DAGs with NO TEARS

Continuous Optimization for Structure Learning

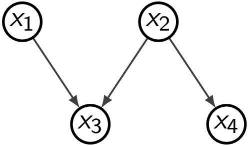
Xun Zheng Bryon Aragam Pradeep Ravikumar Eric Xing

Machine Learning Department
Carnegie Mellon University

November 28, 2018

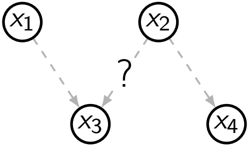
Background

- Graphical models: compact models of $p(x_1, \ldots, x_d)$



$$\xrightarrow{\textit{sample}} \quad \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 4.00 & -1.14 & 0.20 & -2.37 \\ -1.05 & 0.35 & -0.66 & -0.39 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{array}$$

- Structure learning: what graph fits the data best?



$$\xleftarrow{\textit{estimate}} \quad \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 4.00 & -1.14 & 0.20 & -2.37 \\ -1.05 & 0.35 & -0.66 & -0.39 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \end{array}$$

Structure Learning: Where Are We?

	MNs	BNs	Comments
constraint-based	✓	✓	need faithfulness
score-based, local search	✓	✓	combinatorial opt.

Structure Learning: Where Are We?

	MNs	BNs	Comments
constraint-based	✓	✓	need faithfulness
score-based, local search	✓	✓	combinatorial opt.
score-based, global search	✓ †	?*	continuous opt.

†Breakthough in Markov Networks:

- Huge success of methods like graphical lasso
- Widely applied in various fields, *e.g.* bioinformatics

*Challenges in Bayesian Networks:

- Directed graph $\rightarrow$ asymmetric matrix
- Acyclic graph $\rightarrow$ combinatorial constraint

Structure Learning: Where Are We?

	MNs	BNs	Comments
constraint-based	✓	✓	need faithfulness
score-based, local search	✓	✓	combinatorial opt.
score-based, global search	✓[†]	this work	continuous opt.

[†]Breakthough in Markov Networks:

- Huge success of methods like graphical lasso
- Widely applied in various fields, *e.g.* bioinformatics

*Challenges in Bayesian Networks:

- Directed graph $\rightarrow$ asymmetric matrix
- Acyclic graph $\rightarrow$ combinatorial constraint

tl;dr

$$\max_{G} \; score(G) \quad \text{s.t. } G \in DAG \qquad \Longleftrightarrow \qquad \max_{W} \; score(W) \quad \text{s.t. } h(W) \leq 0$$

(combinatorial 😱) (smooth 😎)

tl;dr

$$\max_G \ score(G) \quad \text{s.t. } G \in DAG \qquad \iff \qquad \max_W \ score(W) \quad \text{s.t. } h(W) \leq 0$$

(combinatorial 😱) (smooth 😎)

Smooth Characterization of DAG

Such function exists: $h(W) = \mathrm{tr}(e^{W \circ W}) - d$.
Moreover, simple gradient: $\nabla h(W) = (e^{W \circ W})^T \circ 2W$.

NO TEARS 🚫💧

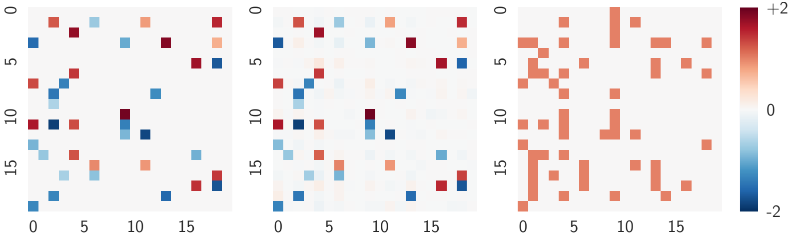
Available at: github.com/xunzheng/notears

```python
def notears_simple(X, max_iter=100, h_tol=1e-8, w_threshold=0.3):
    n, d = X.shape
    w_est, w_new = np.zeros(d * d), np.zeros(d * d)
    rho, alpha, h, h_new = 1.0, 0.0, np.inf, np.inf
    bnds = [(0, 0) if i == j else (None, None) for i in range(d) for j in range(d)]
    for _ in range(max_iter):
        while rho < 1e+20:
            sol = sopt.minimize(_func, w_est, method='L-BFGS-B', jac=_grad, bounds=bnds)
            w_new = sol.x
            h_new = _h(w_new)
            if h_new > 0.25 * h:
                rho *= 10
            else:
                break
        w_est, h = w_new, h_new
        alpha += rho * h
        if h <= h_tol:
            break
    w_est[np.abs(w_est) < w_threshold] = 0
    return w_est.reshape([d, d])
```

30 lines (function, gradient) + 20 lines (optimize) $\approx$ 50 lines

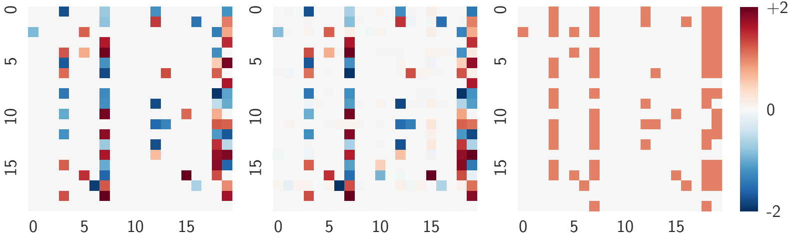
Existing algorithms: $\gg$ 1000 lines

Results: Recovering Erdos-Renyi Graph



ground truth this work baseline (FGS)

Results: Recovering Scale-free Graph



ground truth | this work | baseline (FGS)

- A smooth characterization of DAG:

$$h(W) = \mathrm{tr}(e^{W \circ W}) - d \leq 0 \iff G(W) \in DAG$$

- Use existing solvers for constrained optimization problem:

$$\begin{aligned} \max_{W} \quad & score(W) \\ s.t. \quad & h(W) \leq 0 \end{aligned}$$

- Bridge <u>optimization</u> and <u>structure learning</u>