

# Some Issues in the Learning of Accurate, Interpretable User Models From Sparse Data

Frank Wittig\*

Department of Computer Science, Saarland University  
P.O. Box 15 11 50, D-66041 Saarbrücken, Germany  
wittig@cs.uni-sb.de

**Abstract.** We discuss issues that arise when applying techniques for the learning of Bayesian networks in the user modeling context. We address the problem of sparse data that is often present in user modeling and show how we try to cope with it by introducing available a-priori knowledge into the learning procedures. Particularly, we present initial results concerning the learning of the structural part of a Bayesian network user model.

## 1 Introduction and Background

In this paper, we discuss issues that appear to be especially important when applying techniques for the learning of Bayesian networks (BNs) in the user modeling context. Some of these issues also arise when other machine learning methods are applied to user modeling. For concreteness, we use as an example the procedure of learning a BN that relates features of speech to mental states of the user (e.g., his working memory load). In Müller, Großmann-Hutter, Jameson, Rummer, and Wittig (2001) a similar BN is used. The present paper can be viewed as a description of steps in the development of a methodology for deriving such BNs in an empirical based manner – with focus on the structural learning part.

Before presenting specific results, we will discuss more generally some aspects of applying machine learning methods in the user modeling context. Beyond others, in our research efforts on learning user models we have identified and address the following problems:

- *How can we learn accurate user models when data is sparse?* Often, there are only a few interactions between a user and the user-adaptive system. Sometimes a user interacts only a single time with, e.g., an adaptive help system to recommend things he might want to buy. A more general problem occurs in every adaptive system: How should the system interact with a new user? In some domains this problem can be reduced by using data from more than just a single user to learn a general user model by, e.g., *collaborative filtering* methods that could use all data available from all users so far to make useful recommendations for new users.

---

\* The research described was supported by the German Science Foundation (DFG) in its Collaborative Research Center on Resource-Adaptive Cognitive Processes (<http://www.coli.uni-sb.de/sfb378/>), SFB 378, Project B2 (READY, <http://w5.cs.uni-sb.de/~ready>).

- *How can we deal with potentially large differences between individual users?* A user-adaptive system is typically designed to optimally adapt to an individual user. If large datasets consisting of many users' data can be used to build the user model, how can we take into account large individual differences regarding their behavior, intentions, preferences etc.?
- *How can we ensure that the learned user models remain interpretable?* There are mainly two reasons why user models should be interpretable: (a) an interpretable model makes the knowledge engineering task of building the model far more easier. The system's designer or her domain experts can inspect the learned model and localize potential weaknesses. It is then possible to modify the model in an improve-and-test cycle until the quality demands are met. And, (b) the acceptance of a user-adaptive system increases if it can explain the way it derives its results in a comprehensible manner to its users.
- *How can we prevent learned models from overfitting the data?* This is a well-known problem in many machine learning settings. It is equally important in the user modeling context. Here we have to avoid a specialization of the induced model to specific aspects of the users' behavior or – when learning for users in general – to particular users. Often it is not possible to acquire data on a sufficient number of users to cover all relevant user communities – parts of the user population that share many common aspects relevant for the model – and thus the system may perform worse if it encounters a user that does not belong to any user group that has been covered by the learning data.

Our general approach is based on the assumption that the exploitation of available a-priori knowledge can provide significant contributions to solutions for these problems. Usually, there is much prior knowledge available in the user modeling context, be it part of common knowledge or derived from psychological studies. We developed methods to integrate this kind of information into existing learning algorithms. In all efforts we emphasize the need of keeping the user models interpretable.

The paper is structured as follows: In Section 2 we describe our example domain, in Section 3 we briefly review Bayesian networks and how to learn them followed by a discussion of how to exploit and integrate available a-priori knowledge into these learning procedures. Section 5 describes and discusses the results of an explorative study on learning BN structures for user modeling. The paper is concluded by some remarks on the work presented here.

## 2 Example Domain

We now briefly introduce one of the psychological experiments<sup>1</sup> that we performed to acquire data for the type of analysis presented here. For a more detailed description and discussion of the traditional psychological analysis and results, we refer to Müller et al. (2001).

---

<sup>1</sup> The experiment was designed, implemented, performed and analyzed with contributions by (in alphabetical order) Sylvia Bach, Barbara Großmann-Hutter, Anthony Jameson, Tore Knabe, Christian Müller and Ralf Rummer.

The experimental environment simulated on a computer workstation a situation in which a user is navigating through a crowded airport terminal while asking questions to a mobile assistance system via speech. In each trial, a picture appeared in a corner of the screen, and the subject was to introduce and ask a question related to the picture (e.g., “I’m getting thirsty. Will it be possible to get a beer on the plane?”).

Two independent variables were manipulated orthogonally:

- TIME PRESSURE?(2)<sup>2</sup>: Whether the subject was instructed (a) to finish each utterance as quickly as possible or (b) to create an especially clear and comprehensible utterance, without regard to time.
- SECONDARY TASK?(2): Whether or not the subject was required to “navigate” through the terminal depicted on the screen by pressing arrow keys in order to move the cursor on the screen, avoiding obstacles in the process.

Additionally, we had a third independent variable that we did not manipulate explicitly during the experiment, the variable PICTURE DIFFICULTY(2) that categorized the presented pictures according to their “complexity” to produce a related question. Its values were determined on the basis of judgments of four experts.

In each of the 4 (2 × 2) conditions, each of the subjects produced 20 utterances. There are therefore 80 “observations” of each subject.

The subjects’ speech input was later semi-automatically coded with respect to a wide range of features, including pauses, length, quality of content, and various types of disfluency. For the present study of learning methods, we selected a representative subset of six speech-related variables, which we call *symptoms*:

- QUALITY SYMPTOM?(2): This binary variable has the value “true” when any one of four types of disfluency was present in an utterance.
- NUMBER OF SYLLABLES(3): The number of syllables in the utterance.
- SILENT PAUSES?(2): This binary variable represents the presence/absence of silent pauses in the utterance.
- FILLED PAUSES?(2): The corresponding variable for filled pauses (e.g., “Uhh”)
- ARTICULATION RATE(3): The number of syllables articulated per second of speaking time, not including silent pauses.
- CONTENT QUALITY(4): The average quality rank – between 1 (worst) and 32 (best) – assigned to the utterance by four raters.

On the whole, the traditional analysis of the data from this experiment reveals many statistically significant effects (see Müller et al., 2001) of the independent variables on features of speech input. Some of these effects, however, are rather subtle and complex (i.e., involving statistical interactions) so that it is not a trivial task to construct an adequate user model.

The practical relevance of this experiment lies mainly in the prospect that a mobile assistance system could interpret the features of a user’s speech to make inferences about his current psychological state. In addition, there are situations in which it can be useful for the system to be able to predict particular features of the user’s speech in a

<sup>2</sup> The numbers in parentheses behind the variables represent the number of the variables’ states after discretization for their usage in the BN presented in the next section.

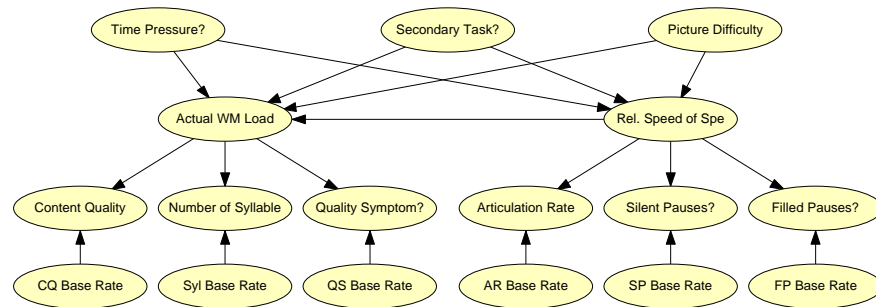
given situation – for example, so as to determine whether to request input via speech or via another modality.

### 3 Learning Bayesian Networks

In this section, we briefly review existing BN learning algorithms that we used as starting points for our research on learning BNs for user modeling.

A *Bayesian network* represents a joint probability distribution over discrete random variables. It consists of (a) a directed acyclic graph (DAG) that represents probabilistic independence relations between these variables and (b) *conditional probability tables (CPTs)* associated with each variable that encode the conditional probability distribution of a variable's values conditioned on its parents' values.

In addition to its ability to make inferences in domains under uncertainty, an aspect of the BN framework that makes BNs quite appealing for user modeling is that the links in the DAG are commonly interpreted as causal influences between the associated variables. Therefore, BNs are a tool that enable us to build quite easily interpretable user models. In principle, even users without scientific background can understand how variables influence others in the network through the causal interpretation of the links<sup>3</sup>.



**Fig. 1.** Bayesian network for the experiment of Section 2

Figure 1 shows a BN that represents a user model for the experiment described in Section 2. On the top row there are the independent variables that have causal influences on the two variables ACTUAL WORKING MEMORY LOAD(3) and RELATIVE SPEED OF SPEECH GENERATION(3). The link between these two variables is based on the assumption that a reduction regarding the speed of speech generation reduces the user's actual working memory load. Each symptom variable is related to one of the two inner variables and a *base rate variable*<sup>4</sup> that represents a user's general tendency with regard

<sup>3</sup> This statement applies as long as the networks do not become overly complex. Then even experienced developers of BN technology may lose insight in how different parts of the networks influence others.

<sup>4</sup> All base rate variables are binary variables modeling the states 'high' and 'low', respectively.

to the value of this particular variable, e.g. if he is a person who normally speaks quite fast, yielding a high average value for the variable ARTICULATION RATE, then this fact is modeled by a high value of the corresponding base rate variable AR BASE RATE.

To learn a BN we have to consider two particular learning tasks that influence each other: (a) learning the values of the CPTs and (b) learning the structure (the DAG) of the BN.

Assuming a given BN structure, the problem of learning a BN is reduced to the task of learning the conditional probabilities in the CPTs. If there are no missing data then learning can be done in a straightforward manner on the basis of relative frequencies which yield maximum likelihood estimates of the CPT entries (see, e.g., Buntine, 1996). In the case of missing values or, even worse, hidden variables – variables whose values are never observed in the data – more sophisticated learning techniques have to be applied. The two most frequently used are the gradient-based *adaptive probabilistic networks* (APN) method developed by Binder, Koller, Russell, and Kanazawa (1997) and the *Expectation Maximization* (EM) algorithm (Dempster, Laird, & Rubin, 1977). Both methods use as quality measure which they try to optimize the *likelihood*  $P(\mathbf{D}|\boldsymbol{\theta})$  of the data  $\mathbf{D}$  regarding the BN under consideration, which is – in the case of a given structure – uniquely characterized by its CPT values  $\boldsymbol{\theta}$ . In our example domain, the variables ACTUAL WORKING MEMORY LOAD and RELATIVE SPEED OF SPEECH GENERATION are hidden variables.

There exist several methods for learning the structure of BNs when data is complete, see e.g. Heckerman (1998). For BNs with hidden variables the SEM (structural EM) algorithm was developed by Friedman (1997). This hybrid algorithm alternates between improvements of the structure (e.g., through a greedy search strategy) and improvements of the CPT values. Regarding the latter part, both algorithms – EM and APN – can be used. The quality measure that is commonly used by SEM is the *Bayesian Information Criterion (BIC)* (Schwarz, 1978):

$$\log P(\mathbf{D}|\boldsymbol{\theta}) - \frac{d}{2} \log n, \quad (1)$$

where  $d$  represents the dimension of the BN, i.e. roughly the overall number of its CPT entries, and  $n$  is the number of cases in  $\mathbf{D}$ . A common interpretation of the BIC measure is given by its division into two parts: (a) the logarithm of the likelihood of the data, given the current model and (b) a term that penalizes more complex structures (with more CPT entries). This quality measure is therefore well suited for our context, since we would like to learn interpretable models. Very complex models are rarely interpretable ones.

## 4 Exploiting Available Prior Knowledge

Now, we refer back to the problems stated in the introduction of this paper – with focus on the structural part of the learning problem. Detailed discussions on several aspects of the learning of a BN’s CPTs for user modeling can be found in Wittig and Jameson (2000) and Jameson and Wittig (2001). We address the relevant issues by the introduction of several types of available a-priori knowledge into the learning procedure:

- *Individual differences*: There exist at least two methods to model individual differences between users: (a) through explicit structural modeling, with for example the base rate variables included in our example BN and (b) through the adaptation of a general user model learned offline – on the basis of data from many users – to the individual user at the system’s runtime. In both situations, a-priori knowledge about the type of individual differences contributes to the right choice of which method to apply in a given context, see, e.g., Jameson and Wittig (2001) for a detailed discussion.
- *Presence of hidden variables*: An important point for the construction of user models in the form of BNs is a-priori knowledge related to the presence (or absence) of hidden variables like ACTUAL WORKING MEMORY LOAD and RELATIVE SPEED OF SPEECH GENERATION in our example. Models that include such variables benefit in two different ways thus justifying the learning task’s increased complexity: (a) in most cases a model with hidden variables has a simpler structure with fewer links yielding often many fewer CPT values that need to be learned and (b) hidden variables contribute to the interpretability of the user model. In Wittig and Jameson (2000) we present a methodology to solve problems that arise when learning interpretable BNs with hidden variables.
- *Explicit knowledge concerning aspects of the user model*: In most domains, it is easier for an expert or even the systems’ developer herself relying on her common knowledge to specify the structural part of the user model than to specify the CPTs. Often, there is a lot of information available on causal relations between the models’ parts. As already mentioned, through the causal interpretation of links, the BN framework is well suited for the user modeling context. Nevertheless, the model building process should benefit from the application of machine learning techniques to induce (some parts of) the structure of a BN representing a user model, especially in domains where different structures of models are possible.

In the following section we present an exploratory study on how the introduction of explicit knowledge concerning several aspects of the model’s structure influences the results of the learning procedure. Initially, we considered two types of prior structural knowledge that can be exploited to limit the search space of potential structures: (a) specification of an initial BN structure for the learning and (b) specification of structural constraints that have to hold during the whole learning process, i.e. forbidden links, links that must be present or nodes that are not allowed to have any parent.

## 5 Tests of Learning Methods

First, we describe the procedure of the experiment that we conducted to study the impact of the specification of structural constraints on the finally learned user models, followed by a discussion of some of the – in our opinion – most interesting results and their implications for the directions of our future work.

### 5.1 Procedure of the Tests

We performed a cross-validation test using the data of the 32 subjects in our experiment, i.e. using data of 31 subjects to learn a user model that was then scored against the data

of the remaining subject. This was done for each of 32 possible combinations starting with the BN structure of Figure 1 with randomly initialized CPTs<sup>5</sup>. As learning procedure the SEM algorithm with an EM algorithm as the inner CPT learning method (with 20 iterations) was used. We performed four tests, each with different sets of structural constraints specified for the learning task:

1. No structural constraint was specified. The SEM algorithm was applied with the BN of Figure 1 as its starting point. This represents the alternative with most degrees of freedom for the learning procedure.
2. The independent variables in our experiment were not allowed to have any parent and there had to be a link from RELATIVE SPEED OF SPEECH GENERATION to ACTUAL WORKING MEMORY LOAD as well as links between the base rate variables and their corresponding symptom variables.
3. In addition to the structural constraints of 2, the learning method had to include all links in the final learned BN that were present in the one used as the starting point for the SEM algorithm's greedy search procedure (cf. Figure 1).
4. No structural learning was allowed. Only the CPTs' values were learned with the EM algorithm, using the structure of Figure 1.

We evaluated the results of these four learning tasks using the *average negative log likelihood per case* measure. We omitted the complexity part of the whole BIC quality measure (see Equation 1) for evaluation because it is not directly relevant to the predictive accuracy of the learned BNs. It will be an interesting issue for further study to relate this part to the problem of ensuring the interpretability of a learned user model.

## 5.2 Results and Discussion

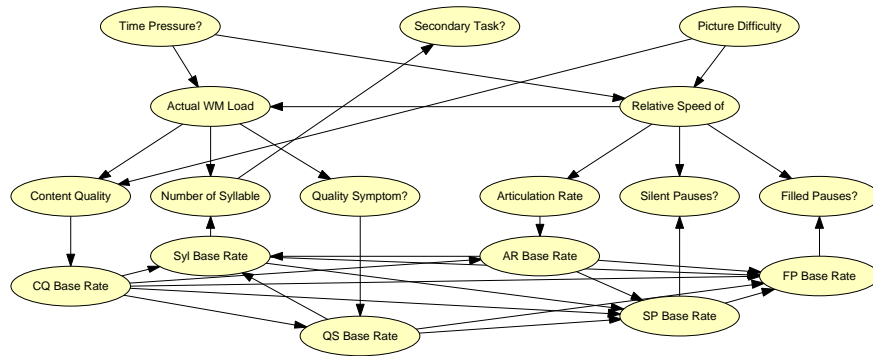
**Learned Structures** Figure 2 shows a prototypical structure learned for the first three learning situations on the basis of 31 subjects<sup>6</sup>. The main result is that in all three cases where structural learning was allowed, the variables representing the base rate variables were strongly connected through new links introduced by the SEM algorithm. In fact, these were always the first changes made in the learning procedure, leading to the largest improvements of the BIC score during the early learning steps. Traditional statistical analyses of these variables confirmed that there are some strong correlations. Furthermore, there is only one profile of base rate values for each subject's 80 observations thus reducing the amount of variation regarding the combination of these values and contributing to the addition of these links during the early learning phase.

Moreover, some links were removed or reversed by the SEM algorithm in the test settings with fewer and no structural constraints (Situations 2 and 1). Particularly, it seems to be quite difficult for the learning algorithm to recognize the causal relationships involving the independent variable SECONDARY TASK?. We already made related observations in the work described in Müller et al. (2001), where the issue is further discussed.

---

<sup>5</sup> An analysis with several differently initialized starting BNs is currently being conducted.

<sup>6</sup> This particular BN was learned in first specified situation (no structural constraints). Overall, the structures show little variation over the different combinations of learning and test data.

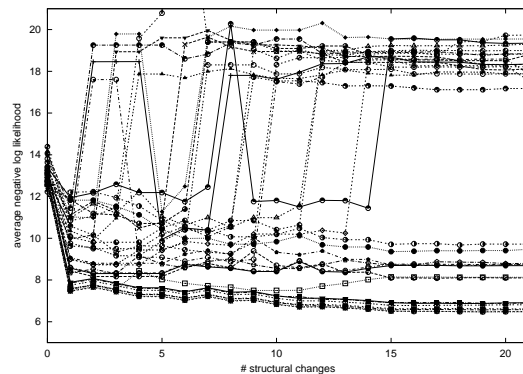


**Fig. 2.** Learned structure (prototypical example)

Only a few results contained links between two symptom variables, for example, a link from NUMBER OF SYLLABLES to ARTICULATION RATE.

**Quality of the Learned Models** The main result concerning the numerical quality of the learned BNs is related to the observation made regarding the interconnectivity of the base rate variables. When scoring the (intermediately) learned BNs against the data of the test subject (see Figure 3)<sup>7</sup>, we can clearly distinguish between two groups within the 32 learned BNs: (a) one group consisting of 15 BNs that performed well and (b) one consisting of 17 BNs that performed significantly worse than the other group.

Figure 4 shows two prototypical curves of the course of structural learning. Searching for an explanation, we had a closer look at the data and observed that the second group of BNs relates to subjects that were quite individual regarding the base rate values, i.e. there were unusual individual values or profiles different from those of almost all other subjects. Each of these subjects showed at least one combination of values of base rate variables that had not been encountered with the subjects used for learning (e.g. for subject 2 there was a low proportion of silent pauses and a high proportion of filled pauses). Consequently, this combination was assigned an extremely



**Fig. 3.** Individual results of the learned BNs

<sup>7</sup> The following particular graphs are based on the results of the first learning situation. The values for the other two situations are very similar. The specification of more or less structural constraints did not have any significant influence on the presented aspects of the results. In all graphs, lower values represent better values.

low value by the learning algorithm, and the fit to this subject’s data was very poor. Looking at the dashed curve on the top of Figure 4 we can see a point where the quality breaks down (in this particular example after the fifth structural change). This is the point in the learning procedure when a link is added whose CPT can not be reliably learned with the available learning data regarding the model’s ability to generalize to new users (regarding the result for subject 2 this was the case when a link from SP BASE RATE to FP BASE RATE was added to the model).

Figure 5 shows the results if we partition the learned BNs into 15 “good” and 17 “bad” ones and evaluate them. Looking at the curve that represents the good BNs, we observe a strong quality improvement after the first SEM learning step and only quite minor improvements from then on. This is due to the large adjustments of the CPT values by the inner EM algorithm during the first structural search step. Later, only small adjustments of these parameters are made in the very similar structures.

(Remember that subsequent structures in the learning procedure differ only by a single added/removed/reversed link.)

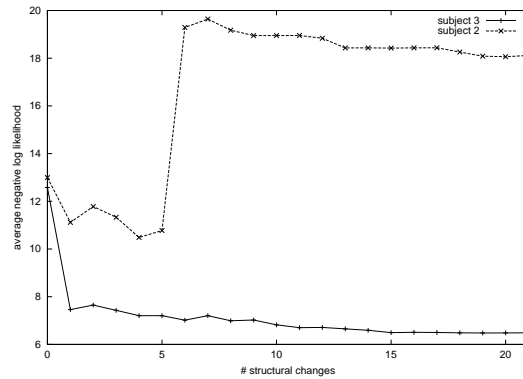


Fig. 4. Results for subjects 2 and 3

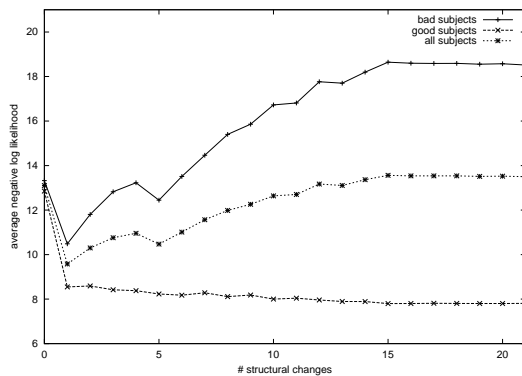


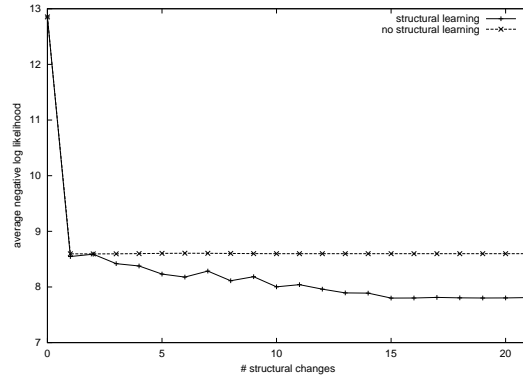
Fig. 5. Good vs. bad results

iterations in each search step, we run the EM algorithm for 420 iterations and took the intermediate results after every twentieth step.

This observation raises the question whether we can benefit from learning (parts of) the BNs structure at all. Figure 6, where we compare the presented results to results for learning the CPTs without any structural learning allowed, indicates that in our domain, structure learning indeed leads to a better quality of the learned user models. These curves are based on the values of the 15 “good” combinations of learning and test datasets only. Since our SEM algorithm performed 20 EM

**Discussion of Results** The results of our exploratory study include every issue we mentioned in the introduction of this paper.

The division of the learned user models into “good” and “bad” ones is due to overfitting in the structural learning caused by large individual differences of the users. Commonly, a link is introduced into the model whose CPT values can not be reliably learned on the basis of the limited learning data that does not represent a adequate sample of the whole population. Since such a situation often occurs in the learning of user models it is worthwhile to use methods to cope with this problem. A possible solution could be to adapt the model at the system’s runtime to the individual user. A method has to be found that detects the “individuality” of the current user and modifies the user model accordingly. The parts of the model that are responsible for the bad performance have to be identified and adjusted in the right way.



**Fig. 6.** Structural vs. no structural learning

In our example the different sets of structural constraints that were specified before learning took place did not yield any significant differences regarding the accuracy of the learned models. Although this is probably not a generalizable result, here we could at least keep the learned user models more interpretable by specifying structural constraints without losing any quality. See Figure 2 for a BN learned without structural constraints that is indeed quite difficult to interpret from a theoretical point of view. The specification of structural constraints yielded far better models with regard to their interpretability, for example, in situations 2 and 3 it is not possible that there is link between a symptom variable like NUMBER OF SYLLABLES and an independent variable like SECONDARY TASK? included.

Nevertheless, we observed an improved performance of the models when structural learning was allowed in addition to the learning of the CPTs. This indicates that it is worthwhile to address the problems that occur within structure learning and try to develop related methods. That is a main part of our ongoing work.

## 6 Concluding Remarks

In this paper, we presented issues that we think to be especially important when applying machine learning techniques for user modeling with Bayesian networks. As a concrete example scenario we used data that we acquired by a psychological experiment for the learning of a user model. We focused on the structural part of the learning problem and showed that these problems indeed play a role in this setting.

Our work in progress includes the development of methods to cope with these issues based on methods that we developed for the learning of the CPTs’ values of BN

user models. As we have done there, we try to achieve this goal through the exploitation of available background knowledge that goes beyond the specification of structural constraints like those we initially studied in this paper.

## Acknowledgments

Anthony Jameson provided valuable comments on the work described in this paper.

## References

- Binder, J., Koller, D., Russell, S., & Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29, 213-244.
- Buntine, W. (1996). A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8, 195-210.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39, 1-38.
- Friedman, N. (1997). Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning*.
- Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In M. I. Jordan (Ed.), *Learning in graphical models*. Cambridge, MA: MIT Press.
- Jameson, A., & Wittig, F. (2001). Leveraging data about users in general in the learning of individual user models. In B. Nebel (Ed.), *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann.
- Müller, C., Großmann-Hutter, B., Jameson, A., Rummer, R., & Wittig, F. (2001). Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In J. Vassileva & P. Gmytrasiewicz (Eds.), *UM2001, User Modeling: Proceedings of the Eighth International Conference*. Berlin: Springer.
- Schwarz, G. (1978). Estimating the dimension of a model. *Annals in Statistics*, 6, 461-464.
- Wittig, F., & Jameson, A. (2000). Exploiting qualitative background knowledge in Bayesian network learning algorithms. In C. Boutilier & M. Goldszmidt (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference* (p. 644-652). San Francisco: Morgan Kaufmann.