Inference with Multivariate Heavy-Tails in Linear Models

Danny Bickson and Carlos Guestrin
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213
{bickson,guestrin}@cs.cmu.edu

Abstract

Heavy-tailed distributions naturally occur in many real life problems. Unfortunately, it is typically not possible to compute inference in closed-form in graphical models which involve such heavy-tailed distributions.

In this work, we propose a novel simple linear graphical model for independent latent random variables, called linear characteristic model (LCM), defined in the characteristic function domain. Using stable distributions, a heavy-tailed family of distributions which is a generalization of Cauchy, Lévy and Gaussian distributions, we show for the first time, how to compute both exact and approximate inference in such a linear multivariate graphical model. LCMs are not limited to stable distributions, in fact LCMs are always defined for any random variables (discrete, continuous or a mixture of both).

We provide a realistic problem from the field of computer networks to demonstrate the applicability of our construction. Other potential application is iterative decoding of linear channels with non-Gaussian noise.

1 Introduction

Heavy-tailed distributions naturally occur in many real life phenomena, for example in computer networks [23, 14, 16]. Typically, a small set of machines are responsible for a large fraction of the consumed network bandwidth. Equivalently, a small set of users generate a large fraction of the network traffic. Another common property of communication networks is that network traffic tends to be linear [8, 23]. Linearity is explained by the fact that the total incoming traffic at a node is composed from the sum of distinct incoming flows.

Recently, several works propose to use linear multivariate statistical methods for monitoring network health, performance analysis or intrusion detection [15, 13, 16, 14]. Some of the aspects of network traffic makes the task of modeling it using a probabilistic graphical models challenging. In many cases, the underlying heavy-tailed distributions are difficult to work with analytically. That is why existing solutions in the area of network monitoring involve various approximations of the joint probability distribution function using a variety of techniques: mixtures of distributions [8], spectral decomposition [13] historgrams [14], sketches [16], entropy [14], sampled moments [23], etc.

In the current work, we propose a novel linear probabilistic graphical model called linear characteristic model (LCM) to model linear interactions of independent heavy-tailed random variables (Section 3). Using the stable family of distributions (defined in Section 2), a family of heavy-tailed distributions, we show how to compute both exact and approximate inference (Section 4). Using real data from the domain of computer networks we demonstrate the applicability of our proposed methods for computing inference in LCM (Section 5).

We summarize our contributions below:

- We propose a new linear graphical model called LCM, defined as a product of factors in the cf domain. We show that our model is well defined for any collection of random variables, since any random variable has a matching cf.
- Computing inference in closed form in linear models involving continuous variables is typically limited to the well understood cases of Gaussians and simple regression problems in exponential families. In this work, we extend the applicability of belief propagation to the stable family of distributions, a generalization of Gaussian, Cauchy and Lévy distributions. We analyze both exact and approximate inference algorithms, including convergence and accuracy of the solution.
- We demonstrate the applicability of our proposed method, performing inference in real settings, using network tomography data obtained from the PlanetLab network.

1.1 Related work

There are three main relevant works in the machine learning domain which are related to the current work: Convolutional Factor Graphs (CFG), Copulas and Independent Component Analysis (ICA). Below we shortly review them and motivate why a new graphical model is needed.

Convolutional Factor Graphs (CFG) [18, 19] are a graphical model for representing linear relation of independent latent random variables. CFG assume that the probability distribution factorizes as a convolution of potentials, and proposes to use duality to derive a product factorization in the characteristic function (cf) domain. In this work we extend CFG by defining the graphical model as a product of factors in the cf domain. Unlike CFGs, LCMs are always defined, for any probability distribution, while CFG may are not defined when the inverse Fourier transform does not exist.

A closely related technique is the Copula method [22, 17]. Similar to our work, Copulas assume a linear underlying model. The main difference is that Copulas transform each marginal variable into a uniform distribution and perform inference in the cumulative distribution function (cdf) domain. In contrast, we perform inference in the cf domain. In our case of interest, when the underlying distributions are stable, Copulas can not be used since stable distributions are not analytically expressible in the cdf domain.

A third related technique is ICA (independent component analysis) on linear models [27]. Assuming a linear model $Y = AX$[1], where the observations Y are given, the task is to estimate the linear relation matrix A, using only the fact that the latent variables X are statistically mutually independent. Both techniques (LCM and ICA) are complementary, since ICA can be used to learn the linear model, while LCM is used for computing inference in the learned model.

2 Stable distribution

Stable distribution [30] is a family of heavy-tailed distributions, where Cauchy, Lévy and Gaussian are special instances of this family (see Figure 1). Stable distributions are used in different problem domains, including economics, physics, geology and astronomy [24]. Stable distribution are useful since they can model heavy-tailed distributions that naturally occur in practice. As we will soon show with our networking example, network flows exhibit empirical distribution which can be modeled remarkably well by stable distributions.

We denote a stable distribution by a tuple of four parameters: $\mathcal{S}(\alpha, \beta, \gamma, \delta)$. We call α as the characteristic exponent, β is the skew parameter, γ is a scale parameter and δ is a shift parameter. For example (Fig. 1), a Gaussian $\mathcal{N}(\mu, \sigma^2)$ is a stable distribution with the parameters $\mathcal{S}(2, 0, \frac{\sigma}{\sqrt{2}}, \mu)$, a Cauchy distribution $\texttt{Cauchy}(\gamma, \delta)$ is stable with $\mathcal{S}(1, 0, \gamma, \delta)$ and a Lévy distribution $\text{Lévy}(\gamma, \delta)$ is stable with $\mathcal{S}(\frac{1}{2}, 1, \gamma, \delta)$. Following we define formally a stable distribution. We begin by defining a unit scale, zero-centered stable random variable.

Definition 2.1. *[25, Def. 1.6] A random variable X is* stable *if and only if $X \sim aZ + b$, $0 < \alpha \leq 2$, $-1 \leq \beta \leq 1$, $a, b \in \mathbb{R}$, $a \neq 0$ and Z is a random variable with characteristic function*[2]

$$E[\exp(iuZ)] = \begin{cases} \exp\left(-|u|^{\alpha}[1 - i\beta \tan(\frac{\pi\alpha}{2}) \operatorname{sign}(u)]\right) & \alpha \neq 1 \\ \exp\left(-|u|[1 + i\beta \frac{2}{\pi} \operatorname{sign}(u) \log(|u|)]\right) & \alpha = 1 \end{cases}. \tag{1}$$

[1] Linear model is formally defined in Section 3.

[2] We formally define characteristic function in the supplementary material.

Next we define a general stable random variable.

Definition 2.2. *[25, Def. 1.7] A random variable X is $\mathcal{S}(\alpha,\beta,\gamma,\delta)$ if*

$$X \sim \begin{cases} \gamma(Z - \beta\tan(\frac{\pi\alpha}{2})) + \delta & \alpha \neq 1 \\ \gamma Z + \delta & \alpha = 1 \end{cases},$$

where Z is given by (1)*. X has characteristic function*

$$E\exp(iuZ) = \begin{cases} \exp(-\gamma^\alpha |u|^\alpha [1 - i\beta\tan(\frac{\pi\alpha}{2})\operatorname{sign}(u)(|\gamma u|^{1-\alpha} - 1)] + i\delta u) & \alpha \neq 1 \\ \exp(-\gamma|u|[1 + i\beta\frac{2}{\pi}\operatorname{sign}(u)\log(\gamma|u|)] + i\delta u) & \alpha = 1 \end{cases}.$$

A basic property of stable laws is that weighted sums of α-stable random variables is α-stable (and hence the family is called stable). This property will be useful in the next section where we compute inference in a linear graphical model with underlying stable distributions. The following proposition formulates this linearity.

Proposition 2.1. *[25, Prop. 1.16]*

a) **Multiplication by a scalar.** *If $X \sim \mathcal{S}(\alpha,\beta,\gamma,\delta)$ then for any $a, b \in \mathbb{R}, a \neq 0$,*

$$aX + b \sim \mathcal{S}(\alpha, \operatorname{sign}(a)\beta, |a|\gamma, a\delta + b)\,.$$

b) **Summation of two stable variables.** *If $X_1 \sim \mathcal{S}(\alpha,\beta_1,\gamma_1,\delta_1)$ and $X_2 \sim \mathcal{S}(\alpha,\beta_2,\gamma_2,\delta_2)$ are independent, then $X_1 + X_2 \sim \mathcal{S}(\alpha,\beta,\gamma,\delta)$ where*

$$\beta = \frac{\beta_1\gamma_1^\alpha + \beta_2\gamma_2^\alpha}{\gamma_1^\alpha + \gamma_2^\alpha}, \quad \gamma^\alpha = \gamma_1^\alpha + \gamma_2^\alpha \;, \quad \delta = \delta_1 + \delta_2 + \xi\,,$$

$$\xi = \begin{cases} \tan(\frac{\pi\alpha}{2})[\beta\gamma - \beta_1\gamma_1 - \beta_2\gamma_2] & \alpha \neq 1 \\ \frac{2}{\pi}[\beta\gamma\log\gamma - \beta_1\gamma_1\log\gamma_1 - \beta_2\gamma_2\log\gamma_2] & \alpha = 1 \end{cases}.$$

Note that both X_1, X_2 have to be distributed with the same characteristic exponent α.

3 Linear characteristic models

A drawback of general stable distributions, is that they do not have closed-form equation for the pdf or the cdf. This fact makes the handling of stable distributions more difficult. This is probably one of the reasons stable distribution are rarely used in the probabilistic graphical models community.

We propose a novel approach for modeling linear interactions between random variables distributed according to stable distributions, using a new linear probabilistic graphical model called LCM. A new graphical model is needed, since previous approaches like CFG or the Copula method can not be used for computing inference in closed-form in linear models involving stable distribution, because they require computation in the pdf or cdf domains respectively. We start by defining a linear model:

Definition 3.1. *(Linear model) Let $X_1, \cdots, X_n$ a set of mutually independent random variables.*[3] *Let $Y_1, \cdots, Y_m$ be a set of observations obtained using the linear model:*

$$Y_i \sim \sum_j A_{ij} X_j \quad \forall_i\,,$$

where $A_{ij} \in \mathbb{R}$ are weighting scalars. We denote the linear model in matrix notation as $Y = AX$.

Linear models are useful in many domains. For example, in linear channel decoding, X are the transmitted codewords, the matrix A is the linear channel transformation and Y is a vector of observations. When X are distributed using a Gaussian distribution, the channel model is called AWGN (additive white Gaussian noise) channel. Typically, the decoding task is finding the most probable X, given A and the observation Y. Despite the fact that X are assumed statistically mutually independent when transmitting, given an observation Y, X are not independent any more, since they are correlated via the observation. Besides of the network application we focus on, other potential application to our current work is linear channel decoding with stable, non-Gaussian, noise.

In the rest of this section we develop the foundations for computing inference in a linear model using underlying stable distributions. Because stable distributions do not have closed-form equations in the pdf domain, we must work in the cf domain. Hence, we define a dual linear model in the cf domain.

[3] We do not limit the type of random variables. The variables may be discrete, continuous, or a mixture of both.

3.1 Duality of LCM and CFG

CFG [19] have shown that the joint probability $p(x, y)$ of any linear model can be factorized as a convolution:

$$p(x, y) = p(x_1, \cdots, x_n, y_1, \cdots, y_m) = \prod_i^* p(x_i, y_1, \cdots, y_m)\,. \tag{2}$$

Informally, LCM is the dual representation of (2) in the characteristic function domain. Next, we define LCM formally, and establish the duality to the factorization given in (2).

Definition 3.2. *(LCM) Given the linear model Y=AX, we define the linear characteristic model (LCM)*

$$\varphi(t_1, \cdots, t_n, s_1, \cdots, s_m) \triangleq \prod_i \varphi(t_i, s_1, \cdots, s_m)\,,$$

where $\varphi(t_i, s_1, \cdots, s_m)$ is the characteristic function[4] *of the joint distribution $p(x_i, y_1, \cdots, y_m)$.*

The following two theorems establish duality between the LCM and its dual representation in the pdf domain. This duality is well known (see for example [18, 19]), but important for explaining the derivation of LCM from the linear model.

Theorem 3.3. *Given a LCM, assuming $p(x, y)$ as defined in (2) has a closed form and the Fourier transform $\mathcal{F}[p(x, y)]$ exists, then the $\mathcal{F}[p(x, y)] = \varphi(t_1, \cdots, t_n, s_1, \cdots, s_m)$.*

Theorem 3.4. *Given a LCM, when the inverse Fourier transform exists, then $\mathcal{F}^{-1}(\varphi(t_1, \cdots, t_n, s_1, \cdots, s_m)) = p(x, y)$ as defined in (2).*

The proof of all theorem is deferred to the supplementary material. Whenever the inverse Fourier transform exists, LCM model has a dual CFG model. In contrast to the CFG model, LCM are always defined, even the inverse Fourier transform does not exist. The duality is useful, since it allows us to compute inference in either representations, whenever it is more convenient.

4 Main result: exact and approximate inference in LCM

This section brings our main result. Typically, exact inference in linear models with continuous variables is limited to the well understood cases of Gaussian and simple regression problem in exponential families. In this section we extend previous results, to show how to compute inference (both exact and approximate) in linear model with underlying stable distributions.

4.1 Exact inference in LCM

The inference task typically involves computation of marginal distribution or a conditional distribution of a probability function. For the rest of the discussion we focus on marginal distribution. Marginal distribution of the node x_i is typically computed by integrating out all other nodes:

$$p(x_i|y) \sim \int_{X \setminus i} p(x, y)\, d_{X \setminus i}\,,$$

where $X \setminus i$ is the set of all nodes excluding node i. Unfortunately, when working with stable distribution, the above integral is intractable. Instead, we propose to use a dual operation called slicing, computed in the cf domain.

Definition 4.1. *(slicing/evaluation)[28, p. 110]*

(a) **Joint cf**. *Given random variables X_1, X_2, the* joint cf *is $\varphi_{X_1,X_2}(t_1, t_2) = E[e^{it_1x_1 + it_2x_2}]$.*
(b) **Marginal cf**. *The* marginal cf *is derived from the joint cf by $\varphi_{X_1}(t_1) = \varphi_{X_1,X_2}(t_1, 0)$. This operation is called slicing or evaluation. We denote the slicing operation as $\varphi_{X_1}(t_1) = \varphi_{X_1,X_2}(t_1, t_2)\Big]_{t_2=0}$.*

The following theorem establishes the fact that marginal distribution can be computed in the cf domain, by using the slicing operation.

[4] Defined in the supplementary material.

for $i \in |T|$
{
Eliminate t_i by computing

$$\phi_{m+i}(N(t_i)) = \prod_{\varphi_j \in N(t_i)} \phi(t_j, s_1, \cdots, s_m) \Big]_{t_i=0}$$

Remove $\phi(t_j, s_1, \cdots, s_m)$ and t_i from LCM.
Add ϕ_{m+i} to LCM.
}
Finally: If $\mathcal{F}^{-1}$ exists, compute

$$p(x_i) = \mathcal{F}^{-1}(\phi_{final}) .$$

Algorithm 1: Exact inference in LCM using LCM-Elimination.

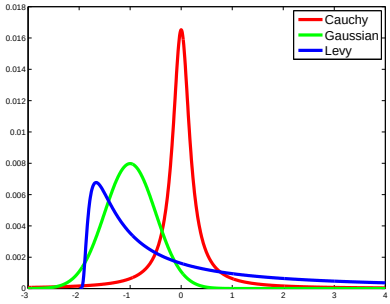

Figure 1: The three special cases of stable distribution where closed-form pdf exists.

Theorem 4.2. *Given a LCM, the marginal cf of the random variable X_i can be computed using*

$$\varphi(t_i) = \prod_j \varphi(t_j, s_1, \cdots, s_m) \Big]_{T \setminus i=0} , \tag{3}$$

In case the inverse Fourier transform exists, then the marginal probability of the hidden variable X_i is given by $p(x_i) \sim \mathcal{F}^{-1}\{\varphi(t_i)\}$.

Based on the results of Thm. 4.2 we propose an exact inference algorithm, LCM-Elimination, for computing the marginal cf (shown in Algorithm 1). We use the notation $N(k)$ as the set of graph neighbors of node k, excluding k[5]. T is the set $\{t_1, \cdots, t_n\}$.

LCM-Elimination is dual to CFG-Elimination algorithm [19]. LCM-Elimination operates in the cf domain, by evaluating one variable at a time, and updating the remaining graphical model accordingly. The order of elimination does not affect correctness (although it may affect efficiency). Once the marginal cf $\varphi(t_i)$, is computed, assuming the inverse Fourier transform exists, we can compute the desired marginal probability $p(x_i)$.

4.2 Exact inference in stable distributions

After defining LCM and showing that inference can be computed in the cf domain, we are finally ready to show how to compute exact inference in a linear model with underlying stable distributions. We assume that all observation nodes Y_i are distributed according to a stable distribution. From the linearity property of stable distribution, it is clear that the hidden variables X_i are distributed according to a stable distribution as well. The following theorem is one of the the novel contributions of this work, since as far as we know, no closed-form solution was previously derived.

Theorem 4.3. *Given a LCM, $Y = AX + Z$, with n i.i.d. hidden variables $X_i \sim \mathcal{S}(\alpha, \beta_{x_i}, \gamma_{x_i}, \delta_{x_i})$, n i.i.d. noise variables with known parameters $Z_i \sim \mathcal{S}(\alpha, \beta_{z_i}, \gamma_{z_i}, \delta_{z_i})$, and n observations $y_i \in \mathbb{R}$, assuming the matrix $A_{n\times n}$ is invertible*[6]*, then*
a) the observations Y_i are distributed according to stable distribution $Y_i \sim \mathcal{S}(\alpha, \beta_{y_i}, \gamma_{y_i}, \delta_{y_i})$ with the following parameters:

$$\gamma_y = |A|^\alpha \gamma_x + \gamma_z, \quad \beta_y = \gamma_y^{-\alpha} \odot [(|A| \odot \text{sign}(A))(\beta_x \odot \gamma_x) + \beta_z \odot \gamma_z], \quad \delta_y = A\delta_x + \xi_y$$

$$\xi_y = \begin{cases} \tan(\frac{\pi\alpha}{2})[\beta_y \odot \gamma_y - A(\beta_x \odot \gamma_x) - \beta_z \odot \gamma_z] & \alpha \neq 1 \\ \frac{2}{\pi}[\beta_y \odot \gamma_y \odot \log(\gamma_y) - A \odot \log(|A|)(\beta_x \odot \gamma_x) - A(\beta_x \odot \gamma_x \odot \log(\gamma_x)) - \beta_z \odot \gamma_z] & \alpha = 1 \end{cases},$$

b) the result of exact inference for computing the marginals $p(x_i|y) \sim \mathcal{S}(\alpha, \beta_{x_i|y}, \gamma_{x_i|y}, \delta_{x_i|y})$ is given in vector notation:

$$\beta_{x|y} = \gamma_{x|y}^{-\alpha} \odot [(|A|^\alpha \odot \text{sign}(A))^{-1}(\beta_y \odot \gamma_y^\alpha)], \quad \gamma_{x|y}^\alpha = (|A|^\alpha)^{-1}\gamma_y^\alpha, \quad \delta_{x|y} = A^{-1}[\delta_y - \xi_x], \tag{4}$$

[5]More detailed explanation of the construction of a graphical model out of the linear relation matrix A is found on [4, Chapter 2.3].

[6]To simplify discussion we assume that the length of both the hidden and observation vectors $|X| = |Y| = n$. However the results can be equivalently extended to the more general case where $|X| = n, |Y| = m, m \neq n$. See for example [6].

Initialize: $m_{ij}(x_j) = 1, \quad \forall A_{ij} \neq 0.$
Iterate until convergence

$$m_{ij}(t_j) = \varphi_i(t_i, s_1, \cdots, s_m) \prod_{k \in N(i) \setminus j} m_{ki}(t_i) \Big]_{t_i = 0}$$

Finally:

$$\varphi(t_i) = \varphi_i(t_i, s_1, \cdots, s_m) \prod_{k \in N(i)} m_{ki}(t_i).$$

(a)

Initialize: $m_{ij}(x_j) = 1, \quad \forall A_{ij} \neq 0.$
Iterate until convergence

$$m_{ij}(x_j) = \int_{x_i} p(x_i, y_1, \cdots, y_m) * \prod_{k \in N(i) \setminus j}^{*} m_{ki}(x_i) dx_i$$

Finally:

$$p(x_i) = p(x_i, y_1, \cdots, y_m) * \prod_{k \in N(i)}^{*} m_{ki}(x_i).$$

(b)

Initialize: $\beta_{x_i|y}, \gamma_{x_i|y}, \delta_{x_i|y} = \mathcal{S}(\alpha, 0, 0, 0), \quad \forall_i .$
Iterate until convergence:

$$\gamma^{\alpha}_{x_i|y} = \gamma^{\alpha}_{y_i} - \sum_{j \neq i} |A_{ij}|^{\alpha} \gamma^{\alpha}_{x_j|y}, \quad \beta_{x_i|y} = \beta_{y_i} \gamma^{\alpha}_{y_i} - \sum_{j \neq i} \operatorname{sign}(A_{ij}) |A_{ij}|^{\alpha} \beta_{x_j|y}, \quad \delta_{x_i|y} = \delta_{y_i} - \sum_{j \neq i} A_{ij} \delta_{x_j|y} - \xi_{x_i|y},$$

$$\xi_{x_i|y} = \begin{cases} \tan(\frac{\pi\alpha}{2})[\beta_{y_i}\gamma_{y_i} - \sum_j A_{ij}\beta_{x_j|y}\gamma_{x_j|y}^{\frac{1-\alpha}{\alpha}}] & \alpha \neq 1 \\ \frac{2}{\pi}[\beta_{y_i}\gamma_{y_i}\log(\gamma_{y_i}) - \sum_{j:A_{ij}\neq 0} A_{ij}\log(|A_{ij}|)\beta_{x_j|y}\gamma_{x_j|y}^{\frac{1-\alpha}{\alpha}} - \sum_j A_{ij}\beta_{x_j|y}\gamma_{x_j|y}\log(\gamma_{x_j|y}^{\frac{1-\alpha}{\alpha}})] & \alpha = 1 \end{cases} \tag{6}$$

Output: $x_i|y \sim \mathcal{S}(\alpha, \beta_{x_i|y}/\gamma^{\alpha}_{x_i|y}, \gamma_{x_i|y}, \delta_{x_i|y})$ (c)

Algorithm 2: Approximate inference in LCM using the (a) Characteristic-Sum-Product (CSP) algorithm (b) Integral Convolution (IC) algorithm. Both are exact on tree topologies. (c) Stable-Jacobi algorithm.

$$\xi_{x|y} = \begin{cases} \tan(\frac{\pi\alpha}{2})[\beta_y \odot \gamma_y - A(\beta_{x|y} \odot \gamma_{x|y})] & \alpha \neq 1 \\ \frac{2}{\pi}[\beta_y \odot \gamma_y \odot \log(\gamma_y) - (A \odot \log(|A|)(\beta_{x|y} \odot \gamma_{x|y}) - A(\beta_{x|y} \odot \gamma_{x|y} \odot \log(\gamma_{x|y}))] & \alpha = 1 \end{cases}, \tag{5}$$

where $\odot$ is the entrywise product (of both vectors and matrices), $|A|$ is the absolute value (entrywise) $\log(A), A^{\alpha}, \operatorname{sign}(A)$ *are entrywise matrix operations and* $\beta_x \triangleq [\beta_{x_1}, \cdots, \beta_{x_n}]^T$ *and the same for* $\beta_y, \beta_z, \gamma_x, \gamma_y, \gamma_z, \delta_x, \delta_y, \delta_z$.

4.3 Approximate Inference in LCM

Typically, the cost of exact inference may be expensive. For example, in the related linear model of a multivariate Gaussian (a special case of stable distribution), LCM-Elimination reduces to Gaussian elimination type algorithm with a cost of $O(n^3)$, where n is the number of variables. Approximate methods for inference like belief propagation [26], usually require less work than exact inference, but may not always converge (or convergence to an unsatisfactory solution). The cost of exact inference motivates us to devise a more efficient approximations.

We propose two novel algorithms that are variants of belief propagation for computing approximate inference in LCM. The first, Characteristic-Slice-Product (CSP) is defined in LCM (shown in Algorithm 2(a)). The second, Integral-Convolution (IC) algorithm (Algorithm 2(b)) is its dual in CFG. As in belief propagation, our algorithms are exact on tree graphical models. The following theorem establishes this fact.

Theorem 4.4. *Given an LCM with underlying tree topology (the matrix A is an irreducible adjacency matrix of a tree graph), the CSP and IC algorithms, compute exact inference, resulting in the marginal cf and the marginal distribution respectively.*

The basic property which allows us to devise the CSP algorithm is that LCM is defined as a product of factor in the cf domain. Typically, belief propagation algorithms are applied to a probability distribution which factors as a product of potentials in the pdf domain. The sum-product algorithm uses the distributivity of the integral and product operation to devise efficient recursive evaluation of the marginal probability. Equivalently, the Characteristic-Slice-Product algorithm uses the distributivity of the slicing and product operations to perform efficient inference to compute the marginal cf in the cf domain, as shown in Theorem 4.4. In a similar way, the Integral-Convolution algorithm uses distributivity of the integral and convolution operations to perform efficient inference in the pdf domain. Note that the original CFG work [18, 19] did not consider approximate inference. Hence our proposed approximate inference algorithm further extends the CFG model.

4.4 Approximate inference for stable distributions

For the case of stable distributions, we derive an approximation algorithm, Stable-Jacobi (Algorithm 2(c)), out of the CSP update rules. The algorithm is derived by substituting the convolution and multiplication by scalar operations (Prop. 2.1 b,a) into the update rules of the CSP algorithm given in Algorithm 2(a).

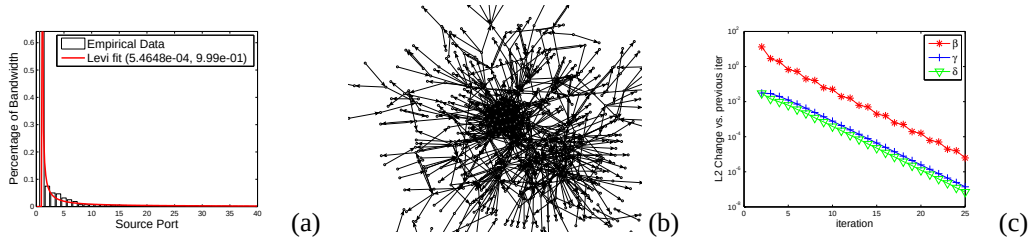

Figure 2: (a) Distribution of network flows on a typical PlanetLab host is fitted quite well with a Levy distribution. (b) The core of the PlanetLab network. 1% of the flows consists of 19% of the total bandwidth. (c) Convergence of Stable-Jacobi.

Like belief propagation, our approximate algorithm Stable-Jacobi is not guaranteed to converge on general graphs containing cycles. We have analyzed the evolution dynamics of the update equations for Stable-Jacobi and derived sufficient conditions for convergence. Furthermore, we have analyzed the accuracy of the approximation. Not surprisingly, the sufficient condition for convergence relates to the properties of the linear transformation matrix A. The following theorem is one of the main novel contributions of this work. It provides both sufficient condition for convergence of Stable-Jacobi as well as closed-form equations for the fixed point.

Theorem 4.5. *Given a LCM with n i.i.d hidden variables X_i, n observations Y_i distributed according to stable distribution $Y_i \sim \mathcal{S}(\alpha, \beta_{y_i}, \gamma_{y_i}, \delta_{y_i})$, assuming the linear relation matrix $A_{n\times n}$ is invertible and normalized to a unit diagonal*[7]*, Stable-Jacobi (as given in Algorithm 2(c)) converges to a unique fixed point under both the following sufficient conditions for convergence (both should hold):*

$$(1) \qquad \rho(|R|^{\alpha}) < 1\,, \qquad\qquad (2) \qquad \rho(R) < 1\,.$$

where $\rho(R)$ is the spectral radius (the largest absolute value of the eigenvalues of R), $R \triangleq I - A$, $|R|$ is the entrywise absolute value and $|R|^{\alpha}$ is the entrywise exponentiation. Furthermore, the unique fixed points of convergence are given by equations (4)-(5)*. The algorithm converges to the* **exact marginals** *for the linear-stable channel.*[8]

5 Application: Network flow monitoring

In this section we propose a novel application for inference in LCMs to model network traffic flows of a large operational worldwide testbed. Additional experimental results using synthetic examples are found in the supplementary material. Network monitoring is an important problem in monitoring and anomaly detection of communication networks [15, 16, 8]. We obtained Netflow PlanetLab network data [10] collected on 25 January 2010. The PlanetLab network [1] is a distributed networking testbed with around 1000 server nodes scattered in about 500 sites around the world. We define a network flow as a directed edge between a transmitting and receiving hosts. The number of packets transmitted in this flow is the scalar edge weight.

We propose to use LCMs for modeling distribution of network flows. Figure 2(a) plots a distribution of flows, sorted by their bandwidth, on a typical PlanetLab node. Empirically, we found out that network flow distribution in a single PlanetLab node are fitted quite well using Lévy distribution a stable distribution with $\alpha = 0.5, \beta = 1$. The empirical means are $\texttt{mean}(\gamma) \approx 1e^{-4}$, $\texttt{mean}(\delta) \approx 1$. For performing the fitting, we use Mark Veillette's Matlab stable distribution package [31].

Using previously proposed techniques utilizing histograms [16] for tracking flow distribution in Figure 2(a), we would need to store 40 values (percentage of bandwidth for each source port). In contrast, by approximating network flow distribution with stable distributions, we need only 4

[7] When the matrix A is positive definite it is always possible to normalize it to a unit diagonal. The normalized matrix is $\mathbf{D}^{-\frac{1}{2}} A \mathbf{D}^{-\frac{1}{2}}$ where $\mathbf{D} = \mathbf{diag}(A)$. Normalizing to a unit diagonal is done to simplify convergence analysis (as done for example in [12]) but does not limit the generality of the proposed method.

[8] Note that there is an interesting relation to the walk-summability convergence condition [12] of belief propagation in the Gaussian case: $\rho(|R|) < 1$. However, our results are more general since they apply for any characteristic exponent $0 < \alpha \leq 2$ and not just for $\alpha = 2$ as in the Gaussian case.

parameters $(\alpha, \beta, \gamma, \delta)$! Thus we dramatically reduce storage requirements. Furthermore, using the developed theory in previous sections, we are able to linearly aggregate distribution of flows in clusters of nodes.

We extracted a connected component of traffic flows connecting the core network 652 nodes. We fitted a stable distribution characterizing flow behavior for each machine. A partition of 376 machines as the observed flows Y_i (where flow distribution is known). The task is to predict the distribution of the unobserved remaining 376 flows X_i, based on the observed traffic flows (entries of A_{ij}). We run approximate inference using Stable-Jacobi and compared the results to the exact result computed by LCM-Elimination. We emphasize again, that using related techniques (Copula method , CFG, and ICA) it is not possible to compute exact inference for the problem at hand. In the supplementary material, we provide a detailed comparison of two previous approximation algorithms: non-parametric BP (NBP) and expectation propagation (EP).

Figure 2(c) plots convergence of the three parameters β, γ, δ as a function of iteration number of the Stable-Jacobi algorithm. Note that convergence speed is geometric. ($\rho(R) = 0.02 << 1$). Regarding computation overhead, LCM-Exact algorithm requires $4 \cdot 376^3$ operations, while Stable-Jacobi converged to an accuracy of $1e^{-5}$ in only $4 \cdot 376^2 \cdot 25$ operations. Additional benefit of the Stable-Jacobi is that it is a distributed algorithm, naturally suitable for communication networks. Source code of some of the algorithms presented here can be found on [3].

6 Conclusion and future work

We have presented a novel linear graphical model called LCM, defined in the cf domain. We have shown for the first time how to perform exact and approximate inference in a linear multivariate graphical model when the underlying distributions are stable. We have discussed an application of our construction for computing inference of network flows.

We have proposed to borrow ideas from belief propagation, for computing efficient inference, based on the distributivity property of the slice-product operations and the integral-convolution operations. We believe that other problem domains may benefit from this construction, and plan to pursue this as a future work.

We believe there are several exciting directions for extending this work. Other families of distributions like geometric stable distributions or Wishart can be analyzed in our model. The Fourier transform can be replaced with more general kernel transform, creating richer models.

Acknowledgement

D. Bickson would like to thank Andrea Pagnani (ISI) for inspiring the direction of this research, to John P. Nolan (American University) , for sharing parts of his excellent book about stable distribution online, Mark Veillette (Boston University) for sharing his stable distribution code online, to Jason K. Johnson (LANL) for assisting in the convergence analysis and to Sapan Bathia and Marc E. Fiuczynski (Princeton University) for providing the PlanetFlow data. This research was supported by ARO MURI W911NF0710287, ARO MURI W911NF0810242, NSF Mundo IIS-0803333 and NSF Nets-NBD CNS-0721591.

References

[1] PlanetLab Network Homepage `http://www.planet-lab.org/`.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Calculation. Numerical Methods.* Prentice Hall, 1989.

[3] D. Bickson. Linear characteristic graphical models Matlab toolbox. Carnegie Mellon university. Available on `http://www.cs.cmu.edu/~bickson/stable/`.

[4] D. Bickson. *Gaussian Belief Propagation: Theory and Application.* PhD thesis, The Hebrew University of Jerusalem, 2008.

[5] D. Bickson, D. Baron, A. T. Ihler, H. Avissar, and D. Dolev. Fault identification via non-parametric belief propagation. *IEEE Tran. on Signal Processing*, to appear, 2010.

[6] D. Bickson, O. Shental, P. H. Siegel, J. K. Wolf, and D. Dolev. Gaussian belief propagation based multiuser detection. In *IEEE Int. Symp. on Inform. Theory (ISIT)*, Toronto, Canada, July 2008.

[7] M. Briers, A. Doucet, and S. S. Singh. Sequential auxiliary particle belief propagation. In *International Conference on Information Fusion*, pages 705–711, 2005.

[8] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1875–1883, May 2007.

[9] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[10] M. Huang, A. Bavier, and L. Peterson. Planetflow: maintaining accountability for network services. *SIGOPS Oper. Syst. Rev.*, (1):89–94, 2006.

[11] A. T. Ihler, E. Sudderth, W. Freeman, and A. Willsky. Efficient multiscale sampling from products of Gaussian mixtures. In *Neural Information Processing Systems (NIPS)*, Dec. 2003.

[12] J. Johnson, D. Malioutov, and A. Willsky. Walk-Sum Interpretation and Analysis of Gaussian Belief Propagation. In *Advances in Neural Information Processing Systems 18*, pages 579–586, 2006.

[13] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, number 4, pages 219–230, New York, NY, USA, October 2004.

[14] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 217–228, New York, NY, USA, 2005. ACM.

[15] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. In *SIGMETRICS '04/Performance '04: Proceedings of the joint international conference on Measurement and modeling of computer systems*, number 1, pages 61–72, New York, NY, USA, June 2004.

[16] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identification of network anomalies using sketch subspaces. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 147–152, New York, NY, USA, 2006. ACM.

[17] H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. In *Journal of Machine Learning Research, to appear.*, 2009.

[18] Y. Mao and F. R. Kschischang. On factor graphs and the Fourier transform. In *IEEE Trans. Inform. Theory*, volume 51, pages 1635–1649, August 2005.

[19] Y. Mao, F. R. Kschischang, and B. J. Frey. Convolutional factor graphs as probabilistic models. In *UAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 374–381, Arlington, Virginia, United States, 2004. AUAI Press.

[20] R. J. Marks II. *Handbook of Fourier Analysis and Its Applications*. Oxford University Press, 2009.

[21] T. P. Minka. Expectation propagation for approximate Bayesian inference. In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[22] R. B. Nelsen. *An Introduction to Copulas*. Springer Serias in Statistics, second edition, 2006.

[23] H. X. Nguyen and P. Thiran. Network loss inference with second order statistics of end-to-end flows. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 227–240, New York, NY, USA, 2007. ACM.

[24] J. P. Nolan. Bibliography on stable distributions, processes and related topics. Technical report, 2010.

[25] J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, Boston, 2010. In progress, Chapter 1 online at `http://academic2.american.edu/~jpnolan`.

[26] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, 1988.

[27] H. Shen, S. Jegelka, and A. Gretton. Fast kernel-based independent component analysis. *Signal Processing, IEEE Transactions on*, 57(9):3498–3511, May 2009.

[28] T. T. Soong. *Fundamentals of Probability and Statistics for Engineers*. Wiley, 2004.

[29] E. Sudderth, A. T. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2003.

[30] V. V. Uchaikin and V. M. Zolotarev. Chance and stability. In *Stable Distributions and their Applications*. Utrecht, VSP, 1999.

[31] M. Veillette. Stable distribution Matlab package. Boston university. Available on `http://math.bu.edu/people/mveillet/`.

[32] A. Yener, R. Yates, and S. Ulukus. CDMA multiuser detection: A nonlinear programming approach. *IEEE Tran. On Communications*, 50(6):1016–1024, 2002.