Bagged Structure Learning of Bayesian Networks

Gal Elidan
Department of Statistics, Hebrew University

Abstract

We present a novel approach for density estimation using Bayesian networks when faced with scarce and partially observed data. Our approach relies on Efron's bootstrap framework, and replaces the standard model selection score by a bootstrap aggregation objective aimed at sifting out bad decisions *during* the learning procedure. Unlike previous bootstrap or MCMC based approaches that are only aimed at recovering specific structural features, we learn a concrete density model that can be used for probabilistic generalization. To make use of our objective when some of the data is missing, we propose a bagged structural EM procedure that does not incur the heavy computational cost typically associated with a bootstrap-based approach. We compare our bagged objective to the Bayesian score and the Bayesian information criterion (BIC), as well as other bootstrap-based model selection objectives, and demonstrate its effectiveness in improving generalization performance for varied real-life datasets.

1 Introduction

Multivariate density estimation is an important challenge in a multitude of domains ranging from bioinformatics to fault diagnosis, where scarcity of the data goes hand in hand with the complexity of the domain. A general framework for coping with this task, that has gained wide popularity in recent decades, is the framework of Bayesian networks (BNs) (Pearl, 1988). These models combine a qualitative graph structure that encodes independencies and quantitative parameters to compactly parameterize a joint distribution, allowing for relatively efficient probabilistic computations and estimation. Yet, in complex real-life scenarios with few and partial instances, inferring such models, and in particular a useful (for density estimation) network structure, is still a formidable challenge.

A common approach for learning the structure of BNs is to search for a beneficial structure using a model selection score that aims to approximate the model's predictive ability by balancing the likelihood of the training data given the model and the model's complexity (Lam and Bacchus, 1994; Schwarz, 1978; Heckerman et al., 1995)). Whether via a standard hill-climbing approach, a search in the space of network equivalence classes (Chickering, 2002), or an ordering based algorithm (Teyssier and Koller, 2005), when training data is scarce and partially observed the quality of a learned Bayesian network can be quite poor (see Section 6).

When the goal is *discovery of structural features*, the Bootstrap (Efron and Tibshirani, 1993), a powerful approach for estimating various properties of a given statistic from limited data, has gained popularity in the context of BN model selection (Friedman et al., 1999, 2000; Pe'er et al., 2001; Steck and Jaakkola, 2003; Deforche et al., 2006). These methods evaluate the confidence of a given feature in the network (e.g. a parent-child relationship) by learning B models from B independently sampled bootstrap datasets. The confidence of a particular feature is then estimated via model averaging as the fraction of its occurrence in the B models. Friedman and Koller (2003) offer an alternative approach for this task that relies on an order-based MCMC sampling procedure.

While the benefit of the above methods for discovering known and novel features from data has been demonstrated (e.g., (Pe'er et al., 2001; Friedman et al., 2000; Deforche et al., 2006; Friedman and Koller, 2003)), none of them give rise to a coherent density model. In fact, Friedman et al. (1999), report that using such measures to define a prior over structures *did not* lead to improved generalization. Thus, with the goal of learning a coherent density model, these approaches offer no solution. Further difficulties arise in the par-

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

tial data scenario: bootstrap-based methods are computationally demanding and the above works do not even explore the merit of these procedures in the face of missing data; using the MCMC approach in this context requires further developments (see discussion in Friedman and Koller (2003)).

Our goal is to carry out robust density estimation in the face of scarce and partial data. That is, we aim to infer a *single* BN model that generalizes well.[1] To do so, we rely on bootstrap datasets to compute a robust model selection criteria. Intuitively, instead of averaging over B inferior models, we want to improve local decisions before it is "too late". That is, we want to increase the robustness of structure modifications *during* learning. We make this idea concrete by applying bootstrap aggregation (bagging) (Breiman, 1996) to the Bayesian Information Criterion (BIC) (Schwarz, 1978). Unlike bootstrap-based bias correction, the extensively used (for classification) bagging approach is aimed at reducing the variance of the estimator, and is particularly apt for the scarce data scenario, where the variance of the model selection score is high. Bagging in our case amount to an objective that averages over multiple model selection "experts", each arising from one of the B bootstrap datasets. Importantly, this objective allows us to learn a concrete density model that, as our evaluation demonstrates, generalizes well.

In the face of missing data, when the maximum likelihood parameters cannot be computed in closed form, computation of the score is demanding, and structure learning can be prohibitive. A common solution is to use the structural expectation maximization (SEM) algorithm (Friedman, 1997) that iterates between costly computation of expected sufficient statistics and maximization given these statistics. To avoid incurring the prohibitive cost of a naive incorporation of our bagged objective within an SEM procedure, we present the Bagged Structural Expectation Maximization (BSEM) algorithm: instead of performing computations for each of the bootstrap datasets *independently*, we compute expected statistics for *all* datasets using the maximum likelihood parameters with respect to the *original data*. This leads to significant savings in running time: the costly computation of the distribution of missing values conditioned on the observed variables and the current model is carried out *only once* for each instance at each BSEM iteration. As our method is equivalent to performing (informed) approximate inference for each bootstrap dataset, unlike standard SEM, our BSEM algorithm is guaranteed to improve the learning objective at each iteration only asymptotically. Yet, in practice, the learning objective always improves for a wide range of scarce datasets.

We demonstrate the effectiveness of our bagging structure learning approach for varied discrete and continuous real-life datasets. In all cases, the single coherent model learned by our bagging approach offers consistent and often significant gains in generalization performance, when compared to learning with the Bayesian score (Heckerman et al., 1995; Geiger and Heckerman, 1994) and the BIC score (Schwarz, 1978), as well as alternative bootstrap-based approaches.

2 Background

A *Bayesian Network* (BN) $\mathcal{M} = \langle \mathcal{G}, \Theta \rangle$ (Pearl, 1988) encodes a joint density over a set of random variables $\mathcal{X} = \{X_1, \ldots, X_N\}$. $\mathcal{G}$ is a directed acyclic graph whose vertices correspond to $\mathcal{X}$ and formally encodes a set of independence statements: X_i is independent of its non-descendants given its parents $\mathbf{Pa}_i$ in $\mathcal{G}$. Θ are the of *conditional probability distributions* (CPDs) $P(X_i \mid \mathbf{Pa}_i)$. It can be easily shown that a Bayesian network defines a unique joint probability distribution over $\mathcal{X}$ given by $P(X_1, \ldots, X_N) = \prod_{i=1}^{n} P(X_i \mid \mathbf{Pa}_i)$. The Markov Blanket of X_i (its parents,children and spouses) is the minimal set of variables that render X_i independent of all other variables.

Given $\mathcal{G}$ and training set $\mathcal{D} = \{\mathbf{x}[1], \ldots, \mathbf{x}[M]\}$ of instances of $\mathbf{X} \subset \mathcal{X}$, we look for the *maximum likelihood* parameters $\hat{\theta}$ that maximize the log-likelihood function $\ell(D : \theta, \mathcal{G}) = \sum_m \log P(\mathbf{x}[m] \mid \mathcal{G}, \theta)$. With fully observed data, the log-likelihood also decomposes according to $\mathcal{G}$: $\ell(D : \theta, \mathcal{G}) = \sum_i \sum_m logP(x_i[m] \mid \mathbf{pa}_i[m], \Theta_{X_i, \mathbf{Pa}_i}, \mathcal{G})$. In this case finding $\hat{\theta}$ is typically straightforward. Bayesian parameter estimation, using appropriate priors, amounts to augmenting the empirical sufficient statistics with *pseudo samples* (DeGroot, 1989). Thus, from now on we view the parameter prior as modifying the empirical distribution and omit explicit references to it.

To learn the structure $\mathcal{G}$, we typically rely on a greedy search that examines local structure changes (add, delete or reverse an edge). This search is guided by a scoring function (e.g. MDL (Lam and Bacchus, 1994), BIC (Schwarz, 1978)) that penalizes the likelihood of the data to limit the model complexity

$$\text{score}(\mathcal{G} \ : \ \mathcal{D}) = \ell(\mathcal{D} : \hat{\theta}, \mathcal{G}) - \text{Pen}_M(\mathcal{G}), \qquad (1)$$

where $\hat{\theta}$ are the maximum-likelihood parameters that correspond to the graph $\mathcal{G}$ and $\text{Pen}_M(\mathcal{G})$ is a penalty function that depends on the structure of the graph and number of instances M in $\mathcal{D}$ but not on the data

[1] One might also consider the (orthogonal to ours) approach of model averaging. However, such methods are typically quite time consuming, both at learning and prediction time.

itself. For the BIC score $\text{Pen}_M(\mathcal{G}) = \frac{1}{2}\log(M)|\Theta_\mathcal{G}|$ where $|\Theta_\mathcal{G}|$ is the number of free parameters associated with the graph structure $\mathcal{G}$. The Bayesian score (Heckerman et al., 1995; Geiger and Heckerman, 1994)

$$\log P(D \mid \mathcal{G}) + \log P(\mathcal{G}) = \sum_m \log P(x[m] \mid x[1], \ldots, x[m-1], \mathcal{G}) + \log P(\mathcal{G})$$

is appealing even when $P(\mathcal{G})$ is uniform since the prequential predictive form (each term in the sum) directly approximates the predictive ability of the model.

3 Bootstrap-based Model Selection

The bootstrap (Efron and Tibshirani, 1993) is a general framework for estimating properties of a statistic $T(\mathcal{D})$ given a dataset $\mathcal{D}$ with M samples, that are assumed to be generated from an *unknown* distribution $\mathcal{F}$. Most commonly, the framework is used to estimate the finite sample bias of the statistic $Bias_T = \boldsymbol{E}_{D\sim\mathcal{F}}[T(\mathcal{D})] - T(\mathcal{F})$, where the expectation is with respect to datasets of size M sampled from $\mathcal{F}$, and $T(\mathcal{F})$ denotes the true statistic that we are trying to estimate (e.g., the log-likelihood function). Intuitively, if we had access to a large number B of independent datasets $\mathcal{D}_b \sim \mathcal{F}$ with M samples, we could approximate the bias of $T()$ by using these datasets as a surrogate for $\mathcal{F}$. However, since we only have access to the single training dataset, in the non-parametric bootstrap we create a proxy to the desired scenario by sampling B random datasets from $\mathcal{D}$ (each dataset is independently constructed by randomly choosing M instances from $\mathcal{D}$, with repetitions). The bootstrap simulated estimate of the bias is then

$$\widehat{Bias}_T = \frac{1}{B}\sum_{b=1}^{B} T(\mathcal{D}_b) - T(\mathcal{D}),$$

where $T(\mathcal{D})$ is the *plug-in* statistic with $\mathcal{D}$ replacing $\mathcal{F}$. Using this estimate, we can get an improved (bias-wise) statistic by correcting the original estimate

$$T_{Boot}(\mathcal{D}) = T(\mathcal{D}) - \widehat{Bias}_T = 2T(\mathcal{D}) - \frac{1}{B}\sum_b T(\mathcal{D}_b). \quad (2)$$

Quite remarkably, in a wide range of settings, $T_{Boot}(\mathcal{D})$ reduces the order of bias of $T(\mathcal{D})$ as a function of N. However, this can come at the cost of increasing the variance of the estimate (see Efron and Tibshirani (1993); Shao and D. Tu (1995) for more details).

The bootstrap framework can also provide a confidence measure on $T()$ by evaluating its distribution over the different bootstrap datasets. This idea underlies previous bootstrap-based approaches for model selection (Friedman et al., 1999, 2000; Pe'er et al., 2001) that estimate the confidence of a feature $f(\mathcal{M}(\mathcal{D}))$ of the model given the data (e.g. the existence of an edge) via $C\left[f(\mathcal{M}(\mathcal{D}))\right] \approx \frac{1}{B}\sum_b 1_f(\mathcal{M}(\mathcal{D}_b))$ where $1_f(\mathcal{M}(\mathcal{D}_b))$ is the indicator function for the feature and $\mathcal{M}(\mathcal{D}_b)$ is the network learned from $\mathcal{D}_b$. Steck and Jaakkola (2003) show that applying standard scores to $\mathcal{D}_b$ leads to an additional bias that results from the bootstrap procedure and suggest an analytical correction term (1/2 the number of parameters). We emphasize that both variants result in a bag of confidence measures and not a concrete BN model. Thus, for the purpose of density estimation which is our goal in this work, an alternative approach must be considered.

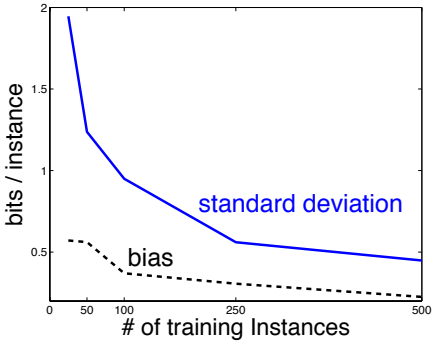

Figure 1: Bias and standard deviation of the log-likelihood per instance function for the synthetic Alarm network (Beinlich et al., 1989). Shown is an estimate based on 500 independent samples (for each training size) and the true structure. Clearly, the variance of the log-likelihood function is significantly higher than its bias, particularly when training data is scarce.

4 Bagging the Likelihood

As noted, previous bootstrap-based approaches for model selection of BNs do not give rise to a coherent density model that can be used for generalization. At the same time, despite its asymptotic appeal, the prequential Bayesian score of Eq. (2) often results in models that generalize poorly in the face of scarce data (see Section 6). To overcome these difficulties, in this work we aim to make use of the finite-sample power of the bootstrap yet learn a concrete density model that is useful in practice. To do so, instead of computing confidence measures via an aggregation of B inferior models, we need to somehow improve the *local decision* (e.g. add an edge) at each step of the learning algorithm. One possibility is to use the bootstrap datasets to estimate the bias of the model selection score of the form Eq. (1), and then apply Eq. (2) to get a better measure for the benefit of structural modifications.

While the above approach appears reasonable, it can be problematic when training data is scarce. In this scenario, the most acute problem is that the variance of the log-likelihood can be quite high, as demonstrated in Figure 1. Thus, because of the bias-variance

trade off, using Eq. (2) to "improve" the score by reducing bias can make things worse. A reasonable alternative motivated by idea of expert aggregation is to average over multiple model selection criteria, each based on a different bootstrap dataset $\mathcal{D}_b$, thereby reducing the variance of the score and ultimately that of the structural decision based on this score.

It is precisely such intuition that motivated the *extensively* used bootstrap aggregation (bagging) approach of Breiman (1996) in the context of classification. In this approach, the original statistic is replaced with $T_{Bag}(\mathcal{D}) = \frac{1}{B}\sum_b T(D_b)$. For model selection scores of the form of Eq. (1) this amounts to replacing the log-likelihood term with its bagged estimate

$$\begin{aligned} T_{Bag}(\mathcal{D}) &\equiv \ell_{Bag}(\mathcal{D} : \{\hat{\theta}^b\}, \mathcal{G}) - \mathrm{Pen}_M(\mathcal{G}) \\ &= (1/B)\textstyle\sum_b \ell(\mathcal{D}_b : \hat{\theta}^b, \mathcal{G}) - \mathrm{Pen}_M(\mathcal{G}) \end{aligned}$$

where $\hat{\theta}^b$ are the maximum likelihood parameters with respect to the bootstrap dataset $\mathcal{D}_b$. It is important to note that T_{Bag} is quite different from the standard bootstrap estimate T_{Boot} of Eq. (2). In fact, although not noted in the original paper of Breiman (1996), T_{Bag} happens to equal (via simple algebra) to *adding* the bootstrap bias estimate to the original estimate of $T(\mathcal{D})$. While this may sound alarming, given the bias-variance trade-off, it should not come as a great surprise when we explicitly aim to reduce variance.

Our bagged objective is the sum of independent terms and thus finding the maximum likelihood parameters $\{\hat{\theta}^b\}$ can be carried out by maximizing the likelihood with respect to each D_b independently $\hat{\theta}^b = \arg\max_\theta \ell(\mathcal{D}_b : \theta)$. This can be done efficiently when the data is complete and, as we demonstrate in Section 6, leads to consistent improvement in generalization performance. In the next section we present a bagged structural expectation maximization approach that allows us to use our bagged score in the computationally intensive scenario of partial observations.

5 Bagged Structural EM (BSEM)

In the case of partial observations, the maximum likelihood parameters can no longer be found in closed form and we typically use the Structural Expectation Maximization (SEM) algorithm (Friedman, 1997; Dempster et al., 1977). Building on the ability to carry out estimation when the data is complete, the SEM algorithm iterates between "guessing" the missing values using the current parameters (E-Step) and then using the "completed" data to maximize the parameters and structure (M-Step). Concretely, given the maximum likelihood parameters $\hat{\theta}^{\mathrm{t}}$ and model $\mathcal{G}^{\mathrm{t}}$ of the previous iteration t, we find the graph and parameters that (locally) maximize the *expected score*

$$\begin{aligned} &Q(\mathcal{G}, \theta \mid \mathcal{G}^{\mathrm{t}}, \hat{\theta}^{\mathrm{t}}) \equiv \\ &\textstyle\sum_m \boldsymbol{E}_{P(h[m]|\hat{\theta}^{\mathrm{t}})}[\log P(x[m], h[m] \mid \theta, \mathcal{G})] - \mathrm{Pen}_M(\mathcal{G}) \end{aligned}$$

where $P(h[m]|\hat{\theta}^{\mathrm{t}})$ is a shorthand for $P(h[m]|x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G}^{\mathrm{t}})$ so that the expectation of the hidden variables $h[m]$ for each instance is computed with respect to the model of the previous iteration. Appealingly, the penalized log-likelihood of the *observed data* can only increase at each iteration of the algorithm, and convergence to a local maximum is guaranteed. Note, however, that this only holds when exact inference is used. When approximate inference is needed, SEM is no longer guaranteed to improve at each iteration or converge at all.

5.1 The BSEM Algorithm

While typically effective, the SEM procedure can be costly as the computation of $P(h[m] \mid x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G})$ is in general NP-hard and often quite demanding. Thus, applying SEM independently to each of the B bootstrap datasets can be prohibitive. Instead, we now present a bagged SEM procedure that offers dramatic computational savings over independent optimization.

The idea is straightforward and builds on the fact that computation of statistics of $\mathcal{D}_b$ amounts to collection of these statistics from instances of $\mathcal{D}$, weighed according to their frequency in $\mathcal{D}_b$. This suggests that in the E-Step, instead of computing $P(h[m] \mid x[m], \hat{\theta}^{b,\mathrm{t}}, \mathcal{G})$ for each b, we compute $P(h[m] \mid x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G})$ *once for each instance*, and then efficiently collect the appropriate statistics for each dataset D_b. Concretely, we define our learning objective to be the *bagged expected score*:

$$\begin{aligned} &Q_{Bag}(\mathcal{G}, \{\theta^b\} \mid \mathcal{G}^t, \hat{\theta}^{\mathrm{t}}) \equiv \\ &\tfrac{1}{B}\textstyle\sum_b \sum_{m \in \mathcal{D}_b} \boldsymbol{E}_{P(h[m]|\hat{\theta}^{\mathrm{t}})}\big[\log P(x[m], h[m]|\theta^b, \mathcal{G})\big] - \mathrm{Pen}(\mathcal{G}). \end{aligned} \tag{3}$$

Note that this can be viewed as standard SEM where (informed) approximate inference is applied to each dataset D_b. Thus, given $\mathcal{G}$ and the posterior probability of $h[m]$ for each instance, finding *each* of the $\hat{\theta}^{b,\mathrm{t}+1}$ that maximize this objective is straightforward as it amounts to maximization with respect to a complete dataset. As an example, for the multinomial sufficient statistics counts $M(x_i, \mathbf{pa}_i)$, using $w_b[m]$ to denote the number of times instance m appears in $\mathcal{D}_b$, we can readily compute the *expected sufficient statistics*

$$\begin{aligned} &\boldsymbol{E}_{P(H|\mathcal{D},\hat{\theta}^{\mathrm{t}})}[M_b(x_i, \mathbf{pa}_i)] = \\ &\textstyle\sum_{m \in \mathcal{D}_b} w_b[m] P(x_i[m] = x_i, \mathbf{pa}_i[m] = \mathbf{pa}_i \mid x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G}). \end{aligned}$$

From these statistics, $\hat{\theta}^{b,\mathrm{t}+1}$ can then be computed in closed form. Estimation for more general parameters of the exponential family is similarly straightforward.

Algorithm 1: Bagged Structural EM.
Input: Training set $\mathcal{D}$. **Output**: $\mathcal{G}$, parameters Θ.

$\mathcal{G}^0 \leftarrow$ empty graph, $\hat{\theta}^0 \leftarrow$ random // initialization

foreach $t = 0, 1, \ldots$ *until convergence* **do**
  // `common E-Step computation`
  compute $P(h[m] \mid x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G}^{\mathrm{t}})$ for each m
  // **greedy** `search upto local maximum`
  $\mathcal{G}^{\mathrm{t}+1} = \arg\max Q_{Bag}(\mathcal{G}, \{\theta^b\}, \mathcal{G}^{\mathrm{t}}, \hat{\theta}^{\mathrm{t}})$
  // `e.g., full parametric EM`
  $\hat{\theta}^{\mathrm{t}+1} = \arg\max Q(\mathcal{G}^{\mathrm{t}+1}, \theta, \mathcal{G}^{\mathrm{t}}, \hat{\theta}^{\mathrm{t}})$
  // `using last completion of the data`
  compute $\{\hat{\theta}^{b,\mathrm{t}+1}\}$

return $\mathcal{G}^{\mathrm{t}+1}, \hat{\theta}^{\mathrm{t}+1}$

We can now proceed to learn the structure of a BN model by plugging our bagged objective of Eq. (3) into the structural EM algorithm (Friedman, 1997) as outlined in Algorithm 1: at each iteration, after computing posterior probabilities *once for each instance* using the standard maximum-likelihood parameters, we search for the structure that (locally) maximizes the bagged expected score (using, for example, a standard greedy approach that relies on local structure modifications). We can then optimize parameters with respect to this structure using standard parametric EM. At convergence, we return the maximum likelihood parameters corresponding to this structure with respect to the original data $\mathcal{D}$.[2] Importantly, the computation of $P(h[m] \mid x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G}^{\mathrm{t}})$ dominates the collection of sufficient statistics and indeed the entire algorithm. Thus, in practice the running time of our algorithm grows slowly in the number of bootstrap datasets B.

5.2 Convergence of BSEM

The obvious question is whether our bagged SEM procedure is guaranteed to improve our bagged likelihood objective of Eq. (3) at each iteration as is the case in the standard SEM algorithm. Since we are in effect performing approximate inference with respect to each bootstrap dataset, this is not always true and a theoretical guarantee only exists asymptotically:

Theorem 5.1 : For CPD parameters that are a smooth continuous function of the sufficient statistics of the data, in the limit $M \to \infty$

$$\ell_{Bag}(\mathcal{D} : \{\hat{\theta}^{b,\mathrm{t}+1}\}, \mathcal{G}^{\mathrm{t}+1}) - \ell_{Bag}(\mathcal{D} : \{\hat{\theta}^{b,\mathrm{t}}\}, \mathcal{G}^{\mathrm{t}}) \quad \dot{\geq} \quad 0$$

where $\dot{\geq}$ denotes greater than or equal in probability.

[2] We can also return a bias corrected or bagged estimate instead. This, however, did not result in noticeable differences in our experiments.

Proof: (**Outline**) Using $\boldsymbol{E}_{P(h)}[logP(x)] = logP(x)$ (since $P(x)$ does not depend on $P(h)$), and straightforward algebra, the likelihood difference in then left-hand side of the theorem equals to

$$\sum_b \sum_{m \in \mathcal{D}_b} \boldsymbol{E}_{P(h[m]|\hat{\theta}^{\mathrm{t}})} \left[\log \frac{P(x[m], h[m]|\hat{\theta}^{b,\mathrm{t}+1}, \mathcal{G}^{\mathrm{t}+1})}{P(x[m], h[m]|\hat{\theta}^{b,\mathrm{t}}, \mathcal{G}^{\mathrm{t}})} \right]$$
$$+ \quad \sum_b \sum_m \boldsymbol{D}(P(h[m]|\hat{\theta}^{\mathrm{t}}) \| P(h[m]|x[m], \hat{\theta}^{b,\mathrm{t}+1}, \mathcal{G}^{\mathrm{t}+1}))$$
$$- \quad \sum_b \sum_m \boldsymbol{D}(P(h[m]|\hat{\theta}^{\mathrm{t}}) \| P(h[m]|x[m], \hat{\theta}^{b,\mathrm{t}}, \mathcal{G}^{\mathrm{t}})),$$

where $\boldsymbol{D}(\|)$ is the Kullback-Leibler divergence, and using the shorthand $P(h[m]|\hat{\theta}^{\mathrm{t}}) \equiv P(h[m]|x[m], \hat{\theta}^{\mathrm{t}}, \mathcal{G}^{\mathrm{t}})$. Our assumption on the parameters implies, using standard asymptotic results (Lehmann, 1999), that as M_b grows $\hat{\theta}^{b,\mathrm{t}} \to \hat{\theta}^{\mathrm{t}}$. Thus, since $P(h[m]|x[m], \theta, \mathcal{G})$ is a smooth function of θ, the second $\boldsymbol{D}(\|)$ approaches zero, and the first $\boldsymbol{D}(\|)$ can only be greater. The result follows from the fact that the first line is non-negative by the construction of the maximization step. ∎

Admittedly, the asymptotic result seems of little use when our goal is to improve model selection in the face of scarce data. However, the above proof provides insight as to when we may expect our procedure to improve the bagged objective at each iteration: when the distribution of the hidden values H given $\hat{\theta}^{b,\mathrm{t}}$ is closer to the distribution of H given $\hat{\theta}^{\mathrm{t}}$ than to the distribution of H given $\hat{\theta}^{b,\mathrm{t}+1}$. Intuitively, this is to be expected since $\hat{\theta}^{b,\mathrm{t}}$ are computed based on samples from the instances used to estimate $\hat{\theta}^{\mathrm{t}}$. In practice, since many independent $\boldsymbol{D}(\|)$ differences are summed over independent instances and datasets, the chances that the overall difference will be positive increase.

As noted, our approach can be viewed as performing approximate inference at each EM iteration and the above limitation is to be expected. That said, exceeding our expectations, for *all repetitions in all domains* considered in our experimental evaluation, the BSEM algorithm always improved the bagged objective, until convergence. Thus, our results suggests that the fact that our method *forces* approximate inference does not have significant practical ramifications.

6 Experimental Evaluation

To evaluate the merit of our method, we learn the structure of a BN using a standard greedy approach with local edge modifications. We compare our **BSEM** approach that uses the bagged objective of Eq. (3) to the standard **BIC** score (Schwarz, 1978). We also compare to the **Bayesian** approach using the BDeu score (Heckerman et al., 1995) with an

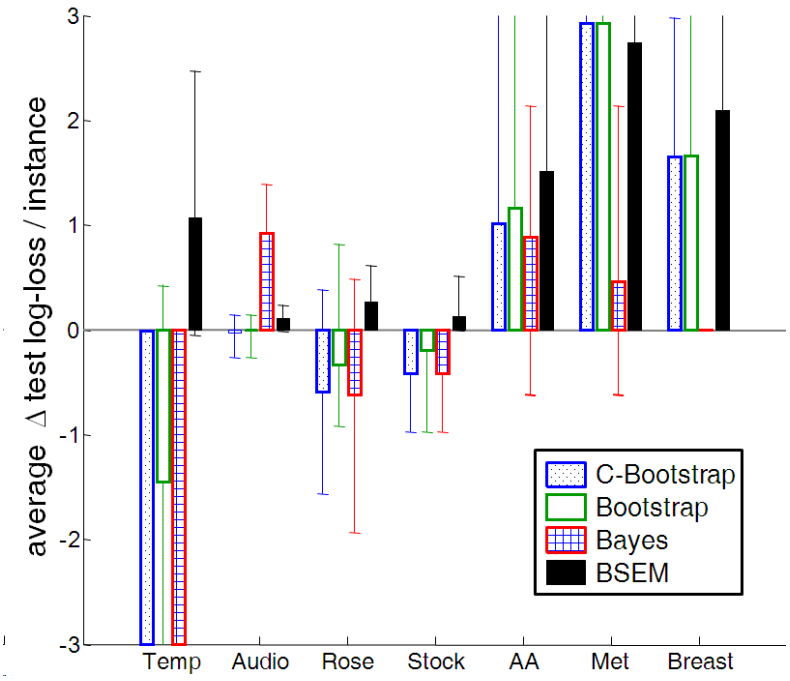

Figure 2: Comparison of our bagged structural EM (**BSEM**) method to structure learning with the **Bayesian** score, the standard **Bootstrap** correction of the BIC score and the **C-Bootstrap** bias corrected variant (Steck and Jaakkola, 2003). Shown is the mean (bar) and full range (error bars) of the test log-loss/instance relative to the **BIC** baseline (dotted line at 0) for the seven real life datasets we consider.

equivalent sample size of 1 (an experiment for other values is included below) for the discrete domains[3], and the BGe score (Geiger and Heckerman, 1994) with an Inverse-Wishart standardized prior of equal strength for the continuous datasets. We also compare to a standard bias correction **Bootstrap** estimate of the BIC score by applying Eq. (2) to the likelihood term. Finally, we also compare a **C-Bootstrap** variant with the additional correction suggested by Steck and Jaakkola (2003). We emphasize that we *integrate* both bootstrap competitors into the learning process and thus produce a coherent model which can be compared to ours. We also note that all the methods considered differ only in the way that the structure of the network is evaluated, and parameter estimation (including the prior) is identical in all cases.

We start by demonstrating the limited ability of all scores to recover structure when faced with scarce training data. We attempted to recover the structure of the 37 variable Alarm network (Beinlich et al., 1989) from synthetically generated instances where 25% of the values were randomly hidden. With 250 instances (a higher instance/variable ratio than in the real-life datasets we consider below), all methods recovered on average only $\sim 30\%$ of the Markov Blanket (MB) neighborhood. Although our objective was slightly (yet consistently) superior to the other scores, differences were small $(1-2\%)$. With 100 instances all scores perform poorly and less than 10% of the MB was recovered on average. Thus exemplifies that in non-trivial domains we cannot aim at a density model and at the same time expect significant true structural recovery. We now turn to our true goal: model selection for the purpose of probabilistic density estimation, which we evaluate using log-probability (or log-loss) performance on unseen test instances. We consider the following real-life datasets:

- **Temperature**: measurements of 54 sensors discretized into 4 bins (Deshpande et al., 2004). 100 training instances were randomly chosen, and 25% of the values were randomly hidden.
- **Audiology**: 69 variables relating to audiology disfunctions (Bareiss and Porter, 1987). 226 instances with missing values were randomly split into equal train and test sets.
- **Rosetta**: partially missing gene expression of 6000 *Saccharomyces cerevisiae* genes in 300 experiments (Hughes et al., 2000). We discretized the data and concentrated on 35 *stationary phase* genes as in [anonymous citation]. 100 train and 200 test instances were chosen randomly.
- **Stock**: up/down changes of 20 US technology stocks in 1516 trading days (Boyen et al., 1999). 100 samples were chosen randomly for training and 25% of the the value were randomly hidden.
- **Gasch**: expression of the baker's yeast genes in 173 experiments (Gasch et al., 2000). As in [anonymous citation], we concentrated on 44 *amino acid* (AA) genes with all values observed and 89 *general*

[3] We also tried a corrected BDeu that corrects the bias of deterministic CPDs (common when the data is sparse). Results however were on average and almost always worse

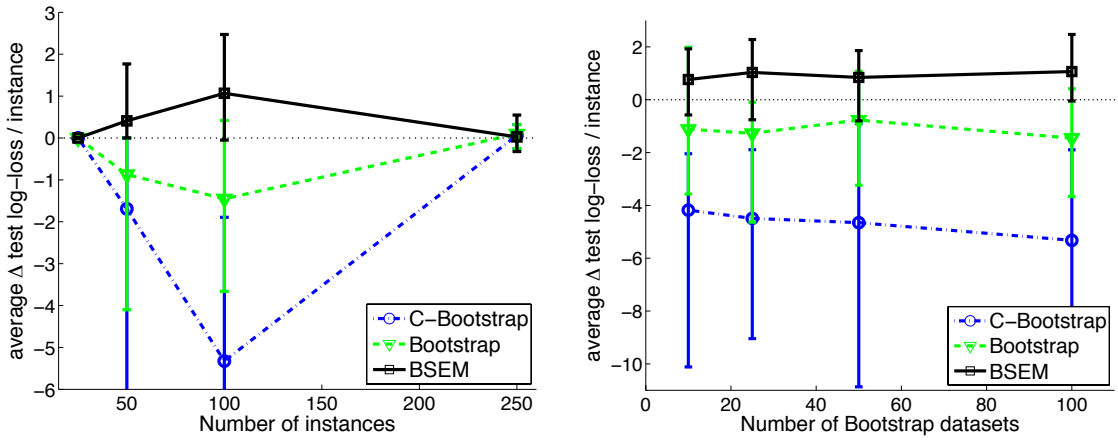

Figure 3: Comparison of our **BSEM** method to the the standard **Bootstrap** correction and the **C-Bootstrap** bias corrected variant for the Temperature dataset (the **Bayesian** approach which was consistently and significantly worse is not included for readability). Shown is the average and full range (across random repetitions) test log-probability (log-loss) per instance relative to the **BIC** baseline (dotted line at 0) as a function of (left) the number of training instances and (right) the number of bootstrap datasets used.

metabolic process (MET) genes. Results reported are for linear Gaussian models.

- **Breast** (prognosis): 198 records with 34 continuous standardized attributes (Street et al., 1996) were randomly split into equal train/test sets. 25% of the values were randomly hidden.

Results reported are over 5 random train/test split repetitions for the larger Met domain and 10 repetitions for all other domains. Figure 2 compares the benefit in test log-loss per instance of test data of the different model selection objectives relative to the standard **BIC** baseline (thin dotted black line at 0). Shown is the average result (bar) as well as the *full range* across the random splits (vertical error bars). While **Bootstrap** (green outline bar) and **C-Bootstrap** (blue dotted bar) work reasonably well on the continuous datasets AA, Met and Breast, these bootstrap variants are inferior to the **BIC** baseline on the discrete domains and often significantly so. This is probably due to the fact that in the discrete scenario, the sufficient statistics can undergo more drastic transitions as the data becomes scarcer. The **Bayesian** approach (grid fill red) offers an advantage only for the Audiology dataset, and is competitive only in one other case (AA). For three datasets, the **Bayesian** approach results in significant degradation with respect to the BIC baseline (result of **Bayesian** for the Breast dataset is missing since using the Bayesian score for the continuous dataset with missing values is too prohibitive). In contrast, our **BSEM** approach (solid black bar) *consistently dominates* the baseline, and is better than *all* other methods in 5 of the 7 domains. In fact, even its worst case performance is quite appealing: across all datasets and random repetitions it is superior to the baseline in 63/65 experiments and is only marginally inferior for 2/65 cases (one repetition for the Temperature and Audio datasets). Importantly, the improvement of our method is often significant: an advantage of, for example, half a bit per instance (y-axis) over as little as 50 test instances amounts to the test data being 2^{25} times more likely. Finally, note that by bagging the likelihood, as expected, we often achieve a significant decrease in variance of the quality of the model learned. This is particularly evident when comparing our method to the **Bayesian** score, which exhibits significant variability for all datasets.

To get a better understanding of the behavior of our algorithm, we consider performance in more detail for the Temperature dataset. Figure 3(a) compares the average (marker) and full range (error bars) test log-loss per instance of the different methods as a function of the number of instances M (the **Bayesian** method which was significantly worse is not shown for readability). At 25 instances, the signal in the data is too weak to overcome the penalty term, and for all random repetitions all methods resulted in the empty network. As M increases, our **BSEM** approach quickly takes advantage of beneficial signals and improves over the baseline as well as the other competitors. As expected, when the M grows further, all methods are essentially equivalent. What is striking is the degradation in performance of the competitors exactly in the "golden range" where our **BSEM** method offers the greatest advantage. To understand this phenomenon, we recall that **BSEM** attempts to average over scores so as to reduce the variance, an apt approach when the signal in the data is still quite noisy. The **Bootstrap** and **C-Bootstrap** scores, on the other hand, attempt to correct the bias of the score thereby increasing its

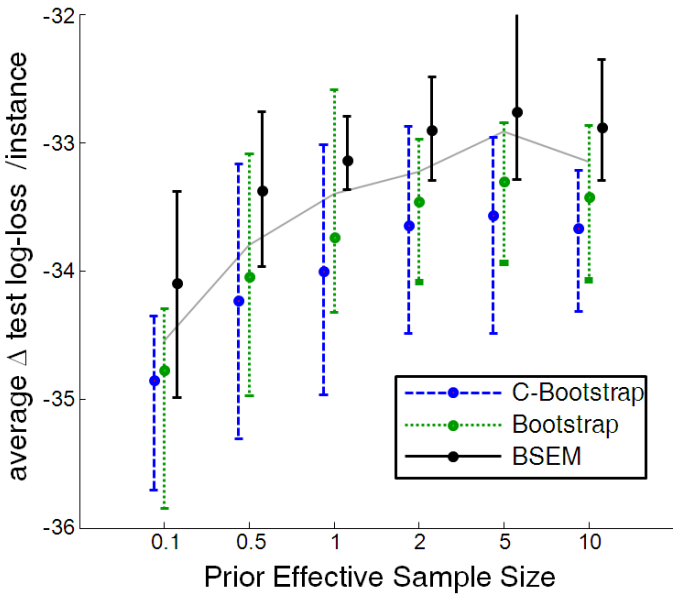

Figure 4: Test log-loss instance performance (y-axis) as a function of the effective sample size of the Dirichlet Prior (x-axis) for the Rosetta domain. Shown is the mean (circle) and full range (error bars) relative to the **BIC** (solid gray line). We compare our bagged structural EM (**BSEM**) method to the standard **Bootstrap** correction of the BIC score and the **C-Bootstrap** bias corrected variant (Steck and Jaakkola, 2003) (the performance of the **Bayesian** score was significantly worse than all others and is not shown for readability).

variance due to the bias-variance trade-off, which ultimately leads to a degradation of performance. It is also interesting to note that **C-Bootstrap** is actually inferior to the standard bootstrap bias correction as it relies on the assumption that "BIC is obviously intended to be applied to the given data" (Steck and Jaakkola, 2003). We argue that in the scarce data scenario, this assumption is problematic since the signals in the given data may be too noisy.

The effect of the number of bootstrap datasets B on our performance is shown in Figure 3(b). As B increases and additional "experts" are available, we benefits from the "wisdom of crowds" phenomenon, mostly in term of worse case performance. With $B = 100$, our method improves over the baseline in 9 out of the 10 random repetitions and is essentially the same in the last. In contrast, the other bootstrap-based methods, which do not attempt to reduce variance, continue to exhibit high variability and a performance that is significantly inferior to the baseline.

Figure 4 explores the behavior of the different model selection scores as a function of the equivalent sample size (ESS) of the prior for the **Rosetta** domain (results were similar for the other discrete domains with the peak performance at ESS=1 or ESS=2). As can be clearly seen, the advantage of our **BSEM** approach over the BIC baseline (solid black) as well as the other competitors is both evident and consistent for ESS values that span two orders of magnitudes.

Finally, we consider the running time of our algorithm. Even with a crude preliminary implementation that caches local scores but does not fully share repeated post-inference computations between different bootstrap datasets, our **BSEM** runs took only 7-12 times longer than standard structure learning with $B = 100$. This is significantly better than a running time factor of B we typically expect when bootstrap aggregation is used to wrap a learning procedure.

7 Discussion and Future Work

In this work we introduced a novel approach for robustly learning a Bayesian network multivariate density model from scarce data. Our approach relies on a bootstrap aggregation model selection objective that sifts out inferior structural choices during the learning procedure. We presented BSEM, an adaptation of the Structural EM algorithm that allows for efficient application of our objective when observations are both scarce and partial. We demonstrated the effectiveness of our approach for probabilistic generalization when compared to learning with the BIC and Bayesian model selection criteria as well as other bootstrap-based variants on a range of real-life datasets.

Our contribution is three-fold. First, in contrast to previous bootstrap-based model selection approaches, our method leads to a concrete multivariate density model that can be used for generalization (Friedman et al., 1999; Steck and Jaakkola, 2003). Second, when the data is particularly scarce, our objective provides a robust model selection criteria that improves performance over BIC (Schwarz, 1978) and the Bayesian score (Heckerman et al., 1995), as well as the other bootstrap-based alternative we considered. Third, we propose a practical EM adaptation that facilitates the use of our approach in the challenging scenario of learning with missing value, where alternative MCMC and model averaging methods are not practical.

While we focused on learning the structure of BNs (a challenge quite sensitive to the scarcity of data), the idea of bagging a model selection objective can be readily adapted to other settings, and we intend to explore the effectiveness of such an approach in improving other hypothesis exploration algorithms. More generally, it would be theoretically interesting and practically useful to develop soft measures that automatically balance the two extremes of either subtracting the expected bias for the maximum likelihood estimate or, as is done in bagging, simply adding it.

Acknowledgements

I am grateful to Amir Globerson for his comments on an earlier draft of this manuscript. Gal Elidan was supported by the Alon fellowship.

References

R. Bareiss and B. Porter. Protos: An exemplar-based learning apprentice. *Proc of the International Workshop on Machine Learning*, 1987.

I. Beinlich and G. Suermondt and R. Chavez and G. Cooper. The ALARM monitoring system. *Proc of the European Conf on AI and Medicine*, 1989.

X. Boyen, N. Friedman, and D. Koller. Discovering the hidden structure of complex dynamic systems. *Conf on Uncertainty in AI (UAI)*, 1999.

L. Breiman. Bagging predictors. *ML*, 1996.

D. Chickering. Optimal structure identification with greedy search. *Journal of ML Research*, 2002.

T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

A. Dawid. Statistical theory: The prequential approach. *J of the Royal Statistical Society*, 1984.

K. Deforche, T. Silander, R. Camacho, Z. Grossman, M. Soares, K. Van Laethem, R. Kantor, Y. Moreau, and A. Vandamme. Analysis of hiv-1 pol sequences using bayesian networks: implications for drug resistance. *Bioinformatics*, 2006.

M. DeGroot. *Probability and Statistics*. 1989.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J of the Royal Statistical Society*, B 39:1–39, 1977.

A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. *Very Large Databases Conf*, 2004.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, London, 1993.

N. Friedman. Learning belief networks in the presence of missing values and hidden variables. *International Conf on ML (ICML)*, 1997.

N. Friedman, M. Goldszmidt, and A. Wyner. Data analysis with Bayesian networks: A bootstrap approach. *Conf on Uncertainty in AI (UAI)*, 1999.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Conf on Research in Computational Molecular Biology (RECOMB)*, 2000.

N. Friedman and D. Koller. Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. *ML*, 2003.

A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel, M. Eisen, G. Storz, D. Botstein, and P. Brown. Genomic expression program in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.

D. Geiger and D. Heckerman. Learning Gaussian networks. *Conf on Uncertainty in AI (UAI)*, 1994.

D. Heckerman. A tutorial on learning with Bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*. Kluwer, Dordrecht, Netherlands, 1998.

D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *ML*, 20:197–243, 1995.

T. Hughes, M. Marton, A. Jones, C. Roberts, R. Stoughton, C. Armour, H. Bennett, E. Coffey, H. Dai, Y. He, M. Kidd, A. King, M. Meyer, D. Slade, P. Lum, S. Stepaniants, D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, S. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–26, 2000.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. 2009.

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

E. Lehmann. *Elements of Large-Sample Theory*. 1999.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 1988.

D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(Suppl 1):S215–24, 2001.

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

J. Shao and D. Tu. *The Jackknife and Bootstrap*, 1995.

H. Steck and T. Jaakkola. Bias-corrected bootstrap and model uncertainty. *Conf on Neural Information Processing Systems (NIPS)*, 2003.

W. Street, O. Mangasarian, and W. Wolberg. An inductive learning approach to prognostic prediction. *International Conf on ML (ICML)*, 1996.

M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. *Conf on Uncertainty in AI (UAI)*, 2005.