Graph Component Analysis via Score-Matching

Advisor: John Lafferty

Dennis Zhao

1 Abstract

We present a generalization of independent component analysis (ICA) and tree component analysis (TCA). Instead of looking for a linear transform that makes the data components independent or fit by a tree-structured graphical model, we seek a transform that makes the data components fit by a sparse graphical model. By using score-matching techniques, we can extend the algorithm used for TCA and relax the tree graphical model constraint. Using a combination of an interactive algorithm and Givens rotations, we can estimate the optimal transform and a sparse graphical structure that best represents the joint density of the data. We then demonstrate the potential of such a method via a simulation on synthetically generated data from a non-tree distribution, which yields promising qualitative results, and propose future simulations that will yield a more quantitative and concrete comparison of the methods.

2 Introduction

Given a multivariate random variable $x \in \mathbb{R}^m$ where the data is assumed to be generated as a linear mixture of independent components, independent component analysis (ICA) [Com94] ICA allows us to find a linear transform $W$ such that the resulting components of $s = Wx$ are as independent as possible. ICA has been applied successfully to many problems in fields such as audio blind source separation or biomedical image processing. It can also be used as a general multivariate density estimation method where, only univariate density estimation needs to be performed after finding the optimal transformation $W$.

To generalize this method, Bach and Jordan proposed tree component analysis (TCA) [BJ13] to relax the independence requirement of the solution. Now, TCA allows us to find the linear
transform $W$ such that the components of $s = Wx$ can be well modeled by a tree-structured graphical model. Furthermore, the topology of the tree is not determined in advance of performing TCA. By weakening the assumption that the underlying components are independent, TCA is then applicable to a wider variety of problems. For example, instruments in a recording are generally not mutually independent, to TCA would allow us to model their dependencies. TCA could also be used as an efficient method for general multivariate density estimation.

In this paper, we further generalize these ideas, now relaxing the requirement on the nature of the data sources; we now have require that the components of $s = Wx$ can be modeled via a sparse graphical model. Similar to the case of TCA, we do not fix the topology of the graph in advance; instead, we search for the best possible graph in our algorithm. One of the main obstacles in generalizing TCA by proposing Graph Component Analysis (GCA), to solve more problems that fall in this class. One of the main obstacles in extending ICA and TCA to GCA is the need to compute the normalizing constant of the joint density; while we usually use the maximum likelihood estimate (MLE), the MLE is usually intractable for high dimensional graphical models as the computation of the normalizing constant is a $d$-fold convolution. To bypass this problem, we use the properties of score-matching in the context of graphical models as proposed in [Jan15]; by using an alternative scoring rule to the log-likelihood, we no longer need to compute the normalizing constant of the distribution.

3 Background and Previous Work

We first give an overview of previous work on ICA, TCA, and score-matching. The ICA model is of the form $x = As$ where $A$ is an invertible matrix and $s$ has independent components. We then want to find $W = A^{-1}$. Note that if $x$ is Gaussian, ICA reduces to Principal Component Analysis (PCA) and the model is not identifiable; more specifically, the optimal matrix $W$ is only defined up to a rotation. Let $\mathcal{D}^W$ denote the set of all distributions $q(x)$ such that $s = Wx$ has independent components. As the best approximation to $p(x)$ by a distribution in $\mathcal{D}^W$, in KL divergence is the product of the marginals of $s = Wx$, ICA then wants to minimize the mutual information of the estimated components of $s = Wx$.

For TCA, in addition to having $x = As$ where again $A$ is an invertible matrix and $s$ has independent components, we also have $s$ factorizes as a tree $T$. We then let $\mathcal{D}^{W,T}$ denote the set of all such distributions associated with matrix $W$ and tree $T$. Hence, the minimum of the KL-
divergence of $p(x)$, the distribution of $x$, and $q(x) \in \mathcal{D}^{W,T}$, is the $T$-mutual information of $s$. In other words, we want to minimize

$$J(x, W, T) = I(s_1, \ldots, s_m) - \sum_{(u,v) \in T} I(s_u, s_v)$$

Finally, Janofsky describes score matching in detail in [Jan15]. Score matching is a method for graph structure learning and parameter estimation based on minimizing the regularized Hyvärinen score of the data. For a given density $q(x)$ that is twice continuously differentiable over $\mathbb{R}^d$ and satisfies $\|p(x)\nabla \log q(x)\| \to 0$ for all $\|x\| \to \infty$, with $X \sim p$ the Hyvärinen score is given as

$$h(x, q) = \frac{1}{2} \|\nabla \log q(x)\|^2 + \Delta \log q(x)$$

where $\Delta \phi(x) = \sum_{i \in V} \frac{\partial^2 \phi(x)}{\partial x_i^2}$

The Hyvärinen score then induces the Fisher Divergence, given as

$$F(p||q) = \mathbb{E}_{X \sim p} \left[ \left\| \nabla \log \frac{p(X)}{q(X)} \right\|^2 \right]$$

This is homogeneous in $q$, so we no longer need to compute the normalizing constant of $q$. This method applies to any continuous pairwise exponential family as long as it follows some weak smoothness and tail conditions. Janofsky also works out the details for model selection for non-parametric pairwise model by choosing the sufficient statistics to be basis elements.

4 Algorithm

Our algorithm will work in the following context. Let $G = (V, E)$ be a graph with $|V| = p$, $s \in \mathbb{R}^p$ be the source, and $x \in \mathbb{R}^p$ be the observed data. Let $A \in \mathbb{R}^{p \times p}$ be invertible with $W = A^{-1}$. We observe $x_1, \ldots, x_n$ from

$$x = As$$

$$q_S(s) = \exp(g(s) - \Psi)$$

We can thus write our model for the density of $x$ as:

$$q_X(x) = |W|q_S(Wx)$$

$$\log q_X(x) = g(Wx) + \log |W| - \Psi.$$ (2)

We further assume

$$g(s) = \sum_{k \in [K]} \theta_k \phi_k(s) = (\theta, \phi(s)).$$
where $\phi_k : \mathbb{R}^p \rightarrow \mathbb{R}$ and $\phi : \mathbb{R}^p \rightarrow \mathbb{R}^K$ is the vector-valued function $\phi(s) = (\phi_1(s), \ldots, \phi_K(s))^T \in \mathbb{R}^K$. We wish to estimate $W$ and as a byproduct, estimate $\theta$.

We first define some notation. Given a function $f : \mathbb{R} \rightarrow \mathbb{R}$, let $\partial^j f$ denote the $j$-th derivative with respect to the argument. For a function $g : \mathbb{R}^p \rightarrow \mathbb{R}$ and for $u,v \in V$, let $\partial_{u_1,u_2,\ldots,u_j}^j g = \frac{\partial^{j} g}{\partial s_{u_1} \ldots \partial s_{u_j}}$. We also have $\partial \psi = (\partial \psi_1, \ldots, \partial \psi_m)^T$. For a vector function $\psi \in \mathbb{R}^m$ whose elements map from $\mathbb{R}^p \rightarrow \mathbb{R}$, define $\partial \psi \in \mathbb{R}^K$ such that $\partial \psi = (\partial \psi_1, \ldots, \partial \psi_m)^T$. Finally, let $\Delta g(s) = \sum_{v \in V} \partial^2_{vv} g(s)$.

Before stating the algorithm, we define the following terms. Note that all of the terms depend implicitly on $\phi$. First, let $f_\theta(\langle w_v, x \rangle)$ be the part of $L(x,W,\theta)$ depending on $\langle w_v, x \rangle$. Now, let $C(Wx) = (J_{\phi}(Wx)J_{\phi}(Wx)^T) = \sum_{v \in V} \partial_v \phi(Wx)\partial_v \phi(Wx)^T$

$\xi(Wx) = \sum_{v \in V} \partial^2_{vv} \phi(Wx)$

$L(\theta,W,x) = \frac{1}{2} \theta^T C(Wx) \theta + \theta^T \xi(Wx) + \frac{\lambda}{2} \| \theta \|^2_2$ (3)

$\hat{C}_W = \hat{E} C(Wx)$

$\hat{\xi}_W = \hat{E} \xi(Wx)$

$\hat{L}(\theta,W) = \hat{E} L(\theta,W,x)$

Using these quantities, we have the following algorithm

**Algorithm 1: GCA.** Define $\hat{C}_W$, $\hat{\xi}_W$, $\hat{L}(\theta,W)$ as in (3). Let $G_{u,v}(\eta)$ be as defined in (7)

**Input:** Dataset $\{x_i\}_{i \in [n]}$, $x_i \in \mathbb{R}^p$ for all $i$, penalty coefficient $\lambda \in \mathbb{R}$, cutoff $\epsilon$, number of steps $T$

**Output:** $\theta \in \mathbb{R}^K$, $W \in \mathbb{R}^{p \times p}$.

Standardize the data so that $\hat{\Sigma} = I_p$ ;

Initialize $W_0 \in M$ randomly ;

While $\hat{L}(\theta,W) > \epsilon$ ;

\[ \theta = (\hat{C}_W + \lambda I_K)^{-1}(-\hat{\xi}_W) ; \]

While not converged and $t \in [T]$, do ;

Select pair of indices $(u(t),v(t))$ satisfying $1 \leq u(t) < v(t) \leq p$ ;

$\eta_{t+1} = \text{argmin}_{\eta} \hat{L}(\theta,W_t G_{u,v}(\eta))$ ;

$W_{t+1} = W_t G_{u,v}(\eta_{t+1})$ ;

$t = t + 1 ;$
4.1 Identifiability

We still have the same scaling and permutation identifiability issues from ICA. The model (1) equivalent to 
\[ x = CA\tilde{s} \]
where \( C \) is a diagonal matrix with non-zero entries and \( \tilde{s} \) is the correspondingly-scaled version of \( s \). That is, the model is only identifiable up to scaling. Hence, we assume 
\[ W'W = WW' = I. \]

Furthermore, the model (1) is only identifiable up to permutation. That is, for any permutation matrix \( P \), 
\[ x = AP^{-1}Ps. \]
Hence, we will estimate the graphical model up to the equivalence class under permutation of the nodes. Note that the score-matching objective is invariant under permutation of the nodes.

4.2 Derivation of Algorithm

We have the objective to be
\[ L(\theta, W) = \frac{1}{2} \mathbb{E} \| \nabla_x \log p_X(x) - \nabla_x \log q_X(x) \|_2^2 = \frac{1}{2} \mathbb{E} \| \nabla_x \log q_X(x) \|_2^2 + \mathbb{E} \text{tr}(\nabla_x^2 \log q_X(x)). \]  
(4)

We now use the transformation (2) and the chain rule to obtain an objective in \( \theta \) and \( W \).
\[ \nabla_x \log q_X(x) = \nabla_x (\log q_S(Wx) + \log |W|) = W^T (\nabla \log q_S(Wx)) \]
and
\[ \nabla_x^2 \log q_X(x) = W^T (\nabla^2 \log q_S(Wx))W. \]

Plugging into (4) and noting that \( WW^T = I_p \), the score-matching objective is
\[ L(\theta, W) = \frac{1}{2} \mathbb{E} (\nabla g(Wx)^T \nabla g(Wx)) + \mathbb{E} \text{tr}(\nabla^2 g(Wx)) \]  
(5)
\[ = \frac{1}{2} \theta^T (\mathbb{E} J_\phi(Wx) J_\phi(Wx)^T) \theta + \theta^T \left( \mathbb{E} \sum_{v \in V} \partial_{vv}^2 \phi(Wx) \right), \]  
(6)

where we utilize the fact that \( \nabla g = (\theta^T \partial_1 \phi, \ldots, \theta^T \partial_p \phi)^T = (\theta^T J_\phi)^T = J_\phi^T \theta \) and \( (\nabla^2 g)_{vv} = \langle \theta, \partial_{vv}^2 \phi \rangle. \)

In our optimization, we use the empirical expectation. With \( \hat{C}_W, \hat{\xi}_W \) as defined in (3), and adding an \( \ell_2 \) penalty for \( \theta \), the empirical score-matching objective becomes as in (3). The optimization problem is
\[ \min_{\theta, W} \hat{L}(\theta, W) \text{ s.t. } WW^T = I_p \]

We now look into \( \theta \) and \( W \) optimization.
4.3 $\theta$ optimization

To optimize $\theta$, fix $W$. Note that the objective is quadratic in $\theta$ and there is a closed-form solution

$$\hat{\theta} = (\hat{C}_W + \lambda I_K)^{-1} \left( -\hat{\xi}_W \right),$$

where $\hat{C}_W, \hat{\xi}_W$ are as in (3). However, if we impose other penalties on $\theta$ (e.g. if we use the graphical model structure where we require $\theta_{vu} = \theta_{uv}$), we must use an iterative algorithm for $\theta$. Here, we will use the Alternating direction method of multipliers (ADMM) algorithm, a variant on standard augmented Lagrangian method that uses partial updates.

4.4 $W$ optimization

To optimize $W$, we use the Givens orthogonal rotation algorithm, described in (1). For $u, v \in V$, define

$$G_{u,v}(\eta) = \begin{pmatrix} 1 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \ldots & \cos \eta & \ldots & -\sin \eta & \ldots & 0 \\ \vdots & \ddots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & \sin \eta & \ldots & \cos \eta & \ldots & 0 \\ \vdots & \ddots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \ldots & 0 & \ldots & 1 \end{pmatrix}$$

Specifically, this matrix has 1’s along the diagonal except for the marked columns, and 0’s elsewhere except for the marked $(u, u), (v, v), (u, v),$ and $(v, u)$ entries. We thus want to find $\argmin_{\eta} \hat{L}(WG_{u,v}(\eta))$ by computing the first derivative of the function with respect to $\eta$.

Fix a pair $(u,v)$; we then write $G_\eta$ for simplicity. We will also let $f_v = \partial_v \phi$, $g_v = \partial^2_{vv} \phi$, and $y_\eta := WG_\eta x$. Our objective in $\eta$ is $\mathbb{E}L(\theta, WG_\eta, x)$, where

$$L(\eta, x) = \mathcal{L}(\theta, WG_\eta, x) = \frac{1}{2} \theta^T \left( \sum_{v \in V} f_v(y_\eta)f_v(y_\eta)^T \right) \theta + \theta^T \left( \sum_{v \in V} g_v(y_\eta) \right)$$

Now let $h : \mathbb{R}^p \rightarrow \mathbb{R}$, let $\eta$ be a scalar, and let $y_\eta := y(\eta) = (y_1(\eta), \ldots, y_p(\eta))^T \in \mathbb{R}^p$ be a vector depending on $\eta$. Then $\partial_\eta h(y_\eta) = \langle \nabla h(y_\eta), \partial_\eta y_\eta \rangle$, so $\partial_\eta y_\eta = W \partial_\eta G_\eta x$ where $\partial_\eta G_{u,v}(\eta)$ is the matrix whose entries are the derivatives of the entries of $G_\eta$ with respect to $\eta$. As $f_v, g_v : \mathbb{R}^p \rightarrow \mathbb{R}^K$, they
are also vectors in $\mathbb{R}^K$). We also have
\[
\partial_\eta f_v(y_\eta) \triangleq (\partial_\eta f_v(y_\eta), \ldots, \partial_\eta f_v(y_\eta))^T
\]
\[
= (\langle \nabla f_v(y_\eta), \partial_\eta y_\eta \rangle, \ldots, \langle \nabla f_v(y_\eta), \partial_\eta y_\eta \rangle)^T
\]
\[
= J_{f_v}(y_\eta)\partial_\eta y_\eta
\]

By the product rule on each element of $f_v(y_\eta)f_v(y_\eta)^T$, we have $\partial_\eta (\theta^T(f_v f_v)^T \theta) = \theta^T((\partial_\eta f_v)f_v^T + f_v(\partial_\eta f_v)^T)\theta$. Thus
\[
\partial_\eta L(\theta,WG_\eta, x) = \frac{1}{2} \theta^T \left( \sum_{v \in V} J_{f_v}(y_\eta)\partial_\eta y_\eta f_v(y_\eta)^T + f_v(y_\eta)\partial_\eta y_\eta^T J_{f_v}(y_\eta)^T \right) \theta + \theta^T \left( \sum_{v \in V} J_{g_v}(y_\eta)\partial_\eta y_\eta \right)
\]

Moving the derivative inside the expectation, we have $\partial_\eta L(\theta, WG_\eta) = E \partial_\eta L(\theta, WG_\eta, x)$.

5 Experiments

In order to test the effectiveness of our proposed GCA, we generate random vectors in $\mathbb{R}^6$ from a graphical model. The original graph has the following structure

![Graphical structure](image)

Figure 1: Graphical structure of generated data in $\mathbb{R}^6$.

and we generate the data via the Nonparanormal distribution. We then run ICA, TCA, and GCA on this data and compare the marginal distribution of each of the components after unmixing.
via the estimated $W$. Furthermore, in the case of TCA and GCA, we also compare the predicted graphical structure from each of the methods. Below are the marginal distributions, accounting for permutations.
Figure 2: Marginal distributions of original data and post-ICA/TCA/GCA of each of the 6 components.
We also plot the density estimates on top of the histograms for the marginal distributions of the data from original, post-ICA, and post-TCA.
Figure 3: Marginal distributions with density estimates of original data and post-ICA/TCA of each of the 6 components.

For each component, we see that GCA is the closest to capturing the original marginal distribution. Generally, the marginal distributions produced by TCA are often very different from the original marginal distributions. Furthermore, while the marginal distributions produced by ICA are sometimes close, they usually do not match the original marginal distributions as well as that of GCA. We can also look at the estimated graph from TCA

Figure 4: Estimated graph of the joint distribution by TCA

As expected, TCA only searches across possible trees and is thus unable to estimate the cyclic
structure from which we generated the original data. Meanwhile, GCA is able to accurately capture the graph structure. Hence, this small experience gives us confidence that GCA is indeed an generalization and improvement on ICA and TCA that performs within our relaxed constraints.

6 Discussion and Future Work

As we have seen from our above simulations, our proposed GCA method is promising in our goal to find a method that extends TCA to more general graphical models. In order to further test the robustness of GCA, we will perform more tests on various types of data (generated from independent sources, trees, larger graphs, etc.) and again compare our results with that of ICA and TCA. Furthermore, we would also like to have a more quantifiable method to compare density estimates and the original density; this can be done via by settling on a distance metric for densities and then calculating the difference empirically. Having a quantifiable metric will allow us to make more concrete comparisons among the methods we are considering.

We will also try to find ways to make our algorithm more efficient; this will certainly be a concern as we test GCA on higher dimensional data and want to ensure that GCA is indeed tractable on larger graphs. Finally, we would also like to analyze the rates of convergence of our algorithm, in a similar fashion to the work done in [Jan15].

7 Acknowledgements

Many thanks to Professor John Lafferty for providing guidance and helping me throughout this project. I would also like to thank Natalie Doss for her work on this project and answering all my questions throughout the process.

References
