ELSEVIER

Neurocomputing 22 (1998) 81–111

NEUROCOMPUTING

Bayesian Kullback Ying–Yang dependence reduction theory

Lei Xu*

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong

Accepted 3 July 1998

Abstract

Bayesian Kullback Ying–Yang dependence reduction system and theory is presented. Via stochastic approximation, implementable algorithms and criteria are given for parameter learning and model selection, respectively. Three typical architectures are further studied on several special cases. The forward one is a general information theoretic dependence reduction model that maps an observation x into a representation y of k independent components, with k detectable by criteria. For the special cases of invertible map $x \rightarrow y$, a general adaptive algorithm is obtained, which not only is applicable to nonlinear or post-nonlinear mixtures, but also provides an adaptive EM algorithm that implements the previously proposed learned parametric mixture method for independent component analysis (ICA) on linear mixtures. The backward architecture provides a maximum likelihood independent factor model for modeling observations from unknown number of independent factors via a linear or nonlinear system under noisy situations. For the special cases of linear or post-nonlinear mixture under Gaussian noise, the simplified adaptive algorithm and the criterion for detecting k are given, with an approximately optimal linear mapping $x \rightarrow y$ suggested. Moreover, if the independent factors are assumed to be standard Gaussians, we are further led to the conventional factor analysis, but with a new adaptive algorithm for its estimation and a criterion for deciding the number of factors. The bi-directional architecture combines the advantages of backward and forward ones. A mean field approximation is presented, with a simplified adaptive parameter learning algorithm and an approximate k-selection criterion. Moreover, its special cases lead to the existing least mean square error reconstruction learning and the one hidden layer deterministic Helmholtz machine, with new findings. Also, a specific degenerate case of bi-directional architecture results in a non-invertible but adaptively implementable forward on to mapping for ICA. Experiments on binary sources are demonstrated with successes. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Dependence reduction; Bayesian Kullback Ying–Yang learning; Model selection; Independent component analysis; Stochastic approximation; Information theoretic; Maximum

* E-mail: lxu@cse.cuhk.edu.hk

PII S0925-2312(98)00051-4

likelihood; Independent factor model; Linear and post-nonlinear mixture; Blind separation; Source number detection; Least mean square error reconstruction; Deterministic Helmholtz machine

1. Introduction

The aim of mapping x into $y=[y^{(1)},\dots,y^{(k)}]^{\mathrm{T}}$ with the dependence among its components reduced as much as possible is a basic principle in a brain perception system [2]. It has been studied in literature under different names such as *dependence reduction, factorial learning, independent component analysis* (*ICA*), *factorial encoding*. Here, we adopt the first one because it actually covers the meanings of the other three and is more appropriate to include those efforts that attempt to get independence but finally may not really reach independence but a reduced dependence (e.g., falling into a local minimum in a cost minimization). We omit a large volume publications relating to these topics and refer to many other papers in this special issue.

The Bayesian Kullback Ying–Yang dependence reduction (BKYY-DR) theory is preliminarily proposed in Ref. [20] under the name of BKYY *factorial encoding* theory for the cases of y being a binary code. Its development to a general form is briefly summarized in Ref. [21, 22] under the current name, with particular focus on the backward architectures. In this paper, new advances will be reported. First, improvements are made on the architecture design of a BKYY-DR system. Second, a unified BKYY-DR framework has been set up, a generic stochastic implementing procedure and a generic model selection criterion are developed. Third, three typical architectures and their relations to several existing ICA or factorial learning methods have been further studied.

As a test bed, we also consider the instantaneous mixture models that the d dimensional observation x comes from k independent sources $y=[y^{(1)},\dots,y^{(k)}]^{\mathrm{T}}$ via

$$x=\begin{cases} g(y,\phi)+e, & \text{general,}\\ g(Ay,\phi)+e, & \text{post-nonlinear, } Ey=0,\ e \text{ is noise, } Ee=0,\ Eye^{\mathrm{T}}=0,\end{cases}$$

$$g(Ay)=[g_1(\hat{x}^1,\phi),\dots,g_d(\hat{x}^d,\phi)],\ Ay=\hat{x}=[\hat{x}^{(1)},\dots,\hat{x}^{(d)}]^{\mathrm{T}}. \qquad (1)$$

with x being wide stationary and ergodic. In the case of $e=0$, $d=k$ and $g(u,\phi)=u$ linear, it reduces to the linear invertible instantaneous mixture and $\hat{y}=Wx$ becomes independent in its components. This is usually called *independent component analysis* (*ICA*). In this case, this $\hat{y}=Wx$ can recover y up to constant unknown scales and any permutation of indices, and thus sources can be blindly separated from this linear invertible instantaneous mixture. This ICA has been studied widely in the literature. We omit to mention one by one all the existing references and refer to many other papers in this special issue. More generally, in the case of $e=0$, $d=k$ but $g(u,\phi)$ nonlinear, the problem of Eq. (1) becomes the post-nonlinear instantaneous mixture studied recently by Taleb and Jutten [14] and others.

Section 2 introduces the BKYY-DR theory. Via stochastic approximation, implementable algorithms and criteria are given for parameter learning and model selection in Section 3. In Section 4, three typical architectures are further studied on special cases. The first two are shown to lead to a generalized information theoretic DR model and a generalized maximum likelihood (ML) independent factor model, respectively, with the existing ICA methods and factor analyses models as special cases and with a number of new results. The bi-directional architecture is shown to combine the advantage of the first two and a mean field approximation is presented for fast implementation of its parameter learning and k-selection. It is also shown to include the existing LMSER learning[1] and the one hidden layer deterministic Helmholtz machine [5] as special cases with new findings. Moreover, a forward non-invertible onto mapping for ICA is got from a specific degenerate case of bi-directional architecture. Experimental results on binary sources are shown in Section 5. Then, we conclude in Section 6.

2. BKYY-DR system and theory

2.1. Basic idea of Bayesian Ying–Yang learning

The details of Bayesian Ying–Yang (BYY) learning system and theory and its applications can be found in Refs. [20,21]. Here, we only introduce its basic idea.

As shown in Fig. 1, the perception tasks can be summarized as the problem of estimating the joint distribution $p(x, y)$ of the observable pattern x in the observable space X and its representation pattern y in the representation space Y. In the Bayesian framework, we have two complementary representations $p(x, y) = p(y|x)p(x)$ and $p(x, y) = p(x|y)p(y)$. We use two sets of models $M_1 = \{M_{y|x}, M_x\}$ and $M_2 = \{M_{x|y}, M_y\}$ to implement each of the two representations:

$$p_{M_1}(x, y) = p_{M_{y|x}}(y|x)p_{M_x}(x), \quad p_{M_2}(x, y) = p_{M_{x|y}}(x|y)p_{M_y}(y). \tag{2}$$

We call M_x a Yang/(visible) model, which describes $p(x)$ in the visible domain X, and M_y a Ying[2]/(invisible) model which describes $p(y)$ in the invisible domain Y. Also, we call the passage $M_{y|x}$ for the flow $x \rightarrow y$ a *Yang*/(male) passage since it performs the task of transferring a pattern/(a real body) into a code/(a seed). We call a passage $M_{x|y}$ for the flow $y \rightarrow x$ a *Ying*/(female) passage since it performs the task of generating a pattern/(a real body) from a code/(a seed). Together, we have a YANG machine M_1 to implement $p_{M_1}(x, y)$ and a *YING* machine M_2 to implement $p_{M_2}(x, y)$. A pair of YING–YANG machines is called a YING–YANG pair or a Bayesian YING–YANG

[1] It is a unsupervised learning rule proposed with both the batch and adaptive gradient algorithms provided firstly in Refs. [15,16], and three years later it has been directly adopted to implement ICA by Karhunen and Joutsensalo [8] under the name of nonlinear PCA, see Section 4.3 for details.

[2] It should be "Yin" in the Mainland Chinese spelling system. However, I prefer to use "Ying" for the beauty of symmetry.

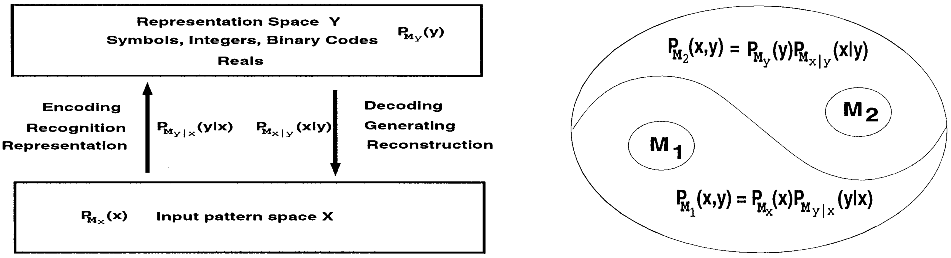


Fig. 1. The joint input-representation spaces X,Y and the Bayesian YING–YANG system.

system. Such a formalization compliments to a famous Chinese ancient philosophy that *every entity in the universe involves the interaction between YING and YANG.*

The task of specifying a Ying–Yang system consists of specifying all the aspects (e.g., the forms of variable and distributions, the architectures and scales, parameters, etc.) of the four components $p_{M_{y|x}}(y|x), p_{M_x}(x), p_{M_{x|y}}(x|y), p_{M_y}(y)$ under a given situation, which is called *learning* in a broad sense. Our learning theory consists of two key principles. **First**, the values of real parameters in a given system design are specified by maximally enhancing the matching or Harmony between the *Ying–Yang* pair, because both the Ying and Yang attempt to describe the same joint distribution. **Second**, those positive integers, that represent the scales or sizes of structures in the given system design, are selected for not only best enhancing the *Ying–Yang* matching but also keeping the entire Ying–Yang system in a minimum complexity. This theory is called Bayesian YING–YANG (BYY) learning theory. As shown in Ref. [20], the BYY theory provides a theoretical guide for a number of existing major learning models in *parameter learning, regularization, structural scale or complexity selection, architecture design and data smoothing.*

In implementation, the Ying–Yang harmony is made by minimizing a harmony measure called *separation functional.* Three categories of separation functionals have been suggested in Ref. [20]. A widely used one is the well-known Kullback divergence between the two representations in Eq. (2). In this special case, the BYY learning is called Bayesian–Kullback YING–YANG (BKYY) learning.

2.2. BKYY dependence reduction system and architecture design

This paper only concentrates on one special case of the BKYY learning that maps input x into a binary or real code $y = [y^{(1)}, \dots, y^{(k)}]^{\mathrm{T}}$ such that k is appropriately decided and the dependence among the components of $y^{(j)}$ are reduced as possible. This aim is imposed by letting

$$p_{M_y}(y) = \prod_{j=1}^{k} p_{M_y}(y^{(j)}). \tag{3}$$

In this case, the learning is realized by the minimization of the Kullback divergence:

$$KL_{M_1,M_2} = \int p_{M_{y|x}}(y|x)p_{M_x}(x)\ln\frac{p_{M_{y|x}}(y|x)p_{M_x}(x)}{p_{M_{x|y}}(x|y)\prod_{j=1}^{k}p_{M_y}(y^{(j)})}\,\mathrm{d}x\,\mathrm{d}y, \tag{4}$$

and thus called *BKYY dependence reduction* (*BKYY-DR*) *system and learning.*

In this learning, the specification of $p_{M_x}(x)$ is usually straightforward. Given a training set $D_x = \{x_i\}_{i=1}^{N}$, we simply let $p_{M_x}(x)$ fixed to a kernel estimate:

$$p_h(x) = \frac{1}{N}\sum_{i=1}^{N} K_h(x - x_i), \quad K_h(r) = \frac{1}{h^d}K\left(\frac{r}{h}\right), \tag{5}$$

with a prefixed kernel function $K(.)$ and a smoothing parameter h. A special case that we often consider is that $N \to \infty$, $h \to 0$.

The specifications of $p_{M_y}(y^{(j)})$ has two choices. One is to let $p_{M_y}(y^{(j)}) = p(y^{(j)})$ free.[3] Thus, min KL_{M_1,M_2} results in

$$p(y^{(j)}) = p_{M_1}(y^{(j)}), \quad p_{M_1}(y) = \int p_{M_{y|x}}(y|x)p_{M_x}(x)\,\mathrm{d}x. \tag{6}$$

The other choice is that $p_{M_y}(y)$ is an independent parametric model with each component density being either Bernoulli for a binary $y^{(j)}$ or a finite mixture for real $y^{(j)}$, that is

$$\begin{aligned} p_{M_y}(y) &= p(y|\theta_k) = \prod_{j=1}^{k} p(y^{(j)}|\xi_j), \quad \theta_k = \{\xi_j\}_{j=1}^{k}, \\ p(y^{(j)}|\xi_j) & \\ &= \begin{cases} p_j^{y^{(j)}}(1-p_j)^{1-y^{(j)}}, p_j = 1/(1+\mathrm{e}^{-\xi_j}) & \text{for binary } y^{(j)}, \\ \sum_{r=1}^{n_{y,j}}\alpha_{r,j}p(y^{(j)}|\xi_{r,j}),\ \alpha_{r,j} > 0,\ \sum_{r=1}^{n_{y,j}}\alpha_{r,j} = 1, \quad \xi_j = \{\xi_{r,j}, \alpha_{r,j}\}_1^{n_{y,j}} & \text{for real } y^{(j)}. \end{cases} \end{aligned} \tag{7}$$

The specifications of $p_{M_{x|y}}(x|y)$ and $p_{M_{y|x}}(y|x)$ are made jointly with three types of combinations, given as follows:

(a) *The Forward architecture*, in which $p_{M_{x|y}}(x|y) = p(x|y)$ is free to be indirectly specified through other components and $p_{M_{y|x}}(y|x) = p(y|x,\theta_{y|x})$ is a parametric model.

From min KL_{M_1,M_2}, the free $p_{M_{x|y}}(x|y) = p(x|y)$ is specified by

$$p_{M_{x|y}}(x|y) = p(x|y) = \frac{p(y|x,\theta_{y|x})p_{M_x}(x)}{p_{M_1}(y)}, \tag{8}$$

where $p_{M_1}(y)$ is given by Eq. (6). In this case, Eq. (4) becomes

$$KL_{M_1,M_2} = \int p_{M_{y|x}}(y|x)p_{M_x}(x)\ln\frac{p_{M_1}(y)}{\prod_{j=1}^{k} p_{M_y}(y^{(j)})}\,\mathrm{d}x\,\mathrm{d}y. \tag{9}$$

[3] A density is said *Free* implies that it is a totally unspecified in the density form $p(a)$ without any constraint. Thus, it is free to change to be indirectly specified by other components.

The parametric $p_{M_{y|x}}(y|x) = p(y|x, \theta_{y|x})$ is implemented by a network or a group of networks in forward architecture with $f(x, \psi)$ or $f_r(x, \psi_r)$ being its output.

When y is real, a typical example is the following Gaussian mixture:

$$p(y|x, \theta_{y|x}) = \sum_{r=1}^{n_{y|x}} \beta_r G(y, f_r(x, \psi_r), \Sigma_{y|x,r}), \quad \beta_r > 0, \sum_{r=1}^{n_{y|x}} \beta_r = 1,$$
$$\theta_{y|x} = \{\beta_r, \psi_r, \Sigma_{y|x,r}\}_1^{n_{y|x}}, \tag{10}$$

where $G(r, m, \Sigma)$ denotes a Gaussian density about r with mean m and covariance Σ.

When y is binary, we can consider the following specific design:

$$p_G(y|x, \theta_{y|x}) = Z_V^{-1} \mathrm{e}^{-(1/\beta)E(y,V)}, \quad Z_V = \int \mathrm{e}^{-(1/\beta)E(y,V)}\, \mathrm{d}y,$$
$$E(y,V) = \sum_{i,j \neq i}^{k} v_{ij} y^{(i)} y^{(j)} + \sum_{j=1}^{k} y^{(j)}[f_j(x, \psi) + c_j], \quad \theta_{y|x} = \{\psi, V, \{c_j\}\}, \tag{11}$$

which is a Gibbs distribution defined by $E(y,V)$ on a fully connected recurrent net with weights v_{ij} and external inputs $f_j(x, \psi), j = 1, \ldots, k$ and bias $c_j, j = 1, \ldots, k$, such that the second-order dependence among $y^{(j)}, j = 1, \ldots, k$ is taken into consideration. The simplest case of V is $v_{ij} = 0$ for all i, j. In general cases, V is required to satisfy conditions that Z_V exists.

For convenience, alternatively, we can also approximate $p_G(y|x, \theta_{y|x})$ by an independent distribution:

$$p(y|x, \theta_{y|x}) = \prod_{j=1}^{k} p_j(x)^{y^{(j)}} (1 - p_j(x))^{(1 - y^{(j)})}, \tag{12}$$

with $p_j(x), j = 1, \ldots, k$ decided such that the divergence $\int p(y|x, \theta_{y|x}) \ln[p(y|x, \theta_{y|x})/p_G(y|x, \theta_{y|x})]\, \mathrm{d}y$ is minimized.

As shown in Appendix A, we have $p_j(x), j = 1, \ldots, k$, are solution of

$$p_j(x) = s(\{p_r(x)\}_{r=1, r \neq j}^{k}) = \frac{1}{1 + \exp[\frac{1}{\beta}(\sum_{r=1, r \neq j}^{k} v_{jr} p_r(x) + f_j(x, \psi) + c_j)]},$$
$$j = 1, \ldots, k. \tag{13}$$

The dynamic process of getting the solution is simple, either sequentially or in parallel by

Choice (*a*): *for* $j = 1, \ldots, k$ *sequentially*, $p_j^{\text{new}}(x) = s(\{p_r^{\text{old}}(x)\}_{r=1, r \neq j}^{k})$ *and* $p_j^{\text{old}}(x) = p_j^{\text{new}}(x)$.

Choice (*b*): *for* $j = 1, \ldots, k$ *in parallel*, $p_j^{\text{new}}(x) = p_j^{\text{old}}(x) + \eta[s(\{p_r^{\text{old}}(x)\}_{r=1, r \neq j}^{k}) - p_j^{\text{old}}(x)]$ *and* $p_j^{\text{old}}(x) = p_j^{\text{new}}(x)$, *where* $\eta > 0$ *is a small enough learning stepsize.*

As shown in Appendix A, both the two processes converge to the solution of Eq. (13).

(b) *The Backward architecture*, also called *Generative Model*, which is complement to the forward architecture. In this case, $p(y|x)$ is free to be indirectly specified through other components, and $p_{M_{x|y}}(x|y) = p(x|y, \theta_{x|y})$ is parametric.

From min KL_{M_1,M_2}, the free $p_{M_{y|x}}(y|x) = p(y|x)$ is specified by

$$p_{M_{y|x}}(y|x) = p(y|x) = \frac{p(x|y,\theta_{x|y})\prod_{j=1}^{k} p_{M_y}(y^{(j)})}{p_{M_2}(x)},$$

$$p_{M_2}(x) = \int p(x|y,\theta_{x|y}) \prod_{j=1}^{k} p_{M_y}(y^{(j)})\,\mathrm{d}y. \tag{14}$$

The parametric $p_{M_{x|y}}(x|y) = p(x|y,\theta_{x|y})$ is implemented by a network or a group of networks in backward architecture with $g(x,\phi)$ or $g_r(x,\phi_r)$ being its output.

When x is real, similarly we have the following Gaussian mixture as a typical example:

$$p(x|y,\,\theta_{x|y}) = \sum_{r=1}^{n_{x|y}} \gamma_r G(x,\, g_r(y,\phi_r),\, \Sigma_{x|y,r}), \quad \gamma_r > 0,\ \sum_{r=1}^{n_{x|y}} \gamma_r = 1,$$

$$\theta_{x|y} = \{\gamma_r,\, \phi_r,\, \Sigma_{x|y,r}\}_1^{n_{x|y}}. \tag{15}$$

When $x = [x^{(1)}, \dots, x^{(d)}]^{\mathrm{T}}$ is binary, similar to Eq. (12) we have

$$p_G(x|y,\theta_{x|y}) = Z_U^{-1} \mathrm{e}^{-(1/\beta)E(x,U)}, \quad Z_U = \int \mathrm{e}^{-(1/\beta)E(x,U)}\,\mathrm{d}x,$$

$$E(x,U) = \sum_{i,j\neq i}^{d} u_{i,j} x^{(i)} x^{(j)} + \sum_{j=1}^{k} x^{(j)}[g_j(y,\,\phi) + d_j], \quad \theta_{x|y} = \{\phi,\, U,\, \{d_j\}\}. \tag{16}$$

The simplest case of U is $u_{ij} = 0$ for all i,j. In general cases, U is required to satisfy conditions that Z_U exists.

Alternatively, similar to Eq. (13) we have

$$p(x|y,\,\theta_{x|y}) = \prod_{j=1}^{k} q_j(y)^{x^{(j)}} (1 - q_j(y))^{(1 - x^{(j)})}, \tag{17}$$

with $q_j(y), j = 1, \dots, k$ decided such that the divergence $\int p(x|y,\theta_{x|y}) \ln [p(x|y,\,\theta_{x|y}) / p_G(x|y,\,\theta_{x|y})]\,\mathrm{d}x$ minimizes.

Similarly, such a set of $q_j(y), j = 1, \dots, k$ is the solution of

$$q_j(y) = s(\{q_r(y)\}_{r=1,r\neq j}^{k}) = \frac{1}{1 + \exp[\frac{1}{\beta}(\sum_{r=1,r\neq j}^{k} v_{jr} q_r(y) + g_j(y,\,\phi) + d_j)]},$$

$$j = 1, \dots, k. \tag{18}$$

and we can also use the previous two iteration choices for the dynamic process of getting the solution.

(c) *The Bi-directional architecture*, in which both $p_{M_{y|x}}(y|x) = p(y|x,\,\theta_{y|x})$ is given by Eqs. (10) and (12) and $p_{M_{x|y}}(x|y) = p(x|y,\,\theta_{x|y})$ by Eqs. (5) and (16).

2.3. BKYY dependence reduction theory

Putting the above-described four components $p_{M_{y|x}}(y|x)$, $p_{M_x}(x)$, $p_{M_{x|y}}(x|y)$, and $p_{M_y}(y)$ into Eq. (4), we can get a so called BKYY-DR theory, which consists of four parts, given as follows:

(1) *First*, with $k, \mathcal{N} = \{\{n_{y,j}\}_1^k, n_{y|x}, n_{x|y}\}$ fixed, we determine[4] $\Theta_k = \{\theta_k, \theta_{y|x}, \theta_{x|y}\}$ by

$$\Theta_k^* = \arg\min_{\Theta_k} KL(\Theta_k, \mathcal{N}),\ KL \text{ given by Eq. (4)}, \tag{19}$$

which is called *parameter learning* and can be implemented by an iterative *Alternative Minimization*:

Step 1: Fix $M_2 = M_2^{\rm old}$, get $M_1^{\rm new} = \arg\min_{M_1} KL_{M_1,M_2}$,

$$\text{Particularly, } p_{M_{y|x}}(y|x) = p(y|x) \text{ given by Eq. (14) for a backward architecture.} \tag{20}$$

Step 2: Fix $M_1 = M_1^{\rm old}$, get $M_2^{\rm new} = \arg\min_{M_2} KL_{M_1,M_2}$,
which guarantees to reduce KL_{M_1,M_2} until it converges to a local minimum at Θ_k^*.

(2) *Second*, with $\mathcal{N} = \{n_{x|y}, n_{y|x}, n_y\}$ fixed, we decide the dimension k by

$$k^* = \min_{k\in\mathcal{K}} k, \quad \mathcal{K} = \{ j : J_1(j) = \min_k J_1(k)\}, \quad J_1(k) = KL(\Theta_k^*, \mathcal{N}). \tag{21}$$

It has been shown in Ref. [21] that $J_1(k) > J_1(k^{\rm o})$ for $k < k^{\rm o}$ and $J_1^{\rm o}(k) = J_1^{\rm o}(k^{\rm o})$ for $k \geq k^{\rm o}$, $k^{\rm o}$ is the correct value for k. That is, among several k values with a same or similar *Ying–Yang* matching value, we select the smallest k – the simplest in complexity.

Alternatively, especially in the case of finite samples in D_x where $J_1(k)$ may still slowly decrease even after $k \geq k^{\rm o}$, as shown in Refs. [23,24] with $0 < \gamma_r$ we can also use

$$k^* = \min_k J_2(k), \quad J_2(k) = J_1(k) - \gamma_r H_{y|x}(k),$$
$$H_{y|x}(k) = \int_{x,y} p_{M^*_{y|x}}(y|x) p_h(x) \ln p_{M^*_{y|x}}(y|x)\, \mathrm{d}x\, \mathrm{d}y \tag{22}$$

for detecting $k^{\rm o}$ as the minimum point of $J_2(k)$, which reduces as k increases and reaches its minimum at $k^{\rm o}$ and then increases as k continues to increase, where the superscript "*" means that this component is obtained after inserting the value Θ_k^* obtained by Eq. (19). Here, we select the best k^* that minimizes both the mismatching of the Ying–Yang pair and its complexity $H_{y|x}(k)$ represented by the entropy of the Yang passage, with the trade-off controlled by γ_r. Usually, we can set $\gamma_r = 1$.

[4] Each of $\{n_{y,j}\}_1^k$,$n_{y|x}$,$n_{x|y}$, and also of θ_k,$\theta_{x|y}$,$\theta_{x|y}$ can be an empty set when its corresponding density is free to be determined by other components.

(3) *Third*, after learning by Eq. (19) and dimension selection by Eqs. (21) and (22), we do structural scale selection by

$$\mathcal{N}^* = \min_{\mathcal{N}} J(\mathcal{N}),\ J(\mathcal{N}) = \begin{cases} KL(\theta^*_{k^*}, \mathcal{N}) & \text{for } J_1, \\ KL(\theta^*_{k^*}, \mathcal{N}) - \gamma_r H_{y|x}(k^*) & \text{for } J_2. \end{cases} \tag{23}$$

(4) *Finally*, after learning, the mapping $x \to y$ can actually be realized via $p_{M_{y|x}}(y|x)$ in the three ways:

(a) Random sampling $\hat{y}$ by the resulted $p_{M_{y|x}}(y|x)$.

(b) Maximum posterior $\hat{y} = \arg\max_y p_{M_{y|x}}(y|x)$.

(c) Taking the regression $E(y|x)$ by

$$E(y|x) = \begin{cases} [p_1(x)\,, \dots, p_k(x)]^{\mathrm{T}} & \text{for binary } y, \\ \sum_{r=1}^{n_{y|x}} \beta_r f_r\ (x, \psi_r) & \text{for real } y. \end{cases} \tag{24}$$

3. Stochastic implementation

3.1. Approximating integral by summation

To tackle the difficulty or high expenses on dealing with the integral operations in KL given in Eq. (4), this paper also uses a variant of the general stochastic sampling implementation technique proposed in Ref. [24].

Given the current x_i, suppose that we can get a set of random samples $D_{y|x_i} = \{y_l\}_{l=1}^{N_i}$. Similar to Eq. (5), we consider the nonparametric estimate

$$p_{h'}(y|x_i) = \frac{1}{N_i} \sum_{l=1}^{N_i} K_{h'}(y - y_l). \tag{25}$$

Then, we adopt the following replacements:

$$\mathrm{d}x \approx \frac{p_{h=0}\ (x)}{p_h(x)} \mathrm{d}x, \quad \mathrm{d}y \approx \frac{p_{h'=0}\ (y|x_i)}{p_{h'}(y|x_i)} \mathrm{d}y \tag{26}$$

in the integral operations over x, y in Eqs. (4) and (22). After ignoring $\int p_h(x) \ln p_h(x)\, \mathrm{d}x$, we get

$$KL_{M_1,M_2} = \mathrm{Avg}\left[\frac{h_{y|x}(i,\, l) - c_x(i,\, l) - c_y(i,\, l)}{p_{h'}(y_l|x_i)}\right], \quad H_{y|x}(k) = \mathrm{Avg}\left[\frac{h_{y|x}(i,\, l)}{p_{h'}(y_l|x_i)}\right],$$

$$\mathrm{Avg}[a(i,\, l)] = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{N_i} \sum_{l=1}^{N_i} a(i,\, l), \quad h_{y|x}(i,\, l) = p_{M_{y|x}}(y_l|x_i) \ln p_{M_{y|x}}(y_l|x_i), \tag{27}$$

$$c_x(i,\, l) = p_{M_{y|x}}(y_l|x_i) \ln p_{M_{x|y}}(x_i|y_l), \quad c_y(i,\, l) = p_{M_{y|x}}(y_l|x_i) \sum_{j=1}^{k} \ln p_{M_y}(y_l^{(j)}).$$

Particularly, from Eq. (9) we get

$$KL_{M_1,M_2} = \mathrm{Avg}\left[\frac{l_y(i,\, l) - c_y(i,\, l)}{p_{h'}(y_l|x_i)}\right], \quad l_y(i,\, l) = p_{M_{y|x}}(y_l|x_i) \ln p_{M_1}(y_l),$$

$$p_{M_1}(y) = \frac{1}{N} \sum_{i=1}^{N} p_{M_{y|x}}(y_l|x_i). \tag{28}$$

In general, the set of random samples $D_{y|x_i} = \{y_l\}_{l=1}^{N_i}$ can be obtained in one of the following ways:

1. When y is binary, we can simply get each bit y_j independently by taking 0 or 1 either with equal probability or according to $p_j(x_i)$ given in Eqs. (12) and (13), and at the same time we set $p_{h'}(y_l|x_i) = 1$ or $p_{h'}(y_l|x_i) = p_{M_{y|x}}(y_l|x_i)$, respectively.
2. When y is real, we can have two choices:
   (a) With probability β_r, get $y_l = f_r(x, \psi_r) + \varepsilon$, with ε being either a Gaussian noise of a zero mean and a given variance matrix (which can be a simplification or approximation of $\Sigma_{y|x,r}$) or a uniform distributed noise on a finite domain (e.g., a sphere or hyper-cubic) centered at $f_r(x, \psi_r)$.
   (b) Get $y_l = \sum_{r=1}^{n_{y|x}} \beta_r f_r\ (x, \psi_r) + \varepsilon$, with ε being either a Gaussian noise of a zero mean and a given variance matrix or a uniform distributed noise on a finite domain (e.g., a sphere or hyper-cubic) centered at $\sum_{r=1}^{n_{y|x}} \beta_r f_r\ (x, \psi_r)$.

3.2. Parameter learning and model selection

By using KL_{M_1,M_2} in Eqs. (27) and (28) to replace KL_{M_1,M_2} in the parameter learning Equations (19) and (20), we get a general batch way stochastic implementation algorithm. If one of $\min_{M_1} KL_{M_1,M_2}$ and $\min_{M_2} KL_{M_1,M_2}$ is not analytically solvable, we can let each minimization replaced by a descent search, e.g, a step of movement along gradient descent direction.

Except for the forward architecture[5], we can also get an adaptive algorithm via descending the instantaneous cost $[h_{y|x}(i, l) - c_x(i, l) - c_y(i, l)]/p_{h'}(y_l|x_i)$ along the gradient direction.

As examples, we give two detailed algorithms below:

Batch algorithm for forward BKYY-DR

Step 1: *Fix* $M_y = M_y^{\mathrm{old}}$, *update* $M_{y|x}$ *to get* $M_{y|x}^{\mathrm{new}}$ *by*

$$\theta_{y|x}^{\mathrm{new}} = \theta_{y|x}^{\mathrm{old}} - \eta \mathrm{Avg}\left[\frac{1}{p_{h'}(y_l|x_i)} \left. \frac{\partial [l_y(i, l) - c_y(i, l)]}{\partial \theta_{y|x}} \right|_{\theta_{y|x}^{\mathrm{old}}} \right],$$

Step 2: *Fix* $M_{y|x} = M_{y|x}^{\mathrm{new}}$, *to get* M_y^{new} *by* $p_{M_y}(y_l^{(j)}) = (1/N)\sum_{i=1}^{N} p(y_l^{(j)}|x_i, \theta_{y|x})$ *when it is free*, *otherwise by*

$$\xi_j^{\mathrm{new}} = \xi_j^{\mathrm{old}} + \eta \frac{1}{p_{h'}(y_l|x_i)} \left. \frac{\partial c_y(i, l)}{\partial \xi_j} \right|_{\xi_j^{\mathrm{old}}}.$$

Adaptive algorithm for backward and bi-directional BKYY-DR

Loop: *For an input* x_i, *get a set of random samples* $D_{y|x_i} = \{y_l\}_{l=1}^{N_i}$, *as described in Section* 3.1.

[5] Where $p_{M_1}(y)$ must be computed by Eq. (28) in a batch way.

Step 1: *For each* y_l, *update* $p_{M_{y|x}}(y|x)$ *by*

(*a*) *either*

$$\theta_{y|x}^{\text{new}} = \theta_{y|x}^{\text{old}} - \eta \frac{1}{p_{h'}(y_l|x_i)} \left.\frac{\partial[h_{y|x}(i,l) - c_y(i,l)]}{\partial\theta_{y|x}}\right|_{\theta_{y|x}^{\text{old}}}$$

for bi-directional architecture;

(*b*) *or Eq.* (14) *for backward architecture, with* $p_{M_2}(x_i)$ *given by one of the three choices below*:

(i) $p_{M_2}(x_i) = \sum_y p(x_i|y, \theta_{x|y})\prod_{j=1}^{k} p_{M_y}(y^{(j)})$.

(ii) $p_{M_2}(x_i) \approx \dfrac{1}{N_i}\sum_{l=1}^{N_i} \dfrac{p(x_i|y_l, \theta_{x|y})\prod_{j=1}^{k} p_{M_y}(y_l^{(j)})}{p_{h'}(y_l|x_i)}$.

(iii) *Use a mean field approximation* $p_{M_2}(x_i) \approx p(x_i|E(y|x_i), \theta_{x|y})$ *with* $E(y|x_i)$ *given by* Eq. (24).

Step 2: (*c*) *Fix* $M_{y|x}$, *update*

$$\theta_{x|y}^{\text{new}} = \theta_{x|y}^{\text{old}} + \eta \frac{1}{p_{h'}(y_l|x_i)} \left.\frac{\partial c_x(i,l)}{\partial\theta_{x|y}}\right|_{\theta_{x|y}^{\text{old}}},$$

(*d*) *Then, to get* M_y^{new} *by*

(i) $\xi_j^{new} = \xi_j^{\text{old}} + \eta \dfrac{1}{p_{h'}(y_l|x_i)} \left.\dfrac{\partial c_y(i,l)}{\partial\xi_j}\right|_{\xi_j^{\text{old}}}$,

(ii) *Or otherwise by* $p(y) = p_{M_1}(y)$ *when* $p_{M_y}(y)$ *is free, where* $p_{M_1}(y)$ *is given by Eq.* (28).

The detailed equations for gradients used in the above algorithms are given in Appendix B.

After parameter learning, with the obtained parameters we do model selection by using KL_{M_1,M_2} and $H_{y|x}(k)$ in Eqs. (7) and (28) to replace KL_{M_1,M_2} in Eqs. (21)–(23) such that we get the stochastic approximations of the criteria $J_1(k)$ and $J_2(k)$ for selecting an appropriate k and $J(\mathcal{N})$ for selecting structural scale.

4. Three BKYY-DR architectures

Our attentions will focus on several special cases that not only lead to several existing models but also provide new insights and new models for solving the BSS problem Eq. (1).

4.1. General information theoretic DR model

4.1.1. General model

For the *forward architecture*, from Eq. (9) we have

$$KL_{M_1,M_2} = \int p_{M_1}(y) \ln \frac{p_{M_1}(y)}{\prod_{j=1}^{k} p_{M_y}(y^{(j)})} \mathrm{d}y. \tag{29}$$

That is, x is mapped into y via $p(y|x, \theta_{y|x})$ which results in $p_{M_1}(y)$ such that a best match is obtained between this $p_{M_1}(y)$ and the independent model $\prod_{j=1}^{k} p_{M_y}(y^{(j)})$.

This general model can be classified into two types:

1. For a free $p_{M_y}(y^{(j)}) = p(y^{(j)}) = p_{M_1}(y^{(j)})$ given by Eq. (6), Eq. (29) is an extension of the MMI ICA model [1] to the non-invertible mappings $x \to y$.
2. For a parametric $p_{M_y}(y^{(j)})$ given by Eq. (7), Eq. (29) is an extension of the information theoretic ICA model given in Ref. [25] to the non-invertible mappings $x \to y$.

For both the two types, by using Eq. (28) the parameter learning can be made by the batch algorithm given in Section 3 and the selection of k can be made by Eqs. (1)–(23).

4.1.2. Invertible forward model

In spite of its advantage of being applicable to non-invertible mappings $x \to y$, Eq. (29) can only be implemented in batch because $p_{M_1}(y)$ must be computed by Eq. (28) in batch. In the special cases of invertible mappings $x \to y$, when $p_{M_y}(y^{(j)}) = p(y^{(j)}|\xi_j)$ is parametric, we have $k = d$ and $p(y|x, \theta_{y|x}) = \delta(y - z)$, $z = f(x,\Psi) = [f_1(x,\Psi), \ldots, f_k(x,\Psi)]^{\mathrm{T}}$ with $f(x,\Psi)$ invertible. From Eqs. (6) and (5) we get

$$p_{M_1}(y) = \int \delta(z - y) p_h(f^{-1}(z)) \left| \frac{\partial f^{-1}(z)}{\partial x^{\mathrm{T}}} \right| \mathrm{d}z = p_h(f^{-1}(y)) \left| \frac{\partial f^{-1}(y)}{\partial x^{\mathrm{T}}} \right|. \tag{30}$$

Putting it into Eq. (29) and discarding irrelevant terms, $\min KL_{M_1,M_2}$ becomes $\max_{\Psi,\theta_k} J(\Psi, \theta_k), \theta_k = \{\xi_j\}$ with

$$J(\Psi, \theta_k) = \frac{1}{N} \sum_{i=1}^{N} \left[\ln \left| \frac{\partial f(x,\Psi)}{\partial x^{\mathrm{T}}} \right|_{x = x_i} + \sum_{j=1}^{k} \ln p(f_j(x_i,\Psi)|\xi_j) \right], \tag{31}$$

which can be implemented adaptively by

Adaptive algorithm for forward BKYY-DR.

Step 1: $\Psi^{\mathrm{new}} = \Psi^{\mathrm{old}} + \eta \delta\Psi|_{\Psi^{\mathrm{old}}}$, $\delta\Psi = \dfrac{\partial[\ln |\frac{\partial f\ (x,\Psi)}{\partial x^{\mathrm{T}}}| + \sum_{j=1}^{k} \ln p(f_j(x,\Psi)|\xi_j)]}{\partial \Psi}$,

Step 2: $y^{(j)} = f_j(x,\Psi^{\mathrm{new}})$, $\zeta_j^{\mathrm{new}} = \zeta_j^{\mathrm{old}} + \eta \frac{\partial\ \ln\ p(y^{(j)}|\xi_j)}{\partial \xi_j}$,

When $p(y^{(j)}|\xi_j) = \sum_{r=1}^{n_{y,j}} \alpha_{r,j} p(y^{(j)}|\xi_{r,j})$, it becomes

(a) $$h_{r,j}(y^{(j)}) = \frac{\alpha_{r,j} p(y^{(j)}|\xi_{r,j})}{p(y^{(j)}|\xi_j)}, \quad \alpha_{r,j} = (1 - \eta)\alpha_{r,j}^{\mathrm{old}} + \frac{\eta h_{r,j}(y^{(j)})}{\alpha_{r,j}} ,$$

(b) $$\zeta_{r,j}^{\mathrm{new}} = \zeta_{r,j}^{\mathrm{old}} + \eta h_{r,j}(y^{(j)}) \frac{\partial \ln p(y^{(j)}|\xi_{r,j})}{\partial \xi_{r,j}}.$$

For $p(y^{(j)}|\xi_{r,j}) = G(y^{(j)}, m_{r,j}, \sigma^2_{r,j})$, it becomes

$$m^{\text{new}}_{r,j} = (1-\eta)m^{\text{old}}_{r,j} + \eta y^{(j)}, \quad \sigma^{2\,\text{new}}_{r,j} = (1-\eta)\sigma^{2\,\text{old}}_{r,j} + \eta(y^{(j)} - m^{\text{new}}_{r,j})^2.$$

We further consider three specific cases:

(1) When $f(x,\Psi)$ is an invertible nonlinear function given by a forward network (e.g., a three-layer perceptron), the above algorithm provides a tool for de-mixing a nonlinear source mixture, and the gradient $\delta\Psi$ can be computed in a way similar to back propagation technique.

(2) When $f(x,\Psi) = Wg(x,\Phi)$ and $g(x,\Phi)$ is an invertible nonlinear function given by a forward network, the above algorithm provides a tool for de-mixing the so called post-nonlinear mixture studied in Ref. [14], by using $g(x,\Phi)$ to remove the nonlinear distortion first and then making the linear de-mixing by W.

(3) When $f(x,\Psi) = Wx$, Eq. (31) reduces into the popular invertible linear mixture, as will be further addressed in detail subsequently.

4.1.3. *Invertible linear mixture*

When $f(x,\Psi) = Wx$, Eq. (31) reduces to

$$J(W) = \ln|W| + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} \ln p(w_j^{\mathrm{T}} x_i|\xi_j), \tag{32}$$

with $W = [w_1, \ldots, w_k]^{\mathrm{T}}$. As shown in Ref. [26], several existing ICA models for de-mixing an invertible linear mixture will lead to a cost in the general form of Eq. (32), although they are different in the specific forms for $p(\cdot|\xi_j)$:

(i) When $p(\cdot|\xi_j)$ is a prefixed estimate of the marginal density of $y^{(j)}$, it leads to the MMI ICA [1].

(ii) When $p(\cdot|\xi_j) = \mathrm{d}s_j(r)/\mathrm{d}r$ with $s_j(r)$ being a prefixed sigmoid nonlinearity, it leads to the INFORMAX ICA model [10,3].

(iii) When $p(\cdot|\xi_j)$ is a general parametric model, it leads to the information theoretic ICA model [25].

(iv) When $p(\cdot|\xi_j)$ is a finite mixture given in Eq. (7), it leads to the learned parametric ICA model [26,27]. In this case, by using the following equations:

$$W^{\text{new}} = W^{\text{old}} + \eta(I + \phi(y)y^{\mathrm{T}})W,$$

$$\phi(y) = [\phi_1(y^{(1)}), \ldots, \phi_k(y^{(k)})]^{\mathrm{T}},$$

$$\phi_j(y^{(j)}) = \frac{\partial \ln p(y^{(j)}|\xi_j)}{\partial y^{(j)}} = \frac{\sum_{r=1}^{k_{y,j}} \alpha_{r,j}\, \partial p(y^{(j)}|\xi_{r,j})/\partial y^{(j)}}{\sum_{r=1}^{k_{y,j}} \alpha_{r,j}\, p(y^{(j)}|\xi^{\text{old}}_{r,j})}. \tag{33}$$

to replace Step 1 of the *adaptive algorithm for Forward BKYY-DR* previously given, we actually get an EM-like new adaptive algorithm for the learned parametric ICA model [26,27]. The updating on W in Eq. (33) is the so called natural gradient technique [1]. As shown first in Ref. [19] and also will be further discussed in Section 4.3, Eq. (33) is also directly applicable to the full row rank non-invertible $k \times d$ matrix W for the non-invertible linear mixture.

For those, existing Eq. (33) based approaches for ICA [12,10,3,1,7], either $p(y^{(j)})$ is a prefixed estimate of the marginal density of $y^{(j)}$[1], or $p(y^{(j)})$ is given by $\mathrm{d}s_j(r)/\mathrm{d}r$ with $s_j(r)$ being a prefixed sigmoid nonlinearity or equivalently $\phi_j(y^{(j)})$ is prefixed at a given nonlinearity [12,10,3,4]. These approaches work well only when all the sources are either sub-Gaussian only or super-Gaussian only. In contrast, the use of Eq. (33) to replace Step 1 of the previous *adaptive algorithm for Forward BKYY-DR* will adapt both W and $p(y^{(j)})$ or equivalently $\phi_j(y^{(j)})$ during learning. As a result, it works well on the sources that can be either sub-Gaussian or super-Gaussian as well as any combination of the both types. Actually, the idea of, using a mixture of logistic densities as a flexibly adjustable density $p(y^{(j)})$ function to loosely learn a source density, was firstly proposed in 1995 by the present author and implemented in a joint paper with his colleague under the name of entropy maximization [9]. In April 1996, the direct application of this idea to ICA is made first in Ref. [29] under the name of *mixture of accumulative distribution functions* or *learned parametric ICA model*, and then later formally published in Refs. [27,28]. Also, the use of a mixture of logistic densities for modeling the marginal density has been made in 1996 by Pearlmutter and Parra [11] under the name of maximum likelihood density estimation.

4.2. Maximum likelihood independent factor model

4.2.1. General multi-channel model

For the backward architecture, from Eqs. (13) and (14) and ignoring the irrelevant term, Eq. (4) will become

$$KL_{M_1,M_2} = -\int p_h(x)\ln p_{M_2}(x)\,\mathrm{d}x, \quad p_{M_2}(x) = \int p(x|y,\theta_{x|y})\prod_{j=1}^{k} p_{M_y}(y^{(j)})\,\mathrm{d}y. \tag{34}$$

Thus, $\min KL_{M_1,M_2}$ is equivalent to the ML modeling on the observation x from k independent factors $\prod_{j=1}^{k} p_{M_y}(y^{(j)})$ via the generative model $p(x|y,\theta_{x|y})$.

Instead of directly modeling by Eq. (34), it can be equivalently implemented via Eq. (27) such that the parameter learning is made by the adaptive algorithm given in Section 3 and the selection of k is made by Eqs. (21)–(23).

Both x and y can be binary or real, with $p_{M_y}(y^{(j)})$ given by Eq. (7) and $p(x|y,\theta_{x|y})$ by Eqs. (5) and (17). Particularly, when $p(x|y,\theta_{x|y})$ is given by Eq. (15), we regard that the observation x is generated from the independent factors $y^{(j)}, j = 1,\dots,k$ with probability γ_r via the channel $G(x, g_r(y,\phi_r)$ out of $n_{x|y}$ possible channels.

4.2.2. Single channel model under Gaussian noise

We further consider the BSS problem Eq. (1) with Gaussian noise $G(e, 0, \Sigma_{x|y})$. In this case, we have only a single channel $p(x|y,\ \theta_{x|y}) = G(x,\ g(y,\phi),\Sigma_{x|y})$ through which x is generated from the independent factors $y^{(j)}, j = 1\ ,\dots,k$. By inserting it into the adaptive algorithm in Section 3, the updating on $\theta_{x|y}$ will have the following detailed equations:

$$m_{x|y} = g(y, \phi^{\mathrm{old}}), \quad \gamma_{il} = \frac{p(y_l|x_i)}{p_{h'}(y_l|x_i)},$$

$$\phi^{\mathrm{new}} = \phi^{\mathrm{old}} + \eta\gamma_{il} \frac{\partial g^{\mathrm{T}}(y_l, \phi)}{\partial \phi}|_{\phi^{\mathrm{old}}} \Sigma_{x|y}^{\mathrm{old}\ -1}(x_i - m_{x|y}^{\mathrm{old}}),$$

$$\Sigma_{x|y}^{\mathrm{new}} = (1-\eta)\Sigma_{x|y}^{\mathrm{old}} + \eta\gamma_{il}(x_i - m_{x|y}^{\mathrm{new}})(x_i - m_{x|y}^{\mathrm{new}})^{\mathrm{T}}.$$

or

$$\begin{aligned}
&m_{x|y} = g(Ay,\phi),\ \phi^{\mathrm{new}} = \phi^{\mathrm{old}} + \eta\gamma_{il} \frac{\partial g^{\mathrm{T}}(A^{\mathrm{old}}y_l, \psi^{\mathrm{old}})}{\partial \phi^{\mathrm{old}}} \Sigma_{x|y}^{\mathrm{old}\ -1}(x_i - m_{x|y}^{\mathrm{old}}),\\
&A^{\mathrm{new}} = A^{\mathrm{old}} + \eta\gamma_{il}\delta A,\\
&\delta A = g'(A^{\mathrm{old}}y_l, \phi^{\mathrm{old}})\Sigma_{x|y}^{\mathrm{old}\ -1}(x_i - m_{x|y}^{\mathrm{old}})y_l^{\mathrm{T}}, \text{ or} \qquad (35)\\
&\delta A = g'(A^{\mathrm{old}}y_l, \psi^{\mathrm{old}})(x_i - m_{x|y}^{\mathrm{old}})y_l^{\mathrm{T}},
\end{aligned}$$

where $p_{M_{y|x}}(y_l|x_i) = p(y_l|x_i)$ is given by Eq. (14).

Particularly, for a linear mixture $g(Ay, \phi) = Ay$, we only need the two updating equations for $\Sigma_{x|y}$, A with simply $g'(Ay, \psi) = 1$.

Furthermore, when $y^{(j)}$ is binary, the updating on ξ_j is simply

$$\xi_j^{\mathrm{new}} = \xi_j^{\mathrm{old}} + \eta\gamma_{il}\left(\frac{y_l^{(j)}}{p_j} - \frac{1-y_l^{(j)}}{1-p_j}\right)\frac{\mathrm{e}^{-\xi_j}}{(1+\mathrm{e}^{-\xi_j})^2}. \qquad (36)$$

4.2.3. Approximate linear forward mapping

Though the mapping $x \to y$ can be realized by the three ways given at the end of Section 2, the computation of $p(y|x)$ is quite time consuming. Hence, we suggest to approximately use a linear mapping $\hat{y} = Wx$, with W decided such that $\min_W E\|y - \hat{y}\|^2$, resulting in

$$W^* = R_{xy}^{\mathrm{T}}R_x^{-1}, \quad R_x = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^{\mathrm{T}}, \quad R_{xy} = \mathrm{Avg}\left[\frac{x_i y_l^{\mathrm{T}}}{ph'(y_l|x_i)}\right]. \qquad (37)$$

Particularly, when $g(Ay, \phi) = Ay$, the learning of the inverse mapping $\hat{y} = Wx$ can be inserted into the algorithm Eq. (35) by adding in

$$W^{\mathrm{new}} = W^{\mathrm{old}} + \eta(y_l - W^{\mathrm{old}}x_i)x_i^{\mathrm{T}}. \qquad (38)$$

Moreover, with this linear mapping we can get directly the set of random samples $D_{y|x_i} = \{y_l\}_{l=1}^{N_i}$ by $y_l = W^{\mathrm{old}}x_i + \varepsilon$, where ε is a Gaussian noise of zero mean and a large enough variance.

4.2.4. Factor analysis as a special case

In the simplest case of the linear mixture with

$$p(x|y, \theta_{x|y}) = G(x, Ay, \Sigma_{x|y}), \quad p_{M_y}(y) = G(y, 0, I_k), \qquad (39)$$

the problem of min KL_{M_1,M_2} by Eq. (34) becomes the problem of finding A, $\Sigma_{x|y}$ as the solution of

$$S_x = \Sigma_{x|y} + AA^{\mathrm{T}}, \quad S_x = \int p_h(x)xx^{\mathrm{T}}\,\mathrm{d}x. \tag{40}$$

That is, we are led to the general form of the conventional factor analysis problem in the literature of statistics [13]. For the special case of $\Sigma_{x|y} = \sigma^2_{x|y}I_d$, it can be simply solved with A obtained from the first k principal components of S_x [23]. However, for the cases of either $\Sigma_{x|y} = \mathrm{diag}[\sigma^2_{x|y,1}, \dots, \sigma^2_{x|y,d}]$ or $\Sigma_{x|y}$ in its general form, the problem of solving Eq. (40) is not easy and usually made by some heuristics.

The new perspective of understanding factor analysis from BKYY-DR can bring us at least several new gifts.

From Eq. (14), we have that $p_{M_{y|x}}(y|x) = p(y|x)$ is specified by

$$p_{M_{y|x}}(y|x) = p(y|x) = \frac{G(x, Ay, \Sigma_{x|y})G(y, 0, I_k)}{\int G(x, Ay, \Sigma_{x|y})G(y, 0, I_k)\,\mathrm{d}y} = G(y, W^{\mathrm{T}}x, \Sigma_{y|x}),$$

$$W = (\Sigma_{x|y} + AA^{\mathrm{T}})^{-1}A, \quad \Sigma_{y|x} = I_k - W^{\mathrm{T}}(\Sigma_{x|y} + AA^{\mathrm{T}})W. \tag{41}$$

That is, the previous approximate linear forward mapping $y_l = W^{\mathrm{old}}x_i + \varepsilon$ is equivalent to the accurate optimal linear mapping by $p(y|x)$, with W given by Eq. (41) and ε being a Gaussian noise of zero mean and covariance $\Sigma_{x|y}$ by Eq. (41).

Therefore, from Eq. (35) we can get a simple adaptive algorithm for solving A, $\Sigma_{x|y}$ as follows:

Adaptive algorithm for factor analysis.

Step 1: For each x_i get $y_l = W^{\mathrm{old}}x_i + \varepsilon$ with ε being a Gaussian noise of zero mean and covariance $I_k - W^{\mathrm{old}\ \mathrm{T}}(\Sigma^{\mathrm{old}}_{x|y} + A^{\mathrm{old}}A^{\mathrm{old}\ \mathrm{T}})W^{\mathrm{old}}$;

Step 2: Update $A^{\mathrm{new}} = A^{\mathrm{old}} + \eta\gamma\delta A$, by either $\delta A = \Sigma^{\mathrm{old}\ -1}_{x|y}(x_i - A^{\mathrm{old}}y_l)y_l^T$ or $\delta A = (x_i - A^{\mathrm{old}}y_l)y_l^{\mathrm{T}}$. Then update $\Sigma^{\mathrm{new}}_{x|y} = (1-\eta)\Sigma^{\mathrm{old}}_{x|y} + \eta\gamma(x_i - A^{\mathrm{new}}y_l)(x_i - A^{\mathrm{new}}y_l)^{\mathrm{T}}$, $W^{\mathrm{new}} = (\Sigma^{\mathrm{new}}_{x|y} + A^{\mathrm{new}}A^{\mathrm{new}\ \mathrm{T}})^{-1}A^{\mathrm{new}}$.

Also, we can use $y_l = W^{\mathrm{old}}x_i$ to map x back to the independent factors. Moreover, as will be shown subsequently, we can have a simple criterion for deciding the number of factors. The two issues are usually not considered in the conventional factor analysis.

4.2.5. *Number of independent factors*

After parameter learning, generally the selection of k is made by using KL_{M_1,M_2} and $H_{y|x}(k)$ in Eq. (27) to replace KL_{M_1,M_2} in Eqs. (21) and (22).

For the case of single channel model with $p(x|y, \theta_{x|y}) = G(x, g(y, \phi), \Sigma_{x|y})$, from Eq. (27) we have $-\mathrm{Avg}[\frac{c_x(i,l)}{p_{h'}(y_l|x_i)}] \approx 0.5\ln|\Sigma_{x|y}| + \mathrm{const}$ and thus we can simply use

$$KL_{M_1,M_2} = 0.5\ln|\Sigma_{x|y}| + \mathrm{Avg}\left[\frac{h_{y|x}(i,l) - c_y(i,l)}{p_{h'}(y_l|x_i)}\right] \tag{42}$$

in Eqs. (21) and (22) for selecting k.

Particularly, in the case Eq. (39), from Eq. (41) and after ignoring irrelevant terms, we can even get

$$2KL_{M_1,M_2} = ln|\Sigma_x| + \mathrm{Tr}[\Sigma_x^{-1}S_x],\ \Sigma_x = \Sigma_{x|y} + AA^{\mathrm{T}}, S_x = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^{\mathrm{T}},$$

$$-2H_{y|x}(k) = \ln|\Sigma_{y|x}| + \mathrm{Tr}[\Sigma_{y|x}^{-1}\Sigma_{y|x}] = \ln|I_k - W^{\mathrm{T}}(\Sigma_{x|y} + AA^{\mathrm{T}})W| + k. \quad (43)$$

Thus, we can directly put them into Eqs. (21) and (22) for selecting k, without using stochastic sampling.

4.2.6. No noise special case.

Finally, it should be noticed that all the above discussions apply to the no noise case $e = 0$. Actually, the situation is even simpler. When k and other parameters are not correct, $\Sigma_{x|y}$ will not be zero during the learning, thus similar to the cases with noise. When k is equal or larger than its correct value, $\Sigma_{x|y}$ will become singular as other parameters converge to the correct values. Hence, this fact can actually be used as a signal for indicating the correct convergence. That is, *we can start the learning at a small value for k and then gradually increase it. We can regard that the learning is completed once* $\Sigma_{x|y}$ becomes singular.

Also when $e = 0$, $g(Ay, \phi) = Ay$, $A^{-1} = W$ and $p(\cdot|\xi_j)$ is an estimate of source density, Eq. (34) will again lead to Eq. (32), which is studied by Gaeta and Lacounme; Pham et al.[6,12] under the name of ML ICA model. The above discussions actually provides a new algorithm for solving this special case.

4.3. Combined features, mean field technique, and two related methods

4.3.1. Combined features

The advantage of the backward BKYY-DR is that the noise in x is considered and thus its effect is also reduced during the mapping $x \to y$ by $p(y|x)$, according to either of the three ways given at the end of Section 2, or approximately by the direct inverse mapping $\hat{y} = Wx$ learned via Eq. (38). The forward BKYY-DR has no consideration on noise. However, it implements the mapping $x \to y$ directly by a parametric model, which is easy and fast. In contrast, the backward BKYY-DR must compute $p(y|x)$ every time with expensive computing cost, except for the case of using the approximate linear mapping $\hat{y} = Wx$ learned via Eq. (38) or for the linear factor analysis case Eq. (39). The bi-directional architecture combines the advantages of both the backward and forward ones. Its parameter learning and model selection on k, $\mathcal{N}$ keep the same as discussed in Section 3.

4.3.2. Mean field approximation

We consider the special case that y is binary, $p_{M_{y|x}}(y|x) = p(y|x, \theta_{y|x})$ is given by Eq. (12) and $p_{M_y}(y)$ is given by Eq. (7). In this case, from Eq. (12) we use

$E(y|x) = [p_1(x), \dots, p_k(x)]^{\mathrm{T}}$ to replace y in $p(x|y, \theta_{x|y})$, resulting in a mean field approximation:

$$KL_{M_{1,2}} = \frac{1}{N}\sum_{i=1}^{N}(J_i^f + J_i^b),$$

$$J_i^b = -\ln p(x_i|E(y|x), \theta_{x|y}),$$

$$J_i^f = \sum_{j=1}^{k}\left[p_j(x_i)\ln\frac{p_j(x_i)}{p_j} + (1-p_j(x_i))\ln\frac{1-p_j(x_i)}{1-p_j}\right], \tag{44}$$

from which we see its minimization with respect to $M_{x|y}$ is only related to J_i^b, and its minimization with respect to M_y results in $p_j = 1/N\sum_{i=1}^{N} p_j(x_i)$. Though its minimization with respect to $M_{y|x}$ involves $J_i^f + J_i^b$ as a whole, it can also be made adaptively by gradient descent search. Thus, the parameter learning can be implemented adaptively by

Adaptive algorithm for bi-directional BKYY-DR in mean field approximation.
Step 1: *For an input*, *let* x_i,

$$\theta_{x|y}^{\mathrm{new}} = \theta_{x|y}^{\mathrm{old}} + \eta\frac{\partial \ln p(x_i|E(y|x),\theta_{x|y})}{\partial\theta_{x|y}}\Big|_{\theta^{\mathrm{old}}_{x|y}},\; p_j^{\mathrm{new}} = (1-\eta)p_j^{\mathrm{old}} + \eta p_j^{\mathrm{old}}(x_i).$$

Step 2: *Update* $v_{ij}^{\mathrm{new}} = v_{ij}^{\mathrm{old}} - \eta\delta v_{ij}$, $c_j^{\mathrm{new}} = c_j^{\mathrm{old}} - \eta\delta c_j$, $\beta^{\mathrm{new}} = \beta^{\mathrm{old}} - \eta\delta\beta$, $\psi^{\mathrm{new}} = \psi^{\mathrm{old}} - \eta\delta\psi$, where δv_{ij}, δc_j, $\delta\beta$, $\delta\psi$ are gradient directions of $J_i^f + J_i^b$. In general cases that $V \neq 0$, the learning of v_{ij} should be constrained by the conditions that Z_V in Eq. (11) exists.

To observe the details of this algorithm, we further consider the special case of

$$p(x|y, \theta_{x|y}) = G(x, Ay, \Sigma_{x|y}), \quad \text{and } v_{ij} = 0, \quad i,j = 1, \dots, k,$$

$$f_j(x, \psi) = \psi_j^{\mathrm{T}}x, \quad \psi = [\psi_1, \dots, \psi_k], \tag{45}$$

that is, the backward architecture describes a linear mixture BSS problem Eq. (1) and there is no direct interaction between $y^{(j)}, j = 1, \dots, k$ in the forward architecture.

In this case, Eq. (13) will become

$$p_j(x) = s(\psi_j^{\mathrm{T}}x + c_j), \quad s(r) = 1/(1 + \mathrm{e}^{\frac{r}{\beta}}), \quad j = 1, \dots, k. \tag{46}$$

After ignoring irrelevant terms, we have

$$2\sum_{i=1}^{N} J_i^b = N\ln|\Sigma_{x|y}| + E^2,$$

$$E^2 = \sum_{i=1}^{N}(x_i - AS(\psi^{\mathrm{T}}x + c))^{\mathrm{T}}\Sigma_{x|y}^{-1}(x_i - AS(\psi^{\mathrm{T}}x + c)),$$

$$S(\psi^{\mathrm{T}}x + c) = [s(\psi_1^{\mathrm{T}}x + c_1), \dots, s(\psi_k^{\mathrm{T}}x + c_k)]^{\mathrm{T}}, \quad c = [c_1, \dots, c_k]^{\mathrm{T}}, \tag{47}$$

from which we have the following detailed updating equations for $\theta_{x|y}$ in Step 1 of the above algorithm:

$$\begin{aligned} &\textit{Either } A^{\text{new}} = A^{\text{old}} + \eta \Sigma_{x|y}^{\text{old}\ -1} \delta A, \textit{ or } A^{\text{new}} = A^{\text{old}} + \eta \delta A, \\ &\delta A = [x_i - A^{\text{old}} S(\psi^{\text{old}\ \mathrm{T}} x_i + c)] S(\psi^{\text{old}\ \mathrm{T}} x_i + c)^{\mathrm{T}}, \qquad (48) \\ &\Sigma_{x|y}^{\text{new}} = (1-\eta)\Sigma_{x|y}^{\text{old}} + \eta [x_i - A^{\text{new}} S(\psi^{\text{old}\ \mathrm{T}} x_i + c)][x_i - A^{\text{new}} S(\psi^{\text{old}\ \mathrm{T}} x_i + c)]^{\mathrm{T}}. \end{aligned}$$

Moreover, from Eqs. (44), (46) and (47), we also have

$$\begin{aligned} &\mathrm{d}p(x_i) = \left[\left(\ln \frac{p_1(x_i)}{p_1} - \ln \frac{1-p_1(x_i)}{1-p_1}\right), \ldots, \left(\ln \frac{p_k(x_i)}{p_k} - \ln \frac{1-p_k(x_i)}{1-p_k}\right)\right]^{\mathrm{T}}, \\ &s'(r) = -\mathrm{e}^{\frac{r}{\beta}}/\beta(1+\mathrm{e}^{\frac{r}{\beta}})^2 = \frac{s^2(r) - s(r)}{\beta}, \\ &S'(\psi^{\mathrm{T}} x_i + c) = \mathrm{diag}[s'(\psi_1^{\mathrm{T}} x_i + c_1), \ldots, s'(\psi_k^{\mathrm{T}} x_i + c_k)], \\ &\mathrm{d}E(x_i) = -(x_i - AS(\psi^{\mathrm{T}} x_i + c))^{\mathrm{T}} \Sigma_{x|y}^{-1} A, \qquad (49) \\ &\delta c = \frac{\partial (J_i^f + J_i^b)}{\partial c} = S'(\psi^{\mathrm{T}} x_i + c)[\mathrm{d}p(x_i) + \mathrm{d}E^{\mathrm{T}}(x_i)], \\ &\delta \beta = \frac{\partial (J_i^f + J_i^b)}{\partial \beta} = -\frac{\psi^{\mathrm{T}} x_i + c}{\beta} \mathrm{d}E(x_i) S'(\psi^{\mathrm{T}} x_i + c)[1,1,\ldots,1]^{\mathrm{T}}, \\ &\delta \psi = \frac{\partial (J_i^f + J_i^b)}{\partial \psi} = x_i [\mathrm{d}p^{\mathrm{T}}(x_i) + \mathrm{d}E_j(x_i)] S'(\psi^{\mathrm{T}} x_i + c). \end{aligned}$$

After parameter learning, by inserting $KL_{M_{1,2}}$ of Eq. (44) with J_i^f of Eq. (44) and J_i^b of either Eq. (44) or Eq. (47) and inserting the following:

$$H_{y|x}(k) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} [p_j(x_i) \ln p_j(x_i) + (1 - p_j(x_i)) \ln (1 - p_j(x_i))], \qquad (50)$$

into Eqs. (21) and (22) we get the approximate criteria $J_1(k)$ and $J_2(k)$ for selecting k, where $p_j(x_i)$ is given by either Eq. (13) in a general case or Eq. (46) in the special case discussed above.

4.3.3. Least-mean square error reconstruction

In Eq. (47), if we further assume that $A = \psi$, $c_j = 0$ and $\Sigma_{x|y} = \sigma_{x|y}^2 I_d$, then $\sigma_{x|y}^2 E^2 = \sum_{i=1}^{N} \| x_i - AS(A^{\mathrm{T}} x_i) \|^2$ is exactly the cost function for the nonlinear one layer special case of the *Least mean square error reconstruction* (*LMSER*) learning first proposed in Refs. [15,16], in which both batch and adaptive gradient algorithms are given, and also in which it was firstly discovered that the sigmoid non-linearity can automatically break the symmetry of the components in the subspace, although it was not tested directly on the ICA problem. Three years later, the LMSER learning and its adaptive algorithm given in Refs. [15,16] has been directly adopted to implement ICA by the authors of Ref. [8] under the name of Nonlinear PCA.

Thus, we see that the LMSER learning is actually a rough approximation to a special case of the bi-directional BKYY-DR learning in a mean field approximation, after ignoring $\sum_{i=1}^{N} J_i^f$ and forcing $A = \psi$, $c_j = 0$. This connection not only provides an interpretation on why the LMSER learning works well for those BSS problems in Ref. [8], but also tells that the above bi-directional BKYY-DR learning in a mean field approximation may bring even better performance for ICA.

First, the minimization of $\sum_{i=1}^{N} J_i^f$ alone functions like the forward architecture studied in Section 4.1, and the minimization of $\sum_{i=1}^{N} J_i^b$ alone functions like the backward architecture studied in Section 4.2. The minimization of $\sum_{i=1}^{N}(J_i^f + J_i^b)$ combines the advantages of both. Second, we are no longer limited to the constraints $A = \psi$, $c_j = 0$ and $\Sigma_{x|y} = \sigma_{x|y}^2 I_d$ made in the LMSER learning. Third, we can decide the dimension k as discussed above, which cannot be made in the LMSER learning too. Moreover, we can also consider various general cases with either or both of $f(x, \psi), g(y, A)$ being not linear, which provides some general guides for understanding the previous proposed nonlinear extensions on the LMSER and PCA learning [17,18].

4.3.4. Helmholtz machine

We still consider that case with Eqs. (45) and (46), but we let $p(x|y, \theta_{x|y})$ to be replaced by Eq. (16) with

$$u_{ij} = 0,\ d_j = 0, j = 1, \dots d,\ g^{(j)}(y, \phi) = \phi_j^{\mathrm{T}} y. \tag{51}$$

Thus, after we add in J_i^b of Eq. (44) by a term $\sum_{j=1}^{d}[x_i^{(j)} \ln x_i^{(j)} + (1 - x_i^{(j)}) \ln (1 - x_i^{(j)})]$, which is irrelevant to the parameters to be learned, we get

$$J_i^b = \sum_{j=1}^{d_x} \left[x_i^{(j)} \ln \frac{x_i^{(j)}}{s(\phi_j^{\mathrm{T}} S(\psi^{\mathrm{T}} x_i))} + (1 - x_i^{(j)}) \ln \frac{1 - x_i^{(j)}}{1 - s(\phi_j^{\mathrm{T}} S(\psi^{\mathrm{T}} x_i))} \right]. \tag{52}$$

If we further let $x_i^{(j)}$ be approximated by its mean $Ex_i^{(j)}$, we find that in this special case of $KL_{M_{1,2}}$ of Eq. (44) becomes exactly the same as the $-\mathscr{F}(\theta, \phi)$ given by Eq. (3.11) in Ref. [5] for the deterministic Helmholtz machine under the special case of one hidden layer. This connection provides not only a new understanding on the deterministic Helmholtz machine, but also a guideline for extending it into various generalized cases with $u_{ij} \neq 0$, $d_j \neq 0$, $g^{(j)}(y, \phi) \neq \phi_j^{\mathrm{T}} y$.

4.3.5. Non-invertible linear mixture

Finally, we consider a specific degenerate case of bi-directional architecture with

$$p(x|y, \theta_{x|y}) = G(x, Ay, \sigma_x^2 I_d),\ p(y|x, \theta_{y|x}) = G(y, Wx, \sigma_y^2 I_k),\ A = W^{\mathrm{T}}(WW^{\mathrm{T}})^{-1}. \tag{53}$$

Thus, we have $WA = I$, $x - Ay = e_x$ is a Gaussian of zero mean and covariance $\sigma_x^2 I_d$, $y - Wx = y - W(Ay + e_x) = -We_x$ is a Gaussian of zero mean and covariance $\sigma_y^2 I_k = WW^{\mathrm{T}}\sigma_x^2 I_k$.

Moreover, we consider the parametric $p_{M_y}(y) = p(y|\theta_k)$ given by Eq. (7) and $p_{M_x}(x) = p_h(x)$ by Eq. (5). When $\sigma_x^2 \to 0$, $p(x|y, \theta_{x|y}) \to \delta(x - Ay)$ and $p(y|x, \theta_{y|x}) \to \delta(y - Wx)$. As shown in Appendix C, from Eq. (4) and after ignoring irrelevant terms, we have that $\min_W KL_{M_1,M_2}$ with $h = 0$ is equivalent to the maximization of the following cost function:

$$J(W) = 0.5\ln|WW^{\mathrm{T}}| + \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} \ln p(w_j^{\mathrm{T}} x_i|\xi_j)\,\mathrm{d}x, \quad W = [w_1, \ldots, w_k], \tag{54}$$

which is different from $J(W)$ in Eq. (32) only in that $\ln|W|$ is replaced by $0.5\ln|WW^{\mathrm{T}}|$. When W is invertible, the two become the same. In other words, Eq. (54) is an extension of Eq. (32) that is applicable to the full row rank non-invertible W.

The maximization of $J(W)$ in Eq. (54) can be implemented by either the gradient descent or the natural gradient descent updating

$$W^{\mathrm{new}} = W^{\mathrm{old}} + \eta\delta W,$$

$$\delta W = \begin{cases} (WW^{\mathrm{T}})^{-1}W + \phi(y)x^{\mathrm{T}} & \text{gradient,} \\ (I + \phi(y)y^{\mathrm{T}})W & \text{natural gradient,} \end{cases} \tag{55}$$

where $\phi(y)$ is the same as in Eq. (33). Interestingly, the equation by the natural gradient actually remains unchanged as that in Eq. (33). Therefore, all the discussions made on $J(W)$ of Eq. (32) apply to the cases of the full row rank non-invertible W in the same way as invertible W, without any particular care.

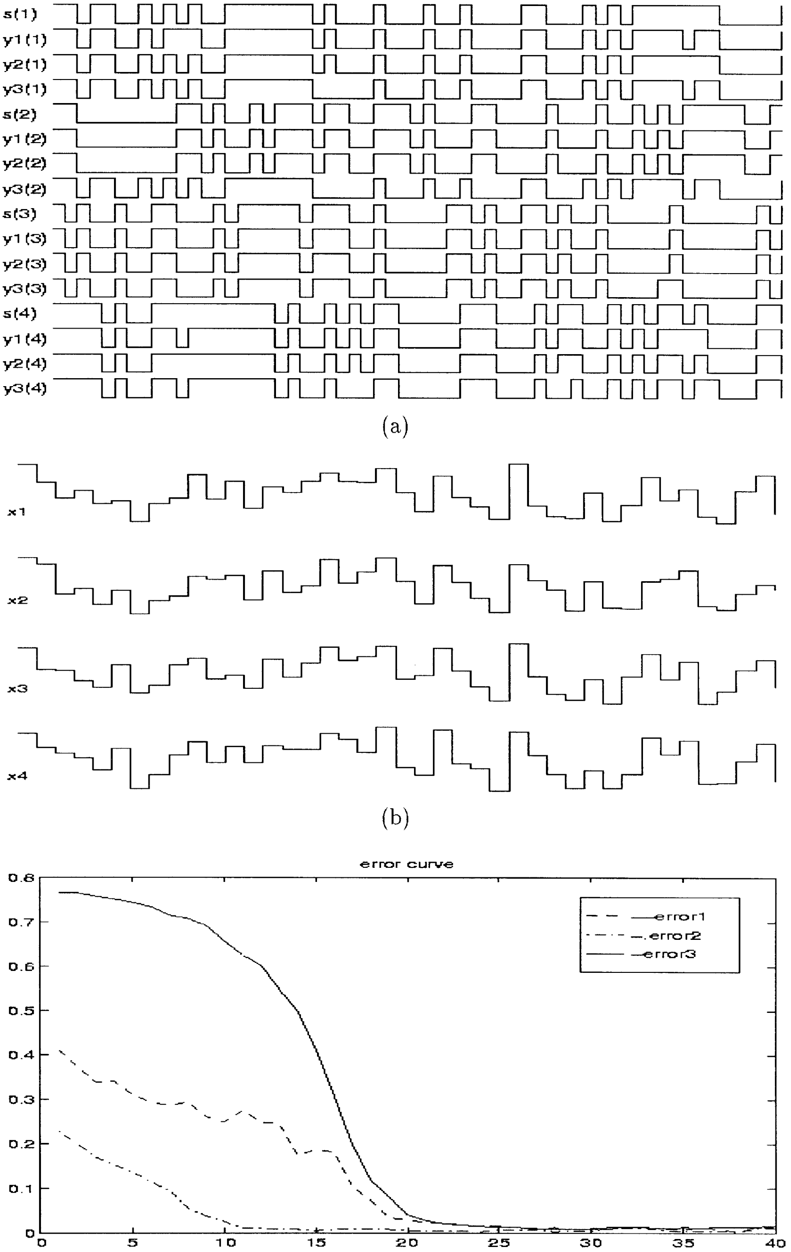
It should also be noted that $-J(W)$ in Eq. (54) is exactly Eq. (19) first given in Section 5 of Ref. [19], and the gradient equation in Eq. (55) is exactly Eq. (20) also first given in Section 5 of Ref. [19].

5. Experiments

Due to space limit, we only give some examples of using the backward BKYY-DR on the BSS problem Eq. (1) with $e \neq 0$ and $g(Ay, \phi) = Ay$ from k^* independent source channels of 0-1 binary Bernoulli signals, randomly sampled according to the ideal probablity $q_j = 0.5$ with a length $N = 200$ for each channel. Experimental results by using the forward BKYY-DR and bi-directional BKYY-DR are given in a separate paper elsewhere.

Here, we use the adaptive algorithm given in Section 4.2, with $p_r(y|x) = p(y|x)$ and thus $p(y_l|x_i)/ph'(y_l|x_i) = 1$ in all the relevant equations.

Shown in Fig. 2 are the results with the number $d = 4$ of sensors and the number $k^* = 4$ of sources. Given in Fig. 2a, $s(1), \ldots, s(k^*)$ are the k^* sources which are mixed by A given in Fig. 3 with a Gaussian noise of covariance matrix $0.02I$ added to form the observations $x(1), x(2), \ldots, x(d)$ by the sensors, as shown in Fig. 2b. It should be noted the noise is already quite large because the magnitudes of signals are finite but

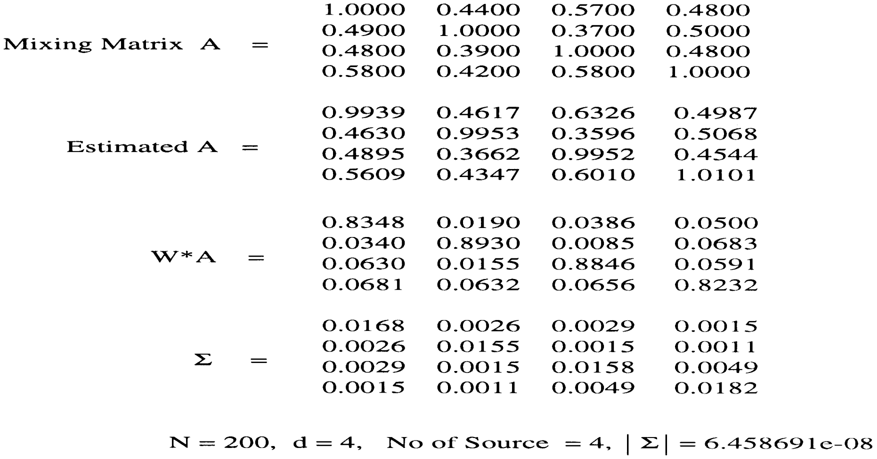


(a)

(b)

(c)

Fig. 2. Results on noisy linear mixture with 4-4-4 for k^*-d-k, where k^* is the correct number of sources, d is the number of sensors, and k is the guessed number of sources.



```
                      1.0000   0.4400   0.5700   0.4800
Mixing Matrix  A  =   0.4900   1.0000   0.3700   0.5000
                      0.4800   0.3900   1.0000   0.4800
                      0.5800   0.4200   0.5800   1.0000

                      0.9939   0.4617   0.6326    0.4987
      Estimated A  =  0.4630   0.9953   0.3596    0.5068
                      0.4895   0.3662   0.9952    0.4544
                      0.5609   0.4347   0.6010    1.0101

                      0.8348   0.0190   0.0386    0.0500
              W*A  =  0.0340   0.8930   0.0085    0.0683
                      0.0630   0.0155   0.8846    0.0591
                      0.0681   0.0632   0.0656    0.8232

                      0.0168   0.0026   0.0029    0.0015
                Σ  =  0.0026   0.0155   0.0015    0.0011
                      0.0029   0.0015   0.0158    0.0049
                      0.0015   0.0011   0.0049    0.0182

      N = 200,  d = 4,   No of Source  = 4,  | Σ | = 6.458691e-08
```

Fig. 3. The detailed data obtained by the adaptive backward BKYY-DR algorithm on the noisy linear mixture with 4-4-4 for k*-d-k.

the noise can be very large although in small probablity. In Fig. 2c, the error curves show how the normalized Hamming distance errors between the recovered signals and the original sources reduce as the learning goes with epoch t. In one epoch, each of 200 samples comes in sequentially.

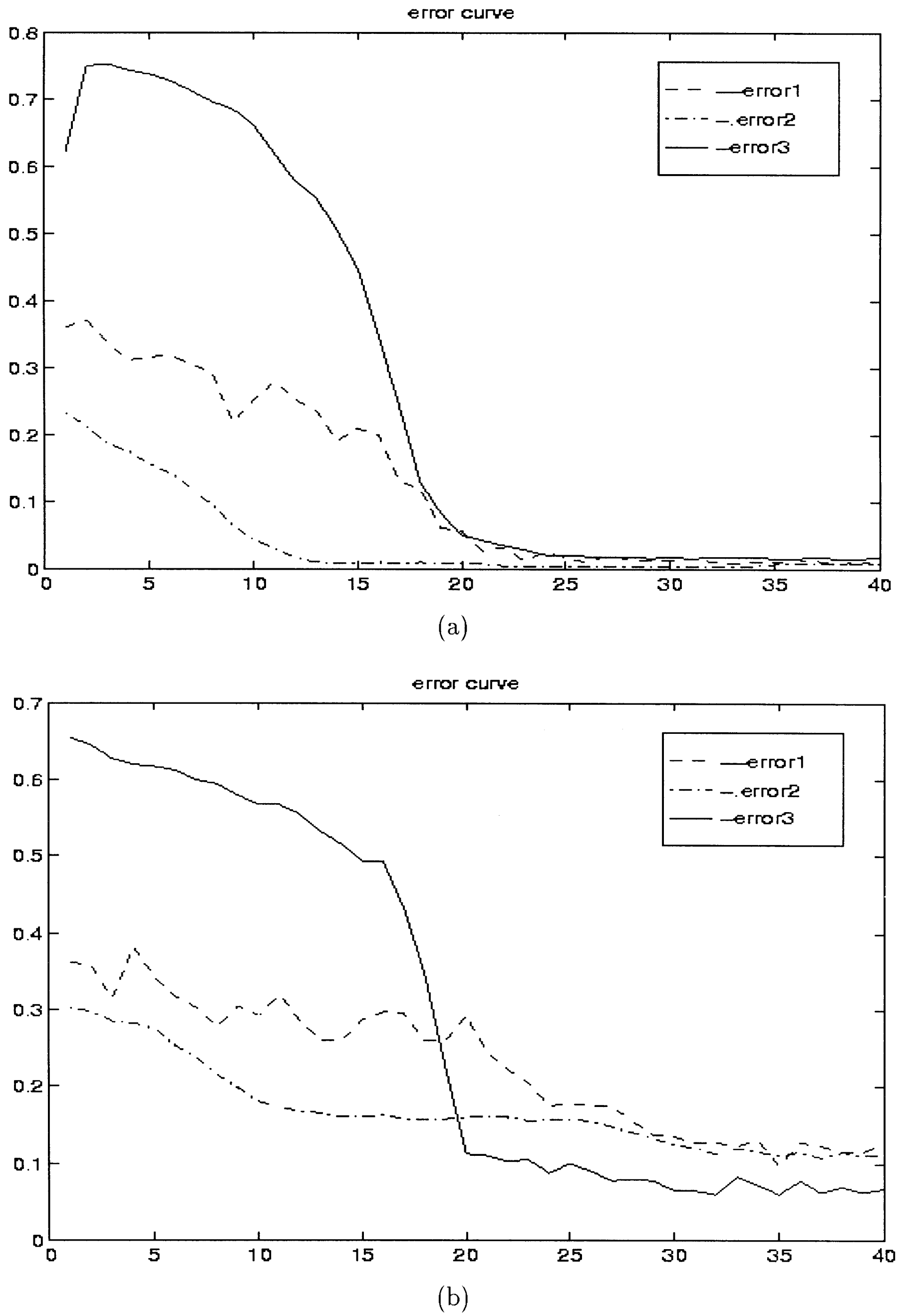
The curves error 1, error 2, error 3 correspond to

1. the random sampling of $\hat{y}$ according to $p(y|x)$,
2. the maximum posterior decision $\hat{y} = \arg\max_y p(y|x)$,
3. the direct inverse mapping $\hat{y} = Wx$ learned by Eq. (38),

which shortly denotes as Types (1) (2) and (3).

After $t = 20$ epochs, the error 1 converges and after about $t = 30$, the other error types converges too, with very small error, which can be further observed in Fig. 2a, where segments of the recovered signals y_j (1), y_j (2) , ..., $y_j(k)$ are listed below each corresponding source, with $j = 1, 2, 3$ for the recovery types corresponding to error 1, error 2, error 3, respectively. On these segments, the recovery Type (2) totally recovered the sources on each channel, the other two recovery types can totally recover the sources on channels 2, 3, although there are a little error on channels 1, 4, from which we can see that the maximum posterior recovery type performs the best. Moreover, in Fig. 3, the estimated matrix A at the last stopping point is listed together with its corresponding value of WA, the estimated noise variances $\Sigma_{x|y}$ and its determinant.

Furthermore, we add in one sensor with $d = 5$ but the other settings are kept the same. Interestingly, we can observe from the recovery errors given in Fig. 4a that the errors of the recovery Types (1) and (2) have been reduced. It indicates that the increasing of the sensor number d may help the separation and reduce the affect of noise. Next, we reduce the number of sensors to 2 only, from the recovery errors given



(a)

(b)

Fig. 4. The Hamming distance error curves. (a) On noisy linear mixture 4-5-4 for *k**-*d*-*k*.; (b) on noisy linear mixture 4-2-2 for *k**-*d*-*k*.

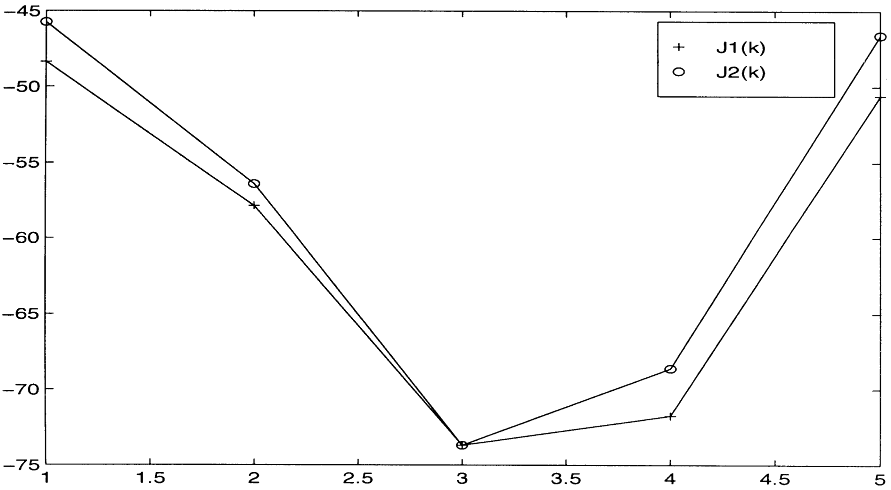


Fig. 5. The curves $J_2(k)$, $J_1(k)$ for source number detection, with the correct source number $k^* = 3$.

in Fig. 4b we still find that two channels of sources have been recovered reasonably good, although there are certain errors. This result indicates that the proposed algorithm can still reasonably work for the cases that the number d of sensors is smaller than the number k of sources.

Shown in Fig. 5 are the results of the detection of k^* by $J_2(k)$, $J_1(k)$ given by Eq. (42) and Eqs. (21) and (22). To smooth out the randomness, for each k, the average values of $J_2(k)$, $J_1(k)$ during the last 20 epochs before stopping are plotted. We see that the correct source number can be detected from the minimum point of both $J_2(k)$ and $J_1(k)$, which performs quite similarly.

6. Conclusions

The BKYY-DR system and theory is suggested for dependence reduction in general and for solving the blind source separation (BSS) problem Eq. (1) with $e \neq 0$ and the unknown number of sources as a particular example. Stochastic approximation based implementable algorithms and criteria are given for parameter learning and model selection, on a general BKYY-DR system as well as on its three typical architectures. Moreover, for the invertible forward architecture, an adaptive algorithm is obtained such that it not only is applicable to nonlinear or post-nonlinear mixtures, but also provides a new EM-type adaptive algorithm for the learned parametric mixture ICA method on linear mixtures. For the special backward architectures used on the linear or post-nonlinear mixtures under Gaussian noise, the simplified adaptive algorithm and the criterion for detecting k are also given, with an approximately optimal linear mapping $x \to y$ suggested. Also, one of its simple special case becomes the conventional factor analysis, with a new adaptive algorithm for estimation and a criterion for deciding the number of factors. The advantage of backward and forward ones are combined in the bi-directional architecture. A mean field approximation is presented for fast implementation of parameter learning and k-selection in the bi-directional architecture and shown to relate to the existing LMSER learning and the one hidden layer deterministic Helmholtz machine with new findings. Moreover, a specific degenerate case of bi-directional architecture is shown to lead to a forward non-invertible onto mapping for ICA that can be implemented adaptively. Finally, experiments on binary sources are demonstrated with successes.

Acknowledgements

This work was supported by HK RGC Earmarked Grant CUHK 339/96E. The author would like to thank Mr. Tajun Wang for his help in preparing the experiments.

Appendix A

We have

$$\begin{aligned} KL(y|x) &= \int p(y|x,\theta_{y|x}) \ln \frac{p(y|x,\theta_{y|x})}{p_G(y|x,\theta_{y|x})} \mathrm{d}y \\ &= \sum_{j=1}^{k} [p_j(x)\ln p_j(x) + (1-p_j(x))\, ln(1-p_j(x))] \\ &\quad + \frac{1}{\beta} E(p_j(x),V) + \ln Z_V, \end{aligned} \tag{56}$$

Since

$$\int p(y|x,\theta_{y|x}) E(y,V)\,\mathrm{d}y = \sum_{i,j\neq i}^{k} v_{ij} p_i(x) p_j(x) + \sum_{j=1}^{k} p_j(x)[f_j(x,\psi) + c_j] = E(p_j(x),V).$$

Furthermore, we have

$$\frac{\partial KL(y|x)}{\partial p_j(x)} = \ln \frac{p_j(x)}{1-p_j(x)} + \sum_{r\neq j}^{k} v_{jr} p_r(x) + f_j(x,\psi) + c_j,$$

$$\frac{\partial^2\ KL(y|x)}{\partial p_j(x)^2} = \frac{1}{p_j(x)} + \frac{1}{1-p_j(x)} > 0.$$

Thus, from $\partial KL(y|x)/\partial p_j(x) = 0$, we know that the solution that minimizes $KL(y|x)$ is given by Eq. (13). Moreover, with $p_r^{\mathrm{old}}(x), r = 1,\dots,k,\ r \neq j$ fixed, $p_j(x) = s(\{p_r^{\mathrm{old}}(x)\}_{r=1,r\neq j}^{k})$ is a minimum point of $KL(y|x)$. Therefore, by the solving iteration of Choice (a), we find these minimum points sequentially such that $KL(y|x)$ will keep reducing until $p_j(x), j = 1, \dots, k$ converge.

In addition, it is obvious that $s(\{p_r^{\text{old}}(x)\}_{r=1,r\neq j}^k) - p_j^{\text{old}}(x), j = 1, \dots, k$ jointly form a descent direction of $KL(y|x)$. Thus, for the solving iteration of Choice (b), as long as $\eta > 0$ is a small enough, $KL(y|x)$ will keep reducing until $p_j\,(x), j = 1, \dots,\, k$ converge.

Appendix B

We have the detailed gradient equations as follows:

$$\frac{\partial h_{y|x}(i,l)}{\partial \theta_{y|x}} = (1 + \ln p(y_l|x_i,\, \theta_{y|x}))p(y_l|x_i,\, \theta_{y|x})\,\frac{\partial \ln p(y_l|x_i,\, \theta_{y|x})}{\partial \theta_{y|x}},$$

$$\frac{\partial c_x(i,l)}{\partial \theta_{y|x}} = p(y_l^{(j)}|x_i,\, \theta_{y|x}) \ln p_{M_{x|y}}(x_i|y_l)\,\frac{\partial \ln p(y_l^{(j)}|x_i,\, \theta_{y|x})}{\partial \theta_{y|x}},$$

$$\frac{\partial c_y(i,l)}{\partial \theta_{y|x}} = \sum_{j=1}^{k} p(y_l^{(j)}|x_i,\, \theta_{y|x}) \ln p_{M_y}(y_l^{(j)})\,\frac{\partial \ln p(y_l^{(j)}|x_i\,, \theta_{y|x})}{\partial \theta_{y|x}},$$

$$\frac{\partial c_x(i,l)}{\partial \theta_{x|y}} = p_{M_{y|x}}(y_l|x_i)\,\frac{\partial \ln p(x_i|y_l, \theta_{x|y})}{\partial \theta_{x|y}},$$

$$\frac{\partial c_y(ij)}{\partial \xi_j} = p_{M_{y|x}}(y_l^{(j)}|x_i)\,\frac{\partial \ln p(y_l^{(j)}|\xi_j)}{\partial \xi_j}. \tag{57}$$

Specifically, when y is real, from Eq. (7) we have

$$h_{r,j}(y^{(j)}) = \frac{\alpha_{r,j} p(y^{(j)}|\xi_{r,j})}{p(y^{(j)}|\xi_j)}, \quad \frac{\partial \ln p(y^{(j)}|\xi_j)}{\partial \alpha_{r,j}} = \frac{h_{r,j}(y^{(j)}) - \alpha_{r,j}}{\alpha_{r,j}},$$

$$\frac{\partial \ln p(y^{(j)}|\xi_j)}{\partial \xi_{r,j}} = h_{r,j}\,(y^{(j)})\,\frac{\partial \ln p(y^{(j)}|\xi_{r,j})}{\partial \xi_{r,j}}. \tag{58}$$

Particularly, when $p(y^{(j)}|\xi_{r,j}) = G(y^{(j)},\, m_{r,j},\, \sigma_{r,j}^2)$, we have

$$\frac{\partial \ln p(y^{(j)}|\xi_{r,j})}{\partial m_{r,j}} = -(y^{(j)} - m_{r,j})\sigma_{r,j}^{-2},$$

$$\frac{\partial \ln p(y^{(j)}|\xi_{r,j})}{\partial \sigma_{r,j}^2} = \sigma_{r,j}^{-2} - (y^{(j)} - m_{r,j})^2 \sigma_{r,j}^{-4}. \tag{59}$$

Moreover, from Eq. (10) we have

$$h_r(y|x) = \frac{\beta_r G(y, f_r(x,\, \psi_r), \Sigma_{y|x,r})}{p(y|x,\, \theta_{y|x})}, \quad \frac{\partial \ln p(y|x, \theta_{y|x})}{\partial \beta_r} = \frac{1}{\beta_r}\,(h_r(y|x) - \beta_r),$$

$$y_{fr}(x) = y - f_r(x, \psi_r), \quad \frac{\partial \ln p(y|x, \theta_{y|x})}{\partial \psi_r} = h_r(y|x)\,\frac{\partial f_r(x,\, \psi_r)}{\partial \psi_r^{\mathrm{T}}}\,\Sigma_{y|x,r}^{-1} y_{fr}(x),$$

$$\frac{\partial \ln p(y|x,\, \theta_{y|x})}{\partial \Sigma_{y|x,r}} = -h_r(y|x)\Sigma_{y|x,r}^{-1}(I - y_{fr}(x) y_{fr}^{\mathrm{T}}(x)\Sigma_{y|x,r}^{-1}). \tag{60}$$

and from Eq. (15) we have

$$h_r(x|y) = \frac{\gamma_r G(x,\, g_r(y,\, \phi_r),\, \Sigma_{x|y,r})}{p(x|y,\, \theta_{x|y})},$$

$$\frac{\partial \ln p(x|y,\, \theta_{x|y})}{\partial \gamma_r} = \frac{1}{\gamma_r}\,(h_r(x|y) - \gamma_r),$$

$$x_{g,r}(y) = x - g_r(y,\, \phi_r),$$

$$\frac{\partial \ln p(x|y,\, \theta_{x|y})}{\partial \phi_r} = h_r(x|y)\,\frac{\partial g_r(y,\phi_r)}{\partial \phi_r^{\mathrm{T}}}\Sigma_{x|y,r}^{-1} x_{g,r}(y),$$

$$\frac{\partial \ln p(x|y,\, \theta_{x|y})}{\partial \Sigma_{x|y,r}} = -\, h_r(x|y)\Sigma_{x|y,r}^{-1}\,(I - x_{g,r}(y)x_{g,r}^{\mathrm{T}}(y)\,\Sigma_{x|y,r}^{-1}). \tag{61}$$

When y is binary, from Eq. (7) we have

$$\frac{\partial \ln p(y^{(j)}|\xi_j)}{\partial \xi_j} = \left(\frac{y^{(j)}}{p_j} - \frac{1-y^{(j)}}{1-p_j}\right)\frac{\mathrm{e}^{-\xi_j}}{(1+\mathrm{e}^{-\xi_j})^2}. \tag{62}$$

From Eqs. (2) and (13), we have

$$\mathrm{T}(p_j(x), y^{(j)}) = \frac{y^{(j)}}{p_j(x)} - \frac{1-y^{(j)}}{1-p_j(x)},$$

$$\frac{\partial \ln p(y|x, \theta_{y|x})}{\partial \psi} = \sum_{j=1}^{k} \mathrm{T}(p_j(x),\, y^{(j)})\,\frac{\partial p_j(x)}{\partial \psi},$$

$$\frac{\partial \ln p(y|x,\, \theta_{y|x})}{\partial c_i} = \sum_{j=1}^{k} \mathrm{T}(p_j(x), y^{(j)})\,\frac{\partial p_j(x)}{\partial c_i},$$

$$\frac{\partial \ln p(y|x,\theta_{y|x})}{\partial v_{il}} = \sum_{j=1}^{k} \mathrm{T}(p_j\,(x),\, y^{(j)})\,\frac{\partial p_j(x)}{\partial v_{il}}. \tag{63}$$

More specifically, from Eq. (13) we have that $\partial p_j(x)/\partial\psi$, $j = 1, \dots, k$ are obtained by solving the following linear equation group:

$$\frac{\beta}{(1-p_j(x))p_j(x)}\,\frac{\partial p_j(x)}{\partial \psi} + \sum_{r=1, r\neq j}^{k} v_{jr}\frac{\partial p_r(x)}{\partial \psi} + \frac{\partial f_j(x,\psi)}{\partial \psi} = 0, \quad j = 1, \dots, k; \tag{64}$$

$\partial p_j(x)/\partial c_i$, $i,j = 1, \dots, k$, are obtained by solving the following linear equation group:

$$\frac{\beta}{(1-p_j(x))p_j(x)}\,\frac{\partial p_j(x)}{\partial c_i} + \sum_{r=1, r\neq j}^{k} v_{jr}\frac{\partial p_r(x)}{\partial c_i} + 1 = 0, \quad i,j = 1, \dots, k; \tag{65}$$

and $\partial p_j(x)/\partial v_{il}, i,j,l = 1, \dots, k, i \neq l$, are obtained by solving the following linear equation group:

$$\frac{\beta}{(1-p_j(x))p_j(x)}\frac{\partial p_j(x)}{\partial v_{il}} + \sum_{r=1,r\neq j}^{k} v_{jr}\frac{\partial p_r(x)}{\partial v_{il}} + p_l(x) = 0, \quad j,l = 1,\dots,k, i = j,$$

$$\frac{\beta}{(1-p_j(x))p_j(x)}\frac{\partial p_j(x)}{\partial v_{il}} + \sum_{r=1,r\neq j}^{k} v_{jr}\frac{\partial p_r(x)}{\partial v_{il}} = 0, \quad i,j,l = 1,\dots,k, i \neq j. \tag{66}$$

Similarly, from Eqs. (7) and (18) we can get $\partial \ln p(x|y,\theta_{x|y})/\partial\phi$, $\partial \ln p(x|y,\theta_{x|y})/\partial d_j$ and $\partial \ln p(x|y,\theta_{x|y})/\partial u_{ij}$.

Appendix C

From Eq. (4) and after ignoring irrelevant terms, we have that $\min KL_{M_1,M_2}$ becomes the minimization of the following cost function:

$$J(W,\sigma_x^2,k) = \int G(y, Wx, WW^{\mathrm{T}}\sigma_x^2 I_k)p_h(x)\ln\frac{G(y,Wx,WW^{\mathrm{T}}\sigma_x^2 I_k)}{G(x,Ay,\sigma_x^2 I_d)\prod_{j=1}^{k}p(y^{(j)}|\xi_j)}\,\mathrm{d}x\,\mathrm{d}y$$

$$= 0.5[-\ln|WW^{\mathrm{T}}| - k\ln\sigma_x^2 + \mathrm{d}\ln\sigma_x^2 + \sigma_x^{-2}T_1] + T_2,$$

$$T_1 = \int G(y,Wx,WW^{\mathrm{T}}\sigma_x^2 I_k)p_h(x)(x-Ay)^{\mathrm{T}}(x-Ay)\,\mathrm{d}y\,\mathrm{d}x,$$

$$T_2 = -\sum_{j=1}^{k}\int G(y, Wx, WW^{\mathrm{T}}\sigma_x^2 I_k)p_h(x)\ln p(y^{(j)}|\xi_j)\,\mathrm{d}x\,\mathrm{d}y. \tag{67}$$

As $\sigma_x^2 \to 0$, we have

$$T_2 \to -\sum_{j=1}^{k}\int p_h(x)\ln p(w_j^{\mathrm{T}}x|\xi_j)\,\mathrm{d}x, \quad W = [w_1,\dots,w_k], \tag{68}$$

which can be regarded as being irrelevant to σ_x^2 for a small σ_x^2. From $\partial J(W,\sigma_x^2,k)/\partial\sigma_x^2 = 0$ and after ignoring the affect of T_2, we have

$$(d-k)\sigma_x^2 = T_1, \quad \sigma_x^{-2}T_1 = d-k,$$

$$J(W,\sigma_x^2,k) = 0.5[-\ln|WW^{\mathrm{T}}| + (d-k)\ln\sigma_x^2 + d - k] + T_2, \tag{69}$$

which is exactly the Eq. (9) first given in Section 3 of Ref. [19]. For a fixed σ_x^2, W is decided by maximizing $J(W) = 0.5\ln|WW^{\mathrm{T}}| + T_2$, which becomes Eq. (54) as $\sigma_x^2 \to 0, h \to 0$, by noticing Eq. (68).

References

[1] S.-I. Amari, A. Cichocki, H. Yang, A new learning algorithm for blind separation of sources, in: D.S. Touretzky et al. (eds.), Advances in Neural Information Processing 8, MIT Press, Cambridge, MA, 1996, pp. 757–763.

[2] H.B. Barlow, Unsupervised learning, Neural Comput. 1 (1989) 295–311.

[3] A.J. Bell, T.J. Sejnowski, An information-maximatization approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.

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

[5] P. Dayan, G.E. Hinton, R.N. Neal, R.S. Zemel, The Helmholtz machine, Neural Comput. 7 (5) (1995) 889–904.

[6] M. Gaeta, J.-L. Lacounme, Source separation without a priori knowledge: the maximum likelihood solution, in: Proc. European Signal Processing Conf. EUSIPCO90, 1990, pp. 621–624.

[7] C. Jutten, From source separation to independent component analysis: an introduction to special session, in: Proc. 1997 European Symp. on Artificial Neural Networks, Bruges, 16–18 April 1997, pp. 243–248.

[8] J. Karhunen, J.J. Joutsensalo, Representation and separation of signals using nonlinear PCA type Learning, Neural Networks 7 (1994) 113–127.

[9] I. King, L. Xu, Adaptive contrast enchancement by entropy maximization with a 1-K-1 constrained network, Proc. 1995 Int. Conf. on Neural Information Processing (ICONIP95), vol. II, 30 October – 3 November, 1995, Beijing, pp. 703–706.

[10] J.-P. Nadal, N. Parga, Nonlinear neurons in the low-noise limit: a factorial code maximizes information transfer, Networks 5 (1994) 565–581.

[11] B.A. Pearlmutter, L.C. Parra, A context-sensitive generalization of ICA, in: Proc. Int. Conf. on Neural Information Processing (ICONIP 96), Hong Kong, 24–27, September 1996, pp. 1235–1239.

[12] D.T. Pham, P. Garat, C. Jutten, Separation of a mixture of independent sources through a maximum likelihood approach, in: J. Vandewalle et al. (Eds.), Signal Processing VI: Theories and Applications, Elsevier, Amsterdam, 1992, pp. 771–774.

[13] S. Sharma, Applied Multivariate Techniques. Wiley, New York, 1995.

[14] A. Taleb, C. Jutten, Nonlinearity source separation: the post-nonlinear mixtures, Proc. 1997 European Symp. on Artificial Neural Networks, Bruges, 16–18 April 1997, pp. 279–284.

[15] L. Xu, Least MSE reconstruction for self-organization: (I), (II), Proc. 1991 International Joint Conference on Neural Networks(IJCNN91), Singapore, 1991, pp. 2363–2373.

[16] L. Xu, Least mean square error reconstruction for self-organizing neural-nets, Neural Networks 6 (1993) 627–648.

[17] L. Xu, Beyond PCA learnings: from linear to nonlinear and from global representation to local representation, Invited Talk, Proc. 1994 Int. Conf. on Neural Information Processing, 17–20 October, Seuol, Korea, 1994, pp. 943–949.

[18] L. Xu, Theories for unsupervised learning: PCA and its nonlinear extensions, Invited Talk, Proc. 1994 IEEE Int. Conf. on Neural Networks, vol.II, 26 June – 2 July, Orlando, Florida, pp. 1252–1257.

[19] L. Xu, Bayesian Ying–Yang learning based ICA models, Neural Networks for Signal Processing VII: Proc. IEEE Signal Processing Society Workshop, 24–26 September, FL, 1997, pp. 476–485.

[20] L. Xu, Bayesian Ying–Yang system and theory as a unified statistical learning approach: (I) unsupervised and semi-unsupervised learning, in: invited paper, S. Amari, N. Kassabov (Eds.), Brainlike Computing and Intelligent Information Systems, Springer, Berlin, 1997, pp. 241–274.

[21] L. Xu, Bayesian Ying – Yang system and theory as a unified stastical learning approach (II): from unsupervised learning to supervised learning and temporal modeling and (III): models and algorithms for dependance reduction, data dimension reduction, ICA and supervised learning, in: K.W. Wong, I. King, D.Y. Yeung (Eds.), Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective (TANC97), Springer, Berlin, 1997, pp. 25–60.

[22] L. Xu, BYY dependence reduction theory and blind source separation, Proc. Int. Joint Conf. on Neural Networks, Anchorage, AK, 5–9 May 1998, Vol. II, pp. 2495–2500.

[23] L. Xu, Bayesian Ying – Yang learning theory for data dimension reduction and determination, J. Comput. Intelligence Finance, Finance Technol. Pub. 6 (5) (1998) 6–18.

[24] L. Xu, Bayesian Ying – Yang system and theory as a unified stastical learning approach: (IV) further advances, Proc. Int. Joint Conf. on Neural Networks, Anchorage, AK, 5–9 May 1998, Vol. II, pp. 1275–1280.

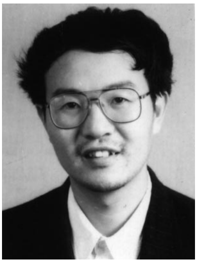
[25] L. Xu, S.-I. Amari, A general independent component analysis framework based on Bayesian–Kullback Ying–Yang Learning, Proc. Int. Conf. on Neural Information Processing (ICONIP 96), Hong Kong, 24–27 September, Springer, Singapore, 1996, pp. 1235–1239.

[26] L. Xu, C.C. Cheung, S. Amari, Further results on nonlinearity and separation capability of a linear mixture ICA method and learned parametric mixture algorithm, in: C. Fyfe (Ed.), Proc. Int. ICSC Workshop on Independence and Artificial neural networks (I&ANN'98), February 9–10, Tenerife, Spain, ICSC Academic Press, New York, 1998, pp. 39–44.

[27] L. Xu, C.C. Cheung, J. Ruan, S.-I. Amari, Nonlinearity and separation capability: further justification for the ICA algorithm with a learned mixture of parametric densities, Proc. European Symp. on Artificial Neural Networks, Bruges, 16–18 April 1997, pp. 291–296.

[28] L. Xu, C.C. Cheung, H.H. Yang, S.-I. Amari, Independent component analysis by the information-theoretic approach with mixture of density, Proc. IEEE Int. Conf. on Neural Networks (IEEE-INNS IJCNN97), vol. III, 9–12 June, Houston, TX, USA, 1997, pp. 1821–1826.

[29] L. Xu, H.H. Yang, S.-I. Amari, Signal source separation by mixtures accumulative distribution functions or mixture of bell-shape density distribution functions, Research Proposal, presented at FRONTIER FORUM (speakers: D. Sherrington, S. Tanaka, L. Xu, J.F. Cardoso), organized by S. Amari, S. Tanaka, A. Cichocki, The Institute of Physical and Chemical Research (RIKEN), Japan, 10 April, 1996.



Lei Xu (Ph.D., IEEE Senior member) is currently a Professor in the Department of Computer Science and Engineering at Chinese University, Hong Kong where he joined in 1993 as a senior lecturer first and then attained the current position in 1996. He has been a professor at Peking University since 1992, where he started as a postdoc in the Department of Maths in 1987 and then became one of ten exceptionally promoted young associate professors of the Peking University in 1988. During 1989–93, he worked as a postdoc or a senior research associate in several universities in Finland, Canada and the USA, including Harvard and MIT. He is an expresident of the Asian–Pacific Neural Networks Assembly, an associate editor of six renowned international academic journals on neurocomputing, including Neural Networks, IEEE Trans. on Neural Networks. He has published over 180 academic papers; given over ten keynote/invited/ tutorial talks as well as served as a program committee member and a session cochair in major International Neural Networks conferences in recent years, including WCNN, IEEE-ICNN, ENNS-ICANN, ICONIP, IJCNN, NNCM. He was also, a program committee chair of ICONIP'96 and a general chair of IDEAL'98. He has received several prestigious Chinese national academic awards (including National Nature Science Award and State Education Council FOK YING TUNG Award) and also some international awards, and is listed in several major Who's Who and the First Five Hundreds publications by CIBC, ABI and Marquis Who's Who.