# Learning Dynamic Bayesian Network Models Via Cross-Validation

Jose M. Peña[1,*], Johan Björkegren[2] and Jesper Tegnér[1,2]
[1]Computational Biology, Dept. of Physics, Linköping University, 58183 Linköping, Sweden
[2]Center for Genomics and Bioinformatics, Karolinska Institutet, 17177 Stockholm, Sweden

## Abstract

We study cross-validation as a scoring criterion for learning dynamic Bayesian network models that generalize well. We argue that cross-validation is more suitable than the Bayesian scoring criterion for one of the most common interpretations of generalization. We confirm this by carrying out an experimental comparison of cross-validation and the Bayesian scoring criterion, as implemented by the Bayesian Dirichlet metric and the Bayesian information criterion. The results show that cross-validation leads to models that generalize better for a wide range of sample sizes.

**Keywords:** Dynamic Bayesian network models, cross-validation, learning.

## 1 Motivation

Let $X^t = \{X_1^t, \ldots, X_I^t\}$ denote a set of $I$ discrete random variables that represents the state of a temporal process at a discrete time point $t$. A dynamic Bayesian network (DBN) is a pair $(G, \theta)$ that models the temporal process by specifying a probability distribution for $X^0, \ldots, X^T$, $p(X^0, \ldots, X^T | G, \theta)$ [7, 18]. The first component of the DBN, $G$, is an acyclic directed graph (DAG) whose nodes correspond to the random variables in $X^0$ and $X^1$. Edges from $X^1$ to $X^0$ are not allowed because they do not conform with the arrow of time. The second component of the DBN, $\theta$, is a set of parameters specifying a conditional probability distribution for each node $X_i^t$ in $G$ given its parents $Pa(X_i^t)$ in $G$, $p(X_i^t | Pa(X_i^t), G, \theta)$. In this paper, all these conditional probability distributions are multinomial, which is the most common choice. We call $G$ the (DBN) model and $\theta$ the (DBN) parameters. A DBN represents $p(X^0, \ldots, X^T | G, \theta)$ through the factorization

$$p(X^0, \ldots, X^T | G, \theta) = \prod_{t=0}^{T} \prod_{i=1}^{I} p(X_i^t | Pa(X_i^t), G, \theta) \tag{1}$$

where $Pa(X_i^t) = \{X_j^{t-1} | X_j^0 \in Pa(X_i^1)\} \cup \{X_j^t | X_j^1 \in Pa(X_i^1)\}$ and $p(X_i^t | Pa(X_i^t), G, \theta) = p(X_i^1 | Pa(X_i^1), G, \theta)$ for $t > 1$. Note that we implicitly assume that $X_i^0, \ldots, X_i^T$ have all the same set of possible values. Note also that our definition of DBNs constrains the temporal processes that can be modelled to be both first-order Markov, i.e. $p(X^t | X^0, \ldots, X^{t-1}, G, \theta) = p(X^t | X^{t-1}, G, \theta)$, and stationary, i.e. $p(X^t | X^{t-1}, G, \theta)$ is the same for all $t$. These constraints can be easily removed. However, they are commonly adopted because they reduce the complexity of the DBNs under consideration, which can be otherwise overwhelming, particularly for large values of $T$ [7, 18].

Learning a DBN model from data aims to find the best model of the unknown probability distribution underlying the temporal process on the basis of a random sample of finite size, i.e. the learning data. The goodness of a model is evaluated with the help of a scoring criterion, which represents our preferences for the models. Let $D = \{D_1, \ldots, D_S\}$ denote the learning data, which consists of $S$ independent and identically distributed time series. Each $D_s$ specifies values for the random variables $X^0, \ldots, X^{T_s}$. The

[*] To whom correspondence should be addressed (e-mail: jmp@ifm.liu.se, phone: +46-13-281000, fax: +46-13-137568).

Bayesian scoring criterion (BSC) is probably the most commonly used scoring criterion. The BSC value of a model $G$ given the learning data $D$ is defined as

$$\log p(D, G) = \log p(D|G) + \log p(G). \qquad (2)$$

For simplicity, $p(G)$ is usually assumed to be uniform. In this paper, we make this assumption as well. Thus, BSC is equivalent to $\log p(D|G)$. This means that BSC scores the likelihood of $G$ having generated $D$. According to [4], BSC can also be interpreted as follows. From the chain rule of probability, we have

$$\log p(D|G) = \sum_{s=1}^{S} \log p(D_s|D_1, \ldots, D_{s-1}, G) \qquad (3)$$

where $p(D_s|D_1, \ldots, D_{s-1}, G)$ represents the predictive accuracy of $G$ for $D_s$ given $D_1, \ldots, D_{s-1}$ after averaging over $\theta$. The log in front of $p(D_s|D_1, \ldots, D_{s-1}, G)$ can be thought of as the utility function for prediction. Thus, BSC scores the accuracy of $G$ as a sequential predictor of $D$ under the log utility function. This means that BSC summarizes not only how well a model fits the learning data but also how well it generalizes to unseen data. Scoring the generalization ability of the models is crucial because it prevents overfitting and, thus, guarantees a good approximation to the unknown probability distribution of the temporal process that generated the learning data.

In this paper, we aim to learn DBN models that generalize well. We interpret the generalization ability of a model $G$ as the expected predictive accuracy for the next time series, $D_{S+1}$, after plugging the maximum likelihood (ML) or maximum a posteriori (MAP) parameters obtained from $D$, $\hat{\theta}$, into $G$, i.e. $E[\log p(D_{S+1}|G, \hat{\theta})]$. As in BSC, we consider the log utility function. This is a very common interpretation of the generalization ability of a model[1] but, unfortunately, BSC does not fully conform to it for the following three reasons, which have been previously discussed in [4]. First, we are interested in the predictive accuracy for $D_{S+1}$ given the $S$ time series already seen, i.e. $D$. In contrast, BSC combines the accuracy of predictions based on $0, 1, 2, \ldots, S-1$ time series, i.e. all the predictions are based on less than $S$ time series and some of them in many less than $S$. Second, we are interested in the expected predictive accuracy for $D_{S+1}$ because $D_{S+1}$ is unknown. In contrast, BSC combines the accuracy of predictions for known time series, i.e. $D_s$ is known when making the prediction based on $D_1, \ldots, D_{s-1}$. Third, we are interested in the predictive accuracy after plugging $\hat{\theta}$ into $G$. In contrast, BSC averages the predictive accuracy over all the possible values of $\theta$. Consequently, BSC is not fully in line with our preferences for the models. As we will see, this substantially harms generalization.

Unfortunately, the exact evaluation of $E[\log p(D_{S+1}|G, \hat{\theta})]$ is computationally unfeasible in all but small domains, because it implies summing over all $D_{S+1}$ and all $G$. That is

$$\begin{aligned} &E[\log p(D_{S+1}|G, \hat{\theta})] \\ &= \sum_{D_{S+1}} p(D_{S+1}|D) \log p(D_{S+1}|G, \hat{\theta}) \\ &= \sum_{D_{S+1}} \Big[ \sum_{G} p(G|D) p(D_{S+1}|D, G) \Big] \log p(D_{S+1}|G, \hat{\theta}). \end{aligned} \qquad (4)$$

In this paper, we propose $K$-fold cross-validation (CV) as a computationally feasible scoring criterion for learning DBN models that generalize well under our interpretation of generalization. The CV value of a model $G$ given the learning data $D$ is computed as follows. First, $D$ is randomly split into $K$ mutually exclusive subsets or folds $D^1, \ldots, D^K$ of approximately equal size. Then, the predictive accuracy of $G$ for $D^k$ after plugging the ML or MAP parameters obtained from $D \setminus D^k$, $\hat{\theta}^k$, into $G$, i.e. $\log p(D^k|G, \hat{\theta}^k)$, is calculated for all $k$. Finally, the CV value of $G$ given $D$ is computed as

$$\frac{1}{S} \sum_{k=1}^{K} \log p(D^k|G, \hat{\theta}^k). \qquad (5)$$

CV is intended to estimate $E[\log p(D_{S+1}|G, \hat{\theta})]$. Obviously, CV departures from this aim in that it

[1] For instance, all those papers that measure the generalization ability of a model as the cross-entropy or log-loss of the model after plugging the ML or MAP parameters into it agree with our interpretation of generalization.

combines the accuracy of predictions based on less than $S$ time series. We believe that this is a minor departure if $K$ is large enough. Thus, we hypothesize that CV complies better than BSC with our interpretation of the generalization ability of a model. The experimental results that we report in Section 2 confirm this hypothesis: CV leads to models that generalize better than those obtained by BSC for a wide range of sample sizes. It is worth mentioning that, in the experiments, we consider two implementations of BSC: On one hand, the Bayesian Dirichlet metric (BD) [7, 8] which calculates BSC exactly and, on the other hand, the Bayesian information criterion (BIC) [7, 23] which is an asymptotic approximation to BSC. CV outperforms both implementations of BSC.

The remaining two sections of this paper are devoted to the experimental comparison of CV and BSC, and to the discussion of this and related works.

# 2 Experiments

In this section, we evaluate CV as a scoring criterion for learning DBN models that generalize well. We use BSC (BD and BIC implementations) as benchmark. All the experiments involve data sampled from known DBNs. This enables us to assess the topological accuracy of the models learnt, in addition to their generalization ability. We first describe the experimental setting.

## 2.1 Experimental Setting

All the learning databases in the experiments involve between 20 and 40 nodes. This prohibits performing an exhaustive search for the highest scoring model and, thus, we turn to heuristics. Specifically, we use a greedy hill-climbing search: We start from the empty graph and, gradually, improve it by applying the highest scoring single edge addition or removal available. This is a popular search strategy due to its simplicity and good performance [7, 8].

The version of CV that we use in the experiments is 10 times 10-fold cross-validation, i.e. we average 10 runs of 10-fold cross-validation with different folds in each run. The folds are the same for all the models evaluated. This setting guarantees a good replicability of the results [2, 14]. It is worth mentioning that CV shares two important properties with BD and BIC. First, CV decomposes into local scores, one for each node and its parents. This means that scoring an edge addition or removal in the greedy hill-climbing search requires computing a single local score. Second, all the sufficient statistics required in each evaluation of CV can be computed in a single pass of the learning data at the expense of storage space. This is typically the most time consuming step in the evaluation of CV, BD and BIC.

CV prevents overfitting by recommending the addition of only those edges that seem to be beneficial for generalization. This excludes the vast majority of false positive edges. However, a considerable number of false positive edges can still get recommended just by chance due to the noisy and/or finite nature of the learning data. Solving this problem is crucial for CV to be a competitive scoring criterion. The overfitting problem of CV has been previously noticed in [9, 19]. Despite these works are not concerned with learning DBN models from data, their arguments are general and apply to this task as well. In [9], the authors suggest that overfitting occurs due to the large variance of CV and propose solving it by adding a penalty to CV. The penalty is up to the user as the authors do not provide any principled method for setting it. In [19], the author claims that overfitting occurs due to testing too many hypotheses, which in our context means testing too many edge additions, and proposes an algorithm to solve it. The algorithm discards the best scoring hypotheses due to the risk of overfitting and returns the next best one.

In this paper, we propose solving the overfitting problem of CV in a new and principled way. We modify the greedy hill-climbing search so as to add an edge only if it significantly improves the CV value of the model. In order to decide upon the significance of the improvement in CV for an edge addition, we propose carrying out a hypothesis test with the improvement in CV as the test statistic and under the null hypothesis that the improvement is just by chance due to the noisy and/or finite nature of the learning data and, thus, the edge should not be added to the model. As we do not know an analytical expression of the probability distribution for the test statistic under the null hypothesis, we empirically estimate it. To be exact, we need to estimate one such probability distribution for each edge that can be added to the model. To keep it simple, we estimate the probability distributions only at the beginning of the greedy hill-climbing search, i.e. when the model is the empty graph. This means that we disregard the complexity of the model in the hypothesis tests. Furthermore, if all the nodes in the learning data have the same cardinality, then we only need to estimate a single probability distribution at the beginning of the search. This is the case in our experiments. Specifically, we empirically estimate the probability distribution for the test statistic under the null hypothesis from the improvement in CV scored by 10000 false positive edge additions to the empty graph that we obtain as follows. First, we replace the values in the learning data with uniformly drawn values and, then, compute the improvement in CV for every possible edge addition to the empty graph. Note that all these edges are false positive. We repeat this process until we gather the improvement in CV for 10000 false positive edges. We obtain the threshold for rejecting the null hypothesis by first sorting the 10000 CV values in descending order and, then, picking the $(100 \cdot \alpha)$-th percentile where $\alpha$ is the significance level. We use $\alpha = 0.001$. As the results below show, this somewhat crude solution works satisfactorily. It is out of the scope of this paper to compare different solutions to the overfitting problem of CV.

The version of BD that we use in the experiments is the so-called BDeu with an equivalent sample size (ESS) of 1 [7, 8]. This is a popular choice based on the results in [8]. For the sake of completeness, we also investigate the effects of increasing ESS. In the experiments, we always compute $\hat{\theta}$ and $\hat{\theta}^k$ as the MAP parameters obtained from $D$ and $D \setminus D^k$, respectively. The prior probability distribution over the parameters is always the same as in BD. See [7, 8] for details.

## 2.2 Experiments with Random DBNs

The first set of experiments involves learning databases of different sizes sampled from random DBNs of different complexities. We consider four model complexities: 20 three-valued nodes with 30 and 50 edges, and 40 three-valued nodes with 60 and 100 edges. We consider learning databases that consist of $S$ independent and identically distributed time series, $S = 3, 5, 10, 25, 50, 100, 250, 500, 1000$. Each time series is of length 10, i.e. it specifies values for $X^0, \ldots, X^9$. Therefore, we consider learning databases of size $10 \cdot S$ observations for each node. For each combination of model complexity and sample size, we generate 100 random DBNs. The model of each of these DBNs is obtained by adding edges to the empty graph such that each edge links a uniformly drawn pair of nodes. To keep it simple, all the edges go from $X^0$ to $X^1$. All the parameters are drawn uniformly from [0, 1]. From each of these DBNs, we sample a learning database of the corresponding size and a testing database with 1000 time series of length 10. For each learning database $D$, we proceed as follows. We run the greedy hill-climbing search described above with CV, BD and BIC as the scoring criterion. We assess the topological accuracy of each model learnt $G$ by computing its precision and recall. Precision is the number of true positive edges divided by the number of true and false positive edges, and it represents the purity of $G$. Recall is the number of true positive edges divided by the number of true positive and false negative edges, and it represents the completeness of $G$. We assess the generalization ability of $G$ as $\log p(D'|G, \hat{\theta})$, where $D'$ is the testing database paired with $D$. This quantity divided by the number of time series in $D'$, also known as log-loss, is commonly used as an approximation of $E[\log p(D_{S+1}|G, \hat{\theta})]$. For each combination of model complexity and sample size, we report the following performance measures. We report the average difference in generalization ability between the 100 models learnt via CV and the 100 induced via BD (BIC). We denote these values by **CV−BD** and **CV−BIC**. Positive values indicate that CV is superior. We also report the average precision and recall of the models learnt via CV, BD and BIC. We denote the precision values by **p CV**, **p BD** and **p BIC**, and the recall values by **r CV**, **r BD** and **r BIC**. Finally, we also report whether the differences in the results of CV and BD (BIC) are statistically significant or

| Size | CV−BD | CV−BIC | p CV | p BD | p BIC | r CV | r BD | r BIC |
|---|---|---|---|---|---|---|---|---|
| 30 | 13243±5579 ✓ | 8326±4149 ✓ | 0.40±0.43 | 0.24±0.12 ✓ | 0.37±0.17 | 0.02±0.03 | 0.10±0.05 ✓ | 0.09±0.05 ✓ |
| 50 | 2536±2291 ✓ | 2587±2154 ✓ | 0.63±0.40 | 0.58±0.25 | 0.60±0.21 | 0.05±0.04 | 0.10±0.05 ✓ | 0.12±0.06 ✓ |
| 100 | -109±705 | -254±758 | 0.94±0.11 | 0.91±0.20 | 0.93±0.12 | 0.15±0.07 | 0.14±0.07 ✓ | 0.18±0.08 ✓ |
| 250 | 575±584 ✓ | 225±456 ✓ | 0.98±0.04 | 0.99±0.03 | 1.00±0.01 ✓ | 0.41±0.10 | 0.27±0.07 ✓ | 0.34±0.08 ✓ |
| 500 | 861±462 ✓ | 410±283 ✓ | 0.99±0.03 | 1.00±0.01 ✓ | 1.00±0.02 ✓ | 0.62±0.11 | 0.41±0.09 ✓ | 0.49±0.10 ✓ |
| 1000 | 944±500 ✓ | 625±395 ✓ | 0.99±0.02 | 1.00±0.00 ✓ | 1.00±0.01 ✓ | 0.85±0.10 | 0.62±0.11 ✓ | 0.68±0.10 ✓ |
| 2500 | 421±290 ✓ | 264±241 ✓ | 0.99±0.02 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.97±0.06 | 0.84±0.10 ✓ | 0.87±0.10 ✓ |
| 5000 | 345±288 ✓ | 155±191 ✓ | 1.00±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.98±0.06 | 0.90±0.10 ✓ | 0.93±0.10 ✓ |
| 10000 | 99±192 ✓ | 84±171 ✓ | 1.00±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 1.00±0.01 | 0.97±0.06 ✓ | 0.97±0.05 ✓ |

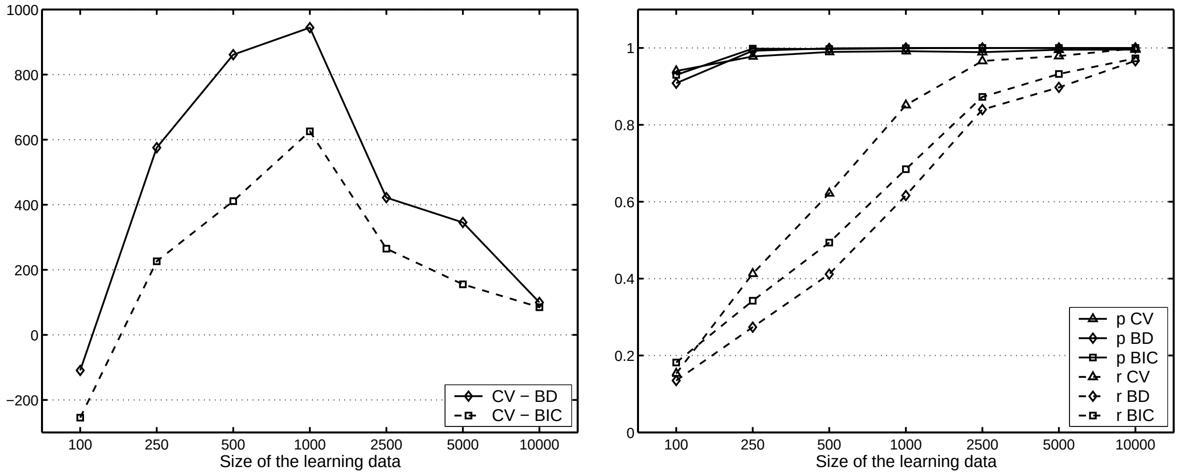

Figure 1: Results of the experiments with the random DBNs of 20 three-valued nodes and 30 edges.

| Size | CV−BD | CV−BIC | p CV | p BD | p BIC | r CV | r BD | r BIC |
|---|---|---|---|---|---|---|---|---|
| 30 | 13655±5162 ✓ | 10695±3919 ✓ | 0.37±0.42 | 0.28±0.14 | 0.34±0.18 | 0.01±0.01 | 0.06±0.04 ✓ | 0.05±0.03 ✓ |
| 50 | 1274±1973 ✓ | 2175±1958 ✓ | 0.55±0.43 | 0.64±0.32 | 0.63±0.24 | 0.02±0.02 | 0.04±0.03 ✓ | 0.06±0.03 ✓ |
| 100 | -244±613 ✓ | -183±588 | 0.88±0.21 | 0.91±0.25 | 0.91±0.16 | 0.07±0.04 | 0.06±0.03 ✓ | 0.09±0.04 ✓ |
| 250 | 565±464 ✓ | 267±358 ✓ | 0.97±0.05 | 1.00±0.01 ✓ | 0.99±0.03 ✓ | 0.21±0.06 | 0.12±0.04 ✓ | 0.16±0.04 ✓ |
| 500 | 1065±525 ✓ | 563±382 ✓ | 0.99±0.02 | 1.00±0.03 | 1.00±0.01 ✓ | 0.38±0.08 | 0.21±0.05 ✓ | 0.27±0.06 ✓ |
| 1000 | 1774±706 ✓ | 1323±601 ✓ | 1.00±0.01 | 1.00±0.00 ✓ | 1.00±0.01 | 0.60±0.10 | 0.34±0.07 ✓ | 0.40±0.08 ✓ |
| 2500 | 1396±528 ✓ | 1014±482 ✓ | 0.99±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.83±0.10 | 0.55±0.08 ✓ | 0.60±0.10 ✓ |
| 5000 | 1385±613 ✓ | 659±421 ✓ | 1.00±0.01 | 1.00±0.00 | 1.00±0.00 | 0.90±0.08 | 0.68±0.10 ✓ | 0.76±0.11 ✓ |
| 10000 | 634±438 ✓ | 546±383 ✓ | 1.00±0.00 | 1.00±0.00 | 1.00±0.00 | 0.97±0.05 | 0.82±0.09 ✓ | 0.84±0.09 ✓ |

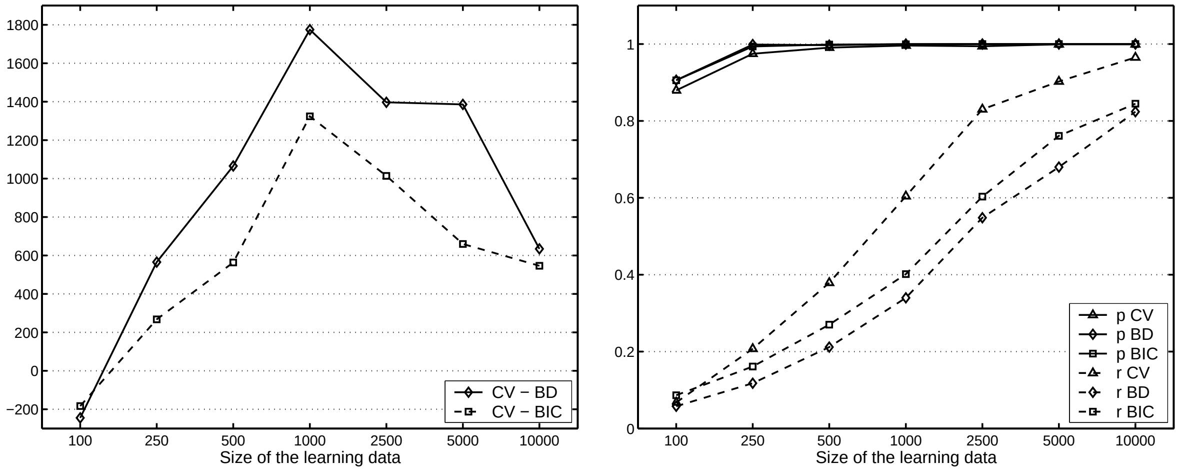

Figure 2: Results of the experiments with the random DBNs of 20 three-valued nodes and 50 edges.

| Size | CV−BD | CV−BIC | p CV | p BD | p BIC | r CV | r BD | r BIC |
|---|---|---|---|---|---|---|---|---|
| 30 | 56389±13130 ✓ | 28254±6636 ✓ | 0.35±0.27 | 0.12±0.05 ✓ | 0.23±0.10 ✓ | 0.02±0.02 | 0.11±0.05 ✓ | 0.08±0.04 ✓ |
| 50 | 9983±4523 ✓ | 8438±3141 ✓ | 0.64±0.30 | 0.40±0.14 ✓ | 0.46±0.13 ✓ | 0.04±0.03 | 0.09±0.04 ✓ | 0.11±0.04 ✓ |
| 100 | -322±1114 | -96±1310 | 0.91±0.09 | 0.90±0.10 | 0.86±0.09 ✓ | 0.14±0.04 | 0.15±0.04 ✓ | 0.19±0.05 ✓ |
| 250 | 1048±671 ✓ | 376±586 ✓ | 0.96±0.04 | 0.99±0.02 ✓ | 0.99±0.02 ✓ | 0.41±0.07 | 0.27±0.05 ✓ | 0.34±0.06 ✓ |
| 500 | 1639±580 ✓ | 875±414 ✓ | 0.98±0.02 | 1.00±0.01 ✓ | 1.00±0.01 ✓ | 0.64±0.07 | 0.42±0.07 ✓ | 0.49±0.07 ✓ |
| 1000 | 1913±666 ✓ | 1332±589 ✓ | 0.99±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.83±0.09 | 0.59±0.07 ✓ | 0.66±0.07 ✓ |
| 2500 | 1039±522 ✓ | 591±392 ✓ | 0.99±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.95±0.06 | 0.80±0.08 ✓ | 0.84±0.08 ✓ |
| 5000 | 772±488 ✓ | 368±295 ✓ | 0.99±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.98±0.04 | 0.89±0.06 ✓ | 0.92±0.05 ✓ |
| 10000 | 211±257 ✓ | 176±227 ✓ | 0.99±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.99±0.02 | 0.95±0.05 ✓ | 0.96±0.05 ✓ |

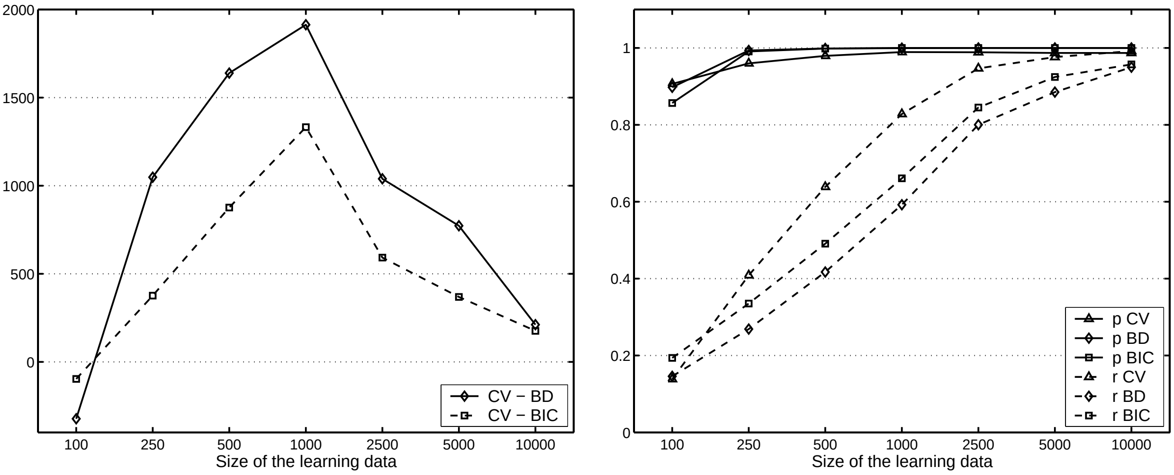

Figure 3: Results of the experiments with the random DBNs of 40 three-valued nodes and 60 edges.

| Size | CV−BD | CV−BIC | p CV | p BD | p BIC | r CV | r BD | r BIC |
|---|---|---|---|---|---|---|---|---|
| 30 | 52350±10899 ✓ | 33778±7335 ✓ | 0.23±0.29 | 0.13±0.04 | 0.20±0.08 | 0.01±0.01 | 0.06±0.02 ✓ | 0.04±0.02 ✓ |
| 50 | 7462±3971 ✓ | 10571±4040 ✓ | 0.64±0.33 | 0.44±0.15 ✓ | 0.40±0.14 ✓ | 0.02±0.01 | 0.04±0.02 ✓ | 0.06±0.02 ✓ |
| 100 | -223±764 | 414±1037 ✓ | 0.86±0.16 | 0.89±0.13 | 0.78±0.13 ✓ | 0.05±0.02 | 0.05±0.02 | 0.08±0.02 ✓ |
| 250 | 869±756 ✓ | 342±577 ✓ | 0.95±0.04 | 0.99±0.03 ✓ | 0.98±0.04 ✓ | 0.21±0.04 | 0.12±0.03 ✓ | 0.16±0.03 ✓ |
| 500 | 1989±748 ✓ | 1011±549 ✓ | 0.98±0.02 | 1.00±0.00 ✓ | 1.00±0.01 ✓ | 0.38±0.06 | 0.21±0.04 ✓ | 0.27±0.05 ✓ |
| 1000 | 3416±883 ✓ | 2499±759 ✓ | 0.99±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.59±0.06 | 0.33±0.05 ✓ | 0.40±0.05 ✓ |
| 2500 | 2729±827 ✓ | 1902±689 ✓ | 1.00±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.79±0.07 | 0.54±0.06 ✓ | 0.60±0.06 ✓ |
| 5000 | 2741±946 ✓ | 1277±533 ✓ | 1.00±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.88±0.06 | 0.66±0.06 ✓ | 0.74±0.07 ✓ |
| 10000 | 1347±659 ✓ | 1130±589 ✓ | 1.00±0.01 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.94±0.05 | 0.79±0.07 ✓ | 0.81±0.07 ✓ |

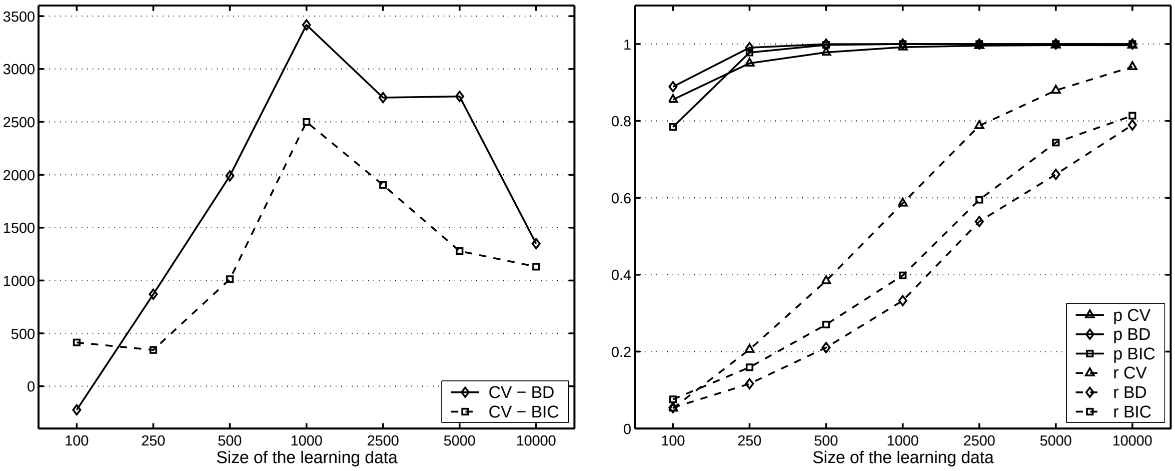

Figure 4: Results of the experiments with the random DBNs of 40 three-valued nodes and 100 edges.

not. For this purpose, we use the Wilcoxon test at a significance level of 0.001. We denote statistical significance by the symbol $\surd$.

Figures 1-4 present the results of the experiments for the four model complexities considered. The table in each figure shows average and standard deviation values, while the graphs only show average values to help visualization. We exclude from the graphs the sample sizes 30 and 50 because some of their average values are too large to be plotted without making the rest unreadable. Clearly, the models learnt via CV generalize better than those induced via BD and BIC. Specifically, CV significantly outperforms BD (BIC) in 32 (33) of the 36 combinations of model complexity and sample size in the evaluation, while BD (BIC) significantly outperforms CV in only one (zero). The precision and recall values clearly indicate that CV, BD and BIC lead to very different models. For the sample sizes smaller than 100, the models learnt via CV score similar low precision but lower recall than those induced by BD and BIC. This means that all the models learnt contain a considerable number of false positive edges, particularly those obtained by BD and BIC. These edges harm generalization, particularly for the models obtained by BD and BIC because they contain more false positive edges than those induced via CV. It is certain that the precision and recall values reported for the sample sizes smaller than 100 also mean that the models learnt by BD and BIC contain more true positive edges than those obtained via CV. However, these edges confer a limited advantage regarding generalization because the corresponding parameters cannot be estimated accurately from such small sample sizes. For the sample sizes larger than 100, the models learnt via CV score similar high precision but higher recall than those induced by BD and BIC. This means that the models learnt hardly contain false positive edges, and that the models obtained by CV contain more true positive edges than those obtained via BD and BIC. This confers advantage regarding generalization to the models learnt via CV. All these observations together lead us to conclude that the models selected by CV are very different from those selected via BD and BIC, though all of them must be supported by the learning databases, otherwise they would not have been selected. Thus, the reason why BD and BIC do not lead to the same models as CV, though they are supported by the learning databases and generalize better, is because there is a mismatch between learning and testing in the case of BD and BIC, i.e. the scoring criterion in the learning phase is not fully in line with the scoring criterion in the testing phase. This confirms our hypothesis in Section 1, namely that CV complies better than BD and BIC with our interpretation of generalization.

Some other conclusions that we obtain from Figures 1-4 follow. Increasing the ratio of the number of edges to the number of nodes in the model sampled affects the results of CV, BD and BIC more noticeably than increasing the number of nodes while keeping the ratio constant. This is not surprising because the higher the ratio, the more complex the model sampled is. We note that the performance of CV degrades less than that of BD and BIC when the ratio is increased (compare Figure 1 with Figure 2, and Figure 3 with Figure 4). It is also worth mentioning that our results for BD and BIC agree with those in [16]: Both scoring criteria produce a considerable number of false positive edges for sample sizes smaller than 100, while they produce a considerable number of false negative edges for sample sizes larger than 100. Finally, we note that BD and BIC should always lead to the true models in our experiments in the large sample limit [3]. Therefore, CV cannot beat BD and BIC in the large sample limit. Our experiments show that this theoretical result can be of limited importance in practice: CV outperforms BD and BIC for a wide range of sample sizes. Reducing the amount of learning data required to converge to the true model is very important if gathering new data is expensive.

## 2.3 Experiments with Yeast DBNs

We complement the previous section with some experiments that involve a real-world DBN model. A brief introduction to this model follows.

Much of a cell's complex behavior can be explained through the concerted activity of genes and gene products. This concerted activity is typically represented as a network of interacting genes and gene

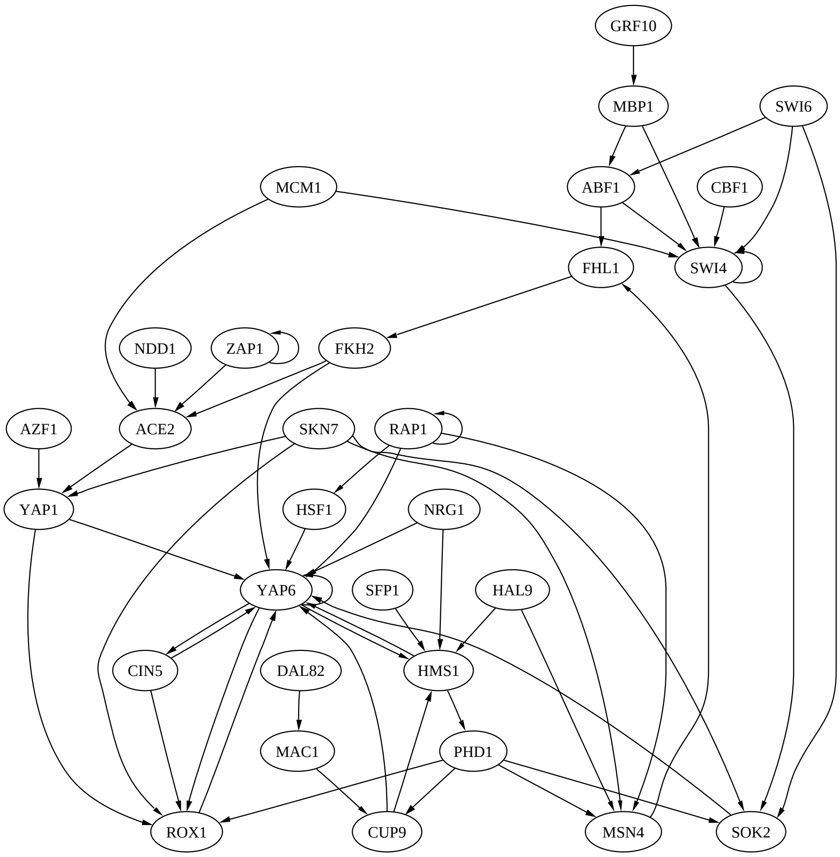

Figure 5: Partial model of the yeast transcriptional regulatory network. An edge $X_i \rightarrow X_j$ should be read as $X_i^0 \rightarrow X_j^1$ in the language of DBN models.

products that we call regulatory network. Identifying this network is crucial for understanding the behavior of the cell which, in turn, can lead to better diagnosis and treatment of diseases. This is one of the most exciting challenges in computational biology. For the last few years, there has been an increasing interest in learning DBN models of regulatory networks from data [7, 10, 13, 17, 20, 22, 27]. It is worth mentioning that there also exist other models of regulatory networks in the computational biology literature, some are more coarse than DBN models, e.g. Boolean network models, and some are less coarse, e.g. differential equation models. See [6, 26] for a review. All in all, the references above prove that DBN models can provide valuable insight into regulatory networks.

The second set of experiments in this paper involves learning databases of different sizes sampled from a partial model of the transcriptional regulatory network of *Saccharomyces cerevisiae*, i.e. baker's yeast. Yeast is typically the testing ground for new algorithms in computational biology. Specifically, we simulate the model in [11], which is based on the findings in [15]. The model involves 30 transcription factors and 56 interactions between them. See Figure 5 for a graphical representation of the model. The nodes represent the transcription factors and the edges the interactions. An edge $X_i \rightarrow X_j$ should be read as $X_i^0 \rightarrow X_j^1$ in the language of DBN models.

| Size | CV−BD | CV−BIC | p CV | p BD | p BIC | r CV | r BD | r BIC |
|---|---|---|---|---|---|---|---|---|
| 30 | 33996±9134 ✓ | 17200±5048 ✓ | 0.33±0.40 | 0.13±0.06 ✓ | 0.24±0.12 | 0.01±0.01 | 0.07±0.03 ✓ | 0.05±0.03 ✓ |
| 50 | 5932±3809 ✓ | 4711±2606 ✓ | 0.52±0.44 | 0.36±0.20 ✓ | 0.45±0.17 | 0.02±0.02 | 0.05±0.03 ✓ | 0.06±0.03 ✓ |
| 100 | -212±994 | -246±914 | 0.84±0.17 | 0.84±0.19 | 0.84±0.13 | 0.08±0.03 | 0.08±0.03 | 0.10±0.03 ✓ |
| 250 | 158±435 ✓ | -30±385 | 0.94±0.07 | 0.98±0.04 ✓ | 0.98±0.04 ✓ | 0.18±0.03 | 0.14±0.03 ✓ | 0.16±0.03 ✓ |
| 500 | 414±318 ✓ | 188±227 ✓ | 0.98±0.04 | 1.00±0.02 | 1.00±0.01 ✓ | 0.26±0.04 | 0.20±0.03 ✓ | 0.22±0.03 ✓ |
| 1000 | 433±226 ✓ | 317±233 ✓ | 0.99±0.02 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.36±0.04 | 0.26±0.03 ✓ | 0.28±0.03 ✓ |
| 2500 | 954±302 ✓ | 748±295 ✓ | 0.99±0.02 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.56±0.05 | 0.35±0.03 ✓ | 0.39±0.03 ✓ |
| 5000 | 1252±288 ✓ | 747±292 ✓ | 0.99±0.02 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.69±0.04 | 0.44±0.04 ✓ | 0.50±0.04 ✓ |
| 10000 | 911±238 ✓ | 720±220 ✓ | 0.98±0.02 | 1.00±0.00 ✓ | 1.00±0.00 ✓ | 0.77±0.02 | 0.56±0.03 ✓ | 0.60±0.03 ✓ |

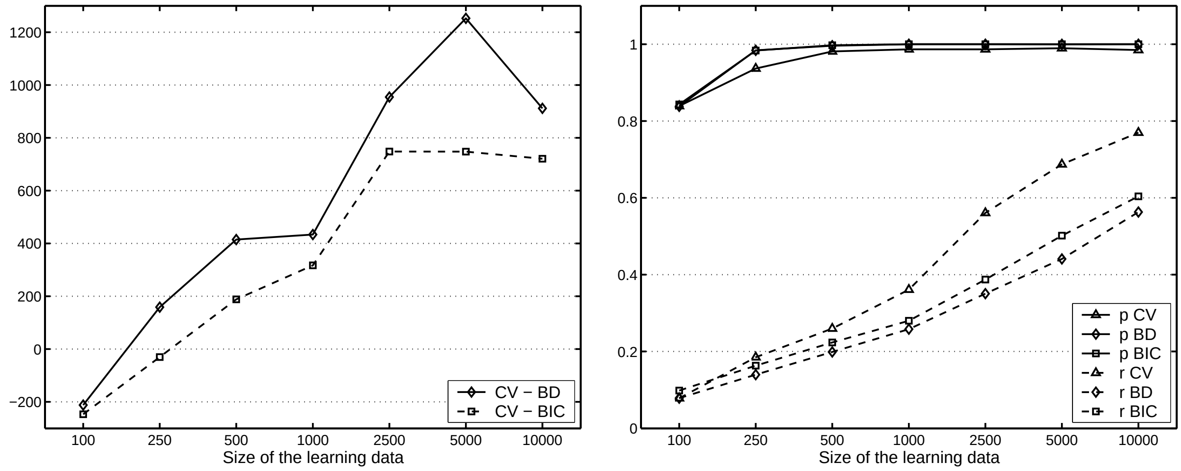

Figure 6: Results of the experiments with the yeast DBNs when ESS is 1.

| Size | CV−BD | CV−BIC | p CV | p BD | p BIC | r CV | r BD | r BIC |
|---|---|---|---|---|---|---|---|---|
| 30 | 45097±3953 ✓ | 4864±1618 ✓ | 0.42±0.41 | 0.10±0.03 ✓ | 0.23±0.12 ✓ | 0.02±0.02 | 0.17±0.05 ✓ | 0.05±0.03 ✓ |
| 50 | 28317±4435 ✓ | 1190±1134 ✓ | 0.62±0.29 | 0.16±0.04 ✓ | 0.45±0.16 ✓ | 0.04±0.02 | 0.17±0.05 ✓ | 0.07±0.03 ✓ |
| 100 | 5096±1212 ✓ | -253±602 ✓ | 0.79±0.15 | 0.35±0.08 ✓ | 0.81±0.14 | 0.09±0.03 | 0.18±0.04 ✓ | 0.10±0.03 |
| 250 | 511±335 ✓ | 182±359 ✓ | 0.92±0.07 | 0.67±0.08 ✓ | 0.98±0.04 ✓ | 0.20±0.04 | 0.24±0.04 ✓ | 0.17±0.04 ✓ |
| 500 | 125±205 ✓ | 264±289 ✓ | 0.95±0.05 | 0.86±0.07 ✓ | 1.00±0.02 ✓ | 0.29±0.03 | 0.29±0.03 | 0.23±0.03 ✓ |
| 1000 | 224±188 ✓ | 437±255 ✓ | 0.98±0.03 | 0.96±0.05 ✓ | 1.00±0.01 ✓ | 0.39±0.05 | 0.34±0.03 ✓ | 0.29±0.03 ✓ |
| 2500 | 797±227 ✓ | 1028±244 ✓ | 0.98±0.02 | 0.99±0.03 | 1.00±0.00 ✓ | 0.58±0.04 | 0.44±0.03 ✓ | 0.39±0.03 ✓ |
| 5000 | 515±233 ✓ | 939±283 ✓ | 0.98±0.02 | 0.99±0.01 ✓ | 1.00±0.00 ✓ | 0.70±0.04 | 0.55±0.03 ✓ | 0.49±0.04 ✓ |
| 10000 | 754±182 ✓ | 844±212 ✓ | 0.99±0.01 | 1.00±0.01 | 1.00±0.00 ✓ | 0.78±0.02 | 0.64±0.03 ✓ | 0.61±0.03 ✓ |

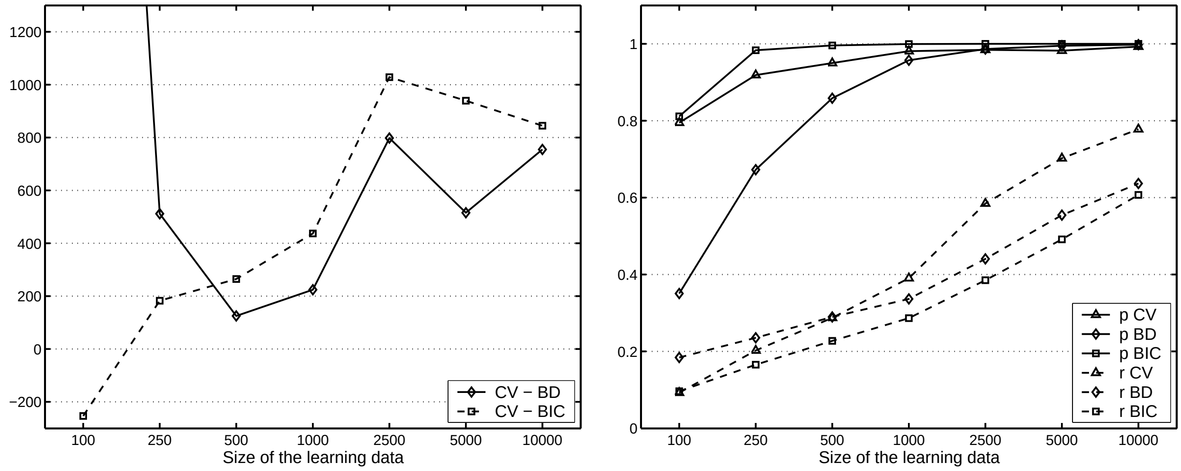

Figure 7: Results of the experiments with the yeast DBNs when ESS is 10.

| ESS | CV | BD | p CV | p BD | r CV | r BD |
|---|---|---|---|---|---|---|
| 1 | -128506±2305 | -128940±2372 ✓ | 0.99±0.02 | 1.00±0.00 ✓ | 0.36±0.04 | 0.26±0.03 ✓ |
| 10 | -128275±2493 | -128505±2459 ✓ | 0.97±0.04 | 0.95±0.05 ✓ | 0.40±0.05 | 0.34±0.03 ✓ |
| 25 | -128158±2554 | -128413±2525 ✓ | 0.97±0.04 | 0.84±0.07 ✓ | 0.42±0.05 | 0.39±0.04 ✓ |
| 50 | -127683±2514 | -127954±2437 ✓ | 0.97±0.04 | 0.73±0.07 ✓ | 0.44±0.06 | 0.45±0.04 |
| 100 | -128659±2308 | -128629±2296 | 0.99±0.03 | 0.62±0.06 ✓ | 0.39±0.07 | 0.51±0.05 ✓ |

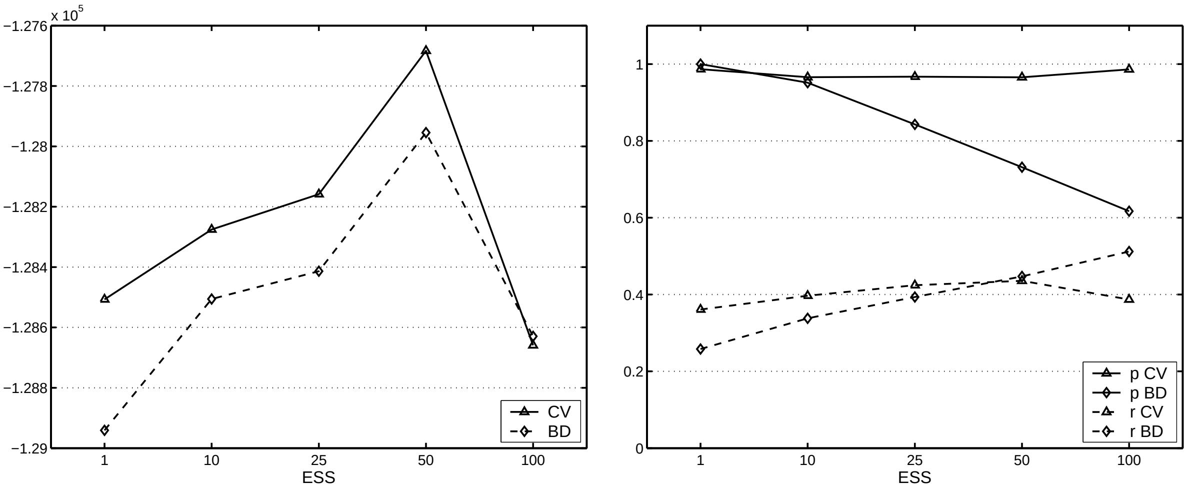

Figure 8: Results of the experiments with the yeast DBNs when increasing ESS. The size of the learning data is 1000.

At first glance, this model seems to be of intermediate complexity compared to the random DBNs in the previous section: All the edges go from $X^0$ to $X^1$ as in the random DBNs, it has 30 nodes while the random DBNs had 20 and 40, and the ratio of the number of edges to the number of nodes is 1.9 while it was 1.5 and 2.5 for the random DBNs. A closer look reveals that the complexity of this model is in the fact that some transcription factors have many regulators, i.e. parents (up to 11 for YAP6). Nodes with many parents were unlikely to occur in the random DBNs. However, this is a characteristic of many regulatory networks [15].

We consider learning databases that consist of $S$ independent and identically distributed time series, $S = 3, 5, 10, 25, 50, 100, 250, 500, 1000$. Each time series is of length 10. Therefore, we consider learning databases of size $10{\cdot}S$ measurements of the expression level of each of the 30 genes in the model. For each sample size, we generate 100 yeast DBNs. The model of each of these DBNs is the one in Figure 5. We assume that each node in the model can take three possible values, corresponding to the gene being up-regulated, down-regulated and unchanged with respect to its expression level in some control population, e.g. the previous time point or the initial time point. All the parameters are drawn uniformly from [0, 1]. We note that the true parameters are unknown in [11, 15], hence the sampling. From each of these DBNs, we sample a learning database of the corresponding size and a testing database with 1000 time series of length 10. For each learning database, we proceed as in the previous section. For each sample size, we report the same performance measures as in the previous section. We note that these experiments are not completely realistic, e.g. all the samples are free of measurement noise and some are too large given the present cost of the measurement technology. In this paper, we aim to reach some general conclusions. Thus, we disregard these domain-specific issues which may, otherwise, bias our conclusions.

Figure 6 summarizes the results of the experiments. They lead us to the same conclusions as those in the previous section, namely that the models learnt via CV generalize better than those induced via BD and BIC and that CV behaves very differently from BD and BIC. Specifically, CV significantly outperforms BD (BIC) in eight (seven) of the nine sample sizes in the evaluation, while BD and BIC never outperform CV significantly. This again confirms our hypothesis in Section 1, namely that there is a mismatch between learning and testing in the case of BD and BIC.

We now study the effects of increasing ESS. Figure 7 shows the results of the experiments when ESS takes value 10. This is another commonly used value based on the results in [8]. The most important observation that can be made from the figure is that CV outperforms both BD and BIC in this scenario as well. Specifically, CV significantly outperforms BD (BIC) in nine (eight) of the nine sample sizes in the evaluation, while BD (BIC) significantly outperforms CV in zero (one). The behavior of BIC relative to CV is consistent with that observed for ESS equal to 1. However, the behavior of BD relative to CV changes considerably from that of ESS equal to 1. The explanation is that increasing ESS reduces the model regularization implicit in BD and, thus, allows more edges to be added to the model [24] (compare the precision and recall values of BD in Figure 6 with those in Figure 7). In our experiments, this degrades generalization for the sample sizes smaller than 500 and improves it for the rest, which suggests that BD has an optimal ESS associated with each sample size. This has been previously noticed in [8, 24]. We elaborate on this issue with the help of Figure 8, which summarizes the results of the experiments for the sample size 1000 when increasing ESS. The figure reports the average generalization ability of the models induced via CV and BD instead of the average differences. We denote these values simply by **CV** and **BD**. For this sample size, increasing ESS up to 50 leads BD to models that generalize better. Therefore, the optimal ESS for BD for this sample size seems to be around 50. However, CV significantly outperforms BD for this ESS too. We believe that, even if the optimal ESS were known in advance for any sample size, BD would not beat CV because the argument of the mismatch between learning and testing still applies to BD. Moreover, our results warn that, while increasing ESS can lead BD to models that generalize better, these models can be very imprecise. Therefore, the assessment of ESS for BD remains a sensitive issue. Note, on the other hand, the robustness of CV in terms of precision and recall across all the values of ESS considered.

# 3 Discussion

BSC is probably the most commonly used scoring criterion for learning DBN models from data. Typically, BSC is regarded as scoring the likelihood of a model having generated the learning data. Alternatively, BSC can be seen as scoring the accuracy of the model as a sequential predictor of the learning data. This alternative view is interesting because it reflects that BSC scores some sort of generalization. In this paper, we are concerned with a different interpretation of generalization, namely the expected accuracy of the model for the next time series after plugging the ML or MAP parameters into the model. Therefore, BSC is not fully in line with our interpretation of generalization, though it is a popular one. This means that the common practice of learning a model via BSC, plugging the ML or MAP parameters into it and, then, using it to predict the next time series to be seen involves a mismatch between the purpose the model was learnt for and the use that is made of it. This can have negative consequences for performance. In this paper, we propose correcting this mismatch via CV. As the experimental results reported show, this is an effective way of solving the problem for a wide range of sample sizes: CV leads to models that generalize better than those induced by BSC (BD and BIC implementations). Furthermore, the models obtained by CV are topologically more accurate than those obtained by BSC for a wide range of sample sizes. Therefore, if the goal is to maximize topological accuracy rather than generalization ability, then CV may still be preferred over BSC. Finally, it is worth mentioning that we expect similar results as the ones in this paper for learning (static) Bayesian network (BN) models that generalize well under our interpretation of generalization, because our arguments for preferring CV over BSC apply to that task as well.

Our work is inspired by [4]. In that work, the authors aim to learn BN models that generalize well, where the generalization ability of a model $G$ is interpreted as the expected predictive accuracy for the next instance, i.e. $E[\log p(D_{S+1}|D,G)]$. The authors compare two scoring criteria for learning BN models that generalize well. First, BSC (BD implementation), which the authors call the scientific criterion. Second, the exact computation of $E[\log p(D_{S+1}|D,G)]$, which the authors call the engineering criterion, that is

$$\begin{aligned} & E[\log p(D_{S+1}|D,G)] \\ &= \sum_{D_{S+1}} p(D_{S+1}|D) \log p(D_{S+1}|D,G) \\ &= \sum_{D_{S+1}} \Big[ \sum_{G} p(G|D) p(D_{S+1}|D,G) \Big] \log p(D_{S+1}|D,G). \end{aligned} \tag{6}$$

The experimental results in [4] show that the engineering criterion outperforms the scientific criterion. There are two differences between our work and [4] that are worth mentioning. First, our interpretation of the generalization ability of a model $G$, i.e. $E[\log p(D_{S+1}|G,\hat{\theta})]$, differs slightly from that in [4], i.e. $E[\log p(D_{S+1}|D,G)]$, because we are interested in using the ML or MAP parameters rather than in averaging over all the parameters. Second and more important, the engineering criterion is computationally unfeasible in all but small domains, because it implies summing over all $D_{S+1}$ and all $G$ (see Equation 6). As a matter of fact, the experiments in [4] do not involve domains with more than six random variables. We recall that the exact evaluation of our interpretation of generalization is computationally unfeasible for the same reasons (see Equation 4). This is our main motivation for proposing CV as a scoring criterion for learning DBN models that generalize well: It aims to estimate the generalization ability of a model while being computationally feasible. A line of further research may be the evaluation of CV under the interpretation of generalization in [4].

There exist several papers that use CV for learning BN models for classification tasks, e.g. [12, 21]. However, to our knowledge, [25] is the only study of CV for learning BN models for general purposes. In [5], the authors mention the possibility of using CV for learning BN models for general purposes but they do not pursue it further. In [25], the authors aim to learn BN models that minimize the cross-entropy which, as discussed in Section 1, agrees with our interpretation of generalization. They experimentally compare CV, BIC and Akaike's information criterion (AIC) [1], and conclude that CV performs the best because AIC and particularly BIC overpenalize the model complexity and, thus, lead to underfitted models. We note that they neither include BD in the comparison nor carry out a search in the space of models. Instead, they generate a set of nested models from the simplest to the most complex passing through the true one and, then, compare how the different scoring criteria behave for that set of models. These issues apart, the main difference between [25] and our work is in the explanation of why CV does the best: The authors of [25] argue that BIC and AIC overpenalize the model complexity, while we argue that BD and BIC do not fully match the ultimate goal, namely generalization. Therefore, we provide an alternative explanation to that in [25]. A criticism of the explanation in [25] is that it is not valid for all sample sizes: Our results and those in [16] show that BIC underpenalizes the model complexity for samples sizes smaller than 100. This is not detected in [25] because all the databases considered are of size 200 or larger. The explanation in [25] does not seem to apply to BD either: In the small ESS limit, BD leads to the complete graph for small sample sizes [24]. We are currently studying the connection between the mismatch and the under and overpenalization. We hope that this paper contributes to a better understanding of the behavior of the different scoring criteria for learning BN and DBN models from data.

# Acknowledgements

We thank Roland Nilsson for providing us with the code for the Wilcoxon test, and Magnus Ekdahl and the three anonymous referees for their valuable comments on this paper. This work is funded by the Swedish Foundation for Strategic Research (SSF) and Linköping Institute of Technology.

# References

[1] Akaike,H. (1974) A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, **19**, 716-723.

[2] Bouckaert,R.R. (2003) Choosing Between Two Learning Algorithms Based on Calibrated Tests. In *Proceedings of the Twentieth International Conference on Machine Learning*, 51-58.

[3] Chickering,D.M. (2002) Optimal Structure Identification with Greedy Search. *Journal of Machine Learning Research*, **3**, 507-554.

[4] Chickering,D.M. and Heckerman,D. (2000) A Comparison of Scientific and Engineering Criteria for Bayesian Model Selection. *Statistics and Computing*, **10**, 55-62.

[5] Cowell,R.G., Dawid,A.P., Lauritzen,S.L. and Spiegelhalter,D.J. (1999) *Probabilistic Networks and Expert Systems*. Springer-Verlag.

[6] D'haeseleer,P., Liang,S. and Somogyi,R. (2000) Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering. *Bioinformatics*, **16**, 707-726.

[7] Friedman,N., Murphy,K. and Russell,S. (1998) Learning the Structure of Dynamic Probabilistic Networks. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 139-147.

[8] Heckerman,D., Geiger,D. and Chickering,D.M. (1995) Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, **20**, 197-243.

[9] Hofmann,R. and Tresp,V. (1998) Nonlinear Markov Networks for Continuous Variables. In *Advances in Neural Information Processing Systems*, **10**, 521-527.

[10] Husmeier,D. (2003) Sensitivity and Specificity of Inferring Genetic Regulatory Interactions from Microarray Experiments with Dynamic Bayesian Networks. *Bioinformatics*, **19**, 2271-2282.

[11] Kauffman,S., Peterson,C., Samuelsson,B. and Troein,C. (2003) Random Boolean Network Models and the Yeast Transcriptional Network. In *Proceedings of the National Academy of Sciences of the USA*, **100**, 14796-14799.

[12] Keogh,E. and Pazzani,M. (2002) Learning the Structure of Augmented Bayesian Classifiers. *International Journal on Artificial Intelligence Tools*, **11**, 587-601.

[13] Kim,S., Imoto,S. and Miyano,S. (2003) Dynamic Bayesian Network and Nonparametric Regression for Nonlinear Modeling of Gene Networks from Time Series Gene Expression Data. In *Proceedings of the First International Workshop on Computational Methods in Systems Biology*, 104-113.

[14] Kohavi,R. (1995) A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1137-1143.

[15] Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.T., Thompson,C.M., Simon,I., Zeitlinger,J., Jennings,E.G., Murray,H.L., Gordon,D.B., Ren,B., Wyrick,J.J., Tagne,J.B., Volkert,T.L., Fraenkel,E., Gifford,D.K. and Young,R.A. (2002) Transcriptional Regulatory Networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799-804.

[16] Moral,S. (2004) An Empirical Comparison of Score Measures for Independence. In *Proceedings of the Tenth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1307-1314.

[17] Murphy,K. and Mian,S. (1999) Modelling Gene Expression Data Using Dynamic Bayesian Networks. Technical Report, Computer Science Division, University of California, Berkeley.

[18] Neapolitan,R.E. (2003) *Learning Bayesian Networks*. Prentice Hall.

[19] Ng,A. (1997) Preventing “Overfitting” of Cross-Validation Data. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 245-253.

[20] Ong,I.M., Glasner,J.D. and Page,D. (2002) Modelling Regulatory Pathways in E. Coli from Time Series Expression Profiles. *Bioinformatics*, **18**, S241-S248.

[21] Pazzani,M. (1995) Searching for Dependencies in Bayesian Classifiers. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, 239-248.

[22] Perrin,B.E., Ralaivola,L., Mazurie,A., Bottani,S., Mallet,J. and d’Alché-Buc,F. (2003) Gene Networks Inference Using Dynamic Bayesian Networks. *Bioinformatics*, **19**, ii138-ii148.

[23] Schwarz,G. (1978) Estimating the Dimension of a Model. *Annals of Statistics*, **6**, 461-464.

[24] Steck,H. and Jaakkola,T. (2003) On the Dirichlet Prior and Bayesian Regularization. In *Advances in Neural Information Processing Systems 15*, 697-704.

[25] Van Allen,T. and Greiner,R. (2000) Model Selection Criteria for Learning Belief Nets: An Empirical Comparison. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 1047-1054.

[26] Wessels,L.F.A., Van Someren,E.P. and Reinders,M.J.T. (2001) A Comparison of Genetic Network Models. In *Proceedings of the Pacific Symposium on Biocomputing*, 508-519.

[27] Zou,M. and Conzen,S.D. (2005) A New Dynamic Bayesian Network (DBN) Approach for Identifying Gene Regulatory Networks from Time Course Microarray Data. *Bioinformatics*, **21**, 71-79.