Tuning Causal Discovery Algorithms

Konstantina Biza KONBIZA@GMAIL.COM
Computer Science Department, University of Crete

Ioannis Tsamardinos TSAMARD.IT@GMAIL.COM
Computer Science Department, University of Crete

Sofia Triantafillou SOT16@PITT.EDU
Department of Biomedical Informatics, University of Pittsburgh

Abstract

There are numerous algorithms proposed in the literature for learning causal graphical probabilistic models. Each one of them is typically equipped with one or more tuning hyper-parameters. The choice of optimal algorithm and hyper-parameter values is not universal; it depends on the size of the network, the density of the true causal structure, the sample size, as well as the metric of quality of learning a causal structure. Thus, the challenge to a practitioner is how to "tune" these choices, given that the true graph is unknown and the learning task is unsupervised. In the paper, we evaluate two previously proposed methods for tuning, one based on stability of the learned structure under perturbations of the input data and the other based on balancing the in-sample fitting of the model with the model complexity. We propose and comparatively evaluate a new method that treats a causal model as a set of predictive models: one for each node given its Markov Blanket. It then tunes the choices using out-of-sample protocols for supervised methods such as cross-validation. The proposed method performs on par or better than the previous methods for most metrics.

Keywords: Causal Discovery, Tuning, Bayesian Networks.

1. Introduction

A wide palette of algorithms for learning causal probabilistic graphical models from observational data has been proposed in the literature. The algorithms differ in terms of their distributional assumptions, theoretical properties, search heuristics for the optimal structure, approximations, or other characteristics that make them more or less appropriate and effective to a given learning task. In addition, most algorithms are "tunable" by employing a set of hyper-parameters that determines their behavior, such as their sensitivity to identifying correlations. The choice of algorithm and corresponding hyper-parameter values (hereafter, called a **configuration**) can have a sizable impact on the learning quality. Practitioners are faced with optimizing the configuration for the task at hand. Unfortunately, given that the problem is unsupervised, standard out-of-sample estimation methods used for supervised problems, such as cross-validation cannot be directly applied.

In this work, we propose an optimization method over a set of configurations, called Out-of-sample Causal Tuning or **OCT**. It is based on the premise that a causal network G induces a set of predictive models, namely a model for each node V, using as predictors the variables in its Markov Blanket, denoted as $MB_G(V)$. The $MB_G(V)$ is the minimal set of nodes that leads to an optimally predictive model for V (under some conditions, see (Tsamardinos and Aliferis, 2003)). Thus, a configuration that outputs a causal network with all the correct Markov Blankets will exhibit the optimal average (over all nodes) out-of-sample prediction power and be selected over all other configurations. Hence, one could optimize the configuration by employing out-of-sample estimation protocols devised for supervised learning problems, such as k-fold cross-validation, hold out, and repeated hold out. *Whether optimizing causal algorithms with respect to predictive performance will also result in optimizing learning with respect to the causal structure is the major research question of the paper.*

The proposed method is comparatively evaluated against three other methods in the literature: **StARS**, and selecting the configuration that minimizes the Bayesian Information Criterion (**BIC**) or the Akaike Information Criterion (**AIC**). StARS selects the configuration that exhibits the highest stability of causal induction to perturbations of the input data, while BIC and AIC select the configurations that fit the data best, according to some parametric assumptions, and penalizing for complexity. In a set of computational experiments over several network sizes and densities, as well as sample sizes, we show that **OCT** performs on par or better than the prior suggestions. However, we also indicate that there is still much room for improvement for novel tuning methodologies.

OCT is an approach exploiting the connection between causal models and predictive models: a causal model implies a set of predictive models. It tries to leverage results in supervised, predictive modeling to improve causal discovery, in this case the use of out-of-sample estimation protocols. Conceivably, other techniques for tuning predictive models, such as the ones in the field of automated machine learning (AutoML) for intelligently searching the space of hyper-parameters could improve causal tuning and causal learning.

2. Problem Definition

In an application of causal discovery algorithms to real data a practitioner is faced with selecting the appropriate algorithm to use. In addition, each algorithm requires the choice of the values of a certain number of hyper-parameters that determine its behavior, such as the sensitivity of identifying patterns. Hyper-parameters differ from model parameters in the sense that the former are set by the user, while the latter are estimated from the data. The impact of the choice of the hyper-parameter values for a given algorithm has been noted in several papers (Raghu et al., 2018; Ramsey and Andrews, 2017). Optimizing over both algorithm and hyper-parameters has been coined the *Combined Algorithm Selection and Hyper-parameter optimization problem* in the supervised learning literature (Thornton et al., 2012), **CASH** for short, or “tuning”. We adopt the same nomenclature in this paper. Notice that we can represent the choice of the learning algorithm with a new hyper-parameter. An instantiation of all hyper-parameter values (including the algorithm) is called a **configuration**.

A related problem in statistics is the problem of *model selection*. In both cases, one optimizes among a set of possible choices. However, there are conceptual differences of perspectives into the problem. Historically in statistics, different models are fit and then the final model is selected among all the ones fit. The choice is often manual by visualizing the model’s fit and residuals. Principled methods for model selection typically score the trade-off between model fitting and model complexity. Such a model selection criterion is the Bayesian Information Criterion (or similarly, the Akaike Information Criterion) scoring fitting using the *in-sample* data likelihood and penalizing for the model’s degrees of freedom. A main observation in model selection is that all models are trained on all training data; selection is based on the in-sample data fit.

In contrast, the CASH perspective (or arguably, the Machine Learning perspective) focuses on the learning algorithm, not the specific models (model instances to be precise). It is not the model that is selected, but the algorithm and its hyper-parameters (the configuration) to be applied on all data. For example, during cross-validating an algorithm several models are produced. None of them is the final model. They only serve to estimate how accurate are the models produced on average by the algorithm. The final model to return is the model trained on all data using the learning algorithm; all other models serve only for estimating performance purposes. Thus, CASH selects algorithms, not models, typically using out-of-sample estimation protocols (e.g., cross-validation).

It is not straightforward to apply the above techniques to the CASH problem in causal discovery. A first reason is that the task is inherently unsupervised. The true causal network is unknown. Thus, there is no direct way of estimating how well a model approximates the true causalities in the data. A second problem has to do with the performance metric to optimize. Performance metrics for typical supervised learning tasks, such as binary classification and regression have reached maturity. However, for causal discovery there is a range of metrics, some considering only the causal structure, parts of the causal structure (e.g., edge presence or absence), and others that consider the networks parameters (e.g., effect sizes). Despite its obvious importance to practitioners, the problem of tuning has not been extensively studied in the context of causal discovery.

3. Preliminaries

Causal Bayesian Networks (CBN) consist of a Directed Acyclic Graph (DAG) $\mathcal{G}$ and a set of probabilities $\mathcal{P}$ (Pearl, 2009). Nodes in the DAG represent variables (we use the terms node and variable interchangeably) and directed edges represent direct causality (in the context of nodes in the graph). Each node can represent a continuous or discrete variable. If $X \rightarrow Y$ in a DAG $\mathcal{G}$, we say that X is a parent of Y and Y is a child of X in $\mathcal{G}$. Two nodes that share a common child are called spouses.

The graph and the distribution are connected through the Causal Markov Condition (CMC). CMC states that every variable is independent of its non-effects given its direct causes. Given the CMC, $P(V)$ can be factorized as $P(X_1, \ldots, X_n) = \prod_i P(X_i | Pa_{\mathcal{G}}(X_i))$, where $Pa_{\mathcal{G}}(X_i)$ denote the parents of X_i in $\mathcal{G}$. Equivalently, the CMC entails a set of conditional independencies expected to hold in the joint probability distribution of variables in the graph. CBNs can only model causally sufficient systems, meaning that no pair of variables share an unmeasured common cause (confounder). Extensions of CBNs such as Maximal Ancestral Graphs (MAGs) (Richardson and Spirtes, 2002) model causal relationships in causally insufficient systems.

There are two major types of methods for learning causal structure: constraint-based and score-based. Constraint-based methods apply tests of conditional independence to a data set, and then try to identify all causal graphs that are consistent with these (in)dependencies. Hyper-parameters of these methods include the type of conditional independence test, the conditioning set size, and the significance threshold for rejecting the null hypothesis. Score-based methods try to identify the graphical model that leads to a factorization that is closest to the one estimated by the observational data (Cooper and Herskovits, 1992). Hyper-parameters of score-based methods include sampling and structure priors. In the CASH perspective, the choice of algorithm is also a hyper-parameter.

In most cases, the causal structure cannot be uniquely identified from the data. Instead, a set of DAGs will entail the same independence relationships (or equivalent factorizations). These graphs are called Markov equivalent, and share the same edges and some orientations. Both constraint-based and score-based algorithms typically return a Partially Directed Acyclic Graph (PDAG), that summarizes the invariant features of Markov equivalent graphs.

4. Approaches to Tuning for Causal Network Discovery

Two prior and one novel approach to tuning for causal discovery are presented. They can be viewed as representatives of three distinct methodologies to the problem, following different principles.

4.1 Network stability and the StARS algorithm

One principle for selecting configuration is based on the *stability of the networks output by the configuration* w.r.t. small changes to the input data. A specific instantiation of this principle is the *StARS* method (Liu et al., 2010) that was initially introduced specifically for tuning the lambda penalty hyper-parameter of the graphical lasso (Friedman et al., 2008); it was latter applied as a tuning method in (Raghu et al., 2018). The basic idea is to select a configuration that minimizes the instability of the network over perturbations of the input data. For a given configuration a, the network instability N_a is computed as the average edge instability. In turn, for each edge (X, Y) the instability ξ_{XY} is computed as follows: the probability p_{XY} of its presence in the network is estimated using subsampling (learning multiple networks using the same hyper-parameters on resamples of the data without replacement). The instability of the edge is defined as $\xi_{XY} \equiv 2p_{XY} \cdot (1 - p_{XY})$, i.e., it is twice the variance of a Bernoulli distribution with parameter p_{XY}. It is low when p_{XY} is close to 0 or 1, and high when it is close to 0.5.

Selecting the configuration with the minimum instability (maximum stability) seems reasonable at a first glance, but it could lead to configurations that consistently select the empty or the full graph. To avoid this situation, the authors of StARS propose the following: first, to order configurations with increasing density, and then "monotonize" their instability metric, i.e., define $\overline{N}(a_j) = max_{i \leq j} N(a_i)$, where $N(a)$ is the average instability for configuration a over all edges. Subsequently, to select the configuration $a^* = \arg\max_{a_i}\{\overline{N}(a_i) | \overline{N}(a_i) \leq \beta\}$, where β is a hyper-parameter of the StARS method. Intuitively, this strategy *selects the configuration that produces the densest network with an accepted value of instability (below a threshold β).* The pseudo-code is presented in Algorithm 1. The value of β is suggested to be 0.05 (Liu et al., 2010; Raghu et al., 2018). Liu et al. (2010) compare StARS to BIC and cross-validation, and find that StARS outperforms the alternative methods for high-dimensional settings. However, we note that StARS was evaluated specifically for tuning the graphical Lasso lambda parameter. It was not evaluated as a method to also tune the algorithm choice or other types of hyper-parameters. This is a research question explored in the experimental section.

StARS is based on the intuition that configurations whose output is very sensitive to the specific samples included in the dataset are undesirable. An alternative metric of a sort of instability appeared in (Roumpelaki et al., 2016). The intuition in this latter work is that marginalizing out some variables, should produce networks whose causal relations do not conflict with the network learned over all variables. An algorithm is proposed to measure consistency under marginalizations. It could also be employed for tuning, similar to StARS. The StARS approach measures sensitivity of a configuration to the specific sampling, while the latter measures sensitivity of the output to the specific choices of variables to measure. Another stability-based approach is in (Meinshausen and Bühlmann, 2010) specifically aiming to control the false positive rate of network edges.

A theoretical advantage of StARS is that it will select configurations that are robust to a few outliers in the data. A disadvantage is that it measures stability with respect to edge presence, i.e., the network skeleton; causal directionality is ignored. This could be ameliorated by considering edge-arrows in the stability calculations. Another obvious disadvantage is that it does not evaluate the model fit to the data. Thus, an algorithm that makes the same systematic error will be favored. This is not a problem when using a single algorithm (graphical Lasso), hence not identified as a problem in the original StARS paper. It is, however, a serious disadvantage that requires more fundamental changes to the basic principles of the StARS algorithm to make it successful to a broader tuning context, one that includes other network algorithms and hyper-parameter types.

Algorithm 1: StARS

```
Input: Dataset D over nodes $\mathbf{V}$, Configurations $\mathbf{A}$,
       Subsamples $\mathbf{S}$, threshold $\beta$
Output: Configuration $a^*$
for $a \in \mathbf{A}$ do
    for $s \in \mathbf{S}$ do
        $\mathcal{G}_{a,s} \leftarrow \texttt{causalAlg}_a(D_s)$ ;
        $D_{a,s} \leftarrow$ Density of $\mathcal{G}_{a,s}$;
    for each pair of variables $X, Y$ do
        $p_{a,XY} \leftarrow$ Frequency of an edge $(X, Y)$ in
          $\{G_{a,s}\}_{s \in \mathbf{S}}$;
        $\xi_{a,XY} = 2p_{a,XY}(1 - p_{a,XY})$;
    $N(a) \leftarrow \overline{\xi}_{a,XY}$ over all edges;
Rank $N(a)$ by increasing density;
$\overline{N}(a_j) \leftarrow max_{i \leq j} N(a_i)$ ;
$a^* = \arg\max_{a_i}\{\overline{N}(a_i) | \overline{N}(a_i) \leq \beta\}$ ;
```

Algorithm 2: BIC selection

```
Input: Dataset D over nodes
       $\mathbf{V}$, Configurations $\mathbf{A}$
Output: Configuration $a^*$
for $a \in \mathbf{A}$ do
    $G_a \leftarrow \texttt{causalAlg}_a(D)$;
    $G'_a \leftarrow \texttt{pdagToDag}(G_a)$;
    $LL_a \leftarrow 2 \log P(D|G'_a)$;
    $BIC_a \leftarrow \log(n)k - LL_a$;
$a^* = \arg\min_{a \in A} BIC_a$;
```

4.2 Balancing Fitting with Model Complexity

Another principle for selecting model and corresponding configuration is to select based on the best trade off between *in-sample* fitting of the data and the model complexity. This approach has also been used for other unsupervised problems, like clustering (Hofmeyr, 2018). A specific instantiation of the principle for causal discovery tuning appeared in (Maathuis et al., 2009) and is based on the Bayesian Information Criterion (BIC). BIC scoring was also compared with StARS (Liu et al., 2010) for learning Markov Networks using the graphical Lasso. BIC scores a causal model based on the likelihood of the data given the causal model and penalizes with the degrees of freedom of the model (Schwarz, 1978). Alternatively, one could use the AIC, which has been shown to be asymptotically equivalent to leave-one-out cross validation (Stone, 1977). When the model is a CBN, the likelihood of the data is computed based on the corresponding factorization :

$$P(D|G) = \prod_{ij} P(x_{ij}|G) = \prod_{ij} P(x_{ij}|Pa(i)) \tag{1}$$

where P is the probability or probability density function, x_{ij} is the value of the i-th variable of the j-th sample, and $Pa(i)$, the parents of the variable i in G. In order to compute $P(x_{ij}|Pa(i))$ a *parametric, statistical model* needs to be fit with outcome each variable i given its *parents* $Pa(i)$. The BIC of a graph is the likelihood, penalized for the number of parameters. Lower values of BIC and AIC are better. Any alternative principle for scoring the trade off of model fitting with complexity could also be employed. Examples include the Minimum Message Length, the Minimum Description Length, PAC error bounds on the model, or even the Kolmogorov complexity.

There are several advantages to BIC and AIC scoring for model selection. First, several algorithms use BIC internally to search and score the best possible causal model, proving its effectiveness. Using Eq.1 one needs to fit models for each node X_i from only its parents in the graph $Pa(X_i)$. In comparison, the proposed method below employs models for each node given its Markov Blanket. The latter is a superset of the parents and thus requires more samples to be fit accurately. There are also some disadvantages, a major one being that it requires the computation of likelihood and the degrees of freedom of the causal model. This is typically possible only with statistical, parametric models such as Gaussian and multinomial employed in the current paper. Computing the BIC for Decision Trees, Random Forests and other types of non-parametric machine learning models is not possible. Thus, *using BIC does not allow the full use of the ML arsenal in predictive modeling*. In addition, BIC and AIC cannot be computed for Maximal Ancestral Graphs with discrete variables.

4.3 Tuning based on Predictive Performance

The main principle to our proposed approach is to treat a causal model as a set of predictive models. Subsequently, we can evaluate the configurations producing causal models using *out-of-sample performance estimation protocols* such as cross-validation. A similar approach has been suggested for other unsupervised learning tasks, such as dimensionality reduction with the PCA algorithm (Perry, 2009) and for clustering (Fu and Perry, 2017). In the framework of potential outcomes, out-of-sample protocols have also been used to improve the estimation of conditional average treatment effects (Saito and Yasui, 2019).

Specifically, a causal model $\mathcal{G}$ induces a Markov Blanket $MB(X)$ for each node of the graph. The $MB(X)$ is the minimal set that renders X conditionally independent of any other node. It is unique for distributions faithful to the graph and it is invariant among all graphs in the same Markov equivalence class. Moreover, under some conditions it is the minimal set of nodes that is necessary and sufficient for optimal prediction of X (Tsamardinos et al., 2003). Hence, assuming that we model the functional relationship between $MB(X)$ and X correctly (e.g., use a learning algorithm that correctly represent the distribution), a best-performing predictive model can be constructed.

We now propose an algorithm that selects the configuration resulting in the best set of predictive models (one for each node), using an out-of-sample protocol. The algorithm is called **OCT** and is described in Algorithm 3. It takes as input a data set over variables $\mathbf{V}$ and a set of configurations of causal discovery algorithms $\mathbf{A}$. The algorithm also takes as input the number of folds K for the cross validation. For each configuration a and each fold k, we estimate a causal graph by running the corresponding configuration $causalAlg_a$ on the training data set D_k^{train}. Subsequently, **OCT** evaluates the predictive performance of $causalAlg_a$ by identifying the Markov Blanket $MB(X)$ of each variable X, building a predictive model for X based on $MB(X)$, and estimating the prediction error of each predictive model on the test set D_k^{test}. The overall performance of $causalAlg_a$ in fold k is the average performance of all of the predictive models (one for each variable) in that fold. Asymptotically, the true causal graph will be among the models that achieve the best performance:

Theorem 1 *Assuming that the following conditions hold: (a) Data are generated by a causal Bayesian Network $\mathcal{G}_{true}$ over variables $\mathbf{V}$. (b) The learning algorithm can exactly learn the conditional distribution of each node $V \in \mathbf{V}$ given its MB. (c) The learning algorithm uses a proper*

Algorithm 3: Out-of-sample Causal Tuning (OCT)

```
Input: Dataset D over nodes $\mathbf{V}$, Configurations $\mathbf{A}$, Folds K
Output: Configuration $a^*$
for $a \in A$ do
    for $k = 1$ to $K$ do
        $\mathcal{G}_{a,k} = \texttt{causalAlg}_a(D_k^{train})$ ;
        for $X \in \mathbf{V}$ do
            $MB_{a,k,X} \leftarrow$ Markov Blanket of $X$ in $\mathcal{G}_{a,k}$;
            $M_{a,k,X} \leftarrow \texttt{fitModel}(X, MB_{a,k,X}, D_k^{train})$;
            $P_{a,k,X} \leftarrow \texttt{evaluatePerf}(M_{a,k,X}, D_k^{test})$
        $P_{a,k} \leftarrow \overline{P_{a,k,X}}$ over $\mathbf{V}$.
    $P_a \leftarrow \overline{P_{a,k}}$ over $\mathbf{k}$.
$a^* \leftarrow \underset{a \in A}{argmax} P_a$
```

scoring criterion. Then any DAG $\mathcal{G}$ for which $MB_{\mathcal{G}}(V) = MB_{\mathcal{G}_{true}}(V)\ \forall\ V \in \mathbf{V}$ will asymptotically have the maximum score.

Proof If a proper scoring rule is used, then the highest performance for each variable V will be obtained by the true probability distribution $P(V|\mathbf{V} \setminus V)$, which is equal to $P(V|MB(V))$. ■

Induced causal models that miss members of a $MB(X)$ will achieve a lower predictive performance than possible, as they lack informational predictors. Causal models that add false positive members of a $MB(X)$ may result in overfitting in finite samples. However, in the large sample limit, Markov Blankets that include a superset of the true set will also achieve the maximum score, and could be selected by **OCT**. To address this problem, we also examined a version of **OCT**, which we call **OCTs**: In this version, among configurations whose performance is similar (not statistically significantly different) to the optimal performance, we pick the one with the smallest Markov Blanket sets (average over all variables and folds).

4.4 Strengths and Limitations

Advantages of **OCT** are that it does not inherently need to make parametric assumptions about the data distribution; one could potentially employ any applicable modelling method in Machine Learning or statistics, and it will asymptotically select the optimal configuration with respect to prediction (assuming the conditions in Theorem 1 hold). On the other hand, there are two major limitations of **OCT**. The first is that it does not directly penalize false positives. Modern learning algorithms are robust to irrelevant or redundant variables. Thus, even if Markov Blankets contain false positives, they will be ignored by the learning algorithms and their predictive performance may still be optimal. Penalizing dense graphs may ameliorate this shortcoming. A second problem is the choice of the predictive modeling algorithm. If the algorithm cannot approximate the true distribution, the procedure can collapse. Ideally, one should perform CASH on each model with obvious impacts on the computational performance of the method.

5. Experimental Setup

Graphs and Data simulation. For our experiments we focused on simulated random DAGs varying the number of nodes and their density. We tested our methods on continuous and discrete data. For each combination of data type, density and sample size, we simulate 20 datasets from random parametrizations, as follows: For discrete data, the number of categories range from 2 to 5 and the conditional probability tables are sampled randomly from a Dirichlet distribution with $a = 0.5$. For continuous data, we simulate linear Gaussian structural equation models with absolute coefficients ranging between 0.1 and 0.9.
Algorithms and Packages.We used a variety of causal discovery algorithms and hyper-parameters as "configurations": We used Tetrad implementations of PC, PC-stable, CPC, CPC-stable and FGES (http://www.phil.cmu.edu/tetrad). We used MMHC from the recent version of Causal Explorer (Aliferis et al., 2003) and the bnlearn package (Scutari, 2010) for discrete and continuous data respectively. For constraint based algorithms, we varied the level of significance for the conditional independence test (0.001, 0.005, 0.01, 0.05 and 0.1) and the maximum conditioning set (4 and unlimited). For FGES we varied the BIC penalty discount (1, 2, 4) for continuous data and the BDeu sample and structure priors (1, 2, 4) for discrete data. Overall, we have 48 algorithm and hyper-parameter combinations (configurations) for continuous and 54 for discrete data sets.
OCT configuration For OCT and continuous data, we used a linear regression model as the learning algorithm. We standardized the data and computed the predictive performance as the square root of the sum of squared residuals over all nodes and samples (pooled residuals over all folds). For the OCTs version, we used a t-test to compare the sum of squared residuals. For discrete data, we used a classification tree. For OCTs, we used accuracy of predictions as the performance metric. Notice that accuracy is not a proper scoring rule.
Evaluation Metrics We evaluate the graphs according to the following metrics. The Structural Hamming Distance (SHD) counts the number of the steps needed to reach the ground truth CPDAG from the PDAG of the estimated graph. These modifications include edge removal, addition and changes in orientation (Tsamardinos et al., 2006). Structural Intervention Distance (SID) counts the number of pairs for which the intervention distribution is falsely estimated on the learnt graph (Peters and Bühlmann, 2015). SIDu and SIDl are the upper and lower limits of SID in the Markov Equivalence class of the network. Due to space constraints, we only show SIDu in the figures. Results for SIDl are similar.

5.1 Comparative Evaluation

In Figure 1 we now compare the performance of **OCT**, **OCTs**, **BIC**, **AIC** and **StARS** for selecting optimal configurations, over increasing density. Each tuning method was run to obtain the selected configuration for each dataset; a network is then produced using the selected configuration on the complete dataset. We also include the performance of the random selection of a configuration with uniform probability (**Random**). The y-axes correspond to the difference of performance achieved by a tuning method compared to the oracle selection for the given metric. An optimal selection of configuration corresponds to 0 difference; lower is better for all plots and metrics. Grey area shows the range from best to worst performance. Performance is estimated on datasets with 50 variables and 1000 samples, simulated from graphs with increasing mean parents per node (2, 4, and 6).

In general **OCT** and/or **OCTs** perform on par or better than other methods w.r.t. most metrics and scenarios, with some exceptions discussed below: For dense continuous networks (6 number of

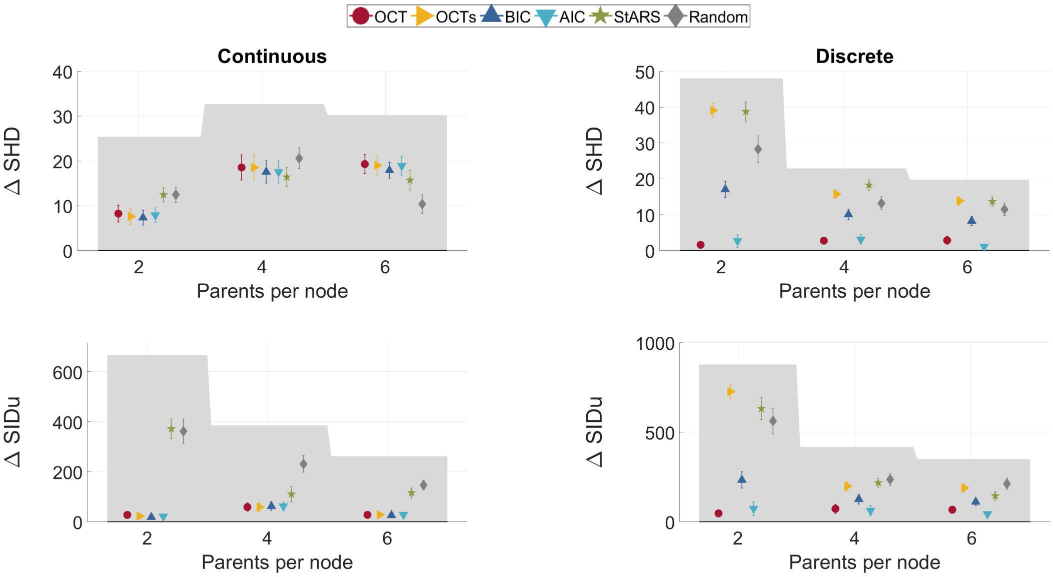

Figure 1: Comparison of the tuning methods for continuous and discrete data over increasing number of parents per node, on datasets with 50 variables and 1000 samples. Each point corresponds to the difference of performance achieved by each method compared to the oracle selection (zero is optimal, lower is better). **OCT** performs on par or better than other methods in many settings.

parents), all tuning methods perform worse than random w.r.t. SHD. OCT selects configurations that lead to denser graphs and have fewer false negative edges. For this reason, the configuration selected by OCT is close to optimal in the SIDu metric. SHD is a metric that penalizes false positives and false negatives equally. However, for probabilistic and causal predictions, false negatives often lead to larger biases than false positives. In addition to dense continuous networks, OCTs performs worse than most algorithms in discrete data. This is due to the use of accuracy as a performance metric. Accuracy is not a proper scoring rule, and per-node accuracies among best-scoring configurations are often identical. Thus, OCTs selects very sparse graphs, leading to high SHDs and SIDus. In the future, we plan to explore additional metrics, such as multi-class AUCs (Tang et al., 2011).

Figure 2 compares the methods over network size and sample size, w.r.t the SHD and SIDu for continuous data with 50 variables and 2 parents per node on average. OCT, OCTs, BIC, and AIC perform similarly, and select configurations that are close to the optimal. Notice that ΔSHD increases with increasing sample size. While this seems counter-intuitive, we note that this is the difference in SHDs between the selected and the optimal configuration. The actual mean SHD is lower for larger sample sizes, as expected.

The average computational time on 10 graphs of 50 nodes is as follows. For continuous data OCT: 90 (1.6), BIC: 49 (0.6), StARS: 136 (2.2) while for discrete data, OCT: 2065 (1599), BIC: 317 (241), StARS: 342 (151) (time in seconds, standard deviation in parenthesis). Thus, OCT is always more expensive than BIC, and more expensive than StARS in the discrete case. However, we note that the computational cost is not prohibitive for graphs of modest size (20-100 nodes).

Our results show that **OCT** and **OCTs** have similar behavior in continuous variables and the performance of **OCT** is better for discrete data. In addition, they perform equally well with **BIC**

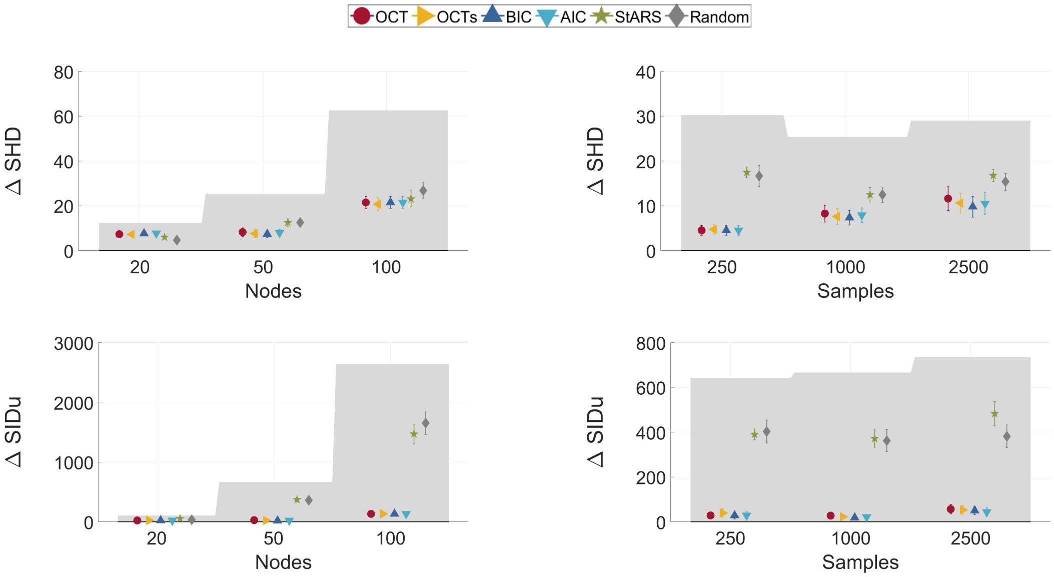

Figure 2: Comparison of the methods over increasing network (left) and sample sizes (right), on continuous datasets with 2 parents per variable. All methods besides StARS perform similarly.

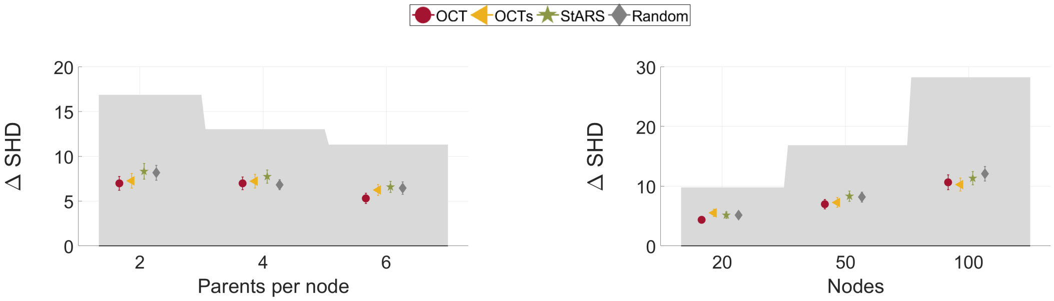

Figure 3: Comparison of the methods over increasing number of parents (left) and network size (right) for networks with hidden confounders. OCT and OCTs marginally increase performance.

and **AIC** approaches. **StARS** consistently underperforms. While the method is able to tune the lambda parameter of graphical Lasso, its success does not seem to transfer to Bayesian Network tuning of algorithms and their hyper-parameters, at least in its current implementation.

Despite the computational cost of **OCT** it is *generalizable to multiple cases where **BIC** or **AIC** cannot be computed.* For example, BIC and AIC are not suitable for models without likelihood estimators. In addition, they are not applicable in discrete causal models with hidden variables. As a proof of concept, we applied our method for tuning algorithms that learn causal graphs with hidden confounders: We simulated 50 DAGs with varying density and number of nodes, and set 30% of the nodes as latent. We used FCI and RFCI implemented in Tetrad, varying the level of significance of conditional independence test and the maximum conditioning set, as before. We tuned the algorithms with OCT, OCTs and StARS. The results (Fig 3), show that both OCT and OCTs behave reasonably, and lead to increased performance, albeit marginally. StARS and random guessing perform slightly worse, but similarly. This is probably due to the fact that the number of algorithmic configurations is smaller, and all algorithms are of the same type (constraint-based).

6. Discussion and Conclusions

It is impossible to evaluate a causal discovery method on its causal predictions based on observational data alone. However, we can evaluate causal models w.r.t. their predictive power. We propose an algorithm, called Out-of-sample Causal Tuning (OCT), that employs this principle to select among several choices of algorithms and their hyper-parameter values to use on a given causal discovery problem. It performs on par or better than two previous other selection methods based on network stability across subsamples and the Bayesian and the Akaike Information Criteria. However, the optimal selection of algorithm and hyper-parameter values depends on the metric of performance, thus no tuning method can simultaneously optimize for all metrics. Even though OCT optimizes the predictive power of the resulting causal models, it still manages to simultaneously optimize the causalities encoded in the network, reasonably well. In the present study we used only synthetic data following parametric distributions: either variables are all continuous following a multivariate Gaussian or are all discrete following a multinomial. A major current limitation of OCT is that it assumes an appropriate predictive modeling algorithm for the data is employed (linear regression and Decision Tree, within the scope of the experiments). The efficient tuning of the predictive modeling required by OCT is a natural next step in this research direction.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 617393

References

C. F. Aliferis, I. Tsamardinos, A. R. Statnikov, and L. E. Brown. Causal explorer: A causal probabilistic network learning toolkit for biomedical discovery. In *METMBS*, 2003.

G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

J. H. Friedman, T. J. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9 3:432–41, 2008.

W. Fu and P. Perry. Estimating the number of clusters using cross-validation. *Journal of Computational and Graphical Statistics*, 2017.

D. P. Hofmeyr. Degrees of freedom and model selection for kmeans clustering. *ArXiv*, 2018.

H. Liu, K. Roeder, and L. A. Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. *NeurIPS*, 24 2:1432–1440, 2010.

M. Maathuis, M. Kalisch, and P. Buhlmann. Estimating high-dimensional intervention effects from observational data. *Annals of Statistics*, 37:3133–3164, 2009.

N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

P. O. Perry. *Cross-validation for unsupervised learning*. PhD thesis, Stanford University, 2009.

J. Peters and P. Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Computation*, 27:771–799, 2015.

V. K. Raghu, A. Poon, and P. V. Benos. Evaluation of causal structure learning methods on mixed data types. *Proceedings of machine learning research*, 92:48–65, 2018.

J. D. Ramsey and B. Andrews. A comparison of public causal search packages on linear, gaussian data with no latent variables. *CoRR*, 2017.

T. Richardson and P. Spirtes. Ancestral graph markov models. *Ann. Statist.*, 30(4):962–1030, 08 2002.

A. Roumpelaki, G. Borboudakis, S. Triantafillou, and I. Tsamardinos. Marginal causal consistency in constraint-based causal learning. In *CFA@UAI*, 2016.

Y. Saito and S. Yasui. Counterfactual cross-validation: Effective causal model selection from observational data. *arXiv preprint arXiv:1909.05299*, 2019.

G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

M. Scutari. Learning bayesian networks with the bnlearn r package. *Journal of Statistical Software, Articles*, 35(3):1–22, 2010.

M. Stone. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):44–47, 1977.

K. Tang, R. Wang, and T. Chen. Towards maximizing the area under the roc curve for multi-class classification problems. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 483–488. AAAI Press, 2011.

C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-weka: combined selection and hyperparameter optimization of classification algorithms. In *KDD '13*, 2012.

I. Tsamardinos and C. F. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *AISTATS*, 2003.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. In *KDD*, pages 673–678, 2003.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.