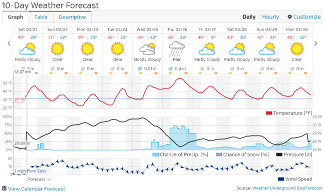
Causal Networks Project Exam DYNOTEARS: Structure Learning from Time-Series Data

Óscar Espitia, Rishabh Upadhyay

December 3, 2021

Introduction 1

Time-series data?
A collection of observations that we measure repeatedly over time



DYNOTEARS is evaluated in different contexts, including stock market data

Introduction 2

Scope

↳ DYNOTEARS is focused on the problem of learning dynamic structures from time-series data.

  ↳ Specifically, it's an approach for learning Bayesian networks

Motivation

↳ BN are graphical models that represent a set of variables and their conditional dependencies via Directed Acyclic graphs (DAGs)

  ↳ Interpretable/explainable

  ↳ Allow us to introduce causal insights about the underlying process

    ↳ In a DAG, the edges provide those clues, e.g. relationship between variables in a system.

Introduction 3

Problem:

↳ learning the edges in a DAG

  ↳ Observations (Dataset)

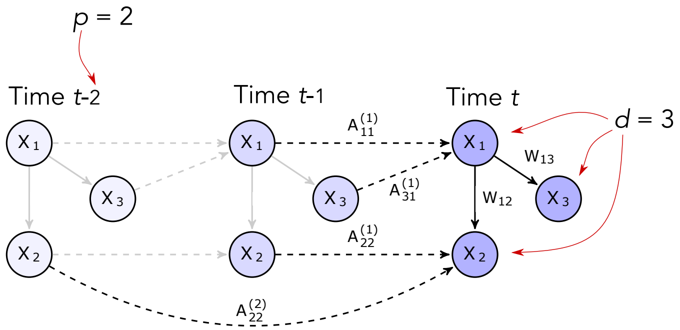
  ↳ Temporal dependencies

Solution:

↳ Learn Dynamic BN

  ↳ model relationships between variables over adjacent time steps.

Dynamic Bayesian Network

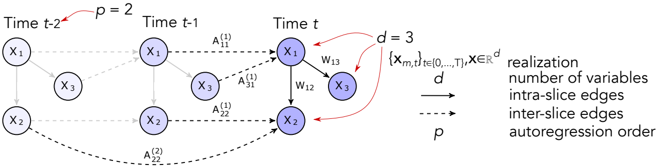


$\{\mathbf{x}_{m,t}\}_{t\in\{0,\ldots,T\}}, \mathbf{x}\in\mathbb{R}^d$

realization of a stationary time series

- d number of variables
- → contemporaneous influence (intra-slice)
- - - → time-lagged influence inter-slice
- p autoregression order

Data modeling 1



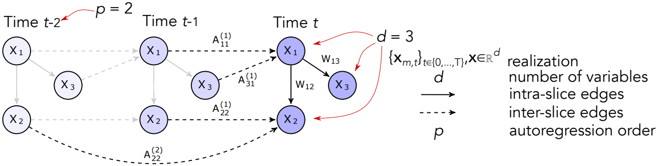
Considering M independent realizations of a stationary time series, the mth time series modeled by the standard Vector Auto-regressive[1] is:

$$\mathbf{x}_{m,t}^{\top} = \mathbf{x}_{m,t}^{\top}\mathbf{W} + \mathbf{x}_{m,t-1}^{\top}\mathbf{A}_1 + \cdots + \mathbf{x}_{m,t-p}^{\top}\mathbf{A}_p + \mathbf{z}_{m,t}^{\top}$$

- $t \in \{p, \cdots, T\}$
- $m \in \{1. \cdots, M\}$
- vector of centered error variables
- Intra-slice and inter-slice edges (weighted matrices with nonzero elements)

[1] Statistical model that captures the relationship and describes the evolution of a set of variables

Data modeling 2

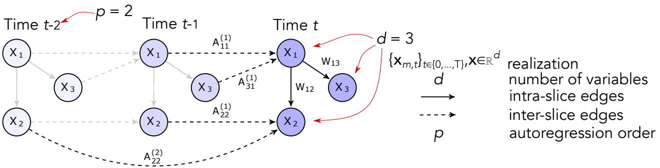


Assuming the network structure as constant across time, the M independent realizations can be modeled as:

$$\mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{Y}_1\mathbf{A}_1 + \cdots + \mathbf{Y}_p\mathbf{A}_p + \mathbf{Z}$$

- $\mathbf{X} \in \mathbb{R}^{n \times d}$ (with $\mathbf{x}_{m,t}^\top$ as rows), $n = M(T + 1 - p)$ (an effective sample size)
- Time lagged versions of $\mathbf{X}$
- matrix of centered error variables
- Intra-slice and inter-slice edges (weighted matrices with nonzero elements)

Data modeling 3

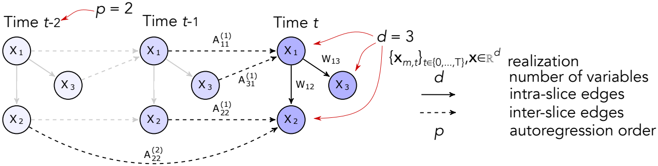


stacking the different $\mathbf{Y}_i$ and $\mathbf{A}_i$, respectively, a compact version of the model is:

$$\mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{Y}\mathbf{A} + \mathbf{Z}$$

- $\mathbf{X} \in \mathbb{R}^{n\times d}$ (with $\mathbf{x}_{m,t}^\top$ as rows), $n = M(T+1-p)$ (an effective sample size)
- Time lagged versions of $\mathbf{X} \to \mathbf{Y} = [\mathbf{Y}_1|\cdots|\mathbf{Y}_p] \in \mathbb{R}^{n\times pd}$
- matrix of centered error variables
- Intra-slice and inter-slice edges (weighted matrices with nonzero elements) $\to \mathbf{A} = [\mathbf{A}_1^\top|\cdots|\mathbf{A}_p^\top]^\top \in \mathbb{R}^{pd\times d}$

Problem statement 1

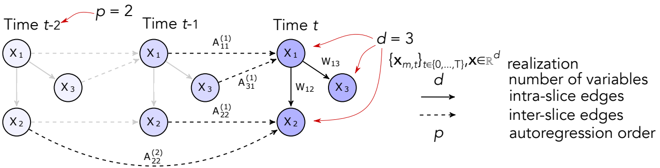


Given the data **X** and **Y** and the (DAGs) matrices **W** and **A**, the problem can be formulated as:

$$\min_{\mathbf{W},\mathbf{A}} \ell(\mathbf{W},\mathbf{A}) \text{ s.t. } \mathbf{W} \text{ is acyclic},$$

$$\ell(\mathbf{W},\mathbf{A}) = \frac{1}{2n}||\mathbf{X} - \mathbf{X}\mathbf{W} - \mathbf{Y}\mathbf{A}||_F^2.$$

Problem statement 2



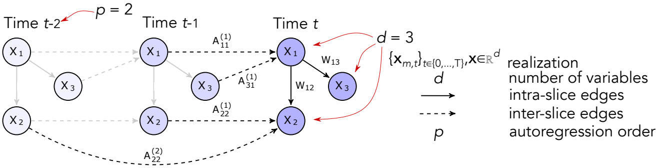
promoting sparsity for **W** and **A**, the problem becomes:

$$\min_{\mathbf{W},\mathbf{A}} f(\mathbf{W},\mathbf{A}) \text{ s.t. } \mathbf{W} \text{ is acyclic},$$

$$f(\mathbf{W},\mathbf{A}) = \ell(\mathbf{W},\mathbf{A}) + \lambda_{\mathbf{W}}||\mathbf{W}||_1 + \lambda_{\mathbf{A}}||\mathbf{A}||_1,$$

with $||\cdot||_1$ stands for the element-wise ℓ_1 norm.

Problem statement 3

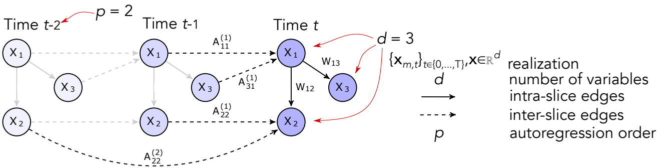


using an equivalent formulation of acyclicity the problem becomes:

$$\min_{\mathbf{W},\mathbf{A}} f(\mathbf{W},\mathbf{A}) \text{ s.t. } h(\mathbf{W}) = 0 \Leftrightarrow \mathbf{W} \text{ is acyclic},$$

$h(\mathbf{W}) = \operatorname{tr} e^{\mathbf{W}\circ\mathbf{W}} - d$ is the trace exponential function and $\circ$ is the Hadamard product.

Problem statement 4



This problem can be solved as a series of unconstrained minimization problems, following the Augmented Lagrangian method:

$$\min_{\mathbf{W},\mathbf{A}} F(\mathbf{W}, \mathbf{A})$$

$$F(\mathbf{W}, \mathbf{A}) = f(\mathbf{W}, \mathbf{A}) + \frac{\rho}{2}(\mathbf{W})^2 + \alpha h(\mathbf{W})$$

The resulting problem can be solved using standard solvers such as L-BFGS-B[2].

[2] an iterative method for solving large scale unconstrained nonlinear optimization problems with limited memory

Experimental Setup

Stimulation of dataset for benchmarks:

- Generating the weighted graphs G_W and G_A.
- Generating data matrices X and Y consistent with these graphs.
- Running all algorithms on X and Y and computing performance metrics.

Baselines:

- NOTEARS and Lasso regression to estimate W and A independently[3].
- SVAR estimation method based on LiNGAM[4].
- tsGFCI[5].

[3] Murphy, K. P. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, University of California, Berkeley

[4] Hyv¨arinen, A., Zhang, K., Shimizu, S., and Hoyer, P. O. (2010). Estimation of a structural vector autoregression model using non-Gaussianity. Journal of Machine Learning Research, 11(May)

[5] Malinsky, D. and Spirtes, P. (2018). Causal structure learning from multivariate time series in settings with unmeasured confounding. In Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery

Stimulated Dataset

Stimulated Dataset:

- Gaussian noise and Exponential.
- 50-500 samples.
- 1 autoregressive term[6].

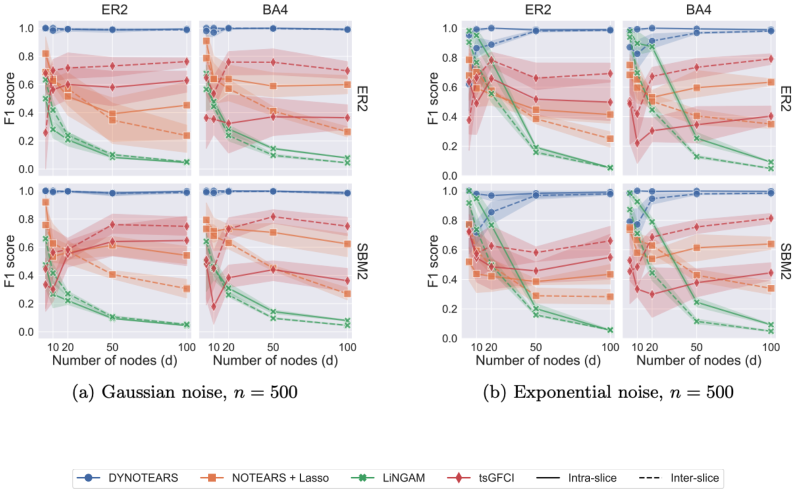
Different combinations are used for Intra-slice and Inter-slice:

- ER2-ER2
- BA4-ER2
- ER2-SBM4
- BA4-SBM4

Barabási–Albert- BA, Stochastic block model- SBM and Erdős–Rényi model- ER

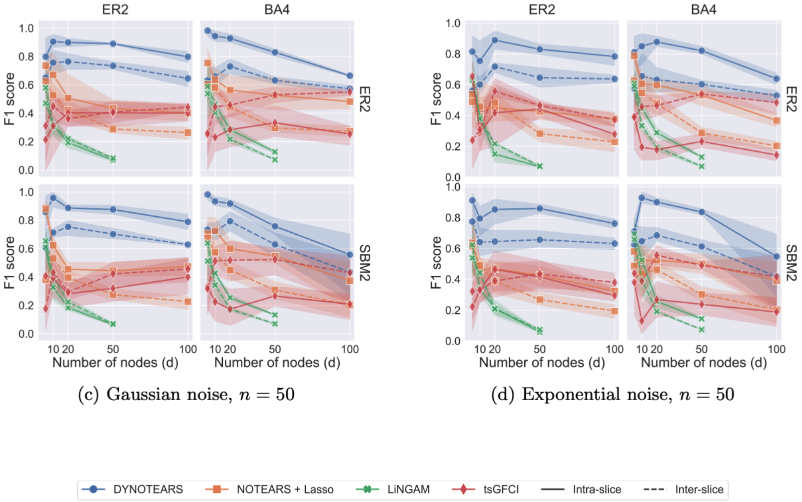
[6] the number of immediately preceding values in the series that are used to predict the value at the present time

Result n=500



(a) Gaussian noise, $n = 500$ (b) Exponential noise, $n = 500$

Result n=50



(c) Gaussian noise, $n = 50$

(d) Exponential noise, $n = 50$

Result's observation

- DYNOTEARS – best-performing algorithm when the number of variables exceeds the number of samples.
- Second-best algorithm is tsGFCI – performance degrade with more edges to the ground-truth graphs.
- LiNGAM is an algorithm designed for non-Gaussian data – performed poor with increase in number of nodes.

Applications

- S&P 100 stock returns: n $=$ 1257 samples (i.e., trading days) and d $=$ 97 variables (i.e., stocks).
- DREAM4 gene expression: gene regulatory networks from gene expression data. 5 independent datasets, each with 10 different time series with 100 variables.

Applications: S&P 100 stock returns

- 400 data points of the series for validation.
- The final graph does not contain inter-slice edges – prediction of future returns is the current return.
- Two stocks influence each other if they belong to the same sector.
- Few stocks also get influenced with other sector – Amazon (AMZN), which is part of the Consumer Cyclical sector, is connected to many of the technology stocks, including Facebook, Netflix, NVIDIA, Google, and Microsoft.

Applications: Dream4 gene expression

- Various DBN models were compared using AUPR and AUCROC.
- DYNOTEARS achieves an average AUROC of 0.664 and an average AUPR of 0.173.
- DYNOTEARS is within one standard deviation of the best performing method.
- Final ranks 4th in AUPR and 8th in AUROC when compared with non DBNs model.

Algorithm	Mean AUPR	Mean AUROC
DYNOTEARS	**0.173**	0.664
G1DBN	0.110	**0.676**
ScanBMA	0.101	0.657
VBSSMb	0.096	0.618
VBSSMa	0.086	0.624
Ebdbnet	0.043	0.643

Conclusion

- Model's the relationships between the variables in a multi-variate time series using a structural vector autoregressive model, where time-invariant structure of the relationships between variables is modelled using DAGs.
- Simplicity in terms of formulating an objective function and in terms of optimizing it.
- It performs well on simulated data across a wide range of parameter choices in the data-generation process.
- Limitations:
  - Behaviour of the algorithm on nonstationary time series data[7].
  - The method was not designed to handle undersampling.
  - Linear data assumption is made purely for simplicity.

[7] have means, variances, and covariances that change over time