NTS-NOTEARS: Learning Nonparametric DBNs With Prior Knowledge

Xiangyu Sun [1] Oliver Schulte [1] Guiliang Liu [2] Pascal Poupart [3]

[1]Simon Fraser University [2]The Chinese University of Hong Kong, Shenzhen [3]University of Waterloo

Abstract

We describe NTS-NOTEARS, **a score-based structure learning method for time-series data to learn dynamic Bayesian networks** (DBNs) that captures nonlinear, lagged (inter-slice) and instantaneous (intra-slice) relations among variables. NTS-NOTEARS utilizes 1D convolutional neural networks (CNNs) to model the dependence of child variables on their parents; 1D CNN is a neural function approximation model well-suited for sequential data. DBN-CNN structure learning is formulated as a continuous optimization problem with an acyclicity constraint, following the NOTEARS DAG learning approach. We show how prior knowledge of dependencies (e.g., forbidden and required edges) can be included as additional optimization constraints.

Scan this QR code for the GitHub repo

Research Gaps

Method	Score-Based	Nonlinear	Temporal	Instantaneous Edges	Acyclic
cMLP	✓	✓	✓	✗	✓
Economy-SRU	✓	✓	✓	✗	✓
GVAR	✓	✓	✓	✗	✓
VAR-LINGAM	✓	✗	✓	✓	✓
PCMCI+*	✗	✓	✓	✓	✓
TCDF*	✓	✓	✓	✓	✗
NOTEARS	✓	✗	✗	✓	✓
GraN-DAG	✓	✓	✗	✓	✓
NOTEARS-MLP	✓	✓	✗	✓	✓
DYNOTEARS*	✓	✗	✓	✓	✓
NTS-NOTEARS	✓	✓	✓	✓	✓

Table 1. Difference between existing methods and NTS-NOTEARS. Starred methods are evaluation baselines.

NTS-NOTEARS MODEL

Temporal CNN Model

We utilize 1D CNNs:

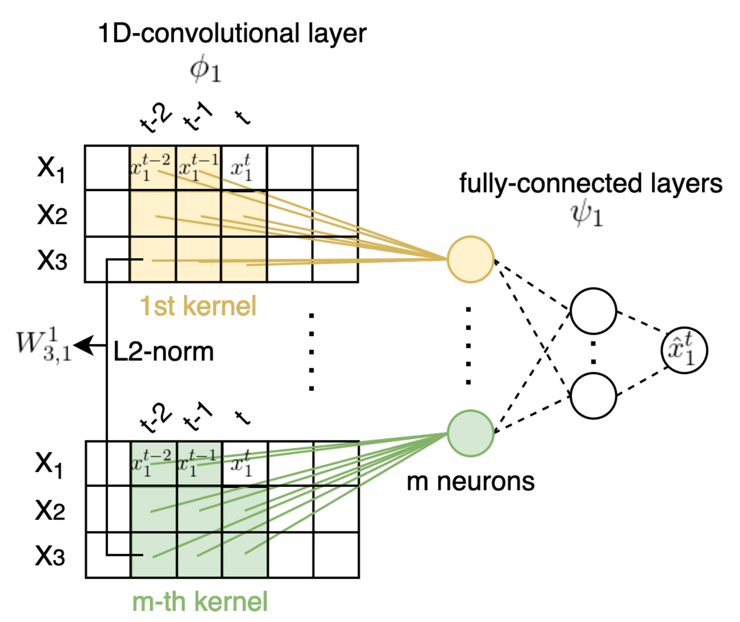
- exploit a sequential or grid topology in the input data. A general MLP does not incorporate data order information.
- Current MLP-based methods concatenate the data. Data concatenation with large datasets may cause memory issues and slow down the training speed.

We train d CNNs jointly where the j-th CNN predicts the expectation of the target variable X_j^t at each time step $t \geq K+1$ given preceding and instantaneous input variables:

$$\mathbb{E}[X_j^t | PA(X_j^t)] = CNN_j(\{\boldsymbol{X}^{t-k} : 1 \leq k \leq K\}, \boldsymbol{X}_{-j}^t)$$

where

- $PA(X_j^t)$ denotes the parents of X_j^t that are defined by the trained CNNs.
- K is a hyperparameter denoting the maximum lag (order).
- The convolutional weights w.r.t. the child variable in the intra-slice t are set to 0.
- For the j-th CNN, the kernel weights are denoted by ϕ_j, the remaining parameters by ψ_j, and $\theta_j = \{\phi_j, \psi_j\}$.



From Local CNNs to Model Weights

- $\phi_{i,j}^k \subset \theta_j$ denotes the m kernel weight parameters for input variable X_i^k in the *first* convolutional layer of the j-th CNN.
- Each entry W_{ij}^k in the weighted adjacency matrix W represents the dependency strength of a directed edge from variable X_i^k to variable X_j^{K+1}.

$$W_{ij}^k = ||\phi_{i,j}^k||_{L^2} \text{ for } k = 1, \ldots, K+1 \tag{1}$$

TRAINING OBJECTIVE

The training objective comprises four components for local functions:

1. Matching the observed child values given the parents.
2. A sparsity penalty for the CNN weights.
3. A regularization term for all parameters.
4. A NOTEARS cyclicity penalty to drive the induced weights to define an acyclic graph.

Let $\mathcal{L}$ denote the least-squares loss, ϕ_j^k be the concatenation of the $\phi_{i,j}^k$ vectors, and $\theta = \{\theta_1, \ldots, \theta_d\}$. The constrained training objective function is defined as:

$$\min_\theta F(\theta)$$
$$\text{subject to } h(W^{K+1}) = 0$$

where

$$F(\theta) = \frac{1}{T-K} \cdot \sum_{t=K+1}^{T} \sum_{j=1}^{d} \mathcal{L}(X_j^t, CNN_{\theta_j}(\{\boldsymbol{X}^{t-k} : 1 \leq k \leq K\}, \boldsymbol{X}_{-j}^t)) + \sum_{k=1}^{K+1} \lambda_1^k \cdot ||\phi_j^k||_{L^1} + \frac{1}{2}\lambda_2 \cdot ||\theta_j||_{L^2}$$

$$h(W^{K+1}) = tr(e^{W^{K+1} \circ W^{K+1}}) - d = 0$$

$tr(A)$ and e^A are the trace and matrix exponential of matrix A, respectively, and $\circ$ is element-wise product. The function h enforces the acyclicity constraint among intra-slice dependencies.

Optimization

We use the L-BFGS-B algorithm to optimize the unconstrained objective.

$$\min_\theta F(\theta) + \frac{\rho}{2} \cdot (h(W^{K+1}))^2 + \alpha \cdot h(W^{K+1}) \tag{2}$$

FROM PRIOR KNOWLEDGE TO OPTIMIZATION CONSTRAINTS

Allowing prior knowledge is often necessary for real-world applications, e.g. forbidden and required edges. Such knowledge can be formalized as constraints on the dependency weights W_{ij}^k:

- b denotes a dependency strength as prior knowledge specified by user
- m is the number of kernels of the convolutional layer of each CNN

Each b is scaled in the following way before being applied to the L-BFGS-B algorithm:

$$\bar{b} = \sqrt{\frac{b^2}{m}} \tag{3}$$

Let $\theta = \{\dot{\theta}, \bar{\theta}\}$ where $\dot{\theta}$ denotes free parameters and $\bar{\theta}$ denotes constrained parameters with lower bounds l and upper bounds u, representing prior knowledge according to Equation (3). The objective function (2) becomes

$$\min_{\dot{\theta}, l_1 \leq \bar{\theta}_1 \leq u_1, l_2 \leq \bar{\theta}_2 \leq u_2, \ldots} F(\theta) + \frac{\rho}{2}(h(W^{K+1}))^2 + \alpha h(W^{K+1})$$

EVALUATION

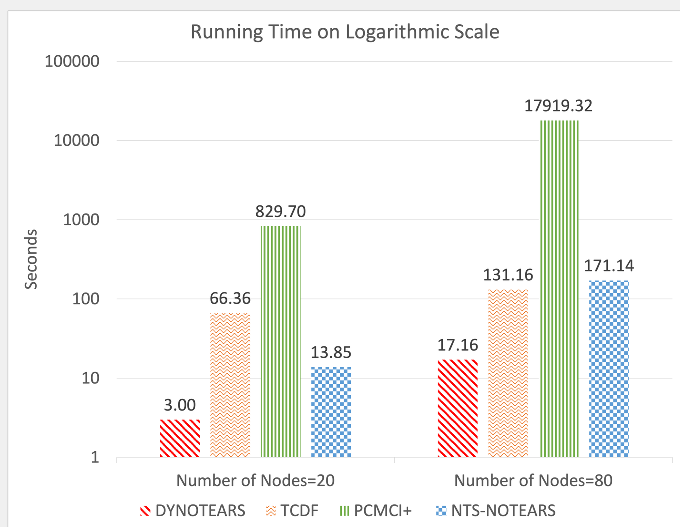
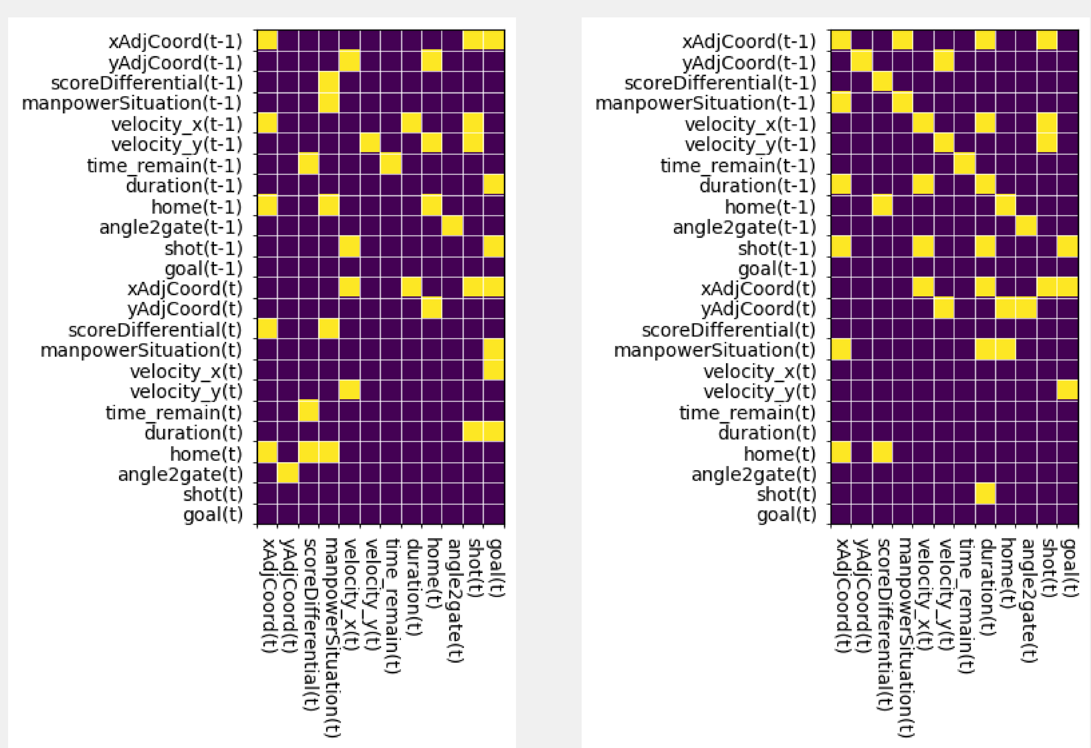


Figure 1. The **average running time** over 10 datasets measured in seconds.

Method	Lorenz 96	fMRI
DYNOTEARS	0.855 (± 0.016)	0.475 (± 0.020)
TCDF	0.459 (± 0.017)	0.347 (± 0.059)
PCMCI+	0.637 (± 0.028)	0.502 (± 0.045)
NTS-NOTEARS	**0.996** (± 0.002)	**0.628** (± 0.023)

Table 2. **Mean F1-scores** (± SE) computed with Lorenz 96 and fMRI benchmarks.



(a) NTS-NOTEARS (b) DYNOTEARS

Figure 2. The DBNs estimated by NTS-NOTEARS and DYNOTEARS with **real-world ice hockey data**.