Competency Based Learning in the Web of Learning Data

Guillaume Durand
National Research Council Canada
100 des Aboiteaux str.
Moncton, Canada
Guillaume.Durand@nrc.gc.ca

Nabil Belacel
National Research Council Canada
100 des Aboiteaux str.
Moncton, Canada
Nabil.Belacel@nrc.gc.ca

Cyril Goutte
National Research Council Canada
1200 Montreal Rd.
Ottawa, Canada
Cyril.Goutte@nrc.gc.ca

ABSTRACT

In this paper, we present, discuss and summarize different research works we carried out toward the exploitation of the Web of data for learning and training purpose (Web of learning data). For several years now, we have conducted efforts to explore this main objective through two complementary directions. The first direction is the scalability and particularly the need to develop methods able to provide learners with adequate learning path in the world of big data. The second direction is related to the transition from Web data to Web of learning data and particularly the extraction of cognitive attributes from Web content. For this purpose, we proposed different text mining techniques as well as the development of competency framework engineering tools. Resulting evidence-based techniques allow us to properly evaluate and improve the relationships between learning materials, performance records and student competencies. Although some questions remain unanswered and challenging technology improvements are still required, promising results and developments are arising.

Keywords

Web learning data recommendation; Web data features extraction; Learning skills engineering.

1. INTRODUCTION

According to Wiley [23], a learning object is "any digital resource that can be reused to support learning". As such, a significant number of Web pages match the definition of learning object and Wikipedia could be seen as the prototypical provider of learning objects. The 37 million articles contained in the online encyclopedia generate around 18 billion views a month. Therefore, Web learning, even informal, is already happening on a large scale thanks to a constantly increasing number of quality contents, better connectivity and web literacy. In the meantime the tools providing web users with learning material have not really evolved since the beginning of the WWW. Wikipedia is still relatively flat in term of content hierarchy, search engines are more interested in tracking users' habits for advertising purpose rather than learning needs and no tools offer learning services similar to what can be found on virtual learning environments. A learning path recommender system providing users guidance to select the right web material to reach a targeted level of knowledge would benefit users looking to mine the web for learning purpose. The same way some big internet names have been able to determine user's purchase interests, we should be able to detect user's abilities and knowledge and align their specificities to the resources available to optimize learning. This vision does not come without deep challenges that we discuss along the different propositions presented in this paper. Among the challenges of transforming the WWW into a real web of learning, is the scalability of the solutions. This is explored in the context of our investigations regarding learning path or curriculum recommendation (Section 2). Over the years, educational data mining and recommendation technologies have proposed significant contributions to provide learners with adequate learning material by recommending educational papers [20] or internet links [13], using collaborative and/or content-based filtering. Other approaches, especially in the course generation research community, address the need for recommending not only the learning objects themselves, but sequences of learning objects [22, 12]. However, none have investigated learning path recommendation for large repositories counting millions of learning objects like the Web is potentially offering. We discuss and present developments regarding learning path in the following section before focusing in section 3 on potential approaches to convert the Web to a Web of learning data and discussing current and further developments in the fourth section of the paper.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890460

2. BUILDING (WEB) LEARNING PATH

2.1 Dynamic Dependencies

To some extent, the Web can be seen as a graph in expansion, where edges are built according to the usage made. For instance, it is usual to build edges according to the hypertext links between pages but they can also be built for a more specific learning perspective considering competencies prerequisite and gains. For this last purpose, let $G = (V, E)$ be a directed graph representing the Web of learning data. Each vertex or node in G corresponds to a learning object. Two vertices are connected if there exists a dependency relation, such that one vertex satisfies the prerequisites of the

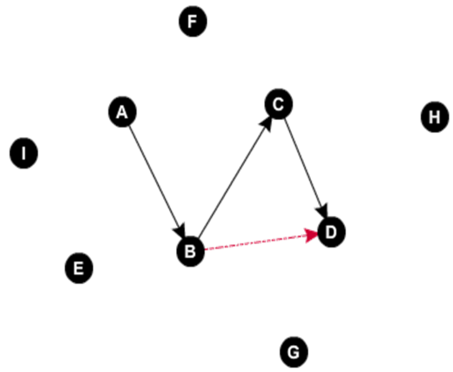


Figure 1: Illustration of the dynamic direct graph principle. If A provides competency 1, B provides competency 2, and D is accessible to a learner with competencies 1 and 2, a new edge should be created to connect B and D.

other. So, an edge $\overrightarrow{AB}$ between two vertices A and B means that the learning object A is accessible from B. Building such dependencies is non-trivial especially if they are not solely based on explicit HTTP links. We will talk more about this problem later in Section 3. Dependencies aside, the issue is now to build a learning path on a large graph, starting from user's initial competencies, and ending at the target competencies.

If the Web was simply a big graph with static edges then, depending on the learning strategy, recommending a learning path could be solved by a shortest path algorithm. However, the edges are not static, but rather dependent on the learning objects consulted previously. For example, let's consider a learning object D that would be accessible to a learner having reached mastery in competencies 1 and 2. Assume that competency 1 is provided by learning objects A and C and competency 2 is provided by learning objects B and C. D is reachable if learning objects A and B are completed or if learning object C is completed. If a learner completes learning object A at time t and learning object B at time $t+1$, the learner will have the competencies required to reach D and a new edge between B and D should be created (Figure 1).

2.2 Heuristic Approach and Graph theory

Finding a learning path, in our model, consists in looking for the shortest path in a large dynamic graph. Due to the complexity of searching such a graph, no deterministic approach is suitable. Therefore we proposed a two-stage algorithm that first reduces the problem state or graph size, then solves the reduced graph.

The first stage can be seen as a loop generating subgraphs or cliques[1], until one such clique is generated whose prerequisites are a subset of the learner's competencies. Cliques are generated in a top-down manner. We begin with the target clique, which is composed of a single learning object (we create a fictitious learning object, β, whose prerequisite competencies correspond to the list of the learner's target competencies). Cliques are then generated by finding every vertex where at least one output competency is found in the prerequisite competencies of the clique (the union of all prerequisite competencies of every learning object within the clique) to which it is a prerequisite. As such, cliques contain the largest possible subset of vertices which satisfies the condition "if every learning object in the clique is completed, then every learning object in the following clique is accessible"[2] while preserving dynamicity constraints.

In the second stage, a greedy algorithm attempts to find a path by considering each clique one after the other and reducing it to a minimal subset of itself which still verifies the condition "if every learning object in the clique is completed, then every learning object in the following clique is accessible". For each clique, the local optimum is obtained when the minimum subset of vertices with a minimum "degree", being the sum of the number of prerequisite competencies and output competencies of the vertex, are found. In other words, the greedy algorithm selects in each clique a set of learning objects minimizing the number of competencies required and gained in order to locally limit the cognitive load of the selected material. Note that the degree function could be calculated to accommodate other learning policies like maximizing learning gains to stimulate curiosity.

2.3 Further Developments

Overall, the clique-based approach is an efficient way to reduce the solution space and check the existence of a solution. However, a greedy search may not lead to the shortest learning path. To solve this issue, we investigated binary integer programming as an alternative [4]. Our implementation of the branch-and-bound (B&B) algorithm solved the accuracy issue but the performance cost is questionable. Alternatives as mentioned by Applegate et al. [1] like branch-and-cut could prove to be faster but likely not as fast as the greedy approach. Moreover, as mentioned in [11], the efficiency of reducing the solution space with the clique mechanism is highly dependent on the dataset topology (average number of gain and prerequisite competencies per learning object) as highlighted by Figure 2. For instance, the calculation time increases differently depending on the variation of the number of output or prerequisite competencies[3]. Mixing algorithms in order to balance computational time and accuracy based on the graph topology might bring interesting developments.

3. WEB TO WEB OF LEARNING DATA

3.1 Competency Mining

Using the Web of learning data requires the algorithms to be able to provide the adequate guidance to learner. The graph model proposed in the previous section to recommend learning paths requires competencies to be identified for each Web page in order to be processed and potentially recommended. However, this information is usually not available in web content. Moreover, editing each web page to manually define required metadata is not an option. Not only because this would take a huge amount of time, but also because extracting competency information is complicated and

[1]Complete subgraphs in which all the learning objects are adjacent to each other.

[2]This condition confers also the completeness property to the subgraphs. By extension, if every learning object is accessible in the following clique then all of them are adjacent.

[3]A more detailed performance study of the greedy algorithm is available in [11].

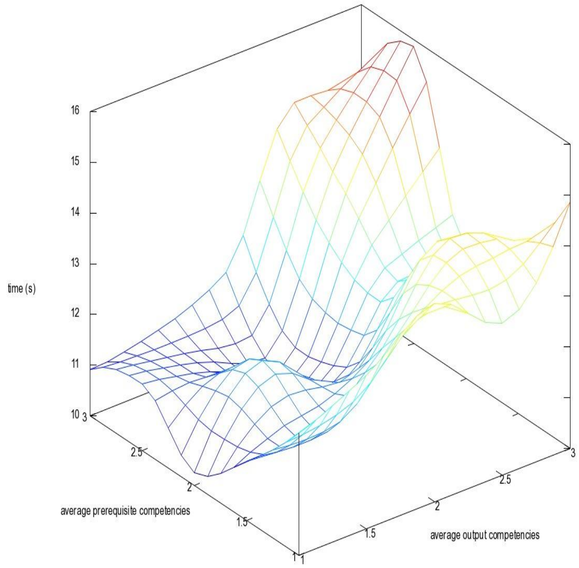


Figure 2: Computational time of the Greedy version of the two stage algorithm with different graph topologies counting 10^5 learning objects and 10^4 competencies (Intel Core 2 Extreme Q9300 CPU at 2.54GHz and 8GB RAM with Apache Cassandra Database on the same machine).

prone to mistakes. Even for experts in cognitive sciences, defining the right granularity as well as the distribution of the competencies among the learning tests is a challenging task [10].

$$\mathrm{Q} = \begin{array}{c} \\ ItemA \\ ItemB \\ ItemC \\ ItemD \end{array} \begin{array}{c} \begin{array}{ccc} Skillc1 & Skillc2 & Skillc3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Figure 3: Example of a Q-matrix illustrating the competencies gained in the dynamic graph example of figure 1.

One promising approach comes from the psychometric and educational data mining (EDM) community where some researchers are trying to automatically discover competencies based on learners performances. They pursue the objective of generating a matrix called Q-matrix that associates learning items and competencies (Figure 3). Desmarais and associates [7, 8] refined expert Q matrices using matrix factorization, with impressive successes. However as non-negative matrix factorization is sensitive to initialization and prone to local minima, a fully automated generation might be out of rich with this method. Sun et al. [19] generated binary Q-matrices using an alternate recursive method that automatically estimates the number of competencies, yielding high matrix coverage rates. Others [18, 6] estimated the Q-matrix under the setting of well-known psychometric models that integrate guess and slip parameters to model the variation between ideal and observed response patterns. They formulated Q-matrix extraction as a latent variable selection problem solved by regularized maximum likelihood, but requiring the number of competencies as input. Finally, Sparse Factor Analysis [17] was recently introduced to address data sparsity in a flexible probabilistic model.

3.2 Competency Description

All of these approaches address competency frameworks generation from slightly different angles but none of these techniques address the problem of providing a textual description of the discovered attributes. This makes them hard to interpret and understand, and may limit their practical usability. In the mean time, it is difficult to extract a full text description of a latent competency fully automatically. However, a lot of textual information is available in test items, whether in the text of the questions, hints or responses. We proposed a simple probabilistic model that extracts, from this text, the keywords that are most relevant to each skill [14]. The intuition is that relevant keywords are not always high frequency word, which tend to be common or topical words. Keywords relevant to a competency are words that are relatively *frequent* in items testing that competency, and relatively *infrequent* in items testing other competencies.

We tested this on a small dataset from the PSLC Datashop [16] containing 823 test items, with competency frameworks ranging from 44 to 108 competencies associated with at least one item. Table 1 shows examples of keywords extracted for 5 competencies with known labels. In our experiments, the labels were removed before the keyword extraction was applied to the associated test items. Note that in most cases, we extracted words from the unseen label, as well as many other related relevant words.

We quantified this process using various metrics measuring coverage and specificity of keywords, on several competency frameworks, and compared our simple probabilistic extraction technique to the common alternative of using the most common (most frequent) words. We found that with our extraction, keywords are used to describe on average 1.2 to 1.4 competencies (maximum 9), whereas the highest frequency keywords describe on average 3 competencies each (maximum 87). In fact some words like “correct” or “incorrect” are highly frequent, but clearly not very informative about competencies.

In our proposed method we only extracted key *words* from the textual data. A straightforward improvement would be to extract longer, more descriptive information such as multiword terms, short snippets from the data or more complicated linguistic structure such as subject-verb-object triples [2]. The data-generated descriptions can also be useful in the generation or the refinement of Q-Matrices. Naming competencies can offer significant information on the consistency of a Q-matrix. This can be used as an alternative or a complement to existing refinement methods based on functional models optimization [8].

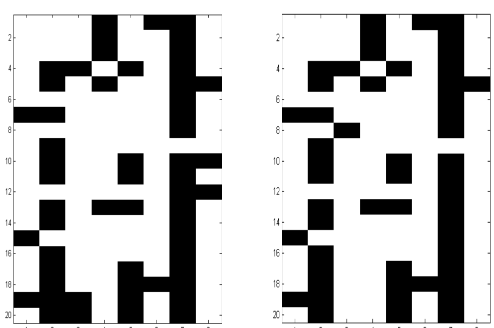
3.3 Competency Frameworks Evaluation

While competency generation seems to provide interesting results toward the objective of automatically extracting competencies from observed performance patterns, the quality of the extracted matrices is questionable. Unfortunately, the predictive quality of such matrices is sometimes only as good as chance in term of predictive accuracy. In order to address this problem, we proposed a method that aims at specifically evaluating the predictive quality of a Q-matrix. For this purpose, we proposed an evaluation method using a deterministic model using matrix factorization techniques.

Competency label	#items	Top 10 extracted keywords
_identify-sr	52	phishing email scam social learned indicate legitimate engineering anti-phishing
_p2p	27	risks mitigate applications p2p protected law file-sharing copyright illegal
_print_quota03	12	quota printing andrew print semester consumed printouts longer unused cost
_vpn	11	vpn connect restricted libraries circumstances accessing need using university
_dmca	9	copyright dmca party notice student digital played regard brad policies
_penalties_dmca	2	penalties illegal possible file-sharing fines 80,000 $ imprisonment high years
_penalties_bandwidth	1	maximum limitations exceed times long bandwidth suspended network access

Table 1: Top 10 keywords extracted, for a sample of competencies, from the text of their test items. "# item" is the number of items associated with the competency. Competency labels (left) are hidden from the extractor.

The evaluation method considers a factorization model; $R = Q \times S$ where R and S are, respectively, the "Results" ($item \times student$) and "Student" ($competencies \times student$) matrices. Assuming R observed and Q known, our method evaluates how well Q would predict unobserved results through a classical 10-fold cross validation algorithm (injecting missing values in R). For each fold, an estimated student matrix $\widehat{S}$ is obtained by solving the system of linear equations $Q \times x = R$. In our experiments, we used the *weighted least squares* method[4], although we could also impose various constraints on the s0tudent matrix using non-negative or Boolean matrix factorization. Test observations are removed from the results matrix R according to the cross-validation framework, and predictions are made for these values using the product of the Q-matrix and the estimated $\widehat{S}$. Preliminary results obtained with this method are conclusive [10]. To illustrate this method, let's consider two Q-matrices presented in Figure 4 and estimate the predictive quality of each Q-matrix. In this example, we consider a well known dataset with its original Q-matrix and a Q-matrix variant that was automatically improved.



(a) Tatsuoka (20x8) (b) Desmarais (20x8)

Figure 4: Graphical representation of the Q-matrices used in our example. Black cells in the Q-matrix (ones) indicate that one of the 20 items (rows) is associated with one of the 8 skills (columns).

Both Q-matrices are related to a dataset involving 2144 middle school students answering 20 items on fraction algebra and requiring the use of the eight following skills [21]:

1. Convert a whole number to a fraction,
2. Separate a whole number from a fraction,
3. Simplify before subtracting,
4. Find a common denominator,
5. Borrow from whole number part,
6. Column borrow to subtract the second numerator from the first,
7. Subtract numerators,
8. Reduce answers to simplest form.

[4]http://en.wikipedia.org/wiki/Least_squares#Weighted_least_squares

The Q-matrix (a) on the left side in Figure 4 was proposed by Tatsuoka [21], while the one on the right (b) was refined (automatically improved from the original (a)) by Desmarais [9]. The refining process resulted in changing the mapping between items and skills slightly: skill 3 ("Simplify before subtracting") was added to item 8 and removed from items 19 and 20, and skill 8 ("Reduce answers to simplest form") was removed for items 10 and 12. Table 2 shows Q-matrix evaluation results calculated with our evaluation method. The Root Mean Square Error (RMSE) as well as the Mean Average Error (MAE) are smaller for Q-matrix (b), showing that the refined matrix has better predictive ability than the original one (lower reconstruction errors). Calculating the RMSE for Q-matrix (a) with another cognitive model like the Additive Factor Model (AFM) yields an error close to .37. The values range obtained with our method tends to corroborate that matrix factorization models lead to prediction errors that are comparable to other cognitive diagnostic models [3] while keeping interesting advantages.

As the weighted least squares algorithm handles missing values without imputing them, this makes the proposed method usable in cases where the observed results are incomplete, such as when learners do not perform the same items, or progress at different paces. Uncompleted items should do not necessarily prevent the Q-matrix from being tested and iteratively improved in parallel with the testing activity. Using a cognitive model with very few parameters is also an advantage to generate and evaluate Q-matrices. In fact, adding parameters to a cognitive model make the evaluation of the Q-matrix more difficult since the expert misconceptions can be more easily compensated by these extra parameters.

3.4 Multi-Relational Competency Frameworks

The generation of competency frameworks is a challenging task. So far, our research effort has focused on relatively simple competency frameworks modeled by Q-matrices. In fact, Q-matrices do not consider different types of associations between tests and competencies. Nonetheless, in the graph model we proposed in Section 2 we considered two types of competencies; the competencies that are required to

Table 2: Results obtained by the Q-matrix evaluation method on expert-made Q-matrices, in root means squared (RMSE) and mean average (MAE) reconstruction errors.

Dataset	(a) Tatsuoka	(b) Desmarais
RMSE	.4051	**.3810**
MAE	.2531	**.2353**

understand the learning material and the competencies that the learning material provides to the learner. Q-matrices as competency frameworks are built around tests that require some competencies to be passed but rarely provide learning gains. As a result, the unidirectional learning competency relationship investigated with Q-matrices is not sufficient to build the graph we discussed in Section 2 to build envisioned Web of learning data. Extracting both competencies gained and required while research focusing on "simple" Q-matrix models is still emerging, represents a very serious challenge. For instance, the question of discriminating competencies required to use a web page and the competencies provided as benefits might not be simple to answer. Many web pages may not explicitly contain any words, metadata or more globally information related to required competencies. For example, a web page presenting techniques of matrix decomposition would require from the reader a good understanding of matrix multiplication but no reference to matrix multiplication might be attached to the document. Even known, the relation between competencies required and competencies gained may not be straight forward to understand and exploit. If a person consulting the page on matrix decomposition does not understand matrix multiplication, the set of competencies gain from reading the page may be limited. Inferring such limitation may require a deeper understanding of competencies dependencies as well as learner knowledge and cognitive capacities that is once again challenging to obtain in such an informal learning environment.

4. DISCUSSION

Online learners are learning without being evaluated like students would be by a teacher that is able to provide guidance based on the normative and formative assessment performed. Nonetheless, this might not necessarily mean that cognitive diagnostic models cannot benefit the Web of learning data research. Even so the cognitive diagnostic models used in competency frameworks generation require the use of test results and that typical Web pages do not measure success or failure, similar metrics could be built. One of the most widely used technologies online is collaborative filtering, using either explicit ratings filled by users or implicit ratings based on different observed behaviors (or both). Using feature extraction and navigational patterns, it should be possible to build metrics that could be used to build web learning predictive models.

However, considering only the informal nature of web learning might narrow the scope of the possible applications. The Web of learning data is composed of an increasing number of shared educational contents. Some of them are shared for indexing purpose and are formatted with rich metadata information through standards like the Learning Object Metadata. While the explicit definitions of competency gained and competency required are not designed in metadata format, related information like learning objectives, or dependencies can be exploited to define competency gains. Competency required can be induced when a hierarchical competency framework is defined, or using the learning object dependencies whenever it is correctly filled in the metadata format. This information, when available, can provide useful initialization values for competency referential refinement.

Scalability is a requirement for learning path recommendation but also for competency frameworks extraction, refinement and evaluation. So far, competency frameworks engineering has been conducted on relatively small datasets compared to the amount of information that might be treated for a Web of learning data. Hubwieser and Mühling [15] clearly embraced this issue by proposing a method to mine competencies in large data sets (tens of thousands of participants, after preprocessing). Their method is particularly interesting since it looks first at the set of items using latent traits analysis to find a set of questions that would evaluate a common competency (joint psychometric construct). Their comparison to several psychometric models (Item Response Theory, IRT) confirms the validity of the competency mapping. So far, the type of competency framework built is very simple and equivalent to a Q-matrix with one competency shared by n items but the authors mention that a multidimensional IRT can potentially be used. Alternatively, our Q-matrix evaluation algorithm could technically be implemented in this method to build and validate Q-matrices with several competencies per item. However, the main limitation of this method may come from scaling the latent trait analysis, which requires a very unbalanced ratio between items and performance observations. As a result, the method is particularly well adapted to situations when a lot of participant and results are available on a small number of items.

Considering the high number of documents and the limited performance information of web learning, promising work may come from the information retrieval community with methods based, for example, on Latent Dirichlet Allocation (LDA) [5]. Like topics in textual data analysis, competencies can be modeled as latent variables that are inferred rather than directly observed. Once topics and their associated distributions have been estimated from a corpus of documents, LDA allows the assignment of new documents to these topics. Similarly, new test items could be associated with estimated latent competencies. Note that in LDA like in the related Matrix Factorization methods, the number of latent topics/competencies must be pre-specified.

5. CONCLUSION

In this paper we discussed ongoing efforts towards a Web of learning data that would use the constantly growing resources available online to benefit web learners needs. A prominent challenge that was initially discussed is to provide learners with customized tailored learning path allowing them to reach target or key competencies. However, it is necessary to recognize that learning requires needs going beyond the navigational constraints the WWW was built upon. Considering the scale of the Web and the dynamic nature of learning we proposed some new sets of algorithms using heuristics in our effort to support web learners more adequately than traditional search engines. We also realized that the information required to discriminate web content for learning purpose goes beyond hyper-links and traditional metadata keywords. Consequently, we proposed an information model in which contents are qualified regarding the competencies they require and the competencies they provide. As latent factors, competencies are difficult to recognize even by knowledge domain experts. For this purpose we presented our results on automated competency extraction providing methods to name them and evaluate competency frameworks predictive quality. Among future developments, we envision the extension of our work on competency frameworks to multi-relational structures including the two types of competencies defined in our information model (competencies gain and required) while taking care as discussed in Section 4, of the scalability of proposed methods.

6. ACKNOWLEDGMENTS

This work is part of the National Research Council Canada program Learning and Performance Support Systems (LPSS), which addresses training, development and performance support in all industry sectors, including education, oil and gas, policing, military and medical devices.

7. REFERENCES

[1] D. Applegate, R. Bixby, V. Chvatal, and W. Cook. On the solution of traveling salesman problems. In *Int. Congress of Mathematicians*, 1998.

[2] T. Atapattu, K. Falkner, and N. Falkner. Acquisition of triples of knowledge from lecture notes: A natural language processing approach. In *7th Intl. Conf. on Educational Data Mining*, pages 193–196, 2014.

[3] B. Beheshti and M. C. Desmarais. Predictive performance of prevailing approaches to skills assessment techniques: Insights from real vs. synthetic data sets. In *7th Intl. Conf. on Educational Data Mining*, 2014.

[4] N. Belacel, G. Durand, and F. Laplante. A binary integer programming model for global optimization of learning path discovery. In *Proceedings of the Workshops held at the 7th International Conference on Educational Data Mining (EDM 2014)*, London, UK, 2014.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, Mar. 2003.

[6] Y. Chen, J. Liu, G. Xu, and Z. Ying. Statistical analysis of Q-matrix based diagnostic classification models. *Journal of the American Statistical Association*, 110(510), 2015.

[7] M. Desmarais. Mapping questions items to skills with non-negative matrix factorization. *ACM-KDD-Explorations*, 13(2):30–36, 2011.

[8] M. Desmarais, B. Beheshti, and P. Xu. The refinement of a Q-matrix: Assessing methods to validate tasks to skills mapping. In *7th Intl. Conf. on Educational Data Mining*, pages 308–311, 2014.

[9] M. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based Q-matrices. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, volume 7926 of *Lecture Notes in Computer Science*, pages 441–450. Springer Berlin Heidelberg, 2013.

[10] G. Durand, N. Belacel, and C. Goutte. Evaluation of expert-based Q-matrices predictive quality in matrix factorization models. In *Design for Teaching and Learning in a Networked World, EC-TEL 2015 conference*, pages 56–69. Springer, 2015.

[11] G. Durand, N. Belacel, and F. Laplante. Graph theory based model for learning path recommendation. *Information Sciences*, 251(10), 2013.

[12] G. Durand, F. Laplante, and R. Kop. A learning design recommendation system based on Markov decision processes. In *17th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD) Workshop on Knowledge Discovery in Educational Data*, 2011.

[13] D. Godoy and L. Amandi. Link recommendation in e-learning systems based on content-based student profiles. In C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.

[14] C. Goutte, S. Léger, and G. Durand. A probabilistic model for knowledge component naming. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.

[15] P. Hubwieser and A. Mühling. Competency mining in large data sets - preparing large scale investigations in computer science education. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR 2014)*, pages 315–322, 2014.

[16] K. Koedinger, R. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC datashop. In C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.

[17] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research*, 15:1959–2008, June 2014.

[18] J. Liu, G. Xu, and Z. Ying. Data-driven learning of Q-matrix. *Applied Psychological Measurement*, 36(7):548–564, 2012.

[19] Y. Sun, S. Ye, S. Inoue, and Y. Sun. Alternating recursive method for Q-matrix learning. In *7th Intl. Conf. on Educational Data Mining*, pages 14–20, 2014.

[20] T. Tang and G. McCalla. Data mining for contextual educational recommendation and evaluation strategies. In C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.

[21] K. K. Tatsuoka. Analysis of errors in fraction addition and subtraction problems. Final report, Computer-based Education Research Laboratory, University of Illinois at Urbana-Champaign, 1984.

[22] C. Ullrich and E. Melis. Complex course generation adapted to pedagogical scenarios and its evaluation. *Educational Technology & Society*, 13(2):102–115, 2010.

[23] D. A. Wiley. *Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy*. The Instructional Use of Learning Objects. D. A. WILEY (Ed.), 2002.