Least-squares Independent Component Analysis*

Taiji Suzuki

Department of Mathematical Informatics, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
s-taiji@stat.t.u-tokyo.ac.jp

Masashi Sugiyama

Department of Computer Science, Tokyo Institute of Technology
and PRESTO, Japan Science and Technology Agency
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan
sugi@cs.titech.ac.jp

Abstract

Accurately evaluating statistical independence among random variables is a key element of Independent Component Analysis (ICA). In this paper, we employ a squared-loss variant of mutual information as an independence measure and give its estimation method. Our basic idea is to estimate the *ratio* of probability densities directly without going through density estimation, by which a hard task of density estimation can be avoided. In this density ratio approach, a natural cross-validation procedure is available for hyper-parameter selection. Thus, all tuning parameters such as the kernel width or the regularization parameter can be objectively optimized. This is an advantage over recently developed kernel-based independence measures and is a highly useful property in unsupervised learning problems such as ICA. Based on this novel independence measure, we develop an ICA algorithm named *Least-squares Independent Component Analysis* (LICA).

1 Introduction

The purpose of Independent Component Analysis (ICA) (Hyvärinen et al., 2001) is to obtain a transformation matrix that separates mixed signals into statistically-independent source signals. A direct approach to ICA is to find a transformation matrix such that independence among separated signals is maximized under some independence measure such as *mutual information* (MI).

*A MATLAB® implementation of the proposed algorithm, LICA, is available from 'http://www.simplex.t.u-tokyo.ac.jp/~s-taiji/software/LICA/index.html'.

Various approaches to evaluating the independence among random variables from samples have been explored so far. A naive approach is to estimate probability densities based on parametric or non-parametric density estimation methods. However, finding an appropriate parametric model is not easy without strong prior knowledge and non-parametric estimation is not accurate in high-dimensional problems. Thus this naive approach is not reliable in practice. Another approach is to approximate the *negentropy* (or negative entropy) based on the *Gram-Charlier expansion* (Cardoso & Souloumiac, 1993; Comon, 1994; Amari et al., 1996) or the *Edgeworth expansion* (Hulle, 2008). An advantage of this negentropy-based approach is that a hard task of density estimation is not directly involved. However, these expansion techniques are based on the assumption that the target density is close to normal and violation of this assumption can cause large approximation error.

The above approaches are based on the probability densities of signals. Another line of research that does not explicitly involve probability densities employs *non-linear correlation*—signals are statistically independent if and only if all non-linear correlations among the signals vanish. Following this line, computationally efficient algorithms have been developed based on a *contrast function* (Jutten & Herault, 1991; Hyvärinen, 1999), which is an approximation of negentropy or mutual information. However, these methods require to pre-specify non-linearities in the contrast function, and thus could be inaccurate if the predetermined non-linearities do not match the target distribution. To cope with this problem, the *kernel trick* has been applied in ICA, which allows one to evaluate all non-linear correlations in a computationally efficient manner (Bach & Jordan, 2002). However, its practical performance depends on the choice of kernels (more specifically, the Gaussian kernel width) and there seems no theoretically justified method to determine the kernel width (see also Fukumizu et al., 2009). This is a critical problem in unsupervised learning problems such as ICA.

In this paper, we develop a new ICA algorithm that resolves the problems mentioned above. We adopt a squared-loss variant of MI (which we call *squared-loss MI*; SMI) as an independence measure and approximate it by estimating the ratio of probability densities contained in SMI directly without going through density estimation. This approach—which follows the line of Sugiyama et al. (2008), Kanamori et al. (2009), and Nguyen et al. (2010)—allows us to avoid a hard task of density estimation. Another practical advantage of this density-ratio approach is that a natural cross-validation (CV) procedure is available for hyper-parameter selection. Thus all tuning parameters such as the kernel width or the regularization parameter can be objectively and systematically optimized through CV.

From an algorithmic point of view, our density-ratio approach *analytically* provides a non-parametric estimator of SMI; furthermore its derivative can also be computed analytically and these properties are utilized in deriving a new ICA algorithm. The proposed method is named *Least-squares Independent Component Analysis* (LICA).

Characteristics of existing and proposed ICA methods are summarized in Table 1, highlighting the advantage of the proposed LICA approach.

The structure of this paper is as follows. In Section 2, we formulate our estimator of

Table 1: Summary of existing and proposed ICA methods.

	Hyper-parameter selection	Distribution
Fast ICA (FICA) (Hyvärinen, 1999)	**Not Necessary**	Not Free
Natural-gradient ICA (NICA) (Amari et al., 1996)	**Not Necessary**	Not Free
Kernel ICA (KICA) (Bach & Jordan, 2002)	Not Available	**Free**
Edgeworth-expansion ICA (EICA) (Hulle, 2008)	**Not Necessary**	Nearly normal
Least-squares ICA (LICA) (proposed)	**Available**	**Free**

SMI. In Section 3, we derive the LICA algorithm based on the SMI estimator. Section 4 is devoted to numerical experiments where we show that our method properly estimate the true demixing matrix using toy datasets, and compare the performances of the proposed and existing methods on artificial and real datasets.

2 SMI Estimation for ICA

In this section, we formulate the ICA problem and introduce our independence measure, SMI. Then we give an estimation method of SMI and derive an ICA algorithm.

2.1 Problem Formulation

Suppose there is a d-dimensional random signal

$$\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)})^\top$$

drawn from a distribution with density $p(\boldsymbol{x})$, where $\{x^{(m)}\}_{m=1}^d$ are statistically independent of each other, and $^\top$ denotes the transpose of a matrix or a vector. Thus, $p(\boldsymbol{x})$ can be factorized as

$$p(\boldsymbol{x}) = \prod_{m=1}^{d} p_m(x^{(m)}).$$

We cannot directly observe the *source* signal $\boldsymbol{x}$, but only a linearly mixed signal $\boldsymbol{y}$:

$$\boldsymbol{y} = (y^{(1)}, \ldots, y^{(d)})^\top := \boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{A}$ is a $d \times d$ invertible matrix called the *mixing matrix*. The goal of ICA is, given samples of the mixed signals $\{\boldsymbol{y}_i\}_{i=1}^n$, to obtain a *demixing matrix* $\boldsymbol{W}$ that recovers the original source signal $\boldsymbol{x}$. We denote the demixed signal by $\boldsymbol{z}$:

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{y}.$$

The ideal solution is $\boldsymbol{W} = \boldsymbol{A}^{-1}$, but we can only recover the source signals up to permutation and scaling of components of $\boldsymbol{x}$ due to non-identifiability of the ICA setup (Hyvärinen et al., 2001).

A direct approach to ICA is to determine $\boldsymbol{W}$ so that components of $\boldsymbol{z}$ are as independent as possible. Here, we adopt SMI as the independence measure:

$$I_s(Z^{(1)}, \ldots, Z^{(d)}) := \frac{1}{2} \int \left(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})} - 1 \right)^2 r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}, \tag{1}$$

where $q(\boldsymbol{z})$ denotes the joint density of $\boldsymbol{z}$ and $r(\boldsymbol{z})$ denotes the product of marginal densities $\{q_m(z^{(m)})\}_{m=1}^d$:

$$r(\boldsymbol{z}) = \prod_{m=1}^{d} q_m(z^{(m)}).$$

Note that SMI is the *Pearson divergence* (Pearson, 1900; Paninsky, 2003; Liese & Vajda, 2006; Cichocki et al., 2009) between $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$, while ordinary MI is the *Kullback-Leibler divergence* (Kullback & Leibler, 1951). Since I_s is non-negative and it vanishes if and only if $q(\boldsymbol{z}) = r(\boldsymbol{z})$, the degree of independence among $\{z^{(m)}\}_{m=1}^d$ may be measured by SMI. Note that Eq.(1) corresponds to the *f-divergence* (Ali & Silvey, 1966; Csiszár, 1967) between $q(\boldsymbol{x})$ and $r(\boldsymbol{z})$ with the squared-loss, while ordinary MI corresponds to the f-divergence with the log-loss. Thus SMI could be regarded as a natural generalization of ordinary MI.

Based on the independence detection property of SMI, we try to find the demixing matrix $\boldsymbol{W}$ that minimizes SMI. Let us denote the demixed samples by

$$\{\boldsymbol{z}_i \mid \boldsymbol{z}_i = (z_i^{(1)}, \ldots, z_i^{(d)})^\top := \boldsymbol{W}\boldsymbol{y}_i\}_{i=1}^n.$$

Our key constraint when estimating SMI is that we want to avoid density estimation since it is a hard task (Vapnik, 1998). Below, we show how this could be accomplished.

2.2 SMI Approximation via Density Ratio Estimation

We approximate SMI via *density ratio estimation*. Let us denote the ratio of the densities $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$ by

$$g^*(\boldsymbol{z}) := \frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}. \tag{2}$$

Then SMI can be written as

$$\begin{aligned} I_s(Z^{(1)}, \ldots, Z^{(d)}) &= \frac{1}{2} \int \left(g^*(\boldsymbol{z}) - 1\right)^2 r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \int \left(g^*(\boldsymbol{z})^2 r(\boldsymbol{z}) - 2g^*(\boldsymbol{z}) r(\boldsymbol{z}) + r(\boldsymbol{z}) \right) \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \int \left(g^*(\boldsymbol{z}) q(\boldsymbol{z}) - 2q(\boldsymbol{z}) + r(\boldsymbol{z}) \right) \mathrm{d}\boldsymbol{z} \\ &= \frac{1}{2} \int g^*(\boldsymbol{z}) q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} - \frac{1}{2}. \end{aligned} \tag{3}$$

Therefore, SMI can be approximated through the estimation of $\int g^*(\boldsymbol{z})q(\boldsymbol{z})\mathrm{d}\boldsymbol{z}$, the expectation of $g^*(\boldsymbol{z})$ over $q(\boldsymbol{z})$. This can be achieved by taking the sample average of an estimator of the density ratio $g^*(\boldsymbol{z})$, say $\widehat{g}(\boldsymbol{z})$:

$$\widehat{I}_s = \frac{1}{2n}\sum_{i=1}^{n}\widehat{g}(\boldsymbol{z}_i) - \frac{1}{2}. \tag{4}$$

We take the least-squares approach to estimating the density ratio $g^*(\boldsymbol{z})$:

$$\begin{aligned}&\inf_g\left[\frac{1}{2}\int\Big(g(\boldsymbol{z})-g^*(\boldsymbol{z})\Big)^2 r(\boldsymbol{z})\mathrm{d}\boldsymbol{z}\right]\\ &\quad=\inf_g\left[\int\Big(\frac{1}{2}g(\boldsymbol{z})^2 r(\boldsymbol{z})-g(\boldsymbol{z})q(\boldsymbol{z})\Big)\mathrm{d}\boldsymbol{z}\right]+\text{constant},\end{aligned}$$

where $\inf_g$ is taken over all measurable functions. Obviously the optimal solution is the density ratio g^*. Thus computing I_s is now reduced to solving the following optimization problem:

$$\inf_g\left[\int\Big(\frac{1}{2}g(\boldsymbol{z})^2 r(\boldsymbol{z})-g(\boldsymbol{z})q(\boldsymbol{z})\Big)\mathrm{d}\boldsymbol{z}\right]. \tag{5}$$

However, directly solving the problem (5) is not possible due to the following two reasons. The first reason is that finding the minimizer over all measurable functions is not tractable in practice since the search space is too vast. To overcome this problem, we restrict the search space to some linear subspace $\mathcal{G}$:

$$\mathcal{G} = \{\boldsymbol{\alpha}^\top\boldsymbol{\varphi}(\boldsymbol{z}) \mid \boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_b)^\top \in \mathbb{R}^b\}, \tag{6}$$

where $\boldsymbol{\alpha}$ is a parameter to be learned from samples, and $\boldsymbol{\varphi}(\boldsymbol{z})$ is a basis function vector such that

$$\boldsymbol{\varphi}(\boldsymbol{z}) = (\varphi_1(\boldsymbol{z}),\ldots,\varphi_b(\boldsymbol{z}))^\top \geq \boldsymbol{0}_b \quad \text{for all } \boldsymbol{z}.$$

$\boldsymbol{0}_b$ denotes the b-dimensional vector with all zeros. Note that $\boldsymbol{\varphi}(\boldsymbol{z})$ could be dependent on the samples $\{\boldsymbol{z}_i\}_{i=1}^n$, i.e., *kernel* models are also allowed. We explain how the basis functions $\boldsymbol{\varphi}(\boldsymbol{z})$ are chosen in Section 2.3.

The second reason why directly solving the problem (5) is not possible is that the expectations over the true probability densities $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$ cannot be computed since $q(\boldsymbol{z})$ and $r(\boldsymbol{z})$ are unknown. To cope with this problem, we approximate the expectations by their empirical averages—then the optimization problem is reduced to

$$\widehat{\boldsymbol{\alpha}} := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^b}\left[\frac{1}{2}\boldsymbol{\alpha}^\top\widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top\boldsymbol{\alpha} + \lambda\boldsymbol{\alpha}^\top\boldsymbol{R}\boldsymbol{\alpha}\right], \tag{7}$$

where we included $\lambda\boldsymbol{\alpha}^\top\boldsymbol{R}\boldsymbol{\alpha}$ ($\lambda > 0$) for avoiding overfitting. λ is called the *regularization*

parameter, and $\boldsymbol{R}$ is some positive definite matrix. $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$ are defined as

$$\widehat{\boldsymbol{H}} := \frac{1}{n^d} \sum_{i_1,\ldots,i_d=1}^{n} \boldsymbol{\varphi}(z_{i_1}^{(1)}, \ldots, z_{i_d}^{(d)}) \boldsymbol{\varphi}(z_{i_1}^{(1)}, \ldots, z_{i_d}^{(d)})^\top, \tag{8}$$

$$\widehat{\boldsymbol{h}} := \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\varphi}(z_i^{(1)}, \ldots, z_i^{(d)}). \tag{9}$$

Differentiating the objective function in Eq.(7) with respect to $\boldsymbol{\alpha}$ and equating it to zero, we can obtain an analytic-form solution as

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1} \widehat{\boldsymbol{h}}.$$

Thus, the solution can be computed very efficiently just by solving a system of linear equations.

Once the density ratio (2) has been estimated, SMI can be approximated by plugging the estimated density ratio $\widehat{g}(\boldsymbol{z}) = \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{\varphi}(\boldsymbol{z})$ in Eq.(4):

$$\widehat{I}_s = \frac{1}{2} \widehat{\boldsymbol{\alpha}}^\top \widehat{\boldsymbol{h}} - \frac{1}{2}. \tag{10}$$

Note that we may obtain various expressions of SMI using the following identities:

$$\int g^*(\boldsymbol{z})^2 r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int g^*(\boldsymbol{z}) q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z},$$
$$\int g^*(\boldsymbol{z}) r(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = 1.$$

Ordinary MI based on the Kullback-Leibler divergence can also be estimated similarly using the density ratio (Suzuki et al., 2008). However, the use of SMI is more advantageous due to the analytic-form solution, as described in Section 3.

2.3 Design of Basis Functions and Hyper-parameter Selection

As basis functions, we propose to use a Gaussian kernel:

$$\varphi_\ell(\boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{v}_\ell\|^2}{2\sigma^2}\right) = \prod_{m=1}^{d} \exp\left(-\frac{(z^{(m)} - v_\ell^{(m)})^2}{2\sigma^2}\right), \tag{11}$$

where

$$\{\boldsymbol{v}_\ell \mid \boldsymbol{v}_\ell = (v_\ell^{(1)}, \ldots, v_\ell^{(d)})^\top\}_{\ell=1}^{b}$$

are Gaussian centers randomly chosen from $\{\boldsymbol{z}_i\}_{i=1}^n$—more precisely, we set $\boldsymbol{v}_\ell = \boldsymbol{z}_{c(\ell)}$, where $\{c(\ell)\}_{\ell=1}^b$ are randomly chosen from $\{1, \ldots, n\}$ without replacement. An advantage

of the Gaussian kernel lies in the factorizability in Eq.(11), contributing to reducing the computational cost of the matrix $\widehat{\boldsymbol{H}}$ significantly:

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n^d} \prod_{m=1}^{d} \left[\sum_{i=1}^{n} \exp\left(-\frac{(z_i^{(m)} - v_\ell^{(m)})^2 + (z_i^{(m)} - v_{\ell'}^{(m)})^2}{2\sigma^2} \right) \right].$$

We use the RKHS (Reproducing Kernel Hilbert Space) norm of $\boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{z})$ induced by the Gaussian kernel as the regularization term $\boldsymbol{\alpha}^\top \boldsymbol{R} \boldsymbol{\alpha}$, which is a popular choice in the kernel method community (Schölkopf & Smola, 2002):

$$R_{\ell,\ell'} = \exp\left(-\frac{\|\boldsymbol{v}_\ell - \boldsymbol{v}_{\ell'}\|^2}{2\sigma^2} \right). \tag{12}$$

In the experiments, we fix the number of basis functions to

$$b = \min(300, n),$$

and choose the Gaussian width σ and the regularization parameter λ by CV with grid search as follows. First, the samples $\{\boldsymbol{z}_i\}_{i=1}^n$ are divided into K disjoint subsets $\{\mathcal{Z}_k\}_{k=1}^K$ of (approximately) the same size (we use $K = 5$ in the experiments). Then an estimator $\widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k}$ is obtained using $\mathcal{Z}\backslash\mathcal{Z}_k$ (i.e., $\mathcal{Z}$ without $\mathcal{Z}_k$) and the approximation error for the hold-out samples $\mathcal{Z}_k$ is computed:

$$J_{\mathcal{Z}_k}^{(K\text{-CV})} = \frac{1}{2} \widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k}^\top \widehat{\boldsymbol{H}}_{\mathcal{Z}_k} \widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k} - \widehat{\boldsymbol{h}}_{\mathcal{Z}_k}^\top \widehat{\boldsymbol{\alpha}}_{\mathcal{Z}\backslash\mathcal{Z}_k},$$

where the matrix $\widehat{\boldsymbol{H}}_{\mathcal{Z}_k}$ and the vector $\widehat{\boldsymbol{h}}_{\mathcal{Z}_k}$ are defined in the same way as $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$, but computed only using $\mathcal{Z}_k$. This procedure is repeated for $k = 1, \ldots, K$ and its average $J^{(K\text{-CV})}$ is computed:

$$J^{(K\text{-CV})} = \frac{1}{K} \sum_{k=1}^{K} J_{\mathcal{Z}_k}^{(K\text{-CV})}.$$

For parameter selection, we compute $J^{(K\text{-CV})}$ for all hyper-parameter candidates (the Gaussian width σ and the regularization parameter λ in the current setting) and choose the parameter that minimizes $J^{(K\text{-CV})}$. We can show that $J^{(K\text{-CV})}$ is an almost unbiased estimator of the objective function in Eq.(5), where the 'almost'-ness comes from the fact that the number of samples is reduced in the CV procedure due to data splitting (Geisser, 1975; Kohave, 1995).

3 The LICA Algorithms

In this section, we show how the above SMI estimation idea could be employed in the context of ICA. Here, we derive two algorithms, which we call *Least-squares Independent Component Analysis* (LICA), for obtaining a minimizer of $\widehat{I}_s$ with respect to the demixing matrix $\boldsymbol{W}$—one is based on a *plain gradient* method (which we refer to as *PG-LICA*) and the other is based on a *natural gradient* method for whitened samples (which we refer to as *NG-LICA*). A MATLAB® implementation of LICA is available from

`http://www.simplex.t.u-tokyo.ac.jp/~s-taiji/software/LICA/index.html`

3.1 Plain Gradient Algorithm: PG-LICA

Based on the plain gradient technique, an update rule of $\boldsymbol{W}$ is given by

$$\boldsymbol{W} \longleftarrow \boldsymbol{W} - \varepsilon \frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}}, \tag{13}$$

where ε (> 0) is the step size. As shown in Appendix, the gradient is given by

$$\frac{\partial \widehat{I}_s}{\partial W_{\ell,\ell'}} = \frac{\partial \widehat{\boldsymbol{h}}^\top}{\partial W_{\ell,\ell'}} \widehat{\boldsymbol{\alpha}} - \frac{1}{2} \widehat{\boldsymbol{\alpha}}^\top \left(\frac{\partial \widehat{\boldsymbol{H}}}{\partial W_{\ell,\ell'}} + \lambda \frac{\partial \boldsymbol{R}}{\partial W_{\ell,\ell'}} \right) \widehat{\boldsymbol{\alpha}}, \tag{14}$$

where, for $\boldsymbol{u}_\ell = \boldsymbol{y}_{c(\ell)}$ and $\boldsymbol{y}_i = (y_i^{(1)}, \ldots, y_i^{(d)})^\top$,

$$\frac{\partial \widehat{h}_\ell}{\partial W_{k,k'}} = \frac{1}{n\sigma^2} \sum_{i=1}^{n} (z_i^{(k)} - v_\ell^{(k)})(u_\ell^{(k')} - y_i^{(k')}) \exp\left(-\frac{\|\boldsymbol{z}_i - \boldsymbol{v}_k\|^2}{2\sigma^2} \right), \tag{15}$$

$$\begin{aligned} \frac{\partial \widehat{H}_{\ell,\ell'}}{\partial W_{k,k'}} = {} & \frac{1}{n^{d-1}} \prod_{m \neq k} \left[\sum_{i=1}^{n} \exp\left(-\frac{(z_i^{(m)} - v_\ell^{(m)})^2 + (z_i^{(m)} - v_{\ell'}^{(m)})^2}{2\sigma^2} \right) \right] \\ & \times \left[\frac{1}{n\sigma^2} \sum_{i=1}^{n} \left((z_i^{(k)} - v_\ell^{(k)})(u_\ell^{(k')} - y_i^{(k')}) + (z_i^{(k)} - v_{\ell'}^{(k)})(u_{\ell'}^{(k')} - y_i^{(k')}) \right) \right. \\ & \left. \times \exp\left(-\frac{(z_i^{(k)} - v_\ell^{(k)})^2 + (z_i^{(k)} - v_{\ell'}^{(k)})^2}{2\sigma^2} \right) \right]. \end{aligned} \tag{16}$$

For the regularization matrix $\boldsymbol{R}$ defined by Eq.(12), the partial derivative is given by

$$\frac{\partial R_{\ell,\ell'}}{\partial W_{k,k'}} = \frac{1}{\sigma^2} (v_\ell^{(k)} - v_{\ell'}^{(k)})(u_{\ell'}^{(k')} - u_\ell^{(k')}) \exp\left(-\frac{\|\boldsymbol{v}_\ell - \boldsymbol{v}_{\ell'}\|^2}{2\sigma^2} \right).$$

In ICA, scaling of components of $\boldsymbol{z}$ can be arbitrary. This implies that the above gradient updating rule can lead to a solution with poor scaling, which is not preferable from a numerical point of view. To avoid possible numerical instability, we normalize $\boldsymbol{W}$ at each gradient iteration as

$$W_{k,k'} \longleftarrow \frac{W_{k,k'}}{\sqrt{\sum_{m=1}^{d} W_{k,m}^2}}. \tag{17}$$

In practice, we may iteratively perform line search along the gradient and optimize the Gaussian width σ and the regularization parameter λ by CV. A pseudo code of the PG-LICA algorithm is summarized in Figure 1.

1. Initialize demixing matrix $\boldsymbol{W}$ and normalize it by Eq.(17).
2. Optimize Gaussian width σ and regularization parameter λ by CV.
3. Compute gradient $\frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}}$ by Eq.(14).
4. Choose step-size ε such that $\widehat{I}_s$ (see Eq.(10)) is minimized (*line-search*).
5. Update $\boldsymbol{W}$ by Eq.(13).
6. Normalize $\boldsymbol{W}$ by Eq.(17).
7. Repeat 2.–6. until $\boldsymbol{W}$ converges.

Figure 1: The LICA algorithm with plain gradient descent (PG-LICA).

3.2 Natural Gradient Algorithm for Whitened Data: NG-LICA

The second algorithm is based on a *natural gradient* technique (Amari, 1998).

Suppose the data samples are *whitened*, i.e., samples $\{\boldsymbol{y}_i\}_{i=1}^n$ are transformed as

$$\boldsymbol{y}_i \longleftarrow \widehat{\boldsymbol{C}}^{-\frac{1}{2}} \boldsymbol{y}_i, \tag{18}$$

where $\widehat{\boldsymbol{C}}$ is the sample covariance matrix:

$$\widehat{\boldsymbol{C}} := \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{y}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{y}_j \right) \left(\boldsymbol{y}_i - \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{y}_j \right)^\top.$$

Then it can be shown that a demixing matrix which eliminates the second order correlation is an *orthogonal matrix* (Hyvärinen et al., 2001). Thus, for whitened data, the search space of $\boldsymbol{W}$ can be restricted to the orthogonal group $O(d)$ without loss of generality.

The *tangent space* of $O(d)$ at $\boldsymbol{W}$ is equal to the space of all matrices $\boldsymbol{U}$ such that $\boldsymbol{W}^\top \boldsymbol{U}$ is *skew symmetric*, i.e., $\boldsymbol{U}\boldsymbol{W}^\top = -\boldsymbol{W}\boldsymbol{U}^\top$. The steepest direction on this tangent space, which is called the *natural gradient*, is given as follows (Amari, 1998):

$$\nabla \widehat{I}_s(\boldsymbol{W}) := \frac{1}{2} \left(\frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}} - \boldsymbol{W} \frac{\partial \widehat{I}_s}{\partial \boldsymbol{W}}^\top \boldsymbol{W} \right), \tag{19}$$

where the canonical metric $\langle \boldsymbol{G}_1, \boldsymbol{G}_2 \rangle = \frac{1}{2} \mathrm{tr}(\boldsymbol{G}_1^\top \boldsymbol{G}_2)$ is adopted in the tangent space. Then the *geodesic* from $\boldsymbol{W}$ in the direction of the natural gradient over $O(d)$ can be expressed by

$$\boldsymbol{W} \exp\left(t \boldsymbol{W}^\top \nabla \widehat{I}_s(\boldsymbol{W}) \right),$$

where $t \in \mathbb{R}$ and 'exp' denotes the matrix exponential, i.e., for a square matrix $\boldsymbol{D}$,

$$\exp(\boldsymbol{D}) = \sum_{k=0}^{\infty} \frac{1}{k!} \boldsymbol{D}^k.$$

1. Whiten the data samples by Eq.(18).
2. Initialize demixing matrix $\boldsymbol{W}$ and normalize it by Eq.(17).
3. Optimize Gaussian width σ and regularization parameter λ by CV.
4. Compute the natural gradient $\nabla\widehat{I}_s$ by Eq.(19).
5. Choose step-size t such that $\widehat{I}_s$ (see Eq.(10)) is minimized over the set (20).
6. Update $\boldsymbol{W}$ by Eq.(21).
7. Repeat 3.–6. until $\boldsymbol{W}$ converges.

Figure 2: The LICA algorithm with natural gradient descent (NG-LICA).

Thus when we perform line search along the geodesic in the natural gradient direction, the minimizer may be searched from the set

$$\left\{\boldsymbol{W}\exp\left(-t\boldsymbol{W}^\top\nabla\widehat{I}_s(\boldsymbol{W})\right)\;\middle|\; t\geq 0\right\}, \tag{20}$$

i.e., t is chosen such that $\widehat{I}_s$ (see Eq.(10)) is minimized and $\boldsymbol{W}$ is updated as

$$\boldsymbol{W}\longleftarrow\boldsymbol{W}\exp\left(-t\boldsymbol{W}^\top\nabla\widehat{I}_s(\boldsymbol{W})\right). \tag{21}$$

Geometry and optimization algorithms on more general structure, the *Stiefel manifold*, is discussed in more detail in Nishimori and Akaho (2005).

A pseudo code of the NG-LICA algorithm is summarized in Figure 2.

3.3 Remarks

The proposed LICA algorithms can be regarded as an application of the general unconstrained least-squares density-ratio estimator proposed by Kanamori et al. (2009) to SMI in the context of ICA.

The optimization problem (5) can also be obtained following the line of Nguyen et al. (2010), which addresses a divergence estimation problem utilizing the *Legendre-Fenchel duality*. SMI defined by Eq.(1) can be expressed as

$$I_s(Z^{(1)},\ldots,Z^{(d)})=\int\frac{1}{2}\left(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}\right)^2 r(\boldsymbol{z})\mathrm{d}\boldsymbol{z}-\frac{1}{2}. \tag{22}$$

If the Legendre-Fenchel duality of the convex function $\frac{1}{2}x^2$,

$$\frac{1}{2}x^2=\sup_y\left(yx-\frac{1}{2}y^2\right),$$

is applied to $\frac{1}{2}\left(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}\right)^2$ in Eq.(22) in a pointwise manner, we have

$$\begin{aligned} I_s(Z^{(1)},\ldots,Z^{(d)}) &= \sup_g \left[\int \Big(\frac{q(\boldsymbol{z})}{r(\boldsymbol{z})}g(\boldsymbol{z}) - \frac{1}{2}g(\boldsymbol{z})^2\Big)r(\boldsymbol{z})\mathrm{d}\boldsymbol{z} - \frac{1}{2}\right] \\ &= -\inf_g \left[\int \Big(\frac{1}{2}g(\boldsymbol{z})^2 q(\boldsymbol{z}) - g(\boldsymbol{z})r(\boldsymbol{z})\Big)\mathrm{d}\boldsymbol{z}\right] - \frac{1}{2}, \end{aligned}$$

where $\sup_g$ and $\inf_g$ are taken over all measurable functions.

SMI is closely related to the kernel independence measures developed recently (Gretton et al., 2005a; Gretton et al., 2005b; Fukumizu et al., 2008). In particular, it has been shown that the *NOrmalized Cross-Covariance Operator* (NOCCO) proposed in Fukumizu et al. (2008) is also an estimator of SMI for $d = 2$. However, there is no reasonable hyper-parameter selection method for this and all other kernel-based independence measures (see also Bach & Jordan, 2002 and Fukumizu et al., 2009). This is a crucial limitation in unsupervised learning scenarios such as ICA. On the other hand, cross-validation can be applied to our method for hyper-parameter selection, as shown in Section 2.3.

4 Experiments

In this section, we investigate the experimental performance of the proposed method.

4.1 Illustrative Examples

First, we illustrate how the proposed method behaves using the following three 2-dimensional datasets:

(a) Sub·Sub-Gaussians: $p(\boldsymbol{x}) = U(x^{(1)}; -0.5, 0.5)U(x^{(2)}; -0.5, 0.5)$,

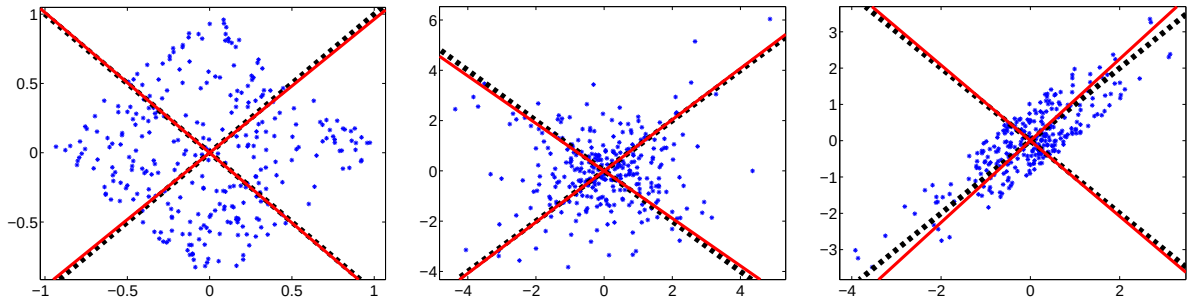
(b) Super·Super-Gaussians: $p(\boldsymbol{x}) = L(x^{(1)}; 0, 1)L(x^{(2)}; 0, 1)$,

(c) Sub·Super-Gaussians: $p(\boldsymbol{x}) = U(x^{(1)}; -0.5, 0.5)L(x^{(2)}; 0, 1)$,

where $U(x; a, b)$ $(a, b \in \mathbb{R}, a < b)$ denotes the uniform density on $[a, b]$ and $L(x; \mu, v)$ $(\mu \in \mathbb{R}, v > 0)$ denotes the Laplace density with mean μ and variance v. Let the number of samples be $n = 300$ and we observe mixed samples $\{\boldsymbol{y}_i\}_{i=1}^n$ through the following mixing matrix:

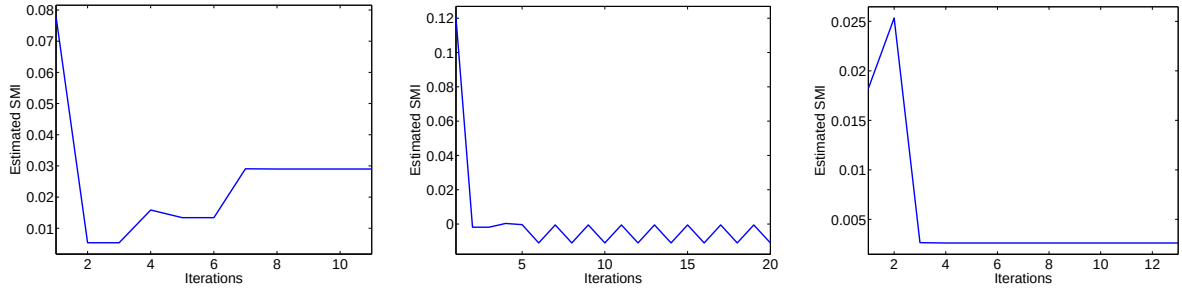
$$\boldsymbol{A} = \begin{pmatrix} \cos(\pi/4) & \sin(\pi/4) \\ -\sin(\pi/4) & \cos(\pi/4) \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

The observed samples are plotted in Figure 3. We employed the NG-LICA algorithm described in Figure 2. Hyper-parameters σ and λ in LICA were chosen by 5-fold CV from the 10 values in $[0.1, 1]$ at regular intervals and the 10 values in $[0.001, 1]$ at regular intervals in log scale, respectively. The regularization term was set to the squared RKHS norm induced by the Gaussian kernel, i.e., we employed $\boldsymbol{R}$ defined by Eq.(12).



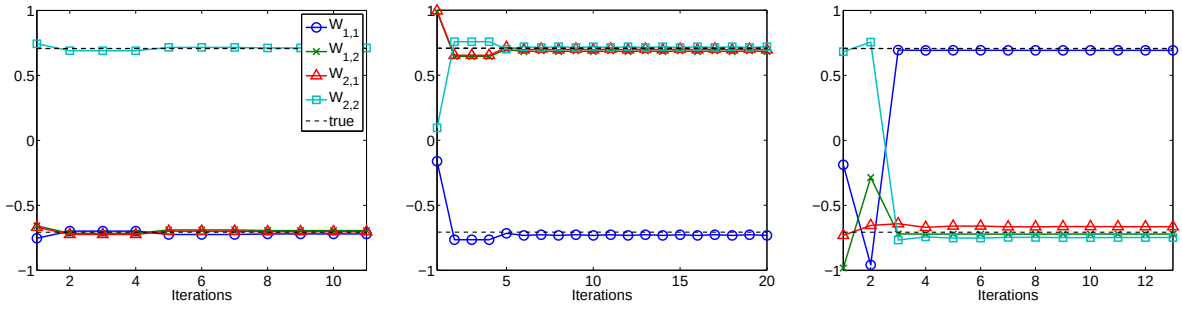
(a) (b) (c)

Figure 3: Observed samples (asterisks), true independent directions (dotted lines) and estimated independent directions (solid lines).



(a) (b) (c)

Figure 4: The value of $\widehat{I}_s$ over iterations.



(a) (b) (c)

Figure 5: The elements of the demixing matrix $\boldsymbol{W}$ over iterations. Solid lines correspond to $W_{1,1}$, $W_{1,2}$, $W_{2,1}$, and $W_{2,2}$, respectively. The dotted lines denote the true values.

The true independent directions as well as the estimated independent directions are plotted in Figure 3. Figure 4 depicts the value of the estimated SMI (10) over iterations and Figure 5 depicts the elements of the demixing matrix $\boldsymbol{W}$ over iterations. The results show that estimated SMI decreases rapidly and good solutions are obtained for all the datasets. The reason why the estimated SMI in Figure 4 does not decrease monotonically is that during the natural gradient optimization procedure, the hyper-parameters (λ and σ) are adjusted by CV (see Figure 2), which possibly causes an increase in the objective values.

4.2 Performance Comparison

Here we compare our method with some existing methods (KICA, FICA, JADE (Cardoso & Souloumiac, 1993)) on artificial and real datasets. We used the datasets (a), (b), and (c) in Section 4.1, the 'demosig' dataset available in the FastICA package[1] for MATLAB®, and '10halo', 'Sergio7', 'Speech4', and 'c5signals' datasets available in the ICALAB signal processing benchmark datasets[2] (Cichocki & Amari, 2003). The datasets (a), (b), (c), 'demosig', Sergio7', and 'c5signals' are artificial datasets. The datasets '10halo' and 'Speech4' are real datasets. We employed the *Amari index* (Amari et al., 1996) as the performance measure (smaller is better):

$$\text{Amari index} := \frac{1}{2d(d-1)} \sum_{m,m'=1}^{d} \left(\frac{|o_{m,m'}|}{\max_{m''} |o_{m,m''}|} + \frac{|o_{m,m'}|}{\max_{m''} |o_{m'',m'}|} \right) - \frac{1}{d-1},$$

where $o_{m,m'} := [\widehat{\boldsymbol{W}}\boldsymbol{A}]_{m,m'}$ for an estimated demixing matrix $\widehat{\boldsymbol{W}}$. We used the publicly available MATLAB® codes for KICA[3], FICA[1] and JADE[4], where default parameter settings were used. Hyper-parameters σ and λ in LICA were chosen by 5-fold CV from the 10 values in $[0.1, 1]$ at regular intervals and the 10 values in $[0.001, 1]$ at regular intervals in log scale, respectively. $\boldsymbol{R}$ was set as Eq.(12).

We randomly generated the mixing matrix $\boldsymbol{A}$ and source signals for artificial datasets, and computed the Amari index between the true $\boldsymbol{A}$ and $\widehat{\boldsymbol{W}}^{-1}$ for $\widehat{\boldsymbol{W}}$ estimated by each method. As training samples, we used the first n samples for Sergio7 and c5signals, and the n samples between the 1001th and (1000+n)-th interval for 10halo and Speech4, where we tested $n = 200$ and 500.

The performance of each method is summarized in Table 2, which depicts the mean and standard deviation of the Amari index over 50 trials. NG-LICA overall shows good performance. KICA tends to work reasonably well for datasets (a), (b), (c) and 'demosig', but it performs poorly for the ICALAB datasets; this seems to be caused by an inappropriate choice of the Gaussian kernel width and local optima. On the other hand, FICA and JADE tend to work reasonably well for the ICALAB datasets, but performs poorly

[1] http://www.cis.hut.fi/projects/ica/fastica
[2] http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/benchmarks/
[3] http://www.di.ens.fr/~fbach/kernel-ica/index.htm
[4] http://perso.telecom-paristech.fr/ cardoso/guidesepsou.html

Table 2: Mean and standard deviation of the Amari index (smaller is better) for the benchmark datasets. The datasets (a), (b), and (c) are taken from Section 4.1. The 'demosig' dataset is taken from the FastICA package. The '10halo', 'Sergio7', 'Speech4', and 'c5signals' datasets are taken from the ICALAB benchmarks datasets. The best method in terms of the mean Amari index and comparable ones based on the one-sided t-test at the significance level 1% are indicated by boldface.

dataset	n	NG-LICA	KICA	FICA	JADE
(a)	200	**0.05(0.03)**	**0.04(0.02)**	0.06(0.03)	**0.04(0.02)**
	500	**0.03(0.01)**	**0.03(0.01)**	**0.03(0.02)**	**0.02(0.01)**
(b)	200	**0.06(0.04)**	0.12(0.15)	0.16(0.20)	0.15(0.17)
	500	**0.04(0.03)**	**0.05(0.04)**	0.11(0.12)	**0.05(0.04)**
(c)	200	**0.08(0.05)**	**0.09(0.06)**	0.14(0.11)	0.13(0.09)
	500	**0.04(0.03)**	**0.04(0.03)**	0.09(0.08)	0.10(0.06)
demosig	200	**0.04(0.01)**	**0.05(0.11)**	0.08(0.05)	0.08(0.08)
	500	**0.02(0.01)**	**0.04(0.09)**	0.04(0.03)	0.04(0.02)
10halo	200	**0.29(0.02)**	0.38(0.03)	0.33(0.07)	0.36(0.00)
	500	**0.22(0.02)**	0.37(0.03)	**0.22(0.03)**	0.28(0.00)
Sergio7	200	**0.04(0.01)**	0.38(0.04)	0.05(0.02)	0.07(0.00)
	500	0.05(0.02)	0.37(0.03)	0.04(0.01)	**0.04(0.00)**
Speech4	200	**0.18(0.03)**	0.29(0.05)	0.20(0.03)	0.22(0.00)
	500	0.07(0.00)	0.10(0.04)	0.10(0.04)	**0.06(0.00)**
c5signals	200	0.12(0.01)	0.25(0.15)	**0.10(0.02)**	0.12(0.00)
	500	**0.06(0.04)**	0.07(0.06)	**0.04(0.02)**	0.07(0.00)

for (a), (b), (c) and 'demosig'; we conjecture that the contrast functions in FICA and the fourth-order statistics in JADE did not appropriately catch the non-Gaussianity of the datasets (a), (b), (c) and 'demosig'. Overall, the proposed LICA algorithm is shown to be a promising ICA method.

5 Conclusions

In this paper, we proposed a new ICA method based on a squared-loss variant of mutual information. The proposed method, named least-squares ICA (LICA), has several preferable properties, e.g., it is distribution-free and hyper-parameter selection by cross-validation is available.

Similarly to other ICA algorithms, the optimization problem involved in LICA is non-convex. Thus it is practically very important to develop good heuristics for initialization and avoiding local optima in the gradient procedures, which is an open research topic to be investigated. Moreover, although our SMI estimator is analytic, the LICA algorithm is still computationally rather expensive due to linear equations and cross-validation. Our

future work will address the computational issue, e.g., by vectorization and parallelization.

Appendix: Derivation of the Gradient of the SMI Estimator

Here we show the derivation of the gradient (14) of the SMI estimator (10). Since $\widehat{I}_s = \frac{1}{2}\widehat{\boldsymbol{h}}^\top \widehat{\boldsymbol{\alpha}} + \frac{1}{2}$ (see Eq.(10)), the derivative of $\widehat{I}_s$ with respect to $W_{k,k'}$ is given as follows:

$$\frac{\partial \widehat{I}_s}{\partial W_{k,k'}} = \frac{1}{2}\widehat{\boldsymbol{h}}^\top \frac{\partial \widehat{\boldsymbol{\alpha}}}{\partial W_{k,k'}} + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}}. \tag{23}$$

Remind that $\frac{\mathrm{d}\boldsymbol{B}(x)^{-1}}{\mathrm{d}x} = -\boldsymbol{B}(x)^{-1}\frac{\mathrm{d}\boldsymbol{B}(x)}{\mathrm{d}x}\boldsymbol{B}(x)^{-1}$ for an arbitrary matrix function $\boldsymbol{B}(x)$. Then the partial derivative of $\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\widehat{\boldsymbol{h}}$ with respect to $W_{k,k'}$ is given by

$$\begin{aligned}\frac{\partial \widehat{\boldsymbol{\alpha}}}{\partial W_{k,k'}} &= -(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})}{\partial W_{k,k'}}(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\widehat{\boldsymbol{h}} + (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}} \\ &= -(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} + (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}}.\end{aligned}$$

Substituting this in Eq.(23), we have

$$\begin{aligned}\frac{\partial \widehat{I}_s}{\partial W_{k,k'}} =& \frac{1}{2}\widehat{\boldsymbol{h}}^\top \left(-(\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} + (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{R})^{-1}\frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}} \right) + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}} \\ =& -\frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{H}}}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} - \frac{\lambda}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \boldsymbol{R}}{\partial W_{k,k'}}\widehat{\boldsymbol{\alpha}} + \widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\boldsymbol{h}}}{\partial W_{k,k'}},\end{aligned}$$

which gives Eq.(14).

Acknowledgments

The authors would like to thank Dr. Takafumi Kanamori for his valuable comments. T.S. was supported in part by the JSPS Research Fellowships for Young Scientists and Global COE Program "The research and training center for new development in mathematics", MEXT, Japan. M.S. acknowledges support from SCAT, AOARD, and the JST PRESTO program.

References

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, *28*, 131–142.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*, 251–276.

Amari, S., Cichocki, A., & Yang, H. H. (1996). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems* (pp. 757–763). MIT Press.

Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, *3*, 1–48.

Cardoso, J.-F., & Souloumiac, A. (1993). Blind beamforming for non-Gaussian signals. *Radar and Signal Processing, IEE Proceedings-F*, *140*, 362–370.

Cichocki, A., & Amari, S. (2003). *Adaptive blind signal and image processing: Learning algorithms and applications*. Wiley.

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Non-negative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. New York: Wiley.

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

Csiszár, I. (1967). Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, *2*, 229–318.

Fukumizu, K., Bach, F. R., & Jordan, M. I. (2009). Kernel dimension reduction in regression. *The Annals of Statistics*, *37*, 1871–1905.

Fukumizu, K., Gretton, A., Sun, X., & Schölkopf, B. (2008). Kernel measures of conditional dependence. *Advances in Neural Information Processing Systems 20* (pp. 489–496). Cambridge, MA: MIT Press.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, *70*, 320–328.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005a). Measuring statistical dependence with Hilbert-Schmidt norms. *Algorithmic Learning Theory* (pp. 63–77). Berlin: Springer-Verlag.

Gretton, A., Herbrich, R., Smola, A., Bousquet, O., & Schölkopf, B. (2005b). Kernel methods for measuring independence. *Journal of Machine Learning Research*, *6*, 2075–2129.

Hulle, M. M. V. (2008). Sequential fixed-point ICA based on mutual information minimization. *Neural Computation*, *20*, 1344–1365.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, *10*, 626–634.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis*. New York: Wiley.

Jutten, C., & Herault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, *24*, 1–10.

Kanamori, T., Hido, S., & Sugiyama, M. (2009). A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, *10*, 1391–1445.

Kohave, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1137–1143). Morgan Kaufmann.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, *22*, 79–86.

Liese, F., & Vajda, I. (2006). On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, *52*, 4394–4412.

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*. to appear.

Nishimori, Y., & Akaho, S. (2005). Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, *67*, 106–135.

Paninsky, L. (2003). Estimation of entropy and mutual information. *Neural Computation*, *15*, 1191–1253.

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, *50*, 157–172.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., & Kawanabe, M. (2008). Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, *60*, 699–746.

Suzuki, T., Sugiyama, M., Sese, J., & Kanamori, T. (2008). Approximating mutual information by maximum likelihood density ratio estimation. *New Challenges for Feature Selection in Data Mining and Knowledge Discovery* (pp. 5–20).

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.