MICA: Multimodal Independent Component Analysis

Shotaro Akaho† Yasuhiko Kiuchi‡ Shinji Umeyama†
†Electrotechnical Laboratory ‡Saitama University
e-mail : akaho@etl.go.jp

Abstract

We extend the framework of ICA (independent component analysis) to the case that there is a pair of information sources. The goal of MICA is to extract statistically dependent pairs of features from the sources, where the components of feature vector extracted from each source are independent. Therefore, the cost function is constructed to maximize the degree of pairwise dependence as well as optimizing the cost function of ICA. We approximate the cost function by two dimensional Gram-Charlier expansion and propose a gradient descent algorithm derived by Amari's natural gradient. The relation between MICA and traditional CCA (canonical correlation analysis) is similar to the relation between ICA and PCA (principal component analysis).

Introduction

Recently, ICA (independent component analysis) has attracted a lot of attention because of its applications to brain science. One important application is data analysis for biomedical recordings (e.g. EEG, MEG, fMRI), in which artifactual noise components can be eliminated by ICA to extract pure neural brain activity[5]. Another application is extracting features in vision systems. Several kinds of image filters such as edge detector can be learned from natural images by ICA[4], which is also considered to be a model of early vision in the brain.

In the present paper, we consider to extract common features from two sets of multidimensional inputs, whereas ICA deals with only one set of the inputs. It is the case, for example, that we have simultaneous recordings of EEG and MEG.

Traditionally, CCA (canonical correlation analysis) is a widely known method to analyze the correlation structure between two sets of variables (see for example [1]). CCA maximizes the correlation coefficient between each pair of extracted features. Although this is equivalent to maximizing mutual information when given variables are jointly Gaussian-distributed, any nonsingular transformation in the canonical space does not change the amount of the mutual information. In CCA, the problem is partly solved by restricting the covariance matrices of feature vector to identity (sphering condition). However, the freedom of rotation transformation still remains for two axes in which the correlation coefficients are identical. Those assumptions and freedom can make difficult to analyze or to visualize each component of extracted features.

The freedom of rotation also occurs in PCA (principal component analysis), and it has been solved by introducing the *independence* criterion in ICA instead of the sphering condition. We adopt that criterion in MICA, and furthermore the maximization of correlation coefficients is replaced by the maximization of mutual information itself to deal with non-Gaussian signals and nonlinear relation.

In the following sections, first we formulate the framework of MICA, next we derive the gradient descent algorithm and show some simulation results, and we conclude the paper by mentioning further applications and the relation to other methods.

Formulation

MICA can be formulated in a similar way to ICA except that MICA deals with a pair of sources (fig. 1): Let us consider unknown pairs of source signals $(u_i^*(t), v_i^*(t))$, $i = 1, \ldots, n$, where $u_i^*(t)$ and $v_i^*(t)$ are statistically related to each other at any fixed time t, while vector components of $\{u_i^*(t)\}$ and $\{v_i^*(t)\}$ are mutually independent respectively. We assume $(\boldsymbol{u}^*(t), \boldsymbol{v}^*(t))$ is stationary and generated from unknown distribution $p(\boldsymbol{u}^*, \boldsymbol{v}^*)$. The condition of the independence can be written by

$$p(\boldsymbol{u}^*) = \prod_{i=1}^{n} p(u_i^*), \qquad p(\boldsymbol{v}^*) = \prod_{i=1}^{n} p(v_i^*).$$

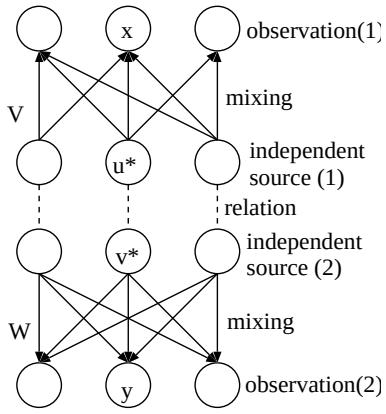

Figure 1: MICA

The degree of dependence between u_i^* and v_i^* can be measured by mutual information as formulated later.

The models of observations $\boldsymbol{x}$, $\boldsymbol{y}$ are given by

$$\boldsymbol{x}(t) = V\boldsymbol{u}^*(t), \qquad \boldsymbol{y}(t) = W\boldsymbol{v}^*(t), \tag{1}$$

where V and W are unknown $n \times n$ nonsingular mixing matrices [1].

The goal of MICA is to recover the source signals by taking linear transformations of observations,

$$\boldsymbol{u}(t) = A\boldsymbol{x}(t), \qquad \boldsymbol{v}(t) = B\boldsymbol{y}(t). \tag{2}$$

Although there remains the ambiguity of the order of pairs and the amplitude of signals, we expect to recover pairwise relation between $u_i(t)$ and $v_i(t)$.

Additive cost function

We have two kinds of constraints: One is to maximize the degree of the statistical dependence between u_i and v_i, and the other is the independence of vector components $\{u_i\}$ and $\{v_i\}$ respectively.

Let us define negative mutual information as a measure of statistical dependence between recovered signals by

$$\mathcal{E}_{\text{dep}} = -\sum_{i=1}^{n} K[p(u_i, v_i)||p(u_i)p(v_i)], \tag{3}$$

where

$$K[p(\boldsymbol{z})||q(\boldsymbol{z})] = \int p(\boldsymbol{z}) \log \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})} \, \mathrm{d}\boldsymbol{z}$$

[1] For the sake of simplicity, we assume that the dimensionality of the observation and the source are identical

denotes Kullback-Leibler divergence which measures the difference between two distributions, and $-\mathcal{E}_{\text{dep}}$ is Shannon's mutual information measuring the amount of information that u_i contains about v_i (and vice versa).

On the other hand, the measure of independence can be defined in the same way as ICA,

$$\mathcal{E}_{\text{ind}} = K[p(\boldsymbol{u})||\prod_{i=1}^{n} p(u_i)] + K[p(\boldsymbol{v})||\prod_{i=1}^{n} p(v_i)], \tag{4}$$

which is equal or larger than zero, and zero is achieved if and only if $\{u_i\}$ and $\{v_i\}$ are independent respectively. Note that $\mathcal{E}_{\text{ind}} = 0$ can not be achieved in general, which is a different situation from the sphering condition in CCA.

Therefore, we have two functions $\mathcal{E}_{\text{dep}}$ and $\mathcal{E}_{\text{ind}}$ to be minimized. A simple way of minimizing two functions simultaneously is to minimize their linear combination,

$$\mathcal{E}_{\text{tot}} = \mathcal{E}_{\text{ind}} + \gamma(t)\mathcal{E}_{\text{dep}}, \tag{5}$$

where $\gamma(t) > 0$ is a weight value. $\mathcal{E}_{\text{tot}}$ can be written in terms of entropy as

$$\begin{aligned} \mathcal{E}_{\text{tot}} \quad = \quad & -H(\boldsymbol{x}) - H(\boldsymbol{y}) - \log|A| - \log|B| \\ & +(1-\gamma)\sum_{i=1}^{n}\{H(u_i) + H(v_i)\} \\ & +\gamma\sum_{i=1}^{n} H(u_i, v_i), \end{aligned} \tag{6}$$

where $H(\boldsymbol{x}) = \mathrm{E}[\log \boldsymbol{x}]$ is a entropy of $\boldsymbol{x}$.

Approximation of the cost function

In order to obtain the gradient descent algorithm, the cost function must be expressed as a function of parameters A and B. In the present paper, we approximate the entropy by using Gram-Charlier expansion. Whereas just one dimensional Gram-Charlier expansion is necessary in ICA for one dimensional marginal entropy $H(u_i)$ and $H(v_i)$ (see Yang et al 1996 in detail), two-dimensional Gram-Charlier expansion is also necessary in MICA, which is complicated in general. In order to get rather simple form of the expansion, first we translate (u_i, v_i) into uncorrelated variables (r_i, s_i) and the gradient is calculated based on $H(r_i, s_i)$.

Decorrelation technique

Without loss of generality, we can assume $\mathrm{E}[u_i] = \mathrm{E}[v_i] = 0$ and $\mathrm{Var}[u_i] = \mathrm{Var}[v_i] = 1$. When they have nonzero means and different variances, we can rescale

them so as to satisfy these conditions.

Since the general form of two-dimensional Gram-Charlier expansion is complicated, we decorrelate (u_i, v_i),

$$\begin{pmatrix} r_i \\ s_i \end{pmatrix} = \begin{pmatrix} c_i^+ & c_i^- \\ c_i^- & c_i^+ \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix}, \tag{7}$$

where

$$\begin{aligned} c_i^+ &= \{(1+\rho_i)^{-1/2} + (1-\rho_i)^{-1/2}\}/2, \\ c_i^- &= \{(1+\rho_i)^{-1/2} - (1-\rho_i)^{-1/2}\}/2, \\ \rho_i &= \mathrm{E}[u_i v_i]. \end{aligned}$$

By this procedure, we obtain

$$H(u_i, v_i) = H(r_i, s_i) + \frac{1}{2}\log(1-\rho_i^2), \tag{8}$$

and

$$\mathrm{E}[r_i s_i] = 0, \qquad \mathrm{Var}[r_i] = \mathrm{Var}[s_i] = 1.$$

Two dimensional Gram-Charlier expansion

We use the following truncated Gram-Charlier expansion,

$$\begin{aligned} p(r_i, s_i) = \phi_i\{1 &+ \\ &\frac{1}{3!}(\beta_i^{3,0} h_i^{3,0} + 3\beta_i^{2,1} h_i^{2,1} 3\beta_i^{1,2} h_i^{1,2} + \beta_i^{0,3} h_i^{0,3}) + \\ &\frac{1}{4!}(\beta_i^{4,0} h_i^{4,0} + 4\beta_i^{3,1} h_i^{3,1} + 6\beta_i^{2,2} h_i^{2,2} + \\ &\quad 4\beta_i^{1,3} h_i^{1,3} + \beta_i^{0,4} h_i^{0,4})\}, \end{aligned}$$

where ϕ_i is 2 dimensional standard Gaussian distribution of r_i and s_i, and $\beta_i^{k,l}$ is a joint cumulant defined by

$$\beta_i^{k,l} = \mathrm{E}[(r_i)^k (s_i)^l] - \beta_0^{k,l}, \tag{9}$$

$$\beta_0^{k,l} = \begin{cases} 3 & \text{if } k=4 \text{ or } l=4, \\ 1 & \text{if } k=l=2, \\ 0 & \text{otherwise,} \end{cases}$$

when $k+l \leq 4$, and r_i and s_i are normalized and uncorrelated; $h_i^{k,l}$ is two dimensional Hermite polynomial of degree (k,l),

$$h_i^{k,l} = \frac{(-1)^{k+l}}{\phi_i} \frac{\partial^{k+l}}{\partial r_i^k s_i^l} \phi_i.$$

The joint entropy of r_i and s_i is approximated by

$$\begin{aligned} H(r_i, s_i) \simeq\ & 1 + \log 2\pi \\ & - \frac{1}{2 \cdot 3!}\left\{ (\beta_i^{3,0})^2 + 3(\beta_i^{2,1})^2 + 3(\beta_i^{1,2})^2 + (\beta_i^{0,3})^2 \right\} \\ & - \frac{1}{2 \cdot 4!}\left\{ (\beta_i^{4,0})^2 + 4(\beta_i^{3,1})^2 + 6(\beta_i^{2,2})^2 \right. \\ & \qquad \left. + 4(\beta_i^{1,3})^2 + (\beta_i^{0,4})^2 \right\}, \end{aligned} \tag{10}$$

up to the second order approximation[2] of $\beta_i^{k,l}$.

Stochastic gradient descent algorithm

Now we obtain the gradient descent by differentiating the approximated cost function.

Learning rule of A is written in the form,

$$\begin{aligned} \frac{\partial A}{\partial t} &= -\eta(t) \frac{\partial \mathcal{E}_{\mathrm{tot}}}{\partial A} A^{\mathrm{T}} A \\ &= \eta(t)\Big[I - (1-\gamma)\boldsymbol{\varphi}^{A}(\boldsymbol{u})\boldsymbol{u}^{\mathrm{T}} + \gamma \boldsymbol{\xi}^{A}(\boldsymbol{u}, \boldsymbol{v})\boldsymbol{u}^{\mathrm{T}} \\ &\qquad + \gamma \mathrm{diag}[\boldsymbol{\psi}^{A}(\boldsymbol{u}, \boldsymbol{v}) + \bar{\boldsymbol{\rho}}] R^{\mathrm{T}} \Big] A, \end{aligned} \tag{11}$$

which is a natural gradient algorithm proposed by Amari, and $\boldsymbol{\varphi}^A$ is just the same which appears in ICA and the other terms are the original/unique terms in MICA as explained later in detail ($\boldsymbol{\xi}^A, \boldsymbol{\psi}^A, \bar{\boldsymbol{\rho}}$ and R are defined in (19), (20), (13) and (14) respectively). Learning rule of B is given in a similar form.

Eq.(11) is an on-line version or stochastic gradient of the algorithm, which is derived by eliminating the average operator E after calculating derivatives (remaining E leads to batch learning).

Linear term

We obtain the gradient for the term $(1/2)\log(1-\rho_i^2)$ in (8) easily, which leads to the last term in (11),

$$-\{\sum_i \frac{1}{2}\frac{\partial}{\partial A}\log(1-\rho_i^2)\} A^{\mathrm{T}} A = \mathrm{diag}[\bar{\boldsymbol{\rho}}] R^{\mathrm{T}} A \tag{12}$$

where

$$\bar{\rho}_i = \frac{\rho_i}{1-\rho_i^2}, \tag{13}$$

and R denotes the covariance matrix of $\boldsymbol{u}$ and $\boldsymbol{v}$,

$$R = \mathrm{E}[\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}]. \tag{14}$$

Note that ρ_i is equal to the diagonal element R_{ii}.

Nonlinear term

The derivative of $H(r_i, s_i)$ can be expressed as

$$\begin{aligned} \frac{\partial H(r_i, s_i)}{\partial A_{ij}} = & \sum_{k,l \in \mathcal{K}} \frac{\partial H(r_i, s_i)}{\partial \beta_i^{k,l}} \times \\ & \mathrm{E}\left[\frac{\partial (r_i)^k (s_i)^l}{\partial r_i} \frac{\partial r_i}{\partial A_{ij}} + \frac{\partial (r_i)^k (s_i)^l}{\partial s_i} \frac{\partial s_i}{\partial A_{ij}} \right], \end{aligned} \tag{15}$$

[2] We need study further whether higher order terms are necessary (the third order consists of 22 terms) from the viewpoint of computational complexity, accuracy, and robustness.

where $\mathcal{K}$ is a set of all indices (k,l) which appear in (10). Then, let

$$f_i = -\sum_{k,l\in\mathcal{K}} \frac{\partial H(r_i,s_i)}{\partial \beta_i^{k,l}} \frac{\partial (r_i)^k (s_i)^l}{\partial r_i}, \quad (16)$$

$$g_i = -\sum_{k,l\in\mathcal{K}} \frac{\partial H(r_i,s_i)}{\partial \beta_i^{k,l}} \frac{\partial (r_i)^k (s_i)^l}{\partial s_i}, \quad (17)$$

where $\partial H/\partial \beta_i^{k,l}$ is obtained easily since $H(r_i,s_i)$ is expressed as the polynomial of $\boldsymbol{\beta}_i$ in (10), (e.g. $\partial H(r_i,s_i)/\partial \beta_i^{3,0} = -\beta_i^{3,0}/6$); hence f_i and g_i are polynomial of $\beta_i^{k,l}$, r_i and s_i. On the other hand, the derivative of (r_i,s_i) by A_{ij} is obtained as

$$\frac{\partial}{\partial A_{ij}} \begin{pmatrix} r_i \\ s_i \end{pmatrix} = \begin{pmatrix} d_i^+ u_i + d_i^- v_i \\ d_i^- u_i + d_i^+ v_i \end{pmatrix} \mathrm{E}[v_i x_j] + \begin{pmatrix} c_i^+ \\ c_i^- \end{pmatrix} x_j, \quad (18)$$

where

$$d_i^+ = -\{(1+\rho_i)^{-3/2} + (1-\rho_i)^{-3/2}\}/4,$$
$$d_i^- = -\{(1+\rho_i)^{-3/2} - (1-\rho_i)^{-3/2}\}/4.$$

As a result, we obtain the terms in (11),

$$\xi_i^A = c_i^+ f_i + c_i^- g_i, \quad (19)$$

$$\psi_i^A = (d_i^+ u_i + d_i^- v_i) f_i + (d_i^- u_i + d_i^+ v_i) g_i. \quad (20)$$

Adaptive online algorithm

The algorithm derived in the previous sections depends on the cumulants $\beta_i^{k,l}$ and covariance matrix R, which are not known in advance when on-line learning. Those values can be estimated by the following adaptive algorithms:

$$\frac{d\beta_i^{k,l}}{dt} = -\mu(t)\{\beta_i^{k,l} - (r_i)^k (s_i)^l + \beta_0^{k,l}\}, \quad (21)$$

$$\frac{dR_{ij}}{dt} = -\mu(t)(R_{ij} - u_i v_j), \quad (22)$$

where $\mu(t)$ is a learning rate.

Simulation

We applied the algorithm for a pair of 3 mixed signals. The source signals $\boldsymbol{u}^*$ used are
(1) uniform random number on $[-1,1]$,
(2) $\sin(2\pi 800t + 6\cos(2\pi 60t))$,
(3) $\sin(2\pi 90t)$,
where t is sampled by 10kHz($\Delta t = 10^{-4}$). Signals $\boldsymbol{v}^*$ corresponding to $\boldsymbol{u}^*$ are taken by $\boldsymbol{v}^* = (|u_1^*|, |u_2^*|, |u_3^*|)^{\mathrm{T}}$. A part of $\boldsymbol{u}^*$ and $\boldsymbol{v}^*$ are shown in fig. 2. Those signals are mixed by random matrix whose elements are generated from the uniform distribution on $[-1,1]$. Mixed signals are shown in fig. 3.

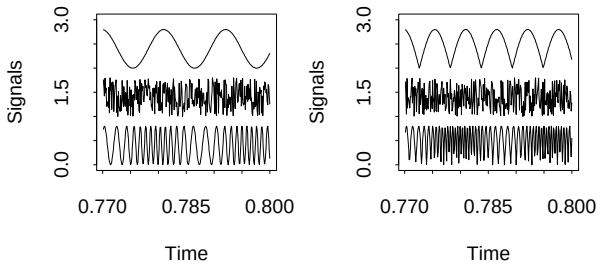

Figure 2: Original signals (left : u_i^*, right : v_i^*)

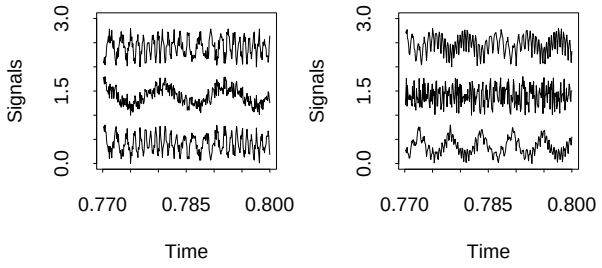

Figure 3: Mixed signals (left : x_i, right : y_i)

We compared two algorithms:
(1) Alg-ICA: running ICA for $\boldsymbol{x}$ and $\boldsymbol{y}$ independently,
(2) Alg-MICA: running MICA

First, we presphered the whole data by CCA, which is almost equivalent to the presphering by PCA in this case since the correlation coefficient between u_i^* and v_i^* is almost zero. Then, we ran each algorithm in their on-line form. The parameters are set as follows: time interval 0.8[secs], weight value $\gamma(t) = 0.2/(t/\Delta t + 1)$, learning constants $\eta(t) = 1000.0$, $\mu(t) = 50.0$. As time was going on we decreased $\gamma(t)$, because dependence relation is tend to be preserved, once it is acquired in the early stage of learning.

Recovered signals $\boldsymbol{u}$ and $\boldsymbol{v}$ for $t = [0.77, 0.8]$ are shown in fig. 4 for Alg-ICA and fig. 5 for Alg-MICA. The pairwise correspondence of signals is recovered correctly in fig. 5.

In literature of ICA, the following error index is often

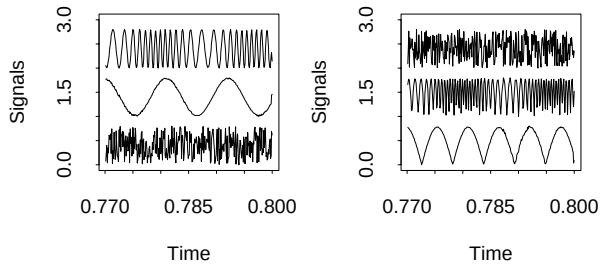

Figure 4: Recovered signals (Alg-ICA. left : u_i, right : v_i)

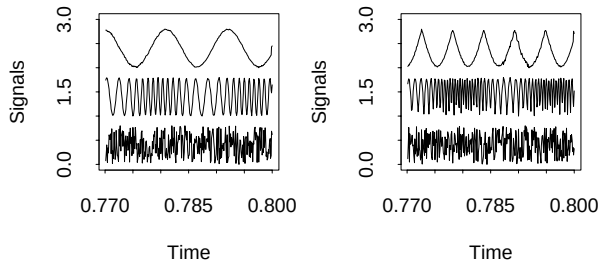

Figure 5: Recovered signals (Alg-MICA. left : u_i, right : v_i)

used to measure the performance of algorithms.

$$\mathcal{E}_{\mathrm{ICA}} = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{|P_{ij}|}{\max_k |P_{ik}|}-1\right)+ \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{|P_{ij}|}{\max_k |P_{kj}|}-1\right), \quad (23)$$

where $P = P^A = AV$ for $\boldsymbol{x}$ and $P = P^B = BW$ for $\boldsymbol{y}$, and $\mathcal{E}_{\mathrm{ICA}}$ is equal to zero when P is equivalent to identity matrix except for permutation and scaling.

In order to evaluate the performance of MICA, let us define the following error index,

$$\mathcal{E}_{\mathrm{MICA}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\left|\frac{{P^A}_{ij}}{\max_k |{P^A}_{ik}|}-\frac{{P^B}_{ij}}{\max_k |{P^B}_{ik}|}\right|, \quad (24)$$

which is equal to zero when P^A and P^B are identical except for scaling.
The independence index $\mathcal{E}_{\mathrm{ICA}}$ is shown in fig. 6, and $\mathcal{E}_{\mathrm{ICA}}$ of Alg-ICA tends to be rather smaller than in Alg-MICA, though they converges to almost zero. On the other hand, the dependence index $\mathcal{E}_{\mathrm{MICA}}$ is shown in fig. 7, where $\mathcal{E}_{\mathrm{MICA}}$ of Alg-ICA is much larger than Alg-MICA.

Concluding remarks

We have proposed a new learning algorithm to extract features from two sets of observations. It maximizes the degree of pairwise mutual information, and the features for each modality are statistically independent. It is straightforward to extend the framework to the case there are more than two sets of observations.

By numerical simulation, we confirmed the algorithm to extract features that are nonlinearly dependent. Another approach to find such a nonlinear relation between two sets of observations is nonlinear CCA, which extends CCA by allowing nonlinear mapping to get features. Asoh et al[2] has proposed the learning algorithm of nonlinear CCA for neural networks. Although that approach is effective in some cases, the features are not independent and it is not clear how the maximization of correlation coefficient is related to the maximization of mutual information without Gaussian assumption.

As stated in the beginning, ICA can be considered to be a model of lower systems in the brain (e.g. early vision), in which only one modality exists. On the other hand, Becker[3] has proposed the algorithm to maximize mutual information itself for discrete domain as a model of cortical self-organization, in which the system receives signals from several kinds of lower systems (e.g. auditory system and vision system). MICA integrates those algorithms in a sophisticated manner, and it is possible to construct a model of higher systems as an extension of lower levels.

References

[1] Anderson, T.W.: *An Introduction to Multivariate Statistical Analysis — Second edition*, John Wiley & Sons, 1984.

[2] Asoh, H., Takechi, O.: An approximation of Nonlinear Canonical Correlation Analysis by Multilayer Perceptrons, *Proc. of ICANN'94*, pp. 713–716, 1994.

[3] Becker, S.: Mutual Information Maximization: Models of Cortical Self-Organization, *Network: Computation in Neural Systems*, Vol. 7, No. 1, 1996.

[4] Bell, A.J., Sejnowski, T.J.: The 'independent components' of natural scenes are edge filters, *Vision Research*, Vol. 37, pp. 3327–3338, 1997.

[5] Jung, T.-P., Humphries, C., Lee, T.-W., Makeig, S., McKeown, M., Iragui, V., Sejnowski, T.J.: Extended ica removes artifacts from electroencephalographic record-

ings, *Advances in Neural Information Processing Systems 10*, 1998.

[6] Lee, T-W.: *Independent Component Analysis — Theory and Applications*, Kluwer, 1998.

[7] Yang, H.-H., Amari, S.: Adaptive Online Learning Algorithms for Blind Separation: Maximum Entropy and Minimum Mutual Information, *Neural Computation*, Vol. 9, pp. 1457–1482, 1997.

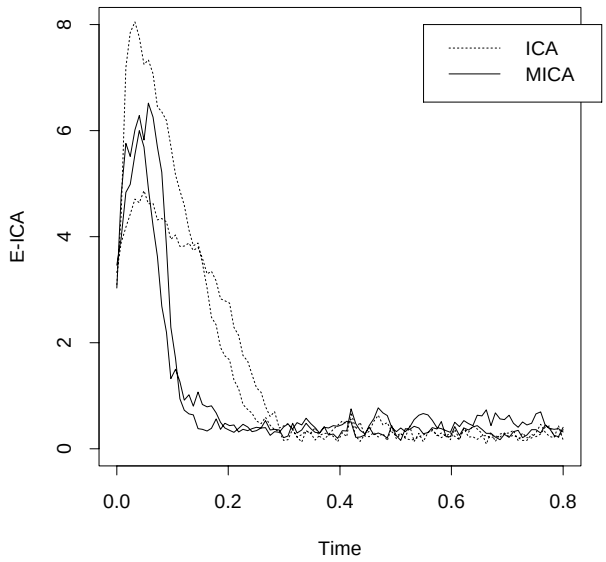

Figure 6: Error index on independence $\mathcal{E}_{\mathrm{ICA}}$.

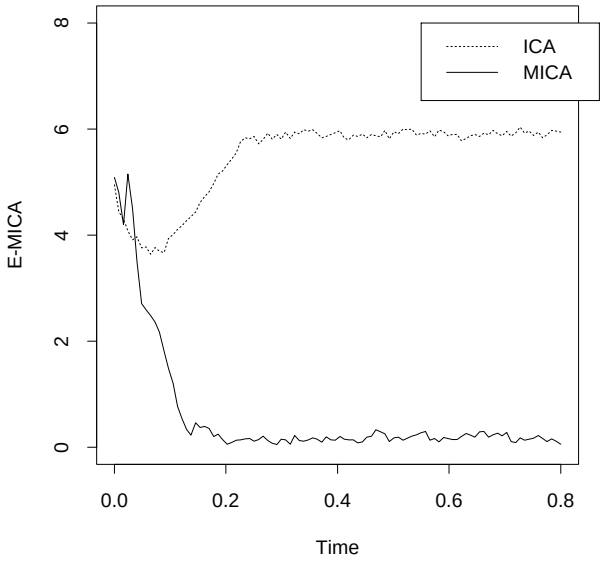

Figure 7: Error index on dependence $\mathcal{E}_{\mathrm{MICA}}$.