Directed Acyclic Graph Structure Estimation using Sparse Bayesian Learning.

Tanvi Nayak

Indian Institute of Technology Dharwad
Mathematics and Computing

Problem Statement

- Recover the weighted adjacency matrix of a Directed Acyclic Graph (DAG) from observed data under a Gaussian SEM model.
- Enforce the acyclicity constraint while estimating dependencies among variables.
- Achieve a sparse and interpretable graph structure that suppresses spurious edges and matches real-world network sparsity.
- Integrate Sparse Bayesian Learning (SBL) to adaptively drive unnecessary edges to zero and improve structural recovery.
- Combine SBL-driven sparsity with the NOTEARS acyclicity formulation into a single optimization framework.

What is a DAG?

- A directed acyclic graph (DAG) is a directed graph with no directed cycles.
- It can be represented by a matrix whose nonzero entries indicate directed edges.

- In the DAG learning problem, we observe a data matrix $X \in \mathbb{R}^{n \times d}$ containing n i.i.d. samples of $X = (X_1, \ldots, X_d)$.
- Let D be the discrete space of all DAGs $G = (V, E)$ on d nodes.
- The task is to find a DAG $G \in D$ that represents the joint distribution $P(X)$.

Example: A Small DAG

- Consider a graph with three nodes: X_1, X_2, and X_3.
- Suppose the directed edges are:

$$X_1 \to X_2, \qquad X_2 \to X_3.$$

- This forms a simple chain with no cycles.

Weighted Adjacency Matrix:

$$W = \begin{pmatrix} 0 & 1.5 & 0 \\ 0 & 0 & -0.8 \\ 0 & 0 & 0 \end{pmatrix}$$

Structural Equation Models (SEMs)

- We represent a directed graph using a weighted matrix $W = [\, w_1 \mid w_2 \mid \cdots \mid w_d \,] \in \mathbb{R}^{d \times d}$.
- Each column w_j contains the weights of all incoming edges to node X_j.
- Let $X = (X_1, \ldots, X_d)$ be the random vector of variables.
- A linear SEM defines each variable as:

$$X_j = w_j^\top X + z_j,$$

  where z_j is an independent noise term.
- In matrix form, stacking all variables and noise terms:

$$X = XW + Z,$$

  where $Z = (z_1, \ldots, z_d)$.

Example: SEM for a Simple DAG

DAG Structure

$$X_1 \;\rightarrow\; X_2 \;\rightarrow\; X_3$$

Weighted Adjacency Matrix

$$W = \begin{pmatrix} 0 & 1.5 & 0 \\ 0 & 0 & -0.8 \\ 0 & 0 & 0 \end{pmatrix}$$

Interpretation

- 1.5 : edge $X_1 \rightarrow X_2$
- -0.8 : edge $X_2 \rightarrow X_3$
- Column w_j lists parents of X_j

Corresponding SEM Equations

$$\begin{aligned} X_1 &= z_1, \\ X_2 &= 1.5\,X_1 + z_2, \\ X_3 &= -0.8\,X_2 + z_3. \end{aligned}$$

General Form

$$X_j = w_j^\top X + z_j$$

- Each variable is a linear function of its parents.
- z_j represents independent noise.

NOTEARS: Acyclicity Constraint

Key Idea (Zheng et al., 2018): A DAG can be characterized using a smooth, differentiable function.

- For a weighted adjacency matrix $W \in \mathbb{R}^{d \times d}$, define

$$h(W) = \mathrm{tr}\left(e^{W \circ W}\right) - d.$$

- **Theorem:** W represents a DAG if and only if $h(W) = 0$.
- The gradient is simple and fully differentiable:

$$\nabla h(W) = \left(e^{W \circ W}\right)^T \circ (2W).$$

NOTEARS: Acyclicity Constraint

- The hadamard product $W \circ W$ removes signs and keeps only squared edge weights.
- The matrix exponential $e^{W \circ W}$ expands into a power series containing $(W \circ W)^k$ terms.
- Powers of an adjacency matrix count directed walks of length k.
- If a cycle exists, some power contributes a positive value to the diagonal, so $\mathrm{tr}(e^{W \circ W}) > d$.
- If no cycles exist, all diagonal terms remain exactly 1, giving $\mathrm{tr}(e^{W \circ W}) = d$.

SEM Likelihood: Single Variable Formulation

- For each variable x_k in the SEM:

$$x_k = Xw_k + e_k, \qquad e_k \sim \mathcal{N}(0, \sigma_k^2 I).$$

- The likelihood of a single column is:

$$P(x_k \mid W, \sigma_k^2) = (2\pi\sigma_k^2)^{-N/2} \exp\left(-\frac{1}{2\sigma_k^2}\|x_k - Xw_k\|_2^2\right).$$

Total Likelihood of the data

- Noise terms $\{e_k\}$ are independent across variables.
- Full likelihood factorizes into column-wise components:

$$P(X \mid W, \sigma^2) = \prod_{k=1}^{d} P(x_k \mid W, \sigma_k^2).$$

Sparse Bayesian Learning: Prior Structure

- SBL imposes sparsity via a hierarchical prior:

$$W_{jk} \sim \mathcal{N}(0, \alpha_{jk}^{-1}), \qquad \alpha_{jk} \sim \mathrm{Gamma}(a, b).$$

- Large α_{jk} shrink W_{jk} toward zero (automatic sparsity).

Full Joint Posterior: Likelihood + Priors

- Combining SEM likelihood with SBL priors:

$$\begin{aligned}\log P(X, W, \alpha) \;=\; & \log P(X \mid W) \\ & + \sum_{j,k} \log P(W_{jk} \mid \alpha_{jk}) \\ & + \sum_{j,k} \log P(\alpha_{jk}).\end{aligned}$$

- Maximizing this posterior produces adaptive sparsity.

Final Optimization Objective: SEM + SBL + NOTEARS

- Objective function:

$$J(W, \Lambda) = \frac{1}{2N} \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \|x_k - Xw_k\|_2^2 + \frac{1}{2} \sum_{j,k} \Lambda_{jk} W_{jk}^2 + \lambda\, h(W),$$

  where

$$h(W) = \mathrm{tr}(e^{W \circ W}) - d.$$

Update Rules: Posterior Statistics and Precision (SBL Step)

- **Posterior covariance for column w_k:**

$$\Sigma_{w_k} = \left(\frac{1}{\sigma_k^2} X^\top X + \mathrm{diag}(\Lambda_k) \right)^{-1}.$$

- **Posterior mean:**

$$\mu_{w_k} = \frac{1}{\sigma_k^2} \Sigma_{w_k} X^\top x_k.$$

- **Precision update (SBL):**

$$\Lambda_{jk}^{\text{new}} = \frac{1 - \Lambda_{jk} (\Sigma_{w_k})_{jj}}{(\mu_{w_k})_j^2}.$$

Update Rules: W-Step and Acyclicity Enforcement (ALM)

- **W-update (Augmented Lagrangian):**

$$W^{+} = \arg\min_{W} \left[\frac{1}{2N} \sum_{k} \frac{1}{\sigma_k^2} \|x_k - Xw_k\|_2^2 + \frac{1}{2} \sum_{j,k} \Lambda_{jk} W_{jk}^2 \right. \\ \left. + \ \alpha \, h(W) + \frac{\rho}{2} h(W)^2 \right].$$

- **Lagrange multiplier update:**

$$\alpha \leftarrow \alpha + \rho \, h(W).$$

- **Penalty update (optional):**

$$\rho \leftarrow c\rho, \qquad c > 1.$$

Proposed Algorithm

Algorithm 1

Input: Data matrix $\mathbf{X}$, initial weights $\mathbf{W}^{(0)}$, regularization parameter λ, initial precision matrix $\mathbf{\Lambda}^{(0)}$, noise variances $\boldsymbol{\sigma}^2$, gamma parameters a , b
Initialize: Lagrange multiplier α_0, penalty parameter ρ_0
for $t = 0, 1, 2, \dots$ until convergence **do**
  (1) Update SBL hyperparameters:
  for $k = 1$ to d **do**
    Compute posterior mean $\boldsymbol{\mu}_{\mathbf{w}_k}^{(t)}$ and covariance $\mathbf{\Sigma}_{\mathbf{w}_k}^{(t)}$
    for $j = 1$ to d **do**
      Update precision: $\Lambda_{jk}^{(t+1)} = \frac{1-\Lambda_{jk}^{(t)}(\mathbf{\Sigma}_{\mathbf{w}_k}^{(t)})_{jj}}{(\boldsymbol{\mu}_{\mathbf{w}_k}^{(t)})_j^2}$
    end for
  end for
  (2) Optimize W via Augmented Lagrangian:
   Initialize: $\mathbf{W}^{(0)}$, α_0, ρ_0
  for $m = 0, 1, 2, \dots$ until inner convergence **do**
    Update weights:

$$\mathbf{W}^{(m+1)} = \arg\min_{\mathbf{W}} \mathcal{L}_{\text{aug}}(\mathbf{W}, \alpha_m, \rho_m)$$

    Update Lagrange multiplier:

$$\alpha_{m+1} = \alpha_m + \rho_m h(\mathbf{W}^{(m+1)})$$

  end for
  Set $\mathbf{W}^{(t+1)} = \mathbf{W}^{(m+1)}$
end for
Thresholding: Apply element-wise thresholding to the final $\mathbf{W}$:

$$W_{jk} \leftarrow \begin{cases} W_{jk} & \text{if } |W_{jk}| \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

Output: Final pruned DAG weight matrix $\mathbf{W}$

Evaluation Metrics and Comparison Setup

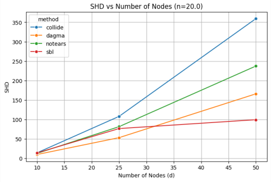
We consider three key metrics for causal graph recovery:

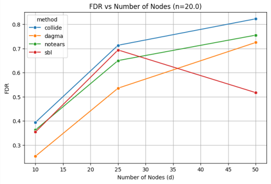
- **SHD** – Measures the number of edge additions, deletions, or flips needed to transform the estimated graph into the ground truth.
- **FDR** – Fraction of discovered edges that are incorrect.
- **Edge Intersection** – Number of correctly recovered edges.

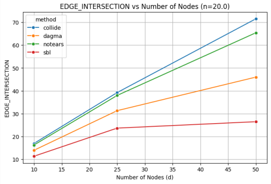
We compare our method (**SBL**) against three baselines: **NOTEARS**, **DAGMA**, and **CoLiDE**.

Note: In all figures that follow, the method CoLiDE is inadvertently labeled as "COLLIDE"; all such instances should be interpreted as referring to CoLiDE.

Nodes vs SHD, FDR, Edge Intersection (Samples = 20)

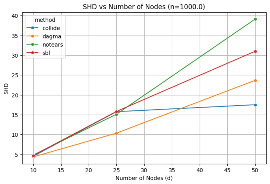


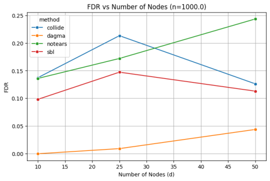


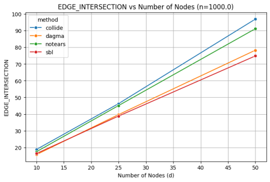


- SBL does not give rise to high SHD or high FDR compared to the other methods at low sample size.
- Consequently, its edge intersection remains on the lower side but follows a stable trend as the number of nodes increases.

Nodes vs SHD, FDR, Edge Intersection (Samples = 1000)

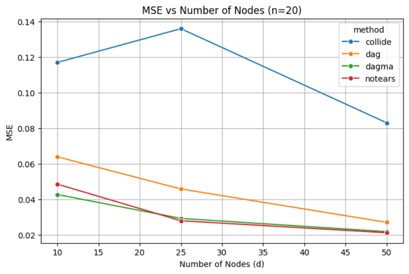


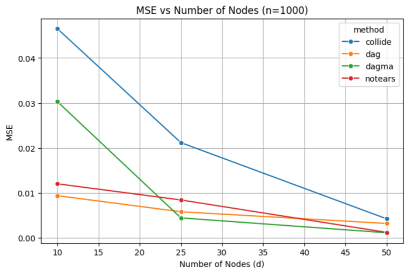




- SBL follows the same overall trend, with SHD and FDR remaining reasonably controlled across node sizes.
- Its edge intersection performance is also quite solid, even if it is not the best among the methods.

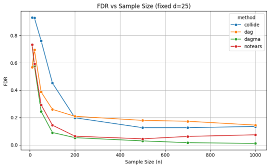
Nodes vs MSE

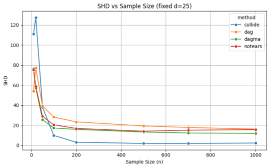


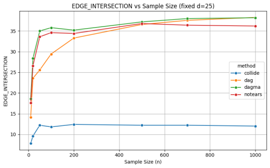


- The MSE is defined as $\|W - \widehat{W}\|_2^2$.
- SBL (labelled as dag in the plot) shows low MSE, not always the lowest but consistently good enough across node sizes.

Sample Size vs SHD, FDR, Edge Intersection





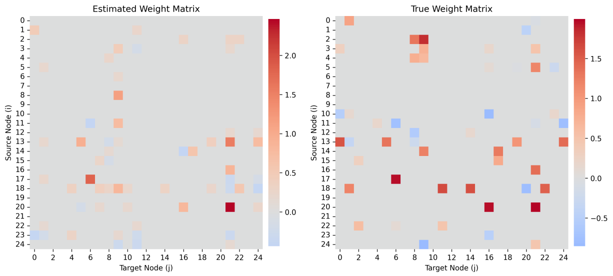


- At low sample sizes, SBL (labelled as `DAG`) shows lower FDR and SHD compared to the other methods.
- As the sample size increases, its FDR and SHD also rise, but the edge intersection remains reasonably high.

.

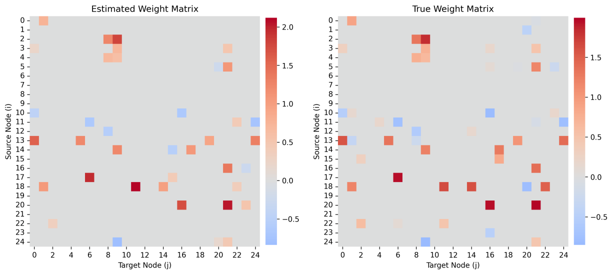
Effect of Sample Size

Nodes = 25, Edges = 50, Samples = 20



Effect of Sample Size

Nodes = 25, Edges = 50, Samples = 1000



Conclusion

- We combined sparse Bayesian learning with the NOTEARS acyclicity formulation to develop a DAG learning method.
- The precision updates promoted sparsity, while the acyclicity term ensured a valid graph structure.
- The results were satisfactory: SHD and FDR remained controlled and edge recovery was reasonably accurate.
- Overall, the approach produced stable estimates and a balanced tradeoff between sparsity and structural correctness.

Future Work

- Improving edge intersection while keeping FDR and SHD within acceptable bounds remains an important direction.
- The pruning threshold is currently selected through trial-and-error; establishing a systematic or theoretically motivated thresholding rule would strengthen the approach.
- Enhancing robustness to noise is essential for applying the method effectively to real-world datasets.

References

- X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing, *DAGs with NO TEARS: Continuous Optimization for Structure Learning*, 2018.
- M. E. Tipping, *Sparse Bayesian Learning and the Relevance Vector Machine*, 2001.
- S. S. Saboksayr and G. Mateos, *COLIDE: Concomitant Linear DAG Estimation*.
- K. Bello, B. Aragam, and P. Ravikumar, *DAGMA: Learning DAGs via M-matrices and a Log-Determinant Acyclicity Characterization*.