A Notes on MDL

Given a sufficiently regular k-dimensional parametric model class $\mathcal{M}_{\mathcal{G}}$ (e.g. a curved exponential family), an essential result from the MDL literature is that $\log p_{\text{Bayes}}(\mathcal{Z} \mid \mathcal{G})$ and $\log p_{\text{plug-in}}(\mathcal{Z} \mid \mathcal{G})$ closely track the maximum log-likelihood of $\mathcal{M}_{\mathcal{G}}$ for all sufficiently regular data $\mathcal{Z} \in \mathcal{X}^n$ (e.g. the MLE lies in the interior of the parameter space); see Grünwald and Roos [2019][Section 2.2] for precise statements and pointers to the literature. Essentially, this result shows that p_{Bayes} and $p_{\text{plug-in}}$ are universal distributions and satisfy Eq. (1), up to a small error.

Theorem 1. *Under sufficient regularity conditions on the model class $\mathcal{M}_{\mathcal{G}}$ and for any sufficiently regular data $\mathcal{Z} \in \mathcal{X}^n$,*

$$\log p(\mathcal{Z} \mid \hat{\theta}^{MLE}(\mathcal{Z}), \mathcal{G}) \leq \log p_{preq}(\mathcal{Z} \mid \mathcal{G}) + \mathcal{O}(\log(n)) + \mathcal{O}(1),$$
$$\log p(\mathcal{Z} \mid \hat{\theta}^{MLE}(\mathcal{Z}), \mathcal{G}) \leq \log p_{Bayes}(\mathcal{Z} \mid \mathcal{G}) + \frac{k}{2}\log(n) + \mathcal{O}(1),$$

where the constants depend on the mismatch between $\mathcal{M}_{\mathcal{G}}$ and the data distribution.

Connection to Data Compression. The use of universal distributions connects to data compression through the Kraft–McMillan inequality [Cover, 1999], which states that a probabilistic model's log-loss on $\mathcal{D}$ is equal to the shortest achievable code-length for encoding $\mathcal{D}$ using a code derived from that model. However, we cannot use the optimal choice $\hat{\theta}^{\text{MLE}}(\mathcal{D})$ for compression since it is not known a priori. Instead, we want a code that can losslessly encode almost as well as $p(\cdot|\hat{\theta}^{\text{MLE}}, \mathcal{G})$ by having a code length near $\min_\theta - \log\, p(\mathcal{Z}|\theta, \mathcal{G})$ for all possible data $\mathcal{Z}$. A reader may recognize this criterion as Eq. (1), so we can alternatively define a universal distribution as one that achieves a code length as closely to that of $p(\cdot|\hat{\theta}^{\text{MLE}}, \mathcal{G})$ as possible for all data $\mathcal{Z}$.

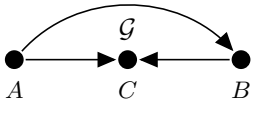
B Faithfulness Example

Let us denote with $\mathcal{I}(\mathcal{G}) = \{X \perp\!\!\!\perp_{\mathcal{G}} Y \mid Z\}$ the set of statistical independence relationships implied by the structure of a BN $(\mathcal{G}, p)$, also called *global Markov independencies*, and let us denote with $\mathcal{I}(p)$ the set of statistical independence relationships satisfied by the distribution p. We must have $\mathcal{I}(\mathcal{G}) \subseteq \mathcal{I}(p)$, i.e. any independence encoded in $\mathcal{G}$ is satisfied by p—we denote this as $X \perp\!\!\!\perp_{\mathcal{G}} Y \mid Z \Rightarrow X \perp\!\!\!\perp_p Y|Z$. If the converse relation holds, namely if any independence satisfied by p is encoded in $\mathcal{G}$ (i.e. $X \perp\!\!\!\perp_p Y \mid Z \Rightarrow X \perp\!\!\!\perp_{\mathcal{G}} Y \mid Z$) giving $\mathcal{I}(\mathcal{G}) = \mathcal{I}(p)$, then we say that p is *faithful* to $\mathcal{G}$. Unfaithfulness can arise for different reasons, for example when dependence along different pathways cancels out [Peters et al., 2017] or when links express deterministic relationships, which can occur in many real-world scenarios (see Koski and Noble [2012], Mabrouk et al. [2014]).

Constraint-based approaches to structure learning assume faithfulness of the distribution p to the structure $\mathcal{G}$ underlying the data generation mechanism, and infer a member of the Markov equivalence class w.r.t. which p is faithful. In the example below we shows that, when p is not faithful, recovering the Markov equivalence class w.r.t. which p is faithful might lead to a preference for more complex CPDs. This issue is not shared by prequential scoring which does not assume faithfulness and infers a structure by taking the complexity of the CPDs into account.

Let us consider the data generation mechanism

- $A =$ random prime number with $2 \leq A \leq M$,
- $B =$ random prime number with $A \leq B \leq M$,
- $C = A \cdot B$.



The DAG $\mathcal{G}$ corresponding to this mechanism is shown in the figure above and has $\mathcal{I}(\mathcal{G}) = \emptyset$.

If we analyze the conditional independence relationships satisfied by the joint distribution p, we realize that any number c with $p(C = c) > 0$ can be uniquely factorized into the numbers A and B that generated it. This is a direct consequence of the fundamental theorem of arithmetic and the fact that the generative process ensures $A \leq B$. Because A does not provide additional information about B given C and vice versa, the set of statistical independence relationships between the variables is given by $\mathcal{I}(p) = \{A \perp\!\!\!\perp B \mid C\}$. Therefore p is *not* faithful to $\mathcal{G}$.

The Markov equivalence class $\mathcal{E}$ of DAGs w.r.t p is faithful is $\mathcal{E} = \{A \leftarrow C \rightarrow B,\ A \rightarrow C \rightarrow B,\ A \leftarrow C \leftarrow B\}$. The structures in $\mathcal{E}$ can be considered simpler than $\mathcal{G}$ as they contain fewer links. However, their joint distributions must contain at least one conditional distribution which requires an integer factorization. For example, $p(A \mid C)$ reads: "perform a factorization of C into prime numbers and set A to the smaller one". Factorization is considered a hard problem and it is commonly used as a one-way function in cryptography because no known algorithms can perform it efficiently.

Applying structure learning methods that infer the Markov equivalence class w.r.t which the distribution is faithful would return $\mathcal{E}$ and, along the way, the methods would have to discover (the existence of) a factorization algorithm (or table) for the natural numbers up to M^2. Therefore, whilst returning simpler structures, these methods would imply significantly more complex conditional distributions (for some intuitive definition of complexity). Prequential scoring instead takes the complexity of the conditional distributions into account, and therefore it is free to prefer a graph which is outside of $\mathcal{E}$ if it finds modeling a multiplication "easier" than factorization.

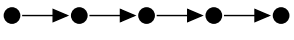
C Case Study on 5 Nodes Chains

In the sections above we argued that prequential scoring not only takes the overall DAG sparsity into account, but also considers the complexity of the CPDs as measured by their sample-efficiency. As a result, prequential scoring often has a preference for the direction of links even when the resulting DAGs contain the same number of edges and belong to the same Markov equivalence class.

In this section, we demonstrate this property by generating data from two chains of 5 variables $X^1, \cdots, X^5$. In the first chain, each variable is an almost linear function of the previous variable in the chain plus some noise. In the second chain, the functional relationship between successive nodes is more non-linear (periodic) and requires a multi-modal CPD when modeling the relationship in the inverse direction. Note that our models are capable of modeling multi-modal regression variables because we use categorical prediction for the discretized target variable as described in Sect. 2.2.

For each of the two chains, we show the top-5 inferred DAGs and their excess scores relative to the best DAG. We observe that for both chains prequential scoring correctly identifies the ground-truth DAG as most likely. However, prequential scoring consistently avoids inverting the direction of the links in the second chain, which is not a property shared by the contenders.

$X^1 \sim \mathcal{N}(0, 1)\ ;\ X^d \sim \mathcal{N}(\sin(X^{d-1}), 0.1^2)$

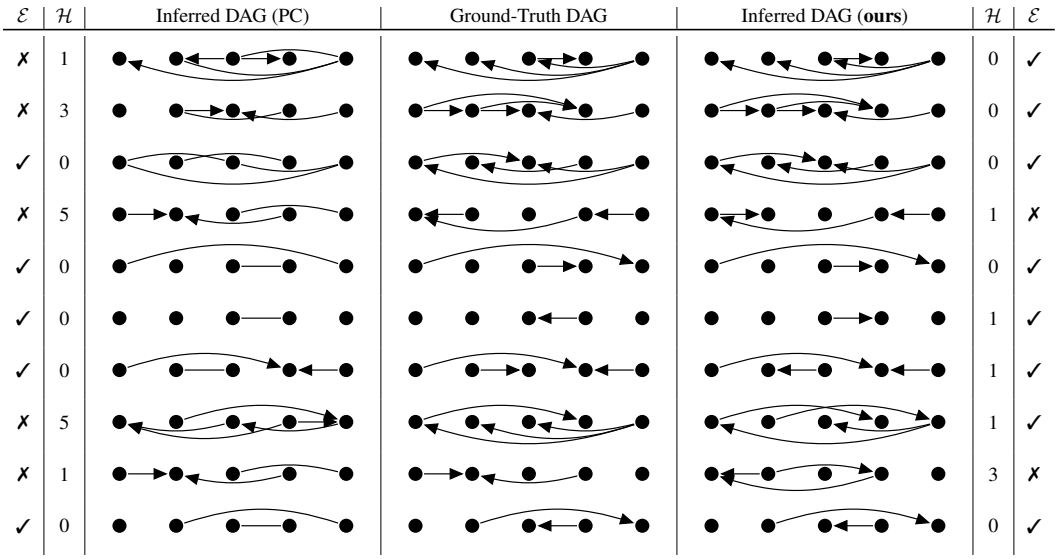
Inferred DAG	$\Delta \log p(\mathcal{D}\|\mathcal{G})$
	0
	1117
	1178
	1319
	1348

$X^1 \sim \mathcal{N}(0, 1)\ ;\ X^d \sim \mathcal{N}(\sin(4 \cdot X^{d-1}), 0.1^2)$

Inferred DAG	$\Delta \log p(\mathcal{D}\|\mathcal{G})$
	0
	2361
	2455
	4569
	4678

D PC Algorithm on Data from Yu et al. [2019]

This section describes the behaviour of the PC algorithm on data generated according to the mechanism introduced by Yu et al. [2019] for validating DAG-GNN described in the main text. We used the implementation in PGMPy [Ankan and Panda, 2015] with χ^2 (`k2`) and linear-regression based Pearson correlation (`pearsonr`) as conditional independence test. The PC algorithm returns a PDAG, which uses undirected edges to represent the Markov equivalence class, so the Hamming distance to ground-truth is not comparable to the one of prequential scoring.



E 5-Node DAGs

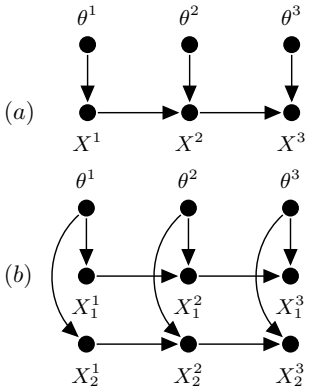
We applied prequential scoring to 20 randomly generated DAGs with nonlinear relationships between variables. Each instance was created by 1) generating a random 5-node DAG with the GNP algorithm [Batagelj and Brandes, 2005] with link probability of 0.25, 2) removing links to make the adjacency matrix lower-triangular, 3) selecting the non-linear functions for each (topologically sorted) variable uniformly from a set of candidate-functions with a the corresponding number of input variables (see Table 3), 4) generating i.i.d. data (sampling a independent $\epsilon \sim \mathcal{N}(0, 0.1^2)$ per variable),

Table 3 shows the 20 generative processes that were created this way. The results when applying prequential scoring to this data can be found in Table 2. DAG-GNN [Yu et al., 2019] with the recommended parameter settings was unstable with respect to random seed and dataset size. We were not able to find a parameter setting that worked reliably, so we omit the results for DAG-GNN on this data.

A	B	C	D	E
$\sin(30\epsilon)$	$\sin(2A) + \epsilon$	$\sin(B^3 - B + \epsilon)$	$(C + \epsilon)^3$	$sgn(A)^1/_{\lvert A\rvert + 0.1} + \epsilon$
10ϵ	$(A + \epsilon)^3$	$\sin(2A) + \epsilon$	$\sin(^1/_{\lvert C\rvert + 0.1} + \epsilon)$	$\sin(2A) + \epsilon$
10ϵ	10ϵ	$sgn(B)^1/_{\lvert B\rvert + 0.1} + \epsilon$	$\sin(2C^3 - B^2) + \epsilon$	$sgn(D)\sin(4DA + \epsilon)$
10ϵ	$sgn(A)^1/_{\lvert A\rvert + 0.1} + \epsilon$	10ϵ	$\sin(2C^3 - A^2) + \epsilon$	$sgn(D)\sin(4DA + \epsilon)$
$\sin(30\epsilon)$	$(A + \epsilon)^3$	$\sin(30\epsilon)$	$(C + \epsilon)^3$	$\sin(C^3 - C + \epsilon)$
10ϵ	$\sin(30\epsilon)$	$\sin(2B^3 - A^2) + \epsilon$	$sgn(C)\sin(4CA + \epsilon)$	$\sin(^1/_{\lvert D\rvert + 0.1} + \epsilon)$
$\sin(30\epsilon)$	$\sin(A^3 - A + \epsilon)$	$\sin(2B)\sin(^1/_{\lvert A\rvert + 0.1}) + \epsilon$	$\sin(4CBA + \epsilon)$	$\sin(2D^3 - C^2) + \epsilon$
$\sin(30\epsilon)$	$\sin(A^3 - A + \epsilon)$	$\sin(2B)\sin(^1/_{\lvert A\rvert + 0.1}) + \epsilon$	$sgn(A)^1/_{\lvert A\rvert + 0.1} + \epsilon$	$\sin(2D)\sin(^1/_{\lvert A\rvert + 0.1}) + \epsilon$
10ϵ	10ϵ	$\sin(4BA + \epsilon)$	$\sin(30\epsilon)$	$\sin(4DCA + \epsilon)$
$\sin(30\epsilon)$	$\sin(A^3 - A + \epsilon)$	$\sin(2B^3 - A^2) + \epsilon$	10ϵ	$\sin(2B)\sin(4BA + \epsilon)$
$\sin(30\epsilon)$	$\sin(30\epsilon)$	$sgn(B)\sin(2BA + \epsilon)$	$\sin(30\epsilon)$	$\sin(2D)\sin(4DA + \epsilon)$
$\sin(30\epsilon)$	$\sin(2A) + \epsilon$	$\sin(4BA + \epsilon)$	$\sin(2C)\sin(^1/_{\lvert B\rvert + 0.1}) + \epsilon$	$sgn(C)^1/_{\lvert C\rvert + 0.1} + \epsilon$
10ϵ	$(A + \epsilon)^3$	$\sin(4BA + \epsilon)$	$sgn(C)\sin(2CA + \epsilon)$	$\sin(4DA + \epsilon)$
$\sin(30\epsilon)$	$(A + \epsilon)^3$	10ϵ	$\sin(4CA + \epsilon)$	$sgn(D)^1/_{\lvert D\rvert + 0.1} + \epsilon$
10ϵ	$\sin(A^3 - A + \epsilon)$	$(B + \epsilon)^3$	$\sin(C^3 - C + \epsilon)$	$\sin(2A) + \epsilon$
$\sin(30\epsilon)$	$sgn(A)^1/_{\lvert A\rvert + 0.1} + \epsilon$	$sgn(B)^1/_{\lvert B\rvert + 0.1} + \epsilon$	$\sin(2C) + \epsilon$	$sgn(D)\sin(4DA + \epsilon)$
$\sin(30\epsilon)$	$sgn(A)^1/_{\lvert A\rvert + 0.1} + \epsilon$	$\sin(2B^3 - A^2) + \epsilon$	10ϵ	$sgn(D)^1/_{\lvert D\rvert + 0.1} + \epsilon$
10ϵ	$\sin(30\epsilon)$	$sgn(B)\sin(2BA + \epsilon)$	$\sin(2C)\sin(^1/_{\lvert B\rvert + 0.1}) + \epsilon$	$\sin(4DA + \epsilon)$
$\sin(30\epsilon)$	$\sin(^1/_{\lvert A\rvert + 0.1} + \epsilon)$	$(B + \epsilon)^3$	$\sin(2B^3 - A^2) + \epsilon$	$\sin(B^3 - B + \epsilon)$
$\sin(30\epsilon)$	$\sin(^1/_{\lvert A\rvert + 0.1} + \epsilon)$	$\sin(B^3 - B + \epsilon)$	$\sin(^1/_{\lvert B\rvert + 0.1} + \epsilon)$	$\sin(30\epsilon)$

Table 3: Generating equations used for the experiments presented in Table 2 ($\epsilon \sim \mathcal{N}(0, 0.1^2)$ for each table cell independently).

F Relation between $\log\, p_{\textbf{Bayes}}(\mathcal{D}\,|\,\mathcal{G})$ and $\log\, p_{\textbf{preq}}(\mathcal{D}\,|\,\mathcal{G})$



In this section we demonstrate that, if $p(\theta\,|\,\mathcal{G}) = \prod_{d=1}^{D} p(\theta^d)$, $\log\, p_{\text{Bayes}}(\mathcal{D}\,|\,\mathcal{G})$ can be written as (omitting dependence on $\mathcal{G}$)

$$\log\, p_{\text{Bayes}}(\mathcal{D}) = \sum_{d=1}^{D} \log \prod_{i=1}^{n} \int_{\Theta^d} p(x_i^d\,|\,\text{pa}(x_i^d), \theta^d) p(\theta^d\,|\,x_{<i})\, d\theta^d, \tag{4}$$

highlighting the connection with the factorization of $p_{\text{plug-in}}(\mathcal{D})$ over CPDs (Eq. (3))

$$\log\, p_{\text{preq}}(\mathcal{D}) = \sum_{d=1}^{D} \log \prod_{i=1}^{n} p(x_i^d\,|\,\text{pa}(x_i^d), \hat{\theta}^d(x_{<i})).$$

To simplify the understanding but without loss of generality, we show the validity of Eq. (4) for the Bayesian network (a). Let $\mathcal{D} = \{x_i := (x_i^1, x_i^2, x_i^3)\}_{i=1}^2$ be a dataset formed by two samples from $\int_\Theta p(X^1, X^2, X^3, \theta) d\theta$. We can view $\mathcal{D}$ as formed by one sample from $\int_\Theta p(X_1, X_2, \theta) d\theta$, where $p(X_1, X_2, \theta) = p(X_1\,|\,\theta) p(X_2\,|\,\theta) p(\theta)$ with $p(X_1\,|\,\theta) = p(X_2\,|\,\theta)$, and $p(X_i := (X_i^1, X_i^2, X_i^3)|\theta) = p(X_i^3\,|\,X_i^2, \theta^3) p(X_i^2\,|\,X_i^1, \theta^2) p(X_i^1\,|\,\theta^1)$, for $i = 1, 2$ (Bayesian network (b)). Using the prequential formulation $\log p(x_1, x_2) = \log\, p(x_2|x_1) p(x_1)$, $\log\, p_{\text{Bayes}}(\mathcal{D}) := \log \int_\Theta p(x_1, x_2, \theta) d\theta$ can be written as

$$\begin{aligned}
\log\, p_{\text{Bayes}}(\mathcal{D}) &= \log\, p(x_2|x_1) p(x_1) = \log \Big\{ \int_\Theta p(x_2, \theta\,|\,x_1) d\theta \Big\} \Big\{ \int_\Theta p(x_1, \theta) d\theta \Big\} \\
&= \log \Big\{ \int_\Theta p(x_2\,|\,\theta, \cancel{x_1}) p(\theta\,|\,x_1) d\theta \Big\} \Big\{ \int_\Theta p(x_1\,|\,\theta) p(\theta) d\theta \Big\} \\
&= \log \Big\{ \int_\Theta p(x_2^3\,|\,x_2^2, \theta^3) p(x_2^2\,|\,x_2^1, \theta^2) p(x_2^1\,|\,\theta^1) p(\theta^1\,|\,x_1) p(\theta^2\,|\,x_1) p(\theta^3\,|\,x_1) d\theta \Big\} \\
&\quad \times \Big\{ \int_\Theta p(x_1^3\,|\,x_1^2, \theta^3) p(x_1^2\,|\,x_1^1, \theta^2) p(x_1^1\,|\,\theta^1) p(\theta^1) p(\theta^2) p(\theta^3) d\theta \Big\} \\
&= \sum_{d=1}^{D} \log \prod_{i=1}^{n} \int_{\Theta^d} p(x_i^d\,|\,\text{pa}(x_i^d), \theta^d) p(\theta^d\,|\,x_{<i})\, d\theta^d,
\end{aligned} \tag{5}$$

where we have used the fact that $p(x_2\,|\,\theta, x_1) = p(x_2\,|\,\theta)$ and that the parameters posterior factorizes over d, i.e. $p(\theta\,|\,x_1) = p(\theta^1\,|\,x_1) p(\theta^2\,|\,x_1) p(\theta^3\,|\,x_1)$.

In the case in which the distribution of the parameters for each CPD and value of parents is Dirichlet with parameter α, a well-known result is

$$\int_{\Theta^d} p(x_i^d = k\,|\,\text{pa}(x_i^d) = l, \theta^d) p(\theta^d\,|\,x_{<i})\, d\theta^d = \frac{N_{k,l} + \alpha}{\sum_m (N_{m,l} + \alpha)}.$$

G Prequential Scoring with Interventional Data

In Fig. 4 we show the influence of interventional data on prequential scoring. We generated observations from the cancer Bayesian network from the bnlearn repository [Taskesen, 2019] with DAG $\mathcal{G}^* = P \to C \leftarrow S, C \to X, C \to D$. For $i \in \{5,000, \ldots, 6000\}$, we generated interventional data by randomly selecting a node from the graph and replacing it with a random value. The excess prequential loss in nats is plotted for the 14 best ranking DAGs during the first half of the sequence and the 14 best after learning from all observations. The interventional samples have a strong influence on the ranking and are necessary for the correct identification of $\mathcal{G}^*$.

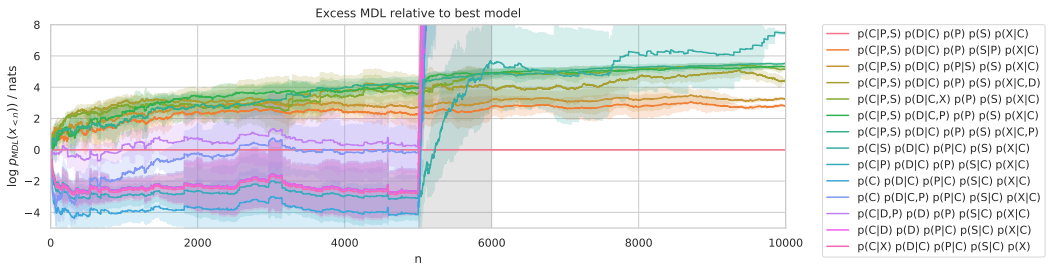


Figure 4: Excess log-loss in nats for $i = 1, \ldots, 10,000$ for the 14 DAGs with lowest excess. Uncertainty bands show standard deviation from 5 different different permutations of the data. Between $5,000 \leq i < 6,000$ (shaded area) interventional samples were supplied: we replaced the value of a random node in the graph with an random value (p = ½). Only due to these interventional samples could the ground-truth DAG $\mathcal{G}^*$ be correctly identified as most likely. Nevertheless, there is no strong evidence in favour of a structure as many of them differ by only a few nats.