JMLR: Workshop and Conference Proceedings vol 52, 403-414, 2016 PGM 2016

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.

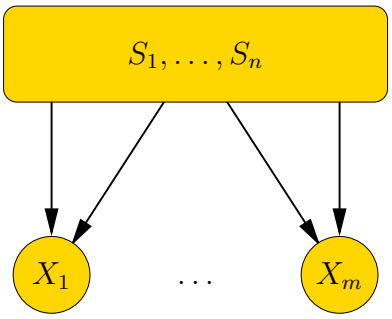
2. Student Skill Models

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.:

$$\begin{aligned} s_i \preceq s_i' \quad &\rightarrow \quad P(X = x|S_i = s_i) \;\leq\; P(X = x|S_i = s_i') \text{ , or} \\ s_i \preceq s_i' \quad &\rightarrow \quad P(X = x|S_i = s_i) \;\geq\; P(X = x|S_i = s_i') \end{aligned}$$

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 $e_{i,j}$

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

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, $f(S_0|e) : \mathrm{R}^m \to \mathrm{R}$, through NN's intermediary hidden layers providing the score value. Reversed structure then provides functions $p_i(X_i|S_0) : \mathrm{R} \to \mathrm{R}$. 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

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

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);

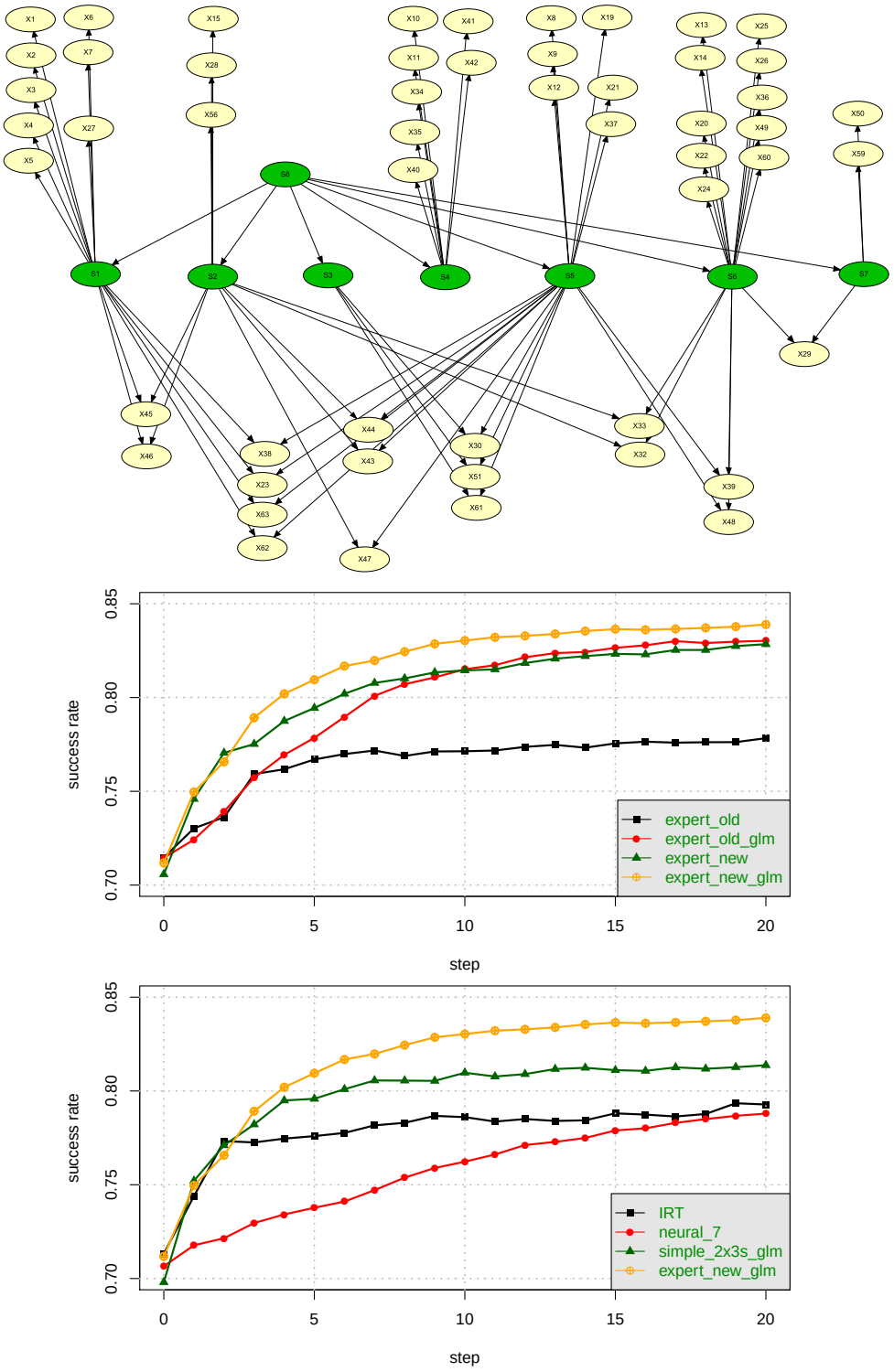


Figure 1: (a) Bayesian network structure (the expert model), (b) Expert Bayesian models success rates, (c) Models comparison success rates

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.