

Least-Squares Independent Component Analysis

Taiji Suzuki

s-taiji@stat.t.u-tokyo.ac.jp

*Department of Mathematical Informatics, University of Tokyo, Bunkyo-ku,
Tokyo 113-8656, Japan*

Masashi Sugiyama

sugi@cs.titech.ac.jp

*Department of Computer Science, Tokyo Institute of Technology and PRESTO,
Japan Science and Technology Agency, Meguro-ku, Tokyo 152-8552, Japan*

Accurately evaluating statistical independence among random variables is a key element of independent component analysis (ICA). In this letter, 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, thereby avoiding the difficult task of density estimation. In this density ratio approach, a natural cross-validation procedure is available for hyperparameter 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*.

1 Introduction ---

The purpose of independent component analysis (ICA) (Hyvärinen, Karhunen, & Oja, 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).

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 nonparametric density estimation methods. However, finding an appropriate parametric model is not easy without strong prior knowledge, and nonparametric 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, Cichocki, & Yang, 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 nonlinear correlation: signals are statistically independent if and only if all nonlinear correlations among the signals vanish. Following this line, computationally efficient algorithms have been developed based on a contrast function (Jutten & Héroult, 1991; Hyvärinen, 1999), which is an approximation of negentropy or mutual information. However, these methods require prespecifying nonlinearities in the contrast function and thus could be inaccurate if the predetermined nonlinearities do not match the target distribution. To cope with this problem, the kernel trick has been applied in ICA, which allows evaluating all nonlinear 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 to be no theoretically justified method to determine the kernel width (see also Fukumizu, Bach, & Jordan, 2009). This is a critical problem in unsupervised learning problems such as ICA.

In this letter, we develop a new ICA algorithm that resolves these problems. 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, Hido, and Sugiyama (2009), and Nguyen, Wainwright, and Jordan (in press), allows us to avoid the difficult task of density estimation. Another practical advantage of this density-ratio approach is that a natural cross-validation (CV) procedure is available for hyperparameter 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 nonparametric estimator of SMI; furthermore, its derivative can also be computed analytically, and these properties are used 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 letter is as follows. In section 2, we formulate our estimator of SMI. In section 3, we derive the LICA algorithm based on the

Table 1: Summary of Existing and Proposed ICA Methods.

	Hyperparameter 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 estimator. Section 4 is devoted to numerical experiments where we show that our method properly estimates the true demixing matrix using toy data sets and compare the performances of the proposed and existing methods on artificial and real data sets.

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,

$$\mathbf{x} = (x^{(1)}, \dots, x^{(d)})^\top,$$

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

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

We cannot directly observe the source signal \mathbf{x} , only a linearly mixed signal \mathbf{y} :

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

where \mathbf{A} is a $d \times d$ invertible matrix called the mixing matrix. The goal of ICA is, given samples of the mixed signals $\{\mathbf{y}_i\}_{i=1}^n$, to obtain a demixing

matrix W that recovers the original source signal x . We denote the demixed signal by z :

$$z = Wy.$$

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

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

$$I_s(Z^{(1)}, \dots, Z^{(d)}) := \frac{1}{2} \int \left(\frac{q(z)}{r(z)} - 1 \right)^2 r(z) dz, \quad (2.1)$$

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

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

Note that SMI is the Pearson divergence (Pearson, 1900; Paninsky, 2003; Liese & Vajda, 2006; Cichocki, Zdunek, Phan, & Amari, 2009) between $q(z)$ and $r(z)$, while ordinary MI is the Kullback-Leibler divergence (Kullback & Leibler, 1951). Since I_s is nonnegative and it vanishes if and only if $q(z) = r(z)$, the degree of independence among $\{z^{(m)}\}_{m=1}^d$ may be measured by SMI. Note that equation 2.1 corresponds to the f -divergence (Ali & Silvey, 1966; Csiszár, 1967) between $q(x)$ and $r(z)$ 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 W that minimizes SMI. Let us denote the demixed samples by

$$\{z_i \mid z_i = (z_i^{(1)}, \dots, z_i^{(d)})^\top := Wy_i\}_{i=1}^n.$$

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

2.2 SMI Approximation by Density Ratio Estimation. We approximate SMI by density ratio estimation. Let us denote the ratio of the densities $q(z)$ and $r(z)$ by

$$g^*(z) := \frac{q(z)}{r(z)}. \tag{2.2}$$

Then SMI can be written as

$$\begin{aligned} I_s(Z^{(1)}, \dots, Z^{(d)}) &= \frac{1}{2} \int (g^*(z) - 1)^2 r(z) dz \\ &= \frac{1}{2} \int (g^*(z)^2 r(z) - 2g^*(z)r(z) + r(z)) dz \\ &= \frac{1}{2} \int (g^*(z)q(z) - 2q(z) + r(z)) dz \\ &= \frac{1}{2} \int g^*(z)q(z) dz - \frac{1}{2}. \end{aligned} \tag{2.3}$$

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

$$\hat{I}_s = \frac{1}{2n} \sum_{i=1}^n \hat{g}(z_i) - \frac{1}{2}. \tag{2.4}$$

We take the least squares approach to estimating the density ratio $g^*(z)$:

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

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:

$$\inf_g \left[\int \left(\frac{1}{2} g(z)^2 r(z) - g(z)q(z) \right) dz \right]. \tag{2.5}$$

However, directly solving problem 2.5 is not possible for two reasons. First, 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} ,

$$\mathcal{G} = \{\alpha^\top \boldsymbol{\varphi}(z) \mid \alpha = (\alpha_1, \dots, \alpha_b)^\top \in \mathbb{R}^b\}, \quad (2.6)$$

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

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

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

The second reason that directly solving problem 2.5 is not possible is that the expectations over the true probability densities $q(z)$ and $r(z)$ cannot be computed since $q(z)$ and $r(z)$ are unknown. To cope with this problem, we approximate the expectations by their empirical averages. Then the optimization problem is reduced to

$$\hat{\alpha} := \operatorname{argmin}_{\alpha \in \mathbb{R}^b} \left[\frac{1}{2} \alpha^\top \hat{\mathbf{H}} \alpha - \hat{\mathbf{h}}^\top \alpha + \lambda \alpha^\top \mathbf{R} \alpha \right], \quad (2.7)$$

where we include $\lambda \alpha^\top \mathbf{R} \alpha$ ($\lambda > 0$) for avoiding overfitting. λ is called the regularization parameter, and \mathbf{R} is some positive definite matrix. $\hat{\mathbf{H}}$ and $\hat{\mathbf{h}}$ are defined as

$$\hat{\mathbf{H}} := \frac{1}{n^d} \sum_{i_1, \dots, i_d=1}^n \boldsymbol{\varphi}(z_{i_1}^{(1)}, \dots, z_{i_d}^{(d)}) \boldsymbol{\varphi}(z_{i_1}^{(1)}, \dots, z_{i_d}^{(d)})^\top, \quad (2.8)$$

$$\hat{\mathbf{h}} := \frac{1}{n} \sum_{i=1}^n \boldsymbol{\varphi}(z_i^{(1)}, \dots, z_i^{(d)}). \quad (2.9)$$

Differentiating the objective function in equation 2.7 with respect to α and equating it to zero, we can obtain an analytic form solution as

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

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

Once the density ratio 2.2 has been estimated, SMI can be approximated by plugging the estimated density ratio $\hat{g}(z) = \hat{\alpha}^\top \boldsymbol{\varphi}(z)$ in equation 2.4:

$$\hat{I}_s = \frac{1}{2} \hat{\alpha}^\top \hat{\mathbf{h}} - \frac{1}{2}. \quad (2.10)$$

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

$$\int g^*(z)^2 r(z) dz = \int g^*(z) q(z) dz,$$

$$\int g^*(z) r(z) dz = \int q(z) dz = 1.$$

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

2.3 Design of Basis Functions and Hyperparameter Selection. As basis functions, we propose to use a gaussian kernel:

$$\varphi_\ell(z) = \exp\left(-\frac{\|z - \mathbf{v}_\ell\|^2}{2\sigma^2}\right) = \prod_{m=1}^d \exp\left(-\frac{(z^{(m)} - v_\ell^{(m)})^2}{2\sigma^2}\right), \quad (2.11)$$

where

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

are gaussian centers randomly chosen from $\{z_i\}_{i=1}^n$. More precisely, we set $\mathbf{v}_\ell = z_{c(\ell)}$, where $\{c(\ell)\}_{\ell=1}^b$ are randomly chosen from $\{1, \dots, n\}$ without replacement. An advantage of the gaussian kernel lies in the factorizability in equation 2.11, contributing to reducing the computational cost of the matrix \widehat{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 reproducing kernel Hilbert space (RKHS) norm of $\boldsymbol{\alpha}^\top \boldsymbol{\varphi}(z)$ induced by the gaussian kernel as the regularization term $\boldsymbol{\alpha}^\top \mathbf{R} \boldsymbol{\alpha}$, a popular choice in the kernel method community (Schölkopf & Smola, 2002):

$$R_{\ell, \ell'} = \exp\left(-\frac{\|\mathbf{v}_\ell - \mathbf{v}_{\ell'}\|^2}{2\sigma^2}\right). \quad (2.12)$$

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 $\{z_i\}_{i=1}^n$ 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 $\hat{\alpha}_{\mathcal{Z} \setminus \mathcal{Z}_k}$ is obtained using $\mathcal{Z} \setminus \mathcal{Z}_k$ (i.e., \mathcal{Z} without \mathcal{Z}_k), and the approximation error for the holdout samples \mathcal{Z}_k is computed:

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

where the matrix $\hat{\mathbf{H}}_{\mathcal{Z}_k}$ and the vector $\hat{\mathbf{h}}_{\mathcal{Z}_k}$ are defined in the same way as $\hat{\mathbf{H}}$ and $\hat{\mathbf{h}}$, but computed using only \mathcal{Z}_k . This procedure is repeated for $k = 1, \dots, K$, 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 hyperparameter 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 equation 2.5, 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 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 \hat{I}_s with respect to the demixing matrix \mathbf{W} . One is based on a plain gradient method (which we refer to as PG-LICA) and the other on a natural gradient method for whitened samples (which we refer to as NG-LICA). A Matlab implementation of LICA is available online at <http://www.simplex.t.u-tokyo.ac.jp/~s-tajiri/software/LICA/index.html>.

3.1 Plain Gradient Algorithm: PG-LICA. Based on the plain gradient technique, an update rule of \mathbf{W} is given by

$$\mathbf{W} \leftarrow \mathbf{W} - \varepsilon \frac{\partial \hat{I}_s}{\partial \mathbf{W}}, \quad (3.1)$$

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

$$\frac{\partial \widehat{I}_s}{\partial W_{\ell, \ell'}} = \frac{\partial \widehat{\mathbf{h}}^\top}{\partial W_{\ell, \ell'}} \widehat{\boldsymbol{\alpha}} - \frac{1}{2} \widehat{\boldsymbol{\alpha}}^\top \left(\frac{\partial \widehat{\mathbf{H}}}{\partial W_{\ell, \ell'}} + \lambda \frac{\partial \mathbf{R}}{\partial W_{\ell, \ell'}} \right) \widehat{\boldsymbol{\alpha}}, \quad (3.2)$$

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

$$\frac{\partial \widehat{h}_\ell}{\partial W_{k, k'}} = \frac{1}{n\sigma^2} \sum_{i=1}^n (z_i^{(k)} - v_\ell^{(k)})(u_\ell^{(k')} - y_i^{(k')}) \exp\left(-\frac{\|z_i - \mathbf{v}_\ell\|^2}{2\sigma^2}\right), \quad (3.3)$$

$$\begin{aligned} \frac{\partial \widehat{H}_{\ell, \ell'}}{\partial W_{k, k'}} &= \frac{1}{n^{d-1}} \prod_{m \neq k} \left[\sum_{i=1}^n \exp\left(-\frac{(z_i^{(m)} - v_\ell^{(m)})^2 + (z_i^{(m)} - v_{\ell'}^{(m)})^2}{2\sigma^2}\right) \right] \\ &\times \left[\frac{1}{n\sigma^2} \sum_{i=1}^n ((z_i^{(k)} - v_\ell^{(k)})(u_\ell^{(k')} - y_i^{(k')}) + (z_i^{(k)} - v_{\ell'}^{(k)})(u_{\ell'}^{(k')} - y_i^{(k')})) \right] \\ &\times \exp\left(-\frac{(z_i^{(k)} - v_\ell^{(k)})^2 + (z_i^{(k)} - v_{\ell'}^{(k)})^2}{2\sigma^2}\right) \Big]. \end{aligned} \quad (3.4)$$

For the regularization matrix \mathbf{R} defined by equation 2.12, the partial derivative is given by

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

In ICA, scaling of components of \mathbf{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 \mathbf{W} at each gradient iteration as

$$W_{k, k'} \leftarrow \frac{W_{k, k'}}{\sqrt{\sum_{m=1}^d W_{k, m}^2}}. \quad (3.5)$$

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 algorithm 1.

3.2 Natural Gradient Algorithm for Whitened Data: NG-LICA. The second algorithm is based on a *natural gradient* technique (Amari, 1998).

Algorithm 1: LICA Algorithm with Plain Gradient Descent (PG-LICA).

1. Initialize demixing matrix \mathbf{W} and normalize it by equation 3.5.
2. Optimize gaussian width σ and regularization parameter λ by CV.
3. Compute gradient $\frac{\partial \hat{I}_s}{\partial \mathbf{W}}$ by equation 3.2.
4. Choose step size ε such that \hat{I}_s (see equation 2.10) is minimized (*line-search*).
5. Update \mathbf{W} by equation 3.1.
6. Normalize \mathbf{W} by equation 3.5.
7. Repeat steps 2–6 until \mathbf{W} converges.

Suppose the data samples are whitened, that is, samples $\{\mathbf{y}_i\}_{i=1}^n$ are transformed as

$$\mathbf{y}_i \leftarrow \hat{\mathbf{C}}^{-\frac{1}{2}} \mathbf{y}_i, \quad (3.6)$$

where $\hat{\mathbf{C}}$ is the sample covariance matrix:

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

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

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

$$\nabla_{\hat{I}_s}(\mathbf{W}) := \frac{1}{2} \left(\frac{\partial \hat{I}_s}{\partial \mathbf{W}} - \mathbf{W} \frac{\partial \hat{I}_s}{\partial \mathbf{W}}^\top \mathbf{W} \right), \quad (3.7)$$

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

$$\mathbf{W} \exp(t \mathbf{W}^\top \nabla_{\hat{I}_s}(\mathbf{W})),$$

Algorithm 2: The LICA Algorithm with Natural Gradient Descent (NG-LICA).

1. Whiten the data samples by equation 3.6.
2. Initialize demixing matrix \mathbf{W} , and normalize it by equation 3.5.
3. Optimize gaussian width σ and regularization parameter λ by CV.
4. Compute the natural gradient $\nabla \widehat{I}_s$ by equation 3.7.
5. Choose step-size t such that \widehat{I}_s (see equation 2.10) is minimized over the set 3.8.
6. Update \mathbf{W} by equation 3.9.
7. Repeat steps 3–6 until \mathbf{W} converges.

where $t \in \mathbb{R}$ and \exp denotes the matrix exponential, that is, for a square matrix \mathbf{D} ,

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

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

$$\{\mathbf{W} \exp(-t \mathbf{W}^\top \nabla \widehat{I}_s(\mathbf{W})) \mid t \geq 0\}. \quad (3.8)$$

That is, t is chosen such that \widehat{I}_s (see equation 2.10) is minimized and \mathbf{W} is updated as

$$\mathbf{W} \leftarrow \mathbf{W} \exp(-t \mathbf{W}^\top \nabla \widehat{I}_s(\mathbf{W})). \quad (3.9)$$

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 algorithm 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 2.5 can also be obtained following the line of Nguyen et al. (in press), which addresses a divergence estimation problem utilizing the Legendre-Fenchel duality. SMI defined by equation 2.1 can be expressed as

$$I_s(Z^{(1)}, \dots, Z^{(d)}) = \int \frac{1}{2} \left(\frac{q(\mathbf{z})}{r(\mathbf{z})} \right)^2 r(\mathbf{z}) d\mathbf{z} - \frac{1}{2}. \quad (3.10)$$

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 $\frac{1}{2}\left(\frac{q(z)}{r(z)}\right)^2$ in equation 3.10 in a pointwise manner, we have

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

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

SMI is closely related to the kernel independence measures developed recently (Gretton, Bousquet, Smola, & Schölkopf, 2005; Gretton, Herbrich, Smola, Bousquet, & Schölkopf, 2005; Fukumizu et al., 2008). In particular, it has been shown that the normalized cross-covariance operator (NOCCO) proposed in Fukumizu, Gretton, Sun, and Schölkopf (2008) is also an estimator of SMI for $d = 2$. However, there is no reasonable hyperparameter 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 hyperparameter 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 two-dimensional data sets:

- Sub-sub-gaussians: $p(x) = U(x^{(1)}; -0.5, 0.5)U(x^{(2)}; -0.5, 0.5)$
- Super-super-gaussians: $p(x) = L(x^{(1)}; 0, 1)L(x^{(2)}; 0, 1)$
- Sub-super-gaussians: $p(x) = U(x^{(1)}; -0.5, 0.5)L(x^{(2)}; 0, 1)$

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 $\{\mathbf{y}_i\}_{i=1}^n$ through the following mixing matrix:

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

The observed samples are plotted in Figure 1. We employed the NG-LICA algorithm described in algorithm 2. Hyperparameters σ and λ in LICA were chosen by five-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, that is, we employed \mathbf{R} defined by equation 2.12.

The true independent directions as well as the estimated independent directions are plotted in Figure 1. Figure 2 depicts the value of the estimated SMI 2.10 over iterations, and Figure 3 depicts the elements of the demixing matrix \mathbf{W} over iterations. The results show that estimated SMI decreases rapidly and good solutions are obtained for all the data sets. The reason the estimated SMI in Figure 2 does not decrease monotonically is that during the natural gradient optimization procedure, the hyperparameters (λ and σ) are adjusted by CV (see algorithm 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 data sets.¹ We used the three data sets, a, b, and c in section 4.1; the demosig data set available in the FastICA package¹ for Matlab; and 10halo, Sergio7, Speech4, and c5signals data sets available in the ICALAB signal processing benchmark data sets¹ (Cichocki & Amari, 2003). Data sets a, b, and c, demosig, Sergio7, and c5signals are artificial data sets. Data sets 10halo and Speech4 are real data sets. We employed the Amari index (Amari et al., 1996) as the performance measure (smaller is better):

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

¹FastICA: <http://www.cis.hut.fi/projects/ica/fastica/>; ICALAB: <http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/benchmarks/>; KICA: <http://www.di.ens.fr/~fbach/kernel-ica/index.htm>; JADE: <http://perso.telecom-paristech.fr/~cardoso/guidesepsou.html>.

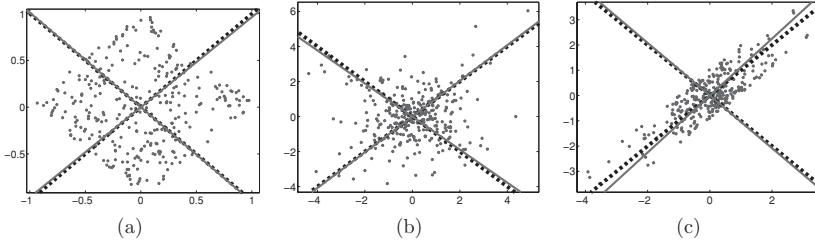


Figure 1: Observed samples (asterisks), true independent directions (dotted lines), and estimated independent directions (solid lines). (For a color version of this figure see the supplemental material, available online at <http://www.mitpressjournals.org/doi/suppl/10.1162/NECO.a.00062>.)

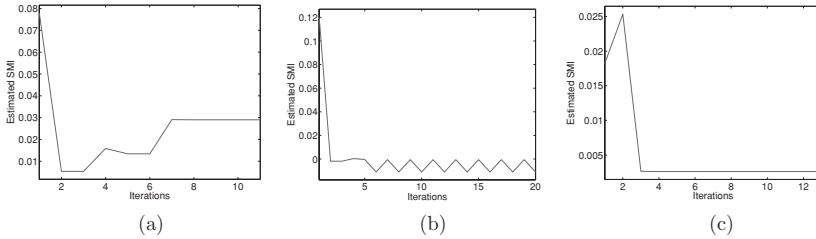


Figure 2: The value of \hat{I}_s over iterations.

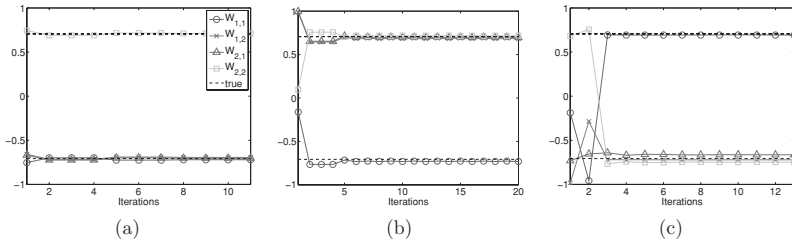


Figure 3: The elements of the demixing matrix W over iterations. Solid lines correspond to $W_{1,1}$, $W_{1,2}$, $W_{2,1}$, and $W_{2,2}$, respectively. The dashed lines denote the true values. (For a color version of this figure see the supplemental material, available online at <http://www.mitpressjournals.org/doi/suppl/10.1162/NECO.a.00062>.)

where $o_{m,m'} := [\hat{W}A]_{m,m'}$ for an estimated demixing matrix \hat{W} . We used the publicly available Matlab codes for KICA, FICA and JADE, where default parameter settings were used. Hyperparameters σ and λ in LICA were chosen by five-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. R was set as equation 2.12.

Table 2: Mean and Standard Deviation of the Amari Index (“Smaller Is Better”) for the Benchmark Data Sets.

Data Set	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)

Notes: Data sets a, b, and c are taken from section 4.1. The demosig data set is from the FastICA package. The 10halo, Sergio7, Speech4, and c5signals data sets are taken from the ICALAB benchmarks data sets. 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 in bold.

We randomly generated the mixing matrix A and source signals for artificial data sets and computed the Amari index between the true A and \widehat{W}^{-1} for \widehat{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 data sets a, b, c, and demosig, but it performs poorly for the ICALAB data sets; this seems to be caused by an inappropriate choice of the gaussian kernel width and local optima. FICA and JADE tend to work reasonably well for the ICALAB data sets, but perform poorly 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 nongaussianity of data sets a, b, c, and demosig. Overall, the proposed LICA algorithm is shown to be a promising ICA method.

5 Conclusion

In this letter, we proposed a new ICA method based on a squared-loss variant of mutual information. The proposed method, least squares ICA

(LICA), has several preferable properties; it is, for example, distribution free and hyperparameter selection by cross-validation is available.

Similar to other ICA algorithms, the optimization problem involved in LICA is nonconvex. Thus, practically it is very important to develop good heuristics for initialization and avoid local optima in the gradient procedures, 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, for example, by vectorization and parallelization.

Appendix: Derivation of the Gradient of the SMI Estimator

Here we show the derivation of gradient 3.2 of the SMI estimator 2.10. Since $\widehat{I}_s = \frac{1}{2}\widehat{\mathbf{h}}^\top \widehat{\boldsymbol{\alpha}} + \frac{1}{2}$ (see equation 2.10), the derivative of \widehat{I}_s with respect to $W_{k,k'}$ is given as

$$\frac{\partial \widehat{I}_s}{\partial W_{k,k'}} = \frac{1}{2}\widehat{\mathbf{h}}^\top \frac{\partial \widehat{\boldsymbol{\alpha}}}{\partial W_{k,k'}} + \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top \frac{\partial \widehat{\mathbf{h}}}{\partial W_{k,k'}}. \quad (\text{A.1})$$

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

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

Substituting this in equation A.1, we have

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

which gives equation 3.2.

Acknowledgments

We thank 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. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in neural information processing systems*, 8 (pp. 757–763). Cambridge, MA: 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*. Hoboken, NJ: 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*. Hoboken, NJ: 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. *Annals of Statistics*, 37, 1871–1905.
- Fukumizu, K., Gretton, A., Sun, X., & Schölkopf, B. (2008). Kernel measures of conditional dependence. In J. C. Platt, D. Köller, Y. Singer, & S. Roweis (Eds.), *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. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In S. Jain, H. U. Simon, & E. Tomita (Eds.), *Algorithmic learning theory* (pp. 63–77). Berlin: Springer-Verlag.
- Gretton, A., Herbrich, R., Smola, A., Bousquet, O., & Schölkopf, B. (2005). 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., & Héroult, 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. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1137–1143). San Francisco: 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 information in statistics and information theory. *IEEE Transactions on Information Theory*, *52*, 4394–4412.
- Nguyen, X., Wainwright, M. J., & Jordan, M. I. (in press). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*.
- 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 Büna, 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. In Y. Saeys, H. Liu, I. Inza, L. Wehenkel, & Y. Van de Peer (Eds.), *New challenges for feature selection in data mining and knowledge discovery* (pp. 5–20). Cambridge, MA: MIT Press.
- Vapnik, V. N. (1998). *Statistical learning theory*. Hoboken, NJ: Wiley.