

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Multilevel Visualisation of Topic Dependency Models for Assessment Design and Delivery: A Hypergraph Based Approach

Kendra M. L. Cooper Ph.D.^{a,*}¹, Hassan Khosravi Ph.D.^b²

^aIndependent Scholar, Canada

^bQueensland University, Australia

ARTICLE INFO

Article History:

Submitted x.x.2019

Revised x.xx.2019

Second Revision x.xx.2019

Accepted x.xx.2019

Keywords:

educational assessment

topic dependency model

hypergraph

visualisation

ABSTRACT

The effective design and delivery of assessments in a wide variety of evolving educational environments remains a challenging problem. Proposals have included the use of learning dashboards, peer learning environments, and grading support systems; these embrace visualisations to summarise and communicate results. In an on-going project, the investigation of graph based visualisation models for assessment design and delivery has yielded promising results. Here, an alternative graph foundation, a two-weighted hypergraph, is considered to represent assessment results and their explicit mapping to one or more learning objective topics. The visualisation approach considers the hypergraph as a collection of levels; the content of these levels can be customized (i.e., filtered) and presented according to user preferences. A case study on generating hypergraph models using commonly available assessment data and a flexible visualisation approach using historical data from an introductory programming course is presented.

© 2019 KSI Research

1. Introduction


Assessment remains a core educational activity, even as environments continue to evolve beyond the traditional classroom. Blended, flipped, and massive open on-line courses are supported by a wide range of assessment tools and techniques [10]. Instructors have many options for assessing the required topics of a course (e.g., CS 100 Introduction to Computer Science) in a particular offering (e.g., CS 100 Term 1 2019 Section 001) [12]. More traditional material includes homework assignments and examinations; emerging material includes question repositories and games. This rich variety also introduces new challenges to educational stakeholders (e.g., students, instructors, administrators, ed-

ucation researchers) with respect to evaluating the coverage of assessment material and communicating achievements. 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 co-ordinators, and researchers also face challenges. Administrators find it challenging to compare the content and difficulty of formal assessments as well as students' outcomes across different offerings of a course. Course designers and co-ordinators find it challenging to ensure the required topics and their relationships (e.g. questions with a combination of topics T_1 , T_2 and T_3) have been assessed. Educational researchers need to compare the achievement results between control and experimental groups.

To enable different educational stakeholders to explore, understand, and communicate insights from educational data sets, a new field known as "Learning Dashboards" has emerged [20, 2, 18]. These dashboards employ data sets on students' engagement and performance that are commonly available in educational settings to provide interactive visualisation widgets and techniques in the educational context. A vari-

In this work, a topic dependency model for educational assessment is introduced. The model, based on a hypergraph foundation, is visualised with a multilevel approach that supports user selected filtering options for achievements and topic coverage.

*Corresponding author

 kendra.m.cooper@gmail.com (K.M.L. Cooper);

h.khosravi@uq.edu.au (H. Khosravi)

ORCID(s): 0000-0003-4771-1371 (K.M.L. Cooper)

¹Independent Scholar, Canada.

²Senior Lecturer, Queensland University, Australia.

ety of traditional plots and charts have been adopted in these dashboards (e.g., pie, box, histogram, radial) to visualise the achievements of students. The research typically considers independent (stand-alone) topics. Additional discussion on the related work in this area is presented in Section 3.

As part of an on-going research project, the authors of this paper have explored a collection of topic dependency models (TDMs) for assessment in which the relationships among topics are considered [14, 8]. The TDMs use a two-weighted undirected graphs foundation to formally represent and visualise 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 consists of a course reference model (to establish the topics and dependencies covered in a course), in addition to classroom models, both static and dynamic. As the work presented in this article leverages these results, a background section on the original TDM collection is presented in Section 2.

In this work, an alternative graph foundation is explored for the TDM collection, a two weighted hypergraph, referred to as TDM_{MH} ; the current results focus on static classroom models. This graph can explicitly visualise the n-ary topic coverage and achievement inherent in questions (i.e., a question can assess 1:N topics). However, visualisations of their more general n-ary hyperedge relationships may be more difficult to understand in the broader community. Here, a method to generate TDM_{MH} using commonly available assessment data and a flexible approach to address the complexity of the graphs are presented. The visualisation is a multilevel filtering approach that considers the hypergraph as a collection of levels. The levels partition a graph based on the number of nodes associated with the hyperedges. The content of these levels can be customised (i.e., filtered) and presented according to user preferences. For example, instead of being presented with the complete hypergraph, the users can select topics, achievement or number of responses (e.g., maximum, minimum, range of values) to view part of it. In addition, the visualisations of the levels can be presented in either a cumulative or accumulative mode. The new models are presented using an illustrative example. In addition, a case study that illustrates how TDM_{MH} can be applied to provide insight for instructors is presented. This study uses historical data from a first-year undergraduate level offering of a course on programming and engineering design at The University of British Columbia. The case study includes a comparison of the use of the newly proposed TDM_{MH} with the original TDM approach.

2. Background: The Original TDM Collection

A collection of TDMs has been presented in the authors' previous work [14, 8] to address some of the challenges stakeholders face that relate to the design, delivery and analysis of assessments. An overview of the collection is presented in Figure 1. The TDM collection is revisited in this section as it provides a foundation for the current work.

2.1. Scenarios of Use

The wide variety of stakeholders interacting with these models (e.g., students, instructors, and so on) are shown at the top of the figure; they have roles inside and outside of the classroom.

The users can select the assessment data of interest. For example, they may need to explore data from formative (e.g. assignments) or summative (e.g., examinations) assessments; the data can be from one or more classes (current or previous). The course data includes a specification of the course topics that need to be assessed. The input assessment data are shown on the left of Figure 1.

A preliminary scenario analysis has identified 40 questions for the stakeholders. For example, instructors interact with the models to explore questions such as:

- What topics and their relationships do I need to assess?
- What topics are 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?
- 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?

Here, scenarios of use include filtering by topic(s), or constraints involving achievement or coverage. Additional filters can be added in a straightforward way. Currently, the scenarios focus on using the static models; the reference and dynamic models for the TDM_{MH} are planned in future work. Two examples, derived from the scenario analysis in [14], are described below.

Topics. How are the students performing on a specific topic? One or more topics of interest can be selected in order to identify topics that may need additional attention. For example, an instructor can select one of many topics covered in a course; the levels are filtered to emphasize the topic(s) of interest.

Achievements. What topics have poor achievement? A constraint for visualising a particular achievement level can be selected. For example, an instructor can select to view the minimum achievement scores over all topics in order to identify those that need additional attention; the levels are filtered to emphasize the achievement scores of interest.

The users can choose to combine filters. In other words, they can filter the visualisations based on one or more constraints such as selecting both a topic and an achievement constraint, e.g., the highest performance involving topic T_1 . The results are filtered on a level-by-level basis.

When visualising the filtered results, the user can also choose to view the results in either the cumulative or accumulative mode. In the accumulative mode, only the results for one level are presented. In the cumulative mode, the re-

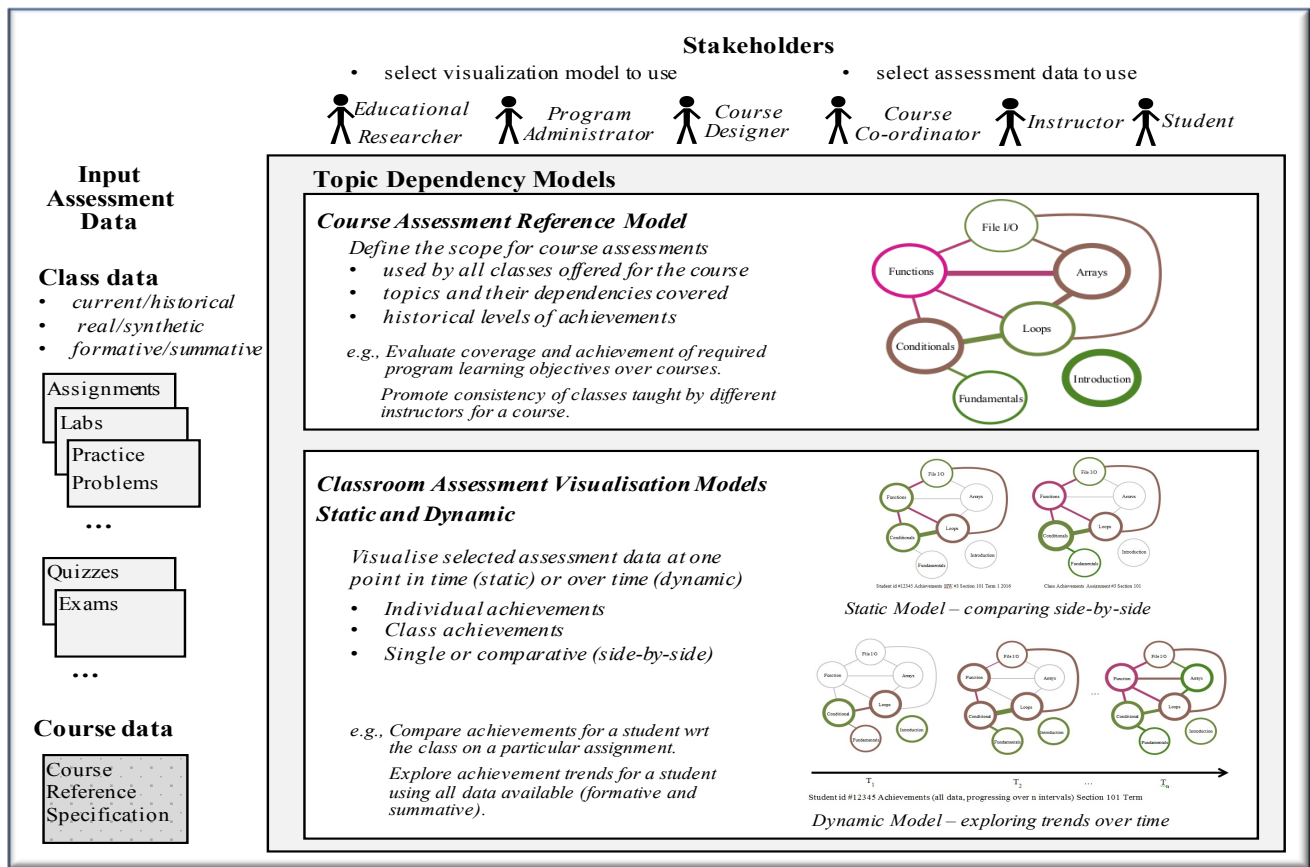


Figure 1: Original TDM model collection: two-weighted undirected graph foundation [14].

sults up to and including the current level of interest are presented.

2.2. Topic Dependency Models

Based on the requests from stakeholders, the input data are selected and transformed into visual models. As illustrated in the main block of Figure 1, the TDM collection consists of two kinds of models: Course Assessment Reference and Classroom Assessment Visualisation (static and dynamic). The Reference model establishes and communicates the required topics and their relationships for a course (e.g., a CS1 course is an introductory programming course covering variables, branching, loops, and so on). It provides a common foundation for all sections of a course offered over time (e.g., CS1 Section 001 Term 1 2019 Instructor A. Smith). The reference model helps to ensure the consistent coverage of topics by different instructors, clearly communicate the scope of the topics to students, and support administrative activities related to monitoring learning objective outcomes.

The Classroom Assessment Visualisation models present assessment data within a class and support comparisons of assessment data between classes at one point in time (static) and over time (dynamic). The assessment data are selectively visualised, e.g., for the whole class, individual students, specific topics, and so on. For example, the user can

choose to visualise static models, in a side-by-side comparison, for a specific assessment (e.g., Assignment 3) for all of the students in two classes. Alternatively, the user can choose to visualise dynamic models to explore the progression of a class over time.

The Reference and Classroom Assessment Visualisation models share a common graph foundation: a two-weighted undirected graph. The vertices represent topics in a course; the edges represent assessment material (e.g., questions) that address the topics the edges are related to. The weights are reflected in the visualisation using a colour palette (achievement) and width (coverage). As the graphs can only represent edges with up to two vertices, questions involving three or more topics must be redistributed in the models as combinations, in order to visualise them. For example, a question covering three topics (e.g., A, B, C) is redistributed with estimations into six relationships for visualisation: A,A; B,B; C,C, A,B; A,C; and B,C.

3. Related Work

The development of TDM_{MH} draws upon the literature from Learning Dashboards and hypergraph based methods. These are briefly presented in this section.

3.1. Learning Dashboards

The learning analytics community continues to actively investigate approaches that support the exploration of learning activities by different stakeholders. With the increase in the use of educational technologies and the advancements in the areas of learning analytics and educational data mining, a new field, commonly known as “Learning Dashboards” has emerged to help make sense of data sets in learning and education [20, 2, 18]. A variety of visualisations such as bar charts [9], pie charts [23], histograms [16], box plots [1], radar graphs [17], and skill meters [6] have been adopted to show the achievements of students for independent (stand-alone) topics. [20] presents a systematic literature review on the use of learning dashboards. Based on the findings of this literature review, the use of graph-based visualisations in learning dashboards has not received much attention. In addition, a number of studies provide strong evidence that opening the model to learners, leading to the notion of Open Learner Models (OLMs) [7], can be effective in helping students learn [3]. The OLMs commonly use a set of individual topics as their underlying structure for modeling learners’ knowledge state, which ignores relationships among topics. An emerging new field applied methods from process mining and sequential data mining to educational data [4] to facilitate better understanding of the educational process. Educational process mining methods predominantly use graphs, but their main focus is on using activity logs to visualise students’ learning process in terms of the time, place, path, pace of learning activities.

3.2. Hypergraph Based Methods

Hypergraphs have been adopted as a foundation for data analytics and visualisations in a wide variety of domains including data warehousing, communication network analysis, geospatial metadata, and cellular biology networks. To support interactive queries in data warehousing systems, a framework for developing dashboards called Dashboard-by-example has been proposed [13]. This framework adopts hypergraph-based techniques to transform dissimilar, heterogeneous data into a homogeneous knowledge space of clusters and partitions. The framework is flexible, as the hypergraph-guided data linkages support the exploration and aggregation of data from multiple perspectives. For the analysis of network traffic traces, available in massive communication logs, a hypergraph based visualisation is proposed in [11]. The network traffic visualisation approach utilises a frequent item set mining method to identify interesting traffic patterns in the large amount of data. The patterns are visualised as hypergraphs with explicit, multi-attribute relationships. A framework to comprehensively address all of the available geospatial metadata standards (i.e., documents) is proposed in [19]. Geospatial metadata describe geographic digital data resources such as earth imagery, geospatial databases and catalogues, and Geographic Information System files. The framework integrates hypergraphs and topic maps, representing the elements and their dependency relationships. The potential for adopting hypergraphs in the domain of cel-

lular biology is introduced in [15]; recently, for example, [21] proposes a framework that adopts hypergraphs and associated hypergraphs to describe, analyse, and identify metabolic network alignments at the full genome level. These alignments are used to discover important similarities and differences between distinct molecular networks: they reveal mappings between components (topological, biological functional) across different networks.

4. Methodology

This section presents the problem definition, graph foundation (hypergraph, level), analysis metrics, and an illustrative example. The graph generation and visualisation algorithm are presented in Section 5, using the example introduced in this section. A case study demonstrating real life examples of how TDM_{MH} can practically be used is presented in Section 6.

4.1. Problem Definition

Based on the related work analysis, a modeling and visualisation approach based on hypergraphs for educational assessment does not appear to be available at this time. The TDM_{MH} (static model) is introduced in this work to help address this gap. More specifically, the research questions are:

- RQ1. How can a two-weighted hypergraph be utilised to model topics, achievements, and coverage in the educational assessment problem domain?
- RQ2. How can the two-weighted hypergraph models representing assessment data be effectively visualised?
- RQ3. How can established graph metrics be used to quantitatively assess the visual complexity of the hypergraph based visualisations of the models?
- RQ4. What are the strengths and limitations of the TDM_{MH} in comparison to the original TDM?

4.2. Hypergraph Foundation of the TDM_{MH}

In this section, the formal definition for the two-weighted hypergraph and the concept of a level are presented.

Definition 1. A two-weighted, undirected hypergraph $G = (V, H)$, where V is the set of vertices representing the topics, and H is the set of hyperedges. A hyperedge $h \in H$ is represented as $h = (C, c_1, c_2)$, where C is the subset of the vertices being connected, and c_1 and c_2 represent the two weighted values for an edge. An edge involving only one node represents a self-loop.

In this work, the nodes are used represent topics; the hyperedges represent the assessment material that covers the topics. The weight c_1 represents the number of learning objects that are tagged with the topics for the hyperedge; and c_2 represents the represents the performance (e.g., achievement) on these learning objects. A simple example of a

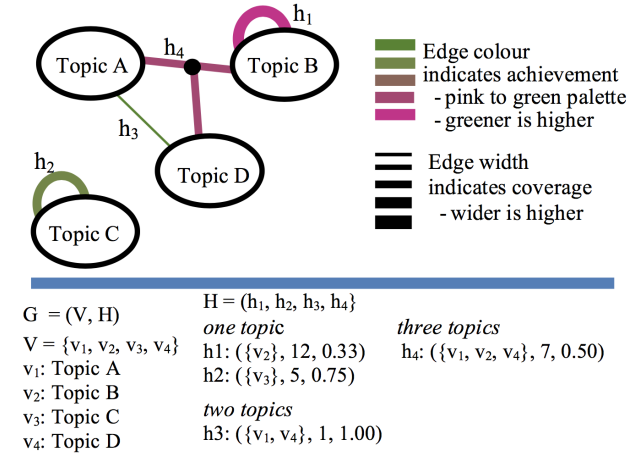


Figure 2: Example of a two-weighted hypergraph: definition and visualisation.

TDM_{MH} visualisation is illustrated in Figure 2. There are four topics in this example (A, B, C, and D); each topic is represented by a node. Topic B and Topic C both have self loops; Topic C is isolated which indicates the topic it represented is assessed on its own. For Topic B, the edge has a dark pink colour and a wide width, reflecting the poor average performance (33%) on the questions answered and the large number of responses. The hyperedge for Topic C is a medium green colour and has a medium width; this represents the good average performance (75%) and a moderate number of responses to questions. A 2-ary hyperedge exists between Topics A and D. This edge is dark green and thin, in representing the excellent performance (100%) on the related questions and a small number of responses. A 3-ary hyperedge exists between Topics A, B, and D. This edge is a brownish pink colour and has a moderate weight, which represents the moderate average performance (50%) and a moderate number of responses to questions on these three topics.

The concept of a level is used in this work to support the visualisation. For example, in the simple example given in Figure 2, there are three levels. $Level_1$ is a subset of the hyperedges with one vertex (i.e., h_1, h_2), $Level_2$ is a subset of the hyperedges with two vertices (i.e., h_3), and $Level_3$ is a subset of the hyperedges with three vertices (i.e., h_4). More generally, $Level_i$ is the subset of hyperedges involving i vertices.

4.3. Analysis Metrics

A basic collection of structural metrics has been selected, which are well-defined for hypergraphs (e.g., [5, 24]). As graphs are a special case of hypergraphs, the metrics can be applied to both hypergraphs and graphs. They are used in this work to help explore the complexity of the TDM_{MH} visualisations in Section 5. They are also applied in the comparative case study in Section 6 to compare and contrast the original TDM with the TDM_{MH} .

In the following definitions, $G = (V, H)$ is an undirected hypergraph, $v \in V$, and $h \in H$. The order of the hypergraph

G is the cardinality of V , i.e., $|V| = n$. In other words, the order of the graph is the number of vertices it has. The size of the hypergraph G is the cardinality of H , i.e., $|H| = m$. In other words, the size of the hypergraph is the number of hyperedges it has.

Degree metrics can be considered at both the individual node level and for a hypergraph as indicators of a graph's complexity. For an individual vertex the degree is the number of hyperedges (ingress or egress) the node is contained in. A node in a graph, denoted $G(v)$, can be contained in one or more hyperedges. For a hyperedge (h_j), $j \in J$ that contains the node v , the degree of that node is $d(v) = |J|$. In Figure 2, for example, the $d(\text{Topic A})$ is two, as it is contained in two hyperedges h_2 and h_3 . The $d(\text{Topic B})$ is three, as the hyperedge h_1 contributes a count of two (egress and ingress) and h_4 also contains the node.

Beyond the individual node degree computations, additional degree metrics are considered at the graph level. These include the total degree of G , denoted by $d_t(G) = |H(v)|$; the average degree of a graph is the total divided by the order, i.e., $d_t(v)/|V|$, or $|H(v)|/|V|$. In addition, the maximal degree of a hypergraph G is denoted by $\Delta(G)$; the minimal degree is denoted by $\delta(G)$. For example, the average, maximal, and minimal degrees of Figure 2 are 2.25, 3, and 2.

4.4. Illustrative Example

An example based on six students, three formative assignments (five questions per assignment), and six topics has been defined by the authors. The following constraints are considered for creating the illustrated example:

- The questions must span a range of one to four topics.
- There must be a range in the individual student achievements in the responses (low to high scores).
- There must be a range in the number of responses to the questions.
- There must be a range in the average scores for the questions.
- Assessments on groups of topics demonstrate a range of achievements (very poor to very strong); one or more questions address the groups of topics.
- The multilevel visualisation algorithm with a list of filtering options (e.g., selected topic(s), achievement, cumulative/accumulative mode) can be clearly illustrated.

Table 1 presents a sample data set meeting these constraints. As input to the TDM_{MH} generation algorithm, the data are formatted into two CSV files: (1) A student-question-answer (SQA) file that contains the student identifier, question identifier, and score (correct/incorrect), and (2) a question-topic (QT) file that contains the question identifier and topics (i.e., tags) it addresses.

Q	Topic List	Student ID					
		S_1	S_2	S_3	S_4	S_5	S_6
Assignment 1							
Q_1	T_1	-	1	-	-	0	1
Q_2	T_3	1	1	1	1	1	1
Q_3	T_4	0	1	0	1	1	1
Q_4	T_1, T_2	1	1	0	1	1	0
Q_5	T_1, T_4	1	-	-	0	1	0
Assignment 2							
Q_6	T_4	0	0	1	0	0	1
Q_7	T_1, T_4	-	0	1	1	0	0
Q_8	T_4, T_5	0	1	0	0	0	1
Q_9	T_1, T_4, T_5	0	0	0	0	-	1
Q_{10}	T_1, T_2, T_4, T_5	0	0	0	1	0	1
Assignment 3							
Q_{11}	T_1, T_4	1	-	0	1	-	1
Q_{12}	T_2, T_6	0	0	0	0	0	0
Q_{13}	T_1, T_2, T_6	-	-	1	1	-	1
Q_{14}	T_1, T_2, T_4, T_5	-	0	1	0	0	1
Q_{15}	T_2, T_4, T_5, T_6	-	0	0	1	1	1

Table 1

Illustrative example: questions, topic lists, and student responses. The value 1 indicates the question is answered correctly; 0 indicates it is answered incorrectly; and - indicates it is not attempted.

5. TDM_{MH} Static Model

In this section, the approach to generating and filtering a TDM_{MH} is presented. Section 5.1 presents methods for generating the graph and Section 5.2 explores filtering mechanisms to address the complexity of hypergraph visualisations. The results are briefly discussed in Section 5.3.

5.1. Generating a TDM_{MH}

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_{MH} . High-level code and notation are presented in Algorithm 1. The algorithm consists of three high level steps: create working dictionaries and matrices; define the TDM_{MH} graph elements (vertices and hyperedges); and visualise (i.e., plot) the TDM_{MH} graph.

The first six steps of Algorithm 1 transform the data in the SQA.csv and QT.csv files into working dictionaries and matrices. In Steps 1, 2, and 3 of the algorithm, three dictionaries are created, $QDict$, $SDict$, $TDict$, to map array and matrix indices to question, student, and topic identifiers. In Steps 4, 5, and 6, three working matrices are created: T , A , and R . The information on topics assigned to each question is represented in matrix T , in which $t_{ij} = 1$ indicates that question i is tagged with topic j and $t_{ij} = 0$ indicates that question i is not tagged with topic j . The correctness of the answers provided by the users are represented in matrix A , where $a_{ij} = 1$ indicates that user i has answered question i correctly, $a_{ij} = 0$ indicates that user u has answered question i incorrectly, and $a_{ij} = n$ indicates that user u has not attempted question i . Matrix R is used to keep track of attempted questions, where $r_{ij} = 1$ if user i has attempted question i and zero otherwise. Figure 3 illustrates the data

Algorithm 1 Generating a TDM_{MH}

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

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: $A \leftarrow CreateA(SQA.CSV, SDict, QDict)$
- 6: $R \leftarrow CreateR(SQA.CSV, SDict, QDict)$

Compute the Graph Elements: Vertices and Edges

- 7: $VList \leftarrow ComputeV(TDict)$
- 8: $HList \leftarrow ComputeH(T, A, D, TDict)$

Create and visualise the Graph

- 9: $TDMStatic \leftarrow CreateTDM(VList, HList)$
- 10: $Visualise(TDMStatic, filters)$

stored in the dictionaries ($QDict$, $SDict$, $TDict$) and the matrices (T , A , R).

Steps 7 and 8 in Algorithm 1 establish the graph vertices and hyperedges. $VList$ stores the list of the vertices of the TDM_{MH} graph and $HList$ stores the list of its hyperedges. The coverage and competency associated with an edge are both computed using T , A and R within the $ComputeH$ function. The coverage (Cov) associated with a hyperedge among a set of vertices $V = v_j, \dots, v_k$ is computed based on the number of attempts that have been made on questions that are tagged with exactly topics that are included in V . It is computed using the following formula:

$$Cov(V) = \sum_{i \in QDict} \left(\prod_{j \in V} t_{ij} \prod_{j \notin V} (1 - t_{ij}) \right) \sum_{u \in SDict} r_{ui} \quad (1)$$

The outer summation loops through all of the questions. For a question i , $(\prod_{j \in V} t_{ij} \prod_{j \notin V} (1 - t_{ij}))$ results in one if question i is tagged with exactly the topics that are included in V and zero otherwise. Together, these two parts identify questions that are tagged with exactly topics that are included in V . The inner summation then counts the number of attempts that have been made on such a question. In the given example, $Cov(\{T_1, T_4\})$ has coverage of 13, which consists of contributions of 4 from Q_5 , 5 from Q_7 , and 4 from Q_{11} .

The achievement ($Achv$) associated with a hyperedge among a set of vertices $V = v_j, \dots, v_k$ is computed as the success rate of answering questions that are tagged with exactly topics, which are included in V . It is computed using the following formula:

$$Achv(V) = \frac{\sum_{i \in QDict} \left(\prod_{j \in V} t_{ij} \prod_{j \notin V} (1 - t_{ij}) \right) \sum_{u \in SDict} a_{ui}}{Cov(V)} \quad (2)$$

QDict:

{'Q₁':0, 'Q₂':1, 'Q₃':2, 'Q₄':3, 'Q₅':4, 'Q₆':5, 'Q₇':6, 'Q₈':7, 'Q₉':8, 'Q₁₀':9, 'Q₁₁':10, 'Q₁₂':11, 'Q₁₃':12, 'Q₁₄':13, 'Q₁₅':14}

SDict:

{'S₁':0, 'S₂':1, 'S₃':2, 'S₄':3, 'S₅':4, 'S₆':5}

TDict:

{'T₁':0, 'T₂':1, 'T₃':2, 'T₄':3, 'T₅':4, 'T₆':5}

Topics (6)		Questions (15)	
Questions (15)	Students (6)	Matrix A	
		$\begin{bmatrix} n & 1 & 0 & 1 & 1 & 0 & n & 0 & 0 & 1 & 0 & n & n & n \\ 1 & 1 & 1 & 1 & n & 0 & 0 & 1 & 0 & 0 & n & 0 & n & 0 \\ n & 1 & 0 & 0 & n & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ n & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & n & 0 & n & 0 & n & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$	
		Matrix R	
		$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	
		Matrix T	
		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$	

Figure 3: Illustrative Example Data Organization. The data represented in Table 1 are loaded and transformed into three dictionaries: *QDict*, *SDict* and *TDict* (Steps 1, 2, and 3 in Algorithm 1). The dictionaries and the input files are used to create three working matrices: T, A, and R (Steps 4, 5, and 6). These matrices are used to create the graph elements (Steps 7, 8). The TDM_{MH} model is created (Step 9) and subsequently visualised (Step 10) (refer to Figure 4).

The numerator of this formula count the number of correct answers given to questions that are tagged with exactly topics included in V . The formula is similar to how $Con(V)$ is computed, but instead of using $\sum_{u \in SDict} r_{ui}$ to count the number of attempts made on questions that are tagged with exactly topics included in V , $\sum_{u \in SDict} a_{ui}$ is used to compute only the number of correct answers. Dividing this number by $Cov(V)$ produces the rate of correctly answering questions tagged with only topics in V . In the given example $Achv(\{T_1, T_4\})$ is computed as $\frac{2+2+3}{13} = \frac{7}{13} = 0.54$. Note that Cov and $Achv$ of a single topic j can be computed via the same two formulas by using $V = v_j$.

Step 9 of Algorithm 1 creates the graph model based on the $VList$ and $HList$ computed in steps 7 and 8 of the algorithm. The lower part of Figure 4 presents the $VList$, $HList$, and two-weighted hypergraph model created for the illustrative example.

5.2. Visualising a TDM_{MH}

Step 10 of Algorithm 1 visualizes the model according to the user selected filtering options and display mode. The TDM_{MH} approach supports filtering combinations including one or more topics as well as one or more achievement conditions.

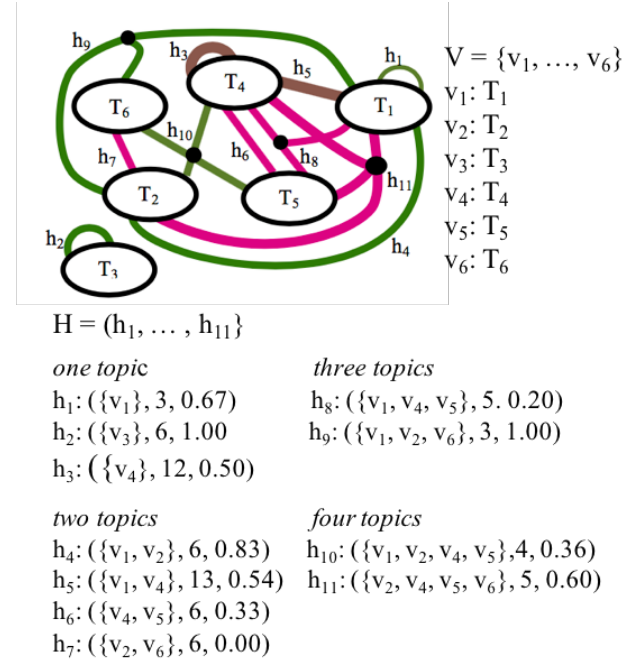


Figure 4: Complete TDM_{MH} for the data set provided in Table 1 using Algorithm 1 (no filtering, default cumulative mode). The example does not have questions that involve five or six topics.

Three straightforward visualisation examples are presented in this section for illustrative purposes: no filtering (default cumulative mode), filtering by one topic (accumulative mode), and filtering by an achievement condition (cumulative mode).

To accomplish the visualisation step, the levels in the graph are first identified based on the structure of the graph. The filters selected by the user are then applied to the levels. Lastly, the filtered levels are displayed in accordance with the mode (accumulative, cumulative).

The illustrative example (with six nodes) has six levels. Table 2 illustrates the levels before any filtering is done. For example, $Level_1$ is the subset of hyperedges $\{h_1, h_2, h_3\}$.

5.2.1. Example 1 (no filtering)

This example illustrates the situation in which no filtering options are selected by the user and the default cumulative mode is used. In step 10 of Algorithm 1, the levels for the graph are first identified (refer to Table 2).

If the user does not select any filters, then all of the levels are displayed by default in the cumulative mode, including all topics, achievements, and coverage of the learning objects. In this situation, the filtered levels are the same as the original levels, i.e., $Level_i$ and $Level'_i$ are the same (refer to Example 1 in Table 2).

The default cumulative mode is used, which results in the visualisation in Figure 4.

The metrics associated with this example are presented in Table 3. The row corresponding to the visualised model are emphasized with the lightgray colour; intermediate assessments are not highlighted. In the cumulative visuali-

Level ID	Level Membership	Example 1 Level' Membership (no filtering)	Example 2 Level' Membership: Topic is T_1	Example 3 Level' Membership: Achievement is $\leq 20\%$ OR $\geq 80\%$
$Level_1$	$\{h_1, h_2, h_3\}$	$\{h_1, h_2, h_3\}$	$\{h_1\}$	$\{h_2\}$
$Level_2$	$\{h_4, h_5, h_6, h_7\}$	$\{h_4, h_5, h_6, h_7\}$	$\{h_4, h_5\}$	$\{h_4, h_7\}$
$Level_3$	$\{h_8, h_9\}$	$\{h_8, h_9\}$	$\{h_8, h_9\}$	$\{h_8, h_9\}$
$Level_4$	$\{h_{10}, h_{11}\}$	$\{h_{10}, h_{11}\}$	$\{h_{10}\}$	\emptyset
$Level_{5+}$	\emptyset	\emptyset	\emptyset	\emptyset

Table 2

Illustrative Example: Levels before and after filtering (Examples 1, 2, and 3).

Example	$\delta(H)$ (min)	$\Delta(H)$ (max)	E (avg)	Order	Size
Example 1 (no filtering, default cumulative mode)					
Level 1'	2	2	2.00	3	3
Level 2'	1	2	1.60	4	5
Level 3'	1	2	1.20	2	5
Level 4'	1	2	1.20	2	5
Levels 1'	2	2	2.00	3	3
Levels 1'-2'	1	4	2.33	7	6
Levels 1'-3'	2	6	3.33	9	6
Levels 1'-4'	2	7	4.67	11	6
Example 2 (filter on topic, accumulative mode)					
Level 1'	2	2	2.00	1	1
Level 2'	2	2	1.33	2	3
Level 3'	2	2	1.20	2	5
Level 4'	1	1	1.00	1	4
Example 3 (filter on achievement, cumulative mode)					
Level 1'	2	2	1.00	1	1
Level 2'	1	2	2.00	2	3
Level 3'	1	2	1.20	2	5
Level 4' (\emptyset)	0	0	0	0	0
Levels 1'	2	2	2.00	1	1
Levels 1'-2'	1	2	1.50	3	4
Levels 1'-3'+	1	3	2.00	5	6

Table 3

Illustrative Example: Metric summaries (Examples 1, 2, and 3).

sation involving Levels 1'-4', the minimum and maximum degree node values occur with the isolated node T_3 , 2, and node T_4 , 6, respectively. The average degree node value is relatively high, 4.67. This indicates a more complex visualisation for the users. All six of the nodes and all 11 of the hyperedges are included in the visualisation.

5.2.2. Example 2 (filtering by topic)

This example illustrates the situation in which the user select the option to filter by one topic (T_1) and the accumulative mode is used. In step 10 of Algorithm 1, the levels for the graph are first identified (refer to Table 2).

For example, if a user requests to filter with respect to topic T_1 , the levels are filtered accordingly. In this case, for example, the filtered $Level'_i$ is the subset of hyperedges $\{h_1\}$, as only h_1 contains T_1 (refer to Example 2 in Table 2).

After the levels are filtered, they are visualised. In the accumulative mode, only the hyperedges for one level are presented at a time. The four filtered levels are visualised in Figure 5.

The metrics associated with Example 2 are presented in

Table 3. The rows corresponding to the visualised model are emphasized with the lightgray colour; all four rows are highlighted in the table to indicate they are visualised. The minimum node degree values ([1..2]) show limited variation; the maximum degree node values ([1..2]) are also low. There are a smaller number of nodes ([1..2]); a small number of hyperedges ([1..2]) are involved in the filtered levels. This results in reduced average node degree values ([1.00..2.00]), reflecting the reduced visual complexity of the individual levels in comparison to Example 1.

5.2.3. Example 3 (filtering by achievement)

This example illustrates the situation in which the user select the option to filter by an achievement condition, which identifies the more extreme performance outcomes in the assessment data for the class ($\leq 20\%$ OR $\geq 80\%$); the cumulative mode is used. In step 10 of Algorithm 1, the levels for the graph are first identified (refer to Table 2).

The levels are filtered according to the achievement condition selected by the user ($\leq 20\%$ OR $\geq 80\%$). There are five hyperedges that meet this achievement condition: h_2, h_4, h_7, h_8, h_9 (refer to Example 3 in Table 2).

After the levels are filtered, they are ready to be visualised. In the cumulative mode, the presentation for a level includes the hyperedges for the current levels and the previous levels. The three (non-empty) filtered levels are visualised in Figure 6.

The metrics associated with this example are presented in Table 3. The rows corresponding to the visualised model are emphasized with the lightgray colour; rows containing intermediate assessments are not highlighted. The minimum node degree values ([1..2]), is low; the maximum degree node values ([2..3]) are also relatively low. There is a wider variation in the number of nodes involved in the cumulative visualisations ([1..6]); a relatively small number of hyperedges ([1..5]) are involved in the filtered levels. This results in reduced average node degree values [1.50..2.00] in the individual levels reflecting the reduced visual complexity of the individual levels in comparison to Example 1; the overall average in the cumulative visualisation involving the three non-empty levels is 2.00.

5.3. Discussion

The straightforward examples explore the use of the two-weighted hypergraph based TDM_{MH} model to generate (RQ1) and visualise (RQ2) assessment data. The representation of

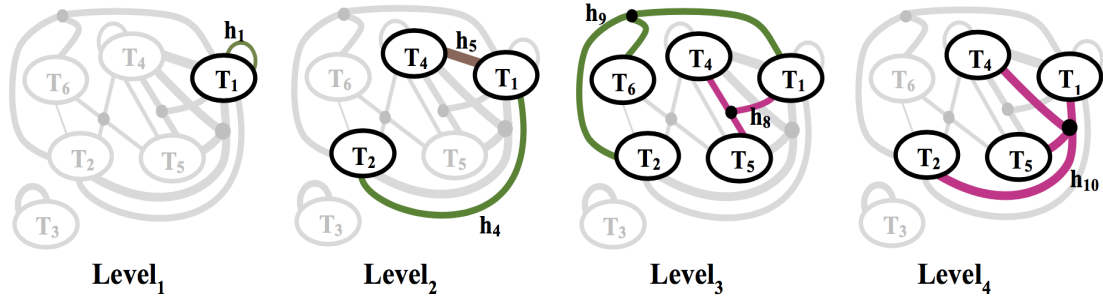


Figure 5: Illustrative Example 2 multilevel visualisation: one topic is selected (T_1) as a filter (accumulative mode).

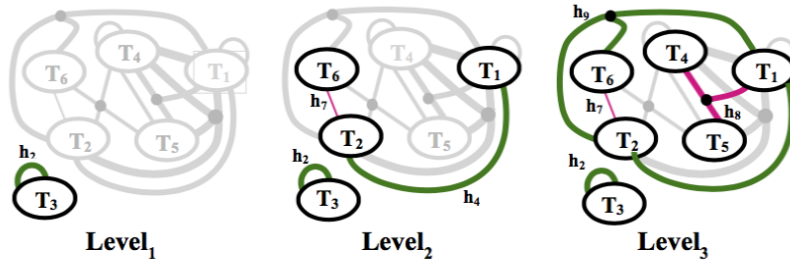


Figure 6: Illustrative Example 3 multilevel visualisation: achievements at the more extreme ranges (very low $\leq 20\%$, very high $\geq 80\%$) of the grades are selected using filter selections (cumulative mode).

the assessment data covering one or more topics is straightforward with the TDM_{MH} as it leverages the n-ary relationships of the hyperedges. The support for displaying all or part of the model of interest using the concepts of levels and filtering is flexible, allowing the user to focus on specific aspects of the assessment data. The preliminary set of metrics are applied, providing some insight on their relationship to the visual complexity of the models (RQ3). The average node index value appears to be useful as an indicator of the visual complexity. In the next section, the TDM_{MH} model is further explored using a historical case study in a self study and a comparative study with the original TDM model (RQ4).

6. Case Study

6.1. Historical Data

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 of the course 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 in nine modules: 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.

The data have been collected from the final exam of the

course, which are captured via the Gradescope platform [22], a system for the on-line assessment of handwritten exams. The final exam of this offering consists of 17 independent sub-questions that formed a total of eight main questions. The questions are tagged using 20 concept-level topics, defined by the instructors, based on the eight modules that are covered in this course. For example, the Strings Module is further decomposed into three topics: String-Length, String-Copy and String-Compare, providing a finer level of granularity.

6.2. TDM_{MH} Visualisation

Three visualisation cases are presented in this section: no filtering, filtering by topic, and filtering by an achievement condition. The cumulative mode is used in all three cases. In these visualisations, a self-loop on a node is represented by the width and colour associated with the circle representing that node. The metrics associated with these cases are presented in Table 4.

6.2.1. Case 1 (no filtering)

Figure 7 shows the TDM_{MH} with no filters for the exam data described in Section 6.1.

It indicates that the exam adequately covers all of the modules of the course and that all modules except Conversions (including the Hex and Octal topics) are highly connected to one another. The exam includes questions that had only a single topic (e.g., h_1 on Hex), two topics (e.g., h_{14} on While Loops and DAQ-Write), three topics (e.g., h_9 on Printing, File Output and 2D Arrays) and four topics (e.g.,

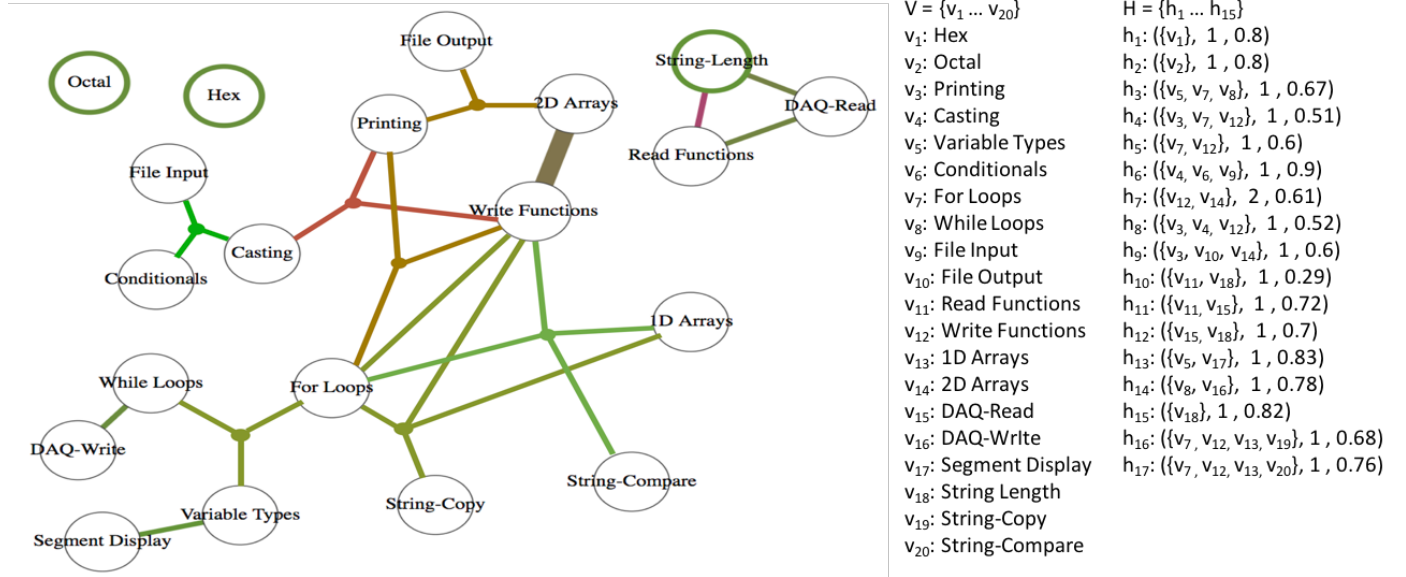


Figure 7: TDM_{MH} for the final exam of an offering of a first-year course on programming and engineering design

Case Study	$\delta(G)$ (min)	$\Delta(G)$ (max)	E (avg)	order	size
Case 1 (no filtering, default cumulative mode)					
Level 1'	2	2	2.00	3	3
Level 2'	1	2	1.40	7	10
Level 3'	1	3	1.0	5	10
Level 4'	1	2	0.80	2	5
All (Levels 1'-4')	1	6	2.15	17	20
Case 2 (filter on topic, Level 4, cumulative mode)					
Level 2'	1	2	1.33	2	3
Level 3'	1	2	1.50	2	4
Level 4'	1	2	1.60	2	5
Levels 1'-4'	1	6	2.25	6	8
Case 3 (filter on achievement, Level 3, cumulative mode)					
Level 2'	1	1	1.00	2	4
Level 3'	1	3	0.67	3	6
Levels 1'-3'	1	3	1.62	5	8

Table 4

Case Study: TDM_{MH} Metric Summaries for Cases 1, 2, and 3.

h_{17} on For Loops, 1D Arrays, Write Functions, and String-Compare). The further decomposition to concept-level topics provides insights which might have not been possible to gauge using module-level tags. For example it shows that students have done 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 are able to do well in 1D-Arrays, but perform quite poorly on questions on 2D-Arrays.

For the first case study, its associated metrics are presented in the top part of Table 4. The row corresponding to the visualised model is emphasized with the lightgray colour; the rows with intermediate assessment values are not highlighted. The minimum degree node value is one; this occurs, for example, with the nodes representing File Input (i.e., v_9),

File Output (i.e., c_10), and others. The maximum degree node value is six which occurs with a node representing the Write Functions topic (v_{12}). The average degree node value is 2.15 for the visualised model; this study includes all of the hyperedges and nodes, 17 and 20 respectively.

6.2.2. Case 2 (filtering by topic)

Figure 8 applies the cumulative topic-based $level_4$ filter on the "Write Functions" concept. This filter will enable instructors to determine the coverage and performance of students on a particular concept. The TDM_{MH} in the given example demonstrates that the Write Function has been extensively covered in the exam. The performance of students on questions that covered write functions in combination with 1D arrays, String-copy, and String-compare have been quite good; however, their performance on questions that cover Write functions in combination with 2D Arrays, Casting, and Printing has not been as good.

The metrics associated with this case are presented in Table 4. The row corresponding to the visualised model is emphasised with the lightgray colour; the rows with intermediate assessments are not highlighted. The minimum degree node value is one (e.g., 2D Arrays (v_{14})). The maximum degree node value is six; this occurs with Write Functions (v_{12}). The average degree node value is 2.25 for the visualised model, which is higher than the average value for Case 1 (no filtering). The Write Functions is the topic that has the highest number of appearances in the hyperedges. Filtering on a topic results in a higher average degree value in this study, as there are more connections among the nodes selected; a subset of six hyperedges and eight nodes are considered.

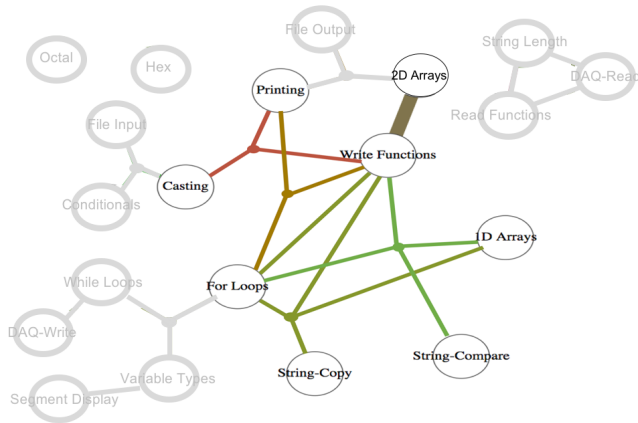


Figure 8: Cumulative topic-based $level_4$ filter on "Write Functions" is selected.

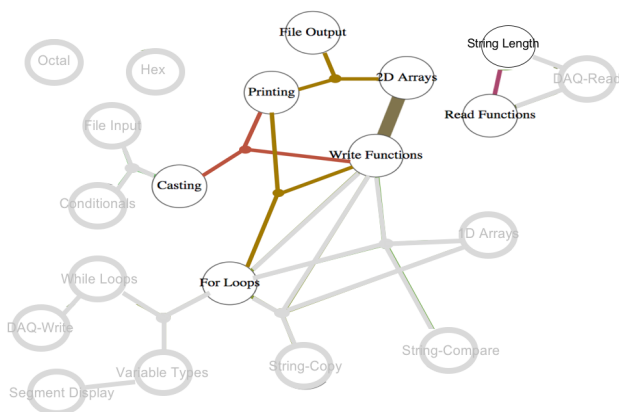


Figure 9: Cumulative achievement $level_3$ filter with " ≤ 60 " is selected.

6.2.3. Case 3 (filtering by achievement)

Figure 9 applies the cumulative achievement $level_3$ filter with " ≤ 60 ". This filter enables instructors to determine gaps in students' knowledge. The TDM_{MH} in the given example demonstrates questions concept combinations that the students have performed poorly on in this exam. Interestingly, all of the hyperedges that include Printing are selected in this filter. This may suggest that there is a general misunderstanding or misconception about how print statements are used.

The metrics associated with this case are presented in Table 4. The row corresponding to the visualised model is emphasised with the lightgray colour; the rows with intermediate assessments are not highlighted. The minimum degree node value is one (e.g., String-Length (v_{18})). The maximum degree node value is three (e.g., Write Functions (v_{12})). The average degree node value, 1.62, is quite low for the visualised model. Here, the filtering selects a subset of five hyperedges and eight nodes for consideration.

6.2.4. Discussion

The three presented cases explore the use of the two-weighted hypergraph based TDM_{MH} model to generate (RQ1)

and visualise (RQ2) assessment data from a historical data set. The support for visualising all or part of the model of interest using the concepts of levels and filtering allows instructors to gain additional insights. The preliminary set of metrics are applied to quantify the complexity of the models (RQ3). The average node index value appears to be useful as an indicator of the visual complexity of the model. As demonstrated by Case 2 and Case 3, applying filters may increase or decrease the average node index of the model compared to Case 1 which had no filters. Therefore, the number of nodes in combination with the average node index value may be a more useful indicator of the visual complexity of the model.

6.3. Comparing the TDM and TDM_{MH} Approaches (no filtering)

This section compares and contrasts the use of the TDM_{MH} with the previously proposed TDM approach in this study, Case 4. The discussion is organised with respect to the four RQs. Figure 10 visualises the final exam results for the data set described in Section 6.1; this study extends Case 1 in 6.2.1 (i.e., no filtering) to include the TDM approach.

The two approaches can both be used to represent the assessment data. Essentially, the nodes represent assessment topics and the two-weighted edges/hyperedges represent the coverage and achievements in the assessment data. A colour palette and line widths can visualise the weights in both the TDM and TDM_{MH} . The common layout of the nodes has been performed manually, with the goal of reducing the number of crossing edges. This layout, however, does not strongly reflect the organisation of topics within a course. For example, three DAQ topics (DAQ-Read, DAQ-Write, Segment Display) are taught within one module of the course; these topics are scattered in the visualisations.

The fundamental difference between the two models is based on their interpretation of how a question i that is tagged with a set of topics $V = v_j, \dots, v_k$ contributes to the coverage and achievement of the models. In the algorithm used for creating the TDM_{MH} hypergraphs, question i only contributes to the coverage and achievement of the hyperedge that is associated with V ($Achv(V)$) (see Section 5.1 for details). In contrast, in the algorithm used for creating the TDM two-weighted graphs, question i contributes to the coverage and achievement for all of the edges that include an element of V (see [14] for details). This approach ensures that a question that is tagged with more than two topics (i.e., $|V| > 2$), where a particular edge is not associated with V , contributes to the computation of achievements within the TDM. Let's consider an example using the models that are visualised in Figure 10. In the TDM_{MH} approach, the question tagged with the three topics of File Input, Conditionals and Casting only contribute to the coverage and achievement of one hyperedge; this is the one associated with these three topics. In the two-weighted TDM, this question contributes to the coverage and achievement of six separate edges: self loops on each of the File Input, Conditionals and Casting topics as well as edges between File Input and Casting, File

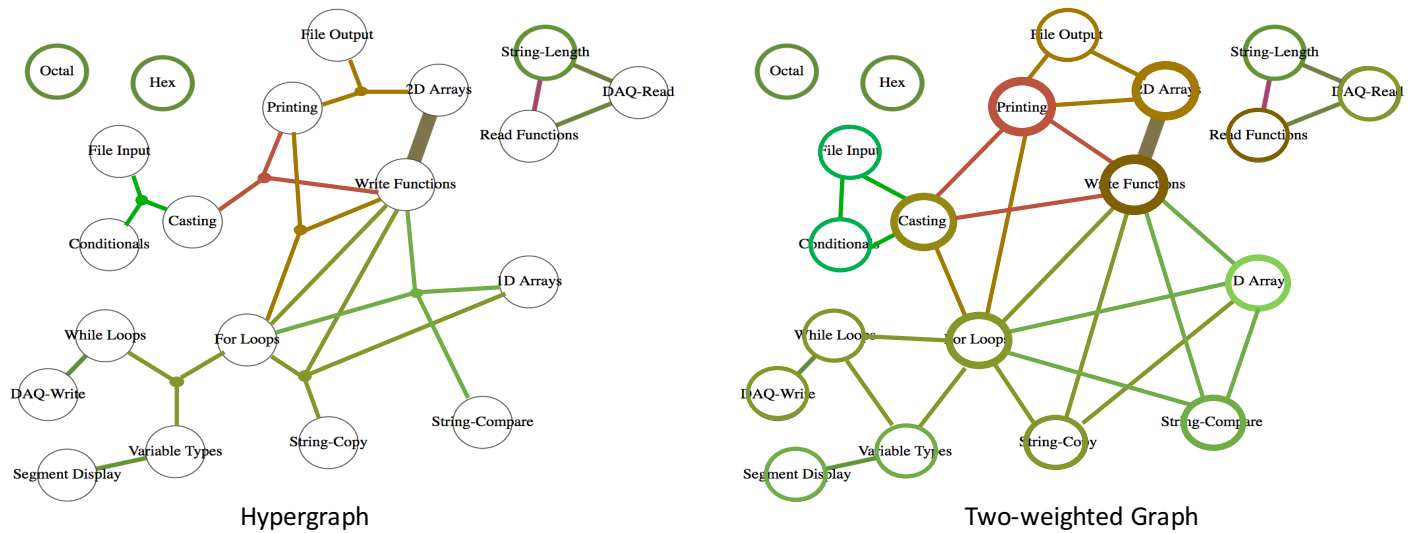


Figure 10: Comparison of the proposed two-weighted hypergraph (TDM_{MH}) and the original two-weighted graph (TDM) for visualising the final exam results of an offering of a first-year course on programming and engineering design

input and Conditionals, and Casting and Conditionals.

Based on the difference in the interpretation of how a question contributes to the coverage and achievement of edges, the two models differ quite significantly. A few of these similarities and differences with reference the metrics provided in Table 5 are presented below.

- With respect to the number of nodes, both visualisations are the same, as this is not impacted by the different graph foundations: 20 topics need to be visualised. Filtering options that impact the number of nodes (e.g., multi-level visualisations in the hypergraphs) are not part of the Case 4 study.
- The TDM_{MH} approach provides explicit information about questions that are tagged with more than two topics, whereas this information is implicitly provided in the two-weighted TDM. For example, as discussed previously, the contribution of a question that is tagged with three topics (e.g., File Input, Conditionals and Casting) is explicitly visualised in the hypergraph approach; however, the contribution of this question is implicitly distributed among six edges in the two-weighted graph approach.
- The TDM_{MH} can have up to 2^n hyperedges where as the two-weighted graph TDM can only have $\frac{n(n-1)}{2}$ edges for a graph with n vertices. This enables TDM_{MH} to model more complex domains with more than one edge between two vertices. For example there are a total of three edges between For Loops and Write Functions. One based on a question that involved only these two topics; on hyperedge based on a question that involved these two topics as well as 1D arrays and Strong Compare; and a third hyperedge based on a question that involved these two topics as well as 1D Arrays and String Copy.
- The TDM_{MH} approach is generally sparser compared to the two-weighted TDM approach. This is because in the TDM_{MH} approach each question can only contribute to one edge. In contrast, in the two-weighted graph approach one question can contribute to multiple edges. For example, in Figure 10, the hypergraph TDM has a total of 17 hyperedges whereas the two-weighted TDM has a total of 49 edges. The average node degree values of the TDM and TDM_{MH} are 2.80 and 2.15, respectively, indicate this measure may provide some insight into the visual complexity of the models.
- The main strength of TDM_{MH} is that it accurately represents the underlying data set. Therefore, the model can be used to determine precisely the achievement and coverage on a set of nodes V . The main limitation of TDM_{MH} is in its inference power as it lacks the ability to infer achievements or coverage on the non-complete subsets of V . As an example, referring to Figure 10, TDM_{MH} provides accurate information about how For Loops have been covered in combination with other topics, but it cannot infer achievement on For Loops by itself.
- The main limitation of the original TDM approach is that it cannot accurately represent the underlying data set; therefore, the model does not have the capacity to determine precisely the achievement and coverage on a set of nodes V . The main strength of this model is in its inference power as it can infer achievements or coverage of subset of nodes of size 1 (self loops) or size 2 (general edges). For the same example given before, the original TDM can be used to infer the achievement and coverage on the topic of For Loops by itself; however, it cannot be used to provide highly

Model	$\delta(G)$ (min)	$\Delta(G)$ (max)	E (avg)	order	size
<i>TDM</i>	2	10	2.80	49	20
<i>TDM_{MH}</i>	1	6	2.15	17	20

Table 5

Case Study: *TDM* and *TDM_{MH}* Metric Summaries (no filtering, default cumulative mode)

accurate information about how the topic of For Loops has been covered in combination with other topics.

7. Conclusions and Future Work

The preliminary results of modelling and visualising topic-level achievements and coverage in the educational assessment problem using a two-weighted, multilevel, hypergraph-based model (called *TDM_{MH}*) are presented. The generation of *TDM_{MH}* utilises matrix computations, which makes it scalable and efficient. The visualisation helps to address the complexity of a hypergraph through a flexible, multilevel approach. A set of metrics are employed to quantitatively assess the visual complexity of the hypergraph based visualisations of the models. A case study that illustrates how *TDM_{MH}* can be applied to provide insight in the context of a large university course is presented. The case study also compares and contrasts the proposed approach with the original *TDM* model that uses a graph instead of a hypergraph.

There are several limitations in the current work which restrict the generalisability of the results. One of the significant limitations of this study is that *TDM_{MH}*s have only been applied to one course from the computer science domain. It is important to acknowledge that these may be hard to comprehend for stakeholders without formal training in algorithmic literacy. A second limitation of the current work is that the case studies only explore the benefits of using *TDM_{MH}*s from an instructor's perspective. A comprehensive stakeholder and scenario analysis study is planned for future work, which would investigate the usability of the model across a range of disciplines.

A number of other directions are also planned for the next steps in this research. (1) Adding layout algorithms that optimise the model visualisation; the algorithm may use a range of options such as reducing the number of crossing edges in the graph or grouping topics with respect to instructional design constraints (e.g., groups of related topics covered in a course). (2) Exploring the use of dynamic models to present a series of *TDM_{MH}*s that illustrate changes on achievements and coverage over time. (3) Exploring the use of 3D and hierarchical models for visualising a *TDM_{MH}*. (4) Exploring additional metrics to improve the quantitative assessment of the *TDM* and *TDM_{MH}* models' complexity.

References

- [1] Albert, D., Nussbaumer, A., Steiner, C., 2010. Towards generic visualisation tools and techniques for adaptive e-learning, in: Proc of the 18th International Conference in Computers in Education, pp. 61–65.
- [2] Argumentation, V., 2003. Software tools for collaborative and educational sense-making. Paul A. Kirschner, Simon J. Buckingham Shum and Chad S. Carr (Eds.) Springer-Verlag: London , 117–135.
- [3] Bodily, R., Kay, J., Alevy, V., Jivet, I., Davis, D., Xhakaj, F., Verbert, K., 2018. Open learner models and learning analytics dashboards: a systematic review, in: Proc of the 8th International Conference on Learning Analytics and Knowledge, pp. 41–50.
- [4] Bogarín, A., Cerezo, R., Romero, C., 2018. A survey on educational process mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.
- [5] Bretto, A., 2013. Hypergraph Theory. Springer.
- [6] Bull, S., Ginon, B., Boscolo, C., Johnson, M., 2016. Introduction of learning visualisations and metacognitive support in a persuadable open learner model, in: Proc of the Sixth International Conference on Learning Analytics & Knowledge, pp. 30–39.
- [7] Bull, S., Kay, J., 2010. Open learner models, in: Advances in Intelligent Tutoring Systems. Springer, pp. 301–322.
- [8] Cooper, K., Khosravi, H., 2018. Graph-based visual topic dependency models: Supporting assessment design and delivery at scale, in: Proc of the 8th International Conference on Learning Analytics and Knowledge, pp. 11–15.
- [9] Dawson, S., Macfadyen, L., Risko, E.F., Foulsham, T., Kingstone, A., 2012. Using technology to encourage self-directed learning: The collaborative lecture annotation system, in: Proc of the conference of the Australian Society for Computers in Tertiary Education.
- [10] Englund, C., Olofsson, A.D., Price, L., 2017. Teaching with technology in higher education: understanding conceptual change and development in practice. Higher Education Research & Development 36, 73–87.
- [11] Glatz, E., Mavromatidis, S., Ager, B., Dimitropoulos, X., 2014. Visualizing big network traffic data using frequent pattern mining and hypergraphs. Computing 96, 27–38.
- [12] Griffin, P., Care, E., 2015. Assessment and Teaching of 21st Century Skills. Springer.
- [13] Hoang, D., Nguyen, B., Tjoa, M., 2012. Dashboard by-example: A hypergraph-based approach to on-demand data warehousing systems, in: Proc of the 2012 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1853–1858.
- [14] Khosravi, H., Cooper, K., 2018. Topic dependency models: Graph-based visual analytics for communicating assessment data. Journal of Learning Analytics 5.
- [15] Klamt, S., Haus, U.U., Theis, F., 2009. Hypergraphs and cellular networks. PLoS computational biology 5, e1000385.
- [16] Leony, D., Pardo, A., de la Fuente Valentin, L., de Castro, D.S., Kloos, C.D., 2012. Glass: a learning analytics visualization tool, in: Proc of the 2nd International Conference on Learning Analytics and Knowledge, pp. 162–163.
- [17] 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, 52–69.
- [18] Park, Y., Jo, I.H., 2015. Development of the learning analytics dashboard to support students' learning performance. Journal of Universal Computer Science 21, 110.
- [19] Rajaram, G., Karnatak, H.C., Venkatraman, S., Manjula, K.R., Krithivasan, K., 2018. A novel computational knowledge-base framework for visualization and quantification of geospatial metadata in spatial data infrastructures. GeoInformatica 22, 269–305.
- [20] Schwendimann, B.A., Rodriguez-Triana, M.J., Vozniuk, A., Prieto, L.P., Boroujeni, M.S., Holzer, A., Gillet, D., Dillenbourg, P., 2016. Perceiving learning at a glance: A systematic literature review of learning dashboard research. IEEE Transactions on Learning Technologies 10, 30–41.
- [21] Shen, T., Zhang, Z., Chen, Z., Gu, D., Liang, S., Xu, Y., Li, R., Wei, Y., Liu, Z., Yi, Y., Xie, X., 2018. A genome-scale metabolic network alignment method within a hypergraph-based framework using a rotational tensor-vector product. Scientific Reports 8.
- [22] Singh, A., Karayev, S., Gutowski, K., Abbeel, P., 2017. Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten

- work, in: Proc of the 4th Conference on Learning@ Scale, pp. 81–88.
- [23] Tervakari, A.M., Silius, K., Koro, J., Paukkeri, J., Pirttilä, O., 2014. Usefulness of information visualizations based on educational data, in: Global Engineering Education Conference, pp. 142–151.
- [24] Voloshin, V., 2009. Introduction to Graph and Hypergraph Theory. Nova Science Publishers.