Cauchy Graphical Models

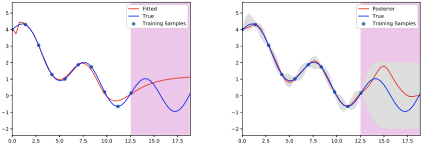
T. Muvunza

Tsinghua-Berkeley Shenzhen Institute

October 18, 2023

Bayesian vs Neural Networks

1. **Causal Inference**: How does drinking 10% more water in the morning reduce aging?
2. **Explainability**: "Doctor, here is my neural net & its 97% accurate!"
3. **Parsimonious**: Imagine a data set with 0 data points and a prior.



(a) (b)

Figure 1: (a) Regression output using NN with 2 hidden layers, (b) Regression output using Gaussian process framework; *Source:* Goan and Fookes (2020).

Challenges in Bayesian Networks

Main Challenges (Opportunities?)

1. Model & Computational Complexity
2. Application (BNN, BCNN, BGCN, etc)
3. Modeling Assumptions & Approximation

Gaussian Assumptions:

- Symmetry: distribution is centred around the mean, μ
- Homoskedastic variance, σ^2

Note:

- Does not model heavy tails
- Does not model asymmetry

$\rightarrow$ BN that apply Gaussian assumptions may lead to suboptimal performance

Research Goal

- A common approach to **BN** involves learning linear regression-based Graphical Models.

However:

- Distribution of microarray intensities show a clear skew (Friedman et al., 2000, 2004; Ben-Dor et al., 2000)
- Financial data is heavy tailed (Muvunza, 2020, 2021)
- Wind Power Forecasting Error is leptokurtic (WPFE) (Hodge and Milligan, 2011);
- Image denoising, (Achim and Kuruoglu, 2005; Rabbani et al., 2006)
- Neural Network parameters (Simsekli et al, 2019)

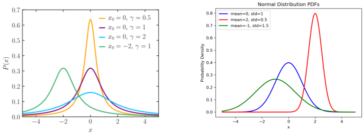
$\rightarrow$ We aim to model **BN** as **Directed-Acyclic Cauchy Graphs (DAGs)**

Why Cauchy?

- Has closed-form solutions
- Has heavy tails
- Is highly skewed
- Is parameterized by scale x_0, and location γ parameters

Challenge:

- Mean & Variance are unknown
- Moments do not exist



(a) Cauchy dist. (b) Gaussian dist.

Figure 2: Gaussian process does not best describe heavy tailed data

Contributions

Our contributions are as follows:

1. We propose novel **Cauchy Graphical Models** (**CGLearn**), a new class of multivariate Cauchy densities that can be represented as Directed-Acyclic Graphs **(DAGs)** with arbitrary network topologies.
2. We conduct extensive experiments on synthetic and real world data & the results demonstrate the efficacy of our approach.
3. We propose Cauchy-based GCN to overcome the lack of generalization and expressiveness inherent in popular techniques used in structural learning.

1. Background and Motivations
2. Cauchy Graphical Models
3. Methodologies
4. Empirical Validation
5. Future Works

Bayesian Network Models

Suppose we have the following **DAG** network:

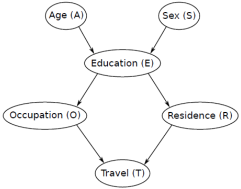


Figure 3: DAG representing dependencies of variables in a network

$$P(A, S, E, O, R, T) = P(A)P(S)P(E|A, S)P(O|E)P(R|E)P(T|O, R)$$

$$P_B(\mathcal{X}) = \prod_{i=1}^{|\mathcal{X}|} p(\texttt{Child}, X_i \mid \texttt{Parents}, P_a(X_i))$$

Cauchy density

The most common parameterization for stable distribution is defined by Samorodnitsky and Taqqu (1994): A random variable X is $S(\alpha, \beta, \gamma, \delta)$ if it has characteristic function:

$$E(\exp^{itX}) = \begin{cases} \exp\left(-\gamma^{\alpha}|t|^{\alpha}\left[1 - i\beta(\tan\frac{\pi\alpha}{2})(\text{sign}t)\right] + i\delta t\right) & \text{if } \alpha \neq 1 \\ \exp\left(-\gamma|t|\left[1 + i\beta\frac{2}{\pi}(\text{sign}t)\ln|t|\right] + i\delta t\right) & \text{if } \alpha = 1 \end{cases}$$

The parameter α is the index of stability and sign$t = 1$ if $t > 0$, 0 if $t = 0$ and -1 if $t < 0$.
$\rightarrow$Cauchy density is derived when $\alpha = 1$ and $\beta = 0$:

$$\begin{aligned} \phi_X(t) &= \exp(-\gamma|t|[1+0] + i\delta t) \\ \phi_X(t) &= \exp(-\gamma|t|[1+0]) \\ \phi_X(t) &= \exp(-\gamma|t|) \end{aligned}$$

Cauchy Density

Given $\phi_X(t)$ as the characteristic function of Cauchy, we can obtain the Fourier Transform as follows:

$$F(x) = \mathcal{F}(\phi_X(t))$$

$$\begin{aligned} F(x) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itX} f(t)dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itX}\phi_X(t)dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itX}e^{-\gamma|t|}dt \\ &= \frac{1}{2\pi}\int_{-\infty}^{0} e^{-itX}e^{\gamma t}dt + \frac{1}{2\pi}\int_{0}^{\infty} e^{-itX}e^{-\gamma t} \\ &= \frac{1}{2\pi}\int_{-\infty}^{0} e^{(\gamma - iX)t}dt + \frac{1}{2\pi}\int_{0}^{\infty} e^{-(\gamma + iX)t}dt \end{aligned}$$

Cauchy Density

$$= \frac{1}{2\pi}\left[\left[\frac{e^{(\gamma - iX)t}}{\gamma - iX}\right]_{-\infty}^{0} - \left[\frac{e^{-(\gamma + iX)t}}{\gamma + iX}\right]_{0}^{\infty}\right]$$

$$= \frac{1}{2\pi}\left[\frac{1}{\gamma - iX} + \frac{1}{\gamma + iX}\right]$$

$$= \frac{1}{2\pi}\left[\frac{2\gamma}{\gamma^2 + x^2}\right]$$

$$\frac{1}{\pi}\left[\frac{\gamma}{\gamma^2 + x^2}\right]$$

The $F(x)$ of $\phi_X(t)$ shown above is the density of the Cauchy distribution. $\square$

Problem Formulation

More formally:

- Given a joint distribution of a finite set of RV $\mathcal{X} = \{X_1, X_2, ..., X_N\}$,
- We define a **BN** $B(G, \Theta)$ consisting of the DAG, & a set of parameters $\Theta = \{\theta_i \mid X_i \in \mathcal{X}\}$, that determine the conditional probability distribution $p(X_i \mid P_a(X_i), \theta)$ for $X_i \in \mathcal{X}$ given the state of its parents $P_a(X_i) \subseteq \mathcal{X} \setminus \{X_i\}$ in G.
- DAG G represents the **factorization** of joint probability density of RV into terms representing each variable X_i and its parents $P_a(X_i)$ such that:

$$P_B(\mathcal{X}) = \prod_{i=1}^{|\mathcal{X}|} p(X_i \mid P_a(X_i), \theta)$$

Problem Formulation

Factorization of joint pdf of RV

$$P_B(\mathcal{X}) = \prod_{i=1}^{|\mathcal{X}|} p(X_i \mid P_a(X_i), \theta)$$

- The dependence of $p(X_i \mid P_a(X_i), \theta)$ on θ_i is usually specified by an appropriately chosen family of parameterized probability densities such as Gaussian.
- Our goal is to use multivariate Cauchy densities to model the RV in $\mathcal{X}$.

Cauchy Graphical Models

Bayesian Networks constructed from Cauchy densities

A Cauchy Graphical Model is a probability distribution over $\mathcal{X}$ such that:

1. $Z_j = X_j - \sum_{X_k \in P_a(X_j)} w_{jk} X_k \sim \text{Cauchy}(\gamma, \delta) \equiv S(1, 0, \gamma, \delta)$
2. Z_j is independent of Z_k if $Z_j \neq Z_k, \forall X_j \in \mathcal{X}$

- where $P_a(X_j) \subseteq \mathcal{X} \setminus \{X_j\}$ are parent nodes of X_j in the DAG G and Θ describes the distribution of the parameters.
- $w_{jk} \in \mathbb{R}, W_j = \{w_{jk} \mid X_k \in P_a(X_j)\}$
- $\theta_j = \{\alpha, \beta_j, \gamma_j, \delta_j\} \cup W_j, \Theta = \{\theta_i \mid X_i \in \mathcal{X}\}$

Given the above conditions, we note that $B(G, \Theta)$ is a Bayesian Network. The transformation matrix from X_i to Z_j is also a BN.

Learning Cauchy Graphical Models

Structure Learning

- Goal for Structure Learning of a BN is to determine the optimal topology that best mirrors the dependencies between RV.
- Score-based Algorithms explore the search space of the *DAG* to maximize a given score. The most common method is Bayesian Information Criterion (**BIC**), Schwarz (1978).
- Given a data set $D = \{D_1, ..., D_N\}$, the $S_{BIC}(B|D)$ for a BN $B(G, \Theta)$ is defined as:

$$S_{BIC}(B|D) = \sum_{D_j \in D} \log[P_B(D_j)] - \sum_{X_i \in \mathcal{X}} \frac{|P_a(X_i)|}{2} \log N$$

1. $P_B(D_j)$ is the marginal likelihood estimator.
2. $\sum_{X_i \in \mathcal{X}} \frac{|P_a(X_i)|}{2} \log N$ is the penalty term.

Learning Cauchy Graphical Models

Structure Learning

- Misra and Kuruoglu (1998, 2016) proposed Minimum Dispersion Criteria *MDC*, which is more efficient than *BIC*.
- MDC selects the Bayesian Network that maximises the score S_{MDC} over the space of all DAG G, and Θ parameters. Formally, the score is defined as:

$$S_{MDC}(B|D) = -\sum_{X_i \in \mathcal{X}} \left\{ N \frac{\log \gamma_i}{\alpha} + \frac{|P_a(X_i)|}{2} \log N \right\}$$

- $S_{MDC} \equiv S_{BIC}$ under specific settings for symmetric α-stable densities, (Misra and Kuruoglu, 2016).

Learning Cauchy Graphical Models

Parameter Learning, γ

- Goal of Parameter Learning in BN is to determine each conditional distribution for a given network.
- Given a Cauchy density, γ denotes the dispersion parameter.
- Finding the conditional distribution is a non-trivial task since the moments do not exist.

$\rightarrow$ Why γ?

1. For Structure Learning, $S_{MDC}(B|D)$
2. To characterize linear dependencies among RV.

Learning Cauchy Graphical Models

Parameter Learning, γ

- Samorodnitsky (1996), Kuruoglu (1998, 2001) showed that **If:**

$$Z \sim S(\alpha, 0, \gamma, 0) \equiv \text{Cauchy}(\gamma, 0)$$

then

$$\mathbb{E}(|Z|^p) = C(p, \alpha)\gamma^{p/\alpha}, -1 < p < \alpha$$

- The l_p of a Cauchy RV is related to it's p-th moment.
- Minimizing $\gamma \equiv$ minimizing p-th order moment.

$$\text{argmin}\, \frac{1}{\alpha} \log \gamma_j \equiv \text{argmin} \|Z_j\|_p \equiv (\sum_{\lambda=1}^{N} |Z_{j,\lambda}|^p)^{1/p} \forall -1 < p < \alpha$$

$$W_j^* = \text{argmin} \log \big(\|Z_j\|_p\big) \equiv \text{argmin} \log \Big((\sum_{\lambda=1}^{N} |Z_{j,\lambda}|^p)^{1/p} \Big)$$

Learning Cauchy Graphical Models

Structure Learning Algorithms

1. **Algorithm 1**: IRLS to minimize l_p norm and obtain regression coefficients.
2. **Algorithm 2**: K2Search, we use a modified version of hill-climbing method to learn the *DAG* consistent with an ordering, σ.
3. **Algorithm 3**: Ordering-Based Search (OBS), we use OBS to search for a local optimum in the space of all *DAGs*
4. **Algorithm 4**: **CGLearn**, a full algorithm for learning the structure and parameters of Cauchy Graphical Models.

Baseline

OLS-based BIC baseline

- For structure learning, we choose OLS-based BIC which is used to learn Gaussian Graphical models.
- We define BIC penalized log-likelihood as $S_{OLS}(B|D)$ as:

$$S_{BIC}(B \mid D) = \sum_{D_j \in D} \log[P_B(D_j)] - \sum_{X_i \in \mathcal{X}} \frac{|P_a(X_i)|}{2} \log N$$

Synthetic data

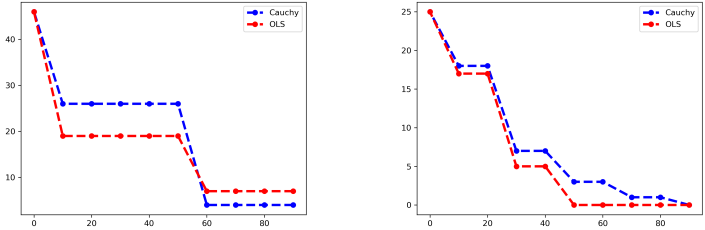
ALARM and CHILD Networks

1. We fix the network topology of ALARM (37, 46) and CHILD (20,25) networks.
2. We simulate data using α-Stable process $S(\alpha = 1, \beta = 0.9, \gamma = 1, \delta = 0)$
3. In our results, we report True and False Positives, Mean Regression Coefficients, Variance of Regression Coefficients and $\log \gamma$.

- **ALARM** is a Bayesian network designed to provide an alarm message system for patient monitoring, (Beinlich et al, 1989)
- The aim of the **CHILD** network is to provide clinical experts with a mechanism to diagnose the type of disease that a child has, (Spiegelhalter and Cowell, 1992)

ALARM Network

- **True Positives** are the number of bootstrap replicates where each true positive edge was found for structure learning.



(a) ALARM network (b) CHILD network

Figure 4: Overall (a) **CGLearn** (192), **OLS** (169) correct edges; (b) **CGLearn** (83), **OLS** (69) correct edges.

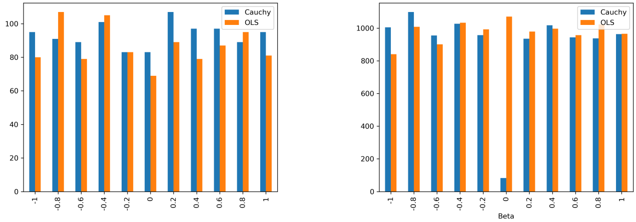
CHILD Network

Varying β

- We fix CHILD network topology and vary β from [-1,1] in steps of 0.2 while fixing $\alpha = 1, \& \gamma = 1$.
- Varying β would allow us to determine how the algorithm performs in symmetrizing the data and learning complex problems.
- Our results determine the sensitivity of the models to changes in the skewness of the data.

CHILD Network

True & False Positives: Varying β



(a) TP Bar plot

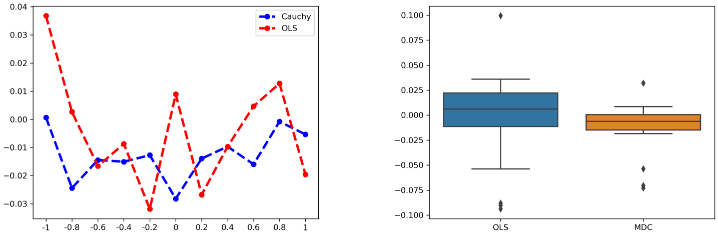
(b) FP Bar plot

Figure 5: (a)Overall, our results show that **CGLearn** (1027 edges) performs better than **OLS** (954 edges); (b) **CGLearn** (9921) performs better than **OLS** (10 766 edges)

CHILD Network

Mean Regression Coefficients

- It is the bias in mean regression-coefficient of each edge for True Positives.



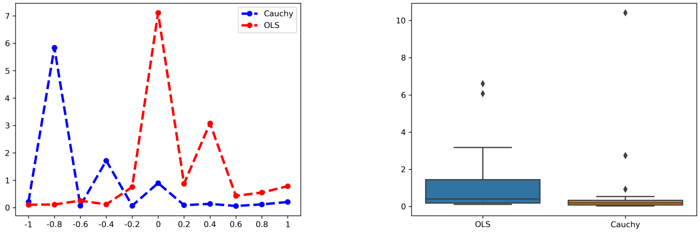
(a) Varying β (b) Box plot (node specific MRC)

Figure 6: (b) **CGLearn** has lower node specific MRC than **OLS**

CHILD Network

Variance of Regression Coefficients

- It denotes the variance about mean regression coefficients for True Positives



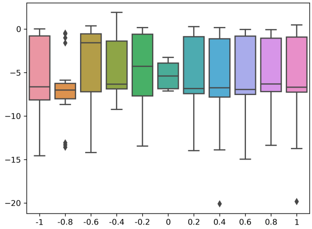
(a) Varying β (b) Box plot

Figure 7: **OLS** generally overestimates **MRC** compared to **CGLearn**

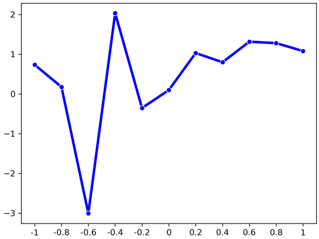
CHILD Network

Log γ

- $\log\gamma$ measures dispersion of noise variable Z



(a) $\log\gamma$



(b) Bias

$\rightarrow$ At lower values of β, our model tends to under/overestimate the dispersion in noise Z

Cross Validation

Gene Expression data

- We perform cross-validation using Gene Expression data for 1 240 subjects and 21 800 Probes
- We apply **CGLearn** to the problem of analyzing differential expression (DE) of a gene between samples.
- We processed the data as follows:
  1. log-intensity for each probe was median-centered
  2. We ranked median-centered probes in decreasing order of variance
  3. We selected the top 100 ranked probes for cross validation.
  4. We compare cross validation results of **CGLearn** against **OLS**.

Cross Validation

Log Fractional Lower Order Moments, LFLOM

$$LFLOM(T|B,p) = \sum_{X_i \in \mathcal{X}} \left[\frac{1}{p} \left(\log \mathbb{E}[|Z_i|^p] \right) \right]$$

$$= \sum_{X_i \in \mathcal{X}} \left[\frac{1}{p} \left(\log \mathbb{E} \left| X_i - \sum_{X_j \in P_a(X_i)} w_{ij} X_j \right|^p \right) \right]$$

Cross Validation

Gene Expression Data

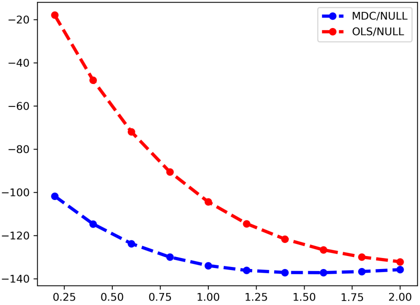


Figure 8: There is a clear departure of the data from Gaussian.

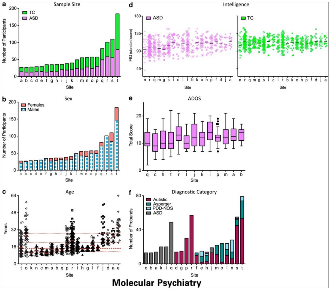
Application: Cauchy GCN

Autism Brain Imaging Data Exchange (ABIDE)

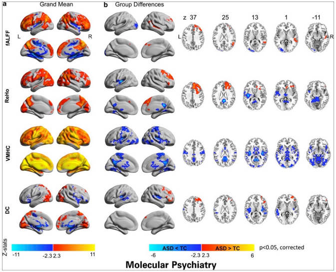
- 16 heterogeneous sites consisting of 539 subjects & 573 typical controls.
- Data consists of structural and resting state f-MRI + 106 phenotypic measures.
- Processing: Configurable Pipeline for the Analysis of Connectomes, C-PAC software (Craddock et al. 2013)
- **TADPOLE** is another popular ASD challenge dataset.

Cauchy GCN

ABIDE Sample composition & Regional abnormalities



(a) Phenotypic sample characteristics



(b) Regional measures of intrinsic functional architecture

Figure 9: There are noticeable differences between ASD and TC subjects

Cauchy GCN

Accuracy

Chebyshev, k	1	2	3	4	5
Cauchy Unweighted	0.545	0.602	**0.682**	0.659	0.659
Cauchy Weighted	0.545	<u>0.648</u>	<u>0.659</u>	<u>0.670</u>	**0.693**
Sex + Site	**0.670**	**0.659**	**0.682**	**0.682**	0.659
Cosine Similarity	0.648	0.625	0.636	<u>0.670</u>	0.648
Complete	<u>0.659</u>	<u>0.648</u>	0.648	**0.682**	<u>0.682</u>
Age+Sex+Site	<u>0.659</u>	**0.659**	**0.682**	<u>0.670</u>	0.670

Table 1: Accuracy for GCN disease prediction with different graph construction techniques. **Bold** denotes the best result and <u>underline</u> denotes the second best result.

Cauchy GCN

Area Under Curve

Chebyshev, k	1	2	3	4	5
Cauchy Unweighted	0.718	0.678	0.732	0.711	0.695
Cauchy Weighted	0.716	<u>0.736</u>	<u>0.739</u>	<u>0.734</u>	<u>0.738</u>
Sex + Site	<u>0.737</u>	0.732	0.730	0.731	0.733
Cosine Similarity	0.729	0.721	0.728	0.705	0.683
Complete	0.725	0.723	0.736	0.721	0.723
Age+Sex+Site	**0.738**	**0.737**	**0.746**	**0.735**	**0.747**

Table 2: Area Under Curve for GCN disease prediction with different graph construction techniques. **Bold** denotes the best result and <u>underline</u> denotes the second best result.

Future Works

- Apply **CGLearn** to other areas to discover hierarchical structures in data.
- Extend **CGLearn** to model dependencies in NN parameters.
- Extend our model to Dynamic Cauchy Graphical Models.

Data and codes:

- Bayesian Networks (Child, Alarm) etc available at `https://www.bnlearn.com/bnrepository/`
- ABIDE data set is available via AWS & upon application at `https://www.nitrc.org/`
- Code: CGLearn is available from authors upon request
- Cauchy-based GCN is available on my github: `https://github.com/TauraiUCB/CGLearn`