An Empirical Study of Efficiency and Accuracy of Probabilistic Graphical Models

Jens Dalgaard Nielsen and Manfred Jaeger
Institut for Datalogi, Aalborg Universitet
Frederik Bajers Vej 7, 9220 Aalborg Ø, Denmark

Abstract

In this paper we compare Naïve Bayes (NB) models, general Bayes Net (BN) models and Probabilistic Decision Graph (PDG) models w.r.t. accuracy and efficiency. As the basis for our analysis we use graphs of size vs. likelihood that show the theoretical capabilities of the models. We also measure accuracy and efficiency empirically by running exact inference algorithms on randomly generated queries. Our analysis supports previous results by showing good accuracy for NB models compared to both BN and PDG models. However, our results also shows that the advantage of the low complexity inference provided by NB models is not as significant as assessed in a previous study.

1 Introduction

Probabilistic graphical models (PGMs) have been applied extensively in machine learning and data mining research, and many studies have been dedicated to the development of algorithms for learning PGMs from data. Automatically learned PGMs are typically used for inference, and therefore efficiency and accuracy of the PGM w.r.t. inference are of interest when evaluating a learned model.

Among some of the most commonly used PGMs are the general Bayesian Network model (BN) and the Naïve Bayes model (NB). The BN model efficiently represents a joint probability distribution over a domain of discrete random variables by a factorization into independent local distributions. The NB model contains a number of *components* and models each discrete random variable as independent of all other variables within each component; it represents a joint probability distribution over all variables as a mixture of the component marginals.Exact inference has linear time complexity in the size of the model when using NB models. For the general BN model exact inference is NP-hard (Cooper, 1987), and known algorithms for exact inference have (worst case) complexity exponential in the size of the model (e.g. the junction tree algorithm (Lauritzen and Spiegelhalter, 1988)). Inference in BNs is therefore often solved approximately using loopy belief propagation (Yedidia et al., 2000) or sampling methods (Neal, 1993). A drawback common to both of these methods is the fact that they may not converge in reasonable time, and if stopped prematurely, the approximation can be very bad.

Model-selection algorithms for learning PGMs typically uses some conventional score-metric, searching for a model that optimizes the metric. Penalized likelihood metrics like BIC, AIC and MDL (Hastie et al., 2001) are a weighted sum of model accuracy and size. When the learned model is to be used for general inference, including a measure for inference complexity into the metric is relevant. Neither BIC, AIC nor MDL explicitly takes inference complexity into account when assessing a given model. Recently, several authors have independently emphasized the importance of considering inference complexity when applying learning in a real domain.

Beygelzimer and Rish (2003) investigate the tradeoff between model accuracy and efficiency. They only consider BN models for a given target distribution (in a learning setting, the target distribution is the empirical distribution defined by the data; more generally, the target distribution could be any distribution one wants to represent). For BNs *treewidth* is an adequate efficiency measure (defined as $k-1$, where k is the size of the largest clique in an optimal junction tree). Tradeoff curves are introduced that plot treewidth against the best possible accuracy achievable with a given treewidth. These tradeoff curves can be used to investigate the *approximability* of a target distribution, where approximability is to be understood as the ability to represent close approximations by small and efficient models.

As an example for a distribution not approximable in this sense, Beygelzimer and Rish (2003) mention the parity distribution, which represents the parity function on n binary inputs. An accurate representation of this distribution requires a BN of treewidth $n-1$, and any BN with a smaller treewidth can approximate the parity distribution only as well as the empty network.

The non-approximability of the parity distribution (and hence the impossibility of accurate models supporting efficient inference) only holds under the restriction to BN models with nodes corresponding exactly to the n input bits. The use of other PGMs, or the use of latent variables in a BN representation, can still lead to accurate and computationally efficient representations of the parity distribution.

Motivated by some distributions refusal to be efficiently approximated by BN models, the PGM language of probabilistic decision graph (PDG) models was developed (Jaeger, 2004). In particular, the parity distribution is representable by a PDG that has inference complexity linear in n. In a recent study an empirical analysis of the approximations offered by BN and PDG models learned from real-world data was conducted (Jaeger et al., 2006). Similar to the tradeoff curves of (Beygelzimer and Rish, 2003), Jaeger et al. (2006) used graphs showing likelihood of data vs. size of the model for the analysis of accuracy vs. complexity. The comparison of PDGs vs. BNs did not produce a clear winner, and the main lesson was that the models offer surprisingly similar tradeoffs when learned from real data.

Also motivated by considerations of model accuracy and efficiency, Lowd and Domingos (2005) in a recent study compared NB and BN models. NB models are a very special kind of latent class model, and can potentially offer accuracy-efficiency tradeoff behaviors that for some distributions differ from those provided by standard BN representation (although NBs do not include the latent class models that allow an efficient representation of the parity distribution). Lowd and Domingos (2005) determine inference complexity empirically by measuring inference times on randomly generated queries. The inferences are computed exactly for NB models, but for BN models approximate methods where used (Gibbs sampling (Neal, 1993) and loopy belief propagation (Yedidia et al., 2000)). Lowd and Domingos (2005) concludes that NB models offer approximations that are as accurate as those offered by BN models, but in terms of inference complexity the NB models are reported to be orders of magnitude faster than BN models.

Our present paper extends these previous works in two ways. First, we conduct a comparative analysis of accuracy vs. efficiency tradeoffs for three type of PGMs: BN, NB and PDG models. Our results show that in spite of theoretical differences BN, NB and PDG models perform surprisingly similar when learned from real data, and no single model is consistently superior. Second, we investigate the theoretical and empirical efficiency of exact inference for all models. This analysis somewhat differs from the analysis in (Lowd and Domingos, 2005), where only approximate inference was considered for BNs. The latter approach can lead to somewhat unfavorable results for BNs, because approximate inference can be much slower than exact inference for models still amenable to exact inference. Our results show that while NB models are still very competitive w.r.t. accuracy, exact inference in BN models is often tractable and differences in empirically measured run-times are typically not significant.

2 Probabilistic Graphical Models

In this section we introduce the three types of models that we will use in our experiments; the general Bayesian Network (BN), the Naïve Bayesian Network (NB) and the Probabilistic Decision Graphs (PDG).

2.1 Bayesian Network Models

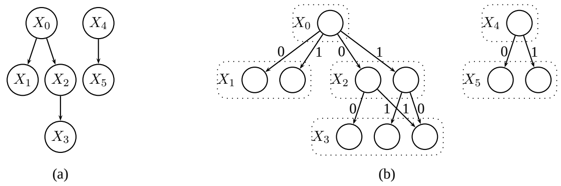
BNs (Jensen, 2001; Pearl, 1988) is a class of probabilistic graphical models that represent a joint probability distribution over a domain $\mathbf{X}$ of discrete random variables through a factorization of independent local distributions or factors. The structure of a BN is a directed acyclic graph (DAG) $G = (\mathbf{V}, \mathbf{E})$ of nodes $\mathbf{V}$ and directed edges $\mathbf{E}$. Each random variable $X_i \in \mathbf{X}$ is represented by a node $V_i \in \mathbf{V}$, and the factorization $\prod_{X_i \in \mathbf{X}} P(X_i | pa_G(X_i))$ (where $pa_G(X_i)$ is the set of random variables represented by parents of node V_i in DAG G) defines the full joint probability distribution $P(\mathbf{X})$ represented by the BN model. By *size* of a BN we understand the size of the *representation*, i.e. the number of independent parameters. Exact inference is usually performed by first constructing a junction tree from the BN. Inference is then solvable in time linear in the size of the junction tree (Lauritzen and Spiegelhalter, 1988), which may be exponential in the size of the BN from which it was constructed.

2.2 Naïve Bayes Models

The NB model represents a joint probability distribution over domain $\mathbf{X}$ by introducing an unobserved, latent variable C. Each state of C is referred to as a *component*, and conditioned on C, each variable $X_i \in \mathbf{X}$ is assumed to be independent of all other variables in $\mathbf{X}$. This yields the simple factorization: $P(\mathbf{X}, C) = P(C) \prod_{X_i \in \mathbf{X}} P(X_i | C)$. Exact inference is computable in time linear in the number of parameters, so the representation size of the NB model is a theoretical measure of the inference complexity.

2.3 Probabilistic Decision Graph Models

PDGs is a fairly new language for probabilistic graphical modeling (Jaeger, 2004; Bozga and Maler, 1999). As BNs and NBs, PDGs represent a joint probability distribution over a domain of discrete random variables $\mathbf{X}$ through a factorization of local distributions. However, the structure of the factorization defined by a PDG is not based on a variable level independence model but on a certain kind of context specific independencies among the variables. A PDG can be seen as a two-layer structure, 1) a forest of tree-structures over all members of $\mathbf{X}$, and 2) a set of rooted DAG structures over *parameter* nodes, each holding a local distribution over one random variable. Figure 1(a) shows a forest F of tree-structures over binary variables $\mathbf{X} = \{X_0, X_1 \ldots, X_5\}$, and figure 1(b) shows an example of a PDG structure based on F. For a complete semantics of the PDG model and algorithms for exact inference with linear complexity in the size of the model, the reader is referred to (Jaeger, 2004).



(a)

(b)

Figure 1: Example PDG. Subfigure (a) shows the a forest-structure F over binary 5 variables, and (b) shows a full PDG structure based on F.

3 Elements of the Analysis

The goal of our analysis is to investigate the quality of PGMs learned from real data w.r.t. accuracy and inference efficiency. The appropriate notion of accuracy depends on the intended tasks for the model. Following (Jaeger et al., 2006; Lowd and Domingos, 2005; Beygelzimer and Rish, 2003) we use log-likelihood of the model given the data ($L(M, D)$) as a "global" measure of accuracy. Log-likelihood score is essentially equivalent to cross-entropy (CE)[1] between the empirical distribution P^D and the distribution P^M represented in model M:

$$CE(P^D, P^M) = -H(P^D) - \frac{1}{|D|} L(M, D), \quad (1)$$

where $H(\cdot)$ is the entropy function. Observe that when $CE(P^D, P^M) = 0$ (when P^D and P^M are equal), then $L(M, D) = -|D| \cdot H(P^D)$. Thus, data entropy is an upper bound on the log-likelihood.

3.1 Theoretical complexity vs. accuracy: SL-curves

We use SL-curves (Jaeger et al., 2006) in our analysis of the theoretical performance of each PGM language. SL-curves are plots of size vs. likelihood. The size of model here has to be the *effective size* for inference. For NB and PDG models this is the size of the model itself. For BN models it is the size of the junction tree constructed for inference. Size could be any model complexity parameter, such that inference has linear time complexity in this parameter.

[1] CE is sometimes also referred to as Kullbach-Liebler (KL) divergence or relative entropy (Cover and Thomas, 1991).

3.2 Empirical complexity and accuracy

The size measure used in the SL-curves described in section 3.1 measures inference complexity only up to a linear factor. Following Lowd and Domingos (2005), we estimate the complexity of exact inference also empirically by measuring execution times for random queries. A query for model M is solved by computing a conditional probability $P^M(\mathbf{Q} = \mathbf{q}|\mathbf{E} = \mathbf{e})$, where $\mathbf{Q}, \mathbf{E}$ are disjoint subsets of $\mathbf{X}$, and $\mathbf{q}, \mathbf{e}$ are instantiations of $\mathbf{Q}$, respectively $\mathbf{E}$. Queries are randomly generated as follows: first a random pair $\langle \mathbf{Q}_i, \mathbf{E}_i \rangle$ of variable subsets is drawn from $\mathbf{X}$. Then, an instance d_i is randomly drawn from the test data. The random query then is $P^M(\mathbf{Q} = d_i[\mathbf{Q}_i]|\mathbf{E} = d_i[\mathbf{E}_i])$, where $d_i[\mathbf{Q}_i], d_i[\mathbf{E}_i]$ are the instantiations of $\mathbf{Q}$, respectively $\mathbf{E}$ in d_i. The empirical complexity is simply the average execution time for random queries. The empirical accuracy is measured by averaging *log*$P^M(\mathbf{Q} = d_i[\mathbf{Q}_i]|\mathbf{E} = d_i[\mathbf{E}_i])$ over the random queries. Compared to the global accuracy measure $L(M, D)$, this can be understood as a measure for "local" accuracy, i.e. restricted to specific conditional and marginal distributions of P^M.

4 Learning

In this section we briefly describe the algorithms we use for learning each type of PGM from data. For our analysis we need to learn a range of models with different efficiency vs. accuracy tradeoffs. For *score based* learning a general λ-score will be used (Jaeger et al., 2006):

$$S_\lambda(M, D) = \lambda \cdot L(M, D) - (1 - \lambda)|M|, \qquad (2)$$

where $0 < \lambda < 1$, and $|M|$ is the size of model M. Equation (2) is a general score metric, as it becomes equivalent to common metrics as BIC, AIC and MDL for specific settings of λ (e.g. BIC corresponds to (2) with $\lambda = \frac{log|D|}{2+\log|D|}$ and AIC corresponds to (2) with $\lambda = \frac{1}{2}$). By optimizing models for different settings of λ we get a range of models offering different tradeoffs between size and accuracy.

4.1 Learning Bayesian Networks

We use the KES algorithm for learning BN models (Nielsen et al., 2003). KES performs model-selection in the space of equivalence classes of BN structures using a semi-greedy heuristic. A parameter $k \in [0 \ldots 1]$ controls the level of greediness, where a setting of 0 is maximally stochastic and 1 is maximally greedy. The algorithm is based on the GES algorithm (Chickering, 2002). For producing a range of different models, we configured KES to use (2) with varying setting of λ as score metric.

The algorithm scores a BN by its own size, not the size of its junction tree. In some cases this leads to learning of models whose junction tree turns out to be intractably large. Such models are not included in our experimental results of section 6. Note that this is not so much symptomatic of a problem with BNs, but of not scoring BNs directly in terms of the size of their junction trees in the learning procedure.

4.2 Learning Naïve Bayes Net models

For learning NB models we have implemented a version of the NBE algorithm (Lowd and Domingos, 2005) for learning NB models. As the structure of NB models is fixed, the task reduces to learning the number of states in the latent variable C, and the parameters of the model. Learning in the presence of the latent components is done by standard Expectation Maximization (EM) approach, following (Lowd and Domingos, 2005; Karciauskas et al., 2004). Learning a range of models is done by incrementally increasing the number of states of C, and outputting the model learned for each cardinality. In this way we obtain a range of models that offer different complexity vs. accuracy tradeoffs. Note that no score function here is required.

4.3 Learning Probabilistic Decision Graphs

Learning of PDGs is done using the model-selection algorithm presented in (Jaeger et al., 2006). Using a score metric the algorithm searches for a structure that is optimal w.r.t. the score. Again we use (2) as score metric, and by varying the setting of λ we produce a range of models with different tradeoff in efficiency vs. accuracy.

5 Experiments

We have produced SL-curves and empirically measured inference times and accuracy, on 5 different datasets (see table 1) publicly available at the UCI repository[2]. These five datasets are a representative sample of the 50 datasets used in the extensive study by Lowd and Domingos (2005).

We used the same versions of the datasets as used by Lowd and Domingos (2005). Specifically, the partitioning into training (90%) and test (10%) sets was the same, continuous variables were discretized into five equal frequency bins, and missing values were interpreted as special states of the variables.

Table 1: Datasets used for experiments.

Dataset	Vars	train	test
Poisonous Mushroom	23	7337	787
King, Rook vs. King	7	25188	2868
Pageblocks	11	4482	574
Abalone	9	3758	419
Image Segmentation	17	2047	263

For measuring the empirical efficiency and accuracy, we generated random queries as described in 3.2. consisting of 1 to 5 query variables $\mathbf{Q}$ and 0 to 5 evidence variables $\mathbf{E}$.

For inference in BN and NB models we used the junction-tree algorithm implemented in the inference engine in Hugin[3] through the Hugin Java API. For inference in PDGs, the method described in (Jaeger, 2004) was implemented in Java. Inferences for BNs were performed on a standard laptop, 1.6GHz Pentium CPU with 512Mb RAM running Linux. PDG inference were performed on a Sun Fire280R, 900Mhz SPARC CPU with 4Gb RAM running Solaris 9.

6 Results

Table 2 shows SL-curves for each dataset and PGM, both for train (left column) and test sets (right column). The circle, triangle and square in each plot marks the NB, BN and PDG model respectively that maximizes BIC-score.

Lowd and Domingos (2005) based their comparison on single BN and NB models. The NB models were selected by maximizing likelihood on a hold-out set. Crosses × in the right column of table 2 indicate the size-likelihood values obtained by the models reported in (Lowd and Domingos, 2005).

The first observation we make from table 2 is that no single model language consistently dominates the others. The plots in the left column shows that BN models have the lowest log-likelihood measured on training data consistently for models larger than some small threshold. For Abalone and Image Segmentation, this characteristic is mitigated in the plots for the test-data, where especially PDGs seem to overfit the training-data and accordingly receives low log-likelihood score on the test-data.

The overall picture in table 2 is that in terms of accuracy, BNs and NBs are often quite similar (King, Rook vs. King is the only exception). This is consistent with what Lowd and Domingos (2005) have found. However, Lowd and Domingos (2005) reported big differences in inference complexity when comparing exact inference in NB to approximate methods in BNs. We do not observe this tendency when considering exact inference for both BNs and NBs. Our results show that we can learn BNs that are within reach of exact inference methods, and that the theoretical inference complexity as measured by effective model size mostly is similar for a given accuracy level for all three PGM languages.

Effective model size measures actual inference time only up to a linear factor. In order to determine whether there possibly are huge (orders of magnitude) differences in these linear factors, we measure the actual inference time on our random queries. The left column of table 3 shows the average inference time for 1000 random queries with 4 query and 3 evidence variables (results for other numbers of query and evidence variables were very similar). We observe that this empirical complexity behaves almost indistinguishably for BN and NB models. This is not surprising, since both models use the Hugin inference engine. The results do show, however, that the different structures of the junction trees for BN and NB models do not have a significant impact on runtime. The linear factor for PDG inference in these experiments is about 4 times larger than that for BN/NB inference. [4] Seeing

[2] http://www.ics.uci.edu/~mlearn

[3] http://www.hugin.com

[4] This factor has to be viewed with caution, since PDG inference was run on a machine with a slower CPU but more main memory. When running PDG inference on the same machine as NB/BN inference, we observed overall a similar performance, but more deviations from a strictly linear behavior (in table 3 still visible to some degree for the Mushroom and Abalone data). These deviations seem mostly attributable to

Table 2: SL-curves for train-sets (left column) and test-sets (right column). Circles, squares and triangles marks the NB, BN, and PDG models that maximizes BIC-score. The crosses marks the NB models reported by Lowd and Domingos (2005). $-H(D)$ (minus data-entropy) is plotted as a horizontal line.

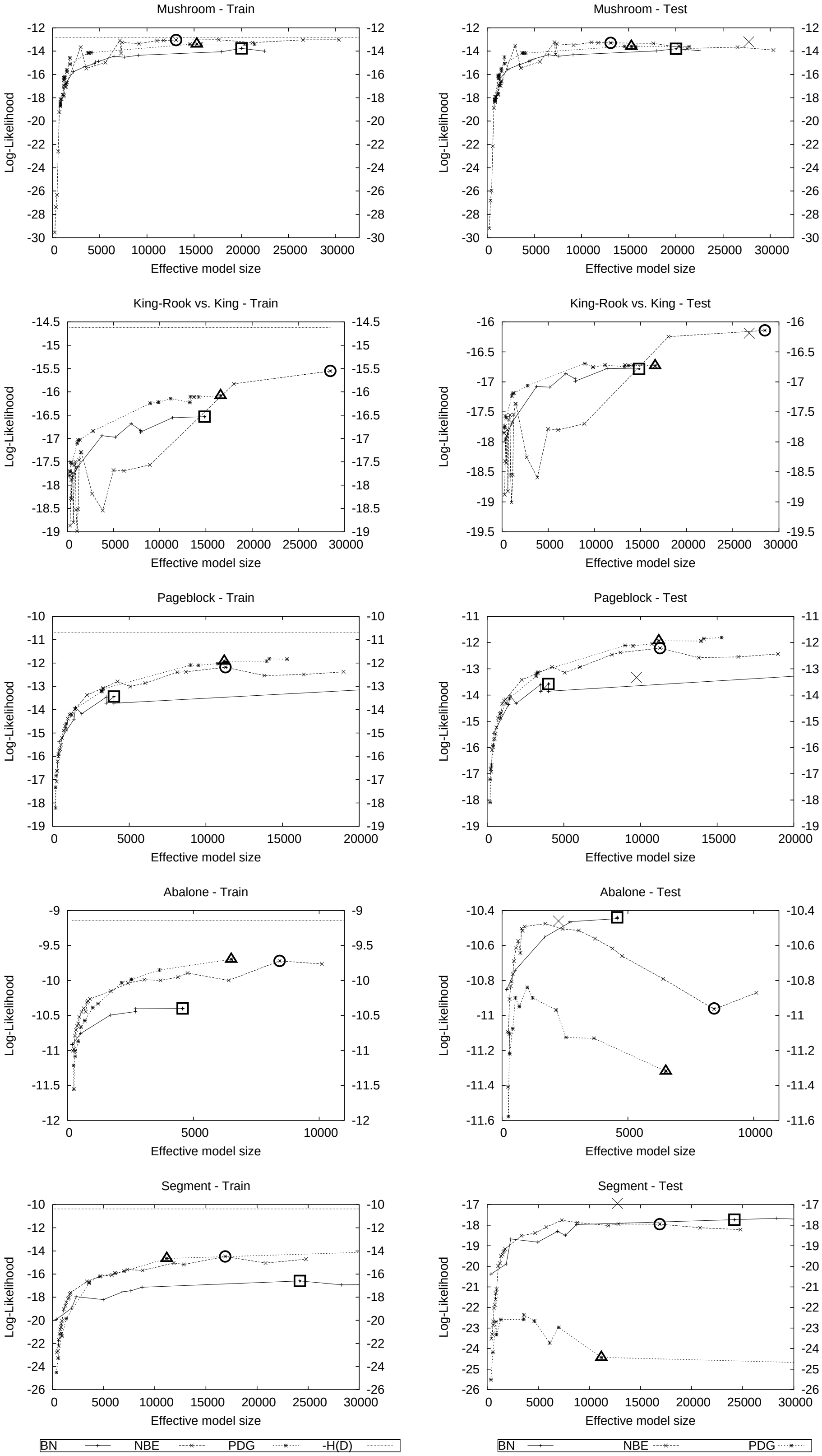
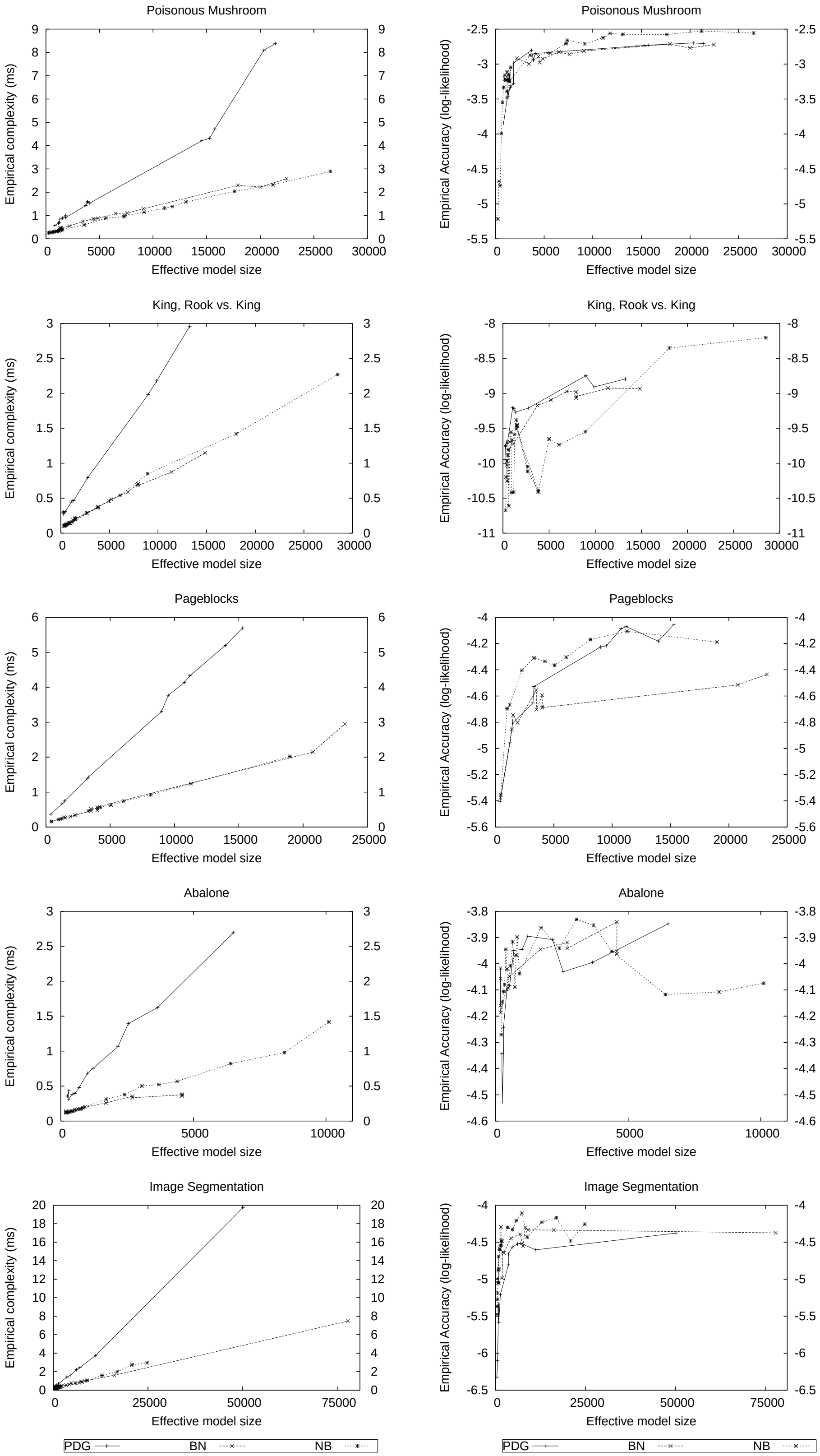


Table 3: Empirical efficiency (left column) and accuracy (right column).



that we use a proof-of-concept prototype Java implementation for PDGs, and the commercial Hugin inference engine for BNs and NBs, this indicates that PDGs are competitive in practice, not only according to theoretical complexity analyses.

The right column in table 3 shows the empirical (local) accuracy obtained for 1000 random queries with 4 query and 3 evidence variables. Overall, the results are consistent with the global accuracy on the test data (table 2, right column). The absolute log-likelihood values are naturally higher, since the queries only refer to a subset of the variables. The differences observed for the different PGMs in table 2 can also be seen in table 3, though the discrepancies tend to become less pronounced on the random queries as on global likelihood (particularly for PDGs in the image segmentation data). One possible explanation for this is that low global likelihood scores are mostly due to a few test cases whose joint instantiation of the variables are given low probability by a model, and that these isolated low-probability configurations are seldom met with in the random queries.

7 Conclusion

Motivated by several previous, independent studies on the tradeoff between model accuracy and efficiency in different PGM languages, we have investigated the performance of BN, NB, and PDG models. Our main findings are: 1) In contrast to potentially widely different performance on artificial examples (e.g. the parity distribution), we observe a relatively uniform behavior of all three languages on real-life data. 2) Our results confirm the conclusions of Lowd and Domingos (2005) that the NB model is a viable alternative to the BN model for general purpose probabilistic modeling and inference. However, the order-of-magnitude advantages in inference complexity could not be confirmed when comparing exact inference methods for both types of models. 3) Previous theoretical complexity analyses for inference in PDG models now have been complemented with empirical results showing also the practical competitiveness of PDGs.

the memory management in the Java runtime environment.

Acknowledgment

We thank Daniel Lowd for providing the preprocessed datasets we used in our experiments.

References

Beygelzimer, A. and Rish, I.: 2003, Approximability of probability distributions, *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems*.

Bozga, M. and Maler, O.: 1999, On the representation of probabilities over structured domains, *Proceedings of the 11th International Conference on Computer Aided Verification*, Springer-Verlag.

Chickering, D. M.: 2002, Optimal structure identification with greedy search, *Journal of Machine Learning Research* **3**, 507–554.

Cooper, G. F.: 1987, Probabilistic inference using belief networks is np-hard, *Technical report*, Knowledge Systems Laboratory, Stanford University.

Cover, T. M. and Thomas, J.: 1991, *Elements of information theory*, Wiley.

Hastie, T., Tibshirani, R. and Friedman, J.: 2001, *The Elements of Statistical Learning*, Springer-Verlag.

Jaeger, M.: 2004, Probabilistic decision graphs - combining verification and ai techniques for probabilistic inference, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **12**, 19–42.

Jaeger, M., Nielsen, J. D. and Silander, T.: 2006, Learning probabilistic decision graphs, *International Journal of Approximate Reasoning* **42**(1-2), 84–100.

Jensen, F. V.: 2001, *Bayesian Networks and Decision Graphs*, Springer.

Karciauskas, G., Kočka, T., Jensen, F. V., naga, P. L. and Lozano, J. A.: 2004, Learning of latent class models by splitting and merging components, *pgm-04*, pp. 137–144.

Lauritzen, S. L. and Spiegelhalter, D. J.: 1988, Local computations with probabilities on graphical structures and their application to expert systems, *Journal of the Royal Statistical Society* **50**(2), 157–224.

Lowd, D. and Domingos, P.: 2005, Naive bayes models for probability estimation, *Proceedings of the Twentysecond International Conference on Machine Learning (ICML)*.

Neal, R. M.: 1993, Probabilistic inference using markov chain monte carlo methods, *Technical report*, Department of Computer Science, University of Toronto.

Nielsen, J. D., Kočka, T. and Peña, J. M.: 2003, On local optima in learning bayesian networks, *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, Acapulco, Mexico, pp. 435–442.

Pearl, J.: 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers.

Yedidia, J. S., Freeman, W. T. and Weiss, Y.: 2000, Generalized belief propagation, *Technical report*, MITSUBISHI ELECTRIC RESEARCH LABORATORY.