

# Summary: A Tutorial on Learning With Bayesian Networks

Markus Kalisch

May 5, 2006

We primarily summarize [4]. When we think that it is appropriate, we comment on additional facts and more recent developments.

## 1 Abstract and Introduction

The advantages of graphical modelling include

- easy handling of missing data
- easy modelling of causal relationships
- easy combination of prior information and data
- easy to avoid overfitting

## 2 Bayesian Approach

- Degree of belief
- Rules of probability are a good tool to measure belief
- Probability is subjective and has to be assessed for instance with a probability wheel. Problems of **precision** and **accuracy** can occur.
- The paper considers as a running example multinomial sampling with Dirichlet priors.

## 3 Bayesian Networks (BN)

A Bayesian network (BN) is defined by

- a network structure (DAG)
- local probability distributions

so that

$$P(x) = \prod_{i=1}^n P(x_i | pa(x_i))$$

To learn a BN, we have to

- choose the variables of the model
- choose the structure of the model
- assess local probability distributions

## 4 Inference in Bayesian Networks

- Direct approach impractical
- There are alternatives, e.g. message passing to speed things up
- Probabilistic inference in BN is NP-hard. Approximations or special-case-solutions are needed.

## 5 Learning Parameters given structure

Goal: Compute posterior for given prior, structure and data. Multinomial sampling is used as an example.

### 5.1 Complete Data

If the data is complete and mutual independence of parameters is assumed, the problem can be reduced to the one-variable case:

$$P(\theta_s | D, S) = \prod_{i=1}^n \prod_{j=1}^{q_i} P(\theta_{ij} | D, S)$$

### 5.2 Incomplete Data

We assume “missing at random”. For every case with missing observations, the posterior will split up into a Dirichlet mixture. I.e., if there are many missing observations, the posterior will be a mixture of Dirichlet mixtures and the number of components in this formula grows exponentially in the number of incomplete cases. Therefore, approximations are needed:

- Gibbs Sampling: A Monte-Carlo Method; usually computer intensive

- Gaussian Approximation: Second order Taylor approximation of log-posterior around the MAP. Needs MAP and a Hessian matrix, which might be hard to get. We can replace the MAP by the ML (good for large sample size) and compute either by gradient-based optimization or EM-algorithm.

## 6 Learning Parameters and structure

For computing the probability of some random event  $T$  given data  $D$ , we can use the formula:

$$P(T|D) = \sum_S P(S|D) \int P(T|\theta, S)P(\theta|D, S)d\theta$$

I.e., one has to sum over all possible structures. Unfortunately, the space of DAGs grows superexponentially in the number of nodes (see Figure 1), i.e., there is no hope for computing this sum analytically in real world examples. Even the number of equivalence classes is conjectured (although not proven) to grow superexponentially (see [3]).

## 7 Model Selection

The last section showed that exact computations in BN learning easily get infeasible. An alternative to averaging over all models is to find the best fitting model, i.e., use model selection. A good criterion is the log posterior probability

$$\log(P(D, S)) = \log(P(S)) + \log(P(D|S))$$

For complete observations, this can be computed exactly. If there are missing values, we run into problems as mentioned before. A good choice then is using the (already mentioned) Gauss approximation. A further simplification, which is valid for large sample sizes, leads directly to the **Bayesian Information Criterion (BIC)**:

$$\log(P(D|S) \approx \log(P(D|\hat{\theta}, S)) - \frac{d}{2} \log(N)$$

Note, that the ML estimator  $\hat{\theta}$  of the parameters for a given structure was used.

## 8 Choice of Priors

We have already seen, that different DAGs can be equivalent in terms of the conditional independence statements they encode (independence equivalence). Furthermore, one could imagine a situation in which a BN over

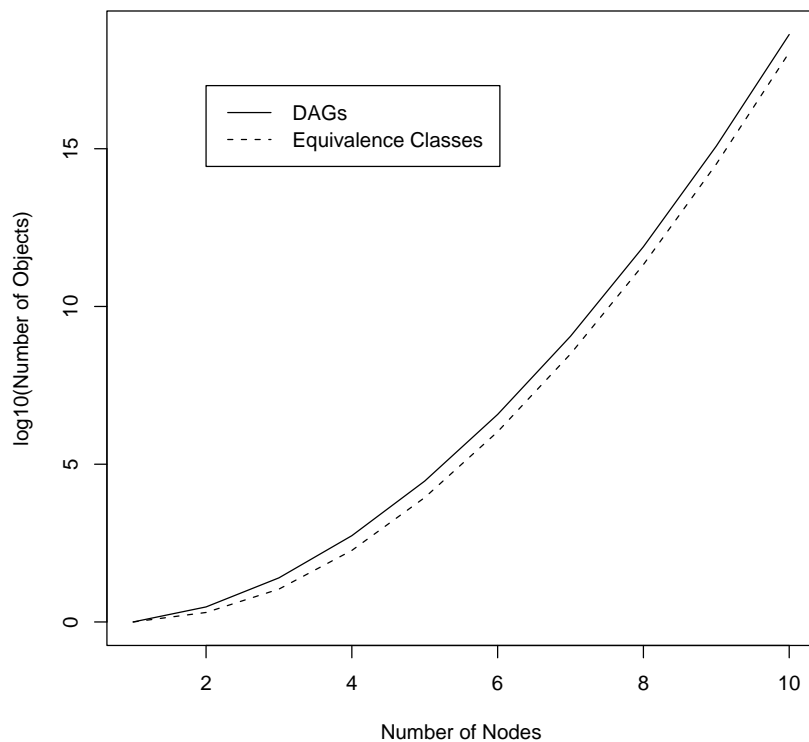


Figure 1: Number of DAGs and Equivalence Classes as function of nodes. Note, that the number of objects is on log10-Scale. Both class sizes grow superexponentially; on average, an equivalence class seems to contain 4 DAGs (which can be seen as a shift between the curves). This graph hints, that a big challenge in learning Bayesian Networks is **structure learning**.

one DAG encodes the same pdf as a BN over a different DAG. These BNs are then called distribution equivalent. [1] showed, that for gaussian distributions and multinomial sampling, these two equivalence statements are identical. In general, one can define **likelihood equivalence**, which implies that observations cannot help to discriminate two equivalent network structures. We have seen already in Figure 1 that, unfortunately, the space of equivalence classes is not much smaller (about a factor of 4) than the DAG space.

## 8.1 Parameters

Heckerman showed, that under a technical assumption (parameter independence, parameter modularity and likelihood equivalence), the Dirichlet distribution with some constraints on the hyperparameters is the most sensible prior distribution.

## 8.2 Structures

Oftentimes, uniform priors for the possible or sensible network structures are used.

# 9 Search Methods

It can be shown, that learning of BNs on discrete nodes which may have  $k \geq 3$  parents is NP-Hard ([2]), even if we have perfect knowledge of all conditional independence relations (independence oracle).

Since there is no hope of finding exact solutions to the learning problem, people usually use heuristics and try to prove correctness if many samples are available (asymptotic). There are two main categories of search algorithms:

## 9.1 Search and Score Methods

We search for the BN, that achieves the highest score (e.g. BIC, see section on Model Selection). Usually, these algorithms work in a greedy way, so they tend to get stuck in local maxima. To avoid this, using random restarts is a good idea. Separable search criterions are used

$$C(S, D) = \prod_{i=1}^n c(x_i, pa_i, D_i)$$

i.e., the updating of the score can be done very quickly. Probably the most renown algorithm of this kind is the **Greedy Equivalence Search** algorithm by [1]. It has been shown, that this algorithm (in the large sample limit) finds the correct equivalence class of the BN.

## 9.2 Constraint based methods

A completely different approach to search-and-score methods was proposed by [6]. They suggest the **PC-algorithm**, which starts with a complete graph and recursively deletes edges based on conditional independence tests. They showed, that given an independence oracle, this algorithm produces the correct structure. [5] showed for gaussian distributions, that the algorithm (in the large sample limit) finds the correct (structure) equivalence class of the BN when estimating conditional independencies from data. (The following is not yet waterproof but seems plausible:) If the underlying DAG is finite, this also holds for multinomial sampling. The parameters have to be learnt separately after this structure learning procedure.

## 10 Learning Causal Relationships

The **causal Markov condition** says: Let  $C$  be a causal graph for  $X$ ; then  $C$  is also a Bayesian-network structure for the joint probability distribution  $X$ . Given this property, causal relationships can be inferred from conditional (in)dependence information.

## References

- [1] D.M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.
- [2] D.M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.
- [3] S.B. Gillispie and M.D. Perlman. Enumerating markov equivalence classes of acyclic digraph models. In M. Goldszmidt, J. Breese, and D. Koller, editors, *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 171–177. Morgan Kaufmann, 2001.
- [4] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Research, 1995.
- [5] M. Kalisch and P. Buehlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. Technical report, ETH Zurich, 2005.
- [6] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. The MIT Press, 2000.