

# Bayesian Student Models Based on Item to Item Knowledge Structures

Michel C. Desmarais and Michel Gagnon

École Polytechnique de Montréal

**Abstract.** Bayesian networks are commonly used in cognitive student modeling and assessment. They typically represent the item-concepts relationships, where items are observable responses to questions or exercises and concepts represent latent traits and skills. Bayesian networks can also represent concepts-concepts and concepts-misconceptions relationships. We explore their use for modeling item-item relationships, in accordance with the theory of *knowledge spaces*. We compare two Bayesian frameworks for that purpose, a standard Bayesian network approach and a more constrained framework that relies on a local independence assumption. Their performance is compared over their respective ability to predict item outcome and through simulations over two data sets. The simulation results show that both approaches can effectively perform accurate predictions, but the constrained approach shows higher predictive power than a Bayesian Network. We discuss the applications of item to item structure for cognitive modeling within different contexts.

## 1 Introduction

There is considerable interest in the use of Bayesian networks (BN) for student modeling and cognitive assessment (for eg., Mislevy, Almond, Yan, & Steinberg, 1999; Conati, Gertner, & VanLehn, 2002; VanLehn, Niu, Siler, & Gertner, 1998; Vomlel, 2004). In part, this interest stems from the ability of a BN to model the uncertainty that is inherent to cognitive modeling. Moreover, BN lend themselves to automatic learning and offer an attractive alternative to the difficult and error prone effort to parametrize a student model by means of human expertise. Finally, this interest can also be attributed to the fact that the field of BN has progressed at a rapid and sustained pace in the last two decades. It has now matured to a level where commercial and open source software packages and libraries allow their use in a relatively simple manner.

The studies on the use of BN in student models have, to our knowledge, exclusively dealt with modeling relationships from test item responses to concepts and misconceptions, and between concepts and misconceptions themselves. However, items also have an internal structure according to the theory of *knowledge spaces* (Doignon & Falmagne, 1999). Doignon and Falmagne (1999) have shown that the structure among items is determined by the order that constrains their learning in time. This order allows the inference of *mastered* (or *non mastered*) items, thereby allowing the assessment of an individual's state of knowledge

from partial evidence, akin to the process of evidence propagation in a Bayesian network.

We describe two approaches to building item to item structures and how to use them for cognitive diagnostic. The first approach incorporates a specific method of inducing the structure of item to item structures. It is combined with an inference model based on simple posterior probability model under the assumption of local independence. The second approach is a generic approach of inducing the structure of a BN and performing inference within such structures. First, we describe theoretical underpinning of item to item structures. It is followed by the description of each approach and by a performance comparison to assess their respective ability to predict item outcome.

## 2 Item to Item Node Structures and the theory of Knowledge Spaces

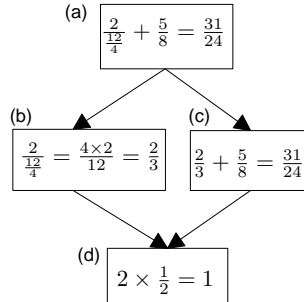
Item to item structures depart from the more common way of building student models, which is based on a hierarchy of concepts with items at the bottom of the hierarchy (see, for eg. Millán, de-la Cruz, & Suárez, 2000; Vomlel, 2004; VanLehn et al., 1998). In contrast to hierarchical structures, item to item structures build structures among observable knowledge item themselves (Falmagne, Koppen, Villano, Doignon, & Johannesen, 1990; Doignon & Falmagne, 1999), bypassing concept links. A number of researchers have worked on the problem of building student models within this framework (Dowling & Hockemeyer, 2001; Kambouri, Koppen, Villano, & Falmagne, 1994). Our own work on Partial Order Knowledge Structures (POKS) (Desmarais, Maluf, & Liu, 1996; Desmarais & Pu, 2005) falls under this line of research as well.

The theory of knowledge spaces asserts that *knowledge items*, i.e. observable elements such as question items, are mastered in a constrained order. Knowledge items define an individual's *knowledge state* as a subset of items that are mastered by that individual. The knowledge space determines which other knowledge state the person can move to. Viewed differently, the knowledge space defines the structure of prerequisites among knowledge items. For example, we learn to solve figure 1's problems in an order that complies with the inverse of the arrow directions. It follows from this structure that if one masters knowledge item (c), it is likely she will also master item (d). Conversely, if she fails item (c), she will likely fail item (a). However, item (c) does not significantly inform us about item (b). This structure defines the following possible knowledge states (subsets of the set  $\{a, b, c, d\}$ ):

$$\{\emptyset, \{d\}, \{c, d\}, \{b, d\}, \{b, c, d\}, \{a, b, c, d\}\}$$

Other knowledge states are deemed impossible (or *unlikely* in a probabilistic framework).

Formally, it can be shown that a directed acyclic graph (DAG), such as the one in figure 1, can represent a knowledge space closed under union and



**Fig. 1.** A simple knowledge space composed of 4 items ( $\{a, b, c, d\}$ ). The partial order constrains possible knowledge states to  $\{\emptyset, \{d\}, \{b, d\}, \{c, d\}, \{b, c, d\}, \{d, b, c, a\}\}$ . In the Knowledge space Theory, the arrows represent *surmise* relations ( $\succ$ ) that determine the order in which we acquire the knowledge items, represented by the graph nodes.

intersection. Closure under union and intersection implies that if we combine two people’s knowledge state, it is a valid subset of the knowledge space, and so is the intersection of their two states. We refer to such structures as *partial order knowledge structures*, or POKS. They represent a variant of the general theory of knowledge space which assumes closure only under union. The assumption of closure under union and intersection allows a considerable reduction the space of knowledge states. It greatly simplifies the algorithms for inducing a knowledge structure from data and reduces the amount of data cases required.

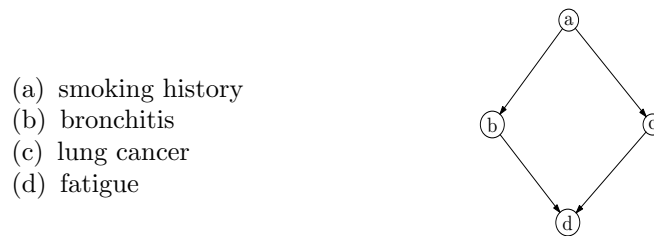
It can be seen that the theory of knowledge spaces and its POKS derivative make no attempt to structure knowledge in a hierarchy of concepts or any other structure containing latent variables (often called *latent traits*). The knowledge state of an individual is solely defined in terms of observable manifestations of skills such as test question. Of course, that does not preclude the possibility to re-structure knowledge items into higher level concepts and skills. In fact, this precisely is what a teacher does for developing a quiz or an exam, for example.

## 2.1 Partial Order Knowledge Structures and Bayesian Networks

Although POKS graphs like the one in figure 1 can be conveniently represented graphically by a DAG that resembles to a BN, the semantics of links is different. BN directed links usually represent causal relationships (although they can represent any kind of probabilistic relationship) and the structure explicitly represents conditional independence between variables. A knowledge space link is similar to a logical implication relation, but it represents a prerequisite, or, to use Doignon and Falmagne terminology, a *surmise* relation. For example, if we

have a surmise relation  $A \succ B$ , it implies that the mastery of  $B$  will precede the mastery of  $A$ , and thus if a student has a success at for  $A$ , that student is likely to have a success for  $B$ . Moreover, its structure represents a partial ordering of the order in which items are likely to be learned.

That difference in the semantics of links has a number of implications. For one, the closures under *union* and *intersection* of POKS implies that, given a relation  $A \rightarrow B$ , or in the poks framework  $A \succ B$ , the absolute frequency of people who master a knowledge item  $A$  will necessarily be smaller or equal to the frequency of  $B$ . This conclusion does not hold for the case of general BN. For example, assume figure 1's structure is the following (a BN taken from Neapolitan, 2004):



It is clear that *smoking history* (a) can be a much more frequent state than *lung cancer* (c) and *bronchitis* (b). This could not happen in POKS. The frequency of (a) cannot be higher than that of (b) and (c). It is also obvious that, whereas the occurrence *lung cancer* could decrease the probability of *bronchitis* by *discounting* (or “*explaining away*”) that later cause as a plausible explanation for *fatigue*, *discounting* does not play a role in the case of knowledge structures (eg. observing node (c) in Figure 1 would not decrease the probability of (b)).

In short, many interactions found in general BN do not occur in POKS. We conjecture that this reduction in the space of possibilities that characterizes POKS, namely the closure under *union* and *intersection*, can be used to infer knowledge structures with algorithms that rely upon stronger assumptions and more constrained search spaces than for the more general BN models.

In fact, structural induction techniques tailored to the Knowledge structures and the POKS frameworks have been devised by a number of researchers. For example, Kambouri et al. (1994) introduced a semi-automated algorithm to construct knowledge structures. They developed an application that combines the use of empirical and an interactive question-answer process with domain experts to successfully construct knowledge structures. Their approach allows the construction of knowledge structures closed under union only, which implies it can represent alternative prerequisites. However, the construction process requires human intervention and cannot be considered as automated learning.

In the current study, we focus on the construction of item to item structures solely from learning approaches. The next section describes a BN learning approach over item to item structures, and a constraint-based structural learning approach to induce POKS.

### 3 Induction and Inference with Partial Order Knowledge Structures

The topology of an item to item structure can be fairly intertwined and complex. Inducing that structure is a difficult task to perform manually. It entails determining the order of mastery among knowledge items. If the set of knowledge items is large, over a few tens of items for example, our own experience is that this task can be very tedious and error prone.

Thus, finding means of learning the item to item knowledge structures from empirical data is imperative. We study two means of learning item to item structures:

- Bayesian Network structural learning;
- a POKS learning and inference algorithm.

Each approach is discussed below. Experiments to compare their respective performance for predicting item responses outcomes is reported later.

#### 3.1 Bayesian Network structural learning for POKS

In spite of the semantic differences between the links of a BN and those of a POKS, the relations of both structures can be thought of as probabilistic implications between nodes. Both can represent evidence that influences the probabilities of neighboring nodes taking on values of true or false, in accordance to a Bayesian framework. It follows that any BN structural learning algorithm is a reasonable candidate for learning item to item structures.

We conducted a study on learning item to item BN structures with the K2 and PC algorithms:

- K2 The K2 algorithm (Cooper & Herskovits, 1992) is based on a Bayesian method that looks for the most probable structure given the observed distribution. It uses a greedy search algorithm over the space of network topologies.
- PC The PC algorithm (Spirtes, Glymour, & Scheines, 2000) falls into the *constraint-based structural learning* approach. It uses conditional independence tests with a given level of significance to determine the structure.

These algorithms are regularly used in the BN learning literature. For the K2 algorithm, the general principle is to maximize the probability of a given topology given observed data. Cooper and Herskovits (1992) originally used a Bayesian metric, but the well known BIC (Bayesian Information Criterion) is also used. The algorithm performs a local search constrained by a given node ordering pattern to reduce the search space. That order is a topological order which defines a subspace of the permissible DAGs. For our experiments, the initial order is obtained first using the Maximum Weight Spanning Tree (MWST) algorithm by (Chow & Liu, 1968) to derive a network topology, and by extracting a topological order from this structure. François and Leray (2003) has shown that

the initial DAG obtained by the MWST is an effective replacement to a random ordering.

We used Ken Murphy’s BNT package for learning the BN structures of all the experiments conducted (<http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html>). The results are reported in the section 4.

### 3.2 POKS Structural Induction

The second approach we study for inducing the relations among items is based on Desmarais and Pu (2005; Desmarais et al., 1996).

This approach can be considered a constraint-based structural learning approaches since it uses conditional independence tests to determine the structure (see Neapolitan, 2004).

The POKS induction algorithm relies on a pairwise analysis of item to item relationships (Desmarais et al., 1996). The analysis attempts to identify the order in which we master knowledge items in accordance to the theory of knowledge spaces (Doignon & Falmagne, 1999) but under the stronger assumption that the skill acquisition order can be modeled by a DAG.

The tests to establish a relation  $A \succ B$  consists in three conditions for which a statistical test is applied:

$$P(B|A) \geq p_c \tag{1}$$

$$P(\bar{A}|\bar{B}) \geq p_c \tag{2}$$

$$P(B|A) \neq P(B) \tag{3}$$

Conditions (1) and (2) respectively correspond to the ability to predict that  $B$  is true given that  $A$  is observed true (*mastered*), and the ability that  $A$  is false (*non mastered*) given that  $B$  is false. The third condition verifies that the conditional probabilities are different from the non conditional probabilities (i.e. there is an interaction between the probability distributions of  $A$  and  $B$ ). These conditions are verified by a Binomial test with parameters:

$p_c$  the minimal conditional probability of equations (1) and (2),  
 $\alpha_i$  the alpha error tolerance level.

For this study,  $p_c$  is set at 0.5. Condition (3) is the independence test verified through a  $\chi^2$  statistic with an alpha error  $\alpha_i < 0.2$ . The high values of alpha errors maximize the number of relations we retain. See Desmarais et al. (1996) and Desmarais and Pu (2005) for further details about the parameters.

### 3.3 Inferences

Once we obtain an item to item structure, an assessment of the probability of success over all items can be computed from partial evidence. In other words, we wish to evaluate the validity of the two frameworks over their item outcome predictive ability. We do not attempt to assess the actual item to item structures

themselves because we have no mean to determine their respective true structure. In fact, that issue belongs to the field of cognitive science and was already thoroughly investigated by Doignon and Falmagne (see Doignon & Falmagne, 1999) and a number of other researchers. Our interest lies in the predictive power of the models which is measured by their ability to perform accurate assessment.

### 3.4 Inference in BN

For the BN structure, there exist a number of standard and well documented algorithms (see, for eg., Neapolitan, 2004). We use the junction-tree algorithm (Jensen, 1996) which performs an exact computation of posterior probabilities within a tree whose vertices's and derived from a triangulated graph, which is itself derived from the DAG in the BN.

### 3.5 Inference in POKS

For the POKS framework, computation of the nodes' probabilities are essentially based on standard bayesian posteriors under the local independence assumption.

Given a series of relations:

$$E_1 \succ H, E_2 \succ H, \dots, E_n \succ H$$

where  $E_i$  stands for an evidence node (parent) and  $H$  stands for a hypothesis node (child), the likelihood ratio of  $H$  is computed according to the following equation :

$$O(H | E_1, E_2, \dots, E_n) = O(H) \prod_i^n \frac{P(E_i | H)}{P(E_i | \bar{H})} \quad (4)$$

In case the evidence is negative for observation  $i$ , then the ratio  $\frac{P(\bar{E}_i | H)}{P(\bar{E}_i | \bar{H})}$  is used. We refer the reader to Desmarais and Pu (2005) for more details.

In the current study, we do not use transitive/recursive propagation to perform inference based on partial evidence as was done for previous studies with POKS (Desmarais & Pu, 2005; Desmarais et al., 1996). Instead, we rely on the fact that if we have strong surmise relations  $A \succ B \succ C$ , then we would also expect to find  $A \succ C$  according to the POKS structural learning algorithm. In other words, if we have  $A \succ B$  and  $B \succ C$ , no probability update is performed over  $C$  upon the observation of  $A$ , unless a link  $A \succ C$  is explicitly derived from the data.

The departure from the original POKS framework (Desmarais et al., 1996) makes the model simpler. It avoids the definition of a scheme to propagate partial evidence: propagating evidence from  $A$  to  $C$  in a structure like  $A \succ B \succ C$ , for example. Given that we expect partial evidence inferences to result in direct, transitive relations, the results are expected to be very similar. This was confirmed in our own experimental results that show that the performance is very close between the two alternatives.

## 4 Predictive Comparison of the BN and POKS Structural Learning approaches

The BN and POKS structural learning approaches are compared over their ability to predict item response outcome. We use data from real tests to conduct simulations and measure the performance of each approach for predicting the outcome over the full set of item answers from a subset of observed answers. This validation technique is identical to the ones used by Vomlel (2004) and by Desmarais and Pu (2005).

### 4.1 Simulation Methodology

The experiment consists in simulating the question answering process with the real subjects. An item is chosen and the outcome of the answer, success or failure, is fed to the inference algorithm. An updated probability of success is computed given this new evidence. All items for which the probability is above 0.5 is considered mastered and all others are considered non-mastered. We then compare the results with the real answers to obtain a measure of how accurate the predictions are. The process is repeated from 0 items administered until all items are “observed”. Observed items take their true value, such that after all items are administered, the score always converges to 1.

The choice of the question to ask is determined by an entropy reduction optimization algorithm. The same algorithm is used for the BN and POKS frameworks (for details on this algorithm, see Vomlel, 2004; Desmarais & Pu, 2005). Essentially, the choice of the next question to administer corresponds to the one that reduces the entropy of a set of network nodes. Items with very high or low probability of success are generally excluded because their expected entropy reduction value will be low.

### 4.2 Data sets

The data sets are taken from the results of two tests administered to human subjects :

1. **Arithmetic test.** Vomlel (Vomlel, 2004) gathered data from 149 pupils who completed a 20 question items test of basic fraction arithmetic for grade 6–8 pupils.
2. **UNIX shell command test.** The last data set is composed of 47 test results over a 33 question item test on knowledge of different Unix shell commands. The question range from simple and essential commands (eg. *cd*, *ls*), to more complex data processing utilities (eg. *awk*, *sed*) and system administration tools (eg. *ps*, *chgrp*).

For each data set, a portion of the data is used for training and the remaining ones for testing. Table 1 provides the size of the training and testing sets along with the average success rate of each test.



**Table 1.** Data sets

Data set	nb. items	nb. data cases			Average success rate
		Training	Test	Total	
Arithmetic	20	100	49	149	61%
Unix	33	30	17	47	53%

For each data set, six training and test sets were randomly sampled from both corpus. All performance reports represent the average over all six sampled sets.

### 4.3 Simulation parameters

A number of parameters must be set for the different algorithms used.

The BN parameters for both the K2 and PC algorithms were initialized with Dirichlet uniform priors, which correspond to Beta priors in the case of binomial variables.

For the PC algorithm, recall that it relies on a conditional independence test. The significance level chosen for this test in our experiment is 0.2, the same value as the one used for the POKS interaction test (see below).

For the POKS structural learning, two parameters need to be set. They are:

$$p_c = 0.5 \text{ and } \alpha_i = 0.2$$

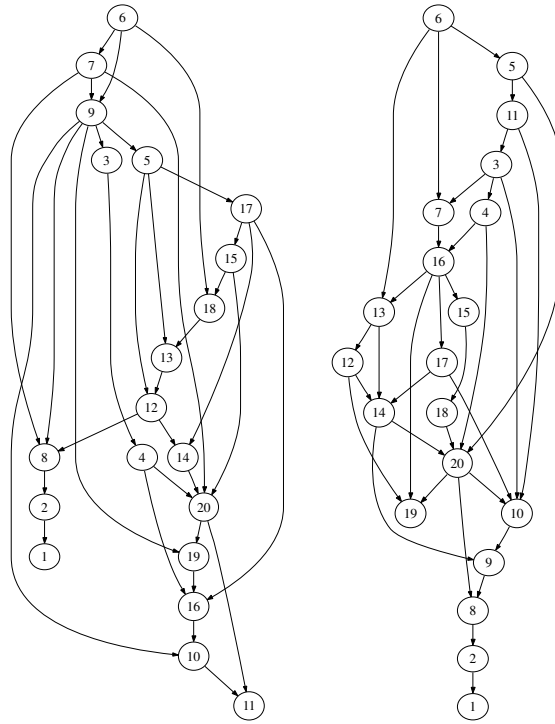
These values were also used in Desmarais et al. (1996) and they are generally appropriate when the number of nodes is below 50.

The simulations are run with Ken Murphy’s BNT package (<http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html>). Note that it was not possible to test the PC algorithm for the Unix test because it resulted in an error with Matlab.

### 4.4 Learned Structures

Over all six randomly sampled sets, the POKS structural learning algorithm created structures that, for the arithmetic data set, contains between 181 and 218 relations, of which 117 to 126 are symmetric, for an average between 9.1 to 10.9 links per node. For the Unix data set, the number of relations varies between 582 and 691, and the number of symmetric relations varies between 348 and 297, and average relations per node that varies between 17.6 to 20.9. The structure of the Unix data set is thus much more populated with an average link per node about twice that of the arithmetic test.

For the BN structural learning results, figure 2 displays the first two structures learned with the K2 algorithm. Recall that the structures of both the K2 and PC algorithms were constrained to 1 parent. It can be seen that the topology differs significantly between the two network shown in this figure. In general,



**Fig. 2.** Two examples of BN structures learned with the K2 algorithm.

about only half of the relations are common between BN from two samples. However and as mentioned, we do not focus on the actual topologies in this study but solely on the ability of the induced structures to perform accurate inferences.

Processing time for learning differs significantly between the two BN structural learning algorithms.

For the arithmetic data, we obtained these values on a 1Ghz pentium PC running Linux :

PC:  $\approx 1080$  seconds and K2:  $\approx 3$  seconds

Learning time for the POKS algorithm is considerably faster than both. It stands around  $80ms$ . However, these numbers should be interpreted with care as the POKS algorithm is a C program whereas the BN software runs under Matlab. Nevertheless, the POKS algorithm has order  $O(ni^2)$  time complexity, where  $n$  is the number of cases and  $i$  is the number of item nodes, which is far less than the K2 algorithm ( $O(ni^4)$  for the current context).

On the performance of the evidence propagation algorithms, the total simulation time for the BN varied between 2120 seconds to 2609 seconds. This is again orders of magnitude greater than for the POKS algorithm which took about 1 second per simulation. Each simulation involves looping over 49 cases and, for each for each of the 20 question item, performing on average 19 inferences

( $n - 1$  items). To find the average time of an inference we would thus divide each simulation time by  $49 * 20 * 19 = 18,620$ .

#### 4.5 Results

The performance results are averaged over all test subjects and random samples. The formula for computing the accuracy of the prediction after each observed item is:

$$\text{Accuracy} = \frac{\sum_i^r \sum_j^m \sum_k^n M_{ijk}}{r m n}$$

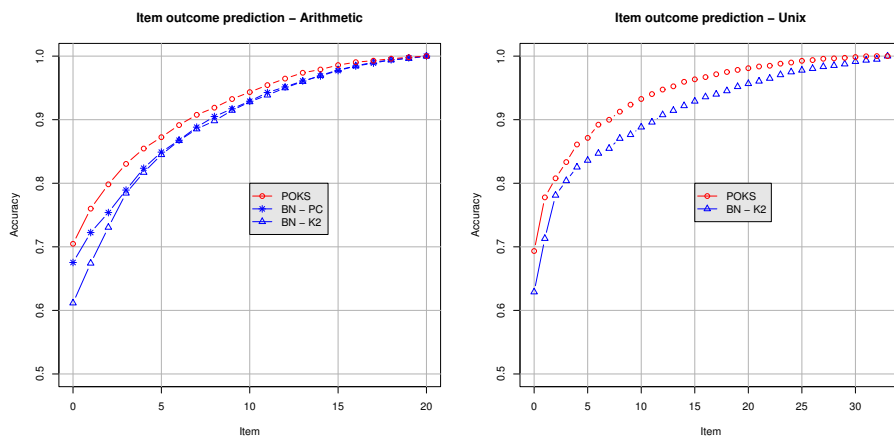
where  $r$  is the total number items in the test,  $m$  is the number of random sample runs of the simulation (6), and  $n$  is the number of test subject cases (17 for the Unix test and 49 for the arithmetic test).  $M_{ijk}$  represents the item outcome prediction to item  $i$  by subject  $k$  for the simulation run  $j$ . It is 1 if the prediction is correct or if it is an observed item, and 0 otherwise.

Figure 3 reports the simulations results for the item outcome predictive performance. It shows that, for both data sets, the POKS algorithm yields more accurate predictions than the two BN algorithms. Although the difference is only a few percentage points, it is relatively significant. For example, after 20 items, the difference between the BN and POKS for the Unix data set is about 95% compared to 98%. Although this represents a 3% difference in absolute values, it should be regarded relative to error reduction. In terms of the remaining error, it represents a 60% relative reduction. Viewed from a different perspective, it means that the accuracy reached by POKS after 20 of the 33 item Unix test is only reached after about 27 items for the BN K2 algorithm. In a context where, for example, we need strong confidence that a specific item is mastered and avoid mistakes, the difference in reliability can be significant.

Looking at the relative error reduction between both tests, we note that the performance difference between algorithms is greater for the Unix than for the arithmetic test. This is potentially due to the fact that this test was meant to discriminate between a wide range of expertise, from basis file manipulation commands to sophisticated data manipulation scripts, whereas the arithmetic test is more focused on closely related skills and notions. The ordering between items that is typical of knowledge structures is more likely to appear for the Unix than the arithmetic test.

We also note that the PC algorithm performs better than the K2 algorithm, apparently due to more accurate priors. However, the difference quickly vanishes after 2 items are observed.

To obtain an estimate of the statistical significance of the performance results, Figure 4 reports the variability of the estimates across subjects. The plot represents the quartiles for the different algorithms. The boxes contain a line indicating the median and span over roughly the upper and lower quartiles around it. The ‘whiskers’ represent the outer quartiles and outliers are shown by themselves. The arithmetic simulation results of Figure 4 contains two series of boxes for the BN results that respectively correspond to the PC and K2 algorithms, whereas the Unix contains only the K2 algorithm results.



**Fig. 3.** Item prediction performance for the Arithmetic and Unix tests. Each line represents the average over all 49 test cases and over 6 simulation samples.

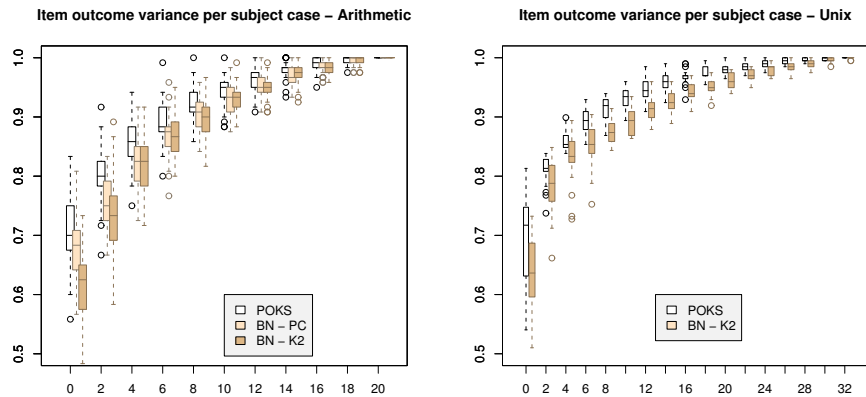
## 5 Discussion

This study shows that item to item structures can be constructed from data and yield effective predictions of item outcome. Two approaches were investigated, namely standard BN induction techniques and the POKS framework, which stands closer to the Naive Bayes family of models.

Experiments over the two data sets show that the POKS framework yields better predictive results for item outcome prediction than does the general BN framework. Although the strong performance of a Naive Bayes framework is by no means uncommon (Domingos & Pazzani, 1997), we conjecture that, for this study, it is consequent with the constrained nature of POKS: knowledge spaces closed under union and intersection. The assumptions made by the POKS framework may be appropriate for the knowledge spaces and, consequently, allow the framework to be applied effectively with small data sets as the ones we experimented with.

Item to item student models are based solely on observable variables. This implies that they can readily be learned through standard and general BN structural learning techniques, or through more specialized techniques such as POKS. The absence of hidden variables makes the learning more robust and more efficient. The POKS approach is particularly efficient for both learning and inference.

These models yield prediction of item responses. As such, they can be used to assess higher level concepts or skills, much in the same manner as we would use test results to assess the mastery of a topic or diagnose learning difficulties. However, they do not model misconceptions, nor build link from specific answers to specific skills or misconceptions as BN can do. Some tutoring systems require such diagnostic in order to provide more relevant remedial material for a student.



**Fig. 4.** Box plot of the variance across subjects. Central quartiles are within the box whereas the two extreme quartiles are represented by the “whiskers”.

Nevertheless, their simplicity and the possibility of learning item to item models with relatively small data samples makes them very attractive for many learning systems such as study guides.

## Acknowledgments

We are grateful to Jiří Vomlel for providing the data used in this experiment. This work has been supported by the National Research Council of Canada.

## 6 References

- Chow, C., & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. Information Theory*, 14(11), 462–467.
- Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4), 371–417.
- Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347.
- Desmarais, M. C., Maluf, A., & Liu, J. (1996). User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction*, 5(3-4), 283–315.
- Desmarais, M. C., & Pu, X. (2005). A bayesian inference adaptive testing framework and its comparison with item response theory. *International Journal of Artificial Intelligence in Education*, 15, 291–323.
- Doignon, J.-P., & Falmagne, J.-C. (1999). *Knowledge spaces*. Berlin: Springer-Verlag.

- Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29, 103–130.
- Dowling, C. E., & Hockemeyer, C. (2001). Automata for the assessment of knowledge. *IEEE Transactions on Knowledge and Data Engineering*.
- Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build test and search them. *Psychological Review*, 97, 201–224.
- François, O., & Leray, P. (2003). Etude comparative d’algorithmes d’apprentissage de structure dans les réseaux bayésiens. *RJCIA03* (pp. 167–180).
- Jensen, F. V. (1996). *An introduction to Bayesian networks*. London, England: UCL Press.
- Kambouri, M., Koppen, M., Villano, M., & Falmagne, J.-C. (1994). Knowledge assessment: tapping human expertise by the query routine. *International Journal of Human-Computer Studies*, 40(1), 119–151.
- Millán, E., de-la Cruz, J.-L. P., & Suárez, E. (2000). Adaptive Bayesian networks for multilevel student modelling. *ITS’00: Proceedings of the 5th International Conference on Intelligent Tutoring Systems* (pp. 534–543). Springer-Verlag.
- Mislevy, R. J., Almond, R. G., Yan, D., & Steinberg, L. S. (1999). Bayes nets in educational assessment: Where the numbers come from. In K. B. Laskey, & H. Prade (Eds.), *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)* (pp. 437–446). S.F., Cal.: Morgan Kaufmann Publishers.
- Murphy, K. P. (2001). *The Bayes net toolbox for MATLAB* (Technical report). University of California at Berkeley; Berkeley, CA.
- Neapolitan, R. E. (2004). *Learning Bayesian networks*. New Jersey: Prentice Hall.
- Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, prediction, and search*. Cambridge, Massachusetts: The MIT Press, 2 edition.
- VanLehn, K., Niu, Z., Siler, S., & Gertner, A. S. (1998). Student modeling from conventional test data: A Bayesian approach without priors. *ITS’98: Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (pp. 434–443). London, UK: Springer-Verlag.
- Vomlel, J. (2004). Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 12(Supplementary Issue 1), 83–100.