Bayesian Networks in Educational Testing

Jiří Vomlel
http://utia.cas.cz/vomlel/

Abstract

In this paper we discuss applications of Bayesian networks to educational testing. Namely, we deal with the diagnosis of person's skills. We show that when modeling dependence between skills we can get better diagnosis faster. We present results of experiments with basic operations that use fractions. The experiments suggest that the test design can benefit from a Bayesian network that models relations between skills, not only in the case of an adaptive test but also when designing a fixed (non-adaptive) test.

1 Introduction

One task in educational testing is diagnosis of the presence or the absence of person's skills. A possible scenario is that an institution provides a certain course and wishes to test the applicants for gaps in their prerequisites for attending the course. This task is more difficult than giving grades to students or the classification of examinees according to their overall level since the goal is not only to say whether an examinee is good or not but to pinpoint each examinee's abilities and weak points.

Typically, a test designer specifies a set of tested skills, abilities, misconceptions, etc. and a bank of questions, tasks, etc. Let $\mathcal{S} = \{S_1, \ldots, S_k\}$ denote the set of tested skills, abilities, misconceptions, etc. and let $\mathcal{X} = \{X_1, \ldots, X_m\}$ denote the bank of questions, tasks, etc.[1] The designer should also specify which skills are directly related to each question. These relations are often probabilistic, especially if a multiple choice test is used.

One approach used in test design is to construct a test that consists of a fixed sequence of questions covering all tested skills. We will call this type of test a *fixed test*. Another approach aims at constructing an optimal test for each examinee. After each response on a question the system selects next question based on the answers of the previous questions. Since this approach requires computers for the test administration it is often referred to as *computerized adaptive testing* (CAT)[26, 24]. Tests that are automatically tailored to the level of the individual examinees will be referred to as *adaptive tests*.

[1] For simplicity, we will call the elements of set $\mathcal{S}$ skills and the elements of $\mathcal{X}$ questions.

Almond and Mislevy[2] proposed to use graphical models[14] for CAT. Every skill S_i and question X_j are represented by a random variable having a finite sets of values $\mathbb{S}_i$ and $\mathbb{X}_j$, respectively. We will use $\mathbf{S}$ to denote the multidimensional random variable $(S_1, \ldots, S_k)$. Similarly, $\mathbf{X}$ will denote the multidimensional random variable $(X_1, \ldots, X_m)$. The model of Almond and Mislevy consists of one *student model* $P(\mathbf{S})$ and one *evidence model* $P(X_j \mid \mathbf{S}^j)$ for each question $X_j \in \mathcal{X}$. $\mathbf{S}^j$ denotes a possibly multidimensional variable $(S_\ell \in \mathcal{S}^j)$, where $\mathcal{S}^j \subseteq \mathcal{S}$. The underlying assumption is that only skills from set $\mathcal{S}^j$ are directly related to question X_j. In the language of probabilistic models it means that question X_j is independent of skills $S_\ell \in \mathcal{S} \setminus \mathcal{S}^j$ given a state of muldimensional variable $\mathbf{S}^j$. The student model describes relations between skills by use of a joint probability distribution $P(\mathbf{S})$ defined on the variables of the student model[2]. Using the approach of Almond and Mislevy we define the overall probabilistic model of the problem as a Bayesian network

$$P(\mathbf{S}, \mathbf{X}) \quad = \quad P(\mathbf{S}) \cdot \prod_{X_j \in \mathcal{X}} P(X_j \mid \mathbf{S}^j) \ .$$

Bayesian networks[10] are a popular class of models from the family of graphical models[14]. We would like to mention that also several other authors applied Bayesian networks to educational testing and tutoring. Millán et al.[17] uses Bayesian networks to model students in computerized adaptive tests. Mislevy et al.[18] discuss how the numerical parameters of probabilistic models can be estimated. Almond et al.[1] discuss models for conditional probability tables in educational assessment. In paper by Conati et al.[6] an interesting application of Bayesian networks to student modeling for coached problem solving is presented.

In this paper, first, we briefly describe the algorithm we used to construct a student model in Section 2. We discuss criteria and methods for the test construction in Section 3. We present a new algorithm exploiting a Bayesian network model for the construction of a fixed test. In Section 4 we introduce student and evidence models for testing basic operations with fractions. In Section 5 we describe the process of learning models and present results of experiments performed with tests built using the learned models.

2 Building models

We used a probabilistic model $P(\mathbf{S}, \mathbf{X})$ represented by a Bayesian network to model the problem. Assume a collected data $D = \{(\mathbf{x}^1, \mathbf{s}^1), \ldots, (\mathbf{x}^n, \mathbf{s}^n)\}$. Different methods are available for structural learning of Bayesian networks models from collected data. There are two classes of the structural learning algorithms: (1) score-based and (2) constraint-based learning algorithms. Algorithms in

[2]This does not mean that we believe that a tested student behaves stochastically. We use a probability model to characterize a population of students about which we have a prior knowledge, e.g., we know that certain skills are often related to other skills.

both classes perform an informed search through the search space of all possible models guided by a search strategy.

Since we aim at correct prediction of absence or presence of skills $\mathcal{S}$ a score used to evaluate different models could be the conditional log-likelihood CLL of a model P given data D

$$CLL(P \mid D) \quad = \quad \sum_{i=1}^{n} \log P(\mathbf{s}^i \mid \mathbf{x}^i) \ .$$

A fundamental problem with this score is that it does not decompose with the model structure. This means that in order to maximize CLL for a fixed model structure we would have to search over the whole space of model parameters[9]. This makes the method computationally expensive.

Experiments of Cheng and Greiner[5] suggest that learning algorithms based on series of conditional independence test provides Bayesian network classifiers that perform well. We have used a constraint-based learning algorithm - the Hugin PC algorithm[8] - a variant of the original PC algorithm of Spirtes et al.[20] to learn the structure of the student model $P(\mathbf{S})$. This algorithm is based on series of conditional independence tests.

Next, we briefly describe the steps of the PC algorithm. A more detailed description of the Hugin implementation of the PC algorithm can be found in a paper by Jensen et al.[11] The algorithm performs following steps:

- Statistical tests for conditional independence between all pairs of variables are performed.
- An undirected link is added between each pair of variables for which no conditional independence was found.
- Colliders[3] are then identified, ensuring that no directed cycles occur. E.g., if variables A and B are found to be dependent, variables B and C are found to be dependent, but variables A and C are found to be conditionally independent given a set of variables not containing B, then this can be represented by the collider structure $A \rightarrow B \leftarrow C$.
- Links whose direction can be derived from the identified conditional independences and colliders are directed.
- The remaining undirected links are directed randomly, so that no directed cycle occurs.

The PC algorithm produces provably correct structures under the assumptions of infinite data sets, perfect tests, and DAG faithfulness[4]. In the case of limited data sets, however, these algorithms often derive too many conditional

[3] A collider is a pair of links directed such that they meet head-to-head in a node. For example, $A \rightarrow B \leftarrow C$ is a collider.

[4] DAG faithfulness means that the data can be assumed to be simulated from a probability distribution that factorizes according to a DAG.

independence statements. Also, they may in some cases leave out important dependence relations[11]. Thus, the learned structure should be inspected by a domain expert. We combined the Hugin PC algorithm applied to collected data with our expert knowledge of the modeled domain. We discuss our approach in detail in Section 5.

3 Test construction

When a model P of the tested domain is built we can use it to construct tests. Every test $\mathbf{t}$ can be represented by a directed tree. See Figure 1 for a simple example. If the student's answer to the first question (X_2) is correct then the second question is X_3, which is more difficult, otherwise the student gets question X_1, which is easier.

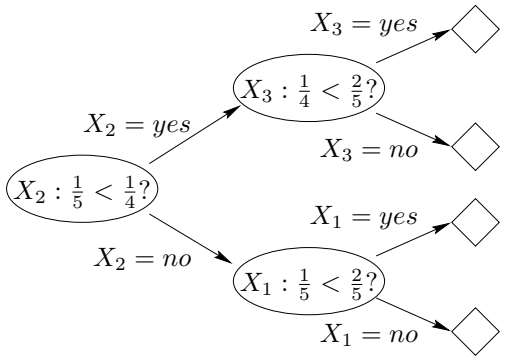

Figure 1: Example of an adaptive test.

Every node corresponds to a question. Every edge $n \rightarrow m$ is labeled by by an answer to the question corresponding to the node n; $outcome(edge)$ is a function that provides the $edge$ labels. The test terminates after a given number of questions is answered or if sufficient information about the tested examinee has been achieved. The set of all terminal nodes of a test $\mathbf{t}$ (leaves of the corresponding tree) will be denoted by $\mathcal{L}(\mathbf{t})$. The root node will be denoted by ϑ. Let $path(n_1, n_k)$ be the sequence of edges constituting the path from node n_1 to node n_k in a test, then

$$\mathbf{e}_n \quad = \quad \bigcup_{edge \in path(\vartheta, n)} outcome(edge)$$

defines the evidence compiled through the performance of questions in $path(\vartheta, n)$. In every node n of test $\mathbf{t}$ we can compute probability $P(\mathbf{e}_n)$ of getting to this node.

Remark 1 Questions can be allowed to be repeated in a test if there are different versions of each question that have the same relation to the tested skills.

Design of a diagnostic test differs according to the goal of the test. Next we discuss different criteria that can be used when designing a test.

3.1 Maximum information

Naturally, every examiner would like to maximize information about the diagnosed examinee at the end of a test. In our context more information corresponds to a decrease in the uncertainty about the presence or the absence of the skills represented by the probability distribution over the skills. It means that we prefer probability distributions that have values close to zero or one. A way to formalize this preference is to aim at a probability distribution minimizing the Shannon entropy. Ben-Bassat[3] used the Shannon entropy and variance to measure the impact of new information.

We will formalize the maximum information approach as a testing process aiming at distributions having minimal value of entropy at the end of the process.

Definition 1 Let $\mathbf{S}$ be a possibly multidimensional random variable with states $\mathbf{s}$ from a finite set $\mathbb{S}$ with probability distribution $P(\mathbf{S})$. Entropy $H(P(\mathbf{S}))$ of $P(\mathbf{S})$ is defined as

$$H\left(P(\mathbf{S})\right) \quad = \quad -\sum_{\mathbf{s}\in\mathbb{S}} P(\mathbf{S}=\mathbf{s}) \cdot \log P(\mathbf{S}=\mathbf{s}) \ .$$

Assume we are interested in all skills of the student model independently, i.e. in marginal probability distributions $P(S_i), i = 1, \ldots, k$.[5] In every node n of a test $\mathbf{t}$ we can compute the total entropy $H(\mathbf{e}_n)$.

Definition 2 Total entropy $H(\mathbf{e}_n)$ in a node n is

$$H(\mathbf{e}_n) \quad = \quad \sum_{S_i\in\mathcal{S}} H(P(S_i \mid \mathbf{e}_n)) \ .$$

We will simplify the notation using function

$$EH(\mathbf{e}_n) \quad = \quad P(\mathbf{e}_n) \cdot H(\mathbf{e}_n) \ .$$

We use the expected value of entropy after a test to define an optimal test.

Definition 3 Let $\mathcal{T}$ be the set of all considered tests. Test $\mathbf{t}^\star$ is optimal iff for all $\mathbf{t} \in \mathcal{T}$

$$\sum_{\ell^\star\in\mathcal{L}(\mathbf{t}^\star)} EH(\mathbf{e}_{\ell^\star}) \quad \leq \quad \sum_{\ell\in\mathcal{L}(\mathbf{t})} EH(\mathbf{e}_\ell) \ .$$

[5] Sometimes the goal of an examiner can be better formalized by use of the joint probability distribution $P(S_1, \ldots, S_k)$ or more dimensional marginals of the joint probability distribution.

In practice it is often impossible to find an optimal test through the evaluation of all considered tests because of the combinatorial explosion. Instead, a greedy heuristic is used to construct a myopically optimal test. A myopically optimal test is a test that consists of questions such that each question minimizes the expected value of entropy after the question is answered.

Definition 4 Let $ch(n)$ denote children of a node n in a test $\mathbf{t}$. A test $\mathbf{t}$ is myopically optimal iff in every nonterminal node n it holds that for all $X \in \mathcal{X}$

$$\sum_{m \in ch(n)} EH(\mathbf{e}_m) \quad \leq \quad \sum_{x \in \mathbb{X}} EH(\mathbf{e}_n \cup \{X = x\}) \ .$$

Other criteria

An alternative to the maximum information criteria is the minimization of *expected decision error*. This criteria is better suited to situations in which the test is used to classify students into certain predefined groups. The criteria is then defined as minimization of probability of misclassification of students at the end of the test.

There can be several *psychometric constraints* on the order of items. For example, if there are related questions then they should be presented together. This restriction leads to the notion of a *testlet*, which is something like a group of related questions. Another restriction can be that certain proportions of questions from different groups are required. Important constraints are those given by *security* issues. They should avoid frequent use of the same questions.

How should a test designer combine several criteria like maximum information, reasonable ordering of questions, security issues, and other constraints? One solution that combines the security issue with a myopically optimal test is that, instead of selecting a most informative question, a probabilistic selection is used with probability of selecting question X being inversely proportional to

$$\sum_{x \in \mathbb{X}} EH(\mathbf{e}_n \cup \{X = x\}) \ .$$

Generally a test designer tries to minimize the number of constraints that are not fulfilled while maximizing the information at the end of a test. These questions are topics of active research[21, 22, 23]. For a review of different test selection strategies see Almond and Madigan[15]. For the criteria often used for item selection in adaptive testing see Meijer and Nering[16].

3.2 Fixed tests

In some situations a fixed test is more suitable than an adaptive test. One of the reasons that a fixed test might be more suitable is that adaptive tests require computers for their application. Computers may not be available or persons taking the tests are expected to have problems with a computer-based test. The design of a fixed test can also benefit from a Bayesian network model. The

same criteria as in the case of adaptive tests can be used. The only difference is the set of considered tests $\mathcal{T}$. While adaptive tests correspond to trees, every fixed test can be represented by a single sequence of questions.

In Table 1 we provide an algorithm for the construction of a fixed test. The

Table 1: Construction of a fixed test.

```
e_list := [∅];
test := [ ];
for i := 1 to |𝒳| do counts[i] := 0;
for position := 1 to test_length do
    new_e_list := [ ];
    for all e ∈ e_list do
        i := most_informative_X(e);
        counts[i] := counts[i] + P(e);
        for all x_i ∈ 𝕏_i do
            append(new_e_list, {e ∪ {X_i = x_i}});
    e_list := new_e_list;
    i* := arg max_i counts[i];
    append(test, X_{i*});
    counts[i*] := 0;
return(test);
```

algorithm performs a greedy search in the graph of the state space corresponding to all possible adaptive tests. It searches for a fixed test, therefore, at each stage it selects only one question.[6] The search is greedy, which means that at each stage the next question is selected using information available looking one step ahead. The question X_i (in the fixed test) maximizes the probability being used at or before the current stage in the myopically optimal adaptive test. This probability can be computed as the total sum of probabilities over all possible states s in the current and all previous stages where the question X_i would be selected, i.e., where

$$most\_informative\_X(\mathbf{e}_s) \quad = \quad i \ ,$$

where $\mathbf{e}_s$ denotes the evidence corresponding to the node of state s in the state space graph. The search continues only in the subspace corresponding to the selected question. We will use an example to better explain how the algorithm works.

Example 1 Figure 2 depicts the state space of all possible tests of the length two. Rectangular nodes correspond to states where a decision on selection of

[6] Every stage in the state space graph corresponds to a position in the test.

the next question is made. Ovals correspond to states where the examinee's answer to a given question is observed. Edges correspond to answers. Assume that question X_2 was selected as the first question. In Figure 2 the selection of the second question is shown. □

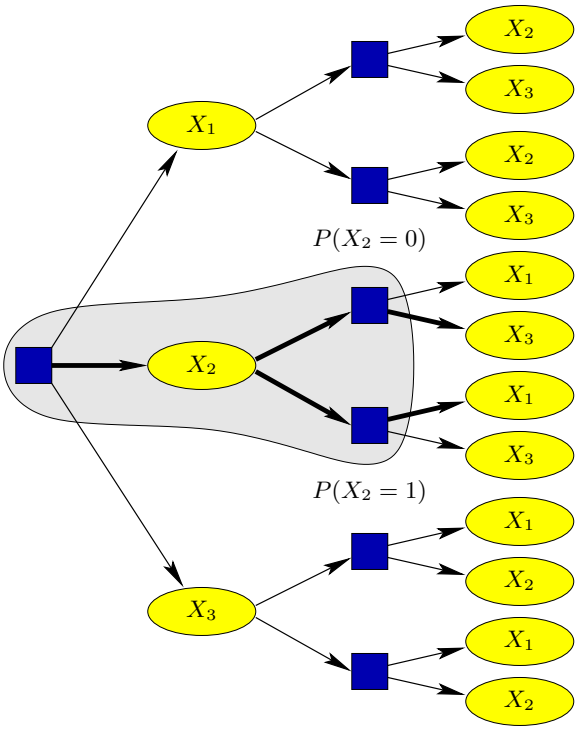

Figure 2: Selection of the second question in the algorithm from Table 1.

The evidence list $e\_list = \{\{X_2 = 0\}, \{X_2 = 1\}\}$. The most informative question for evidence $\{X_2 = 0\}$ is X_3:

$$most\_informative\_X(\{X_2 = 0\}) \quad = \quad 3$$

therefore X_3 gets a count corresponding to $P(X_2 = 0)$, i.e. $counts[3] = P(X_2 = 0)$. Since

$$most\_informative\_X(\{X_2 = 1\}) \quad = \quad 1$$

X_1 gets a count corresponding to $P(X_2 = 1)$, i.e. $counts[1] = P(X_2 = 1)$. A question that has the maximal total count is selected as the second question.

4 Model for basic operations with fractions

4.1 Student model

The learning process that resulted in a student model consisted of several steps. First, a group of students from Aalborg University prepared paper tests that were given to students at Brønderslev High School. Four elementary skills, four operational skills, and abilities to apply operational skills to complex tasks were tested. See Table 2 for elementary and operational skills[7]. 149 students solved the test. The university students analyzed the tests and summarized the results[4]. During this phase, seven types of misconception were discovered. See Table 3 for misconceptions observed in Brønderslev High School.

Table 2: Elementary and operational skills.

Label	Description	Example
CP	Comparison (common numerator or denominator)	$\frac{1}{2} > \frac{1}{3}, \quad \frac{2}{3} > \frac{1}{3}$
AD	Addition (common denominator)	$\frac{1}{7} + \frac{2}{7} = \frac{1+2}{7} = \frac{3}{7}$
SB	Subtraction (common denominator)	$\frac{2}{5} - \frac{1}{5} = \frac{2-1}{5} = \frac{1}{5}$
MT	Multiplication	$\frac{1}{2} \cdot \frac{3}{5} = \frac{3}{10}$
CD	Finding common denominator	$\left(\frac{1}{2}, \frac{2}{3}\right) = \left(\frac{3}{6}, \frac{4}{6}\right)$
CL	Canceling out	$\frac{4}{6} = \frac{2 \cdot 2}{2 \cdot 3} = \frac{2}{3}$
CIM	Conversion to mixed numbers	$\frac{7}{2} = \frac{3 \cdot 2+1}{2} = 3\frac{1}{2}$
CMI	Conversion to improper fractions	$3\frac{1}{2} = \frac{3 \cdot 2+1}{2} = \frac{7}{2}$

An example of a student model is given in Figure 3. We will discuss the model selection process in Section 5.

4.2 Evidence models

For each task or question an evidence model is created. Examples of tasks are

$$
\begin{aligned}
T_1 \quad & \left(\tfrac{3}{4} \cdot \tfrac{5}{6}\right) - \tfrac{1}{8} = \tfrac{15}{24} - \tfrac{1}{8} = \tfrac{5}{8} - \tfrac{1}{8} = \tfrac{4}{8} = \tfrac{1}{2} \\
T_2 \quad & \tfrac{1}{3} - \tfrac{1}{12} = \tfrac{4}{12} - \tfrac{1}{12} = \tfrac{3}{12} = \tfrac{1}{4} \\
T_3 \quad & \tfrac{1}{4} \cdot 1\tfrac{1}{2} = \tfrac{1}{4} \cdot \tfrac{3}{2} = \tfrac{3}{8} \\
T_4 \quad & \left(\tfrac{1}{2} \cdot \tfrac{1}{2}\right) \cdot \left(\tfrac{1}{3} + \tfrac{1}{3}\right) = \tfrac{1}{4} \cdot \tfrac{2}{3} = \tfrac{2}{12} = \tfrac{1}{6} \ .
\end{aligned}
$$

[7] For each operational skill there is a corresponding application skill labeled with the prefix "A" in the student model. Application skills are not listed in Table 2.

Table 3: Misconceptions.

Label	Description	Occurrence
MAD	$\frac{a}{b} + \frac{c}{d} = \frac{a+c}{b+d}$	14.8%
MSB	$\frac{a}{b} - \frac{c}{d} = \frac{a-c}{b-d}$	9.4%
MMT1	$\frac{a}{b} \cdot \frac{c}{b} = \frac{a \cdot c}{b}$	14.1%
MMT2	$\frac{a}{b} \cdot \frac{c}{b} = \frac{a+c}{b \cdot b}$	8.1%
MMT3	$\frac{a}{b} \cdot \frac{c}{d} = \frac{a \cdot d}{b \cdot c}$	15.4%
MMT4	$\frac{a}{b} \cdot \frac{c}{d} = \frac{a \cdot c}{b+d}$	8.1%
MC	$a\frac{b}{c} = \frac{a \cdot b}{c}$	4.0%

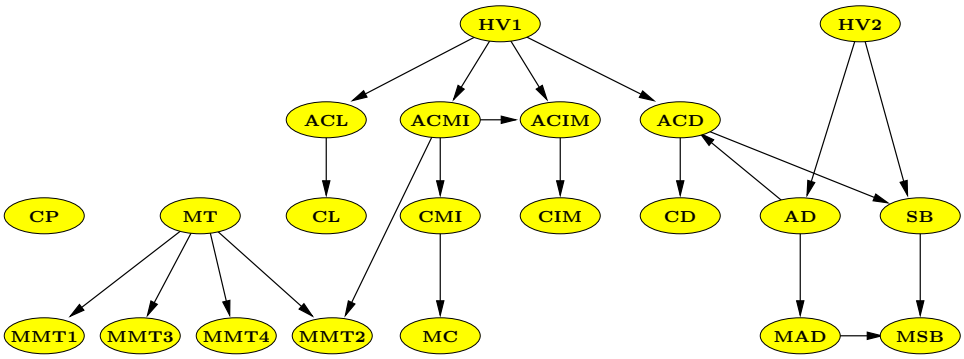

Figure 3: Student model describing relations between skills and misconceptions.

First, assume that a student is able to solve certain tasks if and only if she has the necessary skills and does not have certain misconceptions (later we will relax this assumption). Thus, we can formally describe the tasks T_1,T_2,T_3,and T_4 by logical formulas:

$$
\begin{aligned}
T_1 &\Leftrightarrow MT \,\&\, CL \,\&\, ACL \,\&\, SB \,\&\, \neg MMT3 \,\&\, \neg MMT4 \,\&\, \neg MSB \\
T_2 &\Leftrightarrow SB \,\&\, CL \,\&\, ACL \,\&\, CD \,\&\, ACD \,\&\, \neg MSB \\
T_3 &\Leftrightarrow MT \,\&\, CMI \,\&\, ACMI \,\&\, \neg MMT3 \,\&\, \neg MMT4 \,\&\, \neg MC \\
T_4 &\Leftrightarrow MT\&AD\&CL\&ACL\&\neg MMT1\&\neg MMT2\&\neg MMT3\&\neg MMT4\&\neg MAD
\end{aligned}
$$

Of course, the assumption of deterministic relations between skills and the actual outcome of a task is unrealistic. A student can make a mistake even if she has all abilities needed to solve a given task. On the other hand, a correct answer does not necessarily mean that the student has all abilities since she may guess the

right answer, e.g., in a test where she is to select one answer from a given set of answers. However, it turned out to be reasonable to understand a task variable T_i as the ability to solve the corresponding task and to use a non-deterministic model only for the description of the dependence between the skill T_i and the actual outcome of the corresponding task X_i. Thus we can model "guessing" using conditional probability $P(X_i \mid \neg T_i)$ and "mistakes" using $P(\neg X_i \mid T_i)$. See Figure 4.

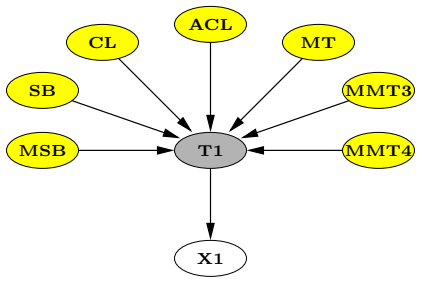

Figure 4: Evidence model of task T_1.

Conditional probability tables representing relations between variables in the student and evidence models can often be simplified, for example, using models of independence of causal influence. Examples of such models are Noisy-OR or Noisy-AND.

5 Experiments

5.1 Building models

We searched model structures using the PC-algorithm of Spirtes et al.[20], implemented in Hugin[8]. At the beginning we ran the PC-algorithm without any constraints on edges. It provided an initial insight into the relations between skills and misconceptions. Using our "expert knowledge" of the domain of fractions we explained some relations with the aid of hidden variables and introduced certain constraints on edges.

The PC-algorithm is based on series of conditional independence tests. An important parameter of these tests is the significance level used in the independence tests. See Figure 5 from which one can see that significance level 5% also appeared to be optimal when learned models were compared using the Bayesian Information Criteria (BIC), a criteria derived by Schwarz[19].

Applying different constraints on the resulting model we learned, using the PC-algorithm, nine different model structures, most of them containing hidden variables introduced by "domain experts", see Table 4.

In most of the models certain edges were required to be present. We calibrated the overall models (composed from one student model and twenty evi-

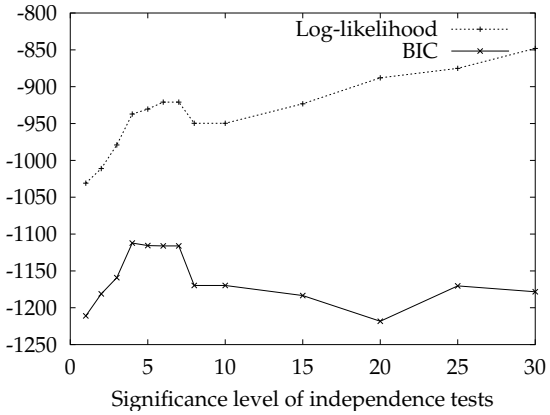

Figure 5: Comparison of models learned by the PC algorithm in Hugin with different significance levels.

dence models) on the collected data. Parameter learning using the EM-algorithm was started from a non-uniform distribution to guide the EM-algorithm[13] to reasonable parameters for hidden variables, the initial experience counts were very low (0.00001) to adjust the learned probability tables to the data. Evidence models were designed as described in Section 4.2; only the conditional probability distributions $P(X_i \mid T_i)$ were estimated from the collected data.

5.2 Comparison of models according to their test performance

We used the paper-test results to test the models. Each data vector in collected data corresponds to the responses of one student. The testing procedure was *leave-one-out cross-validation*. First, we selected 148 of 149 data vectors from the collected data. We learned each model on 148 selected data vectors. Then we tested the model on the remaining data vector. For each model we repeated the procedure 149 times so that each data vector was used for testing once. The reported results are average over all 149 experiments.

Since each question was present only once in the test we could not allow questions to be repeated. Thus all experiments consisted of only twenty questions. We measured how well the models predicted the students' skills. For all nineteen skills we tested whether the most probable state (given observed answers) of each skill is equal to the true state observed in the data. The skill predictive accuracy is the percentage of skills whose predicted state equals the observed state. It was possible to compare the predictions with the true skill values since they were available due to a detail manual analysis of student responses in the paper tests. In Table 5 we compare models' skill predictive accuracy after the 4th, 9th, 14th, and 19th questions in an adaptive test. The

Table 4: Tested models.

(a)	model containing two hidden variables with several restrictions on presence or absence of edges,
(b)	model with two hidden variables and few restrictions, the model is given in Figure 3,
(c)	model with one hidden variable and few restrictions,
(d)	model with no hidden variable and few restrictions,
(e)	model with no hidden variable and only obvious logical constraints as restrictions,
(f)	model with no hidden variable and independent skills,
(g)	Naïve Bayes model with one hidden variable (with two states) being parent of all skills,
(h)	as above, but the hidden variable has three states,
(i)	model without skills - all questions were children of one hidden variable with six states.

confidence intervals correspond to a confidence level 0.95. Model (i) cannot be tested on skill prediction since it does not contain skills in the model.

Table 5: Skill predictive accuracy.

model	4th	9th	14th	19th
(a)	84.52 ± 1.90	88.87 ± 1.61	89.39 ± 1.60	89.42 ± 1.58
(b)	87.93 ± 1.61	90.66 ± 1.42	91.46 ± 1.25	91.44 ± 1.26
(c)	88.51 ± 1.56	90.31 ± 1.41	90.72 ± 1.36	90.89 ± 1.34
(d)	87.76 ± 1.68	90.32 ± 1.41	90.39 ± 1.36	90.43 ± 1.36
(e)	80.79 ± 2.28	88.69 ± 1.35	89.27 ± 1.31	89.47 ± 1.27
(f)	80.24 ± 2.04	87.07 ± 1.58	87.61 ± 1.50	87.71 ± 1.49
(g)	88.32 ± 1.40	89.78 ± 1.19	90.03 ± 1.19	90.07 ± 1.18
(h)	89.10 ± 1.54	90.36 ± 1.26	90.74 ± 1.28	90.82 ± 1.25

We performed cross-validated paired t tests[12, 7] to establish the statistical significance of the difference between performance of models. We compared the skill predictive accuracy after 9^{th} question[8]. This test makes two assumptions: (1) difference of the performance is normally distributed and (2) difference of the performance on different test sets is independent. Since these assumptions may be violated the tests may incorrectly detect difference when no difference

[8]We intend to use the models in tests that would typically consist of no more than ten questions.

exists. However, this error was empirically shown to be small[7].

In Table 6 we present results of pairwise comparisons of models. For two-sided test with 148 degrees of freedom the t value corresponding to a confidence level 0.95 is 1.97601. Thus, for all values exceeding this value the hypothesis that models are equal can be rejected. One can see that difference between some of the models is not very large. We conjecture that it is because too few cases were available for learning, so that the more complex models could not be calibrated well. For subsequent tests we selected model (b). Its structure was presented in Figure 3.

Table 6: Comparison of models' performance using t values. In the last column we present number of significant wins minus number of significant losses.

	vs. (b)	vs. (c)	vs. (d)	vs. (e)	vs. (f)	vs. (g)	vs. (h)	total
(a)	**−3.293**	**−2.650**	**−2.567**	0.308	**2.323**	−1.318	**−2.192**	**−3**
(b)		0.983	0.578	**3.579**	**5.296**	1.734	0.759	**+3**
(c)			−0.265	**2.612**	**4.488**	1.154	−0.004	**+3**
(d)				**2.839**	**4.487**	1.327	0.091	**+3**
(e)					**3.837**	**−1.977**	**−2.739**	**−4**
(f)						**−4.071**	**−4.728**	**−7**
(g)							−1.641	**+2**
(h)								**+3**

To see the importance of the calibration we also tested models that were not well calibrated by giving our original model experience count 50. We observed significantly worse predictions.

In Table 7 we present comparisons of question predictive accuracy. In order to provide a stable criteria for comparing the answer predictions, the presented number is the average computed from the predictions for all questions. This means that the questions that were answered are also included as if they could be asked again. Therefore the question predictive accuracy converges to 100% when all questions are answered. The differences between models are even smaller there.

5.3 Comparison of test design methods

We used model (b) to compare four *test design methods*:

- a test where questions are in the same ascending order, i.e., as they were ordered in the paper tests given to the students,
- a fixed test where questions are taken in the reverse (descending) order,
- the myopically optimal adaptive test, and

Table 7: Question predictive accuracy.

model	4th	9th	14th	19th
(a)	83.26 ± 1.33	91.34 ± 1.08	95.50 ± 0.81	99.53 ± 0.23
(b)	82.95 ± 1.37	90.91 ± 1.05	94.97 ± 0.84	99.36 ± 0.26
(c)	84.53 ± 1.41	90.40 ± 1.05	94.97 ± 0.83	99.43 ± 0.26
(d)	83.36 ± 1.28	90.71 ± 1.06	94.93 ± 0.83	99.33 ± 0.27
(e)	81.07 ± 1.40	90.47 ± 1.13	95.03 ± 0.86	99.43 ± 0.25
(f)	81.68 ± 1.30	90.07 ± 1.09	94.93 ± 0.85	99.40 ± 0.26
(g)	86.75 ± 1.30	91.71 ± 1.03	95.50 ± 0.80	99.46 ± 0.25
(h)	85.97 ± 1.34	91.48 ± 1.04	95.03 ± 0.83	99.43 ± 0.26
(i)	80.81 ± 1.75	89.13 ± 1.22	94.46 ± 0.81	98.93 ± 0.33

- the average fixed test constructed by the method described in Section 3.2.

In Figure 6 we can see how the skill predictive accuracy for different selection methods evolves with more questions being answered.The adaptive test provides best results. The average fixed test provides nearly as good results as the adaptive test. The two fixed tests are substantially worse, especially the one where questions are taken in the same order as they were ordered in the paper tests given to the students. Note that in the adaptive test we needed only six questions to reach predictive accuracy for which the fixed (with questions in the ascending order) test required sixteen questions.

The criterion used to select the next question in the adaptive test was the total expected entropy on skills. In Figure 7 we present how the actual total entropy on skills evolved.Observe that the skill predictive accuracy and the total expected entropy are approximately inversely proportional - the lower the entropy, the better the predictions. A bit surprising is that entropy does not decrease monotonically, especially in the adaptive test. An explanation can be that in the testing data there are certain questions that are answered surprisingly when compared to the answers to other questions. This may increase the uncertainty in the model.

To complete the comparisons we also present a plot showing how the quality of prediction of answers evolved for different selection methods in Figure 8.Our primary goal was not the question predictive accuracy but it is natural that with better knowledge about student's skills, the model better predicts the answers.

6 Conclusions

We provided empirical evidence showing that educational testing can benefit from the application of Bayesian networks. We showed that adaptive tests may substantially reduce the number of questions that are necessary to be asked. We proposed a new method for the design of a fixed test, which provided good results

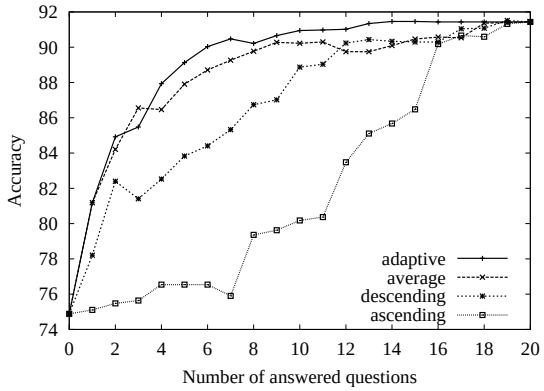

Figure 6: Skill predictive accuracy.

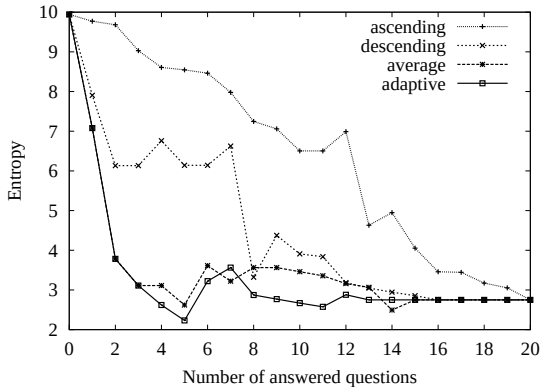

Figure 7: Total entropy of probability of skills.

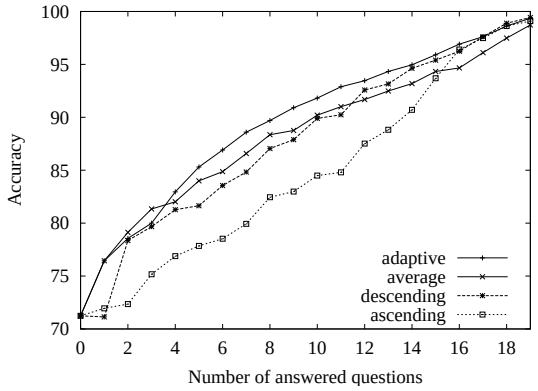

Figure 8: Question predictive accuracy.

on tested data. It may be regarded as a good inexpensive alternative to the computerized adaptive tests when they are not suitable. There are theoretical problems related to the application of Bayesian networks to educational testing that were not studied in this paper. One of them is efficient inference - a problem addressed in our paper[25] presented at the 18^{th} Conference on Uncertainty in AI.

Acknowledgments

This paper is mainly based on results achieved when I was with the Department of Computer Science at Aalborg University, Denmark. I am grateful to Finn V. Jensen and the Decision Support Systems group at Aalborg University for the inspiring environment, Russell G. Almond and Robert J. Mislevy from Educational Testing Service (ETS) for interesting discussions during my visit at ETS, Frank Jensen and Anders L. Madsen for their assistance with Hugin, and Kirsten Bangsø Jensen for organizing the tests in Brønderselev High School. I was supported by the Grant Agency of the Czech Republic through grant number GA ČR 201/01/1482.

References

[1] Russell. G. Almond, Lou Dibello, Frank Jenkins, Deniz Senturk, Robert J. Mislevy, Linda S. Steinberg, and Duanli Yan. Models for conditional probability tables in educational assessment. In *Proc. of the 2001 Conference on AI and Statistics*. Society for AI and Statistics, 2001.

[2] Russell G. Almond and Robert J. Mislevy. Graphical models and computerized adaptive testing. *Applied Psychological Measurement*, 23(3):223–237, 1999.

[3] Moshe Ben-Bassat. Myopic policies in sequential classification. *IEEE Transactions on Computers*, 27(2):170–174, 1978.

[4] Linas Būtėnas, Agnė Brilingaitė, Alminas Čivilis, Xuepeng Yin, and Nora Zokaitė. Computerized adaptive test based on Bayesian network for basic operations with fractions. Student project report, Aalborg University, 2001. `http://www.cs.auc.dk/library`.

[5] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In Kathryn Blackmond Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 101–108. Morgan Kaufmann Publishers, 1999.

[6] Cristina Conati, Abigail S. Gertner, Kurt VanLehn, and Marek J. Druzdzel. On-line student modeling for coached problem solving using Bayesian networks. In Anthony Jameson, Cecile Paris, and Carlo Tasso, editors, *Proc. of the Sixth Int. Conf. on User Modeling (UM97), Chia Laguna, Sardinia, Italy*, pages 231–242. Springer Verlag, 1997.

[7] Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computing*, 10(7):1895–1924, 1998.

[8] Hugin Explorer. ver. 6.0. Computer software, 2002. `http://www.hugin.com`.

[9] N. Friedman, D. Geiger, and M. Goldszmitdt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[10] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer Verlag, New York, 2001.

[11] Frank Jensen, Uffe B. Kjærulff, Michael Lang, and Anders L. Madsen. Hugin - the tool for Bayesian networks and Influence diagrams. In J. A. Gámez and A. Salmeron, editors, *Proceedings of the First European Workshop on Probabilistic Graphical Models, PGM 2002*, pages 211–221, 6-8 November 2002.

[12] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. S. Mellish, editor, *Proceedings of the International Joint Confrence on Artificial Intelligence (IJCAI 95)*, page C. S. Mellish. Morgan Kaufmann Publishers, 1995.

[13] Steffen L. Lauritzen. The EM-algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 1:191–201, 1995.

[14] Steffen L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[15] David Madigan and Russell G. Almond. On test selection strategies for belief networks. In D. D. Fisher and H. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics*, volume V, pages 89–98. Springer Verlag, 1996.

[16] Rob R. Meijer and Michael L. Nering. Computerized adaptive testing: Overview and introduction. *Applied Psychological Measurement*, 23(3):187–194, 1999. Special Issue: Computerized Adaptive Testing.

[17] Eva Millán and José Luis Pérez-de-la-Cruz. A Bayesian diagnostic algorithm for student modeling and its evaluation. *User modeling and User-Adapted Interaction*, 12(2–3):281–330, 2002.

[18] Robert J. Mislevy, Russell G. Almond, Duanli Yan, and Linda Steinberg. Bayes nets in educational assessment: Where do the numbers come from? In Kathryn B. Laskey and Henri Prade, editors, *Proc. of the Fifteenth Conf. on Uncertainty in AI*, pages 437–446, San Francisco, 1999. Morgan Kaufmann Publishers, Inc.

[19] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 7(2):461–464, 1978.

[20] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Number 81 in Lecture Notes in Statistics. Springer Verlag, 1993.

[21] Martha L. Stocking and Charles Lewis. Controlling item exposure conditional on ability in computerized adaptive testing. *Journal of Educational and Behavioral Statistics*, 23:57–75, 1998.

[22] Martha L. Stocking and Len Swanson. A method for severly constrained item selection in adaptive testing. *Applied Psychological Measurement*, 17:277–292, 1993.

[23] Len Swanson and Martha L. Stocking. A model and heuristic for solving very large item selection problems. *Applied Psychological Measurement*, 17:151–166, 1993.

[24] Wim J. van der Linden and Cees A. W. Glas, editors. *Computerized Adaptive Testing: Theory and Practice*. Kluwer Academic Publishers, 2000.

[25] Jiří Vomlel. Exploiting functional dependence in Bayesian network inference. In *Proc. of the* 18^{th} *Conf. on Uncertainty in AI*, pages 528–535. Morgan Kaufmann Publishers, 2002.

[26] Howard Wainer, David Thissen, and Robert J. Mislevy. *Computerized Adaptive Testing: A Primer*. Mahwah, N.J., Lawrence Erlbaum Associates, second edition, 2000.