MODEL SELECTION FOR BELIEF NETWORKS WHEN LEARNING WITH INCOMPLETE DATA

CHUNG-CHIEH SHAN

Abstract. To learn the structure and parameters of a belief network from data, a model selection criterion must be specified that makes a bias-variance tradeoff and selects a network structure with the appropriate complexity. The model selection criterion in the learning procedure interacts with the parameter estimation method. Van Allen empirically compared various model selection criteria in learning settings with complete data [10, §4]. Frequency counting, the canonical parameter estimation method for complete data, optimizes globally; Expectation-Maximization (EM), the canonical parameter estimation method for incomplete data, optimizes locally. To investigate how model selection criteria behave differently when the data is incomplete, we extended Van Allen's experimental design and conducted additional experiments. We found that the effect of incomplete data on model selection is similar to that of reduced sample size, and is consistent with the results for complete data.

1. Introduction

Consider the problem of learning a belief network given random samples from the true distribution. A procedure for this task is essentially a search algorithm that explores the space of belief networks and optimizes some objective function. This objective function evaluates each visited network by estimating how close it is to the truth. Because the network structure of the true distribution is often not available to the learning procedure, it is important for the objective function to take into account the structure of the network being evaluated. The part of the objective function that evaluates the network structure is known as the *model selection criterion*.

In [10, §4], Van Allen conducted an empirical comparison of various model selection criteria. Based on his empirical study, Van Allen suggested that some criteria based on penalizing model complexity, such Minimum Description Length (MDL), may be dominated by their complexity penalty portion, thus underfitting the data when the sample size is not large enough. At the same time, the experiments indicated that the cross-validation criterion is "suitable for a wide range of problems," in the sense that it more reliably emulates the behavior of a loss function whose value is unknown to the learning procedure.

In Van Allen's experiments, all samples were fully observed—there were no missing values or variables in the data. Parameter estimation could thus be performed globally and quickly. Hence, the role of model selection criteria in Van Allen's experiments was to select a belief network structure which, when assigned its globally

Date: January 20, 2001.

This paper is submitted as a final project term paper for Computer Science 282 (Avi Pfeffer, Probabilistic Reasoning) in Fall 2000 at Harvard University.

optimal set of parameters, defines a probability distribution that is as close to truth as possible.

On the other hand, when learning from incomplete data, parameter estimation must be performed locally, and at greater computational expense, canonically using the well-known Expectation-Maximization (EM) algorithm. Because EM is a local optimization procedure, it is prone to being trapped in local minima in the search space. We know that, in general, simpler network structures have lower-dimensional parameter spaces, which in turn tend to contain fewer local minima for both the parameters optimized at each M-step and the expected sufficient statistics computed at each E-step. Unlike in Van Allen's experiments, the role of model selection criteria when learning from incomplete data is to select a belief network structure which, when assigned its *EM-learned* set of parameters, defines a probability distribution that is as close to truth as possible. This shift in role might justify a more severe complexity penalty such as that imposed by MDL. The question now arises as to whether Van Allen's results comparing various model selection criteria still apply when learning from incomplete data.

In this paper, we reproduce Van Allen's results and use a similar experimental design to investigate how incomplete data affects the behavior of model selection criteria. In §2, we specify the task of learning a belief network from sample data. In §3–4, we explain how model selection criteria are invoked in conjunction with a parameter estimation method. In §5, we enumerate and define the criteria studied by Van Allen, including MDL and cross-validation. In §6–7, we describe our experimental design and review our results. Finally, in §8, we conclude with a discussion of our findings.

2. The learning task

To introduce our framework and notation, let $X_1, \ldots, X_n$ be the n random variables in the domain to be learned. We assume that all variables are discrete. A *belief network* over these variables is a pair (θ, h), where h denotes the *structure* of the network (a directed acyclic graph over the random variables), and θ denotes the *parameters* of the network (local conditional probability tables). Each belief network (θ, h) represents a joint distribution $P(X_1, \ldots, X_n) = P(X \mid \theta, h)$ over the variables $X_1, \ldots, X_n$.

The task of learning a belief network from random samples can be stated as follows. Let $P_0(X_1, \ldots, X_n)$ denote the *truth*, an unknown joint probability distribution over $X_1, \ldots, X_n$. Suppose that we have available to us m independent random samples $d = (d_1, \ldots, d_m)$, each of which is taken from the distribution P_0. Sometimes the data is *complete*, meaning all n variables are observed in all m samples. Only learning with complete data is considered in [10]. When the data is *incomplete*, some values are *missing*, i.e., withheld from and invisible to the learning procedure. In addition to reproducing results in [10], our empirical study examines two categories of learning situations with incomplete data:

(1) *Missing values*: Some values in the data are unobserved. In each of our experiments in this category, we fix the total number of missing values to a certain percentage of all values, but each variable may have a different number of missing values, and each sample may also have a different number of missing values.

(2) *Missing variables*: Some of the n variables in the data are unobserved. In each of our experiments in this category, we fix the set of missing variables, so the same set of variables are missing from all sample data, and no other value is missing from the data.

The goal of the learning procedure is to compute a belief network (θ, h) given d, so as to minimize some *loss function* determined by the unknown true distribution P_0. The loss function is usually a measure of how close the learned model P is to the true model P_0. Note that the loss function depends on the truth, and so is unknown to the learning procedure. The objective function, on the other hand, depends only on the data, and so can be computed by learning procedure. Ideally, the behavior of the objective function over the space of networks considered is qualitatively similar to that of the loss function.

A loss function widely used to evaluate belief network learning strategies is *KL-divergence* [6]. The KL-divergence of a learned distribution P from the true distribution P_0 is given by

$$\mathrm{Err}(P_0; P) = \sum_x P_0(x) \cdot \frac{\log P_0}{\log P}, \tag{1}$$

in which x ranges over each possible value $(x_1, \ldots, x_n)$ of the random variables $X_1, \ldots, X_n$. We can rewrite this definition as

$$\mathrm{Err}(P_0; P) = \mathrm{EDL}(P_0; P) - \mathrm{EDL}(P_0; P_0), \tag{2}$$

where the second term is a constant, and the first term

$$\mathrm{EDL}(P_0; P) = -\sum_x P_0(x) \log P(x) \tag{3}$$

is the expected description length of encoding samples from P_0 using the optimal code code based on P. The KL-divergence is non-negative; it is zero when $P_0 = P$, at which point $\mathrm{EDL}(P_0; P)$ is equal to $\mathrm{EDL}(P_0; P_0)$, the entropy of P_0.

Following common practice and [10], we evaluate each learned network by its KL-divergence from the truth. One complication arises from incomplete data: When some variables are missing, the value labels on these variables are no longer expressed; in other words, the data is invariant with respect to any permutation of the values of any missing variable. Hence, it is no longer meaningful to measure KL-divergence across all variables; instead, we marginalize the learned model and the truth over the observed variables, and measure the KL-divergence between the resulting (smaller) joint distributions.

Even when all variables are observed, the data may still be invariant with respect to some (non-identity) permutations of the values of some variable. This is the case precisely when two or more possible values of a variable fail to appear in any sample. This is possible even with complete data, but more likely when some values are missing, when the sample size is small, or when marginal probabilities are uneven in the true distribution. In this case, we do not evaluate learned networks differently from common practice; in other words, we only exclude a variable from our computation of the KL-divergence if it is not observed at all in the data.[1]

[1]We only encountered this situation in our experiments for the LiabilityCost variable in the Insurance network, which has 4 possible values and the uneven marginal probability distribution $(0.964, 0.018, 0.011, 0.007)$.

3. The learning procedure

One simple approach to learning is *maximum likelihood* (ML). In this approach, we choose the network (θ, h) that maximizes the likelihood $P(d \mid \theta, h)$ of the data d. Note that any distribution that can be represented by one network structure can also be represented by a more complex network structure. Therefore, the ML approach always chooses a maximally complex model. A complex model allows parameters to be chosen to fit the data more closely, but it also increases the number of parameters that need to be determined. This high dimensionality of the parameter space causes the learned model to be easily swayed away from the truth by the variance inherent in the data. In short, we say that ML learning often *overfits* the data, choosing a network structure more complex than the optimal or true one. Other approaches to learning may *underfit* the data, choosing a network structure simpler than the optimal or true one.

Because the data does not tell us directly how complex the truth is, any learning procedure faces the task of selecting a network structure h of the correct complexity. A more parsimonious structure promises greater generalization power at the risk of being oblivious to the data; a more complex structure enables a better fit to the data at the risk of being swayed by it. Thus, underfitting increases *bias error* while overfitting increases *variance error*. Bias error and variance error together comprise the error of the learned model from the truth. Any learning procedure makes a *bias-variance tradeoff* when they determine the structure of the learned network and hence its complexity.

Van Allen divides the basic belief network learning procedure into two parts:

(1) *Parameter estimation*: For a given network structure, determine the optimal parameters under the structure.
(2) *Model selection*: For a given parameter estimation method, determine the network structure for which the method produces the best network overall.

It is clear from the division above that the optimal network structure depends on the parameter estimation method. Strictly speaking, then, each model selection criterion should only be used in conjunction with the parameter estimation method for which it was devised. For reasons explained in [10, §4.1.7], however, Van Allen used the same parameter estimation method in conjunction with all model selection criteria he evaluated.

4. Parameter estimation

The parameter estimation method Van Allen used was not the ML estimate but the *Bayes estimate* (also called the average or predictive distribution). The Bayes estimate is based on the Bayesian approach to learning, and is defined as follows. Let h be the given (fixed) structure for which parameters are to be estimated, and $P(\Theta \mid h)$ be some prior distribution over these parameters. The Bayes estimate of the true distribution, given d and h, is the distribution

$$\begin{aligned}\tilde{P}(X \mid d, h) &= \int_\theta P(X \mid \theta, h) P(\theta \mid d, h)\, d\theta \\ &= \frac{\int_\theta P(X \mid \theta, h) P(d \mid \theta, h) P(\theta \mid h)\, d\theta}{\int_\theta P(d \mid \theta, h) P(\theta \mid h)\, d\theta}.\end{aligned} \tag{4}$$

In words, we take as our learned model the average of all models with the given structure, weighted by their posterior probability. The posterior probability is (proportional to) the product of the likelihood and the prior. Based on an information theory result by Shannon, we know that the Bayes estimate $\tilde{P}$ is the Bayes-optimal strategy for minimizing the expected KL-divergence [10, §4.1.6].

The Bayes estimate is different from the *maximum a posteriori* (MAP) estimate, which is defined as the model that maximizes the posterior, with no averaging across models:

$$\begin{aligned}\hat{\theta}(X \mid d, h) &= \operatorname{argmax}_\theta P(X \mid \theta, h) P(\theta \mid d, h) \\ &= \operatorname{argmax}_\theta P(X \mid \theta, h) P(d \mid \theta, h) P(\theta \mid h), \\ \hat{P}(X \mid d, h) &= P(X \mid \hat{\theta}, h).\end{aligned} \tag{5}$$

As a special case, the MAP estimate with the uniform prior is equal to the ML estimate.

We can consider the Bayes and MAP estimates as different ways of summarizing the posterior distribution $P(\theta \mid d, h)$ around its peak. Note that the MAP estimate only takes into account the location of the peak, while the Bayes estimate takes into account the location as well as sharpness of the peak.

When the random variables $X_1, \ldots, X_n$ are all discrete, and the local conditional probability distributions are unrestricted multinomial distributions, it is convenient to take the parameter prior $P(\theta \mid h)$ to be a product of Dirichlet distributions, one for each local conditional probability table [5, §2]. The Dirichlet family of priors has several useful properties:

(1) With complete data, both the Bayes estimate and the MAP estimate with a Dirichlet prior are computable in closed form.
(2) The Dirichlet prior with all hyperparameters set to 1 is equal to the uniform prior.
(3) With complete data, the Bayes estimate with a Dirichlet prior happens to be equal to the MAP estimate with another Dirichlet prior—one with all hyperparameters incremented by 1.
(4) In particular, with complete data, the Bayes estimate with the uniform prior happens to be equal to the MAP estimate with the Dirichlet prior with all hyperparameters set to 2.
(5) Incrementing all hyperparameters by 1 is equivalent to 'imagining' that every possible combination of parent and child values in the belief network has been observed one extra time. Therefore, the Dirichlet prior with all hyperparameters set to 2 is sometimes referred to as the *add-one prior*.

Based on these properties, a commonly applied technique in belief network learning is to estimate parameters by ML with every observation frequency count incremented by 1. This practice is sometimes referred to as the *add-one trick*. With complete data, the ML estimate produced by the add-one trick is equivalent to the MAP estimate with the add-one prior, which in turns is equivalent to the Bayes estimate with the uniform prior.

With complete data, the add-one trick is simply an exact, closed-form computation of the Bayes estimate. All theoretical justifications for the Bayes estimate apply to the add-one trick—avoiding overfitting, minimizing expected KL-convergence, and the Bayesian philosophy. With incomplete data, however, the theoretical justification for the add-one trick becomes much weaker. There is no known method for computing the Bayes estimate given incomplete data; the canonical method for learning network parameters from incomplete data is the Expectation-Maximization (EM) algorithm, a local optimization algorithm for MAP parameter estimation [5, §6.3]. In the case of incomplete data, the Bayes estimate is no longer the MAP estimate with all Dirichlet prior hyperparameters incremented by 1. Yet the add-one trick is often 'generalized' from complete data to incomplete data as follows: Instead of using EM to estimate ML parameters (i.e., MAP parameters with the uniform prior), use EM to estimate MAP parameters with the add-one prior.

As far as we know, there has been no study of how closely the MAP parameter estimate with the add-one prior approximates the Bayes estimate when the data is incomplete. The best justification we know of for running EM with the add-one prior is overfitting avoidance. In our experiments, we follow this common practice of running EM with the add-one prior, and denote the parameters thus learned with $\tilde{\theta}(d,h)$. Note that, with incomplete data, $\tilde{\theta}$ is no longer a deterministic function, as EM is not deterministic. Of course, in the case of complete data, our usage of the symbol $\tilde{\theta}$ reduces to the (deterministic) Bayes estimate defined above.[2]

5. Model selection

In §3 above, we introduced the concept of a model selection criterion, whose goal is to determine the optimal network structure. More specifically, a model selection criterion is a scoring function that assigns to each network structure a numerical score. For convenience, we assign *lower* scores to better networks; consequently, the optimal network structure is the one that minimizes the criterion. In the Bayesian framework, the criterion can be regarded as an unnormalized negative log prior on the space of network structures under consideration.

Van Allen examined the following criteria in [10, §4]. All criteria we describe below are normalized by m, the total number of samples.

5.1. Fit to data.

The *fit* of the model to the data is defined as the (normalized negative log) likelihood of the data under the estimated parameters:

$$\mathrm{Fit}(h,d) = -\frac{1}{m}\sum_{i=1}^{m} \log \tilde{P}(X = d_i \mid d, h). \tag{6}$$

This criterion measures only how expressive the network structure is, i.e., how well the parameters can be adjusted to describe the data, but does not attempt to correct for variance in the data.

5.2. The Bayesian Information Criterion.

The *Bayesian Information Criterion* (BIC), approximately equal to the *Minimum Description Length* criterion (MDL), is defined by

$$\mathrm{BIC}(h,d) = \mathrm{Fit}(h,d) + \frac{\mathrm{Dim}(h)\log m}{2m}, \tag{7}$$

[2] Despite our maintenance of the status quo on this issue, we believe that it deserves further investigation. A more principled approach to approximating the Bayes estimate given incomplete data may be pointed out by [5, §6.2]. The idea is to approximate the posterior as a multivariate Gaussian distribution over the parameter space, centered at the MAP parameter estimate. To compute this Gaussian, we need to know (1) the MAP parameter estimate and (2) the negative Hessian of the log-posterior at the MAP parameter estimate. For (1), EM remains the canonical approach. For (2), "Thiesson [9] demonstrates that, for unrestricted multinomial distributions, the second derivative can be computed using Bayesian-network inference."

where $\mathrm{Dim}(h)$ is the parameter space dimensionality of the network structure h.[3] In the formula above, the first term is the number of bits required to describe the data given the learned model, and the second term is the number of bits required to describe the learned model itself. The criterion thus minimizes the total description length of the model and the samples. Intuitively, the second term above penalizes complexity of the network.[4]

5.3. **The Akaike Information Criterion.** The *Akaike Information Criterion* (AIC) is another criterion that penalizes complexity of the network, though less severely. It is defined by

$$\mathrm{AIC}(h, d) = \mathrm{Fit}(h, d) + \frac{\mathrm{Dim}(h) \log e}{m}, \tag{8}$$

where e is the base of the natural log, and $\log e$ converts to bits.

5.4. **Cross-validation.** Van Allen examined the version of *cross-validation* known as *two-fold cross-validation*. This technique partitions the data $(d_1, \ldots, d_m)$ into two subsamples $(d_1, \ldots, d_{m/2})$ and $(d_{m/2+1}, \ldots, d_m)$, and uses each subsample to test a network learned from the other subsample:

$$\begin{aligned} \mathrm{XV}(h, d) = -\frac{1}{m}\Bigg(&\sum_{i=1}^{m/2} \log \tilde{P}\big(X = d_i \mid (d_{m/2+1}, \ldots, d_m), h\big) \\ &+ \sum_{i=m/2+1}^{m} \log \tilde{P}\big(X = d_i \mid (d_1, \ldots, d_{m/2}), h\big)\Bigg). \end{aligned} \tag{9}$$

Cross-validation simulates learning parameters from half the data, then tests the learned network on the other (withheld) half of the data. This technique requires learning two networks (one from each half of the data) for validation purposes, in addition to learning a final network from all of the data as usual.

5.5. **The prequential criterion.** The *prequential criterion* uses each sample instance to test a network trained on all preceding sample instances:

$$\mathrm{Preq}(h, d) = -\frac{1}{m} \sum_{i=1}^{m} \log \tilde{P}\big(X = d_i \mid (d_1, \ldots, d_{i-1}), h\big). \tag{10}$$

The prequential criterion is similar in spirit to cross-validation. In terms of Bayesian theory, the prequential criterion has the advantage of being equivalent to the likelihood of the data under the prior [5, §8.1]. However, it is much more expensive to compute—it requires learning m networks from increasingly large portions of the data, in addition to learning a final network from all of the data as usual.

---

[3] The dimensionality of the parameter space is the number of free parameters that can be adjusted. For example, the binomial distribution has a 1-dimensional parameter space.

[4] The BIC is derived as a large-sample approximation to the Bayesian posterior. The authors of [4] argue that the standard BIC approximation is no longer applicable when the sample is incomplete, and offer an alternative. Regardless, our experiments adhere to the formula above.

5.6. **Bootstrapping.** The *bootstrapping* criterion tests the fit of the given sample on a network learned from not the given sample but a resampling thereof:

$$\mathrm{Boot}(h, d) = -\frac{1}{m} \sum_{i=1}^{m} \log \tilde{P}(X = d_i \mid d', h), \tag{11}$$

where d' is a sample of size m, generated from d by resampling with replacement. Bootstrapping can be thought of as another method for simulating sampling variance, distinct from cross-validation and the prequential criterion: Instead of withholding from learning some data and using that data later for testing, bootstrapping uses all available data for both learning and testing, but resamples during learning to simulate sampling variance.

6. Empirical study

To characterize the behavior of the model selection criteria described in §5 above, we performed an empirical study consisting of the following steps:

(1) *Pick a true distribution.* We tried using the following belief networks as the truth:
   (a) Randomly generated networks with 10 binary variables and 10, 20 or 30 links. To generate each random network, we first selected a network structure uniform-randomly, then assigned the conditional probabilities, also uniform-randomly.
   (b) The Alarm network [1], a real-world benchmark network with 37 variables and 46 links. Each variable has between 2 and 4 possible values; there are 509 free parameters in total.
   (c) The Insurance network [2], another real-world benchmark network, with 27 variables and 52 links. Each variable has between 2 and 5 possible values; there are 1008 free parameters in total.

   We retrieved the Alarm and Insurance networks in BIF format from the Bayesian Network Repository [3], then converted them for use in the Bayes Net Toolbox [7] using a conversion tool we built [8].
(2) *Generate a sequence of hypotheses that includes the truth.* Let t denote the number of links in the true network. We chose a number $T > t$.[5] We uniformly-randomly generated a sequence of $T + 1$ candidate network structures $(h_0, \ldots, h_T)$, such that:
   (a) Each structure h_i has the same random variables as the truth; i.e., represents the same domain as the truth.
   (b) Each structure h_i contains exactly i links.
   (c) The links in each hypothesis h_i is a properly subset of the links in h_{i+1}, missing exactly one link compared to h_{i+1}. In other words, the sequence of structures is nested.
   (d) The structure h_t is the true structure.

[5]For the Alarm network, where $t = 46$, we chose $T = 70$. For the Insurance network, where $t = 52$, we chose $T = 80$. For the randomly generated networks, where $t \in \{10, 20, 30\}$, we chose $T = 45$, the number of links in a fully connected network with 10 nodes. These choices are same as those in Van Allen's original experiments.

(3) *Generate sample data.* To generate sample data for the learning procedure, we first produced m samples from the true distribution, where m is a number we chose to be either 50 or 100. We then optionally introduced missing data in one of the following two ways:
   (a) We introduced missing values by uniformly-randomly occluding μmn values out of the mn that comprise the complete data, where μ is a parameter we chose between 10% and 50%.
   (b) For the Insurance network, we also tried introducing missing variables by occluding the 12 out of 27 variables specified as hidden in [2, §5.3.3].

   We never introduced both missing variables and missing values in the same experiment.
(4) *Learn network parameters.* For each hypothesis h_i, we ran the EM algorithm with the add-one prior, repeating 5 times and thus learning 5 different sets of network parameters.
(5) *Compute KL-divergence.* For each distribution P learned in step 4, we marginalized P and P_0 over the observed variables to become P^o and P_0^o. Hence $P^o = P$ and $P_0^o = P_0$ except when we introduced missing variables for the Insurance network. We then computed the expected description length $\mathrm{EDL}(P_0^o; P^o)$ as well as the true entropy $\mathrm{EDL}(P_0^o; P_0^o)$. The difference between the expected description length and the true entropy is the KL-divergence. We took the average over the 5 trials for each hypothesis.
(6) *Evaluate model selection criteria.* We used the networks learned in step 4 to compute the fit-to-data score Fit and the complexity penalty scores BIC and AIC. We averaged the scores over the 5 trials for each candidate structure. We performed additional learning to compute the two-fold cross-validation score XV, repeating 5 times and taking the average. In a few of our experiments, we also performed a large amount of additional learning and computed Preq, again repeating 5 times and taking the average. We did not examine Boot in our experiments.

Our experimental design generalizes Van Allen's [10, §4]. In particular, the latter is equivalent to the special case where there is no missing data and so no repeated learning trials are necessary.

7. Results

Unfortunately, we did not have time to collect enough data to conduct a quantitative and comprehensive analysis that is meaningful. Nevertheless, a qualitative, case-by-case examination of our results can provide preliminary insights and point the way towards more detailed investigation in the future.

7.1. **Random networks.**

Let us first examine our results when the truth is a randomly generated network and there is no missing data. Figures 1 and 2 show typical experiment results for when the true network is randomly generated, and the data is complete. Each graph displays the results from one experiment. The experiments in both figures used the same true network, with 10 nodes and 20 links. The difference between the two experiments is that $m = 50$ samples were provided to the learning procedure in Figure 1, and $m = 100$ in Figure 2.

The x-axis shows the complexity of each hypothesis structure in terms of number of links. The left edge of the graph corresponds to the hypothesis with no

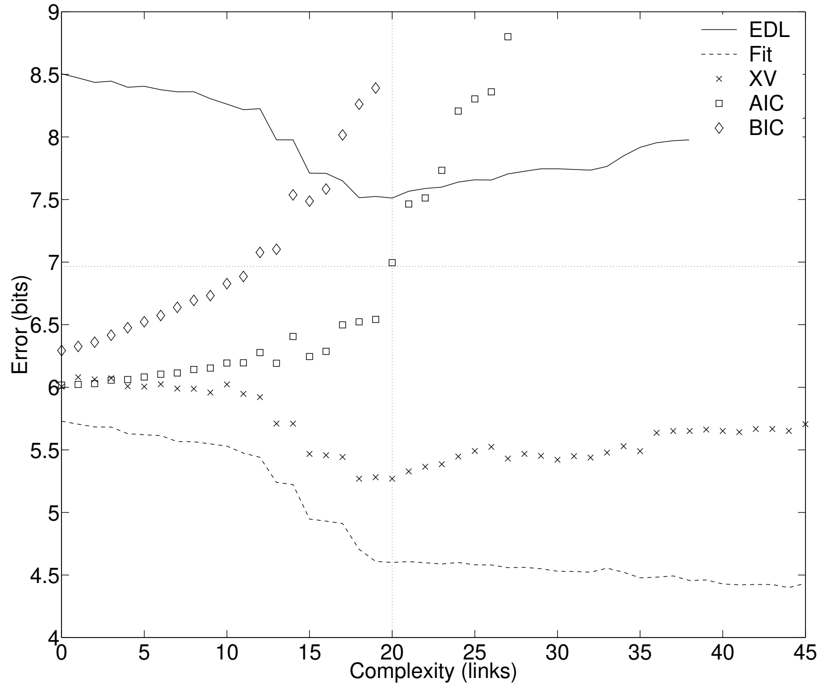

FIGURE 1. Random network results: $t = 20$, $m = 50$, $\mu = 0$.

link whatsoever. The true structure is indicated with a vertical dotted line. The hypotheses to the left of the dotted line underfit; the ones to the right overfit.

The y-axis shows various criterion scores in terms of bits. The solid curve across the graph shows the expected description length, and the dotted line underneath shows the true entropy. The difference between the two is the KL-divergence of the learned model from the truth, averaged over 5 trials as usual. From left to right, the KL-divergence curve shows the bias-variance tradeoff in action as the hypothesis structures shift from being simpler than the truth to being more complex than the truth. As shown in Figures 1 and 2, the KL-divergence usually decreases as the candidate structure increases in complexity from 0 to the true complexity, then increases as the structure begins to overfit instead of underfit. This confirms our intuition that the network structure that enables learning a distribution closest to the true distribution is the true structure.

The ideal model selection criterion is one with the same qualitative behavior as KL-divergence. In terms of the hypothesis sequences that we used in our experiments, we would like a criterion to have the same shape as KL-divergence, or at least be generally U-shaped and achieve minimum at the same place as KL-divergence.

The fit-to-data score, shown in the graph with a dashed curve, generally decreases as the hypothesis complexity increases from zero to the true complexity and beyond. Therefore, if we were to use the Fit score to select a model when learning these randomly generated networks, we would overfit. Note that, because we used the Bayes estimate $\tilde{P}$ rather than the MAP estimate $\hat{P}$ for parameter estimation, the Fit score does not always strictly decrease with respect to hypothesis complexity.

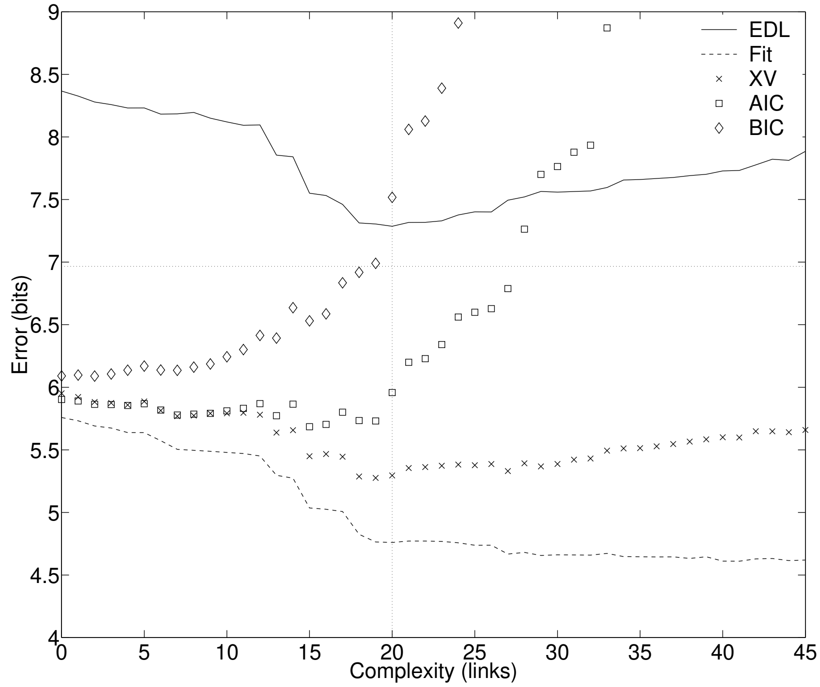

FIGURE 2. Random network results: $t = 20$, $m = 100$, $\mu = 0$.

Compared to randomly generated true networks, this effect becomes much more significant for the Alarm and Insurance networks, as we will see in §7.2.

When the amount of data is relatively small, say $m = 50$ samples, BIC and AIC significantly underfit, often achieving minima at zero hypothesis complexity. When $m = 100$ samples are available to the learning procedure, the performance of BIC and AIC starts to improve, in that they become less sensitive to sampling variance and more similar to KL-divergence. According to [10, §4.3], eventually both criteria converge on selecting the true network. In other words, both BIC and AIC have a tendency to overfit that diminishes as the sample size increases. Because BIC penalizes complexity more severely than AIC does, BIC has a stronger tendency to overfit when the sample size is small.

Compared to BIC and AIC, the two-fold cross-validation criterion is much more similar to KL-divergence in terms of shape and minima. In our experiments as well as those reported in [10, §4.3], the cross-validation curve almost always follows the overall trend of KL-divergence, and usually achieves minima at or next to where KL-divergence does. Cross-validation also often replicates the bumps and other irregularities in the KL-divergence curve on a smaller scale. This behavioral similarity between cross-validation and KL-divergence holds even when the sample size is the smaller $m = 50$.

As for the prequential criterion, we only computed it in a few experiments, because parameter estimation with incomplete data is computationally expensive (much more so than with complete data). These experiments of ours bear out the report in [10, §4.3] that the prequential criterion and two-fold cross-validation “give

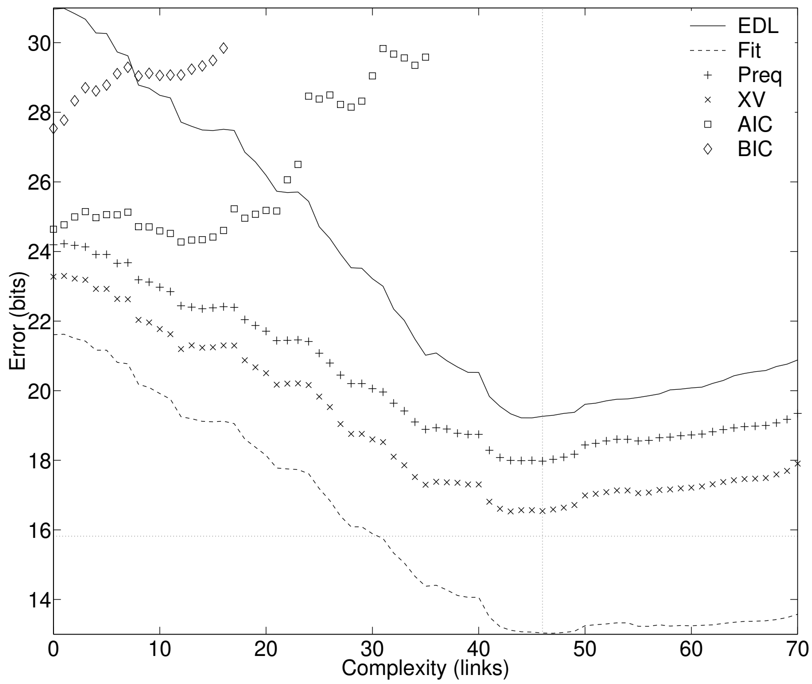

FIGURE 3. Alarm network results: $m = 50$, $\mu = 0$.

very close evaluations," except "the prequential criterion appears to be slightly less sensitive to underlying fluctuations in [KL-divergence]." In any case, because it is so expensive to compute, the prequential criterion—at least in its standard form—is not suitable as a model selection criterion when learning with incomplete data. We revisit this point below in §8.

Returning to the behavior of KL-divergence, note that it does not always achieve its minimum at the true structure hypothesis. The minimum KL-divergence often occurs at a hypothesis a few links *simpler* than the truth; this is often the case when less data is available to the learning procedure. This observation is consistent with our intuition that the magnitude of variance error relative to bias error is larger when fewer samples are available, so the optimal bias-variance tradeoff point occurs at a simpler hypothesis. When the data is complete, the parameter estimation procedure is deterministic, so the minimum KL-divergence never occurs at any hypothesis more complex than the truth. We could not discern any correlation between the number of links in the true network and the offset from the true hypothesis to the hypothesis where the KL-divergence is minimized.

7.2. **Benchmark networks.** Turning from random networks to benchmark networks, let us consider Figures 3, 4 and 5, which show typical results for when the true network is either Alarm or Insurance, and the data is complete. These results and the results for random networks are mostly similar. To summarize: KL-divergence is minimized at the true structure complexity, except sometimes—more often when the sample size is smaller—it is minimized at slightly lower than the true structure complexity. (Roughly speaking, the hypothesis that minimizes

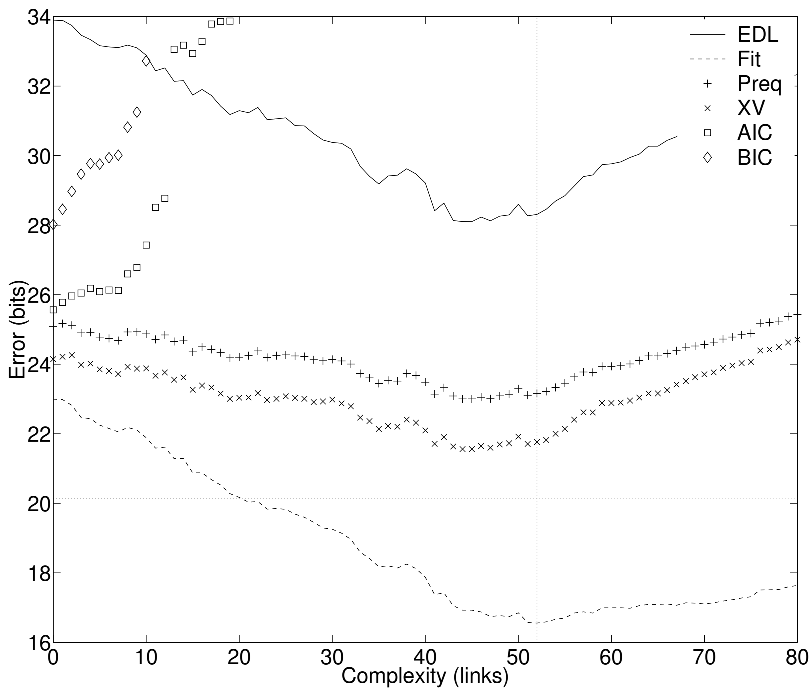

FIGURE 4. Insurance network results: $m = 50$, $\mu = 0$.

KL-divergence is simpler than the true structure most often for the Insurance network, followed by Alarm and then the randomly generated networks. This reflects the fact that Insurance has the most free parameters and therefore is hardest to estimate parameters for, followed by Alarm and then the randomly generated networks.) The BIC and AIC criteria still both underfit the truth, especially BIC and in particular when the sample size is smaller. As before, two-fold cross-validation emulates KL-divergence relatively well.

Our results for benchmark networks and for random networks differ prominently in one regard, namely the behavior of the fit-to-data score Fit. Recall that, for random networks, Fit tends to monotonically decrease with respect to the hypothesis complexity, even though it occasionally increases slightly before diving further. These occasional increases can be attributed to our using the Bayes estimate $\tilde{P}$ rather than the MAP estimate $\hat{P}$ for learning parameters; overall, Fit as a model selection criterion still grossly overfits. For Alarm and Insurance, however, Fit as a criterion actually comes quite close to selecting the true structure. For these networks, Fit manages to emulate the general shape, minima location, and sometimes even small-scale irregularities of KL-divergence to an extent comparable to that of cross-validation.

As we described above in §4, the Bayes estimate is an average model that takes into account the location of the posterior peak in parameter space as well as its sharpness, unlike the MAP estimate, which only takes into account the location of the peak. Intuitively, the Bayes estimate hedges its bets toward the prior and away from the data if there are too many degrees of freedom in parameter space—and too

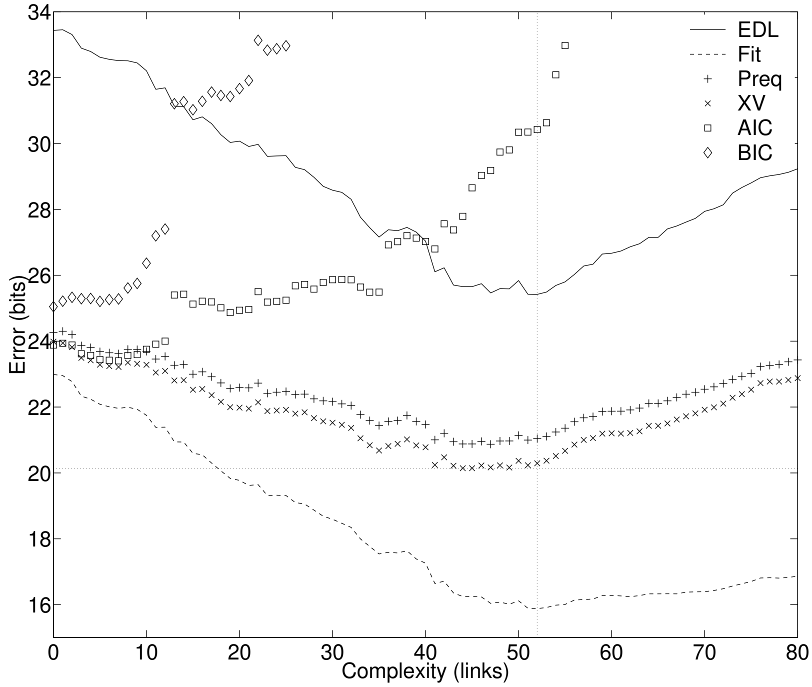

FIGURE 5. Insurance network results: $m = 100$, $\mu = 0$.

few samples—for the learning procedure to accurately estimate the true parameters from the data. While this intuition does help us understand what is going on in Figures 3, 4 and 5, that we used the Bayes estimate rather than the MAP estimate to learn parameters cannot fully explain the rebound of Fit in benchmark networks, since it did not occur in random networks under the same experimental design.

In [10, §4], Van Allen attributed the rebound of Fit in benchmark networks to "the fact that the benchmark networks are locally low entropy—the conditional probability distributions for each node tend to be highly skewed." More specifically, when the conditional probability distributions are highly skewed, the average model specified by the Bayes estimate is more distant from the peak model specified by the MAP estimate; consequently, the gap between the fit-to-data of the average model and that of the peak model is greater. This explanation is reasonable, but we have yet to understand why the magnitude of the additional gap across the hypothesis sequence is so well-tuned to the location of the true structure as we see in our results.

In Figure 4, Fit reaches minimum at $t = 52$, exactly the true structure; EDL, on the other hand, reaches minimum at $t = 45$, presumably because there was not enough data to properly estimate parameters for the true structure. Indeed, several cases in our experiments such as that in Figure 4 suggest that, a significant minority of the time, the hypothesis selected by Fit comes even closer to the truth than a simpler hypothesis selected by KL-divergence. Of course, we can regard these cases as Fit overfitting *KL-divergence*; after all, we began this study by specifying KL-divergence as the loss function that model selection criteria should strive to

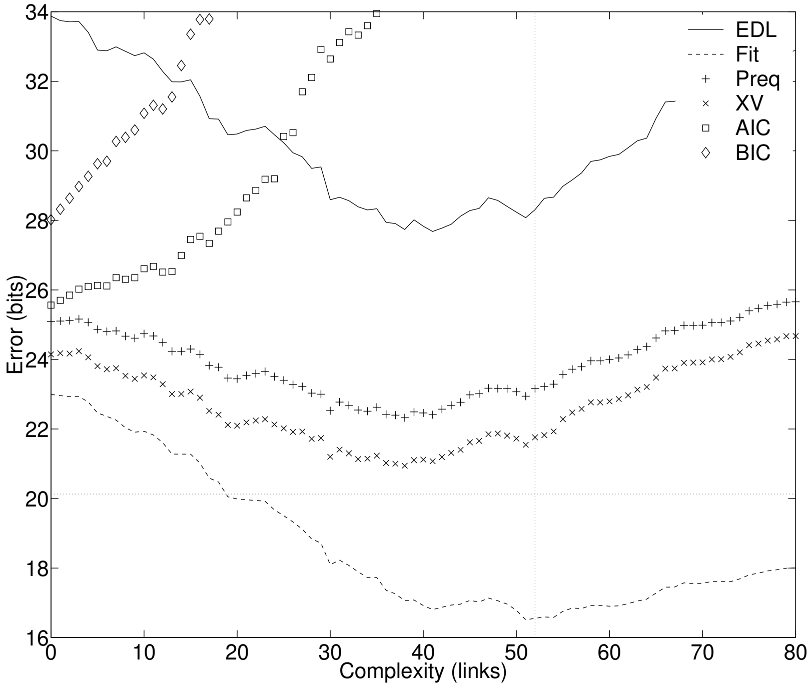

FIGURE 6. Insurance network results: $m = 50$, $\mu = 0$. This experiment differs from the one shown in Figure 4 only in the sequence of candidate network structures evaluated.

minimize. On the other hand, we also wonder whether it would be meaningful to say in these cases that Fit, a score easily known to the learning procedure, selects the true *hypothesis* better than EDL, a function supposedly more informative but never known to the learning procedure.

Before moving on to experiments involving incomplete data, we note another phenomenon that appears with some frequency when the information in the data is not quite enough to estimate the parameters for the true structure. Not only do BIC and AIC select models that underfit the true structure, so does two-fold cross-validation, and to a greater extent than KL-divergence does (although, as one might expect, still to a lesser extent than BIC and AIC do). Figure 3 marginally fits this description: The true structure has 46 links, KL-divergence selects the structure with 45 links, and cross-validation selects the structure with 43 links. (Meanwhile, Preq, AIC, BIC, and Fit select structures with 46, 12, 0, and 47 links, respectively.) An experiment that cannot claim to be typical but more strikingly exhibits the phenomenon is shown in Figure 6. In this experiment, the true network is Insurance, which has 52 links; EDL, XV, Preq, AIC, BIC, and Fit select structures with 41, 38, 38, 0, 0, 51 links, respectively. (Perhaps incidentally, this is also an experiment in which the network selected by Fit underfits the true structure less than the network selected by KL-divergence does.)

One explanation for why two-fold cross-validation underfits when the sample size is small is that it evaluates how well model parameters can be estimated given *half* as much data as is actually available to the learning procedure. If this explanation

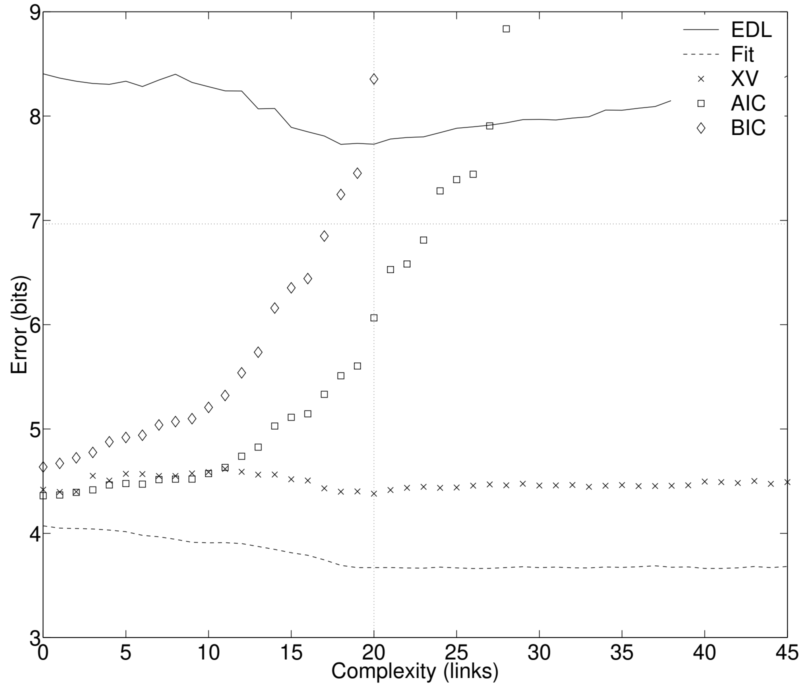

FIGURE 7. Random network results: $t = 20$, $m = 50$, $\mu = 30\%$.

is correct, r-fold cross-validation, which divides the data into $r > 2$ subsamples, would underfit less. Unfortunately, regardless of whether r-fold cross-validation underfits, it requires r passes of parameter estimation to compute, compared to 2 passes to compute two-fold cross-validation.

7.3. **Incomplete data.** Figures 7–11 show typical results learning random networks as well as benchmark networks with incomplete data. In all these experiments, we occluded 30% of all sample values in the data, except in Figure 11, where we occluded the 12 out of 27 variables specified as hidden in [2, §5.3.3]. The true networks and candidate structure sequences in these experiments are identical to those shown in the preceding figures; the only difference between these experiments and the previous ones is that the data is incomplete.

Having discussed many experiments in the preceding sections where the data contains less information than parameter estimation appears to require, we are hard-pressed to identify any qualitative differences between our results with complete data and with incomplete data. All the observations we have made so far continue to apply: Briefly, AIC and BIC always select models that underfit significantly; XV and KL-divergence sometimes select models that underfit somewhat. For random networks, Fit monotonically decreases, but for the two benchmark networks that we tried, the structure complexity selected by Fit lies roughly between the true structure and the structure selected by XV and KL-divergence.[6]

[6]A minor difference caused by incomplete data is that the parameter estimation procedure is no longer deterministic. As a result, in rare occasions KL-divergence achieves its minimum at a hypothesis several links more complex than the truth.

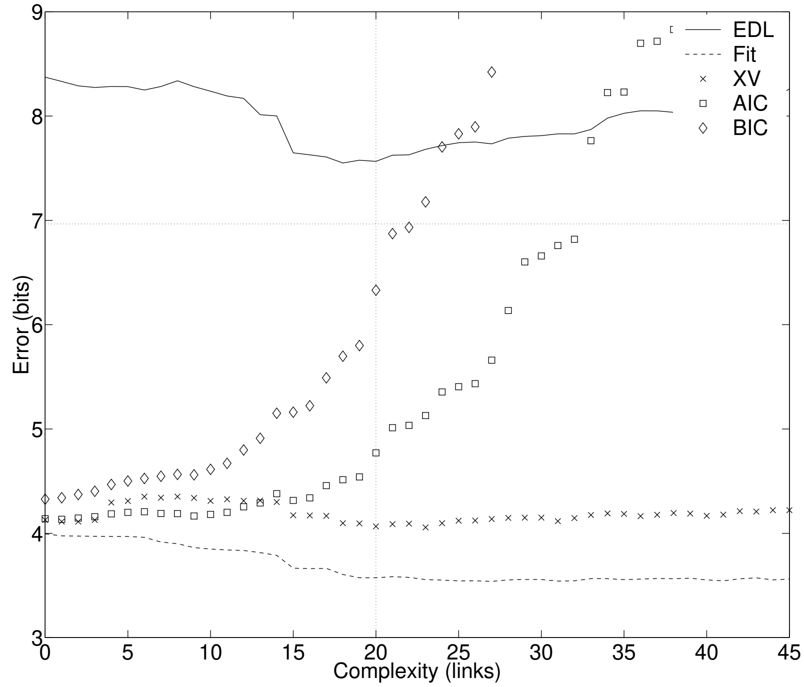

FIGURE 8. Random network results: $t = 20$, $m = 100$, $\mu = 30\%$.

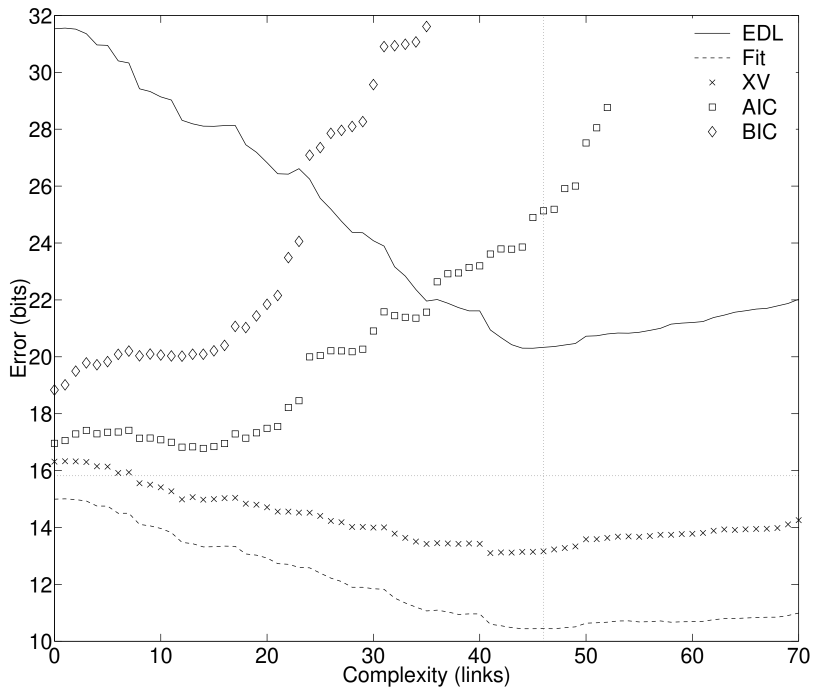

FIGURE 9. Alarm network results: $m = 50$, $\mu = 30\%$.

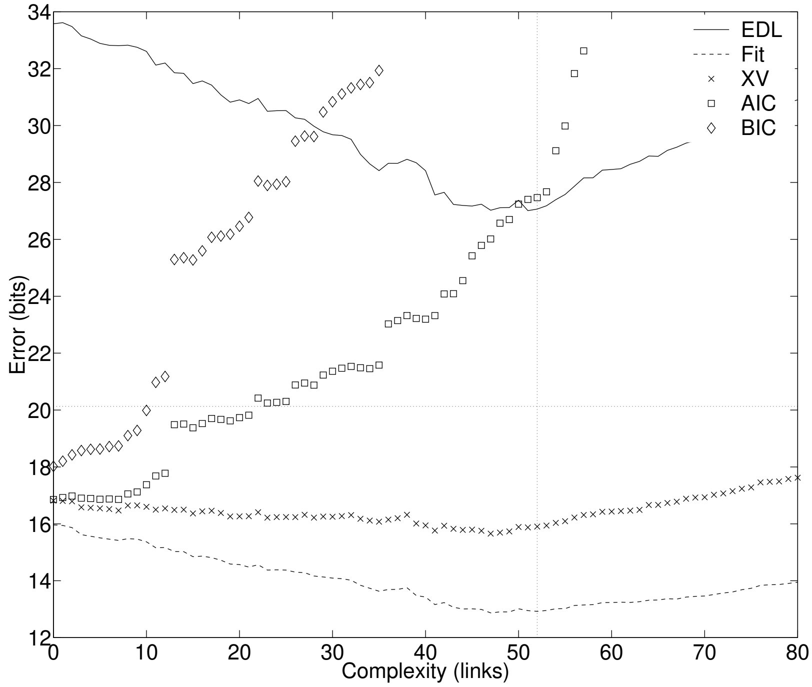

FIGURE 10. Insurance network results: $m = 100$, $\mu = 30\%$.

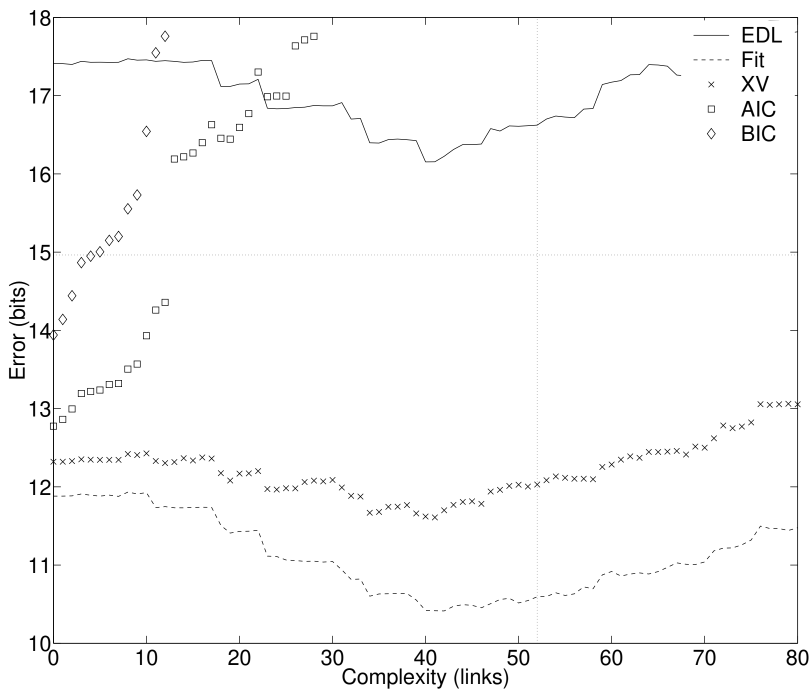

FIGURE 11. Insurance network results: $m = 100$, missing variables.

Intuitively, both the scarcity of sample instances and the scarcity of data values reduce the total amount of information in the data, making the high dimensionality of the parameter space more problematic for the learning procedure. From this perspective, we may tentatively compare Figure 1 against Figure 8 and Figure 4 against Figures 10–11, finding them to be generally similar. The curves for the cross-validation and fit-to-data scores are noticeably flatter in the experiments with incomplete data. This is probably because the expected sufficient statistics computed by the E-step in EM is less skewed (i.e., contains less entropy) than fully observed samples. In any case, the flatness does not seem to significantly affect the performance of the criteria under consideration.

8. Discussion

As explained in §1, the original motivation for this study was the question: How do model selection criteria behave differently when parameter estimation must be performed by local optimization? Taking as our starting point Van Allen's study [10, §4], which characterized the behavior of model selection criteria when parameter estimation is performed by global optimization, we extended the experimental design and conducted some preliminary experiments.

Though both the design of our experiments and their results raised issues relevant to belief network learning and model selection, we did not find that the incompleteness of the data per se to be consequential. Missing data reduced the amount of information available to the learning procedure, just as reducing the number of sample instances did. In either case, the complexity penalties we examined continued to be overly severe, selecting a model that underfits the data. On the other hand, cross-validation was able to closely predict the behavior of KL-divergence, our loss function, only underfitting somewhat when the amount of sample data is low.

In this study, we only examined two-fold cross-validation, taking it to represent the cross-validation technique overall. For learning situations where the amount of sample data is indeed low, general r-fold cross-validation may turn out to underfit less than two-fold cross-validation. Alternatively, a modified version of the prequential criterion that processes multiple sample instances at once might be computationally feasible, practically useful and theoretically grounded. Additional studies would be needed to explore these ideas. For example, it might be theoretically if not practically interesting to compare cross-validation with the standard prequential criterion for learning with incomplete data.[7]

When the true distribution is a benchmark network rather than a randomly generated network, the fit-to-data score consistently achieves minimum around the structure complexity of the true network. This phenomenon, in our opinion, remains partially unexplained.

A complication unavoidable in practical learning procedures that both Van Allen and us sidestepped in our experiments is how to perform structure search. Obviously, a practical learning procedure must not require as input a nested hypothesis sequence containing the truth. The experiments described here and in [10, §4] give us some intuition for various proposed model selection criteria, but our experiments

[7] We did compute the prequential score in a few incomplete-data experiments. In these experiments, the prequential criterion and cross-validation behaved very similarly. Then again, the missing value ratio μ was only set to a low 10%.

should not be regarded as, for example, simulations of structural learning of belief networks. The focus of this paper is the interaction between parameter estimation methods and model selection criteria; it would be worthwhile to investigate the interaction between structure search methods and model selection criteria as well.

References

[1] I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proc. 2'nd European Conf. on AI and Medicine*, 1989.

[2] J. Binder, D. Koller, S. Russell, K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29 : 213–244, 1997.

[3] N. Friedman. Bayesian network repository. Available online at `http://www-nt.cs.berkeley.edu/home/nir/public_html/Repository`.

[4] D. Geiger, D. Heckerman, and C. Meek. Asymptotic model selection for directed networks with hidden variables. Technical Report MSR-TR-96-07, Microsoft, 1996. Available online at `ftp://ftp.research.microsoft.com/pub/tr/tr-96-07.ps`.

[5] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft, 1995; revised January 1996. Available online at `ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.ps`.

[6] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22 : 79–86, 1951.

[7] K. Murphy. Bayes net toolbox for MATLAB. Available online at `http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html`.

[8] C. Shan. BIF-BNT converter. Available online at `http://www.digitas.harvard.edu/~ken/bif2bnt`.

[9] B. Thiesson. Score and information for recursive exponential models with incomplete data. Technical report, Institute of Electronic Systems, Aalborg University, 1995.

[10] T. Van Allen. Handling uncertainty when you're handling uncertainty: Model selection and error bars for belief networks. Master of Science thesis, Department of Computer Science, University of Alberta, 2000. Available online at `http://www.cs.ualberta.ca/~vanallen/thesis.ps`.

Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA
E-mail address: `ken@digitas.harvard.edu`