Cauchy Graphical Models

Taurai Muvunza GNF18@MAILS.TSINGHUA.EDU.CN
Yang Li YANGLI@SZ.TSINGHUA.EDU.CN
Ercan Engin Kuruoglu[*] KURUOGLU@SZ.TSINGHUA.EDU.CN
Tsinghua-Berkeley Shenzhen Institute, Institute of Data and Information, Shenzhen International Graduate School, Tsinghua University

Editors: J.H.P. Kwisthout & S. Renooij

Abstract

A common approach to learning Bayesian networks involves specifying an appropriately chosen family of parameterized probability density such as Gaussian. However, the distribution of most real-life data is leptokurtic and may not necessarily be best described by a Gaussian process. In this work we introduce Cauchy Graphical Models (CGM), a class of multivariate Cauchy densities that can be represented as directed acyclic graphs with arbitrary network topologies, the edges of which encode linear dependencies between random variables. We develop CGLearn, the resultant algorithm for learning the structure and Cauchy parameters based on Minimum Dispersion Criterion (MDC). Experiments using simulated datasets on benchmark network topologies demonstrate the efficacy of our approach when compared to Gaussian Graphical Models (GGM).

Keywords: Cauchy distribution; probabilistic graphical models; Bayesian networks; heavy tails.

1. Introduction

Bayesian networks are a class of graphical models that allow for a representation of the probabilistic dependencies between a given set of random variables as directed acyclic graphs (Nagarajan et al., 2013). A comprehensive introduction and notation to graphical models can be found in Lauritzen (1996).

Definition 1 *Given a set of finite random variables $\mathcal{X} = \{X_1, X_2, ..., X_N\}$, we define a Bayesian network $B(\mathcal{G}, \Psi)$ specified by directed acyclic graph (DAG) $\mathcal{G}$ whose nodes denote random variables in $\mathcal{X}$ and a set of parameters $\Psi = \{\psi_i | X_i \in \mathcal{X}\}$ that determine the conditional probability distribution $p(X_i | pa_{\mathcal{G}}(X_i), \psi)$ for $X_i \in \mathcal{X}$ given the state of its parents $pa_{\mathcal{G}}(X_i) \subseteq \mathcal{X} \setminus \{X_i\}$ in $\mathcal{G}$.*

Bayesian networks allow for the factorization of joint probability density of random variables as a product of the conditional probability distributions as follows:

$$P_B(\mathcal{X}) = \prod_{i=1}^{|\mathcal{X}|} p(X_i | \mathrm{pa}_{\mathcal{G}}(X_i), \psi) \tag{1}$$

[*] Corresponding Author

To ensure that the factorization $P_B(\mathcal{X})$ is well defined, DAGs do not have self-loops and the dependence of $p(X_i|\mathrm{pa}_{\mathcal{G}}(X_i), \psi)$ on ψ_i when learning Bayesian networks is usually specified by an appropriately chosen family of parameterized probability densities such as Gaussian. However, studies have shown that real-world data, for instance, microarray intensities, functional MRI data and stock prices exhibit heavy tails that can not be best described by a Gaussian process (Nolan, 2020). We solve this challenge by characterizing the dependency structure of Bayesian networks with multivariate Cauchy densities.

The Cauchy density is specified as:

$$f(x) = 1/\sigma\pi \left[1 + \left(\frac{x-\mu}{\sigma}\right)^2\right] \text{for} -\infty \leq x \leq \infty \tag{2}$$

and the distribution function is given by

$$F(x) = \int_{-\infty}^{x} f(t)dt = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x-\mu}{\sigma}\right) \tag{3}$$

where μ and σ are the location and scale parameters, respectively (Bloch, 1966). The simplicity offered by pedagogically attractive, tractable closed-form expressions make Cauchy more effective to model heavy-tailed data. Our choice to model continuous random variables with Cauchy is motivated by several theoretical guarantees which demonstrate that Cauchy distribution possess optimality properties in handling impulsive noise (Verdú, 2023). Our key **contributions** are summarized as follows:

1. We propose Cauchy Graphical Models (CGM) that can be represented as multivariate DAGs with arbitrary network topologies to model impulsive noise in random variables.

2. We introduce Minimum Dispersion Criterion (MDC), a score-based method to select the optimal DAG network of the CGM.

3. We conduct an extensive experimental campaign using synthetic data on benchmark Bayesian networks to validate the efficacy of our approach.

2. Related Work

Graphical models can be grouped into directed and undirected graphs (Maathuis et al., 2018). Mixed graphs are a special case of chain graphs that were introduced to unify the directed and undirected edges by imposing DAG structure on a disjoint subset of vertices, (Ali et al., 2009). Richardson et al. (2023) relaxed the conditional independence assumption and proposed conditional acyclic directed mixed graphs with nested Markov by incorporating inequality constraints that condition fixed vertices on variables that index the distribution.

Bayesian networks are a class of graphical models that encode the joint probability distribution for a set of variables. Besides inferring causal relationships, Bayesian networks provide a compact representation of high-dimensional data as a smaller subset of key dependent random variables. Different probability distributions have been proposed to characterize the dependency structure of DAGs. For instance, conditional Gaussian distribution (Bottcher, 2001), mixtures of truncated exponentials (Cobb and Shenoy, 2005), Gaussian distribution (Schmidt et al., 2007), copula (Elidan, 2010), mixture of Gaussians (Shenoy and West, 2011) and pair-copula (Bauer and Czado, 2016). While these methods have achieved significant performance, they heavily rely on the Guassianity assumption which is not only assumed for mathematical simplicity but also has maximum entropy among all real-valued distributions specified by mean and covariance (Cover, 1999). In this work we compare our approach to Gaussian Graphical Models (GGM) learned by Ordinary Least Squares method as they are the most representative and commonly used form of Bayesian networks.

Cauchy distribution has gained popularity across a wide range of applications due to its ability to handle impulsive noise better than the Gaussian and Laplace models. For example, it has been used to model gene expression data in computational biology (Khondoker et al., 2006). Besides, Hodge and Milligan (2011) found that Cauchy distribution outperformed Gaussian, Beta and Weibull when modeling Wind Power Forecasting Error. Additionally, Cauchy distribution has profound applications in modeling stock returns (Mahdizadeh and Zamanzade, 2019) and image denoising (Jiang et al., 2023). The Cauchy probability distribution has peculiar features due to its heavy tails and the difficulty in estimating its parameters (Johnson et al., 1995). A significant body of scholarship has been devoted to estimating the parameters of a Cauchy process. Earlier works focused on the estimation of scale and location parameters using maximum likelihood method (Antle and Bain, 1969; Ferguson, 1978). Approximations of the parameters of Cauchy distribution, albeit inefficient were also proposed earlier in literature. For example, Rothenberg et al. (1964) suggested that the maximum likelihood is difficult to calculate and interpret, and introduced the sample median as the simplest consistent estimator of the location parameter. Meanwhile, Blom (1958); Barnett (1966) considered linear estimation of the location parameter based on sample order statistics, while Chan (1970) estimated both parameters of the Cauchy distribution by considering quantiles of a small sample taken from a large sample. Koutrouvelis (1982) proposed estimation method of the location and scale using empirical characteristic function while Howlader and Weiss (1988) introduced a Bayesian approach to estimate Cauchy parameters. A new method for parameter estimation based on l_p-norm was proposed for non-Gaussian data and distributions with unknown closed form solutions (Rice and White, 1964). Ekblom and Henriksson (1969) were among the first to use l_p- norm to estimate the dispersion of a Cauchy density for different values of p. Given an ordered sample of $\mathcal{X} = \{X_1, X_2, X_3, ..., X_n\}$, and a distribution function characterised by $F(X - a)$, where a is the location parameter, the l_p estimate of the data minimizes:

$$L_p(X) = \left(\frac{1}{n-1}\sum_{i=1}^{n}|X_i - a|^p\right)^{1/p} \text{ for } p \geq 1 \tag{4}$$

The l_p- norm minimization has been extensively studied to address under-determined systems in signal processing and statistical estimation (Daubechies et al., 2010). However, its application to multivariate graphical models remains notably scarce.

3. Methodology

Definition 2 *We define a Cauchy Graphical Model (CGM) $B(\mathcal{G}, \Psi)$ as a probability distribution over $\mathcal{X}$ such that:*

$$\xi_j = X_j - \sum_{X_k \in pa_{\mathcal{G}}(X_j)} w_{jk} X_k \sim \mathit{Cauchy}(\sigma_j, \mu_j) \tag{5}$$

ξ is a noise random variable independent of ξ_k if $\xi_j \neq \xi_k, \forall X_j \in \mathcal{X}$, where $\text{pa}_{\mathcal{G}}(X_j) \subseteq \mathcal{X} \setminus \{X_j\}$ are parent nodes of X_j in the directed acyclic graph $\mathcal{G}$, and the distribution of the parameters is represented as follows:

$$w_{jk} \in \mathbb{R}, W_j = \{w_{jk} | X_k \in \text{pa}_{\mathcal{G}}(X_j)\} \tag{6}$$

$$\psi_j = \{\sigma_j, \mu_j\} \cup W_j, \tag{7}$$

$$\Psi = \{\psi_i | X_i \in \mathcal{X}\} \tag{8}$$

Lemma 3 *Given the above conditions, $B(\mathcal{G}, \Psi)$ is a Bayesian network.*

Proof We let ξ obey Markovian property and infer a non-unique ordering, τ, that is consistent with the DAG such that $\text{pa}_{\mathcal{G}}(X_j) \subseteq \{X_1 ... X_j - 1\}$. It suffices that $B(\mathcal{G}, \Psi)$ is a Bayesian network since the transformation matrix $\mathcal{T}_{\xi_i \to X_i}$ is lower triangular with each diagonal entry equal to 1 $\forall \xi_j$ independent noise variables and the Jacobian of $\mathcal{T}_{\xi_i \to X_i}$ is also equal to 1. Formally:

$$P_B(\xi_1, ... \xi_{|\mathcal{X}|}) = \prod_{j=1}^{|\mathcal{X}|} f(\xi_j | \sigma_j, \mu_j) \tag{9}$$

$$p(X_j | \text{pa}_{\mathcal{G}}(X_j), \psi) = f(\xi_j | \sigma_j, \mu_j) \tag{10}$$

$$P_B(\mathcal{X}) = P_B(\xi_1, ..., \xi_{|\mathcal{X}|}) \cdot \left| \frac{\partial(\xi_1, ..., \xi_{|\mathcal{X}|})}{\partial(X_1, ..., X_{|\mathcal{X}|})} \right| \tag{11}$$

$$P_B(\mathcal{X}) = \prod_{j=1}^{|\mathcal{X}|} p(X_j | \text{pa}_{\mathcal{G}}(X_j), \psi) \cdot \left| \frac{\partial(\xi_1, ..., \xi_{|\mathcal{X}|})}{\partial(X_1, ..., X_{|\mathcal{X}|})} \right| \tag{12}$$

$$P_B(\mathcal{X}) = \prod_{j=1}^{|\mathcal{X}|} p(X_j | \text{pa}_{\mathcal{G}}(X_j), \psi_j) \tag{13}$$

■

In (9), the probability distribution of noise variable ξ is expressed as a product of individual probability density functions. The conditional probability of the observed variable X_j given by $p(X_j | \text{pa}_{\mathcal{G}}(X_j), \psi)$ in (10) is equivalent to the distribution of the corresponding noise variables ξ_j. We simplify this expression and show that it is equivalent to the joint distribution in (1). The proof shows that a linear transformation of a Cauchy distributed random variable is Cauchy distributed.

3.1. Learning Cauchy Graphical Models

We have developed CGLearn[1] software to learn the parameters and structure of Cauchy Graphical Models and the algorithm is shown in Algorithm 1. The graphical structure of CGM is learned in an iterative process. We explore different non-unique variable orderings using Ordering Based Search (OBS). For each ordering, we perform structure learning using K2Search by greedily adding edges to the network that maximizes the MDC score over the space of all directed acyclic graphs G and Ψ parameters. Once a structure is proposed, we employ the Iterative Reweighted Least Squares (IRLS) to accurately estimate

[1] The code, data and instructions to reproduce the results in this paper can be accessed from this anonymized repository: CGLearn

Algorithm 1 CGLearn `//Structure and Parameter Learning Algorithm for CG Models`

Input: Data matrix D_{IN}, number of random restarts Nreps

Output: Cauchy Graphical model $B(G, \Psi)$ over $\mathcal{X}$

```
Symmetrize input data matrix $D \leftarrow D_{IN}$
Initialize $B(G, \Psi) = \emptyset$
for $i = 1$ to Nreps do
    Initialize a random ordering $\sigma$
    $B_\sigma(\mathcal{G}, \Psi) = OBS(D, \sigma)$    //Using Ordering-based search, Algorithm 4
    if $MDC_s(B_\sigma|D) > MDC_s(B|D)$ then
        $B = B_\sigma$
    end if
end for
```

the parameters using a connection between the l_p-norm of the noise random variable ξ and its $p-$th moment. This process of optimizing the ordering, learning the structure and refining the parameters continues until the best network structure, characterized by the highest MDC score, is identified.

3.1.1. Minimum Dispersion Criterion

Maximizing the DAG structure is complicated since the search space of the DAG increases exponentially as the number of nodes increases. Algorithms to identify the graph structure of Bayesian networks fall into three categories: score-based, constraint-based and hybrid algorithms, and in this paper we focus on score-based methods as they offer significant computational advantage when learning the structure of the network (Nagarajan et al., 2013). Network scores are a goodness of fit test that measure how well the whole DAG mirrors the dependence structure of the data. Notable examples of network scores include the Bayesian Information Criterion (BIC) (Schwarz, 1978), Minimum Dispersion Criterion (MDC) (Stuck, 1978) and Minimum Description Length (Rissanen et al., 2007).

While Gaussian-based models rely on BIC for model selection, part of our main contribution lies in proposing the use of MDC to select the optimum DAG for CGM. We employ MDC since it is shown to be an optimal and computationally efficient model selection criterion for symmetric, heavy tailed noise (Cline and Brockwell, 1985).

GGM-based BIC selects the Bayesian network that maximizes the score over the space of all possible DAG $\mathcal{G}$, and parameters Ψ. Given a data matrix $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ the BIC score for a Bayesian network $B(\mathcal{G}, \Psi)$ is given by:

$$BIC_s(B|\mathcal{D}) = \sum_{D_j \in \mathcal{D}} \log[P_B(D_j)] - \sum_{X_i \in \mathcal{X}} \frac{|\text{pa}_{\mathcal{G}}(X_i)|}{2} \log N \tag{14}$$

where $P_B(D_j)$ is the marginal likelihood estimator and $\sum_{X_i \in \mathcal{X}} \frac{|\text{pa}_{\mathcal{G}}(X_i)|}{2} \log N$ is the penalty term.

Definition 4 *We define Cauchy-based MDC score which selects the Bayesian network that maximises the score MDC_s over the space of all DAG $\mathcal{G}$ and Ψ parameters to be expressed as:*

$$MDC_s(B|\mathcal{D}) = -\sum_{X_i\in\mathcal{X}}\left\{N\mathrm{log}\sigma_i + \frac{|\mathrm{pa}_{\mathcal{G}}(X_i)|}{2}\mathrm{log}N\right\} \tag{15}$$

Symmetrization. Because the Cauchy distribution is inherently symmetric, it is essential to symmetrize the data to minimize the effects of skewness and ensure accurate, unbiased parameter estimation. Therefore, every CGM can be associated with a symmetric CGM with identical topological structure and regression parameters. Given a data set $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ denoting a CGM $B(\mathcal{G}, \Psi)$, let $\widehat{\mathcal{D}} = \{\widehat{D_1}, \widehat{D_2}, ..., \widehat{D_{Nd}}\}$ represent bootstrapped realizations specified as $\widehat{X_{i,\lambda}} = X_{i,2\lambda} - X_{i,2\lambda-1}, \forall\lambda \in \{1, 2, ..., N_d = [N/2]\}$

The new data samples $\widehat{D_\lambda} = \{\widehat{X_{i,\lambda}}|X_i \in \mathcal{X}\}$ represent independent realizations of random variables $\widehat{\mathcal{X}} \equiv \{\widehat{X_i}|X_i \in \mathcal{X}\}$. The transformation halves the total number of samples but does not change the distribution of the data. The resampled noise from the symmetrized data is also Cauchy distributed and can be represented as:

$$\widehat{\xi_j} \equiv \widehat{X_j} - \sum_{\widehat{X_k}\in\mathrm{pa}_{\mathcal{G}}(X_j)} w_{jk}\widehat{X_k} \sim \mathrm{Cauchy}(\sigma_j, \mu_j) \tag{16}$$

where $\widehat{\xi_j}$ represent resampled independent noise variables. Learning CGM can be separated into parameter and structure learning.

3.1.2. Parameter Learning

Estimating the parameters of the Cauchy distribution is a non-trivial task since the distribution is so heavy that the mean and variance do not exist. We apply properties of l_p-norm minimization to solve this challenge.

Lemma 5 *Given $\mathcal{X} = \{X_1, X_2, ..., X_n\}$, as a sequence of independent Cauchy distributed random variables, the sum of independent copies $\sum_{i=1}^N X_i$ also has the Cauchy distribution.*

Given the above Lemma and assuming symmetric data with $\mu = 0$ such that $\xi \sim \mathrm{Cauchy}(\sigma, 0)$, we can invoke stable properties (Samoradnitsky, 2017) and express the expectation of ξ as:

$$\mathbb{E}(|\xi|^p) = C(p)\sigma^p, -1 < p < 1 \tag{17}$$

Thus, the dispersion of a symmetrized Cauchy random variable ξ is associated with its moments using the above equation. More specifically, if ξ represents noise random variables, then within a constant term $C(p)$, minimizing the dispersion $\log\sigma_j$ is equivalent to minimizing the l_p-norm $\|\xi_j\|_p$ as follows:

$$\begin{aligned}\mathrm{argmin}\log\sigma_j &\equiv \mathrm{argmin}\|\xi_j\|_p \\ &\equiv (\sum_{\lambda=1}^{N}|\xi_{j,\lambda}|^p)^{1/p}, -1 < p < 1\end{aligned} \tag{18}$$

Definition 6 *Let W_j be the regression coefficients such that $W_j = \{w_{jk}|X_k \in pa_{\mathcal{G}}(X_j)\}$. We define $\sigma_j(W_j)$ to denote the dispersion parameter of the distribution of $\xi_j = X_j - \sum_{X_k\in pa_{\mathcal{G}}(X_j)} w_{jk}X_k$. The MDC-based Cauchy then selects regression parameters:*

$$W_j^* = \mathrm{argmin}\log\sigma_j(W_j)$$

$$W_j^* = \mathrm{argmin}\log\big(\|\xi_j\|_p\big) \equiv \mathrm{argmin}\log\Big((\sum_{\lambda=1}^{N}|\xi_{j,\lambda}|^p)^{1/p}\Big) \tag{19}$$

Algorithm 2 IRLS //Find the least l_p norm regression coefficients

```
Input: N dimensional vector for realizations of the child node $Y, N \times M$ matrix $X$ of realizations of the parent set $\mathrm{pa}_{\mathcal{G}}(Y)$.
Parameter: tolerance $\epsilon$ and $p \in (0, 2]$
Output: Vector of co-efficients $W^* = \mathrm{argmin}_W \|Y - XW\|_p$
Initialize $W$ with OLS coefficients $W = (X^T X)^{-1}(X^T Y)$
repeat
    Initialize buffer for current regression coefficients $\beta = W$
    Initialize a diagonal $N \times N$ matrix $\Omega$ from $\beta$ for weighted least squares regression
        $\Omega_{ij} = \delta_{ij}(Y_i - (XW)_i)^{p-2} \;\; \forall i, j \in \{1, ...N\}$
    Update regression coefficients vector $W = (X^T \Omega X)^{-1}(X^T \Omega Y)$
until Change in regression coefficients is within tolerance $\|\beta - W\|_2 < \epsilon$
```

We estimate the dispersion parameter $\log \sigma_j$ after structure learning by computing the l_p-norm and using the connection between the l_p-norm of the Cauchy random variable ξ_j and its $p-$th moment. When estimating $\log \sigma$, we generalize the formula $\log C(p)\sigma^p$ by ignoring the constant $\log C(p)$ since it is common to all candidate structures. In our experiments, regression coefficients are learned using Iterative Reweighted Least Squares (IRLS) algorithm with $p = 1$ (Daubechies et al., 2010) as shown in Algorithm 2. While IRLS is non-convex for $p < 1$, convergence proofs have been provided for $1 < p < 3$ and $p = 1$ in Osborne (1985) and Daubechies et al. (2010), respectively. We consider l_1 minimization on CGM since it is robust to outliers, computationally efficient and yields sparse solutions Daubechies et al. (2010). Furthermore, Rice and White (1964) and Ekblom and Henriksson (1969) studied the l_p-norm of several symmetrical distributions, for $-\infty < p < +\infty$, and showed that l_1 is the best l_p estimate for the Cauchy distribution.

3.1.3. Structure Learning

The main goal of learning DAG structure is to determine which nodes should be included in the network generally by using an algorithm that searches the DAG space to maximize a given network score. We search for the local optimum in the search space of all DAGs using the Ordering based Search (OBS) (Teyssier and Koller, 2012) as shown in Algorithm 4. The algorithm takes an initial ordering, τ, and learns a directed acyclic graph that is consistent with τ using K2Search (Cooper and Herskovits, 1992).

The principle of K2Search algorithm is to assume that each node lacks parents firstly and then to search for them in the preceding nodes by greedily adding edges until the MDC score reaches a local optimum as illustrated in Algorithm 3. Gaussian based graphical models (Schmidt et al., 2007) use linear regression to find a set of potential neighbors. On the contrary, our modified version of hill-climbing based K2Search uses l_p-norm instead. We compare our method to GGM learned with OLS since they are the most representative and commonly used form of Bayesian networks.

Algorithm 3 K2Search `//Perform structure learning of the DAG`

Input: Symmetrized Data matrix D and fixed ordering τ

Output: Cauchy Graphical Model $B(\mathcal{G}, \Psi)$ given τ

```
Initialize $B(\mathcal{G}, \Psi) = \emptyset$
for $i = 2$ to $|\mathcal{X}|$ do
     //Find the optimal parent set pa$_{\mathcal{G}}(\tau_i)$
   //by greedily adding edges starting from pa$_{\mathcal{G}}(\tau_i) = \emptyset$
   repeat
   Initialize $noChange = true$
   Initialize $best = FS(\tau_i, \text{pa}_{\mathcal{G}}(\tau_i)|D)$
   $Add\text{pa}_{\mathcal{G}} = \emptyset$                //Search for a potential parent
   for $X_j \in \{\tau_1, ...\tau_{i-1}\} \setminus \text{pa}_{\mathcal{G}}(\tau_i)$ do
      Estimate regression weights $W_{\tau_i}$ for parent set $\text{pa}_{\mathcal{G}}(\tau_i) \cup X_j$ using IRLS
      if $FS(\tau_i, \text{pa}_{\mathcal{G}}(\tau_i) \cup X_j|D) > best$ then
         $best = FS(\tau_i, \text{pa}_{\mathcal{G}}(\tau_i) \cup X_j|D)$          //Update best score
         $Add\text{pa}_{\mathcal{G}} = X_j$                    //Possible new parent
         $noChange = false$
      end if
   end for
   $\text{pa}_{\mathcal{G}}(\tau_i) = \text{pa}_{\mathcal{G}}(\tau_i) \cup Add\text{pa}_{\mathcal{G}}$             //Add the new parent
   until $noChange$ is $true$               //Repeat until local optimum
end for
```

4. Empirical Validation

To validate our model, we conduct extensive experiments on synthetic and genetic datasets. We discuss the experiments and results in the following sections.

4.1. Empirical Validation on Synthetic Data

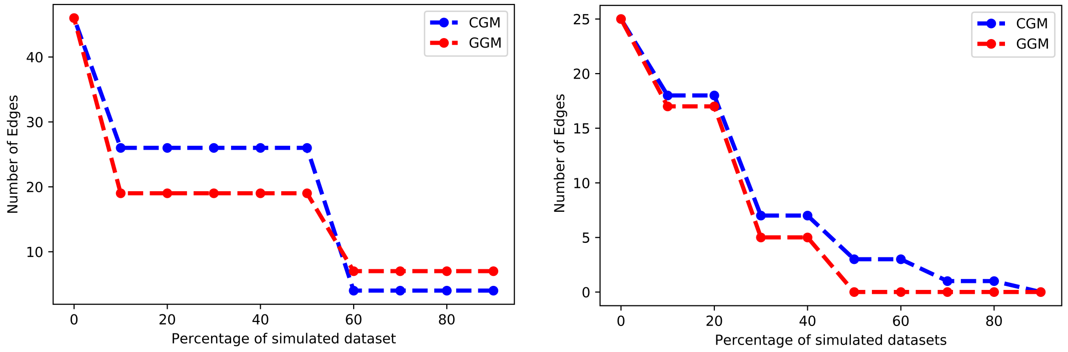
We present the findings of CGM using benchmark network topologies from the Bayesian network repository[2]: `(number of nodes, number of edges)`, ALARM`(37,46)` and CHILD `(20,25)` networks. ALARM is a Bayesian network designed to provide an alarm message system for patient monitoring (Beinlich et al., 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 et al., 1993)

Synthetic data was generated from a Cauchy distribution, Cauchy$(1, 0)$. We assigned Cauchy additive noise to every node $X_i \in \mathcal{X}$ with the same Cauchy parameters while regression coefficients sampled uniformly at random from $[-\frac{\rho}{2}, +\frac{\rho}{2}]$ were assigned on each edge. We fixed the network topology and noise parameters, and performed experiments using 100 simulated datasets, each containing 2000 samples from CGM with randomly chosen regression weights. We fixed $\rho = 1$.

Moreover, to test the robustness of CGM's performance, we varied the dispersion σ from 1 to 10 in steps of 1. This approach also evaluates our model's sensitivity to learning

[2] https://www.bnlearn.com/bnrepository/

```
Algorithm 4 OBS. //Searches for a local optimum in the space of all DAGs
Input: Symmetrized Data matrix $D$ and initial ordering $\tau$
Output: Cauchy Graphical Model $B(G, \Psi)$ over $\mathcal{X}$
Initialize Cauchy Graphical Model $B$ =K2Search$(D, \tau)$
for $i = 1$ to $|\mathcal{X}| - 1$ do
    Initialize $T_i\tau = Twiddle(i, \tau)$  //New ordering $T_i\tau$ by swapping $\tau_i$ and $\tau_{i+1}$
    $\tilde{B}$ = K2Search$(D, T_i\tau)$  //Compute the optimum $\tilde{B}$ given $T_i\tau$
    $DS(i) = MDC_s(\tilde{B}|D) - MDC_s(B|D)$  //Set Delta score for the twiddle
end for
repeat
Initialize $noChange = true$
Find $a$ =argmax $DS(i)$  //Find the best twiddle $T_a\tau$
$\tilde{B}$ =K2Search$(D, T_a\tau)$  //Compute the optimum given $T_a\tau$
if $MDC_s(\tilde{B}|D) > MDC_s(B|D)$ then
    $\tau = T_a\tau, B = \tilde{B}$  //Accept the swap and update $\tau, B$
    $DS(a-1)$ if $(a > 1)$  //Update the delta scores for
    neighbors $a - 1$
    $DS(a+1)$ if $(a < |\mathcal{X}| - 1$  //and $a + 1$, if valid
    $noChange = false$
end if
until $noChange$ is $true$  //Repeat until local optimum
```



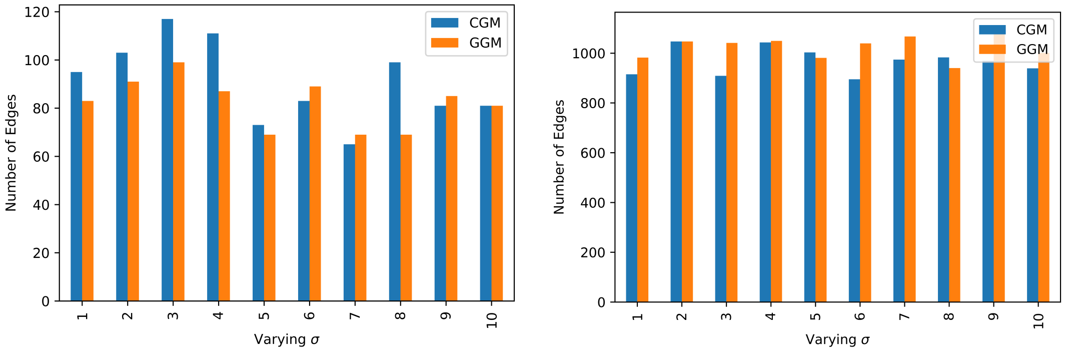
(a) ALARM

(b) CHILD

Figure 1: True Positives for ALARM and CHILD networks. The vertical axis denotes the number of edges in the network and horizontal axis represents the percentage of simulated data sets where the directed edge was learned.

difficult problems. We compare our method to GGM and report results for True/False Positives (TP/FP) which denotes the number of bootstrap replicates where each true/false positive edge was found for structure learning, Mean Regression Coefficients (MRC) and Variance of MRC (VMRC) which represent the mean regression-coefficient of each edge and the variance about mean regression coefficients for true positives, respectively.

For brevity, Figure 1 measures the frequency of correctly inferred edges that consistently appear in multiple simulated datasets. Overall, results for ALARM and CHILD networks show that CGM outperformed GGM as it inferred a higher number of edges. Meanwhile,



(*a*) True Positives

(*b*) False Positives

Figure 2: True and False Positives for different values of σ. CGM inferred a higher number of TP than GGM in most cases (a). In (b), CGM is better than GGM at not inferring incorrect edges (Type 1 errors).

a well-performing model should typically yield a higher sum of accurately inferred edges across all datasets. We note that in total, CGM inferred 192 correct edges on ALARM network, while GGM had 169. On the CHILD network, CGM learned 83 correct edges and GGM inferred only 69 edges in total.

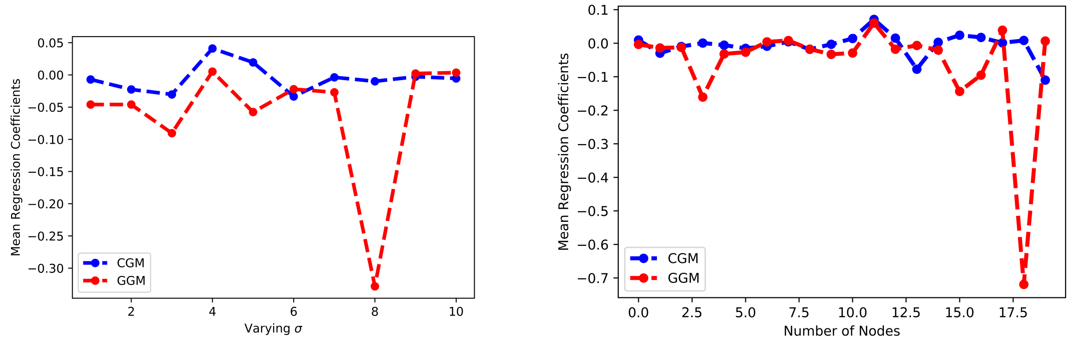
CGM's robustness to changes in dispersion, σ. To validate our model's performance in learning complex problems, we show results for different values of σ. For each value of σ, we performed ten experiments and summed the number of consistently inferred edges across many datasets. Figure 2 displays the comparative performance of CGM (blue) and GGM (yellow) on CHILD network.

The number of correctly inferred direct edges was higher for CGM compared to GGM. CGM inferred 908 correct direct edges in total, which is 10.67% more than the 822 edges learned by GGM. The consistent performance of CGM given different values of σ demonstrates the robustness of our approach against GGM.

With regards to FP, GGM incorrectly inferred 10 256 edges which is 5.65% more than the 9 677 inferred by CGM. Although both methods are quite competitive, results show that CGM is better at not inferring incorrect edges while GGM continuously display poor performance on TP (Type II errors) and FP (Type I errors).

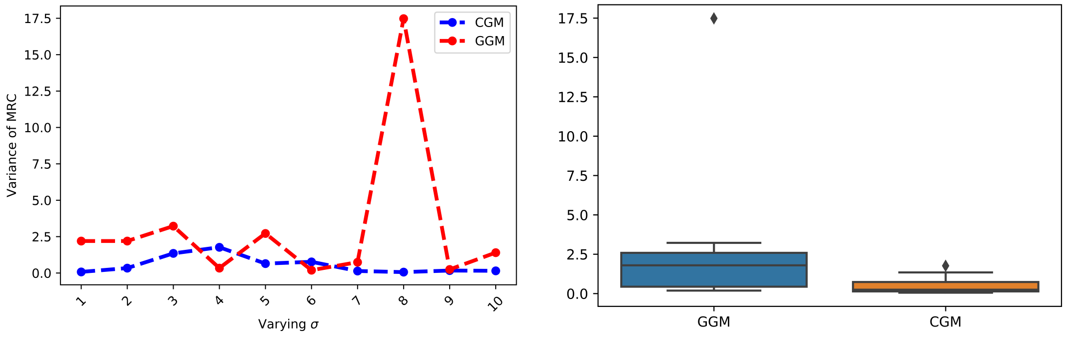
Mean and Variance of regression coefficients. For MRC and VMRC, we averaged the results of ten experiments and computed the mean and variance across all nodes for each value of σ. Figure 3 shows the comparative performance of both approaches. Results show that GGM is generally characterized by larger deviations in mean and variance, as shown in Figures 3 and 4, respectively. This shows that for heavy tailed data, GGM is not reliable at estimating the regression coefficients, while CGM remains robust in sensitivity to changes in the distribution of data. Overally, the node-specific variance obtained using CGM is lower and concentrated around zero while GGM exhibits larger variance as shown in Figure 4(b).

Dispersion parameter log σ. We assessed the accuracy and robustness of CGM in estimating the dispersion parameter of the noise variables. GGM is not considered in this experiment since it assumes Gaussian noise. We present results of $\log \sigma$ calculated as



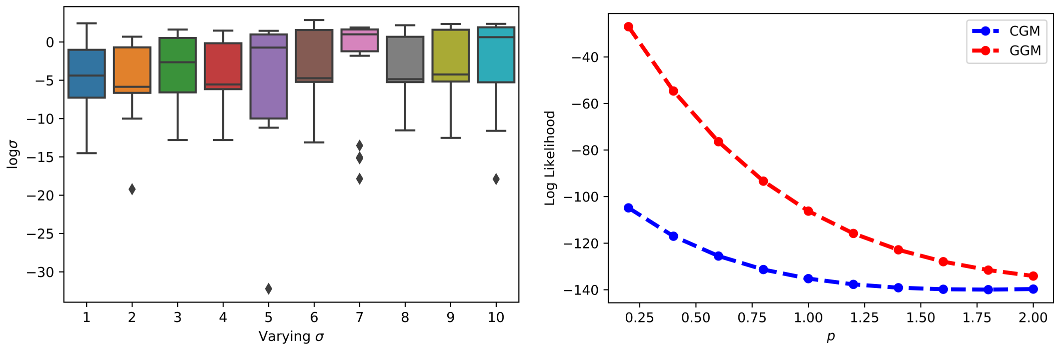
(a) MRC (b) MRC at node level

Figure 3: MRC for different values of σ. On average, GGM yields higher MRC compared to CGM. Node-specific MRC is also higher for GGM in (b).



(a) VMRC (b) VMRC at node level

Figure 4: VMRC for different values of σ. CGM shows a lower VMRC demonstrating the reliability of the model in estimating MRC. Node-specific VMRC is also lower for CGM compared to GGM as shown in the box plot.



(a) $\log \sigma$ (b) LFLOM Cross Validation

Figure 5: Estimates of $\log \sigma$ for different values of σ are shown in (a). Cross validation results in terms of LFLOM in (b) show that there is a clear departure of the data from the Gaussian process

$\log \sigma = 1/|\mathcal{X}| \sum_i \log \sigma_i$ for different values of σ in Figure 5. The results show relatively large variability as σ increases, indicating the difficulty in estimating the dispersion. It is important to note that estimating the dispersion of Cauchy noise random variables is a challenging parameter domain for most existing methods (Johnson et al., 1995).

4.2. Empirical Validation on Genetic Data

Analysis of gene differentiation show that microarrays contain a number of error sources and while the measurement errors remain unknown, they are generally believed to follow a Cauchy process (Vetterling, 2002). We perform cross validation on real world genetic dataset. The data consist of 21 800 gene probes of 1 240 individuals from 7 population groups. Although the dataset was later expanded, details of the preliminary version can be accessed in Stranger et al. (2012). The data was processed by taking the log intensities and centering them around the median. The median-centred probes were ranked in decreasing order of variance and we selected the top 100 probes and performed cross validation on them.

We compared CGM and GGM by reporting the goodness of fit in terms of Log Fractional Lower Order Moments (LFLOM) for a graphical model B on the test set $T = \{T_1; ...T_N\}$ as follows:

$$\begin{aligned} LFLOM(T|B,p) &= \sum_{X_i \in \mathcal{X}} \left[\frac{1}{p} (\log \mathbb{E}[|\xi_i|^p]) \right] \\ &= \sum_{X_i \in \mathcal{X}} \left[\frac{1}{p} \left(\log \mathbb{E}|X_i - \sum_{X_j \in \mathrm{pa}_{\mathcal{G}}(X_i)} w_{ij} X_j|^p \right) \right] \end{aligned} \tag{20}$$

where w_{ij} denotes the regression coefficients learned by CGM or GGM. Our cross validation tests whether the variation in a child node X_i can be explained by the set of its parents $\mathrm{pa}_{\mathcal{G}}(X_i)$. In that case we expect LFLOM to be small. In our analysis, we are interested in assessing the performance of each approach by considering each gene as a random variable. We performed a ten-fold cross validation of LFLOM for CGM and GGM and the averaged results in Figure 5(b) show the difference between optimal network for each of the approaches and an empty network without edges (NULL).

The results demonstrate a clear departure of the noise variable ξ_i from Gaussian, with both approaches showing a narrower difference as p approaches 2. This is along expected lines since LFLOM is identical to the negative log-likelihood of GGM as the noise variable ξ_i is symmetrized before cross validation.

5. Conclusions

We proposed CGM that can be represented as DAGs with arbitrary network topologies. We introduced MDC score to select the optimal DAG network of the CGM. We empirically validate the resultant algorithm, CGLearn on synthetic and gene expression data. Cross validation results and experiments on benchmark Bayesian networks demonstrate the efficacy of our approach compared to Gaussian-based graphical models. In the future, we hope to build dynamic CGM to cater for more complex application scenarios.

Acknowledgments

Ercan E. Kuruoglu was supported in part by Tsinghua Shenzhen International Graduate School Start-up fund under Grant QD2022024C and Shenzhen Science and Technology Innovation Commission under Grant JCYJ20220530143002005. Yang Li was supported in part by the Natural Science Foundation of China (Grant 62371270). Yang Li and Ercan E. Kuruoglu were both supported by Shenzhen Key Laboratory of Ubiquitous Data Enabling (No.ZDSYS20220527171406015).

References

R. A. Ali, T. S. Richardson, and P. Spirtes. Markov Equivalence for Ancestral Graphs. 2009.

C. E. Antle and L. J. Bain. A property of maximum likelihood estimators of location and scale parameters. *SIAM Review*, 11(2):251–253, 1969.

V. Barnett. Evaluation of the maximum-likelihood estimator where the likelihood equation has multiple roots. *Biometrika*, 53(1/2):151–165, 1966.

A. Bauer and C. Czado. Pair-copula Bayesian networks. *Journal of Computational and Graphical Statistics*, 25(4):1248–1271, 2016.

I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89: Second European Conference on Artificial Intelligence in Medicine, London, August 29th–31st 1989. Proceedings*, pages 247–256. Springer, 1989.

D. Bloch. A note on the estimation of the location parameter of the Cauchy distribution. *Journal of the American Statistical Association*, 61(315):852–855, 1966.

G. Blom. *Statistical Estimates and Transformed Beta-variables.* PhD thesis, Almqvist & Wiksell, 1958.

S. Bottcher. Learning Bayesian networks with mixed variables. In *International Workshop on Artificial Intelligence and Statistics*, pages 13–20. PMLR, 2001.

L. K. Chan. Linear estimation of the location and scale parameters of the Cauchy distribution based on sample quantiles. *Journal of the American Statistical Association*, 65(330): 851–859, 1970.

D. B. Cline and P. J. Brockwell. Linear prediction of ARMA processes with infinite variance. *Stochastic Processes and their Applications*, 19(2):281–296, 1985.

B. R. Cobb and P. P. Shenoy. Nonlinear deterministic relationships in Bayesian networks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 27–38. Springer, 2005.

G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

T. M. Cover. *Elements of Information Theory.* John Wiley & Sons, 1999.

I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively Reweighted Least Squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

H. Ekblom and S. Henriksson. L_p-criteria for the estimation of location parameters. *SIAM Journal on Applied Mathematics*, 17(6):1130–1141, 1969.

G. Elidan. Copula Bayesian networks. *Advances in Neural Information Processing Systems*, 23, 2010.

T. S. Ferguson. Maximum likelihood estimates of the parameters of the Cauchy distribution for samples of size 3 and 4. *Journal of the American Statistical Association*, 73(361):211–213, 1978.

B.-M. Hodge and M. Milligan. Wind power forecasting error distributions over multiple timescales. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–8, 2011. doi: 10.1109/PES.2011.6039388.

H. Howlader and G. Weiss. On Bayesian estimation of the Cauchy parameters. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 350–361, 1988.

M. Jiang, X. Feng, C. Wang, H. Zhang, et al. Robust color image watermarking algorithm based on synchronization correction with multi-layer perceptron and Cauchy distribution model. *Applied Soft Computing*, 140:110271, 2023.

N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions, Volume 2*, volume 289. John Wiley & Sons, 1995.

M. R. Khondoker, C. A. Glasbey, and B. J. Worton. Statistical estimation of gene expression using multiple laser scans of microarrays. *Bioinformatics*, 22(2):215–219, 2006.

I. A. Koutrouvelis. Estimation of location and scale in Cauchy distributions using the empirical characteristic function. *Biometrika*, 69(1):205–213, 1982.

S. L. Lauritzen. *Graphical Models*, volume 17. Clarendon Press, 1996.

M. Maathuis, M. Drton, S. Lauritzen, and M. Wainwright. *Handbook of Graphical Models.* CRC Press, 2018.

M. Mahdizadeh and E. Zamanzade. Goodness-of-fit testing for the Cauchy distribution with application to financial modeling. *Journal of King Saud University-Science*, 31(4): 1167–1174, 2019.

R. Nagarajan, M. Scutari, and S. Lèbre. Bayesian Networks in R. *Springer*, 122:125–127, 2013.

J. P. Nolan. *Univariate Stable Distributions.* Springer, 2020.

M. R. Osborne. *Finite Algorithms in Optimization and Data Analysis.* John Wiley & Sons, Inc., 1985.

J. R. Rice and J. S. White. Norms for smoothing and estimation. *SIAM Review*, 6(3): 243–256, 1964.

T. S. Richardson, R. J. Evans, J. M. Robins, and I. Shpitser. Nested Markov properties for acyclic directed mixed graphs. *The Annals of Statistics*, 51(1):334–361, 2023.

J. Rissanen et al. *Information and Complexity in Statistical Modeling*, volume 152. Springer, 2007.

T. J. Rothenberg, F. M. Fisher, and C. B. Tilanus. A note on estimation from a Cauchy sample. *Journal of the American Statistical Association*, 59(306):460–463, 1964.

G. Samoradnitsky. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance.* Routledge, 2017.

M. Schmidt, A. Niculescu-Mizil, K. Murphy, et al. Learning graphical model structure using L1-regularization paths. In *AAAI*, volume 7, pages 1278–1283, 2007.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.

P. P. Shenoy and J. C. West. Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning*, 52(5):641–657, 2011.

D. J. Spiegelhalter, A. P. Dawid, S. L. Lauritzen, and R. G. Cowell. Bayesian analysis in expert systems. *Statistical Science*, pages 219–247, 1993.

B. E. Stranger, S. B. Montgomery, A. S. Dimas, L. Parts, O. Stegle, C. E. Ingle, M. Sekowska, G. D. Smith, D. Evans, M. Gutierrez-Arcelus, et al. Patterns of cis regulatory variation in diverse human populations. *PLoS Genetics*, 8(4):e1002639, 2012.

B. Stuck. Minimum error dispersion linear filtering of scalar symmetric stable processes. *IEEE Transactions on Automatic Control*, 23(3):507–509, 1978.

M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, 2012.

S. Verdú. The Cauchy Distribution in Information Theory. *Entropy*, 25(2):346, 2023.

W. T. Vetterling. *Numerical Recipes Example Book (C++): The Art of Scientific Computing.* Cambridge University Press, 2002.