Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

DISSERTATION THESIS

Adaptive Testing using Bayesian Networks

2020

Martin Plajner

Bibliografický záznam

Autor:	Ing. Martin Plajner, České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská, Katedra softwarového inženýrství
Název práce:	Adaptivní testování znalostí pomocí bayesovských sítí
Studijní program:	Aplikace přírodních věd
Studijní obor:	Matematické inženýrství
Školitel:	Ing. Jiří Vomlel, PhD., Ústav teorie informace a automatizace Akademie věd ČR
Akademický rok:	2019/2020
Počet stran:	170
Klíčová slova:	Bayesovské sítě; počítačové adaptivní testování; monotonicita; učení parametrů; model studenta

Bibliographic Entry

Author:	Ing. Martin Plajner, Czech Technical Universtiy in Prague, Faculty of Nuclear Sciences and Physical Engineering, Department of Software Engineering
Title of Dissertation:	Adaptive Testing Using Bayesian Networks
Degree Programme:	Applications of Natural Sciences
Field of Study:	Mathematical Engineering
Supervisor:	Ing. Jiří Vomlel, PhD., The Institute of Information Theory and Automation Czech Academy of Sciences
Academic Year:	2019/2020
Number of Pages:	170
Keywords:	Bayesian networks; computerized adaptive testing; monotonicity; parameter learning; student model

Acknowledgments

I would like to thank the following people who helped me during my PhD and were a great support to me for the whole time.

Especially, my supervisor Jirka Vomlel who took great care of me and taught me a lot not only in the field of scientific research. I appreciate his time he invested in our work and his thoroughness during revisions of our articles. I am very glad I had the opportunity to have him personally as my supervisor as I can not imagine a better one.

The whole team of the Department of Decision-Making Theory of the Institute of Information Theory and Automation under the Czech Academy of Sciences. Especially, to Milan Studený, Václav Kratochvíl, Martin Kružík, and Marie Kolářová who were always helpful and willing to provide advice when needed. I will always cherish the memory of the time we spent together at the institute and conferences.

Last, but not least, to my wife and family who were understanding during the whole time of my PhD. Even in the times when I was approaching deadlines and I was not able to focus on them rather than articles, during the waiting time for the review decision, and while I was deep in thoughts on a complex problems. I know it may not have been always easy and I feel in debt they were all there for me.

This work was supported by the following grants and funding

- The Grant Agency of the Czech Technical University in Prague, Grants
  No. SGS16/175/OHK3/2T/14 and
  No. SGS17/198/OHK4/3T/14
- Czech Science Foundation
  Project No. 16-12010S and
  Project No. 19-04579S.

ABSTRAKT

Testování lidských dovedností je v dnešním světě velmi často opakovanou a běžnou úlohou. Metodologie, s kterou je prováděna, zůstala po mnoho let beze změny. Je zde však potenciál pro zlepšení tohoto procesu. Jedna z možností je využití konceptu počítačového adaptivního testu. Tento koncept má za cíl sestavení modelu studenta schopného měřit jeho nepozorované dovednosti a na tomto základě předpovídat jeho výsledky v testování. Tato snaha nám umožuje vytvářet kratší a přesnější verze testů, protože jsme schopni pokládat takové otázky, které lépe pasují k danému studentovi.

V této disertační práci je náš výzkum zaměřen na koncept počítačového adaptivního testování s využitím bayesovských sítí jakožto modelu studenta. Představujeme metodologii provádění testů s pomocí těchto modelů a ověřujeme přidanou hodnotu adaptivního testu oproti klasickému testování. Toto ověření je provedeno na umělých datech a dále na dvou empirických sadách. První z těchto sad byla sesbírána jako středoškolský test z matematiky a druhá sada je tvořena výsledky české státní maturity z matematiky. Naše testy prokázaly, že využití konceptu adaptivního testu snižuje potřebnou délku testování a poskytuje věrohodnější výsledky. Navíc lze model studenta využít k získání dalších informací o konkrétním studentovi namísto pouhých odpovědí v testu.

V našem výzkumu předkládáme vyhodnocení efektivity využití bayesovkých sítí jako modelu studenta a experimentální potvrzení tohoto přístupu. Dále jsme identifikovali, popsali a otestovali vliv speciální vlastnosti těchto modelů, monotonicity. Monotonicita vyžaduje, aby model sploval určité podmínky kladené na jeho parametry. Empiricky jsme potvrdili, že tyto podmínky zlepšují kvalitu modelu naučeného z dat a to především v situaci, kdy je učebních dat malý objem. Navrhli a představili jsme novou metodu učení parametrů bayesovských sítí zajišující dodržení těchto podmínek. Tato metoda učí modely, které jsou monotonní a ty dle našich experimentů dosahují lepších výsledků v aplikacích ve srovnání s nemonotonními modely stejně tak jako ve srovnání s monotonními modely naučenými pomocí konkurenčních metod. Monotonicita je významná vlastnost, která napomáhá procesu učení a umožuje nám naučit spolehlivější parametry. Na místech, kde je v praxi monotonicita očekávána, jsou navíc tyto model snáze přijímány experty v odvětví. Oblast uplatnění monotonních modelů je široká a jedná se o běžnou vlastnost modelované reality. Použití přesahuje z adaptivního testování do dalších oblastí, kde je učení modelů s malými datovými vzorky běžným jevem. Na těchto místech může monotonicita výrazně pomoci.

ABSTRACT

Testing of human skills and abilities is a task which is being repeated frequently in the modern world. The testing methodology has remained the same for a long time but there are ways to potentially improve this process. One way is by using the concept of computerized adaptive testing. This concept aims at modeling a student, measuring his/her (unobservable) skills and, based on those results, predicting his/her outputs in testing. This effort allows us to create a shorter and more precise test as we are able to ask questions suiting the particular student better.

In this dissertation thesis, our research is centered around the concept of computerized adaptive testing using Bayesian networks as student models. We present the methodology of facilitating the adaptive test with this type of model and verify the added value of using the concept of CAT over the classical approach. The verification is performed either on artificial data or on two empirical datasets. One dataset is collected as a mathematics test at high schools, the second is the official results dataset of Czech National Final High School Exam. Our tests proved that using the adaptive approach in testing decreases the length of the test and provides more reliable results. Moreover, we can use the student model to extract more information about the student rather than just the score of a single test.

In our research we use Bayesian networks as student models. We provide an evaluation of their effectiveness for this task and experimental proofs. We have identified, described and tested the effect of a special condition of these models, monotonicity. The monotonicity condition requires a model to satisfy special conditions placed on its parameters. We empirically proved that this condition improves the quality of the model which is learned from data, especially in cases where the learning dataset is small. We derive and present a new method for learning monotone parameters. This method uses learned models which are monotone. Based on our experiments these models provide better results than non-monotone methods and competitive monotone methods. Monotonicity is an important concept which helps learning models and allows us to learn more reliable parameters. Monotone models are more likely to be accepted by final users in areas where monotonicity is to be expected. The application area of such models is large as it is a quite common feature of modeled reality. Their application spans over the domain of CAT to other domains as well, where learning with a small dataset may be a common problem and monotonicity can help a lot there.

Contents

1. Introduction and Current State of Art

This dissertation thesis is a research project of Computerized Adaptive Testing, its applications, and theoretical background of student models. The main focus is placed on Bayesian networks, their theoretical improvements, and practical applications. The research has been carried out as a part of the author's PhD studies at the Faculty of Nuclear Sciences and Physical Engineering of the Czech Technical University and the Institute of Information Theory and Automation of the Czech Academy of Sciences (UTIA, CAS), with the aid and co-authorship of Jiří Vomlel at UTIA, CAS. This work is structured as a collection of papers which have been published in recent years and presented in academic journals and at prestigious peer-reviewed conferences. In the first part we provide an overview of the whole thesis, its workflow and a summary of its methodology and achieved results.

The thesis lies in the intersection of two main areas:

- Computerized Adaptive Testing and
- Bayesian networks.

In this section we give a brief overview of both main areas and later address their overlap in our research work.

1.1. *Computerized Adaptive Testing*

Testing human abilities and human knowledge is a very common task in modern society. The computerized form of testing is also getting increased attention with the growing use of computers, smart phones and other devices which allow us to easily contact the test audience. Computerized Adaptive Testing (CAT) (Wainer and Dorans 2015; Almond and Mislevy 1999; van der Linden and Glas 2000, 2010) is a concept of testing where a student is performing a computer-administered and -controlled test. The computer system selects questions for the student to be tested and it evaluates his/her performance. This is being done in order to create a shorter version of the test by asking correct questions (tailored to each particular student). If performed properly, the measurement of the student's ability/knowledge is more precise (Pine and Weiss 1978), the test is fairer, the student is better motivated, and less time is consumed (Moe and Johnson 1988; Tonidandel et al. 2002).

The process can be divided into two phases: model creation and testing. In the first one, the student model is created while, in the second one, the model is used to actually test the students. The student model is a construction which is used to model the actual student. The model should describe the student and his/her skills as closely to reality as possible. There are many different model types (Almond and Mislevy 1999; Culbertson 2014; Cowell et al. 1999) that can be used for adaptive testing. In this work, we cover Item Response Theory (IRT), which is a model regularly used for CAT, Bayesian and Neural networks (BNs and NNs); both of these models are commonly used for a large variety of tasks in many areas of artificial intelligence. The testing part always follows the same scheme regardless of the selected model. With the prepared and calibrated model, CAT testing repeats the following steps:

- The next question to be asked is selected.
- This question is asked and an answer is obtained.
- This answer is inserted into the model.
- The model (which provides estimates of the student's skills) is updated.
- (optional) Answers to all questions are estimated based on the current estimates of his or her skills.

This procedure is repeated until we reach a termination criterion. There are many different stopping criteria. It can be a time restriction, the number of questions, or a confidence interval of the estimated variables (i.e., reliability of the test).

The concept of CAT can be used for not only testing but also teaching. The student model is a powerful tool with a lot of information about the student. It models his/her abilities, which can subsequently be used to point him/her in the direction most appropriate for his/her future studies. It means that there will be no unnecessary time spent on too simple tasks and, on the contrary, the student will focus on the topics he/she is not as strong in. This application is very promising in the modern era of e-learning and remote and automated procedures.

1.2. *Bayesian networks*

In this part, we provide an informal introduction to Bayesian networks (BNs), their scope and references. The formal definition, as well as specific notation, is always included in the individual papers relevant to the specific theory of the particular paper, or can be found in, e.g., in Pearl (1988); Nielsen and Jensen (2007).

A Bayesian Network, as a probabilistic graphical model, is a structure representing conditional independence statements. It consists of the following components:

- a set of variables (nodes);
- a set of oriented edges;
- a set of conditional probabilities.

Edges between variables have to form a directed acyclic graph (DAG). Each variable is either continuous or discrete with a finite list of mutually exclusive states. For each variable, a conditional probability distribution conditioned by its parents is defined in terms of either the conditional probability table or a function defining the distribution. In the entire research project, except for minor exceptions, we are restricted to the discrete variables.

An example of a BN is given in Figure 1. This example network was used to model students in the adaptive testing scenario. We can see skill nodes, (S1 - S8) representing student knowledge, and question nodes, representing individual questions in the test (X1 - X26_3). Links between nodes represent relationships in the student model. Each link says there is a connection between a particular skill and a chance to score a certain amount of points for an answer to the question it connects to. Each node has an associated conditional probability table which defines these relationships. Skills are hidden (unobserved) variables meaning that in no case are we are able to obtain their real values. This is caused by the fact that a student's skill can never be directly measured.

Example 1.1. We have created an example which is used throughout this introductory part. Based on Figure 1, there is a test in mathematics. This test has 26 questions and some of them have sub-questions. We have identified eight different skills which a student should have in order to complete the test correctly. Thus, one question (X6) gives an analytical expression of a circle written in parametric form in its text. The task is to plot the circle. In order to complete this task the student must

- know how to work with analytical geometry;
- know how to work with equations and;
- know how to plot in a chart.

These skills are represented by nodes S1, S4, and S5. There are connections from these skill nodes and the question node X6 establishing the causality. With different levels of each skill, there is a varied chance that the student will answer correct or incorrect. If s/he knows the first two parts, s/he is likely to get points for correctly finding the center's

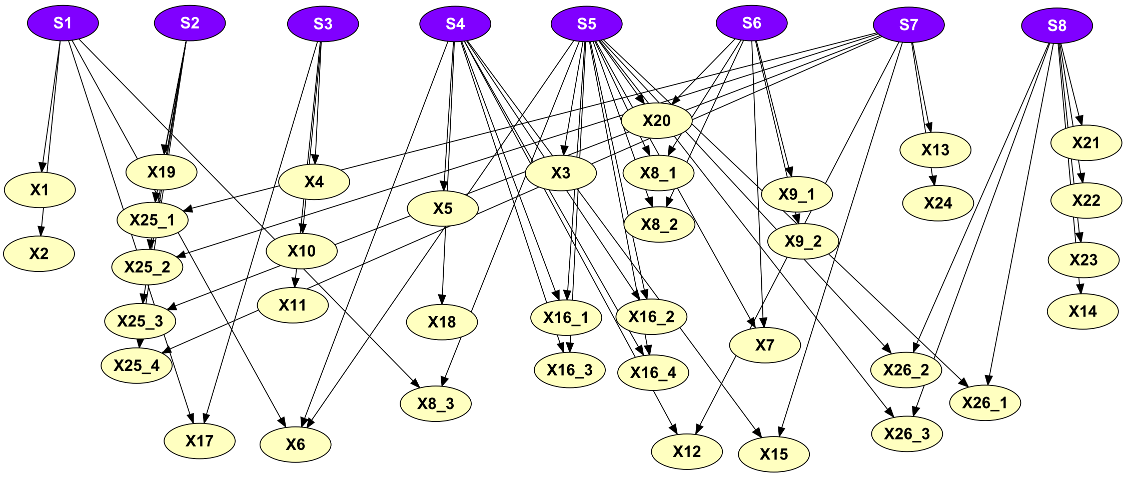


Figure 1. An example of a Bayesian network used for CAT.

coordinates and the diameter of the circle, but not for plotting it. There is also always a risk of making a computation error. Each combination of the skills has a defined probability of scoring a certain amount of points (one probability for each point option). A combination of all possible point scores based on all possible skill combinations creates a conditional[1] probability table defining the relationship, which is an inseparable part of the BN definition.

1.3. *Monotonicity*

The definition of monotonicity is present, for example, in Plajner and Vomlel (2017). At this point, we provide an informal introduction to allow the reader to go through the following text without the need to study the formal definition.

Monotonicity, in terms of Bayesian network features, is a behavior of the causal relationship. Consider a causal connection between two nodes A and B where A is the parent of B. It means that the probability on B is conditioned by the state of A. If there is no monotonicity then there is no restriction on the behavior of this connection in a general BN. On the other hand, in a case that the monotonicity is present, increasing the state value of A increases the expected result of B. In other words, a higher value of A yields a higher chance of higher state of B. There is of course the possibility of an opposite/negative effect, but the rationale would be the same without the loss of generality.

This behavior is commonly observed in the real world. For example, studies prove (e.g. in Cornfield et al. (2009)) that a higher number of cigarettes a person smokes per day causes a higher risk of lung cancer. While modeling such a causal connection, it is very reasonable to expect such behavior in the model as well. Users of the probability model are usually very unsatisfied and puzzled if they observe different behavior.

In the case of student models, simply said, monotonicity condition means that a higher level of student skills leads to a better result. This condition sounds very reasonable, although there is generally no insurance that a BN model will learn it from the provided data. The main cause of not learning this feature (if it is actually present) may be a small learning dataset. The learning algorithm then has too few data points to infer monotonicity, which may not be visible in this reduced set. Another issue, connected to the previous one, is the fact that most learning techniques are heuristics – they may therefore end in a non-monotone solution of a local optimum even though the global monotone solution in fact

[1] Probability of obtaining certain amount of points conditioned by the combination of the states of the skills.

exists.

In this thesis, we have expended a great deal of effort to generally describe the effect of monotonicity in terms of CAT and BNs. In the papers we cite, an algorithm is presented to ensure monotone BN models and thus improve the model quality and user satisfaction.

Example 1.2. In terms of Example 1.1, the simplified monotonicity means, for example, that an increase of the level of skill S1 increases the chance of scoring more points in question X6, i.e., if student Adam improves his skill in analytical geometry, the chance of his answering the particular question should not decrease. Or, if there is a second student, Eve, who has a level of skill higher than Adam's, the expected value of her points scored in that question should be at least as high as Adam's (by the definition of monotonicity, an equality is acceptable). To demonstrate the effect, we present an example conditional probability table of the node X6 with parents S1, S4, and S5. In order to simplify the table, all parents are binary, which means that they are either 0 (not having the skill) or 1 (having the skill). The question has three possible states 0 through 2 (points). The top three rows of the table set the parent configuration while the bottom three rows are probabilities of scoring the respective amount of points. An example selection of two violations of monotonicity is highlighted.

By the blue color in the last row, a higher-ranked state has a lower probability of scoring. This means that improving one skill would actually decrease the chance of being successful. That contradicts the monotonicity condition.

In the second situation, which is colored red, it is necessary to operate with the cumulative distribution function. The monotonicity requires that the chance of scoring at least one point can not get higher while having the higher-ordered level of skills. We compare a setting of having the skill S1 against having skills S1 and S4. The chance of scoring zero or one point is $0.7 (= 0.45 + 0.25)$ in the first case and $0.75 (= 0.4 + 0.35)$ in the second case. Nevertheless, the monotonicity requires that higher-ordered states have this kind of cumulative probability smaller than or equal to the lower-ordered ones. Simply said, the risk of a failure cannot be higher if you are more skilled.

The condition saying that scoring at least one point can not get higher while having the higher level of skills may sound strange at first, but there are two other possible ways how to look at the problem. It means that scoring small amount of points is less probable with better knowledge. Or, from the other side if you get to the highest point value it necessarily means that higher-ordered skill states have to provide at least the same or higher probability of scoring this highest value.

S1	0	0	0	0	1	1	1	1
S4	0	0	1	1	0	0	1	1
S5	0	1	0	1	0	1	0	1
$X6 = 0$	0.85	0.75	0.35	0.10	0.45	0.40	0.35	0.10
$X6 = 1$	0.13	0.20	0.25	0.35	0.25	0.35	0.30	0.20
$X6 = 2$	0.02	0.05	0.40	0.55	0.30	0.25	0.35	0.70

Table 1. An example CPT where monotonicity conditions are violated.

1.4. *Student Models and Related Work*

As explained in Section 1.1, CAT is a concept of testing student knowledge; it is very dependent on the underlying student model. In practice, there have been many variations of student models and many different approaches; IRT, which is briefly mentioned above, is one common option in the domain. Nevertheless, research (e.g., Almond and Mislevy (1999); Almond et al. (2015)) shows that other models may be useful for solving this task as well. At the beginning of our research project, we decided to use BNs in order to model a student. In the study Plajner and Vomlel (2016a) for this dissertation thesis we explored different options of models, and performed experimental evaluation of their capabilities. It has proved useful to utilize the power of BNs to model students.

There are many benefits of BNs being used for creating a student model with them. Two main advantages are

- the graphical representation enabling comprehensible model structure; and
- a good interpretation of learned models, connections, and causalities.

The first benefit is especially important for experts in the application fields who do not have a strong mathematical background. A graphical representation is easy to understand, create and validate. The second benefit is extremely useful when we want to understand the reasoning behind the decision taken by the model, or in order to extract general knowledge from the model. These aspects are especially interesting in comparison with another popular model, namely, Neural networks. A detailed comparison can be found in Plajner and Vomlel (2016b). These types of models have their difficulties and their use is less practical from the point of user interactions. More details about the interpretations of NNs and associated research can, e.g., be found in the recent work Fan et al. (2020).

2. Dissertation Thesis Goals and Time Line

This dissertation thesis sets up a series of goals which have been refined, added to and solved during its lifetime. The main goals we pursue are:

(1) Validate the possibility of using BNs for CAT and test it on a small data set;
(2) Compare different types of student models;
(3) Test CAT approach and student models on a large data set;
(4) Evaluate the benefit of monotonicity in the student model;
(5) Propose and test an algorithm to learn monotone BNs used for CAT;
(6) Generalize the algorithm for monotone learning; and
(7) Connect theoretical and practical discoveries and test them on a large dataset.

The entire time line of the thesis is, in a comprehensive way, displayed in Figure 2. It shows individual activities often leading to a published paper. Table 2 presents a list of these papers as they are displayed in the time line with a short-hand notation. In the time line, we can also find associated research goals as they were addressed. The two first lines grant an overview of two different data sets used for the research.

3. Methodology

The methodology of this thesis is presented in this Section by individual research goals established in Section 2. For each goal, we go through the motivation and the steps taken in order to fulfill it. This is an overview of the entire process and it should clarify the consequences of this thesis. The obtained results can be found in Section 4; and specific details are given in the respective papers which are cited after this introductory part. The connections between goals and resulting papers are displayed in Figure 2.

Type	Reference\Title	Published
[C]	Plajner and Vomlel (2015) Bayesian Network Models for Adaptive Testing	26.11.2015 (CEUR proceedings) Bayesian Modeling Applications Workshop at UAI
[O]	Plajner (2016) Probabilistic Models for Computerized Adaptive Testing	28.01.2016 Study for Dissertation
[O]	Plajner and Vomlel (2016a) Probabilistic Models for Computerized Adaptive Testing: Experiments	28.01.2016 CAT experiments on ArXive
[C]	Plajner and Vomlel (2016b) Student Skill Models in Adaptive Testing	07.09.2016 International conference on Probabilistic Graphical Models
[C]	Plajner and Vomlel (2017) Monotonicity in Bayesian Networks for Computerized Adaptive Testing	11.07.2017 European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty
[C]	Plajner et al. (2017) Question Selection Methods for Adaptive Testing with Bayesian Networks	18.09.2017 Czech-Japan seminar
[C]	Plajner and Vomlel (2018) Gradient Descent Parameter Learning of Bayesian Networks under Monotonicity Restrictions	06.06.2018 Workshop on Uncertainty Processing
[J]	Plajner and Vomlel (2019) Learning Bipartite Bayesian Networks under Monotonicity Restrictions	20.11.2019 International Journal of General Systems vol.49
[O]	Plajner and Vomlel (2020) Monotonicity in Practice of Adaptive Testing	01.09.2020 Finalization and synergies paper on ArXive

Table 2. List of papers corresponding to the research project. Legend: [J] journal paper, [C]: conference paper, [O]: other paper.

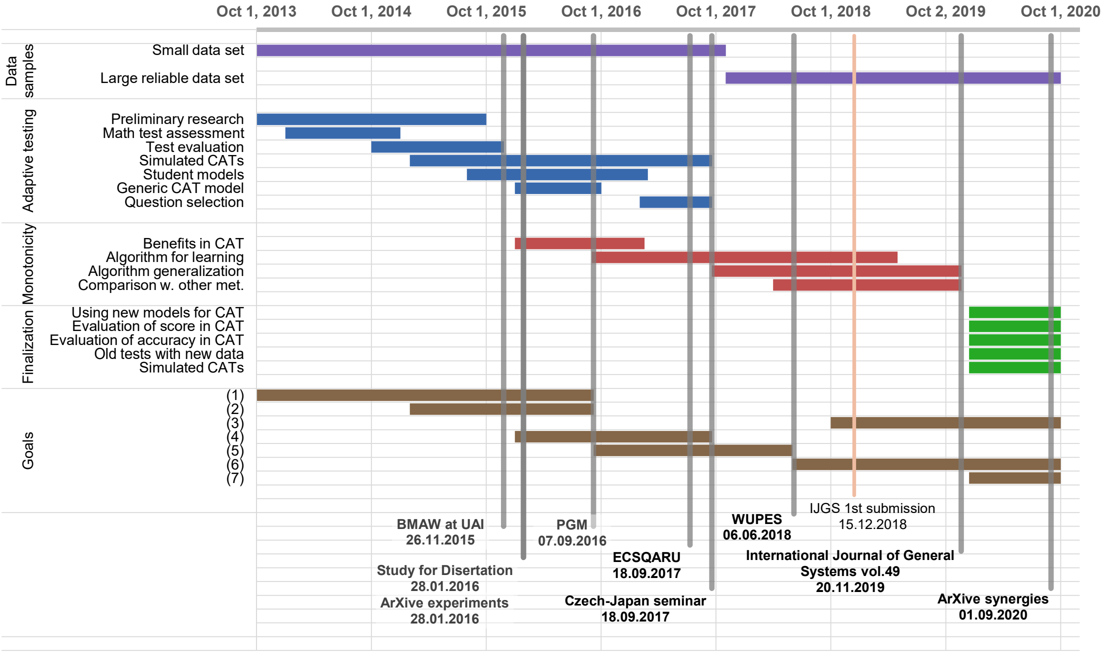


Figure 2. Project time line with activities.

3.1. *Validate the possibility of using BNs for CAT and test it on a small data set*

This task is introduced in the first phase of this thesis. We are following previous research of other scientists in this field, showing that it is appropriate to use BNs for CAT, e.g., Vomlel (2004a,b); Almond et al. (2015). In order to proceed with our own research, we validate this possibility and create a framework which allows us to perform experiments in the this thesis. This task has been split into two distinct parts, namely:

- CAT framework; and
- data collection.

To finish this task we must perform a thorough evaluation of the current state of the art in two domains: BNs and CAT. We decided that it is necessary to obtain a data source which would be suitable for the subsequent research. It has seemed that obtaining data ourselves is a good option. Based on the background in teaching mathematics and analysis of previous research, a domain of mathematics has been selected. To create a valid test we carried out research in the psychometric field which is an important part to correctly solve the problem. We have designed a test in mathematics to be assessed by students of high schools in Prague. A total of 281 students answered in the he test. This data set is a good starting point for initial experiments. Later, we created a new research goal of obtaining a new, larger dataset, to perform more experiments.

The second part of this goal lies in the preparation of a framework for facilitating experiments. We created a framework for BNs in CAT, which is implemented in the programming language R. This framework can be used to train models for CAT from the test data, to run simulations of adaptive testing, and to facilitate the entire process of adaptive testing itself. We implemented the framework and used it to run a series of tests and simulated adaptive testing runs. The focus is on simulated CATs where models are learned from the subset data sample and then the simulated CAT procedure is performed on the remaining data points. Among the answers produced from the testing, we have validated that the approach of using BN for CAT is possible. Throughout work on the thesis, the correctness of the BN approach has repeatedly been proved.

As a result of this effort, a paper describing our early discoveries was presented at a peer-reviewed Bayesian Modeling Applications Workshop, organized parallel with the Uncertainty in Artificial Intelligence 2015 Conference held in Amsterdam, Netherlands. One of the described outputs is also a student model based on the test handed to students which is used for experiments in following goals as well.

This framework provides the functionality which allows us to work with the student model:

- train model parameters with the EM algorithm based on the data;
- train model parameters using an isotonic regression EM algorithm (implemented in the latter stage of the project);
- train model parameters using penalized and regular gradient descent method (implemented in the latter stage of the project);
- obtain the best question to be asked based on the current state of the model and the selected criterion;
- insert the evidence of an answered question (points value); and
- update the model based on the inserted evidence.

Example 3.1. We can, for example, insert the evidence of correctly answered question X17. This question is connected to skill S1. Updating the model with the correct answer increases the estimated level of skill S1. This increase of skill increases the chance of a correct answer to question X6. This sequence of operations is performed during the CAT procedure and after each answer the model gets more precise, it produces better information about the student skills, and predicts their answers more precisely.

3.2. *Compare different types of student models*

After the initial research and assessment of the possibility to use BNs as student models, our solution must be compared with different model types available for the same task. This goal is set to provide sufficient comparison data regarding different possibilities of student modeling in CAT. The BN solution has provided promising results, even though it is necessary to evaluate it in contrast to other solutions. For the comparison, we selected the standard CAT model, Item Response Theory (IRT) and Neural networks which are very popular these days.

We benefit greatly from the work in previous goals, reusing the CAT framework and further generalizing it to provide better functionality. Different models are used for CAT student modeling and simulations. In this goal, we have proven that the selected advanced models (BNs) than the original IRT. Also, student models based on Neural networks did not prove efficient. This observation is most likely connected to the fact that data available to model the students are usually quite small in volume. That means that the benefit of BNs ability to combine expert knowledge with underlying data is very important to provide reliable models. The result of this goal is clearly visible in Figure 3, published in the paper presented at PGM 2016 (see below). The chart shows the average success ratio of predictions of answers in the CAT simulations. In other words, how precise the model is during the course of the testing. In this figure it is clearly visible that IRT and NN methods score results which are worse than those achieved with BN models.

As a part of the work on this goal, we have evaluated and described similarities and differences of using various models for CAT tasks. This activity results in the generalized framework for CAT where different student models can be used while the rest of the system remains unchanged. This concept is described in a paper which was presented at the peer-reviewed International Conference on Probabilistic Graphical Models 2016, which was held in Lugano, Switzerland.

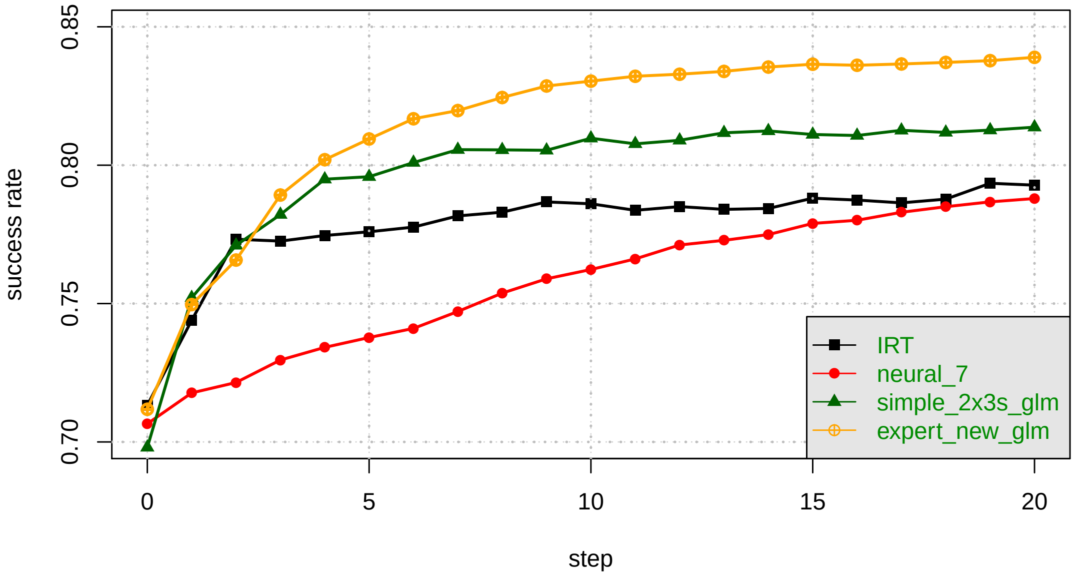


Figure 3. Comparison of answer prediction success in the CAT procedure of different models as published in the paper at PGM 2016.

During the work on this task, we started to observe a special behavior of student models and connections in these models. Specifically, it became clear that monotonicity feature of the models is a feature of great importance and needs to be further evaluated. During our work we noticed that in some cases the model showed unexpected behavior. For example, in case of two students the one who has higher overall level of skills, the chance of answering correctly to some questions was lower. This behavior contradicts natural expectations and reduces the model's credibility. Based on these observations, a new research goal has arisen as monotonicity condition is able to disallow these situations.

3.3. *Test CAT approach and student models on a large data set*

The original data set we collected during the first phase of this thesis was sufficient for initial experiments. Nevertheless, for the large-scale testing and proper continuation of our work, a larger and more reliable data set was required. We managed to obtain data from the official Czech National Final Exam which is taken at the end of high school. This data set is both large and very reliable, as the motivation of students to perform at their best is unhindered. We were able to obtain this set after negotiations with the office in charge of the testing (CERMAT) and the Ministry of Education. The data we obtained are anonymous. Individual answers of each student are included, which is sufficient for all tasks we need to do and to proceed with the research.

We used this larger dataset in subsequent research to test new ideas and topics, and to experimentally evaluate our theoretical advances. At the time of the acquisition of this new dataset, the research was mainly focused on the monotonicity branch. This dataset was used in all papers after 2017. Most of these papers are about monotonicity, with the most important one published in the International Journal of General Systems in 2019. The best overview of CAT simulations with this dataset is presented in the final paper published for open access on ArXive in 2020. In the paper, we can find a chart which is also included here as Figure 4. This Figure shows the error of estimating the final score of a test[2] during the test procedure[3] for different methods of learning the BN model parameters[4]. We can clearly observe that, regardless of the learning method, a fixed version of the test is much slower in error reduction during the testing procedure. This behavior clearly verifies CAT as the correct approach.

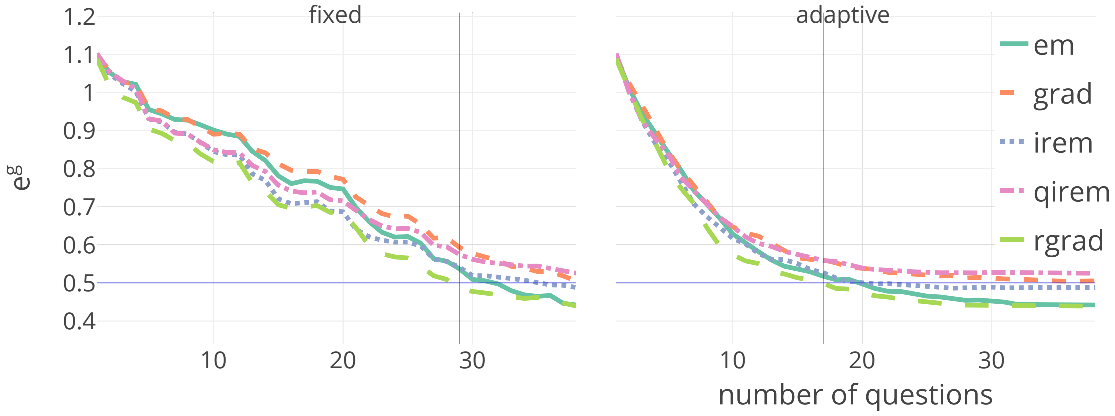


Figure 4. Evolution of the grade prediction error based on skills, fixed and adaptive question selection.

3.4. *Evaluate the benefit of monotonicity in the student model*

While working with adaptive tests and student models, we have noticed that models which hold the monotonicity constraints usually perform better than those which violate these conditions. In other words, models violating monotonicity show an unusual and unexpected behavior. It is very questionable if an increase of a skill decreases the chance of a correct answer to the connected question.

An explanatory example of this behavior is already described in Example 1.2.

BNs generally allow models not to be monotone if the underlying learning data does not have this quality. This problem has initiated a new research path to propose an algorithm which would learn a monotone model even though the data might not exactly point towards it. The first step was taken in the paper presented at the peer-reviewed International Conference on Probabilistic Graphical Models 2016 in Lugano, Switzerland. In this paper, simple generalized linear models were used to learn parameters of BNs. This technique, due to its inner structure, always leads to a monotone solution but it is quite simple and does not allow us to model complex relationships. This initial test showed the benefit of introducing the monotonicity into the BN learning and justified the continuation of this research path.

3.5. *Propose and test an algorithm to learn monotone BNs used for CAT*

After discovering the positive effect of monotonicity in BNs used for CAT, we decided to investigate this topic thoroughly. There are different reasons for data possibly not indicating monotonicity even though it is expected. Mostly it is the volume of data in connection to the parameter space. There are usually a lot of local extremes in the space of model parameters and with a small volume of learning data it is easily possible to obtain a sub-optimal solution because a majority of learning techniques are heuristic.

[2] Y axis where the exact formula is given in the respective paper. The lower the error, the better.

[3] x axis

[4] Legend. Methods are not important in terms of this Example.

These solutions can then easily be non-monotone as well. For example, in the case of the very common EM algorithm, the final solution largely depends on the starting point and the chance of obtaining a non-monotone solution is even higher. Another important aspect is that, because of the small amount of data, the learning output of a monotone model might appear worse than that of a non-monotone one. Then, while comparable on a larger data set, monotonicity provides better results as it is shown in our papers, for example, Plajner and Vomlel (2017).

Motivated by the findings about the monotonicity, we followed research which has been done in this area and have discovered that there is a space to propose an algorithm better than those already available in the community. We derived and tested an algorithm based on the gradient descent, constrained to lead the solution to the monotone area of possible parameters. The main gradient descent optimization function is the log-likelihood of the model given the training data. That means the model parameters should provide the best fit to the data. In addition, this criterion is penalized in order to point the solution towards monotonicity. This algorithm can be used to learn monotone BN parameters.

Monotone models have a large application area, not only in adaptive testing but also in other domains. They are of interest wherever we need to learn models from a small volume of data and/or when we know that the monotonicity conditions apply. This requirement can be based on user expectations, or on common knowledge of general behavior of the reality to be modeled. In these cases monotonicity helps the models to learn better and more reliable parameters by adding the information which is not visible in data. This information is thus used as additional data samples; namely, having them should naturally introduce monotonicity to the model.

The algorithm is mathematically established and described in the paper which was presented at the peer-reviewed Fourteenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty 2017 in Lugano, Switzerland.

Example 3.2. In Example 1.2, we elaborate on the effect of monotonicity and the example presents its violation. This example clearly illustrates how it may not be reasonable not to include monotonicity conditions. If monotonicity were not present, it would be possible to have a lower probability value for answering correctly even with a higher level of skill (or more skills present). This behavior is hard to explain to the community of specialists using the model; teachers in this particular case of a test in mathematics.

3.6. *Generalize the algorithm for monotone learning*

The algorithm for monotone learning has proved successful as it can learn BN models for CAT, which score better in experiments. Nevertheless, the initial design of the algorithm is limited to a specific shape of BNs. The largest issue is that it could work with binary question nodes, i.e., only correct/incorrect answers. In many tests, there is a necessity to use finer resolutions. This goal requires a generalization of the algorithm to be able to learn models which would better suit the needs of CAT.

This generalization was performed and presented at the peer-reviewed Workshops on Uncertainty Processing 2018 in Třeboň, Czech Republic. The paper was well received at the conference and, after the conference, certain improvements were introduced to create a paper which was submitted to the International Journal of General Systems.

A period of the review process followed, with additional modifications and improvements to the paper. The final, very well refined, version of the paper was accepted in November 2019.

The final version of the algorithm allows the user to learn parameters of models which have multiple states both on questions and skills. This is a large step forward in comparison with the previous version; the difference between binary and multiple state nodes is very significant in terms of this algorithm. Moreover, an analysis of the concurrent algorithm irEM (see van der Gaag et al. (2004)) was performed showing the weaknesses and imperfections of this method. A comparison with our proposed method can be found in our papers.

In one of the figures presented in the IJGS paper, we show the KL divergence values between the learned parameters with the aid of different methods and parameters in the model, which was used to create artificial testing data. The first two methods in the plot legend (EM and gradient) are non-monotone, the remaining four are monotone and the last one (res gradient) is our proposed method. In this figure we can see that monotone methods are always better in this domain; especially, for small learning set sizes, our proposed method provides the best results.

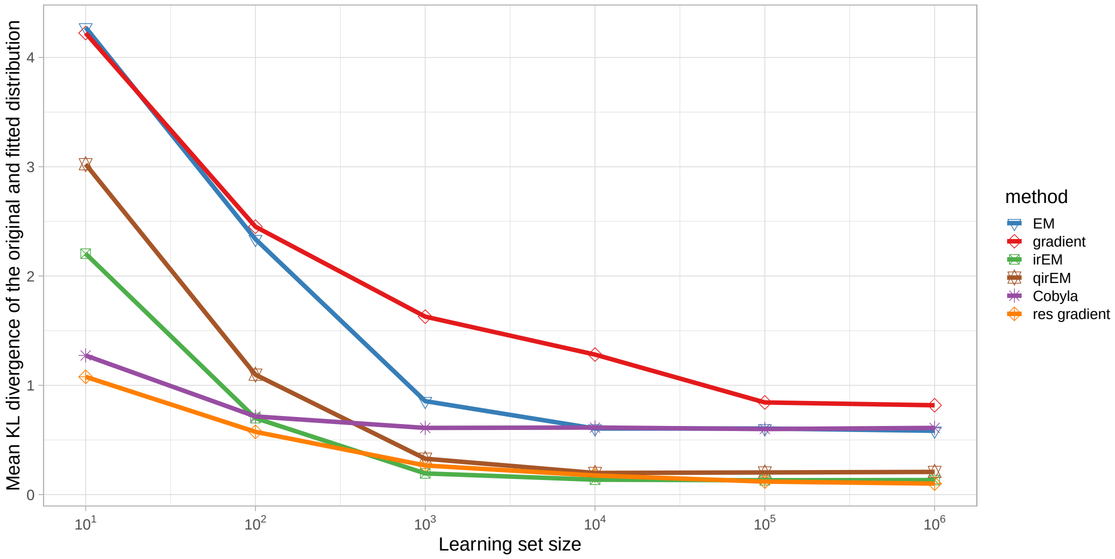


Figure 5. KL divergence based on the learning set size for different learning methods.

3.7. *Connect theoretical and practical discoveries and test them on a large dataset*

Working on this thesis, we have followed two main paths; adaptive testing and parameter learning in terms of monotonicity. These two parts are separate, yet they are always interconnected by experiments and the motivation of the use for CAT. In the first part, we rather focus on the procedure of testing and different types of evaluation, in the second part we work more on the underlying theory of Bayesian networks and parameter learning especially in terms of monotonicity.

As we want to provide an unified view of this thesis as a whole, we have decided to complete a final goal. This goal was set to fill the gaps and answer some questions which were asked during the thesis and remained unanswered. From the previous work, we have the following state:

- An implemented framework for adaptive testing which is mainly used with a smaller dataset;
- the algorithm for monotone parameters learning for BN models;
- the large reliable dataset;
- the question of how precise the adaptive testing is; and
- the question of how to evaluate the score of a test.

The last goal aims at combining these research results together: to use a new dataset to train monotone BN student models, and use these models in the adaptive testing framework in order to get results of the simulated CAT procedure and to evaluate this process. In addition, it is necessary to answer the question of how to evaluate the test in terms of scoring, which is necessary for the real application. This last goal is thus a finalization and mutual interconnection of all the previous tasks providing a comprehensive overview of the entire project.

The solution of this task is described in the paper which is published online at the same time as this thesis is finalized. This paper brings an unified view of the topic. It clearly shows the possibilities and options of CAT and the role of monotone BNs used as student models. The process of measuring the score is described and tested, and the results of these tests are promising. The paper wraps up this thesis while it also shows further possible study paths. The larger data set is intensively used in this goal and thoroughly tested. This paper is the final interconnecting element between the two research paths which have been overlapping and influencing each other during the whole thesis, both lying in the intersection of the two main research areas.

4. Results

In this Section we present the main results of this thesis by citing the individual papers in which they were published. All of the papers included in this dissertation are given here exactly in the forms in which they were published.. For each paper, its main ideas, theoretical work, experiments and results are summarized in this Section.

4.1. *Bayesian Network Models for Adaptive Testing*

This paper (Plajner and Vomlel 2015) is published in the proceedings of the peer-reviewed Bayesian Modeling Applications Workshop, which was held at the conference Uncertainty in Artificial Intelligence 2015 in Amsterdam, Netherlands (CEUR Proceedings). This paper is an introductory paper of this thesis. It describes the problem of adaptive testing, used collected dataset and computerized adaptive test procedure. After defining the CAT process and the role of BNs in it, we presented several different BN student models. These models were tested in CAT simulation and compared against each other. Among the most important results of this paper are answers to very important questions

- validation of the possibility to use BNs for CAT;
- overview of the model quality in the adaptive test simulation;,
- discovery that larger and more complex models are better but require more input for learning; and
- the fact that additional student information can quickly be replaced by the answers to the questions and it is not that important to know more about the students.

This paper sets the research path for further research and it validates the usage of student data without additional student information, i.e., anonymous data.

4.2. *Probabilistic Models for Computerized Adaptive Testing*

This paper (Plajner 2016) is published online on the open-access ArXive. It is a study for this dissertation thesis. In the study for the dissertation, we bring a thorough explanation of the concept of CAT. We go through its advantages and disadvantages in great detail. Further, we present the data sample which was collected as a test of mathematics at Czech high schools. We perform a complete evaluation of the test in terms of psychometric analysis. In the final part of this paper, we create student models with the aid of three different techniques

(1) Classical Item Response Theory;
(2) Bayesian networks; and
(3) Neural networks.

These models are described in the paper and the way they are to be used in the adaptive procedure is presented. In this paper, the concluding part is very important as it summarizes a lot of research questions. It creates a summary of the current state of the art and sets the research possibilities to be further addressed in subsequent papers. There is the first mention of the requirement to reduce the space of parameters while learning BN models. This requirement further leads to the research of monotonicity, which is one of the most important benefits of this thesis.

4.3. *Probabilistic Models for Computerized Adaptive Testing: Experiments*

The paper (Plajner and Vomlel 2016a) is closely connected to the previous one (Plajner 2016). The previous paper grants a theoretical overview, research questions and ideas. This paper provides experiments to support the theoretical work done in Plajner (2016). These experiments are performed on the small dataset which was already described. We tested multiple methods during the testing process and we evaluated each of them as well as comparing them.

4.4. *Student Skill Models in Adaptive Testing*

The paper Plajner and Vomlel (2016b) is published in the proceedings of the peer-reviewed Eighth International Conference on Probabilistic Graphical Models in Lugano, Switzerland (Proceedings of Machine Learning Research). In this paper, we presented a generic model which ties different student models together. We set the common framework, which can be used for CAT testing with various model types. Within this framework, we present the question of selecting the methodology in order to facilitate CAT tests. We instantiated the generic model with three different model types: Bayesian networks, Item Response Theory model, and Neural networks. All these models fit the framework and we performed experiments with them. In this paper we also describe the monotonicity property and elaborate on its usefulness during parameter learning. Its connection to all three model types is discussed.

We present results of experiments with different student models from three model types. Some models fulfill the monotonicity condition, while others do not. The results are compared with each other and discussed. The most important scientific output of this paper is the empirical evidence that monotonicity improves model results, which leads to the following papers further examining the monotonicity condition.

4.5. *Monotonicity in Bayesian Networks for Computerized Adaptive Testing*

This paper (Plajner and Vomlel 2017) is published in the proceedings of the peer-reviewed conference European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty 2017 in Lugano, Switzerland (Lecture Notes in Computer Science). In this paper we describe the concept of monotonicity. We elaborate on the fact that even though it is a restriction on the parameter space of the learning procedure, it may, in specific cases, provide better results. We present an algorithm which learns monotone parameters of a BN model. This algorithm is based on the gradient-descent method and restricts the learning so that the resulting model is monotone.

The proposed algorithm is tested on two datasets. One dataset is artificial, while the other contains real data. The artificial dataset is used in order to ensure that the generating model is monotone and we are introducing monotonicity correctly. There is a strong expectation that a student model is monotone, but it cannot be proved for sure as skills are not observed. We compare results obtained by learning monotone models with non-monotone models. The comparison is done also with monotone models learned by another technique, isotonic regression EM. Based on our empirical evaluation, the proposed gradient method performs very well and outperforms the other methods. This statement especially holds for small learning set sizes. With larger learning datasets, differences start to disappear. It is caused by a naturally increased presence of monotonicity in the data sample.

This paper's most significant research output is the algorithm for monotone parameter learning. Nevertheless, this version of the algorithm is tailored to specific model structures, which is limiting. Further research was needed in order to generalize the algorithm.

4.6. *Question Selection Methods for Adaptive Testing with Bayesian Networks*

The paper by Plajner et al. (2017) was written together with an undergraduate student from the Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University. It was presented at Czech-Japan Seminar on Data Analysis and Decision Making 2017 in Třeboň. It is published in its proceedings. This paper answers the question that arose in the initial phase of the dissertation. The quality and speed of the CAT process is very dependent on the way in which the questions are selected. There are many possible concepts for suitable selection of the next question. In this paper we have explored three possibilities. Maximization of

- expected skills entropy reduction;
- expected skills variance; or
- expected question variance.

The first one aims at the reduction of the uncertainty on student skills, the second aims at the best students differentiation possible, and the last one aims at the best chance to predict question results. All three methods were compared with each other and also with the classical sequential method. It was proved that, without any doubt, all three methods perform far better than the sequential one. No clear ordering can be established among the advanced methods themselves. Nevertheless, it was shown that they behave differently and user should always keep that in mind to select the right criterion according to his/her goals.

4.7. *Gradient Descent Parameter Learning of Bayesian Networks under Monotonicity Restrictions*

This paper (Plajner and Vomlel 2018) was presented at the peer-reviewed Workshop on Uncertainty Processing 2018 in Třeboň, Czech Republic. It further generalizes the proposed method of restricted gradient-descent parameter learning. We present this generalization to use the method for both questions and skills with multiple states. This step allows Bayesian networks to be more general and to cover wider application areas. In our case, it allows us to use models and tests which have more possible points for each question to score rather than just correct and incorrect answer.

In this paper we thoroughly describe the monotonicity condition. Next, we derive the gradient-descent algorithm to learn parameters of Bayesian networks. The gradient descent works with the log likelihood criterion, which it tries to maximize in order to achieve the model which best fits the learning data. This gradient descent is further penalized by a function taking into account violations of the monotonicity conditions. That leads the solution of the algorithm towards the monotone area of the parameter space. The penalization has its control parameter, which allows us to set the strength of the penalty and, in fact, sets the necessity of maintaining the monotonicty. In this paper, the algorithm cannot ensure that monotonicity is met and the resulting solution may remain non-monotone, even though it is unlikely.

The algorithm is tested on two datasets. The first one is artificially generated from a synthetic model. This model is created to be monotone and is further used to measure the distance of the solution from the known generating parameters. The second set is an empirical set obtained from the Czech National Final high school exam. Based on this set, an expert BN model is created measuring the skills of the students who took this test. This second dataset is large, and the associated model is complex. In this paper, we measure the score of the learned models by two criteria for the artificial model. First, we measure the ratio between log-likelihood of the generating model and the learned model given the same data points. Second, we measure the mean distance between the learned and generating parameters to see how far the learned model is from the original one. For the empirical dataset, we measure only the log-likelihood of the full data sample because we have no actual parameters to compare with. As the comparison method against the proposed restricted gradient, we use the EM algorithm.

Results of experiments in this paper clearly show the benefit of the monotone principles of learning parameters for the student model. In all observed criteria, the proposed method works better, especially for small learning datasets. In this paper, a comparison with other monotone methods is missing, and there is no theoretical guarantee of a monotone solution for the restricted gradient method.

4.8. *Learning Bipartite Bayesian Networks under Monotonicity Restrictions*

This paper Plajner and Vomlel (2019) was published in the International Journal of General Systems vol.49 in 2019. It is an extension of the previous paper presented at WUPES 2018 (Plajner and Vomlel 2018). It works with the penalized restricted gradient-descent method for monotone parameter learning.

In addition to the previous paper, we use two other possible methods for learning monotone models to provide the proper comparison.

The first one is the isotonic regression EM method (irEM) (van der Gaag et al. 2004), which was used before also in our paper Plajner and Vomlel (2017). To use this method in this paper, we had to generalize it to work with multiple state variables (both parent and child nodes); it was originally defined only for binary variables by the authors.

The second method is the bounded non-linear optimization (Powell 1994), which performs the search for the optimal solution only in the polytope defined by the monotonicity conditions.

Another addition to the previous paper is an introduction and theoretical description of the modification ensuring a monotone solution while using our restricted gradient method. This modification is important whenever there is a need of strictly ensuring monotonicity, which was previously impossible. This also sets up a fair ground for comparisons with other monotone methods that ensure monotone solutions.

This paper provides much more detailed experimental section than the previous one. It runs its experiments on a total of six methods for parameter learning:

- gradient (unrestricted) (Cauchy 1847);

- EM (unrestricted) Lauritzen (1995);
- irEM (van der Gaag et al. 2004);
- qirEM (van der Gaag et al. 2004);
- bounded non-linear method Cobyla (Powell 1994);
- restricted gradient descent (Plajner and Vomlel 2019);

where qirEM is a variation of irEM.

We compare these methods with each other and in this comparison we observe that the proposed restricted gradient method provides significantly better results especially for small learning set sizes. This conclusion is also supported by the a version of the metric, which is Wilcoxon's test, measuring whether the difference in the learned parameters is significantly better in the case of the artificial model. This test also proves that our method provides significantly better results. The next novelty in the experimental evaluation is the way in which we measure the distance of the learned parameters from the generating ones for the artificial model. In this paper, we use KL divergence rather than the simple average distance. This method provides a fairer comparison as it is more sensitive to extremes. This updated methodology of measuring the fit quality also speaks in favor of the proposed method. All the conclusions referred to above hold for both empirical and artificial datasets.

In addition to the results in the previous paragraph, we have also observed and described the behavior of irEM and qirEM methods. The irEM method may end up in the point of the optimization process where the solution oscillates between monotone and non-monotone solutions due to its two-step optimization process. This leads to an increase of computational time as well as possibly blocking the possibility of reaching the optimal solution.

4.9. *Monotonicity in Practice of Adaptive Testing*

The final paper of this collection of papers finalizes this thesis Plajner and Vomlel (2020). It was published in an open-access ArXive repository in 2020. We have shown how Bayesian networks can be used for adaptive testing of students and their skills. Later, we have taken the advantage of monotonicity restrictions in order to learn models that better fit the data. This paper provides a synergy between these two phases, as it evaluates the Bayesian Network models used for computerized adaptive testing and learned with the proposed restricted gradient descent method for monotone parameter learning.

This paper is rather focused on the adaptive part of the procedure, and the tests are performed with the large dataset, Czech National Final high school exam, which was not used for the CAT tests before. Moreover, in this paper, we present the methodology to predict the final student score, its distribution, and its reliability interval. Along with that, we elaborate on the student grading and ranking. These parts are very important in order to use BN models in CAT in practice for two reasons:

- they are actually able to assess the students; and
- they are able to effectively control the flow of the CAT procedure and stop the testing when appropriate.

In experiments presented in this paper, we empirically prove that a test can significantly be shortened while using the concept of CAT without a significant loss of precision. Further we show that models learned with the proposed restricted gradient method perform better in most criteria. There are certain points for which the unrestricted methods perform better, which is caused by the freedom in their parameter learning. Nevertheless, in the overall view, the advantages of using the monotone models are clearly seen. The last observation available in the paper is the view of the evolution and the reduction of the confidence interval of the score prediction. We show how the uncertainty of the prediction decreases in the processes of CAT.

Summarized, this final paper provides a link through the research of this thesis connecting the theoretical advances in parameter learning with a series of elaborated tests and experiments performed on the large dataset.

5. Scientific benefit and future work

This thesis describes the research which was performed in two main areas, Bayesian networks parameter learning and Computerized Adaptive Testing. The research brings practical and theoretical improvements of the current state of the art. From the practical point of view we have established a clear methodology for

- using Bayesian networks in the adaptive testing field;
- predicting the student final score; and
- establishing when and how to stop the CAT process.

These results are beneficial for the work in the field of adaptive testing as well as intelligent tutoring systems and teaching.

From the theoretical point of view, this thesis brings new results in the field of monotonicity and parameter learning of Bayesian networks. We have shown how the monotonicity conditions affect the learning of parameters and the resulting model. We present experimental proofs that applying monotonicity to a model increases its quality. The resulting models have a better accuracy when used for practical tasks. We present a novel method to learn monotone parameters. This method is based on the gradient descent and restricted to maintain monotonicity. The learned models have been extensively tested to prove the benefit of montonicity; these benefits have successfully been confirmed.

There is still open future work on this topic. One open question is further generalization of the proposed method to work with even more general Bayesian networks. The final method only works with a special, but very useful, structure of BNs, bipartite BNs; further generalization to general BNs would enlarge the possible area of use. One possible direction is that of healthcare and diagnostics, where monotone models could be helpful in finding connections between symptoms and diseases. Another possible path for further research is the exploration of monotone effect and testing in domains different from CAT.

It would be interesting to use monotone models in an intelligent tutoring system to provide a predictive platform for students to improve their learning process and allow them to learn more effectively.

List of Publications

Plajner, M. and Vomlel, J. (2015). Bayesian Network Models for Adaptive Testing. In *Proceedings of the Twelfth UAI Bayesian Modeling Applications Workshop*, pages 24–33, Amsterdam, The Netherlands. CEUR-WS.org

Plajner, M. (2016). Probabilistic Models for Computerized Adaptive Testing, arXiv: 1703.09794

Plajner, M. and Vomlel, J. (2016a). Probabilistic Models for Computerized Adaptive Testing: Experiments. Technical report, arXiv: 1601.07929

Plajner, M. and Vomlel, J. (2016b). Student Skill Models in Adaptive Testing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 403–414. JMLR.org

Plajner, M. and Vomlel, J. (2017). Monotonicity in Bayesian Networks for Computerized Adaptive Testing. In Antonucci, A., Cholvy, L., and Papini, O., editors, *ECSQARU 2017*, pages 125–134, Cham. Springer International Publishing

Plajner, M., Magauina, A., and Vomlel, J. (2017). Question Selection Methods for Adaptive Testing with Bayesian Networks. In *Proceedings of the 20th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty CZECH-JAPAN SEMINAR 2017, Pardubice, Czech Republic*, pages 164–175

Plajner, M. and Vomlel, J. (2018). Gradient Descent Parameter Learning of Bayesian Networks under Monotonicity Restrictions. Workshop on Uncertainty Processing (WUPES'18). Publishing House of the Faculty of Mathematics and Physics Charles University

Plajner, M. and Vomlel, J. (2019). Learning bipartite Bayesian networks under monotonicity restrictions. *International Journal of General Systems*, 49(1):88–111

Plajner, M. and Vomlel, J. (2020). Monotonicity in practice of adaptive testing, arXiv: 2009.06981

Bibliography

Aleksander, I. and Morton, H. (1995). *An Introduction to Neural Computing*. Information Systems. International Thomson Computer Press.

Almond, R. G. and Mislevy, R. J. (1999). Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237.

Almond, R. G., Mislevy, R. J., Steinberg, L. S., Yan, D., and Williamson, D. M. (2015). *Bayesian Networks in Educational Assessment*. Statistics for Social and Behavioral Sciences. Springer New York.

Altendorf, E. E., Restificar, A. C., and Dietterich, T. G. (2005). Learning from Sparse Data by Exploiting Monotonicity Constraints. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*.

Anton, F. and Ruef, M. (2010). *Woodcock-Johnson: Mezinárodní Edice II. Uživatelská příručka*. WMF Press.

Cauchy, A. (1847). Methode generale pour la resolution des systemes d'equations simultanees. *C.R. Acad. Sci. Paris*, 25:536–538.

Cornfield, J., Haenszel, W., Hammond, E. C., Lilienfeld, A. M., Shimkin, M. B., and Wynder, E. L. (2009). Smoking and lung cancer: recent evidence and a discussion of some questions*. *International Journal of Epidemiology*, 38(5):1175–1191.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer.

Culbertson, M. J. (2014). *Graphical Models for Student Knowledge: Networks, Parameters, and Item Selection*. PhD thesis, University of Illinois at Urbana.

Culbertson, M. J. (2015). Bayesian Networks in Educational Assessment: The State of the Field. *Applied Psychological Measurement*, 40(1):3–21.

de Campos, C. P., Tong, Y., and Ji, Q. (2008). Constrained Maximum Likelihood Learning of Bayesian Networks for Facial Action Recognition. *Computer Vision – ECCV 2008*, 5304:168–181.

Díez, F. J. and Druzdzel, M. J. (2007). Canonical Probabilistic Models for Knowledge Engineering. Technical report, Research Centre on Intelligent Decision-Support Systems.

Druzdzel, J. and Henrion, M. (1993). Efficient Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 548–553. AAAI Press.

Fan, F., Xiong, J., and Wang, G. (2020). On interpretability of artificial neural networks (preprint). *arXiv:2001.02522v1*.

Feelders, A. (2007). A new parameter Learning Method for Bayesian Networks with Qualitative Influences. In Parr, R. and van der Gaag, L. C., editors, *{UAI} 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, pages 117–124. {AUAI} Press.

Feelders, A. J. and van der Gaag, L. C. (2005). Learning Bayesian Network Parameters with Prior Knowledge about Context-Specific Qualitative Influences. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*.

Hambleton, R. K., Swaminathan, H., and Rogers, H. J. (1991). *Fundamentals of item response theory*, volume 21. SAGE Publications.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY.

Haykin, S. S. (2009). *Neural Networks and Learning Machines.* Number v. 10 in Neural networks and learning machines. Prentice Hall.

Helmstadter, G. C. (1964). *Principles of Psychological Measurement.* New York: Appleton-Century-Crofts.

Hugin (2014). Explorer, ver. 8.0, Comput. Software 2014, http://www.hugin.com.

Ibrahim, J. G., Chen, M.-H., Lipsitz, S. R., and Herring, A. H. (2005). Missing-Data Methods for Generalized Linear Models. *Journal of the American Statistical Association*, 100(469):332–346.

Johnson, S. G. (2018). The NLopt nonlinear-optimization package. Technical report.

Kjærulff, U. B. and Madsen, A. L. (2008). *Bayesian Networks and Influence Diagrams.* Springer.

Kraft, D. (1994). Algorithm 733: TOMPFortran modules for optimal control calculations. *ACM Transactions on Mathematical Software*, 20(3):262–281.

Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19(2):191–201.

Lín, V. (2005). Complexity of Finding Optimal Observation Strategies for Bayesian Network Models. In *Proceedings of the conference Znalosti*, Vysoké Tatry.

Lord, F. M. and Novick, M. R. (1968). *Statistical Theories of Mental Test Scores.* (Behavioral science : quantitative methods). Addison-Wesley.

Madsen, A. L. and Jensen, F. V. (1999). Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113:203–245.

Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data.* PhD thesis, Carnegie Mellon University, Pittsburgh.

Masegosa, A. R., Feelders, A. J., and van der Gaag, L. (2016). Learning from incomplete data in Bayesian networks with qualitative influences. *International Journal of Approximate Reasoning*, 69:18–34.

McCall, W. (1922). *How to measure in education.* Macmillan Company, New York.

Millán, E., Loboda, T., and Pérez-de-la Cruz, J. L. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683.

Millán, E., Trella, M., Pérez-de-la Cruz, J. L., and Conejo, R. (2000). Using Bayesian Networks in Computerized Adaptive Tests. In Ortega, M. and Bravo, J., editors, *Computers and Education in the 21st Century*, pages 217–228. Springer.

Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika*, 59(4):439–483.

Moe, K. C. and Johnson, M. F. (1988). Participants' Reactions To Computerized Testing. *Journal of Educational Computing Research*, 4(1):79–86.

Nielsen, T. D. and Jensen, F. V. (2007). *Bayesian Networks and Decision Graphs (Information Science and Statistics).* Springer.

Olesen, K. G., Kjaerulff, U., Jensen, F., Jensen, F. V., Falck, B., Andreassen, S., and Andersen, S. K. (1989). A Munin Network for the Median Nerve-A Case Study on Loops. *Applied Artificial Intelligence*, 3(2-3):385–403.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc.

Pesonen, E., Eskelinen, M., and Juhola, M. (1998). Treatment of missing data values in a neural network based decision support system for acute abdominal pain. *Artificial Intelligence in Medicine*, 13(3):139–146.

Pine, S. M. and Weiss, D. J. (1978). A Comparison of the Fairness of Adaptive and Conventional Testign Strategies. Technical report, University of Minnesota, Minneapolis.

Plajner, M. (2016). Probabilistic Models for Computerized Adaptive Testing, arXiv: 1703.09794.

Plajner, M., Magauina, A., and Vomlel, J. (2017). Question Selection Methods for Adaptive Testing with Bayesian Networks. In *Proceedings of the 20th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty CZECH-JAPAN SEMINAR 2017, Pardubice, Czech Republic*, pages 164–175.

Plajner, M. and Vomlel, J. (2015). Bayesian Network Models for Adaptive Testing. In *Proceedings of the Twelfth UAI Bayesian Modeling Applications Workshop*, pages 24–33, Amsterdam, The Netherlands. CEUR-WS.org.

Plajner, M. and Vomlel, J. (2016a). Probabilistic Models for Computerized Adaptive Testing: Experiments. Technical report, arXiv: 1601.07929.

Plajner, M. and Vomlel, J. (2016b). Student Skill Models in Adaptive Testing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 403–414. JMLR.org.

Plajner, M. and Vomlel, J. (2017). Monotonicity in Bayesian Networks for Computerized Adaptive Testing. In Antonucci, A., Cholvy, L., and Papini, O., editors, *ECSQARU 2017*, pages 125–134, Cham. Springer International Publishing.

Plajner, M. and Vomlel, J. (2018). Gradient Descent Parameter Learning of Bayesian Networks under Monotonicity Restrictions. Workshop on Uncertainty Processing (WUPES'18). Publishing House of the Faculty of Mathematics and Physics Charles University.

Plajner, M. and Vomlel, J. (2019). Learning bipartite Bayesian networks under monotonicity restrictions. *International Journal of General Systems*, 49(1):88–111.

Plajner, M. and Vomlel, J. (2020). Monotonicity in practice of adaptive testing, arXiv: 2009.06981.

Powell, M. J. D. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis, eds. S. Gomez and J.-P. Hennart (Kluwer Academic: Dordrecht)*, pages 51–67.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Rasch, G. (1960). *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut.

Rasch, G. (1993). *Probabilistic Models for Some Intelligence and Attainment Tests. Expanded Ed.* MESA Press.

Restificar, A. C. and Dietterich, T. G. (2013). Exploiting monotonicity via logistic regression in Bayesian network learning. Technical report, Corvallis, OR : Oregon State University.

Rijmen, F. (2008). Bayesian networks with a logistic regression model for the conditional probabilities. *International Journal of Approximate Reasoning*, 48(2):659–666.

Savicky, P. and Vomlel, J. (2007). Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika*, 43(5):747–764.

Schlesinger, M. I. and Hlaváč, V. (2002). *Ten Lectures on Statistical and Structural Pattern Recognition.* Computational Imaging and Vision. Springer Netherlands.

Stocking, M. L. and Lewis, C. (2000). Methods of Controlling the Exposure of Items in CAT. In van der Linden, W. J. and Glas, G. A., editors, *Computerized Adaptive Testing: Theory and Practice*, pages 163–182. Springer Netherlands, Dordrecht.

Tonidandel, S., Quiñones, M. A., and Adams, A. A. (2002). Computer-adaptive testing: the impact of test characteristics on perceived performance and test takers' reactions. *The Journal of applied* *psychology*, 87(2):320–32.

Urbánek, T., Denglerová, D., and Širček, J. (2011). *Psychometrika.* Portál.

van der Gaag, L., Bodlaender, H. L., and Feelders, A. J. (2004). Monotonicity in Bayesian networks. *20th Conference on Uncertainty in Artificial Intelligence (UAI '04)*, pages 569–576.

van der Gaag, L. C. and de Waal, P. (2006). Multi-dimensional Bayesian Network Classifiers. In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models, PGM 2006*, pages 107–114.

van der Linden, W. J. and Glas, C. A. W. (2000). *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers.

van der Linden, W. J. and Glas, C. A. W., editors (2010). *Elements of Adaptive Testing.* Springer New York, NY.

van der Linden, W. J. and Hambleton, R. K. (2013). *Handbook of Modern Item Response Theory.* Springer New York.

van der Linden, W. J. and Veldkamp, B. P. (2004). Constraining Item Exposure in Computerized Adaptive Testing With Shadow Tests. *Journal of Educational and Behavioral Statistics*, 29(3):273–291.

Vomlel, J. (2004a). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):83–100.

Vomlel, J. (2004b). Buliding Adaptive Test Using Bayesian Networks. *Kybernetika*, 40(3):333–348.

Wainer, H. and Dorans, N. J. (2015). *Computerized Adaptive Testing: A Primer.* Routledge.

Weiss, D. J. and Kingsbury, G. G. (1984). Application of Computerized Adaptive Testing to Educational Problems. *Journal of Educational Measurement*, 21:361–375.

Wellman, M. P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303.

Attached Articles

Bayesian Network Models for Adaptive Testing

Martin Plajner
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4
Prague 8, CZ-182 08
Czech Republic

Jiří Vomlel
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4
Prague 8, CZ-182 08
Czech Republic

Abstract

Computerized adaptive testing (CAT) is an interesting and promising approach to testing human abilities. In our research we use Bayesian networks to create a model of tested humans. We collected data from paper tests performed with grammar school students. In this article we first provide the summary of data used for our experiments. We propose several different Bayesian networks, which we tested and compared by cross-validation. Interesting results were obtained and are discussed in the paper. The analysis has brought a clearer view on the model selection problem. Future research is outlined in the concluding part of the paper.

1 INTRODUCTION

The testing of human knowledge is a very large field of human effort. We are in touch with different ability and skill checks almost daily. The computerized form of testing is also getting an increased attention with the growing spread of computers, smart phones and other devices which allow easy impact on the target groups. In this paper we focus on the Computerized Adaptive Testing (CAT) (van der Linden and Glas, 2000; Almond and Mislevy, 1999).

CAT aims at creating shorter tests and thus it takes less time without sacrificing its reliability. This type of test is computer administered. The test has an accompanied model which models a student (a student model). This model is constructed based on samples of previous students. During the testing the model is updated to reflect abilities of one particular student who is in the process of testing. At the same time we use the model to adaptively select next questions to be asked in order to ask the most appropriate one. This leads to collection of significant information in shorter time and allows to ask less questions. We provide an additional description of the testing process in the Section 4 and more information can be found also in (Millán et al., 2000). It seems that there is a large possibility of applications of CAT in the domain of educational testing (Vomlel, 2004a; Weiss and Kingsbury, 1984).

In this paper we look into the problem of using Bayesian network models (Kjærulff and Madsen, 2008) for adaptive testing (Millán et al., 2010). Bayesian network is a conditional independence structure and its usage for CAT can be understood as an expansion of the Item Response Theory (IRT) (Almond and Mislevy, 1999). IRT has been successfully used in testing for many years already and experiments using Bayesian networks in CAT are also being made (Mislevy, 1994; Vomlel, 2004b).

We discuss the construction of Bayesian network models for data collected in paper tests organized at grammar schools. We propose and experimentally compare different Bayesian network models. To evaluate models we simulate tests using parts of collected data. Results of all proposed models are discussed and further research is outlined in the last section of this paper.

2 DATA COLLECTION

We designed a paper test of mathematical knowledge of grammar school students focused on simple functions (mostly polynomial, trigonometric, and exponential/logarithmic). Students were asked to solve different mathematical problems[1] including graph drawing and reading, calculation of points on the graph, root finding, description of function shape and other function properties.

The test design went through two rounds. First, we prepared an initial version of the test. This version was carried out by a small group of students. We evaluated the first version of the test and based on this evaluation we made changes before the main test cycle. Problems were updated and changed to be better understood by students. Few prob-

[1] In this case we use the term mathematical "problem" due to its nature. In general tests, terms "question" or "item" are often used. In this article all of these terms are interchangeable.

lems were removed completely from the test, mainly because the information benefit of the problem was too low due to its high or low difficulty. Moreover we divided problems into subproblems in the way that:

(a) it is possible to separate the subproblem from the main problem and solve it independently or

(b) it is not possible to separate the subproblem, but it represents a subroutine of the main problem solution.

Note that each subproblem of the first type can be viewed as a completely separate problem. On the other hand, subproblems of the second type are inseparable pieces of a problem.

Next we present an example of a problem that appeared in the test.

Example 2.1. Decide which of the following functions

$$\begin{aligned} f(x) &= x^2 - 2x - 8 \\ g(x) &= -x^2 + 2x + 8 \end{aligned}$$

is decreasing in the interval $(-\infty, -1]$.

The final version of test contains 29 mathematical problems. Each one of them is graded with 0–4 points. These problems have been further divided into 53 subproblems. Subproblems are graded so that the sum of their grades is the grade of the parent problem, i.e., it falls into the set $\{0, \ldots, 4\}$. Usually a question is divided into two parts each graded by at most two points[2]. The granularity of subproblems is not the same for all of them and is a subset of the set $\{0, \ldots, 4\}$. All together, the maximal possible score to obtain in the test is 120 points. In an alternative evaluation approach, each subproblem is evaluated using the Boolean values (correct/wrong). The answer is evaluated as correct only if the solution of the subproblem and the solution method is correct unless there is an obvious numerical mistake.

We organized tests at four grammar schools. In total 281 students participated in the testing. In addition to problem solutions, we also collected basic personal data from students including age, gender, name, and their grades in mathematics, physics, and chemistry from previous three school terms. The primal goal of the tests was not the student evaluation. The goal was to provide them valuable information about their weak and strong points. They could view their result (the scores obtained in each individual problem) as well as a comparison with the rest of the test group. The comparisons were provided in the form of quantiles in their class, school and all participants.

The Table 1 shows the average scores of the grammar schools (the higher the score the better the results). We also computed correlations between the score and average grades from Mathematics, Physics, and Chemistry from previous three school terms. The grades are from the set $\{1, 2, 3, 4, 5\}$ with the best grade being 1 and the worst being 5. These correlations are shown in the Table 2. Negative numbers mean that a better grade is correlated with a better result, which confirms our expectation.

Table 1: Average test scores of the four grammar schools.

GS1	GS2	GS3	GS4	Total
42.76	46.68	46.35	43.65	44.53

Table 2: Correlation of the grades and the test total score.

Mathematics	Physics	Chemistry
-0.60	-0.42	-0.41

3 BAYESIAN NETWORK MODELS

In this section we discuss different Bayesian network models we used to model relations between students' math skills and students' results when solving mathematical problems. All models discussed in this paper consists of the following:

- A set of n variables we want to estimate $\{S_1, \ldots, S_n\}$. We will call them skills or skill variables. We will use symbol S to denote the multivariable $(S_1, \ldots, S_n)$ taking states $s = (s_1, \ldots s_n)$.
- A set of m questions (math problems) $\{X_1, \ldots, X_m\}$. We will use the symbol X to denote the multivariable $(X_1, \ldots, X_m)$ taking states $x = (x_1, \ldots, x_m)$.
- A set of arcs between variables that define relations between skills and questions and, eventually, also inbetween skills and inbetween questions.

The ultimate goal is to estimate the values of skills, i.e., the probabilities of states of variables $S_1, \ldots, S_n$.

3.1 QUESTIONS

The solution of math problems were either evaluated using a numeric scale or using a Boolean scale as explained in the previous section. Although the numeric scale carries more information and thus it seems to be a better alternative, there are other aspects discouraging such a choice. The main problem is the model learning. The more the states the higher the number of model parameters to be learned.

[2] There is one exception from this rule: The first problem is very simple and it is divided into 8 parts, each graded by zero or one point (summing to the total maximum of 8).

With a limited training data it may be difficult to reliably estimate the model parameters.

We consider two alternatives in our models. Variables corresponding to problems' solutions (questions) can either be

- Boolean, i.e. they have two states only 0 and 1 or
- integer, i.e. each X_i takes m_i states $\{1, \ldots, m_i\}$, $m_i \in \mathbb{N}$, where m_i is the maximal number points for the corresponding math problem.

In Section 5 we present results of experiments with both options.

3.2 SKILL NODES

We assume the student responses can be explained by skill nodes that are parents of questions. Skill nodes model the student abilities and, generally, they are not directly observable. Several decisions are to be made during the model creation.

The first decision is the number of skill nodes itself. Should we expect one common skill or should it rather be several different skills each related to a subset of questions only? In the later case it is necessary to specify which skills are required to solve each particular question (i.e. a math problem). Skills required for the successful solution of a question become parents of the considered question.

Most networks proposed in this paper have only one skill node. This node is connected to all questions. The student is thus modelled by a single variable. Ordinarily, it is not possible to give a precise interpretation to this variable.

We created two models with more than one skill node. One of them is with the Boolean scale of question nodes and the other is with the numeric scale. We used our expert knowledge of the field of secondary school mathematics and our experiences gained during the evaluation of paper tests. In these model we included 7 skill nodes with arcs connecting each of them to 1 – 4 problems.

Another issue is the state space of the skill nodes. As an unobserved variable, it is hard to decide how many states it should have. Another alternative is to use a continuous skill variable instead of a discrete one but we did not elaborate more on this option. In our models we have used skill nodes with either 2 or 3 states ($s_i \in \{1, 2\}$ or $s_i \in \{1, 2, 3\}$).

We tried also the possibility of replacing the unobserved skill variable by a variable representing a total score of the test. To do this we had to use a coarse discretization. We divided the scores into three equally sized groups and thus we obtained an observed variable having three possible states. The states represent a group of students with "bad", "average", and "good" scores achieved. The state of this variable is known if all questions were included in the test. Thus, during the learning phase the variable is observed and the information is used for learning. On the other hand, during the testing the resulting score is not known – we are trying to estimate the group into which would this test subject fall. In the testing phase the variable is hidden (unobserved).

3.3 ADDITIONAL INFORMATION

As mentioned above, we have collected not only solutions to problems but also additional personal information about students. This additional information may improve the quality of the student model. On the other hand it makes the model more complex (more parameters need to be estimated). It may mislead the reasoning based solely on question answers (especially later when sufficient information about a student is collected from his/her answers). The additional variables are $Y_1, \ldots, Y_\ell$ and they take states $y_1, \ldots, y_\ell$. We tested both versions of most of the models, i.e. models with or without the additional information.

3.4 PROPOSED MODELS

In total we have created 14 different models that differ in factors discussed above. The combinations of parameters' settings are displayed in the Table 3. One model type is shown in the Figure 1. It is the case of "tf_plus" which is a network with one hidden skill node and with the additional information[3]. Models that differ only by number of states of variables have the same structure. Models with the "obs" infix in the name and "o" in the ID have the skill variable modified to represent score groups rather than skill (as explained earlier in the part 3.2). Models without additional information do not contain the part of variables on the right hand side of the skill variable S_1. Figure 2 shows the structure of the expert models with 7 skill variables in the middle part of the figure.

4 ADAPTIVE TESTS

All proposed models are supposed to serve for adaptive testing. In this section we describe the process of adaptive testing with the help of these models.

At first, we select the model which we want to use. If this model contains additional information variables it is necessary to insert observed states of these variables before we start selecting and asking questions. Next, following steps are repeated:

- The next question to be asked is selected.
- The question is asked and a result is obtained.
- The result is inserted into the network as evidence.

[3] Please note that the missing problems and problem numbers are due to the two-cycled test creation and problems removal.

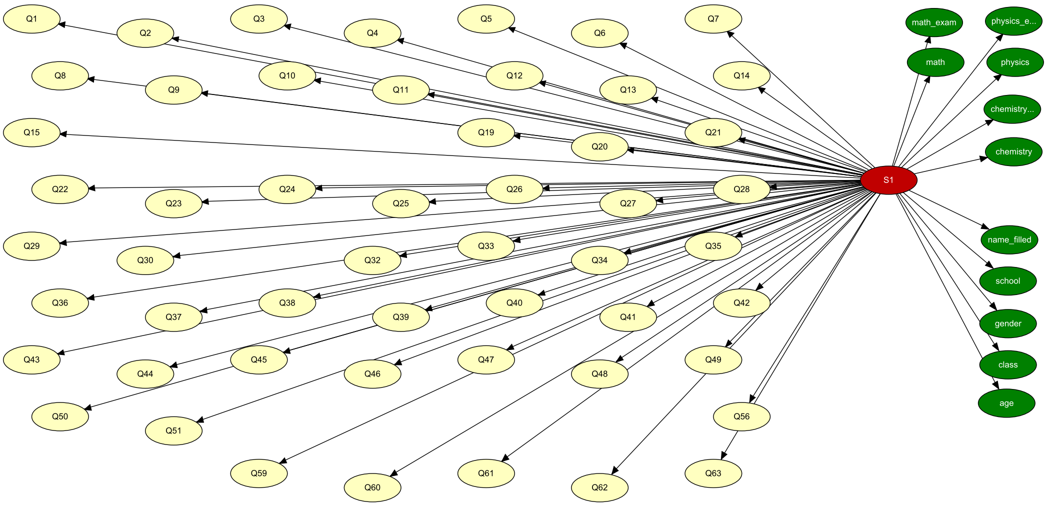


Figure 1: Bayesian network with one hidden variable and personal information about students

ID	Model name	No. of skill nodes	No. of states of skill nodes	Problem variables	Additional info
b2	tf_simple	1	2	Boolean	no
b2+	tf_plus	1	2	Boolean	yes
b3	tf3s_simple	1	3	Boolean	no
b3+	tf3s_plus	1	3	Boolean	yes
b3o	tf3s_obssimple	1	3	Boolean	no
b3o+	tf3s_obsplus	1	3	Boolean	yes
b2e	tf_expert	7	2	Boolean	no
n2	points_simple	1	2	numeric	no
n2+	points_plus	1	2	numeric	yes
n3	points3s_simple	1	3	numeric	no
n3+	points3s_plus	1	3	numeric	yes
n3o	points3s_obssimple	1	3	numeric	no
n3o+	points3s_obsplus	1	3	numeric	yes
n2e	points_expert	7	2	numeric	no

Table 3: Overview of Bayesian network models

- The network is updated with this evidence.
- (optional) Subsequent answers are estimated.

This procedure is repeated as long as necessary. It means until we reach a termination criterion which can be either a time restriction, the number of questions, or a confidence interval of the estimated variables. Each of these criterion would lead to a different learning strategy (Vomlel, 2004b), but because such strategy would be NP-Hard (Lín, 2005). We have chosen an heuristic approach based on greedy entropy minimization.

4.1 SELECTING NEXT QUESTION

One task to solve during the procedure is the selection of the next question. It is repeated in every step of the testing and it is described below.

Let the test be in the state after $s-1$ steps where

$$\mathcal{X}_s = \{X_{i_1} \dots X_{i_n} \mid i_1, \dots, i_n \in \{1, \dots, m\}\}$$

are unobserved (unanswered) variables and

$$e = \{X_{k_1} = x_{k_1}, \dots, X_{k_o} = x_{k_o} | k_1, \dots, k_o \in \{1, \dots, m\}\}$$

is evidence of observed variables – questions which were already answered and, possibly, the initial information. The goal is to select a variable from $\mathcal{X}_s$ to be asked as the next question. We select a question with the largest expected information gain.

We compute the cumulative Shannon entropy over all skill variables of S given evidence e. It is given by the following

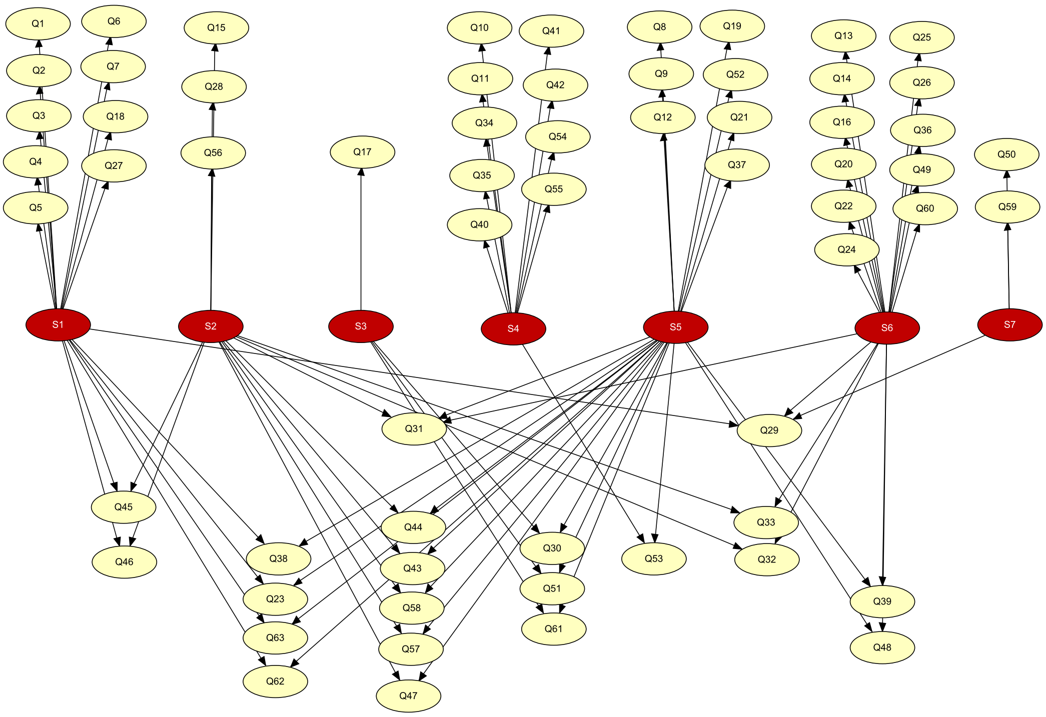


Figure 2: Bayesian network with 7 hidden variables (the expert model)

formula:

$$H(e) \quad = \quad \sum_{i=1}^{n} \sum_{s_i} -P(S_i = s_i|e) \cdot \log P(S_i = s_i|e) \ .$$

Assume we decide to ask a question $X' \in \mathcal{X}_s$ with possible outcomes $x'_1, \ldots, x'_p$. After inserting the observed outcome the entropy over all skills changes. We can compute the value of new entropy for evidence extended by $X' = x'_j$, $j \in \{1, \ldots, p\}$ as:

$$H(e, X' = x'_j) \quad = \\ \sum_{i=1}^{n} \sum_{s_i} \begin{array}{l} -P(S_i = s_i|e, X' = x'_j) \\ \cdot \log P(S_i = s_i|e, X' = x'_j) \end{array} \ .$$

This entropy $H(e, X' = x'_j)$ is the sum of individual entropies over all skill nodes. Another option would be to compute the entropy of the joint probability distribution of all skill nodes. This would take into account correlations between these nodes. In our task we want to estimate marginal probabilities of all skill nodes. In the case of high correlations between two (or more) skills the second criterion would assign them a lower significance in the model. This is the behavior we wanted to avoid. The first criterion assigns the same significance to all skill nodes which seems to us as a better solution. Given the objective of the question selection, the greedy strategy based on the sum of entropies provides good results. Moreover, the computational time required for the proposed method is lower.

Now, we can compute the expected entropy after answering question X':

$$EH(X', e) \quad = \quad \sum_{j=1}^{p} P(X' = x'_j|e) \cdot H(e, X' = x'_j) \ .$$

Finally, we choose a question X^* that maximizes the information gain $IG(X', e)$

$$\begin{aligned} X^* \quad &= \quad \arg\max_{X' \in \mathcal{X}_s} IG(X', e) \ , \ \text{where} \\ IG(X', e) \quad &= \quad H(e) - EH(X', e) \ . \end{aligned}$$

4.2 INSERTION OF THE SELECTED QUESTION

The selected question X^* is given to the student and his/her answer is obtained. This answer changes the state of variable X^* from unobserved to an observed state x^*. Next, the question together with its answer is inserted into the vector of evidence e. We update the probability distributions $P(S_i|e)$ of skill variables with the updated evidence e. We also recompute the value of entropy $H(e)$. The question X^* is also removed from $\mathcal{X}_s$ forming a set of unobserved variables $\mathcal{X}_{s+1}$ for the next step s and selection process can be repeated.

4.3 ESTIMATING SUBSEQUENT ANSWERS

In experiments presented in the next section we will use individual models to estimate answers for all subsequent questions in $\mathcal{X}_{s+1}$. This is easy since we enter evidence e and perform inference to compute $P(X' = x'|e)$ for all states of $X' \in \mathcal{X}_{s+1}$ by invoking the distribute and collect evidence procedures in the BN model.

5 MODEL EVALUATION

In this section we report results of tests performed with networks proposed in Section 3 of this paper. The testing was done by 10-fold cross-validation. For each model we learned the corresponding Bayesian network from $\frac{9}{10}$ of randomly divided data. The model parameters were learned using Hugin's (Hugin, 2014) implementation of the EM algorithm. The remaining $\frac{1}{10}$ of the dataset served as a testing set. This procedure was repeated 10 times to obtain 10 networks for each model type.

The testing was done as described in Section 4. For every model and for each student from the testing data we simulated a test run. Collected initial evidence and answers were inserted into the model. During testing we estimated answers of the current student based on evidence collected so far. At the end of the step s we computed probability distributions $P(X_i|e)$ for all unobserved questions $X_i \in \mathcal{X}_{s+1}$. Then we selected the most probable state of X_i:

$$x_i^* = \arg\max_{x_l} P(X_i = x_l|e) \ .$$

By comparing this value to the real answer x_i' we obtained a success ratio of the response estimation for all questions $X_i \in \mathcal{X}_{s+1}$ of test (student) t in step s

$$\begin{aligned} \mathrm{SR}_s^t &= \frac{\sum_{X_i \in \mathcal{X}_{s+1}} I(x_i^* = x_i')}{|\mathcal{X}_{s+1}|} \text{ , where} \\ I(expr) &= \begin{cases} 1 & \text{if } expr \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The total success ratio of one model in the step s for all test data (N = 281) is defined as

$$\mathrm{SR}_s = \frac{\sum_{t=1}^{N} \mathrm{SR}_s^t}{N} \ .$$

We will refer to the success rate in the step s as to elements of $\mathrm{sr} = (\mathrm{SR}_0, \mathrm{SR}_1, \ldots)$, where SR_0 is the success rate of the prediction before asking any question.

ID/Step	0	1	5	15	25	30
b2	0.714	0.761	0.766	0.778	0.798	0.835
b2+	0.749	0.768	0.768	0.778	0.797	0.829
b3	0.714	0.745	0.776	0.803	0.843	0.857
b3+	0.746	0.754	0.78	0.801	0.831	0.859
b3o	0.714	0.747	0.782	0.8	0.832	0.864
b3o+	0.747	0.761	0.785	0.799	0.83	0.865
b2e	0.715	0.73	0.767	0.776	0.781	0.768
n2	0.684	0.708	0.73	0.713	0.745	0.776
n2+	0.717	0.732	0.731	0.717	0.75	0.778
n3	0.684	0.723	0.745	0.758	0.781	0.79
n3+	0.684	0.724	0.743	0.757	0.77	0.776
n3o	0.686	0.721	0.745	0.751	0.77	0.779
n3o+	0.716	0.729	0.743	0.752	0.773	0.779
n2e	0.684	0.699	0.735	0.738	0.737	0.715

Table 4: Success ratios of Bayesian network models

Table 4 shows success rates of proposed networks for selected steps $s = 0, 1, 5, 15, 25, 30$. The network ID corresponds to the ID from the Table 3. The most important part of the tests are the first few steps, which is because of the nature of CAT. We prefer shorter tests therefore we are interested in the early progression of the model (in this case approximately up to the step 20). During the final stages of testing we estimate results of only a couple of questions which in some cases may cause rapid changes of success rates. Questions which are left to the end of the test do not carry a large amount of information (because of the entropy selection strategy). This may be caused by two possible reasons. The first one is that the state of the question is almost certain and knowing it does not bring any additional information. The second possibility is that the question connection with the rest of the model is weak and because of that it does not change much the entropy of skill variables. In the latter case it is also hard to predict the state of such question because its probability distribution also does not change much with additional evidence.

From an analysis of success rates we have identified clusters of models with similar behavior. For models with integer valued questions and also for models with Boolean questions three clusters of models with similar success ratio emerged:

- models with skill variable of 3 states,
- models with skill variable of 2 states, and
- the expert model.

We selected the best model from each cluster to display success ratios SR_s in steps s in Figure 3 for Boolean questions and in Figure 4 for integer valued questions. We made the following observations:

- Models with the skill variable with 3 states were more successful.

	b2+	b3	b2e	n2+	n3	n2e
AZT	0.5	1.9	7.5	18.1	47.4	81.7
AS	0.002	0.006	0.026	0.047	0.081	0.121

Table 5: Avg. number of zeros/sparsity of different models

- Models with skill variable with 2 states were better at the very end of tests, but this test stage is is not very important for CAT since the tests usually terminates at early stages as explained above.
- The expert model achieved medium quality prediction in the middle stage but its prediction ability decreases in the second half of the tests.

We would like to point out that the distinction between models is basically only by differences of skill variables used in the models. The influence of additional information is visible only at the very beginning of testing. As can be seen in the Table 4 "+" models are scoring better in the initial estimation and then in the first one. After that both models follow almost the same track. In the late stages of the test, models with additional information are estimating worse than their counterparts without information. It suggests that models without additional information are able to derive the same information by getting answers to few questions (in the order of a couple of steps).

It is easy to observe that the expert model does not provide as good results as other models especially during the second half of the testing. As was stated above the second part of the testing is not as important, nevertheless we have investigated causes for these inaccuracies. The main possible reason for this behavior may be the complexity of this type of model. With seven skill nodes and various connections to question nodes this model contains a significantly higher number of parameters to be fitted. It is possible that our limited learning sample leads to over-fitting. We have explored the conditional probability tables (CPTs) of models used during cross-validation procedure to see how sparse they are. Our observation is shown in the Table 5. The number AZT is the average of the total number of zeros in cross-validation models for the specific configuration and AS is the average sparsity of CPTs rows in these models. We can see that in the same type of scales (Boolean or numeric) the sparsity of expert models is significantly higher. This can be improved by increasing data volume or decreasing the model's complexity. This finding is consistent with the above explained possible cause for inaccuracies. In addition we can observe that there is also an increase in sparsity when more skill variables states are introduced. It seems to us as a good idea to further explore the space between one skill variable and seven skill variables as well as the number of their states to provide a better insight into this problem and to draw out more general conclusions.

In Figures 5 and 6 we compare which questions were often selected by the tested models at different stages of the tests. Figure 5 is for Boolean questions and Figure 6 for integer valued questions. Only three models (the same as for success ratio plots) were selected because other models share common behavior with others from the same cluster. On the horizontal axis there is the step when the question was asked, on the vertical axis are questions by their ID. The darker the cell in the graph the more tests used the corresponding variable in the corresponding time. Even though it provides only a rough presentation it is possible to notice different patterns of behavior. Especially, we would like to point out the clouded area of the expert model where it is clear that the individual tests were very different. Expert models are apparently less sure about the selection of the next question. This may be caused by a large set of skill variables which divide the effort of the model into many directions. This behavior is not necessarily unwanted because it provides very different test for every test subject which may be considered positive, but it is necessary to maintain the prediction success rates.

6 CONCLUSION AND FUTURE RESEARCH

In this paper we presented several Bayesian network models designed for adaptive testing. We evaluated their performance using data from paper tests organized at grammar schools. In the experiments we observed that:

- Larger state space of skill variables is beneficial. Clearly, models with 3 states of the hidden skill variable behave better during the most important stages of the tests. Test with hidden variables with more than 3 states are still to be done.
- Expert model did not score as good as simpler models but it showed a potential for its improvements. The proposed expert model is much more complex than other models in this paper and probably it can improve its performance with more data collected.
- Additional information provided improves results only during the initial stage. This fact is positive because obtaining such additional information may be hard in practice. Additionally, it can be considered politically incorrect to make assumption about student skills using this type of information.

In the future we plan to explore models with one or two hidden variables having more than three states, expert models with skill nodes of more than 2 states, and try to add relations between skills into the expert model to improve its performance. We would also like to compare our current results with standard models used in adaptive testing like the Rash and IRT models.

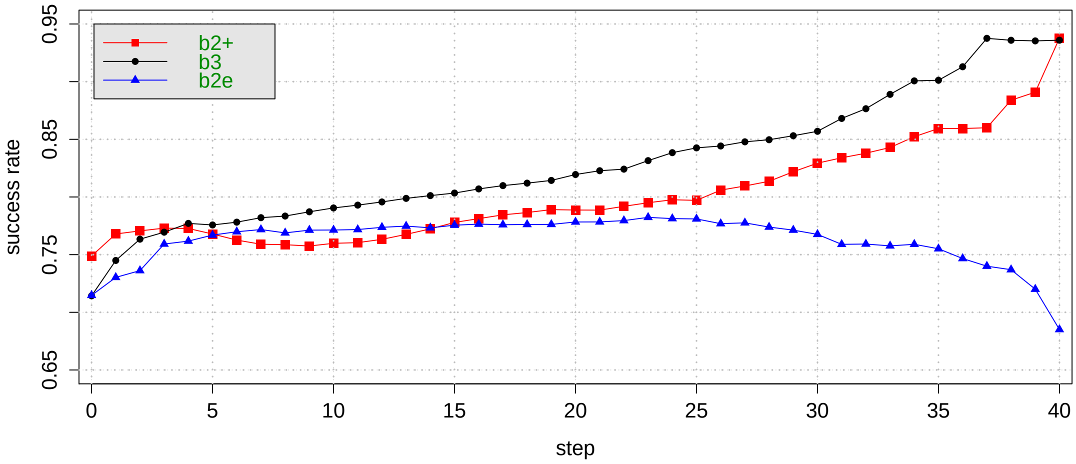


Figure 3: Success ratios for models with Boolean questions

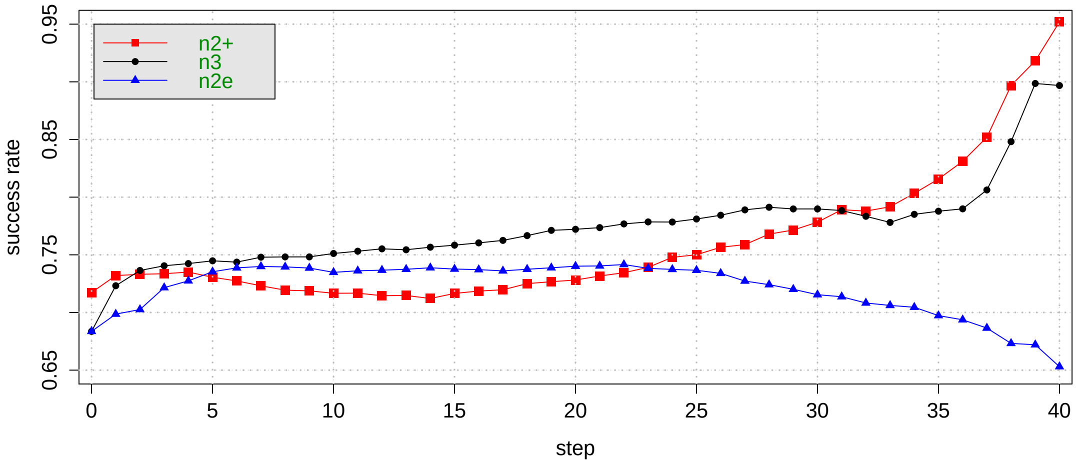


Figure 4: Success ratios for models with integer valued questions

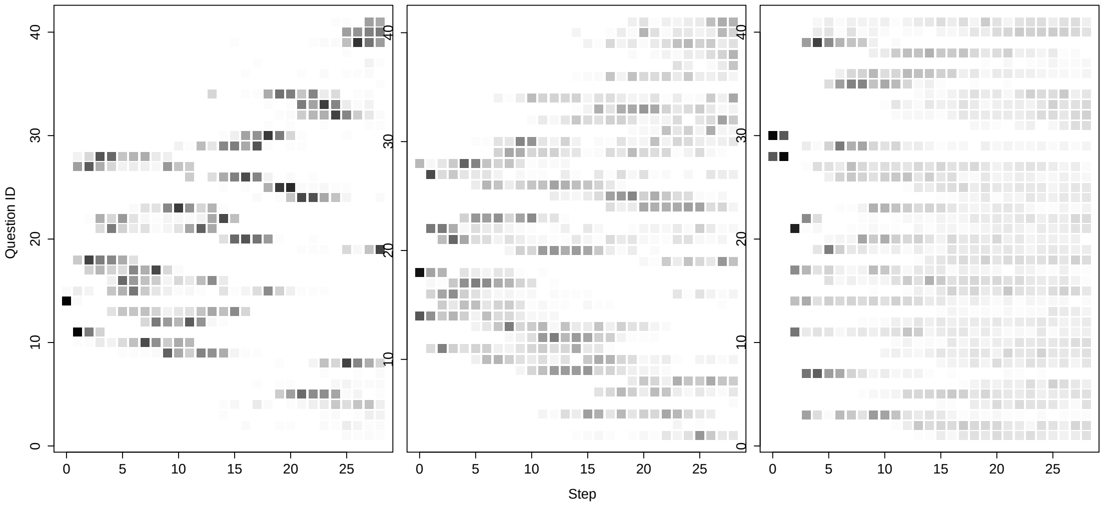


Figure 5: Relative occurrence of questions (on vertical axis) into models with Boolean scale. From left “b2+”,“b3”,“b2e”

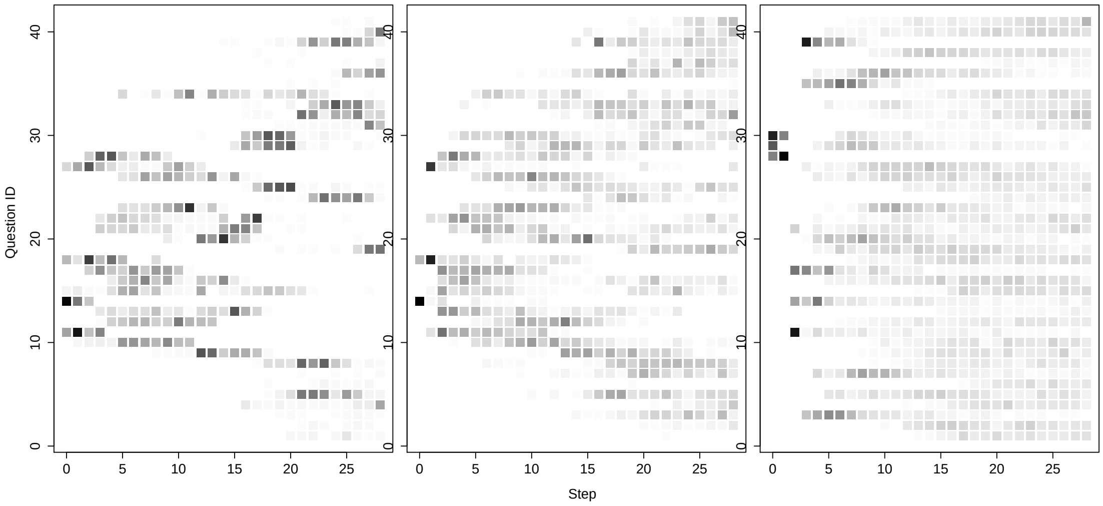


Figure 6: Relative occurrence of questions (on vertical axis) into models with numeric scale. From left “n2+”,“n3”,“n2e”

Acknowledgements

The work on this paper has been supported from GACR project n. 13-20012S.

References

Almond, R. G. and Mislevy, R. J. (1999). Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237.

Hugin (2014). Explorer, ver. 8.0, comput. software 2014, http://www.hugin.com.

Kjærulff, U. B. and Madsen, A. L. (2008). *Bayesian Networks and Influence Diagrams*. Springer.

Lín, V. (2005). Complexity of finding optimal observation strategies for bayesian network models. In *Proceedings of the conference Znalosti*, Vysoké Tatry.

Millán, E., Loboda, T., and Pérez-de-la Cruz, J. L. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683.

Millán, E., Trella, M., Prez-de-la Cruz, J., and Conejo, R. (2000). Using bayesian networks in computerized adaptive tests. In Ortega, M. and Bravo, J., editors, *Computers and Education in the 21st Century*, pages 217–228. Springer.

Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika*, 59(4):439–483.

van der Linden, W. J. and Glas, C. A. (2000). *Computerized Adaptive Testing: Theory and Practice*. Springer.

Vomlel, J. (2004a). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):83–100.

Vomlel, J. (2004b). Buliding Adaptive Test Using Bayesian Networks. *Kybernetika*, 40(3):333–348.

Weiss, D. J. and Kingsbury, G. G. (1984). Application of Computerized Adaptive Testing to Educational Problems. *Journal of Educational Measurement*, 21:361–375.

Czech Technical University in Prague

Faculty of Nuclear Sciences and Physical Engineering

Probabilistic Models for Computerized Adaptive Testing

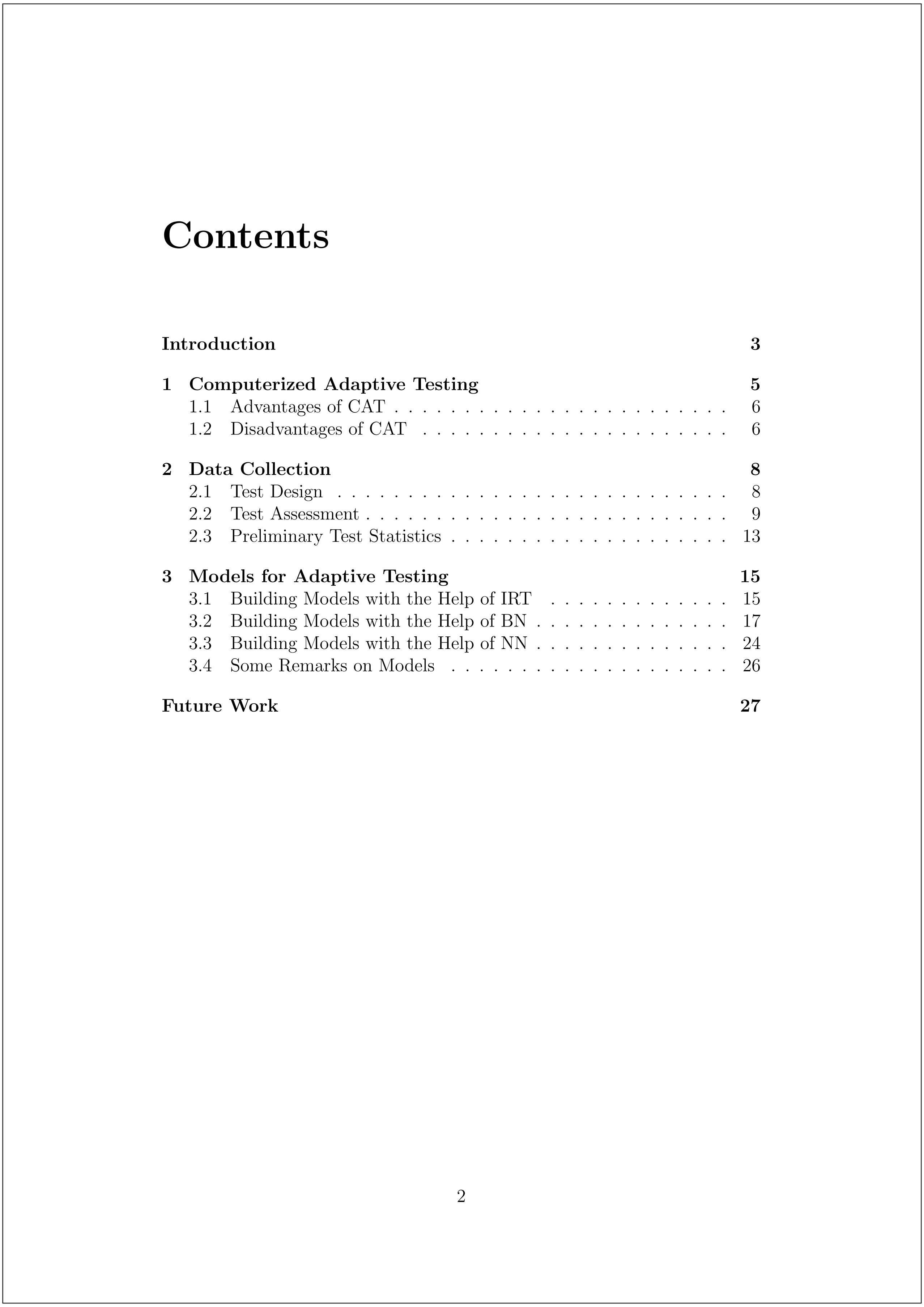
Study for dissertation thesis

Abstract:

In this paper we follow our previous research in the area of Computerized Adaptive Testing (CAT). We present three different methods for CAT. One of them, the item response theory, is a well established method, while the other two, Bayesian and neural networks, are new in the area of educational testing. In the first part of this paper, we present the concept of CAT and its advantages and disadvantages. We collected data from paper tests performed with grammar school students. We provide the summary of data used for our experiments in the second part. Next, we present three different model types for CAT. They are based on the item response theory, Bayesian networks, and neural networks. The general theory associated with each type is briefly explained and the utilization of these models for CAT is analyzed. Future research is outlined in the concluding part of the paper. It shows many interesting research paths that are important not only for CAT but also for other areas of artificial intelligence.

Key words: computerized, adaptive, testing, Bayesian networks, IRT, CAT, neural networks

Author: Ing. Martin Plajner
Supervisor: Ing. Jiří Vomlel, Ph.D.
Year: 2016



Contents

Introduction

Educational testing is an important part of our lives in the modern society. Every person participates in a large number of tests which are used to assess his/her level of knowledge, quality, or skill in a certain domain. There are many different possibilities how to design a test for a specific purpose. The theory of test creation, administration, validation, etc. (in general called psychometrics) is extensive (for example, [Helmstadter, 1964, Lord and Novick, 1968, Rasch, 1960, Mislevy, 1994], and many others). The process of the creation of a good test is long, contains many steps, and can be performed in many different ways. In the classical approach we first identify the target ability (the ability we want to measure). Afterwards, there are repeated cycles of adding new questions, testing the test on a small set of examinees (to see if the questions are measuring the right ability in a correct way), and removing unsuitable questions. In the end we end up with one satisfactory version of the test. This test is fixed in its questions (i.e., every student taking the test will have the same question). This approach does not take into account the individuality of each examinee. It is clear that for a skilled examinee the test will necessarily contain a lot of questions which are too easy and vice versa. The time which is being spent by solving these questions could be used to better explore his/her actual skill level. This can be achieved by asking questions with an appropriate difficulty.

There are banks of questions which are suitable to measure the ability of a student (sometimes they are quite limited – for example if we measure a physical ability there might be a limited number of possible questions – and sometimes they are unlimited – in mathematics we can create as many different problems to solve as we want). Questions for a test are selected from this bank. If we use one set of questions for every test there will be a lot of possibly good questions which are never asked (those which remained in the question bank unselected). The same set of questions for every test also, in some cases, encourages cheating (which of course can usually be solved by other methods, but it requires additional steps).

Some tests do not follow the outline explained in the paragraph above and tries to utilize the whole bank of questions. One way of doing that is for example used in the Czech driving license test[1]. It is a computer test where 25 questions are randomly selected from approximately a thousand of possible questions. This approach negates the possibility of learning all the questions by heart as well as cheating by looking into your neighbor's sheet. Another

[1] http://www.mdcr.cz/cs/Silnicni_doprava/etesty/etesty.htm(Czech language)

test in a similar manner is done by the Faculty of Medicine of the Charles University[2] as an entrance exam test. Questions for the entrance exam are selected from a set (book) of possible questions for each test (i.e., question bank). Several versions of a test are prepared for every entrance exam session. Both of these selection processes (driver's license and entrance exam) remove some complications mentioned above but produce new problems. In the driver's license test, where the question selection is done automatically by the system, it is hard to ensure the overall difficulty of each test will be approximately the same. Cases where a lot of easy questions, or on the contrary a lot of difficult questions, is selected might occur. In the approach of the medical faculty this unfair combination can be avoided by a careful test composition. The test composition is done by hand by specialists. These specialists sometimes have shifted notion of the difficulty of individual questions (some things they may think of as very easy are actually hard for young students). There is also definitely a lot of effort and time involved in the preparation of every entrance exam round.

Computerized Adaptive Testing (CAT) offers a way to overcome some of the limitations given by the classical testing approach. The examinee is answering questions presented to him/her by a computer system. This system is centered on a student model. There are many ways to construct a student model. One way is a model composition by experts. Another is to construct the model from a data set of many previously tested examinees. These examinees have to be tested without the adaptive approach to obtain a basis for the model creation. Afterwards, the model can be further updated and extended with new cases even while being in use.

During the course of testing the student model is updated to reflect abilities of the tested student and as a part of that process an estimate of student's level of knowledge is updated as well. This provides us an actual estimate of student's abilities in every phase of testing. At the same time the model is used to select the next question. The next selected question is the most appropriate one. An appropriate question suits certain criteria, usually providing the best information about the student at the current stage of testing. Questions are selected from a bank of questions. This bank can be similar to a question bank for the classical test. Adaptive testing is performed until a criterion is reached. There is a variety of possible criteria. Usually we want to stop the test when the confidence of the estimate of student's skill is above a certain significant value. Other practical limitations might affect this criterion such as the total time of the test or the number of asked questions. The adaptive testing concept brings many advantages but also some disadvantages over the classical testing approach. These aspects are detailed in the following chapters. Further we present three different model types for CAT. Experimental results of these models with empirical data are in an associated paper [?]

[2] `http://www.lf1.cuni.cz/prijimaci-rizeni`(Czech language)

Chapter 1

Computerized Adaptive Testing

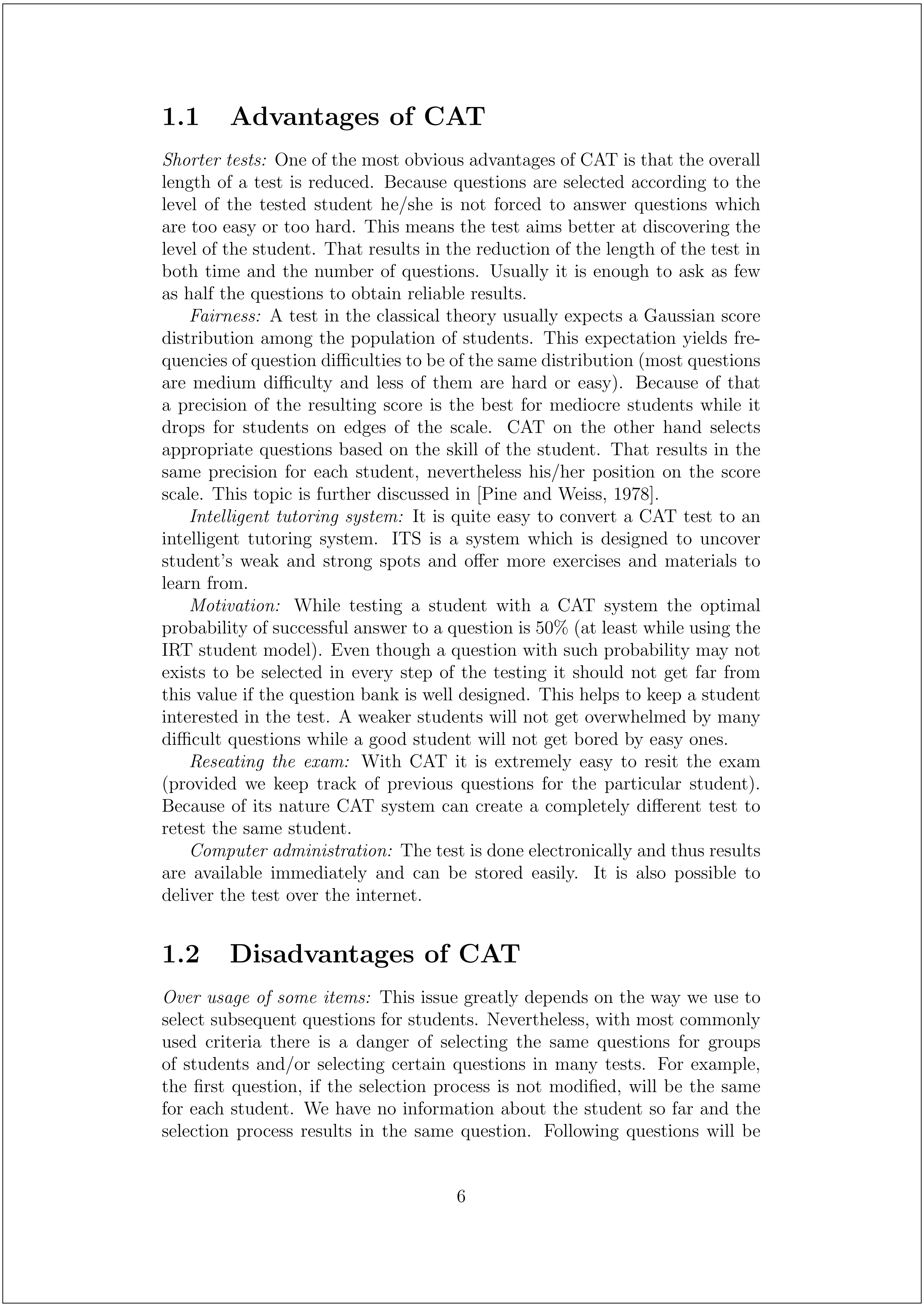
This chapter introduces the concept of Computerized Adaptive Testing (CAT) and summarizes its advantages and disadvantages.

CAT is a concept of testing which is getting large scientific attention for about two decades [van der Linden and Glas, 2000, Wainer and Dorans, 2015, van der Linden and Glas, 2010]. With CAT we build computer administered and computer controlled tests. The computer system is selecting questions for a student taking the test and evaluating his/her performance.

The process can be divided into two phases: model creation and testing. In the first one the student model is created while in the second one the model is used to actually test examinees. There are many different model types [Almond and Mislevy, 1999, Culbertson, 2014, Cowell et al., 1999] which can be used for adaptive testing. In this work we are going to cover Item Response Theory (IRT), which is a model regularly used for CAT, Bayesian and neural networks (BNs and NNs), which are both models commonly used in many areas of artificial intelligence for a large variety of tasks. We will pay closer attention to these models later on but regardless of the model we choose the testing part follows always the same scheme. With the prepared and calibrated model, CAT testing repeats following steps.

- The next question to be asked is selected.
- This question is asked and an answer is obtained.
- This answer is inserted into the model.
- The model (which provides estimates of the student's skills) is updated.
- (optional) Answers to all questions are estimated given the current estimates of student's skills.

This procedure is repeated until we reach a termination criterion. There are many different stopping criteria. It can be a time restriction, the number of questions, or a confidence interval of the estimated variables (i.e., reliability of the test).



1.1 Advantages of CAT

Shorter tests: One of the most obvious advantages of CAT is that the overall length of a test is reduced. Because questions are selected according to the level of the tested student he/she is not forced to answer questions which are too easy or too hard. This means the test aims better at discovering the level of the student. That results in the reduction of the length of the test in both time and the number of questions. Usually it is enough to ask as few as half the questions to obtain reliable results.

Fairness: A test in the classical theory usually expects a Gaussian score distribution among the population of students. This expectation yields frequencies of question difficulties to be of the same distribution (most questions are medium difficulty and less of them are hard or easy). Because of that a precision of the resulting score is the best for mediocre students while it drops for students on edges of the scale. CAT on the other hand selects appropriate questions based on the skill of the student. That results in the same precision for each student, nevertheless his/her position on the score scale. This topic is further discussed in [Pine and Weiss, 1978].

Intelligent tutoring system: It is quite easy to convert a CAT test to an intelligent tutoring system. ITS is a system which is designed to uncover student's weak and strong spots and offer more exercises and materials to learn from.

Motivation: While testing a student with a CAT system the optimal probability of successful answer to a question is 50% (at least while using the IRT student model). Even though a question with such probability may not exists to be selected in every step of the testing it should not get far from this value if the question bank is well designed. This helps to keep a student interested in the test. A weaker students will not get overwhelmed by many difficult questions while a good student will not get bored by easy ones.

Reseating the exam: With CAT it is extremely easy to resit the exam (provided we keep track of previous questions for the particular student). Because of its nature CAT system can create a completely different test to retest the same student.

Computer administration: The test is done electronically and thus results are available immediately and can be stored easily. It is also possible to deliver the test over the internet.

1.2 Disadvantages of CAT

Over usage of some items: This issue greatly depends on the way we use to select subsequent questions for students. Nevertheless, with most commonly used criteria there is a danger of selecting the same questions for groups of students and/or selecting certain questions in many tests. For example, the first question, if the selection process is not modified, will be the same for each student. We have no information about the student so far and the selection process results in the same question. Following questions will be the same for groups of students. These groups shrink with more answered questions as the number of possible combinations of answers increase. This behavior can be reduced by having a large question bank containing many different questions with similar properties (i.e., difficulty). Moreover, it is possible to modify the selection process to ensure a wider spread of selected questions over the question bank (with the cost of decreased precision).

Initial data collection: Prior to starting a test using CAT it is necessary to obtain a large set of data (full test results) from a representative population. This data is used to create and calibrate the student model used for testing. Results used for this creation need to come from a full length tests (optionally it would be possible, but not preferable, to have a several sub-tests). This means students participating in the initial testing are required to fill answers to many items. Another option is to build a model with the help of an expert in the field but even this approach is time consuming.

Computer administration: In order to test students it is necessary to create an environment for such testing on the computer. Also it is necessary for students to have access to a computer rather than having just a pen.

Results perception: Last but not least, there might be some issues with the perception of results by students taking the test. It may be hard to explain to them and for them to comprehend the fact, that even though they got completely (or partly) different questions they are sorted on the same scale (sometimes even obtaining the same score). It may seem unfair and incomparable because of the question selection process. The feeling may be the same as with the the Czech driving license test mention in the introduction but there the selection is done at random. In reality CAT tests tend to be more fair the regular paper-pen tests [Moe and Johnson, 1988, Tonidandel et al., 2002].

Chapter 2

Data Collection

To support the creation of a student model we have collected empirical data. We designed a paper test of mathematical knowledge of grammar school students. The test focuses on simple functions (mostly polynomial, trigonometric, and exponential/logarithmic). Students were asked to solve various mathematical problems[1] including graph drawing and reading, calculating points on the graph, root finding, describing function shapes and other function properties.

2.1 Test Design

When we were creating the test for the data collection we performed several steps. First, we prepared an initial version of the test. This version was carried out by a small group of students and took about 80 minutes to be solved. We evaluated this first version and based on this evaluation we made changes before the main test cycle. It was necessary to limit the time of the test to 45 minutes to make it fit one school lesson. Some questions were removed completely from the test. They were mainly those where the information benefit of the problem was too low due to their high or low difficulty (i.e. only a few students answered them correctly or incorrectly). There was no assumption that all the students should be able to finish all the questions in time which is a usual way to create school test. In this case we were targeting the number of questions to allow the best students to finish just in time. This allowed us to remove less questions than we normally would. Remaining problems were updated and changed to be better understandable. Moreover we divided problems into subproblems in the way that:

(a) it is possible to separate the subproblem from the main problem and solve it independently or

(b) it is not possible to separate the subproblem, but it represents a subroutine of the main problem solution.

[1] In this case we use the term mathematical "problem" due to its nature. In general tests, terms "question" or "item" are often used. In this article all of these terms are interchangeable.

Note that each subproblem of the first type can be viewed as a completely separate problem. On the other hand, subproblems of the second type are inseparable pieces of a problem.

The final version of the test contains 29 mathematical problems. These problems have been further divided into 53 subproblems. Subproblems are graded so that the sum of their grades is the grade of the parent problem, i.e., it falls into the set $\{0, \ldots, 4\}$. Usually a question is divided into two parts each graded by at most two points[2]. The granularity of subproblems is not the same for all of them and is a subset of the set $\{0, \ldots, 4\}$. All together, the maximal possible score in the test is 120 points.
In an alternative evaluation approach, each subproblem is evaluated using the Boolean values (correct/wrong). An answer is evaluated as correct only if the solution of the subproblem and the solution method is correct unless there is an obvious numerical mistake.

We organized tests at four grammar schools. In total 281 students participated in testing. In addition to answers to questions, information about students was collected. This includes mostly some personal factors as gender, age, and grades from mathematics, physics, and chemistry from the recent period. These factors will be used to better differentiate between students and to better predict their performance as well as to verify the validity of the test. The goal of the test was to pinpoint students' weak and strong points and to provide them with valuable information about their skills. Students are able to view their results (the scores obtained in each individual question). We also provide them with a comparison with specific groups of students (their class, school, and all participants). Comparisons are provided in the form of quantiles in the respective group.

2.2 Test Assessment

In the following section we present a psychometric analysis of the test. This kind of analysis should be done for every large scale test. It might not be necessary to perform all actions which are presented below for CAT. Nevertheless we will use these results to compare classical approach and CAT as well as to point out some interesting relations. Moreover it proves that the paper test we used to collect data provide reasonable results.

True scores and reliability
The goal of every test is to measure a certain variable. This variable reflects examinee's skill, ability or level of another quality (some psychiatric test might be measuring person's empathy). In terms of IRT and CAT this variable is a part of the student model described in the Section 3.1. Even in the classical test there is a certain variable. A test is just a tool created to measure this variable. As always, when measuring anything, the measurement process is obstructed with measurement errors. These errors are

[2] There is one exception from this rule: The first problem is very simple and it is divided into 8 parts, each graded by zero or one point (summing to the total maximum of 8 points).

caused by many different factors (the examinee could have a bad day, be ill, guess the answer, or get distracted while solving a single question,...) and it is reasonable to expect them to have a significant influence on the final value. The value obtained as a measurement x of the variable X is called a raw score and is in the form

$$x = \tau + e$$

where τ is the true score and e is an additive error.

There is an obvious question whether the raw score is influenced more by the true score or the error. For many measurements the maximum-likelihood estimator of the error is the variance of many consecutive measurements of the same factor. In our case it proves to be impractical to measure one person multiple times for obvious reasons. It is not as well possible to use the variance of many different examinees as their true values most likely differ. The variability of scores in the data set is then caused by actual differences between examinees (different true scores) as well as errors. It is usually expected that the data set satisfies homoskedasticity condition[3]. With this assumption true scores and errors are statistically independent and thus the observed variance σ_x is a sum of variances of true scores σ_τ and errors σ_e.

$$\sigma_x = \sigma_\tau + \sigma_e$$

The best possible situation is that the variance of the measured variable X is fully modeled by true scores. This situation is very unlikely to happen. To determine the level of the relationship we use the value called reliability[4] which is defined as follows:

$$r_{xx} = \frac{\sigma_\tau}{\sigma_x} = \frac{\sigma_\tau}{\sigma_\tau + \sigma_e}$$

The higher the value the better. Unfortunately variables σ_τ in the nominator as well as σ_e in the denominator of the second fraction are hidden (unobservable) variables and as such we are unable to evaluate their variance. The reliability has to be estimated with a different approach.

There are many possible approaches and we will elaborate more into one of them which is known as Cronbach's alpha coefficient. The idea is that items of the test are measuring the same factor and thus they should correlate with each other. The amount of pair wise correlations for q questions is $k = \frac{q(q-1)}{2}$. All these correlations are put together in the Cronbach's alpha coefficient which can be calculated as

$$r_{xx} \approx \alpha = \frac{n}{n-1}\left(1 - \frac{\sum_{i=1}^{n} \sigma_i^2}{\sigma_t^2}\right)$$

where σ_i is the variance of the ith item of the test, σ_t is the variance of the whole test and n is the number of items in the test. The coefficient should

[3] Homoskedasticity means that the size of an error is not correlated with the size of the measured variable

[4] Note that reliability is a well established and very important property of a test among psychometric comunity

reach high values. According to [Helmstadter, 1964] any value below 0.5 means the test is of no use. Quality results are produced with the coefficient over 0.9.

For our data set (281 students) the following values were calculated:
Cronbach's alpha for the numeric classification: $\alpha = 0.914$
Cronbach's alpha for the Boolean classification: $\alpha = 0.925$
These values show reasonably high reliability of the test.

Normalization and standard scores

It may not be very efficient to use directly the score a student obtained in the test. This score is called the raw score. The question with raw score is that it may not distinguish between individual students as well as it could. For example in case we have 3 results with scores of 20, 40 and 60 respectively. It would seem to us that the gap between the first and the second pair is the same. It definitely is in terms of raw score, but it may not be in terms of real abilities of students. If there are a lot of students who score between 20 and 40 and just a few in the interval between 40 and 60 then the skill gap from the second to the third may not be as wide as it appears. In order to better categorize students, scores are usually normalized. With normalized scores it is easier to evaluate the position of a student in the test for a specialist who is used to work with normalized scores. There are many different types of standard scores and most of them are obtained by a linear transformation of raw scores (note that it means that the order of examinees is not changed by this kind of transformation) by the following formula

$$x' = \mu' + \sigma' \frac{(x - \mu)}{\sigma}$$

Where x' is the transformed score, μ' and σ' are desired mean and variance values of the standardized score, μ and σ are previous mean and variance values and x is the raw score.

To apply these transformations it is required that the raw score belong to the Gaussian distribution (ideally with the mean value in the middle of possible scores). Standardized scores differ in the chosen parameters of μ' and σ' and some special selections are generally recognized. The most commonly used is the z-score with the mean value 0 and the variance 1. Another well known standard score is the IQ score ($\mu' = 100$, $\sigma' = 15$) used mostly for intelligence testing. Other well known scores are also stens, stenines, percentiles, and t-scores.

The set of scores obtained from our data set most likely do not belong to Gaussian distribution. The visual proof is displayed in the Figure 2.1 where it can be clearly seen that it does not resemble the Gaussian distribution. The Shapiro-Wilk normality test also rejects the null hypothesis of the Gaussian distribution by resulting with $p - value = 3.648 \cdot^{-7}$. The solution to this problem is provided by the McCall's area standardization [McCall, 1922, Urbánek et al., 2011] which transforms raw scores to the Gaussian distribution. We performed this step at first and then we transformed scores to the standardized score scales. To illustrate these scales, a short excerpt from whole scale tables for the z-score and the IQ score is shown in the Table 2.1.

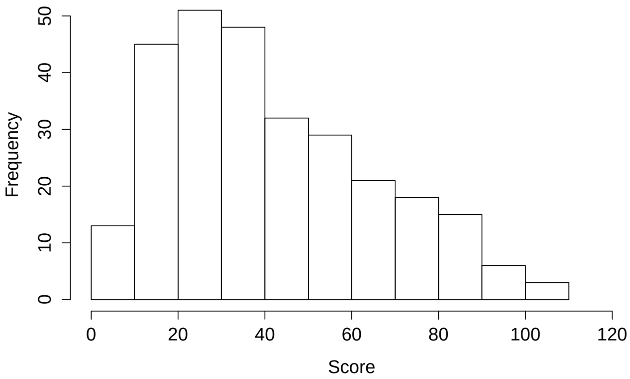


Figure 2.1: Score frequencies

From this table we can see that the center of the normalized score scale is around 40 points of raw score (z value of 0 and IQ of 100). Maximum score obtained was 107 points and minimum 0. The transformed scale at these points has opposite (and extreme) values. The space between center of 40 points and these extremes is the same at both sides for normalized scores but it is not for raw scores. There is the same amount of students scoring in any two intervals of the same length ending/starting at the center on the normalized score scale (for example the same amount of students scored in intervals (-1,0) and (0,1) on the z-score, which corresponds to raw scores of (20, 40) and (40, 72 - not in the table)).

Table 2.1: Standardized scores

raw	0	20	40	60	80	100	107
z	-2.91	-1.00	-0.01	0.67	1.16	2.06	2.91
IQ	56	85	100	110	117	131	144

Validity

Another question it is important to ask is whether the test is actually measuring the factor it is supposed to measure (i.e. in our case if the score obtained reflects mathematical skills rather than for example the ability to read the question or the writing skill of the examinee). This characteristic is called validity and there are many different ways of proving the test is valid. Most validity proofs come from the outside of the test. One way is to let an examinee to answer a new different test measuring the same factor (ideally a test which is already well established). Another way is to consult other factors known about the examinee, which is what was performed in our case.

As was mentioned above, in addition to solutions to individual problems student's grades from subjects (mathematics, physics, and chemistry) were obtained. It is reasonable to expect a correlation between these grades and the score reached. The correlation is present and its values are shown in the following paragraphs. Because of this fact, although the complete validation would require more thorough examination, it is expected that the test is valid.

2.3 Preliminary Test Statistics

In this section we present an overview of results obtained in testing. This should provide an idea of skills of students, prove validity as mentioned in the paragraph above and will also be referenced later on for comparison.

Table 2.2: Average test scores of the four grammar schools.

	GS1	**GS2**	**GS3**	**GS4**	**Total**
Males	51.40	40.08	47.77	51.03	48.48
Females	42.53	54.86	44.45	38.81	43.06
Together	42.76	46.68	46.35	43.65	44.53

The Table 2.2 shows the reached scores divided by gender and school. We calculated Pearson's correlation coefficients of score with other factors. Results are shown in the Table 2.3. The correlation test is associated with its p-value, where the null hypothesis is correlation of 0 (no correlation). It means we can say that the correlation between score and all grades (math, physics, and chemistry) is present. The negative value of correlation means that better grade (lower value) yields better score (higher value) which is expected. Furthermore we can see that the grade in mathematics has the highest correlation while physics and chemistry lower. Another significant correlation is interestingly between the fact that the student filled his/her name and his/her score. Positive value shows that those students who filled their name scored better in the test. On the other hand we can not reject the null hypothesis for gender, so there most likely is no statistically significant correlation between gender and score[5].

Table 2.3: Correlations of the score with other factors

	Gender	**Mathematics**	**Physics**	**Chemistry**	**Name**
Correlation	-0.10	-0.59	-0.42	-0.41	0.22
p-value	0.08	2.20E-16	3.63E-12	2.65E-11	0.18E-4

Some questions were in the form of word problems with a connection to everyday life (calculating savings, time to finish a job,etc.). These questions

[5] Females were encoded as 1 and males as -1. Negative value would show worst score for females, but it is statistically insignificant.

were correlated with the score independently as well. The result is displayed in the Table 2.4. In the first column it is possible to see that there is a strong and statistically significant correlation of the score obtained in these questions with the total score. Also in this case there is not a strong correlation with the gender of the student even though a bit higher and on the edge of rejection of statistical insignificance. The trend of correlations with grades is preserved but the strength of correlation is lower. In connection with previous results, it leads to an assumption that students with worse grades from these subjects answered correctly rather this kind of questions than other questions.

Table 2.4: Correlations of word problems with other factors

	Score	**Gender**	**Mathematics**	**Physics**	**Chemistry**
Correlation	0.69	-0.19	-0.38	-0.25	-0.27
p-value	2.20E-16	0.16E-3	3.16E-10	7.99E-5	2.25E-5

Chapter 3

Models for Adaptive Testing

We remind, as was mentioned in the Section 1, the process of an adaptive test.

- The next question to be asked is selected.
- This question is asked and an answer is obtained.
- This answer is inserted into the model.
- The model (which provides estimates of the student's skills) is updated.
- (optional) Answers to all questions are estimated given the current estimates of student's skills.

In this section we will take a closer look on the model structures for different approaches. Also we will discuss the question selection process from step 1 of the list above. Insertion to the model (step 3) and consequent update of the model (steps 4 and 5) is always done with respective tools for the particular model and will not be extensively discussed here. This topic, especially for BNs, is covered in [Plajner and Vomlel, 2015]. We performed experiments on empirical data with different models of following model types. Results of these experiments are not a part of this paper but they are available in [**?**].

3.1 Building Models with the Help of IRT

The beginning of Item Response Theory (IRT) stems back to about 5 decades ago [Lord and Novick, 1968, Rasch, 1960, Rasch, 1993]. This approach is different from the Classical Test Theory (CTT) and it is getting scientific attention ever since. IRT allows more specific measurement of certain abilities of an examinee. Internationally, there is a large amount of tests adapting this concept. It has stronger assumptions but it also provide stronger results. Nevertheless, its spread is not as high as could have been expected. This smaller impact might be caused by the fact that there is a requirement for a stronger statistical and theoretical preparation of a test creator than in

Question	a	b	c
1	-2	0.3	0
2	0	1.5	0
3	5	0.7	0

Table 3.1: IRFs' parameters

the CTT. In the Czech Republic there is just a few large normalized tests which use this concept [Urbánek et al., 2011][1].

IRT expects a student to have an ability (skill) which directly influences his/her chance of answering a question correctly. This ability is called latent ability or latent trait θ. When we have only one variable[2], it is common to refer to it as proficiency variable. We will stay with the more general skill variable term because we will have more variables in the following parts (Bayesian networks). Every question of the IRT model has an associated item response function (IRF) which is a probability of a successful answer given θ. There are more variants of the shape of this IRF but mostly a 3 parametric model is used (often called 3PL). These parameters reshape a standard logistic function. The resulting IRF, as the probability of a correct answer to *i-th* with the ability of θ, is given by a formula

$$p_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta - b_i)}}$$

where c_i is a parameter for guessing, a_i sets the scale of the question (this sets its discrimination ability - a steeper curve better differentiate between students), b_i is the difficulty of the question (horizontal position of a curve in space). An example of typical IRFs is shown in the Figure 3.1. The dependence of a correct answer probability P based on the skill θ is displayed. The Table 3.1 shows parameters of these functions.

IRFs are created either by setting parameters manually or automatically through machine learning procedures. Manual creation is done by field experts based on their knowledge and experiences in the field. Automatic creation is done from collected data as most likelihood estimates of IRFs' parameters.

3.1.1 Adaptive Test Procedure

Building CAT model with IRT is very straightforward. IRT itself, as was described above, is in the form prepared to be used for CAT. With the model fitted from sample data or created by domain experts we have IRFs for every question. In every phase of the test we can compute an estimate of the latent skill θ based on answers x: $p(\theta|x)$. For this estimations Empirical Bayes or Multiple Imputation methods of IRT are used. Knowing the value

[1] One of them is, for example, the Woodcock–Johnson test [Anton and Ruef, 2010]

[2] There are variants of multidimensional IRT model where it is possible to have more then one latent variable but in this section we are going to discuss only models with one latent variable.

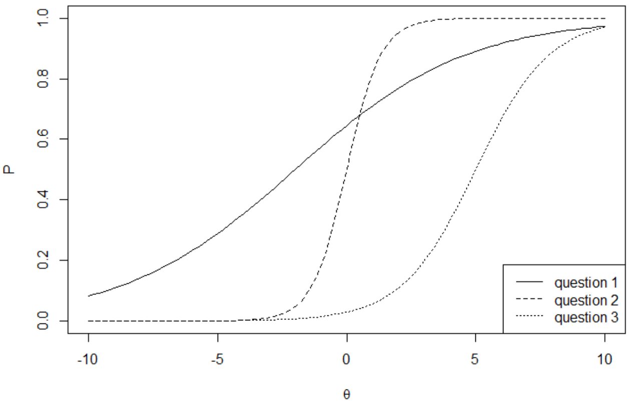


Figure 3.1: Item Response Functions

of the latent skill we know probabilities of a correct answers $p_i(\theta)$ and an incorrect answers $q_i(\theta)$ to every question[3]. More importantly, we are able to calculate the information provided by asking the question. This is called item information and it is given by the formula

$$I_i(\theta) = \frac{(p_i'(\theta))^2}{p_i(\theta)q_i(\theta)}$$

where p_i' is the derivation of the item response function p_i. There is an example of typical item information functions (with the same parameters of items as in the Table 3.1) in the Figure 3.2. This item information provides one, and most straightforward, way of the next question selection. In every step the question X^* which is selected is one with the highest item information.

$$X^*(\theta) = \arg\max_i I_i(\theta)$$

This approach minimizes the standard error of the test procedure [Hambleton et al., 1991] because the standard error of measurement SE_i produced by $i - th$ item is defined as

$$SE_i(\theta) = \frac{1}{\sqrt{I_i(\theta)}}.$$

This means that the better precision of difficulty we are able to achieve while asking questions the less error of measurement.

3.2 Building Models with the Help of BN

In this section we go over the basic definitions of Bayesian networks, more details can be found in [Nielsen and Jensen, 2007, Kjærulff and Madsen,

[3] With 3 parametric model these two numbers do not necessarily sum to 1

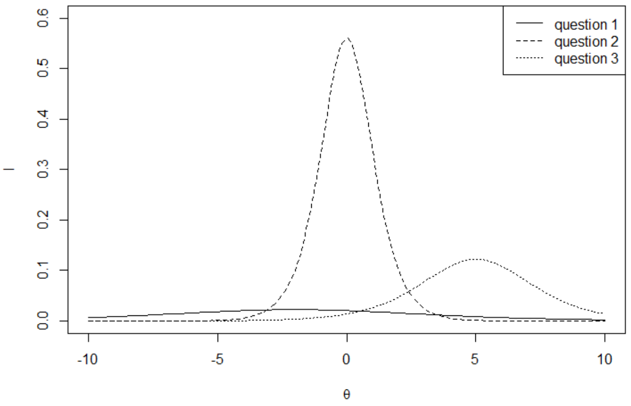


Figure 3.2: Item Information Functions

2008]. The use of BNs in educational assessment is discussed in [Almond et al., 2015, Culbertson, 2015, Millán et al., 2010]. This section is focused on the creation Bayesian networks models for CAT. This topic is also discussed, for example, in [Vomlel, 2004b, Vomlel, 2004a].

Bayesian network is a probabilistic graphical model, a structure representing conditional independence statements. It consists of the following:

- a set of variables (nodes),
- a set of edges,
- a set of conditional probabilities.

Edges between variables have to form a directed acyclic graph (DAG). Each variable has a list of mutually exclusive states. For each variable a conditional probability distribution conditioned by its parents is defined, e.g., variable A with parents B_1,B_2,...,B_n has the conditional probability table[4] $P(A|B_1, B_2, ..., B_n)$.

To build a BN model for adaptive testing we need to perform 3 steps:

1. define nodes of the BN,
2. define connections between nodes, and
3. specify initial values of conditional probability tables.

Types of Nodes

We will divide nodes of a BN into three sets.

[4] Note that the variables with no parents have the table in the form $P(A)$

- A set of n variables we want to estimate $\{S_1, \ldots, S_n\}$. These variables represent latent skills (abilities, knowledge) of a student. We will call them skills or skill variables. We will use symbol $\boldsymbol{S}$ to denote the multivariable $\boldsymbol{S} = (S_1, \ldots, S_n)$ taking states $\boldsymbol{s} = (s_{1,i_1}, \ldots, s_{n,i_n})$.

- A set of m variables representing eventual additional information about the student $\{I_1, \ldots, I_m\}$. We will use the symbol $\boldsymbol{I}$ to denote the multivariable $\boldsymbol{I} = (I_1, \ldots, I_m)$ taking states $\boldsymbol{i} = (i_1, \ldots, i_m)$.

- A set of p questions (math problems) $\{X_1, \ldots, X_p\}$. We will use the symbol $\boldsymbol{X}$ to denote the multivariable $\boldsymbol{X} = (X_1, \ldots, X_p)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_p)$.

Skills

Skill nodes model the student abilities and, generally, they are not directly observable. It means they are hidden variables of the model and their value is not known prior to the model creation. Several decisions concerning skill nodes are to be made during the model creation.

The first decision is the number of skill nodes itself. Should we expect one common skill or should it rather be several different skills each related to a subset of questions only? In the later case it is necessary to specify which skills are involved in solution of each particular question (i.e. a math problem). These skills become parents of the considered question. Possible relations between them are discussed in the last chapter.

This way we create variables with a given meaning (specific student ability). It is not possible to cover all the necessary skills to solve a question. Also, there are some other aspect important for a question's solution. During the interpretation of a CAT result we have to be careful. Even though we have given the variable the meaning, it is possible that the model learned a combination of this meaning with other factors. Nevertheless, if the model was properly constructed the meaning of the variable should converge to the intended meaning. For example, we have two skills overlapping over some questions and not overlapping over other questions. It means that a skill 1 is needed to solve some questions, skill 2 is needed as well to solve a subset of them and then some others. If we select a student who has (by the model) only one of these skills, we should be able to observe that he/she answered correctly only to a part of questions of the corresponding skills. If this is true and we can specify the skill needed to solve this type of questions, then it is reasonable to assume that the skill interpretation converges to the intended one.

Another decision we are facing is about the size of a state space of skill nodes. As an unobserved variable, it is hard to decide how many states it should have. Another alternative is to use a continuous skill variable instead of a discrete one but we did not elaborate more on this option. For BNs no suitable apparatus to handle continuous parents exists. It would be possible though to create different kind of models with continuous parents. The use of a discrete state space can be in a way viewed as sampling (or discretizing) the continuous skill variable of the student. It may seem reasonable to create many states for each skill variable but each state increases the total number of parameters of the model (the exact rate depends on the structure). This means that these models are too complex. It may be hard to learn a statistically reliable and stable model if the complexity is high. Conditional probability tables may end up very sparse and that limits the generalization ability of the BN.

Skills are ordinal variables.A variable S_i with possible states $\{s_{i,1}, \ldots, s_{i,n_i}\}$ and an arbitrary state $s_{i,0}$ which is one but first state of the variable, has a probability

$$P(S \leq s_{i,j})$$

of the variable S_i being in one of states $s_{i,k}, k \in \{1, \ldots j\}$ (previous to j). We define

$$P(S_i \leq s_{i,0}) \overset{\text{def}}{=} 0$$

$$P(S \leq s_{i,n_i}) \overset{\text{def}}{=} 1.$$

Then for the probability of the ordinal variable S_i being in the state $s_{i,j}$ the following has to be satisfied:

$$P(S_i = s_{i,j}) = P(S_i \leq s_{i,j}) - P(S_i \leq s_{i,j-1}).$$

If this assumption would not be taken into account it may cause inconsistent results. For example, consider a student answering to a question that is dependent on one skill (with 3 states). Without the ordinality assumption a BN model could result in a following probability distribution:

- With a low level of the skill the chance of a correct answer is high.
- With a medium level of the skill the chance of a correct answer is low.
- With a high level of the skill the chance of a correct answer is high.

This situation is impossible with our definition of a skill to be a reasonable requirement for the question's solution. If the probability of a correct answer is high with the low level of the skill it has to be higher for the medium level and even higher for the high level[5].

It is also possible to use a different BN model where unobserved skill variables are replaced by observed variables. The easiest way is to introduce the total test score as a variable into the model. To do this it is necessary to use a coarse discretization. At first, total scores are divided into n groups and by that we obtain an observed variable having n possible states. The states represent a group of students with similar scores achieved. During the learning phase the variable is observed and the information is used for learning. On the other hand, during the testing, the resulting score is not known – we are trying to estimate the group into which a test subject falls. In the testing phase the variable is again hidden (unobservable).

Combinations of both types of skill variables are also possible.

[5] It would be possible to have an inverse situation if we had a skill which negates a correct solution, but it still would have to be monotonic.

Information about a Student

Information nodes gather additional information we have about a student. They are observed variables. The number of their states correspond to the possible options of the specific piece of information. One state, labeled as “unknown”, may be also included for missing values. This should be done especially if it has any information value in the context of the test. For example, if a student does not enter his/her grade of mathematics it may indicate that he does not feel very confident. If it is the case, then this information can be used to our advantage. This additional information may improve the quality of the student model.

Questions

The last type of nodes are question nodes. This node type holds answers to individual questions. Its state space depends on the number of possible answers to a question. As it was already mentioned it is difficult to build a computerized system for evaluation of answers which do not use the multiple choice question type. In some cases it may be possible to have open answers to questions but in most cases these would be too hard to process. With multiple choice, a question node has two possible state spaces:

1. one state for each possible answer,
2. one state for the correct answer and one for any incorrect answer.

The former case is more informative. It gives us a possibility to differentiate between students not only based on the fact that the answer is correct/incorrect but as well on the fact which incorrect one it is. Nevertheless, it has some limitations. The more the states the higher the number of model parameters to be learned. With a limited training data it may be difficult to reliably estimate model parameters. It requires larger data set to learn from.

Another aspect is the concept of fairness. It is questionable if it is fair to make distinctions based on wrong answers. On one hand, a classical test usually do not do this. If the answer is wrong then it does not matter which one it is. On the other hand the selected answer brings additional information about the student’s ability and there is no theoretical obstacle why not to use it.

Connections between Nodes

The last step in the BN model creation is to define a set of arcs between variables (nodes), i.e., network structure. This set defines relations between skills, questions, and additional information, eventually, also inbetween them. This task is usually done by the domain expert by hand. There are algorithms for automated structure learning, but these algorithms are for general cases and usually do not provide usable results for this specific purpose. We discuss the automated creation more in the last chapter. The expert who is creating a structure has to pinpoint which variables should be connected. Usually, we connect skill nodes with questions in a way that skill nodes are parents of questions when the skill is needed to a question's solution. The connection can be of

- a deterministic relation, where we have to assign all the values of an associated conditional probability table (discussed in the next section), or
- a specific relation, e.g., AND, OR, etc. (discussed in the last section).

Then, there may be connections between skills if we want to further specify them. For example, we can create a common skill that is brought down to two sub-skills. This common skill have a connection to the two sub-skills, but possibly no questions. If we want to include some additional information the expert also has to define what it influences. It depends on the type of the piece of information. It can influence a skill, or even a question directly if it is an important factor for its solution.

3.2.1 Model Learning

The last action to complete a BN is to define conditional probability tables (CPT) for each node. Values in CPTs represent probabilities of the variable being in a state conditioned by a configuration of its parents for every state of the variable and every combination of its parents. First, we manually input values into CPTs. Values should reflect a general expectation and are created with expert knowledge in the field of the test. These probabilities serve as a starting point for the learning algorithm. Next we learn the model with the standard EM algorithm using collected data. This operation modifies values in CPTs to better reflect the data.

BNs have a large advantage since they can learn from missing data (with some unknown values). The EM algorithm, that is used for learning, has no problems operating with missing data. Also, during the prediction process unknown values are simply not inserted into the network and the prediction is performed without this knowledge.

3.2.2 Adaptive Test Procedure

During the adaptive test we use standard BN inference methods to update the network. These methods estimate probabilities of skill variables as well as probabilities of a success in unanswered questions.

One task to solve during the CAT procedure is the selection of the next question. It is repeated in every step of the testing and it is described below.

Let the test be in the state after $s-1$ steps. It means that s questions were already answered and they form the evidence e:

$$e = \{X_{i_1} = x_{i_1}, \ldots, X_{i_n} = x_{i_n} | i_1, \ldots, i_n \in \{1, \ldots, m\}\}.$$

Remaining questions

$$\mathcal{X}_s = \boldsymbol{X} \setminus e$$

are unobserved (unanswered).

The goal is to select a question from $\mathcal{X}_s$ to be asked next. We select a question with the largest expected information gain.

We compute the cumulative Shannon entropy over all skill variables of S given evidence e. It is given by the following formula:

$$H(e) = \sum_{i=1}^{n} \sum_{j=1}^{i_n} -P(S_i = s_{i,j}|e) \cdot \log P(S_i = s_{i,j}|e).$$

Assume we decide to ask a question $X' \in \mathcal{X}_s$ with possible outcomes $x'_1, \ldots, x'_p$. After inserting the observed outcome the entropy over all skills changes. We can compute the value of new entropy for evidence extended by $X' = x'_j$, $j \in \{1, \ldots, p\}$ as:

$$H(e, X' = x'_j) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \begin{array}{l} -P(S_i = s_{i,j}|e, X' = x'_j) \\ \cdot \log P(S_i = s_{i,j}|e, X' = x'_j) \end{array} .$$

This entropy $H(e, X' = x'_j)$ is the sum of individual entropies over all skill nodes. Another option would be to compute the entropy of the joint probability distribution of all skill nodes. This would take into account correlations between these nodes. In our task we want to estimate marginal probabilities of all skill nodes. In the case of high correlations between two (or more) skills the second criterion would assign them a lower significance in the model. This is the behavior we wanted to avoid. The first criterion assigns the same significance to all skill nodes which is a better solution. For our problem, the greedy strategy based on the sum of entropies provides good results. Moreover, the computational time required for the proposed method is lower.

Now, we can compute the expected entropy after answering question X':

$$EH(X', e) \quad = \quad \sum_{j=1}^{p} P(X' = x'_j|e) \cdot H(e, X' = x'_j) \ .$$

Finally, we choose a question X^* that maximizes the information gain $IG(X', e)$

$$\begin{aligned} X^* &= \arg\max_{X' \in \mathcal{X}_s} IG(X', e) \ , \text{ where} \\ IG(X', e) &= H(e) - EH(X', e) \ . \end{aligned}$$

3.2.3 Obtaining Total Score from Skills

BN models usually produce estimates of student skills. In some cases this is more useful than a regular score. On the other hand if we want to obtain a score in terms of achieved points we have to transform these skills. First, we define a score SC as a weighted sum of skills $(S_1 = s_1, \ldots, S_n = s_n)$:

$$SC \stackrel{\text{def}}{=} \sum_{i=1}^{n} s_i C_i$$

C_i is a weight associated with the i-th skill. These weights define the maximum score

$$SC_{max} = \sum_{i=1}^{n} s_{i,n_i} C_i,$$

where s_{i,n_i} is the last possible state of S_i.

The weights C_i can be set to any value. There are two special cases. The first is, when all the weights are set to be equal

$$C_i = C.$$

Then the impact of each skill on the total score depends on the number of the skill's states. The second is, when we want all the skills to have the same impact on the score. Then weights have to be set to

$$C_i = \frac{n_{max}}{n_i} C,$$

where $n_{max} = \max_i n_i$ and C is a scaling constant.

During the testing process, the states of skills $S_1, \ldots, S_n$ are unknown. We use their estimates to compute an expected value of the total score:

$$E(SC) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} P(S_i = s_{i,j}) s_{i,j} C_i$$

For a tested student this expected total score is our estimate of the real total score.

3.3 Building Models with the Help of NN

Neural networks are models for approximations of non-linear functions. We present a brief overview of NNs. For more details about NNs, please refer to [Haykin, 2009, Aleksander and Morton, 1995].

There are three different parts of a NN:

1. an input layer,
2. several hidden layers, and
3. an output layer.

Each layer consist of several nodes called neurons. These neurons have connections to the next layer. Usually the connections are formed from one neuron to all neurons in the next layer. Every connection has an associated weight. These weights are used to calculate a value of a neuron from values of its predecessors. A substantial difference between NNs and previously described IRT and BN models is that NNs are learned by supervised learning. In this scenario an output (result), that is to be predicted by a NN, has to be known during learning. IRT and BNs are usually working with unsupervised learning [Schlesinger and Hlaváč, 2002]. For them we do not need to know the output. That allow us to learn models for unknown values of skills. During the learning it is not necessary[6] to input other values than question

[6] BNs are able to learn even in a supervised fashion knowing the output value

responses. This pattern does not work with neural networks. It is necessary to provide target values during the learning algorithm. In this case we can use score results of the test as target values. It means, that the NN model can not predict the skill of a student but it can predict his/her score directly.

The input layer of the NN model for CAT is created by as many nodes as the number of questions is. For every question we are feeding its result into the neuron. We have two options how to encode information about the answer. It is either 1 (or 1,...,n if it is possible to have more points) for a correct answer and 0 for an incorrect one. Inserting 0 to a node means that there will not be any activation of such node. If we want to activate it even for an incorrect answer we have to encode it as -1.

There is a general problem of missing data with NNs [Hastie et al., 2009, Pesonen et al., 1998]. In order to produce a result NN has to obtain values to all its nodes. There are many different methods to overcome this problem. In our research we input either a value of 0 (there wont be any activation of a neuron then) or an average score for a question (hopefully producing an average result from that question).

The number and the size of hidden layers is up to our choice. There is no specific rule how to choose the best specifications of a NN.

The last output layer contains only a single node. The value of this node corresponds to predicted score of a student.

Learning of the NN model is done by a standard back propagation algorithm from collected data.

3.3.1 Selecting the Next Question

In the two previous models we have used the entropy reduction criterion to select the next question. The entropy was measured on the skill variables. We have no skill variables with NN, only a score output. Measuring an entropy in this case is not possible, because reducing the entropy of total score would mean that we are trying to push a student to some specific score value. With score there is no reason for this, with skills we wanted a student to reach a certain level of skill. Instead, we propose a simple criterion to deal with the selection of the next question. We want the selected question to provide us as much information as possible about the student. That means that a student who answers incorrectly should be as far as possible on the score scale from the one who answers correctly. Let the $SC|X_{i,x}$ be the score prediction after answering the i-th question's state x, $P(X_{i,x})$ the probability of state x to be the answer to question i. $P(X_{i,x})$ can be obtained, for example, by statistical analysis of answers. We select a question X^* maximizing the variance of predicted scores:

$$X^* = \arg\max_i \underset{\mathrm{x}}{\mathrm{Var}}(SC|X_{i,x}) = \sum_x P(X_{i,x})(SC|X_{i,x} - \overline{SC|X_i}),$$

where

$$\overline{SC|X_i} = \sum_x P(X_{i,x}) SC|X_{i,x}$$

is the mean values of predicted scores.

3.4 Some Remarks on Models

Scoring

Both IRT and BN models usually estimate the skill of a student. We have to perform transformations to the score scale if we want to produce a score of the test. For BN it was discussed in the Section 3.2.3. For IRT models a similar procedure can be applied. NN models predict the resulting score directly thus there is no conversion needed.

Question nodes

BN allow us to exploit every answer to a question as an information about the student. This may help the adaptive test to evolve faster in case there are some answers which are "more" wrong that other wrong answers. As was mentioned above it requires large data samples for learning to avoid overfitting.

Additional Information

BN and NN models allow us to include additional information about a student. This is not possible in the standard version of IRT-CAT.

Future Work

In this section we present a brief overview of research problems of our further interest.

Constrained question selection

Adaptive tests constructed with the IRT model have various well established methods of selecting next questions. As it was discussed in the first chapter, the selection process should be carefully managed to prevent the overuse of some questions as well as very similar question combinations for many participants. One way IRT researchers are solving this issue is through a series of constraints in the selection process [van der Linden and Veldkamp, 2004, Stocking and Lewis, 2000]. We plan to analyze the use of such constraints for BN models. The research will be conducted in order to define the correct constraints, their application and their impact on the CAT procedure.

Precision of skills measurement

Stopping rules for CAT used in IRT are of two kinds.

- Practical, e.g., limited time of the test or a specific number of asked questions, or
- Statistical, e.g., reliability of the test (precision of measured latent skill).

The goal of this research path is to provide a similar criteria for BN-CAT models as well. We will establish a criterion able to provide a statistically sound precision of estimates. Afterwards, we will review the effect of using this criterion as the stopping rule for CAT.

Model quality criterion

So far in our research we were evaluating a model quality based on its predictions of answers to remaining questions. This criterion provide reasonable results but it also at the same time has some drawbacks. First of all, for some models similar to each other it is often hard to order them in terms of quality. The prediction accuracy varies over the test run and the order of models changes in different steps (different numbers of asked questions). Next, some models are able to predict the student's skills very well while their prediction power back to answers is not that good. These models have

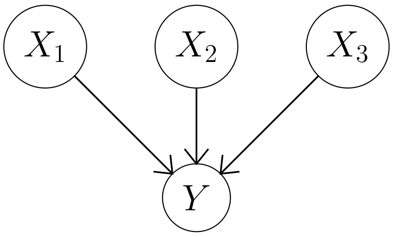


Figure 3.3: Simple Bayesian network

a disadvantage compared to other models (with better backwards precision) if we use the current criterion. This applies especially for some of neural networks models.

We will focus on the design and testing criteria that are able to take into account more aspects of models for CAT. We would like to better describe, and experimentally verify on data, criteria that work with skills estimates. Specifically, we will measure the quality of a model as a correlation of the predicted score with the real score. Also we want to compare the final ordering of students based on predictions and real values.

Model creation

One reason of the small spread of the CAT use is in the high amount of work required to create a student model. The process itself may be very useful due to its highly organized character and statistically strong results, but it requires a certain level of expertise in modeling and statistics.

We want to address an option of an automated model creation based on data. Algorithms for BN structure learning in a general case exist, some examples are in [Margaritis, 2003]. These algorithms cover general structure learning which may serve well in some areas but in case of CAT it usually does not reflect our situation very well. We would like to explore special types of models which would fit better for our needs as the CAT model. This leads to learning a model which contains a local structure. It is an interesting theoretical task for the whole BN community, not limited only to CAT.

Local structure in BNs

Bayesian networks encode conditional probabilities. These probabilities can be encoded in many ways. One of the most common is to define a conditional probability table (CPT). This table defines a conditional probability of a variable for every combination of its parents. For example, let us consider the network in the Figure 3.3. The CPT for this network with a child Y and parents X_1, X_2, X_3 is in the Table 3.2 (values are chosen at random). In the general case we have to specify 2^n parameters where n is the number of parents. In the example case, we need $2^3 = 8$ parameters. This amount of parameters correspond to binary nodes. For nodes with more than two states, the total amount of parameters is even higher.

X_1	1	1	1	1	0	0	0	0
X_2	1	1	0	0	1	1	0	0
X_3	1	0	1	0	1	0	1	0
$Y(0)$	0.05	0.41	0.12	0.67	0.85	0.5	0.9	0.08
$Y(1)$	0.95	0.59	0.78	0.33	0.15	0.5	0.1	0.92

Table 3.2: CPT for the BN in 3.3

If we have additional information about the structure (relations) of BN variables we can use this information to our benefit. The local structure in a BN allow us to specify these relations and encode conditional probabilities efficiently. The local structure concept is sometimes called as canonical models. A more thorough introduction to the theory of canonical models can be found in [Díez and Druzdzel, 2007]. Basically, we establish a function that prescribes how to compute a child's value from its parents. The function can be of many different types, but for our illustrative example, we will now consider the noisy OR function only. If there is OR local structure (without noise) it means, that the value of Y is:

$$Y = X_1 \vee X_2 \vee X_3$$

Because the relation is encoded directly in the formula, there is no need to specify a CPT. In this case, we need to know only the values of X_1, X_2, X_3. To this model we introduce noise by adding auxiliary variables Z_1, Z_2, Z_3. The network then changes from the one in the Figure 3.3 to the one in the Figure 3.4. In this case we need to include probabilities

$$P(Z_i|X_i)$$

which specify the noise. This forms the noisy OR local structure. The total number of parameters that we have to specify is n (there might be one additional inhibitor variable with connection to each Z_i making it to $2n$ parameters). In our example we have to specify 3 or 6 parameters. It means there is a difference of $O(n)$ for models with a local structure compared to $O(2^n)$ for models without a local structure.

We will explore learning strategies for BNs with a local structure. It means, we have to modify the learning process of the general BN to this special case. During the general structure learning we use criteria for the model ranking. AIC/BIC[7] criteria are popular. These criteria takes into account the prediction quality of a model, but also penalize it for its size. It is necessary to adapt these criteria to the specialized case of BNs with the local structure. AIC/BIC work with the number of nodes to compute parameters, but with the local structure the reduction of number of parameters is essential.

It is clear that exploiting a local structure in a BN has many advantages.

- First, it is easier to learn a statistically reliable model with less parameters.

[7] Akaike Iformation Criterion/Bayesian Information Criterion

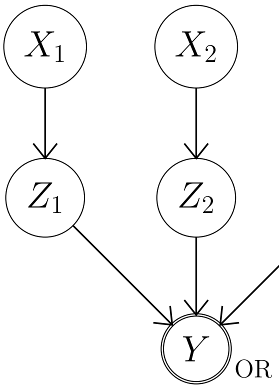


Figure 3.4: Bayesian network with noisy OR

- We are able to create more complex models where computational operation will be quickly solvable.
- It is possible to store this model in less space.
- Last but not least, we do not need to specify a large number of conditional probabilities. These probabilities are often obtained from experts and it may be difficult to get reliable estimates of them for large CPTs. With the local structure, we have to specify significantly less conditional probability values.

Acknowledgements

The work on this paper has been supported from GACR project n. 16-12010S.

Bibliography

[Aleksander and Morton, 1995] Aleksander, I. and Morton, H. (1995). *An Introduction to Neural Computing*. Information Systems. International Thomson Computer Press.

[Almond and Mislevy, 1999] Almond, R. G. and Mislevy, R. J. (1999). Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237.

[Almond et al., 2015] Almond, R. G., Mislevy, R. J., Steinberg, L. S., Yan, D., and Williamson, D. M. (2015). *Bayesian Networks in Educational Assessment*. Statistics for Social and Behavioral Sciences. Springer New York, New York, NY.

[Anton and Ruef, 2010] Anton, F. and Ruef, M. (2010). *Woodcock-Johnson: Mezinárodní Edice II. Uživatelská příručka*. WMF Press.

[Cowell et al., 1999] Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer.

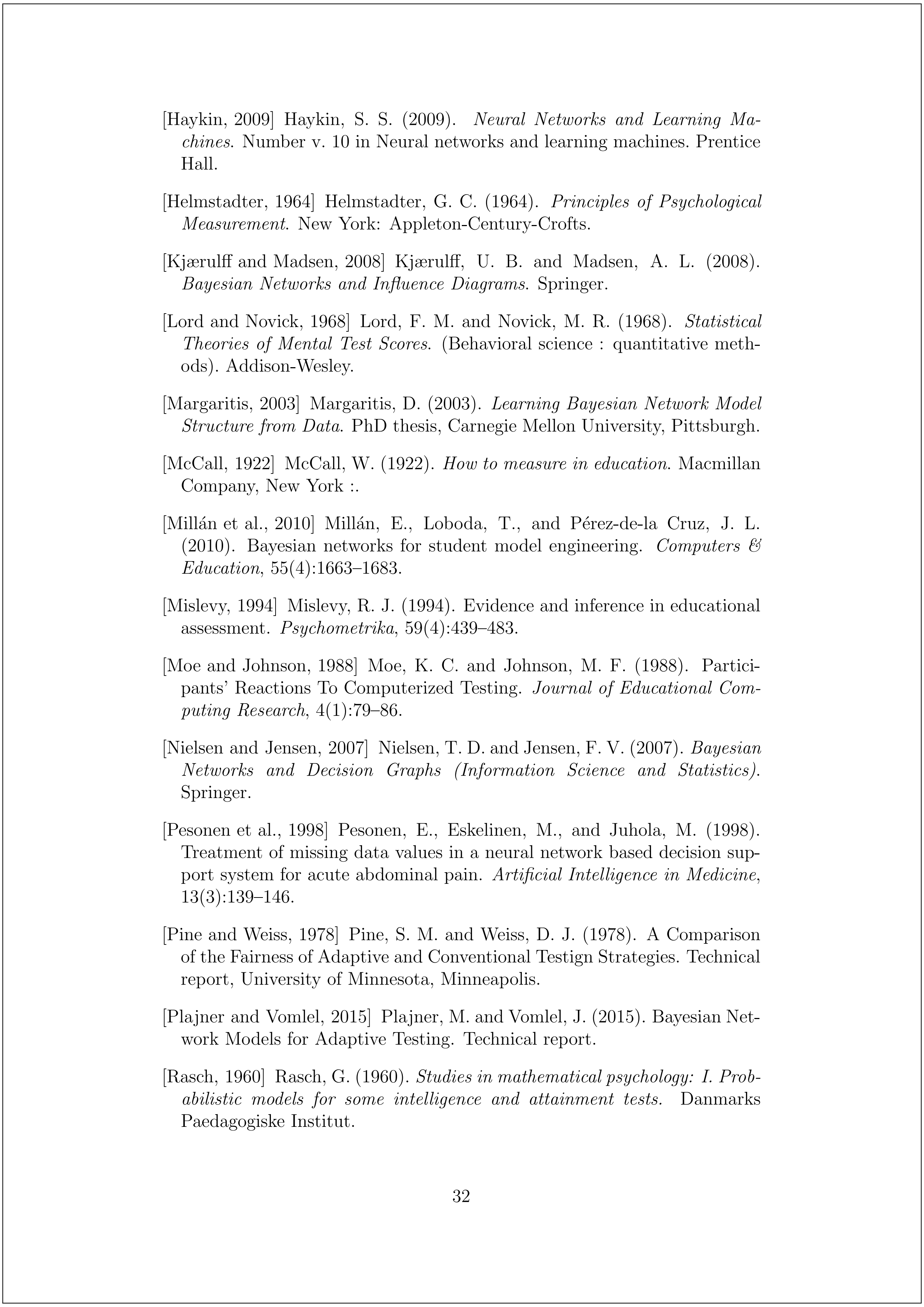
[Culbertson, 2014] Culbertson, M. J. (2014). *Graphical Models for Student Knowledge: Networks, Parameters, and Item Selection*. PhD thesis, University of Illinois at Urbana.

[Culbertson, 2015] Culbertson, M. J. (2015). Bayesian Networks in Educational Assessment: The State of the Field. *Applied Psychological Measurement*, 40(1):3–21.

[Díez and Druzdzel, 2007] Díez, F. J. and Druzdzel, M. J. (2007). Canonical Probabilistic Models for Knowledge Engineering. Technical report, Research Centre on Intelligent Decision-Support Systems.

[Hambleton et al., 1991] Hambleton, R. K., Swaminathan, H., and Rogers, H. J. (1991). *Fundamentals of item response theory*, volume 21. SAGE Publications.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, New York, NY.



[Haykin, 2009] Haykin, S. S. (2009). *Neural Networks and Learning Machines.* Number v. 10 in Neural networks and learning machines. Prentice Hall.

[Helmstadter, 1964] Helmstadter, G. C. (1964). *Principles of Psychological Measurement.* New York: Appleton-Century-Crofts.

[Kjærulff and Madsen, 2008] Kjærulff, U. B. and Madsen, A. L. (2008). *Bayesian Networks and Influence Diagrams.* Springer.

[Lord and Novick, 1968] Lord, F. M. and Novick, M. R. (1968). *Statistical Theories of Mental Test Scores.* (Behavioral science : quantitative methods). Addison-Wesley.

[Margaritis, 2003] Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data.* PhD thesis, Carnegie Mellon University, Pittsburgh.

[McCall, 1922] McCall, W. (1922). *How to measure in education.* Macmillan Company, New York :.

[Millán et al., 2010] Millán, E., Loboda, T., and Pérez-de-la Cruz, J. L. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683.

[Mislevy, 1994] Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika*, 59(4):439–483.

[Moe and Johnson, 1988] Moe, K. C. and Johnson, M. F. (1988). Participants' Reactions To Computerized Testing. *Journal of Educational Computing Research*, 4(1):79–86.

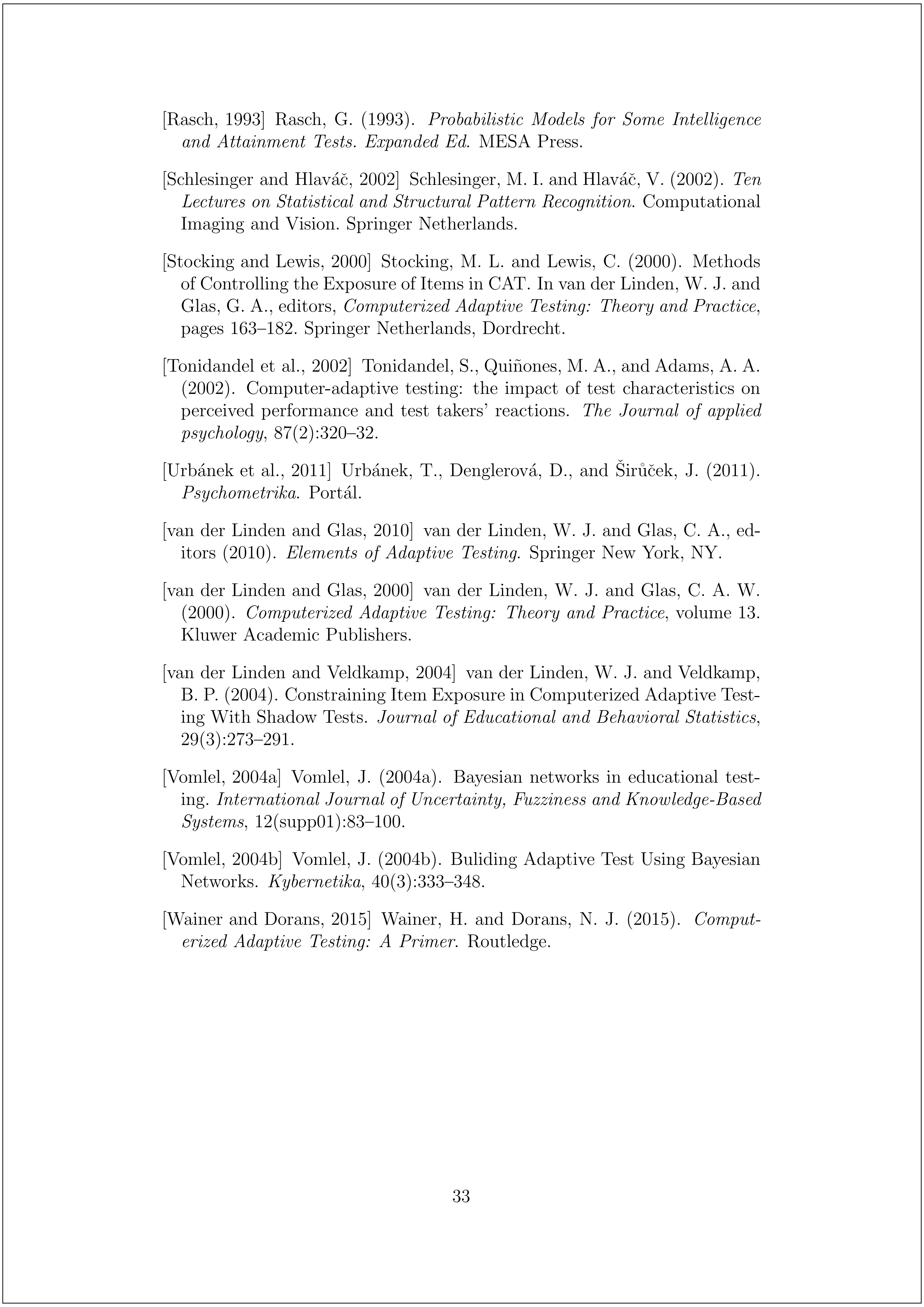
[Nielsen and Jensen, 2007] Nielsen, T. D. and Jensen, F. V. (2007). *Bayesian Networks and Decision Graphs (Information Science and Statistics).* Springer.

[Pesonen et al., 1998] Pesonen, E., Eskelinen, M., and Juhola, M. (1998). Treatment of missing data values in a neural network based decision support system for acute abdominal pain. *Artificial Intelligence in Medicine*, 13(3):139–146.

[Pine and Weiss, 1978] Pine, S. M. and Weiss, D. J. (1978). A Comparison of the Fairness of Adaptive and Conventional Testign Strategies. Technical report, University of Minnesota, Minneapolis.

[Plajner and Vomlel, 2015] Plajner, M. and Vomlel, J. (2015). Bayesian Network Models for Adaptive Testing. Technical report.

[Rasch, 1960] Rasch, G. (1960). *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut.



[Rasch, 1993] Rasch, G. (1993). *Probabilistic Models for Some Intelligence and Attainment Tests. Expanded Ed.* MESA Press.

[Schlesinger and Hlaváč, 2002] Schlesinger, M. I. and Hlaváč, V. (2002). *Ten Lectures on Statistical and Structural Pattern Recognition.* Computational Imaging and Vision. Springer Netherlands.

[Stocking and Lewis, 2000] Stocking, M. L. and Lewis, C. (2000). Methods of Controlling the Exposure of Items in CAT. In van der Linden, W. J. and Glas, G. A., editors, *Computerized Adaptive Testing: Theory and Practice*, pages 163–182. Springer Netherlands, Dordrecht.

[Tonidandel et al., 2002] Tonidandel, S., Quiñones, M. A., and Adams, A. A. (2002). Computer-adaptive testing: the impact of test characteristics on perceived performance and test takers' reactions. *The Journal of applied psychology*, 87(2):320–32.

[Urbánek et al., 2011] Urbánek, T., Denglerová, D., and Širůček, J. (2011). *Psychometrika.* Portál.

[van der Linden and Glas, 2010] van der Linden, W. J. and Glas, C. A., editors (2010). *Elements of Adaptive Testing.* Springer New York, NY.

[van der Linden and Glas, 2000] van der Linden, W. J. and Glas, C. A. W. (2000). *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers.

[van der Linden and Veldkamp, 2004] van der Linden, W. J. and Veldkamp, B. P. (2004). Constraining Item Exposure in Computerized Adaptive Testing With Shadow Tests. *Journal of Educational and Behavioral Statistics*, 29(3):273–291.

[Vomlel, 2004a] Vomlel, J. (2004a). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):83–100.

[Vomlel, 2004b] Vomlel, J. (2004b). Buliding Adaptive Test Using Bayesian Networks. *Kybernetika*, 40(3):333–348.

[Wainer and Dorans, 2015] Wainer, H. and Dorans, N. J. (2015). *Computerized Adaptive Testing: A Primer.* Routledge.

Probabilistic Models for Computerized Adaptive Testing: Experiments

Martin Plajner
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4
Prague 8, CZ-182 08
Czech Republic

Jiří Vomlel
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4
Prague 8, CZ-182 08
Czech Republic

Abstract

This paper follows previous research we have already performed in the area of Bayesian networks models for CAT. We present models using Item Response Theory (IRT - standard CAT method), Bayesian networks, and neural networks. We conducted simulated CAT tests on empirical data. Results of these tests are presented for each model separately and compared.

1 INTRODUCTION

All of us are in touch with different ability and skill checks almost every day. The computerized form of testing is also getting an increasing attention with the spread of computers, smart phones and other devices which allow easy contact with target groups. This paper focuses on the Computerized Adaptive Testing (CAT) (van der Linden and Glas, 2000; Almond and Mislevy, 1999; Almond et al., 2015) and it follows a previous research paper (Plajner and Vomlel, 2015).

In this previous paper we explained the concept of CAT. Next, we describe our empirical data set. The use of Bayesian networks for CAT was discussed and we constructed different types of Bayesian network models for CAT. These models were tested on empirical data. The results were presented and discussed.

In this paper we present two additional model types for CAT: Item Response Theory (IRT) and neural networks. Moreover, new BN models are proposed in this paper. We conducted simulated CAT tests on the same empirical data as in the previous paper. This allows us to make comparisons of two new model types (BN and NN) with the CAT standard IRT model. Results are presented for each model separately and then they are all compared.

2 CAT PROCEDURE AND MODEL EVALUATION

All models proposed in this paper are supposed to serve for adaptive testing. In this section we briefly outline the process of adaptive testing[1] with the help of these models and methods for their evaluation. For every model we used similar procedures. The specific details for each model type are discussed in the corresponding sections. At this point we discuss the common aspects.

In every model type we have the following types of variables. For some models they have a different specific name because of an established naming convention of the corresponding method. Nevertheless, the meaning of these variables is the same and we explain differences for each model types. In this paper we use two types of variables:

- A set of n variables we want to estimate $\mathcal{S} = \{S_1, \ldots, S_n\}$. These variables represent latent skills (abilities, knowledge) of a student. We will call them skills or skill variables. We will use symbol $\boldsymbol{S}$ to denote the multivariable $\boldsymbol{S} = (S_1, \ldots, S_n)$ taking states $\boldsymbol{s} = (s_{1,i_1}, \ldots, s_{n,i_n})$.
- A set of p questions $\mathcal{X} = \{X_1, \ldots, X_p\}$. We will use the symbol $\boldsymbol{X}$ to denote the multivariable $\boldsymbol{X} = (X_1, \ldots, X_p)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_p)$.

We collected data from paper tests conducted by grammar schools' students. The description of the test and its statistics can be found in the paper (Plajner and Vomlel, 2015). All together, we have obtained 281 test results. Experiments were performed with each model of each type that is described in following sections. We used 10-fold cross-validation method. We learned each model from $\frac{9}{10}$ of randomly divided data. The remaining $\frac{1}{10}$ of the data set served as a testing set. This procedure was repeated 10 times to obtain 10 learned student models with the same structure and different parameters.

[1] Additional information about CAT can be found in (Wainer and Dorans, 2015)

With these learned models we simulate CAT using test sets. For every student in a test set a CAT procedure consists of the following steps:

- The next question to be asked is selected.
- This question is asked and an answer is obtained.
- This answer is inserted into the model.
- The model (which provides estimates of the student's skills) is updated.
- (optional) Answers to all questions are estimated given the current estimates of student's skills.

This procedure is repeated as long as necessary. It means until we reach a termination criterion, which can be, for example, a time restriction, the number of questions, or a confidence interval of the estimated variables. Each of these criteria would lead to a different learning strategy (Vomlel, 2004b), but finding a global optimal selection with these strategies would be NP-hard (Lín, 2005). We have chosen an heuristic approach based on greedy optimization methods. Methods of the question selection differ for each model type and are explained in the respective sections. All of them use the greedy strategy to select questions.

To evaluate models we performed a simulation of CAT test for every model and for every student. During testing we first estimated the skill(s) of a student based on his/her answers. Then, based on these estimated skills we used the model to estimate answers to all questions $\boldsymbol{X}$. Let the test be in the step s ($s-1$ questions asked). At the end of the step s (after updating the model with a new answer) we compute marginal probability distributions for all skills $\boldsymbol{S}$. Then we use this to compute estimations of answers to all questions, where we select the most probable state of each question $X_i \in \mathcal{X}$:

$$x_i^* = \arg\max_{x_i} P(X_i = x_i|\boldsymbol{S}).$$

By comparing this value to the real answer to $i{-}th$ question x_i' we obtain a success ratio of the response estimates for all questions $X_i \in \mathcal{X}$ of a test result t (particular student's result) in the step s

$$\begin{aligned} \mathrm{SR}_s^t &= \frac{\sum_{X_i\in\mathcal{X}} I(x_i^* = x_i')}{|\mathcal{X}|}\text{ , where} \\ I(expr) &= \begin{cases} 1 & \text{if } expr \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The total success ratio of one model in the step s for all test data is defined as

$$\mathrm{SR}_s = \frac{\sum_{t=1}^{N} \mathrm{SR}_s^t}{N} .$$

SR_0 is the success rate of the prediction before asking any questions.

3 ITEM RESPONSE THEORY

The beginning of Item Response Theory (IRT) stem back to 5 decades ago and there is a large amount of resources available, for example, (Lord and Novick, 1968; Rasch, 1960, 1993). IRT allows more specific measurements of certain abilities of an examinee. It expects a student to have an ability (skill) which directly influences his/her chance of answering a question correctly. When we have only one variable[2], it is common to refer to it as a proficiency variable. This ability is called latent ability or a latent trait θ. The trait θ corresponds to the general skill S_1 defined in the Section 2. Every question of the IRT model has an associated item response function (IRF) which is a probability of a successful answer given θ.

We fitted our data on the 2 parametric IRT model. It means that characteristic Item Response Functions, as the probability of a correct answer to *i-th* given the ability θ, are computed by the formula

$$p_i(\theta) = \frac{1}{1 + e^{-a_i(\theta - b_i)}}$$

where a_i sets the scale of the question (this sets its discrimination ability - a steeper curve better differentiate between students), b_i is the difficulty of the question (horizontal position of a curve in space).

For question selection step of CAT we use item information of a question i that it is given by the formula

$$I_i(\theta) = \frac{(p_i'(\theta))^2}{p_i(\theta)q_i(\theta)}$$

where p_i' is the derivation of the item response function p_i. This item information provides one, and most straightforward, way of the next question selection. In every step the question X^* which is selected is one with the highest item information.

$$X^*(\theta) = \arg\max_i I_i(\theta)$$

This approach minimizes the standard error of the test procedure (Hambleton et al., 1991) because the standard error of measurement SE_i produced by $i - th$ item is defined as

$$SE_i(\theta) = \frac{1}{\sqrt{I_i(\theta)}}.$$

This means that the better precision of difficulty we are able to achieve while asking questions the smaller error of measurement.

The result of CAT simulation is displayed in the Figure 1. We can notice that this model is able to choose correct

[2] There are variants of multidimensional IRT model where it is possible to have more then one ability but in this section we are going to discuss only models with one only.

questions to ask very quickly and its prediction success rises after asking the first two. After these questions it can not improve much any more. This is caused by the simplicity of the model.

4 BAYESIAN NETWORKS

In this section we use Bayesian networks (BN) as CAT models. Details about BNs can be found in (Nielsen and Jensen, 2007; Kjærulff and Madsen, 2008). The use of BNs in educational assessment is discussed in (Almond et al., 2015; Culbertson, 2015; Millán et al., 2010). This topic is also discussed, for example, in (Vomlel, 2004a,b).

A Bayesian network is a probabilistic graphical model, a structure representing conditional independence statements. It consists of the following:

- a set of variables (nodes),
- a set of edges,
- a set of conditional probabilities.

Specific details about the use of BNs for CAT can be found in (Plajner and Vomlel, 2015). Types of nodes in our BNs correspond to types of variables defined in the Section 2. In this paper we use question nodes with only Boolean states, i.e., question is either correct or incorrect. Edges are defined usually between skills and questions (we present examples of connections in figures). Conditional probability values have been learned using standard EM algorithm for BN learning.

In this paper we use a modified method for model scoring compared to the method used in our previous research. The current method is described in the section 2. The difference is that in this case we estimate answers to all questions in the question pool and then compare to real answers in every step. In the previous version we were estimating answers only to unanswered questions in every step. It led to a skewed results interpretation because the value in the denominator of the success rate

$$\mathrm{SR}_s^t = \frac{\sum_{X_i \in \mathcal{X}} I(x_i^* = x_i')}{|\mathcal{X}|}$$

was decreasing in every step. The modified version is comparing all questions and because of that the denominator stays the same in every step.

From previous models we selected the model marked as "b3" and "expert". The former means that it has Boolean answer values, there is one skill variable having 3 states and no additional information (personal data of students) was used. See Figure 2 for its structure. The later is an expert model with 7 skill nodes (each having 2 states), Boolean answer values and no additional information about students was used. See Figure 3 for its structure.

In this paper we present three new BN models. The first two are modifications of "b3" model. They have the same structure and differ only in the number of states of their skill node. We present experiments with 4 and 9 states. We performed experiments with different numbers of states as well, but they do not provide more interesting results. Next, we add a modified expert model. This modified model has also Boolean questions and no additional information. We have added one state to 7 skill nodes from the previous version (they have 3 states in total now). The reason for this addition is an analysis of the question selection criterion. We select questions by minimizing the expected entropy over skill nodes. With only two states it means that we are pushing a student into one or the other side of the spectrum (basically, we want him to be either good or bad). With 3 states we allow them to approach mediocre skill quality as well. Moreover, we realized that the model structure as in the Figure 3 has only skills that are very specialized. We introduce a new 8th skill node which connect previous 7 skill nodes. Its representation is an overall mathematical skill combining all other skills. It allows skills on the lower level to influence each other and to provide evidence between themselves. The final model structure is in the Figure 4.

All models are summarized in the Table 1. Results of CAT simulation with BN models are displayed in the Figure 5. Increasing the number of states of one skill node improved prediction accuracy of the model (simple_4s, simple_9s), but only slightly. As we can see, one additional state (4 states in total) is better than more states (9). This confirms our expectation that simply adding node states can not improve the model quality for long due to over fitting of the model. Next, we can observe that there is a large difference between the new and the old expert model. The success rate of the new version exceeds all other models. Adding additional skill node connecting other skills proved to be a correct step. Possibilities in the model structure are still large and it remains to be explored how to create the best possible structure.

5 NEURAL NETWORKS

Neural networks are models for approximations of nonlinear functions. For more details about NNs, please refer to (Haykin, 2009; Aleksander and Morton, 1995).

There are three different parts of a NN:

1. an input layer,
2. several hidden layers, and
3. an output layer.

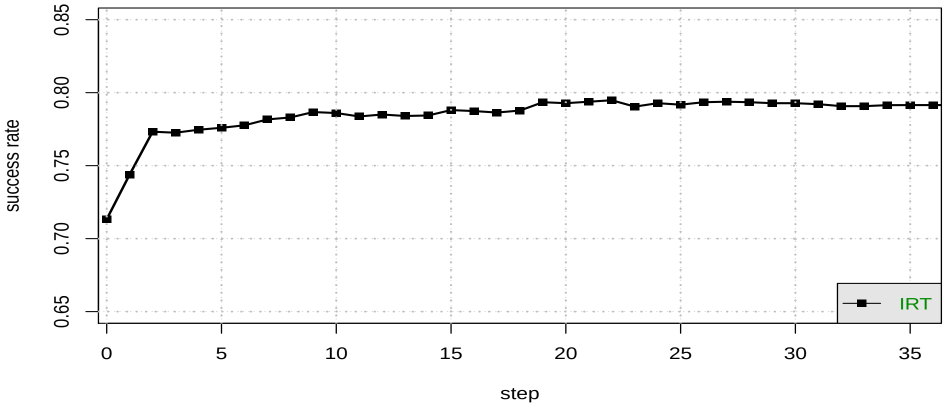


Figure 1: Success rates of IRT model

Model name	Figure	No. of skill nodes	No. of states of skill nodes
simple_3s	2	1	3
simple_4s	2	1	4
simple_9s	2	1	9
expert_old	3	7	2
expert_new	4	7+1	3

Table 1: Overview of Bayesian network models

We use NN as a student model. We feed student answers to the input layer. These values are transformed to the hidden layer(s). There is no general rule how to choose the number of hidden layers and their size. In our case we performed experiments with one hidden layer of different sizes. The hidden layer then further transforms to the output layer. NNs are not suitable for unsupervised learning. Because of that, we do not estimate an unknown student skill in the output layer. We would not have any target value needed during the learning step of the NN. Instead of that, we estimate the score (the test result) of a student directly. The score of a student is known for every student at the time of learning. The output layer then provide an estimate of this score. Nevertheless, this score is a corresponding variable to skill variables described in the Section 2

To select the next question we use the following procedure. We want the selected question to provide us as much information as possible about the tested student. That means that a student who answers incorrectly should be as far as possible on the score scale from another who answers correctly. Let the $S|X_{i,x}$ be the score prediction after answering the $i-th$ question's state x, $P(X_{i,x})$ the probability of state x to be the answer to the question i. $P(X_{i,x})$ can be obtained, for example, by statistical analysis of answers. We select a question X^* maximizing the variance of predicted scores:

$$\begin{aligned} X^* &= \arg\max_i \underset{\mathrm{x}}{\mathrm{Var}}(SC|X_{i,x}) \\ &= \sum_x P(X_{i,x})(SC|X_{i,x} - \overline{SC|X_i}), \text{where} \\ \overline{SC|X_i} &= \sum_x P(X_{i,x}) SC|X_{i,x} \end{aligned}$$

is the mean value of predicted scores.

In our experiment we used only one hidden layer with many different numbers of hidden neurons. From them we select models with 3, 5, and 7 neurons in the hidden layer because they provide the most interesting results. The structure of the network with 5 hidden neurons is in the Figure 6. Results of CAT simulation with NN models are displayed in the Figure 7. As we can see in this figure, the quality of estimates while using NNs increases very slowly. This may be caused by the question selection criterion. If we were selecting better questions, it is possible that the success rate would be increasing faster. It remains to be explored which selection criterion would provide such questions. Nevertheless, this better question selection does not change the

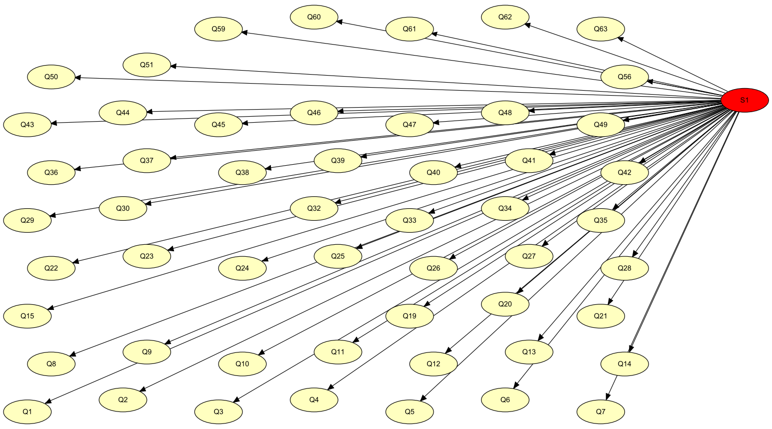


Figure 2: Bayesian network with one hidden variable and personal information about students

final prediction power of the model (the maximal success rate would not be exceeded). This prediction power could be increased by using a different version of NNs. More specifically, we will perform experiments with one of the recurrent versions of NNs, i.e., Elman's networks or Jordan's networks.

6 MODEL COMPARISON AND CONCLUSIONS

We present a graphical comparison of all three model types in the Figure 8. One model is selected from each type. We can see that the neural network model scored the worst result. This may be further improved by a better NN structure and better question selection process. The new BN expert model is scoring the best. Even in this case we believe that further improvements are possible to increase its success rate. We will focus our future research into methods for BN models creation and criteria for their comparison. Especially, we would like to use a concept of the local structure in BN models (Díez and Druzdzel, 2007). That would allow us to create more complex models, yet with less parameters to be estimated during learning. Both previous models can be compared with the IRT model which is the standard in the field of CAT.

Acknowledgements

The work on this paper has been supported from GACR project n. 16-12010S.

References

Aleksander, I. and Morton, H. (1995). *An Introduction to Neural Computing*. Information Systems. International Thomson Computer Press.

Almond, R. G. and Mislevy, R. J. (1999). Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237.

Almond, R. G., Mislevy, R. J., Steinberg, L. S., Yan, D., and Williamson, D. M. (2015). *Bayesian Networks in Educational Assessment*. Statistics for Social and Behavioral Sciences. Springer New York, New York, NY.

Culbertson, M. J. (2015). Bayesian Networks in Educational Assessment: The State of the Field. *Applied Psychological Measurement*, 40(1):3–21.

Díez, F. J. and Druzdzel, M. J. (2007). Canonical Probabilistic Models for Knowledge Engineering. Technical report, Research Centre on Intelligent Decision-Support Systems.

Hambleton, R. K., Swaminathan, H., and Rogers, H. J. (1991). *Fundamentals of item response theory*, volume 21. SAGE Publications.

Haykin, S. S. (2009). *Neural Networks and Learning Machines*. Number v. 10 in Neural networks and learning machines. Prentice Hall.

Kjærulff, U. B. and Madsen, A. L. (2008). *Bayesian Networks and Influence Diagrams*. Springer.

Lín, V. (2005). Complexity of Finding Optimal Observation Strategies for Bayesian Network Models. In *Proceedings of the conference Znalosti*, Vysoké Tatry.

Lord, F. M. and Novick, M. R. (1968). *Statistical Theories of Mental Test Scores*. (Behavioral science : quantitative methods). Addison-Wesley.

Millán, E., Loboda, T., and Pérez-de-la Cruz, J. L. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683.

Nielsen, T. D. and Jensen, F. V. (2007). *Bayesian Networks*

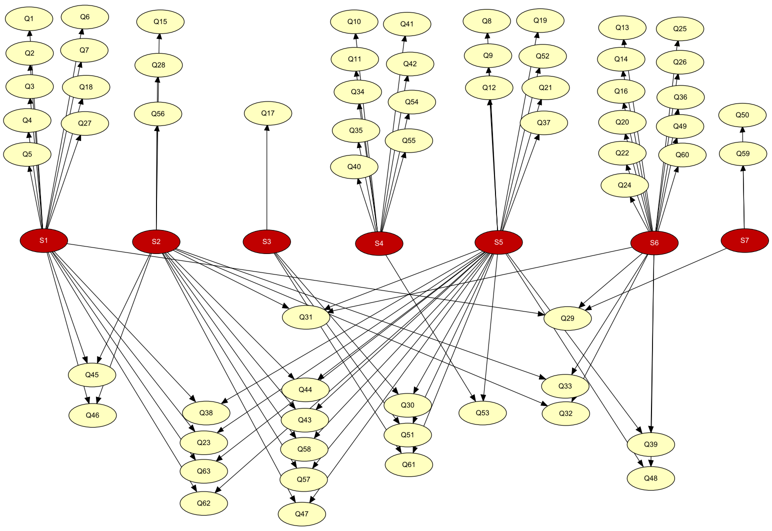


Figure 3: Bayesian network with 7 hidden variables (the old expert model)

and Decision Graphs (Information Science and Statistics). Springer.

Plajner, M. and Vomlel, J. (2015). Bayesian Network Models for Adaptive Testing. Technical report, ArXiv: 1511.08488, http://arxiv.org/abs/1511.08488.

Rasch, G. (1960). *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut.

Rasch, G. (1993). *Probabilistic Models for Some Intelligence and Attainment Tests. Expanded Ed.* MESA Press.

van der Linden, W. J. and Glas, C. A. W. (2000). *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers.

Vomlel, J. (2004a). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):83–100.

Vomlel, J. (2004b). Buliding Adaptive Test Using Bayesian Networks. *Kybernetika*, 40(3):333–348.

Wainer, H. and Dorans, N. J. (2015). *Computerized Adaptive Testing: A Primer*. Routledge.

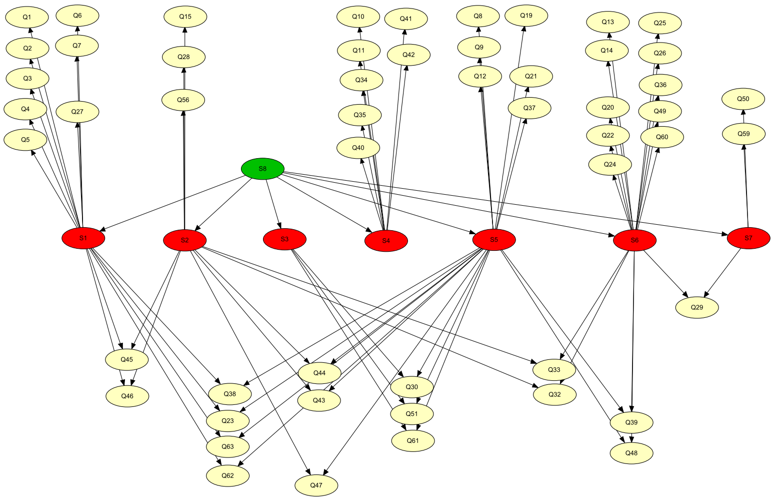


Figure 4: Bayesian network with 7+1 hidden variables (the new expert model)

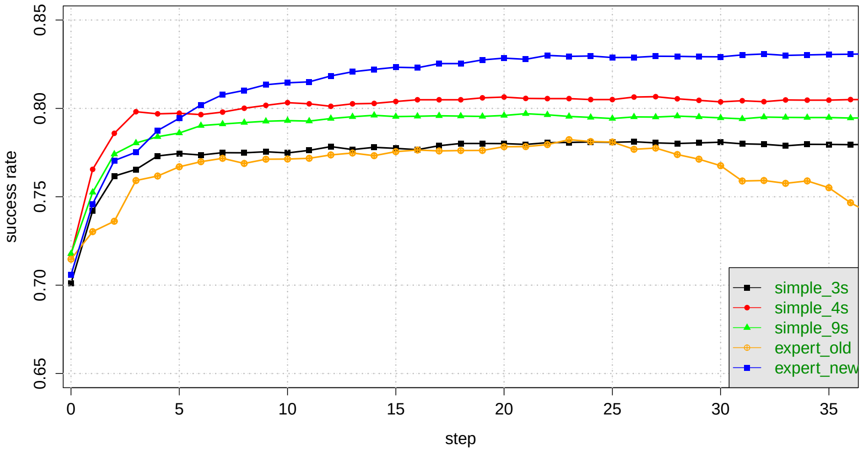


Figure 5: Results of CAT simulation with BNs

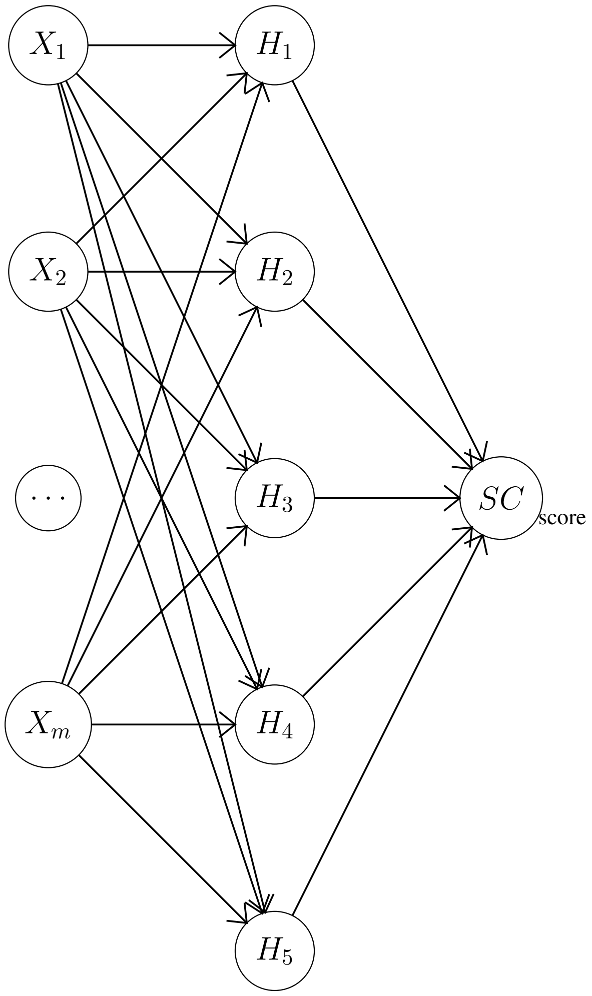


Figure 6: Neural network with 5 hidden neurons

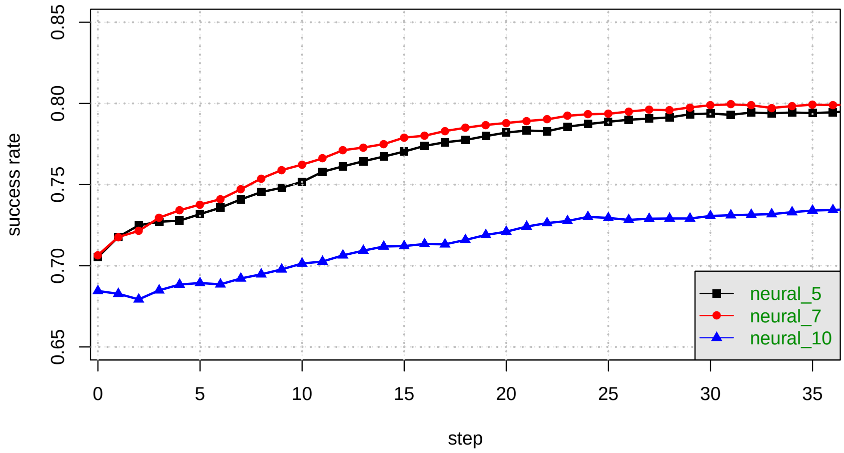


Figure 7: Results of CAT simulation with NNs

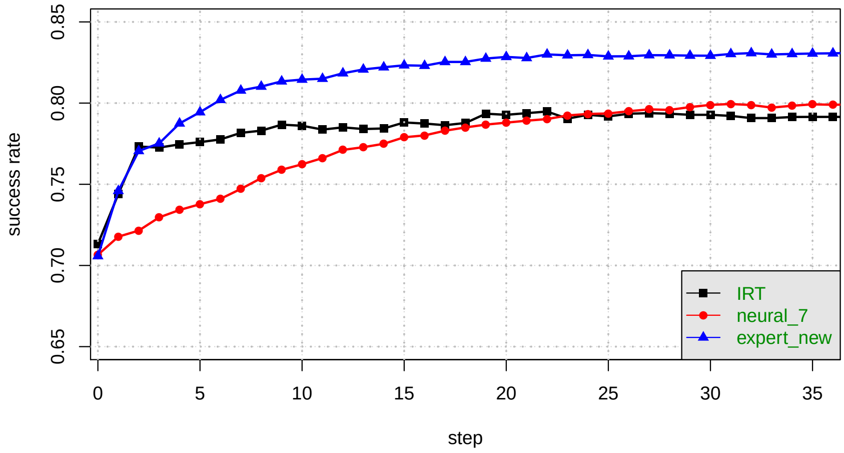


Figure 8: CAT simulation results comparison

Student Skill Models in Adaptive Testing

Martin Plajner PLAJNER@UTIA.CAS.CZ
Jiří Vomlel VOMLEL@UTIA.CAS.CZ
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4
Prague 8, CZ-182 08
Czech Republic

Abstract

This paper provides a common framework, a generic model, for Computerized Adaptive Testing (CAT) for different model types. We present question selection methods for CAT for this generic model. We use three different types of models, Item Response Theory, Bayesian Networks, and Neural Networks, that instantiate the generic model. We illustrate the usefulness of a special model condition – the monotonicity – and discuss its inclusion in these model types. With Bayesian networks we use specific type of learning using generalized linear models to ensure the monotonicity. We conducted simulated CAT tests on empirical data. Behavior of individual models was assessed based on these tests. The best performing model was the BN model constructed by a domain expert; its parameters were learned from data under the monotonicity condition.

Keywords: Bayesian Networks; Computerized Adaptive Testing; Generalized Linear Models; Item Response Theory.

1. Introduction

Testing human abilities and human knowledge is frequent in the modern society. The computerized form of testing is also getting an increasing attention with the growing spread of computers, smart phones and other devices which allow easy contact with the test audience. This paper focuses on Computerized Adaptive Testing (CAT) (Wainer and Dorans, 1990; Almond and Mislevy, 1999; van der Linden and Glas, 2000, 2010). CAT is a concept of testing where an examinee is performing a computer administered and computer controlled test. The computer system selects questions for a student taking the test and it evaluates his/her performance. This is being done in order to create a shorter version of the test by asking correct questions (tailored for each particular student). If performed properly the measurement of student's ability/knowledge has better precision (Pine and Weiss, 1978), the test is more fair, the student is better motivated, and less time is consumed (Moe and Johnson, 1988; Tonidandel et al., 2002).

In this paper we introduce a framework for CAT. This framework is formed by a generic model and associated methods. The goal is to provide a unifying probabilistic graphical model for diverse models. The CAT process can be divided into two phases: model creation and testing. In the former, the student model is created. In the later, the model is used to actually test examinees. In the Section 2 we present a generic structure which is further used to nest different probabilistic models. This allows us to summarize similarities in different modeling approaches. Next, in the Section 3 we discuss the procedure of testing and associated methods. After establishing this generic structure, we present specific examples of models to be filled into it. We go through the use of Item Response Theory (IRT), which is a model regularly used for CAT and Bayesian and neural networks (BNs and NNs), which are both models commonly used in many areas of artificial intelligence for a large variety of tasks. We conducted simulated CAT tests on an empirical dataset which we collected for this purpose. This allows us to compare two model types (BN and NN) which are new in the field of CAT with the standard IRT model. The overview of the dataset, experimental setup and experimental results are presented in the concluding parts of this paper.

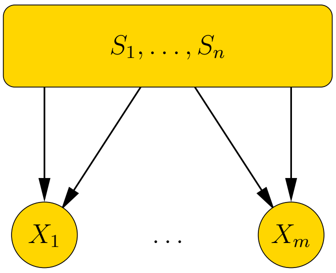
2. Student Skill Models

The student model is a tool which models a student. It provides assumptions about his/her skills, expected score and other variables. There are many different student model types (Culbertson, 2015) which can be used for adaptive testing. In this work we present a common framework which views them as special cases of one generic model for CAT.

2.1 Generic Student Model

The generic student model has the following two types of variables:

- A set of n variables we want to estimate $\mathcal{S} = \{S_1, \ldots, S_n\}$. These variables represent skills (abilities, knowledge) of a student. We will call them skills or skill variables. We will use symbol $\boldsymbol{S}$ to denote the multivariable $\boldsymbol{S} = (S_1, \ldots, S_n)$ taking states $\boldsymbol{s} = (s_{1,i_1}, \ldots, s_{n,i_n})$.
- A set of m questions $\mathcal{X} = \{X_1, \ldots, X_m\}$. We will use the symbol $\boldsymbol{X}$ to denote the multivariable $\boldsymbol{X} = (X_1, \ldots, X_m)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_m)$.



Skills $\mathcal{S}$ are either continuous or discrete variables. In the continuous case they provide values which can be interpreted as levels of skills. They also naturally make an ordering of students. Discrete variables can be Boolean (true/false) or categorical. Boolean variables inform us that a student has or has not the particular skill. Categorical variables are sampled from the continuous case. Their states are different skill levels a student can have. Ordering of students can be done by the value of expected skill computed from probabilities of each state. In addition we differentiate between observed and unobserved skills (in the training sample). In the case of observed skills we measure them by a certain metric (for example, score of the test), or they are produced by an expert from a test results analysis. In the case of unobserved skills their states are not known even for students with complete test results.

Questions $\mathcal{X}$ are discrete variables having Boolean or categorical states. Boolean for correct/incorrect answers, categorical for multiple choice answers. The subset of questions which are already answered forms evidence

$$e = \{X_{i_1} = x_{i_1}, \ldots, X_{i_k} = x_{i_k} | i_1, \ldots, i_k \in \{1, \ldots, m\}\}.$$

Links connecting skills $\mathcal{S}$ and questions $\mathcal{X}$ define the relationship between these two sets. $\mathcal{S}_{pa(i)} \subseteq \mathcal{S}$, with the respective multivariable $\boldsymbol{S}_{pa(i)} = \boldsymbol{s}_{pa(i)}$, denotes parents of the question

X_i. Then the probability of a correct answer to $i-th$ question (or the probabilities of the specific item answered) is: $P(X_i = x_i|\boldsymbol{S}_{pa(i)})$ has to be provided in the model. In the case of continuous skills these probabilities are given by a continuous link function $p_i(X_i = 1|\boldsymbol{S}_{pa(i)})$ giving the probability of a correct answer based on $\boldsymbol{S}_{pa(i)}$. In the case of discrete skills, probabilities are in the form of conditional probability tables (CPTs). Because the state of $\boldsymbol{S}_{pa(i)}$ is directly influenced by the evidence e we will also use shorthanded notation $p_i(X_i = 1|e)$ and $P(X_i = x_i|e)$. We assume all questions are conditionally independent given skills, i.e., $X_i \perp\!\!\!\perp X_j|\boldsymbol{S}, \forall i \neq j$. The joint probability distribution is then $P(\boldsymbol{X}, \boldsymbol{S}) = P(\boldsymbol{S}) \cdot \prod_{i=1}^{m} P(X_i|\boldsymbol{S}_{pa(i)})$

All together it forms a graphical probabilistic model. It is formed by vertices $\mathcal{S} \cup \mathcal{X}$, edges between them and associated parameters with these edges. In order to create this model, we have to establish its structure and learn parameters. In this paper we will not discuss the former and we will focus only on the later. One way of obtaining necessary parameters is to ask an expert to provide them based on his/her knowledge of the field. This option is very demanding (in terms of knowledge of the expert as well as time) because the space of parameters associated with the model is very large. The other way is to learn probabilities by a machine learning approach from collected data. Even this approach has issues with the large space of parameters and a large volume of quality samples has to be provided in order to obtain statistically reliable estimations. The automated learning of parameters is discussed in this paper.

2.2 Monotonicity

For the needs of adaptive testing it is reasonable to require relations between skills and questions to be isotone in the distribution (the model to be monotonic) (van der Gaag et al., 2004). First, we create an ordering on states s, s' of i-th student skill S_i: $s_i \preceq s'_i$. It means that we are able to say which of these states is better (or the same). The monotonic model then ensures that probabilities of higher ordered states are also always higher (isotone) or always lower (antitone), i.e.:

$$\begin{array}{lcl} s_i \preceq s'_i & \rightarrow & P(X = x|S_i = s_i) \ \leq \ P(X = x|S_i = s'_i) \text{ , or} \\ s_i \preceq s'_i & \rightarrow & P(X = x|S_i = s_i) \ \geq \ P(X = x|S_i = s'_i) \end{array}$$

For example, to avoid the following situation: “With the low level of student’s skills the probability of a correct answer is small. With the medium level the probability is large. And with the high level it is small again.” Skill states should reflect a certain ability level, thus we expect a positive or negative correlation of the skill and student’s answers.

3. Testing Process

Regardless of the model we choose the testing part follows always the same scheme. With the prepared and calibrated model, CAT repeats following steps:

- A question is selected, this question is asked and an answer is obtained.
- The answer is inserted into the model, the model (which provides estimates of the student’s skills) is updated.
- (optional) Answers to all questions are estimated given the current estimates of student’s skills.

This procedure is repeated as long as necessary which means until we reach a termination criterion. This criterion can be either a time restriction, the number of questions, or a confidence interval of the estimated variables. Each of these criteria would lead to a different learning strategy (Vomlel, 2004a), but finding an optimal strategy is NP-hard for these criteria (Lín, 2005). We have chosen an heuristic approach based on greedy optimization methods. This approach selects the next question during the testing in every step based on a given rule. There is a large variety rules which can be used for this task. We present some of them in the following section.

3.1 Question selection criteria

In this section we present three various criteria for question selection C_j, where $j \in \{1, 2, 3\}$ is an index of a criterion. Each of them works with the evidence about the student e and outputs a value for the question X_i. The selected question X^* is a question from all unanswered questions maximizing this criterion given the evidence:

$$X^*(e) = \arg\max_{X_i} C_j(X_i, e)$$

3.1.1 ITEM INFORMATION

For the continuous variables $\mathcal{S}$, links to questions are given by functions $p_i(X_i = 1|e)$ (for binary questions). The item information that is given by $i - th$ question is then

$$C_1(X_i, e) = I(X_i, e) = \frac{(p_i'(X_i = 1|e))^2}{p_i(X_i = 1|e)(1 - p_i(X_i = 1|e))}$$

where p_i' is the derivation of p_i. This item information provides one, and most straightforward, way of the next question selection in the continuous case. It is derived form the Item Response Theory's classical way of measuring information, e.g., in van der Linden and Hambleton (2013). This approach minimizes the standard error of the test procedure in each step because the standard error of measurement $\mathrm{SE}(X_i, e)$ produced by the question X_i is defined as

$$\mathrm{SE}(X_i, e) = \frac{1}{\sqrt{I(X_i, e)}}.$$

This means that the smallest error is produced by questions which are steep and their probability of a correct answer is close to 50% given the current level of skill.

3.1.2 ENTROPY REDUCTION

This approach is based on reducing the expected value of entropy after asking a question. In the following text we provide formulas for discrete case, but with minimal changes it is applicable to continuous variables as well. The cumulative Shannon entropy over all skill variables of $\mathcal{S}$ given the evidence e is

$$H(e) = \sum_{k=1}^{n} \sum_{\ell=1}^{i_n} -P(S_k = s_{k,\ell}|e) \cdot \log P(S_k = s_{k,\ell}|e).$$

The entropy $H(e)$ is the sum of individual entropies over all skill nodes. Another option would be to compute the entropy of the joint probability distribution of all skill nodes. This would take into

model	skill variables type	no. skill variables	QS criterion
IRT	continuous, unobserved	1	item information
BN	discrete, unobserved	1. . . many	entropy reduction
NN	continuous, observed	1	students separation

Table 1: Models summary

account correlations between these nodes. In our task we want to estimate marginal probabilities of all skill nodes. In the case of high correlations between two (or more) skills the second criterion would assign them a lower significance in the model. This is the behavior we wanted to avoid. The first criterion assigns the same significance to all skill nodes which is a better solution. For our problem, the greedy strategy based on the sum of entropies provides good results. Moreover, the computational time required for the proposed method is lower.

Assume we decide to ask a question $X_i \in \mathcal{X}_s$ with possible outcomes $x_1, \ldots, x_{p_i}$. The new value of entropy is then computed as $H(e_{i,j}) = H(e \cup \{X_i = x_j\})$. The expected entropy after answering question X_i is

$$\begin{aligned} EH(X_i, e) &= \sum_{j=1}^{p} P(X_i = x_j|e) \cdot H(e_{i,j}) \ . \\ C_2(X_i, e) = IG(X_i, e) &= H(e) - EH(X_i, e) \end{aligned}$$

gives us the information gain criterion.

3.1.3 STUDENTS SEPARATION MAXIMIZATION

The last criterion, we are proposing, maximizes the distance between students within skills. That means that a student who answers incorrectly should be as far as possible on the skill scale from the one who answers correctly. We present this criteria for a single skill variable S_1 while an extension to more variables is possible. Let $s_j|e_{i,j}$ be the predicted value of skill S_1 given extended evidence $e_{i,j} = e \cup \{X_i = x_j\}$ and $(\bar{s}|e_{i,j})$ be its mean value. Then we get the variance of S_1 given evidence $e_{i,j}$

$$C_3(X_i, e) = \sum_{j=1}^{p} ((\bar{s}|e_{i,j}) - (s_j|e_{i,j}))^2 \cdot P(X_i = x_j|e) \ .$$

4. Specific Models for CAT

We present three specific model types fitting into the generic CAT student model: Item Response Theory (IRT), Bayesian networks (BN) and neural networks (NN). Basic properties of these models are summarized in Table 1. We used question selection criteria with models as indicated in the column QS selection. These presented choices are the most natural for the particular model but in general, with modifications, they should be interchangeable.

4.1 Item Response Theory

The beginning of Item Response Theory (IRT) stems back to 5 decades ago and there is a large amount of literature available, for example, Rasch (1960); Lord and Novick (1968). IRT allows more precise measurement of a certain ability of an examinee than classical test theory[1]. It is expected a student has a skill[2] which directly influences his/her chances of answering questions correctly. In this case skills of the generic model defined in Section 2 reduce to $\mathcal{S} = \{S_1\}$. It is a continuous variable. Links of generic model are filled by item response functions (IRF) which are probabilities of a successful answer given S_1. In our research we use 2PL IRT model which is in the form

$$p_i(X_i = 1|S_1 = s_1) = \frac{1}{1 + e^{-a_i(s_1 - b_i)}}$$

where a_i sets the scale of the IRF (the discrimination ability - a steeper curve $=$ better differentiation between students), b_i is the difficulty of the question (the position of a curve in space), $a_i, b_i \in R$. Generally, we observe small (positive or negative) numbers. In this case there is one link from the skill S_1 to each question in $\mathcal{X}$. Parameters of IRFs are usually fitted using maximum likelihood estimation from dataset. It is also possible to obtain these parameters from an expert. Given the format of item response functions, IRT[3] model satisfies monotonicity property as described in Section 2.

4.2 Bayesian Networks

Bayesian networks are probabilistic graphical models, their structure represents conditional independence statements. Details about BNs can be found in, for example, (Pearl, 1988; Nielsen and Jensen, 2007; Kjærulff and Madsen, 2008). The use of BNs in educational assessment is discussed, e.g., by Almond and Mislevy (1999); Vomlel (2004a,b); Millán et al. (2010); Almond et al. (2015); Culbertson (2015).

A Bayesian network consists of: a set of variables (nodes), a set of edges, a set of conditional probabilities. In our case the set of variables is formed by questions $\mathcal{X}$ and skills $\mathcal{S}$ from the generic model. The number of skills can vary from 1 to many. The set of edges is formed by connections between skills and from skills to questions where one questions can have more influencing skills. An example can be found in Figure 1(a). All variables are discrete. Each variable has an associated CPT which describes a probability for every configuration of its parents (structure given by edges).

Parameters can be obtained from an expert in the field, or we can use a machine learning approach from dataset. Skills in the model are not observed and thus it is necessary to use a method capable of handling missing data. Most often, the EM algorithm (Lauritzen, 1995) is used.

4.3 Monotonicity in BNs

When using the general EM algorithm the monotonicity property cannot be ensured. In order to make sure that the model satisfies the monotonicity property it is necessary to restrict CPTs to be only of a specific form. We build on ideas from Rijmen (2008); Restificar and Dietterich (2013), where generalized linear models are used to create CPTs.

[1] Classical test theory focuses on the test as a whole, measuring the score as a sum of questions with the same difficulty. IRT on the other hand views questions as individual items with different difficulties.

[2] In the field of IRT it is often called ability or proficiency.

[3] The structure of IRT model could be also modeled by a special type of Bayesian Network but we will not go into details in this article.

The CPT of a question X_i is from a binomial family model (glm model with the logit link function). $\boldsymbol{\alpha_i}, \boldsymbol{\beta_i}$ are its parameters and the model takes the form:

$$P(X_i = 1 | \mathcal{S}_{pa(i)} = s_{pa(i)}) \quad = \quad \frac{\exp(\boldsymbol{\alpha}_i + \boldsymbol{\beta}_i^T s_{pa(i)})}{1 + \exp(\boldsymbol{\alpha}_i + \boldsymbol{\beta}_i^T s_{pa(i)})} \ .$$

By calculating this value for every possible state combination of affecting skills, we are able to fill the CPT. The problem with finding the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is that with the glm model we usually observe variables from $\mathcal{S}$. In this case they are unknown. The situation is solvable with a version of the EM algorithm for GLM models. Ibrahim et al. (2005) presents an algorithm for partially unobserved variables. This approach ensures the model is not violating the monotonicity property.

4.4 Neural Networks

Neural networks are models for approximations of non-linear functions. For more details about NNs, please refer, e.g., to Aleksander and Morton (1995); Haykin (2009). There are three different parts of a NN: an input layer, several hidden layers, and an output layer.

In our NN model the input layer is formed by questions $\mathcal{X}$ from the generic model defined in Section 2. From this layer the NN transforms to intermediate hidden layers. Nodes of these hidden layers represent unobserved uninterpretable skill variables. There is no general rule how to choose a number of hidden layers and their size. Variants we experimented with are further detailed in Section 5. The intermediate layers tranform to the output node which is a single observed student skill. This skill (S_0) is directly measured by the score of a test. The output node and hidden layers form skills $\mathcal{S}$. The choice of using an observed variable in this case is because NNs are not suitable for unsupervised learning, unless having special structure. We need to have a target value during the learning step of the NN. The score of a student is known for every student at the time of learning. During the CAT test the output layer then provides an estimate of the score of the currently tested student. For inverse estimations of answers based on student's skill the NN structure is reversed. These two networks are learned separately and each performs its own task.

Links between nodes form a function, $f(S_0|e) : \mathrm{R}^m \rightarrow \mathrm{R}$, through NN's intermediary hidden layers providing the score value. Reversed structure then provides functions $p_i(X_i|S_0) : \mathrm{R} \rightarrow \mathrm{R}$. These function break down to the regular NN neuron activation and combination functions (for example, multi layered perceptron or radial basis functions). Learning methods are also common NN methods, i.e., usually backpropagation.

5. Experiments

To verify the concepts presented in this paper we have collected empirical data. We designed a paper test of mathematical knowledge of grammar school students. The test focuses on simple functions (mostly polynomial, trigonometric, and exponential/logarithmic). Students were asked to solve various mathematical problems[4] including graph drawing and reading, calculating points on the graph, root finding, describing function shapes and other function properties. All together, we have obtained 281 test results. Details about data can be found in Plajner and Vomlel (2015). The

[4] In this case we use the term mathematical "problem" due to its nature. In general tests, terms "question" or "item" are often used. In this article all of these terms are interchangeable.

model evaluation was done for each model of each type that is described in following sections. We used 10-fold cross-validation method.

5.1 Results evaluation

To evaluate models we performed a simulation of the CAT test for every model and for every student. During testing we first estimated the skill(s) of a student based on his/her answers. Then, based on these estimated skills we used the model to estimate answers to all questions $X_i \in \mathcal{X}$. More specifically: Let the test be in the step s ($s - 1$ questions asked). At the end of the step s (after updating a model with new answer) we compute marginal probability distributions for all skills $\boldsymbol{S}$. Then we use this to compute estimations of answers to all questions, where we select the most probable state of each question[5] $X_i \in \mathcal{X}$:

$$x_i^* = \arg\max_{x_i'} P(X_i = x_i' | \boldsymbol{S}).$$

By comparing this value to the real answer to $i - th$ question x_i for each question we obtain a success ratio

$$\text{SR} = \frac{\sum_{X_i \in \mathcal{X}} f(x_i^* = x_i)}{|\mathcal{X}|} \text{ , where } f(expr) = \begin{cases} 1 & \text{if } expr \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

The total success ratio of a model in a step is the average of success ratios of all tests in the same step. We compare models based on this total success ratios. The quality of models could be assessed also in other ways. One of the main goals of a student model is to predict abilities of students. As such it would be reasonable to measure the quality of these predictions. Unfortunately, this is hard to achieve because these skills are usually hidden variables. It is possible to create an indicator such as student's overall performance or his/her known qualities. Due to the nature of our data set we do not have any of these options and because of that we decided to use the approach described above.

5.2 Models

We have performed testing with different model versions. The best IRT, BN, and neural network models are compared together in Figure 1(c). We select the most important representatives from each group. Below we present an overview of these versions.

IRT is a commonly used model that can be considered as a base model to compare with. We especially wanted to provide a comparison with other models. As we can see in the Figure 1(c) this model's performance is exceeded by many other models.

The first group of BN models, we experimented with, has one or two skill nodes which connect to all questions. These skill nodes have different number of states. We selected the best performing model and it is labeled as "simple_2x3" as it has two skill nodes each having 3 states. To satisfy the monotonicity requirement of BN models we have implemented a version of the EM algorithm. Models which are learned using this algorithm are labeled with additional "glm". The rest is learned with the Hugin EM algorithm (Hugin, 2014). The source code of our version of the EM algorithm and other algorithms used (including BN inference) is implemented in R language and it is available at the author's web page (`http://staff.utia.cas.cz/plajner`).

[5] We remind that all questions are conditionally independent given skills, i.e., $X_i \perp\!\!\!\perp X_j | \boldsymbol{S}, \forall i \neq j$.

The second group of BN models is based on our expert knowledge in the field of the test. We identified several skills each connecting to a specific subset of questions which are relevant to the skill represented by the variable. One version of this network is shown in Figure 1(a). In this particular case there are 7+1 skill nodes. 7 nodes connect directly to questions and the last one connects these skills together. This model is called "expert_new". In our experiments it appeared that the connection of skill nodes provides a substantial improvement in the performance of the models. The version of the same model, without the skill connecting all other skill nodes, is also included as "expert_old".

The result of the best performing BN model of the first group, "simple_2x3", is presented in Figure 1(c). Results of BN expert models are displayed in Figure 1(b). In this graph we can compare the performance of models learned with glm method and their counterparts. We can observe that glm models are scoring similarly during first steps but quickly outperform those with the general EM algorithm. The best BN expert model can be compared with other models in Figure 1(c).

Some of the most important facts resulting from experiments with BN models are: (1) Models with the monotonicity requirement provide better results than models without this requirement. (2) Adding a higher level node to the expert model causes significant boost in the model's performance. We believe that it is caused by the possibility of an easier transition of evidence through the network from a skill to another skill.

In our experiments with NNs we used only one hidden layer with different numbers of hidden neurons. From them we select the model with 7 neurons in the hidden layer because it provides the best results. The result of CAT simulation with this NN model is displayed in Figure 1(c). As we can see in this figure, the quality of estimates while using NNs increases very slowly. We believe this is caused by the question selection criterion. If we were selecting better questions, it is possible that the success rate would be increasing faster. It remains to be explored which selection criterion would provide such questions. Nevertheless, this better question selection does not change the final prediction power of the model (the maximal success rate in the last steps would not be exceeded). This prediction power could be increased by using a modified structure of the NN. Additional research is needed to show which NN structure is better suited for this task. In this paper we verified the general possibility of using NNs for CAT.

6. Conclusions and Future Work

In this paper we established a common generic model for CAT. This model was instantiated by three different model types. The first one, IRT, serves as a reference point. The second type were BNs which we studied the most. Especially, we discussed parameter learning which ensures the monotonicity. In experiments this method produced better results than the same model without the monotonicity condition. This is the most important empirical result of this paper and we believe that every CAT model should consider monotonicity. The third model type, NNs, did not provide the most convincing results. However, we believe that further improvements are possible.

In the future research we would like to focus on BN models because from models, we have experimented with, we see the best potential in BNs. Possible combinations and variations in the model structures are vast and it remains to be explored how to search for the best BN structure. In this article we used generalized linear models to ensure monotonicity in BNs. It is possible that this approach may introduce additional unwanted behavior. One way to resolve this is to use less restricting techniques for ensuring monotonicity, such as, for example, in Masegosa et al. (2016);

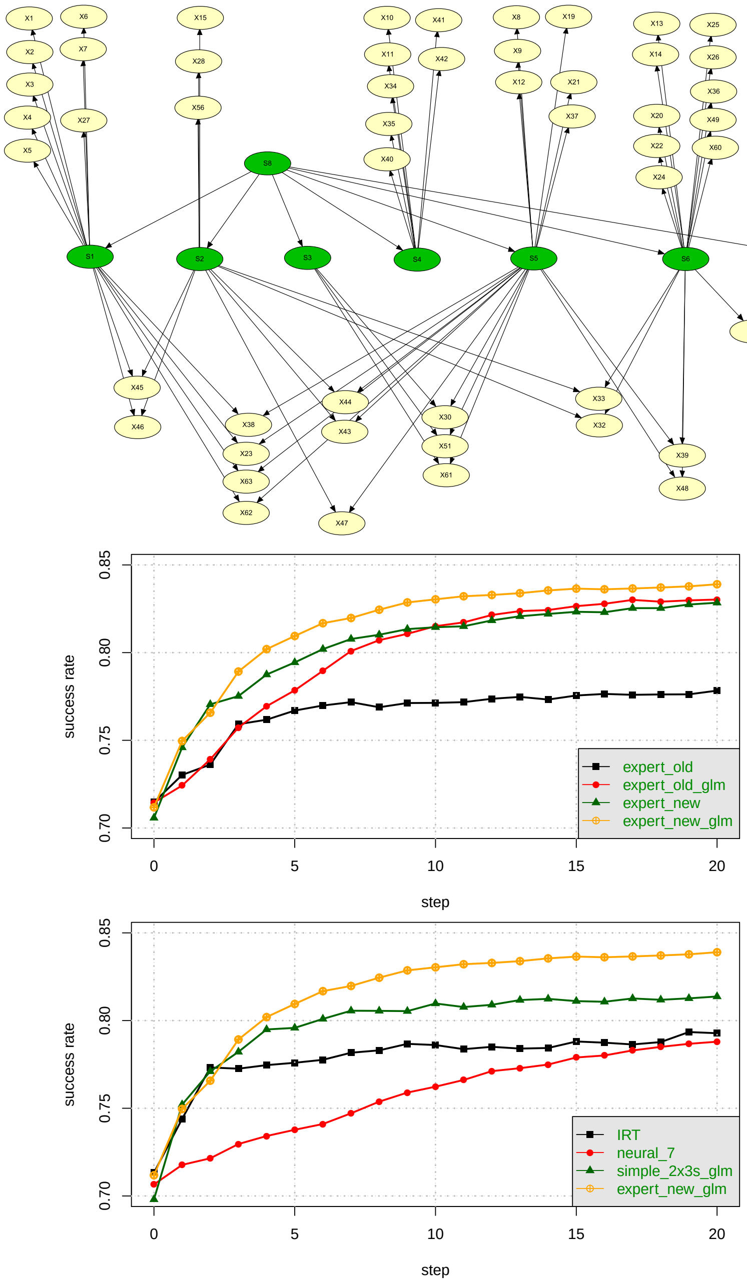


Figure 1: (a) Bayesian network structure (the expert model), (b) Expert Bayesian models success rates, (c) Models comparison success rates

de Campos et al. (2008). We plan experiments to verify the impact of glm models properties and to compare it to the less restricting option. Furthermore, we would like to introduce CPTs with a local structure (Díez and Druzdzel, 2007) which would allow us to get even larger control of the form of the BN model.

Acknowledgments

The work on this paper has been supported by
the Czech Science Foundation, GACR project No. 16-12010S and by
the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/175/OHK3/2T/14.

References

I. Aleksander and H. Morton. *An Introduction to Neural Computing*. Information Systems. International Thomson Computer Press, 1995.

R. G. Almond and R. J. Mislevy. Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237, 1999.

R. G. Almond, R. J. Mislevy, L. S. Steinberg, D. Yan, and D. M. Williamson. *Bayesian Networks in Educational Assessment*. Springer New York, 2015.

M. J. Culbertson. Bayesian Networks in Educational Assessment: The State of the Field. *Applied Psychological Measurement*, 40(1):3–21, 2015.

C. P. de Campos, Y. Tong, and Q. Ji. Constrained Maximum Likelihood Learning of Bayesian Networks for Facial Action Recognition. *Computer Vision – ECCV 2008*, 5304:168–181, 2008.

F. J. Díez and M. J. Druzdzel. Canonical Probabilistic Models for Knowledge Engineering. Technical report, Research Centre on Intelligent Decision-Support Systems, 2007.

S. S. Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2009.

Hugin. Explorer, ver. 8.0, Comput. Software 2014, http://www.hugin.com, 2014.

J. G. Ibrahim, M.-H. Chen, S. R. Lipsitz, and A. H. Herring. Missing-Data Methods for Generalized Linear Models. *Journal of the American Statistical Association*, 100(469):332–346, 2005.

U. B. Kjærulff and A. L. Madsen. *Bayesian Networks and Influence Diagrams*. Springer, 2008.

S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19(2):191–201, 1995.

V. Lín. Complexity of Finding Optimal Observation Strategies for Bayesian Network Models. In *Proceedings of the conference Znalosti*, Vysoké Tatry, 2005.

F. M. Lord and M. R. Novick. *Statistical Theories of Mental Test Scores*. (Behavioral science : quantitative methods). Addison-Wesley, 1968.

A. R. Masegosa, A. J. Feelders, and L. C. van der Gaag. Learning from incomplete data in Bayesian networks with qualitative influences. *International Journal of Approximate Reasoning*, 69:18–34, 2016.

E. Millán, T. Loboda, and J. L. Pérez-de-la Cruz. Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683, 2010.

K. C. Moe and M. F. Johnson. Participants' Reactions To Computerized Testing. *Journal of Educational Computing Research*, 4(1):79–86, jan 1988.

T. D. Nielsen and F. V. Jensen. *Bayesian Networks and Decision Graphs (Information Science and Statistics)*. Springer, 2007.

J. Pearl. *Probabilistic reasoning in intelligent systems:networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

S. M. Pine and D. J. Weiss. A Comparison of the Fairness of Adaptive and Conventional Testign Strategies. Technical report, University of Minnesota, Minneapolis, 1978.

M. Plajner and J. Vomlel. Bayesian Network Models for Adaptive Testing. Technical report, ArXiv: 1511.08488, nov 2015.

G. Rasch. *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut, 1960.

A. C. Restificar and T. G. Dietterich. Exploiting monotonicity via logistic regression in Bayesian network learning. Technical report, Oregon State University, 2013.

F. Rijmen. Bayesian networks with a logistic regression model for the conditional probabilities. *International Journal of Approximate Reasoning*, 48(2):659–666, 2008.

S. Tonidandel, M. A. Quiñones, and A. A. Adams. Computer-adaptive testing: the impact of test characteristics on perceived performance and test takers' reactions. *The Journal of applied psychology*, 87(2):320–32, apr 2002.

L. C. van der Gaag, H. L. Bodlaender, and A. J. Feelders. Monotonicity in Bayesian networks. *20th Conference on Uncertainty in Artificial Intelligence*, pages 569–576, 2004.

W. J. van der Linden and C. A. W. Glas. *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers, 2000.

W. J. van der Linden and C. A. W. Glas, editors. *Elements of Adaptive Testing*. Springer NY, 2010.

W. J. van der Linden and R. K. Hambleton. *Handbook of Modern Item Response Theory*. Springer NY, 2013.

J. Vomlel. Buliding Adaptive Test Using Bayesian Networks. *Kybernetika*, 40(3):333–348, 2004a.

J. Vomlel. Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(supp01):83–100, 2004b.

H. Wainer and N. J. Dorans. *Computerized Adaptive Testing: A Primer*. Routledge, 1990.

Monotonicity in Bayesian Networks for Computerized Adaptive Testing*

Martin Plajner[1,2] and Jiří Vomlel[2]

[1] Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University, Prague
Trojanova 13, Prague, 120 00, Czech Republic
plajner@utia.cas.cz,
http://staff.utia.cas.cz/plajner/

[2] Institute of Information Theory and Automation,
Czech Academy of Sciences,
Pod vodárenskou věží 4, Prague 8, 182 08, Czech Republic
vomlel@utia.cas.cz,
http://www.utia.cas.cz/vomlel/

Abstract. Artificial intelligence is present in many modern computer science applications. The question of effectively learning parameters of such models even with small data samples is still very active. It turns out that restricting conditional probabilities of a probabilistic model by monotonicity conditions might be useful in certain situations. Moreover, in some cases, the modeled reality requires these conditions to hold. In this article we focus on monotonicity conditions in Bayesian Network models. We present an algorithm for learning model parameters, which satisfy monotonicity conditions, based on gradient descent optimization. We test the proposed method on two data sets. One set is synthetic and the other is formed by real data collected for computerized adaptive testing. We compare obtained results with the isotonic regression EM method by Masegosa et al. which also learns BN model parameters satisfying monotonicity. A comparison is performed also with the standard unrestricted EM algorithm for BN learning. Obtained experimental results in our experiments clearly justify monotonicity restrictions. As a consequence of monotonicity requirements, resulting models better fit data.

Keywords: computerized adaptive testing, monotonicity, isotonic regression EM, gradient method, parameters learning

1 Introduction

In our previous research Plajner and Vomlel (2015) we focused on Computerized Adaptive Testing (CAT) (Almond and Mislevy, 1999; van der Linden and

* This work was supported by the Czech Science Foundation (project No. 16-12010S) and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS17/198/OHK4/3T/14.

Glas, 2000). We used artificial student models to select questions during the course of testing. We have shown that it is useful to include monotonicity conditions while learning parameters of these models (Plajner and Vomlel, 2016b). Monotonicity conditions incorporate qualitative influences into a model. These influences restrict conditional probabilities in a specific way to avoid unwanted behavior. Some models we use for CAT include monotonicity naturally, but in this article we focus on a specific family of models, Bayesian Networks, which do not. Monotonicity in Bayesian Networks is discussed in literature for a long time. It is addressed, for example, by Wellman (1990); Druzdzel and Henrion (1993) and more recently by ,e.g., Restificar and Dietterich (2013); Masegosa et al. (2016). Monotonicity restrictions are often motivated by reasonable demands from model users. In our case of CAT it means we want to make sure that students having certain skills will have a higher probability of answering questions depending on these skills correctly. Moreover, assuming monotonicity we can learn better models, especially when the data sample is small. In our work we have so far used monotonicity attained by logistic regression models of CPTs. This has proven useful but it is restrictive since it requires a prescribed CPT structure.

In this article we extends our results in the domain of Bayesian Networks. We present a gradient descent optimum search method for learning parameters of CPTs respecting monotonicity conditions. First, we establish our notation and monotonicity conditions in Section 2. Our method is derived in Section 3. We have implemented the method and performed tests. For testing we used two different data sets. First, we used a synthetic data set generated from a monotonic model (CPTs satisfying monotonicity) and second, we used real data set collected earlier. Experiments were performed on these data sets also with the isotonic regression EM (irem) method described by Masegosa et al. (2016) and the ordinary EM learning without monotonicity restrictions. In Section 4 of this paper we take a closer look at the experimental setup and present results of described tests. The last section brings an overview and a discussion of the obtained results.

2 BN Models and Monotonicity

2.1 Notation

In this article we use Bayesian Networks. Details about BNs can be found in, for example, Pearl (1988); Nielsen and Jensen (2007). We restrict ourselves to the following BN structure. Networks have two levels. In compliance with our previous articles, variables in the parent's level are addressed as skill variables S. The children level contains questions-answers variables X. Example network structures, which we also used for experiments, are shown in Figure 1 and 2.

- We will use symbol $\boldsymbol{X}$ to denote the multivariable $(X_1, \ldots, X_n)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_n)$. The total number of question variables is n, the set of all indexes of question variables is $\boldsymbol{N} = \{1, \ldots, n\}$. Question variables are binary and they are observable.

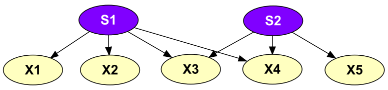


Fig. 1. Artificial Model

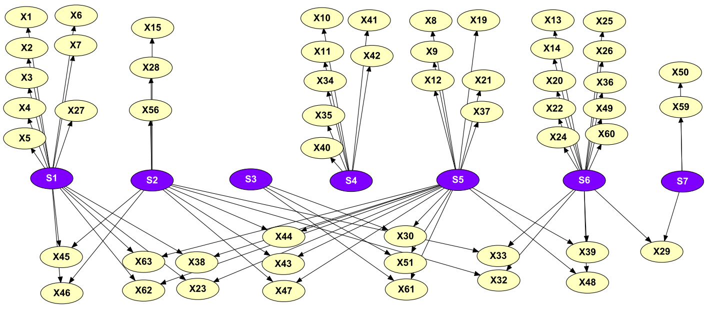


Fig. 2. CAT Model Network

- We will use symbol $\boldsymbol{S}$ to denote the multivariable $(S_1, \dots, S_m)$ taking states $\boldsymbol{s} = (s_1, \dots, s_m)$. The set of all indexes of skill variables is $\boldsymbol{M} = \{1, \dots, m\}$. Skill variables have variable number of states[3], the total number of states of a variable S_j is m_j and individual states are $s_{j,k}, k \in \{1, \dots, m_j\}$. The variable $\boldsymbol{S}^i = \boldsymbol{S}^{pa(i)}$ stands for a multivariable same as $\boldsymbol{S}$ but containing only parent variables of the question X_i. Indexes of these variables are $\boldsymbol{M}^i \subseteq \boldsymbol{M}$. The set of all possible state configurations of $\boldsymbol{S}^i$ is $Val(\boldsymbol{S}^i)$. Skill variables are all unobservable.

CPT parameters for a question variable X_i for all $i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i)$ are

$$\theta_{i,\boldsymbol{s}^i} = P(X_i = 0 | \boldsymbol{S}^i = \boldsymbol{s}^i), \;\; \boldsymbol{\theta}_i = (\theta_{i,\boldsymbol{s}^i})_{\boldsymbol{s}^i \in Val(\boldsymbol{S}^i)} \;\; .$$

We will also use $\theta_{i,\boldsymbol{s}} = \theta_{i,\boldsymbol{s}^i}$ with the whole parent set $\boldsymbol{S}$, where variables from $\boldsymbol{S} \setminus \boldsymbol{S}^i$ do not affect the value. Probabilities of a correct answer to a question X_i given state configuration $\boldsymbol{s}^i$ is $P(X = 1 | \boldsymbol{S}^i = \boldsymbol{s}^i) = 1 - \theta_{i,\boldsymbol{s}^i}$ (binary questions).

Parameters of parent variables for $j \in \boldsymbol{M}$ are

$$\rho_{j,s_j} = P(S_j = s_j), \;\; \boldsymbol{\rho}_j = \left(P(S_j = s_{j'})\right), j' \in \{1, \dots, m_j\} \;\; .$$

Parameter vector $\boldsymbol{\rho}_j$ is constrained by a condition $\sum_{s_j=1}^{m_j} \rho_{j,s_j} = 1$. To remove this condition we reparametrize this vector to

$$\rho_{j,s_j} = \frac{exp(\mu_{j,s_j})}{\sum_{s'_j=1}^{m_i} exp(\mu_{j,s'_j})} \;\; .$$

[3] In our experiments we use parents with 3 states, but the following theory applies to any number of states.

The whole vector of parameters is then

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n, \boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_m)\,, \text{ or } \boldsymbol{\mu} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m) \quad ,$$

where the meaning of $\boldsymbol{\mu_j}$ is the same as $\boldsymbol{\rho_j}$ but in this case vectors contain reparametrized variables. The transition from $\boldsymbol{\mu}$ to $\boldsymbol{\theta}$ is simply done with the reparametrization above and will be used without further notice. The total number of elements in the vector $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ is

$$l_{\boldsymbol{\mu}} = l_{\boldsymbol{\theta}} = \sum_{i \in \boldsymbol{N}} \prod_{j \in \boldsymbol{M^i}} m_j + \sum_{l \in \boldsymbol{M}} m_l \quad .$$

2.2 Monotonicity

The concept of monotonicity in BNs has been discussed in literature since the last decade of the previous millennium (Wellman, 1990; Druzdzel and Henrion, 1993). Later its benefits for BN parameter learning were addressed, for example, by van der Gaag et al. (2004); Altendorf et al. (2005). This topic is still active, e.g., Feelders and van der Gaag (2005); Restificar and Dietterich (2013); Masegosa et al. (2016).

We will consider only variables with states from $\mathbb{N}_0$ with their natural ordering, i.e., the ordering of states of skill variable's S_j for $j \in \boldsymbol{M}$, is

$$s_{j,1} \prec \ldots \prec s_{j,m_j} \quad .$$

For questions we use natural ordering of its states ($0 \prec 1$).

A variable S_j has monotone, resp. antitone, effect on its child if for all $k, l \in \{1, \ldots, m_j\}$:

$$\begin{aligned} s_{j,k} \preceq s_{j,l} &\Rightarrow P(X_i = 1 | S_j = s_{j,k}, \boldsymbol{s}) \;\leq\; P(X_i = 1 | S_j = s_{j,l}, \boldsymbol{s}) \quad \text{, resp.} \\ s_{j,k} \preceq s_{j,l} &\Rightarrow P(X_i = 1 | S_j = s_{j,k}, \boldsymbol{s}) \;>\; P(X_i = 1 | S_j = s_{j,l}, \boldsymbol{s}) \quad . \end{aligned}$$

where $\boldsymbol{s}$ is the configuration of other remaining parents of question i without S_j. For each question $X_i, i \in \boldsymbol{M}$ we denote by $\boldsymbol{S}^{i,+}$ the set of parents with a monotone effect and by $\boldsymbol{S}^{i,-}$ the set of parents with an antitone effect.

Next, we create a partial ordering $\preceq_i$ on all state configurations of parents $\boldsymbol{S}^i$ of the i-th question, where for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i})$:

$$\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i \Leftrightarrow \left(s^i_j \preceq r^i_j,\ j \in \boldsymbol{S}^{i,+}\right) \text{ and } \left(r^i_j \preceq s^i_j,\ j \in \boldsymbol{S}^{i,-}\right) \quad .$$

The monotonicity condition then requires that the question probability of correct answer is higher for a higher order parent configuration, i.e., for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i})$:

$$\begin{aligned} \boldsymbol{s}^i \preceq_i \boldsymbol{r}^i &\Rightarrow P(X_i = 1 | \boldsymbol{S}^i = \boldsymbol{s}^i) \;\leq\; P(X_i = 1 | \boldsymbol{S}^i = \boldsymbol{r}^i) \quad , \\ \boldsymbol{s}^i \preceq_i \boldsymbol{r}^i &\Rightarrow P(Xi = 0 | \boldsymbol{S}^i = \boldsymbol{s}^i) \;\geq P(Xi = 0 | \boldsymbol{S}^i = \boldsymbol{r}^i) \;\Leftrightarrow\; \theta_{i,\boldsymbol{s}^i} \;\geq\; \theta_{i,\boldsymbol{r}^i} \quad . \end{aligned}$$

In our experimental part we consider only isotone effect of parents on their children. The difference with antitone effects is only in the partial ordering.

3 Parameter Gradient Search with Monotonicity

To learn parameter vector $\boldsymbol{\mu}$ we develop a method based on the gradient descent optimization. We follow the work of Altendorf et al. (2005) where they use a gradient descent method with exterior penalties to learn parameters. The main difference is that we consider models with hidden variables.

We denote by $\boldsymbol{D}$ the set of indexes of observations vectors. One vector $x^k, k \in \boldsymbol{D}$ corresponds to one student and an observation of i-th variable X_i is x_i^k. The number of occurrences of the k-th configuration vector in the data sample is d_k.

We use the model structure as described in Section 2, i.e., unobserved parent variables and observed binary children variables. With sets $\boldsymbol{I}_0^k$ and $\boldsymbol{I}_1^k$ of indexes of incorrectly and correctly answered questions, we create following products based on observations in the k-th vector:

$$p_0^k(\boldsymbol{\mu}, \boldsymbol{s}, k) = \prod_{i \in \boldsymbol{I}_0^k} \theta_{i,\boldsymbol{s}}, \quad p_1^k(\boldsymbol{\mu}, \boldsymbol{s}, k) = \prod_{i \in \boldsymbol{I}_1^k} (1 - \theta_{i,\boldsymbol{s}}), \quad p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) = \prod_{j=1}^{m} exp(\mu_{j,s_j}).$$

We work with the log likelihood:

$$\begin{aligned} LL(\boldsymbol{\mu}) &= \sum_{k \in \boldsymbol{D}} d_k \cdot log \left(\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} \prod_{j=1}^{m} \frac{exp(\mu_{j,s_j})}{\sum_{s'_j=1}^{m_j} exp(\mu_{j,s'_j})} \cdot p_0^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \cdot p_1^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \right) \\ &= \sum_{k \in \boldsymbol{D}} d_k \cdot log \Big(\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \cdot p_0^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \cdot p_1^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \Big) - \\ &\quad - N \cdot \sum_{j=1}^{m} log \sum_{s'_j=1}^{m_j} exp(\mu_{j,s'_j}) \ . \end{aligned}$$

The partial derivatives of $LL(\mu)$ with respect to $\theta_{i,\boldsymbol{s}^i}$ for $i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i)$ are

$$\frac{\delta LL(\boldsymbol{\mu})}{\delta \theta_{i,\boldsymbol{s}^i}} = \sum_{k \in \boldsymbol{D}} d_k \cdot \frac{(-2x_i^k + 1) \cdot p_\mu(\boldsymbol{\mu}, \boldsymbol{s}^i) \cdot p_0^k(\boldsymbol{\mu}, \boldsymbol{s}^i, k) \cdot p_1^k(\boldsymbol{\mu}, \boldsymbol{s}^i, k)}{\theta_{i,\boldsymbol{s}^i} \cdot \sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \cdot p_0^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \cdot p_1^k(\boldsymbol{\mu}, \boldsymbol{s}, k)} \ .$$

and with respect to $\mu_{i,l}$ for $i \in \boldsymbol{M}, l \in \{1, \ldots, m_i\}$ are

$$\begin{aligned} \frac{\delta LL(\boldsymbol{\mu})}{\delta \mu_{i,l}} &= \sum_{k \in \boldsymbol{D}} d_k \cdot \frac{\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})}^{s_i = l} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \cdot p_0^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \cdot p_1^k(\boldsymbol{\mu}, \boldsymbol{s}, k)}{\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \cdot p_0^k(\boldsymbol{\mu}, \boldsymbol{s}, k) \cdot p_1^k(\boldsymbol{\mu}, \boldsymbol{s}, k)} - \\ &\quad -N \cdot \frac{exp(\mu_{i,l})}{\sum_{l'=1}^{m_i} exp(\mu_{k,l'})} \ . \end{aligned}$$

3.1 Monotonicity Restriction

To ensure monotonicity we use a penalty function

$$p(\theta_{i,\boldsymbol{s}^i}, \theta_{i,\boldsymbol{r}^i}) = exp(c \cdot (\theta_{i,\boldsymbol{r}^i} - \theta_{i,\boldsymbol{s}^i}))$$

for the log likelihood:

$$LL'(\boldsymbol{\mu}, c) = LL(\boldsymbol{\mu}) - \sum_{i \in \boldsymbol{N}} \sum_{\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i} p(\theta_{i,\boldsymbol{s}^i}, \theta_{i,\boldsymbol{r}^i}),$$

where c is a constant determining the strength of the condition. Theoretically, this condition does not ensure monotonicity but, practically, selecting high values of c results in monotonic estimates. If the monotonicity is not violated, i.e. $\theta_{i,\boldsymbol{r}^i} < \theta_{i,\boldsymbol{s}^i}$ then the penalty value is close to zero. Otherwise, the penalty is raising exponentially fast with respect to $\theta_{i,\boldsymbol{r}^i} - \theta_{i,\boldsymbol{s}^i}$. In our experiments we have used the value of $c = 40$ but any value higher than 20 provided almost identical results.

Partial derivatives with respect to $\mu_{i,l}$ remain unchanged. Partial derivatives with respect to $\theta_{i,\boldsymbol{s}^{\boldsymbol{i}}}$ are:

$$\frac{\delta LL'(\boldsymbol{\mu}, c)}{\delta \theta_{i,\boldsymbol{s}^{\boldsymbol{i}}}} = \frac{\delta LL(\boldsymbol{\mu})}{\delta \theta_{i,\boldsymbol{s}^{\boldsymbol{i}}}} + c \sum_{\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i} p(\theta_{i,\boldsymbol{s}^i}, \theta_{i,\boldsymbol{r}^i}) - c \sum_{\boldsymbol{r}^i \preceq_i \boldsymbol{s}^i} p(\theta_{i,\boldsymbol{r}^i}, \theta_{i,\boldsymbol{s}^i})$$

Using the penalized log likelihood, $LL'(\boldsymbol{\mu}, c)$, and its gradient

$$\nabla(LL(\boldsymbol{\mu}, c)) = \Big(\frac{\delta LL'(\boldsymbol{\mu}, c)}{\delta \theta_{i,\boldsymbol{s}^{\boldsymbol{i}}}}, \frac{\delta LL(\boldsymbol{\mu})}{\delta \mu_{j,l}} \Big) \ ,$$

for $i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i),\ j \in \boldsymbol{M}, l \in \{1, \ldots, m_j\}$, we can apply the standard gradient method optimization to solve the problem. In order to ensure probability values of $\boldsymbol{\theta}_i, i \in \boldsymbol{N}$ it is necessary to use a bounded optimization method.

4 Experiments

For testing we use two different Bayesian Network models. The first one is an artificial model and we use simulated data. The second model is one of the models we used for computerized adaptive testing and we work with real data (for details please refer to Plajner and Vomlel (2016a)). In both cases we learn model parameters from data. Parameters are learned with our gradient method, isotonic regression EM[4] and the standard unrestricted EM algorithm. The learned model quality is measured by the log likelihood of the whole data sample including the training subset. This is done in order to provide results comparable between different training set sizes.

[4] We have implemented the irem algorithm based on the article (Masegosa et al., 2016). We extended the method to work with parents with more states than 2 (the article considers only binary variables). Questions (children) remain binary which makes the extension easy.

4.1 Artificial Model

The first model is displayed in Figure 1. This model was created to provide simulated data for testing. The structure of the model is similar to models we use in CAT modeling with two levels of variables. Parents S_1 and S_2 have 3 possible states and children $X_1, \ldots, X_5$ are binary. We have instantiated the model with random parameters vector $\boldsymbol{\theta}^*$ satisfying monotonicity conditions. We drew a random sample of 100 000 cases from the model.

For parameters learning we use random subsets of size k of 10, 20, 50, 100, 1 000, 10 000, 50 000, and 100 000-(full data set) cases. For each size (except the last one) we use 10 different sets. Next, we prepared 15 initial parameter configurations for the fixed Bayesian Network structure (Fig. 1). These networks have starting parameters $\boldsymbol{\theta}_i$ generated at random, but in such a way, that they satisfy monotonicity conditions. The assumption of monotonicity is part of our domain expert knowledge. Therefore we can use it to speed up the process and avoid local optima. Parameters of parent variables are uniform and initial vectors are the same for each method. In our experiment we learn network parameters for each initial parameter setup for each set in a particular set size (giving a total of 150 learned networks for one set size). The learned parameter vectors are $\boldsymbol{\theta}_{i,j}$ for j-th subset of data.

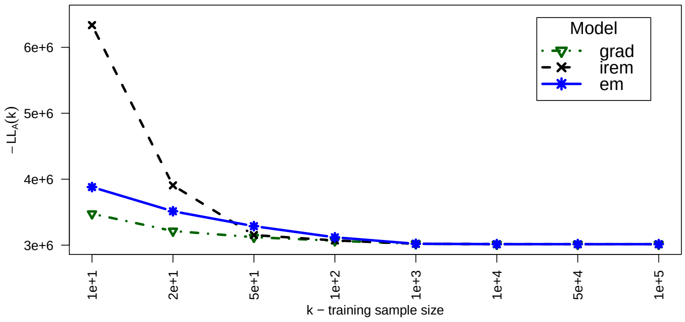


Fig. 3. Negative log likelihood for the whole sample and different training set sizes for the artificial model.

The average log likelihood for the whole data sample

$$LL_A = \frac{\sum_{j=1}^{10} \sum_{i=1}^{15} LL(\boldsymbol{\theta}_{i,j})}{150}$$

is shown in Figure 3 for each set size. In case of this model we are also able to measure the distance of learned parameters from the generating parameters in addition to the log likelihood. First we calculate an average error for each learned model:

$$e_{i,j} = \frac{|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{i,j}|}{l_{\boldsymbol{\theta}}} \quad ,$$

where || is L1 norm. Next we average over all results in one set size:

$$e = \frac{\sum_{j=1}^{10} \sum_{i=1}^{15} e_{i,j}}{150} \quad .$$

Resulting values of e are displayed in Figure 4 for each set size.

4.2 CAT Model

The second model is the model we used for CAT (Plajner and Vomlel, 2016b). Its structure is displayed in Figure 2. Parent variables $S_1, \ldots, S_7$ have 3 states and each one of them represents a particular student skill. Children nodes X_i are variables representing questions which are binary. Data associated with this model were collected from paper tests of mathematical skills of high school students. In total the data sample has 281 cases. For more detailed overview of tests refer to Plajner and Vomlel (2016a). For learning we use random subsets of size of 1/10, 2/10, 3/10, and 4/10 of the whole sample. Similarly to the previous model, we drew 10 random sets for each size and initiated models by 15 different initial random monotonic starting parameters $\boldsymbol{\theta}_i$.

After learning we compute log likelihoods of the whole data set and we create averages for each set size $LL_A(k)$ as with the previous model. Resulting values are in Figure 5. In this case we cannot compare learned parameters because the real parameters with real are unknown.

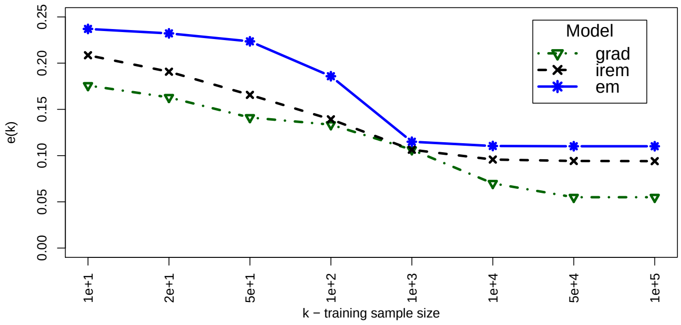


Fig. 4. Mean difference of parameters of learned and generating networks for different set sizes for the artificial model.

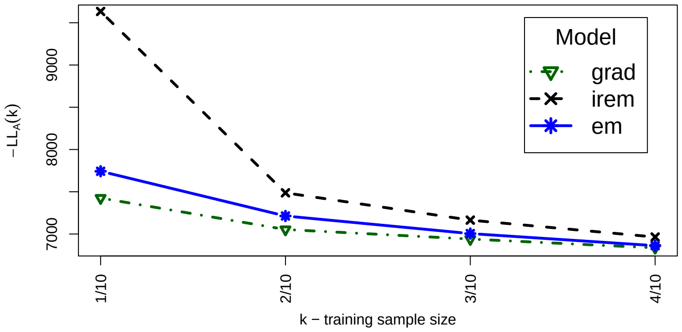


Fig. 5. Negative log likelihood for the whole sample and different training set sizes for the CAT model.

5 Conclusions

In this article we have presented a gradient based method for learning parameters of Bayesian Network under monotonicity restrictions. The method was described and then tested on two data sets. In Figures 3 and 5 it is clearly visible that this method achieves the best results from three tested methods (especially for small training samples). The irem method has problems with small training samples and the log likehood in those cases is low. This is a consequence of the fact that it moves to monotonic solution from a poor EM estimate and in these cases ensuring monotonicity implies log likelihood degradation. We can also observe that for the training sets larger than 1000 data vectors the EM algorithm stabilizes in its parameter estimations. It means that at about $k = 1000$ the EM algorithm found the best model it can and increasing training size does not improve the result. Nevertheless, as we can observe in Figure 4 parameters of learned networks are always closer to the generating parameters while considering monotonicity for both the irem and the gradient methods than for the standard EM.

These results verify usefulness of monotonicity for learning Bayesian Networks. A possible extension is to enlarge the theory of gradient based method to work with more general network structures.

Bibliography

Almond, R. G. and Mislevy, R. J. (1999). Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237.

Altendorf, E. E., Restificar, A. C., and Dietterich, T. G. (2005). Learning from Sparse Data by Exploiting Monotonicity Constraints. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005).*

Druzdzel, J. and Henrion, M. (1993). Efficient Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 548–553. AAAI Press.

Feelders, A. J. and van der Gaag, L. (2005). Learning Bayesian Network Parameters with Prior Knowledge about Context-Specific Qualitative Influences. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005).*

Masegosa, A. R., Feelders, A. J., and van der Gaag, L. (2016). Learning from incomplete data in Bayesian networks with qualitative influences. *International Journal of Approximate Reasoning*, 69:18–34.

Nielsen, T. D. and Jensen, F. V. (2007). *Bayesian Networks and Decision Graphs (Information Science and Statistics).* Springer.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc.

Plajner, M. and Vomlel, J. (2015). Bayesian Network Models for Adaptive Testing. In *Proceedings of the Twelfth UAI Bayesian Modeling Applications Workshop*, pages 24–33, Amsterdam, The Netherlands. CEUR-WS.org.

Plajner, M. and Vomlel, J. (2016a). Probabilistic Models for Computerized Adaptive Testing: Experiments. Technical report, ArXiv:1601.07929.

Plajner, M. and Vomlel, J. (2016b). Student Skill Models in Adaptive Testing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 403–414. JMLR.org.

Restificar, A. C. and Dietterich, T. G. (2013). Exploiting monotonicity via logistic regression in Bayesian network learning. Technical report, Corvallis, OR : Oregon State University.

van der Gaag, L., Bodlaender, H. L., and Feelders, A. J. (2004). Monotonicity in Bayesian networks. *20th Conference on Uncertainty in Artificial Intelligence (UAI2004)*, pages 569–576.

van der Linden, W. J. and Glas, C. A. W. (2000). *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers.

Wellman, M. P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303.

Question Selection Methods for Adaptive Testing with Bayesian Networks

Martin PLAJNER[1] and Amal MAGAUINA[1] and Jiří VOMLEL[2]

[1] *Faculty of Nuclear Sciences and Physical Engineering,*
Czech Technical University, Prague
Trojanova 13, Prague, 120 00, Czech Republic
martin.plajner@fjfi.cvut.cz

[2] *Institute of Information Theory and Automation,*
Czech Academy of Sciences,
Pod vodárenskou věží 4, Prague 8, 182 08, Czech Republic

Abstract

The performance of Computerized Adaptive Testing systems, which are used for testing of human knowledge, relies heavily on methods selecting correct questions for tested students. In this article we propose three different methods selecting questions with Bayesian networks as students' models. We present the motivation to use these methods and their mathematical description. Two empirical datasets, paper tests of specific topics in mathematics and Czech language for foreigners, were collected for the purpose of methods' testing. All three methods were tested using simulated testing procedure and results are compared for individual methods. The comparison is done also with the sequential selection of questions to provide a relation to the classical way of testing. The proposed methods are behaving much better than the sequential selection which verifies the need to use a better selection method. Individually, our methods behave differently, i.e., select different questions but the success rate of model's predictions is very similar for all of them. This motivates further research in this topic to find an ordering between methods and to find the best method which would provide the best possible selections in computerized adaptive tests.

Keywords: Computerized Adaptive Testing, Question Selection, Bayesian Networks.

1 Introduction

In our research we focus on Computerized Adaptive Testing (CAT). In CAT there is not a single static version of a test distributed to many students but an individual test is dynamically created during the course of testing for each individual participant. The next question is selected with regard to student's previous answers. This leads to several benefits as a better student assessment, a better motivation, etc. [4, 8]

We employ Bayesian networks for our research in this domain. The most recent papers we have published consider the beneficial effect of monotonicity conditions while learning model parameters. In this paper we aim at the testing process itself while the network is already learned. We take a closer look at the question selection procedure. There are many options how to select the next question from a bank of possible questions. This selection process is crucial for a successful adaptive testing procedure because the order in which questions are selected affects the rate in which the model improve its estimations. There is so far no definite answer which objective function produces the best possible results. In this article we discuss several question selection functions and compare them on two real models.

The paper is organized as follows. First, we describe the concept of Computerized Adaptive Testing, our models, and the notation we use. A short overview of two empirical data sets is presented. Both sets contain results of a paper (written) test collected for the purpose of our research. The first dataset is formed by results of high school tests of mathematical skills in the domain of functions; the second dataset has been collected from test results of foreign students of Czech language. In Section 4, we propose three different types of methods and to compare we also use linear selection process (questions are asked in the same order as they are ordered in the set of possible questions). All methods are tested on two available data sets. Results of experiments are presented in Section 5 of this paper where methods are compared and contrasted. The concluding section summarizes our results and points out possibilities for further improvements in this area.

2 Computerized Adaptive Testing

CAT is a concept of testing which is getting a large scientific attention for about two decades [9, 10, 12]. With CAT we build computer administered and computer controlled tests. The computer system is selecting questions for a student taking the test and evaluating his/her performance.

The process can be divided into two phases: model creation and testing. In the first phase the student model is created while in the second phase the model is used to actually test examinees. There are many different model types usable for adaptive testing as can be found, for example, in [1, 2, 3]. In this work we are working with Bayesian Networks. Regardless of the model the testing part follows the same scheme. With a prepared and calibrated model, CAT repeats following steps:

- The next question to be asked is selected.
- This question is asked and an answer is obtained.
- This answer is inserted into the model.
- The model (which provides estimates of the student's skills) is updated.
- Answers to all questions are estimated given the current estimates of student's skills. (optional)

This procedure is repeated until a termination criterion is reached. Criteria can be of various types, for example, a time restriction, a number of questions, or a confidence interval of the estimated variables (i.e., reliability of the test).

In this article we consider the first step of the testing procedure which is the question selection procedure.

3 Bayesian Network Models

We use Bayesian Networks (BNs) to model students. Details about BNs can be found in, for example, [6, 5]. We restrict ourselves to the following BN structure. Networks have two levels, variables in the parent's level are addressed as skill variables $S \in \mathcal{S}$ where $\mathcal{S}$ is the set of all skills. The children level contains question variables $X \in \mathcal{X}$ where $\mathcal{X}$ is the set of all questions.

- We will use symbol $\boldsymbol{X}$ to denote the multivariable $(X_1, \ldots, X_n)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_n)$. The total number of question variables is n, the set of all indexes of question variables is $\boldsymbol{N} = \{1, \ldots, n\}$. Question variables are binary and they are observable.
- We will use symbol $\boldsymbol{S}$ to denote the multivariable $(S_1, \ldots, S_m)$ taking states $\boldsymbol{s} = (s_1, \ldots, s_m)$. The set of all indexes of skill variables is $\boldsymbol{M} = \{1, \ldots, m\}$. In this article we use only binary skill variables. The set of all possible state configurations of $\boldsymbol{S}$ is $Val(\boldsymbol{S})$. Skill variables are all unobservable.

3.1 Data and specific models

To test our theoretical methods we have collected empirical data. We obtained two different data sets which are described here.

First, we designed a paper test of mathematical knowledge of grammar school students. The test focuses on simple functions (mostly polynomial, trigonometric, and exponential/logarithmic).

Students were asked to solve various mathematical problems (reffered as questions) including graph drawing and reading, calculating points on the graph, root finding, describing function shapes and other function properties. Questions are open (the mathematical problem's solution has to be included) and results are stored as binary correct/wrong values. In total 281 participants took the test. For the purpose of this paper this data set is modeled by two different Bayesian network models. One of them is shown in Figure 1. It consists of 53 questions and 8 skill nodes. These skill nodes represent different student skills connected to questions. This models is further referred to as Mathematical knowledge test Model (MM).

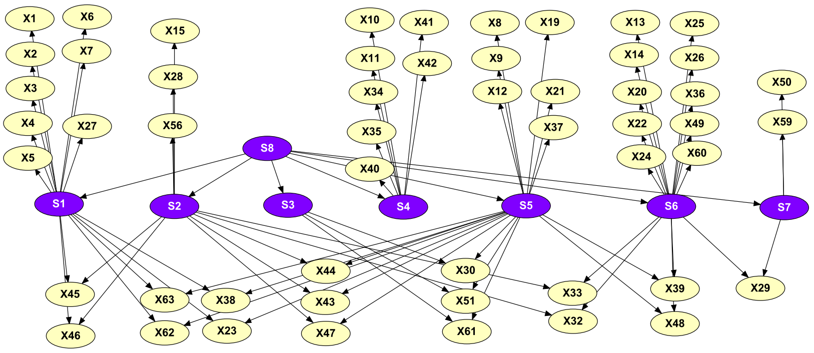


Figure 1: CAT MM structure

The second dataset was collected with a test of Czech language for non-native speaker students. This test contained multiple choice questions with four possible answers. One answer was correct. The test was assessed in a binary way where each question was either correct or incorrect. This test contains 30 questions and 143 students participated in the testing process. The model which was created by a domain expert is shown in Figure 2. Apart from 30 question nodes it has 11 skill nodes. Each skill, again, represents a specific ability a student should have to answer a connected question correctly. The skills include abilities related to morphology, vocabulary, conjugation, etc. This model is referred to as Czech language test Model (CM)[1].

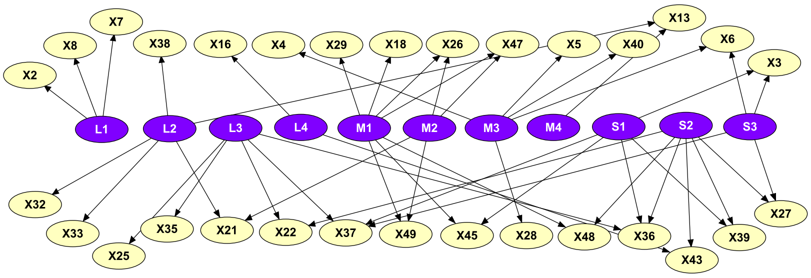


Figure 2: CAT CM structure

4 Question Selection Methods

The task of the question selection is repeated in every step of testing of an individual student. Its process is described in detail below.

[1] More detailed information can be found (in Czech) in the master thesis of Amal Magauina available at https://dspace.cvut.cz/

We define the question evidence e as:

$$e = \{X_{i_1} = x_{i_1}, \ldots, X_{i_n} = x_{i_n} | i_1, \ldots, i_n \in \boldsymbol{N}\}.$$

where $\{i_1, \ldots, i_n\} = \boldsymbol{I}$ are indexes of already answered questions. Remaining questions are unobserved (unanswered) $\hat{\mathcal{X}} = \{X_i | i \in \boldsymbol{N} \setminus \boldsymbol{I}\}$.

The goal is to select a question from $\hat{\mathcal{X}}$ to be asked next. The selection is dependent on a criterion function which may take different forms. Below, we describe three possible question selection methods. In this paper we also use, as a comparison method, a sequential selection. While using the sequential selection, the question we select is simply chosen in the same order as they are ordered in the question input list. This type of question selection is often used in non-adaptive tests where questions are always asked in the same sequence.

Three methods, we present further, are:

- Maximization of the Expected Entropy Reduction (also called Information Gain)
- Maximization of the Expected Skills Variance
- Maximization of the Expected Question Variance

The motivation for selecting these three possibilities is discussed for each criterion separately.

4.1 Maximization of the Expected Entropy Reduction

The purpose of an adaptive test is to provide the best possible information about a tested student. Each student is modeled by his skills. The criterion described in this section uses the Shannon entropy calculated over all skill values which we define in this section. It is a measure of the certainty of skills estimation. Because of that we want to select a question which provides the largest expected information gain if asked, i.e., a question which reduces uncertainty the most. This method is further referred to as Skills' Entropy.

We compute the cumulative Shannon entropy over all skill variables of S given the evidence e:

$$H(e) = \sum_{j \in \boldsymbol{M}} \sum_{s=0}^{1} -P(S_j = s|e) \cdot \log P(S_j = s|e) \ . \tag{1}$$

Assume we decide to ask a question $\hat{X} \in \hat{\mathcal{X}}$. After inserting the observed outcome the entropy over all skills changes. We can compute the value of new entropy for evidence extended by $\hat{X} = \hat{x}$ as:

$$H(e, \hat{X} = \hat{x}) = \sum_{j \in \boldsymbol{M}} \sum_{s=0}^{1} -P(S_j = s|e, \hat{X} = \hat{x}) \cdot \log P(S_j = s|e, \hat{X} = \hat{x}) \ . \tag{2}$$

This entropy $H(e, \hat{X} = \hat{x})$ is the sum of individual entropies over all skill nodes. Another option would be to compute the entropy of the joint probability distribution of all skill nodes. This would take into account correlations between these nodes. In our task we want to estimate marginal probabilities of all skill nodes. In the case of high correlations between two (or more) skills the latter criterion would assign them a lower significance in the model. This is the behavior we wanted to avoid. The first criterion assigns the same significance to all skill nodes which is a better solution. Moreover, the computational time required for the proposed method is lower.

Now, we can compute the expected entropy after answering question $\hat{X}$:

$$EH(\hat{X}, e) \quad = \quad \sum_{\hat{x}=0}^{1} P(\hat{X} = \hat{x}|e) \cdot H(e, \hat{X} = \hat{x}) \ . \tag{3}$$

Finally, we choose a question X^* that maximizes the information gain $IG(\hat{X}, e)$

$$X^* \quad = \quad \arg\max_{\hat{X} \in \hat{\mathcal{X}}} IG(\hat{X}, e) \ , \text{ where} \tag{4}$$

$$IG(\hat{X}, e) \quad = \quad H(e) - EH(\hat{X}, e) \ . \tag{5}$$

4.2 Maximization of the Expected Skills Variance

With this criterion we want to select a question which leads to the largest variance of state probabilities of skill variables. The rationale behind this selection is very similar to the one discussed in the previous method. The goal is to provide the most accurate estimation of student's skills and also to provide the best separation of students based on their skills. We measure the variance between skill's state probabilities (student having the skill). The variance is measured for two possible answers to one question, i.e., correct and incorrect. The criterion searches for a question which provides the largest variance in these two possibilities. This method is further referred to as Skills' Variance.

We consider unanswered question $\hat{X} \in \hat{\mathcal{X}}$ to be asked. First, we establish following notation:

$$p_0^j = P(S_j = 1|\hat{X} = 0, e) \ ,$$
$$p_1^j = P(S_j = 1|\hat{X} = 1, e) \ ,$$

where $S_j \in \mathcal{S}$. The symbol p_0^j stands for the probability of a student having the examined skill S_j even though the answer to the question $\hat{X}$ was incorrect. p_1^j is the case where the answer was correct and the student has the skill S_j. Naturally, the value of p_1^j should be larger than p_0^j. We compute the average value $\overline{p}^j$:

$$\overline{p}^j = P(\hat{X} = 0|e) \cdot p_0^j + P(\hat{X} = 1|e) \cdot p_1^j \ .$$

Then, the expected variance of states' probabilities of the skill S_j after answering the question $\hat{X}$ can be obtained using the following formula:

$$var_j(S_j|e, \hat{X}) = (\overline{p}^j - p_0^j)^2 \cdot P(\hat{X} = 0|e) + (\overline{p}^j - p_1^j)^2 \cdot P(\hat{X} = 1|e) \ . \tag{6}$$

This value has to be computed for each skill in the model. Afterwards, we compute the average of these values for the question $\hat{X}$:

$$var(\mathcal{S}|e, \hat{X}) = \frac{1}{m} \sum_{j \in \boldsymbol{M}} var_j(S_j|e, \hat{X}) \ . \tag{7}$$

We select a question which has the highest average value computed from (7):

$$X^* \quad = \quad \arg\max_{\hat{X} \in \hat{\mathcal{X}}} var(\mathcal{S}|e, \hat{X}) \ . \tag{8}$$

Maximization of (6) can be viewed as a generalization of the criterion of student separation described in our previous article [7]. The difference is that in this case we consider the probability of $S_j = 1$ after answering $\hat{X}$ instead of the most probable state of S_j.

4.3 Maximization of the Expected Question Variance

Previous two criteria aimed at skills directly. This third one aims at questions instead. From all unanswered questions we want to find a question with the highest expected variance of correct answer probabilities for all possible state combinations. This criterion is motivated as follows: if the question's correct answer probability varies a lot with changing skill states it means that this question is significantly affected when student skills shifts. It follows from the Bayes rule that this question also has a significant influence on the skills. This method is further referred to as Questions' Variance.

The expected variance of the question's $\hat{X}$ correct answer probability is computed given the following formula:

$$var(\hat{X}|e) = \sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} (P(\hat{X} = 1|e) - P(\hat{X} = 1|\boldsymbol{s}, e))^2 \cdot P(\boldsymbol{s}|e) \ . \tag{9}$$

A question with the highest value of expected variance given by (9) is selected to be asked next. The use of this function for computations during testing is impractical because of its computational complexity as it would take long time to select the next question. We propose an approximation of Formula (9). We compute the variance for a single skill node and then take into account their combined average instead of the full computation over all states' combinations.

We establish following notation:

$$r_0^j = P(\hat{X} = 1|S_j = 0) \ ,$$
$$r_1^j = P(\hat{X} = 1|S_j = 1) \ ,$$

where $S_j \in \mathcal{S}$, $\hat{X} \in \hat{\mathcal{X}}$. r_0^j stands for the probability, that the student answers correctly to the question even though he/she has no skill in question. r_1^j is the same situation while the student has all examined skills. Intuitively, the value r_1^j has to be larger than r_0^j.

With the average value

$$\overline{r}^j = P(S_j = 0|e) \cdot r_0^j + P(S_j = 1|e) \cdot r_1^j$$

we can compute the expected variance of correct answer probability for the question $\hat{X}$ using the next formula:

$$\begin{aligned} var_j(\hat{X}|e) &= (\overline{r}^j - r_0^j)^2 \cdot P(S_j = 0|e) + (\overline{r}^j - r_1^j)^2 \cdot P(S_j = 1|e) \ , \\ var(\hat{X}|e) &= \frac{1}{m} \sum_{j \in \boldsymbol{M}} var_j(\hat{X}|e) \ . \end{aligned} \tag{10}$$

A question X^* we select is maximizing this variance.

$$X^* \quad = \quad \arg\max_{\hat{X} \in \hat{\mathcal{X}}} var(\mathcal{S}|e, \hat{X}) \ . \tag{11}$$

The value $(\overline{r}^j - r_s^i)$ can be viewed as differential of $P(\hat{X} = 1|S_j = s)$ of skill variables S_j that have only two states $s \in \{0, 1\}$. Therefore, if P is the probability density function of the continuous skill variable S_j, we can view it as a finite equivalent of the probability $P(\hat{X} = 1|S_j = s)$ derivative with respect to s. It means that $var(\hat{X}|e)$ is similar to Fisher's information which is a commonly used criterion for IRT (Item Response Theory) [11] – another possible type of model for CAT. More detailed explanation of this criterion in the case of continuous skill variables can be found in [7].

5 Experiments

5.1 Experimental Setup

To evaluate models we have done experiments on both data sets with models MM and CM described above. We have used 10 fold cross-validation method for both data sets. Models were first learned using standard EM algorithm from learning data. Next, we performed a simulation of CAT test for every model and for every student using testing data.

During simulated testing we first estimated the skills of a student based on his/her answers. At the start of each step we compute marginal probability distributions for all skills $\boldsymbol{S}$. This happens before selecting a new question and updating the model with the new answer. We use evidence e obtained in previous steps which is at the start of testing empty. Then, based on estimated skills we predict answers to all questions, where we select the most probable state of each question $X \in \mathcal{X}$:

$$x^* = \arg\max_x P(X = x|\boldsymbol{S}) \ . \tag{12}$$

By comparing this value to the real answer x' of the question X we obtain a success rate of the response estimates for all questions $X \in \mathcal{X}$ of a test result t (particular student's result) in one step

$$\mathrm{SR}^t \quad = \quad \frac{\sum_{X \in \mathcal{X}} I(x^* = x')}{|\mathcal{X}|} \ , \text{ where} \tag{13}$$

$$I(expr) \quad = \quad \begin{cases} 1 & \text{if } expr \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

The total success rate of one model in one step for all test data is defined as

$$\mathrm{SR} \;\; = \;\; \frac{\sum_{t=1}^{D} \mathrm{SR}^t}{D} \;, \qquad (15)$$

where D is the dataset size.

5.2 Experimental Results

Average results of simulated tests for both data sets, i.e., both models described above, are displayed in graphs 3 and 4 for each model separately. Graphs show success rates SR in the first 30 steps. Step 0 is the state before asking any questions. At this point the prediction is based only on data itself. There is no evidence and the selection criterion adds no benefit. Therefore the SR is the same over all cases for a single model. For comparison we include also the sequential selection as described in Section 4.

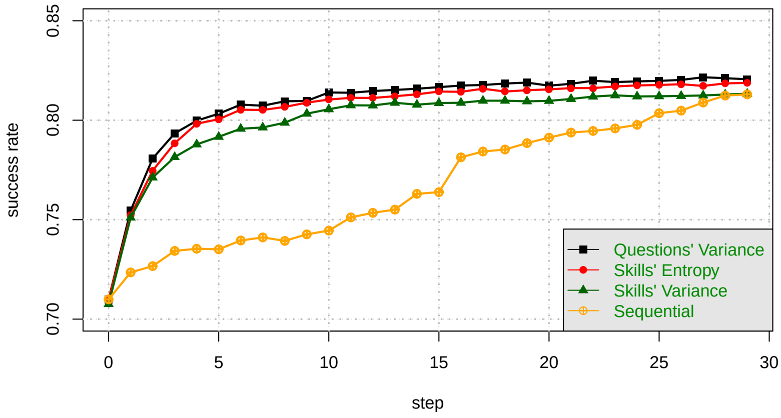


Figure 3: MM success rates for first 30 questions of simulated testing

As we can see in these graphs the worst performing method is the sequential selection. It is apparent that the rate in which this method improves its estimates is lower than the rate of the remaining methods. This is caused by the fact that it is selecting questions which are not most informative in the current test situation for the tested student.

The Skills' Variance method of the selection has the lowest performance (or same) from three proposed methods in both models. As explained below it is the only method which has statistically significantly worse results in one instance. Particular reasons for this behavior has to be explored further as there might be many possible causes.

Questions selected by individual methods are not the same even though the success rate of question estimates is very similar. The selected questions are displayed in the Tables 3 and 4. Numbers displayed correspond to the total number of selections of the particular question in the MM and the CM in the first five steps of simulated testing. Questions which were not selected at all are not included in the tables. By inspecting these tables we can easily see that there are differences in individual methods. Some questions were not selected at all by one method while the other two methods selected them in some cases only. For example, in the MM the question X42 was not selected by Skills' Entropy while Questions' Variance selected it in 112 cases in the first five steps. Nevertheless, we can see a trend of good questions which are selected very often and soon in the process of testing by all methods. For example, in the MM the question X43 was

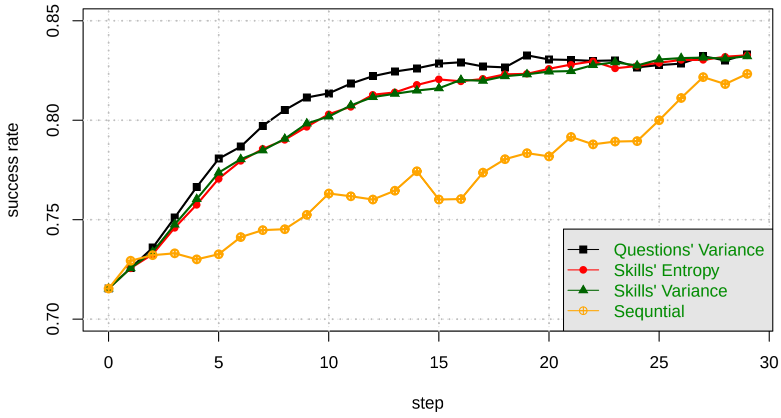


Figure 4: CM success rates for first 30 questions of simulated testing

selected for all students by Skills' Entropy and only for 7/10 of the dataset by the two remaining methods.

5.3 Wilcoxon tests

To confirm our conclusions described above we used the Wilcoxon signed-rank test. Tables 1 and 2 contain p-values obtained from Wilcoxon tests to compare the success rates of two criteria. An alternative hypothesis is that the overall success rate of the i-th criterion (row index) is greater than the overall success rate of the j-th criterion (column index).

Table 1: MM Wilcoxon tests p-values

	sequential	**Skills' Entropy**	**Skills' Variance**	**Questions' Variance**
sequential	-	1	1	1
Skills' Entropy	$1.17 \cdot 10^{-5}$	-	$1.62 \cdot 10^{-1}$	$9.39 \cdot 10^{-1}$
Skills' Variance	$2.20 \cdot 10^{-4}$	$8.40 \cdot 10^{-1}$	-	$9.99 \cdot 10^{-1}$
Questions' Variance	$2.04 \cdot 10^{-8}$	$6.15 \cdot 10^{-2}$	$9.22 \cdot 10^{-3}$	-

Table 2: CM Wilcoxon tests p-values

	sequential	**Skills' Entropy**	**Skills' Variance**	**Questions' Variance**
sequential	-	1	1	1
Skills' Entropy	$1.10 \cdot 10^{-4}$	-	$5.14 \cdot 10^{-1}$	$8.76 \cdot 10^{-1}$
Skills' Variance	$8.77 \cdot 10^{-5}$	$4.92 \cdot 10^{-1}$	-	$8.58 \cdot 10^{-1}$
Questions' Variance	$7.72 \cdot 10^{-6}$	$1.27 \cdot 10^{-1}$	$1.46 \cdot 10^{-1}$	-

As we can see in both cases, p-values of the sequential selection compared to all other criteria are much smaller than the borderline of $\alpha = 0.05$. This confirms the fact that the sequential selection provides the worst results. The table of the MM also shows that the success rate of Questions' Variance method is greater than the SR of Skills' Variance method (p-value $= 9.22 \cdot 10^{-3}$). This shows there is statistically important improvement in success rates of the former over the latter method. All other pairs of different selection criteria show statistically insignificant difference within the selected confidence interval. Therefore, for the remaining pairs we can not establish any statistically sound order.

6 Conclusions and Future Work

This article considered different ways of selecting questions during the procedure of Computerized Adaptive Testing. We presented three different types of methods to select questions during CAT which were afterwards tested. For testing we used two data sets collected for this purpose.

The first important empirical observation is that the question selection method has a significant impact on the quality of predictions during the CAT procedure. In the comparisons all three proposed methods clearly outperformed the sequential selection. The motivation to study these methods is thus valid.

The next observation is that three proposed methods behave differently. In this case the difference in the quality of prediction is not large, it is statistically insignificant, but the methods are distinguishable since they select different questions. The first step in the future research is to provide generalizations of these methods to support multi-state skill variables. It seems that especially for Skills' Variance it may be very beneficial to test a model with skill nodes having more than two states. It is necessary to show if we can improve these methods and establish any ordering between them which would be valid generally over different models.

Acknowledgement
This work was supported by the Czech Science Foundation (project No. 16-12010S) and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS17/198/OHK4/3T/14.

Table 3: The frequency of questions X1-X61 selections during simulated testing using criterion (a) Skills' Entropy, (b) Skills' Variance, (c) Questions' Variance for the MM model. Questions which were never selected are not included.

	X3	X5	X7	X10	X11	X12	X13	X19	X25	X26	X27	X29	X30	X32	X33	X38	X39	X41	X42	X43	X44	X50	X51	X61
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	281	-	-	-	-
2	-	-	-	-	-	-	-	118	-	-	-	-	-	149	-	-	-	14	-	-	-	-	-	-
3	-	62	10	5	-	-	-	70	-	8	-	-	-	-	-	-	-	38	-	-	-	-	68	20
4	-	9	-	11	-	1	8	-	-	23	7	25	18	-	28	-	-	113	-	-	-	16	2	20
5	-	24	-	13	13	14	-	9	3	1	9	14	49	-	19	-	-	33	-	-	5	38	10	27

	X3	X5	X7	X10	X11	X12	X13	X19	X25	X26	X27	X29	X30	X32	X33	X38	X39	X41	X42	X43	X44	X50	X51	X61
1	-	-	-	-	-	-	-	-	-	-	-	-	-	56	-	-	-	-	-	197	28	-	-	-
2	-	-	-	-	-	-	-	49	-	-	-	-	-	105	-	-	-	108	-	10	9	-	-	-
3	10	53	-	-	-	-	-	48	-	-	-	-	-	14	38	-	-	35	9	15	-	-	45	14
4	4	6	5	7	-	1	-	16	-	13	-	-	-	-	11	3	-	80	35	13	9	49	18	11
5	9	8	2	1	-	-	8	32	-	24	1	14	9	-	13	11	-	32	20	2	7	40	20	28

	X3	X5	X7	X10	X11	X12	X13	X19	X25	X26	X27	X29	X30	X32	X33	X38	X39	X41	X42	X43	X44	X50	X51	X61
1	-	-	-	-	-	-	-	-	-	-	-	-	-	84	-	-	-	-	-	197	-	-	-	-
2	-	-	-	-	-	-	-	113	-	-	-	-	-	15	-	-	-	106	-	-	47	-	-	-
3	-	16	3	-	-	-	-	22	-	-	-	-	-	21	4	-	-	61	66	25	-	-	49	14
4	6	59	7	2	-	1	-	-	-	-	-	-	2	5	17	16	-	18	36	12	-	47	39	14
5	12	14	-	10	-	1	-	12	7	17	1	2	15	25	10	16	5	18	10	-	-	70	10	26

Table 4: The frequency of questions X2-X49 selections during simulated testing using criterion (a) Skills' Entropy, (b) Skills' Variance, (c) Xuestions' Variance for the CM model. Xuestions which were never selected are not included.

	X2	X3	X4	X6	X7	X13	X16	X18	X21	X22	X26	X28	X35	X36	X37	X39	X43	X45	X47	X48	X49
1	-	-	-	-	-	-	-	-	-	-	-	-	143	-	-	-	-	-	-	-	-
2	-	14	-	-	-	-	86	-	-	43	-	-	-	-	-	-	-	-	-	-	-
3	19	4	-	13	-	4	25	-	-	15	-	-	-	-	-	-	3	-	-	60	-
4	12	15	8	14	8	12	9	-	-	12	6	-	-	-	1	6	5	-	-	11	24
5	26	22	1	13	2	13	10	-	5	7	10	-	-	4	-	20	1	2	-	7	-

	X2	X3	X4	X6	X7	X13	X16	X18	X21	X22	X26	X28	X35	X36	X37	X39	X43	X45	X47	X48	X49
1	-	-	-	-	-	-	-	-	-	-	-	-	143	-	-	-	-	-	-	-	-
2	-	14	-	-	-	-	77	-	-	52	-	-	-	-	-	-	-	-	-	-	-
3	14	4	-	18	-	-	29	-	-	6	-	-	-	-	-	-	3	-	-	69	-
4	4	15	7	19	-	8	21	-	-	8	6	-	-	-	-	4	6	-	1	17	27
5	15	33	3	31	-	5	9	-	2	14	9	2	-	-	-	-	2	2	1	11	4

	X2	X3	X4	X6	X7	X13	X16	X18	X21	X22	X26	X28	X35	X36	X37	X39	X43	X45	X47	X48	X49
1	-	-	-	-	-	-	-	-	-	-	-	-	143	-	-	-	-	-	-	-	-
2	23	18	-	-	-	-	29	-	-	58	-	-	-	-	-	-	-	-	-	15	-
3	16	25	-	15	-	-	13	-	-	-	6	-	-	-	-	-	3	-	-	65	-
4	20	29	-	18	3	-	13	-	-	33	6	-	-	1	-	4	1	2	-	13	-
5	10	29	2	11	6	-	24	3	-	9	5	-	-	6	-	20	8	1	-	3	6

References

[1] R. G. Almond and R. J. Mislevy. Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237, 1999.

[2] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer, 1999.

[3] M. J. Culbertson. *Graphical Models for Student Knowledge: Networks, Parameters, and Item Selection.* PhD thesis, University of Illinois at Urbana, 2014.

[4] K. C. Moe and M. F. Johnson. Participants' Reactions To Computerized Testing. *Journal of Educational Computing Research*, 4(1):79–86, jan 1988.

[5] T. D. Nielsen and F. V. Jensen. *Bayesian Networks and Decision Graphs (Information Science and Statistics).* Springer, 2007.

[6] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., dec 1988.

[7] M. Plajner and J. Vomlel. Student Skill Models in Adaptive Testing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 403–414. JMLR.org, 2016.

[8] S. Tonidandel, M. A. Quiñones, and A. A. Adams. Computer-adaptive testing: the impact of test characteristics on perceived performance and test takers' reactions. *The Journal of applied psychology*, 87(2):320–32, apr 2002.

[9] W. J. van der Linden and C. A. W. Glas. *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers, 2000.

[10] W. J. van der Linden and C. A. W. Glas, editors. *Elements of Adaptive Testing.* Springer New York, NY, 2010.

[11] W. J. van der Linden and R. K. Hambleton. *Handbook of Modern Item Response Theory.* Springer New York, 2013.

[12] H. Wainer and N. J. Dorans. *Computerized Adaptive Testing: A Primer.* Routledge, 2015.

Gradient Descent Parameter Learning of Bayesian Networks under Monotonicity Restrictions

Martin Plajner[1,2]
plajner@utia.cas.cz
Faculty of Nuclear Sciences and Physical Engineering[1],
Czech Technical University, Prague
Trojanova 13, Prague,
120 00, Czech Republic

Jiří Vomlel[2]
Institute of Information Theory and Automation[2],
Czech Academy of Sciences,
Pod vodárenskou věží 4, Prague 8, 182 08, Czech Republic

Abstract

Learning parameters of a probabilistic model is a necessary step in most machine learning modeling tasks. When the model is complex and data volume is small the learning process may fail to provide good results. In this paper we present a method to improve learning results for small data sets by using additional information about the modelled system. This additional information is represented by monotonicity conditions which are restrictions on parameters of the model. Monotonicity simplifies the learning process and also these conditions are often required by the user of the system to hold.

In this paper we present a generalization of the previously used algorithm for parameter learning of Bayesian Networks under monotonicity conditions. This generalization allows both parents and children in the network to have multiple states. The algorithm is described in detail as well as monotonicity conditions are.

The presented algorithm is tested on two different data sets. Models are trained on differently sized data subsamples with the proposed method and the general EM algorithm. Learned models are then compared by their ability to fit data. We present empirical results showing the benefit of monotonicity conditions. The difference is especially significant when working with small data samples. The proposed method outperforms the EM algorithm for small sets and provides comparable results for larger sets.

1 Introduction

In our research we address Computerized Adaptive Testing (CAT) [1, 13]. CAT is a concept of testing latent student abilities which allows us to create shorter tests, asking less questions in a shorter time while keeping the same level of information. This task is performed by asking the right questions for each individual student. Questions are selected based on a student model. In common practice experts often use Item Response Theory models [10] (IRT) which are well explored and have been in use for a long time. Nevertheless, we have focused our attention on a different family of models to model a student using Bayesian Networks (BNs) since they offer more options in the modelling process. It is for example possible to model more complex influences between skills and questions as BNs are not limited to connecting each skill with each question as well as we can introduce connections between skills themselves.

During our research we noticed that there are certain conditions which should be satisfied in this specific modelling task. We especially focused on monotonicity conditions. Monotonicity conditions incorporate qualitative influences into a model. These influences restrict conditional probabilities inside the model in a specific way to avoid unwanted behavior. Monotonicity in Bayesian Networks has been discussed in the literature for a long time. It is addressed, selecting the most relevant to our topic, by [14, 3] and more recently by ,e.g., [11, 5]. Monotonicity restrictions are often motivated by reasonable demands from model users. In our case of CAT it means we want to guarantee that students having certain skills will have a higher probability of answering questions correctly.

Certain types of models include monotonicity naturally by the way they are constructed. In the case of general BNs this is not true. In order to satisfy these conditions we have to introduce restrictions to conditional probabilities during the process of parameter learning.

In our previous work we first showed that monotonicity conditions are uself in the context of CAT [8]. Later we applied these conditions to Bayesian Network [9]. In this article we extend our earlier presented gradient descent optimum search method for BN parameter learning under monotonicity conditions. The last article covers only specific BNs. It works solely with binary children variables in the model (yes/no answers in terms of CAT). The extension we present in this article provides a tool to include monotonicity in BN models with multiple-state children nodes. Additionally, in this article we perform experiments on a new dataset. It is consists of data from the Czech high school state final exam. This data source contains a large volume of reliable data, and it is very useful for the empirical verification of our ideas.

We implemented the new method in R language and performed experimental verification of our assumptions. We used two data sets. The first one, a synthetic data set, is generated from artificial models satisfying monotonicity conditions. The second one, an empirical data set, is formed by data from the Czech high school final exam. Experiments were performed on these data sets also with the ordinary EM learning without monotonicity restrictions in order to compare these two approaches.

The structure of this article is as follows. First, we establish our notation and describe monotonicity conditions in detail in Section 2. Next, we present the extended method in Section 3. In Section 4 of this paper, we take a closer look at the experimental setup and present results of our experiments. The last section contains an overview and a discussion of the obtained results.

2 BN Models and Monotonicity

2.1 Notation

In this article we use the new gradient descent method for BNs which are used to model students in the domain of CAT. Details about BNs can be found, for example, in [7, 6]. We restrict ourselves to the BNs that have two levels. In compliance with our previous articles, variables in the parent level are addressed as skill variables S. The children level contains questions variables X. Examples of network structures, which we also used for experiments, are shown in Figures 1 and 2.

- We use the symbol $\boldsymbol{X}$ to denote the multivariable $(X_1, \ldots, X_n)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_n)$. The total number of question variables is n, the set of all indexes of question variables is $\boldsymbol{N} = \{1, \ldots, n\}$. Question variables' individual states are $x_{i,t}, t \in \{0, \ldots, n_i\}$ and they are observable. Each question can have a different number of states, the maximum number of states over all variables is $N^{max} = \max_i(n_i) + 1$. States are integers with natural ordering specifying the number of points obtained in the $i - th$ question[1].
- We use the symbol $\boldsymbol{S}$ to denote the multivariable $(S_1, \ldots, S_m)$ taking states $\boldsymbol{s} = (s_1, \ldots, s_m)$. The set of all indexes of skill variables is $\boldsymbol{M} = \{1, \ldots, m\}$. Skill variables have a variable number of states, the total number of states of a variable S_j is m_j, and individual states are $s_{j,k}, k \in \{1, \ldots, m_j\}$. The variable $\boldsymbol{S}^i = \boldsymbol{S}^{pa(i)}$ stands for a multivariable containing only parent variables of the question X_i. Indexes of these variables are $\boldsymbol{M}^i \subseteq \boldsymbol{M}$. The set of all possible state configurations of $\boldsymbol{S}^i$ is $Val(\boldsymbol{S}^i)$. Skill variables are unobservable.

The BN has CPT parameters for all questions $X_i, i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i)$ which define conditional probabilities as

$$P(X_i = t | \boldsymbol{S} = \boldsymbol{s}) = \theta^t_{i,\boldsymbol{s}^i} \quad ,$$

and for all parent variables $S_j, j \in \boldsymbol{M}$ as

$$P(S_j = s_j) = \tilde{\theta}_{j,s_j} \quad .$$

[1] The interpretation of points is very complex and has to be viewed as per question because we use the CAT framework. In this context getting one point in one question is not the same as one point in another.

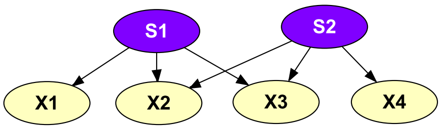


Figure 1: An artificial BN model

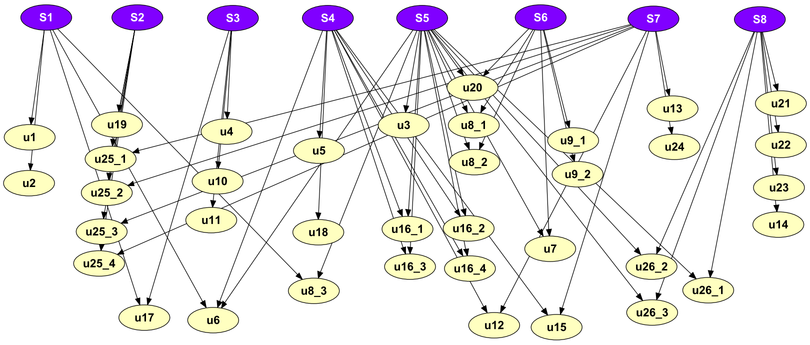


Figure 2: A BN model for CAT

From the definition above it follows that parameters are constrained to be between zero and one and to sum up to one. For question variable the condition is $\sum_{t=0}^{n_i} \theta^t_{i,\boldsymbol{s}^i} = 1, \ \forall i, \boldsymbol{s}^i$ and for parent variables it is $\sum_{s_j} \tilde{\theta}_{j,s_j} = 1, \ \forall j$. To remove this condition for the later use in the gradient method we reparametrize parameters

$$\begin{aligned} \theta^t_{i,\boldsymbol{s}^i} &= \frac{exp(\mu^t_{i,s_i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,s_i})} \\ \tilde{\theta}_{j,s_j} &= \frac{exp(\tilde{\mu}_{j,s_j})}{\sum_{s'_j=1}^{m_i} exp(\tilde{\mu}_{j,s'_j})} \ . \end{aligned}$$

The set of all question parameters $\theta^t_{i,\boldsymbol{s}^i}$ and all skills parameters $\tilde{\theta}_{j,s_j}$ is $\boldsymbol{\theta}$ without the reparametrization and $\boldsymbol{\mu}$ with the reparametrization.

2.2 Monotonicity

The concept of monotonicity in BNs has been discussed in the literature since the last decade of the previous millennium [14, 3]. Later its benefits for BN parameter learning were addressed, for example, by [12, 2]. This topic is still active, e.g., [4, 11, 5].

We consider only variables with states from $\mathbb{N}_0$ with their natural ordering, i.e., the ordering of states of skill variable S_j for $j \in \boldsymbol{M}$ is

$$s_{j,1} \prec \ldots \prec s_{j,m_j} \ .$$

A variable S_j has a monotone effect on its child X_i if for all $k, l \in \{1, \ldots, m_j\}, t' \in \{0, \cdots, n_i\}$:

$$s_{j,k} \preceq s_{j,l} \quad \Rightarrow \quad \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \ \geq \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s})$$

and antitone effect:

$$s_{j,k} \preceq s_{j,l} \quad \Rightarrow \quad \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \ \leq \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s}) \ ,$$

where $\boldsymbol{s}$ is a configuration of remaining parents of question i without S_j. For each question $X_i, i \in \boldsymbol{M}$ we denote by $\boldsymbol{S}^{i,+}$ the set of parents with a monotone effect and by $\boldsymbol{S}^{i,-}$ the set of parents with an antitone effect.

The conditions above are defined for states of question variable X_i in the set $\{0, \cdots, (n_i - 1)\}$. Given the property of conditional probabilities, i.e.

$$\theta_{i,\boldsymbol{s}^i}^{n_i} = 1 - \sum_{t=0}^{n_i - 1} \theta_{i,\boldsymbol{s}^i}^{t} \ ,$$

it holds for the state n_i in the form for monotonic:

$$s_{j,k} \preceq s_{j,l} \quad \Rightarrow \quad P(X_i = n_i | S_j = s_{j,k}, \boldsymbol{s}) \ \leq \ P(X_i = n_i | S_j = s_{j,l}, \boldsymbol{s})$$

and for antitonic:

$$s_{j,k} \preceq s_{j,l} \quad \Rightarrow \quad P(X_i = n_i | S_j = s_{j,k}, \boldsymbol{s}) \ \geq \ P(X_i = n_i | S_j = s_{j,l}, \boldsymbol{s})$$

Next, we create a partial ordering $\preceq_i$ on all state configurations of parents $\boldsymbol{S}^i$ of the i-th question, where for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i})$:

$$\boldsymbol{s}^i \quad \preceq_i \quad \boldsymbol{r}^i \Leftrightarrow \left(s_j^i \preceq r_j^i, \ j \in \boldsymbol{S}^{i,+}\right) \text{ and } \left(r_j^i \preceq s_j^i, \ j \in \boldsymbol{S}^{i,-}\right) \ .$$

The monotonicity condition then requires that the probability of an incorrect answer is higher for a lower order parent configuration (chances of correct better answers increasing for higher ordered parents' states), i.e., for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i}), k \in \{0, \ldots, (n_i - 1)\}$:

$$\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i \quad \Rightarrow \quad \sum_{t=0}^{k} P(X_i = t | \boldsymbol{S}^i = \boldsymbol{s}^i) \ \geq \ \sum_{t=0}^{k} P(X_i = t | \boldsymbol{S}^i = \boldsymbol{r}^i) \ .$$

In our experimental part we consider only the monotone effect of parents on their children. The difference with antitone effects is only in the partial ordering.

3 Parameter Gradient Search with Monotonicity

To learn the parameter vector $\boldsymbol{\mu}$ we have developed a method based on gradient descent optimization. We follow the work of [2] where authors use a gradient descent method with exterior penalties to learn parameters. The main difference is that we consider models with hidden variables. In this article we generalize the method from [9] to multistate question variables.

We denote by $\boldsymbol{D}$ the set of indexes of question vectors. One vector $x^k, k \in \boldsymbol{D}$ corresponds to one student and an observation of i-th variable X_i is x_i^k. The number of occurrences of the k-th configuration vector in the data sample is d_k.

We use the model as described in Section 2 having unobserved parent variables and observed children variables. With sets $\boldsymbol{I}_t^k, t \in \{0, \ldots, N^{max}\}$ of indexes of questions answered with the point gain of t points, we define the following products based on observations in the k-th vector:

$$p^t(\boldsymbol{\mu}, \boldsymbol{s}, k) = \prod_{i \in \boldsymbol{I}_t^k} \frac{exp(\mu_{i,\boldsymbol{s}}^t)}{\sum_{t'=0}^{n_i} exp(\mu_{i,\boldsymbol{s}}^{t'})},\ t \in \{0, \cdots, N^{max}\}; \quad p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \quad = \quad \prod_{j=1}^{m} exp(\tilde{\mu}_{j,s_j}).$$

We work with the log likelihood of data modelled by BN with the parameter vector $\boldsymbol{\mu}$:

$$\begin{aligned} LL(\boldsymbol{\mu}) &= \sum_{k \in \boldsymbol{D}} d_k \cdot log\left(\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} \prod_{j=1}^{m} \frac{exp(\tilde{\mu}_{j,s_j})}{\sum_{s_j'=1}^{m_j} exp(\tilde{\mu}_{j,s_j'})} \cdot \prod_{t=0}^{N^{max}} p^t(\boldsymbol{\mu}, \boldsymbol{s}, k) \right) \\ &= \sum_{k \in \boldsymbol{D}} d_k \cdot log\Big(\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \prod_{t=0}^{N^{max}} p^t(\boldsymbol{\mu}, \boldsymbol{s}, k)\Big) - N \cdot \sum_{j=1}^{m} log \sum_{s_j'=1}^{m_j} exp(\tilde{\mu}_{j,s_j'}) \ . \end{aligned}$$

In the gradient descent optimization we need partial derivatives to establish the gradient. The partial derivatives of $LL(\boldsymbol{\mu})$ with respect to $\mu_{i,\boldsymbol{s}^i}$ for $i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i)$ are

$$\frac{\delta LL(\boldsymbol{\mu})}{\delta \mu_{i,\boldsymbol{s}^i}^t} =$$

$$\sum_{k \in \boldsymbol{D}} d_k \cdot \quad \frac{I(t, i, \boldsymbol{s^i}, k) - (\sum_{t'=0}^{n_i} exp(\mu_{i,\boldsymbol{s}^i}^{t'}) - exp(\mu_{i,\boldsymbol{s}^i}^{t})) \cdot p_\mu(\boldsymbol{\mu}, \boldsymbol{s}^i) \prod_{t=0}^{N^{max}} p^t(\boldsymbol{\mu}, \boldsymbol{s}, k)}{\sum_{t'=0}^{n_i} exp(\mu_{i,\boldsymbol{s}^i}^{t'}) \cdot \sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} \Big(p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \prod_{t=0}^{N^{max}} p^t(\boldsymbol{\mu}, \boldsymbol{s}, k)\Big)},$$

$$\text{where} \qquad I(t, i, \boldsymbol{s^i}, k) = \begin{cases} exp(\mu_{i,\boldsymbol{s}^i}^t), & \text{if } t = k \\ 0, & \text{otherwise} \end{cases}$$

and with respect to $\tilde{\mu}_{i,l}$ for $i \in \boldsymbol{M}, l \in \{1, \ldots, m_i\}$ are

$$\begin{aligned}\frac{\delta LL(\boldsymbol{\mu})}{\delta \tilde{\mu}_{i,l}} \quad = \quad & \sum_{k \in \boldsymbol{D}} d_k \cdot \frac{\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})}^{s_i = l} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \prod_{t=0}^{N^{max}} p^t(\boldsymbol{\mu}, \boldsymbol{s}, k)}{\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} p_\mu(\boldsymbol{\mu}, \boldsymbol{s}) \prod_{t=0}^{N^{max}} p^t(\boldsymbol{\mu}, \boldsymbol{s}, k)} - \\ & - N \cdot \frac{exp(\tilde{\mu}_{i,l})}{\sum_{l'=1}^{m_i} exp(\tilde{\mu}_{k,l'})} \ .\end{aligned}$$

3.1 Monotonicity Restriction

To ensure monotonicity we use a penalty function which penalizes solutions that do not satisfy monotonicity conditions

$$C(\theta_{i,\boldsymbol{s}^i}, \theta_{i,\boldsymbol{r}^i}, t', c) = exp(c \cdot (\sum_{t=0}^{t'} \theta_{i,\boldsymbol{r}^i}^t - \sum_{t=0}^{t'} \theta_{i,\boldsymbol{s}^i}^t))$$

for the log likelihood:

$$LL'(\boldsymbol{\theta}, c) = LL(\boldsymbol{\theta}) - \sum_{i \in \boldsymbol{N}} \sum_{\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i} \sum_{t'=0}^{N^{max}} C(\theta_{i,\boldsymbol{s}^i}, \theta_{i,\boldsymbol{r}^i}, t', c),$$

and in the case of reparametrized parameters:

$$LL'(\boldsymbol{\mu}, c) = LL(\boldsymbol{\mu}) - \sum_{i \in \boldsymbol{N}} \sum_{\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i} \sum_{t'=0}^{N^{max}} C(\frac{exp(\mu_{i,\boldsymbol{s}^i}^t)}{\sum_{t'=0}^{n_i} exp(\mu_{i,\boldsymbol{s}^i}^{t'})}, \frac{exp(\mu_{i,\boldsymbol{r}^i}^t)}{\sum_{t'=0}^{n_i} exp(\mu_{i,\boldsymbol{r}^i}^{t'})}, t', c),$$

where c is a constant determining the slope of the penalization function. The higher the value the more strict the penalization is. Theoretically, this condition does not ensure monotonicity but, practically, selecting high values of c results in monotonic estimates. If the monotonicity is not violated then the penalty value is close to zero. Otherwise, the penalty is raising exponentially fast. In our experiments we have used the value of $c = 200$ but any value higher than 100 provided almost identical results.

After adding the penalized part to the log likelihood, partial derivatives with respect to $\mu_{i,l}$ remain unchanged. Partial derivatives with respect to $\mu_{i,\boldsymbol{s}^i}^t$ change. The reparametrization causes the derivatives to become very complex. Due to limited space in this paper we do not include their full description here.

Using the penalized log likelihood, $LL'(\boldsymbol{\mu}, c)$, and its gradient $\nabla(LL'(\boldsymbol{\mu}, c))$ we can use standard gradient descent optimization methods to find the paramters of BN models.

4 Experiments

We designed tests to verify our assumptions. We want to show that if we learn parameters of BNs with little amount data it is beneficial to use monotonicity

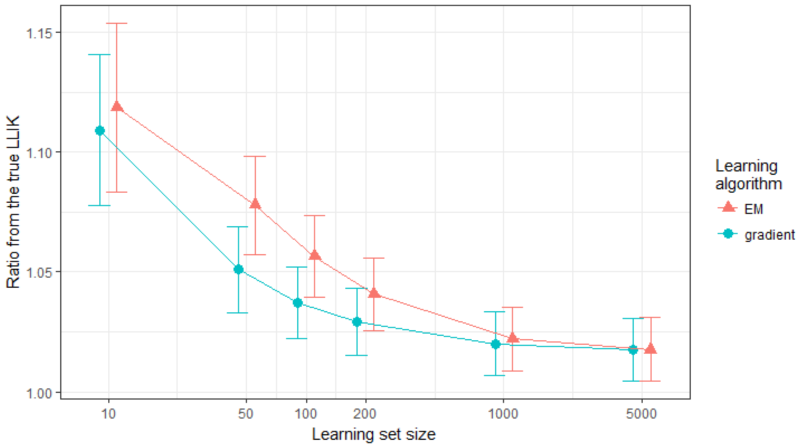


Figure 3: Artificial model: The ratio between the fitted and the real log likelihood (measured on the whole data set) obtained by models trained with EM and the restricted gradient methods for different training set sizes. Notice the logarithmic scale of the x axis. Curves are slightly misaligned in the direction of the x-axis to avoid overlapping.

constraints. We designed two experiments to test the method described above. The first one works with artificial (synthetic data); the other uses a real world empiric data sample.

Parameters are learned with our gradient method and the standard unrestricted EM algorithm. In both cases, we learn model parameters from subsets of data of different sizes. The quality of the parameter fit is measured by the log likelihood. The log likelihood is measured on the whole data set to provide results comparable between subsets of different sizes.

4.1 Artificial Model

The structure of the first model is shown in Figure 1. This model reflects the usual model structure used in CAT where there are two levels of variables, one level of questions, and one level of parents (skills). Parents S_1 and S_2 have 3 possible states and children X_1, X_2, X_3, X_4 also have three states. The model was set up with 10 different sets of parameters $\boldsymbol{\theta}^*_a$ satisfying the monotonicity conditions. Furthermore, every model produced 10 000 test cases.

To learn parameters of these models we drew random subsets of size d of 10, 50, 100, 200, 1 000, 5 000. Ten different sets for each size (indexed by b). Next, we created 10 initial starting points (indexed by c) for the model learning phase. The structure of both generating and learning models is the same and is shown in

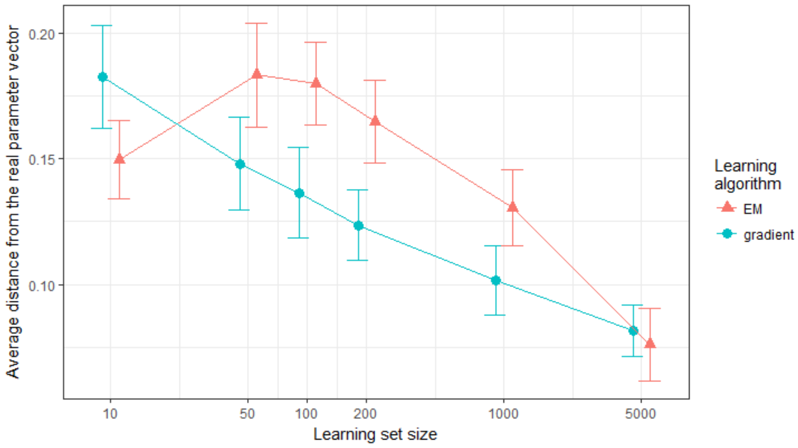


Figure 4: Artificial model: Mean parameter distance between real and fitted parameters in models trained with the EM and restricted gradient methods for different training set sizes. Notice the logarithmic scale of the x axis. Curves are slightly misaligned in the direction of the x-axis to avoid overlapping.

Figure 1. Starting parameter vectors $\boldsymbol{\theta}_b$ are randomized so that they satisfy the monotonicity conditions. Parameters of all parent variables are uniform. Starting points are the same for both the EM and the gradient method alike. In this setup we have 10 different original models, 10 different observation subsets, and 10 different starting parameters, which gives 1 000 combinations for each set size. Each combination has a set of parameters $\boldsymbol{\theta}^d_{a,b,c}, a, b, c \in \{1, \ldots, 10\}$. We performed tests for all these combinations and the results are evaluated as follows.

We measure the log likelihood on the whole data set in order to keep results comparable. The resulting log likelihood after learning is compared with the log likelihood obtained with the real model and then averaged over all instances. This process gives us the average percentual difference between the original and fitted model. For the set size d:

$$LR^d = \frac{\sum_{a,b,c} \dfrac{LL(\boldsymbol{\theta}^*_a)}{LL(\boldsymbol{\theta}^d_{a,b,c})}}{1000}$$

Resulting valus for all set sizes are shown in Figure 3. In this artificial setup we are also able to measure the distance of learned parameters from the generating parameters. First we calculate an average error for each learned model:

$$e^d_{i,j} = \frac{|\boldsymbol{\theta}^*_a - \boldsymbol{\theta}^d_{a,b,c}|}{|\boldsymbol{\theta}|} \quad ,$$

where || is the L1 norm. Next we average over all results in one set size d:

$$e^d = \frac{\sum_{a,b,c} e_{i,j}}{1000} \ .$$

The summary of results is shown in Figure 4.

4.2 CAT Model

The second model is the model presented in Figure 2 and we use it for our CAT research. Parent variables $S_1, \ldots, S_8$ have 3 states and each one of them represents a particular student skill. Children nodes U_i are variables representing questions which have a various number of states (based on the evaluation of the specific question). This model was learned from data contained in the data sample collected from the Czech high school final exam[2]. The data set contains answers from over 20 000 students who took the test in the year 2015. We created the model structure based on our expert analysis and assigned skills to questions. To learn parameters we use random subsets of size of 10, 50, 100, and 500 cases of the whole sample. We drew 10 random sets for each size. Models were initiated with 10 different initial random starting parameters $\boldsymbol{\theta}_i$.

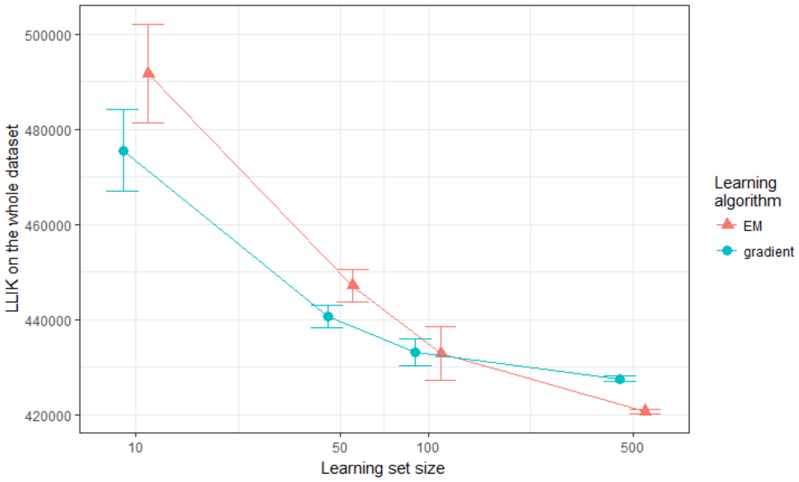


Figure 5: BN model for CAT empirical data: LLIK scored on the whole dataset for models trained with the EM and restricted gradient methods for different training set sizes. Notice the logarithmic scale of the x axis. Curves are slightly misaligned in the direction of the x-axis to avoid overlapping.

[2] The test is accessible here (Czech language):`http://www.statnimaturita-matika.cz/wp-content/uploads/matematika-test-zadani-maturita-2015-jaro.pdf`

For the learned models we computed the log likelihood for the whole data set. These values are then averaged over all results of the same size $LL_A(k)$ similarly to the artificial model. Results are presented in Figure 5. In this case we cannot compare learned parameters because the real parameters are unknown.

5 Conclusions

In this article we presented a new gradient based method for learning parameters of Bayesian Networks under monotonicity restrictions. The method was described and then tested on two data sets. In Figures 3 and 5 it is clearly visible that the newly proposed method provide better results than the general EM algorithm for small set sizes. When the size of learning set grows both method are getting more accurate and fitting data better. As we can see in results of the artificial model, both methods converge to the same point which is almost identical to the log likelihood of the model with real parameters. The speed of convergence is slower for the gradient method, nevertheless in the artificial case, it is not outperformed by the EM algorithm. In the case of empirical data, we can observe the same notion where for small set sizes the new gradient method is scoring better results. In this case EM is getting better log likelihood for larger data sets. This is caused by the fact that for these larger sets monotonicity restrictions start to make the learning process harder. For smaller sets they are showing the right path and guiding the learning process to a better solution. For larger sets they are restricting parameters and making the process harder. On the other hand, in case when we use the gradient method, we are working with learned model satisfying monotonicity conditions which may be desirable given its purpose.

This article shows that it is possible to benefit from monotonicity conditions. It presents the method to be used to learn parameter of BNs under these conditions. A possible extension of our work is to design a method which would use gradient descent optimization in a polytope defined by monotonicity conditions instead of using a penalty function. This approach has certain benefits as it ensures ending with strictly monotonic solution, on the other hand the current method allows small deviations from monotonicity if data strongly contradicts it.

References

[1] R. G. Almond and R. J. Mislevy. Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237, 1999.

[2] E. E. Altendorf, A. C. Restificar, and T. G. Dietterich. Learning from Sparse Data by Exploiting Monotonicity Constraints. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*, 2005.

[3] J. Druzdzel and M. Henrion. Efficient Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 548–553. AAAI Press, 1993.

[4] A. J. Feelders and L. van der Gaag. Learning Bayesian Network Parameters with Prior Knowledge about Context-Specific Qualitative Influences. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*, 2005.

[5] A. R. Masegosa, A. J. Feelders, and L. van der Gaag. Learning from incomplete data in Bayesian networks with qualitative influences. *International Journal of Approximate Reasoning*, 69:18–34, 2016.

[6] T. D. Nielsen and F. V. Jensen. *Bayesian Networks and Decision Graphs (Information Science and Statistics).* Springer, 2007.

[7] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., dec 1988.

[8] M. Plajner and J. Vomlel. Student Skill Models in Adaptive Testing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 403–414. JMLR.org, 2016.

[9] M. Plajner and J. Vomlel. Monotonicity in Bayesian Networks for Computerized Adaptive Testing. In A. Antonucci, L. Cholvy, and O. Papini, editors, *ECSQARU 2017*, pages 125–134, Cham, 2017. Springer International Publishing.

[10] G. Rasch. *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut, 1960.

[11] A. C. Restificar and T. G. Dietterich. Exploiting monotonicity via logistic regression in Bayesian network learning. Technical report, Corvallis, OR : Oregon State University, 2013.

[12] L. van der Gaag, H. L. Bodlaender, and A. J. Feelders. Monotonicity in Bayesian networks. *20th Conference on Uncertainty in Artificial Intelligence (UAI '04)*, pages 569–576, 2004.

[13] W. J. van der Linden and C. A. W. Glas. *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers, 2000.

[14] M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303, 1990.

Learning Bipartite Bayesian Networks under Monotonicity Restrictions

Martin Plajner[a,b] and Jiří Vomlel[b]

[a]Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University
Trojanova 13, Prague 120 00, Czech Republic
[b]Institute of Information Theory and Automation, Czech Academy of Sciences,
Pod vodárenskou věží 4, Prague 8, 182 08, Czech Republic

ARTICLE HISTORY
Compiled October 29, 2019

ABSTRACT
Learning parameters of a probabilistic model is a necessary step in machine learning tasks. We present a method to improve learning from small datasets by using monotonicity conditions. Monotonicity simplifies the learning and it is often required by users. We present an algorithm for Bayesian Networks parameter learning. The algorithm and monotonicity conditions are described, and it is shown that with the monotonicity conditions we can better fit underlying data.

Our algorithm is tested on artificial and empiric datasets. We use different methods satisfying monotonicity conditions: the proposed gradient descent, isotonic regression EM, and non-linear optimization. We also provide results of unrestricted EM and gradient descent methods. Learned models are compared with respect to their ability to fit data in terms of log-likelihood and their fit of parameters of the generating model. Our proposed method outperforms other methods for small sets, and provides better or comparable results for larger sets.

KEYWORDS
Bayesian Networks; monotonicity; parameter learning; isotonic regression; gradient method; computerized adaptive testing

1. Introduction

Our research is focused in the domain of Computerized Adaptive Testing (CAT) working with Bayesian Networks (BNs) to model students' abilities, which is also addressed, for example, by (Almond and Mislevy 1999; van der Linden and Glas 2000). CAT is a concept of testing latent student abilities, which allows creating shorter tests, asking fewer questions while keeping the same level of information. This task is performed by asking each individual student the right questions. Questions are selected based on a student model. In common practice, experts often use Item Response Theory models (IRT) (Rasch 1960), which are well explored and have been in use for a long time. Nevertheless, we have focused our attention on a different family of models. The reason is that Bayesian Networks provide us with better relationships in the model. It is, for example, possible to model more complex influences between skills and questions because BNs are not limited to connecting each skill with each question; moreover,

CONTACT martin.plajner@fjfi.cvut.cz

we can introduce relationships between skills themselves. We address the topic of the model selection in larger detail in our previous work, e.g., Plajner and Vomlel (2016b).

During our research, we have noticed that there are certain conditions which should be satisfied in this specific modeling task. We have especially been focused on monotonicity conditions. Monotonicity conditions incorporate qualitative influences into a model. These influences restrict conditional probabilities inside the model in a specific way to avoid unwanted behavior. Monotonicity in Bayesian Networks has been discussed in the literature for a long time. The most relevant papers are Wellman (1990); Druzdzel and Henrion (1993) and more recently Restificar and Dietterich (2013); Masegosa, Feelders, and van der Gaag (2016). Monotonicity restrictions are often motivated by reasonable demands from model users. In our case of CAT, it means we want to guarantee that students having a higher level of skill(s) will have a higher probability of answering questions correctly. As another example of monotonicity usage, imagine a BN that is learned to predict the effect of commercial promotions of products in retail stores. There are certain factors which should have an isotone effect. For example, secondary placement in the store, i.e., the position in the store's layout. A better position should provide a better result. If it does not, it is most likely caused by other factors or noise in the data. In this case, we want the learned effect to be isotone and our proposed algorithm can be used to provide it.

Certain types of models include monotonicity naturally, due the way in which they are constructed. This is not true in the case of general BNs. In order to satisfy these conditions, we have to introduce restrictions to conditional probabilities during the process of parameter learning.

In our previous work we showed that monotonicity conditions are useful in the context of CAT (Plajner and Vomlel 2016b). Later we applied these conditions to Bayesian Networks (Plajner and Vomlel 2017). In this article, we present a gradient descent optimum search method for BN parameter learning under monotonicity conditions. The algorithm we present provides a tool to include monotonicity in the BN models with multiple-state variables. We implemented the new method in R language and performed experimental verification of our assumptions. Experiments were performed on two datasets. The first one, a synthetic dataset, is generated from artificial models satisfying monotonicity conditions. The second one, an empirical dataset, is newly obtained and it consists of data from the Czech high school state final exam. This second dataset contains a large volume of reliable data, and it is very useful for the empirical verification of our approach. Experiments on these datasets were performed with various parameter learning methods both satisfying and not satisfying the monotonicity restrictions. The results are compared to show differences between individual methods and the approaches with and without considering monotonicity.

In contrast to our previously published articles, this paper brings significant modifications and improvements. Here, we establish a way of using our proposed gradient descent algorithm for BNs that have other than binary variables. We also modified the irEM method, which we use as for reference in work with multi-state variables. In this article we add a new dataset, which is based on large scale real-world data in a domain where the monotonicity should apply. Moreover, we have revised the way to evaluate models in order to create a more precise and comprehensive evaluation. This step includes adding to the comparison additional monotonicity-ensuring methods.

The structure of this article is as follows. First, we establish our notation and describe monotonicity conditions in detail in Section 2.1. Next, we present different methods for learning parameters under monotonicity conditions in Section 3 and afterwords we present our proposed method in Section 3.1. In Section 4, we take a

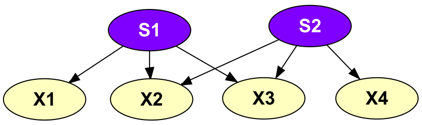


Figure 1. An artificial BN model

closer look at the experimental setup and present results of our experiments. Section 5 contains an overview and a discussion of the obtained results.

2. BN Models and Monotonicity

2.1. *Models and Adaptive Testing*

In our work we focus on computerized adaptive testing and assessing student knowledge and abilities, using Bayesian Networks with a specific structure. The structure is a bipartite network, which consists of a layer of skills and a layer of questions. Skills are parents in our structure and correspond to specific abilities a student may or may not have. Individual states of these skills are interpreted as levels of knowledge. This interpretation is generally difficult as skills are unobserved variables. Having monotonicity constraints in our models, we are able to introduce an ordering of these levels and refer to them as increasing (or decreasing) qualities of skills. Children in the bipartite structure are question nodes, which correspond to particular questions in a test. Levels of these nodes correspond to the points obtained by solving the specific problem (the problem can be divided into sub-problems with different scores). These models are described in further detail in Plajner and Vomlel (2016a).

2.2. *Notation*

In this article, we use BNs to model students in the domain of CAT. Details about BNs can be found, for example, in Pearl (1988); Nielsen and Jensen (2007). The model we use can be considered a special BN structure such as Multi-dimensional Bayesian Network Classifier which is described, e.g., in van der Gaag and de Waal (2006). We restrict ourselves to the BNs that have two levels. In compliance with our previous articles, variables in the parent level are skill variables S. The child level contains question variables X. Examples of network structures, which we also used for experiments, are shown in Figures 1 and 2.

- We use the symbol $\boldsymbol{X}$ to denote the multivariable $(X_1, \ldots, X_n)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_n)$. The total number of question variables is n, the set of all indices of question variables is $\boldsymbol{N} = \{1, \ldots, n\}$. Question variables' individual states are $x_{i,t}, t \in \{0, \ldots, n_i\}$ and they are observable. Each question can have a different number of states; the maximum number of states over all variables is $N^{\max} = \max_i(n_i) + 1$. States are integers with the natural ordering.[1].
- We use the symbol $\boldsymbol{S}$ to denote the multivariable $(S_1, \ldots, S_m)$ taking states

[1] In our case, points are specifying the score obtained in the question i. The interpretation of points is very complex and has to be viewed as per question because we use the CAT framework. In this context, getting one point in one question is not the same as one point in another.

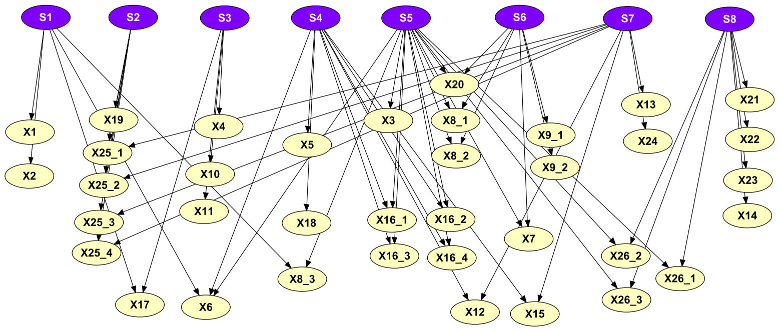


Figure 2. A BN model for CAT

$\boldsymbol{s} = (s_1, \ldots, s_m)$. The set of all indices of skill variables is $\boldsymbol{M} = \{1, \ldots, m\}$. Skill variables have variable numbers of states, the number of states of a variable S_j is m_j, and individual states are $s_{j,k}, k \in \{1, \ldots, m_j\}$. The variable $\boldsymbol{S}^i = \boldsymbol{S}^{pa(i)}$ stands for a multivariable containing the parent variables of the question X_i. Indices of these variables are $\boldsymbol{M}^i \subseteq \boldsymbol{M}$. The set of all possible state configurations of $\boldsymbol{S}^i$ is $Val(\boldsymbol{S}^i)$. Skill variables are unobservable.

The BN is defined by, along with its structure, parameters of all questions $X_i, i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i)$ which define conditional probabilities as

$$\theta^t_{i,\boldsymbol{s}^i} = P(X_i = t | \boldsymbol{S}^i = \boldsymbol{s}^i) \ ,$$

and, parameters of all skills $S_j, j \in \boldsymbol{M}$ as

$$\tilde{\theta}_{j,s_j} = P(S_j = s_j) \ .$$

From the definition above it follows that the parameters are constrained to be between zero and one with constraints for question variables $\sum_t \theta^t_{i,\boldsymbol{s}^i} = 1, \ \forall i, \boldsymbol{s}^i$ and, for parent variables, $\sum_{s_j} \tilde{\theta}_{j,s_j} = 1, \ \forall j$. To avoid these constraints in our gradient method, we reparametrize

$$\begin{aligned} \theta^t_{i,\boldsymbol{s}^i} &= \frac{exp(\mu^t_{i,\boldsymbol{s}^i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s}^i})} \\ \tilde{\theta}_{j,s_j} &= \frac{exp(\tilde{\mu}_{j,s_j})}{\sum_{s'_j=1}^{m_i} exp(\tilde{\mu}_{j,s'_j})} \ . \end{aligned}$$

The set of all question parameters $\theta^t_{i,\boldsymbol{s}^i}$ and all skills parameters $\tilde{\theta}_{j,s_j}$ is denoted by $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$ is the set of reparameterized parameters. The symbol $\boldsymbol{\mu_{i,s^i}} = \left\{ \mu^{t'}_{i,\boldsymbol{s}^i}, t' \in \{0, \ldots, n_i\} \right\}$ stands for the set of parameters for all states of a question X_i given one parent configuration $\boldsymbol{s}^i$. Theoretically, $\mu^{t'}_{i,\boldsymbol{s}^i} \in \mathbb{R}, \forall i, \forall t'$ but for the practical computational issues we forbid the two extreme values of $\boldsymbol{\theta}$, i.e., 0 and 1. We elaborate more on exact bounds in the experimental section of this paper in Section 4.

2.3. *Monotonicity*

The concept of monotonicity in BNs has been discussed in the literature since the 1990s, see Wellman (1990); Druzdzel and Henrion (1993). Later, its benefits for BN parameter learning were addressed, for example, by van der Gaag, Bodlaender, and Feelders (2004); Altendorf, Restificar, and Dietterich (2005); Feelders and van der Gaag (2005). This topic is still active, see, e.g., Restificar and Dietterich (2013); Masegosa, Feelders, and van der Gaag (2016).

We consider only variables with states from $\mathbb{N}_0$ with their natural ordering, i.e., the ordering of states of skill variable S_j for $j \in \boldsymbol{M}$ is

$$s_{j,1} \prec \ldots \prec s_{j,m_j} \ .$$

A variable S_j has an *isotone effect* on its child X_i if for all $k, l \in \{1, \ldots, m_j\}, t' \in \{0, \cdots, n_i - 1\}$ the following holds[2]:

$$s_{j,k} \preceq s_{j,l} \ \ \Rightarrow \ \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \ \geq \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s})$$

and *antitone effect*:

$$s_{j,k} \preceq s_{j,l} \ \ \Rightarrow \ \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \ \leq \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s}) \ ,$$

where $\boldsymbol{s}$ is a configuration of the remaining parents of question i without S_j. For each question $X_i, i \in \boldsymbol{M}$ we denote by $\boldsymbol{S}^{i,+}$ the set of parents with an isotone effect and by $\boldsymbol{S}^{i,-}$ the set of parents with an antitone effect.

The conditions above are defined for the states of question variable X_i in the set $\{0, \cdots, n_i - 1\}$. The sum property of conditional probabilities

$$\sum_{t=0}^{n_i} \theta_{i,\boldsymbol{s}^i}^t = 1 \ ,$$

implies that, for n_i in the case of the isotone effect:

$$s_{j,k} \preceq s_{j,l} \ \ \Rightarrow \ \ P(X_i = n_i | S_j = s_{j,k}, \boldsymbol{s}) \ \leq \ P(X_i = n_i | S_j = s_{j,l}, \boldsymbol{s})$$

and in the case the antitone effect:

$$s_{j,k} \preceq s_{j,l} \ \ \Rightarrow \ \ P(X_i = n_i | S_j = s_{j,k}, \boldsymbol{s}) \ \geq \ P(X_i = n_i | S_j = s_{j,l}, \boldsymbol{s})$$

Next, we define a partial ordering $\preceq_i$ on all state configurations of parents $\boldsymbol{S}^i$ of the i-th question, if for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i})$:

$$\boldsymbol{s}^i \ \ \preceq_i \ \ \boldsymbol{r}^i \Leftrightarrow \left(s_j^i \preceq r_j^i, \ j \in \boldsymbol{S}^{i,+}\right) \text{ and } \left(r_j^i \preceq s_j^i, \ j \in \boldsymbol{S}^{i,-}\right) \ .$$

[2] Note that for n_i this formula always holds since $\sum_{t=0}^{n_i} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) = 1 \quad \forall i, \forall j, \forall k$

The monotonicity condition requires that the probability of an incorrect answer is higher for a lower order parent configuration (the chance of a correct answer increases for higher ordered parents' states), i.e., for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i}), k \in \{0, \ldots, n_i - 1\}$:

$$\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i \;\;\Rightarrow\;\; \sum_{t=0}^{k} P(X_i = t | \boldsymbol{S}^i = \boldsymbol{s}^i) \;\geq\; \sum_{t=0}^{k} P(X_i = t | \boldsymbol{S}^i = \boldsymbol{r}^i) \;\; .$$

In our experimental part, we consider only the isotone effect of parents on their children. The difference with antitone effects is only in the partial ordering.

3. Learning Model Parameters under Monotonicity Conditions

Different methods can be used to learn model parameters while satisfying monotonicity conditions. In this Section, we will outline some of them and then we will describe our newly proposed method. All optimization methods we consider are optimizing the log-likelihood of the model. Methods, in the order as they are described below, are:

- Isotonic regression EM (**irEM**)
- Bounded non-linear optimization (**Cobyla**)
- Restricted gradient method (**res gradient**)

Isotonic Regression EM

Isotonic Regression EM was proposed in Masegosa, Feelders, and van der Gaag (2016). The authors propose a method for parameter learning which ensures convergence to monotonicity satisfying parameters. The method is a modification of the well-known EM algorithm where the M-step is modified to contain an isotonic regression step. This step, in the case of a solution not complying with the monotonicity conditions, moves the solution to the border of the admissible parameter space. The steps of the algorithm are applied iteratively as in the case of the regular EM. In each step a new solution, starting from the previous point, is found. This solution may or may not satisfy the monotonicity conditions. If it does not, isotonic regression is performed to satisfy them. As we show later in this paper, this behavior has a tendency to end at the border of the admissible parameter space. This behavior may imply that the algorithm fails to provide an optimal solution.

We have implemented the generalized version of the irEM algorithm working with multiple state parent variables in our previous paper (Plajner and Vomlel 2017). In the present paper we further generalize the irEM method to work with multiple states of children variables as well.

The authors of the irEM algorithm also provide quick-irEM, abbreviated to qirEM, a version of the algorithm which is a speed optimization modification. In this case the isotonic regression step is performed only once after the EM algorithm converges. In experiments, we have tested this version of the algorithm as well.

Bounded non-linear optimization

The monotonicity constraints form a subspace in the whole parameter space of CPTs' parameters. A possible approach is to apply an optimization method for finding an optimum only inside this subspace. In that case the solution would satisfy the monotonicity constraints and should be optimal (locally or globally based on the algorithm and properties of the space itself). In our experiments we used various methods from NLOPT library for non-linear optimization problems (Johnson 2018). From among methods available in the library, we selected Sequential Least-Squares Quadratic Programming (Kraft 1994) and Constrained Optimization BY Linear Approximations (Cobyla) (Powell 1994) methods. Reasons to select these methods are that they are able to work in our domain of restricted space and non-linear inequalities formed by the monotonicity constraints. They are local optimization techniques and as such they do not guarantee global optimum. We have also experimented with global optimization methods but the time required for these methods to converge was extensive and this is why we decided to skip experiments with these methods.

Restricted Gradient Method

We propose to use the Restricted Gradient Search method (which is our proposed method) to find parameters of a BN under monotonicity restrictions. This method uses the gradient descent optimum search technique. It takes a penalized log-likelihood function to be optimized in order to find the solution of this problem. The penalization encourages the solution to leave the non-admissible area of nonmonotonic parameters and leads the gradient towards a monotonic solution. As such, this method does not strictly ensure monotonicity to hold. Nevertheless, there are two important comments to be made. The strength of the restriction is variable and setting high restriction values effectively enforces the solution to be monotonic. Moreover, if the solution is not monotonic the reason might be that the underlying data strongly contradicts it. This method provides an option to balance data evidence and the monotonicity restrictions and allows to create a non monotonic solution. Even though this is possible to achieve, there is no general rule how to weight these influences. It depends on the data and the model and requires expertise to evaluate. If the user is not sure, we propose to use large penalty values to practically ensure a monotonic solution. This method is described in detail in the following Section.

3.1. *Parameter Gradient Search with Monotonicity*

We have developed a method based on gradient descent optimization. We follow the work of Altendorf, Restificar, and Dietterich (2005) where the authors use a gradient descent method with exterior penalties. The main difference is that we consider models with hidden variables. In this article, we generalize the method from Plajner and Vomlel (2017) to multi-state question variables.

We denote by $\boldsymbol{D}$ the set of indices of the question vectors. One vector $x^k, k \in \boldsymbol{D}$ corresponds to one student and an observation of i-th variable X_i is x_i^k. The number of occurrences of the k-th configuration vector in the data sample is d_k.

We use the BN model described in Section 2.1 where we have unobserved parent variables and observed children variables. The parent variables correspond to skills and the number of their levels set the levels of quality/ability of the skill. The child nodes correspond to questions and the number of levels is the number of possible points obtained in the particular question. Let $\boldsymbol{I}_t^k, t \in \{0, \ldots, N^{\max}\}$ be sets of indices of the questions in a state t. Then, we define the following products based on the observations in the k-th vector[3]:

$$\begin{aligned} p^t(\boldsymbol{\mu},\boldsymbol{s},k) &= \begin{cases} 1, & \text{if } \boldsymbol{I}_t^k=\emptyset \\ \prod_{i\in \boldsymbol{I}_t^k} \dfrac{exp(\mu^t_{i,\boldsymbol{s}^i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s}^i})}, & \text{otherwise} \end{cases} \quad , \quad t\in\{0,\cdots,N^{\max}\}, \\ p_\mu(\boldsymbol{\mu},\boldsymbol{s},k) &= \prod_{t=0}^{N^{\max}} p^t(\boldsymbol{\mu},\boldsymbol{s},k) \\ p_{\tilde{\mu}}(\boldsymbol{\mu},\boldsymbol{s}) &= \prod_{j=1}^{m} exp(\tilde{\mu}_{j,s_j}). \end{aligned}$$

We work with the log-likelihood of data modeled by BN with the parameter vector $\boldsymbol{\mu}$:

$$\begin{aligned} LL(\boldsymbol{\mu}) &= \sum_{k\in\boldsymbol{D}} d_k\cdot log\left(\sum_{\boldsymbol{s}\in Val(\boldsymbol{S})}\prod_{j=1}^{m}\frac{exp(\tilde{\mu}_{j,s_j})}{\sum_{s'_j=1}^{m_j} exp(\tilde{\mu}_{j,s'_j})}\cdot p_\mu(\boldsymbol{\mu},\boldsymbol{s},k)\right) \\ &= \sum_{k\in\boldsymbol{D}} d_k\cdot log\left(\sum_{\boldsymbol{s}\in Val(\boldsymbol{S})} p_{\tilde{\mu}}(\boldsymbol{\mu},\boldsymbol{s})p_\mu(\boldsymbol{\mu},\boldsymbol{s},k)\right) - N\cdot\sum_{j=1}^{m} log\sum_{s'_j=1}^{m_j} exp(\tilde{\mu}_{j,s'_j}) \ . \end{aligned}$$

Monotonicity Restrictions for the Gradient Search

To enforce monotonicity into the model we apply a penalty function which penalizes solutions that do not satisfy the monotonicity conditions. We will use the following penalization function for the log-likelihood:

$$C(\boldsymbol{\mu_{i,s^i}},\boldsymbol{\mu_{i,r^i}},\hat{t},c) \quad = \quad max\left(0, c\cdot\left(\frac{\sum_{t=0}^{\hat{t}} exp(\mu^t_{i,\boldsymbol{r}^i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{r}^i})} - \frac{\sum_{t=0}^{\hat{t}} exp(\mu^t_{i,\boldsymbol{s}^i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s}^i})}\right)^p\right) \quad ,$$

and the penalized log-likelihood is

$$LL'(\boldsymbol{\mu},c) \quad = \quad LL(\boldsymbol{\mu}) - \sum_{i\in\boldsymbol{N}}\sum_{\boldsymbol{s}^i\preceq_i\boldsymbol{r}^i}\sum_{\hat{t}=0}^{N^{\max}-1} C\left(\boldsymbol{\mu_{i,s^i}},\boldsymbol{\mu_{i,r^i}},\hat{t},c\right) \quad ,$$

where p sets the degree of the polynomial function and it takes only odd values, c is a constant determining the slope of the penalization function, and $\hat{t}$ is the level of the question node. The higher the value of c the more strict the penalization is. Theoretically, this condition does not ensure monotonicity but, practically, selecting high values of c results in monotonic estimates. The polynomial penalty uses an odd degree polynomial function. We discuss the size of the penalty in the following Section.

Using the penalized log-likelihood, $LL'(\boldsymbol{\mu},c)$, and its gradient $\nabla(LL'(\boldsymbol{\mu},c))$, we can use standard gradient descent optimization methods to learn the parameter vector $\boldsymbol{\mu}$ of BN models. We provide formulas to compute the gradient in Appendix A.

[3] As we use only reparameterized parameters in our gradient method, we provide only formulas with the reparametrization, i.e., the parameter vector $\boldsymbol{\mu}$ as was introduced in Section 2.2

3.2. *Ensuring Monotonicity with the Penalization*

Penalization described above may provide a solution which is not monotone. This behavior is observable especially in instances in which the data strongly contradict the monotonicity conditions. The solution will always be close to the admissible region but the distance depends on the strength of the penalization. It depends on the specific application whether we require a strictly monotone result or not. In many cases it may be acceptable to break these conditions in order to get a better data fit. When the training sample is very small, it is particularly easy to have data that contradicts monotonicity. However, in some situations, we need to enforce monotonicity. It is particularly easy to measure the distance from the border of the admissible region. We can use the iterative process to ensure monotonicity. If the final parameter vector after the optimization violates the monotonicity conditions, we restart the optimization with a stronger penalization and use the end point as a new starting point. This process is repeated until the monotone solution is reached. Nevertheless, in this Section, we also provide a way to ensure monotonicity conditions by setting a strong enough penalization.

In order to be able to do that and to compare this method with other strictly monotone methods we propose the following concept. Below we use the penalization $C(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c)$.

The penalization of the log-likelihood described above and detailed in Appendix A has to lead the gradient method to the admissible area. We need to ensure that for each $\mu^t_{i,\boldsymbol{s^i}}$ with the parent configuration $\boldsymbol{s^i}$ in the term

$$\begin{aligned} \frac{\partial LL'(\boldsymbol{\mu}, c)}{\partial \mu^t_{i,\boldsymbol{s^i}}} &= \frac{\partial LL(\boldsymbol{\mu})}{\partial \mu^t_{i,\boldsymbol{s^i}}} && (1) \\ &\quad - \sum_{\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i} \sum_{\hat{t}=0}^{N^{\max}} \frac{\partial C\left(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c\right)}{\partial \mu^t_{i,\boldsymbol{s^i}}} && (2) \\ &\quad - \sum_{\boldsymbol{r}^i \preceq_i \boldsymbol{s}^i} \sum_{\hat{t}=0}^{N^{\max}} \frac{\partial C\left(\boldsymbol{\mu_{i,r^i}}, \boldsymbol{\mu_{i,s^i}}, \hat{t}, c\right)}{\partial \mu^t_{i,\boldsymbol{s^i}}} \quad , && (3) \end{aligned}$$

the gradient part of $LL(\boldsymbol{\mu}, c)$ (1) is not larger than the penalization terms (2) and (3) while the parameter vector $\boldsymbol{\mu}$ is not in the admissible region. The two terms (2) and (3) of the penalization gradient are generated by the monotonicity conditions where each condition generates one item to the outer sum for one or both of them. The first term (2) is for the situation $\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i$ and the second one (3) for the opposite instance $\boldsymbol{r}^i \preceq_i \boldsymbol{s}^i$. For a single parameter $\mu^t_{i,\boldsymbol{s^i}}$ these two gradient parts have opposite effects.

We need to analyze the partial ordering of skill configurations

$$\boldsymbol{r}^i \preceq_i \boldsymbol{s}^i, \boldsymbol{s}^i \preceq_i \boldsymbol{r}^i,\ \boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S}^i)$$

determining conditions of the question X_i and, more specifically, a single parameter $\mu^t_{i,\boldsymbol{s^i}}$. Because the penalty and its gradient is zero when the condition is not violated, we can omit the state configurations for which the condition holds and work only with the configurations for which the penalty is positive, i.e., all pairs $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val'(\boldsymbol{S}^i) \subseteq$

$Val(\boldsymbol{S}^i)$ for which

$$C(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c) > 0 \ .$$

Given this reduced set $Val'(\boldsymbol{S}^i)$ and the partial ordering, there is always one state configuration which is the first in the partial ordering. It means that there exists at least one configuration $\hat{\boldsymbol{s}}^i$ for which

$$\left\{\hat{\boldsymbol{s}}^i \preceq_i \boldsymbol{r}^i\right\} = \emptyset, \forall \boldsymbol{r}^i \in Val'(\boldsymbol{S}^i) \ .$$

In the part of the gradient corresponding to parameter $\mu^t_{i,\boldsymbol{s}^i}$ one of the two sums $(\boldsymbol{r}^i \preceq_i \hat{\boldsymbol{s}}^i)$ is zero. If we are able to ensure that the penalization part of the gradient is always larger outside of the admissible region for this parameter, it will be moved to the admissible region by the gradient method. After this step, the whole solution either is in the admissible region, or we can use the same process to move another parameter to the admissible region as long as there are any parameters outside of the region.

The penalization drops towards zero as it gets closer to the border of the admissible region. This behavior creates computational difficulties. The cause of these difficulties lies in the fact that very small values of penalization can be outweighed by improvements of the log-likelihood by shifting parameters outside of the admissible space. Thus, it would be hard to ensure monotonicity in such conditions. To avoid these issues, we shrink the admissible region by adding a small margin β to the penalization function:

$$\begin{gathered} C'_{2,p}(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c, \beta) = \\ max\left(0, c \cdot \left(\frac{\sum_{t=0}^{\hat{t}} exp(\mu^t_{i,\boldsymbol{r}^i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{r}^i})} - \frac{\sum_{t=0}^{\hat{t}} exp(\mu^t_{i,\boldsymbol{s}^i})}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s}^i})} + \beta\right)^p\right) \ . \end{gathered}$$

This makes the lowest possible value at the border of the admissible region to be

$$C^*_{2,p}(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c, \beta) = c \cdot \beta^p$$

As the penalization function is growing rapidly outside of the admissible region, it is sufficient to ensure the gradient inequality between terms (1), (2) and (3) in the formula above at the border.

Based on the reasoning above and the formulas to compute the gradient, we set the constant $c = c^*$ to ensure monotonicity in the following way

$$\begin{aligned} \left|\frac{\partial LL(\boldsymbol{\theta})}{\partial \theta^t_{i,\boldsymbol{s}^i}}\right| &< \left|\frac{\partial C\left(\boldsymbol{\theta_{i,s^i}}, \boldsymbol{\theta_{i,r^i}}, \hat{t}, c\right)}{\partial \theta^t_{i,\boldsymbol{s}^i}}\right| \\ \frac{1}{(\tilde{\theta}_-)^m \cdot (\theta_-)^n} &< cp\beta^{p-1} \cdot (\theta_-)^2 \\ c &> (\tilde{\theta}_-)^{-m} \cdot (\theta_-)^{-n-2} \cdot \beta^{1-p} \ , \end{aligned}$$

where $\beta = 0.01$ and $p = 3$ are chosen constants for the penalization. $\tilde{\theta}_-$ and θ_- are the minimal possible parameters values as

$$\begin{aligned}\tilde{\theta}_- &= \frac{exp(\tilde{\mu}_-)}{(m-1)exp(\tilde{\mu}_+) + exp(\tilde{\mu}_-)} \\ \theta_- &= \frac{exp(\mu_-)}{(n-1)exp(\mu_+) + exp(\mu_-)}\end{aligned} \quad ,$$

where $\tilde{\mu}_-, \tilde{\mu}_+, \mu_-, \mu_+$ are the bounds on reparameterized parameters which we use to prevent the probability values from reaching zero or one. In our case we use the maximum of 3 and the minimum of -3, which effectively changes the interval of probabilities of a three-state variable to approximately $[0.0012; 0.995]$.

3.3. *Isotonic Regression EM for Variables with Multiple States*

As mentioned earlier we use the isotonic regression EM method as a comparison method to our proposed gradient approach. Our algorithm is designed for variables having multiple states. The original irEM algorithm as it is published in Masegosa, Feelders, and van der Gaag (2016) only works with binary variables. In our previous paper (Plajner and Vomlel 2017) we detailed our implementation of this method to work with parent variables in bipartite networks having multiple states. In order to be able to make the full comparison with the method proposed in this paper, we also implemented the irEM algorithm based on original work of Masegosa, Feelders, and van der Gaag (2016), and Feelders (2007), where more information about the generalization to non-binary cases can be found, to work with multi-state child variables as well. We provide details of our implementation in this Section.

For the sake of simplicity, we describe the implementation for isotone effects only as antitone effects are simple reversions. In our case of multiple states, a variable S_j has an *isotone effect* on its child X_i if for all $k, l \in \{1, \ldots, m_j\}, t' \in \{0, \cdots, n_i - 1\}$:

$$s_{j,k} \preceq s_{j,l} \quad \Rightarrow \quad \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \;\geq\; \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s})$$

More information about the stochastic dominance which is used here to model monotonicity in terms of cumulative distributions can be found in Wellman (1990).

The difference between the binary case and the multi-state case lies in the cumulative probability. In the binary case there was only one series of inequalities for $t = 0$. Nevertheless, the structure of inequalities is the same as in the original irEM algorithm for each level of $t \in \{0, \ldots, n_i\}$. We propose the use of a series of isotonic regression steps. Each step works with a single level of t, i.e., with cumulative probabilities of question X_i in separate sets

$$I^t = \left\{ \sum_{t'=0}^{t} \theta_{i,\boldsymbol{s}^i}^{t'} \right\}, \forall t \in \{0, \ldots, n_i\} \ .$$

We perform the isotonic regression algorithm for each set separately with weights as relative frequencies in the same way as in the standard irEM algorithm to obtain new cumulative probabilities. These probabilities are afterward converted back to non-cumulative probabilities, i.e., individual variables.

4. Experiments

In experiments we would like to verify that if we learn parameters of BNs from a small volume of data it is beneficial to use monotonicity constraints. We designed two experiments to compare the methods discussed in this paper. In the first experiment we use artificial (synthetic) data; the other uses a real world empiric data sample. There are two model versions for each dataset. One with binary and one with ternary question nodes, creating a total of four different model types we worked with.

Parameters are learned using the methods described above: our gradient method, unrestricted gradient descent, irEM, qirEM, regular EM, and Cobyla from NLOPT methods family[4]. For all model types, we learn the model parameters from subsets of data of different sizes. The quality of the parameter fit is measured by the log-likelihood of the learned models. The log-likelihood is measured on the whole dataset to provide results comparable between learning subsets of different sizes.

We implemented the methods in R and its various built-in packages to ease this process (R Development Core Team 2008), the NLOPT package mentioned above, and for computations of the regular EM algorithm, the Hugin (Hugin 2014) engine was used as the most time efficient tool. One important point to mention is that we restricted the parameters of the learned conditional probability tables to be from the specific interval $[\epsilon, 1-\epsilon]$ where $\epsilon \in [0,1]$ is a chosen small number; we used $\epsilon = 10^{-3}$. This step is carried out in order to avoid extreme parameter values. When the learning sample is very small, the networks parameters tend to move towards zero or one, but we know it should not be the case in reality. These limits are very similar for the reparameterized case of our gradient method as described in Section 3.2. The gradient method is penalized by the constraint described in Section 3.1 and it takes the parameter p defining the degree of the polynomial function. In our experiments we have always used the third degree as it proves, empirically, to converge fastest to the solution.

4.1. *Artificial Model*

The structure of the first model is shown in Figure 1. This model has a typical model structure used in CAT where there are two levels of variables, one level of questions, and one level of skills (parents). Skills S_1 and S_2 have three possible states and questions X_1, X_2, X_3, X_4 are either binary or ternary, creating two different sets for further testing. Models were set up with ten different sets of parameters $\boldsymbol{\theta}_a^*$ satisfying the monotonicity conditions. Furthermore, each model was used to generate one million of data samples (test results of a student, i.e., answers to questions). Parent variables were unobserved in all cases.

To learn the parameters of these models, we drew random subsets of size $d = 10^k$, where $k \in \{1,2,3,4,5,6\}$. Note that for $k=6$ the subset is the set itself. Ten different sets for each size (indexed by b) were generated. Next, we created ten initial

[4] The reason to include regular EM and unrestricted gradient methods is to further verify the benefit of using the monotonicity constraints. We want to provide a reader with a comparison also between the restricted and unrestricted cases.

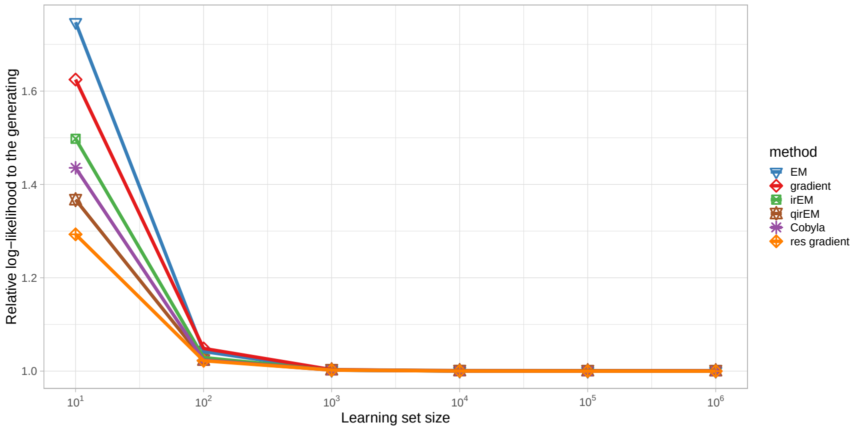


Figure 3. Artificial model, binary questions: The ratio between log-likelihoods of the fitted and the generating models.

starting points (indexed by c) for the model learning phase. The structures of both the generating and the learned models are fixed to be the same as it is shown in Figure 1. The starting parameter vectors $\boldsymbol{\mu}_b$ and the corresponding $\boldsymbol{\theta}_b$ were randomly generated from the interval $[0.01, 0.99]$. The starting points were the same for all methods. In this setup, we have ten different original models, ten different observation subsets, and ten different starting parameters, which provides us with a thousand combinations for each set size and each model. Each model $M^d, d \in \left\{10^k, k \in \{1, \ldots, 10\}\right\}$ is specified by a set of parameters $\boldsymbol{\theta}^d_{a,b,c}, a, b, c \in \{1, \ldots, 10\}$. We performed experiments for all these combinations and the results are evaluated as follows.

We measure the log-likelihood on the whole dataset in order to keep the results comparable. The log-likelihood of each learned model is compared with the log-likelihood of the generating model and then averaged over all instances of (a, b, c). This process gives us the average log-likelihood ratio between the generating and the fitted model for each subset of size d:

$$LR^d = \frac{1}{1000} \sum_{a,b,c} \frac{LL(\boldsymbol{\theta}^*_a)}{LL(\boldsymbol{\theta}^d_{a,b,c})} \quad .$$

In this artificial setup we are also able to measure the distance of the probability distributions of learned parameters Q from the the distribution of generating parameters P. First, we calculate the average Kullback–Leibler divergence for each learned model:

$$D_{KL}(\theta^*_a || \theta^d_{a,b,c}) = \frac{1}{n} \sum_{i=1}^{n} D_{KL}(P(X_i|\theta^*_a) || Q(X_i|\theta^d_{a,b,c})) \quad .$$

Next we average over all results for each subset of size d:

$$D^d_{KL} = \frac{1}{1000} \sum_{a,b,c} D_{KL}(\theta^*_a || \theta^d_{a,b,c}) \quad .$$

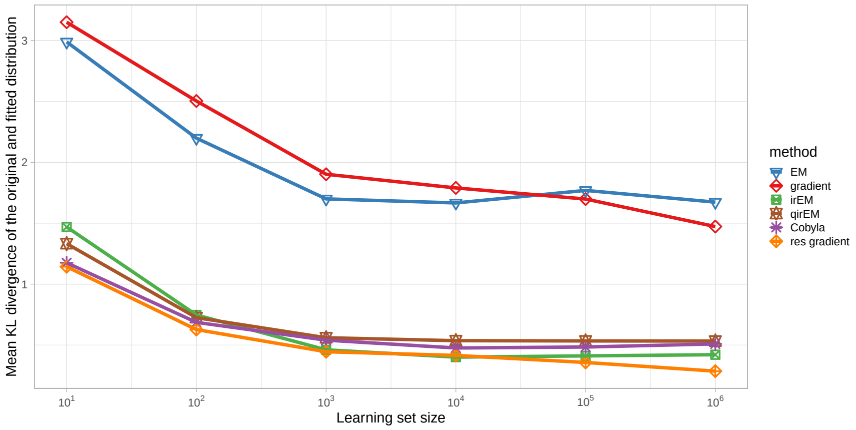


Figure 4. Artificial model, binary questions: The mean KL divergence of the fitted and generating probability distributions.

We require all methods which are restricted to satisfy the monotonicity conditions. As described in Section 3.2, we can ensure this behavior by setting a high value of the penalization parameter c. In this setting $c = 10^{20}$ satisfies this condition. Even though it is possible to use this penalization, it is very high and in certain cases it is numerically hard to reach convergence without the algorithm failing. Instead we use a smaller penalization of $c = 10^5$ which is, together with offset $\beta = 0.01$, sufficient for practical purposes to satisfy the condition, although it does not theoretically guarantee it. For each solution we verify whether it lies in the admissible region or not and the solution which does not will not be used. For the actual penalization settings none of solutions obtained in our experiments lied outside the admissible region and, because of that, all restricted methods are comparable as they provide solutions under the same restrictions. For better detail, we measured the situation for a much smaller penalization of $c = 100$ and the smallest learning set size, where the danger of not satisfying the monotonicity conditions is the highest. Even in this case only under 10% of initial solutions end outside the admissible region.

Binary Question Variables

In this Section we present results for the artificial model with binary question variables. The resulting values of the relative log-likelihood LR measured on the whole dataset for all set sizes are shown in Figure 3. Figure 4 then shows the KL divergence of the learned parameter distributions from the parameters of the generating distribution. In both Figures, the horizontal axis has the logarithmic scale.

As we can see in Figure 3 all methods converge to the same log-likelihood value very quickly. Differences are mostly in the smaller set sizes of 10 and 100 observations. Unrestricted methods are clearly performing worse than the methods using the restrictions. The isotonic regression and NLOPT methods provide similar results; and the methods of restricted gradient provide the best solutions for small sets. In the case of the KL divergence (in Figure 4), we can clearly see that monotonicity helps us obtain parameters which are closer to the real ones. In order to establish a sound ordering of the methods, we performed Wilcoxon's test. The null hypothesis was that one method is not giving better (lower) results. The p-values resulting from this test are presented in Table B1. We can see that, in most cases, the restricted-gradient methods outperform the other methods at a significant level. The Cobyla and irEM methods are scoring very similarly against other methods but when pair-wise compared, the irEM is performing better.

This model is small and all methods converge to a solution quite fast. Nevertheless, the EM and irEM methods are the fastest as they use the graph decomposition and update the CPTs separately. In the case of other methods, the structure remains complex (which is caused by unobserved parent variables) and such computations are more time-consuming for larger networks. The main problem is the increasing state space created by the state combinations of parents. As the number of parameters increases, the number of conditions increases as well, and computing the gradient also takes considerably more time. Especially in the case of NLOPT methods, this problem is significant.

Ternary Question Variables

The same testing scenario was used for ternary question variables. The results for relative log-likelihood are shown in Figure 5 and the divergence values of parameter distributions in Figure 6. These Figures are constructed in the same way as those in the binary case. These results are very similar to the case with binary question variables. Wilcoxon's test results for this case are displayed in Table B2. They are almost identical to the previous case. The main difference is that the performance of the Cobyla method has significantly decreased. We can also observe that the order of methods is not exactly the same in both figures. The first figure shows the ability of methods to fit data. It is measured by the log-likelihood criterion. The second figure shows the distance of the fitted parameters to the parameters of the generating distribution. Models that have a high log-likelihood need not necessarily represent the best fit when it is measured by the distance of the parameters. Therefore we provide both views.

An interesting point to point out is that, unlike the other methods, the Cobyla one did not reach exactly the log-likelihood ratio 1 and the estimates of its parameters are leveled at an early stage. The reason for such behavior lies in the computational demands of this family of methods. The time which is sufficient for other methods is not sufficient for the Cobyla method to converge. We have also tested other possible methods from the NLOPT family, including the global optimization method. The global optimization method had problems finding a solution even in the binary problem. Another NLOPT method, Sequential Least-Squares Quadratic Programming (SLSQP), which is a very fast local optimization method working well for the binary scenario, was not able to reach the solution either for this specific problem.

Example of Isotonic Regression EM Behavior

We observed a problematic behavior of the irEM algorithm, which happens when the algorithm repeatedly leaves the space of admissible solutions during the EM step. We illustrate this behavior using a simple example. Figure 7 shows the log-likelihood during the fitting process. For the irEM algorithm, iterations are broken down into two consequent steps - the EM step and the isotonic regression step; the latter moves the parameters to the border of the admissible region. We can observe oscillations which are caused by leaving and re-entering the admissible region. This behavior creates an

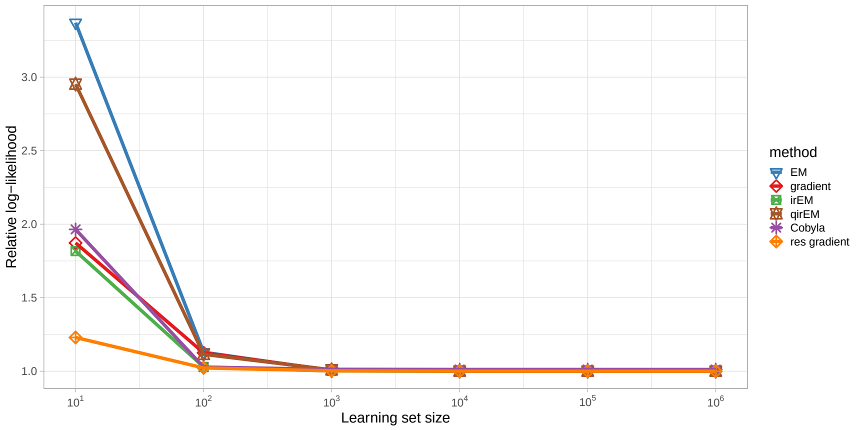


Figure 5. Artificial model, ternary questions: The ratio between log-likelihoods of the fitted and the generating models.

obstacle to finding the better solution in the similar way as local extremes do. The irEM algorithm fails to find a better solution which is reachable by the method from inside of the admissible space (starting parameters already monotonic). In Figure 8 we display the number of the violated monotonicity conditions. In fact, this behavior may also cause problems with the stopping criteria as the algorithm returns to the border possibly very close to the previous state after the ir step with a very similar log-likelihood value.

In Figure 7 we also present results of qirEM method. This method runs as the EM algorithm and performs the isotonic step after EM iterations. We can observe that the fitted log-likelihood is smaller than for the two other methods, but in the last step as the qirEM method satisfies the monotonicity conditions the log-likelihood rises above both concurrent methods. qirEM thus provides a valid solution but it is a heuristic which can potentially provide worse log-likelihood fits.

4.2. *CAT Model*

The structure of the second tested model is presented in Figure 2. Parent variables $S_1, \ldots, S_8$ have 3 states and each of them represents a particular student skill. Child nodes X_i are variables representing questions that have different numbers of states (based on the evaluation of the specific question). We learned this model from the data of the Czech high school final exam[5]. This dataset contains answers from over 20,000 students who took the test in the year 2015. We created the model structure based on our expert analysis and assigned questions to relevant skills. We used random subsamples of the whole data sample with sizes of 10, 40, 160, 640, and 2560. We drew 10 random sets for each size. Models were initiated with 10 different random parameter vectors $\boldsymbol{\mu}_i$ and the corresponding $\boldsymbol{\theta}_i$.

This model was learned using our restricted gradient method and unrestricted gradient and EM methods for reference. In this case, we do not compare to the irEM

[5] The test assignment and its solution are accessible in the Czech language at:`http://www.statnimaturita-matika.cz/wp-content/uploads/matematika-test-zadani-maturita-2015-jaro.pdf`

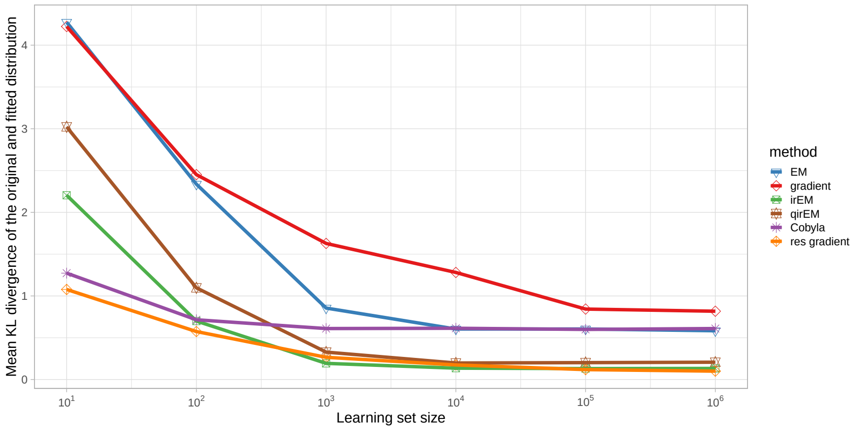


Figure 6. Artificial model, ternary questions: The mean KL divergence of the fitted and generating probability distributions.

method as we are not able to measure the divergence of the parameter distributions and the comparison would not be informative. The NLOPT family methods failed to obtain any solution in the given time (4 hours).

We compute the log-likelihood of the learned models on the whole dataset. These values are then averaged similarly to the artificial model. The results are presented in Figure 9. In this case, we cannot compare the learned parameters with the real ones because the latter are unknown. In the Figure, we can observe that, for empirical data, the restricted gradient methods provide better results for small datasets. The differences in the log-likelihood get lower for larger sets but, even in these cases, parameters of EM and unrestricted gradient learning are usually not monotonic. The parameter space is very large and these methods get easily trapped in a local optimum outside of the monotonicity region.

5. Conclusions

In this article we present a new gradient based method for learning parameters of Bayesian Networks under monotonicity restrictions. Our method is tested on two datasets. When considering the log-likelihood criterion, it is clearly visible in Figures 3, 5 and 9 that the new method provides better results than other methods for small training set sizes. When the size of the learning set grows, all methods are getting more accurate and fit the data better. The results obtained by all tested methods are very similar in terms of the log-likelihood criterion - except for the non-linear optimization approach, which in some cases, failed to obtain any solution due to computational difficulties. For synthetic data, all methods converge to models with the same log-likelihood values, which are nearly identical with those of the log-likelihood of the generating model. In the case of empirical data, we can observe the same behavior. Again, for small training set sizes the new gradient method is scoring better. All methods converge to identical log-likelihood values for large training data sets. Nevertheless, even for the large sets, parameters learned by a non-monotone method, such as EM or the unrestricted gradient, remain non-monotone. The parameter space

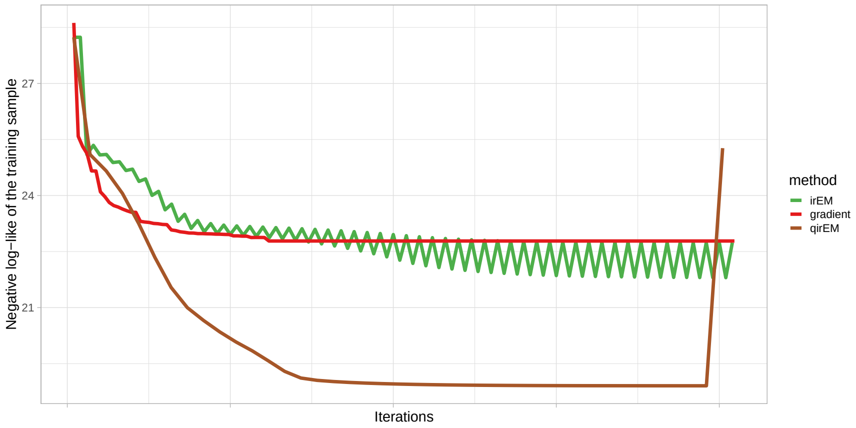


Figure 7. The evolution of the log-likelihood on the training sample during learning iterations of methods. The x axis has not the same scale for all methods as the speed of convergence and the number of steps is not relevant in this case. For the irEM method we display both steps of the iteration in sequence (EM and ir).

is large and it is easy for these methods to get stuck in a local extreme with a not worse log-likelihood value but breaking the monotonicity conditions.

With the synthetic data generated from an artificial model, we are able to compare the fitted parameters with those of the generating model. These comparisons show that the newly proposed method is able to provide results which are closer to the original parameters in all cases. The only drawback of the new method is that it requires longer computational time than the irEM algorithm.

To summarize, we have shown that the learning methods can improve their behavior if they make use of valid monotonicity conditions. We have thus presented a new method that can be used to learn parameters of BNs under the monotonicity conditions. This method performs better in terms of the log-likelihood as well as of a distance from the original model parameters.

There are still open issues concerning the monotonicity conditions and learning parameters under them. One point to address is a generalization of our proposed algorithm to work on general BN structures rather than on bipartite graphs only. The potential application area of the general models is large. One example where we can use the monotonicity conditions regarding promotions planning is mentioned in the introduction of this paper. Another example is disease modeling where we could introduce the monotonicity to model increasing chances of a disease occurrence for higher levels of negative effects such as smoking. Hence it would be beneficial to further explore this research topic to provide larger possibilities to learn and use monotone models.

Acknowledgements

This work was supported by the Czech Science Foundation (Project No. 16-12010S and Project No. 19-04579S) and by the Grant Agency of the Czech Technical University in Prague, Grant No. SGS17/198/OHK4/3T/14.

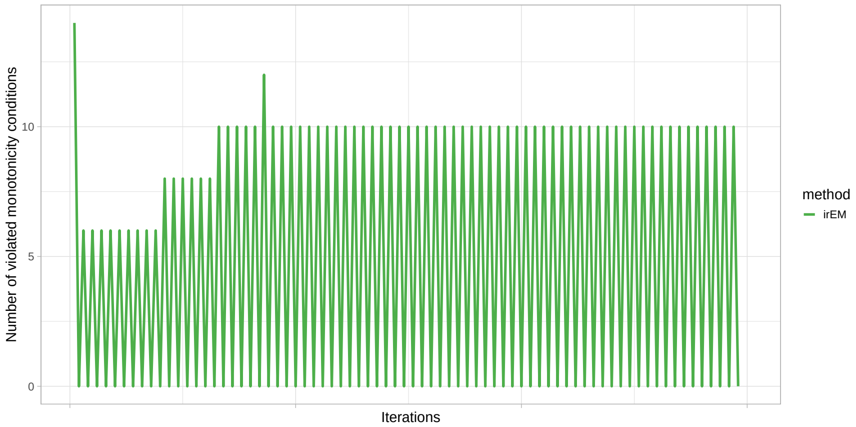


Figure 8. The evolution of the number of violated monotonicity constraints on the training sample for one example case - the binary artificial model. The irEM method displays both steps (EM and ir) in sequence.

References

Almond, R. G., and R. J. Mislevy. 1999. “Graphical Models and Computerized Adaptive Testing.” *Applied Psychological Measurement* 23 (3): 223–237.

Altendorf, E. E., A. C. Restificar, and T. G. Dietterich. 2005. “Learning from Sparse Data by Exploiting Monotonicity Constraints.” *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)* .

Druzdzel, J., and M. Henrion. 1993. “Efficient Reasoning in Qualitative Probabilistic Networks.” In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 548–553. AAAI Press.

Feelders, A. J. 2007. “A new parameter Learning Method for Bayesian Networks with Qualitative Influences.” In *UAI 2007, Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, Vancouver, BC, Canada, July 19-22, 2007*, 117–124.

Feelders, A. J., and L. C. van der Gaag. 2005. “Learning Bayesian Network Parameters with Prior Knowledge about Context-Specific Qualitative Influences.” *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)* .

Hugin. 2014. “Explorer, ver. 8.0, Comput. Software 2014, http://www.hugin.com.” .

Johnson, S. G. 2018. *The NLopt nonlinear-optimization package*. Technical Report.

Kraft, D. 1994. “Algorithm 733: TOMP–Fortran modules for optimal control calculations.” *ACM Transactions on Mathematical Software* 20 (3): 262–281.

Masegosa, A. R., A. J. Feelders, and L. C. van der Gaag. 2016. “Learning from incomplete data in Bayesian networks with qualitative influences.” *International Journal of Approximate Reasoning* 69: 18–34.

Nielsen, T. D., and F. V. Jensen. 2007. *Bayesian Networks and Decision Graphs (Information Science and Statistics)*. Springer.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc.

Plajner, M., and J. Vomlel. 2016a. *Probabilistic Models for Computerized Adaptive Testing: Experiments*. Technical Report. ArXiv:.

Plajner, M., and J. Vomlel. 2016b. “Student Skill Models in Adaptive Testing.” In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, 403–414. JMLR.org.

Plajner, M., and J. Vomlel. 2017. “Monotonicity in Bayesian Networks for Computerized Adap-

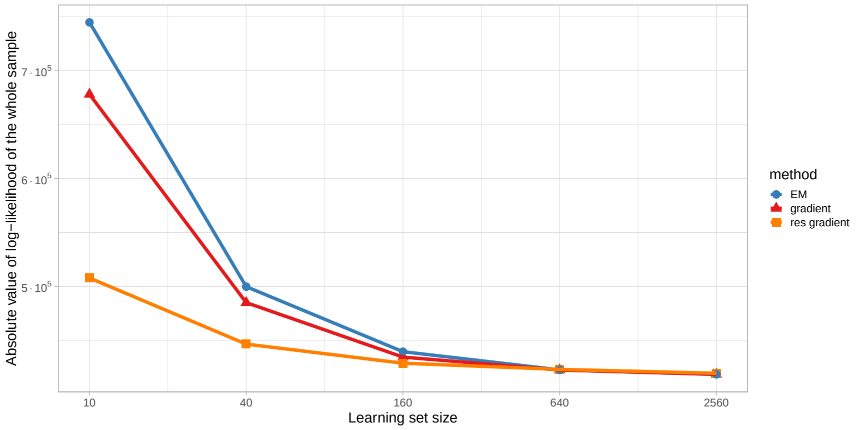


Figure 9. BN model for CAT empirical data: LLIK scored on the whole dataset for models trained with the EM and restricted gradient methods for different training set sizes. Notice the modified logarithmic scale on the x axis ($x = log_4(x'/10)$).

tive Testing." In *ECSQARU 2017*, edited by Alessandro Antonucci, Laurence Cholvy, and Odile Papini, Cham, 125–134. Springer International Publishing.

Powell, M. J. D. 1994. "A direct search optimization method that models the objective and constraint functions by linear interpolation." *Advances in Optimization and Numerical Analysis, eds. S. Gomez and J.-P. Hennart (Kluwer Academic: Dordrecht)* 51–67.

R Development Core Team. 2008. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rasch, G. 1960. *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut.

Restificar, A. C., and T. G. Dietterich. 2013. *Exploiting monotonicity via logistic regression in Bayesian network learning.* Technical Report. Corvallis, OR : Oregon State University.

van der Gaag, L. C., H. L. Bodlaender, and A. J. Feelders. 2004. "Monotonicity in Bayesian networks." *20th Conference on Uncertainty in Artificial Intelligence (UAI '04)* 569–576.

van der Gaag, L. C., and Peter de Waal. 2006. "Multi-dimensional Bayesian Network Classifiers." 01, 107–114.

van der Linden, W. J., and C. A. W. Glas. 2000. *Computerized Adaptive Testing: Theory and Practice.* Vol. 13. Kluwer Academic Publishers.

Wellman, M. P. 1990. "Fundamental concepts of qualitative probabilistic networks." *Artificial Intelligence* 44 (3): 257–303.

Appendix A. Monotonicity Restricted Gradient

In the gradient descent optimization, we need partial derivatives to establish the gradient. The partial derivatives of $LL(\boldsymbol{\mu})$ with respect to $\mu_{i,\boldsymbol{s}^i}$ for $i \in \boldsymbol{N}, \boldsymbol{s}^i \in Val(\boldsymbol{S}^i)$

are

$$\frac{\partial LL(\boldsymbol{\mu})}{\partial \mu^t_{i,\boldsymbol{s^i}}} =$$

$$\sum_{k \in \boldsymbol{D}} d_k \cdot \quad \frac{I(t,i,\boldsymbol{s^i},k) - \left(\left(\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s^i}}) \right) - exp(\mu^t_{i,\boldsymbol{s^i}}) \right) \cdot p_{\tilde{\mu}}(\boldsymbol{\mu}, \boldsymbol{s^i}) p_{\mu}(\boldsymbol{\mu}, \boldsymbol{s^i}, k)}{\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s^i}}) \cdot \sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} \left(p_{\tilde{\mu}}(\boldsymbol{\mu}, \boldsymbol{s}) p_{\mu}(\boldsymbol{\mu}, \boldsymbol{s}, k) \right)},$$

where

$$I(t,i,\boldsymbol{s^i},k) \quad = \quad \begin{cases} exp(\mu^t_{i,\boldsymbol{s^i}}), & \text{if } t = k \\ 0, & \text{otherwise} \end{cases}$$

and with respect to $\tilde{\mu}_{j,l}$ for $j \in \boldsymbol{M}, l \in \{1, \ldots, m_j\}$ are

$$\frac{\partial LL(\boldsymbol{\mu})}{\partial \tilde{\mu}_{j,l}} \quad = \quad \sum_{k \in \boldsymbol{D}} d_k \cdot \frac{\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})}^{s_j = l} p_{\tilde{\mu}}(\boldsymbol{\mu}, \boldsymbol{s}) p_{\mu}(\boldsymbol{\mu}, \boldsymbol{s}, k)}{\sum_{\boldsymbol{s} \in Val(\boldsymbol{S})} p_{\tilde{\mu}}(\boldsymbol{\mu}, \boldsymbol{s}) p_{\mu}(\boldsymbol{\mu}, \boldsymbol{s}, k)} - N \cdot \frac{exp(\tilde{\mu}_{j,l})}{\sum_{l'=1}^{m_j} exp(\tilde{\mu}_{k,l'})} \ .$$

The partial derivative of the penalization function $C(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c)$ is

$$\begin{aligned} \frac{\partial C(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c)}{\partial \mu^t_{i,\boldsymbol{s^i}}} \quad &= \quad -p \cdot C_{p-1}(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c) \\ &\quad \cdot \frac{g(\mu^t_{i,\boldsymbol{s^i}}, \hat{t}) \cdot \left(\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s^i}}) \right) - exp(\mu^t_{i,\boldsymbol{s^i}}) \cdot \sum_{t'=0}^{\hat{t}} exp(\mu^{t'}_{i,\boldsymbol{s^i}})}{\left(\sum_{t'=0}^{n_i} exp(\mu^{t'}_{i,\boldsymbol{s^i}}) \right)^2}, \end{aligned}$$

where

$$g(\mu^t_{i,\boldsymbol{s^i}}, \hat{t}) = \begin{cases} 0, & \text{if } t > \hat{t} \\ exp(\mu^t_{i,\boldsymbol{s^i}}), & \text{if } t \leq \hat{t} \end{cases}$$

and

$$\frac{\partial C(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c)}{\partial \mu^t_{i,\boldsymbol{r^i}}} \quad = \quad -\frac{\partial C(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c)}{\partial \mu^t_{i,\boldsymbol{s^i}}}$$

The partial derivative of $LL'(\boldsymbol{\mu}, c)$ with respect to $\mu^t_{i,\boldsymbol{s^i}}$ is then

$$\begin{aligned} \frac{\partial LL'(\boldsymbol{\mu}, c)}{\partial \mu^t_{i,\boldsymbol{s^i}}} \quad &= \quad \frac{\partial LL(\boldsymbol{\mu})}{\partial \mu^t_{i,\boldsymbol{s^i}}} - \sum_{\boldsymbol{s^i} \preceq_i \boldsymbol{r^i}} \sum_{\hat{t}=0}^{N^{\max}} \frac{\partial C\left(\boldsymbol{\mu_{i,s^i}}, \boldsymbol{\mu_{i,r^i}}, \hat{t}, c\right)}{\partial \mu^t_{i,\boldsymbol{s^i}}} \\ &\quad - \sum_{\boldsymbol{r^i} \preceq_i \boldsymbol{s^i}} \sum_{\hat{t}=0}^{N^{\max}} \frac{\partial C\left(\boldsymbol{\mu_{i,r^i}}, \boldsymbol{\mu_{i,s^i}}, \hat{t}, c\right)}{\partial \mu^t_{i,\boldsymbol{s^i}}} \ . \end{aligned}$$

and the partial derivatives with respect to $\tilde{\mu}_{i,l}$ are not affected by the penalization as the parents do not appear in the penalization function.

$$\frac{\partial LL'(\boldsymbol{\mu})}{\partial \tilde{\mu}_{j,l}} = \frac{\partial LL(\boldsymbol{\mu})}{\partial \tilde{\mu}_{j,l}}$$

Together

$$\frac{\partial LL'(\boldsymbol{\mu})}{\partial \tilde{\mu}_{j,l}} \quad \text{for} \quad \{\tilde{\mu}_{j,l} | j \in \boldsymbol{M}, l \in \{1, \ldots, m_j\}\}$$

and

$$\frac{\partial LL'(\boldsymbol{\mu}, c)}{\partial \mu^t_{i,\boldsymbol{s}^i}} \quad \text{for} \quad \{\mu^t_{i,\boldsymbol{s}^i} | i \in \boldsymbol{N}, t \in \{1, \ldots, N^{\max}\}\}$$

form the gradient $\nabla LL'(\boldsymbol{\mu}, c)$.

Appendix B. Wilcoxon's tests

This appendix contains two tables, Table B1 and Table B2, with results of Wilcoxon's test for the KL divergences of generating and fitted probability distributions in artificial models with binary and ternary variables.

Table B1. Wilcoxon's test to compare results for artificial model with binary variables for different sizes of the learning sets. This test statistically verifies whether a method in the row statistically fits significantly better the generating parameters than another method in the column (H0: there is no shift in their distributions).

10	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5005	**0.0000**	0.0099	**0.0000**	**0.0000**	0.1533
irEM	1.0000	0.5005	0.9986	**0.0000**	**0.0000**	1.0000
qirEM	0.9902	0.0014	0.5005	**0.0000**	**0.0000**	0.8996
EM	1.0000	1.0000	1.0000	0.5005	0.0708	1.0000
gradient	1.0000	1.0000	1.0000	0.9295	0.5005	1.0000
Cobyla	0.8473	**0.0000**	0.1009	**0.0000**	**0.0000**	0.5005

100	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5005	**0.0002**	0.0055	**0.0000**	**0.0000**	0.0966
irEM	0.9998	0.5005	0.8708	**0.0000**	**0.0000**	0.9679
qirEM	0.9946	0.1297	0.5005	**0.0000**	**0.0000**	0.8624
EM	1.0000	1.0000	1.0000	0.5005	0.0025	1.0000
gradient	1.0000	1.0000	1.0000	0.9976	0.5005	1.0000
Cobyla	0.9038	0.0323	0.1382	**0.0000**	**0.0000**	0.5005

1000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5005	0.1313	**0.0000**	**0.0000**	**0.0000**	**0.0010**
irEM	0.8692	0.5005	**0.0000**	**0.0000**	**0.0000**	0.0099
qirEM	1.0000	1.0000	0.5005	**0.0000**	**0.0000**	0.9556
EM	1.0000	1.0000	1.0000	0.5005	0.0132	1.0000
gradient	1.0000	1.0000	1.0000	0.9869	0.5005	1.0000
Cobyla	0.9990	0.9902	0.0446	**0.0000**	**0.0000**	0.5005

1000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5005	0.8738	**0.0000**	**0.0000**	**0.0000**	0.0027
irEM	0.1267	0.5005	**0.0000**	**0.0000**	**0.0000**	0.0013
qirEM	1.0000	1.0000	0.5005	**0.0000**	**0.0000**	0.9951
EM	1.0000	1.0000	1.0000	0.5005	0.1048	1.0000
gradient	1.0000	1.0000	1.0000	0.8956	0.5005	1.0000
Cobyla	0.9973	0.9987	0.0050	**0.0000**	**0.0000**	0.5005

100000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5005	0.2825	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	0.7183	0.5005	**0.0000**	**0.0000**	**0.0000**	0.0031
qirEM	1.0000	1.0000	0.5005	**0.0000**	**0.0000**	0.9984
EM	1.0000	1.0000	1.0000	0.5005	0.8101	1.0000
gradient	1.0000	1.0000	1.0000	0.1905	0.5005	1.0000
Cobyla	1.0000	0.9970	0.0017	**0.0000**	**0.0000**	0.5005

1000000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5151	0.1965	0.0026	**0.0000**	**0.0000**	0.0038
irEM	0.8237	0.5151	0.0827	**0.0000**	**0.0000**	0.1577
qirEM	0.9981	0.9284	0.5151	**0.0000**	**0.0000**	0.6981
EM	1.0000	1.0000	1.0000	0.5151	0.7255	1.0000
gradient	1.0000	1.0000	1.0000	0.3019	0.5177	0.9999
Cobyla	0.9972	0.8612	0.3304	**0.0000**	**0.0001**	0.5177

Table B2. Wilcoxon's test to compare results for artificial model with ternary variables for different sizes of the learning sets. This test statistically verifies whether a method in the row statistically fits significantly better the generating parameters than another method in the column (H0: there is no shift in their distributions).

10	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**	1.0000
qirEM	1.0000	1.0000	0.5000	**0.0000**	**0.0000**	1.0000
EM	1.0000	1.0000	1.0000	0.5000	0.9995	1.0000
gradient	1.0000	1.0000	1.0000	**0.0005**	0.5000	1.0000
Cobyla	1.0000	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.5000
100	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**
qirEM	1.0000	1.0000	0.5000	**0.0000**	**0.0000**	1.0000
EM	1.0000	1.0000	1.0000	0.5000	**0.0000**	1.0000
gradient	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
Cobyla	1.0000	1.0000	**0.0000**	**0.0000**	**0.0000**	0.5000
1000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5001	0.8917	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	0.1084	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**
qirEM	1.0000	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**
EM	1.0000	1.0000	1.0000	0.5000	**0.0000**	1.0000
gradient	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
Cobyla	1.0000	1.0000	1.0000	**0.0000**	**0.0000**	0.5000
10000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5001	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**
qirEM	1.0000	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**
EM	1.0000	1.0000	1.0000	0.5000	**0.0000**	**0.0000**
gradient	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
Cobyla	1.0000	1.0000	1.0000	1.0000	**0.0000**	0.5000
100000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**	**0.0000**
qirEM	1.0000	1.0000	0.5000	**0.0000**	**0.0000**	**0.0000**
EM	1.0000	1.0000	1.0000	0.5000	**0.0000**	**0.0000**
gradient	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
Cobyla	1.0000	1.0000	1.0000	1.0000	**0.0000**	0.5000
1000000	res gradient	irEM	qirEM	EM	gradient	Cobyla
res gradient	0.5005	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
irEM	1.0000	0.5005	**0.0000**	**0.0000**	**0.0000**	**0.0000**
qirEM	1.0000	1.0000	0.5005	**0.0000**	**0.0000**	**0.0000**
EM	1.0000	1.0000	1.0000	0.5005	0.0996	0.0260
gradient	1.0000	1.0000	1.0000	0.9009	0.5005	0.3323
Cobyla	1.0000	1.0000	1.0000	0.9742	0.6686	0.5005

Monotonicity in practice of adaptive testing

Martin Plajner[1,2] PLAJNER@UTIA.CAS.CZ

Jiří Vomlel[1] VOMLEL@UTIA.CAS.CZ

[1] *Institute of Information Theory and Automation,*
Academy of Sciences of the Czech Republic
[2] *Faculty of Nuclear Sciences and Physical Engineering,*
Czech Technical University

Abstract

In our previous work we have shown how Bayesian networks can be used for adaptive testing of student skills. Later, we have taken the advantage of monotonicity restrictions in order to learn models fitting data better. This article provides a synergy between these two phases as it evaluates Bayesian network models used for computerized adaptive testing and learned with a recently proposed monotonicity gradient algorithm. This learning method is compared with another monotone method, the isotonic regression EM algorithm. The quality of methods is empirically evaluated on a large data set of the Czech National Mathematics Exam. Besides advantages of adaptive testing approach we observed also advantageous behavior of monotonic methods, especially for small learning data set sizes. Another novelty of this work is the use of the reliability interval of the score distribution, which is used to predict student's final score and grade. In the experiments we have clearly shown we can shorten the test while keeping its reliability. We have also shown that the monotonicity increases the prediction quality with limited training data sets. The monotone model learned by the gradient method has a lower question prediction quality than unrestricted models but it is better in the main target of this application, which is the student score prediction. It is an important observation that a mere optimization of the model likelihood or the prediction accuracy do not necessarily lead to a model that describes best the student.

Keywords: Monotonicity; Adaptive Testing, Bayesian Network; Gradient Method; Isotonic Regression; Parent Divorcing.

1. Introduction

Computerized Adaptive Testing (CAT) is a concept of testing latent student abilities, which allows creating shorter tests, asking fewer questions while obtaining the same level of information. This task is performed by asking each individual student the most informative questions selected based on a student model. In practice, experts often use the Item Response Theory models (IRT) (Rasch, 1960), which are well explored and have been in use for a long time. We work with Bayesian Networks (BNs) to model students' abilities instead. This approach can be also found, for example, in (Almond and Mislevy, 1999; van der Linden and Glas, 2000).

Over the last few years we addressed different topics from the domain of CAT. We focused mainly on two topics. The first one is the adaptive testing itself and the use of BN models to perform it, see e.g., Plajner and Vomlel (2016b). The second topic concerns the effect of monotonicity restrictions while learning the model, e.g., Plajner and Vomlel (2020). The current article takes the best from both topics and joins them together in a synergy. Here, we use monotone models to perform simulated adaptive tests. For this purpose we use a data set of the Czech Nation Mathematics Exam[1]. This exam serves as a high school evaluation exam and the final grade from this exam is considered important. In this article we introduce an approach for inferring the final score and for the prediction of the expected grade of a student. We also provide a method for establishing the 95% confidence interval of the score which does not require a specific distribution assumption. We observe the evolution of the grade prediction quality during the test and the improvement of the confidence interval. We apply monotone methods, namely our proposed restricted gradient Plajner and Vomlel (2016b) and the isotonic regression EM by Masegosa et al. (2016), as well as the standard (non monotone) EM, and the gradient methods to compare with.

IRT assumes that there is a hidden variable of a student's skill. This approach motivated us to use a structured Bayesian network to model student's skills. In this article we show that the choice of model evaluation criteria is critical in order to select the right model for the given task. It depends whether we want to create a model which predicts the vector of student answers the best or a model which model student's skills the best. The discovery we uncover in this article is that this distinction is also important in the model selection. Sometimes, the best option is to measure the accuracy of answers prediction or the overall fit of data, i.e. likelihood. The reasonable expectation is that when we are able to do this task the best the model would also model the student the best. Nevertheless, as we discuss in the following sections it is not always the case. We can find models which have worse answer prediction accuracy but they better indicate the student skills as it is reflected by the final score/grade obtained in the test. In other words, the model is less certain about the individual's answers but despite that it models the student better.

The article is structured as follows. In Section 2 we go through the necessary notation and describe the models used. Section 3 brings the methodology for student scoring and grading as well as the formulas to evaluate the precision of models. In Section 4 we describe the experimental settings used for the empirical evaluation and results of these experiments are summarized in Section 5. Finally, Section 6 concludes the paper and recollect the main observations and benefits of this paper.

2. BN Models and Monotonicity

2.1 Models and Adaptive Testing

In our work we focus on computerized adaptive testing and assessing student knowledge and abilities, using Bayesian Networks with a specific structure. The structure is a bipartite network, which consists of a layer of skills and a layer of questions. Skills are parents in our structure and correspond to specific abilities a student may or may not have. Individual states of these skills are interpreted as levels of knowledge. This interpretation is generally difficult as skills are unobserved variables. Having monotonicity constraints in our models, we are able to introduce an ordering of these levels and refer to them as increasing (or decreasing) qualities of skills. Children in the bipartite structure are question nodes, which correspond to particular questions in a test. Levels of these nodes correspond to the points obtained by solving the specific problem (the problem can be divided into sub-problems with different scores). These models are described in further detail in Plajner and Vomlel (2016a).

[1] The test assignment and its solution are accessible in the Czech language at: http://www.statnimaturita-matika.cz/wp-content/uploads/matematika-test-zadani-maturita-2015-jaro.pdf

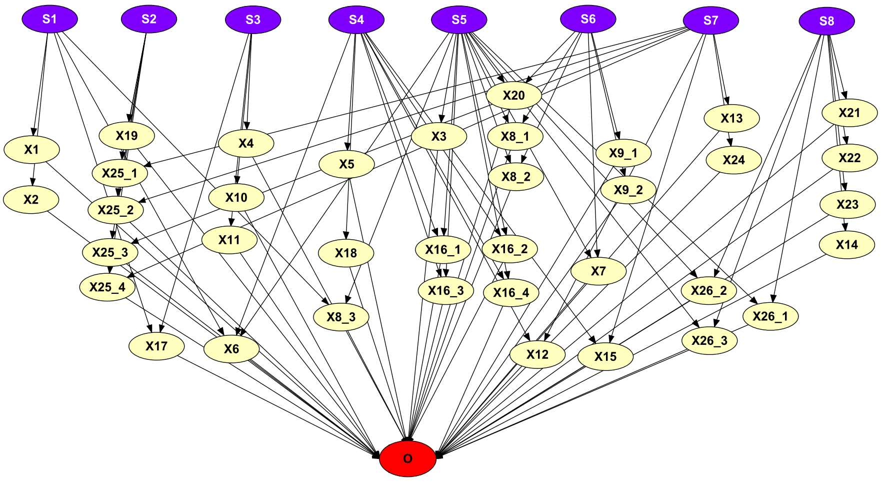


Figure 1: A BN model for CAT

2.2 Notation

We use BNs to model students knowledge. Details about BNs can be found, for example, in Pearl (1988); Nielsen and Jensen (2007). The model we use can be considered a special BN structure such as Multi-dimensional Bayesian Network Classifier which is described, e.g., in Gaag and de Waal (2006). We restrict ourselves to BNs that have two levels of nodes. In compliance with our previous articles, variables in the parent level are skill variables S. The child level contains question variables X. An example of a BN structure, which we also used in experiments, is shown in Figure 1.

- We use the symbol $\boldsymbol{X}$ to denote the multivariable $(X_1, \ldots, X_n)$ taking states $\boldsymbol{x} = (x_1, \ldots, x_n)$. The total number of question variables is n, the set of all indices of question variables is $\boldsymbol{N} = \{1, \ldots, n\}$. Question variables' individual states are $x_{i,t}, t \in \{0, \ldots, n_i\}$ and they are observable. Each question can have a different number of states; the maximum number of states over all variables is $N^{\max} = \max_i(n_i) + 1$. States are integers with the natural ordering.

- We use the symbol $\boldsymbol{S}$ to denote the multivariable $(S_1, \ldots, S_m)$ taking states $\boldsymbol{s} = (s_1, \ldots, s_m)$. The set of all indices of skill variables is $\boldsymbol{M} = \{1, \ldots, m\}$. Skill variables have a variable number of states, the number of states of a variable S_j is m_j, and the individual states are $s_{j,k}, k \in \{1, \ldots, m_j\}$. The variable $\boldsymbol{S}^i = \boldsymbol{S}^{pa(i)}$ stands for a multivariable containing the parent variables of the question X_i. Indices of these variables are $\boldsymbol{M}^i \subseteq \boldsymbol{M}$. The set of all possible state configurations of $\boldsymbol{S}^i$ is $Val(\boldsymbol{S}^i)$. Skill variables are unobservable.

- We use the symbol O to denote the score node taking states $o_k, k \in \{0, \ldots, \sum_{i \in \boldsymbol{N}} x_{i,n_i}\}$. Its state space is the set of all possible sums of question points and it is modeled as sum rule with questions as parents as shown in Figure 1. The maximum number of points is refered to as $o^m = \sum_{i \in \boldsymbol{N}} x_{i,n_i}$.

2.3 Monotonicity

The concept of monotonicity in BNs has been discussed in the literature since the 1990s, see Wellman (1990); Druzdzel and Henrion (1993). Later, its benefits for BN parameter learning were addressed, for example, by van der Gaag et al. (2004); Altendorf et al. (2005); Feelders and van der Gaag (2005). This topic is still active, see, e.g., Restificar and Dietterich (2013); Masegosa et al. (2016).

We consider only variables with states from $\mathbb{N}_0$ with their natural ordering, i.e., the ordering of states of skill variable S_j for $j \in \boldsymbol{M}$ is

$$s_{j,1} \prec \ldots \prec s_{j,m_j} \ .$$

A variable S_j has an *isotone effect* on its child X_i if for all $k, l \in \{1, \ldots, m_j\}, t' \in \{0, \cdots, n_i - 1\}$ the following holds[2]:

$$s_{j,k} \preceq s_{j,l} \ \ \Rightarrow \ \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \ \geq \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s})$$

and *antitone effect*:

$$s_{j,k} \preceq s_{j,l} \ \ \Rightarrow \ \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) \ \leq \ \sum_{t=0}^{t'} P(X_i = t | S_j = s_{j,l}, \boldsymbol{s}) \ ,$$

where $\boldsymbol{s}$ is a configuration of the remaining parents of question i without S_j. For each question $X_i, i \in \boldsymbol{M}$ we denote by $\boldsymbol{S}^{i,+}$ the set of parents with an isotone effect and by $\boldsymbol{S}^{i,-}$ the set of parents with an antitone effect.

Next, we define a partial ordering $\preceq_i$ on all state configurations of parents $\boldsymbol{S}^i$ of the i-th question, if for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i})$:

$$\boldsymbol{s}^i \quad \preceq_i \quad \boldsymbol{r}^i \Leftrightarrow \left(s^i_j \preceq r^i_j,\ j \in \boldsymbol{S}^{i,+}\right) \text{ and } \left(r^i_j \preceq s^i_j,\ j \in \boldsymbol{S}^{i,-}\right) \ .$$

The monotonicity condition requires that the probability of an incorrect answer is higher for a lower order parent configuration (the chance of a correct answer increases for higher ordered parents' states), i.e., for all $\boldsymbol{s}^i, \boldsymbol{r}^i \in Val(\boldsymbol{S^i}), k \in \{0, \ldots, n_i - 1\}$:

$$\boldsymbol{s}^i \preceq_i \boldsymbol{r}^i \ \ \Rightarrow \ \ \sum_{t=0}^{k} P(X_i = t | \boldsymbol{S}^i = \boldsymbol{s}^i) \ \geq \ \sum_{t=0}^{k} P(X_i = t | \boldsymbol{S}^i = \boldsymbol{r}^i) \ .$$

In our experimental part, we consider only the isotone effect of parents on their children. The difference with antitone effects is only in the partial ordering.

[2] Note that for n_i this formula always holds since $\sum_{t=0}^{n_i} P(X_i = t | S_j = s_{j,k}, \boldsymbol{s}) = 1 \quad \forall i, \forall j, \forall k$

3. Score prediction and student grading

In testing it is important to associate a score and/or a grade to a particular student who is being tested. In the adaptive test there are multiple possible options to receive these values. Some options to obtain the student score are described in Plajner and Vomlel (2016a) where, for example, we used estimated skills of the student to compute the score. Nevertheless, the most natural approach seems to be to compute the expected value of the score using the probability distribution of questions' answers in the current state of the student model. In our application we use two different ways how to compute the final score. In both cases we first infer probability distributions of skills of the particular student and then

A. obtain the expected score of remaining unanswered questions, or

B. obtain the expected score of all questions (i.e. also those that were already answered).

Each option is appropriate for a particular scenario. The first one is used in the case the student is tested and we want to estimate his/her result. Questions which were answered define the part of the total score known with certainty and only remaining questions add uncertainty to the total score. This way the test can be evaluated in the just manner. The second approach is more suited for the adaptive learning scenario where we estimate the student score to measure his/her abilities. In this case each question node actually represents a set of similar questions in the test battery. In principal a similar question can be asked again and the answer does not need to be necessary the same, albeit it is most probable it would be.

It is also important to observe not only the expected value but also the distribution of the score. We model the score distribution by an additional node in the Bayesian network, the score node O. This node has as many states as there are possible points to be obtained in the test. The node probability distribution is given by a simple sum rule of its parents (questions). The problem which we have to address in this case is that the dimension of the CPT of this node is very large. We work with 37 questions which have two or more states. Even if they were binary the full state space would be of dimension of 2^{37}. This value is very large and it does not allow direct inference due to the memory size limit. We use the parent divorcing method as described in Olesen et al. (1989). Another option is to use the rank-one decomposition as it is described in Savicky and Vomlel (2007). The reduction of the computational time is very significant as it is outlined it in Figure 2. We show the increase of the time necessary to perform the inference based on the number of questions we connect together. The computational time of the inference is computed only for smaller number of questions as it is not feasible for the standard case in larger numbers.

In this way we obtain the distribution of student's score over the point scale as shown, for example, in Figure 6. Using this distribution we can estimate the expected score and its 95% confidence interval as well. This confidence interval is obtained in the following manner. We sort the states of the node O (points scale) in terms of the states' probabilities in the descending manner. We select all states until the total cumulative probability exceeds 0.95. The probability distribution over tends to be similar to the Gaussian distribution but it is not a rule. The advantage of the proposed approach is that it does not require a specific distribution assumption.

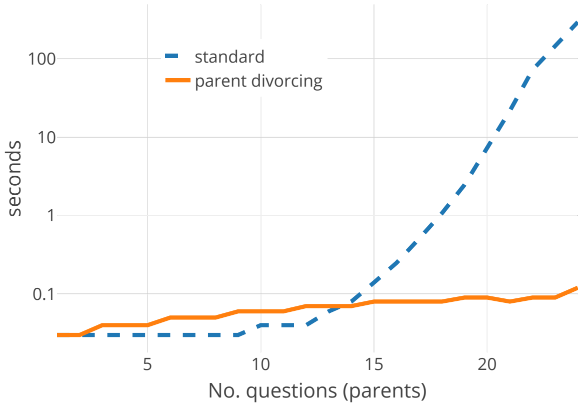


Figure 2: Time of inference in the standard and parent divorcing approach (please note the vertical log scale).

We establish two key performance measures to evaluate different methods in the adaptive testing scenarios. We measure

$$\text{the accuracy of answers prediction} \qquad a^Q = \frac{\sum_{i \in \boldsymbol{N}} I(x_i = x_i^*)}{n} \tag{1}$$

$$\text{the abs. error of the total score prediction} \qquad e^S = |o^* - \sum_{j=1}^{o^m} j \cdot P(O = j)| \tag{2}$$

in each step of the adaptive test for each learning method used, where the function $I(x_i = x_i^*)$ returns one as the maximum likelihood state equals the observed state for the question i and o^* is the real obtained score.

From the perspective of a student the most important measure is the test grade; especially in our special case of the National Exam. The problem of assigning a grade to a student can be viewed as a classification problem which aims at placing a student into the correct grade category. We assume there are G grades where each grade is given for the resulting score in a range of points $G_i, i \in 1, \ldots, G$. The expected grade g is then established from the score variable as

$$g = \underset{i \in 1,\ldots,G}{\text{argmax}} \Big(\sum_{j \in G_i} P(O = j) \Big) \ . \tag{3}$$

The error of this classification is then computed as

$$e^g = \sum_{i \in 1,\ldots,G} |g^* - i| \cdot \sum_{j \in G_i} P(O = j) \ , \tag{4}$$

where g^* is the observed grade.

4. Experimental setup

For experiments in this article we use the data set of the Czech National Mathematics Exam. This exam is taken at the end of the high school and the same test is taken by each student in the same term. Given the nature of this test these data set is valid in terms of student motivation to complete the test as good as possible and data quality is high.

Our experiments were performed according to the following scheme. From all available tests we first drew a random subset to serve as a training set. With each set we train BN models with different learning methods, namely regular EM, regular gradient (grad), isotonic regression EM (irEM), quick irEM (qirEM), and restricted gradient (rgrad). There are 10 random starting points, same for each method to start the learning process at. From the resulting 10 learned models we select the winning model based on the optimization criteria which is the log-likelihood value measured on the training sample. In our previous article Plajner and Vomlel (2020) we compared individual methods on the log-likelihood of the complete data set. In this article the main focus is on adaptive testing usage and we simulate the adaptive testing scenario. The procedure above is performed 10 times with different data selected for learning for each learning set size of 10, 40, and 160 students (i.e. test results). Final ten results for each learning set size are then averaged over the measured metrics.

These learned models are further used in the adaptive testing scenario. We select 100 students which did not figure in any previously selected sets. These students are tested in the simulated test. Tests are performed in two different ways

- fixed and
- adaptive.

The first one is selected in order to provide better insight into comparison of methods. In the adaptive version of testing different questions may be selected for each method in each step. This fact makes the comparison harder in some aspects. On the other hand the ability of a model to be used adaptively is a desired one and we provide comparison of both approaches as well.

5. Results

5.1 Student classification

The grading in the Czech National Mathematics Exam is given by the following scheme.
0-16 points: 5; 17-25 points: 4; 26-34 points: 3; 35-43 points: 2; 44-52 points: 1
In the experiments each student is assigned the expected final grade in every step and the error e^g of this assignment is measured as described in (4). Figure 3 shows the evolution of this error during the adaptive test with models with the learning set of size 10. We show only the version B of questions' answers predictions based only on the inferred skill. Because of that it does not end in zero as we never have the absolute certainty of a student score while knowing only his/her skills. This corresponds to the real-life situation with the margin for errors and mistakes during taking the test even by the best students. In this figure we can see that the restricted gradient method provides the best results. In the end of the test it is on the same level as unrestricted EM and better than other methods. Due to the limited space, we do not include the case A of questions' answers where we predict only remaining questions because it behave very similarly in the most important part of the

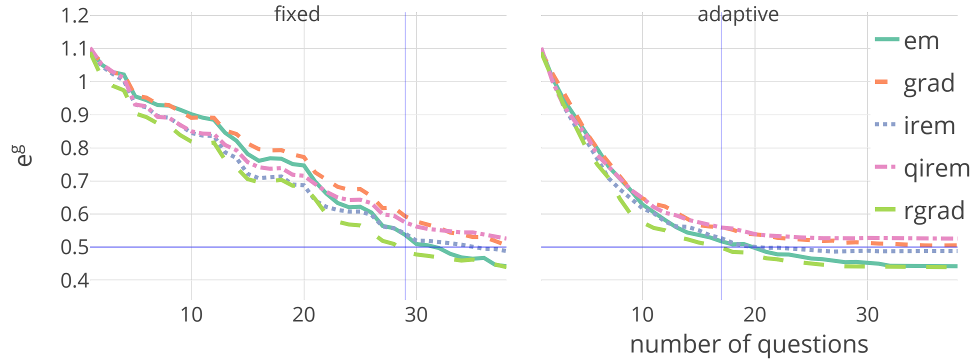


Figure 3: Evolution of the grade prediction error based on skills (B) for models of the learning set size 10, fixed and adaptive question selection.

testing, i.e. aprox. the first half. In the second half it converges to zero as we already know all the answers and thus also the final grade is known.

Special attention should be given to the comparison of the fixed and adaptive approach. Notice the horizontal and vertical lines which mark the threshold of passing the error of 0.5 in all cases. This error threshold is passed in the question 29 and 17 for the fixed and adaptive variants respectively. This observation provides several outcomes. The first one is that using the adaptive version of test significantly reduces the number of questions we have to ask in order to obtain the same level of information. By inspecting the adaptive version of the skills variant further we can see that after asking the first 17 adaptive questions we obtain almost as much information about student skills as possible which gives an option of shortening the test.

5.2 Score and answers prediction

Figure 4 shows the measures of the grade prediction error e^S and the answers prediction accuracy a^Q as they are defined in equations 1 and 2. By inspecting this figure we can see that the restricted gradient method outperforms all other methods in the grade predictions. The only exception where it is slightly worse is the middle part of test for the largest learning set. The highest difference is in the smallest learning set where its benefit is visible the best. In the prediction of answers, restricted gradient method is better in the early stages of testing. For larger learning sets together with other monotone methods (irem and qirem). In the smallest set it is the best of all tested methods. Nevertheless, the best method in the final parts of testing is unrestricted EM. This difference between prediction quality of score and answers is very interesting and it is discussed further in Conclusions section.

Figures 5 and 6 show the evolution of the score prediction as the states of the node O with its confidence interval as it is described in Section 3. Results displayed are obtained from the adaptive test simulation of an individual test with models learned from 10 observations with the rgrad method and the irEM method. The first figure shows the expected value and its confidence interval during the whole test for both methods. We can see that in this particular case both methods shift to the

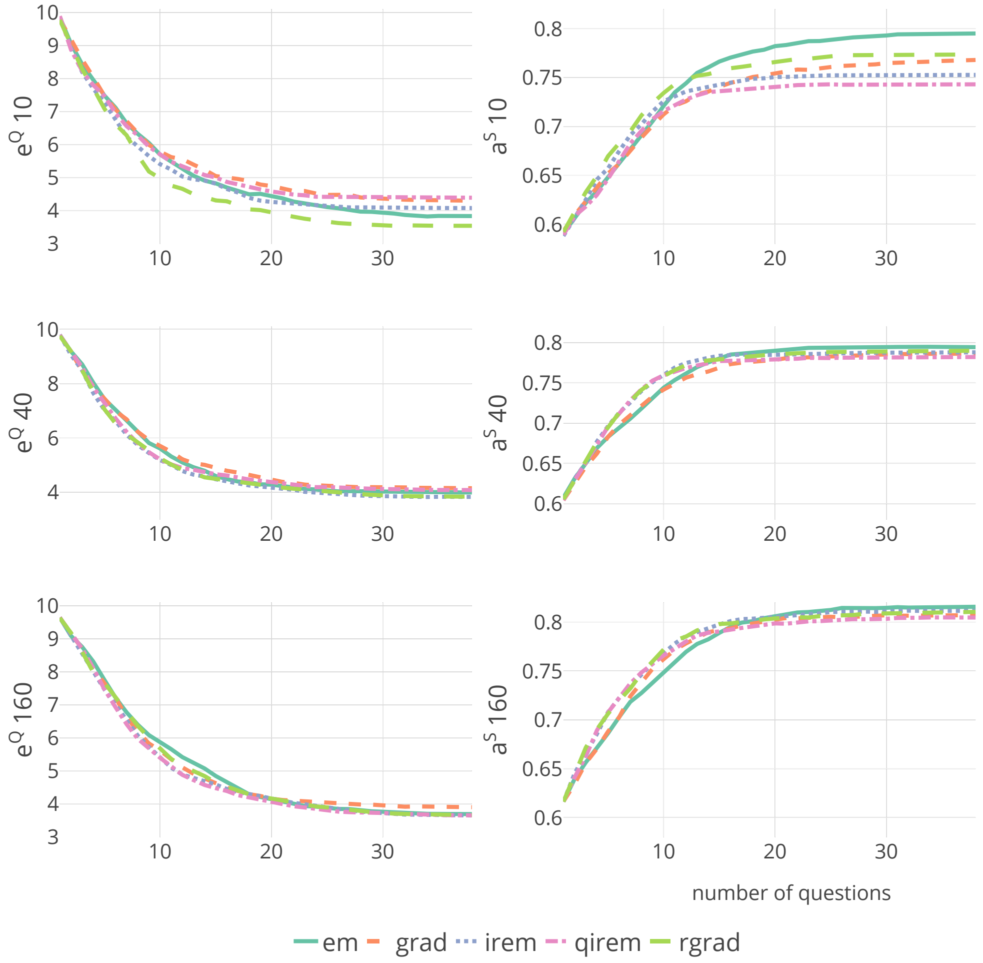


Figure 4: Evolution of the abs. error of the total score prediction e^S and the accuracy of answers prediction a^S for different learning set sizes.

better predictions quite quickly. In this case irEM is faster and at the fifth question its prediction of the total score is better. Nevertheless, irEM stays at the same level for the rest of the test while the rgrad method improves and its final assumption is only approximately one point of the real total score. Another important fact to notice is the shrink of the confidence interval. For the rgrad method it starts at the width of 17 points and it ends at the width of 7 points. This situation is further detailed in Figure 6 where we show the probability distribution at the start and the end of the same test[3].

[3] For the sake of visualization simplicity we display the most probable score instead of the expected score in this case.

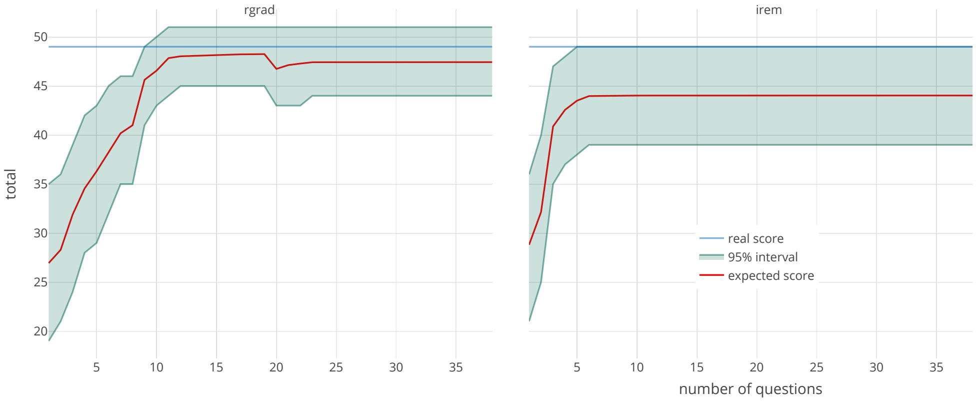


Figure 5: Evolution of the of the total score prediction and its confidence interval for an individual test during the adaptive procedure for the restricted gradient and irem methods, 10 learning samples.

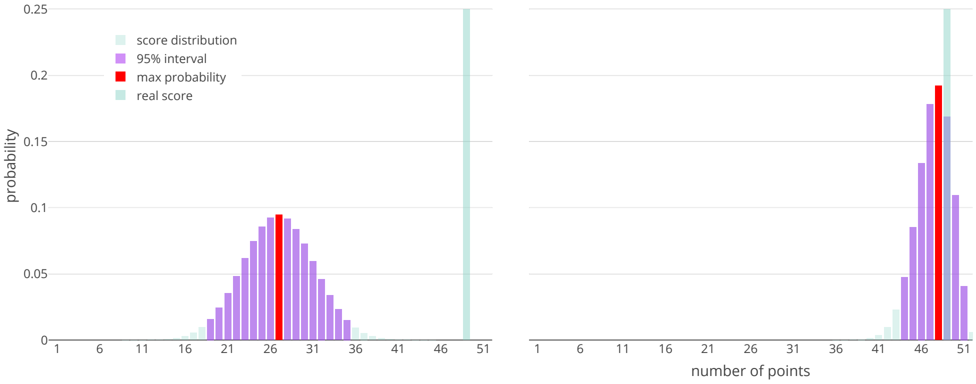


Figure 6: The expected score and its confidence interval for an individual test at the start and the end of testing. Restricted gradient method, 10 learning samples.

6. Conclusions

This article explored the impact of monotonicity restrictions in BN models used to model students in the Czech National Mathematics Exam. It also showed the benefit of adaptivity in testing with this specific data set. In experiments we used monotone and non monotone methods and performed comparisons using different evaluation criteria.

The first observation is the benefit of the adaptive approach to testing. As it can be clearly seen in Figure 3 the number of questions we need to ask is reduced by one third. This creates the space either reducing the length of the test, or using the extra time to increase the precision by asking other questions better tailored for the particular student.

Another new aspect we discussed is the prediction of the total score of a student which is an indicator of his/her skills. We proposed a methodology for measuring the score including the corresponding confidence interval. We compared results of monotone methods and we showed the evolution of the score and the confidence interval during the testing.

Last but not least, we would like to emphasize that monotonicity improves the quality of the grade, score, and question answers predictions. Especially, when the learning set is small and at the first stages of testing. Our empirical results show that the restricted gradient method we propose provides the best results of all tested methods at the first stages of the test. At the later stages of the adaptive test, the regular EM algorithm learning method provided models which were the most precise in terms of individual question answers. This is caused by its flexibility in learning. As EM is not restricted by monotonicity it can learn dependencies monotone methods can not and that allows it to model question answers more precisely in some cases. This result is interesting in the context of the score prediction quality which is an observable indicator of the student skills. When this metric is used for the model evaluation, the EM models were outperformed by the restricted gradient models despite the restricted gradient models prediction of individual answers was worse. The reason is that the monotone models are able to better model the student himself/herself. They are not certain about individual questions but they better infer the score since it is based on their skill model which better characterizes the tested student. This observation means that it is important to keep in mind the purpose of a model while learning it. This is a general observation valid not only for CAT but also for other applications.

Acknowledgments

This work was supported by the Czech Science Foundation (Project No. 19-04579S) and by the Grant Agency of the Czech Technical University in Prague, Grant No. SGS17/198/OHK4/3T/14.

References

R. G. Almond and R. J. Mislevy. Graphical Models and Computerized Adaptive Testing. *Applied Psychological Measurement*, 23(3):223–237, 1999.

E. E. Altendorf, A. C. Restificar, and T. G. Dietterich. Learning from Sparse Data by Exploiting Monotonicity Constraints. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*, 2005.

J. Druzdzel and M. Henrion. Efficient Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 548–553. AAAI Press, 1993.

A. J. Feelders and L. C. van der Gaag. Learning Bayesian Network Parameters with Prior Knowledge about Context-Specific Qualitative Influences. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)*, 2005.

L. C. Gaag and P. de Waal. Multi-dimensional Bayesian Network Classifiers. In *Proceedings of the 3rd European Workshop on Probabilistic Graphical Models, PGM 2006*, pages 107–114, oct 2006.

A. R. Masegosa, A. J. Feelders, and L. van der Gaag. Learning from incomplete data in Bayesian networks with qualitative influences. *International Journal of Approximate Reasoning*, 69:18–34, 2016.

T. D. Nielsen and F. V. Jensen. *Bayesian Networks and Decision Graphs (Information Science and Statistics)*. Springer, 2007.

K. G. Olesen, U. Kjaerulff, F. Jensen, F. V. Jensen, B. Falck, S. Andreassen, and S. K. Andersen. A munin network for the median nerve-a case study on loops. *Applied Artificial Intelligence*, 3 (2-3):385–403, jan 1989.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., dec 1988.

M. Plajner and J. Vomlel. Probabilistic Models for Computerized Adaptive Testing: Experiments. Technical report, ArXiv:, 2016a.

M. Plajner and J. Vomlel. Student Skill Models in Adaptive Testing. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, pages 403–414. JMLR.org, 2016b.

M. Plajner and J. Vomlel. Learning bipartite Bayesian networks under monotonicity restrictions. *International Journal of General Systems*, 49(1):88–111, 2020.

G. Rasch. *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Danmarks Paedagogiske Institut, 1960.

A. C. Restificar and T. G. Dietterich. Exploiting monotonicity via logistic regression in Bayesian network learning. Technical report, Corvallis, OR : Oregon State University, 2013.

P. Savicky and J. Vomlel. Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika*, 43(5):747–764, 2007.

L. van der Gaag, H. L. Bodlaender, and A. J. Feelders. Monotonicity in Bayesian networks. *20th Conference on Uncertainty in Artificial Intelligence (UAI '04)*, pages 569–576, 2004.

W. J. van der Linden and C. A. W. Glas. *Computerized Adaptive Testing: Theory and Practice*, volume 13. Kluwer Academic Publishers, 2000.

M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303, 1990.