Estimating Dependency Structures for non-Gaussian Components with Linear and Energy Correlations

Hiroaki Sasaki[1] hsasaki@cc.uec.ac.jp
Michael U. Gutmann[2] michael.gutmann@helsinki.fi
Hayaru Shouno[1] shouno@uec.ac.jp
Aapo Hyvärinen[3,4] aapo.hyvarinen@helsinki.fi

[1]Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan
[2]Department of Mathematics and Statistics and HIIT, University of Helsinki, Helsinki, Finland
[3]Department of Computer Science and HIIT, University of Helsinki, Helsinki, Finland
[4] ATR Cognitive Mechanisms Laboratories, Kyoto, Japan

Abstract

The statistical dependencies which independent component analysis (ICA) cannot remove often provide rich information beyond the ICA components. It would be very useful to estimate the dependency structure from data. However, most models have concentrated on higher-order correlations such as energy correlations, neglecting linear correlations. Linear correlations might be a strong and informative form of a dependency for some real data sets, but they are usually completely removed by ICA and related methods, and not analyzed at all. In this paper, we propose a probabilistic model of non-Gaussian components which are allowed to have both linear and energy correlations. The dependency structure of the components is explicitly parametrized by a parameter matrix, which defines an undirected graphical model over the latent components. Furthermore, the estimation of the parameter matrix is shown to be particularly simple because using score matching, the objective function is a quadratic form. Using artificial data, we demonstrate that the proposed method is able to estimate non-Gaussian components and their dependency structures, as it is designed to do. When applied to natural images and outputs of simulated complex cells in the primary visual cortex, novel dependencies between the estimated features are discovered.

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

1 Introduction

Recent studies suggest that the statistical dependencies which independent component analysis (ICA) [Comon, 1994, Hyvärinen and Oja, 2000] cannot remove provide rich information. By modeling correlations between squared values (energy correlations) of neighboring components or inside a group of components, topography of the components and phase invariant pooling emerged from natural images [Hyvärinen and Hoyer, 2000, Hyvärinen et al., 2001, Mairal et al., 2011]. Thus, modeling statistical dependencies is important to obtain information beyond the ICA components.

The early work cited above used pre-fixed dependency structures. More recent work is more flexible in the sense that the structure is estimated from data itself. A typical approach is to construct a hierarchical model with ICA-like linear components in the first layer and some parameters in the second layer. The second-layer parameters, when estimated from the data, capture statistical dependencies between the first-layer components. Karklin and Lewicki [2005] proposed a model where the first and second layers consist of linear and density components, respectively. The density components in the model are related to the variances of the linear components. The model provided abstract higher-order structures when it was estimated from natural images. Osindero et al. [2006] proposed a hierarchical topographic model which estimates not only the linear components but also connections among them. A related two-layer model was also proposed by Köster and Hyvärinen [2010].

All the methods mentioned above have concentrated on higher-order statistical dependencies and ignore linear correlations. However, in many practical situations, linear correlations can be observed [Gómez-Herrero et al., 2008, Coen-Cagli et al., 2012]. In fact,

it is possible that in real-data scenarios, the real underlying components are linearly correlated, but ICA is biased towards uncorrelated components which is why such correlations cannot usually be observed in the ICA results. Therefore, it is important to develop methods which incorporate linearly correlated components.

Here, we present a new method to estimate non-Gaussian components and their dependency structures, including linear correlations. The dependency structures are parametrized by a matrix where each off-diagonal element models how strongly a pair of two components is dependent. Thus, the dependency structures can be easily understood and visualized, for example by a graph. Moreover, since the estimation of the components is not biased towards uncorrelated sources, new kinds of underlying components can be estimated. Compared with previous methods, this method is a generalization of ICA and the more recent correlated topographic analysis (CTA) [Sasaki et al., 2013].

Another advantage of the proposed method is the simplicity of the estimation of the dependency parameters. Estimation of hierarchical models is often difficult because the second layer makes the partition function intractable, which results in an intractable likelihood function. Recently, several estimation methods such as contrastive divergence [Hinton, 2002], score matching [Hyvärinen, 2005] and noise contrastive estimation [Gutmann and Hyvärinen, 2012] were proposed to cope with this problem. When score matching is applied to our proposed model, the objective function for the estimation of the dependency parameters takes a quadratic form, and can be optimized by standard methods.

This paper is organized as follows: In Section 2, we formulate a probability distribution for non-Gaussian components and discuss relationships to distributions assumed in previous work. It is argued that the proposed method is a generalization of ICA and CTA. Then, the details for estimating the components and dependency parameters are described. Section 3 shows results for artificial data which demonstrate that the method estimates dependency structures correctly. Application to two kinds of real data, natural images and outputs of simulated complex cells, is the topic of Section 4. Finally, we discuss connections to past work and conclude the paper in Section 5.

2 Estimation of Dependency Structures

2.1 Probabilistic Modeling of Dependency Structures

As in previous work, we suppose that data $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top$ is generated from the linear mixing model:

$$\boldsymbol{x} = \mathbf{A}\boldsymbol{s}, \tag{1}$$

where $\mathbf{A}$ is an unknown square mixing matrix and $\boldsymbol{s} = (s_1, s_2, \ldots, s_d)^\top$ is the source vector of non-Gaussian components. As in ICA, we estimate the model (1) from data. To further model the dependencies of the sources or components, we introduce the following generative model for the sources:

$$\boldsymbol{s} = \boldsymbol{\sigma} \odot \boldsymbol{z}, \tag{2}$$

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_d)^\top$ and $\boldsymbol{z} = (z_1, z_2, \ldots, z_d)^\top$ are the vectors of non-negative and Gaussian random variables, respectively. $\boldsymbol{\sigma}$ and $\boldsymbol{z}$ are statistically independent. As proven in [Hyvärinen et al., 2001], this model generates super-Gaussian components s_i, and statistical dependencies within $\boldsymbol{\sigma}$ or $\boldsymbol{z}$ make the generated components s_i dependent. For example, if σ_i and σ_j have energy correlations (that is, σ_i^2 and σ_j^2 are correlated), and z_i and z_j are linearly correlated, then the generated component s_i is super-Gaussian and has both linear and energy correlations with s_j [Sasaki et al., 2013].

In previous work [Sasaki et al., 2013], a probability distribution for sources having linear and energy correlations was derived as an approximation of the likelihood of the model (2), giving

$$\tilde{p}(\boldsymbol{s}; \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{Z(\boldsymbol{a}, \boldsymbol{b})} \prod_i^d \exp\left(-\sqrt{a_i}|s_i| - \sqrt{b_i}|s_i - s_{i+1}|\right), \tag{3}$$

where $Z(\boldsymbol{a}, \boldsymbol{b})$ is the unknown partition function, and a_i and b_i are non-negative parameters. In (3), the terms $|s_i|$ ensure that the components are super-Gaussian, and the terms $|s_i - s_{i+1}|$ encode statistical dependencies. The dependency structure is, however, limited to topographic neighbors. To remove this limitation, we here extend (3) so that one component s_i can be dependent on several s_j; we propose the following model for the sources:

$$\tilde{p}(\boldsymbol{s}; \mathbf{M}) = \frac{1}{Z(\mathbf{M})} \prod_i^d \exp\left(-m_{i,i}|s_i| - \sum_{j>i} m_{i,j}|s_i - s_j|\right), \tag{4}$$

where $\mathbf{M}$ is a symmetric matrix containing the non-negative parameters $m_{i,j}$. It can be seen that $m_{i,j}$ corresponds to statistical dependency of the two components s_i and s_j. Their effect can be understood in the same way as the precision matrix for a Gaussian distribution: If $m_{i,j} = 0$, s_i and s_j are statistically independent conditioned on all other variables, while they are strongly dependent (given all other variables), if $m_{i,j}$ is large. Moreover, the variables s_j for which $m_{i,j} > 0$ form the Markov blanket of s_i, so that the matrix $\mathbf{M}$ allows us to read out conditional independencies between the components.

We next discuss the basic properties of the probability distribution (4). If, for $j \neq i$, $m_{i,j} \to 0$, $\tilde{p}$ approaches a Laplacian distribution. Laplacian distributions are often used to model components in ICA. If $m_{i,i} \to 1$, $m_{i,i+1} \to 1$ and for $j \neq i, i+1$, $m_{i,j} \to 0$, $\tilde{p}$ approaches the distribution (3) derived in CTA [Sasaki et al., 2013]. Thus, the distribution (4) includes those assumed in ICA and CTA as special cases, and by estimating $\mathbf{M}$ from data, a more general method is obtained.

2.2 Estimation Method

Based on the proposed component model $\tilde{p}(\boldsymbol{s}; \mathbf{M})$, we estimate the dependency parameters $\mathbf{M}$ and the mixing model (1). For any fixed $\mathbf{M}$, the mixing matrix in (1) can be estimated by maximizing the likelihood. The objective function for $\mathbf{W} = \mathbf{A}^{-1}$ to perform maximum likelihood estimation can be derived as

$$J(\mathbf{W}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{d} m_{i,i} G(\boldsymbol{w}_i^\top \boldsymbol{x}(t)) + \sum_{j>i}^{d} m_{i,j} G(\boldsymbol{w}_i^\top \boldsymbol{x}(t) - \boldsymbol{w}_j^\top \boldsymbol{x}(t)) - \log |\det \mathbf{W}| \quad (5)$$

where $\boldsymbol{x}(t)$ is the t-th observation of a data vector and $\boldsymbol{w}_i$ denotes the i-th row vector in $\mathbf{W}$; we replace $|\cdot|$ in (4) by $G(\cdot) = \log\cosh(\cdot)$ for numerical stability. The estimation of the dependency parameters $\mathbf{M}$, in contrast, is difficult because we do not know the partition function $Z(\mathbf{M})$ in (4).

To cope with the problem of an intractable partition function, several estimation methods have been recently proposed. Score matching has a property which is particularly useful for our purposes: For the continuous exponential family, the objective function given by score matching is a quadratic form [Hyvärinen, 2007, Section 4].[1] Based on (4), the proposed distribution $\tilde{p}(\boldsymbol{s}; \mathbf{M})$ can be manipulated to the form of an exponential family,

$$\log \tilde{p}(\boldsymbol{s}; \mathbf{M}) = \sum_{k=1}^{d(d+1)/2} \theta_k F_k(\boldsymbol{s}) - \log Z(\mathbf{M}), \quad (6)$$

where

$$\theta_k = \begin{cases} m_{i,i}, & 1 \leq k \leq d, \\ m_{i,j}, & d+1 \leq k \leq \frac{d(d+1)}{2}, \end{cases} \quad (7)$$

$$F_k(\boldsymbol{s}) = \begin{cases} -G(s_i), & 1 \leq k \leq d, \\ -G(s_i - s_j), & d+1 \leq k \leq \frac{d(d+1)}{2}. \end{cases} \quad (8)$$

The objective function given by score matching is the following quadratic form:

$$J_{SC}(\boldsymbol{\theta}) = \frac{1}{2} \boldsymbol{\theta}^\top E\{\mathbf{K}(\boldsymbol{s}) \mathbf{K}(\boldsymbol{s})^\top\} \boldsymbol{\theta} + \boldsymbol{\theta}^\top \left(\sum_{i=1}^{d} E\{\boldsymbol{h}_i(\boldsymbol{s})\} \right), \quad (9)$$

where E denotes the sample average, the (k,i)-th element in $\mathbf{K}(\boldsymbol{s})$ is $\partial F_k(\boldsymbol{s}) / \partial s_i$, and $\boldsymbol{h}_i(\boldsymbol{s})$ is the i-th column vector in $\mathbf{H}$ whose (k,i)-th element is $\partial^2 F_k(\boldsymbol{s}) / \partial s_i^2$. Note that we further have the constraint that all θ_k are non-negative.

To estimate $\mathbf{W}$ and $\mathbf{M}$, we propose the following optimization algorithm:

Optimization Algorithm for W and M

Input: Data vectors $\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(T)$.

- Initialization: Set $\mathbf{M}$ to the identity matrix, and the elements in $\mathbf{W}$ are randomly determined.
- Alternate between Step 1 and Step 2.

Step 1 Minimize $J(\mathbf{W})$ with respect to $\mathbf{W}$ by the nonlinear conjugate gradient method of [Rasmussen, 2006]. After 10 iterations, normalize each row vector in $\mathbf{W}$ to $\|\boldsymbol{w}_i\| = 1$ and move to Step 2.

Step 2 Minimize the quadratic form $J_{SC}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ under the constraint of non-negativity.

For Step 2, we used the MATLAB function *quadprog.m*. The specific initialization for $\mathbf{M}$ in the algorithm worked well empirically.

[1] If the parameters are not constrained, score matching gives the closed-form solution when the probability distribution belongs to the exponential family. In our case, however, the parameters $m_{i,j}$ are constrained to be non-negative.

3 Simulations on artificial data

In this section, we investigate if the model (1) can be identified by the proposed method and if the estimated parameters in $\mathbf{M}$ correctly reflect the dependency structures in the sources.

3.1 Methods

We generated sources according to the model (2) in two different cases taken from [Sasaki et al., 2013]. Case 1 is an "independent" source where all the components are statistically independent. Case 2 is a "topographic" source where all pairs of only neighboring components, s_i and s_{i+1} for all i, have both linear and energy correlations and the others are independent. The boundary condition is ring-like in this source. As in [Sasaki et al., 2013], for Case 2 sources, the order (topography) of the components can be determined because the dependencies approximately correspond to distances between the components.

Using generated sources $\boldsymbol{s}$, the data $\boldsymbol{x}$ were created from the model (1) where the elements in $\mathbf{A}$ were randomly determined. The only preprocessing step was whitening based on PCA. The dimension of data was $d = 10$, and the total number of samples was $T = 20,000$.

We evaluated the results by the performance matrix $\mathbf{P} = \mathbf{WA}$ where $\mathbf{W}$ is the estimated inverse of $\mathbf{A}$, and by visualizing $\mathbf{M}$. If our estimation was performed correctly, $\mathbf{P}$ should be a permutation matrix and $\mathbf{M}$ should correctly reflect the dependency structure embedded in the sources. The ordering of the rows of $\mathbf{P}$ and $\mathbf{M}$ are coupled: If $\mathbf{P}$ is permuted, that is, if the labeling of the features is changed, the order in the matrix $\mathbf{M}$ changes as well. For the visualization of $\mathbf{P}$, the ordering of the features was changed. We determined the ordering based on the estimated $\mathbf{M}$. The ordering (permutation) algorithm used is a simple greedy algorithm; the details are provided in the supplementary material. Note that the indeterminacy of the permutation is just the same indeterminacy which is always encountered in ICA. It is only a problem for the visualization and validation of the estimation results; it does not change the features and dependency structure learned.

3.2 Results

The results for independent sources (Case 1) are shown in Figure 1. The estimated performance matrix is close to a permutation matrix (Figure 1(a)). Furthermore, $\mathbf{M}$ is a diagonal matrix (Figure 1(b)). Since the sources are statistically independent, these results mean that estimation was performed correctly: the method learned that the sources are all independent.

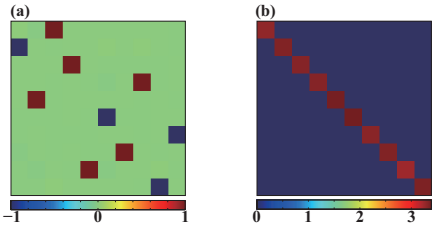

Figure 1: Artificial data, Case 1 (independent sources): (a) Estimated performance matrix $\mathbf{P}$. $\mathbf{P}$ is re-scaled so that its absolute maximum value is one. (b) Dependency matrix $\mathbf{M}$.

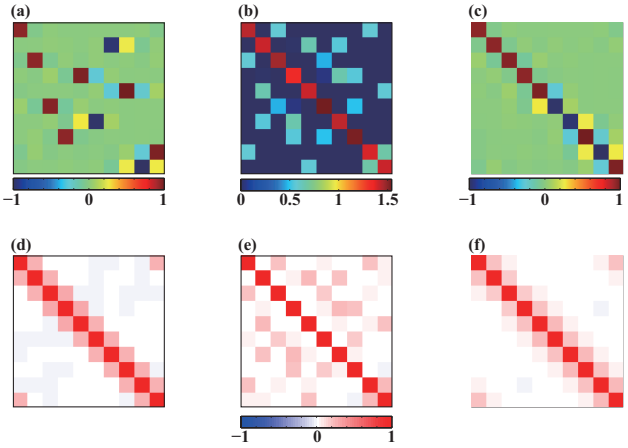

Figure 2: Artificial data, Case 2 (topographic sources): (a) Estimated performance matrix $\mathbf{P}$ and (b) dependency matrix $\mathbf{M}$. (c) Performance matrix, permuted based on $\mathbf{M}$. (d) Correlation matrix of the original sources. (e,f) Correlation matrix of the estimated sources before and after permutation.

The results for topographic sources (Case 2) are shown in Figure 2. Figures 2(a) and (b) visualize the matrix $\mathbf{P}$ and $\mathbf{M}$ before permutation. Figure 2(c) shows $\mathbf{P}$ after permutation based on $\mathbf{M}$. The permuted $\mathbf{P}$ is close to a diagonal matrix, which indicates that the estimate of the features is reasonable. The result is, however, not as good as for Case 1 above in terms of identifiability of the sources. Figures 2(d) and (e) show the linear correlations matrices for the original and estimated sources before permutation. These matrices have different structures. After permutation, however, the linear correlation matrix for the sources is approximately the same as the one for the original sources (Figure 2(d) and (f)). For the energy correlation matrix, the structure was recovered likewise (results not shown). These results mean that the estimated $\mathbf{M}$ contained the information of the topographic depen-

dency structure in the original sources, and that the method worked well.

In summary, the two sets of results indicate that the estimation method works reasonably well and that the estimated $\mathbf{M}$ reflects dependency structures contained in the sources.

4 Application to Real Data

Next, we apply the proposed method to two kinds of real data: natural image patches and outputs of simulated complex cells in the primary visual cortex for natural image inputs.

4.1 Natural Image Patches

ICA-like one-layer and more advanced two-layer models were applied to natural image patches before [Osindero et al., 2006, Hyvärinen et al., 2009, Köster and Hyvärinen, 2010, Gutmann and Hyvärinen, 2012, Sasaki et al., 2013]. Our purpose here is to investigate what kind of inter-relationships between the features the proposed method identifies.

4.1.1 Methods

Natural image patches $\boldsymbol{x}(t)$ of size 12 by 12 pixels were randomly extracted from complete natural scenes.[2] The total number of patches was $T = 100,000$. As preprocessing, first, the DC component of each patch was removed, and then, the norm of each patch was normalized to be one. We further preprocessed the image patches by whitening and dimensionality reduction by PCA. We retained $d = 60$ dimensions.

4.1.2 Results

Groups of basis vectors and the profiles of the rows of $\mathbf{M}$ are presented in Figure 3. Moreover, for a fixed basis vector $\boldsymbol{a}_i$, the basis vectors $\boldsymbol{a}_j$ with the top five largest values of $m_{i,j}$ are shown. Figures 3(a-1,b-1,c-1) indicate that these basis vectors show similar properties in spatial positions or orientations. In addition, the $m_{i,j}$ are sparse (Figure 3(a-3,b-2,c-2)). This means that the Markov blankets in the source space are rather small and formed by the features with similar properties.

In order to compactly visualize all the features and the complete matrix $\mathbf{M}$, we first fitted a Gabor function to each basis vector, and made a line icon representing its spatial position, orientation and length. Second, we pooled the icons with weight $m_{i,j}$, which resulted in the desired compact visualization. Several examples of line icons are shown in Figure 3(a-2). Figures 3(a-4,b-3,c-3) show two kinds of pooling patterns: Pooling similar orientations from spatially distant features (Figure 3(a-4,b-3)) and pooling diverse orientations from spatially close features (Figure 3(c-3)).

Figure 3(d) visualizes the complete set of features and the matrix $\mathbf{M}$ using the icons. Inspection of the figure suggests that the two pooling patterns above are the dominant instances. This is also visible in the undirected graph shown in Figure A in the supplementary material.

To investigate the pooling properties (Markov blankets) more rigorously, we analyzed the relative distance and orientation of the features. Relative distance and orientation are defined as the distance of the center positions and the absolute difference of the orientations between two basis vectors. For relative orientation, the range is from 0 to $\pi/2$. Figure 4(a) shows a scatter plot of the relative distances and orientations. In the figure, to make the points, the top five basis vectors per one basis vector are selected as in Figure 3 and for the five pairs, the relative distance and orientation are computed. Finally, all the points for the relative distance and orientation are summarized in the figure. As the relative distance gets larger, the relative orientation tends to get smaller as well. The slope of the fitted line is clearly negative. This is in agreement with our qualitative analysis above: When the features are spatially close, the learned weights $m_{i,j}$ pool over diverse orientations, and when spatially distant, the pooling occurs over similar orientations.

We further investigated if the model really found linearly correlated components in contrast to energy-correlation based methods [Karklin and Lewicki, 2005, Osindero et al., 2006, Köster and Hyvärinen, 2010]. Figure 4(b) shows a scatter plot for the linear and energy correlation coefficients between the estimated s_i and the corresponding top five s_j selected as in Figure 3. The method did find strongly linearly correlated components. In fact, the average of the linear correlation coefficients for all the points in the figure is 0.669, with standard deviation 0.068. Furthermore, the linear correlation coefficients are correlated with the energy correlation coefficients.

4.2 Outputs of Complex Cells

Next, we applied the method to the outputs of simulated complex cells in the primary visual cortex. ICA and non-negative sparse coding were applied to this kind of data and long-contour features emerged [Hoyer and Hyvärinen, 2002, Hyvärinen et al., 2005]. Recently, a topographic map of these features was learned

[2] We used the natural scenes in the *contournet* MATLAB package, which is available at `http://www.cs.helsinki.fi/u/phoyer/software.html`.

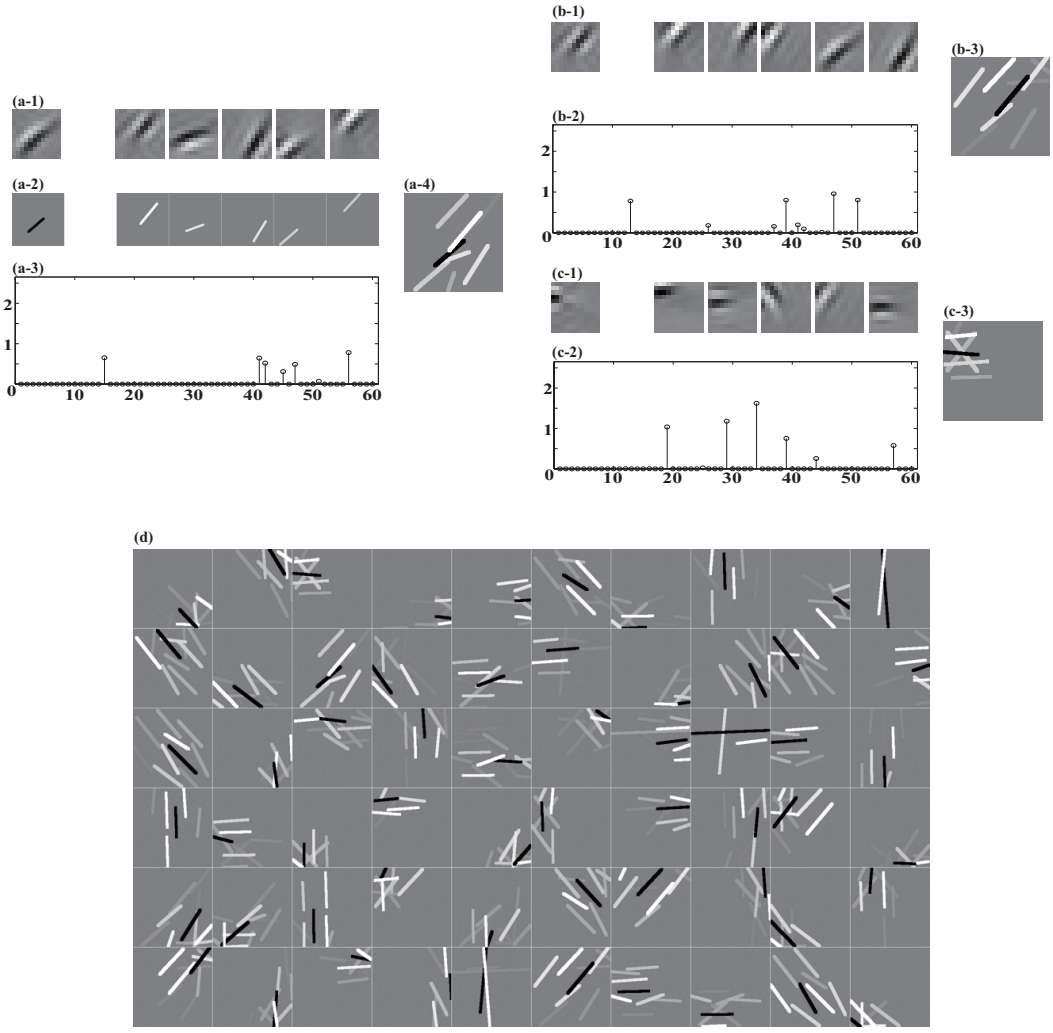

Figure 3: Examples of observed pooling properties. (a-1,b-1,c-1) For the leftmost fixed basis vector $\boldsymbol{a}_i$, the basis vectors $\boldsymbol{a}_j$ with the top five largest values of $m_{i,j}$ are shown. (a-2) Line icons of the basis vectors presented in (a-1). Each line represents spacial position, orientation and length of a basis vector. The color from gray to white indicates the values of $m_{i,j}$ for fixed i. The black line represents the leftmost basis vector $\boldsymbol{a}_i$. (a-3,b-2,c-2) The profile of $m_{i,j}$ for fixed i. (a-4,b-3,c-3) Pooled icons to visualize the learned features and $m_{i,j}$ in a compact manner. (d) Visualization of all learned features and the matrix $\mathbf{M}$.

too [Sasaki et al., 2013]. However, the topographic map was obtained by using a pre-fixed dependency structure. Our purpose here is to relax this assumption and to investigate the inter-relation between the features learned from the outputs of the complex cells.

4.2.1 Methods

The outputs of the simulated complex cells $\boldsymbol{x}$ are computed as

$$x'_k = \left(\sum_{x,y} W^o_k(x,y) I(x,y)\right)^2 + \left(\sum_{x,y} W^e_k(x,y) I(x,y)\right)^2, \quad x_k = \log(x'_k + 1.0), \tag{10}$$

where $I(x,y)$ is a 24×24 natural image patch,[3] and $W^o_k(x,y)$ and $W^e_k(x,y)$ are odd- and even-symmetric Gabor functions with the same spatial positions, orientation and frequency. The total number of outputs is $T = 100,000$. The complex cells exist on a 6 by 6 spatial grid and 4 orientation grid. Thus, there are 144 cells in total.

As preprocessing, the DC component of each $\boldsymbol{x}$ was removed. Then, the norm of $\boldsymbol{x}$ was constrained to be one. Finally, whitening and dimensionality reduction were performed simultaneously by PCA. The retained dimension was $d = 80$.

[3] For the computation of the complex cell outputs, we used again the *contournet* package.

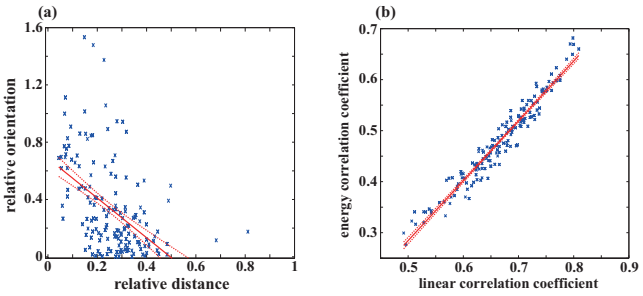

Figure 4: (a) Scatter plot for relative distance and orientation. The solid red line is the fitted regression line (its slope is -1.385). The dotted lines are confidence bounds of the solid one. (b) A comparison of linear and energy correlation coefficients between the estimated sources. In this figure, the coefficients are selected like the five pairs of basis vectors are selected in Figure 3(a-1,b-1,c-1). The same linear fitting was performed as in (a), the slope of the regression line is 1.173. For (a) and (b), the elements relating to the low frequency basis vectors were omitted.

4.2.2 Results

The learned features were qualitatively similar to those found in [Sasaki et al., 2013]. However, learning $m_{i,j}$ unveiled a novel dependency structure which we visualize in Figures 5(a)-(d): There seem to be strong (conditional) dependencies between long contours.

The complete set of the features and the learned dependency structure is displayed using an undirected graph (Figure 6). In this graph, each node represents a feature and each link corresponds to a $m_{i,j}$ which has a value larger than 0.4. Most of the long contour features are positioned on the left part of the graph, while star-like features, which are the spatially localized ones, are on the right part. This result shows that long contours tend to be (conditionally) dependent on each other, and star-like features also have a similar tendency. The features with less clear structures have only weaker links $m_{i,j}$.

It was shown before that the star-like features may not reflect properties of natural images but properties of the fixed complex-cell stage [Sasaki et al., 2013]. With a threshold of 0.4, these features are mostly separated from the longer contours. It is important future work to choose this threshold more objectively using statistical techniques.

5 Discussion and Conclusion

We proposed a new method for the estimation of non-Gaussian components and their dependency structures. The dependency structures are represented by a parameter matrix $\mathbf{M}$ and their interpretation is easy. As described in Section 2.1, this method includes ICA and CTA as special cases depending on the values of the matrix $\mathbf{M}$. In addition, due to a useful property of score matching [Hyvärinen, 2007], $\mathbf{M}$ can be estimated by solving a standard optimization problem.

Dependency structures were modeled in previous work [Hyvärinen and Hoyer, 2000, Hyvärinen et al., 2001, Mairal et al., 2011, Sasaki et al., 2013]. In those methods, the dependency structures were fixed a priori, while the proposed method estimates the structure from data. Thus, our method is more flexible. Other methods have been proposed to estimate dependency structures from data [Karklin and Lewicki, 2005, Osindero et al., 2006, Köster and Hyvärinen, 2010]. However, those methods did not explicitly take into account linear correlations of the sources, while our method is able to estimate linearly correlated components. Further related work is the estimation of tree-dependent structures [Bach and Jordan, 2003, Zoran and Weiss, 2009]. However, these methods restrict graphs to have a tree-structure. In our method, no such a restriction is imposed.

In simulations for artificial data in Section 3, we demonstrated that the dependency parameters can be correctly estimated for different kinds of sources: For independent sources (Case 1), the estimated $\mathbf{M}$ was a diagonal matrix and for topographic sources (Case 2), the order (topography) of the components was recovered from $\mathbf{M}$. As a limitation, estimation accuracy for the dependent topographic sources was worse than for the independent sources. An important task for the future is to address this shortcoming.

In simulations with natural images, interesting pooling properties were observed: Pooling various orientations from spatially nearby features and pooling similar orientations from far features. This might be because natural images contain many long contours, textures and junctions. Detailed investigation of this point is an important point for future work. For simulated complex cells, ICA, non-negative sparse coding, and CTA were applied in previous work [Hoyer and Hyvärinen, 2002, Hyvärinen et al., 2005, Sasaki et al., 2013]. While the features estimated by our method are similar as those in previous work, the estimated matrix $\mathbf{M}$ revealed interesting relationships between these features.

Typical ICA methods remove linear correlations alto-

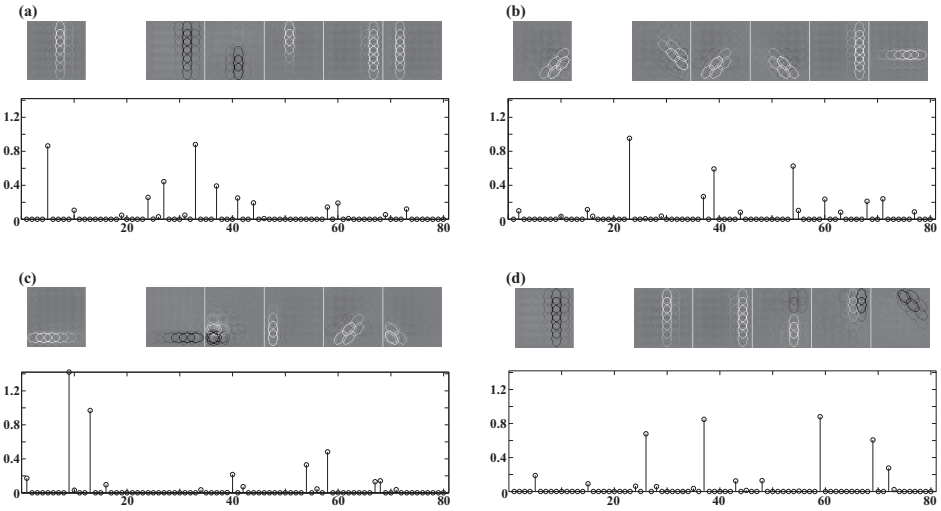

Figure 5: Basis vectors $\boldsymbol{a}_j$ with the five largest $m_{i,j}$ for the leftmost fixed basis vector $\boldsymbol{a}_i$. The stem plots below the basis vectors visualize the values $m_{i,j}$ for fixed i. We see that the strongest (conditional) dependencies occur among long contours, typically of the same orientation.

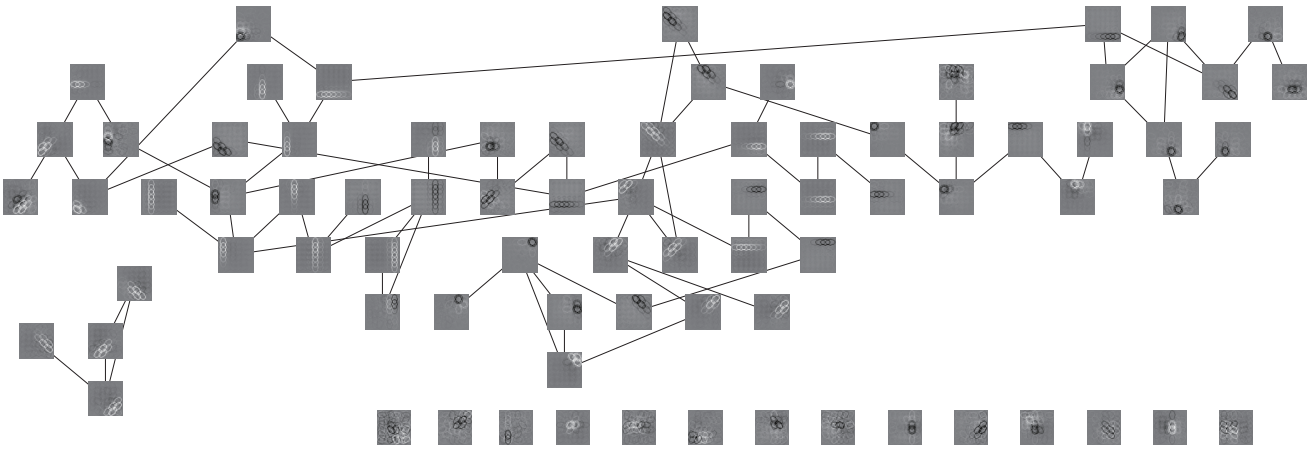

Figure 6: An undirected graph for the estimated features. Each node corresponds to a feature, and only the links which have $m_{i,j}$ larger than 0.4 are displayed. The features aligned in the lower part have no links $m_{i,j}$ larger than 0.4.

gether, thus leaving no room for their further analysis. This is in contrast to energy correlations, which remain even after ICA, and could in many cases be estimated and analyzed after ordinary ICA. Here, we have demonstrated that if the components are allowed to be linearly correlated, the learned dependency structure is not trivial at least for some real data sets. The linear correlation thus provides interesting information which could not have been obtained without a method which allows such correlations in the first place.

Acknowledgements

H. Sasaki was supported by JSPS KAKENHI Grant Number 23·7556 (Grant-in-Aid for JSPS Fellows). M. U. Gutmann acknowledges financial support by the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170). H. Shouno was supported by MEXT/JSPS KAKENHI Grant Numbers 50263231, 21103001. A. Hyvärinen was supported by the Academy of Finland, Computational Science Program and the Finnish Centre-of-Excellence in Algorithmic Data Analysis.

References

F. Bach and M. Jordan. Beyond independent components: trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

R. Coen-Cagli, P. Dayan, and O. Schwartz. Cortical surround interactions and perceptual salience via natural scene statistics. *PLoS Computational Biology*, 8(3):e1002405, 2012.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

G. Gómez-Herrero, M. Atienza, K. Egiazarian, and J. Cantero. Measuring directional coupling between EEG sources. *Neuroimage*, 43(3):497–508, 2008.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.

G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8): 1771–1800, 2002.

P. Hoyer and A. Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593–1605, 2002.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51(5): 2499–2512, 2007.

A. Hyvärinen and P. Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

A. Hyvärinen, P. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

A. Hyvärinen, M. Gutmann, and P. Hoyer. Statistical model of natural stimuli predicts edge-like pooling of spatial frequency channels in V2. *BMC Neuroscience*, 6:12, 2005.

A. Hyvärinen, J. Hurri, and P. Hoyer. *Natural Image Statistics: A probabilistic approach to early computational vision.* Springer-Verlag, 2009.

Y. Karklin and M. Lewicki. A hierarchical Bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural Computation*, 17(2):397–423, 2005.

U. Köster and A. Hyvärinen. A two-layer model of natural stimuli estimated with score matching. *Neural Computation*, 22(9):2308–2333, 2010.

J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Convex and network flow optimization for structured sparsity. *Journal of Machine Learning Research*, 12:2681–2720, 2011.

S. Osindero, M. Welling, and G. Hinton. Topographic product models applied to natural scene statistics. *Neural Computation*, 18(2):381–414, 2006.

C. Rasmussen. Conjugate gradient algorithm, version 2006-09-08. 2006.

H. Sasaki, M. Gutmann, H. Shouno, and A. Hyvärinen. Correlated topographic analysis: estimating an ordering of correlated components. *Machine Learning*, 92(2-3):285–317, 2013.

D. Zoran and Y. Weiss. The "Tree-Dependent Components" of natural images are edge filters. In *Advances in Neural Information Processing Systems*, volume 22, pages 2340–2348, 2009.

Supplementary Material for "Estimating Dependency Structures for non-Gaussian Components with Linear and Energy Correlations"

Hiroaki Sasaki[1] hsasaki@cc.uec.ac.jp
Michael U. Gutmann[2] michael.gutmann@helsinki.fi
Hayaru Shouno[1] shouno@uec.ac.jp
Aapo Hyvärinen[3,4] aapo.hyvarinen@helsinki.fi

[1]Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan
[2]Department of Mathematics and Statistics and HIIT, University of Helsinki, Helsinki, Finland
[3]Department of Computer Science and HIIT, University of Helsinki, Helsinki Finland
[4] ATR Cognitive Mechanisms Laboratories, Kyoto, Japan

A Details for the Permutation Algorithm

Here, we describe the permutation algorithm used in Section 3. The goal of this algorithm is to estimate an order vector $\boldsymbol{k} = (k_1, k_2, \ldots, k_d)$ where $k_i \in \{1, 2, \ldots, d\}$ so that $\mathbf{M}$ in Figure 2(b) approaches a tridiagonal matrix. The algorithm is as follows:

Greedy Permutation Algorithm

Input: $\mathbf{W}$ and $\mathbf{M}$.

- Initialization: Set $\hat{k}_1 = 1$ and the remaining index set $\mathbf{I} = \{2, \ldots, d\}$, and make a matrix with zero diagonal elements by $\mathbf{M}' = \mathbf{M} - \mathbf{M}_{diag}$ where $\mathbf{M}_{diag}$ denotes the diagonal matrix whose diagonal elements are the diagonal ones in $\mathbf{M}$.
- Repeat the following procedures from $i = 2$ to $i = d$:
  - Find $\hat{k}_i$ by
$$\hat{k}_i = \arg\max_{j \in \mathbf{I}} m'_{\hat{k}_{i-1}, j} \tag{A1}$$
where $m'_{\hat{k}_{i-1},j}$ denotes the $(\hat{k}_{i-1}, j)$-th element in $\mathbf{M}'$.
  - Update $\mathbf{I}$ by removing $\hat{k}_i$:
$$\mathbf{I} \leftarrow \mathbf{I} \setminus \hat{k}_i. \tag{A2}$$

Output: $\hat{\boldsymbol{k}} = (\hat{k}_1, \hat{k}_2, \ldots, \hat{k}_d)$.

By (A1), $m_{\hat{k}_{i-1}, \hat{k}_i}$ for all i are made to take a large value, and thus, $\hat{M}$, whose (i, j)-th element is $m_{\hat{k}_i, \hat{k}_j}$, should be close to a tridiagonal matrix. Using $\hat{\boldsymbol{k}}$, the row vectors in $\mathbf{W}$ were permuted as $\mathbf{W}_p = (\boldsymbol{w}_{\hat{k}_1}, \boldsymbol{w}_{\hat{k}_2}, \ldots, \boldsymbol{w}_{\hat{k}_d})^\top$ where $\boldsymbol{w}_i$ denotes the i-th row vector in $\mathbf{W}$, and in Figure 2(c) and (d), we visualized the performance matrix as $\mathbf{W}_p\mathbf{A}$ and correlation matrix for the permuted sources $\mathbf{W}_p\boldsymbol{x}$.

B Undirected Graph for Natural Images

This section presents an undirected graph for natural images as in Figure 6 for the outputs of simulated complex cells. The graph is depicted in Figure A. It shows that features with similar orientations or positions tend to be (conditionally) dependent.

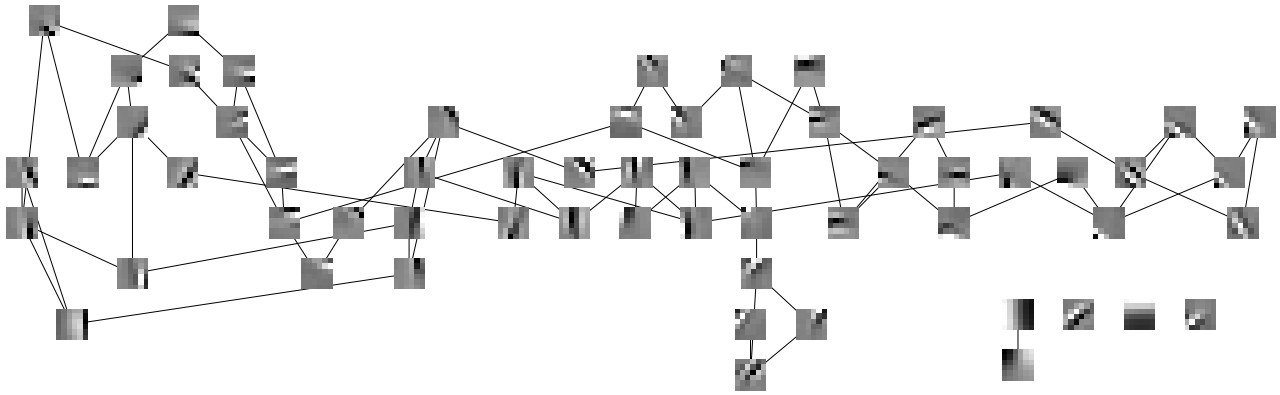

Figure A: An undirected graph for natural images. Each node corresponds to a feature estimated from natural images, and only the links which have $m_{i,j}$ larger than 0.8 are displayed.