# Gradient-Based Neural DAG Learning

**Sébastien Lachapelle**[1] **Philippe Brouillard**[1] **Tristan Deleu**[1] **Simon Lacoste-Julien**[1,2]

[1]Mila, Université de Montréal
[2]Canada CIFAR AI Chair

## 1 Introduction

Structure learning and causal inference have many important applications in different areas of science such as genetics [5, 12], biology [13] and economics [7]. *Bayesian networks* (BN), which encode conditional independencies using *directed acyclic graphs* (DAG), are powerful models which are both interpretable and computationally tractable. *Causal graphical models* (CGM) [12] are BNs which support *interventional* queries like: *What will happen if someone external to the system intervene on variable X?* Recent work suggests that causality could partially solve challenges faced by current machine learning systems such as robustness to out-of-distribution samples, adaptability and explainability [8, 6]. However, structure and causal learning are daunting tasks due to both the combinatorial nature of the space of structures and the question of *structure identifiability* [12]. Nevertheless, these graphical models known qualities and promises of improvement for machine intelligence renders the quest for structure/causal learning appealing. The problem of structure learning can be seen as an inverse problem in which the learner tries to infer the causal structure which has generated the observation.

In this work, we propose a novel score-based method [5, 12] for structure learning named GraN-DAG which makes use of a recent reformulation of the original *combinatorial problem* of finding an optimal DAG into a *continuous constrained optimization problem*. In the original method named NOTEARS [18], the directed graph is encoded as a *weighted adjacency matrix* $W$ which represents coefficients in a linear *structural equation model* (SEM) [7]. To enforce acyclicity, the authors propose a constraint which is both efficiently computable and easily differentiable.

Most popular score-based methods for DAG learning usually tackle the combinatorial nature of the problem via greedy search procedures relying on multiple heuristics [3, 2, 11]. Moving toward the continuous paradigm allows one to use gradient-based optimization algorithms instead of hand-designed greedy search algorithms.

Our first contribution is to extend the work of [18] to deal with nonlinear relationships between variables using neural networks (NN) [4]. GraN-DAG is general enough to deal with a large variety of parametric families of conditional probability distributions. To adapt the acyclicity constraint to our nonlinear model, we use an argument similar to what is used in [18] and apply it first at the level of *neural network paths* and then at the level of *graph paths*. Our adapted constraint allows us to exploit the full flexibility of NNs. On both synthetic and real-world tasks, we show GraN-DAG outperforms other approaches which leverage the continuous paradigm, including DAG-GNN [16], a recent nonlinear extension of [18] independently developed which uses an evidence lower bound as score.

Our second contribution is to provide a missing empirical comparison to existing methods that support nonlinear relationships but tackle the optimization problem in its discrete form using greedy search procedures such as CAM [2]. We show that GraN-DAG is competitive on the wide range of tasks we considered.

Contact: sebastien.lachapelle@umontreal.ca
NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks

## 2 Background

### 2.1 Causal graphical models

We suppose the natural phenomenon of interest can be described by a random vector $X \in \mathbb{R}^d$ entailed by an underlying CGM $(P_X, \mathcal{G})$ where $P_X$ is a probability distribution over $X$ and $\mathcal{G} = (V, E)$ is a DAG [12]. Each node $i \in V$ corresponds to exactly one variable in the system. Let $\pi_i^{\mathcal{G}}$ denote the set of parents of node $i$ in $\mathcal{G}$ and let $X_{\pi_i^{\mathcal{G}}}$ denote the random vector containing the variables corresponding to the parents of $i$ in $\mathcal{G}$. We assume there are no hidden variables. In a CGM, the distribution $P_X$ is said to be *Markov* to $\mathcal{G}$ which means we can write the probability density function (pdf) as $p(x) = \prod_{i=1}^{d} p_i(x_i|x_{\pi_i^{\mathcal{G}}})$ where $p_i(x_i|x_{\pi_i^{\mathcal{G}}})$ is the conditional pdf of variable $X_i$ conditioned on $X_{\pi_i^{\mathcal{G}}}$. A CGM can be thought of as a BN in which directed edges are given a causal meaning, allowing it to answer queries regarding *interventional distributions* [5].

### 2.2 Structure identifiability

In general, it is impossible to recover $\mathcal{G}$ only given samples from $P_X$. It is, however, customary to rely on a set of assumptions to render the structure fully or partially *identifiable*.

**Definition 1.** *Given a set of assumptions $A$ on a CGM $\mathcal{M} = (P_X, \mathcal{G})$, its graph $\mathcal{G}$ is said to be identifiable from $P_X$ if there exists no other CGM $\tilde{\mathcal{M}} = (\tilde{P}_X, \tilde{\mathcal{G}})$ satisfying all assumptions in $A$ such that $\tilde{\mathcal{G}} \neq \mathcal{G}$ and $\tilde{P}_X = P_X$.*

There are many examples of graph identifiability results for continuous variables [11, 9, 14, 17] as well as for discrete variables [10]. Those results are obtained by assuming that the conditional pdf $p_i \; \forall i$ belongs to a specific parametric family $\mathcal{P}$. For example, if one assumes that

$$X_i | X_{\pi_i^{\mathcal{G}}} \sim \mathcal{N}(f_i(X_{\pi_i^{\mathcal{G}}}), \sigma_i^2) \;\; \forall i \tag{1}$$

where $f_i$ is a nonlinear function satisfying some mild regularity conditions, then $\mathcal{G}$ is identifiable from $P_X$ (see [11] for the complete theorem and its proof). We will make use of this results in Section 4.

### 2.3 NOTEARS: Continuous optimization for structure learning

Structure learning is the problem of learning $\mathcal{G}$ using a data set of $n$ samples $\{x^{(1)}, ..., x^{(n)}\}$ from $P_X$. *Score-based* approaches [12] cast this estimation problem as an optimization problem over the space of DAGs, i.e. $\hat{\mathcal{G}} = \arg\max_{\mathcal{G} \in \text{DAG}} \text{Score}(\mathcal{G})$. The score is usually the maximum likelihood of your data given a certain model. Most score-based methods embrace the combinatorial nature of the problem via greedy search procedures [3, 2]. We now present the work of [18] which approaches the problem from a continuous optimization perspective.

To cast the combinatorial optimization problem into a continuous constrained one, [18] proposes to encode the graph $\mathcal{G}$ on $d$ nodes as a weighted adjacency matrix $U = [u_1 | \dots | u_d] \in \mathbb{R}^{d \times d}$ which represents (possibly negative) coefficients in a linear *structural equation model* (SEM) [7] of the form $X_i := u_i^\top X + N_i \;\; \forall i$ where $N_i$ is a noise variable. Let $\mathcal{G}_U$ be the directed graph associated with the SEM and let $A_U$ be the (binary) adjacency matrix associated with $\mathcal{G}_U$. One can see that the following equivalence holds:

$$(A_U)_{ij} = 0 \iff U_{ij} = 0 \tag{2}$$

To make sure $\mathcal{G}_U$ is acyclic, the authors propose the following constraint on $U$:

$$\operatorname{Tr} e^{U \odot U} - d = 0 \tag{3}$$

where $e^M \triangleq \sum_{k=0}^{\infty} \frac{M^k}{k!}$ is the *matrix exponential* and $\odot$ is the Hadamard product. It can be shown that $\mathcal{G}_U$ is acyclic iff the constraint is satisfied (see [18] for a proof).

The authors propose to use a regularized negative least square score (maximum likelihood for a Gaussian noise model). The resulting continuous constrained problem is

$$\max_U \mathcal{S}(U, \mathbf{X}) \triangleq -\frac{1}{2n} \|\mathbf{X} - \mathbf{X}U\|_F^2 - \lambda \|U\|_1 \quad \text{s.t.} \quad \operatorname{Tr} e^{U \odot U} - d = 0 \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the design matrix containing all $n$ samples. The nature of the problem has been drastically changed: we went from a combinatorial to a continuous problem. The difficulties of combinatorial optimization have been replaced by those of non-convex optimization, since the feasible set is non-convex. Nevertheless, a standard numerical solver for constrained optimization such has an *augmented Lagrangian method* (AL) [1] can be applied to get an approximate solution.

## 3 GraN-DAG: Gradient-based neural DAG learning

We propose a new nonlinear extension to the framework presented in Section 2.3. For each variable $X_i$, we learn a fully connected neural network with $L$ hidden layers parametrized by $\phi_{(i)} \triangleq \{W_{(i)}^{(1)}, \dots, W_{(i)}^{(L+1)}\}$ where $W_{(i)}^{(\ell)}$ is the $\ell$th weight matrix of the $i$th NN. Each NN takes as input $X_{-i} \in \mathbb{R}^d$, i.e. the vector $X$ with the $i$th component masked to zero, and outputs $\theta_{(i)} \in \mathbb{R}^m$, the $m$-dimensional parameter vector of the desired distribution family for variable $X_i$. The fully connected NNs have the following form

$$\theta_{(i)} \triangleq W_{(i)}^{(L+1)} g(\dots g(W_{(i)}^{(2)} g(W_{(i)}^{(1)} X_{-i})) \dots) \ \ \forall i \tag{5}$$

where $g$ is a nonlinearity applied element-wise. Let $\phi \triangleq \{\phi_{(1)}, \dots, \phi_{(d)}\}$ represents all parameters of all $d$ NNs. Without any constraint on its parameter $\phi_{(i)}$, neural network $i$ models the conditional pdf $p_i(x_i|x_{-i}; \phi_{(i)})$. Note that the product $\prod_{i=1}^d p_i(x_i|x_{-i}; \phi_{(i)})$ is not a valid joint pdf since it does not decompose according to a DAG. We now show how one can constrain $\phi$ to make sure the product of all conditionals outputted by the NNs is a valid joint pdf. The idea is to define a new weighted adjacency matrix $A_\phi$ similar to the matrix $U$ encountered in Section 2.3, which can be directly used inside the constraint of Equation 3 to enforce acyclicity.

### 3.1 Neural network connectivity

Before defining the weighted adjacency matrix $A_\phi$, we need to focus on how one can make some NN outputs unaffected by some inputs. Since we will discuss properties of a single NN, we drop the NN subscript $(i)$ to improve readability.

We will use the term *neural network path* to refer to a computation path in a NN. For example, in a NN with two hidden layers, the sequence of weights $(W_{h_1 j}^{(1)}, W_{h_2 h_1}^{(2)}, W_{k h_2}^{(3)})$ is a NN path from input $j$ to output $k$. We say that a NN path is *inactive* if at least one weight along the path is zero. We can loosely interpret the *path product* $|W_{h_1 j}^{(1)}||W_{h_2 h_1}^{(2)}||W_{k h_2}^{(3)}| \geq 0$ as the strength of the NN path, where a path product equal to zero if and only if the path is inactive. Note that if all NN paths from input $j$ to output $k$ are inactive (i.e. the sum of their path products is zero), then output $k$ does not depend on input $j$ anymore since the information in input $j$ will never reach output $k$. The sum of all path products from every input $j$ to every output $k$ can be easily computed by taking the product of all the weight matrices in absolute value.

$$C \triangleq |W^{(L+1)}| \dots |W^{(2)}||W^{(1)}| \in \mathbb{R}_{\geq 0}^{m \times d} \tag{6}$$

where $|W|$ is the element-wise absolute value of $W$. It can be verified that $C_{kj}$ is the sum of all NN path products from input $j$ to output $k$. To have $\theta$ independent of variable $X_j$, it is sufficient to have $\sum_{k=1}^m C_{kj} = 0$. This is useful since, to render our architecture acyclic, we need to force some neural network inputs to be inactive (this corresponds to removing edges in our graph).

### 3.2 A weighted adjacency matrix

We now define a weighted adjacency matrix $A_\phi$ that can be used in constraint of Equation 3.

$$(A_\phi)_{ji} \triangleq \begin{cases} \sum_{k=1}^m \left(C_{(i)}\right)_{kj}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $C_{(i)}$ denotes the connectivity matrix of the NN associated with variable $X_i$.

As the notation suggests, $A_\phi \in \mathbb{R}_{\geq 0}^{d \times d}$ depends on all weights of all NNs. Moreover, it can effectively be interpreted as a weighted adjacency matrix similarly to what we presented in Section 2.3, since we

have that

$$(A_\phi)_{ij} = 0 \implies \theta_{(j)} \text{ does not depend on variable } X_i \tag{8}$$

We note $\mathcal{G}_\phi$ to be the directed graph entailed by parameter $\phi$. We can now write our adapted acyclicity constraint:

$$h(\phi) \triangleq \operatorname{Tr} e^{A_\phi} - d = 0 \tag{9}$$

This guarantees acyclicity. The argument is identical to the linear case, except that now we rely on implication (8) instead of (2).

### 3.3 A differentiable score and its optimization

We propose solving the maximum likelihood optimization problem

$$\max_\phi \mathbb{E}_{X \sim P_X} \sum_{i=1}^{d} \log p_i(X_i | X_{\pi_i^\phi}; \theta_{(i)}) \quad \text{s.t.} \quad \operatorname{Tr} e^{A_\phi} - d = 0 \tag{10}$$

where $\pi_i^\phi$ denotes the set of parents of variable $i$ in graph $\mathcal{G}_\phi$. Note that $\sum_{i=1}^{d} \log p_i(X_i | X_{\pi_i^\phi}; \theta_{(i)})$ is a valid log-likelihood function when constraint (9) is satisfied.

As suggested in [18], we apply an augmented Lagrangian approach to get an approximate solution to program (10). Augmented Lagrangian methods consist of optimizing a sequence of subproblems for which the exact solutions are known to converge to a stationary point of the constrained problem under some regularity conditions [1]. We approximately solve each subproblem using RMSprop [15], a stochastic gradient descent variant popular for NN.

## 4 Experiments

We empirically compare GraN-DAG to various baselines (both in the continuous and combinatorial paradigm), namely DAG-GNN [16], NOTEARS [18], RESIT

We first present a comparison on synthetic data sets. We sampled 10 graphs (e.g. with 50 nodes and an average of 200 edges) and data distributions of the form $X_i | X_{\pi_i^\mathcal{G}} \sim \mathcal{N}(f_i(X_{\pi^\mathcal{G}}), \sigma_i^2)$ with $f_i \sim \mathcal{GP}$ and evaluated different methods using SHD and SID (we report the average and the standard deviation over those data sets). Note that we are in the identifiable case presented in Section 2.2. GraN-DAG, NOTEARS and CAM all make the correct gaussian assumption in their respective models. In Table 1 we report a subset of our results. GraN-DAG outperforms other continuous approaches while being competitive with the best performing discrete approach we considered.

In addition, we considered a well known real world data set which measures the expression level of different proteins and phospholipids in human cells [13] (the ground truth graph has 11 nodes and 17 edges). We found GraN-DAG to be competitive with other approaches.

Table 1: Evaluation of different methods for structure learning. The synthetic experiment has 50 nodes and an average of 200 edges. We have more experiments which show roughly this same ranking.

| | | Synthetic | | Protein data set | |
|---|---|---|---|---|---|
| | | SHD | SID | SHD | SID |
| Continuous | **GraN-DAG** | **102.6±21.2** | **1060.1±109.4** | **13** | **47** |
| | DAG-GNN | 191.9±15.2 | 2146.2±64 | 16 | 44 |
| | NOTEARS | 202.3±14.3 | 2149.1±76.3 | 21 | 44 |
| Discrete | **CAM** | **98.8±20.7** | **1197.2±125.9** | **12** | **55** |
| | RANDOM | 708.4±234.4 | 1921.3±203.5 | 21 | 60 |

Our implementation of GraN-DAG can be found here.

**Acknowledgments**

This research was partially supported by the Canada CIFAR AI Chair Program, Calcul Québec (www.calculquebec.ca), Compute Canada (www.computecanada.ca) and by a Google Focused Research award. The authors would like to thank Rémi Le Priol, Tatjana Chavdarova, Charles Guille-Escuret and Yoshua Bengio for insightful discussions as well as Florian Bordes for technical support.

## References

[1] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[2] P. Bühlmann, J. Peters, and J. Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *Annals of Statistics*, 2014.

[3] D.M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 2003.

[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[5] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. MIT Press, 2009.

[6] S. Magliacane, T. van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J.M. Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems 31*, 2018.

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

[8] J. Pearl. The seven tools of causal inference, with reflections on machine learning. *Commun. ACM*, 2019.

[9] J. Peters and P. Bühlman. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 2014.

[10] J. Peters, D. Janzing, and B. Schölkopf. Causal inference on discrete data using additive noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[11] J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 2014.

[12] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, 2017.

[13] K. Sachs, O. Perez, D. Pe'er, D.A. Lauffenburger, and G.P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 2005.

[14] S. Shimizu, P.O. Hoyer, A. Hyvärinen, and A. Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 2006.

[15] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[16] Y. Yu, J. Chen, T. Gao, and M. Yu. DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[17] K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009.

[18] X. Zheng, B. Aragam, P.K. Ravikumar, and E.P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems 31*, 2018.