

# Semi-Automatic Generation of Competency Maps Based on Educational Data Mining

David Alfonso<sup>1</sup>, Angeles Manjarrés<sup>1,\*</sup>, Simon Pickin<sup>2</sup>

<sup>1</sup>Departamento de Inteligencia Artificial, Universidad Nacional de Educación a Distancia, C/Juan del Rosal, 16, 28040, Madrid, Spain

<sup>2</sup>Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid, C/Profesor Jose García Santesmases, 9, 28040, Madrid, Spain

## ARTICLE INFO

### Article History

Received 11 Sep 2018

Accepted 18 Jun 2019

### Keywords

Educational data mining

Concept maps

Bayesian networks

Recommender systems

E-Learning content authoring tools

## ABSTRACT

We propose a semi-automatic method for the generation of educational-competency maps from repositories of multiple-choice question responses, using Bayesian structural learning and data-mining techniques. We tested our method on a large repository of responses to multiple-choice exam questions from an undergraduate course in Languages and Automata Theory at Spain's national distance-learning university (UNED). We also draw up guidelines and best practices, with a view to defining an educational data-mining methodology and to contributing to the development of educational data-mining tools.

© 2019 The Authors. Published by Atlantis Press SARL.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

The development of data mining (DM) and analytics in the field of education came fairly late, compared to the situation in other fields. The first articles and workshops on educational data mining (EDM) began only a decade ago, roughly coinciding with the start of the Pittsburgh Science of Learning Center Data Shop [1], the largest open repository of data on the interactions between students and e-learning systems, which has supported research in many different EDM areas. Progress and interest in the EDM field has increased in the last few years. As reflected in survey papers on this topic [2–10], the objectives and application areas of EDM are wide ranging:

- Improving academic institutional effectiveness;
- Improving student learning processes and collaboration in e-learning systems (adaptive Learning Management Systems, educational Recommender Systems, Intelligent Tutoring Systems, Test and Quiz systems, learning object repositories . . . ), some being web systems, others not;
- Helping educators in defining pedagogical strategies, identifying at-risk students, and responding accordingly, and so on.

The educational data explored is also diverse:

- Data of student–computer interaction in logs of those e-learning systems that provide a navigation trace;

- Data repositories that contain academic information of a large number of students;

and different DM techniques have been used, for example,

- Regression, clustering, classification,
- Text and web mining,
- Association-rule mining (the most common),
- Distillation of data for human judgment,
- Model-based discovery,

involving machine-learning and statistical models, for example, decision trees, Bayesian networks (BNs), and neural networks. Methodologies for the knowledge-discovery process are particularly diverse, usually being specific to the research interests and the issues addressed.

The surveys show that, in the educational field, the knowledge-discovery process must be adapted, and specific DM techniques and associated development tools used. Such tools guide the selection of specific DM methods and algorithms from the wide range available, facilitating their use and configuration, and automating EDM tasks wherever possible for educators who are not experts in DM. Specific tools to automate and support preprocessing tasks that EDM users must currently do manually are also of interest. Significant progress has recently been made in this direction, and new tools are emerging constantly, as reported on in [9,11,12]. However, as stated in [12,13] no one tool is ideally suited to conducting the entire

\*Corresponding author. Email: [amanja@dia.uned.es](mailto:amanja@dia.uned.es)

EDM process, and the integration of EDM tools into e-learning environments alongside other tools such as course-designer tools remains work in progress.

## 1.1. Motivation for Our Work

Compared to other available sources of educational data, repositories of assessment data have not been widely exploited as an EDM data source. According to [8], the mining of such data, usually based on the Q-matrix concept [14] represented about 7% of EDM research in 2010, recent literature on this topic being even less abundant (see [6]). Such repositories contain valuable information on individual and collective strengths and weaknesses of students in the competencies of the learning domain, and on the relations between these competencies. Their analysis thus has great potential with regard to defining pedagogical strategies and learning pathways. Because the interpretation of such data is difficult and time-consuming when the number of students is very large, it would be of great interest to provide tools that allow this knowledge to be semi-automatically extracted and exploited, and integrate them into e-learning systems.

With regard to educational domain modeling, surveys of the available literature also reflect only a mild interest in automatic construction of concept maps (CMs) based on DM. CMs allow us to represent and organize knowledge in graphical form. A particular type of CM currently used in e-learning systems is that of learning competencies or “competency maps.” On the basis of these maps, student models representing learning progress are created, the instructional design is adapted and students are provided with personalized learning objects. In recent years, considerable effort has been made to propose standards and specifications for the definition of competencies linked to learning objectives in different educational domains (e.g. see [15]), as well as to record them as part of a student model (e.g. see [16,17]). See [18] for a review of competency models.

The manual creation of CMs is time-consuming and dependent on experts in the field [19]. CMs usually reveal an expert perspective since educational programs tend to be structured according to conventional criteria that are based on this perspective, in part due to the fact that lecturers/teachers often find it hard to see students’ difficulties and to identify the most intuitive learning pathways. CMs that reflect learning pathways that are more natural for a student who is not familiar with the subject are of great interest to the educational community, especially if they can be automatically generated, and may be used in e-learning systems for modeling both the learning domain and the progression of students in the acquisition of competencies.

## 1.2. Salient Aspects of Our Work

The research reported on in this article concerns the use of probabilistic graphical models for association rule mining of responses to multiple-choice exam questions and the subsequent semi-automatic construction of competency maps for a given learning domain from these models, enhancing the research previously conducted in this area with a generic (applicable across multiple educational domains) rigorous and systematic methodological

approach. By competency, we understand any educational outcome, either the acquisition of knowledge or development of skills (i.e., both conceptual and procedural competencies).

The method used is not fully automatic, since effective EDM must be based on models, mainly due, according to [4,6], to the need to explicitly account for the multilevel hierarchy and nonindependence in educational data. For this reason, the use of models drawn from the psychometrics literature is increasingly common in EDM publications [6,20]. Until the end of the last decade, EDM research mainly dealt with relationship-mining methods [5]. However, as stated in [3] DM, and in particular EDM, is increasingly geared towards discovery with models and distillation of data for human judgment. [10] highlights the use of these methods in educational recommender systems.

The research presented here was carried out inductively, seeking to validate our hypotheses in a particular domain before formulating a general procedure. For the experimentation we chose a UNED (*Universidad Nacional de Educación a Distancia*, the Spanish national distance-learning university) Computer Science degree course: “Languages and Automata Theory.” In order to optimize our results, we used teaching expertise in each of the EDM phases identified in [21], essentially by applying CRoss Industry Standard Process for DM (CRISP-DM) [22], currently a widely-used open standard DM process model. At the end of the experiment, we distilled our findings into the first version of a general methodology for carrying out DM on multiple-choice test responses via the application of any chosen DM techniques (in particular, not specific to probabilistic models). By contrast, the solutions to be found in the literature are either *ad hoc* (educational-domain specific) or do not attempt to cover the whole data-mining process.

To summarize what distinguishes our approach, we provide a novel method for the semi-automatic generation of CMs using automatically-learned BNs as an intermediate representation and employing a generic and rigorous methodological approach to EDM including

- Comprehensive and systematic trials of different candidate algorithms, preprocessing operations, filtering criteria, and so on;
- Designing specific EDM algorithms;
- Assessing results by standard metrics;
- Treating missing values [23], an ineluctable step when using real-world data, ignored in much of the related work;
- Emphasizing domain expert support;
- Experimenting with large repositories.

In Section 2, we present a review of the research in CMs and their construction through semi-automatic DM, paying special attention to the use of BNs with multiple-choice test responses. In Section 3, we compare our research with the most similar research results among those reported on in the literature. In Section 4, we give an overview of the use of our method in the “Languages and Automata Theory” course. We present this overview in terms of the initial version of the methodology that we have developed for the generation of CMs from repositories of multiple-choice test responses via the generation of BNs using structural learning

algorithms. In Section 5, we present our conclusions, reflecting on the potential impact of our research on educational settings and indicating promising future research lines.

## 2. STATE OF THE ART

### 2.1. EDM in Repositories of Test Results

In the literature, we find different automatic approaches to the analysis of multiple-choice question responses, the usual goal being to help instructors generate new improved multiple-choice tests (see [24,25]).

The other main goal in exploring multiple-choice test responses, this being the goal that has motivated our research, is discovering or improving student models concerning knowledge of a learning domain. Computer adaptive testing (CAT) applications [26] are the earliest examples of the use of intelligent user-modeling techniques and adaptive interfaces in educational applications. The principle behind CAT is to analyze the behavior of the learner and build a dynamic model of his/her knowledge and abilities in order to adjust test characteristics according to this model. As stated in [27], the item response theory (IRT) has been the prevalent CAT approach for the last three decades. In CAT applications, the Q-matrix method [14] has been used to create conceptual models that represent relationships between concepts and questions [28], to group student's multiple-choice question responses according to concepts [14], or to define an item-order effect model that suggests the most effective item sequences to facilitate learning [29].

CAT applications have become more sophisticated in recent years with the use of ontologies and computational linguistic techniques. Thus, [30] carries out an ontology-driven assessment of responses to questionnaires that may include open questions. Sophisticated evaluation and construction of a student knowledge model are carried out using error taxonomies and domain ontologies designed by expert teachers.

#### 2.1.1. Bayesian methods in EDM

BNs and Markov decision processes were rarely used in the early days of EDM but have been prominent techniques since around 2005, as stated in [5]; in the recent review of [6] they are used in 12% of the compiled research works. In most cases the BN is knowledge engineered and the probability values are calibrated subjectively (as in [31]).

The automated construction of a BN remains a difficult problem that involves complex algorithms and considerable computing resources. In practice, heuristics and some degree of expert intervention are required.

The earliest reference we have found in the literature concerning empirically-derived conceptual domain models obtained from multiple-choice test responses using BNs is [32]. In this approach, the categorizing of concepts and test items into a hierarchy representing dependencies between skills and student misconceptions involves some knowledge engineering. Conditional probabilities were all calibrated from empirical data. The results show that an adaptive test based on such a BN can correctly identify student skills with an accuracy of approximately 90% after the ninth question

and performs significantly better than a test with a fixed order of questions.

[33] contains a comparative study of different metrics for finding the most informative questions to create a student knowledge-model. The authors state that when using fully automatic methods not supplemented by knowledge engineering, the domain models are necessarily simple; for example, in [27] the nodes of the BN merely represent test questions.

In [34], a method to construct light ontologies (i.e., only including concepts and relations) of university course or degree domains is proposed. Ontologies that are manually designed by experienced lecturers are automatically mapped to BNs where each node corresponds to a concept of the ontology and can take a value of yes or no, depending on whether the student knows the concept or not. Then, student multiple-choice test responses are used to confirm the BN links having a high cross-entropy value. The ontologies and the derived BNs are used in a web-based educational tool. The tool is capable of inferring student knowledge states, on the basis of tests used throughout a course, and of proposing questions and learning resources on the basis of these states according to the strategy defined by the teacher. Students using the tool throughout the course obtained better final exam results than students not using it. However, it is of interest to mention that in earlier work [35], instead of manually constructing ontologies and using BNs to validate them, the authors proposed the automatic learning of ontologies using BN structural learning. We assume that this research line was discarded.

Finally, [36] describes an intelligent tutoring system concerning children's knowledge of the decimal system, the core of which is a BN that represents the domain knowledge. The system can be used to (a) predict what types of understanding errors a learner is most likely to make, given his/her answers, (b) update the parameters of the BN with new information, and (c) determine what kind of questions a learner is more likely to answer incorrectly, to facilitate proposing new questions (in the form of games) or theoretical items. The researchers used a large data set of student multiple-choice questionnaire responses and, although the automatic methods generated BNs with good performance, the experts decided to use manually-created BNs, since, in their opinion, automatically learned relationships can be difficult to understand and may differ too much from those created manually. In Section 3.3, we compare our approach with theirs and discuss why their conclusions are seemingly at odds with ours in this regard.

### 2.2. Automatic Generation of CMs

CMs are graphical languages for representing and organizing knowledge and learning processes. Originally described by Novak [37,38], they were conceived as an educational tool to facilitate both learning and assessment, as well as teaching. A CM comprises concepts (words or phrases inside an ellipse) and links (lines connecting two ellipses). Links may have associated linking phrases describing the relationship between the two concepts. A proposition is a phrase or sentence formed by two or more concepts, linked together by their linking phrases, which has a full meaning. CMs usually have a hierarchical structure with the more general concepts above and the more specific ones below, although other topologies may also be used [39]. [25] defines a CM model called the concept

effect model, representing prerequisite relations, of cardinality  $m \times n$ , between competencies of a learning domain.

Only a few articles of those surveyed in [8] concern the development of CMs using DM techniques (10 out of 300 surveyed). [40] describes the state of the art regarding the generation of CMs: in most cases, the dependence on human intervention is very high, giving an idea of the difficulty that automatic generation of CMs entails.

The methods discussed in [8,40] involve various statistical, machine-learning, and multiple-choice test-mining techniques, but in none of them are BNs used as intermediate structures between the raw data and the CM, as in our work. Neither does more recent work with similar objectives to ours use intermediate BNs [41–43]. The interest of using BNs as an intermediate format is discussed in Section 3.3.

The methods most relevant to our research, from the perspective of the origin of the data used, are those based on multiple-choice test responses and those where other structured data, such as ontologies, are used.

[44] reports on a method to automatically build CMs from multiple-choice test responses using fuzzy rules, and raise the issue of its potential usefulness in adaptive learning systems. Initially, each question must be associated to one or more concepts, indicating that the understanding of such concepts is relevant to answering the question.

The intelligent concept diagnosis system described in [19] is based on the dynamic creation of a CM from multiple-choice test responses by using DM learning techniques. In this case the lecturers also assign a weight to each concept according to its relevance for answering each question. From the CM thus constructed, the system uses the RIP algorithm [19], which computes the most appropriate learning path given the concepts the learner misunderstood, to make personalized recommendations on the study of a set of concepts. The system has been validated in a real-world environment with 245 students, with the following conclusions:

- Questions that are too easy (resp. too difficult), lead to a CM that is too simple (resp. too complex), and does not represent the actual learning state of the student;
- Large numbers of questions involving data memorization did not give rise to representative CMs either;
- The majority of the students with whom the system was tested made significant progress in their learning, the less advantaged students being the ones who obtained the greatest benefit from having the corrective path instruction.

[45] describes an algorithm that generates CMs from existing Web Ontology Language (OWL) ontologies, using the different types of relationships that are described in ontologies as the basis of the construction algorithm.

### 3. OUR APPROACH AND RELATED WORK

In the following section we place our research in the context of the state of the art described in the previous section, essentially according to the criteria established in [37] to classify methods for the

automatic generation of CMs. Note that research in the area of generating CMs from DM of exams is sparse.

#### 3.1. Goal of the Research

We conclude from our study of the literature that the results obtained so far are neither very positive nor very useful to the educational community since the software tools and methods described are not easily used across multiple learning domains. In general, the methods used are *ad hoc* and are mainly intended to be applied in CATs. We agree with [9] on the importance of integrating EDM components that are independent of the domain of learning into e-learning systems and tools for designing e-learning courses.

The ultimate goal of our research was to support lecturers and teachers and other course-content authors in adaptive e-course authoring processes, assisting them in semi-automatically building CMs that reflect more intuitive ways of learning. We propose a set of guidelines that facilitate a methodologically-rigorous DM process for obtaining such CMs from arbitrary repositories of multiple-choice test responses. In the literature we have not found any similar methodological proposal that is applicable in any domain.

We chose to represent competency models using CMs because they are familiar and easy to interpret, both to lecturers/teachers and students. We chose to experiment with the “Languages and Automata Theory” course since, according to the lecturers we consulted, the students on this course tend to jump to the next step in the study plan without having strengthened the competencies required in previous steps. Automata theory is an application domain that is not commonly addressed, neither in the EDM nor in the recommender-system and ontology-building literature, the most common application domain being programming. Finally we had access to repositories of multiple-choice test responses of a large number of students and we could rely on experienced Automata Theory lecturers.

#### 3.2. Data Sources Used

Except that of [36], in which 2500 multiple-choice test papers were used, previous studies on automatic learning of BN structures from multiple-choice test responses have only used small data sets. The second largest we found was that of [34], who validate their approach using between 300 and 500 test responses. Researchers usually apply their algorithms to relatively small ontologies, containing from four to eight concepts, each corresponding to a course topic, and either carry out *ad hoc* tests or they do not use real data (the case in [44]). Our use of Automata Theory experts allowed us to define a very complete and refined ontology to label the BNs (from 9 to almost 40 concepts).

Our experiment, on the other hand, did not involve data collection since we used existing data sources, namely our university’s exam-responses database. This enabled us to use a much larger data set than that used in most similar studies, consisting of almost 1500 multiple-choice exams. When large repositories of tests or exams are used, in general, these tests do not evaluate the same competencies. The heterogeneous tests considered in [46], for example, contain questions from different university-level courses, due to the fact that the research objective was to establish benchmarking between organisations.

### 3.3. Data Preprocessing and Construction Method

**Data preprocessing.** In educational contexts, data pre-processing is a very important and complicated task which, according to [9], sometimes takes up more than half of the total time spent solving a DM problem. Each specific educational problem requires the data to be converted to an appropriate form. None of the articles referenced, except for [36], dealt with the pre-processing phase in the systematic manner in which we have, using our knowledge of the learning domain.

**Domain knowledge support.** As is often the case in the educational field (as stated in [8]), the DM process we carried out involved data with intrinsic semantic information and *a priori* relational structure: hierarchical relationships between the competencies of the educational domain and Q-matrix like relationships [14] between multiple-choice test questions and the competencies they assess. We therefore decided to benefit from expert knowledge as in [34,36], while in [19,35,44], the experts do not intervene in the EDM process. We then elaborated a methodological proposal which we indicate how expert teachers or lecturers can support the EDM process.

**Construction method.** Our method for the semi-automatic generation of CMs, using BN as intermediate representation formalism, is novel. A CM is a data-representation format that is oriented towards presentation rather than processing, that is, towards representing information in a way that is understandable to humans. In consequence, a CM used to represent the results of DM provides a high-level abstract view of the information being mined which, in the case of a CM generated via association-rule mining of multiple-choice exam questions, contains only the most significant (unweighted) relationships. However, for many other uses of the results of data-mining, a more detailed, less abstract, data-representation format, oriented towards processing rather than presentation, would be more suitable. Using a format of the latter type as an intermediate representation in the generation of CMs facilitates using the same association-rule mining results for multiple purposes without affecting the quality of the generated CM, if the intermediate format is well-chosen. In our case, we are interested in performing probabilistic inference in the context of educational recommender systems. As well as providing a good basis for CM generation, BNs are a good choice of intermediate format for us due to the nature of this target use in recommender systems, to the fact that there is a large existing BN tool base and to the advantages of BNs for EDM discussed in Section 2.1.1. As concrete examples of such tasks, we cite identifying key competencies (those that many others depend on), identifying competencies that are difficult to acquire, and obtaining a deep understanding of the learning state of each student.

In choosing our algorithms, we have taken into account the specificity of EDM, choosing methods for hierarchical DM and longitudinal data modeling as suggested in [9], while in the work reported on in the literature the authors do not generally justify their choices. We have also developed additional algorithms that could be reused in other educational domains.

In particular, we used the hill-climbing (HC) [47] algorithm, based on heuristic search, and the hybrid algorithm max-min hill-climbing (MMHC) [48]. Regarding the metrics, we use the Bayesian Information Criterion (BIC) metric [49] to give a probabilistic

interpretation of the structure of BNs. This metric has the characteristic of favoring simpler networks, thus avoiding overfitting.

HC is a simple algorithm that is useful when there are no experts in the domain who know the structure. It is well suited to performing domain discovery, as is our case. As for MMHC, it has been used in problems with high dimensionality and with the aim of discovering causal relationships in the data, as shown in [36]. Being a hybrid algorithm, it tries to take advantage of heuristic search and independence tests. In addition, it does not require an initial ordering of the nodes, so it is easier to use than other algorithms, obtaining equal or better results, in a much shorter execution time (as stated in [50]).

[34] reports on a similar BN approach to ours, though the focus of our work is more similar to that of an earlier work by the same authors, reported on in [35], an article of only two pages where little details are provided of the EDM process carried out. The structural-learning algorithms used in our research are more recent, and work with both more variables and more data. In addition, we experiment with various ways of preprocessing and aggregating data to determine which ones lead to better results.

In [36], the most rigorous and meticulous approach reported on in the literature, although the automatic methods generated BNs with good performance, the experts preferred to use the manually-created BNs. But we contend that, in this case, the fact that the automatically-constructed BNs do not enrich those constructed manually is due to the domain of application being limited and extremely well-studied, that is, for which expert knowledge is plentiful. These characteristics enabled the authors to use a very methodical knowledge-engineering process, relying on very elaborate, precise, specific tests, and categorizations of comprehension errors. On the other hand, although our experiment concerned only one undergraduate degree course, by design, all the EDM steps could be directly applied to any other course. A generic proposal such as ours clearly cannot rely on the different domains to which it is applicable being extremely well-studied. In addition, we define a method to semi-automatically generate CMs, with which teachers will be more familiar, from BNs.

In much of the recent work concerning the generation of CMs from multiple-choice test responses such as [41-43], the efficiency of the generation algorithms is a primary consideration. Our CM generation, on the other hand, uses BNs as an intermediate format, in spite of the fact that learning BNs is known to be hard problem, due to the fact that the use of BNs brings important benefits, as discussed above. Moreover, in this context, computationally-efficient heuristic search, even though it does not guarantee an optimal result, has been shown to produce good solutions. In particular, hill-climbing algorithms are generally considered to offer a good trade-off between efficiency and quality of the models learned. For our CM generation, therefore, we study which of the well-known heuristic search algorithms developed by BN researchers are most suitable to our purposes and, in common with the rest of the literature on generating BNs from multiple-choice test responses, do not explicitly discuss the efficiency of the algorithms studied, since the difference between the different candidates in this regard is not of great importance here.

**Missing values.** Missing values is a problem widely discussed in the field of DM and statistical inference [23]. Among the options

to deal with them in the learning of BNs, that is, eliminating the instances with missing values, imputing the missing values using various methods, or learning the structure of the BN using the data with missing values [17,18]; in the research referenced in this article we have only found examples of the first case. Thus, the algorithm in [44] is limited to data sets without missing values, which is a significant constraint on combining the results of different student examinations. By contrast, we do take into account this possibility and propose various solutions to address it.

**Setting up the map.** Concerning the process of setting up the map, in the research reviewed, when missing values are not taken into account, a different map is generated for the set of students that sat each test (see e.g. [19]). By contrast, taking account of missing values allows data combined from multiple exams or tests to be used as the basis for the map generation.

In [44], to transform the graph of questions into a CM, simple substitution of questions by the concepts associated with them is made, directly eliminating duplicate links by calculating their mean. The CMs generated do not have link phrases, which is common in the reviewed methods, since the creation of these phrases is usually based on a model previously designed by experts. We view the results of the structural learning of BNs probabilistically, assuming that the links represent correlations between variables, not causal mechanisms. In our methodology, guidance is given for the manual generation of the link phrases.

Another reason for generating CMs is that when this facility is integrated into an authoring tool or a recommender system, in the absence of sufficient data to generate the BN, it will always be possible for the teacher to manually enter the CMs.

### 3.4. Validation of the Results

In none of the reviewed articles are metrics used to assess the generated BNs or CMs. The validation of the results reported on in the literature is either subjective (by teachers or other experts, as in [35]) or objective (through user experiences where students and teachers experiment with educational tools based on CMs, as in [19,34,36]). In some cases, as in [44], no validation is reported. We used validation based on both the criteria of two lecturers of the “Languages and Automata Theory” course, and well-known metrics both for BN and CM assessment. We also check whether the CM enables us to predict correlations between correct (resp. incorrect) answers using different exams from those used during the DM process.

## 4. GENERATING CMS VIA EDM

In the course of our experiments (reported on in detail in [51]), we have developed a set of guidelines for the semi-automatic generation of CMs that we consider of sufficient generality to be applied in different learning domains and with different DM techniques. We have also collected expert knowledge in DM that is particularly relevant in this EDM context. We have essentially followed CRISP-DM, a methodology that integrates common approaches to DM problems through six major interdependent phases of a cyclic nature [22]. In this section, we also summarize how we specialized the phases of this methodology to tackle our EDM process and provide a high-level view of our methodology in the form of BPMN [52] process diagrams (see Figures 1–4, where Figures 1 and 2 are generic while Figures 3 and 4 are instantiated for our experimentation with BNs).

### 4.1. Understanding the DM Problem

#### 4.1.1. Educational objectives as DM objectives

Our objective is to explore the relationships between answers to different test questions in order to infer from them relationships between competencies of an educational domain. This goal requires the collaboration of experts in the field to identify the competencies required to answer each test question correctly. The correlations between competencies disclose, in general terms, that students often possess them or not simultaneously, which means that some are prerequisites of others or that they share a common set of more basic competencies as subcompetencies. In our case, the link sentences of the CMs are “is a relevant subcompetency of,” “is an essential subcompetency of,” and “have subcompetencies in common with” (see [17] for other possibilities).

#### 4.1.2. Project success

The project success will be measured in terms of the degree to which the CMs obtained effectively reflect the learners’ cognitive structures. This is done firstly by having expert lecturers or teachers assess them, secondly by checking them against responses to exams different from those used during the DM process, and thirdly by validating, via user experiences, e-learning tools that make use of them.

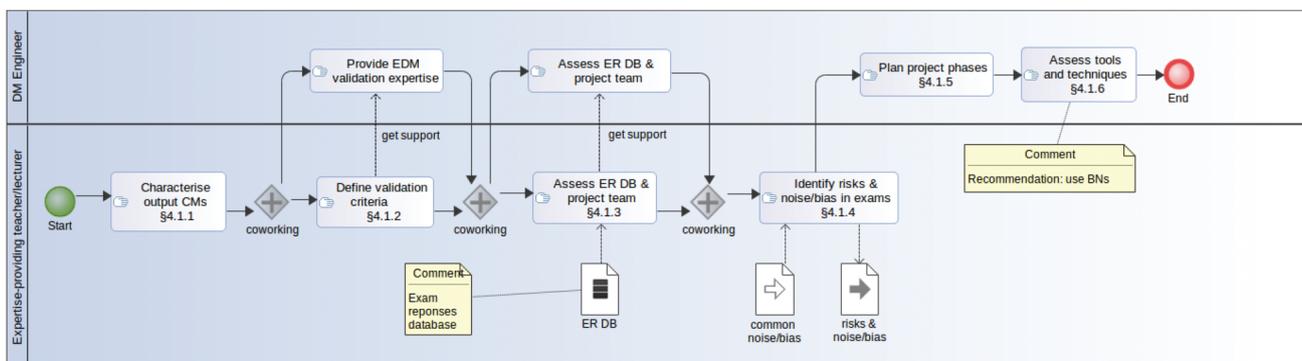


Figure 1 | Understanding the educational data mining (EDM) problem.

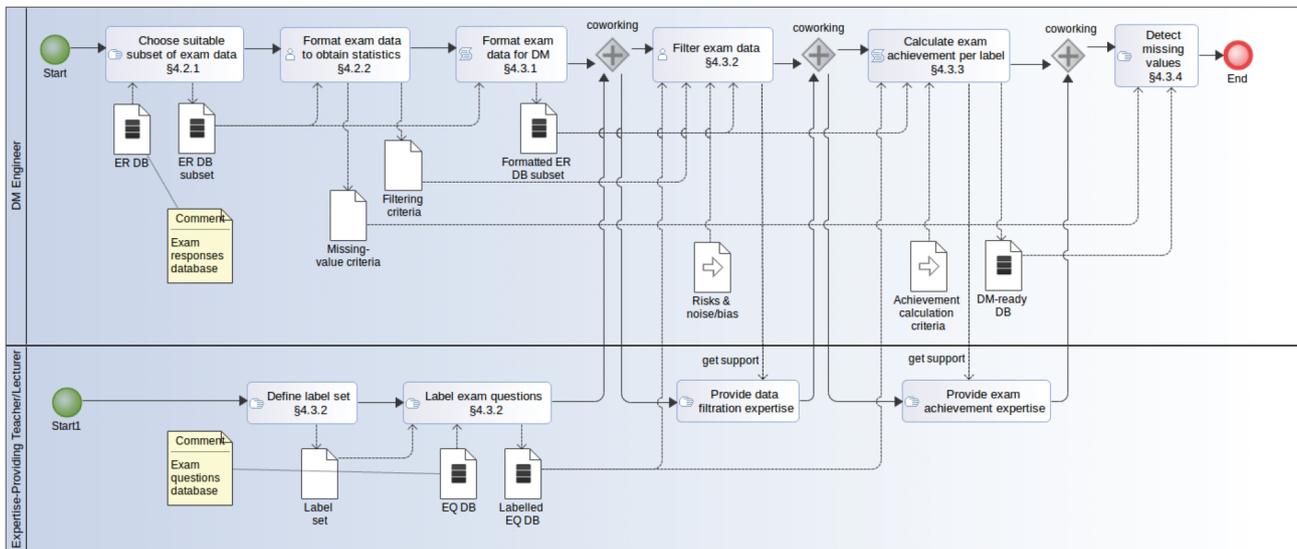


Figure 2 | Understanding and preparation of exam data.

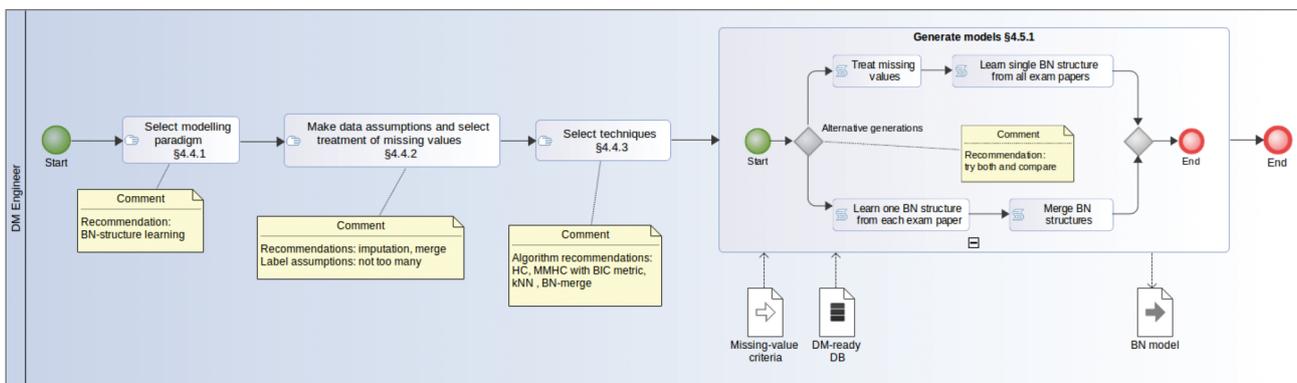


Figure 3 | Modeling choices and model generation instantiated for our experimentation with Bayesian networks (BNs).

#### 4.1.3. Assess situation

The data sources are repositories of multiple-choice exams. The project team should include experienced lecturers or teachers, experts in the learning domain, and software engineers, experts in DM. In our case, we initially considered exams from 2004 to 2012 of the “Languages and Automata Theory” course. Two researchers had been lecturers and authors of relevant training materials and exams of this course for more than 10 years.

#### 4.1.4. Risks and contingencies

After a comprehensive and systematic trial of different filtering criteria, guided by the use of metrics (see Section 4.5.3) we identified some sources of noise and other factors that could distort or invalidate the results:

- In many exams, different competencies are not measured equally exhaustively.
- Students tend to give correct responses to easy questions and to questions on topics that have been regularly asked about in previous years’ exams, and incorrect responses to difficult questions: correlations will therefore appear between easy questions and between difficult questions.

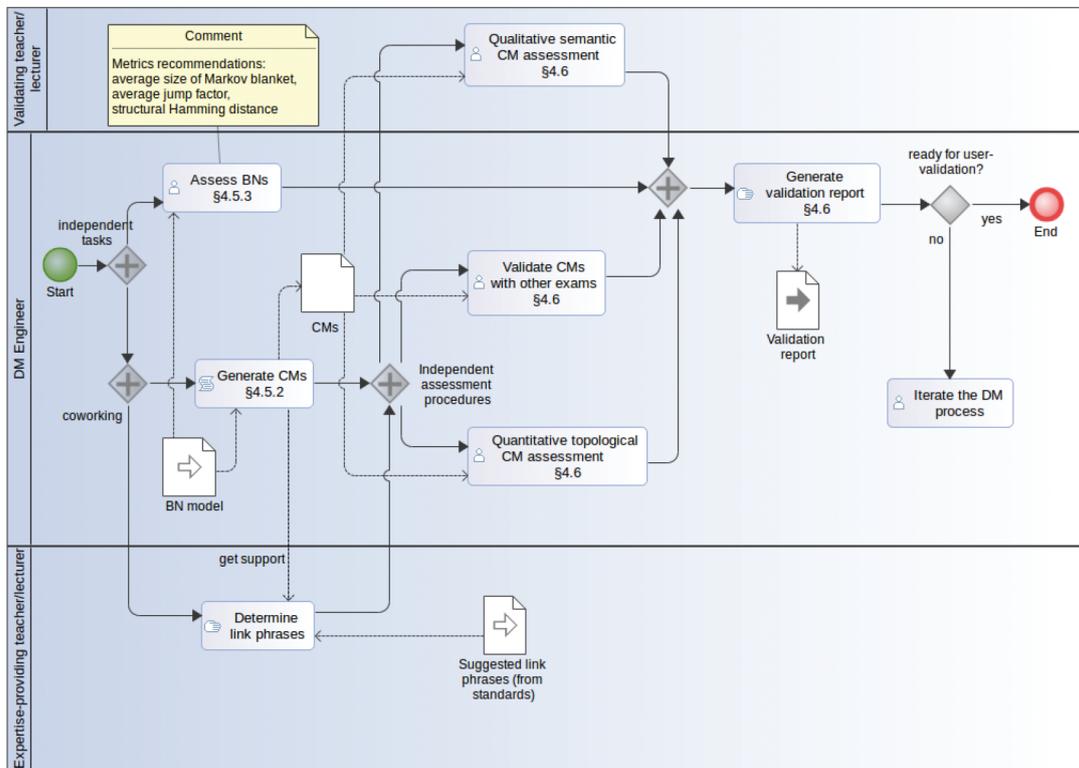
- Some multiple-choice questions can be answered correctly by process of elimination.
- Some students respond randomly to some multiple-choice questions (only low-scoring students since questions answered incorrectly receive negative marks).
- Some questions are “trick” questions or can be answered correctly by applying logic or common sense.
- Examinations with few students can distort the results.
- Some concepts and competencies are associated to a large proportion of the questions on any given topic.

In our experiment, we have attempted to minimize these problems through selection and filtering of data.

Finally, a crucial aspect is to maintain the confidentiality of student information, by anonymizing the data.

#### 4.1.5. Project plan

In the DM area of concern, the participation of experts at every stage of the process is critical. When performing the series of DM trials, the results of each trial should be interpreted by the expert



**Figure 4** | Concept map (CM) generation and validation instantiated for our experimentation with Bayesian networks (BNs).

lecturers or teachers using the educational domain knowledge in order to then design the next trial (possibly with new filtering criteria, new data types and formats, different DM techniques, and different treatment of missing values).

#### 4.1.6. Initial assessment of tools and techniques

We used the Python programming language [53], which facilitates processing chains and code reuse, the R language [54], which has a large number of modules for statistical analysis, and GNU Octave [55], which has simple and powerful matrix-handling. The choice of Bayesian techniques has already been justified.

### 4.2. Data Understanding

#### 4.2.1. Initial data collection

The initial data is the completed examinations available in the University database. For our experiment, we restricted the final exams used to 2004–2006 (4 exams per year, so a total of 12 exams).

#### 4.2.2. Data description and exploration

First, a data matrix, whose rows correspond to students who sat the exam and columns to exam questions, should be built for each exam. Then basic statistical results are obtained, in order to understand the data characteristics, and based on these, filtering criteria and treatment of missing values are derived (see Table 1).

We look for correlations between questions, in the sense of pairs of questions that students tend to answer either both correctly or both incorrectly (see Figure 5).

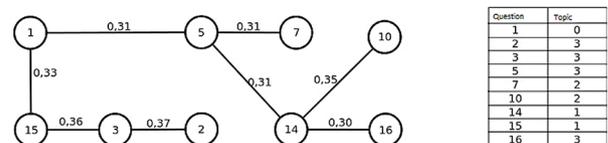
### 4.3. Data Preparation

The experienced lecturers or teachers provide data-filtration and attribute-derivation or -transformation criteria to transform the original raw data into a shape suitable for the chosen mining algorithm.

**Table 1** | Basic statistics of 2006 exams.

Year	Session	Week #	Stud.	% Qual. $\geq 5$	Av. Qual.	% Blank	# Stud.	Year
2006	J	1	112	71	5.7392	20	524	
		2	290	54	5.0067	19		
	S	1	122	35	3.7084	24		
		2	0	0	0.0000	0		

J: June; S: September; using GNU / Octave.



**Figure 5** | Correlation between exam questions in a 2004 exam. Only correlations  $\geq 0.30$  shown. 0: General concepts, 1: Finite states machines, 2: Push down automata, 3: Turing machines.

### 4.3.1. Data set description

The data is arranged in a table with one row per student who sat at least one exam and one column per question in an exam. The initial matrix-element values are the following: the number of the answer the student selected (in our case, one, two, or three) or if the answer was left blank, 0. We stored the exam data from 2004 to 2006 as a single matrix of 1415 rows, by 144 columns.

### 4.3.2. Data selection and enrichment

An expert classifies the questions of the multiple-choice tests into topics and labels them by a list of concepts (the theoretical concepts involved in the question, see Table 2) and competencies (the conceptual or procedural competencies that the students show when they answer a question correctly), paying attention to the considerations regarding filtering mentioned in the section “risks and contingencies.” The topic, concept, and competency labels are useful for the construction of CMs and for filtering.

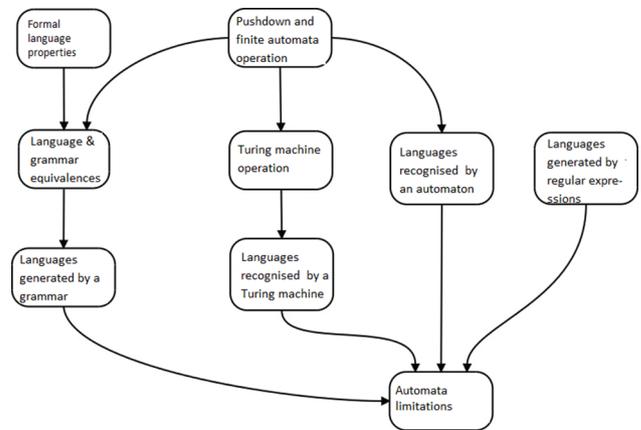
In our experiment, the basic statistical analysis showed that three exam questions (of 144) were correctly answered by a very high number of students (95%–100%). These were filtered out, as were five questions that could not be associated with a specific set of concepts or competencies. Finally, labels in the first quartile, when ordered decreasingly by number of appearances in the exam questions, were also discarded.

The ideal label set is one containing few labels, all of which appear in all the tests (thus reducing missing values) with the same frequency of occurrence, but which are specific enough to be of interest. The test sets that give best results are those in which each test evaluates the same competencies and each competency is concerned in the same number of questions of that test. This is not often the case, one reason being that there is a tendency to assess more frequently those competencies that are considered more important. In the end, we considered four topics, 47 concepts, 9 general competencies, and 27 specific competencies.

The DM can be carried out based on any of the types of label discussed above. Here, we concentrate on generating BNs y CMs from the competency labels, though we also associated to the questions the following extra label types: 1) Type, indicating whether the questions are purely rote, concern establishing relationships between concepts, have a practical component, or consist of solving purely practical exercises. 2) Difficulty, calculated from the percentage of students who correctly answered the question. The difficulty and type labels would be useful in a recommender system for generating tests adapted to each student’s learning level. In Figure 6, we present

**Table 2** | List of some concepts with number of questions to which each has been assigned (in brackets).

Words (59)	Chomsky normal form (10)
Finite state machine (35)	Regular grammar (10)
Regular language (33)	Abnormal termination (10)
Turing machine (30)	Kleene star (9)
Equivalence (26)	Transitions (8)
Decidability (10)	Generative grammar (6)
State (10)	Pushdown stack (2)



**Figure 6** | A manually built concept map (CM) showing dependencies between general competencies of “Languages and Automata Theory” course.

a manually built CM showing dependencies between general competencies of “Languages and Automata Theory” course.

### 4.3.3. Preprocessing of test responses

First, initial values are transformed into values from the set {0,1} indicating whether the answer is correct or not. Blank responses, are treated as incorrect, supposing that the student judged his/her knowledge insufficient to risk answering. Calculated values can be of several different types:

- Binary discrete, for example, success or failure in the questions associated to a label (the percentage of correct answers to obtain the value success is given by a threshold);
- Discrete multinomial, for example, all, some or none;
- Continuous, for example, a percentage in the range [0, 1] (numerical values indicate the percentage of questions bearing the label that have been answered correctly).

In our experiment, binary discrete values and continuous values were calculated; and, within the first, different thresholds for the number of correct answers were tried.

### 4.3.4. Origin of missing values

Missing values arise here when tests are considered jointly, since the labels have the same meaning in all exams but not all the labels will appear in each, that is, a given exam may, or may not, have questions labeled with a given topic, competency, or concept. According to the different mechanisms that can lead to the absence of data, our case is classified as Missing Completely At Random (MCAR).

## 4.4. Modeling

### 4.4.1. Select modeling

The methodology admits different DM techniques. In particular, the literature and our experimentation support the use of BNs in EDM, specifically, the automatic learning of BNs from multiple-choice test responses.

#### 4.4.2. Data assumption/missing values

There are several options to treat missing values when learning BNs [56,57]. Best results were obtained for imputation of missing values using the kNN algorithm. However, if the percentage of missing values is high (over 40%), imputation of them is not recommended. We also studied treating the data for each exam independently, thereby avoiding missing values but limiting the competencies that appear in each BN to those appearing in the corresponding exam. This leads to one BN per exam, which must then be merged into a single BN. As for missing values,

A particular problem when generating from the concept label is that some concepts have different meanings depending on their context (i.e., course topic). Also, if all the concepts are used together, the generated BN is very large, difficult to validate and unreliable. For this reason, we suggest multiple-choice questions be grouped by topic.

#### 4.4.3. Techniques

The large number of DM tools and techniques available prohibits exhaustively experimenting with all of them.

We limit ourselves to recommending the techniques that worked best among the ones we chose to experiment with. For BN structural learning, we used the HC [47] and MMHC [48] algorithms with the BIC metric.

Based on trials using different metrics, we advise setting the threshold for the binary discrete data at 100% (having experimented with 50%, 75%, and 100%), using MMHC, filtering noise sources, and using the k-nearest neighbors (kNN) algorithm for imputation of missing values. Instead of having to choose between merging individual-exam BNs and using a single BN with imputation of missing values, we performed both calculations in order to compare the results.

After evaluating the different options using quality metrics we concluded that for high numbers of missing values, the structural learning of a BN for each exam and the subsequent merge of these BNs into a single BN containing the most relevant links was the most effective strategy. To this end, we developed a BN-merge algorithm that is knowledge-domain independent and therefore reusable in other educational areas. The merged BN contains the most relevant links while maintaining the property of being a directed acyclic graph. The link weights are calculated by weighting their strength with the number of data items used when they were automatically learned. The pseudocode for the algorithm used is shown in Figure 7.

### 4.5. Model Building

#### 4.5.1. Model description

We suggest the use of the software package “bnlearn” for R [58]. Regarding treatment of missing values, among the various techniques available we used eliminating instances with missing values, unidimensional imputation with the median, and multidimensional imputation kNN. We used the R packages Hmisc [59]

```

Input: bna[n] (BN array), Output: bn (BN)
# 1) Merge in a BN avoiding bidirectional links. The
# resulting BN will be connected if at least one of the nodes
# of each input BN appears in at least one other node.
bn = newBN()
for each network in bna
  for each link in bna.links()
    if (link already treated) then continue
    # We add the link weight in all exams where it appears
    nor_weight = sum(bna.link.weight)
    if (link.inverse() exists in bna) then
      inverse_weight = sum(bna.link.inverse().weight)
      if (total_weight < total_inverse_weight) then
        bn.add_link(link, nor_weight + inverse_weight)
      else
        bn.addlink(inverse_link, nor_weight + inverse_weight)
    else
      bn.add_link(link, nor_weight)
# 2) Transform the graph into an acyclic directed graph
# eliminating the links of less weight that are part of a cycle
while there exist (bn.strongly_connected(), length > 1)
  erase_link = nothing
  for each strongly connected component (scc)
    in bn.strongly_connected() do
      if (scc already treated) then continue
      if (cfc.minimum_weight_link() < link_to_erase.weight())
        link_to_erase = scc.minimum_weight_link()
# The graph still connected, since the link was part of a cycle
bn.erase_weight(link_to_erase)
return(bn)

```

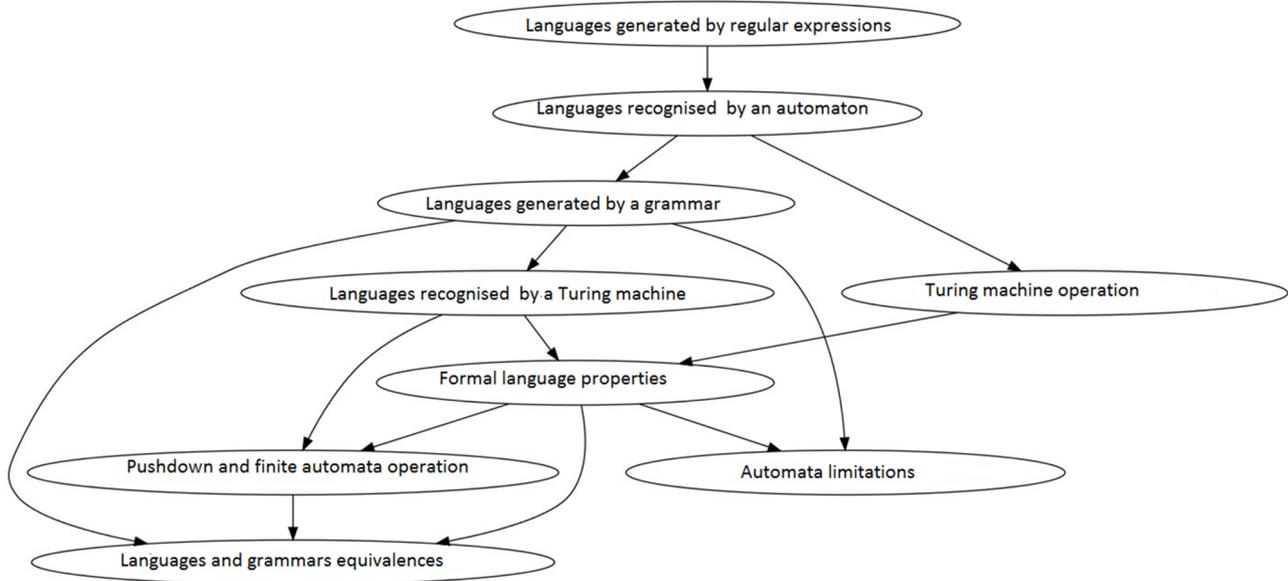
**Figure 7** | Pseudocode for the Bayesian network (BN)-merge algorithm.

and VIM [60] for the two types of imputation, respectively, the best results being obtained with kNN.

In [51], the BN structural learning experiments we carried out are described in detail, specifying in each case the options chosen (for the labels, data sets, filtering, learning algorithm, and missing-value treatment criteria), the validation used, and the conclusions drawn. In Figures 8–10 we show examples of the BNs generated in different experiments.

#### 4.5.2. Post-processing DM results

Figure 10 illustrates the result of merging the BNs learned from the competency labels of nine exams into a single BN, where links between the nodes indicate correlations occurring in one or more of the examinations. The nodes of the generated CM correspond to those of the BN but the links with the least weight in the BN are not present in the generated CM (unless needed to ensure the graph is connected). The resulting CMs are then completed manually by experts adding the appropriate link phrases. The experts also review the orientation of the links and decide when they represent causality. If they cannot determine the direction of a link (if they can, the relationship is propaedeutic), in many cases it is possible to interpret it as a relationship of some other kind, for example, one of those identified in [17].



**Figure 8** | Bayesian network (BN) learned with the max-min hill climbing (MMHC) algorithm from a single exam, eliminating missing values, using binary-valued data obtained with a threshold of 100% and filtering possible noise sources. The structural Hamming distance (SHD) value of this network is 16.

### 4.5.3. Model assessment

Metrics guide the research by providing criteria to determine which algorithms, parameters, preprocessing operations, filtering criteria, and so on, are more suitable. The metrics we used to validate the learned BN were Average Size of Markov Blanket, and Average Jump factor, which focus on the structure of the network and its complexity (visual and probabilistic), and the structural Hamming distance (SHD), which we used to compare the networks learned with those built manually by the expert lecturers. In Tables 3 and 4 we show some properties of the BNs generated in different experiments. Figure 11 illustrates the effect of filtering noise in different experiments.

### 4.6. Validation and Deployment

Expert lecturers made a qualitative semantic assessment of our results stating that the maps generated in our experiment were quite expressive, easy to read and understand, and that the

**Table 3** | Properties of the generated exam-specific BNs and the corresponding merged BN in one of the experiments.

Year	Session	Week	Nodes	Nonrelated Nodes	Links	Av. Jump Factor	Av. Size Markov
2004	F	1	12	1	12	1	2.17
		2	12	3	8	0.67	1.5
2005	F	1	15	5	7	0.47	0.93
		2	9	2	6	0.67	1.56
2006	S	1	11	2	5	0.45	0.91
		1	12	8	2	0.17	0.33
		2	11	2	7	0.64	1.27
Merged BN	S	1	14	4	9	0.64	1.29
		24	0	47	1.96	6.5	

BN, Bayesian network.

**Table 4** | Properties of the generated topic-specific BNs using the concept label aggregated from all examinations using kNN imputation of missing values.

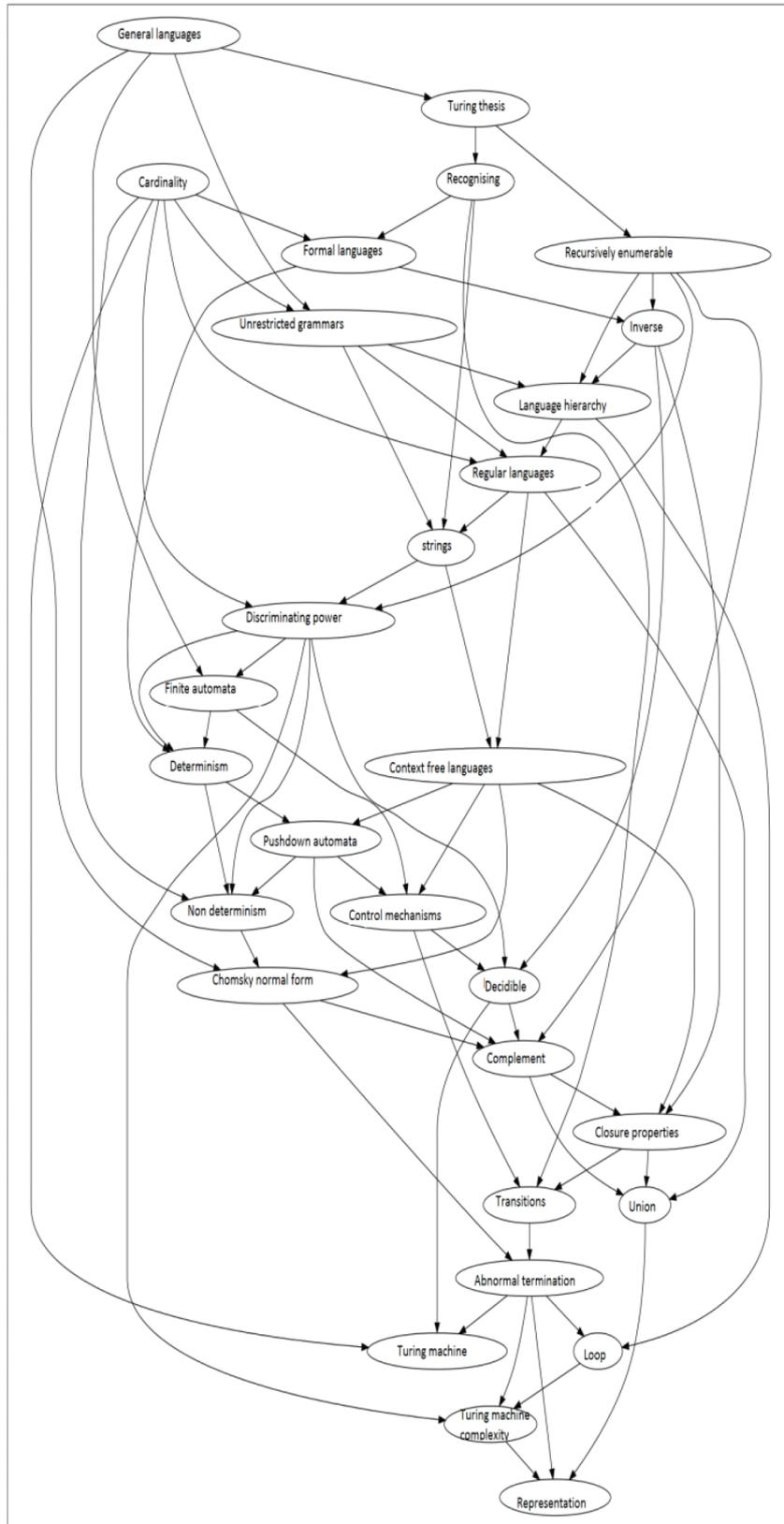
BN	Nodes	Nonrelated Nodes	Links	Av. Jump Factor	Av. Size Markov
Topic 1	29	0	78	2.69	9.10
Topic 2	25	0	61	2.44	8.40
Topic 3	29	0	69	2.38	8.14

BN, Bayesian network; kNN, k-nearest neighbors.

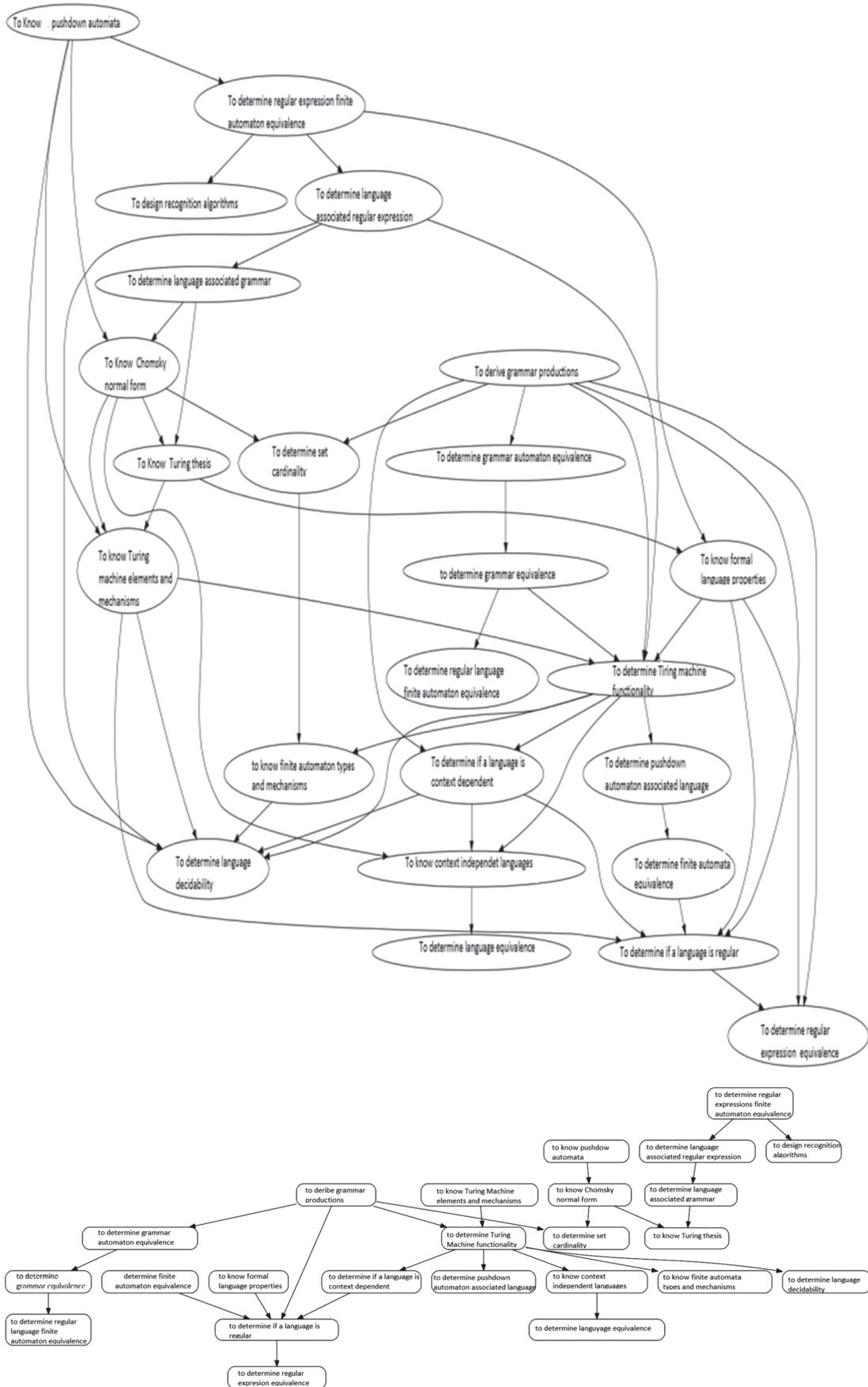
Note that no node is isolated and that every concept has a high number of links with other concepts.

relationships between nodes effectively revealed “deep dependencies” between competencies that reflect the way students learn. Thus, the five most strongly-weighted links did not appear in the manually-created graphs but made sense for the expert lecturers, who interpreted them as revealing typical student study practices. Thus, for example, to determine if a language is context-independent, the students tend first to determine if it is regular.

Among the features that can be used to assess the quality of a CM, [39] distinguishes between semantic features and topological features. With regard to the quantitative analysis of the map structure, they define a topological taxonomy that classifies CMs into seven levels of increasing structural complexity according to five criteria (the existence of hierarchical structure, size of concept labels, presence of linking phrases, number of branching points, and number of cross links). According to these criteria, the maps generated in our experiment were classified in levels zero and one. Apart from the fact that these criteria include some very specific concepts such as label length, these values reflect the fact that, for our purposes, reliability of the links and overall simplicity are desirable characteristics of the resulting CM, the former implying erring on the side of caution when deciding whether or not to include a link. Figure 10 shows an example of a generated BN and the CM obtained from it.



**Figure 9** | Bayesian network (BN) learned with the max–min hill climbing (MMHC) algorithm using binary-valued data, 100% threshold, relating concepts from the topic “Turing machines and recursively enumerable languages,” and with k-Nearest Neighbors (kNN) imputation of missing values.



**Figure 10** Top: The Bayesian network (BN) resulting from the merge of 9 independently learned BNs, each generated from the competency data of one exam. Bottom: The derived concept map (CM) containing 29 links, a reduction of 68% w.r.t. the BN, which has 69.

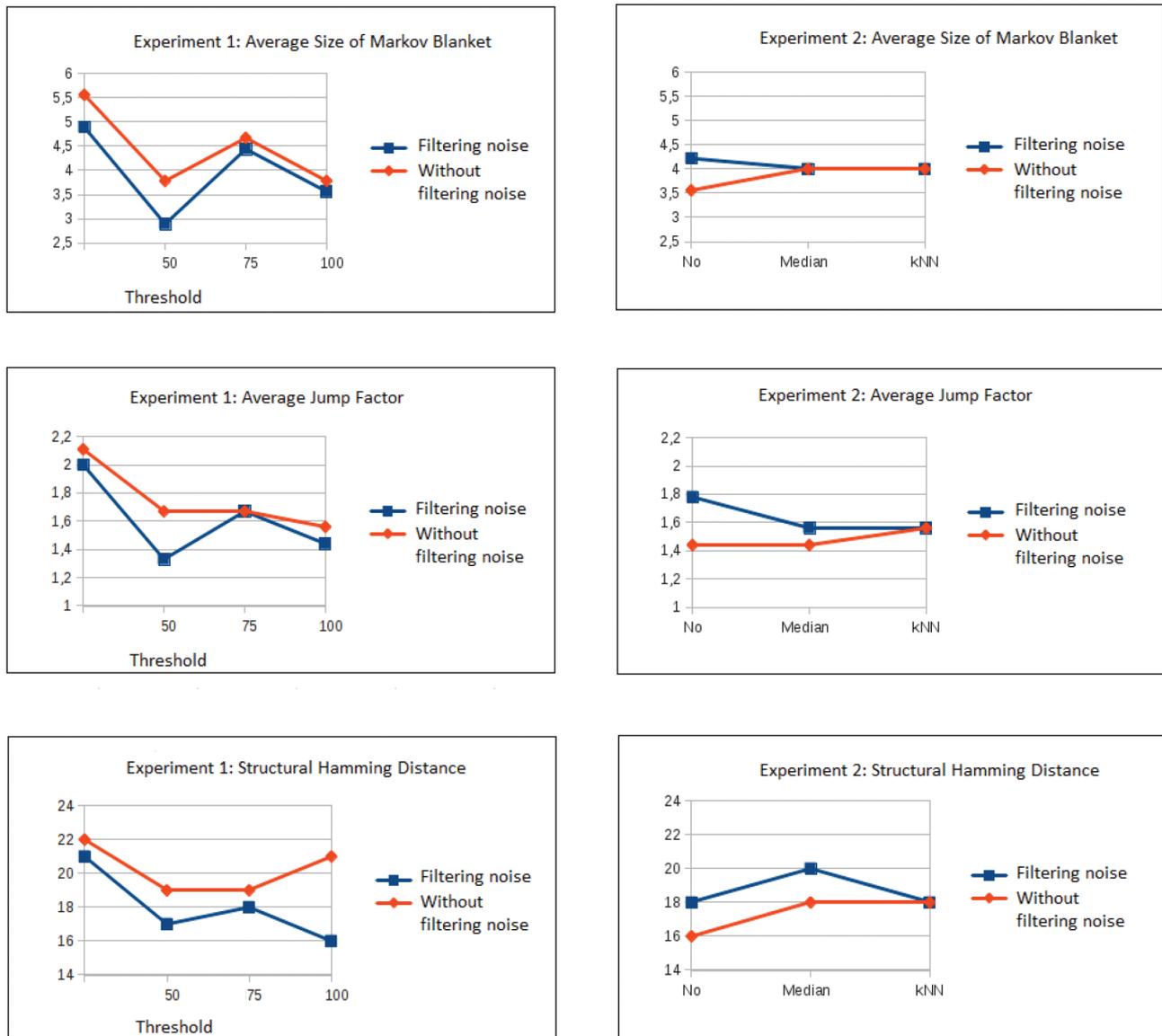


Figure 11 | Effect of filtering noise in different experiments.

As an additional validation of the generated CMs, we checked whether the links between competencies of the CM enable us to predict correlations between correct (resp. incorrect) answers using different exams from those used during the DM process. Finally, we also carried out an initial validation of the use of the generated CMs in a recommender system prototype. For more details regarding the validation of the CMs generated for the Automata Theory course, see [51].

## 5. DISCUSSION OF THE RESULTS

The CMs obtained from our experiment with the Automata Theory course, validated both by our expert lecturers and by using standard metrics, reflect individual and collective strengths and weaknesses of students in the competencies, and the relationships between competencies. Automata Theory is an application domain that is not commonly addressed, neither in the EDM nor in the recommender-system and ontology-building, literature.

It is well proven that the rigorous application of a methodology as widely validated as CRISP-DM is one the best guarantees of the success of a alfonsoDM process, and therein lies the main interest of our proposal. Additionally, based on our experience, we have provided a set of well-founded recommendations and warnings to tackle the EDM process, and guidelines for using educational domain knowledge to assist its different phases. Given the generality of our approach, all the EDM steps are applicable to any other course, while at the same time recognizing the need to experiment in other educational domains that could pose particular challenges. In the literature, no other proposal of a general methodology like ours is reported, only *ad hoc* solutions. Automatic procedures have little or no advantages over what experts can provide if a rigorous DM process is not followed, particularly in the curing of data since, as we have seen, test response data are very susceptible to noise, so that if the examinations have not been designed systematically, various distortions will appear.

## 6. CONCLUSIONS AND FUTURE WORK

The integration of EDM tools into e-learning environments, alongside other tools, has been little explored. We contribute to this area by proposing a novel method to exploit repositories of student assessment data as an EDM data source, an issue that has been little explored in the EDM literature, for the automatic construction of competency maps and other kinds of CMs, using BNs as an intermediate format. We also define the first version of a generic methodology for carrying out DM on multiple-choice test responses via the application of any chosen DM techniques (in particular, not specific to probabilistic models).

Previous studies on automatic learning of BN structures from multiple-choice test responses lack a generic, systematic and rigorous approach, do not use very innovative learning algorithms, and use considerably smaller data sets and many fewer concepts than those employed here (almost 1500 multiple-choice exams and from nine to almost 40 concepts). However, the use of large data sets such as ours to generate a single BN requires sophisticated treatment of missing values. If the percentage of missing values is large, we found that an alternative approach of avoiding them by generating multiple BNs and then merging them into a single BN gives better results. The merge algorithm that we have developed for this purpose is reusable across multiple educational domains.

The semi-automatic generation of CMs (that better reflect a student perspective), as opposed to just using BNs, raises possibilities for improving e-learning tools, in particular educational recommender systems.

Regarding the benefits for teachers, CMs are graphical tools that are easy to understand for those teachers not already familiar with them and the generated CMs are closer to the student perspective, being richer than those manually-generated by an expert. These maps would help to identify more learning pathways that are more natural for students. The BNs generated as intermediate format facilitate the identification of those competencies that are more difficult to acquire, finding the most frequently-made mistakes to learn from, recognizing at-risk students, defining pedagogical strategies, and so on, thereby facilitating student follow-up. Automatic reports that provide an individual and global overview of student learning progress would be of great help to teachers in analyzing the test results, a difficult and time-consuming task. Based on the propaedeutic relationships represented in the BN, a student's answers provide a broad understanding of his/her learning state.

Regarding the benefits for students, the generated CMs could be used to integrate introspective capabilities in a recommender system, due to the ability to access a graphical representation of the progress of one's own learning (i.e., an open student model). On the basis of the intermediate BMs, the instructional design of the corresponding course can be automatically adapted and students can be provided with personalized recommendations. An educational tool can indicate grades of possession of competencies, further helping the students to see their strengths and weaknesses, and which areas of the domain they need to concentrate on.

Our research has an additional potential impact on online educational settings, contributing to the integration of authoring tools embedding DM supporting facilities for the whole DM process in educational systems, with the aim of semi-automatically extracting and exploiting the knowledge implicit in student assessment data.

Once supported, refined, and completed in other experiments, the guidelines we provide could form the basis for an EDM development methodology with an associated software platform equipped with a library of DM techniques to give support in all the DM phases, guiding the selection of specific DM methods and their configuration. DM tasks could be automated to the point that lecturers might only need to provide a mapping from multiple-choice test items to the competencies required to answer them. In the absence of sufficient data to automatically generate the BNs and CMs, the recommender system could still be used, since the lecturer or teacher could manually enter maps of a given knowledge domain. Once collections of test responses are provided, the tool could recommend additional relations on manually-created maps.

## CONFLICT OF INTEREST

David Alfonso, Angeles Manjarrés and Simon Pickin declare that they have no conflict of interest.

## ACKNOWLEDGMENTS

This work was partially supported by the Spanish MEC project TIN2015-65845-C3-1-R (MINECO/FEDER).

## REFERENCES

- [1] K.R. Koedinger, R.S. Baker, K. Cunningham, A. Skogsholm, B. Leber, J. Stamper, A data repository for the EDM community: the PSLC DataShop, in: C. Romero, S. Ventura, M. Pechenizkiy, R.S.J.D. Baker (Eds.), *Handbook of Educational Data Mining*, CRC Press, Boca Raton, 2010, pp. 43–56.
- [2] R.S. Baker, Data mining for education, *Int. Encyclopedia Educ.* 7 (2010), 112–118.
- [3] R.S. Baker, Educational data mining: an advance for intelligent systems in education, *IEEE Intell. Syst.* 29 (2014), 78–82.
- [4] R.S. Baker, P.S. Inventado, Educational data mining and learning analytics, in: J.A. Larusson, B. White (Eds.), *Learning Analytics: From Research to Practice*, Springer Verlag, New York, 2014, pp. 61–75.
- [5] R.S. Baker, K. Yacef, The state of educational data mining in 2009: a review and future visions, *JEDM-J. Educ. Data Mining.* 1 (2009), 3–17.
- [6] A. Dutt, M.A. Ismail, T. Herawan, A systematic review on educational data mining, *IEEE Acces.* 5 (2017), 15991–16005.
- [7] P. Guleria, M. Sood, Data mining in education: a review on the knowledge discovery perspective, *Int. J. Data Mining Knowledge Manage. Process.* 4 (2014), 47.
- [8] C. Romero, S. Ventura, Educational data mining: a review of the state of the art, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 40 (2010), 601–618.
- [9] C. Romero, S. Ventura, Data mining in education, *Wiley Interdiscip. Rev. Data Mining Knowledge Discov.* 3 (2013), 12–27.
- [10] B. Bakhshinategh, O.R. Zaiane, S. ElAtia, D. Ipperciel, Educational data mining applications and tasks: a survey of the last 10 years, *Educ. Info. Technol.* 23 (2018), 537–553.
- [11] E. García, C. Romero, S. Ventura, C. De Castro, A collaborative educational association rule mining tool, *Internet High. Educ.* 14 (2011), 77–88.

- [12] S. Slater, V. Joksimović, R.S. Kovanovic, D. Baker, D. Gasevic, Tools for educational data mining: a review, *J. Educ. Behav. Stat.* 42 (2017), 85–106.
- [13] P.K. Udupi, N. Sharma, S.K. Jha, Educational data mining and big data framework for e-learning environment, in *Proceeding of 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO 2016)*, IEEE, Noida, India, 2016, pp. 258–261.
- [14] T. Barnes, The q-matrix method: mining student response data for knowledge, in *American Association for Artificial Intelligence 2005 (Educational Data Mining Workshop)*, Pittsburgh, PA, 2005, pp. 1–8.
- [15] ISO/IEC JTC 1/SC 36, ISO/IEC 20006-1:2014. Information Technology for Learning, Education and Training—Information Model for Competency, Part 1: Competency General Framework and Information Model, ISO/IEC, Geneva, Switzerland, 2014.
- [16] CEN/TC 353: CEN EN 15981, *European Learner Mobility, Achievement Information (EuroLMAI)*, CEN, Brussels, Belgium, 2011.
- [17] J. Chapman, S.A. Sutton (Eds.), Licensed under a creative commons attribution 4.0 international license, ASN Description Framework Schema, Achievements Standard Network, 2017. <http://standards.asn.desire2learn.com/>
- [18] M. El Asame, M. Wakrim, Towards a competency model: a review of the literature and the competency standards, *Educ. Info. Technol.* 23 (2018), 225–236.
- [19] C. Lee, G.G. Lee, Y. Leu, Application of automatically constructed concept map of learning to conceptual diagnosis of e-learning, *Expert Syst. Appl.* 36 (2009), 1675–1684.
- [20] N.A. Rashid, M.N. Taib, S. Lias, N.B. Sulaiman. Z.H. Murat, R.S.S.A. Kadir, EEG theta and alpha asymmetry analysis of neuroticism-bound learning style, in *Proceeding of 3rd International Congress on Engineering Education (ICEED)*, IEEE, Kuala Lumpur, Malaysia, 2011, pp. 71–75.
- [21] J. Han, J. Pei, M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, Waltham, MA, 2011.
- [22] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, *CRISP-DM 1.0 Step-by-Step Data Mining Guide*, SPSS, Richmond, VA, 2000.
- [23] T.D. Pigott, A review of methods for missing data, *Educ. Res. Eval.* 7 (2001), 353–383.
- [24] A. Merceron, K. Yacef, Educational data mining: a case study, in: C-K Looi, G.I. McCalla, B. Bredeweg, J. Breuker (Eds.), *Proceeding of Artificial Intelligence in Education*, IOS Press, Amsterdam, The Netherlands, 2005, pp. 467–474.
- [25] G.J. Hwang, A data mining approach to diagnosing student learning problems in sciences courses, in: J. Wang, (Eds.), *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, IGI Global, Hershey, PA, 2008, pp. 2928–2942.
- [26] D.J. Weiss, G. Kingsbury, Application of computerized adaptive testing to educational problems, *J. Educ. Meas.* 21 (1984), 361–375.
- [27] M.C. Desmarais, X. Pu, A Bayesian student model without hidden nodes and its comparison with item response theory, *Int. J. Artif. Intell. Educ.* 15 (2005), 291–323.
- [28] P.I. Pavlik Jr, H. Cen, K.R. Koedinger, Learning factors transfer analysis: using learning curve analysis to automatically generate domain models, in: T. Barnes, M.C. Desmarais, C. Romero, S. Ventura (Eds.), *Proceeding of the 2nd International Conference on Educational Data Mining*, Córdoba, Spain, 2009, pp. 121–130.
- [29] Z. Pardos, N. Heffernan, Determining the significance of item order in randomized problem sets, in: T. Barnes, M.C. Desmarais, C. Romero, S. Ventura (Eds.), *Proceeding of the 2nd International Conference on Educational Data Mining*, Córdoba, Spain, 2009, pp. 111–120.
- [30] F. Bouarab-Dahmani, M. Si-Mohammed, C. Comparot, P.J. Charrel, Automated evaluation of learners with ODALA: application to relational databases E-learning, *Int. J. Comput. Intell. Syst.* 3 (2010), 357–369.
- [31] C. Conati, C.A. Gertner, K. Vanlehn, Using Bayesian networks to manage uncertainty in student modeling, *User Model. User-Adap. Interac.* 12 (2002), 371–417.
- [32] J. Vomlel, Bayesian networks in educational testing, *Int. J. Uncertainty Fuzziness Knowledge-Based Syst.* 12 (2004), 83–100.
- [33] L.M. Rudner, An examination of decision-theory adaptive testing procedures, in *Proceeding of Annual meeting of the American Educational Research Association*, New Orleans, LA, 2002.
- [34] F. Colace, M. De Santo, Ontology for e-learning: a Bayesian approach, *IEEE Trans. Educ.* 53 (2010), 223–233.
- [35] F. Colace, M. De Santo, M. Vento, An automatic algorithm for building ontologies from data, in *Proceeding of the International Conference on Information and Communication Technologies: From Theory to Applications*, IEEE, Damascus, Syria, 2004, pp. 117–118.
- [36] K.B. Korb, A.E. Nicholson, *Bayesian Artificial Intelligence*, second ed., CRC Press, Boca Raton, FL, 2010.
- [37] J.D. Novak, A.J. Cañas, The theory underlying concept maps and how to construct and use them, *Práxis Educativa.* 5 (2010), 9–29.
- [38] J.D. Novak, D.B. Gowin, *Learning How to Learn*, Cambridge University Press, Cambridge, England, 1984.
- [39] A. Valerio, D.B. Leake, A.J. Cañas, Automatic classification of concept maps based on a topological taxonomy and its application to studying features of human-built maps, in: A.J. Cañas, P. Reiska, M.K. Åhlberg, J.D. Novak (Eds.), *Concept Maps: Connecting Educators*, *Proceeding of the Third International Conference on Concept Mapping*, IHMC, Tallinn University, University of Helsinki, Tallinn, Helsinki, 2008, pp. 122–129.
- [40] J.H. Kowata, D. Cury, M. Boeres, A review of semi-automatic approaches to build concept maps, in *Proceeding of 4th International Conference on Concept Mapping*, Universidad de Chile, Viña del Mar, Chile, 2010, pp. 40–48.
- [41] N. Gupta, V. Mayank, M. Geetha, S. Rai, S. Karanth, A tree based approach for concept maps construction in adaptive learning systems, in *Proceeding of International Conference on Applications and Techniques in Information Security*, Springer, Singapore, 2018, pp. 172–182.
- [42] X. Huang, K. Yang, V.B. Lawrence, An efficient data mining approach to concept map generation for adaptive learning, in *Proceeding of Industrial Conference on Data Mining*, Springer, Leipzig, Germany, 2015, pp. 247–260.
- [43] A. Acharya, D. Sinha, Construction of automated concept map of learning using hashing technique, in *Proceeding of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, Springer, Bhubaneswar, India, 2015, pp. 567–578.
- [44] S.M. Bai, S.M. Chen, Automatically constructing concept maps based on fuzzy rules for adapting learning systems, *Expert Syst. Appl.* 35 (2008), 41–49.
- [45] V. Graudina, J. Grundspenkis, Concept map generation from OWL ontologies, in *Proceedings of the Third International*

- Conference on Concept Mapping, IHMC, Tallinn University, University of Helsinki, Tallinn, Helsinki, 2008, pp. 263–270.
- [46] R.A. Carrasco, P. Villar, M.J. Hornos, E. Herrera-Viedma, A linguistic multi-criteria decision making model applied to the integration of education questionnaires, *Int. J. Comput. Intell. Syst.*, 4 (2011), 946–959.
- [47] D. Margaritis, Learning Bayesian Network Model Structure from Data (No. CMU-CS-03-153), Carnegie-Mellon University Pittsburgh Pa School of Computer Science, Pittsburgh, PA, 2003.
- [48] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, *Mach. Learn.* 65 (2006), 31–78.
- [49] G. Schwarz, Estimating the dimension of a model, *Ann. Stat.* 6 (1978), 461–464.
- [50] L.E. Brown, I. Tsamardinos, C.F. Aliferis, A comparison of novel and state-of-the-art polynomial bayesian network learning algorithms, in: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05), AAAI Press, Pittsburgh, PA, 2 (2005), 739–745.
- [51] D. Alfonso, Aprendizaje computacional de mapas conceptuales en términos de modelos gráficos probabilistas. Aplicación al diseño de un sistema recomendador para la asignatura de Teoría de Autómatas I, Computational learning of concept maps using probabilistic graphical models. Application to the design of a recommender system for the course: automata Theory I, Master Thesis, Universidad Nacional de Educación a Distancia (UNED), Spain, 2013. <http://www.ia.uned.es/docencia/posgrado/tfm-memorias/AlfonsoGaspar-sept13.pdf>
- [52] OMG, Business Process Model and Notation (BPMN). Version 2.0, OMG, Needham, MA, 2011.
- [53] G. Van Rossum, F.L. Drake, Python Language Reference Manual, Network Theory Limited, United Kingdom, 2011.
- [54] R. Ihaka, R. Gentleman, R: a language for data analysis and graphics, *J. Comput. Graph. Stat.* 5 (1996), 299–314.
- [55] J.W. Eaton, D. Bateman, S. Hauber, GNU Octave Manual Version 3. Artpower International Publishing, Shenzhen, China, 2015.
- [56] D. Heckerman, A tutorial on learning with Bayesian networks, in: D.E. Holmes, (Eds.), *Innovations in Bayesian Networks*, Springer Verlag, Berlin, Heidelberg, 156, (2008), 33–82.
- [57] J.H. Lin, P.J. Haug, Exploiting missing clinical data in Bayesian network modeling for predicting medical problems, *J. Biomed. Informatics.* 41 (2008), 1–14.
- [58] M. Scutari, Learning Bayesian networks with the bnlearn R package, *J. Stat. Softw.* 35 (2010).
- [59] M. Templ, A. Kowarik, P. Filzmoser, Iterative stepwise regression imputation using standard and robust methods, *Comput. Stat. Data Anal.* 55 (2011), 2793–2806.
- [60] A. Kowarik, M. Templ, Imputation with the R Package VIM, *J. Stat. Softw.* 74 (2016)