Model Selection Criteria for Learning Belief Nets: An Empirical Comparison

Tim Van Allen VANALLEN@CS.UALBERTA.CA
Russ Greiner GREINER@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2H1 Canada

Abstract

We are interested in the problem of learning the dependency structure of a belief net, which involves a trade-off between simplicity and goodness of fit to the training data. We describe the results of an empirical comparison of three standard model selection criteria — viz., a Minimum Description Length criterion (MDL), Akaike's Information Criterion (AIC) and a Cross-Validation criterion — applied to this problem. Our results suggest that AIC and Cross-Validation are both good criteria for avoiding overfitting, but MDL does not work well in this context.

1. Introduction

In learning a model of a data-generating process from a random sample, a fundamental problem is finding the right balance between the complexity of the model and its goodness of fit to the training data. A more complex model can usually achieve a closer fit to the training data, but this may be because the model reflects not just significant regularities in the data but also minor variations due to random sampling that give no information about the underlying process. The true error of a model comes from two sources: (1) the bias in the structure of the model, which prevents it from representing the underlying process, and (2) the variance of estimating the parameters from a limited sample. More complex model structures have less bias but their parameters usually have higher variances. Finding the best model, based on a random sample, requires finding a balance between the competing objectives of reducing the bias of the model and reducing its variance for the sample in question.

Algorithms for model-learning generally divide into two components: a search algorithm, for finding the best model in a given class, and a criterion for comparing models. Handling the bias-variance trade-off is primarily a matter of choosing the criterion to be applied. One approach is to add a complexity penalty to the training error so that more complex models have to fit the data considerably better than smaller models in order to outscore them. Two standard criteria are *Minimum Description Length* (MDL) (Rissanen, 1989) and *Akaike's Information Criterion* (AIC) (Bozdogan, 1987). Another approach is to use only part of the sample to set the parameters, and use the rest of the sample to get an unbiased estimate of the true error. This latter approach is called *Cross-Validation* (Stone, 1974).

In this paper, we compare these three model selection criteria, in the context of learning belief nets (defined in Section 2.1). We had two goals in carrying out this research. The first was to find a good criterion for learning belief net structures. The second was to understand the issues and trade-offs involved in model-selection. Consequently, we have both prescriptive and descriptive results. We hypothesized that MDL would be less effective than either Cross-Validation or AIC, and in fact we found this to be the case: using the MDL criterion to select a model results in *drastically underfitting* the data in many cases. Therefore, we endorse the use of AIC and Cross-Validation as model selection criteria for learning belief nets, and caution against the use of MDL. Our descriptive results explain the reasons for MDL's poor performance, and illustrate some of the issues and trade-offs involved in model selection.

The outline of this paper is as follows. The rest of this section discusses previous research related to this topic. Section 2 gives background information on belief nets, model selection and the criteria under consideration. Section 3 describes the experimental design we used. Section 4 presents the results of our experiments, using both case studies and comprehensive tests to establish our claims. Section 5 contains a discussion of the results and the issues involved in model selection. It also responds to some possible criticisms of our research.

1.1 Related Work

There is a considerable literature on learning belief networks, and in particular, on learning their structure; see Heckerman (1995) for a detailed overview of the subject.Note that many researchers, including Lam and Bacchus (1994), Suzuki (1996), and Friedman and Goldszmidt (1996) explicitly use the MDL criterion (or something close to it) to evaluate candidate networks. Our work suggests a problem with this MDL framework. Friedman and Yakhini (1996) carry out an analysis of the sample requirements for various complexity penalty approaches to belief net learning. While that work also addresses suitability of various selection criteria, its analysis is theoretical and based on asymptotic behaviour; by contrast we are empirically investigating small sample behaviour over a different class of criteria.

Linhart and Zucchini (1986) provide an overview of the general problem of model selection, covering AIC and Cross-Validation, but not MDL. Rissanen (1989) gives a detailed development of the *Minimum Description Length Principle*, which is the information-theoretic view of induction that the MDL criterion is based on. Bozdogan (1987) provides an easy to read derivation of AIC and a discussion of its use. Kearns *et al* (2000) describe an experiment related to our own, covering a similar range of criteria, but applied to the problem of function learning rather than distribution learning. Their study also found problematic behaviour for MDL.

2. Background

Notation: We use capital letters to denote random variables, and use expressions such as $\mathrm{P}(X=x)$ and $\mathrm{P}(X=x \mid Y=y)$ to denote probabilities. We use expressions such as $\mathrm{P}(X)$ and $\mathrm{P}(X \mid Y=y)$ to represent distributions, and $\mathrm{P}(X \mid Y)$ to represent a set of conditional distributions. In general, unbound variables are implicitly universally quantified over their domains, so an "equation" such as $\mathrm{P}(X) = \mathrm{P}(Y)\,\mathrm{P}(X \mid Y)$ means $\forall x\ \forall y\ [\mathrm{P}(X=x) = \mathrm{P}(Y=y)\mathrm{P}(X=x \mid Y=y)]$. Functions of random variables are also random variables.

In what follows, we will assume that we are dealing with learning a network over n discrete variables, $\{X_1, X_2 \ldots X_n\}$, given a sample s of size m, drawn from the distribution $\mathrm{P}(S \mid T=t)$, where T is a random variable ranging over the family of models under consideration, and t is the true model. Note that all logs are to base 2.

2.1 Belief Networks

A belief network is a representation for a joint distribution over a set of random variables $X = \{X_1, X_2 \ldots X_n\}$. It consists of two parts: a dependency graph, and a set of conditional probability functions. The dependency graph is a directed acyclic graph, whose vertices are the random variables. Each variable is assumed to be conditionally independent of all other variables, given its parents and children in the graph. Thus, the joint probability distribution over all variables is given by the pseudo-equation:

$$\mathrm{P}(X_1, X_2 \ldots X_n) \quad = \quad \prod_{i=1}^{n} \mathrm{P}(X_i \mid \mathrm{Pa}(X_i)),$$

where $\mathrm{Pa}(X_i)$ is the vector of X_i's parents. We will restrict our attention to the case where these conditional probability functions are unrestricted — that is, where they are given by a complete table listing all joint assignments to $\langle X_i, \mathrm{Pa}(X_i) \rangle$. Hence, the number of parameters required to represent the function is exponential in the number of parents.

2.2 Model Selection

The negative log-likelihood is a standard measure of training error. It is given by:

$$\mathrm{DL}(s,h) \quad = \quad -\log \mathrm{P}(S=s \mid T=h),$$

for a sample s and a hypothesis h. We denote it DL because it is the description length of s, encoded using an optimal code based on the distribution given by h. When s is a sequence of i.i.d. instances $x_1, x_2, \ldots, x_m$, of X, then:

$$\mathrm{DL}(s,h) = -\sum_{i=1}^{m} \log \mathrm{P}(X = x_i \mid T=h).$$

KL-divergence (Kullback & Leibler, 1951) is a standard measure of error for distribution learning. If t is the "true" model, and h is a hypothesized model, the KL-divergence of h from t is given by:

$$\mathrm{KLD}(t \,\|\, h) = \sum_{x} \mathrm{P}(X=x \mid T=t) \log \frac{\mathrm{P}(X=x \mid T=t)}{\mathrm{P}(X=x \mid T=h)},$$

where x ranges over all possible assignments to X. It measures the expected cost of encoding instances from $\mathrm{P}(X \mid T=t)$ using a code based on $\mathrm{P}(X \mid T=h)$. Note that it can also be written as:

$$\mathrm{KLD}(t \,\|\, h) = \mathrm{E}[\mathrm{DL}(X,h)] - \mathrm{E}[\mathrm{DL}(X,t)],$$

where the expectations are taken under $P(X \mid T = t)$. The second term is the *entropy* of $P(X \mid T = t)$. As it does not depend on h, the first term alone is sufficient to compare models. We can form an estimate of this first term:

$$\text{info}(h; s) = \frac{1}{m}\text{DL}(s, h),$$

The problem, of course, is that if we use s to estimate the parameters of h, then h depends on s and so $\text{info}(h; s)$ is a biased estimator, tending to favour more complex models. In general, the more complex h is, the more it will be tuned to s, and the worse the bias in this estimate will be. Note that there are two different kinds of bias involved here: as networks become more complex, they become less representationally biased (*i.e.*, they can represent a larger class of distributions), but their parameters have higher variance under sampling, and so training error becomes *more* biased as an estimator of true error.

One solution to this problem is to partition s into two subsamples, a training sample and a validation sample. We can then use the training sample to fit the parameters, and the validation sample to estimate the discrepancy. This removes the bias in the estimate of $E[\text{DL}(X, h)]$, but increases the variance in the $E[\text{DL}(X, h)]$ criterion. One way to (seemingly) avoid the increase in variance is to partition s into k subsamples, and repeat the training process k times, each time reserving one subsample for validation, and afterward combining these estimates. The logical extreme is to divide the sample into m subsamples of 1 datum each. This family of methods goes under the generic name of Cross-Validation, being respectively called "simple", "k-fold", and "leave-one-out" Cross-Validation (Stone, 1974; Linhart & Zucchini, 1986). For our experiments, we used the simple version, dividing the sample into two equal size subsamples, one for training and one for validation.

$$\text{X}V(h;\ s) = \text{info}(h(s_1); s_2)$$

where s has been split into disjoint halves s_1 and s_2, and $h(s_1)$ is the hypothesis h instantiated using the instances s_1. Note that the final parameters of the chosen model will be estimated on the full data set s; it is only for model *selection* that we need to withhold data from the parameter estimation process (*i.e.*, use $h(s_1)$ rather than $h(s)$).

Another approach is to add a complexity penalty to the goodness of fit term, to counteract the bias introduced by overfitting to the data. The penalty may be a function of both the sample size and the number of parameters of the model. Except for sample size, it does not otherwise depend on the data. The problem with this approach is that any such penalty function cannot, in principle, remove the bias from the estimate, because the bias of this estimate depends on the the parameters of the true model. Since the complexity penalty in no way depends on the true distribution, it cannot exactly counteract the bias — it introduces a bias of its own.

There are several well-known penalty functions, each motivated by different theoretical considerations and each appropriate for a particular class of learning problems. The Minimum Description Length (MDL) criterion is based on an information-theoretic view of induction as data compression; see Rissanen (1989) for a detailed development. It is equivalent to the *Bayesian Information Criterion*, which was introduced originally by Schwarz (1978) and given a Bayesian interpretation. The information-theoretic interpretation of the MDL criterion is as the length of an encoding of the sample as a two part code. The basic idea is to use the model to define a code for the sample: encode the sample by first encoding the model, and then encoding the data using the code given by the model. If the model captures significant features of the data, this encoding will be considerably smaller than simply sending the sample as is. On the other hand, if the model represents too much about the sample, the encoding size will increase. This trade-off is similar to the bias-variance trade-off (Linhart & Zucchini, 1986). Our version differs from the standard form in that we have normalized everything by $1/m$ so we can compare it across sample sizes and with other criteria. Some low order terms (all positive) have been dropped as well — it will be seen that this has no *negative* impact on the criterion. Our MDL criterion is given by:

$$\text{MDL}(h; s) = \text{info}(h; s) + \frac{k \log m}{2m}$$

where k is the number of parameters of h.

Akaike's Information Criterion (AIC) comes from a different theoretical perspective. It is an explicit attempt to correct the overfitting bias. See Bozdogan (1987) for a derivation of the criterion. The complexity penalty is considerably smaller than MDL's. Our version of AIC is given by:

$$\text{AIC}(h; s) = \text{info}(h; s) + \frac{k \log e}{m},$$

where e is the base of the natural logarithm, and $\log e$ is simply a conversion from nats to bits.

3. Experimental Design

We wanted to observe the behaviour of these criteria while varying the training sample size and the true distribution. Because the space of network structures is huge for even a modest number of variables, a systematic exploration of that space was an unrealistic goal. Instead, we focussed on trajectories through that space; in particular, trajectories from the simplest to the most complex structure that pass through the true structure. In short, we repeated many experiments of the following form:

1. Generate all edges over n vertices.
2. Randomly order these edges.
3. Pick a number of edges, $d \in [0..\binom{n}{2}]$, for the true model.
4. Make the true structure from the first d edges in the randomized list.
5. Generate random probabilities for the parameters of the true model.
6. Generate samples of various sizes from the true model.
7. For $i = 0$ to $\binom{n}{2}$, construct a hypothesis structure with the first i edges in the randomized list. (We will later refer to the i-edge structure as h_i.)
8. For each hypothesis structure, and each criteria, evaluate the criteria based on the generated samples, and then fit the parameters.

We generated the parameters for the true model from different distributions; however, generally we used a uniform distribution over $(0, 1)$. We experimented with different values for n, and found that 10 was a sufficiently large number to give us interesting results; for larger numbers of variables the results simply scaled up without changing qualitatively. We used binary valued variables only; this also had no qualitative impact on results, but made the computation and analysis easier. We describe our results in the section that follows.

4. Results

Using the experimental apparatus described in the previous section, we could observe the behaviour of each criterion across a spectrum of complexities and a range of sample sizes. What we observed followed a remarkably consistent pattern, one which suggests caution in applying the MDL principle.

Figures 1-3 are “snapshots”, taken at different sample sizes, of the results for one particular true model. These graphs show the criteria compared across the

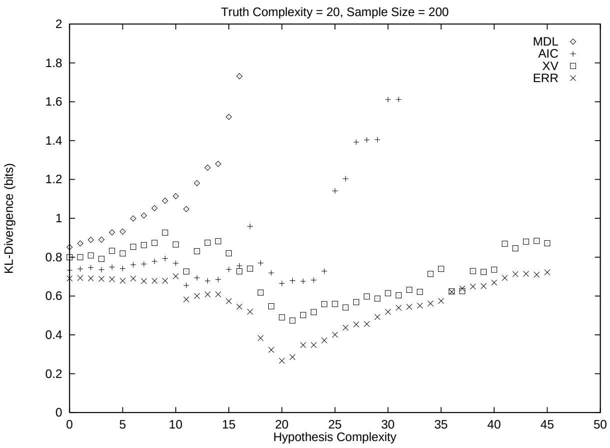

Figure 1. Case Study 1: $m = 200$.

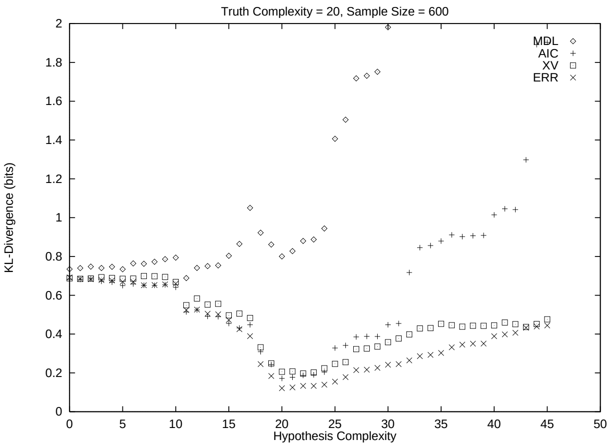

Figure 2. Case Study 1: $m = 600$.

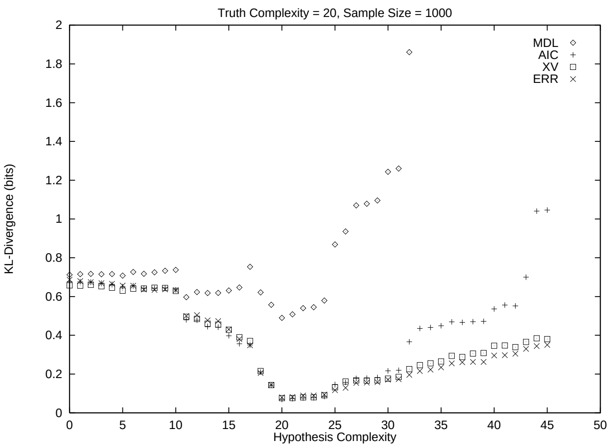

Figure 3. Case Study 1: $m = 1000$.

complexity spectrum (marked out in number of *dependencies*, not number of parameters) when they are evaluated on samples of size 200, 600 and 1000. Four values are plotted for each hypothesis structure: (ERR) the true error, which is the KL-divergence of the network with parameters estimated from the sample, (MDL) the MDL criterion, (AIC) the AIC criterion, and (XV) the Cross-Validation criterion. To scale everything, the true entropy of the distribution has been subtracted from each criterion. The true model is the one with 20 edges.

Given this set of hypotheses, an ideal learner using a criteria $\gamma(h; s)$ would pick a hypothesis with the lowest γ-value. So for the $m = 200$ graph, the MDL-based learner would pick h_0 (*i.e.*, the the 0-edge structure), the AIC-based learner would select either h_{11} or h_{20}, and the Cross-Validation-based would pick h_{21}. While all are wrong (recall the true structure is h_{20}) note that the structure returned by MDL, h_0, is the worst, in that its KL-divergence is 0.65, while the answer returned by AIC may have KL-divergence of either 0.60 (h_{11}) or 0.22 (h_{20}) and the answer for XV has KL-divergence of 0.23 (h_{21}). Here, MDL is clearly doing poorly. For $m = 600$ MDL picks h_{11} (KL of 0.55), AIC picks h_{20} (KL of 0.1) and Cross-Validation picks h_{22} (KL of 0.1); and for $m = 1000$, all three (correctly) pick h_{20}. In all cases, we see that Cross-Validation finds a structure that is close to optimal, while MDL does not, at least for small samples.

In general, a criterion that is well suited for optimizing a function is one that has the same general shape as the function, and in particular, has the same local and global minima. The graphs show that XV and AIC acquire the relevant properties of the ERR function on much smaller samples than does MDL. We have observed a threshold pattern here: often a small increase in sample size can make a large difference in the optimality of MDL's preferred structure. That is, before observing a critical quantity of data MDL will typically prefer very sub-optimal models (*i.e.*, models that are too small); on reaching that size, it returns the optimal model. This is because MDL has a high bias for simplicity, and this bias dominates its behaviour on smaller samples, in that the actual fit to the data has relatively little impact here. Note that the data is sufficient to find a good model — otherwise XV and AIC would not have found good models.

Figure 4 shows the complexity penalties of MDL $(k \log m)/(2m)$ and AIC $(k \log e)/m$, plotted against the actual amount of overfitting measured $(\mathrm{E}[\mathrm{DL}(X, h)] - \mathrm{info}(h; s)$ for each h), on a sample of size 400, for the same truth as Figures 1-3. You can

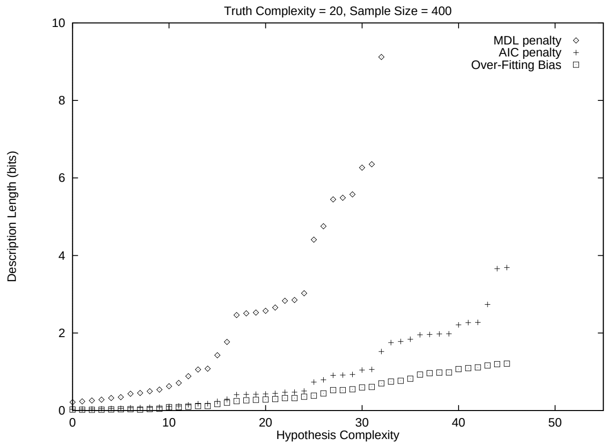

Figure 4. Case Study 1: Penalties vs. Overfit; $m = 400$.

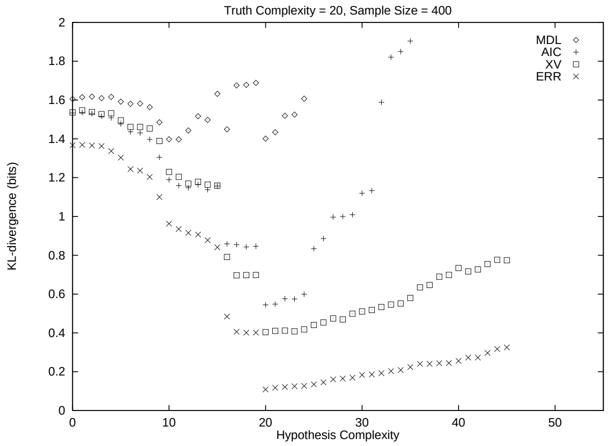

Figure 5. Case Study 2: A lower entropy truth; $m = 400$.

see that AIC does a reasonably good job of matching the overfitting, until the network complexity gets too high. The MDL penalty is much larger than than the amount of overfitting.

Figure 5 is a snapshot of the same experiment on a different truth; here we used the same true structure, but changed the parameters to be either high or low (0.9 or 0.1). This generation scheme tended to produce lower entropy distributions, which have more potential for data compression. We see the same pattern, albeit for a smaller sample size (400 is shown here). MDL is just beginning to overcome its bias, while Cross-Validation and AIC prefer the ideal structure. Note that Cross-Validation does not do such a great job of predicting *what* the true error of a model is — here it consistently overestimates — but it does a good job of predicting the *relative* error between models, which is sufficient for model selection.

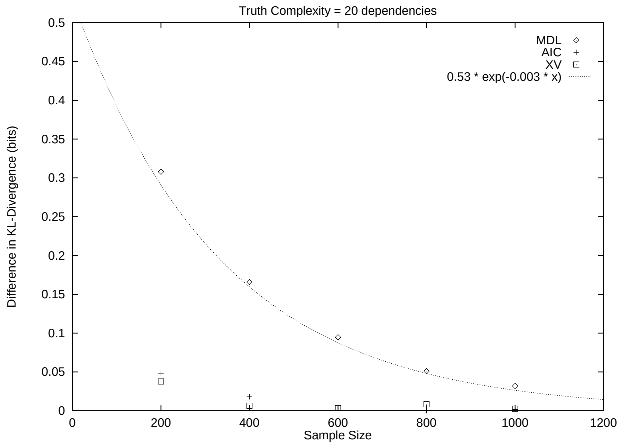

Figure 6. Comprehensive Study: Empirical Convergence Rates; $e = 20$.

Figure 6 shows the empirical convergence rates for truths with 20 dependencies, whose parameters were drawn from a uniform distribution. Each point represents an average of 30 experiments, where each experiment involved generating a new truth. The y-axis represents the difference in KL-divergence between the best-scoring model for the criterion and the model with the lowest true error. As the values go to zero, the criterion is converging on the ideal structure. Note that, while the curve created by plotting the averages is a smooth exponential curve (a best fit is shown for MDL), the actual behaviour of the criteria on any particular truth might not be smooth convergence. In many cases, MDL actually tends to shift almost immediately from preferring high error networks to preferring the ideal structure (as seen in Figures 3 and 4).

Table 1 summarizes the results of our comprehensive study. For each $\langle$sample-size, truth complexity$\rangle$ combination we carried out 30 experiments of the type described above (using a uniform distribution to generate the true parameters). For each experiment, for each criterion, we took the network that scored the best and subtracted its error from the lowest error attained by any network. We summarize the 30 values thus obtained by giving their mean and median. Note that, because we manipulated the number of dependencies, rather than the number of parameters directly, there was a considerable amount of variance in these values. This is because the number of parameters depends on the graph structure, not just on the number of dependencies, though it tends to increase exponentially as the number of dependencies increases. In fact, even sorting the networks into buckets based on the number of parameters did little to reduce the variance: the parameter values have a large effect on the relative difficulty of learning the distribution. We use a large font to indicate the "winners" in each cell, however, the differences are more important than distinguishing the best. Where MDL "won" (low left), for example, the other methods also did quite well in attaining low error; but where MDL did poorly (high right), it did *very* poorly relative to the other criteria.

4.1 The Real World

Our reviewers suggested complementing our random experiments with "real-world" problems. Therefore we carried out additional experiments with the *Alarm* and *Insurance* networks, which are commonly used in belief net studies. Both the Alarm and Insurance networks are sparse: the Alarm network has 37 variables but only 46 links, and the Insurance network has 27 variables and 52. What we observed on these networks essentially mirrored our results on random distributions. For each network we generated samples of size 200, 500 and 1000. Then we computed the three criteria under consideration for a range of network structures, including networks whose edges were a subset of the true network, and networks created by adding edges to the true network. Because of the large sample spaces for these distributions we did not exactly compute the true entropy and KL-divergences, but instead used a large sample (10,000 data) empirical approximation.

Each cell in the table below summarizes the results of 10 experiments of this kind. First we computed the scores for each network across the spectrum of complexity; while doing this we also computed the KL-divergence of each network from the true model. Next we found the best-scoring network for each criterion, and determined the difference between its KL-divergence and the lowest KL-divergence obtained by any network tested (this was not always the true network structure). Finally, we determined the minimum, median and maximum of these differences for each criteria over the 10 experiments. As we observed in our random experiments, the MDL score would lead to significant underfitting if used as a model selection criterion; AIC was considerably better but slightly less effective than Cross-Validation on small sample sizes.

5. Discussion

Our empirical results show that optimizing for the MDL criterion can be a risky strategy for learning belief net structures. While MDL does seem to work for sufficiently large samples, it can be arbitrarily worse for even slightly smaller samples; therefore there is no

Table 1. Comprehensive Study.

	$d = 0$			$d = 10$			$d = 20$			$d = 30$		
m	MDL	AIC	XV	MDL	AIC	XV	MDL	AIC	XV	MDL	AIC	XV
200												
μ	**0.0015**	0.0074	0.0210	0.0618	**0.0138**	0.0258	0.3079	0.0483	**0.0377**	0.4705	0.2008	**0.0477**
M	**0.0000**	**0.0000**	0.0117	0.0044	**0.0000**	0.0096	0.3167	**0.0031**	**0.0031**	0.4419	0.1861	**0.0275**
400												
μ	**0.0000**	0.0020	0.0050	0.0277	**0.0036**	0.0141	0.1658	0.0181	**0.0064**	0.4965	0.0864	**0.0231**
M	**0.0000**	**0.0000**	0.0014	**0.0000**	**0.0000**	0.0038	0.1332	**0.0000**	**0.0000**	0.4884	0.0521	**0.0000**
600												
μ	**0.0000**	0.0017	0.0016	0.0111	**0.0010**	0.0049	0.0946	**0.0008**	0.0033	0.3601	0.0589	**0.0143**
M	**0.0000**	**0.0000**	0.0001	**0.0000**	**0.0000**	**0.0000**	0.0461	**0.0000**	**0.0000**	0.3143	0.0031	**0.0000**
800												
μ	**0.0003**	0.0022	0.0032	0.0023	**0.0014**	0.0047	0.0510	**0.0001**	0.0084	0.3684	0.0294	**0.0020**
M	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	0.0006	**0.0000**	**0.0000**	**0.0000**	0.3408	**0.0000**	**0.0000**
1000												
μ	**0.0000**	0.0023	0.0027	**0.0013**	0.0023	0.0036	0.0319	**0.0016**	0.0027	0.3150	0.0232	**0.0032**
M	**0.0000**	**0.0000**	0.0003	**0.0000**	**0.0000**	**0.0000**	0.0160	**0.0000**	**0.0000**	0.3504	**0.0000**	**0.0000**

Table 2. Results for Insurance

Criterion	Min	Median	Max
	$m = 200$		
MDL	0.851738	3.097284	6.504755
AIC	0.238702	1.856821	3.755595
XV	0.000000	0.000000	0.238702
	$m = 500$		
MDL	0.681383	2.651849	4.677080
AIC	0.000000	0.000000	0.681383
XV	0.000000	0.000000	0.097622
	$m = 1000$		
MDL	0.000000	0.615123	2.103506
AIC	0.000000	0.000000	0.083442
XV	0.000000	0.000000	0.000000

Table 3. Results for Alarm

Criterion	Min	Median	Max
	$m = 200$		
MDL	0.489475	0.883749	1.033580
AIC	0.489475	0.876087	0.971317
XV	0.000000	0.086332	0.595293
	$m = 500$		
MDL	0.679741	1.226129	1.300100
AIC	0.000000	0.221338	0.875011
XV	0.000000	0.000538	0.145014
	$m = 1000$		
MDL	0.000000	0.474193	0.995653
AIC	0.000000	0.004413	0.210303
XV	0.000000	0.000000	0.008826

guarantee of graceful degradation. Furthermore, there is no way to know *a priori* whether MDL has sufficient data to be effective. By contrast, we found Cross-Validation to be a "safe bet", one which was never that bad. (Table 1 shows that Cross-Validation's average error never exceeded 0.1.) AIC's performance was in-between, but closer to Cross-Validation, in terms of its risk.

It is known that any complexity penalty has bias, and therefore will do better for some learning problems at the expense of others. Of course, a learner should use any available prior knowledge, and if that supports some complexity penalty such as AIC or MDL, then that criterion should be used. On the other hand, there are many cases where one has no prior knowledge; and Cross-Validation minimizes the worst-case loss without sacrificing too much in terms of average performance. (Table 1 shows it was the minimax over the three criteria.) This is consistent with the Cross-Validation's other name, "the jackknife" — *i.e.*, a jack of all trades but master of none.

We now address some possible criticisms of our findings. First, some might argue that the MDL criterion is not intended to optimize for true error, but simply to implement the MDL principle. Even if Cross-Validation prefers a more complex network that has lower true error, they might argue that it is not *justified* in doing so. However, anyone using MDL to optimize for description length should be aware that they may obtain very bad generalization error.

A related argument is that the truth tends to be simple, or that we have strong prior beliefs that it is, and therefore we should bias our model-selection algorithms in favour of simplicity. A prior expectation of simplicity can be incorporated into the training error term, however, using a Bayesian prior. It may also be reflected in search strategy or representational choice. It need not be part of the same mechanism used to handle overfitting.

Third, there are other possible MDL criteria, and perhaps we chose the wrong one for our task of learning belief nets. For example, our MDL criterion implicitly assumes that the parameters can vary independently, which is not the case. Note however that the AIC criterion was based on the same assumption, but it still performed fairly well. Moreover, the natural parameterization for many (perhaps most) learning domains exhibits some parameter-dependencies, which means the rigorous application of these asymptotic criteria creates technical difficulties. Cross-Validation does not have this problem. Note also that our results corroborate the results of Kearns et al. (2000) even though they were considering a different domain.

Fourth, it is sometimes argued that natural, or "real-world" problems have a special structure, and therefore results based on random experiments have little value. This is true primarily when prescriptive results are not complemented by descriptive results, however. We have attempted to show why the MDL criterion may lead to underfitting (a large simplicity bias that overwhelms the goodness of fit on small samples), and why it is a risky strategy (it exhibits phase-transition behaviour). Our experiments on real-world data sets confirmed our random experiments, but more importantly, we carried out exploratory analyses on a large number of problem types, seeking falsifying evidence, but found none.

There are several ways to extend the research, in particular, by examining the interactions of different encoding schemes (Friedman & Goldszmidt 1996), and search strategies with model selection. It would also be of interest to compare these methods to greedy algorithms and/or hypothesis testing methods. For more information, including a more complete description of our data and results, see `www.cs.ualberta.ca/~vanallen/models.html`.

6. Conclusion

We carried out an empirical study to compare three criteria for selecting belief network structures: MDL, AIC and Cross-Validation. We found Cross-Validation was an effective criterion for a wide range of sample sizes and across the broad spectrum of truth complexities, both in terms of number of parameters and parameter values. AIC was also effective, over a somewhat smaller range of truth complexities. MDL, by contrast, required much larger sample sizes to reach the same level of performance as either Cross-Validation or AIC. Based on our experience: for learning belief net structures, if there is no prior knowledge, we advise using Cross-Validation; if there is a prior expectation of simplicity, we advise using AIC; and we advise against the use of MDL.

Acknowledgements

Both authors gratefully acknowledge support from NSERC.

References

Bozdogan, H., (1987). Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika, 52, 3*, 345-370.

Cooper, G. F., & Herskovits, E., (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*, 309-347.

Friedman, N., & Goldszmidt, M., (1996). Learning Bayesian networks with local structure. *UAI 1996.*

Friedman, N., & Yakhini, Z. (1996). On the sample complexity of learning Bayesian networks. *UAI'96.*

Heckerman, D. E. (1995). *A tutorial on learning with Bayesian networks* (Technical Report MSR-TR-95-06). Microsoft Research.

Kearns, M., Mansour, Y., Ng, A. Y., & Ron, D., (2000). An experimental and theoretical comparison of model selection methods. *To appear in Machine Learning.*

Kullback, S., & Leibler, R. A., (1951). On information and sufficiency. *Annals of Mathematical Statistics, 22*, 79-86.

Lam, W., & Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, *10*, 269–293.

Linhart, H., & Zucchini, W.,, (1986). *Model selection.* New York: John Wiley & Sons.

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry.* Singapore: World Scientific.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *J. Royal Statistical So-*

ciety, Series B, 36, 44-47.

Suzuki, J. (1996). Learning bayesian belief networks based on the minimum description length principle: An efficient algorithm using the branch and bound technique. *Machine Learning*.