175). https://dx.doi.org/https://doi.org/10.1007/11774303$_1$7

Cooper, K., & Khosravi, H. (2018). Graph-based visual topic dependency models: Supporting assessment design and delivery at scale. In *Proceedings of the 8th international conference on learning analytics and knowledge* (pp. 11–15). New York, NY, USA: ACM. https://dx.doi.org/10.1145/3170358.3170418

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

Dawson, S., Macfadyen, L., Risko, E. F., Foulsham, T., & Kingstone, A. (2012). Using technology to encourage self-directed learning: The collaborative lecture annotation system.

Denny, P., Hamer, J., Luxton-Reilly, A., & Purchase, H. (2008). Peerwise: Students sharing their multiple choice questions. In *Proceedings of the fourth international workshop on computing education research* (pp. 51–58). New York, NY, USA: ACM. https://dx.doi.org/10.1145/1404520.1404526

D'Mello, S., Lehman, B., Sullins, J., Daigle, R., Combs, R., Vogt, K., ... Graesser, A. (2010). A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In *Proceedings of the 10th international conference on intelligent tutoring systems - volume part i* (pp. 245–254). Berlin, Heidelberg: Springer-Verlag. https://dx.doi.org/10.1007/978-3-642-13388-6$_2$9

Ellis, C. (n.d.). Broadening the scope and increasing the usefulness of learning analytics: The case for assessment analytics. *British Journal of Educational Technology*, *44*(4), 662-664. https://dx.doi.org/10.1111/bjet.12028

Englund, C., Olofsson, A. D., & Price, L. (2016, 04). Teaching with technology in higher education: understanding conceptual change and development in practice. *Higher Education Research Development*, 1-15. https://dx.doi.org/10.1080/07294360.2016.1171300

GitHub. (2017). *Topic dependency models tool repository.* Retrieved from https://github.com/hkhosrav/Topic-Dependency-Models

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95. https://dx.doi.org/10.1109/MCSE.2007.55

Khan, I., & Pardo, A. (2016). Data2u: Scalable real time student feedback in active learning environments. In *Proceedings of the sixth international conference on learning analytics & knowledge* (pp. 249–253). https://dx.doi.org/10.1145/2883851.2883911

Khosravi, H. (2017). Recommendation in personalised peer-learning environments. *arXiv preprint arXiv:1712.03077*.

Khosravi, H., Cooper, K., & Kitto, K. (2017). Riple: Recommendation in peer-learning environments based on knowledge gaps and interests. *JEDM-Journal of Educational Data Mining*, *9*(1), 42-67. Retrieved from https://jedm.educationaldatamining.org/index.php/JEDM/article/view/239

Khosravi, H., & Cooper, K. M. (2017). Using learning analytics to investigate patterns of performance and engagement in large classes. In *Proceedings of the 2017 acm sigcse technical symposium on computer science education* (pp. 309–314). https://dx.doi.org/10.1145/3017680.3017711

Kirk, A. (2012). *Data visualization: a successful design process*. Packt Publishing Ltd.

May, M., George, S., & Prevot, P. (2011). Travis to enhance online tutoring and learning activities: Real–time visualization of students tracking data. *Interactive Technology and Smart Education*, *8*(1), 52-69. https://dx.doi.org/10.1108/17415651111125513

Oxman, S., Wong, W., & Innovations, D. (2014). White paper: Adaptive learning systems. *Integrated Education Solutions*.

Pardos, Z., & Heffernan, N. (2011). Kt-idem: introducing item difficulty to the knowledge tracing model. *User Modeling, Adaption and Personalization*, 243–254. https://dx.doi.org/https://doi.org/10.1007/978-3-642-22362-4$_2$1

Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 conference on artificial intelligence in education: Building learning systems that care: From knowledge representation to affective modelling* (pp. 531–538). Amsterdam, The Netherlands, The Netherlands: IOS Press. Retrieved from http://dl.acm.org/citation.cfm?id=1659450.1659529

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in neural information processing systems* (pp. 505–513). Curran Associates, Inc.

Riordan T., L. G. (2009). Collaborative and systemic assessment of student learning: From principles to practice. In J. G. (Ed.), *Assessment, learning and judgement in higher education.* Springer. https://dx.doi.org/10.1007/978-1-4020-8905-3$_2$

Schwendimann, B. A., Rodriguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., ... Dillenbourg, P. (2017). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies*, *10*(1), 30–41. https://dx.doi.org/10.1109/TLT.2016.2599522

Sha, L., & Hong, P. (2017). Neural knowledge tracing. In *International conference on brain function assessment in learning* (pp. 108–117). https://dx.doi.org/https://doi.org/10.1007/978-3-319-67615-9$_1$0

Singh, A., Karayev, S., Gutowski, K., & Abbeel, P. (2017). Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In *Proceedings of the fourth (2017) acm conference on learning@ scale* (pp. 81–88). https://dx.doi.org/10.1145/3051457.3051466

Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. (2013). Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education* (pp. 171–180).

11. Appendix

The following two tables show the dataset used for the creation of the TDM presented in Figure 9b.

Table 4. List of the 27 concepts used in Figure 9b (presented three per row)

'BCNF', 'Redundancy', 'Views',
'Relational_Model', ' OLAP_Queries', 'OLAP',
'Keys', 'Weak_Entities', 'ER_Diagrams',
'Functional Dependancies', 'Normalization', 'Superkey',
'Materialization', 'DBMS', 'Relational_Algebra',
'Datalog', 'relations', 'Minimal_Cover','
Constraints', 'XML', 'HRU_Algorithm',
'OLTP', Data_Warehouse', 'Foreign_Keys',
'3NF', 'SQL', ' Nested_Queries'

Table 5. List of 57 edges used in Figure 9b (presented three per row)

['Keys', 'Relational_Model', 83.83, 74.0], ['Keys', 'SQL', 25.67, 81.0], ['Keys', 'Normalization', 13.87, 69.0],
['Normalization', 'Relational_Model', 12.83, 73.0], ['Datalog', 'OLAP', 3.9, 72.0], ['Functional_Dependancies', '3NF', 13.17, 44.0],
['Keys', 'ER_Diagrams', 127.8, 72.0], ['OLAP', 'Materialization', 3.17, 42.0], ['Normalization', 'Functional_Dependancies', 18.53, 71.0],
['Relational_Model', 'Nested_Queries', 12.27, 73.0], ['Normalization', '3NF', 17.93, 68.0], ['Relational_Model', 'Foreign_Keys', 27.33, 63.0],
['Relational_Model', 'Weak_Entities', 16.2, 84.0], ['Normalization', 'BCNF', 6.2, 78.0], ['Relational_Model', 'Relational_Algebra', 17.6, 79.0],
['Relational_Model', 'Functional_Dependancies', 26.93, 71.0], ['SQL', 'OLAP', 5.9, 76.0], ['SQL', 'Relational_Model', 130.53, 76.0],
['SQL', 'Datalog', 7.57, 85.0], ['Materialization', 'Data_Warehouse', 8.83, 60.0], ['OLAP', 'Data_Warehouse', 38.83, 79.0],
['SQL', 'ER_Diagrams', 2.2, 91.0], ['Datalog', 'Relational_Model', 4.73, 66.0], ['Functional_Dependancies', 'Foreign_Keys', 3.6, 56.0],
['SQL', 'Foreign_Keys', 35.83, 66.0], ['SQL', 'Nested_Queries', 38.93, 44.0], ['Normalization', 'Normalization', 49.37, 73.0],
['Keys', 'Keys', 711.27, 75.0], ['Datalog', 'Datalog', 527.4, 71.0], ['SQL', 'SQL', 1714.87, 69.0], ['OLAP', 'OLAP', 307.73, 76.0],
['Materialization', 'Materialization', 18.83, 71.0], ['XML', 'XML', 367.0, 69.0], ['ER_Diagrams', 'ER_Diagrams', 634.5, 64.0],
['Data_Warehouse', 'Data_Warehouse', 176.5, 76.0], ['Functional_Dependancies', 'BCNF', 32.37, 84.0], ['Datalog', 'Relational_Algebra', 0.83, 40.0],
['Weak_Entities', 'Weak_Entities', 51.87, 84.0], ['3NF', '3NF', 115.33, 57.0], ['BCNF', 'BCNF', 247.87, 74.0],
['Relational_Algebra', 'Nested_Queries', 10.1, 83.0], ['Keys', 'Weak_Entities', 5.67, 85.0], ['Keys', '3NF', 17.83, 75.0],
['Keys', 'Functional_Dependancies', 147.87, 84.0], ['SQL', 'Weak_Entities', 2.2, 91.0], ['Keys', 'BCNF', 37.2, 75.0],
['Keys', 'Foreign_Keys', 42.0, 66.0], ['Relational_Model', 'ER_Diagrams', 128.17, 60.0], ['Relational_Model', 'Relational_Model', 523.87, 74.0],
['Relational_Model', 'OLAP', 3.9, 72.0], ['ER_Diagrams', 'Functional_Dependancies', 10.57, 78.0], ['3NF', 'BCNF', 48.17, 54.0],
['ER_Diagrams', 'Weak_Entities', 7.87, 87.0], ['Foreign_Keys', 'Foreign_Keys', 175.67, 75.0], ['Nested_Queries', 'Nested_Queries', 38.93, 44.0],
['Relational_Model', 'BCNF', 4.6, 93.0], ['SQL', 'Relational_Algebra', 64.1, 73.0], ['ER_Diagrams', 'Foreign_Keys', 12.73, 71.0],
['Relational_Algebra', 'Relational_Algebra', 438.93, 70.0], ['Functional_Dependancies', 'Functional_Dependancies', 413.7, 76.0]

ISSN 1929-7750 (online).