Mach Learn (2013) 92:285–317
DOI 10.1007/s10994-013-5351-x

Correlated topographic analysis: estimating an ordering of correlated components

Hiroaki Sasaki · Michael U. Gutmann · Hayaru Shouno · Aapo Hyvärinen

Received: 10 January 2013 / Accepted: 8 April 2013 / Published online: 14 May 2013

Abstract This paper describes a novel method, which we call correlated topographic analysis (CTA), to estimate non-Gaussian components and their ordering (topography). The method is inspired by a central motivation of recent variants of independent component analysis (ICA), namely, to make use of the residual statistical dependency which ICA cannot remove. We assume that components nearby on the topographic arrangement have both linear and energy correlations, while far-away components are statistically independent. We use these dependencies to fix the ordering of the components. We start by proposing the generative model for the components. Then, we derive an approximation of the likelihood based on the model. Furthermore, since gradient methods tend to get stuck in local optima, we propose a three-step optimization method which dramatically improves topographic estimation. Using simulated data, we show that CTA estimates an ordering of the components and generalizes a previous method in terms of topography estimation. Finally, to demonstrate that CTA is widely applicable, we learn topographic representations for three kinds of real data: natural images, outputs of simulated complex cells and text data.

Editors: Zhi-Hua Zhou, Wee Sun Lee, Steven Hoi, Wray Buntine, and Hiroshi Motoda.

H. Sasaki (✉) · H. Shouno
Graduate School of Informatics and Engineering, The University of Electro-Communications,
1-5-1 Chofugoaka, Chofu, Tokyo, 182-8585, Japan
e-mail: hsasaki@cc.uec.ac.jp

H. Shouno
e-mail: shouno@uec.ac.jp

M.U. Gutmann
Department of Mathematics and Statistics, Helsinki Institute for Information Technology HIIT,
University of Helsinki, Helsinki, Finland
e-mail: michael.gutmann@helsinki.fi

A. Hyvärinen
Department of Mathematics and Statistics, Department of Computer Science, Helsinki Institute for
Information Technology HIIT, University of Helsinki, Helsinki, Finland
e-mail: aapo.hyvarinen@helsinki.fi

Keywords Independent component analysis · Topographic representation · Natural image statistics · Higher order features · Natural language processing

1 Introduction

Many recent methods to analyze multidimensional data $\mathbf{x} = (x_1, \ldots, x_d)^\top$ are based on the linear mixing model

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

where $\mathbf{A}$ is the mixing matrix and $\mathbf{s} = (s_1, \ldots, s_d)^\top$ is the source vector of non-Gaussian latent variables. A special instance of (1) is independent component analysis (ICA) where all the components in $\mathbf{s}$ are statistically independent (Hyvärinen and Oja 2000). The goal of ICA and related methods is to estimate both $\mathbf{A}$ and $\mathbf{s}$ from observations of $\mathbf{x}$ only. The model (1) in ICA was proven to be identifiable up to the order, signs, and scales of the components (Comon 1994). ICA has been used in a wide range of fields such as computational neuroscience (Hyvärinen et al. 2009), natural language processing (Honkela et al. 2010) and MEG/EEG analysis (Vigário et al. 2000).

However, real data do often not follow the assumptions made in ICA. For instance, the components in $\mathbf{s}$ may not be statistically independent. When such components are estimated with ICA, statistical dependencies between the estimates can be observed, in violation of the independence assumption made. For natural images, for example, the conditional variance of an estimated component s_i may depend on the value of another component s_j: As $|s_j|$ increases, the conditional variance of s_i grows. This means that the conditional distribution of s_i becomes wider as $|s_j|$ increases, which gives the conditional histogram a characteristic bow-tie like shape (Simoncelli 1999; Karklin and Lewicki 2005). An alternative formulation of this dependency is energy correlation, $\operatorname{cov}(s_i^2, s_j^2) > 0$: both s_i^2 and s_j^2 tend to be co-active (Hyvärinen et al. 2009).

Therefore, it seems important to relax the independence assumption. Topographic ICA (TICA) is based on this idea (Hyvärinen et al. 2001). The key point of TICA is to arrange the components on an one- or two-dimensional grid or lattice, and allow nearby components to have energy correlations, while far-away components are assumed statistically independent. Thus, energy correlations define the proximity of the components and can be used to fix their ordering. Osindero et al. (2006) proposed another related method and their results for natural image data were similar to those obtained with TICA, although their estimations were overcomplete in contrast to the ones in TICA. Karklin and Lewicki (2005) proposed a hierarchical model where the second layer learns variance components. Further related work includes tree-like modeling of the dependencies of the components (Bach and Jordan 2003; Zoran and Weiss 2009).

The components in TICA are constrained to be linearly uncorrelated. However, uncorrelated components are not always optimal. In fact, both linear and energy correlations can be observed in many practical situations. Consider the outputs of two collinearly aligned Gabor-like filters. As natural images often contain long edges, their outputs have both linear and energy correlations (Coen-Cagli et al. 2012). Such linear correlations make the conditional histogram of the outputs have a tilted bow-tie like shape. Coherent sources in MEG or EEG data can be linearly correlated too, due to neural interactions (Gómez-Herrero et al. 2008). As we will see later, another example occurs in the analysis of text data.

In this paper, we propose a new statistical method which we call correlated topographic analysis (CTA). In CTA, topographically nearby components have linear and energy correlations, and those dependencies are used to fix the ordering as in TICA. Since CTA is sensitive to both kinds of correlations, only one kind (linear or energy) needs to exist in the data. CTA thus generalizes TICA for topography estimation.

In addition to proposing the statistical model of CTA, we propose an optimization method that performs better than standard procedures in terms of local optima. This method dramatically improves topography estimation, and we verify its performance on simulated as well as real data.

This paper is organized as follows. Section 2 motivates the estimation of topographic representations, and presents the new statistical method CTA. CTA is introduced as a special case of a more general framework which also includes ICA and TICA. In Sect. 3, we use simulated data to verify identifiability of the linear mixing model in (1) for sources with various dependency structures, and compare the performances of ICA, TICA and CTA. In Sect. 4, CTA is applied on three kinds of real data: natural images, outputs of simulated complex cells and text data. The applicability on such a wide range of data sets suggests that CTA may be widely applicable. Connections to previous work are discussed in Sect. 5. Section 6 concludes this paper.

2 Correlated topographic analysis

We start by motivating the estimation of topographic representations. Then, we introduce a generative model for the sources **s** in order to model ICA, TICA and CTA in a unified way, and describe the basic properties of the components in CTA. We then derive an approximation of the likelihood for CTA and propose a method for its optimization.

2.1 Motivation for estimating topographic representations

The foremost motivation for estimating topographic representations is visualization. Plotting the components with the topographic arrangement enables us to easily see the interrelationships between components. This is particularly true if the topographic grid is two dimensional and can thus be plotted on the plane.

A second motivation is that the topography learned from natural inputs such as natural images, natural sound, or text, might model cortical representations in the brain. This is based on the hypothesis that in order to minimize wiring length, neurons which interact with each other should be close to each other, see e.g. Hyvärinen et al. (2009). Minimizing wiring seems to be important to keep the volume of the brain manageable, and possibly to speed up computation as well.

An example is computation of complex cell outputs based on simple cell outputs in primary visual cortex (V1). Simple cells are sensitive to an oriented bar or an edge at a certain location in visual space, while complex cells are otherwise similar, but invariant to local sinusoidal phases of visual stimuli. Computationally, such a conversion can be achieved by pooling the squares of the outputs of the simple cells which have similar orientation and spatial location, but different phases. A topographic representation where simple cells are arranged as observed in V1 could minimize the wiring needed in such a pooling because the pooling is done over nearby cells. Such a minimum-wiring topography was found to emerge from natural images using TICA (Hyvärinen et al. 2001).

Related to minimum wiring, the topography may also enable simple definition of new, higher-order features. Summation of the features in a topographic neighborhood (possibly after a nonlinearity such as squaring) may even in general lead to interesting new features, just as in the case of simple cell pooling explained above.

2.2 The generative model

We begin with the following generative model for the latent source vector $\mathbf{s}$ in (1),

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

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_d)$ and $\mathbf{z} = (z_1, \dots, z_d)$ are statistically independent. The two key points of the generative model (2) are the following:

1. If $\mathbf{z}$ is multivariate Gaussian with mean $\mathbf{0}$ and the elements in $\boldsymbol{\sigma}$ are positive random variables, which is what we assume in the following, the components in $\mathbf{s}$ are super-Gaussian, i.e., sparse (Hyvärinen et al. 2001).
2. By introducing linear correlations in $\mathbf{z}$ and/or energy correlations in $\boldsymbol{\sigma}$, the components in $\mathbf{s}$ will have linear and/or energy correlations. This point will be made more precise in the following.

A special case of the model in (2) results in ICA:

Case 1 If all the elements in $\mathbf{z}$ and $\boldsymbol{\sigma}$ are statistically independent, then $\mathbf{s}$ is a vector with independent sparse sources, and (2) gives the source model of ICA.

The source model of TICA can also be obtained as a special case:

Case 2 If all the elements in $\mathbf{z}$ are uncorrelated, but the squares of nearby elements in $\boldsymbol{\sigma}$ are correlated, then $\mathbf{s}$ is a vector formed by sparse sources with energy correlations (and no linear correlations) within a certain neighborhood, and thus (2) gives the source model of TICA.

Here, we introduce the following two further cases:

Case 3 If nearby elements in $\mathbf{z}$ are correlated, but all the elements in $\boldsymbol{\sigma}$ are statistically independent, then $\mathbf{s}$ is a sparse source vector whose elements have linear correlations (and zero or weak energy correlations) within a certain neighborhood.

Case 4 If nearby elements in $\mathbf{z}$ and the squares of nearby elements in $\boldsymbol{\sigma}$ are correlated, then $\mathbf{s}$ is a sparse source vector whose elements have linear and energy correlations within a certain neighborhood, and (2) gives the source model of CTA.

The statistical dependencies of the above four cases for $\boldsymbol{\sigma}$ and $\mathbf{z}$ are summarized in Table 1.

In the following, we concentrate on Case 4 (both energy and linear correlations). We do not explicitly consider Case 3 (linear correlations only), but we will show below with simulations that CTA identifies its sources and estimates the ordering of the components as well. This is natural since the model in Case 4 uses both linear and energy correlations to model topography, while Case 3 uses linear ones only.

Table 1 Dependencies of pairs of nearby elements in $\boldsymbol{\sigma}$ and $\mathbf{z}$ on four cases of sources and the corresponding source model

	Case 1	Case 2	Case 3	Case 4
$\boldsymbol{\sigma}$	$\operatorname{cov}(\sigma_i^2, \sigma_j^2) = 0$	$\operatorname{cov}(\sigma_i^2, \sigma_j^2) \neq 0$	$\operatorname{cov}(\sigma_i^2, \sigma_j^2) = 0$	$\operatorname{cov}(\sigma_i^2, \sigma_j^2) \neq 0$
$\mathbf{z}$	$\operatorname{cov}(z_i, z_j) = 0$	$\operatorname{cov}(z_i, z_j) = 0$	$\operatorname{cov}(z_i, z_j) \neq 0$	$\operatorname{cov}(z_i, z_j) \neq 0$
Model	ICA	TICA	not explicitly considered	CTA

2.3 Basic properties of the model

We give here basic properties of the CTA generative model (Case 4 above) and discuss the differences to TICA (Case 2). Regarding the mean, linear correlation and energy correlation in the model, the following can be shown in general:

- The mean values of all the components are zero.

$$E\{s_i\} = E\{\sigma_i\}E\{z_i\} = 0. \tag{3}$$

- Nearby components, s_i and s_j, are correlated if and only if z_i and z_j are linearly correlated. From the property (3), this is proven by

$$\mathrm{cov}(s_i, s_j) = E\{\sigma_i\sigma_j\}\underbrace{E\{z_i z_j\}}_{\mathrm{cov}(z_i,z_j)} = E\{\sigma_i\sigma_j\}\mathrm{cov}(z_i, z_j). \tag{4}$$

  Thus, $\mathrm{cov}(s_i, s_j)$ is the same as $\mathrm{cov}(z_i, z_j)$ up to the positive multiplication factor $E\{\sigma_i\sigma_j\}$. The linear correlation coefficient of the components has an upper bound (Appendix A).
- The energy correlation for s_i and s_j can be computed as

$$\begin{aligned}\mathrm{cov}\big(s_i^2, s_j^2\big) &= E\big\{s_i^2 s_j^2\big\} - E\big\{s_i^2\big\}E\big\{s_j^2\big\},\\ &= E\big\{\sigma_i^2\sigma_j^2\big\}E\big\{z_i^2 z_j^2\big\} - E\big\{\sigma_i^2\big\}E\big\{\sigma_j^2\big\}E\big\{z_i^2\big\}E\big\{z_j^2\big\},\\ &= E\big\{z_i^2\big\}E\big\{z_j^2\big\}\mathrm{cov}\big(\sigma_i^2, \sigma_j^2\big) + 2E\big\{\sigma_i^2\sigma_j^2\big\}\mathrm{cov}(z_i, z_j)^2, \end{aligned}\tag{5}$$

  where we used the formula valid for Gaussian variables with zero means, $E\{z_i^2 z_j^2\} = E\{z_i^2\}E\{z_j^2\} + 2E\{z_i z_j\}^2$ which is proven by Isserlis' theorem (Isserlis 1918; Michalowicz et al. 2009). From (5), the energy correlation is caused by the energy correlation for $\boldsymbol{\sigma}$ and the squared linear correlation for $\mathbf{z}$. Thus, to prove that $\mathrm{cov}(s_i^2, s_j^2) > 0$, it is enough to prove that $\mathrm{cov}(\sigma_i^2, \sigma_j^2) > 0$. In the literature of TICA (Hyvärinen et al. 2001), $\mathrm{cov}(\sigma_i^2, \sigma_j^2)$ is conjectured to be positive when each σ_i takes the following form,

$$\sigma_i = \phi_i\Bigg(\sum_{j \in N(i)} u_{i+j}\Bigg), \tag{6}$$

  where $N(i)$ is an index set to determine a certain neighborhood, $\phi_i(\cdot)$ denotes a monotonic nonlinear function and u_i is a positive random variable. We follow this conjecture. The energy correlation coefficient of the components has also an upper bound (Appendix A).

The same analysis has been done for the TICA generative model (Case 2) in Hyvärinen et al. (2001). In the model, the sources are linearly uncorrelated, and, regarding energy correlation, only the first term in (5) is nonzero because the elements in $\mathbf{z}$ are statistically independent. Thus, compared to TICA, in CTA, there exist linear correlations and the energy correlations are stronger as well.

2.4 Probability distribution and its approximation

We derive here a probability distribution for $\mathbf{s}$ to estimate the CTA generative model. We make the assumption that the precision matrix $\boldsymbol{\Lambda}$ of $\mathbf{z}$ takes a tridiagonal form, and thus, the distribution of $\mathbf{z}$ is given by

$$p(\mathbf{z};\boldsymbol{\Lambda}) = \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\exp\left(-\frac{1}{2}\mathbf{z}^\top\boldsymbol{\Lambda}\mathbf{z}\right),$$
$$= \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\prod_{i=1}^{d}\exp\left\{-\frac{1}{2}\left(z_i^2+2\lambda_i z_i z_{i+1}\right)\right\}, \tag{7}$$

where the boundary of z_i is ringlike, i.e., $z_{i\pm d}=z_i$. All the diagonal elements in $\boldsymbol{\Lambda}$ are 1, the $(i,i+1)$-th elements are denoted by λ_i and the others are 0. For $\boldsymbol{\sigma}$, we suppose that each element is given by

$$\sigma_i = (u_{i-1}+u_i+v_i)^{-1/2}, \tag{8}$$

where u_i and v_i are independent positive random variables and statistically independent from each other. Such a mixture of u_{i-1} and u_i creates energy correlations in the source vector $\mathbf{s}$, while v_i generates a source-specific variance. By assuming (8), we follow the conjecture in TICA that energy correlations are positive, as in (6). We assume inverse Gamma distributions for $\mathbf{u}$ and $\mathbf{v}$,

$$p(\mathbf{v},\mathbf{u};\mathbf{a},\mathbf{b}) = \prod_{i=1}^{d}\sqrt{\frac{a_i}{2\pi}}v_i^{-3/2}\exp\left(-\frac{a_i}{2v_i}\right)\times\prod_{i=1}^{d}\frac{b_i}{2}u_i^{-2}\exp\left(-\frac{b_i}{2u_i}\right). \tag{9}$$

The a_i and b_i are positive scale parameters. If a scale parameter approaches zero, the corresponding variable converges to zero in the sense of distribution. For example, if $b_i \to 0$ for all i, the u_i approach zero, which decouples the σ_i from each other. A sketch of the process which generates the sources $\mathbf{s}$ and data $\mathbf{x}$ is depicted in Fig. 1.

Inserting (2) into (7) gives the conditional distribution for $\mathbf{s}$ given $\boldsymbol{\sigma}$,

$$p(\mathbf{s}|\boldsymbol{\sigma};\boldsymbol{\lambda}) = \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\prod_{i=1}^{d}\frac{1}{\sigma_i}\exp\left\{-\frac{1}{2}\left(\frac{s_i^2}{\sigma_i^2}+2\lambda_i\frac{s_i s_{i+1}}{\sigma_i\sigma_{i+1}}\right)\right\}. \tag{10}$$

We show in Appendix B that Eq. (8) transforms (10) as

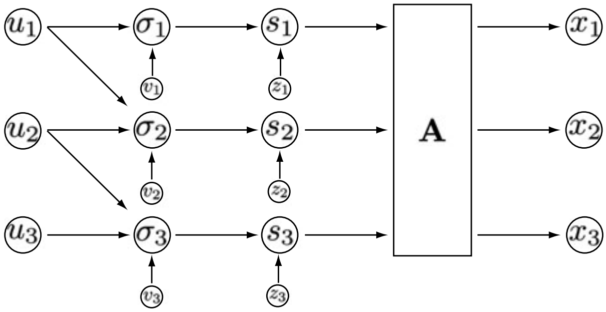


Fig. 1 A sketch of the process generating data $\mathbf{x}$. Adjacent elements of $\mathbf{z}$ are not statistically independent

$$
\begin{aligned}
p(\mathbf{s}|\mathbf{v},\mathbf{u};\boldsymbol{\lambda}) &= \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\prod_{i=1}^{d}\sqrt{u_{i-1}+u_i+v_i}\exp\Big[-\frac{1}{2}\big\{(u_{i-1}+u_i+v_i)s_i^2 \\
&\quad + 2\lambda_i\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})}s_i s_{i+1}\big\}\Big], \\
&= \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\prod_{i=1}^{d}\sqrt{u_{i-1}+u_i+v_i}\exp\Big[-\frac{1}{2}\big\{s_i^2 v_i + \big(s_i^2+s_{i+1}^2\big)u_i \\
&\quad + 2\lambda_i s_i s_{i+1}\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})}\big\}\Big].
\end{aligned} \tag{11}
$$

To obtain the distribution for $\mathbf{s}$, we need to integrate out $\mathbf{u}$ and $\mathbf{v}$ in (11) using (9) as prior distributions. However, this seems to be intractable. Therefore, we resort to two approximations,

$$\sqrt{u_{i-1}+u_i+v_i} \approx c_i\sqrt{u_i}, \tag{12}$$

$$\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})} \approx d_i u_i, \tag{13}$$

where c_i and d_i are two unknown positive scaling parameters which do not depend on $\mathbf{u}$ and $\mathbf{v}$. The above approximations are similar to what has been done for TICA (Hyvärinen et al. 2001, Eq. (3.7)). Below we analyze the implications of these approximations. With (12) and (13), an approximation of (11) is

$$
\begin{aligned}
\tilde{p}(\mathbf{s}|\mathbf{v},\mathbf{u};\boldsymbol{\lambda}) &= \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\prod_{i=1}^{d}c_i\sqrt{u_i}\exp\Big[-\frac{1}{2}\big\{s_i^2 v_i + \big(s_i^2+s_{i+1}^2+2\lambda_i d_i s_i s_{i+1}\big)u_i\big\}\Big], \\
&\propto \prod_{i=1}^{d}\sqrt{u_i}\exp\Big[-\frac{1}{2}\big\{s_i^2 v_i + \big(s_i^2+s_{i+1}^2+2\lambda_i d_i s_i s_{i+1}\big)u_i\big\}\Big],
\end{aligned} \tag{14}
$$

where we dropped terms not depending on $\mathbf{s}$, $\mathbf{v}$, or $\mathbf{u}$. The additional parameters c_i from (12) do not affect the functional form of the approximation. The parameters d_i from (13) and λ_i occur only as a product. We thus replace them by the new parameter $\varrho_i = \lambda_i d_i$. By calculating the integral over $\mathbf{u}$ and $\mathbf{v}$, see Appendix C for details, we obtain the following approximation for the probability distribution of $\mathbf{s}$,

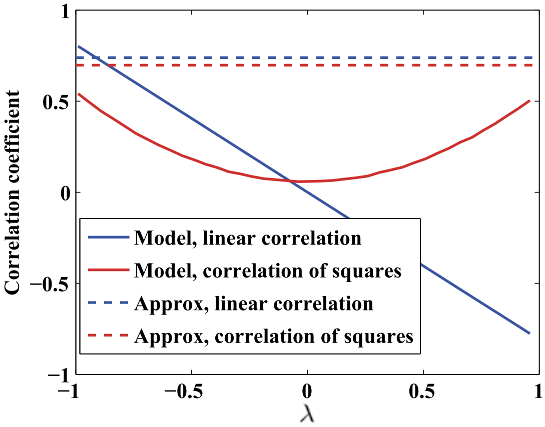
$$\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b}) \propto \prod_i \exp\Big(-\sqrt{a_i}|s_i| - \sqrt{b_i}\sqrt{s_i^2+s_{i+1}^2+2\varrho_i s_i s_{i+1}}\Big). \tag{15}$$

We use the proportionality sign because we do not know the partition function which normalizes $\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b})$.

The approximation in (15) relates to ICA, TICA, and CTA as follows: In the limit where $b_i \to 0$, $\tilde{p}$ becomes the Laplace distribution, as often used in ICA with sparse sources (Case 1). In the limit where $a_i \to 0$ and $\varrho_i = 0$ for all i, we obtain TICA (Case 2). Using the fixed values $a_i = b_i = 1$ and $\varrho_i = -1$, we obtain

$$\tilde{p}(\mathbf{s}) \propto \prod_{i=1}^{d}\exp\big(-|s_i| - |s_i - s_{i+1}|\big), \tag{16}$$

Fig. 2 Comparing the generative model in (2) with the approximation in (16) in terms of their correlation structure. The *blue* and *red solid curves* show the correlation coefficient of the components and their squared values, respectively, for the generative model. The *horizontal dashed lines* show the correlation coefficients for the approximation. We find that the approximation has qualitatively similar correlation coefficients as the generative model for a λ close to −1 (Color figure online)



which serves as approximative distribution for the CTA model (Case 4) with positively correlated sources, as we justify in more detail below. Note that this distribution has been previously used as a prior for the regression coefficients in the fused lasso for supervised learning (Tibshirani et al. 2005). However, our application on modeling latent variables is very different.

2.5 Accuracy of the approximation

The two approximations (12) and (13) were used to derive (16). To analyze the implications of these approximations, we compared (16) with the generative model in (2) in terms of correlation and sparsity of the sources.

For the comparison, we sampled from (2) using $d = 2$ sources and the fixed values $a_i = b_i = 1$ for different values of $\lambda_i = \lambda$. We sampled from (16), with $d = 2$, using slice sampling.[1] For both models, we drew 10^6 samples to compute the correlation coefficient between the two sources, the correlation coefficient between their squared values, and their kurtosis.

Figure 2 shows the correlation coefficients for (2) as a function of λ (curves with solid lines), and the correlation coefficients for the approximation (16) as dashed horizontal lines. The plot suggests that the approximation has qualitatively similar correlation coefficients as the generative model for a λ close to −1.

For the generative model, we found that the (excess) kurtosis of the sources was independent of λ, with a value around 3.4. For the approximation, we obtained a value around 2.1. This means that both the original model and the approximation yield sparse sources.

To conclude, we confirmed that the approximation has qualitatively similar properties as the generative model for a λ close to −1. The limitations of the approximation are that the sources are more strongly energy correlated but less sparse than in the original generative model for λ close to −1.

2.6 Objective function and its optimization

Using the approximative distribution (16), we can compute the log-likelihood for $\mathbf{x}$ and obtain the following objective function to estimate the parameter matrix $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_d)^\top =$

[1] We used MATLAB's `slicesample.m` with a burn-in period of 50,000 samples.

$\mathbf{A}^{-1}$:

$$J(\mathbf{W}) = J_1(\mathbf{W}) + J_2(\mathbf{W}), \tag{17}$$

$$J_1(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d} G\big(\mathbf{w}_i^\top \mathbf{x}(t)\big) + \log|\det \mathbf{W}|, \tag{18}$$

$$J_2(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d} G\big(\mathbf{w}_i^\top \mathbf{x}(t) - \mathbf{w}_{i+1}^\top \mathbf{x}(t)\big), \tag{19}$$

where we replaced $|\cdot|$ with $G(\cdot) = \log\cosh(\cdot)$ for numerical reasons. The vector $\mathbf{x}(t)$ denotes the t-th observation of the data, $t = 1, 2, \dots, T$. Note that J_1 is the log-likelihood for an ICA model and that J_2 models the topographic part, being sensitive to the order as well as the signs of the $\mathbf{w}_i$.

We now describe a method to optimize the objective function in (17) because basic gradient methods tend to get stuck in local maxima as we will see in the next section. The proposed algorithm includes the following three steps:

Algorithm 1: Three-Step Optimization

1. Maximize $J_1(\mathbf{W})$ only, based on a conjugate gradient method (Rasmussen 2006) as

$$\mathbf{W}^{(1)} = \arg\max_{\mathbf{W}} J_1(\mathbf{W}). \tag{20}$$

2. Compute $\mathbf{s}^{(1)}(t) = \mathbf{W}^{(1)}\mathbf{x}(t)$. Optimize the order and signs of $s_i^{(1)}(t)$ in J_2 as

$$\mathbf{k}^*, \mathbf{c}^* = \arg\max_{\mathbf{k},\mathbf{c}} \left[-\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d} G\big(c_i s_{k_i}^{(1)} - c_{i+1} s_{k_{i+1}}^{(1)}\big) \right], \tag{21}$$

   where $\mathbf{k} = (k_1, \dots, k_d)$ is an index vector with $k_i \in \{1, \dots, d\}$ and $k_i \neq k_j$ for $i \neq j$, and $\mathbf{c} = (c_1, \dots, c_d)$ denotes a sign vector: $c_i \in \{-1, 1\}$. The vectors $\mathbf{k}^*$ and $\mathbf{c}^*$ transform $\mathbf{W}^{(1)}$ to $\mathbf{W}^{(2)} = (c_1^* \mathbf{w}_{k_1^*}^{(1)}, \dots, c_d^* \mathbf{w}_{k_d^*}^{(1)})^\top$ where $\mathbf{w}_i^{(1)}$ denotes the i-th row vector in $\mathbf{W}^{(1)}$. This optimization will be done by Algorithm 2 given below.
3. Maximization of $J(\mathbf{W})$ using $\mathbf{W}^{(2)}$ as the initial values on $\mathbf{W}$.

$$\mathbf{W}^{(3)} = \arg\max_{\mathbf{W}} J(\mathbf{W}), \tag{22}$$

   where the conjugate gradient method in Step 1 is applied again.

The final output of the algorithm is $\mathbf{W}^{(3)}$. Step 1 corresponds to performing ICA, and Step 2 gives the optimal order and the optimal signs of the ICA components in the sense of the objective function J_2. In Step 3, $\mathbf{W}^{(2)}$ is used as initial value of $\mathbf{W}$. Therefore, Step 1 and Step 2 can be interpreted as a way to find a good initial value.

In Step 2, we have to solve a combinatorial optimization problem, which is computationally very difficult. However, we can see that the problem (21) has a nestedness property, in other words, we can divide the main problem into subproblems. So we can efficiently solve it. For example, suppose $c_1 = 1$ and $k_1 = 1$. When we want to find the optimal c_2 and k_2

given these c_1 and k_1, we end up with solving a smaller subproblem, which is to maximize the two terms, $f_2(k_3, c_3) = \arg\max_{k_2, c_2}[-\frac{1}{T}\sum_{t=1}^{T}\{G(s_1^{(1)} - c_2 s_{k_2}^{(1)}) + G(c_2 s_{k_2}^{(1)} - c_3 s_{k_3}^{(1)})\}]$ because the other terms do not include k_2 and c_2. Then, we can reuse $f_2(k_3, c_3)$ in finding the optimal c_3 and k_3. Under this situation, dynamic programming (DP) (Bellman 1957; Bellman and Dreyfus 1962; Held and Karp 1962) is one efficient optimization method. The description of the resulting DP algorithm is as follows:

Algorithm 2: Finding the optimal order and signs

Input: ICA components, $\mathbf{s}^{(1)}(1), \mathbf{s}^{(1)}(2), \ldots, \mathbf{s}^{(1)}(T)$

1. Initialization: Fix the index and sign of the first component, $c_1 = 1$ and $k_1 = 1$, and compute and store $f_1(k_2, c_2) = -\frac{1}{T}\sum_{t=1}^{T} G(s_1^{(1)}(t) - c_2 s_{k_2}^{(1)}(t))$ as a table. (The index and sign of the first component can be arbitrarily fixed.)
2. Repeat the computation of the tables for the conditionally optimal indices, signs and values to the subproblems: maximizing the subsets of the objective function in (21) from $i = 2$ to $i = d - 1$:
   (a) Compute

$$h(k_i, c_i, k_{i+1}, c_{i+1}) = -\frac{1}{T}\sum_{t=1}^{T} G\big(c_i s_{k_i}^{(1)}(t) - c_{i+1} s_{k_{i+1}}^{(1)}(t)\big) \tag{23}$$

   for all possible combinations of (k_i, c_i) and (k_{i+1}, c_{i+1}).

   (b) Compute the conditionally optimal values of subsets of the objective function $f_i(k_{i+1}, c_{i+1})$ and store those values in tabular form,

$$f_i(k_{i+1}, c_{i+1}) = \max_{k_i, c_i}\big[f_{i-1}(k_i, c_i) + h(k_i, c_i, k_{i+1}, c_{i+1})\big]. \tag{24}$$

   Simultaneously, create the tables for the conditionally optimal indices and signs:

$$\hat{k}_i(k_{i+1}, c_{i+1}), \hat{c}_i(k_{i+1}, c_{i+1}) = \arg\max_{k_i, c_i}\big[f_{i-1}(k_i, c_i) + h(k_i, c_i, k_{i+1}, c_{i+1})\big], \tag{25}$$

   where $k_{i+1} \neq k_i$ and $k_{i+1}, k_i \notin \{\hat{k}_{i-1}(k_i, c_i), \ldots, \hat{k}_2(\hat{k}_3, \hat{c}_3)\}$.
3. Compute the optimal index and sign of the last component by

$$k_d^*, c_d^* = \arg\max_{k_d, c_d}\left[f_{d-1}(k_d, c_d) - \frac{1}{T}\sum_{t=1}^{T} G\big(c_d s_{k_d}^{(1)}(t) - s_1^{(1)}(t)\big)\right]. \tag{26}$$

4. Sequentially find the optimal indices k_i^* and signs c_i^* from $i = d - 1$ to $i = 2$ by using the tables,

$$\begin{aligned} k_{d-1}^* &= \hat{k}_{d-1}\big(k_d^*, c_d^*\big), & c_{d-1}^* &= \hat{c}_{d-1}\big(k_d^*, c_d^*\big), \\ k_{d-2}^* &= \hat{k}_{d-2}\big(k_{d-1}^*, c_{d-1}^*\big), & c_{d-2}^* &= \hat{c}_{d-2}\big(k_{d-1}^*, c_{d-1}^*\big), \\ &\vdots & &\vdots \\ k_2^* &= \hat{k}_2\big(k_3^*, c_3^*\big) & c_2^* &= \hat{c}_2\big(k_3^*, c_3^*\big) \end{aligned}$$

Output: The optimal indices (order) $\mathbf{k}^* = (1, k_2^*, \ldots, k_d^*)$ and signs $\mathbf{c}^* = (1, c_2^*, \ldots, c_d^*)$.

The last term in the right-hand side of (26) was added because of the ring-like boundary. The MATLAB package of CTA by which several results presented in this paper can be reproduced is available at http://www.cs.helsinki.fi/u/ahyvarin/code/cta.

We now briefly describe the run-time cost of the optimization. When data is high-dimensional, most of the time is spent on the dynamic programming part (Algorithm 2). The computation of (23) is T times additions, and the additions are repeated $4(d-i+1)(d-i)$ times to make the i-th table (24). This means that the computational cost for addition is approximately $O(4T\sum_{i=2}^{d-1}(d-i+1)(d-i)) = O(Td^3)$. Thus, as the dimension of the data increases, more computational time is needed. But, as we will see below, this algorithm dramatically improves results in terms of topography estimation.

3 Identifying simulated sources

In this section, we investigate how the objective function in (17) can be used to estimate the model (1) with sources generated according to the four cases outlined in the previous section, and compare the performances of ICA, TICA and CTA.

3.1 Methods

We generated sources $\mathbf{s}$ according to the four cases of (2). We sampled $\mathbf{z}$ from a Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{C}$: In Case 3 and Case 4, all the diagonal elements are 1, the $(i, i+1)$-th element $c_{i,i+1} (= c_{i+1,i})$ is 0.4 with a ring-like boundary, and the other elements are 0. In Case 1 and Case 2, $\mathbf{C}$ is the identity matrix. For $\boldsymbol{\sigma}$, each element in Case 2 and Case 4 is generated as $\sigma_i = r_{i-1} + r_i + r_{i+1}$ where r_i is sampled from the exponential distribution with mean 1. In Case 1 and Case 3, $\sigma_i = r_i$. After generating $\mathbf{s}$, the mean and variance of all the components s_i are standardized to zero and one, respectively. The dimension of $\mathbf{s}$ and the number of samples are $d = 20$ and $T = 30{,}000$, respectively.

For the generated sources of Case 3, we verified that the energy correlation was very weak: the mean of the energy correlation coefficient in s_1^2 and s_2^2 and its standard deviation in 100 source sets were 0.0192 and 0.0102, respectively.

Then, the data $\mathbf{x}$ was generated from the model (1) where the elements of $\mathbf{A}$ were randomly sampled from the standard normal distribution. The preprocessing consisted of whitening based on PCA.

For the estimation of ICA, we perform only Step 1 in Sect. 2.6. For TICA, Step 1 is performed as in CTA, but the objective functions in Step 2 and Step 3 are replaced by

$$J_{TICA}(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d}\sqrt{0.1 + \left(\mathbf{w}_i\mathbf{x}(t)\right)^2 + \left(\mathbf{w}_{i+1}\mathbf{x}(t)\right)^2} + \log|\det \mathbf{W}|. \tag{27}$$

In Step 2 for TICA, we do not optimize the signs of the components because (27) is insensitive to the change of signs. However, we do optimize the ordering using DP, and thus the TICA algorithm used here is an improved version of the original algorithm by Hyvärinen et al. (2001) in terms of topography estimation.

We visualize the estimation results by showing the performance matrix $\mathbf{P} = \mathbf{WA}$. If the estimation of the ordering is correct, $\mathbf{P}$ should be close to a diagonal matrix, or a circularly shifted diagonal matrix because of the ring-like boundary.

3.2 Results

We first show the effectiveness of our three-step optimization method in optimizing J. Then, we show the results of the comparison between ICA, TICA and CTA.

3.2.1 *Escaping from local maxima*

To clarify the necessity of the optimization method described in Sect. 2.6, we first show the result obtained by the conjugate gradient method only, which is equivalent to performing only Step 3 with a random initial value on $\mathbf{W}$. A performance matrix $\mathbf{P}$ for sources generated according to Case 4 is shown in Fig. 3(a). Obviously, $\mathbf{P}$ is different from a (shifted) diagonal matrix. This means that the order of the estimated components is almost random, and that the estimation is incorrect. To clarify the situation, we conducted an additional experiment where $\mathbf{W}$ was initialized with the true matrix $\mathbf{A}^{-1}$. The resulting matrix $\mathbf{P}$ is shown in Fig. 3(b): $\mathbf{P}$ is much closer to the identity matrix and a quite good estimate. Then, we compared the values of the objective function J for the two initial conditions, the numbers are shown below Fig. 3(a) and (b). The comparison shows that the bad solution was a local maximum, and that we thus need an optimization method to escape from it.

A simple approach to escape from local maxima would be to permute the estimated components. However, such a permutation changes the structure of the covariance matrix, and thus provides a bad fit to the model, decreasing the objective function. In fact, as can be seen in Fig. 4(a) and (b), the structure of the covariance matrices for the original source vector and its estimate in the local maximum are qualitatively similar. Instead of permuting at the end, we empirically found it useful to permute the components at the beginning, after an initial estimation with ICA. The performance matrix $\mathbf{P} = \mathbf{W}^{(1)}\mathbf{A}$ obtained by using ICA (performing only Step 1 in Sect. 2.6) is shown in Fig. 5(a). For Fig. 5(b), the order of the row vectors in $\mathbf{W}^{(1)}$ was manually determined so that the maximum element on each row of $\mathbf{P}$ is located on a (shifted) diagonal. For (c), their signs were also changed manually. A comparison of the values of J_2, shown in Fig. 5,[2] indicates that changing also the signs

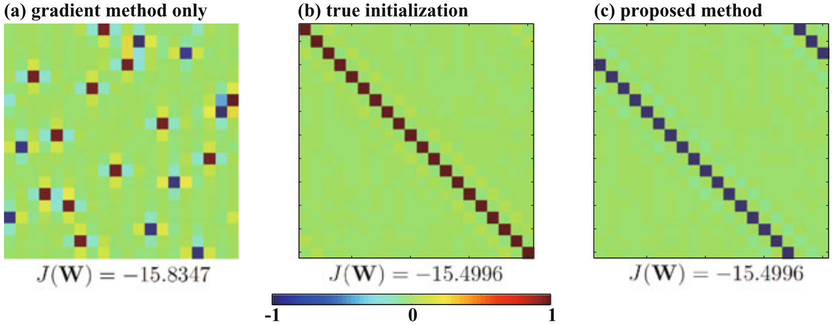


Fig. 3 Performance matrices from (**a**) the conjugate gradient method only, (**b**) the true initialization and (**c**) the proposed optimization method. All performance matrices are normalized by the absolute maximum value in each $\mathbf{P}$, and the data are generated using Case 4 sources. The value of the objective function $J(\mathbf{W})$ in (17) is denoted below each figure (Color figure online)

[2] J_1 is insensitive to any change of the order and signs of the components. Therefore, we computed only J_2 instead of J.

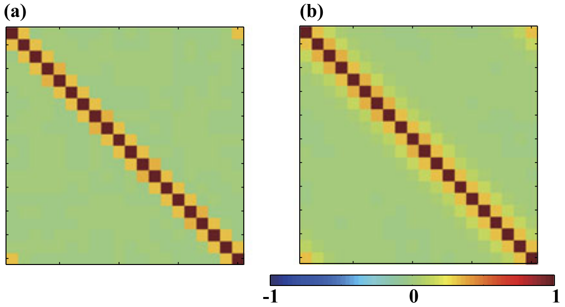


Fig. 4 Covariance matrices of (**a**) the original components; and components estimated by (**b**) the conjugate gradient method only and by (**c**) the proposed optimization method. The components are standardized (Color figure online)

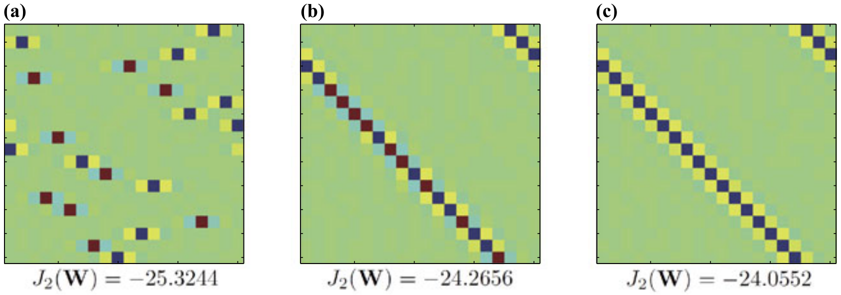


Fig. 5 (**a**) The performance matrix obtained by ICA. (**b**) The performance matrix permuted manually so that the maximum absolute value in each row is on the diagonal, and (**c**) the signs in the matrix are changed as well. The value of $J_2(\mathbf{W})$ in (19) is shown below each figure (Color figure online)

increases the objective function. This evidence strongly suggests that we should optimize not only the order, but also signs of the components estimated by ICA. This motivates the three-step optimization method in Sect. 2.6.

Figure 3(c) shows the result when the three-step optimization method is applied to our example. The performance matrix is close to a shifted identity matrix, and the value of the objective function equals the one in Fig. 3(b). This means that our estimation is performed correctly. Furthermore, note that the signs of the diagonal elements of $\mathbf{P}$ in Fig. 3(c) all agree. This means that CTA solves also the sign indeterminacy problem in ICA. This is impossible for TICA because the objective function (27) is insensitive to the signs of the components.

3.2.2 Comparison to ICA and TICA

Next, we perform 100 trials for each of the four cases of sources, and compare the performance of the three methods.

To quantify how well the components are estimated, we use the Amari index (AI) (Amari et al. 1996). To further investigate how well the topography was estimated, we define a topography index (TI). To compute TI, like for AI, we first normalize $\mathbf{P}$ in order to take the scale indeterminacy of ICA into account. After taking the absolute values of all the elements in $\mathbf{P}$, each row and column is divided by its maximum value which gives the matrices $|\mathbf{P}_1|$ and $|\mathbf{P}_2|$, respectively. Next, we compute the sums over all possible shifted diagonals in $|\mathbf{P}_1|$ and $|\mathbf{P}_2|$, and extract the maximum values, which are denoted by S_1 and S_2. Examples of

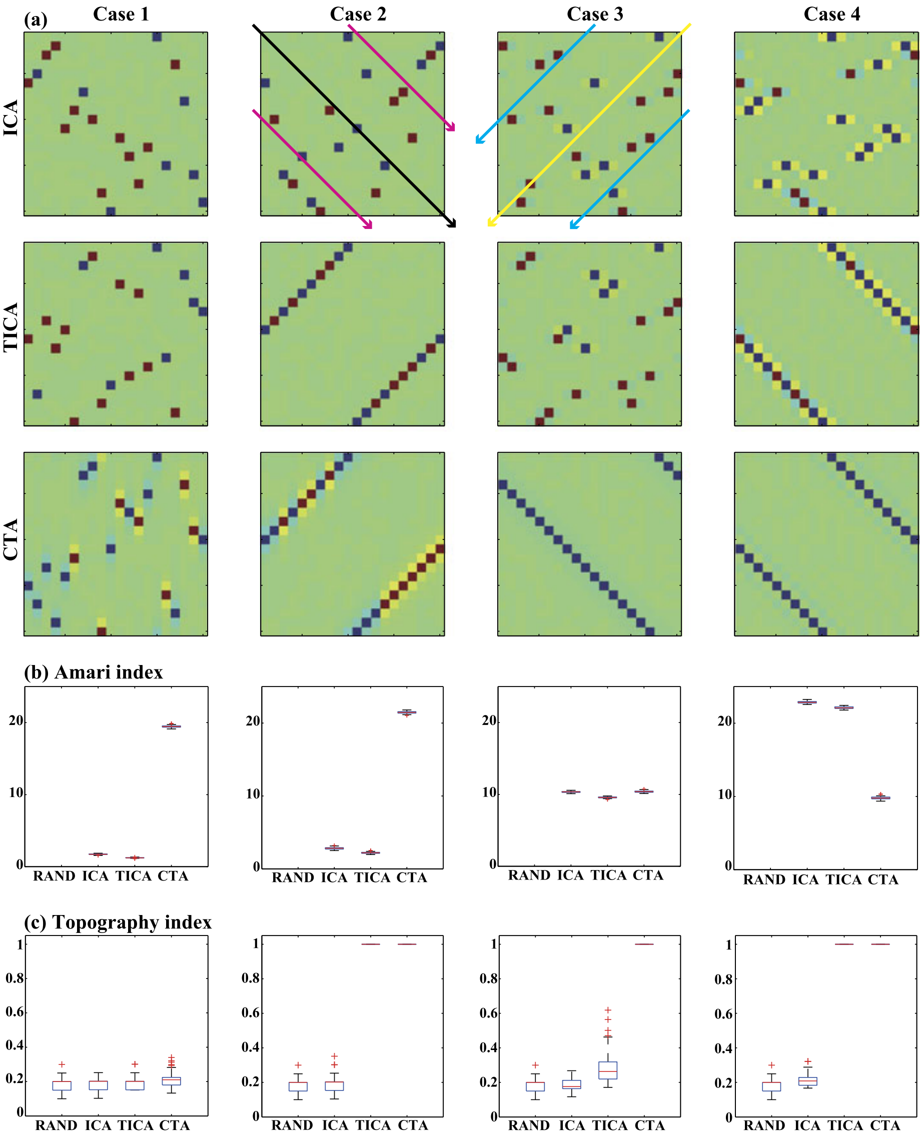


Fig. 6 (**a**) Examples of performance matrices for ICA, TICA and CTA in the four cases of sources. The *arrows* in the figure represent examples of circularly shifted diagonal trajectories in the computation of the topography index. (**b**) and (**c**) depict box plots of Amari index and topography index, respectively, obtained in 100 trials. RAND in (**b**) and (**c**) gives the baseline obtained using 100 random matrices and 100 random permutation matrices, respectively. Amari index for RAND was around 250. For Amari index, smaller means better performance. For topography index, larger means better (Color figure online)

shifted diagonal paths along which we compute the sums are depicted in Fig. 6(a). TI is finally given by

$$\mathrm{TI} = \frac{S_1 + S_2}{2d}. \tag{28}$$

Matrices which show the best performance, giving TI $= 1$, are diagonal or circularly shifted diagonal ones.

Performance matrices for one of the 100 trials are presented in Fig. 6(a). CTA shows the best performance for sources from Case 2 to Case 4 for topography estimation. TICA cannot estimate the topography for Case 3. Regarding AI (Fig. 6(b)), CTA is not as good as ICA and TICA in Case 1 and Case 2. This is presumably because CTA forces the estimated components to be correlated even if they are not. For Case 3 and Case 4, CTA shows almost the same or a better performance than ICA and TICA. Regarding TI (Fig. 6(c)), only CTA can estimate the ordering of the components in all three topographic cases (Case 2, Case 3 and Case 4). TICA cannot estimate the topography for Case 3. We conclude that CTA shows the best performance among the three methods and generalizes TICA for topography estimation. The performance of CTA is weaker in the case of sources with no linear correlations in terms of identifiability, but it is at least as good as ICA or TICA in the case of sources with linear ones.

4 Application to real data

In this section, CTA is applied to three kinds of real data: natural images, outputs of simulated complex cells in V1, and text data.

For natural images and outputs of complex cells, the objective function in (17) is extended to a two-dimensional lattice so that a component is dependent with eight adjacent components. The extended objective function is given by

$$J(\mathbf{W}) = J_1(\mathbf{W}) + J_2(\mathbf{W}), \tag{29}$$

$$J_1(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d_y}\sum_{j=1}^{d_x} G\big(\mathbf{w}_{i,j}^\top \mathbf{x}(t)\big) + \log|\det \mathbf{W}|, \tag{30}$$

$$J_2(\mathbf{W}) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{d_y}\sum_{j=1}^{d_x}\big[G\big(\mathbf{ds}_{i,j}^{rh}(t)\big) + G\big(\mathbf{ds}_{i,j}^{lv}(t)\big) + G\big(\mathbf{ds}_{i,j}^{ll}(t)\big) + G\big(\mathbf{ds}_{i,j}^{lr}(t)\big)\big], \tag{31}$$

where $\mathbf{w}_{i,j}$ represents the row vector in $\mathbf{W}$ that corresponds to the component at position (i, j) on the two-dimensional lattice. Further, $\mathbf{ds}_{i,j}^{rh} = \mathbf{w}_{i,j}^\top\mathbf{x}(t) - \mathbf{w}_{i,j+1}^\top\mathbf{x}(t)$, $\mathbf{ds}_{i,j}^{lv} = \mathbf{w}_{i,j}^\top\mathbf{x}(t) - \mathbf{w}_{i+1,j}^\top\mathbf{x}(t)$, $\mathbf{ds}_{i,j}^{ll} = \mathbf{w}_{i,j}^\top\mathbf{x}(t) - \mathbf{w}_{i+1,j-1}^\top\mathbf{x}(t)$, and $\mathbf{ds}_{i,j}^{lr} = \mathbf{w}_{i,j}^\top\mathbf{x}(t) - \mathbf{w}_{i+1,j+1}^\top\mathbf{x}(t)$ are the differences to the right horizontal, lower vertical, lower left and lower right component, respectively. The optimization method in Sect. 2.6 was modified according to this extension: we extended Step 2 for the two-dimensional lattice and used (29) as the objective function in Step 3. Details about the extension of Step 2 can be found in Appendix D.

4.1 Natural images

Here, we apply CTA to natural image patches.

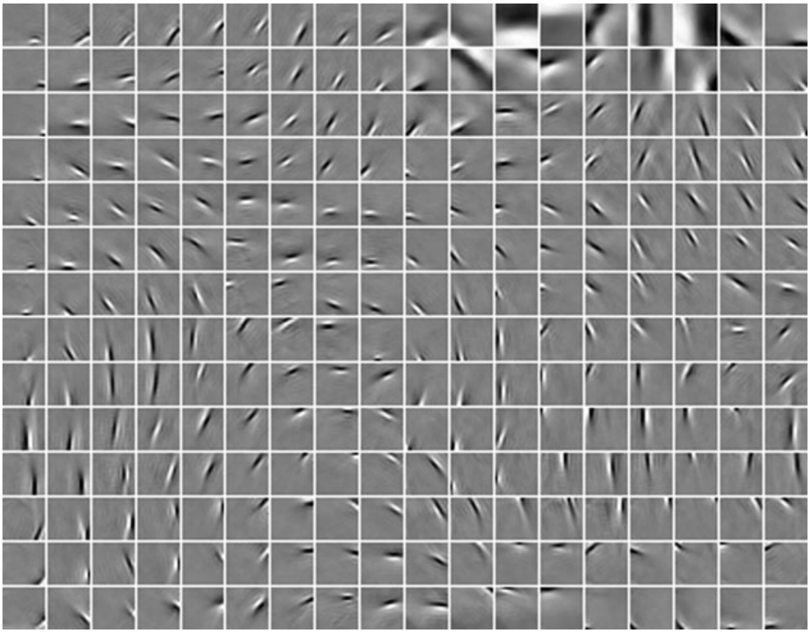


Fig. 7 Estimated basis vectors from natural image patches

4.1.1 *Methods*

The data $\mathbf{x}(t)$ are 20 by 20 image patches which are extracted from natural images.[3] The total number of patches is 100,000. As preprocessing, the DC component of each patch is removed, and whitening and dimensionality reduction are performed by PCA. We retain 252 dimensions.

4.1.2 *Results*

The estimated basis vectors are presented in Fig. 7. Each basis vector has spatially localized, oriented and band-pass like properties as seen in previous work (Olshausen and Field 1996; Bell and Sejnowski 1997). Furthermore, there is a clear topographic organization; similar basis vectors tend to be close to each other. A similar topographic relation can be seen in TICA (Hyvärinen and Hoyer 2001).

To quantify the similarity between adjacent basis vectors and clarify the difference to TICA, we fitted Gabor functions to each basis vector. The scatter plots of the fitted Gabor parameters for pairs of adjacent basis vectors are depicted in Fig. 8. Spatial locations and orientation have strong correlations (Fig. 8(a), (b) and (c)). A large portion of basis vectors prefers high frequency (Fig. 8(d)). These results show that adjacent basis vectors have similar properties. Quite similar results were obtained by TICA (Hyvärinen and Hoyer 2001). The phase parameter however shows a clear difference. Figure 8(e) shows that there seems to exist four clusters in the scatter plot for the phases. This is in contrast to TICA where

[3] The natural images here were taken from the software package associated with the book (Hyvärinen et al. 2009), available at http://www.naturalimagestatistics.net.

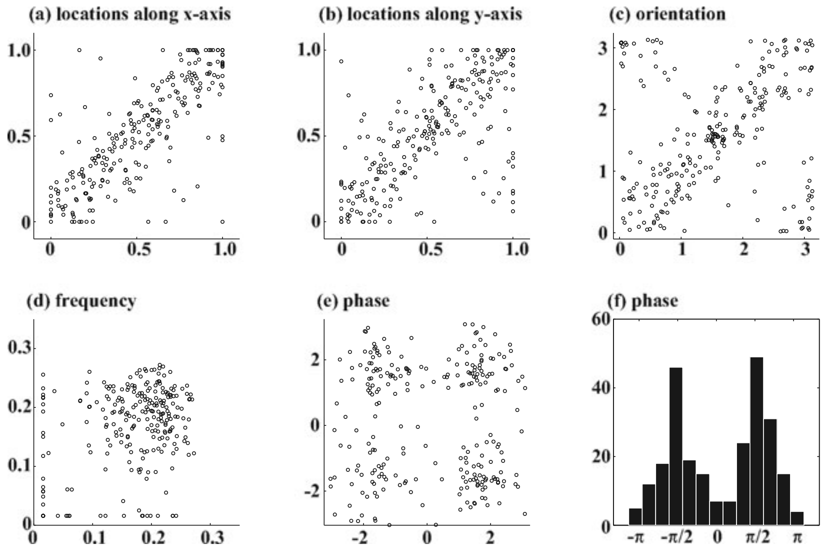


Fig. 8 Scatter plots of fitted Gabor parameters for pairs of adjacent basis vectors. (**a**) and (**b**) show the spatial location along x- and y-axis. (**c**) shows the orientation, (**d**) the spatial frequency and (**e**) the phase. (**f**) Histogram of the phase parameter

there is no clear structure in the scatter plot (Hyvärinen and Hoyer 2001, Fig. 5). In fact, the phase parameters are dominantly $\pm\pi/2$ (Fig. 8(f)). This result means that most of the basis vectors have odd-symmetric spatial patterns, i.e., they represent edges, instead of bars.

4.2 Simulated complex cells

Next, CTA is applied to the outputs of simulated complex cells in V1 when stimulated with natural images. ICA and its related methods have been applied to this kind of data before (Hoyer and Hyvärinen 2002; Hyvärinen et al. 2005). Our purpose here is to investigate what kind of topography emerges for the learned higher-order basis vectors.

4.2.1 Methods

The output of a complex cell x_k is computed by the energy model:[4]

$$x'_k = \left(\sum_{x,y} W_k^o(x,y)I(x,y)\right)^2 + \left(\sum_{x,y} W_k^e(x,y)I(x,y)\right)^2, \tag{32}$$

$$x_k = \log\bigl(x'_k + 1.0\bigr), \tag{33}$$

where $I(x,y)$ is a 24 by 24 natural image patch, and $W_k^o(x,y)$ and $W_k^e(x,y)$ are even and odd symmetric Gabor functions with the same parameters except for their phases. The total number of the patches is $T = 100{,}000$. The complex cells are arranged on a two-dimensional 6 by 6 grid, and at each point, there are cells with four different orientations

[4] The *contournet* MATLAB package is used to compute the outputs of complex cells and available at http://www.cs.helsinki.fi/u/phoyer/software.html.

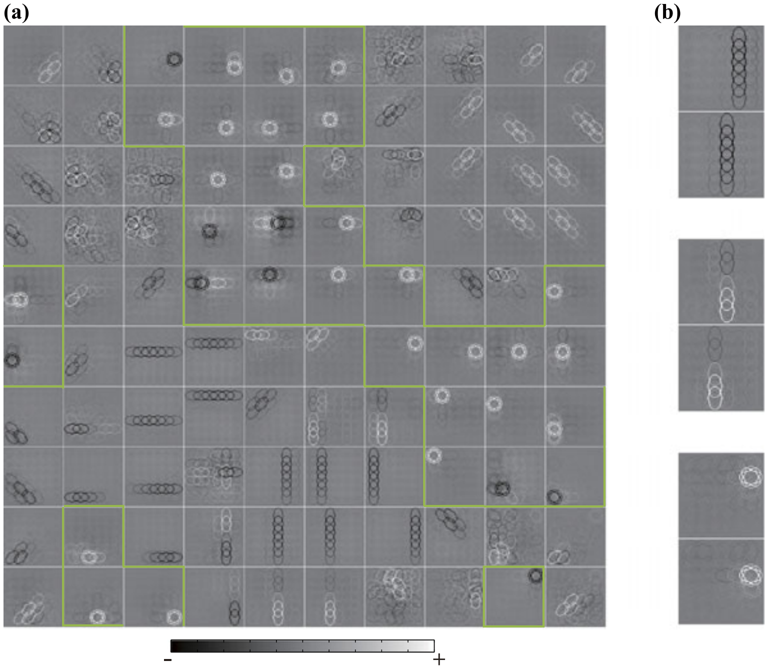


Fig. 9 (**a**) Higher order basis obtained from natural images. Note that the boundary condition of the map is ring-like. (**b**) Prominent features in (**a**), which are long contours, end-stopping and star-like features. The *green lines* in (**a**) separate the star-like features from the other ones. These outlines were determined manually (Color figure online)

and one frequency band. The total number of cells is 6 × 6 × 4 = 144. The vector **x** is then analyzed by CTA. Preprocessing is the same as in Sect. 4.1.1. The retained PCA dimension is 100.

4.2.2 Results

The topographic map of higher-order basis vectors is shown in Fig. 9(a). We visualized the basis vector as in previous work (Hoyer and Hyvärinen 2002; Hyvärinen et al. 2005). For each basis vector, each ellipse represents the spatial extent of the oriented filters, $W_k^o(x,y)$ and $W_k^e(x,y)$, and its orientation is the orientation which a complex cell detects. In Fig. 9(b), three prominent features are highlighted, which represent long contours, end-stopping and star-like features. On the map, the three kinds of basis vectors are separated from each other and have systematic order relationships. Furthermore, nearby long contour features tend to have the same orientation.

Next, to test if the learned features might be due to artifacts introduced by the fixed complex cell model, we performed the same experiment when $I(x,y)$ is Gaussian noise. Such a $I(x,y)$ was sampled from the Gaussian distribution with mean $\mathbf{0}$ and the covariance matrix equal to the one in the natural images used in Fig. 9. The map of higher order basis vectors for the noise input is depicted in Fig. 10. Star-like features are still present, but there are no long contour and end-stopping features. Therefore, we conclude that long contours, end-stopping features and the learned topography are due to the properties of natural images.

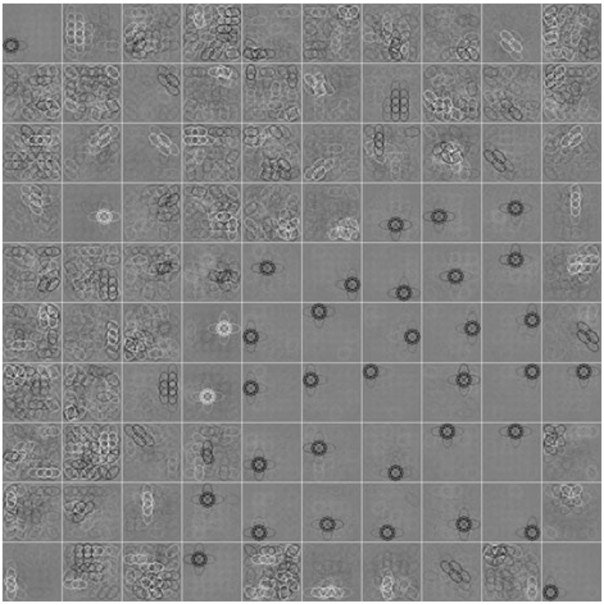


Fig. 10 Higher order basis for Gaussian noise inputs

For comparison, we performed the same experiment by TICA.[5] The estimated higher order basis is presented in Fig. 11(a). As in Fig. 9(a), star-like features and long contours exist. However, those features are not as well topographically aligned as those in CTA. The star-like features for TICA are more scattered on the map of the higher order basis, which disturbs the map of the features that are related to the properties of natural images. For CTA, the end-stopping features and the long contours are more neatly separated from the star-like features, which makes the learned topographic map better (Fig. 9(a)). Furthermore, most of the long contours in TICA seem to be shorter than those in CTA (Fig. 9(b) and (c)). Thus, CTA estimates longer contours and a cleaner topography than TICA does.

4.3 Text data

Our final application of CTA is for text data. Previously, ICA has been applied to similar data. Kolenda et al. (2000) analyzed a set of documents and showed that ICA found more easily interpretable structures than the more traditional latent semantic analysis (LSA). Honkela et al. (2010) analyzed word contexts in text corpora. ICA gave more distinct features reflecting linguistic categories than LSA. We apply here CTA to this kind of context-word data. The purpose is to see what kind of inter-relationships CTA identifies between the latent categories.

4.3.1 Methods

We constructed the context-word data as in Honkela et al. (2010). First, the most frequent $T = 200{,}000$ words were collected from 51,126 Wikipedia articles written in English; these

[5] For TICA extended to a two-dimensional lattice, we simply maximized the objective function only by the conjugate gradient method (Rasmussen 2006), and did not optimize the order of the components because the functional form of the objective function in TICA is different from the one in CTA. Therefore, the optimization method described in Appendix D could not be applied.

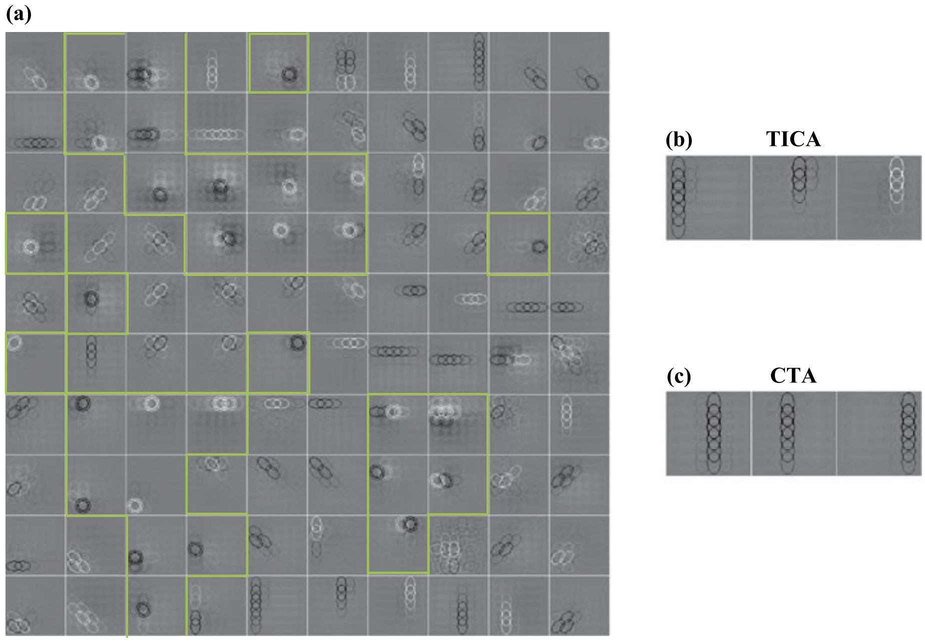


Fig. 11 (**a**) Higher order basis estimated by TICA for natural image inputs. The *green lines* separate the star-like features from the other kind of features. Note that the boundary condition of the map is ring-like. (**b**, **c**) Three examples of vertical contours estimated by TICA and CTA (Color figure online)

are called "collected words" in what follows. Next, we listed the context words occurring among the two words before or two words after each collected word and then took the most frequent 1,000 words. For each pair of collected and context word, we computed the joint frequency, and organized the values into a matrix $\mathbf{Y}$ of size 1,000 by 200,000. Finally, we obtained the context-word matrix $\mathbf{X} = (\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T))$ by transforming each element of $\mathbf{Y}$ as $x_i(t) = \log(y_{i,t} + 1)$.

As preprocessing, we made the mean of each row of $\mathbf{X}$ zero, and standardized its variance to one. Then, the data was whitened by PCA, and the dimension of the data was reduced from 1,000 to 60. Unlike in the experiments of natural images and outputs of complex cells, we assume here an one-dimensional topography and estimate $\mathbf{W}$ as described in Sect. 2.6. After the estimation, the context-word data can be represented as $\mathbf{X} = \mathbf{AS}$ where $\mathbf{S}$ is a 60 by 200,000 category-word matrix. Note that in the context of the text data, we call the rows in $\mathbf{S}$ "categories".

To quantify if the words in each category are similar to those in the same and adjacent category, we compute a similarity metric between two words using WordNet (Miller 1995; Fellbaum 1998) and the natural language toolkit (NLTK) (Bird et al. 2009). WordNet is a large lexical database where words are assigned to sets of synonyms (synsets), each expressing a lexicalized concept (Miller 1995). Since WordNet contains a network of synsets, one can compute the similarity between two words based on simple measures, e.g., the distance between synsets. For the computation of the similarity, first, we picked the top 40 words in each category, that are the words with the largest $|s_i(t)|$. Then, we computed similarities between all possible combinations of words within categories and between adjacent ones. The words which are not assigned to synsets were omitted from this analysis. In addition,

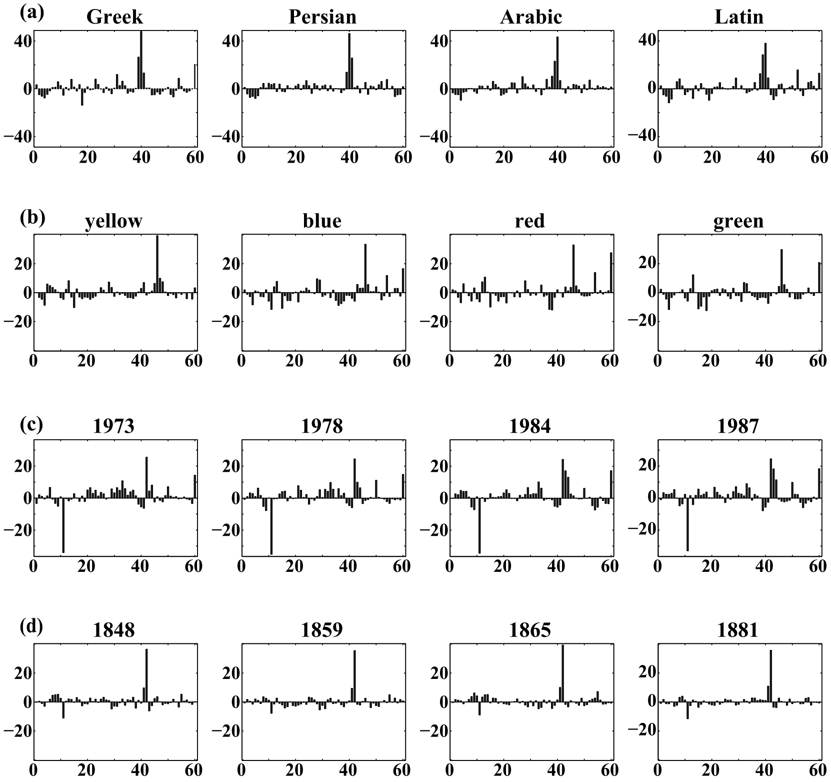


Fig. 12 Latent representations of (**a**) languages, (**b**) colors, (**c**) late 1900's and (**d**) late 1800's. Each plot shows a column of the matrix **S**. The columns are 60 dimensional vectors which contain the activations of the latent categories

categories in which all the top 40 words had no synsets were omitted.[6] To compute the similarity, we used the algorithm path_similarity in NLTK which is based on the shortest path. When words had more than two synsets, we computed similarities with all possible combinations of synsets and selected the maximum value. As a baseline, we computed similarities to 1,600 pairs of words which were randomly selected from 200,000 "collected words".

4.3.2 Results

We first show examples of latent representations of words (columns of **S**). In Fig. 12(a), latent representations of the names of four languages peak at the same category and show large responses around the peak. A similar property can be observed for the colors in Fig. 12(b). Honkela et al. (2010) obtained similar results: semantically similar words tend to have similar latent representations. Another interesting representation is found for numbers of years. Numbers for the late 1900's have a strong negative peak at category 11 and a positive peak at category 42 (Fig. 12(c)), while the late 1800's have positive peak at category 42 only

[6] For example, there were no synsets in the categories consisting of numbers, such as "the late 1900's" and "the late 1800's" in Fig. 12(c) and (d).

Table 2 Two examples of a topographic ordering between three categories. Denoting the k-th row of the matrix $\mathbf{S}$ by $\mathbf{S}^k$, the words with the top ten absolute values of a $\mathbf{S}^k$ are shown

Example 1: Number			Example 2: Media		
$\mathbf{S}^7$	$\mathbf{S}^8$	$\mathbf{S}^9$	$\mathbf{S}^{27}$	$\mathbf{S}^{28}$	$\mathbf{S}^{29}$
hours	few	6	published	marvel	band's
week	several	3	reports	comic	album
month	already	4	report	comics	pop
weeks	various	13	review	fantasy	albums
days	have	16	articles	batman	solo
months	numerous	8	detailed	x-men	band
day	frequently	21	newspaper	manga	rock
hour	two	23	journal	fiction	songs
year	eight	11	fiction	spider-man	blues
summer	many	32	books	superman	punk

Table 3 Another two examples of a topographic ordering between three categories. "undergrad." in $\mathbf{S}^{38}$ is an abbreviation of "undergraduate"

Example 3: Job titles and Names			Example 4: Politics and Education		
$\mathbf{S}^{31}$	$\mathbf{S}^{32}$	$\mathbf{S}^{33}$	$\mathbf{S}^{36}$	$\mathbf{S}^{37}$	$\mathbf{S}^{38}$
actress	actor	minister	minister	constitution	graduate
singer	scott	deputy	politician	constitutional	education
lord	smith	prime	government	parliament	sciences
actor	jr	ali	parliament	courts	engineering
songwriter	haward	appointed	poet	federal	undergrad.
governor	allen	elected	troops	court	medical
musician	lee	ahmed	election	senate	faculty
chairman	johnson	pierre	citizens	legislative	institute
secretary	anthony	mohammad	actress	law	school
naval	wilson	singh	party	supreme	science

(Fig. 12(d)). As another example, we found semantic categories for American states (results are not shown).

Regarding the relation between categories, CTA finds topographic representations where semantically similar categories are often near to each other (Table 2 and Table 3). Categories $\mathbf{S}^7$ (7-th row of $\mathbf{S}$), $\mathbf{S}^8$ and $\mathbf{S}^9$, represent units of time, "quantifiers" and roman numerals, respectively (left panel in Table 2). Another topographic order is for categories related to mass media (right panel in Table 2). Job titles and names are close to each other (left panel in Table 3), we found that categories for political and educational words are close to each other as well (right panel in Table 3).

To quantify how similar words within or between categories are, we computed the similarities between words as described above. Fig. 13 shows the mean of the top 30 similarity

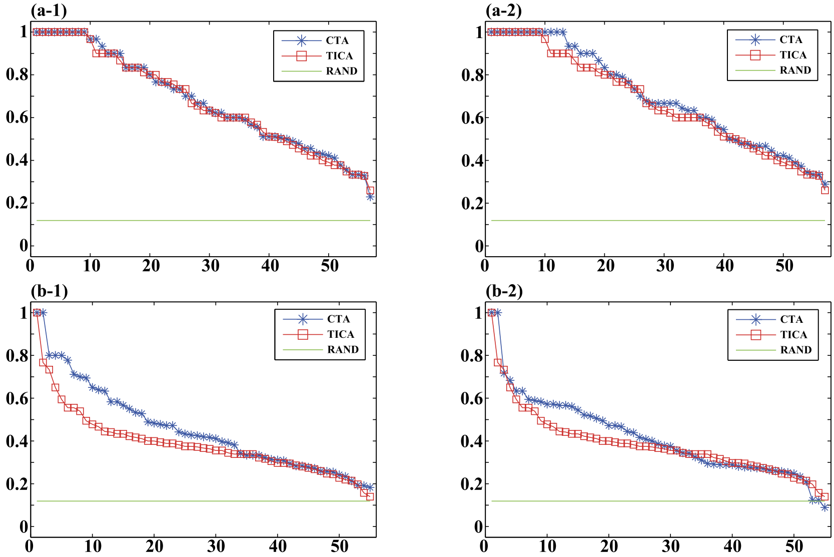


Fig. 13 Mean of top 30 similarity values (**a**) within categories and (**b**) between adjacent categories. For visualization, categories are sorted in descending order. (**a-1**) and (**b-1**) depict the best CTA and best TICA run in the sense of each objective function; (**a-2**) and (**b-2**) are for the worst CTA and best TICA run. RAND is the mean similarity for pairs of randomly selected words from the 200,000 collected words. For further details, we refer to the text body (Color figure online)

values at each category[7], and the categories are sorted in descending order for visualization. For the baseline, we first performed 1,000 times runs using pairs of randomly chosen words, and then computed the similarity as done above at each run. The baseline in Fig. 13 is the mean of those 1,000 runs. In the figure, we presented two cases of results: (a-1) and (b-1) depict the results for the best CTA run in the sense of having the largest value of the objective function, while (a-2) and (b-2) are for the worst CTA run. In total, we performed nine runs with different random initial conditions. For the similarities within categories (Fig. 13(a-1) and (a-2)), all are higher than the baseline similarity for random words. This means that CTA identifies semantically meaningful categories. Figure 13(b-1) and (b-2) clearly indicate that adjacent categories in CTA tend to contain similar words. Thus, CTA not only identifies semantically meaningful categories, but furthermore, it arranges them so that adjacent categories include semantically similar words.

We performed the same experiment and analysis for TICA. The best run results are shown in Fig. 13, too. Figure 13(a) shows that CTA and TICA have almost the same curves for similarity values within categories. However, for CTA, the curve for the similarities between adjacent categories is typically higher than for TICA (Fig. 13(b)). We performed one sided t-tests to each data in the two curves of Fig. 13(b). The null hypothesis of the test is that μ^{CTA} is less than μ^{TICA} where μ^{CTA} denotes the mean of the points forming the CTA curve in Fig. 13(b), and μ^{TICA} denotes the mean for the TICA curve. Note that we did not test if the CTA curve itself is higher than the TICA one because the points in Fig. 13 are sorted only for visualization and thus, there is no particular order-relationship between the points

[7] Some categories, which have less than 30 similarity values, were also omitted because the algorithm could not define the similarity for some pairs of synsets.

in the two curves. For Fig. 13(b-1) and (b-2), the p-values are 0.045 and 0.162, respectively. Thus, in the best result for CTA, the difference is statistically significant at 0.05 level (Fig. 13(b-1)). Even in the worst case, the performance of CTA seems intuitively better although the difference is not statistically significant (Fig. 13(b-2)). Therefore, we conclude that CTA identifies a better topography for text data as well.

5 Discussion

First, we summarize the connections between CTA and TICA. Then, we discuss the connection to other related work.

5.1 Connection to topographic independent component analysis

Section 2 showed that TICA and CTA are closely connected. We see their source models as special instances of the generative model (2), or of the distribution in (15). The distribution (16) which we used to define the CTA objective function is obtained from (15) by fixing the parameters $a_i = 1$, $b_i = 1$ and $\varrho_i = -1$ for all i. Ideally, we would estimate all these parameters. This is however difficult because we do not know the analytical expression of the partition function in (15). Therefore, we had to leave this challenge to future work. A possible approach is to use score matching (Hyvärinen 2006) or noise-contrastive estimation (Gutmann and Hyvärinen 2012).

The foremost difference between CTA and TICA is the additional assumption of linear correlations in the source vector $\mathbf{s}$. The sensitivity to linear correlation improved the topography estimations on artificial data as well as on real data as discussed in Sect. 3.2.2 and Sect. 4. A drawback of this sensitivity is that the identifiability of CTA becomes worse than ICA or TICA when the sources have no linear correlations (Fig. 6). To fix this drawback, we should estimate the amount of linear correlations. This could be achieved by estimating the ϱ_i, which is, as mentioned above, a topic that we had to leave to future work.

5.2 Connection to other related work and possible application

Structured sparsity is a concept related to topographic analysis. Mairal et al. (2011) applied dictionary learning on natural images using structured sparsity and the results were similar to TICA. The main difference is that they did not learn linearly correlated components like CTA. As discussed above, incorporating linear correlation can have advantages in topography estimation.

For natural images, Osindero et al. (2006) proposed another energy-based model which has an objective very similar to TICA, and produces similar results on natural images. Again, the difference to our method is that linear correlations between components are not explicitly modeled. Their model allows for overcomplete bases, which by necessity introduces some linear correlations. But it seems that their model still tries to minimize linear correlations instead of explicitly allowing them.

For the outputs of complex cells Hoyer and Hyvärinen (2002) first discovered long contours by applying a non-negative sparse coding method to the data. Hyvärinen et al. (2005) applied ICA to the outputs with multiple frequency bands and found long broadband contours. Comparing with our results, the main difference is the topography of the estimated features: in Fig. 9(a), similar features are close to each other, while they are randomly organized in the work cited above. The reason is that the previously used methods assume that the components are statistically independent. In addition, the end-stopping behavior that emerges for CTA was not seen in previous work.

For the results of text data, Honkela et al. (2010) applied ICA to the same kind of word data. Categories similar to ours were learned. Since Honkela and colleagues used ICA, there were no relationships between the categories. In contrast to their results, our method estimates a topographic representation where nearby categories include semantically similar words.

We have focused on learning data representations in this paper. CTA might also be useful for engineering applications. Recently, Kavukcuoglu et al. (2009) proposed an architecture for image recognition by creating a new feature through a topographic map which is learned by a method similar to TICA. Hence, we would expect that CTA is equally applicable in such tasks, with its additional sensitivity to linear correlations possibly being an advantage. However, such a study is out of scope of this paper, and we leave it to future work.

6 Conclusion

We proposed correlated topographic analysis (CTA) which is an extension of ICA to estimate the ordering (topography) of correlated components. In the proposed method, nearby components s_i are allowed to have linear and energy correlations; far-away components are as statistically independent as possible. In previous work, only higher order correlations were introduced. Our method generalizes those methods: if either linear or energy correlations in the components are present, CTA can estimate the topography. In addition, since optimization by gradient methods tends to get stuck in local maxima, we proposed a three-step optimization method which dramatically improved topography estimation.

Besides validating the properties of CTA using artificial data, we applied CTA to three kinds of real data sets: natural images, outputs of simulated complex cells, and text data. For natural images, similar basis vectors were close to each other, and we found that most basis vectors represented edges, not bars. In the experiment using the outputs of simulated complex cells, new kinds of higher-order features emerged and, moreover, similar features were systematically organized on the lattice. Finally, we showed for text data that CTA identifies semantic categories and orders them so that adjacent categories are connected by the semantics of the words which they represent.

Acknowledgements The authors are very grateful to Timo Honkela and Jaakko J. Väyrynen for providing us the text data, and wish to thank Shunji Satoh and Jun-ichiro Hirayama for their helpful discussion. H. Sasaki was supported by JSPS KAKENHI Grant Number 23·7556 (Grant-in-Aid for JSPS Fellows). H. Shouno was partially supported by Grand-in-Aid for Scientific Research (C) 21500214 and on Innovative Areas, 21103008, MEXT, Japan. A. Hyvärinen and M.U. Gutmann were supported by the Academy of Finland (CoE in Algorithmic Data Analysis, CoE in Inverse Problems Research, CoE in Computational Inference Research, and Computational Science Program).

Appendix A: An upper bound of the linear and energy correlation coefficient

The linear correlation coefficient $\rho^{\mathrm{lin}}_{s_i,s_j}$ for s_i and s_j reveals another relationship of linear correlations in **s** and **z** as

$$\left|\rho^{\mathrm{lin}}_{s_i,s_j}\right| = \left|\frac{E\{s_i s_j\}}{\sqrt{E\{s_i^2\}E\{s_j^2\}}}\right| = \frac{E\{\sigma_i\sigma_j\}}{\sqrt{E\{\sigma_i^2\}E\{\sigma_j^2\}}}\underbrace{\left|\frac{E\{z_i z_j\}}{\sqrt{E\{z_i^2\}E\{z_j^2\}}}\right|}_{|\rho^{\mathrm{lin}}_{z_i,z_j}|} \leq \left|\rho^{\mathrm{lin}}_{z_i,z_j}\right|, \qquad (34)$$

where $\rho^{\mathrm{lin}}_{z_i,z_j}$ denotes the linear correlation coefficient for z_i and z_j, and the Cauchy-Schwartz inequality, $E\{\sigma_i\sigma_j\}^2 \le E\{\sigma_i^2\}E\{\sigma_j^2\}$, was applied.

The energy correlation coefficient $\rho^{\mathrm{ene}}_{s_i,s_j}$ for s_i and s_j also has an upper bound:

$$\rho^{\mathrm{ene}}_{s_i,s_j} = \frac{\mathrm{cov}(s_i^2, s_j^2)}{\sqrt{E\{(s_i^2 - E\{s_i^2\})^2\}E\{(s_j^2 - E\{s_j^2\})^2\}}} < \frac{1}{3}\rho^{\mathrm{ene}}_{\sigma_i,\sigma_j} + \left(\rho^{\mathrm{lin}}_{z_i,z_j}\right)^2, \tag{35}$$

where $\rho^{\mathrm{ene}}_{\sigma_i,\sigma_j}$ is the correlation coefficient of the squares of σ_i and σ_j. Inequality (35) is proven below. The energy correlation of s_i and s_j is defined by

$$\begin{aligned}\rho^{\mathrm{ene}}_{s_i,s_j} &= \frac{\mathrm{cov}(s_i^2, s_j^2)}{\sqrt{E\{(s_i^2 - E\{s_i^2\})^2\}E\{(s_j^2 - E\{s_j^2\})^2\}}}, \\ &= \frac{E\{\sigma_i^2\sigma_j^2\}E\{z_i^2 z_j^2\} - E\{\sigma_i^2\}E\{\sigma_j^2\}E\{z_i^2\}E\{z_j^2\}}{\sqrt{(E\{\sigma_i^4\}E\{z_i^4\} - E\{\sigma_i^2\}^2E\{z_i^2\}^2)(E\{\sigma_j^4\}E\{z_j^4\} - E\{\sigma_j^2\}^2E\{z_j^2\}^2)}}, \\ &= \frac{\mathrm{cov}(\sigma_i^2, \sigma_j^2)}{\sqrt{(3E\{\sigma_i^4\} - E\{\sigma_i^2\}^2)(3E\{\sigma_j^4\} - E\{\sigma_j^2\}^2)}} \\ &\quad + \frac{2E\{\sigma_i^2\sigma_j^2\}(\rho^{\mathrm{lin}}_{z_i,z_j})^2}{\sqrt{(3E\{\sigma_i^4\} - E\{\sigma_i^2\}^2)(3E\{\sigma_j^4\} - E\{\sigma_j^2\}^2)}},\end{aligned} \tag{36}$$

where we used the two formulas valid for Gaussian variables with zero mean, $E\{z_i^2 z_j^2\} = E\{z_i^2\}E\{z_j^2\} + 2E\{z_i z_j\}^2$ and $E\{z_i^4\} = 3E\{z_i^2\}^2$ which are proven by Isserlis' theorem (Isserlis 1918; Michalowicz et al. 2009). The first term in (36) gives the following inequality,

$$\frac{\mathrm{cov}(\sigma_i^2, \sigma_j^2)}{\sqrt{(3E\{\sigma_i^4\} - E\{\sigma_i^2\}^2)(3E\{\sigma_j^4\} - E\{\sigma_j^2\}^2)}} < \frac{1}{3}\rho^{\mathrm{ene}}_{\sigma_i,\sigma_j}, \tag{37}$$

where $3E\{\sigma_i^4\} - E\{\sigma_i^2\}^2 > 3(E\{\sigma_i^4\} - E\{\sigma_i^2\}^2) = 3E\{(\sigma_i^2 - E\{\sigma_i^2\})^2\}$. For the second term in (36), first, using Jensen's inequality, $E\{\sigma_i^2\}^2 \le E\{\sigma_i^4\}$,

$$\frac{2E\{\sigma_i^2\sigma_j^2\}(\rho^{\mathrm{lin}}_{z_i,z_j})^2}{\sqrt{(3E\{\sigma_i^4\} - E\{\sigma_i^2\}^2)(3E\{\sigma_j^4\} - E\{\sigma_j^2\}^2)}} \le \frac{E\{\sigma_i^2\sigma_j^2\}}{\sqrt{E\{\sigma_i^4\}E\{\sigma_j^4\}}}\left(\rho^{\mathrm{lin}}_{z_i,z_j}\right)^2. \tag{38}$$

By applying the Cauchy-Schwartz inequality, $E\{\sigma_i^2\sigma_j^2\}^2 \le E\{\sigma_i^4\}E\{\sigma_j^4\}$, to the above inequality, the second term in (36) is bounded by the square of the linear correlation coefficient in z_i and z_j:

$$\frac{2E\{\sigma_i^2\sigma_j^2\}(\rho^{\mathrm{lin}}_{z_i,z_j})^2}{\sqrt{(3E\{\sigma_i^4\} - E\{\sigma_i^2\}^2)(3E\{\sigma_j^4\} - E\{\sigma_j^2\}^2)}} \le \left(\rho^{\mathrm{lin}}_{z_i,z_j}\right)^2. \tag{39}$$

We obtain (35) from (37) and (39).

Appendix B: Calculations for Eq. (11)

Here, we describe the details for obtaining (11). The equation before (11) is

$$p(\mathbf{s}|\mathbf{v},\mathbf{u};\boldsymbol{\lambda}) = \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}} \prod_{i=1}^{d} \sqrt{u_{i-1}+u_i+v_i}\exp\biggl[-\frac{1}{2}\bigl\{(u_{i-1}+u_i+v_i)s_i^2 + 2\lambda_i\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})}s_i s_{i+1}\bigr\}\biggr]. \tag{40}$$

This equation can be rewritten as

$$p(\mathbf{s}|\mathbf{v},\mathbf{u};\boldsymbol{\lambda}) = \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}} \exp\Biggl[-\frac{1}{2}\Biggl\{\underbrace{\sum_{i=1}^{d}(u_{i-1}+u_i+v_i)s_i^2}_{g(\mathbf{s},\mathbf{u},\mathbf{v})} + 2\sum_{i=1}^{d}\lambda_i\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})}s_i s_{i+1}\Biggr\}\Biggr] \times \Biggl(\prod_{i=1}^{d}\sqrt{u_{i-1}+u_i+v_i}\Biggr). \tag{41}$$

We expand $g(\mathbf{s},\mathbf{u},\mathbf{v})$ as follows:

$$g(\mathbf{s},\mathbf{u},\mathbf{v}) = \sum_{i=1}^{d} v_i s_i^2 + (u_d+u_1)s_1^2 + (u_1+u_2)s_2^2 + \cdots + (u_{d-2}+u_{d-1})s_{d-1}^2 + (u_{d-1}+u_d)s_d^2, \tag{42}$$

where the ring like boundary is applied. By rewriting the terms behind the first summation in (42) with respect to each u_i, we have

$$\begin{aligned} g(\mathbf{s},\mathbf{u},\mathbf{v}) &= \sum_{i=1}^{d} v_i s_i^2 + \bigl(s_1^2+s_2^2\bigr)u_1 + \bigl(s_2^2+s_3^2\bigr)u_2 + \cdots + \bigl(s_{d-1}^2+s_d^2\bigr)u_{d-1} + \bigl(s_d^2+s_1^2\bigr)u_d, \\ &= \sum_{i=1}^{d} v_i s_i^2 + \bigl(s_i^2+s_{i+1}^2\bigr)u_i. \end{aligned} \tag{43}$$

We obtain Eq. (11) by inserting (43) into (41),

$$p(\mathbf{s}|\mathbf{v},\mathbf{u};\boldsymbol{\lambda}) = \frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}} \exp\Biggl[-\frac{1}{2}\Biggl\{\sum_{i=1}^{d} v_i s_i^2 + \bigl(s_i^2+s_{i+1}^2\bigr)u_i + 2\sum_{i=1}^{d}\lambda_i\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})}s_i s_{i+1}\Biggr\}\Biggr]$$

$$\times\left(\prod_{i=1}^{d}\sqrt{u_{i-1}+u_i+v_i}\right),$$

$$=\frac{|\boldsymbol{\Lambda}|^{1/2}}{(2\pi)^{d/2}}\prod_{i=1}^{d}\sqrt{u_{i-1}+u_i+v_i}\exp\Big[-\frac{1}{2}\big\{v_i s_i^2+\big(s_i^2+s_{i+1}^2\big)u_i$$

$$+2\lambda_i\sqrt{(u_{i-1}+u_i+v_i)(u_i+u_{i+1}+v_{i+1})}s_i s_{i+1}\big\}\Big]. \qquad (44)$$

Appendix C: Deriving the approximation $\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b})$

In this appendix, we give the detailed description of how to derive the probability distribution $\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b})$ in (15).

To obtain $\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b})$, we have to calculate the integral

$$\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b})=\int_0^\infty\int_0^\infty\tilde{p}(\mathbf{s}|\mathbf{v},\mathbf{u};\boldsymbol{\varrho})p(\mathbf{v},\mathbf{u};\mathbf{a},\mathbf{b})d\mathbf{v}d\mathbf{u},$$

$$\propto\prod_{i=1}^{d}\int_0^\infty v_i^{-3/2}\exp\left\{-\frac{1}{2}\left(s_i^2 v_i+\frac{a_i}{v_i}\right)\right\}dv_i$$

$$\times\prod_{i=1}^{d}\int_0^\infty u_i^{-3/2}\exp\left[-\frac{1}{2}\left\{\big(s_i^2+s_{i+1}^2+2\varrho_i s_i s_{i+1}\big)u_i+\frac{b_i}{u_i}\right\}\right]du_i. \qquad (45)$$

To calculate this integral, we use the following formula (Andrews and Mallows 1974),

$$\int_0^\infty\exp\left\{-\frac{1}{2}\left(\alpha^2y^2+\frac{\beta^2}{y^2}\right)\right\}dy=\left(\frac{\pi}{2\alpha^2}\right)^{1/2}\exp\big(-|\alpha\beta|\big). \qquad (46)$$

By change of variable $x=y^{-2}$,

$$\int_0^\infty x^{-3/2}\exp\left\{-\frac{1}{2}\left(\beta^2x+\frac{\alpha^2}{x}\right)\right\}dx=\left(\frac{2\pi}{\alpha^2}\right)^{1/2}\exp\big(-|\alpha\beta|\big). \qquad (47)$$

The formula (47) gives (15) as

$$\tilde{p}(\mathbf{s};\boldsymbol{\varrho},\mathbf{a},\mathbf{b})\propto\prod_{i=1}^{d}\exp\Big(-\sqrt{a_i}|s_i|-\sqrt{b_i}\sqrt{s_i^2+s_{i+1}^2+2\varrho_i s_i s_{i+1}}\Big). \qquad (48)$$

Appendix D: Optimizing components on a two-dimensional lattice

We first present the optimization algorithm. Then, we demonstrate its effectiveness using natural image data.

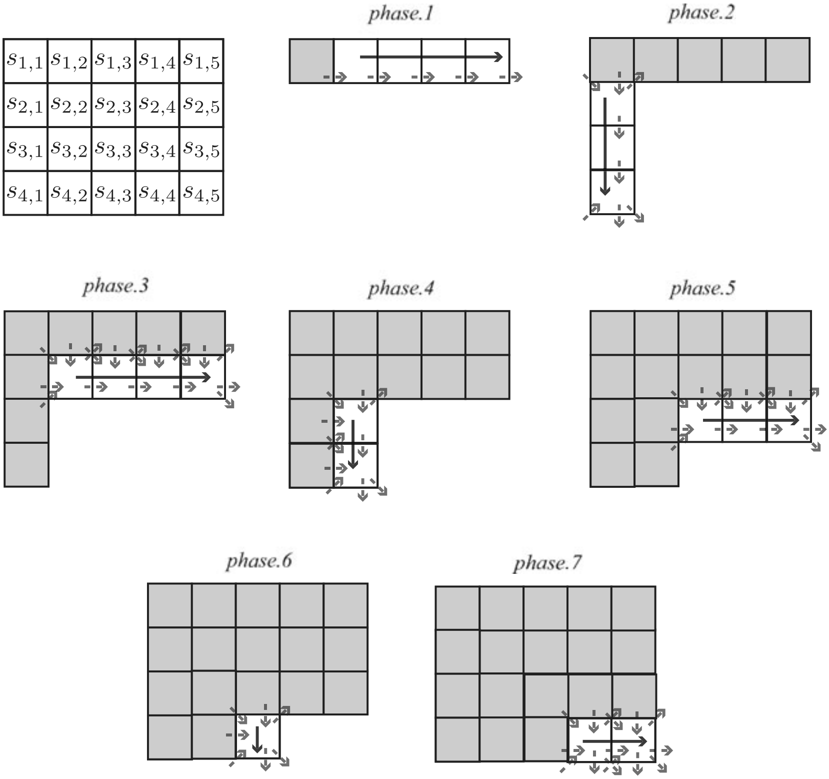

$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,4}$ $s_{1,5}$
$s_{2,1}$ $s_{2,2}$ $s_{2,3}$ $s_{2,4}$ $s_{2,5}$
$s_{3,1}$ $s_{3,2}$ $s_{3,3}$ $s_{3,4}$ $s_{3,5}$
$s_{4,1}$ $s_{4,2}$ $s_{4,3}$ $s_{4,4}$ $s_{4,5}$
phase.1
phase.2
phase.3
phase.4
phase.5
phase.6
phase.7

Fig. 14 The flow of the optimization method for components on a two-dimensional lattice. *Black solid arrows* represent the direction of the optimization, and *gray dashed arrows* represent the statistical dependency to the already optimized components to be evaluated in each phase. *Grayed-out cells* depict the components already optimized in the former phases

D.1 Algorithm

Here, we describe the optimization method on a two-dimensional lattice. Each component is denoted by $s_{i,j}$ for $i = 1, \dots, d_x$ and $j = 1, \dots, d_y$, where d_x and d_y represent the size of the lattice along with horizontal and vertical directions, respectively. For simplicity, we suppose that $d_y \leq d_x$. In the method, the only difference to the optimization method on an one-dimensional lattice is Step 2 and the objective function J in Step 3. Since the objective function in Step 3 is replaced by (29), we explain the algorithm in Step 2 below.

The algorithm is a straightforward extension of the case of an one-dimensional lattice. The key idea is to optimize the order and signs of components alternately along the horizontal and vertical directions. We call each such optimization a "phase" in the algorithm. A sketch of the method is depicted in Fig. 14. In each phase, each optimization problem reduces to that of the one-dimensional lattice. Therefore, we can expect to apply the method of the one-dimensional lattice with small modifications. In fact, the modifications between the methods of the one- and two-dimensional lattice are twofold: (1) one cannot use the indices selected in the former phases and (2) components are optimized at each phase while evaluating dependencies to the components already optimized in the former phases. For example, the problem at phase 3 can be formulated as

$$k^*_{2,j}, c^*_{2,j} = \arg\max_{k_{2,j}, c_{2,j}} L_1 + L_2, \tag{49}$$

$$L_1 = -\frac{1}{T}\sum_{t=1}^{T}\sum_{l=2}^{d_x} G\big(c_{2,l} s_{k_{2,l}}(t) - c_{2,l+1} s_{k_{2,l+1}}(t)\big), \tag{50}$$

$$\begin{aligned} L_2 = -\frac{1}{T}\sum_{t=1}^{T}\Bigg[\sum_{l=2}^{d_x}\sum_{m=-1}^{1}\big\{G\big(c_{2,l} s_{k_{2,l}}(t) - s^*_{1,l+m}(t)\big)\big\} + G\big(c_{2,2} s_{k_{2,2}}(t) - s^*_{2,1}(t)\big) \\ + G\big(c_{2,2} s_{k_{2,2}}(t) - s^*_{3,1}(t)\big) + G\big(c_{2,2} s_{k_{2,d_x}}(t) - s^*_{3,1}(t)\big)\Bigg], \end{aligned} \tag{51}$$

where $j = 2, 3, \dots, d_x$ and $s^*_{i,j}(t) = c^*_{i,j}\mathbf{W}^{(1)}_{k^*_{i,j}}\mathbf{x}(t)$, which is an already optimized component in former steps. $k_{2,j}$ in (49) must be selected from the remaining index set $\{2, \dots, d_x d_y\} \setminus \{k^*_{1,2}, \dots, k^*_{1,d_x}, k^*_{2,1}, \dots, k^*_{d_y,1}\}$. The brief description of the algorithm is as follows:

Algorithm 3: Finding an optimal order and signs on a two-dimensional lattice

```
Input: ICA components, $\mathbf{s}^{(1)}(1), \mathbf{s}^{(1)}(2), \dots, \mathbf{s}^{(1)}(T)$.
1. Assumption: $d_y \le d_x$ (or $d_x \le d_y$)
2. Initialization: $c_{1,1} = 1$, $k_{1,1} = 1$ and $\mathbb{I} = \{2, 3, \dots, d_y d_x\}$.
3. Repeat to find optimal indices and signs at each phase from $n = 1$ to $n = d_y$ (or $n = d_x$):
    (a) Apply the algorithm 2 with the modifications given above to the components on the
        $n$-th row (or column) and obtain the optimal indices from $\mathbb{I}$ and signs.
    (b) Update the components on the $n$-th row by using the optimal indices and signs.
    (c) Update $\mathbb{I}$ by removing the obtained optimal indices.
    (d) Apply the algorithm 2 with the modifications given above to the components on the
        $n$-th column (or row) and obtain the optimal indices from $\mathbb{I}$ and signs.
    (e) Update the components on the $n$-th column by using the optimal indices and signs.
    (f) Update $\mathbb{I}$ by removing the obtained optimal indices.
Output: the optimal indices (order) and signs.
```

MATLAB code can be obtained at http://www.cs.helsinki.fi/u/ahyvarin/code/cta/.

D.2 Effectiveness of the optimization method on natural images

We compare results for natural images obtained by the proposed optimization method and the conjugate gradient method only. The experimental methods are described in Sect. 4.1.

Basis vectors estimated by the conjugate gradient method only are presented in Fig. 15. It seems that nearby basis vectors are less similar than those in Fig. 7. The scatter plots of the Gabor parameters shown in Fig. 16 reveal that the spatial locations of adjacent basis vectors in Fig. 15 have weaker correlations than those in Fig. 8. Furthermore, the objective function from the proposed optimization method is larger than the one from the conjugate gradient method only:

- proposed method: $J(\mathbf{W}) = -233.331$

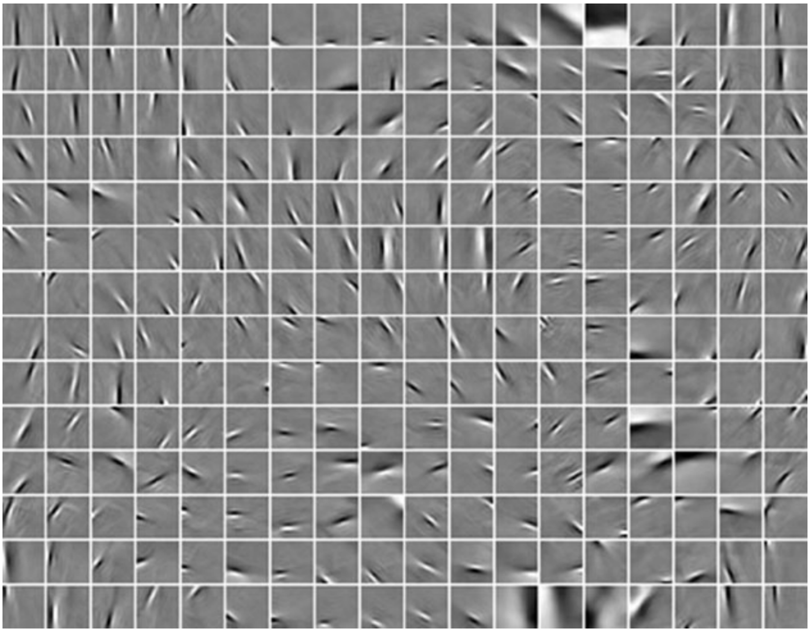


Fig. 15 Estimated basis vectors from natural images using the conjugate gradient method only

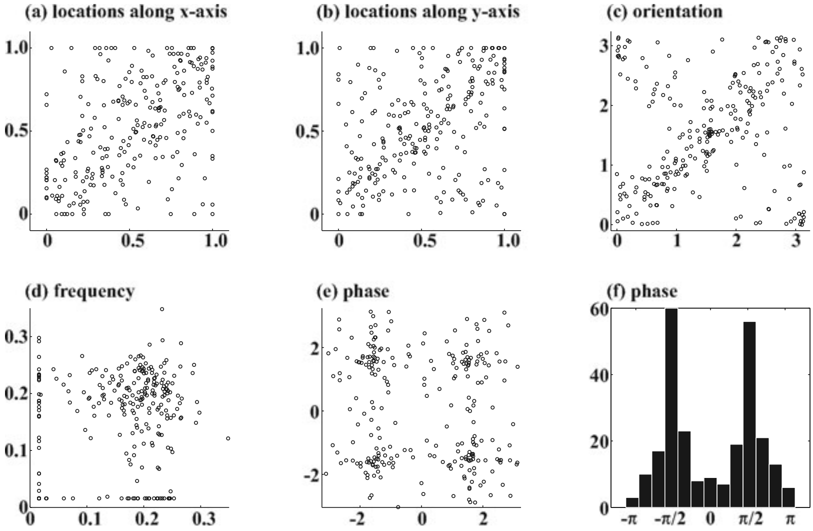


Fig. 16 Scatter plots of Gabor parameters for the pairs of adjacent basis vectors in Fig. 15

– conjugate gradient method: $J(\mathbf{W}) = -233.623$

Thus, the proposed optimization method on a two-dimensional lattice works better than the conjugate gradient method only.

References

Amari, S., Cichocki, A., & Yang, H. H. (1996). A new learning algorithm for blind signal separation. In *Advances in neural information processing systems* (Vol. 8, pp. 757–763).

Andrews, D. F., & Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *36*(1), 99–102.

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

Bell, A. J., & Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Research*, *37*(23), 3327–3338.

Bellman, R. E. (1957). *Dynamic programming*. Princeton: Princeton University Press.

Bellman, R. E., & Dreyfus, S. E. (1962). *Applied dynamic programming*. Princeton: Princeton University Press.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. Sebastopol: O'Reilly Media.

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

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

Fellbaum, C. (1998). *WordNet: an electronic lexical database*. Cambridge: MIT.

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

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

Held, M., & Karp, R. M. (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, *10*(1), 196–210.

Honkela, T., Hyvärinen, A., & Väyrynen, J. J. (2010). WordICA—emergence of linguistic representations for words by independent component analysis. *Natural Language Engineering*, *16*(03), 277–308.

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

Hyvärinen, A. (2006). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, *6*, 695–708.

Hyvärinen, A., & Hoyer, P. O. (2001). A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, *41*(18), 2413–2423.

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

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

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

Hyvärinen, A., Hurri, J., & Hoyer, P. O. (2009). *Natural image statistics: a probabilistic approach to early computational vision*. Berlin: Springer.

Isserlis, L. (1918). On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, *12*(1/2), 134–139.

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

Kavukcuoglu, K., Ranzato, M. A., Fergus, R., & Le-Cun, Y. (2009). Learning invariant features through topographic filter maps. In *IEEE conference on computer vision and pattern recognition, 2009. CVPR 2009* (pp. 1605–1612). New York: IEEE.

Kolenda, T., Hansen, L. K., & Sigurdsson, S. (2000). Independent components in text. In *Advances in independent component analysis* (pp. 229–250). Berlin: Springer.

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

Michalowicz, J. V., Nichols, J. M., Bucholtz, F., & Olson, C. C. (2009). An Isserlis' theorem for mixed Gaussian variables: application to the auto-bispectral density. *Journal of Statistical Physics*, *136*(1), 89–102.

Miller, G. A. (1995). Wordnet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607–609.

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

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

Simoncelli, E. P. (1999). Modeling the joint statistics of images in the wavelet domain. In *Proc. SPIE, 44th annual meeting* (Vol. 3813, pp. 188–195).

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *67*(1), 91–108.

Vigário, R., Särelä, J., Jousmäki, V., Hämäläinen, M., & Oja, E. (2000). Independent component approach to the analysis of EEG and MEG recordings. *IEEE Transactions on Biomedical Engineering*, *47*(5), 589–593.

Zoran, D., & Weiss, Y. (2009). The "tree-dependent components" of natural images are edge filters. In *Advances in neural information processing systems* (Vol. 22, pp. 2340–2348).