Graph-based Visual Topic Dependency Models

Supporting Assessment Design and Delivery at Scale

Kendra Cooper
Independent Scholar
kendra.m.cooper@gmail.com

Hassan Khosravi
University of Queensland
h.khosravi@uq.edu.au

ABSTRACT

Educational environments continue to rapidly evolve to address the needs of diverse, growing student populations, while embracing advances in pedagogy and technology. In this changing landscape ensuring the consistency among the assessments for different offerings of a course (within or across terms), providing meaningful feedback about students' achievements, and tracking students' progression 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 visualises the required topics and their dependencies at a course level (e.g., CS 100) and assessment achievement 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. An algorithm is presented to create a TDM (static achievement for a cohort). An open-source, proof of concept implementation of the TDMs is under development; the current version is described briefly in terms of its support for visualising existing (historical, test) and synthetic data generated on demand.

CCS CONCEPTS

• **Human-centered computing** → *Visual analytics*; • **Mathematics of computing** → *Graph algorithms*; • **Social and professional topics** → *Student assessment*;

KEYWORDS

visual analytics, graph algorithms, student assessment

1 INTRODUCTION

Educational environments continue to rapidly evolve 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, and MOOCs; these are supported by a wide range of tools and techniques [8]. As the environments change assessment remains a core educational activity: the design and delivery of high quality assessments at scale becomes even more challenging.

LAK '18, March 7–9, 2018, Sydney, NSW, Australia

ACM ISBN 978-1-4503-6400-3/18/03…$15.00
https://doi.org/10.1145/3170358.3170418

The design of assessments can involve many forms of material [10]. More traditional material includes homework assignments and examinations; recently emerging material includes question repositories and games. The various forms provide interesting possibilities for covering the required topics of a course (e.g. CS 100) in a particular offering (e.g., CS 100 Term 1 2016 Section 001). However, this rich variety also introduces new challenges to educational stakeholders with respect to evaluating the coverage of assessment material and communicating achievements [10].

A wide range of educational stakeholders are involved including those in a classroom (students, instructors) and outside a classroom (course co-ordinator, course designer, program administrator, educational researcher) [3, 19]. 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 personalised feedback at scale is recognized as a difficult problem. 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 a, b and c) have been assessed. Educational researchers need to compare the achievement results between control and experimental groups.

The learning analytics community continues to actively investigate approaches that support the exploration of learning activities by different stakeholders and the challenges they face including communicating and understanding achievements, course content, and difficulty for individuals and groups of students (e.g., [15]).

With the increase in the use of educational technologies and the advancements presented by the learning analytics and educational data mining communities, the topics of visualising and tracing learners' achievements in addition to knowledge maps have received significant attention. In the field of learning analytics, "Learning Dashboards" have emerged to help make sense of educational data sets [11]. A variety of visualisations such as bar charts [4], box plots [1], radar graphs [16], and skill meters [2] have been adopted to show the achievements of students for independent (stand-alone) topics. In the field of educational data mining, learners' knowledge maps have been traced [18] and are commonly used in intelligent tutoring systems (e.g., [6]), adaptive learning environments (e.g. [17]), and recommender systems (e.g., [14]).

Beyond the communication of achievement results, the use of learning analytics for the design and delivery of assessment material seems, as yet, to be under-developed and under-explored [7] offering many research opportunities. A search of the literature reveals that investigation on combining knowledge on multiple topics has received very little attention.

In this work a collection of graph-based Topic Dependency Models (TDMs) is proposed to help reduce the research gap on the assessment analytic challenges described above. The collection addresses 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 visualise assessment data within a class and support comparisons of assessment data between classes. The third is the need for models that visualise assessment data trends over time.

The collection of models shares a common graph foundation (two-weighted graph). Stakeholders can select the classroom level model and the assessment data to use. The collection is presented using illustrative examples drawn from a high-enrolment introductory course in C programming for engineering students at a research-intensive university. The course has a hybrid educational environment that combines elements of a flipped classroom, traditional lecture and lab sessions, and a question repository, which results in a wide variety of assessment data. The course topics and dependencies span numerous topics: introduction (e.g., base conversions of integers); fundamentals (e.g., built-in data types); conditionals; loops; file I/O; arrays; and functions.

The remainder of this paper is organised as follows. Section 2 presents the underlying formal definition of the TDMs, which is based on a two-weighted graph. Section 3 introduces the TDM collection. An open-source, proof of concept implementation of the TDMs is under development [9]; the current version is described briefly in Section 4. Finally, conclusions and future work are discussed in Section 5.

2 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 with a range of stakeholders. Graphs and their visualisations are widely used to demonstrate the structure of complex data in a formal way, and they may be adopted for representing TDMs. Graphs can summarise 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. Based on the formal definition of graphs presented in [20], the underlying structure of TDMs is defined as follows:

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. An edge $e \in E$ is represented as $e = (v, w, c_1, c_2)$, where v and w are vertices (topics) being connected, c_1 represents the number of learning objects that are tagged with both topics v and w (e.g., coverage), and c_2 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.*

The use of two-weighted, undirected graphs as the formal underlying definition of the TDM allows for visualisations that have straightforward interpretations. As with any visualisation, however, training is needed to assist those not familiar with graphs to understand the representations. In the adopted visualisation (C_1) is mapped to the thickness of an edge to represents coverage and (C_2) is mapped to a colour of an edge to represent achievement.

Figure 1 illustrates an example of a topic dependency model. 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 edge thickness indicates coverage based on the number of questions, where a thicker line indicates a higher coverage. Edge colour spanning on a gradient from red to green indicates achievement, where a greener line indicates a higher achievement. For example,the edge between conditionals and loops indicates that the combination of these topics has received a high coverage in the assessment and the average class score on these questions is quite high. A self-loop on a topic (e.g., Arrays) indicates that there are questions tagged with only that topic. The structure of the graph reveals that topic Introduction is not covered in combination with any other topic.

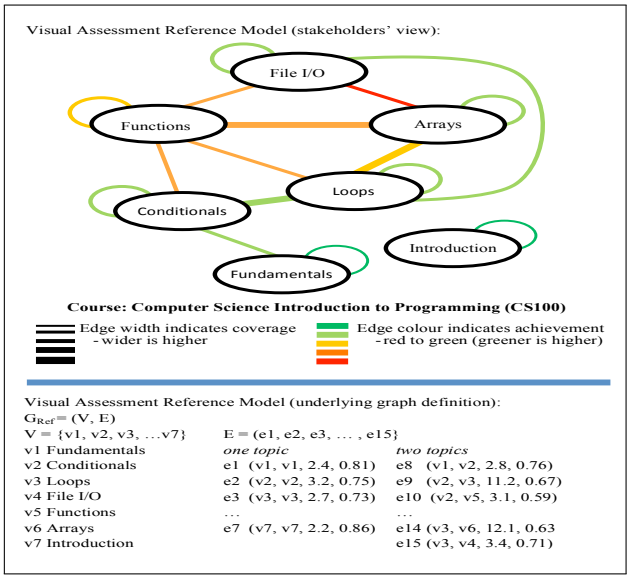

Figure 1: Example of a Topic Dependency Model

A limitation of using two-weighted graphs as the underlying definition of the model is that they are unable to visually represent the coverage of questions that are tagged with more than two topics (e.g., the graph is unable to show the coverage and achievement of questions that are tagged with a, b, and c; instead questions that are tagged with these three topics would contribute to the coverage and achievement of three edges, (a,b), (a,c) and (b,c)).

An alternative underlying graph that could be considered for defining TDMs is a two-weighted hypergraph. This graph that allow hyperedges, which can relate more than two nodes. Hypergraphs have the ability to visually present coverage of questions that are tagged with more than two topics; however, since they are

not as commonly adopted, their visualisations may not be as easy to understand. The evaluation of the trade-off between the two alternative underlying foundations is left as future work; TDMs that use two-weighted graphs are adopted in this paper.

3 THE TDM COLLECTION

The TDM collection is organized into three sub-collections: course assessment reference; static class level visualisations, and dynamic class level visualisations.

3.1 Course Assessment Reference Model

The Course Assessment Reference TDM 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. It 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.

The Course Assessment Reference TDM is illustrated in Figure 1. Assessments on required topics and their dependencies are represented with coloured edges reflecting historical coverage and achievements. For example, assessment material on the Introduction topic is required (the edge is a self-loop). The dark green colour of this edge indicates classes have historically performed very well on this topic. A number of assessments related to Functions are also required. For example, Functions, Functions and File I/O, Functions and Arrays, Functions and Loops, and Functions and Conditions are required for this course. These edges are in shades of orange, indicating students have historically performed adequately on these topics and dependencies. Assessments on File I/O and Arrays are also required for this course; this dependency is in red indicating students have historically performed poorly on this. Orange or red edges highlight on-going issues in the course, which could be explored by course designers, class co-ordinators, and instructors.

3.2 Static Class Level Visualisation Models

The static class level TDMs support visualisations at one point in time of achievement assessment data, both within a classroom and across classrooms (within a term, between terms) for comparison purposes. Achievement data from alternative sources (e.g., homework assignments, question repositories) can be selected, either for individual students or a cohort.

3.2.1 Visualising Assessment Data Within a Class. As an example, the Class Achievement TDM is illustrated in Figure 2. In the illustrated models the achievement results for Assignment #3 are presented for one student (Taylor) on the left and the entire cohort on the right. Taylor's performance on questions spanning Fundamentals and Conditionals is very good indicated by the green coloured edge. However, Taylor's performance on Conditionals is lower (orange edge) and poor on Loops, Conditionals and Loops (red edges). This visualisation can help direct Taylor to focus on additional problems covering Conditionals and Loops. The cohort, in contrast, has performed very well on average across multiple topics indicated by green edges: Conditionals; Fundamentals and Conditionals; and Loops.

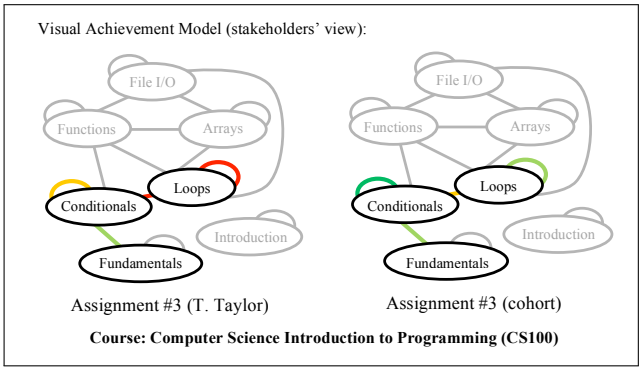

Figure 2: Visualising Assessment Achievements (Assignment #3 Individual and Cohort)

3.2.2 Visualising Assessment Data Across Classes. The Assessment Achievement TDM is illustrated in Figure 3. In the illustrated models the achievement results for Assignment #3 are presented for two classes. The assignment achievement results on the left are for a current instructor, Class 1 (Term 1 2016 Section 001). The assignment achievement results on the right are from Class 2 held the previous year (Term 1 2015 Section 002). The visualisations indicate the assessments do not cover the same topics (different nodes, edges involved). For example, Class 1's assignment includes questions on the Introduction topic and the Fundamentals topic, whereas Class 2's assignment does not. Class 1's assignment does not include questions on Loops, whereas Class 2's assignment includes questions on Loops in addition to Loops and Conditionals. Class 1 has performed adequately on the Conditionals topic (orange edge); Class 2 has performed very well (green edge). Based on the assessment coverage, Class 2 appears to be further ahead in the course compared to Class 1, as Loops are typically covered after Conditionals in introductory programming courses. This visualisation can help direct the current instructor to investigate the underlying reasons for the distinct coverage and achievements.

3.3 Dynamic Class Level Visualisation Models

The dynamic class level TDMs support visualisations of assessment data both within a classroom and across classrooms for comparison purposes, helping to identify trends over time. As with the static models, achievement data from alternative sources (e.g., homework assignments, question repositories) can be selected, either for individual students or a cohort. As an example, the Class Achievement Progression TDM is illustrated in Figure 4.

In the illustrated model achievement results on few topics are initially available (e.g., Introduction, Fundamentals); the light green colour indicates the class is performing well on the Introduction topic (green edge), but only adequately on the Fundamentals topic (orange edge). Later in the course, achievement data on additional topics are available and the visualisation indicates an improving trend on achievements in earlier topics. The last visualisation in

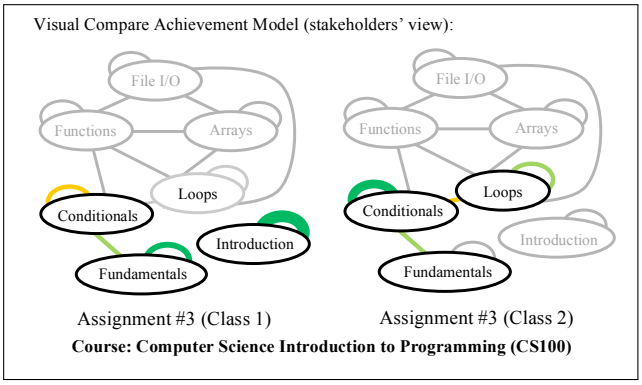

Figure 3: Visualising Assessment Data Across Classes (Comparing Achievements: Assignment #3 in Two Classes)

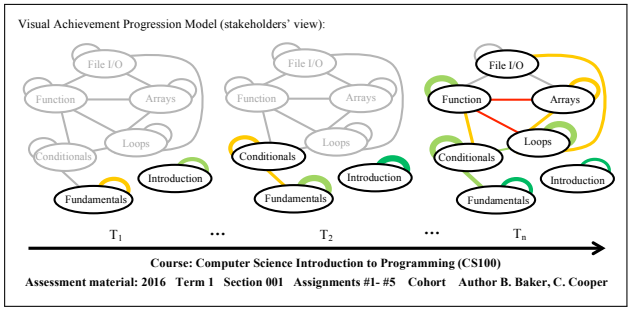

Figure 4: Visualising Class Assessment Data With Dynamic Models

time (Tn) indicates the students are performing very well on many topics (Introduction, Fundamentals, Conditionals, Loops, and Functions). The achievements for questions on Arrays, Functions and Conditionals, as well as Arrays and Loops are adequate (orange edge); however questions on Functions and Loops as well as Functions and Arrays are poor (edges in red). Some topics are easily seen to be missing (e.g., File I/O, File I/O and Functions, File I/O and Arrays) as they remain gray. Achievement data on these topics and their dependencies needs to be added, either by the instructor (e.g., designing/grading an Assignment) or students (designing, contributing, and answering questions in a peer-learning repository).

4 PROOF OF CONCEPT TOOL

An open-source tool that visualises TDMs is under development [9]. Currently, the tool provides capabilities to visualise static achievement data from existing data (historical, test) or synthetic data created on demand with the tool. It is a web-based application, developed using javascript and python; well-established libraries are utilised to help ensure high quality (e.g., pylot [12]).

4.1 Algorithm Overview

The algorithm transforms commonly available input data (student achievements/grades for specific questions and the mapping from the questions to the course topics) into a TDM (static achievement model for the cohort). The algorithms for the other TDMs in the collection are presented elsewhere due to space constraints. The algorithm consists of three high level steps: create working dictionaries and matrices; define the TDM graph elements (vertices and edges); and visualise (i.e., plot) the TDM graph. The algorithm and notation are in Algorithm 1. The implementation of this algorithm is in the file topicDependency.py in [9].

Algorithm 1 Generating a Static Achievement Model for a Cohort

Require: $SQA.\ QT$

Create Working Dictionaries and Matrices

Create dictionaries for efficient indexing

$Question\_Dict \leftarrow Create\_QDict(SQA)$

$Student\_Dict \leftarrow Create\_SDict(SQA)$

$Topic\_Tag\_Dict \leftarrow Create\_TDict(QT)$

Create T, A matrices using SQA, QT, and dictionaries

$T\_Matrix \leftarrow Create\_T(SQA, Student\_Dict, Topic\_Tag\_Dict)$

$A\_Matrix \leftarrow Create\_A(QT, Student\_Dict, Question\_Dict)$

Define the TDM Graph Elements: Vertices and Edges

$V\_List \leftarrow Define\_Vertices(Topic\_Tag\_Dict)$

$E\_List \leftarrow Define\_Edges(T\_Matrix, A\_Matrix)$

Visualise the TDM Graph

$E\_List' \leftarrow Scale\_Edge\_Values(E\_List)$

$Create\_TDM\_Plot(V\_List, E\_List')$

4.2 Synthetic Data

The tool allows users to create and visualise synthetic data sets. To create the data set the user can set the following characteristics around the student population: the number of students, the diversity in the students' backgrounds for 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 data sets. The full implementation is available within dataGenerator.py in [9]. The user interface is straightforward for this feature. Figure 5 presents a screenshot of the interface for generating synthetic data sets and visualising the results.

4.3 Real Data

The behaviour of the TDMs within the preliminary tool are explored using real data, which includes test input data files and a historical data set from a first-year programming course at a research-intensive university. The historical data set is captured in the course by using PeerWise [5], which is a free web-based system in which students can answer, rate, and discuss multiple-choice questions created by their peers. In total 377 students authored 1111 questions, assigned 1700 tags that cover 10 topics, and answered 21432 questions. The user interface is straightforward for this feature: the user selects the student question achievement and question topic input data in csv files to visualise. Figure 6 presents a screenshot of the resulting TDM.

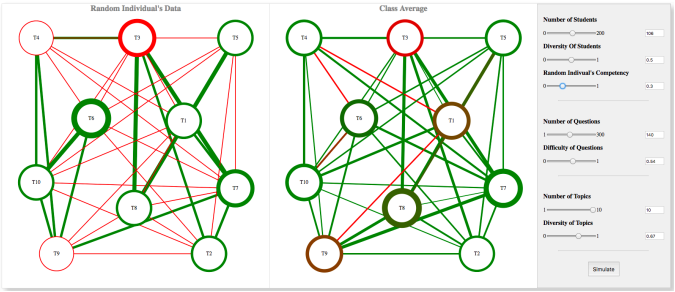

Figure 5: Screenshot of the TDM Visualisation Tool: Application to Synthetic Data

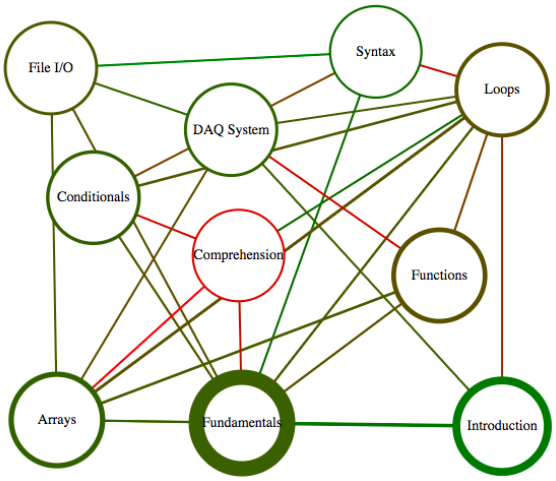

Figure 6: Screenshot of the TDM Visualisation Tool: Application to Historical Data

5 CONCLUSION 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 (within and among terms), providing meaningful feedback about students' achievements, and tracking students' progression 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 visualise assessment data within a class and support comparisons of assessment data between classes (static); the third is the need for classroom level models that visualise 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 algorithm to create one of the TDMs (static achievement for the cohort) and preliminary tool support are also reported.

In the next steps of the research a prototype is planned to implement the TDM collection as a visualisation algorithm within an educational learning dashboard called RiPPLE [13]. The dashboard will serve as a research platform to explore additional research questions such as the: scaffolding required for stakeholders to effectively use the TDM representations (including gamified material and usability studies); trade-offs in adopting alternative graph definitions (e.g., two-weighted hypergraph); impact of guidelines on identifying the topics of course material (e.g., instructor/student defined, number, prioritisation); need for additional features (e.g., identify the point in time when a threshold achievement level is achieved in a dynamic TDM); and the value in considering additional characteristics of the assessment data (e.g., difficulty of questions, levels of knowledge addressed based on a cognitive taxonomy such as Bloom's).

REFERENCES

[1] Dietrich Albert, Alexander Nussbaumer, and Christina Steiner. 2010. Towards Generic Visualisation Tools and Techniques for Adaptive E-Learning. In *Proc. 18th int. conf. Computers in Education*. 61–65.
[2] Susan Bull, Blandine Ginon, Clelia Boscolo, and Matthew Johnson. 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*. ACM, 30–39.
[3] David Carless. 2006. Differing perceptions in the feedback process. *Studies in Higher Education* 31, 2 (2006), 219–233. https://doi.org/10.1080/03075070600572132 arXiv:http://dx.doi.org/10.1080/03075070600572132
[4] Shane Dawson, Leah Macfadyen, Evan F Risko, Tom Foulsham, and Alan Kingstone. 2012. Using technology to encourage self-directed learning: The collaborative lecture annotation system. (2012).
[5] Paul Denny, John Hamer, Andrew Luxton-Reilly, and Helen Purchase. 2008. PeerWise: students sharing their multiple choice questions. In *Proceedings of the fourth international workshop on computing education research*. ACM, 51–58.
[6] Sidney DâĂŹMello, Blair Lehman, Jeremiah Sullins, Rosaire Daigle, Rebekah Combs, Kimberly Vogt, Lydia Perkins, and Art Graesser. 2010. A time for emoting: When affect-sensitivity is and isnâĂŹt effective at promoting deep learning. In *Intelligent tutoring systems*. Springer, 245–254.
[7] Cath Ellis. 2013. Broadening the scope and increasing the usefulness of learning analytics: The case for assessment analytics. *British Journal of Educational Technology* 44, 4 (2013), 662–664.
[8] Claire Englund, Anders D Olofsson, and Linda Price. 2017. Teaching with technology in higher education: understanding conceptual change and development in practice. *Higher Education Research & Development* 36, 1 (2017), 73–87.
[9] GitHub. 2017. Topic Dependency Models Tool repository. (2017). https://github.com/hkhosrav/Topic-Dependency-Models
[10] Patrick Griffin and Esther Care. 2014. *Assessment and teaching of 21st century skills: Methods and approach*. Springer.
[11] Adrian Holzer, Andrii Voznuik, Denis Gillet, Maria Jesús Rodríguez-Triana, Beat A Schwendimann, Luis P Prieto, Mina Shirvani Boroujeni, and Pierre Dillenbourg. 2016. Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies* (2016).
[12] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55
[13] Hassan Khosravi. 2017. Recommendation in Personalised Peer-Learning Environments. *arXiv preprint arXiv:1712.03077* (2017).
[14] Hassan Khosravi, Kendra Cooper, and Kirsty Kitto. 2017. RiPLE: Recommendation in Peer-Learning Environments Based on Knowledge Gaps and Interests. *JEDM-Journal of Educational Data Mining* 9, 1 (2017), 42–67.
[15] Hassan Khosravi and Kendra ML Cooper. 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*. ACM, 309–314.
[16] Madeth May, Sebastien George, and Patrick Prevot. 2011. TrAVis to enhance online tutoring and learning activities: Real–time visualization of students tracking data. *Interactive Technology and Smart Education* 8, 1 (2011), 52–69. https://doi.org/10.1108/17415651111125513
[17] Steven Oxman, William Wong, and DVX Innovations. 2014. White paper: Adaptive learning systems. *Integrated Education Solutions* (2014).
[18] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*. 505–513.
[19] Loacker G. Riordan T. 2009. *Collaborative and Systemic Assessment of Student Learning: From Principles to Practice*. Springer.
[20] Douglas West. 2000. *Introduction to Graph Theory, Second Edition*. Prentice Hall.