# Non-negative Weighted DAG Structure Learning

**Samuel Rey**

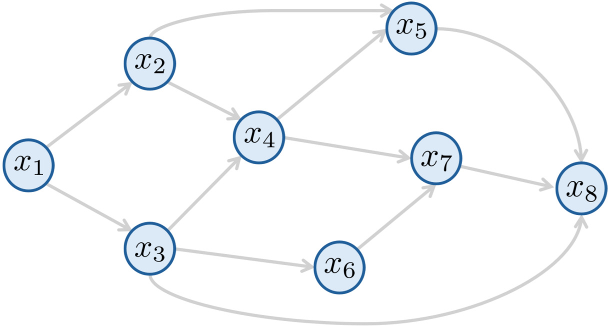
Collaborators: Gonzalo Mateos, S. Saman Saboksayr

King Juan Carlos University
samuel.rey.escudero@urjc.es

March 5, 2026

# The DAG learning problem



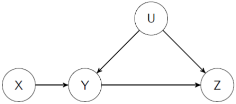



- Consider a set of random variables with unknown dependencies
  - $\Rightarrow$ The underlying structure is a directed acyclic graph (DAG)
  - $\Rightarrow$ Our goal is to learn the DAG structure from the observed data

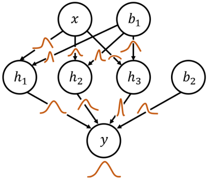
# Why DAGs?



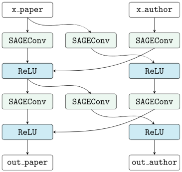
- DAGs have become prominent models in various ML applications
  - ⇒ DAG edges may encode **causal interpretations** [peters17]
  - ⇒ Conditional independencies among variables in Bayesian networks
  - ⇒ Applications: biology [Sachs05], genetics [Zhang13], finance [Sanford12]




Causal inference




Bayesian networks

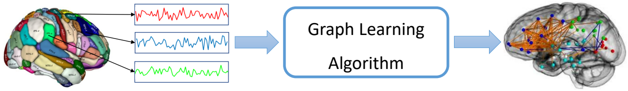



Neural networks

## DAGs vs. Causality

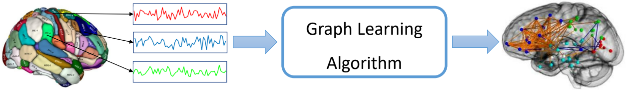
- Causality can always be expressed as a DAG but the converse is not true
  - ⇒ Not every DAG encodes causal relations

- Graph learning aims to infer the graph structure from nodal observations
  - ⇒ Ill-posed problem requires structure (signal model, sparsity...)
  - ⇒ Directionality introduces additional complexity [Marques20]




- Learning the DAG structure comes with **challenges**
  - ⇒ Imposing acyclicity is a combinatorial constraint
  - ⇒ Multiple DAGs may generate the same data distribution

- Graph learning aims to infer the graph structure from nodal observations
  - ⇒ Ill-posed problem requires structure (signal model, sparsity...)
  - ⇒ Directionality introduces additional complexity [Marques20]




- Learning the DAG structure comes with **challenges**
  - ⇒ Imposing acyclicity is a combinatorial constraint
  - ⇒ Multiple DAGs may generate the same data distribution

**Can we impose additional structure to simplify the DAG learning problem?**

# Roadmap

Universidad Rey Juan Carlos

**Background on DAG learning**

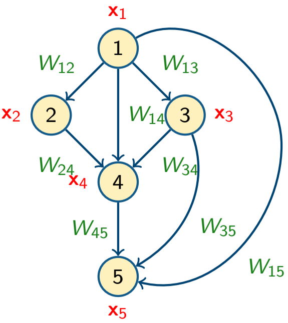
**Non-negative DAG learning**

**Numerical evaluation**

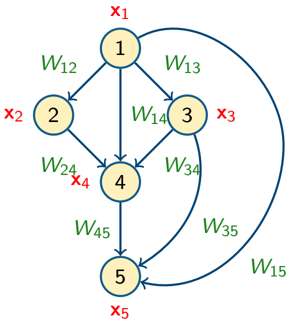
**Concluding remarks**

# DAGs and linear SEM

- DAG $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = d$ nodes
  - $\Rightarrow$ Adjacency matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$
  - $\Rightarrow$ Entry $W_{ij} \neq 0$ indicates a directed link $i \rightarrow j$
  - $\Rightarrow$ $\mathbf{W}$ can be permuted into **upper triangular**
- Define a graph signal $\mathbf{x} \in \mathbb{R}^d$
  - $\Rightarrow$ $\mathcal{D}$ encodes conditional independence on $\mathbf{x}$
  - $\Rightarrow$ $x_i$ depends on parents $PA_i = \{j \in \mathcal{V} : W_{ji} \neq 0\}$



- DAG $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = d$ nodes
  - $\Rightarrow$ Adjacency matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$
  - $\Rightarrow$ Entry $W_{ij} \neq 0$ indicates a directed link $i \rightarrow j$
  - $\Rightarrow$ $\mathbf{W}$ can be permuted into **upper triangular**
- Define a graph signal $\mathbf{x} \in \mathbb{R}^d$
  - $\Rightarrow$ $\mathcal{D}$ encodes conditional independence on $\mathbf{x}$
  - $\Rightarrow$ $x_i$ depends on parents $PA_i = \{j \in \mathcal{V} : W_{ji} \neq 0\}$




- Linear structural equation model (SEM) to generate $\mathbf{X} \in \mathbb{R}^{d \times n}$ consists of

$$x_i = \sum_{j \in PA_i} W_{ji} x_j + z_i \quad \Longrightarrow \quad \mathbf{X} = \mathbf{W}^\top \mathbf{X} + \mathbf{Z}$$

  - $\Rightarrow$ Exogenous input $\mathbf{Z}$ with diagonal covariance matrix
  - $\Rightarrow$ Identifiable for non-Gaussian or homoscedastic Gaussian noise

# Problem statement

Universidad Rey Juan Carlos

- Given the data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ adhering to a linear SEM
- Learn the adjacency matrix $\mathbf{W}$ solving score-minimization problem

$$\min_{\mathbf{W}} \quad F(\mathbf{W}; \mathbf{X}) \ \text{ subject to } \ \mathbf{W} \in \mathbb{D}$$

$\Rightarrow$ With desired score function $F(\mathbf{W}; \mathbf{X})$ and space of DAGs $\mathbb{D}$

- Learning a DAG solely from observational data $\mathbf{X}$ is NP-hard [Chickering96]

  $\Rightarrow$ Combinatorial acyclicity constraint $\mathbf{W} \in \mathbb{D}$ difficult to enforce

- Given the data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ adhering to a linear SEM
- Learn the adjacency matrix $\mathbf{W}$ solving score-minimization problem

$$\min_{\mathbf{W}} \; F(\mathbf{W}; \mathbf{X}) \; \text{subject to} \; \mathbf{W} \in \mathbb{D}$$

$\Rightarrow$ With desired score function $F(\mathbf{W}; \mathbf{X})$ and space of DAGs $\mathbb{D}$

- Learning a DAG solely from observational data $\mathbf{X}$ is NP-hard [Chickering96]
  $\Rightarrow$ Combinatorial acyclicity constraint $\mathbf{W} \in \mathbb{D}$ difficult to enforce

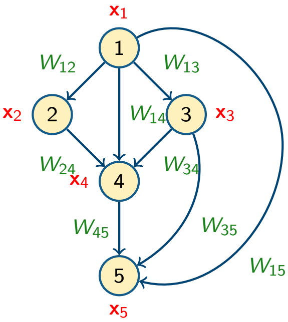
**Discrete optimization**

- Combinatorial search method with likelihood / Bayesian scoring functions
  $\Rightarrow$ Resort to greedy search due to scalability issues [Ramsey17]

Universidad Rey Juan Carlos

- If the causal (partial) order were known
  - $\Rightarrow$ **W** expressed as an upper-triangular matrix

$$\mathbf{W} = \begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} \\ 0 & 0 & 0 & W_{24} & 0 \\ 0 & 0 & 0 & W_{34} & W_{35} \\ 0 & 0 & 0 & 0 & W_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$




- Exploit parameterization $\mathbf{W} \in \mathbb{D} \Leftrightarrow \mathbf{W} = \mathbf{\Pi}^\top \mathbf{U} \mathbf{\Pi}$
  - $\Rightarrow$ **U** is an upper-triangular weight matrix
  - $\Rightarrow$ Permutation matrix $\mathbf{\Pi}$ encodes causal ordering

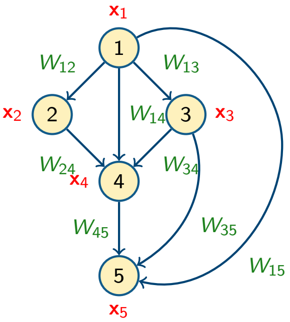
# Order-based methods



- If the causal (partial) order were known

  $\Rightarrow$ $\mathbf{W}$ expressed as an upper-triangular matrix

$$\mathbf{W} = \begin{bmatrix} 0 & W_{12} & W_{13} & W_{14} & W_{15} \\ 0 & 0 & 0 & W_{24} & 0 \\ 0 & 0 & 0 & W_{34} & W_{35} \\ 0 & 0 & 0 & 0 & W_{45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$




- Exploit parameterization $\mathbf{W} \in \mathbb{D} \Leftrightarrow \mathbf{W} = \mathbf{\Pi}^\top \mathbf{U} \mathbf{\Pi}$

  $\Rightarrow$ $\mathbf{U}$ is an upper-triangular weight matrix

  $\Rightarrow$ Permutation matrix $\mathbf{\Pi}$ encodes causal ordering

- Search over exact DAGs in an end-to-end differentiable fashion

  $\Rightarrow$ Learn permutations [Charpentier22], bi-level optimization [Deng23]

- Recovering the causal ordering is challenging with limited data

- NOTEARS characterized acyclicity via smooth function $h(\mathbf{W}) : \mathbb{R}^{d \times d} \to \mathbb{R}$
  - ⇒ The zero level set corresponds to DAGs [Zheng18]

$$h(\mathbf{W}) = 0 \iff \mathbf{W} \in \mathbb{D}$$

  - ⇒ The acyclicity function is $h(\mathbf{W}) = \mathrm{tr}\left(e^{\mathbf{W} \circ \mathbf{W}}\right) - d$

# Continuous acyclicity functions



- NOTEARS characterized acyclicity via smooth function $h(\mathbf{W}) : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$
  - $\Rightarrow$ The zero level set corresponds to DAGs [Zheng18]

$$h(\mathbf{W}) = 0 \iff \mathbf{W} \in \mathbb{D}$$

  - $\Rightarrow$ The acyclicity function is $h(\mathbf{W}) = \mathrm{tr}\left(e^{\mathbf{W} \circ \mathbf{W}}\right) - d$
- Idea: diagonal of $(\mathbf{W} \circ \mathbf{W})^k$ contains information about $k$-length cycles
  - $\Rightarrow$ Term $k!$ can lead to instabilities

$$e^{\mathbf{W}} = \sum_{k=0}^{\infty} \frac{\mathbf{W}^k}{k!} =$$

# Continuous acyclicity functions



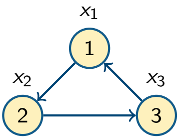
- NOTEARS characterized acyclicity via smooth function $h(\mathbf{W}) : \mathbb{R}^{d \times d} \to \mathbb{R}$
  - $\Rightarrow$ The zero level set corresponds to DAGs [Zheng18]

$$h(\mathbf{W}) = 0 \iff \mathbf{W} \in \mathbb{D}$$

  - $\Rightarrow$ The acyclicity function is $h(\mathbf{W}) = \operatorname{tr}\left(e^{\mathbf{W} \circ \mathbf{W}}\right) - d$

- Idea: diagonal of $(\mathbf{W} \circ \mathbf{W})^k$ contains information about $k$-length cycles
  - $\Rightarrow$ Term $k!$ can lead to instabilities

$$e^{\mathbf{W}} = \sum_{k=0}^{\infty} \frac{\mathbf{W}^k}{k!} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}}_{\text{self-loops}} + \frac{1}{2} \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\text{cycles of size 2}} + \frac{1}{6} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{cycles of size 3}} + \cdots$$



# From discrete to continuous

- Continuous acyclicity constraint offers alternative representation of $\mathbb{D}$
  $\Rightarrow$ From combinatorial search to non-convex continuous optimization

$$\min_{\mathbf{W}} \; F(\mathbf{W};\mathbf{X}) \text{ s. to } \mathbf{W} \in \mathbb{D} \iff \min_{\mathbf{W}} \; F(\mathbf{W};\mathbf{X}) \text{ s. to } h(\mathbf{W}) = 0$$

- Huge **breakthrough** enabling more efficient DAG learning methods!

# From discrete to continuous



- Continuous acyclicity constraint offers alternative representation of $\mathbb{D}$
  $\Rightarrow$ From combinatorial search to non-convex continuous optimization

$$\min_{\mathbf{W}} F(\mathbf{W}; \mathbf{X}) \text{ s. to } \mathbf{W} \in \mathbb{D} \iff \min_{\mathbf{W}} F(\mathbf{W}; \mathbf{X}) \text{ s. to } h(\mathbf{W}) = 0$$

- Huge **breakthrough** enabling more efficient DAG learning methods!
- This discovery propelled the design of different acyclicity functions
  $\Rightarrow$ DAGMA is the state-of-the-art with appealing features [Bello22]

$$h_{dagma}(\mathbf{W}) = d \log(s) - \log\det(s\mathbf{I} - \mathbf{W} \circ \mathbf{W}), \quad s > \rho(\mathbf{W} \circ \mathbf{W})$$

# Current limitations



- The product $\mathbf{W} \circ \mathbf{W}$ is a key component of acyclicity functions
  $\Rightarrow$ Introduces additional non-convexity and important **limitations**
- Every DAG is a stationary point of the acyclicity functions

$$\nabla h_{notears}(\mathbf{W}) = \left(e^{\mathbf{W} \circ \mathbf{W}}\right)^{\top} \circ 2\mathbf{W} \quad \Longrightarrow \quad \nabla h_{notears}(\mathbf{W}) = \mathbf{0} \text{ for all } \mathbf{W} \in \mathbb{D}$$

# Current limitations



- The product $\mathbf{W} \circ \mathbf{W}$ is a key component of acyclicity functions
  - $\Rightarrow$ Introduces additional non-convexity and important **limitations**
- Every DAG is a stationary point of the acyclicity functions

$$\nabla h_{notears}(\mathbf{W}) = \left(e^{\mathbf{W} \circ \mathbf{W}}\right)^{\top} \circ 2\mathbf{W} \quad \Longrightarrow \quad \nabla h_{notears}(\mathbf{W}) = \mathbf{0} \text{ for all } \mathbf{W} \in \mathbb{D}$$

- The KKT condition of the DAG learning problem is

$$\nabla F(\mathbf{W}^{\star}; \mathbf{X}) + \lambda \nabla h(\mathbf{W}^{\star}) = \mathbf{0}$$

  - $\Rightarrow$ The optimal DAG must also be a stationary point of $F(\mathbf{W}; \mathbf{X})$ [Wei20]
  - $\Rightarrow$ If $F$ is convex $\nabla F(\mathbf{W}; \mathbf{X}) = \mathbf{0}$ holds only for minimizers of $F$
  - $\Rightarrow$ Leads to numerical instability and convergence issues

- **Our idea:** assume adjacency matrix $\mathbf{W}$ has **non-negative entries**
  $\Rightarrow$ Remove dependency on $\mathbf{W} \circ \mathbf{W}$ for acyclicity

**Proposition**

For any $\mathbf{W} \in \mathbb{R}_{+}^{d \times d}$ with bounded spectral radius $\rho(\mathbf{W}) < s$, $\mathbf{W} \in \mathbb{D}$ iff

$$h_{ldet}(\mathbf{W}) := d \log(s) - \log\det(s\mathbf{I} - \mathbf{W}) = 0$$

- Highlight of the proof: $-\log\det(\mathbf{I} - \mathbf{W}) = \sum_{k=1}^{\infty} \frac{\mathrm{tr}(\mathbf{W}^k)}{k}$

- **Our idea:** assume adjacency matrix $\mathbf{W}$ has **non-negative entries**
  - $\Rightarrow$ Remove dependency on $\mathbf{W} \circ \mathbf{W}$ for acyclicity

**Proposition**

For any $\mathbf{W} \in \mathbb{R}_+^{d \times d}$ with bounded spectral radius $\rho(\mathbf{W}) < s$, $\mathbf{W} \in \mathbb{D}$ iff

$$h_{ldet}(\mathbf{W}) := d\log(s) - \log\det(s\mathbf{I} - \mathbf{W}) = 0$$

- Highlight of the proof: $-\log\det(\mathbf{I} - \mathbf{W}) = \sum_{k=1}^{\infty} \frac{\mathrm{tr}(\mathbf{W}^k)}{k}$
- DAGs are not stationary points of our acyclicity function

$$\nabla h_{ldet}(\mathbf{W}) = (s\mathbf{I} - \mathbf{W})^{-\top} \quad \Longrightarrow \quad \nabla h_{ldet}(\mathbf{W}) \neq \mathbf{0} \text{ for all } \mathbf{W} \in \mathbb{D}$$

  - $\Rightarrow$ Acyclicity without $\mathbf{W} \circ \mathbf{W}$ holds for other functions as well

# Non-negative DAG learning



- Learning DAGs with non-negative weights amounts to solving

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \quad \frac{1}{2n}\|\mathbf{X} - \mathbf{W}^\top \mathbf{X}\|_F^2 + \alpha \sum_{i,j=1}^{d} W_{ij}$$

$$\text{s. t.} \quad \mathbf{W} \geq 0, \quad h_{ldet}(\mathbf{W}) = 0$$

$\Rightarrow$ Least squares with $\ell_1$ regularization to account for linear SEM

- Amenable optimization landscape in exchange for negative connections

- Learning DAGs with non-negative weights amounts to solving

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \quad \frac{1}{2n}\|\mathbf{X} - \mathbf{W}^\top \mathbf{X}\|_F^2 + \alpha \sum_{i,j=1}^{d} W_{ij}$$
$$\text{s. t.} \quad \mathbf{W} \geq 0, \quad h_{ldet}(\mathbf{W}) = 0$$

$\Rightarrow$ Least squares with $\ell_1$ regularization to account for linear SEM

- Amenable optimization landscape in exchange for negative connections
- To obtain a DAG the acyclicity constraint must be accurately satisfied
- Resort to the method of multipliers to solve the constrained problem
  - $\Rightarrow$ Well-known method with convergence guarantees
  - $\Rightarrow$ Features of our $h_{ldet}(\mathbf{W})$ avoid numerical issues

# Method of multipliers



- Iterative algorithm based on the augmented Lagrangian

$$L_c(\mathbf{W}, \lambda) = F(\mathbf{W}; \mathbf{X}) + \lambda h(\mathbf{W}) + \frac{c}{2} h(\mathbf{W})^2$$

$\Rightarrow$ Lagrange multiplier $\lambda$ and penalty parameter $c$

# Method of multipliers



- Iterative algorithm based on the augmented Lagrangian

$$L_c(\mathbf{W}, \lambda) = F(\mathbf{W}; \mathbf{X}) + \lambda h(\mathbf{W}) + \frac{c}{2} h(\mathbf{W})^2$$

$\Rightarrow$ Lagrange multiplier $\lambda$ and penalty parameter $c$

**DAG learning algorithm**

**Step 1:** Estimate DAG by solving $\mathbf{W}^{(k+1)} = \arg\min_{\mathbf{W} \geq 0} L_{c^{(k)}}(\mathbf{W}, \lambda^{(k)})$

**Step 2:** Update Lagrange multiplier $\lambda^{(k+1)} = \lambda^{(k)} + c^{(k)} h(\mathbf{W}^{(k+1)})$

**Step 3:** Update penalty parameter $c^{(k+1)}$

# Method of multipliers

Universidad Rey Juan Carlos

- Iterative algorithm based on the augmented Lagrangian

$$L_c(\mathbf{W}, \lambda) = F(\mathbf{W}; \mathbf{X}) + \lambda h(\mathbf{W}) + \frac{c}{2} h(\mathbf{W})^2$$

$\Rightarrow$ Lagrange multiplier $\lambda$ and penalty parameter $c$

**DAG learning algorithm**

**Step 1:** Estimate DAG by solving $\mathbf{W}^{(k+1)} = \arg\min_{\mathbf{W} \geq 0} L_{c^{(k)}}(\mathbf{W}, \lambda^{(k)})$

**Step 2:** Update Lagrange multiplier $\lambda^{(k+1)} = \lambda^{(k)} + c^{(k)} h(\mathbf{W}^{(k+1)})$

**Step 3:** Update penalty parameter $c^{(k+1)}$

- Non-convex $h(\mathbf{W}) \Longrightarrow$ Convergence to constrained solution not guaranteed

$\Rightarrow$ In practice we observe recovery of the ground truth!

# Beyond standard SEMs



- Proposed acyclicity function can be readily applied to other scores

**1)** CoLiDE uses score function accounting for noise covariance $\mathbf{\Sigma}$ [Saboksayr24]

$\Rightarrow$ Jointly estimate non-negative $\mathbf{W}$ and $\mathbf{\Sigma}$ for heteroscedastic data

# Beyond standard SEMs

Universidad Rey Juan Carlos

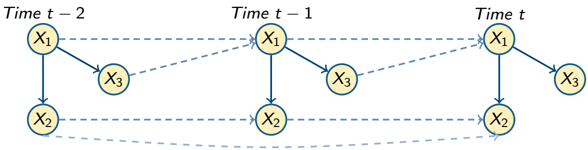
- Proposed acyclicity function can be readily applied to other scores

**1)** CoLiDE uses score function accounting for noise covariance $\mathbf{\Sigma}$ [Saboksayr24]

$\Rightarrow$ Jointly estimate non-negative $\mathbf{W}$ and $\mathbf{\Sigma}$ for heteroscedastic data

**2)** Consider time series signals $\mathbf{x}_t$ adhering to SVARM [Demiralp03]

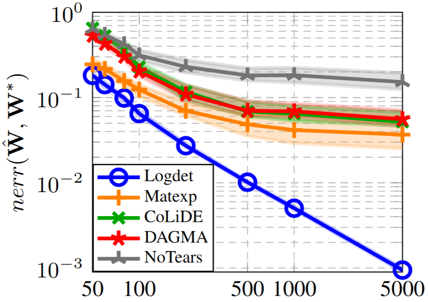
$$\mathbf{x}_t = \mathbf{W}^\top \mathbf{x}_t + \sum_{p=1}^{P} \mathbf{A}_p^\top \mathbf{x}_{t-p} + \mathbf{z}_t$$




$\Rightarrow$ $\mathbf{W}$ and $\mathbf{A}_p$ capture instantaneous and lagged dependencies

$\Rightarrow$ Joint estimation of DAG $\mathbf{W}$ and $\mathbf{A}$ adapting the score function

- Non-negative ER graphs with $d = 100$ nodes and average degree 4
  - $\Rightarrow$ Signals sampled from linear SEM with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$



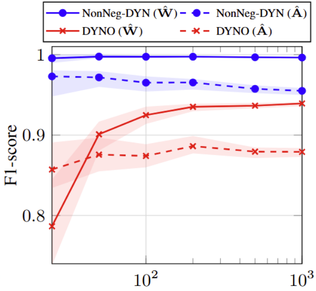
(a) Number of samples $n$

$$nerr(\hat{\mathbf{W}}, \mathbf{W}^*) = \frac{\|\hat{\mathbf{W}} - \mathbf{W}^*\|_F^2}{\|\mathbf{W}^*\|_F^2}$$

- Proposed acyclicity constraints outperform alternatives
  - $\Rightarrow$ The error goes to 0 as number of samples grows

# Synthetic evaluation - Test case 2

- $n = 5000$ signals sampled from a SVARM with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I})$
  - $\Rightarrow$ $P = 2$ time-lagged matrices $\mathbf{A}_p$ with average degree of 1

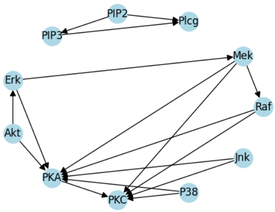



(b) Number of nodes

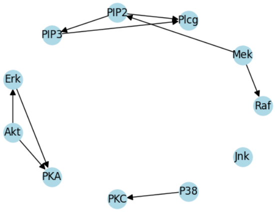
- Proposed acyclicity results in almost perfect Fscore
  - $\Rightarrow$ Better recovery of $\mathbf{W}$ helps in estimation of $\mathbf{A}$
- Methods based on continues acyclicity can manage large DAGs

Universidad Rey Juan Carlos

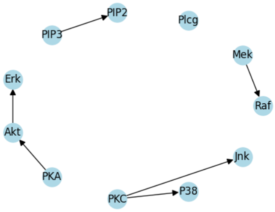
- Sachs dataset contains protein measurements from human system cells
  - ⇒ Comprises 11 nodes, 17 edges and 853 observations
  - ⇒ DAG from experimental methods validated by biological community




Ground Truth




Non-negative
SHD=10, F1=0.61




DAGMA
SHD = 15, F1=0.17

# Conclusions

- DAGs encode causal and dependency relations
  - $\Rightarrow$ Imposing acyclicity to learn the DAG is non-trivial
- Recent development of smooth functions to impose acyclicity
  - $\Rightarrow$ Frames DAG learning as a **continuous optimization** problem
- Assuming **non-negativity** leads to a more tractable problem
  - $\Rightarrow$ Impose acyclicity without the term $\mathbf{W} \circ \mathbf{W}$
  - $\Rightarrow$ Principled constrained optimization method with empirical convergence
  - $\Rightarrow$ Flexible strategy applicable to different score functions
- Ongoing and future work: characterizing convergence

# Questions?




Questions at: **samuel.rey.escudero@urjc.es**