Toward Data-Driven Design of Educational Courses: A Feasibility Study

Rakesh Agrawal
Data Insights Laboratories
ragrawal@acm.org

Behzad Golshan
Boston University
behzad@bu.edu

Evangelos Papalexakis
Carnegie Mellon University
epapalex@cs.cmu.edu

A study plan is the choice of concepts and the organization and sequencing of the concepts to be covered in an educational course. While a good study plan is essential for the success of any course offering, the design of study plans currently remains largely a manual task. We present a novel data-driven method, which given a list of concepts can automatically propose candidate plans to cover all the concepts. Our method uses Wikipedia as an external source of knowledge to both identify which concepts should be studied together and how students should move from one group of concepts to another. For our experimental validation, we synthesize study plan for a course defined by a list of concept names from high school physics. Our user study with domain experts finds that our method is able to produce a study plan of high quality.

1. Introduction

Education is a key determinant of economic growth and prosperity (Hanushek and Woessmann, 2007; World-Bank, 1999). While the issues in devising a high-quality educational system are multi-faceted and complex, education theorists and practitioners universally recognize the importance of well-designed courses for successful outcomes (Coates, 2010; McNeil, 2014). Several factors go into designing an effective course (Davis, 2009; Fink, 2013). It starts with a clear articulation of learning objectives in terms of the knowledge and skills the students are expected to acquire from the course, which helps organize the course content and determine appropriate assessments and instructional strategies. The course needs to be student-centered and should incorporate prior knowledge and experience students bring to the course. It should also incorporate situational constraints such as the number of class meetings and the technologies available for covering the course material. Finally, the course must have a sound study plan. The study plan for a course refers to the choice of concepts to be taught and their organization and sequencing.

The course design currently is largely an art and there is a lack of automated tools to assist teachers design a course (Graves, 1996). We initiate in this paper an effort to bring computational perspective to the problem of course design. Needless to say, attempting to solve a problem of

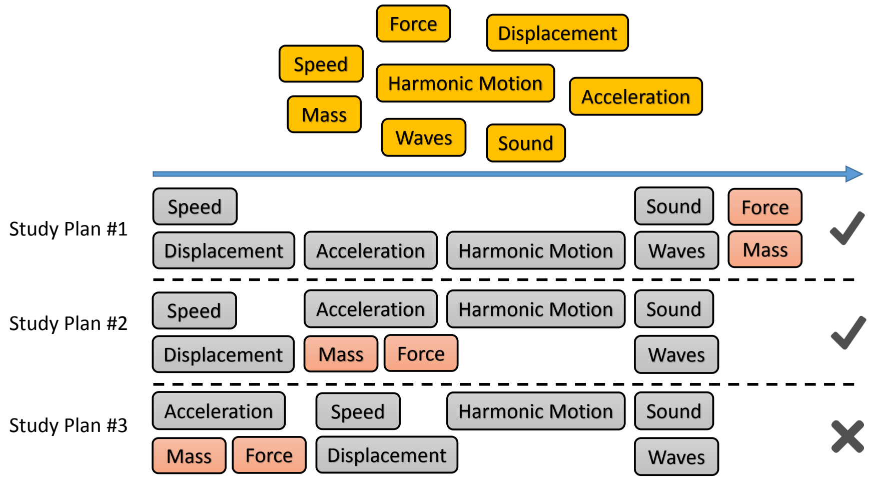

Figure 1: A set of concepts and three study plans

this magnitude in one fell swoop will be too audacious a goal. We, therefore, tackle here an important, hard subproblem - that of automating the task of designing a study plan for a course.

In order to appreciate the complexity of the problem, think of a teacher who is planning to introduce a new course. Having decided the set of concepts she would like to teach in the course, she would then need to decide the subset of concepts she should cover together because of inter-dependencies between them. Next, she would need to decide the order in which these concepts should be taught. Each of these problems is hard because of the combinatorics involved. Consider, for example, the set of concepts shown in Fig. 1. The figure also shows three different study plans. Within a study plan, we have vertically stacked the set of concepts that a teacher might decide to teach together. The first study plan starts by focusing on various concepts pertaining to motion (`Displacement`, `Speed`, and `Acceleration`), without describing how they are related to a concept like `Force`. The second plan, on the other hand, introduces the relationship between `Mass`, `Force`, and `Acceleration` as soon as possible. Both are viable plans, although what concepts get taught together and the order in which they are taught is different. The third plan is problematic though because the concept of `Acceleration` builds upon `Speed` and should not be studied before `Speed`. In general, manual design of study plans has led to some concepts being presented before all the prerequisites have been covered (Vuong et al., 2011) or some non-essential concepts being introduced too soon (Ohland et al., 2004).

We propose a data-driven approach to algorithmically synthesize study plan for a course offering that the teachers can adapt to suit their style and preferences. Although there might be many possible study plans for a given set of concepts, we define the common characteristics that make a good study plan and use them for deriving the plans. These characteristics have been abstracted from the properties of good educational material discussed extensively in the education literature (Gray and Leary, 1935; Paas et al., 2003; Pollock et al., 2002; Witte and Faigley, 1981). The input to our system, referred henceforth as *Socrates*, is the names of the concepts that the teacher intends to cover in the proposed course. These names are the identifiers for the material that teaches the corresponding concepts. The teacher may also specify if two concepts are related. If this information is not provided, Socrates can mine the relationships using Wikipedia and machine learning. The set of concepts together with relationships between them comprise the concept graph for the course. From its input, Socrates first identifies the "learning units" for the course. A learning unit is a set of related concepts that should be studied together. Socrates then outputs one or more study plans, each of which is a total order over the learning units. Learning units are ordered to respect the prerequisite relationships as much as possible (Gagne and Briggs, 1974; Scheines et al., 2014; Vuong et al., 2011). Their ordering also strives to preserve the locality of reference. That is, if a learning unit builds upon the concepts discussed in another learning unit, then this unit is arranged to appear as soon as possible after the other unit in order to avoid students having to recall concepts covered way back in the study plan.

For testing the viability of our approach, we applied Socrates end-to-end to a set of publicly available concept phrases, and generated the study plan for a course intending to teach these concepts. The study plan was generated completely automatically, without any manual intervention or guidance. It is incidental that our input consisted of a list of physics phrases; Socrates did not make any explicit use of the subject information. We validated the quality of our synthesized study plan through a user study in which we asked ten knowledgeable judges to assess its quality. The judges uniformly found the study plan to be of high quality, and definitely worth using as a strong starting point.

Our methodology is general and does not have domain or subject dependency. An assumption underlying approach is that the set of input concepts represents a cohesive body of knowledge. Another assumption is that if the user has not specified dependency relationship between the input concepts, then there are sufficiently many Wikipedia articles on the concepts that the course intends to cover. These articles are used to infer prerequisite structure between the input concepts.

1.1. Paper layout

The paper is structured as follows. We start with a discussion of related work in §2.. In §3.1., we provide certain axioms that a good study plan should comply with and then use them to formally define our problem in §3.2.. We present the techniques underlying Socrates in §3.3..

In §4., we present the results of applying Socrates to a list of concepts representing a physics course for which a study plan needs to be synthesized. We present the user study with domain experts to evaluate the quality of the study plan produced in §5.. We conclude with a summary and directions for future work in §6..

2. Related Work

To the best of our knowledge, we are the first to study the problem of automatic synthesis of study plans. However, many of the subproblems that our system needs to solve have been studied in the past. In this section, we describe past work related to ours at the overall goal level. We discuss work related to our subproblems later in the paper along with the description of our system.

Concept maps are graphical tools developed for modeling and representing a student's knowledge (Novak and Cañas, 2008). A concept map consists of concept names that are usually enclosed in circles or boxes, and the relationships between concepts that are indicated by labeled lines linking two concepts. Concept maps are manually constructed, and depending upon the elicitation technique the practitioner uses, one can arrive at very different maps (Ruiz-Primo and

Shavelson, 1996). An analysis of 43 concept maps related to the topics of electricity and magnetism has been presented in (Nousiainen and Koponen, 2010). The maps used in this analysis were collaboratively developed by students and teachers, but the final maps differed because of the differences in their understanding of the subject matter. This study illustrates how expert opinions can diverge dramatically and different experts can provide different plans for the same course (Nathan and Koedinger, 2000). It also brings home the point that asking experts to provide study plans can get prohibitively costly in time and effort (Scheines et al., 2014). Interestingly, this analysis also shows that contrary to the common belief that the concept maps in physics are hierarchical, clustering and cyclicity are equally important features.

The knowledge structure, proposed in (Falmagne and Doignon, 2011), encodes a body of information as a domain (the set of all elementary items of information), together with states of knowledge (all items mastered by some individual at some point of time), arranged in a lattice. A learning space is a special type of knowledge structure in which a student can move from one knowledge state to another by learning just one additional item. A computational procedure is provided in (Doignon, 2014) for creating learning spaces, but it is effective only for a small number of items and requires access to very many test results.

Related to learning spaces is the idea of structuring information using formal concept analysis (Ganter et al., 2005). The focus of work in (Desmarais et al., 1995) is on assessing a student's knowledge state, either through monitoring a student's behavior or through selective questioning, but not in uncovering the domain's knowledge structure. There is also recent work on estimating prerequisites in the presence of noisy data, again using test results (Brunskill, 2011). Finally, the work in (Hsieh and Wang, 2010) uses concept hierarchy to compute mutual relationships among a set of documents to decide a suitable path for navigating through them.

In contrast, we start with a set of concept names that represent in the mind of the teacher the course content she wants to teach. Given the non-availability of student observational data in this setting, we mine the Web to learn the prerequisite relationships between the concepts. We note that much of the existing work on inferring prerequisite relationships makes use of the data on differential student performance on tests and exercises or data on student interactions with educational material and tutoring platforms (e.g. (Chang et al., 2014; Pavlik et al., 2008; Scheines et al., 2014; Vuong et al., 2011)). Our concept graph, thus formed, can be viewed as a special form of concept map that includes constraints on the admissible learning paths for the course. The structure of our study plans is quite different from a learning space. Our students do not move from mastering one concept to next. Rather, we cluster closely-related concepts into learning units and the students advance from studying a learning unit to another, albeit we do also provide how the teacher can organize a learning unit.

3. Models and Algorithms

3.1. Axioms

We endeavour to define the common characteristics of good study plans. To do so, we first introduce the notion of learning units, the building blocks of the study plans.

3.1.1. Learning units

A learning unit can be thought of as a group of concepts that are closely related, and thus, they should be discussed and covered together. A well-designed study plan aims to organize the set of concepts in a course into cohesive learning units such that they are independent from each other to the extent possible. More precisely, a proper grouping of concepts into learning units follows the following axioms:

- **Cohesion:** Each learning unit must consist of concepts that are closely related. For instance, concepts like `Speed` and `Displacement` are tightly connected while concepts like `Speed` and `Magnet` are not.
- **Isolation:** Concepts that belong to different learning units must be independent as much as possible.
- **Unity:** It is inevitable that a single concept may appear in multiple learning units. For instance, a fundamental concept such as `Force` is related to both `Acceleration` and `Electromagnetic force` and may belong to both a learning unit on "Basic Law's of Motion" and a learning unit on "Electromagnetism". However, it is still desirable to discuss each concept in a single unit to the extent possible.

These axioms have been abstracted from the properties of good educational material, discussed extensively in the education literature (Gray and Leary, 1935; Paas et al., 2003; Pollock et al., 2002; Witte and Faigley, 1981).

3.1.2. Ordering learning units

Once a desirable set of learning units is identified, a study plan can be formed by ordering these learning units. One can consider two main criteria for defining a good order:

- **Difficulty:** This criterion aims to order the learning units based on their level of difficulty. The idea is that students should start with easier concepts and progress to more and more complex concepts (Falmagne and Doignon, 2011; Fink, 2013; Wauters et al., 2011).
- **Prerequisites:** An alternative approach to ordering learning units is to make sure that the concepts that are prerequisites for other concepts appear earlier in the study plan (Gagne and Briggs, 1974; Scheines et al., 2014; Vuong et al., 2011).

We mainly focus on the prerequisites criteria since the difficulty levels of concepts are subjective, while there is little disagreement over respecting the prerequisite relationships. We further refine the prerequisites criteria using the following axioms:

- **Prerequisite compliance:** Learning units must be ordered to respect the prerequisite relationships as much as possible.
- **Locality of references:** If learning unit B builds upon the concepts discussed in learning unit A, then B should be ordered to appear soon after A. Otherwise, students will need to recall concepts that were covered way back in the study plan, increasing comprehension burden.

3.2. Problem Definition

Based on the axioms just stated, we can define our problem in a graph-theoretic setting. Our problem definition postulates as input a *Concept Graph*, which is described next.

Concept Graph: Consider a directed graph $G = \langle V, E\rangle$, where each node represents a concept name and each directed edge (u, v) represents that concept u is a prerequisite of v. Note that if two concepts u and v are mutually dependent, then both (u, v) and (v, u) are included in G.

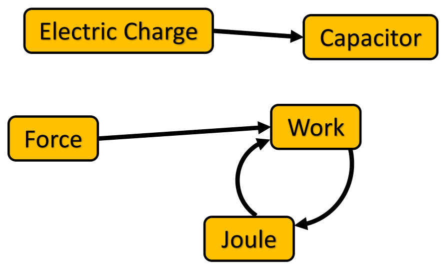

Figure 2: A sample concept graph

Figure 2 shows a sample concept graph. In this graph, `Electric Charge` is a prerequisite for `Capacitor`, whereas `Work` and `Joule` are mutually dependent. On the other hand, `Electric Charge` and `Force` are unrelated.

Problem Statement: We can now formally define the problem of creating study plans from a given concept graph as follows.

Problem (Study Plan Design Problem)**.** *Given a concept graph $G = \langle V, E \rangle$ with $n > 0$ nodes, and the number of desired learning units m ($m \leq n$), output an ordered vector of learning units $\mathcal{L} = \langle L_1, L_2, \cdots, L_m \rangle$ to*

$$\begin{aligned} \textit{Minimize: } & f(\mathcal{L}) \\ \textit{s.t. } & \forall i : L_i \subseteq V, L_i \neq \phi, \textit{and} \\ & \cup_i L_i = V. \end{aligned}$$

We still need to define the objective function $f(\mathcal{L})$ to complete this formulation. For each node $v \subset V$, let us use $\pi(v)$ to refer to the index of the first learning unit in which concept v appears. Using this notation, we define $f(\mathcal{L})$ as follows.

$$\begin{aligned} f(\mathcal{L}) = & \sum_{\substack{\pi(u)<\pi(v)\\(u,v)\in E}} (\pi(v) - \pi(u)) * C_r, \\ & + \sum_{\substack{\pi(u)>\pi(v)\\(u,v)\in E}} (\pi(u) - \pi(v)) * C_p + \\ & + \sum_{\substack{\pi(u)=\pi(v)\\(u,v)\notin E}} C_d. \end{aligned} \tag{1}$$

Explanation: Each term in Eq. 1 penalizes the violation of some of the axioms, and the values C_r, C_p, and C_d are simply the costs associated with each type of violation.

The first term measures to what extent the locality of references axiom is violated. More precisely, if concept u is a prerequisite of v then while studying concept v, students have to recall a concept that was covered $\pi(v) - \pi(u)$ units earlier, and the value C_r is the cost of recalling an earlier concept.

The second term in Eq. 1 captures the violation of the prerequisite compliance axiom. That is, when concept v is covered earlier compared to its prerequisite u, a cost C_p is associated to the gap $\pi(u)-\pi(v)$, which represents how long it would take the students to learn about prerequisite concept u.

Note that the first two terms only focus on how well the learning units are ordered. Unlike the first two terms, the final term evaluates the quality of the learning units. If two concepts u and v are placed together in a learning unit without an edge connecting them, then this violates the cohesion axiom, which is penalized with the cost C_d. Note that since the total number of edges in the graph is constant, optimizing for cohesion would also optimize for the isolation axiom.

Finally, observe that none of these terms directly optimizes for the unity axiom. But it is clear that placing a concept in multiple learning units would create more edges between learning units. Therefore, the violation of the unity axiom is naturally penalized by the first two terms.

3.2.1. Computational complexity

It is easy to show that the problem of minimizing the proposed objective function $f(\mathcal{L})$ is NP-hard. Assume that the optimal ordering of learning units is ignored, (i.e., when C_r and C_p are set to zero). In this case, the objective function simply counts the number of missing edges in each partition (see the last term in Eq. 1). Now, if one considers each edge to represent an agreement, and each missing edge to be a disagreement between its endpoint, then the objective becomes minimizing the number of disagreements in each partition. This problem is a variation of the well-known correlation-clustering problem, and it is known to be NP-hard (Bansal et al., 2002).

Furthermore, our problem is closely related to another NP-hard combinatorial problem, called the *Minimum Linear Arrangement Problem* (MinLA) (Silvestre, 1998), which we will discuss in more detail momentarily.

3.3. Our Method

Given a concept graph that captures the prerequisite relationships among the concept, our method generates a study plan by applying the following three steps: (1) Finding learning units, (2) Ordering learning units, and (3) Organizing learning units. We discuss in §4. how the concept graphs can be automatically obtained by mining the relationship between concepts in the Wikipedia webgraph if the concept graph was not available. For now, assume that the concept graph is provided as part of the input. Next, we provide a detailed description of the three steps of our method.

3.3.1. Finding learning units

The problem of finding cohesive learning units can be viewed as a community detection problem, for which many algorithms exist in the literature (Fortunato, 2010). In particular, the algorithm of Reichardt and Bornholdt (Reichardt and Bornholdt, 2006) works quite well for our task. An efficient implementation of this algorithm, referred to as the Spin-glass algorithm, is available in the igraph package of the R statistical software.[1]

The Spin-glass algorithm requires two input parameters. The first parameter is an upper bound on the number of communities that the algorithm can produce. We tune this parameter by

[1] `inside-r.org/packages/cran/igraph/docs/spinglass.community`

looking at the stability of the reported results. Spin-glass is a randomized algorithm which may return a different set of learning units in each trial. However, if the underlying datasets consists of well-separated coherent units, the algorithm would consistently report the same units. We exploit this fact to tune the number of desired learning units. More precisely, we select the largest number of units for which Spin-glass returns stable results. The second parameter determines the cost of having missing edges in a cluster, which in our formulation is captured by the cost parameter C_d (see Equation 1). We have experimentally observed that setting $C_d = 1.3$ yields the most coherent learning units.

3.3.2. Ordering learning units

This step finds an optimal ordering of the learning units computed in the previous step. The goal is to arrange the learning units in an ordered vector $\mathcal{L}$, such that it would minimize the objective function $f(\mathcal{L})$ described in Eq. 1.

Given a particular ordering of learning units $\mathcal{L}$, we call an edge (u, v) a forward edge if the learning unit that covers concept v is placed after the learning unit containing concept u (i.e., if $\pi(u) < \pi(v)$). Similarly, we call an edge (u, v) a backward edge if $\pi(u) > \pi(v)$. Finally, we define the length of an edge (u, v) to be $|\pi(v) - \pi(u)|$. Using this notation, it is easy to see that the objective function $f(\mathcal{L})$ penalizes the length of forward and backward edges by cost C_r and C_p, respectively.

Unfortunately, as the following theorem shows, finding an optimal ordering of learning units is NP-hard.

Theorem 1. *For a given concept graph $G = \langle V, E\rangle$ and a set of learning units $\{L_1, L_2, \cdots, L_m\}$, the problem of finding an ordering $\mathcal{L}$ of the learning units that minimizes $f(\mathcal{L})$ is NP-hard.*

Proof. If we set the cost of both backward and forward edges to the same value (i.e., if $C_r = C_p$), then this problem becomes an instance of Minimum Linear Arrangement (MinLA) problem that is known to be NP-hard (Silvestre, 1998). The MinLA problem is the problem of ordering the nodes of a graph such the total length of the edges in the arrangement is minimized. $\square$

Although finding the optimal ordering is NP-hard, the number of desired learning units in our application is usually small. Thus, one can exhaustively examine all possible ordering of learning units to find the optimal solution. We also note that many practical approximation algorithms for solving the MinLA problem have been studied in the literature. Especially, the simulated-annealing technique described in (Silvestre, 1998) can be used to solve our problem. One only needs to specify the costs C_r and C_p in the objective function, and apply the same optimization method.

In practice, we set the cost parameters as follows: $C_r = 10$ and $C_p = 1$. These values may seem arbitrary, but (a) they clearly emphasize that violating prerequisite relationships is undesirable, and (b) we experimentally observe that as long as there is a significant gap between C_r and C_p, we obtain the same ordering of learning units. In other words, these parameters do not require fine-tuning as long as $C_r \gg C_p$.

3.3.3. Organizing learning units

At this point, our method has created a solution to the study plan design problem. However, our method tries to further organize the concepts within learning units to provide user with a better visualization of the structure of a learning unit. This visualization highlights which concepts build upon other concepts in the learning unit by arranging concepts into a hierarchy.

For a learning unit L_j, first form the induced subgraph G_j, defined by the concepts inside L_j. Now find the strongly connected components[2] of G_j and contract each of them into a single node to obtain the condensed graph G_j^C. Note when a group of concepts form a strongly-connected component, they are all considered to be prerequisites of each other (called sometimes, corequisites). Such concepts should be discussed and studied concurrently.

Next, compute the transitive reduction G_j^T of G_j^C in order to remove redundant edges (Aho et al., 1972). Since G_j^C is a DAG, G_j^T will be unique. Finally, apply a topological sort to G_j^T to obtain a sequential ordering of the strongly connected components. Clearly, the outcome of the topological sort may not be unique. Thus, this step may allow additional flexibility in organizing a learning unit.

Figure 3 shows a sample learning unit for which the strongly-connected components have been identified and organized sequentially.

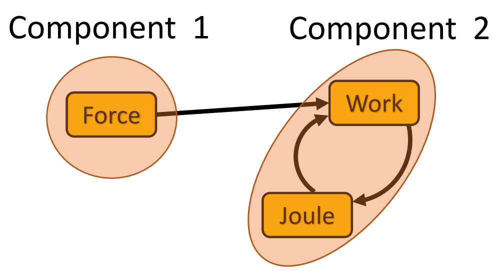

Figure 3: Organizing concepts within a learning unit

4. Test Driving the Methodology

In order to field test our methodology, we applied Socrates to a list of concepts representing a physics course for which a teacher might want to design a study plan. We selected physics because of interest in the education community in applying computing to this field (Koponen and Pehkonen, 2010; Nousiainen and Koponen, 2010).

4.1. The physics dataset

CK12.org is a non-profit organization, dedicated to promoting public-domain educational content. At their website[3], they maintain subject-wise lists of concepts that a high-school student is expected to know by the time of graduation.

The list for physics contains 193 concept names. We provide this list as input to Socrates and ask it to synthesize study plan for a course aimed at teaching these concepts. Note that "Wikipedia's Glossary of Physics", which includes all Wikipedia articles related to physics,

[2] A strongly connected component of a directed graph is a maximal subgraph where there exist a path between any pairs of nodes.

[3] www.ck12.org

consists of 425 articles. So it is reasonable to have 200 concepts in a high-school level physics course.

We emphasize that it is incidental that our input is a list of physics phrases - Socrates did not make any explicit use of this subject information. It only helped when deciding who to recruit for our user study.

4.2. Creating the concept graph

Note that our dataset contains concept names, but not relationships between them. Here, we discuss how one can create a concept graph from a given list of concept names by mining Wikipedia to infer prerequisite relationships. Wikipedia has been used as a knowledge source to infer semantic relatedness in several studies in the past (Medelyan et al., 2009; Ponzetto and Strube, 2007; Suchanek et al., 2008). We describe below how exactly Socrates forms the concept graph, although it is easy to conceive alternative approaches, including employing techniques from the education literature based on tests and user navigational over the course content (Brunskill, 2011; Falmagne and Doignon, 2011; Vuong et al., 2011).

The concept graph is formed in two steps: (1) Creating a webgraph, and (2) Correcting the edges in the webgraph so obtained to yield the concept graph.

4.2.1. Creating a webgraph

We first use web search to map each concept name to a Wikipedia article. More specifically, we use the concept name as the query, constraining the search results to originate from the Wikipedia site. We treat the first search result to be the Wikipedia article that best corresponds to the queried concept. For instance, the top Wikipedia page for concept `Vector Addition`is titled "Euclidean Vector" which we consider as the corresponding Wikipedia article for the concept despite the slight mismatch. Once all the Wikipedia articles are found corresponding to all of the input concepts, we extract the webgraph defined by the retrieved articles. That is a graph in which nodes are the articles, and each edge (u, v) indicates that article u has a hyperlink to article v.

In some cases, multiple concept names map to the same Wikipedia article (e.g., both `Motion and force` and `Types of forces` are mostly discussed in the Wikipedia article titled "Force (Physics)"). For our dataset, we end up with 155 Wikipedia articles after this mapping. When more than one concepts are mapped to the same Wikipedia article, we treat this article as the representative for all of them. Consequently, when such a node in the webgraph is assigned to a learning unit, we consider all the associated concepts to be part of that learning unit.

Amongst this set of Wikipedia articles, 16 had no hyperlink to other articles in the set. These singleton nodes arise due to the following reasons. Firstly, some of the phrases listed on the CK12 website are too generic and cannot be tied to any of the articles. For instance, "Problem Solving" is a concept phrase in the CK12 dataset for which a Wikipedia article exists, but it is easy to see why such an article has no links to any article on a physics concept. Another reason is that Wikipedia can have multiple articles on a single concept. For instance, concept `Force` has a regular Wikipedia page, but there exist another page on `Force` that aims to disambiguate if the term "force" is being discussed in the area of physics or politics. Also, one can find another page on `Force` which is written in what Wikipedia calls the *simple* language. Thus, a link from one article might be missing simply because it is hyperlinked to a different article, which is not in our set.

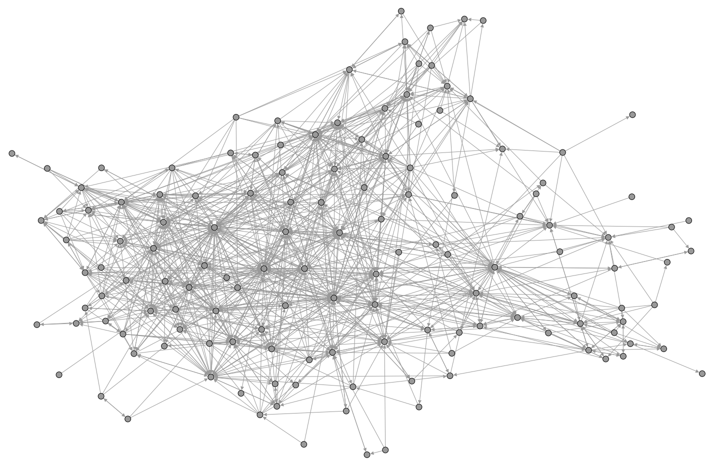

Figure 4: The CK12 physics webgraph

The singleton nodes are not helpful since the community detection algorithms rely on edges to classify the nodes. After removing such nodes, we end up with 139 concepts (and 1009 edges between them), which were then used by Socrates to create the study plan. Table 1 contains these concepts and Fig. 4 shows the webgraph for them.

4.2.2. Correcting edges

The webgraph obtained above should not be directly used as the concept graph. For instance, the Wikipedia article on `Capacitor` has hyperlinks to articles on `Joule` and `Laser`. Arguably, `Joule` should be considered a prerequisite for studying the concept of `Capacitor`. So, the direction of the edge between `Capacitor` and `Joule` should be reversed. Similarly, since the concepts `Capacitor` and `Laser` are only remotely related, an edge between them should not be included in the concept graph.

Despite the differences between the webgraph and the concept graph, we observe that the edges in the webgraph are a superset of the edges in the concept graph. So, we use a classification model that tries to identify which edges should remain in the concept graph, and what their direction should be. Our model uses a set of graphical features (e.g., degree of the node) as well as features extracted from the Wikipedia article (e.g., the number of languages the article is available in). A supervised learning approach for inferring dependency relationship between two Wikipedia pages has also been presented in (Talukdar and Cohen, 2012).

For obtaining the concept graph from the webgraph, we need to classify each edge (u, v) in the webgraph into one of the following categories:

1. **unrelated:** Indicates that there is no prerequisite relationship between concepts u and v. In other words, this edge should not exist in the concept graph.

2. **prerequisite:** Indicates that concept u is indeed a prerequisite of v. Thus, this edge must be included in the concept graph.

3. **reverse:** Indicates that concept v is a prerequisite of u. Thus, the opposite edge (i.e., (v, u)) should be included in the concept graph.

To perform this classification, we define a set of features for every edge in the concept

Actual Edge Type \ Prediction Outcome	prereq.	unrelated	reverse
prereq.	29	142	1
unrelated	9	668	8
reverse	0	110	42

Figure 5: The confusion matrix for edge type prediction

graph. The following is a short description of these features and why they are considered to have predictive power for our classification task:

- **In-degree:** For each edge (u, v), the in-degree of both endpoints u and v are considered as a feature. It is more likely for fundamental concepts to have higher in-degrees since many other concepts build upon them and have to cite them on their Wiki page.
- **Out-degree:** This can be considered as the opposite of the previous feature. Basically, a node with a high out-degree is likely to be an advanced concept.
- **Number of languages:** For each page on Wikipedia, we can find the number of languages in which the article is available. We have observed that fundamental concepts are usually available in more languages than advanced concepts.
- **Number of categories:** Wikipedia also provides a set of categories for each article. For instance, the article for the concept `Velocity` belongs to "Motion", "Kinematics", "Velocity", and "Concept in physics" categories. Intuitively, fundamental concepts belong to fewer categories.
- **Common neighbors:** For every edge (u, v), this feature counts the number of neighbors the nodes u and v have in common. This feature helps the model identify to what extent the two concepts are related. We also include a normalized version of this feature by computing the Jaccard similarity between the neighbors of u and v.
- **Common categories:** For every edge (u, v), this feature counts the number of categories that the nodes u and v have in common. Similar to the previous case, we also include a normalized version of this feature by using the Jaccard similarity.

For assessing the effectiveness of these features, we first created a ground-truth dataset by manually labeling all edges. That is, for each pair of concepts, we decided whether one is a prerequisite of the other or not. Then we used these features to build a decision-forest classifier (Ho, 1998), and did 10-fold cross validation to measure the accuracy. The learned model obtained a precision of 0.74 and recall of 0.73. Figure 5 shows the confusion matrix.

Clearly, the classification model is not perfect in predicting the prerequisite relationships. But there is only one prerequisite edge that is incorrectly classified as a reverse edge and there is no reverse edge that is incorrectly classified as a prerequisite edge. In other words, the model has a tendency to over-classify edges as unrelated, and thus some edges will be erroneously missing from the concept graph. However, for other edges, the model quite accurately predicts which concept is a prerequisite of the other.

This implies that in practice, we will end up with an accurate concept graph, but which is sparser compared to the ideal. Figure 6 shows the ideal and computed graphs. The number of edges in the webgraph, the ideal concept graph and the computed concept graph are 1009, 324 and 94 respectively.

(a) Ideal concept graph (b) Computed concept graph

Figure 6: The ideal and computed concept graph for the CK12 dataset

Fortunately, using the sparser graph for ordering learning units yields the same result as ideal. The reason is that normally the number of learning units is small and relatively few correct prerequisite edges are needed to rigorously order them. For instance, there are only seven learning units in our dataset. Out of the 94 edges in the computed concept graph, 66 turn out to be across learning units which is sufficient to provide a rigorous ordering of the seven learning units.

4.3. Study plan

We next present the generated study plan. Table 1 shows the seven computed learning units and their optimal ordering (encoded in the names of the learning units).[4] Each column of the table shows all the concepts from the corresponding learning unit in the alphabetical order. We observe the following:

- Each learning unit mostly covers concepts related to a specific area in Physics. We can associate the seven learning units created by Socrates with the following areas, respectively: "Classical mechanics", "Kinematics", "Atomic models", "Thermodynamics", "Electromagnetism", "Light and waves", and "Theory of relativity".
- Early units in the order cover concepts from classical physics (e.g., `Force` and `Velocity`), while the units toward the end cover advanced concepts (e.g., `diffraction grating` and `theory of relativity`).

[4] The spinglass community detection algorithm we use for creating the learning units is a randomized algorithm, which implies that multiple runs of the algorithm can create different results. For our dataset, seven was the maximum number of units that could be created while maintaining the stability of the units across multiple runs.

Unit 1(20 concepts)	**Unit 2** (21 concepts)	**Unit 3** (14 concepts)	**Unit 4** (18 concepts)
buoyancy	acceleration	atom	calorimetry
euclidean vector	angular momentum	bohr model	change of state
force	angular velocity	conservation of energy	combined gas law
free body diagram	centripetal force	elastic collision	conversion of units
friction	circular motion	inelastic collision	double-slit experiment
impulse	displacement	kinetic energy	energy
inclined plane	keplers laws of ...	mass versus weight	energy conversion
mechanical advantage	kinematics	momentum	fluid pressure
oil-drop experiment	moment of inertia	nuclear reaction	heat engine
physics	motion	nuclear reactor	heat transfer
quark	motion graphs and ...	potential energy	measurement
range of a projectile	newtons' laws of ...	radiation	metric unit
rolling	orbital motion	radiocarbon dating	pressure
scalar	pendulum	spring	quantum mechanics
scientific method	projectile motion		specific heat capacity
simple machine	rotation around a ...		specific latent heat
torque	simple harmonic motion		temperature
trajectory of a projectile	uniform acceleration		universal gas law
wheel and axle	vector addition		
work	velocity		
	weightlessness		
Unit 5 (27 concepts)	**Unit 6** (28 concepts)	**Unit 7** (11 concepts)	
ammeter	beat	doppler effect	
capacitor	color	general relativity	
capacitors in circuits	concave lens	half-life	
electric charge	conduction	length contraction	
electric current	curved mirror	mathematical physics	
electric field	diffraction	newtons law of univer...	
electric power	diffraction grating	rc time constant	
electrical efficiency	double convex lens	special relativity	
electrical network	electromagnetic radiation	speed of light	
electrical resistor	electromagnetic spectrum	theory of relativity	
electricity and magnetism	laser	time dilation	
electromagnetic induc...	lens		
electromagnetism	light		
electromotive force	longitudinal wave		
electrostatics	loudness		
history of electric motor	mechanical wave		
internal resistance	mirror		
lorentz force	plane mirror		
magnet	reflection		
magnetic field	refraction		
motor-generator	sound		
power	standing wave		
resistors in series	total internal reflection		
series and parallel cir...	transverse wave		
transformer	wave		
tube sound	wave equation		
voltage	wave interference		
	wave speed		

Table 1: The study plan synthesized by Socrates for the CK12 physics dataset. Each column of the table corresponds to one learning unit and lists the included concepts in the alphabetical order. The ordering of the learning units is encoded in their names. Thus, Unit 1 needs to be studied first and Unit 7 last.

- The learning units are largely cohesive, though they each contain few concepts that are out of place (e.g., Unit 1 includes `Oil drop experiment` that could fit better in Unit 5 related to electricity). Some units (e.g., Unit 7) are more cohesive than others.

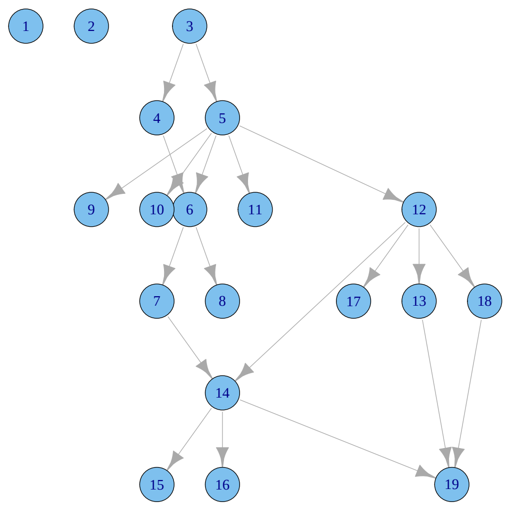

Figure 7: Structure of Unit 2

ID	Concepts	ID	Concepts
1	weightlessness	11	simple harmonic motion
2	projectile motion	12	acceleration
3	motion	13	centripetal force
4	moment of inertia	14	newton's laws of motion
5	vector addition	15	kepler's laws of ...
	displacement	16	orbital motion
	velocity	17	motion graphs
6	angular velocity	18	uniform acceleration
7	angular momentum	19	circular motion
8	rotation around a ...		
9	kinematics		
10	pendulum		

Table 2: Mapping between node numbers in Fig. 7 and the set of corresponding concepts

- The learning units are not balanced in number of concepts they contain. This outcome is not surprising as we did not include such a balance in our objective.

Overall, we were pleasantly surprised by the quality of the study plan, particularly since the only input given to Socrates was a list of phrases and the rest of the process was completely automated. We validate these observations through a user study later in the paper.

4.4. Organization within learning units

Recall Socrates also provides the organization of concepts within a learning unit. Figure 7 shows the structure for Unit 2. We selected Unit 2 for this illustration because this unit has the richest interconnection between the concepts.

Every node in the figure corresponds to a strongly connected component (SCC) in the concept graph of Unit 2. Table 4.3. shows which concepts are associated with each of the nodes. Most SCCs have one concept, except SCC 5, which has three concepts. SCCs 1 and 2 have no prerequisite relationship with each other or any other SCC in Unit 2. This can happen because the edges in the concept graph are a subset of edges in the webgraph.

We see that SCCs 3, 4, and 5 at the top correspond to basic concepts such as `Motion`, `Moment of inertia`, `Displacement`, `Velocity`, and `Vector Addition`. On the other hand, the SCCs at the bottom of the hierarchy correspond to concepts concerning advanced forms of motion, namely `Circular motion`, `Orbital motion`, and `Kepler's law of planetary motion`.

While the suggested ordering for most concepts appears reasonable, there are few concepts for which the ordering can be debated. For instance, consider SCC 9 which is about `Kinematics`. This term describes the area that is being studied in this learning unit. Thus, it might make sense to move this concept to the beginning of this unit. On the other hand, it is expected to see `Kinematics` appearing after `Motion` in the middle of the ordering as most people would agree that one should know what `Motion` is before the person can learn about `Kinematics`.

5. User Study

To quantify the quality of the study plan produced by Socrates, we first considered comparing the plan against an existing plan. Unfortunately, there is no standard plan to compare against - different education systems follow different plans. The CK12 website also did not provide a study plan and translating textbooks back to study plans is a non-trivial, open problem.

We, therefore, set up a user study. The goal of the study is twofold: 1) evaluate the cohesion of each learning unit, and 2) evaluate the ordering of the units.

5.1. Study Participants

We recruited ten volunteers to participate in our study. Our pool of volunteers has a mix of degree of specialization in physics, ranging from high school/college education to Ph.D. level education to professional experience teaching physics.

One in our pool is a high school physics teacher (group 1), one is a physics graduate student (group 2), two are computer science graduate students with physics major and minor respectively in their undergraduate studies (group 3), two are mechanical engineering graduate students (group 4), and four have a purely computer science background (three graduate students and one professor) (group 5). None of the authors provided judgments.

5.2. Survey design

We asked the following two questions on the study plan given in Table 1:

Q1: Please count the number of *odd concepts* that you feel do not belong to a particular unit. If all concepts in a unit look related, please answer 0.

Q2: We have given you a specific ordering of the units. Would you change anything in the ordering? If so, please write the new ordering using the numbers that we have already assigned to the units (e.g. if units 1 and 2 should be swapped, the new ordering would be $2, 1, 3, 4, 5, 6, 7$). If you would not change the ordering, please respond with ”No”.

After some initial testing, we determined that these questions were a good compromise between the required time to complete the survey and the amount of information they provide with respect to the effectiveness of our techniques.

5.3. Quality of judgments

It is important to have an idea of the inter-judge agreement in a survey like ours. We present the result of our analysis with respect to the responses to Q1. An appropriate way of analyzing the agreement is by using Krippendorff’s α statistical measure (Krippendorff, 1970), which is applicable to the current scenario of judges assigning a value to a specific variable.

The overall agreement measured by Krippendorff’s α for our ten judges turns out to be 0.29. This indicates that there is fair but imperfect agreement. When we look at the agreement of judges within the same background group, the agreement between judges with computer science background (group 5) and judges with mechanical engineering background (group 4) is 0.25 and 0.22 respectively. However, the agreement between judges of group 3 (computer science graduate students with major and minor in physics) is higher and has a value of 0.53, and the agreement between judges of groups 4 and 5 (who have professional specialization in physics) is 0.49.

In addition to agreement between judges, it is also important to identify potentially harsh and lenient judges. We, therefore, conduct an analysis of variance (ANOVA) for the ratings of each judge. The box-plot of Fig. 8 shows the results. We see that the ratings provided by the judges do not differ significantly from each other and we cannot statistically pull apart different judges in terms of harshness. There are, however, harsh outliers such as seven odd concepts for Unit 3 as given by a particular judge.

5.4. Results for Q1

Table 3 summarizes the results of the survey for the first question. More specifically, it shows the minimum, maximum, mean, and the median of values that our participants reported as the number of odd concepts in each learning unit.

Table 3: Survey results for Q1 (number of odd concepts)

	Min	**Max**	**Median**	**Mean**	**# Concepts**
Unit 1	1	6	3.0	3.4	20
Unit 2	0	3	1.0	1.1	20
Unit 3	1	7	3.5	3.7	14
Unit 4	0	5	2.0	1.8	18
Unit 5	0	4	1.0	1.0	26
Unit 6	0	3	0.5	0.9	28
Unit 7	0	5	1.0	1.4	11

In general, we observe that for the majority of learning units, the number of odd concepts is very low, demonstrating good cohesion in the units. Moreover, one of the participants (in group 5) found six odd concepts in Unit 1, and another (in group 4) found seven odd concepts in Unit 3, raising the respective averages rather disproportionately. With respect to Unit 1, the judge who found six odd concepts commented that he was not sure of the main subject of the unit, and trying to guess it, he came up with this rating. Looking at Unit 1, we see that there exist some quite general concepts such as `physics` and `euclidean vector` that understandably can distract the focus. We drill down further into Unit 3 in §5.6..

Overall, all judges expressed their high appreciation of the cohesion of the learning units, especially when informed that these learning units were produced algorithmically without manual intervention.

5.5. Results for Q2

With respect to the ordering of the learning units, Socrates proves to be quite robust. In particular, all but two of the participants were satisfied with the given ordering. One participant that belongs to group 3 (computer science graduate student with physics undergraduate major) and one that belongs to group 2 (physics graduate student), suggested orderings $1, 2, 6, 5, 3, 4, 7$ and $1, 2, 3, 4, 6, 5, 7$. The common thread between the two suggestions is that Unit 6 should be placed before Unit 5. It is important to note though, as also observed by one of them, these ordering variations are biased by the particular order in which the judges were first taught these concepts/units.

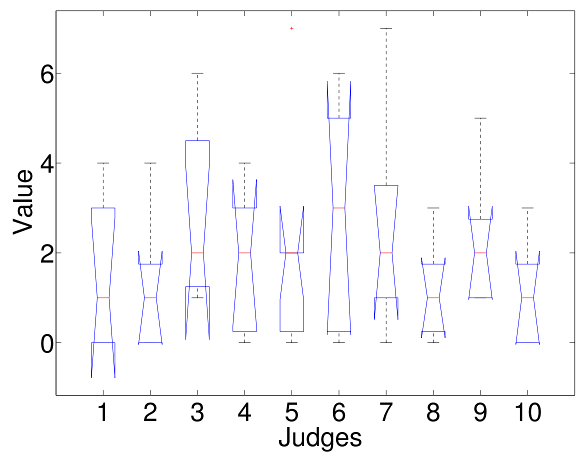

Figure 8: ANOVA results per judge. In the box-plot, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually (per Matlab documentation)

5.6. Comments by participants

Apart from questions Q1 and Q2, the two judges of group 3 voluntarily gave us detailed comments on the structure of the learning units, and in particular regarding Unit 3. Both, independently, observed that Unit 3 contains two fairly cohesive units and should be separated into two units: one on energy and the other on nuclear/particle physics. This phenomenon of cohesive units within the generated learning units is possibly an artifact of the small number of units.

Our automatically generated study plans are intended to provide powerful starting points that the teachers can improve and adapt to suit their style and preferences. Socrates' usefulness in that respect is also affirmed by our user study: anecdotally, judge #10 who is a high school physics teacher was very satisfied by the resulting units and described them as "very clever".

6. Discussion and Future Work

Our major contributions in this work are as follows:

- We introduced the problem of algorithmically synthesizing study plan for an educational course offering from the sole input of concept phrases representing the course content. Given the paucity of automated tools to provide assistance to teachers in course planning and the likely rapid increase in new course offerings in the emerging education milieu (Coates, 2010), this problem has large societal significance.

- We provided a novel, yet pragmatic, solution to the problem. In the course of developing this solution, we abstracted and rigorously defined the problem of study plan synthesis as a combinatorial optimization problem and studied its complexity. This problem definition is likely to be of independent interest to data mining researchers and practitioners. Under mild assumptions, our solution methodology is quite general and does not have domain or subject dependency.

- We tested our solution using a set of publicly available concept phrases, and generating a study plan for a course intending to teach these concepts. The study plan was generated completely automatically, without any manual intervention or guidance. It is incidental that our input consisted of a list of physics phrases; our system did not make any explicit use of the subject information. It only helped when recruiting the experts for our user study.

- We validated the quality of our synthesized study plan through a user study in which we asked ten knowledgeable judges to assess its quality. The judges uniformly found the study plan to be of high quality, and definitely worth using as a strong starting point.

- The axioms underlying our definition of a good study plan have been abstracted from the properties of good education material discussed in the educational literature. This paper thus serves as a bridge between the education sciences and data mining.

- In the course of this study, we created datasets (list of concept phrases, concept graph) that the educational data mining researchers might use in their own work. These datasets can be useful more broadly for algorithm researchers (*cf.* "Choosing a particular class of graphs to serve as benchmark for the MinLA problem is difficult because no real large instances for this problem exist" (Silvestre, 1998)). Finally, our results can also provide a baseline for future research in algorithmic course design.

While this study does demonstrate the feasibility and promise of applying data mining to the important problem of education course design, much more experimental validation is required for baking our techniques before deploying them in the wild. Randomized control trial is often deemed the gold standard for impact evaluation (Glennerster and Takavarasha, 2013). However, we need to overcome some serious challenges like the ones pointed out in (Agrawal et al., 2014) before applying randomized control trials to the task of determining the effectiveness of study plans generated by our techniques. We also need to confirm that our techniques are equally effective for non-physical-science courses.

We envision three major future technical directions. First, we would like to incorporate user modeling into our framework. This extension will allow students to create plans to suit their background, level of preparedness, and interest, and thus pave path to personalized education. Second, we would like to investigate how we can use human inputs, both explicit and implicit, to enrich our solution while still respecting privacy. Finally, we will like to tackle more problems related to course design beyond the problem of designing study plans.

References

Agrawal, R., Jhaveri, M. H., and Kenthapadi, K. 2014. Evaluating educational interventions at scale. In *1st ACM conference on Learning@scale*. ACM, 207–208.

Aho, A. V., Garey, M. R., and Ullman, J. D. 1972. The transitive reduction of a directed graph. *SIAM Journal on Computing 1,* 2, 131–137.

Bansal, N., Blum, A., and Chawla, S. 2002. Correlation clustering. In *Machine Learning*. 238–247.

Brunskill, E. 2011. Estimating prerequisite structure from noisy data. In *Proceedings of the 4th International Conference on Educational Data Mining*. 217–222.

CHANG, H.-S., HSU, H.-J., AND CHEN, K.-T. 2014. Modeling exercise relationships in e-learning: A unified approach. Tech. rep., Institute of Information Science, Taipei,Taiwan.

COATES, M. 2010. *Shaping a New Educational Landscape: Exploring possibilities for education in the 21st century*. Bloomsbury Publishing.

DAVIS, B. G. 2009. *Tools for teaching*. John Wiley & Sons.

DESMARAIS, M. C., MALUF, A., AND LIU, J. 1995. User-expertise modeling with empirically derived probabilistic implication networks. *User modeling and user-adapted interaction 5,* 3-4, 283–315.

DOIGNON, J.-P. 2014. Learning spaces, and how to build them. In *Formal Concept Analysis*. Springer, 1–14.

FALMAGNE, J.-C. AND DOIGNON, J.-P. 2011. *Learning Spaces*. Springer.

FINK, L. D. 2013. *Creating significant learning experiences: An integrated approach to designing college courses*. John Wiley & Sons.

FORTUNATO, S. 2010. Community detection in graphs. *Physics Reports 486,* 3, 75–174.

GAGNE, R. M. AND BRIGGS, L. J. 1974. *Principles of instructional design.* Holt, Rinehart & Winston.

GANTER, B., STUMME, G., AND WILLE, R. 2005. *Formal concept analysis: Foundations and applications*. Springer.

GLENNERSTER, R. AND TAKAVARASHA, K. 2013. *Running randomized evaluations: A practical guide*. Princeton University Press.

GRAVES, K. 1996. *Teachers as course developers*. Cambridge University Press.

GRAY, W. AND LEARY, B. 1935. *What makes a book readable*. University of Chicago Press.

HANUSHEK, E. A. AND WOESSMANN, L. 2007. The role of education quality for economic growth. *Policy Research Department Working Paper 4122, World Bank*.

HO, T. K. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20,* 8, 832–844.

HSIEH, T.-C. AND WANG, T.-I. 2010. A mining-based approach on discovering courses pattern for constructing suitable learning path. *Expert systems with applications 37,* 6, 4156–4167.

KOPONEN, I. AND PEHKONEN, M. 2010. Coherent knowledge structures of physics represented as concept networks in teacher education. *Science & Education 19,* 3, 259–282.

KRIPPENDORFF, K. 1970. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement 30,* 1, 61–70.

MCNEIL, J. D. 2014. *Contemporary curriculum: In thought and action*. John Wiley & Sons.

MEDELYAN, O., MILNE, D., LEGG, C., AND WITTEN, I. H. 2009. Mining meaning from wikipedia. *International Journal of Human-Computer Studies 67,* 9, 716–754.

NATHAN, M. J. AND KOEDINGER, K. R. 2000. An investigation of teachers' beliefs of students' algebra development. *Cognition and Instruction 18,* 2, 209–237.

NOUSIAINEN, M. AND KOPONEN, I. 2010. Concept maps representing knowledge of physics: Connecting structure and content in the context of electricity and magnetism. *Nordic Studies in Science Education 6,* 2, 155–172.

NOVAK, J. D. AND CAÑAS, A. J. 2008. The theory underlying concept maps and how to construct and use them. Tech. Rep. IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition.

OHLAND, M. W., YUHASZ, A. G., AND SILL, B. L. 2004. Identifying and removing a calculus prerequisite as a bottleneck in clemson's general engineering curriculum. *Journal of Engineering Education 93,* 3, 253–257.

PAAS, F., RENKL, A., AND SWELLER, J. 2003. Cognitive load theory and instructional design: Recent developments. *Educational Psychologist 38,* 1.

PAVLIK, P. I., CEN, H., WU, L., AND KOEDINGER, K. R. 2008. Using item-type performance covariance to improve the skill model of an existing tutor. In *Proceedings of the 1st International Conference on Educational Data Mining*. 77–86.

POLLOCK, E., CHANDLER, P., AND SWELLER, J. 2002. Assimilating complex information. *Learning and Instruction 12,* 1.

PONZETTO, S. P. AND STRUBE, M. 2007. Knowledge derived from wikipedia for computing semantic relatedness. *J. Artif. Intell. Res.(JAIR) 30*, 181–212.

REICHARDT, J. AND BORNHOLDT, S. 2006. Statistical mechanics of community detection. *Physical Review E 74,* 1, 016110.

RUIZ-PRIMO, M. A. AND SHAVELSON, R. J. 1996. Problems and issues in the use of concept maps in science assessment. *Journal of research in science teaching 33,* 6, 569–600.

SCHEINES, R., SILVER, E., AND GOLDIN, I. 2014. Discovering prerequisite relationships among knowledge components. In *Proceedings of the 7th International Conference on Educational Data Mining*. 355–356.

SILVESTRE, J. 1998. Approximation heuristics and benchmarkings for the MinLA problem. In *Algorithms and Experiments (ALEX '98): Building Bridges Between Theory and Applications*. 112–128.

SUCHANEK, F. M., KASNECI, G., AND WEIKUM, G. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web 6,* 3, 203–217.

TALUKDAR, P. P. AND COHEN, W. W. 2012. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *7th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 307–315.

VUONG, A., NIXON, T., AND TOWLE, B. 2011. A method for finding prerequisites within a curriculum. In *Proceedings of the 4th International Conference on Educational Data Mining*. 211–216.

WAUTERS, K., DESMET, P., AND VAN DEN NOORTGATE, W. 2011. Acquiring item difficulty estimates: a collaborative effort of data and judgment. In *Proceedings of the 4th International Conference on Educational Data Mining*. 121–128.

WITTE, S. AND FAIGLEY, L. 1981. Coherence, cohesion, and writing quality. *College Composition and Communication 32,* 2.

WORLD-BANK. 1999. *Knowledge for Development: World Development Report: 1998/99*. Oxford University Press.