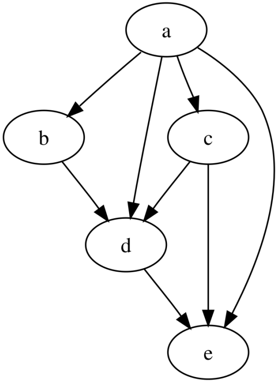
Diferentiable Constraint Function: NOTEARS

Iker Cumplido Esteban

31.01.2025



Outline

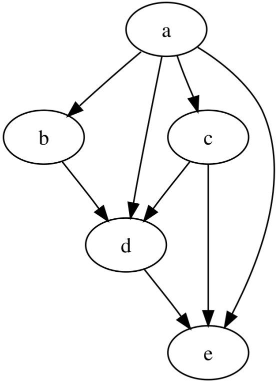
1. Introduction
2. Background
3. Methodology
4. Experiments
5. Conclusion

Outline

1. Introduction
2. Background
3. Methodology
4. Experiments
5. Conclusion

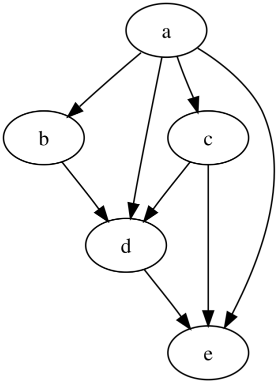
Motivation I

- **Causal Discovery:** Identify cause-effect structure from observational data
- **DAGs:** Represent variables and relationships
- **Challenge:** Ensuring acyclicity involves discrete searches over DAGs
  - NP-hard
  - Grows superexponentially with nodes



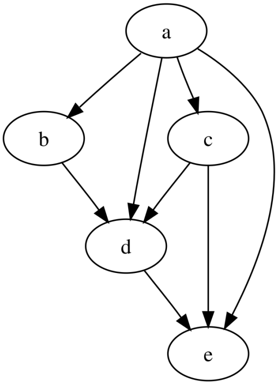
Motivation I

- **Causal Discovery:** Identify cause-effect structure from observational data
- **DAGs:** Represent variables and relationships
- ~~**Challenge:** Ensuring acyclicity involves discrete searches over DAGs~~
  - ~~NP-hard~~
  - ~~Grows superexponentially with nodes~~



Motivation II

- **Causal Discovery:** Identify cause-effect structure from observational data
- **Directed Acyclic Graphs (DAGs):** Represent variables and relationships
- ~~**Challenge:** Ensuring acyclicity usually involves discrete searches over DAGs (NP-hard)~~
  - ~~NP-hard~~
  - ~~Grows superexponentially with nodes~~



NOTEARS ([7])

- **Key Idea:** Convert DAG constraint into a **smooth, differentiable** function
- Allows continuous optimization (no discrete DAG heuristic search)

Outline

1 Introduction

2 Background

3 Methodology

4 Experiments

5 Conclusion

Traditional Approaches

- **Constraint-based (e.g. PC Algorithm):**
  - Uses conditional independence tests
  - Faithfulness assumption
- **Score-based (e.g. GES):**
  - Optimizes a score (like BIC)
  - Superexponential DAG space; often uses greedy search
- **Functional Causal Models (e.g. LiNGAM):**
  - Assumes linear or nonlinear forms
  - Exploit asymmetries in the data to infer causal direction

Traditional Approaches

- **Constraint-based (e.g. PC Algorithm):**
  - Uses conditional independence tests
  - Faithfulness assumption
- **Score-based (e.g. GES):**
  - Optimizes a score (like BIC)
  - Superexponential DAG space; often uses greedy search
- **Functional Causal Models (e.g. LiNGAM):**
  - Assumes linear or nonlinear forms
  - Exploits asymmetries in the data to infer causal direction

NOTEARS

NOTEARS at a Glance

- Assumes a **linear** functional form
- **Continuous** DAG optimization rather than discrete search
- **Smooth & differentiable** acyclicity constraint
- Encourages **sparsity** with regularization

NOTEARS at a Glance

- Assumes a **linear** functional form
- **Continuous** DAG optimization rather than discrete search
- **Smooth & differentiable** acyclicity constraint
- Encourages **sparsity** with regularization

Extensions

- Nonlinear NOTEARS (MLP) [8]
- GOLEM [2]
- DAGMA [1]
- DYNOTEARS [3]

Outline

1. Introduction
2. Background
3. Methodology
4. Experiments
5. Conclusion

Assumptions (1/2)

- **1. Causal Sufficiency (No Latent Confounders) [5]:**
  - No hidden variables that jointly affect observed variables
  - Ensures direct effects are captured correctly

Common Cause Principle (Reichenbach, 1956)

If two random variables X and Y are statistically dependent ($X \not\perp\!\!\!\perp Y$), then there exists a third variable Z that causally influences both.

- Z may coincide with either X or Y as a special case
- Given Z, X and Y become independent: $X \perp\!\!\!\perp Y \mid Z$

Assumptions (1/2)

- 1. Causal Sufficiency (No Latent Confounders) [5]:
  - No hidden variables that jointly affect observed variables
  - Ensures direct effects are captured correctly
- **2. Acyclicity:**
  - True causal graph is a DAG
  - Prohibits feedback loops
- 3. Linearity of Relationships:
  - Original NOTEARS: linear SEM for each variable
  - Later extensions handle nonlinearity

Assumptions (1/2)

- 1. Causal Sufficiency (No Latent Confounders) [5]:
  - No hidden variables that jointly affect observed variables
  - Ensures direct effects are captured correctly
- 2. Acyclicity:
  - True causal graph is a DAG
  - Prohibits feedback loops
- **3. Linearity of Relationships:**
  - Original NOTEARS: linear SEM for each variable
  - Later extensions handle nonlinearity

Assumptions (2/2)

- **4. Identifiability & Additive Noise:**
  - Key Idea: Each variable X_j is influenced by:
    - Its parent variables $\mathrm{Pa}(X_j)$
    - An independent additive noise term z_j, uncorrelated with $\mathrm{Pa}(X_j)$
  - Noise Types and Identifiability of linear additive model:
    - Gaussian Noise: identifiable only for known & equal noise variances [4]
    - Non-Gaussian Noise: fully identifiable
- 5. Large Sample Size:
  - Enough data to reliably estimate structure
- 6. Regularization:
  - ℓ_1-penalty for sparsity
- No Faithfulness Assumption Needed
  - Assumes all conditional independencies are due to the graph structure

Assumptions (2/2)

- 4. Identifiability & Additive Noise:
  - Key Idea: Each variable X_j is influenced by:
    - Its parent variables $\mathrm{Pa}(X_j)$
    - An independent additive noise term z_j, uncorrelated with $\mathrm{Pa}(X_j)$
  - Noise Types and Identifiability of linear additive model:
    - Gaussian Noise: identifiable only for known & equal noise variances [4]
    - Non-Gaussian Noise: fully identifiable
- **5. Large Sample Size:**
  - Enough data to reliably estimate structure
- 6. Regularization:
  - ℓ_1-penalty for sparsity
- No Faithfulness Assumption Needed
  - Assumes all conditional independencies are due to the graph structure

Assumptions (2/2)

- 4. Identifiability & Additive Noise:
  - Key Idea: Each variable X_j is influenced by:
    - Its parent variables $\mathrm{Pa}(X_j)$
    - An independent additive noise term z_j, uncorrelated with $\mathrm{Pa}(X_j)$
  - Noise Types and Identifiability of linear additive model:
    - Gaussian Noise: identifiable only for known & equal noise variances [4]
    - Non-Gaussian Noise: fully identifiable
- 5. Large Sample Size:
  - Enough data to reliably estimate structure
- **6. Regularization:**
  - ℓ_1-penalty for sparsity
- No Faithfulness Assumption Needed
  - Assumes all conditional independencies are due to the graph structure

Assumptions (2/2)

- 4. Identifiability & Additive Noise:
  - Key Idea: Each variable X_j is influenced by:
    - Its parent variables $\mathrm{Pa}(X_j)$
    - An independent additive noise term z_j, uncorrelated with $\mathrm{Pa}(X_j)$
  - Noise Types and Identifiability of linear additive model:
    - Gaussian Noise: identifiable only for known & equal noise variances [4]
    - Non-Gaussian Noise: fully identifiable
- 5. Large Sample Size:
  - Enough data to reliably estimate structure
- 6. Regularization:
  - ℓ_1-penalty for sparsity
- **No Faithfulness Assumption Needed**
  - Assumes all conditional independencies are due to the graph structure

NOTEARS Approach

- **Input:** Data matrix $X \in \mathbb{R}^{n \times d}$
- **Goal:** Learn a weighted adjacency matrix W such that:

$$\min_{W \in \mathbb{R}^{d \times d}} F(W) \quad \text{subject to} \quad G(W) \in \text{DAGs}$$

  - $G(W) \in$ DAGs ensures the learned graph is a DAG
  - Minimizing $F(W)$ ensures good data fit

General Optimization Problem

- **Reformulation:**
  - Original goal (discrete constraint):

$$\min_{W} F(W) \quad \text{s.t.} \quad G(W) \in \mathrm{DAGs}$$

  - Rewritten as (continuous constraint):

$$\min_{W \in \mathbb{R}^{d \times d}} F(W) \quad \text{s.t.} \quad h(W) = 0$$

  - $h(W) = 0 \iff W$ is acyclic
- **Challenge:** Nonconvex optimization $\rightarrow$ Local minima might occur

Key Benefit:

Solving $h(W) = 0$ allows efficient graph learning using standard solvers

SEM & Loss Function (Part 1: Linear SEM)

- **Structural Equation Model (SEM): General Form**

$$X_j = f_j(\mathrm{Pa}(X_j); W) + \epsilon_j$$

  - f_j: Function modeling relationships between $\mathrm{Pa}(X_j)$ and X_j
  - W: Weighted adjacency matrix
  - ϵ_j: Independent noise term uncorrelated with $\mathrm{Pa}(X_j)$

SEM & Loss Function (Part 1: Linear SEM)

- **SEM: General Form**

$$X_j = f_j(\mathrm{Pa}(X_j); W) + \epsilon_j$$

- **For Linear SEM (NOTEARS):**

$$X_j = \sum_{k=1}^{d} W_{kj} X_k + \epsilon_j$$

  - Linear combination of parent variables weighted by W_{kj}
  - Noise ϵ_j captures randomness or unexplained variation

SEM & Loss Function (Part 1: Linear SEM)

- **SEM: General Form**

$$X_j = f_j(\mathrm{Pa}(X_j); W) + \epsilon_j$$

- **For Linear SEM (NOTEARS):**

$$X_j = \sum_{k=1}^{d} W_{kj} X_k + \epsilon_j$$

  - Linear combination of parent variables weighted by W_{kj}
  - Noise ϵ_j captures randomness or unexplained variation

- **When do we have an edge?**
  - $W_{kj} \neq 0$: X_k is a parent of X_j ($X_k \to X_j$)
  - $W_{kj} = 0$: No causal connection between X_k and X_j

SEM & Loss Function (Part 2: Least Squares Loss)

- **Loss Function: Least Squares (LS)**

$$F(W) = \frac{1}{2n} \|X - XW\|_F^2$$

  - How well does W reconstruct X using the SEM?

SEM & Loss Function (Part 2: Least Squares Loss)

- **Loss Function: Least Squares (LS)**

$$F(W) = \frac{1}{2n} \|X - XW\|_F^2$$

  - How well does W reconstruct X using the SEM?
- **When is LS loss not consistent?**
  - Gaussian Noise: only if noise variances are known and equal
  - Small datasets: may lead to suboptimal recovery

SEM & Loss Function (Part 3: Regularization)

- **Sparsity via ℓ_1-regularization:**

$$F_W = \frac{1}{2n}\|X - XW\|_F^2 + \lambda\|W\|_1$$

  - $\lambda > 0$: Controls trade-off between fitting the data and sparsity
  - $\|W\|_1$: Enforces sparsity in the adjacency matrix W
- **Effect of λ:**
  - Smaller $\lambda \Rightarrow$ More edges; better fit but less sparse
  - Larger $\lambda \Rightarrow$ Fewer edges; enforces sparsity but may miss relationships
- **Final Output:** Minimizing F_W yields a sparse DAG suitable for causal inference

Characterization of Acyclicity (Part 1)

- **Theorem [7]:**

$$h(W) = \mathrm{trace}\left(e^{W \odot W}\right) - d$$

  - $h(W) = 0 \iff W$ is a DAG
  - Quantifies the "DAG-ness" of the graph
  - $(W \odot W)$: elementwise square (handles negative weights)

Characterization of Acyclicity (Part 1)

- **Theorem [7]:**

$$h(W) = \mathrm{trace}\left(e^{W \odot W}\right) - d$$

  - $h(W) = 0 \iff W$ is a DAG
  - Quantifies the "DAG-ness" of the graph
  - $(W \odot W)$: elementwise square (handles negative weights)

- **Matrix Exponential:**

$$e^{W \odot W} = \sum_{k=0}^{\infty} \frac{(W \odot W)^k}{k!}$$

  - **Trace of $e^{W \odot W}$:** Measures the sum of weights of all closed walks in the graph
  - For acyclic graphs, there are no closed walks, so $\mathrm{trace}(e^{W \odot W}) = d$

Characterization of Acyclicity (Part 2)

- **Matrix Exponential Trace:**

$$\mathrm{tr}(e^W) = \sum_{k=0}^{\infty} \frac{\mathrm{tr}(W^k)}{k!}$$

  - For a DAG: $\mathrm{tr}(W^k) = 0 \, \forall \, k \geq 1$ (no closed walks)
  - For cyclic graphs: some $\mathrm{tr}(W^k) > 0$, indicating closed paths

Characterization of Acyclicity (Part 2)

- **Matrix Exponential Trace:**

$$\mathrm{tr}(e^W) = \sum_{k=0}^{\infty} \frac{\mathrm{tr}(W^k)}{k!}$$

  - For a DAG: $\mathrm{tr}(W^k) = 0 \; \forall \, k \geq 1$ (no closed walks)
  - For cyclic graphs: some $\mathrm{tr}(W^k) > 0$, indicating closed paths

- **Acyclic Graphs (W is a DAG):**

$$h(W) = \mathrm{tr}(I) - d = 0$$

  - I: Identity matrix with $\mathrm{tr}(I) = d$

Characterization of Acyclicity (Part 2)

- **Matrix Exponential Trace:**

$$\mathrm{tr}(e^W) = \sum_{k=0}^{\infty} \frac{\mathrm{tr}(W^k)}{k!}$$

  - For a DAG: $\mathrm{tr}(W^k) = 0 \, \forall \, k \geq 1$ (no closed walks)
  - For cyclic graphs: some $\mathrm{tr}(W^k) > 0$, indicating closed paths

- **Acyclic Graphs (W is a DAG):**

$$h(W) = \mathrm{tr}(I) - d = 0$$

  - I: Identity matrix with $\mathrm{tr}(I) = d$

- **Cyclic Graphs (W has cycles):**
  - Some power W^k has a nonzero diagonal entry (closed path exists)
  - $\mathrm{tr}(e^W) > d \quad \Rightarrow \quad h(W) > 0$

Acyclic Example: Matrix Definition

$$W^{(\mathrm{A})} = \begin{bmatrix} 0 & 0.5 \\ 0 & 0 \end{bmatrix}$$

- Only $X_1 \to X_2$ edge
- Expect $h(W^{(\mathrm{A})}) = 0$

$$W^{(\mathrm{A})} \circ W^{(\mathrm{A})} = \begin{bmatrix} 0 & 0.25 \\ 0 & 0 \end{bmatrix}$$

- Each entry squared: $0.5^2 = 0.25$
- Denote this as $M^{(\mathrm{A})}$

Acyclic Example: Step 2 (Matrix Exponential)

$$e^{M^{(\mathrm{A})}} \approx I + M^{(\mathrm{A})} + \frac{(M^{(\mathrm{A})})^2}{2!}$$

- $(M^{(\mathrm{A})})^2 = 0$ matrix because second row is zero
- So,

$$e^{M^{(\mathrm{A})}} \approx \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0.25 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix}$$

$$\mathrm{tr}\left(e^{M^{(\mathrm{A})}}\right) = 1 + 1 = 2 \quad h(W^{(\mathrm{A})}) = 2 - d = 2 - 2 = 0$$

Conclusion: $W^{(\mathrm{A})}$ is **acyclic**

Cyclic Example: Matrix Definition

$$W^{(\mathrm{C})} = \begin{bmatrix} 0 & 0.5 \\ 0.3 & 0 \end{bmatrix}$$

- Now $X_1 \to X_2$ and $X_2 \to X_1$ (cycle)
- Expect $h(W^{(\mathrm{C})}) > 0$

$$W^{(\mathrm{C})} \circ W^{(\mathrm{C})} = \begin{bmatrix} 0 & 0.25 \\ 0.09 & 0 \end{bmatrix}$$

- Denote as $M^{(\mathrm{C})}$
- Non-zero entries on both off-diagonal positions

Cyclic Example: Step 2 (Matrix Exponential)

$$e^{M^{(\mathrm{C})}} \approx I + M^{(\mathrm{C})} + \frac{(M^{(\mathrm{C})})^2}{2!} + \frac{(M^{(\mathrm{C})})^3}{3!}$$

- $(M^{(\mathrm{C})})^2$ and $(M^{(\mathrm{C})})^3$ partially non-zero
- This adds positive diagonals

$$\mathrm{tr}\big(e^{M^{(\mathrm{C})}}\big) > 2 \quad \Rightarrow \quad h(W^{(\mathrm{C})}) = \mathrm{tr}\big(e^{M^{(\mathrm{C})}}\big) - d > 0$$

Conclusion: $W^{(\mathrm{C})}$ has a cycle; hence, $h(W^{(\mathrm{C})}) > 0$

Acyclicity Example: Wrap-up

Summary

- Acyclic $W \Rightarrow h(W) = 0$
- Cyclic $W \Rightarrow h(W) > 0$
- The matrix exponential term detects closed walks

Optimization: Augmented Lagrangian

$$\min_{W} F(W) \quad \text{s.t.} \quad h(W) = 0$$

- Introduce a penalty parameter ρ and multiplier α:

$$\mathcal{L}(W, \alpha, \rho) = F(W) + \frac{\rho}{2}\big(h(W)\big)^2 + \alpha\, h(W)$$

- Helps handle $h(W) = 0$ smoothly
- Minimizing $\mathcal{L} \Rightarrow h(W) \approx 0$ plus low $F(W)$

Optimization: Algorithm Steps

1. **Primal Update:**

$$W_{t+1} \leftarrow \arg\min_{W} \mathcal{L}(W, \alpha_t, \rho)$$

2. **Dual Ascent:**

$$\alpha_{t+1} \leftarrow \alpha_t + \rho\, h(W_{t+1})$$

3. Repeat until $h(W_{t+1}) < \epsilon$ (ϵ = tolerance)

Result: A matrix W^* that $\approx$ satisfies $h(W) = 0$

Algorithm: NOTEARS [7]

Pseudocode

Initialize: (W_0, α_0), $c \in (0,1)$, $\epsilon > 0$, $\omega > 0$

For $t = 0, 1, 2, \ldots$ do:

(a) Solve primal: $W_{t+1} \leftarrow \arg\min_W \mathcal{L}^\rho(W, \alpha_t)$

subject to $h(W_{t+1}) < c\, h(W_t)$

(b) Dual ascent: $\alpha_{t+1} \leftarrow \alpha_t + \rho\, h(W_{t+1})$

(c) If $h(W_{t+1}) < \epsilon$: $\widetilde{W}_{\mathrm{ECP}} \leftarrow W_{t+1}$; break

Return: $\widehat{W} := \widetilde{W}_{\mathrm{ECP}} \circ 1\big(|\widetilde{W}_{\mathrm{ECP}}| > \omega\big)$

Thresholding

$$\widehat{A}_{kj} = \begin{cases} 1, & \text{if } |W^*_{kj}| > \delta, \\ 0, & \text{otherwise.} \end{cases}$$

- Converts continuous W^* into a binary adjacency matrix $\widehat{A}$
- δ: hyperparameter controlling which edges remain
- Large $\delta \Rightarrow$ fewer edges, smaller $\delta \Rightarrow$ more edges

Nonlinear Extension (Part 1)

- Original linear NOTEARS: $X_j = \sum_k W_{kj} X_k + \epsilon_j$
- **General SEM:**

$$X_j = f_j(\mathrm{Pa}(X_j); W) + \epsilon_j$$

- Example: MLP-based approach for each variable

Nonlinear Extension (Part 2)

- **Adjacency matrix from partial derivatives:**

$$W(f) \leftarrow \frac{\partial f_j}{\partial X_i}$$

  - Captures how each input influences an output in a neural network
- Maintains $h(W(f)) = 0$ for DAG constraint
- More flexible

Method Summary

- NOTEARS introduces a **smooth acyclicity function** $h(W)$
- Discrete DAG search $\rightarrow$ **continuous optimization** problem
- Uses **augmented Lagrangian** approach to enforce $h(W) = 0$
- ℓ_1-regularization for sparsity + thresholding for final DAG
- Extensions to **nonlinear** relationships

Outline

1. Introduction
2. Background
3. Methodology
4. Experiments
5. Conclusion

Experiment 1: Varsortability

Motivation:

- **Varsortability:**
  - How well does causal order align with order of increasing marginal variances
  - Variance of a variable X_j depends on:
    - Variances of its parent variables
    - Weights of causal edges ($w_{i \to j}$)
    - Noise term (z_j)

Experiment 1: Varsortability

Motivation:

- **Varsortability:**
  - How well does causal order align with order of increasing marginal variances
  - Variance of a variable X_j depends on:
    - Variances of its parent variables
    - Weights of causal edges ($w_{i \to j}$)
    - Noise term (z_j)
- **Issue with High Varsortability:**
  - Variance accumulation downstream $\implies$ synthetic data easily learned
  - MSE-based methods may exploit variance structure

Experiment 1: Varsortability

Motivation:

- **Varsortability:**
  - How well does causal order align with order of increasing marginal variances
  - Variance of a variable X_j depends on:
    - Variances of its parent variables
    - Weights of causal edges ($w_{i \to j}$)
    - Noise term (z_j)
- **Issue with High Varsortability:**
  - Variance accumulation downstream $\Longrightarrow$ synthetic data easily learned
  - MSE-based methods may exploit variance structure
- **Goal of Experiment:**
  - Test if causal direction inference breaks under specific noise variance conditions

Experiment 1 Setup (Part 1)

- **Simple 2-variable model**: $A \to B$.

$$A = N_A, \quad B = wA + N_B$$

- Compare MSE of $B \sim A$ vs. $A \sim B$
- **Key Condition**:

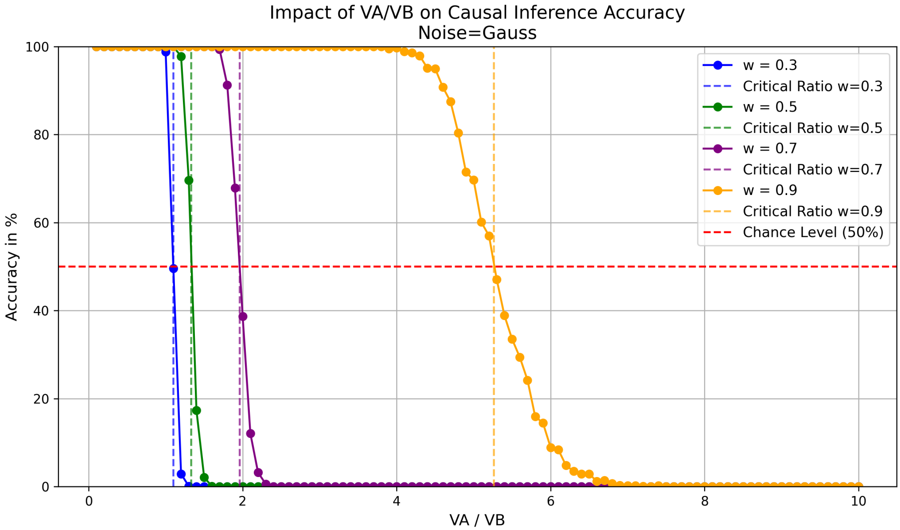
$$(1 - w^2)\, V_A < V_B \quad \Longrightarrow \quad \text{correct direction}$$

Experiment 1 Setup (Part 2)

We vary:

- **Causal Weight:** $w \in \{0.3, 0.5, 0.7, 0.9\}$
- **Noise Ratio:** $V_A/V_B \in [0.1, \ldots, 10]$
- **Noise Distributions:** Gaussian, exponential, Gumbel
- **Sample Size:** $n = 2000$ (1000 repetitions)
- **Standardization:** Compare standardized vs. non-standardized data

Experiment 1.1: Impact of Ratio



Experiment 1.1 Interpretation

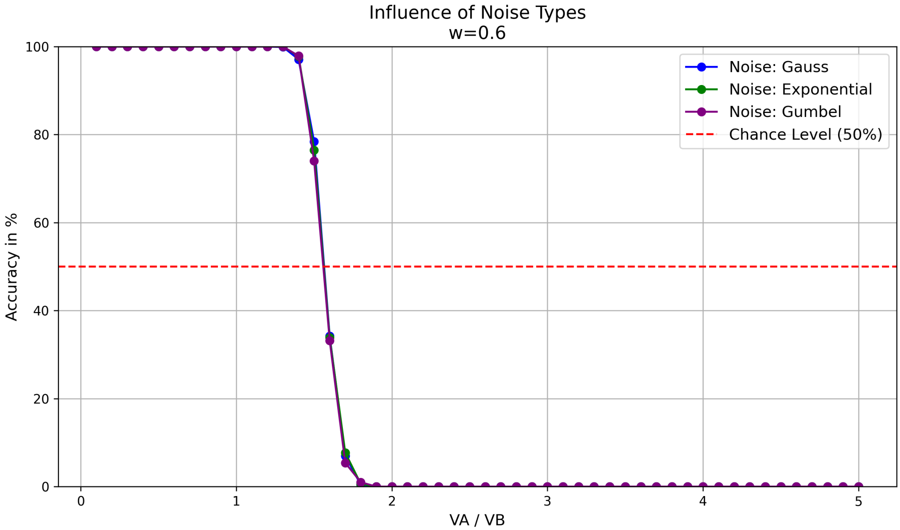
Interpretation:

- Validates [6]:

$$(1 - w^2)V_A V_B \implies \text{Inference fails}$$

- MSE-based methods depend heavily on variance ratios, making them vulnerable to high varsortability

Experiment 1.2: Noise Distributions

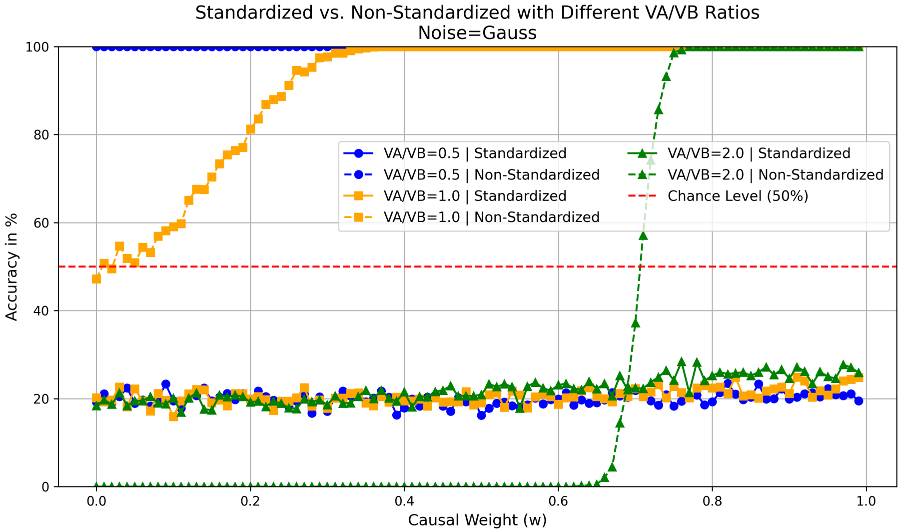


Experiment 1.2 Interpretation

Interpretation:

- The limitation extends beyond Gaussian noise
- Root issue lies in the reliance on variance-based direction inference, not the noise distribution itself

Experiment 1.3: Standardization



Experiment 1.3 Interpretation

Interpretation:

- Standardization **removes the scale information** that is critical for inferring causal direction $\Longrightarrow$ MSE-based direction inference relies heavily on the variance structure

Experiment 2 Overview

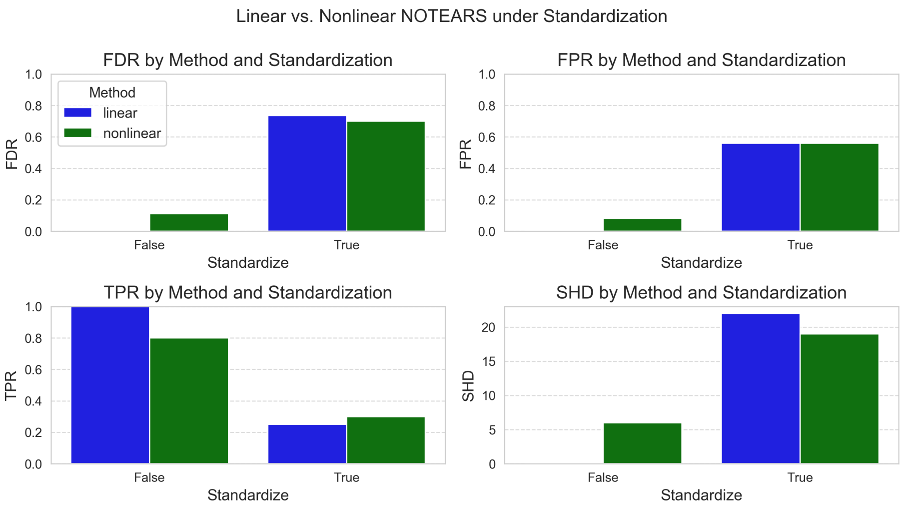
Motivation:

- Compare *linear* vs. *nonlinear* NOTEARS under standardized/non-standardized data
- Investigate how different data-generating mechanisms (linear vs. nonlinear) affect performance
- Assess hyperparameters: λ (regularization) and threshold for adjacency

Experiment 2 Setup

- **Data Generation:**
  - $d = 10$ variables, Erdős-Rényi random graph with $s_0 = 20$ edges
  - $n = 1000$ samples, Gaussian noise (scale = 1.0)
- **Metrics:**
  - False Discovery Rate (FDR)
  - False Positive Rate (FPR)
  - True Positive Rate (TPR)
  - Structural Hamming Distance (SHD)

Experiment 2.1: Linear vs Nonlinear

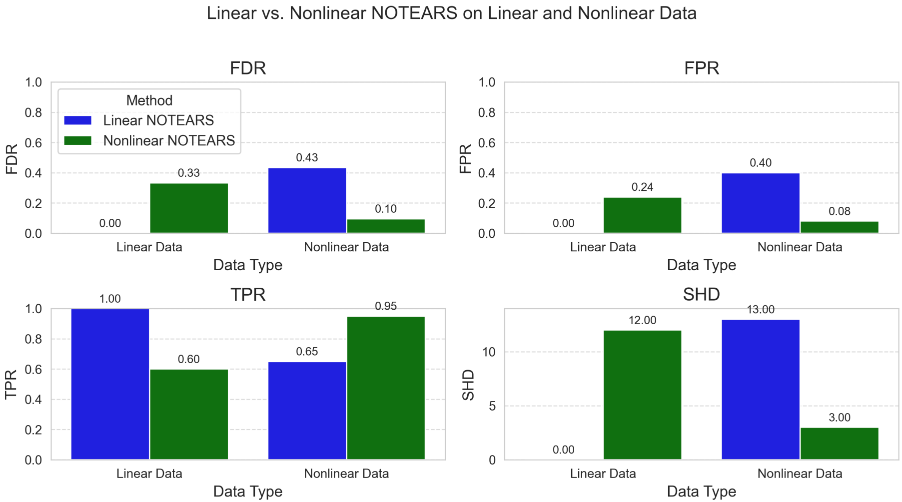


Experiment 2.1: Interpretation

Interpretation:

- Scale information is crucial for NOTEARS, and standardization removes it
- Nonlinear extension doesn't solve the standardization issue

Experiment 2.2: Linear vs Nonlinear on Different Data

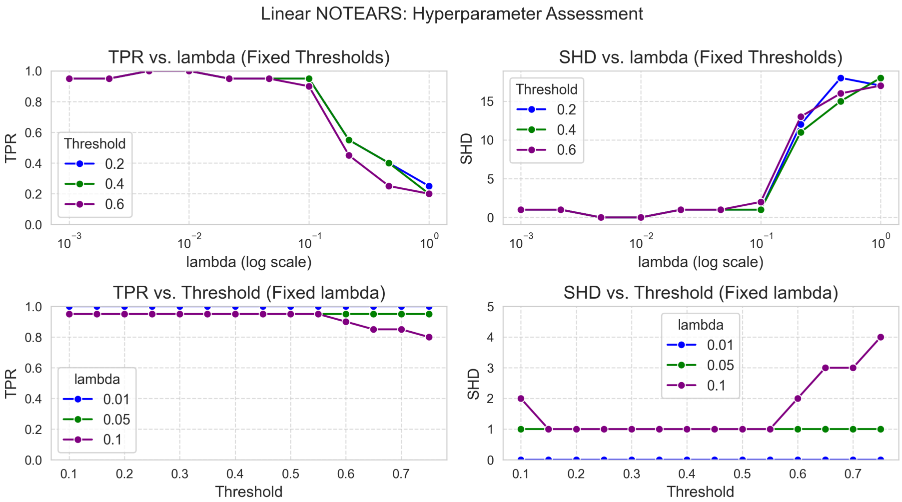


Experiment 2.2: Interpretation

Interpretation:

- If data linear, nonlinear model can overfit or add false edges
- If data nonlinear, linear NOTEARS systematically misrepresents dependencies

Experiment 2.3: Hyperparameter Assessment



Experiment 2.3: Interpretation

Interpretation:

- Must tune λ carefully: moderate value to avoid over- or under-sparsity
- δ can still matter if noise or scale is different

Outline

1 Introduction

2 Background

3 Methodology

4 Experiments

5 Conclusion

Conclusion

Continuous Approach

- NOTEARS uses a **smooth function** $h(W) = 0$ to enforce acyclicity
- Avoids discrete DAG search (NP-hard)

Conclusion

Continuous Approach

- NOTEARS uses a **smooth function** $h(W) = 0$ to enforce acyclicity
- Avoids discrete DAG search (NP-hard)

Varsortability Issue

- High varsortability artificially boosts performance in synthetic data
- Standardization can remove scale info, affecting accuracy

Conclusion

Continuous Approach

- NOTEARS uses a **smooth function** $h(W) = 0$ to enforce acyclicity
- Avoids discrete DAG search (NP-hard)

Linear vs. Nonlinear

- Linear NOTEARS best if data truly linear
- Nonlinear version handles more complex dependencies

Varsortability Issue

- High varsortability artificially boosts performance in synthetic data
- Standardization can remove scale info, affecting accuracy

Conclusion

Continuous Approach

- NOTEARS uses a **smooth function** $h(W) = 0$ to enforce acyclicity
- Avoids discrete DAG search (NP-hard)

Linear vs. Nonlinear

- Linear NOTEARS best if data truly linear
- Nonlinear version handles more complex dependencies

Varsortability Issue

- High varsortability artificially boosts performance in synthetic data
- Standardization can remove scale info, affecting accuracy

Limitations

- Unknown noise variances (usual in real-world data)
- Local minima in nonconvex optimization

References I

- Kevin Bello, Bryon Aragam, and Pradeep Ravikumar.
  Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization.
  Advances in Neural Information Processing Systems, 35:8226–8239, 2022.

- Ignavier Ng, AmirEmad Ghassami, and Kun Zhang.
  On the role of sparsity and dag constraints for learning linear dags.
  Advances in Neural Information Processing Systems, 33:17943–17954, 2020.

- Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam.
  Dynotears: Structure learning from time-series data.
  In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. Pmlr, 2020.

References II

Jonas Peters and Peter Bühlmann.
Identifiability of gaussian structural equation models with equal error variances.
Biometrika, 101(1):219–228, 2014.

Hans Reichenbach.
The Direction of Time.
Dover Publications, Mineola, N.Y., 1956.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald.
Beware of the simulated dag! causal discovery benchmarks may be easy to game.
Advances in Neural Information Processing Systems, 34:27772–27784, 2021.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing.
Dags with no tears: Continuous optimization for structure learning.
Advances in neural information processing systems, 31, 2018.

References III

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing.
Learning sparse nonparametric dags.
In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. Pmlr, 2020.