Temporal BYY Learning for State Space Approach, Hidden Markov Model, and Blind Source Separation

Lei Xu, *Senior Member, IEEE*

***Abstract*—Temporal BYY (TBYY) learning has been presented for modeling signal in a general state space approach, which provides not only a unified point of view on Kalman filter, hidden Markov model (HMM), independent component analysis (ICA), and blind source separation (BSS) with extensions, but also further advances on these studies, including a higher order HMM, independent HMM for binary BSS, temporal ICA (TICA), and temporal factor analysis for real BSS without and with noise. Adaptive algorithms are developed for implementation and criteria are provided for selecting an appropriate number of states or sources. Moreover, theorems are given on the conditions for source separation by linear and nonlinear TICA. Particularly, it has been shown that not only non-Gaussian but also Gaussian sources can also be separated by TICA via exploring temporal dependence. Experiments are also demonstrated.**

***Index Terms*—BYY learning, factor analysis, hidden Markov model, independent component analysis (ICA), inverse mapping, Kalman filtering, source separation, state space, time series.**

I. Introduction

In the classic state space model

$$y_t = By_{t-1} + \varepsilon_t, \quad x_t = Ay_t + e_t \tag{1}$$

$$\hat{y}_t = Wx_t, \quad \text{or} \quad y_t = Wx_t + \zeta_t \tag{2}$$

a series of random observations $\mathbf{x} = \{x_t\}_{t=1}^T$ is described by (1) through a series of hidden states $\mathbf{y} = \{y_t\}_{t=1}^T$, with stochastic disturbances ε_t, e_t. The problem of describing how $\mathbf{x}$ is generated from $\mathbf{y}$ is called *modeling*, and its inverse problem that recovers $\mathbf{y}$ by (2) from $\mathbf{x}$ is called *inverse mapping or state recovery*.

Given an observation $\mathbf{x}$ only, the above problems are usually not well defined. However, it may become well defined under some assumption. In the past decades, studies are made extensively on the model equations (1) and (2) with various extensions, which can be summarized basically along three lines of developments.

The **first line** is classical in the literature of control theory and signal processing [11]. Given A and B known, and ε_t, e_t are mutually uncorrelated Gaussian white noises with known co-variances, the task of recovering $\hat{y}_t$ can be adaptively implemented by Kalman filter [4]. Various extensions are also made to the cases of ε_t, e_t with unknown co-variances, of nonlinear models and of non-Gaussian ε_t, e_t [4].

The **second line** is called *blind source separation (BSS)* [13], which is popular in the recent literature of neural networks and signal processing. In general, BSS refers to the problems of recovering source signal $\mathbf{y}$ from observation $\mathbf{x}$ only, i.e., making the inverse mapping in "blind." Though it is generally impossible, it does become possible for x_t generated from a simple model

$$x_t = Ay_t, \quad \text{where } A \text{ is a unknown invertible matrix} \tag{3}$$

and the components of y_t are mutually independent and at most one of them is Gaussian. In this case, $\hat{y}_t = Wx_t$ recovers y_t up to constant scales and a permutation of components if there is an W that makes $\hat{y}_t$ by (2) become component-wise independent [7], [20]. Thus, it is called *independent component analysis (ICA)* [13]. Many advance are achieved on ICA and its extensions, which can be briefly summarized into three stages. In the first stages, the learning on W is made with a prefixed estimation on the distribution of each component either heuristically (e.g., the sigmoid used in [3]) or based on kurtosis estimation or density expansion [2], [7]. Several learning algorithms for W have been proposed from different perspectives, ranging from contrast functions [5], [7], [12], to maximum likelihood [10], information-maximization [3], [16], and minimum mutual information [2]. Usually, these algorithms work well on the cases that the components of y_t are either all sub-Gaussians or all super-Gaussians. In the second stage, it is realized that the estimation on the distribution of each component should be learned simultaneously together with learning on W such that whether a component is super-Gaussian or sub-Gaussian can be automatically detected during learning in order to work on any combination of super-Gaussian or sub-Gaussian components of y_t. Such an idea has been implemented through adaptively estimating either the distribution of each component by a learned parametric mixture [27], [29], [30] or the kurtosis of each component [17]. In the third stage, extensions have been made toward the cases that 1) the dimension of x_t is larger than k instead of A being invertible [27], [29] and 2) some specific nonlinear system $x_t = G(y_t)$ instead of the linear model equation (3) [19], [26]. A more detailed review on the advances of ICA, as well as the relations of ICA to *factorial learning and Helmholtz machine, nonlinear Hebbian learning, nonlinear PCA and LMSER self-organization*, are referred to in [26].

Studies have also been made on noisy model $x_t = Ay_t + e_t$ with unknown A but still the independence assumption on the components of y_t. Here, we need not only to solve A and the

Manuscript received July 23, 1997; revised July 6, 1999. This work was supported under the HK RGC Earmarked Grant CUHK 339/96. The associate editor coordinating the review of this paper and approving it for publication was Prof. Yu-Hen Hu.

The author is with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: lxu@cse.cuhk.edu.hk).

Publisher Item Identifier S 1053-587X(00)04917-5.

statistical properties of e_t but also to get W to recover $\hat{y}_t$. For Gaussians y_t and e_t, when samples are i.i.d., the task reduces into the classical factor analysis in the literature of statistics [1]. Extensions to non-Gaussians y_t but Gaussian e_t are attempted under the name of ICA [6], [29]. However, being different from the case $e_t = 0$ [20], [7], in a noisy case, the fact that making $\hat{y}_t$ be independent on its components can no longer ensure the conclusion that $\hat{y}_t$ recovers y_t up to only constant scales and a permutation of its components. Strictly speaking, the name ICA is no longer appropriate for a BSS problem on a noisy model. In [26], this BSS problem is studied under the name of *dependence reduction* with not only new results on ICA for the cases of unknown source number and nonlinear model $x_t = G(y_t)$, but also two architectures for BSS on the noisy model.

The **third line** is the study on hidden Markov model (HMM) in the literature of speech processing [18]. In a classical HMM, we deal with the discrete model

$$x_t = 1, \ldots, d, \quad y_t = 1, \ldots k$$
$$p_y = p(y_t = j \,|\, y_{t-1} = i), \quad p_{x|y} = p(x_t = r \,|\, y_t = j).$$

Its relation to (1) can be observed from the fact that (1) can be equivalently described by $p(\varepsilon_t) = p(y_t \,|\, By_{t-1})$ and $p(e_t) = p(x_t \,|\, Ay_t)$. In HMM studies, $p_y, p_{x|y}$ are unknown and to be solved from observations $\mathbf{x}$, which is a *blind modeling* problem solved usually by the Baum algorithm for maximum likelihood estimation [18].

Bayesian Ying-Yang (BYY) learning system and theory is proposed as a unified statistical learning theory which is firstly proposed in 1995 [31] and rather systematically developed in past five years. Some recent reviews are referred to [21], [24], [26], and [28]. As a further development of Bayesian Ying-Yang (BYY) learning, temporal BYY (TBYY) learning system and theory is presented in Section II as a general state space model. In Section III, the connections of TBYY to Kalman filter and HMM are given with new extensions. In Section IV, the condition for BSS by linear and nonlinear temporal ICA is studied. Three methods are proposed for BSS without and with noise. Moreover, criteria are provided for selecting the number of states or sources. Experiments are given in Section V. We conclude in Section VI.

II. TBYY Learning

A. TBYY Learning System

We consider a general probabilistic state space model that describes the relation between $\mathbf{x}, \mathbf{y}$ by the joint density $p(\mathbf{x}, \mathbf{y})$ in two Bayesian representations

$$p_{M_1}(\mathbf{x}, \mathbf{y}) = p_{M_{y|x}}(\mathbf{y} \,|\, \mathbf{x}) p_{M_x}(\mathbf{x})$$
$$p_{M_2}(\mathbf{x}, \mathbf{y}) = p_{M_{x|y}}(\mathbf{x} \,|\, \mathbf{y}) p_{M_y}(\mathbf{y}). \tag{4}$$

On one hand, p_{M_1} is called Yang model, representing the observation space or called Yang space by p_{M_x} and the pathway $\mathbf{x} \to \mathbf{y}$ by $p_{M_{y|x}}$ is called Yang or forward pathway. On the other hand, we have the Ying model p_{M_2} that represents the invisible state space or Ying space by p_{M_y} and the Ying or backward pathway $\mathbf{y} \to \mathbf{x}$ by $p_{M_{x|y}}$. Such a pair of Ying-Yang models[1] is called *temporal Bayesian Ying-Yang (TBYY) learning system*.

The task of specifying all the aspects of $p_{M_{y|x}}, p_{M_x}, p_{M_{x|y}}$, p_{M_y} is called *learning* in a broad sense.

The input of observation to the system is functioned by p_{M_x}. Given a realization $\bar{\mathbf{x}} = \{\bar{x}_t\}_{t=1}^T$, p_{M_x} is usually specified by a nonparametric method based on $\bar{\mathbf{x}}$. In this paper, it is given by (11).

A twofold role is taken by p_{M_y}, which is understood from

$$p_M(\mathbf{y}) = \int p_{M_{y|x}}(\mathbf{y} \,|\, \mathbf{x}) p_{M_x}(\mathbf{x})\, d\mathbf{x}$$
$$p_M(\mathbf{x}) = \int p_{M_{x|y}}(\mathbf{x} \,|\, \mathbf{y}) p_{M_y}(\mathbf{y})\, d\mathbf{y}. \tag{5}$$

On one hand, it is the source model that $\mathbf{x}$ is generated via the Ying model. On the other hand, it is a target model matched by $p_M(\mathbf{y})$ that represents $\mathbf{x}$ via the Yang model. A certain structure can be designed for p_{M_y} according to the nature of problem and *a priori* knowledge. First, we choose the representation form for the state y_t. It can be discrete, e.g., a number $y_t = 1, \ldots, k$ or a k-bit binary code. It can also be a k-dimension real vector. Then, we specify a structure in a parametric density form[2] for p_{M_y} with a set θ_y of finite number unknown parameters, where a specific value of θ_y represents a specific density in the family of all the densities that share this given structure.

Moreover, we design the structures for each of two components $p_{M_{y|x}}$, $p_{M_{x|y}}$. First, we exclude those structures with the relationship between $\mathbf{x}$ and $\mathbf{y}$ broken, i.e., either $p_{M_{x|y}}(\mathbf{x}|\mathbf{y}) = p_{M_{x|y}}(\mathbf{x})$ or $p_{M_{y|x}}(\mathbf{y}|\mathbf{x}) = p_{M_{y|x}}(\mathbf{y})$. Then, we consider two types of structures. One is described by a parametric density with a set of finite number unknown parameters, similar to the above p_{M_y}, e.g., we have parameter sets $\theta_{y|x}$ and $\theta_{x|y}$ for $p_{M_{y|x}}$, $p_{M_{x|y}}$. The other is called *structure-free*, which means no any structural constraints such that p_{M_a} for each $a \in \{x \,|\, y, y \,|\, x\}$ is free to take any element of $\mathcal{P}_a$, where $\mathcal{P}_{x|y}$ and $\mathcal{P}_{y|x}$ denote the family of all the densities in the form $p(\mathbf{x} \,|\, \mathbf{y})$ and $p(\mathbf{y} \,|\, \mathbf{x})$, respectively.

A combination of structures for $p_{M_{y|x}}, p_{M_y}, p_{M_{x|y}}$ specifies a system architecture. There are three kinds of architectures, featured by the structures of $p_{M_{y|x}}, p_{M_{x|y}}$

- *backward architecture or shortly B-architecture* which consists of a parametric density $p_{M_{x|y}}$ for directly implementing the backward pathway, and a structure-free $p_{M_{y|x}}$ with no structure for directly implementing the forward pathway;
- *forward architecture or F-architecture* which consists of a parametric density $p_{M_{y|x}}$ for directly implementing the forward pathway and a structure-free $p_{M_{x|y}}$;
- *bi-directional architecture or BI-architecture* where both $p_{M_{y|x}}, p_{M_{x|y}}$ are parametric densities for directly implementing the bi-directional pathways.

[1] A discrete distribution is also described as a density, e.g., $p(y) = q\delta(y-1) + (1-q)\delta(y)$ describes Bernoulli distribution for a binary $y = 0$ or $y = 1$, where $\delta(y) = 0$ for $y \neq 0$, when $y = 0$ it becomes $\delta(y) = \lim_{h \to 0} h^{-1}$.

[2] A discrete distribution is automatically understand as a parametric density since it is always specified by a finite number of parameters.

The case that both $p_{M_{y|x}}$, $p_{M_{x|y}}$ are structure-free is useless because both pathways cannot be implemented.

Given all the structures designed, there remain two tasks. One is to specify all the unknown parameters $\Theta = \{\theta_{y|x}, \theta_y, \theta_{x|y}\}$, which is called *parameter learning*. The other is to decide k that companies the representation of y_t, which is called *state space complexity selection* or shortly *model selection*, since this k is an indicator of the complexity of the state space or the representation model.

B. TBYY Learning Theory

We use "Ying-Yang harmony" as the fundamental principle. Namely, we decide (Θ, k) such that the Ying model p_{M_2} and the Yang model p_{M_1} to be best harmony in a sense that we minimize both the mismatch between the two models and the diversification of the resulted Ying-Yang system.

Mathematically, we use a functional $H(M)$ to measure the degree of harmony between p_{M_1} and p_{M_2}. It further consists of a measure $Df(M)$ for the mismatching between the two models and a measure $DI(M)$ for the diversification of the resulted Ying-Yang system, such that $H(M)$ is maximized when $Df(M)$ and $DI(M)$ are both minimized.

Generally, the mismatching between two densities p, q can be measured by the so-called f-divergence

$$Df(p,q) = \int p(x) f\left(\frac{q(x)}{p(x)}\right) dx, \quad f(1) = 0$$
$$\frac{d^2 f(u)}{d^2 u} > 0 \text{ on } [0, \infty) \tag{6}$$

which was first studied by Csiszar in 1967; a nice introduction can be found in [8]. It includes Kullback divergence as a special case when $f(x) = -\ln x$

$$KL(p,q) = \int p(x) \ln \frac{p(x)}{q(x)} dx. \tag{7}$$

As shown in [21], we have the following.

Definition 1: $H(M) = -[Df(M) + DI(M)]$. $Df(M)$ is the f-divergence between p_{M_1} and p_{M_2}. The system diversification $DI(M)$ is defined as the negation of the f-divergence between a system density p_M and a standard density u_s, denoted as $-Df(p_M, u_s)$, where u_s stands for the most diversified density on the same support of p_M.

This $DI(M)$ is justified since the more the system density is close to the u_s, the more diversified the system density is, and the larger the negation of the f-Divergence is.

As shown in [21], either p_{M_1} or p_{M_2} could be used as p_M. Also, different choices are available for choosing u_s according to different types of the support of p_M. Moreover, the symmetry $Df(p,q) = Df(q,p)$ is usually not satisfied. Hence, we have several specific forms for the definition of $H(M)$.

This paper focuses on a typical situation defined by the following.

Definition 2: The Kullback divergence equation (7) is used as $Df(p,q)$ in Definition 1, with (a) p_{M_1} chosen as p_M and (b) $u_s = u_y p_{M_x}$ in a correspondence of $p_{M_1} = p_{M_{y|x}} p_{M_x}$, where u_y is the uniform density on the same support S_y of p_{M_y}.

In this case, after ignoring the term $\int p_{M_x}(\mathbf{x}) \ln p_{M_x}(\mathbf{x})\, d\mathbf{x}$, which is irrelevant to Θ, k, from (4) and (7) we have

$$\begin{aligned} H(\Theta,k) &= -[KL(\Theta,k) + DI(\Theta,k)] \\ &= \int p_{M_1} \ln p_{M_2}\, d\mathbf{x}\, d\mathbf{y} \\ KL(\Theta,k) &= \int p_{M_x}(\mathbf{x}) KL(\mathbf{x})\, d\mathbf{x} \\ DI(\Theta,k) &= \int p_{M_x}(\mathbf{x}) DI(\mathbf{x})\, d\mathbf{x} \\ KL(\mathbf{x}) &= \int p_{M_{y|x}}(\mathbf{y}\,|\,\mathbf{x}) \ln \frac{p_{M_{y|x}}(\mathbf{y}\,|\,\mathbf{x})}{p_{M_{x|y}}(\mathbf{x}\,|\,\mathbf{y}) p_{M_y}(\mathbf{y})}\, d\mathbf{y} \\ DI(\mathbf{x}) &= -\int p_{M_{y|x}}(\mathbf{y}\,|\,\mathbf{x}) \ln p_{M_{y|x}}(\mathbf{y}\,|\,\mathbf{x})\, d\mathbf{y}. \end{aligned} \tag{8}$$

In implementation, we maximize $H(M)$ stepwisely by either *Learning procedure I* or *procedure II*. Readers are referred to [21] for the procedure II. The procedure I consists of two parts as follows.

- *Parameter learning* by

$$\Theta^* = \arg\min_{\Theta} KL(\Theta, k), \quad \text{for each given } k. \tag{9}$$

- *Model selection* by

$$\begin{aligned} k^* &= \arg\max_k H(k) \\ H(k) &= -[KL(\Theta^*, k) + DI(\Theta^*, k)]. \end{aligned} \tag{10}$$

C. Recursive Implementation

We consider that $\mathbf{x}$ is causal, i.e., x_t only depends on those past $x_\tau, \tau < t$ but not on any future $x_\tau, \tau > t$. Given a realization $\bar{\mathbf{x}} = \{\bar{x}_t\}_{t=1}^T$, we have

$$\begin{aligned} p_{M_x}(\mathbf{x}) &= \prod_{t=1}^{T} p_{M_x}(x_t\,|\,\mathbf{x}_{t-1}) \\ \mathbf{x}_{t-1} &= [x_{t-1}, \ldots, x_1]^T \\ p_{M_x}(x_t\,|\,\mathbf{x}_{t-1}) &= \delta(x_t - \bar{x}_t), \quad \text{at } \mathbf{x}_{t-1} = \bar{\mathbf{x}}_{t-1} \end{aligned} \tag{11}$$

where $\mathbf{x}_0$ means empty, and here $p_{M_x}(x_t\,|\,\mathbf{x}_{t-1})$ is only partially defined to avoid any over-assumption.

Putting p_{M_x} by (11) into (8), recursively from $t = 1$ to T we get

$$\begin{aligned} KL(\Theta,k) &= \int p_{M_{y|x}}(\mathbf{y}\,|\,\bar{\mathbf{x}}) \ln \frac{p_{M_{y|x}}(\mathbf{y}\,|\,\bar{\mathbf{x}})}{p_{M_{x|y}}(\bar{\mathbf{x}}\,|\,\mathbf{y}) p_{M_y}(\mathbf{y})}\, d\mathbf{y} \\ DI(\Theta,k) &= -\int p_{M_{y|x}}(\mathbf{y}\,|\,\bar{\mathbf{x}}) \ln p_{M_{y|x}}(\mathbf{y}\,|\,\bar{\mathbf{x}})\, d\mathbf{y}. \end{aligned} \tag{12}$$

We further impose the causal assumption

$$\begin{aligned} p_{M_{y|x}}(y_t\,|\,\mathbf{x}, \mathbf{y}_{t-1}) &= p_{M_{y|x}}(y_t\,|\,\bar{x}_t, \bar{\mathbf{x}}_{t-1}, \mathbf{y}_{t-1}) \\ p_{M_{x|y}}(x_t\,|\,\mathbf{y}, \mathbf{x}_{t-1}) &= p_{M_{x|y}}(x_t\,|\,y_t, \bar{\mathbf{x}}_{t-1}, \mathbf{y}_{t-1}) \end{aligned} \tag{13}$$

where $\mathbf{y}_{t-1} = [y_t, y_{t-1}, \ldots, y_1]^T$ with $\mathbf{y}_0$ being empty. Then, we get

$$\begin{aligned} KL(\Theta, k) &= \sum_{t=1}^{T} KL_t(\Theta) \\ KL_t(\Theta) &= KL_t^{(1)} - KL_t^{(2)} - KL_t^{(3)} \\ KL_t^{(1)} &= \int p_{M_{y|x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1}) KL_t^{(1)}(\mathbf{y}_{t-1})\, d\mathbf{y}_{t-1} \\ KL_t^{(2)} &= \int p_{M_{y|x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1}) KL_t^{(2)}(\mathbf{y}_{t-1})\, d\mathbf{y}_{t-1} \\ KL_t^{(3)} &= \int p_{M_{y|x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1}) KL_t^{(3)}(\mathbf{y}_{t-1})\, d\mathbf{y}_{t-1} \end{aligned} \tag{14}$$

$$\begin{aligned} KL_t^{(1)}(\mathbf{y}_{t-1}) &= \int p_{M_{y|x}}(y_t \,|\, \bar{x}_t, \bar{\mathbf{x}}\mathbf{y}_{t-1}) \\ &\quad \times \ln p_{M_{y|x}}(y_t \,|\, \bar{x}_t, \bar{\mathbf{x}}\mathbf{y}_{t-1})\, dy_t \\ KL_t^{(2)}(\mathbf{y}_{t-1}) &= \int p_{M_{y|x}}(y_t \,|\, \bar{x}_t, \bar{\mathbf{x}}\mathbf{y}_{t-1}) \\ &\quad \times \ln p_{M_{x|y}}(\bar{x}_t \,|\, y_t, \bar{\mathbf{x}}\mathbf{y}_{t-1})\, dy_t \\ KL_t^{(3)}(\mathbf{y}_{t-1}) &= \int p_{M_{y|x}}(y_t \,|\, \bar{x}_t, \bar{\mathbf{x}}\mathbf{y}_{t-1}) \\ &\quad \times \ln p_{M_y}(y_t \,|\, \mathbf{y}_{t-1})\, dy_t \\ p_{M_{y|x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1}) &= \prod_{\tau=1}^{t-1} p_{M_{y|x}}(y_{t-\tau} \,|\, \bar{x}_{t-\tau}, \bar{\mathbf{x}}\mathbf{y}_{t-1-\tau}) \\ \bar{\mathbf{x}}\mathbf{y}_{t-1} &= \{\bar{\mathbf{x}}_{t-1}, \mathbf{y}_{t-1}\}. \end{aligned}$$

For a B-architecture, the structure-free $p_{M_{y|x}}$ is decided by minimizing $KL(\Theta, k)$. From (12), we get

$$\begin{aligned} p_{M_{y|x}}(\mathbf{y} \,|\, \bar{\mathbf{x}}) &= \frac{p_{M_{x|y}}(\bar{\mathbf{x}} \,|\, \mathbf{y}) p_{M_y}(\mathbf{y})}{p_M(\mathbf{x})} \\ KL(\Theta, k) &= -\ln p_M(\mathbf{x}) \\ p_M(\mathbf{x}) &= \int p_{M_{x|y}}(\mathbf{x} \,|\, \mathbf{y}) p_{M_y}(\mathbf{y})\, d\mathbf{y}. \end{aligned} \tag{15}$$

In this case, (9) becomes the maximum likelihood (ML) estimation of the density $p_M(\mathbf{x})$.

For a F-architecture, the structure-free $p_{M_{x|y}}$ is also decided by minimizing $KL(\Theta, k)$ in (14), resulting in $p_{M_{x|y}} = \delta(x_t - \bar{x}_t)$ and $KL_t^{(2)} = -\ln \delta(x_t - \bar{x}_t)$ which is irrelevant to Θ, k. Thus, we have

$$KL_t(\Theta) = KL_t^{(1)} - KL_t^{(3)}. \tag{16}$$

By noticing that $DI(\Theta, k) = -\sum_{t=1}^{T} KL_t^{(1)}$, we can also similarly get the recursive form for $H(k)$ by

$$\begin{aligned} H(k) &= -[KL(\Theta, k) + DI(\Theta, k)] \\ &= \sum_{t=1}^{T} \left(KL_t^{(2)} + KL_t^{(3)} \right). \end{aligned} \tag{17}$$

D. Approximate Implementation

Except some simple cases, the integrals over $\mathbf{y}_{t-1}$ are usually difficult or expensive in implementation. To simplify it, we consider $\int p(u) T(u)\, du$ by Taylor expansion of $T(u)$ around the mean $\hat{u} = \int u p(u)\, du$

$$\begin{aligned} T(u) &\approx T(\hat{u}) + (u - \hat{u})^T G(\hat{u}) + c_T (u - \hat{u})^T H(\hat{u})(u - \hat{u}) \\ c_T &= \begin{cases} 0, & \text{the first-order expansion only} \\ 0.5, & \text{up to the second-order expansion} \end{cases} \end{aligned}$$

where $G(u)$, $H(u)$ are the gradient and Hessian of $T(u)$, respectively. Since $\int p(u)(u - \hat{u})\, du = 0$, we get

$$\int p(u) T(u)\, du \approx T(\hat{u}) + c_T \,\mathrm{Tr}[\Sigma H(\hat{u})] \tag{18}$$

where Σ is the covariance matrix of $p(u)$ and $\mathrm{Tr}[C]$ is the trace of matrix C.

Arranging $\mathbf{y}_{t-1}$ in a vector and regarding it as u and regarding $p_{M_{y|x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1})$ as $p(u)$, in help of (18), from (14) and (17), we get[3]

$$\begin{aligned} KL_t(\Theta) &= \sum_{t=1}^{T} \big\{ KL_t(\hat{\mathbf{y}}_{t-1}) \\ &\quad + c_T \,\mathrm{Tr}\big[\Sigma_{t-1}^{y}(\hat{\mathbf{y}}_{t-1}) H_{KL}(\hat{\mathbf{y}}_{t-1})\big] \big\} \\ KL_t(\mathbf{y}_{t-1}) &= KL_t^{(1)}(\mathbf{y}_{t-1}) - KL_t^{(2)}(\mathbf{y}_{t-1}) \\ &\quad - KL_t^{(3)}(\mathbf{y}_{t-1}) \\ H(k) &= -\sum_{t=1}^{T} \big\{ H_t(\hat{\mathbf{y}}_{t-1}) + c_T \,\mathrm{Tr}\big[\Sigma_{t-1}^{y} H_H(\hat{\mathbf{y}}_{t-1})\big] \big\} \\ H_t(\mathbf{y}_{t-1}) &= KL_t^{(2)}(\mathbf{y}_{t-1}) + KL_t^{(3)}(\mathbf{y}_{t-1}) \end{aligned} \tag{19}$$

where H_{KL}, H_H are the Hessians of $KL_t(\mathbf{y}_{t-1})$, $H_t(\mathbf{y}_{t-1})$ at $\hat{\mathbf{y}}_{t-1}$, and Σ_{t-1}^{y} is the covariance matrix of $p_{M_{y|x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1})$ with respect to $\mathbf{y}_{t-1}$.

Moreover, $\hat{\mathbf{y}}_{t-1}$ is obtained recursively at each $t > 2$ by

$$\begin{aligned} \hat{y}_{t-1} &= \int p_{M_{y|x}}(\mathbf{y}_{t-2} \,|\, \bar{\mathbf{x}}_{t-2}) y_{t-1}(\mathbf{y}_{t-2})\, d\mathbf{y}_{t-2} \\ \hat{y}_{t-1}(\mathbf{y}_{t-2}) &= \int y_{t-1} p_{M_{y|x}}(y_{t-1} \,|\, \bar{x}_{t-1}, \bar{\mathbf{x}}\mathbf{y}_{t-2})\, dy_{t-1}, \quad \text{or} \\ \hat{y}_{t-1} &\approx y_{t-1}(\hat{\mathbf{y}}_{t-2}) + c_y \,\mathrm{Tr}\big[\Sigma_{t-2}^{y} H_{y|x}(\hat{\mathbf{y}}_{t-2})\big] \end{aligned} \tag{20}$$

with $c_y = 0$ and $c_y = 0.5$ for the first and second-order approximation, respectively. $H_{y|x}$ are the Hessians of $\hat{y}_{t-1}(\mathbf{y}_{t-2})$ at $\hat{\mathbf{y}}_{t-2}$. Furthermore, Σ_{t-2}^{y} are also obtained similarly by

$$\begin{aligned} \Sigma_{t-1}^{y} &= \int p_{M_{y|x}}(\mathbf{y}_{t-2} \,|\, \bar{\mathbf{x}}_{t-2}) \Sigma_{t-1}^{y}(\mathbf{y}_{t-2})\, d\mathbf{y}_{t-2} \\ \Sigma_{t-1}^{y}(\mathbf{y}_{t-2}) &= \int (y_{t-1} - \hat{y}_{t-1})(y_{t-1} - \hat{y}_{t-1})^T \\ &\quad \times p_{M_{y|x}}(y_{t-1} \,|\, \bar{x}_{t-1}, \bar{\mathbf{x}}\mathbf{y}_{t-2})\, dy_{t-1}. \end{aligned} \tag{21}$$

Letting $\sigma_{i,j}^2$ and $\sigma_{i,j}^2(\mathbf{y}_{t-2})$ denote the (i,j)th element of Σ_{t-1}^{y}, and $\Sigma_{t-1}^{y}(\mathbf{y}_{t-2})$, respectively, we have approximately

$$\sigma_{i,j}^2 \approx \sigma_{i,j}^2(\mathbf{y}_{t-2}) + c_\sigma \,\mathrm{Tr}\big[\Sigma_{t-2}^{y} H_{ij}(\hat{\mathbf{y}}_{t-2})\big] \tag{22}$$

[3]When y_t is binary or discrete, we can still regard that it is real with $p_{M_{y|x}}$ in a type of δ-density. That is, we can always use (18) as long as a Taylor expansion is legal for the part $T(u)$.

with $c_\sigma = 0$ and $c_\sigma = 0.5$ for the first- and second-order approximation, respectively. H_{ij} is the Hessian of $\sigma^2_{i,j}(\mathbf{y}_{t-2})$ at $\hat{\mathbf{y}}_{t-2}$.

For a B-architecture, when we consider $c_T = 0$, the minimization of $KL_t(\mathbf{y_{t-1}})$ in (19) with respect to each free $p_{M_{y\,|\,x}}(y_t\,|\,\bar{x}_t, \bar{\mathbf{x}}\hat{\mathbf{y}}_{t-1})$ will result in

$$\begin{aligned} &KL_t(\Theta, k) = -\ln p_M(\bar{x}_t\,|\,\bar{\mathbf{x}}\hat{\mathbf{y}}_{t-1}) \\ &\bar{\mathbf{x}}\hat{\mathbf{y}}_{t-1} = \{\bar{\mathbf{x}}_{t-1}, \hat{\mathbf{y}}_{t-1}\}, \\ &p_{M_{y\,|\,x}}(y_t\,|\,\bar{x}_t, \bar{\mathbf{x}}\hat{\mathbf{y}}_{t-1}) \\ &\quad = \frac{p_{M_{x\,|\,y}}(\bar{x}_t\,|\,y_t, \bar{\mathbf{x}}\hat{\mathbf{y}}_{t-1}) p_{M_y}(y_t\,|\,\hat{\mathbf{y}}_{t-1})}{p_M(\bar{x}_t\,|\,\bar{\mathbf{x}}_{t-1})}. \end{aligned} \tag{23}$$

E. Markovian Convention and Parameterization

We call the number q of samples in a set $\mathbf{z_{t-1}} = \{\mathbf{z_{t-1}}, \ldots, \mathbf{z_{t-q}}\}$ the order of the set, or we say the set has an order q. In previous discussions, the notations of $\mathbf{x}_{t-1}$, $\mathbf{y}_{t-1}$ are in their most general case of order $q = t-1$. However, the finite order Markovian convention has been widely adopted, i.e., what happens at the current t relates to only a finite number of past samples. Thus, in the rest of the paper, we take a convention that $\mathbf{x}_{t-1}, \mathbf{y}_{t-1}$ always have finite orders q_x, q_y, with either $q_x = q_y$ or $q_x \neq q_y$. Moreover, each of q_x, q_y may have different values as it locates after the conditioning bar "|" in $p_{M_{y\,|\,x}}, p_{M_{x\,|\,y}}, p_{M_y}$.

The finite order convention also facilitates to describe $p_{M_{y\,|\,x}}$, $p_{M_{x\,|\,y}}$, p_{M_y} in parameter structures. These structures share a common feature that each is a density of a n-dimensional vector u conditioning on a d-dimensional vector v. In the sequel, we introduce some examples.

For a binary u with each component u_j taking either 0 or 1, we consider the structure of the so-called generalized linear regression [15]

$$\begin{aligned} E(u\,|\,\{v_j\}) = f(\nu) &= [f(\nu_1), \ldots, f(\nu_n)]^T \\ \nu &= \theta v + c_\varepsilon \\ f(r) &\text{ is usually a monotonic function.} \end{aligned} \tag{24}$$

Specifically, we consider two such structures. One is the so called *Softmax* structure [15]

$$p_S(u\,|\,\nu) = \sum_{i=1}^n u_i f(\nu_i) \Big/ \sum_{i=1}^n f(\nu_i) \qquad \text{e.g., } f(r) = e^r \tag{25}$$

for the case of decisive binary code u, i.e.,

$$\text{each component } u_j \text{ is either 0 or 1,} \quad \text{and} \quad \sum_{j=1}^n u_j = 1. \tag{26}$$

The other one is the independent Bernoulli structure

$$\begin{aligned} p_B(u\,|\,\nu) &= \prod_{i=1}^n f(\nu_i)^{u_i}(1 - f(\nu_i))^{1-u_i} \\ &0 < f(r) < 1 \text{ is a sigmoid function} \\ &\text{e.g., } f(r) = 1/(1+e^{-r}). \end{aligned} \tag{27}$$

For the case of real vector u, we consider a mixture of m Gaussian regressions

$$p_G(u\,|\,v) = \sum_{i=1}^m p_S(b_i\,|\,\omega) G\left(u, \theta^{(i)} v + c_\varepsilon^{(i)}, \Sigma^{(i)}\right) \tag{28}$$

where $G(u, \mu, \Sigma)$ denotes a Gaussian of mean μ and covariance Σ. $p_S(b_i\,|\,\omega), \omega = \phi v + c_\omega$ is given by a softmax structure as in (25) and usually called the gating net [24], which weights each Gaussian regression according to the current input and reduces into a constant $\alpha_i > 0$ when $\omega = c_\omega$. Also, $b_i = [b_1, \ldots, b_m]^T$ with its ith element being 1 and all the others being 0. Typical examples for the structure equation (28) is the mixture-of-expert models and the extended normalized RBF net [24].

F. Gradient Based Algorithm

Under a fixed k, we can implement (9) recursively from $t = 1$ to T by updating Θ in a gradient based technique to reduce each $KL_t(\Theta, k)$. Observing (14) and (19), though we should consider $KL_t(\Theta)$ as a whole for updating $p_{M_{y\,|\,x}}$, we only need consider $KL_t^{(2)}$ for updating $p_{M_{x\,|\,y}}$ and $KL_t^{(3)}$ for updating p_{M_y}, respectively. Therefore, at each t we can implement the following two steps.

Step 1) (a) For the B-architecture, get $p_{M_{y\,|\,x}}$ by (15) or $p_{M_{y\,|\,x}}(y_t\,|\,\bar{x}_t, \bar{\mathbf{x}}\hat{\mathbf{y}}_{t-1})$ by (23),
(b) For the BI-architecture and F-architecture, update $\theta^{\text{new}}_{y\,|\,x} = \theta^{\text{old}}_{y\,|\,x} - \lambda \nabla_{\theta^{\text{old}}_{y\,|\,x}} KL_t(\Theta)$;

Step 2) Update $\theta^{\text{new}}_y = \theta^{\text{old}}_y + \lambda \nabla_{\theta^{\text{old}}_y} KL_t^{(3)}$, and except for the F-architecture, also update

$$\theta^{\text{new}}_{x\,|\,y} = \theta^{\text{old}}_{x\,|\,y} + \lambda \nabla_{\theta^{\text{old}}_{x\,|\,y}} KL_t^{(2)} \tag{29}$$

where $\lambda > 0$ is a stepsize, and ∇_θ means taking gradient with respect to θ. This algorithm is given in a unified form that applies to both (14) and (19), with each of the three architectures. When (19) is used, $KL_t^{(1)}(\hat{\mathbf{y}}_{t-1}), KL_t^{(2)}(\hat{\mathbf{y}}_{t-1}), KL_t^{(3)}(\hat{\mathbf{y}}_{t-1})$ are used in place of the above $KL_t^{(1)}, KL_t^{(2)}, KL_t^{(3)}$, with $\hat{y}_{t-1}$ given by (20). We can make several different levels of approximation by choosing each of c_T, c_y, and c_σ to 0 or 1.

Specifically, for the structures (27) and (28), the implementation of (29) often encounters the costs and gradients given in Tables I and II.

III. Identifiable Model and Source Recovery

A. Identifiable Model

Definition 3: Given k and the structures of $p_{M_{x\,|\,y}}(\mathbf{x}\,|\,\mathbf{y})$, $p_{M_y}(\mathbf{y})$ with parameters $\theta = \{\theta_{x\,|\,y}, \theta_y\}$, a model $p_M(\mathbf{x})$ in (5) is said to be identifiable if each specific value of θ uniquely specifies a density of $p_M(\mathbf{x})$.

Given a $p_{M_x}(\mathbf{x})$, if it is generated from an identifiable model with both k and the structures of $p_{M_{x\,|\,y}}(\mathbf{x}\,|\,\mathbf{y}), p_{M_y}(\mathbf{y})$ known, the constraint $p_M(\mathbf{x}) = p_{M_x}(\mathbf{x})$ will lead us to a unique solution of θ which recovers the original densities of $p_{M_{x\,|\,y}}(\mathbf{x}\,|\,\mathbf{y})$, $p_{M_y}(\mathbf{y})$ that generate $p_{M_x}(\mathbf{x})$. Moreover, it also uniquely specifies $p_{M_{y\,|\,x}}$ by (15). Further putting it into (5), we get

$$\int p_{M_{y\,|\,x}}(\mathbf{y}\,|\,\mathbf{x}) p_{M_x}(\mathbf{x})\, d\mathbf{x} = p_M(\mathbf{y}) = p_{M_y}(\mathbf{y}). \tag{30}$$

TABLE I
BINARY-STATE BASED COSTS AND GRADIENTS

Some Binary-State Based Costs
$L_b = -\int p_B(u\|\mu)\ln p_B(u\|\nu)du = -\sum_{i=1}^{n} c_h(f(\mu_i), f(\nu_i))$,
$H_b = \int p_B(u\|\mu)\ln p_B(u\|\mu)du = \sum_{i=1}^{n} c_h(f(\mu_i), f(\mu_i))$,
$c_h(p,q) = p\ln q + (1-p)\ln(1-q), \quad D_f = d_g[f(\mu_i) - f^2(\mu_i)]$,
$L_{gb} = -\int p_B(u\|\mu)\ln G(x, \varphi_0 u + \rho, \Sigma_e)du = 0.5\{\ln\|\Sigma_e\| + Tr[\Sigma_e^{-1}(ee^T + \varphi_0 D_f \varphi_0^T)]$,
$\nu = \theta v + c_\varepsilon,\ \mu = \psi z + c_\zeta,\ \rho = \varphi w + c_e, e = x - (\varphi_0 f(\mu) + \rho)$,
Related Gradient Directions
$h = [h_1, \cdots, h_d]^T,\ h_i = f'(\nu_i)(\frac{f(\mu_i)}{f(\nu_i)} - \frac{1-f(\mu_i)}{1-f(\nu_i)}),\ f'(r) = \frac{df(r)}{dr}$,
$\nabla_\theta L_b = hv^T,\ \nabla_{c_\theta} L_b = h,\ \nabla_\psi(H_b + L_b) = gz^T$,
$\nabla_{c_\zeta}(H_b + L_b) = g,\ g = [g_1, \cdots, g_d]^T,\ g_i = f'(\mu_i)[\ln\frac{f(\mu_i)}{1-f(\mu_i)} - \ln\frac{f(\nu_i)}{1-f(\nu_i)}]$,
$\nabla_{\Sigma_e} L_{gb} = \Sigma_e^{-1} - \Sigma_e^{-1}\{ee^T + \varphi_0 D_f \varphi_0^T\}\Sigma_e^{-1}$,
$\delta_1 = d_g[f'(\mu_i)]\varphi_0^T\Sigma_e^{-1}\{e + \varphi_0[\vec{1} - 2f(\mu)]\},\ \nabla_\psi L_{gb} = \delta_1 z^T,\ \nabla_{c_\zeta} L_{gb} = \delta_1$,
$\nabla_{\varphi_0} L_{gb} = \varphi_0^T\Sigma_e^{-1}\{\varphi_0 D_f - ef^T(\mu)\},\ \delta_2 = -\Sigma_e^{-1}e,\ \nabla_\varphi L_{gb} = \delta_2 w^T,\ \nabla_{c_e} L_{gb} = \delta_2$,
ν_i, μ_i are the i-th elements of ν, μ, $\vec{1}$ is a vector with all the elements being 1, $d_g[r_i]$ is a diagonal matrix with its i-th element r_i, and $c_0 = n\ln 2\pi$ was ignored in L_{gb}.

TABLE II
GAUSSIAN-STATE BASED COSTS AND GRADIENTS

Some Gaussian-State Based Costs
$H_g = \int G(u, f(\mu), \Sigma_\zeta)\ln G(u, f(\mu), \Sigma_\zeta)du = -0.5(\ln\|\Sigma_\zeta\| + n)$,
$L_g = -\int G(u, f(\mu), \Sigma_\zeta)\ln G(u, \nu, \Sigma_\varepsilon)du = 0.5\{Tr[\Sigma_\zeta\Sigma_\varepsilon^{-1}]$
$+\ln\|\Sigma_\varepsilon\| + (f(\mu) - \nu)^T\Sigma_\varepsilon^{-1}(f(\mu) - \nu)\}$,
$L_{gg} = -\int G(u, f(\mu), \Sigma_\zeta)\ln G(x, \varphi_0 u + \rho, \Sigma_e)du =$
$0.5\{Tr[\Sigma_e^{-1}\varphi_0\Sigma_\zeta\varphi_0^T] + \ln\|\Sigma_e\| + e^T\Sigma_e^{-1}e\}$,
Related Gradient Directions
$\nabla_{\Sigma_\zeta}(H_g + L_g + L_{gg}) = 0.5(\Lambda_\varepsilon^{-1} + \varphi_0^T\Sigma_e^{-1}\varphi_0 - \Sigma_\zeta^{-1})$,
$\nabla_\psi(L_g + L_{gg}) = d_g[f'(\mu_i)]\{\Lambda_\varepsilon^{-1}(f(\mu) - \nu) - \varphi_0^T\Sigma_e^{-1}e\}z^T$,
$\nabla_{c_\zeta}(L_g + L_{gg}) = d_g[f'(\mu_i)]\{\Lambda_\varepsilon^{-1}(f(\mu) - \nu) - \varphi_0^T\Sigma_e^{-1}e\}$,
$\nabla_{\Sigma_\varepsilon} L_g = 0.5\{\Sigma_\varepsilon^{-1} - \Sigma_\varepsilon^{-1}[\Sigma_\zeta + (f(\mu) - \nu)(f(\mu) - \nu)^T]\Sigma_\varepsilon^{-1}\}$,
$\nabla_\theta L_g = -\Sigma_\varepsilon^{-1}(f(\mu) - \nu)v^T,\ \nabla_{c_\varepsilon} L_g = -\Sigma_\varepsilon^{-1}(f(\mu) - \nu)$,
$\nabla_{\Sigma_e} L_{gg} = 0.5\{\Sigma_e^{-1} - \Sigma_e^{-1}[\varphi_0\Sigma_\zeta\varphi_0^T + ee^T]\Sigma_e^{-1}\},\ \delta_2 = -\Sigma_e^{-1}e$,
$\nabla_\varphi L_{gg} = \delta_2 w^T,\ \nabla_{c_e} L_{gg} = \delta_2,\ \nabla_{\varphi_0} L_{gg} = \varphi_0^T\Sigma_e^{-1}[\varphi_0\Sigma_\zeta - ef^T(\mu)]$,
μ, ν, e are given in Tab. 1, and constants are ignored in H_g, L_g, L_{gg}

In this case, we say that $p_{M_{y|x}}$ recovers the source $\mathbf{y}$ in distribution. Thus, we have the following.

Theorem 1: Given a $p_{M_x}(\mathbf{x})$ obtained from observations, a necessary condition for recovering the original source in distribution is that the observations comes from an identifiable model.

This theorem indicates three issues to be considered for source recovery. First, we should only consider those structures of $p_{M_{x|y}}, p_{M_y}$ such that its $p_M(\mathbf{x})$ in (5) is identifiable for some $k^* > 0$. Second, we should select an appropriate k^*, in help of the model selection (10). Also, we should understand the nature of the problem to be solved and best use a priori knowledge such that the structures of $p_{M_{x|y}}, p_{M_y}$ can be designed appropriately.

B. Kalman Filter, Extensions, and Model Identification

The simplest case of identifiable model is that $p_{M_x}(\mathbf{x})$ is from a model with $p_{M_{x|y}}$, p_{M_y} specified already. The well known Kalman filter is such an example.

We can implement (29) based on the gradient directions given in Table I, which leads to the specific algorithm given in Table IV.

When only the first-order serial relationship is considered as in (1), from (15) we get

$$\begin{aligned} p_{M_{y|x}}(y_t \,|\, \bar{x}_t) &= \frac{p_{M_{x|y}}(\bar{x}_t \,|\, y_t) p_{M_y}(y_t \,|\, \bar{x}_{t-1})}{p_M(\bar{x}_t)} \\ p_M(\bar{x}_t) &= \int p_{M_{x|y}}(\bar{x}_t \,|\, y_t) p_{M_y}(y_t)\, dy_t \\ p_{M_y}(y_t \,|\, \bar{x}_{t-1}) &= \int p_{M_y}(y_t \,|\, y_{t-1}) \\ &\quad \times p_{M_{y|x}}(y_{t-1} \,|\, \bar{x}_{t-1})\, dy_{t-1}. \end{aligned} \tag{31}$$

Putting them on (1), when A, B are known and ε_t, e_t in (1) are from Gaussians of zero means and known Σ_{ε_t} and Σ_{e_t}, respectively, we have $p_{M_y}(y_t \,|\, y_{t-1}) = G(y_t, By_{t-1}, \Sigma_{\varepsilon_t})$, $p_{M_{x|y}}(\bar{x}_t \,|\, y_t) = G(x_t, Ay_t, \Sigma_{e_t})$. In a comparison with (5.8.8) and (5.8.9) in [4], we find that the first equation in (31) becomes exactly the Kalman filter.

We can extend the Kalman filter to nonlinear models and non-Gaussian noises by letting $p_{M_{y|x}}(y_t \,|\, \bar{x}_t, y_{t-1})$ in the structure of (28) with $\theta_{y|x}$ learned by Step 1 in (29). Details are in [22].

Also, we can consider in (1) with Σ_{ε_t} and Σ_{e_t} or some part of A, B unknown. We recover these unknown by (29). The task is called system identification in literature of control theory. The

TABLE III
THE UPDATING EQUATIONS FOR $\min_\theta KL_t(\theta)$ IN TFA

$$\Sigma_{\zeta,t+1} = \Sigma_{\zeta,t} - \lambda(\Lambda_{\varepsilon,t}^{-1} + \sigma_{e,t}^{-2} A_t^T A_t - \Sigma_{\zeta,t}^{-1}),\ z = \Lambda_{\varepsilon,t}^{-1}\varepsilon_t - \sigma_{e,t}^{-2} A_t^T e_t,$$
$$K_{t+1} = K_t - \lambda z \mathbf{x}_t^T,\ H_{t+1} = H_t - \lambda z \hat{\mathbf{y}}_{t-1}^T,\ c_{\zeta,t+1} = c_{\zeta,t} - \lambda z,$$
$$\Lambda_{\varepsilon,t+1} = (1-\lambda)\Lambda_{\varepsilon,t} + \lambda diag[\Sigma_{\zeta,t} + \varepsilon_t \varepsilon_t^T],\ E_{t+1} = E_t + \lambda \varepsilon_t \hat{\mathbf{y}}_{t-1}^T,$$
$$c_{\varepsilon,t+1} = c_{\varepsilon,t} + \lambda\varepsilon_t,\ \sigma_{e,t+1}^2 = (1-\lambda)\sigma_{e,t}^2 + \lambda d^{-1}[Tr[A_t \Sigma_{\zeta,t} A_t^T] + \|\bar{e}_t\|^2],$$
$$A_{t+1} = A_t - \lambda[A_t \Sigma_{\zeta,t} - e_t \hat{y}_t^T],\ B_{t+1} = B_t + \lambda \epsilon_t \hat{\mathbf{y}}_{t-1}^T,\ c_{e,t+1} = c_{e,t} + \lambda e_t,$$

where $diag[A]$ takes the diagonal part of A.

TABLE IV
AN INDEPENDENT HIGHER ORDER HMM ALGORITHM

Step 1: update K, H, c_ζ by gradient descent with the gradient given by $-[\nabla_\psi(H_b + L_b) + \nabla_\psi L_{gb}]$ for updating K, H and by $-[\nabla_{c_\zeta}(H_b + L_b) + \nabla_{c_\zeta} L_{gb}]$ for updating c_ζ;

Step 2: update E by gradient descent with the gradient $-\nabla_\theta L_b$, and update c_ε by gradient descent with the gradient $-\nabla_{c_\varepsilon} L_b$. Then update $\sigma_{e,t+1}^2 = (1-\lambda)\sigma_{e,t}^2 + \lambda d^{-1}[Tr[A_t \Lambda_{f,t} A_t^T] + \|e_t\|^2]$, $A_{t+1} = A_t - \lambda[A_t \Lambda_{f,t} - e_t \hat{y}_t^T]$, and update B, c_e as in Tab.3.

task is closely related to find a suitable number k of states, which can be made by (10). From (14), (17), and (31), we have

$$H(k) = \sum_{t=1}^{T} p_M(\bar{x}_t) - \sum_{t=1}^{T} \int p_{M_{y|x}}(y_{t-1} \mid \bar{x}_{t-1}) p_{M_{y|x}}(y_t \mid \bar{x}_t, y_{t-1}) \times \ln p_{M_{y|x}}(y_t \mid \bar{x}_t, y_{t-1})\, dy_{t-1}\, dy_t \quad (32)$$

which can be recursively computed during the implementation of the Kalman filter.

C. HMM, State Selection, and Higher Order HMM

The HMM equation (4) is another simplest identifiable model, where $p_{M_{x|y}}, p_{M_y}$ are unknown but usually can be uniquely specified because y_t has a simple representation.

It follows from (4) that $p_{M_y}(y_t \mid y_{t-1}) = p_y$, $p_{M_{x|y}}(\bar{x}_t \mid y_t) = p_{x|y}$. From (15) and (31), the minimization of $KL(\Theta, k)$ with respect to $p_{M_{x|y}}, p_{M_y}$ leads to the ML learning on the HMM, which can be implemented either recursively by (29) or optimally in a batch way by the Baum algorithm [18].

We give some new results. First, we can select k by (32) after parameter learning, where the integrals become summations because discrete densities.

Second, from (29) we can get an approximate but adaptive learning algorithm for estimating parameters in the HMM equation (4). Details are referred to [25].

Third, though the HMM equation (4) can be extended to include a higher order serial relation, the implementing complexity of the Baum algorithm will grow rapidly. Here, we consider a variant model for higher order HMM. We let $p_{M_{y|x}}(y_t \mid \bar{x}_t, \bar{\mathbf{x}}\mathbf{y}_{t-1})$ and $p_{M_y}(y_t \mid \hat{\mathbf{y}}_{t-1})$ to have the structure of (25) with u as y_t and ν given by (24) with $\bar{x}_t, \bar{\mathbf{x}}\mathbf{y}_{t-1}$ as v for $p_{M_{y|x}}$ and $\hat{\mathbf{y}}_{t-1}$ as v for p_{M_y}. Similarly, we let $p_{M_{x|y}}(\bar{x}_t \mid y_t, \bar{\mathbf{x}}\mathbf{y}_{t-1})$ to have (a) the structure of (25) for $x_t = 1, \ldots, d$, (b) the structure of (27) for a binary vector x_t and (c) the structure of (28) for a real vector x_t.

We use (29) for learning in the help of the approximation by (19) and (20), and use (10) for selecting k. Though this higher order HMM learning does not exactly perform the ML learning, the implementation complexity will not rapidly increase with the orders of $\bar{\mathbf{x}}\mathbf{y}_{t-1}$. The details are referred to [22].

IV. INDEPENDENT MODEL AND SOURCE SEPARATION

A. Independent Model and Dependent Reduction

The vector representation $y_t = [y_t^{(1)}, \ldots, y_t^{(k)}]$ is much powerful than the simplest form in the HMM equation (4). However, it makes the model complexity increase rapidly with k.

One solution is to impose the independence assumption on the components of $y_t = [y_t^{(1)}, \ldots, y_t^{(k)}]$, i.e.,

$$p_{M_y}(\mathbf{y}) = \prod_{j=1}^{k} p_{M_y}\left(\mathbf{y}^{(j)}\right), \quad \text{so}$$
$$p_{M_y}(y_t \mid \mathbf{y}_{t-1}) = \prod_{j=1}^{k} p_{M_y}\left(y_t^{(j)} \,\middle|\, \mathbf{y}_{t-1}^{(j)}\right) \quad (33)$$

where $\mathbf{y}^{(j)}, \mathbf{y}_{t-1}^{(j)}$ denotes the j-th row of $\mathbf{y}, \mathbf{y}_{t-1}$, respectively. In this case, we call $p_M(\mathbf{x})$ by (5) *independent model*. Together with (33), the constraint $p_M(\mathbf{x}) = p_{M_x}(\mathbf{x})$ will make it more likely to find a unique specification of $p_{M_{x|y}}(\mathbf{x} \mid \mathbf{y})$, $p_{M_y}(\mathbf{y})$ for recovering the source in distribution.

We can also let the Yang model based $p_M(\mathbf{y})$ by (5) to match $p_{M_y}(\mathbf{y})$ by (33), which means that $p_{M_{y|x}}$ maps observations into their representations with dependence among components reduced. That is, we get the extension of *dependence reduction* [26] to temporal models. Particularly, we get the extension of ICA to temporal models when

$$p_M(\mathbf{y}) = \prod_{j=1}^{k} p_M\left(\mathbf{y}^{(j)}\right). \quad (34)$$

We call it *temporal ICA or shortly TICA*. The TICA includes the conventional ICA as a special case that each series $\mathbf{y}^{(j)}$ consists of i.i.d. samples.

To get TICA, it is necessary to impose the assumption

$$p_{M_{y|x}}(\mathbf{y} \mid \mathbf{x}) = \prod_{j=1}^{k} p_{M_{y|x}}\left(\mathbf{y}^{(j)} \,\middle|\, \mathbf{x}\right)$$
$$p_{M_{y|x}}(y_t \mid \bar{x}_t, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) = \prod_{j=1}^{k} p_{M_{y|x}}\left(y_t^{(j)} \,\middle|\, \bar{x}_t, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}^{(j)}\right). \quad (35)$$

A particular example is that $p_{M_{y|x}}$ is a delta density. Moreover, we also consider that $p_{M_x}(\mathbf{x})$ is generated from the source $p_{M_y}(\mathbf{y})$ via a delta density $p_{M_{x|y}}$ too, that is, we have

$$p_{M_{y|x}}(\mathbf{y} \mid \mathbf{x}) = \delta(\mathbf{y} - f(\mathbf{x}))$$
$$p_{M_{x|y}}(\mathbf{x} \mid \mathbf{y}) = \delta(\mathbf{x} - g(\mathbf{y})) \quad (36)$$

where $f(\cdot)$ and $g(\cdot)$ are general notations for deterministic mappings from $\mathbf{x} \to \mathbf{y}$ and $\mathbf{y} \to \mathbf{x}$, respectively, and with the causal assumption equation (13) they are recursively implemented by

$$y_t = f(x_t, \bar{\mathbf{x}}\mathbf{y}_{t-1}, \theta_{y|x}), \quad x_t = g(y_t, \bar{\mathbf{x}}\mathbf{y}_{t-1}, \theta_{x|y}). \quad (37)$$

In these cases, we can set up a deep connection between TICA and BSS by the following theorems.

Theorem 2: Given $\mathbf{x} = g(\mathbf{y})$, when $f \circ g(\mathbf{y}) = f(g(\mathbf{y}))$ is invertible and the resulted $p_M(\mathbf{y})$ is not Gaussian, the mapping $\mathbf{y} = f(\mathbf{x})$ that satisfies (34) leads us to

$$f \circ g(\mathbf{y}) = \Pi \left[h_1^T\left(\mathbf{y}^{(1)}\right), \ldots, h_k^T\left(\mathbf{y}^{(k)}\right)\right]^T$$
$$h_j(\cdot) \text{ is a function on } \mathbf{y}^{(j)} \quad \text{and}$$
$$\Pi \text{ is a permutation matrix.} \quad (38)$$

Moreover, when $f(\cdot)$, $g(\cdot)$ are both linear, we have simply $h_j(\mathbf{y}^{(j)}) = b_j\mathbf{y}^{(j)} + c_j$, where b_j, c_j are constants.

Proof: From (5) we get $p_{M_x}(\mathbf{x}) = p_M(\mathbf{x}) = \int \delta(\mathbf{x} - g(\mathbf{y}))p_{M_y}(\mathbf{y})\,d\mathbf{y}$ and $p_M(\mathbf{y}) = \int \delta(\mathbf{y} - f(\mathbf{x}))p_{M_x}(\mathbf{x})\,d\mathbf{x} = \int \delta(\mathbf{y} - f \circ g(\mathbf{y}'))p_{M_y}(\mathbf{y}')\,d\mathbf{y}'$. Moreover, if $f \circ g$ is invertible, with $\mathcal{U} = f \circ g(\mathbf{y}')$ we get $p_M(\mathbf{y}) = \int \delta(\mathbf{y} - \mathcal{U})p_{M_y}((f \circ g)^{-1}\mathcal{U})|D(\mathcal{U})|^{-1}\,d\mathcal{U} = p_{M_y}((f \circ g)^{-1}\mathbf{y})|D(\mathbf{y})|^{-1}$, where $D(\mathbf{y})$ is the Jacobian matrix of $f \circ g(\mathbf{y})$ with respect to $\mathbf{y}$. Moreover, when (33) is realized, we have $\prod_{j=1}^k p_M(\mathbf{y}^{(j)}) = |D(\mathbf{y})|^{-1}p_{M_y}(f \circ g(\mathbf{y}))$, which means that $D(\mathbf{y})$ should also be factorable with respect to $\mathbf{y}^{(j)}$ and thus $f \circ g(\mathbf{y}) = A[h_1^T(\mathbf{y}^{(1)}), \ldots, h_k^T(\mathbf{y}^{(k)})]^T$ with A being a $k \times k$ constant matrix. It further follows that A must be a permutation matrix Π since $p_M(\mathbf{y})$ is non-Gaussian. When $f(\cdot)$, $g(\cdot)$ are both linear, $D(\mathbf{y})$ should be a constant and thus $h_j(\mathbf{y}^{(j)}) = b_j\mathbf{y}^{(j)} + c_j$. □

Theorem 3: Given $\mathbf{x} = g(\mathbf{y})$, when $f \circ g(\mathbf{y}) = f(g(\mathbf{y}))$ is invertible and the resulted $p_M(\mathbf{y})$ is Gaussian, the mapping $\mathbf{y} = f(\mathbf{x})$ that satisfies (34) leads us to either (a) $f \circ g(\mathbf{y}) = dg[b_j]\Phi dg[d_j^{-1/2}]\mathbf{y} + c$ when the resulted $\mathbf{y}$ consists of i.i.d. samples with each sample y_t having the variance matrix $dg[d_j]$, or (b) $f \circ g(\mathbf{y}) = dg[b_j]\Pi\mathbf{y} + c$ when the samples of $\mathbf{y}$ are not i.i.d., where Φ is an orthogonal matrix, Π is a permutation matrix, c is a constant vector, and $b_j \neq 0$ is a constant. Moreover, $dg[r_j]$ denotes a diagonal matrix with its jth diagonal element being r_j.

Proof: Only when p_{M_y} is Gaussian and $f \circ g(\mathbf{y})$ is the form $A\mathbf{y} + c$ with A being a $k \times k$ matrix, the resulted $p_M(\mathbf{y})$ becomes Gaussian. When the samples of $\mathbf{y}$ are i.i.d., we have $p_M(\mathbf{y}) = \prod_{t=T}^k p_M(y_t)$ and $p_M(y_t) = \prod_{j=1}^k p_M(y_t^{(j)})$. For each t, $p_M(y_t)$ is invariant for a transform $dg[b_j]\Phi dg[d_j^{-1/2}]y_t + c$, where Φ is an orthogonal matrix and c is constant. Because the samples are i.i.d., the $dg[b_j]\Phi$ and c are same for all the values of t, thus we have $A = dg[b_j]\Phi dg[d_j^{-1/2}]$. When the samples of $\mathbf{y}$ are independent but not from an identical distribution, though $p_M(y_t)$ is still invariant for an orthogonal transform, the orthogonal matrix Φ generally varies with t, except the special case that $\Phi = \Pi$ is a permutation matrix. Finally, when the samples of $\mathbf{y}$ are serially correlated, by stacking all the columns of $\mathbf{y}$ into a big vector $\mathrm{vec}[\mathbf{y}]$, we have that $p_M(\mathrm{vec}[\mathbf{y}])$ is Gaussian with a nondiagonal covariance matrix, from which we know that $p_M(\mathrm{vec}[\mathbf{y}])$ is invariant only for a permutation among the elements of $\mathrm{vec}[\mathbf{y}]$. Writing $\mathrm{vec}[\mathbf{y}]$ back we find that a permutation is allowed only among the rows of $\mathbf{y}$ in order to keep the serial order for different t. Also, there are always unknown factors due to the form $\prod_{j=1}^k p_M(\mathbf{y}^{(j)})$. Thus, we have again $A = dg[b_j]\Pi$. □

From Theorems 2 & 3, we have the following conclusions.

Conclusion 1: If the observation is generated from a source via a unknown nonlinear system $g(\mathbf{y})$, according to Theorem 2, the TICA mapping $\mathbf{y} = f(\mathbf{x})$ will result in (38) which has de-coupled the "cross talks" between channels of the source but failed in preserving the waveforms of the source. In this case, no matter p_{M_y} is Gaussian or not, $p_M(\mathbf{y})$ is non-Gaussian. Furthermore, if we know the nonlinear function form of $g(\mathbf{y})$ but with a set of unknown parameters, and if we are able to design the function form of $f(\mathbf{x})$ with a set $\theta_{y|x}$ of unknown parameters such that there are specifications of $\theta_{y|x}$ that turns $f \circ g(\mathbf{y})$ into an invertible constant matrix while there is no specification that leads $f \circ g(\mathbf{y})$ to the form of (38) with nonlinear $h_j(\cdot)$, then it follows from Theorems 2 and 3 that the TICA mapping $\mathbf{y} = f(\mathbf{x})$ can recover the waveforms of either any non-Gaussian sources or those Gaussian sources with samples being not i.i.d.

Conclusion 2: Given that the observation is generated from a source via a unknown linear system $g(\mathbf{y})$ and we also use a linear mapping $f(\mathbf{x})$ with a set $\theta_{y|x}$ of unknown parameters to recover the source. In this case, $p_M(\mathbf{y})$ is Gaussian only when the source p_{M_y} is Gaussian. If there are specifications of $\theta_{y|x}$ that turns $f \circ g(\mathbf{y})$ into an invertible matrix, it follows from Theorems 2 and 3 that the TICA mapping by $\mathbf{y} = f(\mathbf{x})$ can recover the waveforms of either any non-Gaussian sources or those Gaussian sources with samples being not i.i.d.

The above conclusions can be regarded as further developments of the existing conditions on ICA for BSS [20] in the cases of i.i.d. samples via the invertible linear system equation (3). The conclusion 1 is also a further development on the nonlinear ICA condition for BSS from a specific post-nonlinear structure [19] to general nonlinear structures.

For the cases of non i.i.d. samples via linear or nonlinear time-delayed systems, it is not necessary to request that the system is invertible as long as the system is "information preserved" such that $f \circ g(\mathbf{y}) = f(g(\mathbf{y}))$ is invertible, e.g., for the observations from (3), the condition for TICA performs BSS can be relaxed to the cases that A is a full rank $d \times k, d > k$ matrix. Moreover, we can find that Gaussian sources of non i.i.d. samples via (3) are still separable by TICA. Even interestingly, we have the following.

Conclusion 3: Gaussian sources of i.i.d. samples via a linear time delay system $x_t = \sum_{r=0}^p A_r y_{t-r}$ are also separable by TICA.

Since the resulted $p_M(\mathbf{y})$ via this delay system and a linear f will be a Gaussian that is not i.i.d. in serial samples. We can understand this point more clearly by equivalently rewriting the delay system into $x_t = A_b y_t^b$ in the same form as (3), with $A_b = [A_0, \ldots, A_p]$. Now, the enlarged source $y_t^b = [y_t^T, \ldots, y_{t-p}^T]^T$ is no longer i.i.d. because both y_t^b and y_{t-1}^b consist of y_t in different locations.

B. Temporal ICA for Real BSS: Without and with Noise

In the rest of this paper, the notation $\mathbf{y}_\tau$ is understood as a vector that is obtained by stacking up each of past sample vectors $y_\tau, y_{\tau-1}, \ldots, y_{\tau-p}$, and the notation $\mathbf{x}_\tau$ is understood similarly. The dimensions of $\mathbf{x}_\tau$ and $\mathbf{y}_\tau$ can be either the same or different.

We consider $p_{M_y}(y_t \,|\, \mathbf{y}_{t-1})$ in (33) to be an independent density in a general case. We let each $p_{M_y}(y_t^{(j)} \,|\, \mathbf{y}_{t-1}^{(j)})$ given by $p_G(u \,|\, v)$ in (28), with $u = y_t^{(j)}$ and $v = \mathbf{y}_{t-1}^{(j)}$ and all the parameters denoted by a set $\theta_y^{(j)}$. Moreover, we denote $p_{M_y}(y_t \,|\, \mathbf{y}_{t-1}) = p(y_t \,|\, \mathbf{y}_{t-1}, \theta_y)$ with $\theta_y = \{\theta_y^{(j)}\}$.

We consider a deterministic F-architecture

$$\begin{aligned} y_t &= f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x}), \quad \text{e.g.} \\ y_t &= K\mathbf{x}_t + H\mathbf{y}_{t-1} + c_\zeta. \end{aligned} \tag{39}$$

Thus, $p_{M_{y\,|\,x}}(y_t \,|\, \bar{\mathbf{x}}_t, \mathbf{y}_{t-1}) = \delta(y_t - f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x}))$. In this case, we can easily get $p_{M_{y\,|\,x}}(\mathbf{y}_{t-1} \,|\, \bar{\mathbf{x}}_{t-1})$ by (14) and put it into (16), resulting in

$$\begin{aligned} KL_t^{(1)} &= \ln \delta(y_t - f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x})) \\ KL_t^{(3)} &= \ln p(f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x}) \,|\, \mathbf{y}_{t-1}, \theta_y) \end{aligned} \tag{40}$$

where the past samples are recursively obtained by (39).

From the fact that $\delta(u - F(v,\psi)) = \lim_{V_u \to 0}(1/\delta V_u)$, where δV_u is the differential volume of the manifold $u = F(v,\psi)$ at v, we have $\delta V_u = c|D_F D_F^T|^{0.5}\delta V_v$, $D_F = \partial F(v,\psi)/\partial v^T$, where δV_v is the differential volume in the space of v, and both $c, \delta V_v$ are irrelevant to parameter ψ and thus can be ignored. Let $u = \mathbf{y}_t, \xi = \bar{\mathbf{x}}_t$ and $v = [u^T, \xi^T]^T$, also let $F(v,\psi)$ to consist of (39) plus the identity mapping from $\bar{\mathbf{y}}_{t-1}$ to $\bar{\mathbf{y}}_{t-1}$ itself. After some derivations we can get $|D_F D_F^T| = |D_f D_f^T|$, $D_f = \partial f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x})/\partial \bar{\mathbf{x}}_t^T$ and get $KL_t^{(1)} = -0.5 \ln |D_f D_f^T|$ after ignoring terms irrelevant to parameter $\theta_{y\,|\,x}$. Putting it into (40), from (16) we summarize as follows.

We implement TICA to recover y_t by (39) from x_t recursively from $t = 1$ to T. At each t, we update the parameters $\theta_{y\,|\,x}, \theta_y$ by maximizing

$$\begin{aligned} -KL_t(\theta_{y\,|\,x}, \theta_y) &= 0.5 \ln \left| D_f D_f^T \right| \\ &\quad + \ln p(f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x}) \,|\, \mathbf{y}_{t-1}, \theta_y) \\ D_f &= \partial f(\bar{\mathbf{x}}_t, \mathbf{y}_{t-1}, \theta_{y\,|\,x})/\partial \bar{\mathbf{x}}_t^T \\ -KL_t(K, H, \theta_y) &= 0.5 \ln |KK^T| + \ln p(K\bar{\mathbf{x}}_t + H\mathbf{y}_{t-1} \\ &\quad + c_\zeta \,|\, \mathbf{y}_{t-1}, \theta_y). \end{aligned} \tag{41}$$

Specifically, K, H, θ_y can be adaptively updated by

$$\begin{aligned} K_{t+1} &= K_t + \lambda[I + \phi(y_t)(K_t\bar{\mathbf{x}}_t)^T]K_t \\ H_{t+1} &= H_t + \lambda\phi(y_t)\mathbf{y}_{t-1}^T, \quad c_{\zeta,t+1} = c_{\zeta,t+1} + \lambda\phi(y_t) \\ \theta_{t+1,y}^{(j)} &= \theta_{t,y}^{(j)} + \lambda \frac{\partial \ln p\left(y_t^{(j)} \,\middle|\, \mathbf{y}_{t-1}, \theta_{t,y}^{(j)}\right)}{\partial \theta_{t,y}^{(j)}} \\ \phi(y_t) &= \left[\frac{\partial \ln p\left(y_t^{(1)} \,\middle|\, \mathbf{y}_{t-1}, \theta_{t,y}^{(1)}\right)}{\partial y_t^{(1)}} \right. \\ &\quad \left. \ldots, \frac{\partial \ln p\left(y_t^{(k)} \,\middle|\, \mathbf{y}_{t-1}, \theta_{t,y}^{(k)}\right)}{\partial y_t^{(k)}} \right]^t \end{aligned} \tag{42}$$

where $\lambda > 0$ is stepsize. Also, $\theta_y^{(j)}$ can be updated by the EM algorithm-like adaptive versions proposed in [25]. Particularly, $\phi(y_t)$ becomes linear for Gaussian source $p_{M_y}(y_t \,|\, \mathbf{y}_{t-1})$.

We can extend this study to the case $x_t = Ay_t + B\mathbf{y}_{t-1} + c_e + e_t$ with Gaussian noise e_t, that is,

$$\begin{aligned} p_{M_{x\,|\,y}}(x_t \,|\, y_t, \mathbf{y}_{t-1}) &= G\left(e_t, 0, \sigma_e^2 I_d\right) \\ e_t &= x_t - (Ay_t + B\mathbf{y}_{t-1} + c_e). \end{aligned} \tag{43}$$

In this case, from (14) we get again the same $KL_t^{(1)}$, $KL_t^{(3)}$ as in (40). Moreover, we get $KL_t^{(2)} = \ln G(e_t, 0, \sigma_e^2 I_d)$. That is, from (14) we now update $\theta = \{K, H, \theta_y, A, B, c_\zeta, c_e, \sigma_e^2\}$ by maximizing

$$\begin{aligned} -KL_t(\theta) &= -KL_t(K, H, \theta_y) \\ &\quad + \ln G\left(x_t, Ay_t + B\mathbf{y}_{t-1} + c_e, \sigma_e^2\right) \end{aligned} \tag{44}$$

with $y_t = K\bar{\mathbf{x}}_t + H\mathbf{y}_{t-1} + c_\zeta$. In this case, we recover y_t by considering both TICA and ML modeling of x_t by (43). Corresponding to (42), we get y_t by (39) and

$$\begin{aligned} g_t &= \phi(y_t) + \sigma_{e,t}^{-2} A^T e_t \\ K_{t+1} &= K_t + \lambda[I + g_t(K_t\bar{\mathbf{x}}_t)^T]K_t, \quad \text{or} \\ g_t &= \sigma_{e,t}^2 \phi(y_t) + A^T e_t \\ K_{t+1} &= K_t + \lambda\left[\sigma_{e,t}^2 I + g_t(K_t\bar{\mathbf{x}}_t)^T\right] K_t \\ H_{t+1} &= H_t + \lambda g_t \mathbf{y}_{t-1}^T, \quad c_{\zeta,t+1} = c_{\zeta,t+1} + \lambda g_t \\ A_{t+1} &= A_t + \lambda e_t y_t^T, \quad B_{t+1} = B_t + \lambda e_t \mathbf{y}_{t-1}^T \\ c_{e,t+1} &= c_{e,t} + \lambda e_t \\ \sigma_{e,t+1}^2 &= (1 - \lambda)\sigma_{e,t}^2 + \lambda d^{-1}\|e_t\|^2 \end{aligned} \tag{45}$$

with θ_y updated by (42) still.

Together with the parameter learning, from (17) we also get

$$\begin{aligned} H(k) = \sum_{t=1}^{T} \Bigg[& 0.5\left(d \ln \sigma_{e,t}^2 - \frac{\|e_t\|^2}{\sigma_{e,t}^2}\right) \\ & + \ln p(K\bar{\mathbf{x}}_t + H\mathbf{y}_{t-1} + c_\zeta \,|\, \mathbf{y}_{t-1}, \theta_y) \Bigg] \end{aligned} \tag{46}$$

in a recursive way for model selection by (10).

C. Temporal Factor Analysis for Real BSS

Via the δ density, the deterministic F-architecture equation (39) avoids the integrals over $y_t, \mathbf{y}_{t-1}$, without using the approximation given in Section II-D.

Here, we extend δ density to Gaussians. We still consider (43) but with

$$\begin{aligned} p_{M_{y\,|\,x}}(y_t \,|\, \bar{\mathbf{x}}_t, \mathbf{y}_{t-1}) &= G(y_t, K\mathbf{x}_t + H\mathbf{y}_{t-1} + c_\zeta, \Sigma_\zeta) \\ p_{M_y}(y_t \,|\, \mathbf{y}_{t-1}) &= G(y_t, E\mathbf{y}_{t-1} + c_\varepsilon, \Lambda_\varepsilon) \end{aligned} \tag{47}$$

where Λ_ε is diagonal. When K, H, Σ_ζ are free or equivalently $p_{M_{y\,|\,x}}$ is free, from (47) and (15) we can get an extension of factor analysis [1] to time series and thus we call it *temporal factor analysis (TFA)*. We can also regard it as a hidden independent AR factor model (HAR) because the observations are obtained via a probabilistic time delay system $p_{M_{x\,|\,y}}$ from

k un-correlated auto-regressive (AR) time series as factors described by p_{M_y}. In this case, (9) becomes ML learning on $p_M(\mathbf{x})$ modeled by $p_{M_{x|y}}$ and p_{M_y}.

Moreover, we are able to get a detailed implementing algorithm for (29), with $KL_t^{(1)}$, $KL_t^{(2)}$, $KL_t^{(3)}$ obtained by analytically solving the integrals in (14). We leave the details elsewhere.

Here, we consider its implementation in the first-order approximation by (19) and (20) with $c_T = c_y = c_\sigma = 0$. That is, we have $\hat{\mathbf{y}}_{t-1}$ recursively obtained by the linear regression in (39) and $KL_t^{(1)}$, $KL_t^{(2)}$, $KL_t^{(3)}$ given by H_g, L_{gg}, L_g in Table II. After ignoring constants, we get the following cost to be minimized

$$\begin{aligned} 2KL_t(\theta) &= \ln\frac{\sigma_e^{2d}|\Lambda_\varepsilon|}{|\Sigma_\zeta|} + \sigma_e^{-2}(\mathrm{Tr}[A\Sigma_\zeta A^T] + \|e_t\|^2) \\ &\quad + \mathrm{Tr}\left[\Sigma_\zeta\Lambda_\varepsilon^{-1}\right] + \varepsilon_t^T\Lambda_\varepsilon^{-1}\varepsilon_t \\ \varepsilon_t &= \hat{y}_t - \hat{y}_t^-, \quad y_t^- = E\hat{\mathbf{y}}_{t-1} + c_\varepsilon \\ \hat{y}_t &= K\mathbf{x}_t + H\hat{\mathbf{y}}_{t-1} + c_\zeta \end{aligned} \tag{48}$$

where e_t is given by (43) which represents the reconstruction error to the observation, and ε_t is the difference between *a priori* state estimate $\hat{y}_t^-$ and *a posteriori* state estimate $\hat{y}_t$.

Moreover, based on the gradient directions given in Table II, we can get a detailed algorithm in Table III, by using either directly gradient descent or its modification obtained by multiplying a positive definite matrix (e.g., $\Lambda_\varepsilon\nabla_{\Lambda_\varepsilon}L_g\Lambda_\varepsilon$ and $\Lambda_\varepsilon\nabla_\theta L_g$) to the gradient descent direction.

From (17) we also get the model selection

$$\begin{aligned} -H(k) &= \sum_{t=1}^{T}[KL_t(\theta^*) + 0.5\ln|\Sigma_{\zeta,t}|], \quad \text{or} \\ -H(k) &\approx 0.5\left\{kT + \sum_{t=1}^{T}\left[\ln|\Lambda_{\varepsilon,t}| + d\ln\sigma_{e,t}^2\right]\right\} \end{aligned} \tag{49}$$

where d, k is the dimension of x_t, y_t, respectively. The approximation comes from roughly regarding that $(1/T)\sum_{t=1}^{T}\mathrm{Tr}[\Sigma_e^{-1}\{A_t\Sigma_{\zeta,t}A_t^T + e_te_t^T\}] \approx \mathrm{Tr}[I_n] = n$ and $(1/T)\sum_{t=1}^{T}\mathrm{Tr}[\Lambda_{\varepsilon,t}^{-1}\,\mathrm{diag}[\Sigma_{\zeta,t} + \varepsilon_t\varepsilon_t^T]] = \mathrm{Tr}[I_k] = k$.

D. Independent Higher Order HMM for Binary BSS

We consider the cases that the state is a binary vector $y_=[y_t^{(1)}, \ldots, y_t^{(k)}]^T$. Still, we consider (43) but now both $p_{M_{y|x}}(y_t \,|\, \bar{\mathbf{x}}_t, \mathbf{y}_{t-1}), p_{M_y}(y_t \,|\, \mathbf{y}_{t-1})$ are given by $p_B(u \,|\, \nu)$ in (27) with $u = y_t$ as well as $\nu = \hat{y}_t$ for $p_{M_{y|x}}$ and $\nu = \hat{y}_t^-$ for p_{M_y}, respectively. This case can be regarded as a variant of higher order HMM for time series modeling with independent binary codes as states. The recovery of these binary codes from $\bar{x}_t$ can be regarded as separating k independent binary sources from the observation $\bar{x}_t$.

Since y_t is binary, the integrals in (14) over $y_t, \mathbf{y}_{t-1}$ become summations. We are able to get a detailed implementing algorithm for (29), with $KL_t^{(1)}, KL_t^{(2)}, KL_t^{(3)}$ obtained without the approximation given in Section II-D. We leave the details elsewhere.

Here, we still consider the first-order approximation by (19) and (20) with $c_T = c_y = c_\sigma = 0$, for fast implementation. That is, we have $\hat{\mathbf{y}}_{t-1}$ recursively obtained by (20) and $KL_t^{(1)}$, $KL_t^{(2)}$, $KL_t^{(3)}$ are given by H_b, L_b, L_{gb} in Table I. After ignoring constants, we get the following cost to be minimized:

$$\begin{aligned} KL_t(\theta) &= \sum_{j=1}^{k}\left[f\left(\hat{y}_t^{(j)}\right)\ln\frac{f\left(\hat{y}_t^{(j)}\right)}{f\left(y_t^{-(j)}\right)}\right. \\ &\quad \left. + \left(1 - f\left(\hat{y}_t^{(j)}\right)\right)\ln\frac{1 - f\left(\hat{y}_t^{(j)}\right)}{1 - f\left(y_t^{-(j)}\right)}\right] \\ &\quad + 0.5\sigma_e^{-2}\{\mathrm{Tr}[A\Lambda_f A^T] + \|e_t\|^2\} + 0.5\,d\ln\sigma_e^2 \\ \Lambda_f &= dg\left[f\left(\hat{y}_t^{(j)}\right)\left(1 - f\left(\hat{y}_t^{(j)}\right)\right)\right] \end{aligned} \tag{50}$$

with $\hat{y}_t, \hat{y}_t^-, \hat{x}_t$ given by (48). The minimization of the above last term alone can be regarded as an extension of the least square reconstruction based nonlinear PCA firstly proposed in [32] where an adaptive algorithm is proposed and the separation property by sigmoid nonlinearity is firstly discovered, which has been later applied to ICA and BSS with success [14].

We can also implement (29) by a stochastic sampling algorithm as given in [23].

Moreover, we do model selection by

$$\begin{aligned} H(k) &= \sum_{t=1}^{T}\left[H_t(f) - 0.5\sigma_{e,t}^{-2}\left\{\mathrm{Tr}\left[A_t\Lambda_{f,t}A_t^T\right] + \|e_t\|^2\right\}\right. \\ &\quad \left. - 0.5\,d\ln\sigma_{e,t}^2\right] \\ H_t(f) &= \sum_{j=1}^{k}\left[f\left(\hat{y}_t^{(j)}\right)\ln f\left(y_t^{-(j)}\right)\right. \\ &\quad \left. + \left(1 - f\left(\hat{y}_t^{(j)}\right)\right)\ln\left(1 - f\left(y_t^{-(j)}\right)\right)\right], \quad \text{or} \\ H(k) &\approx \sum_{t=1}^{T}\left[H_t(f) - 0.5\,d\ln\sigma_{e,t}^2\right] \\ &\text{when } \sigma_{e,t}^2 = \tfrac{1}{T}\textstyle\sum_{t=1}^{T}\left\{\mathrm{Tr}\left[A\Lambda_{f,t}A_t^T\right] + \|e_t\|^2\right\} = n. \end{aligned} \tag{51}$$

V. Experiments

We only show some experiments on using the temporal ICA algorithm (42), shortly denoted as TICA, for solving the real BSS problems. To illustrate the advantages of taking temporal relation in consideration, the experiments are made in comparison with the nontemporal counterpart of (42), called the learned parametric mixture based ICA [27], shortly denoted as LPM-ICA, which is equivalent to a degenerated case of (42) with $H = 0, c_\zeta = 0$.

Two data sets have been tested. The Data-Set-1 is generated according to the model equation (1) with (a) $B = \mathrm{diag}[0.8, -0.7, 0.9]$ and ε_t being i.i.d. from a Gaussian with zero mean and variance 0.04; (b) x_t is obtained via the mixing matrix A with the noise e_t being Gaussian of zero mean and variance $0.01I_3$. The Data-Set-2 is generated from i.i.d Gaussian sources y_t with zero mean and covariance $\mathrm{diag}[0.04, 0.0625, 0.25]$ by a time-delay system $x_t = Ay_t + By_{t-1} + e_t$ with e_t being Gaussian of zero mean

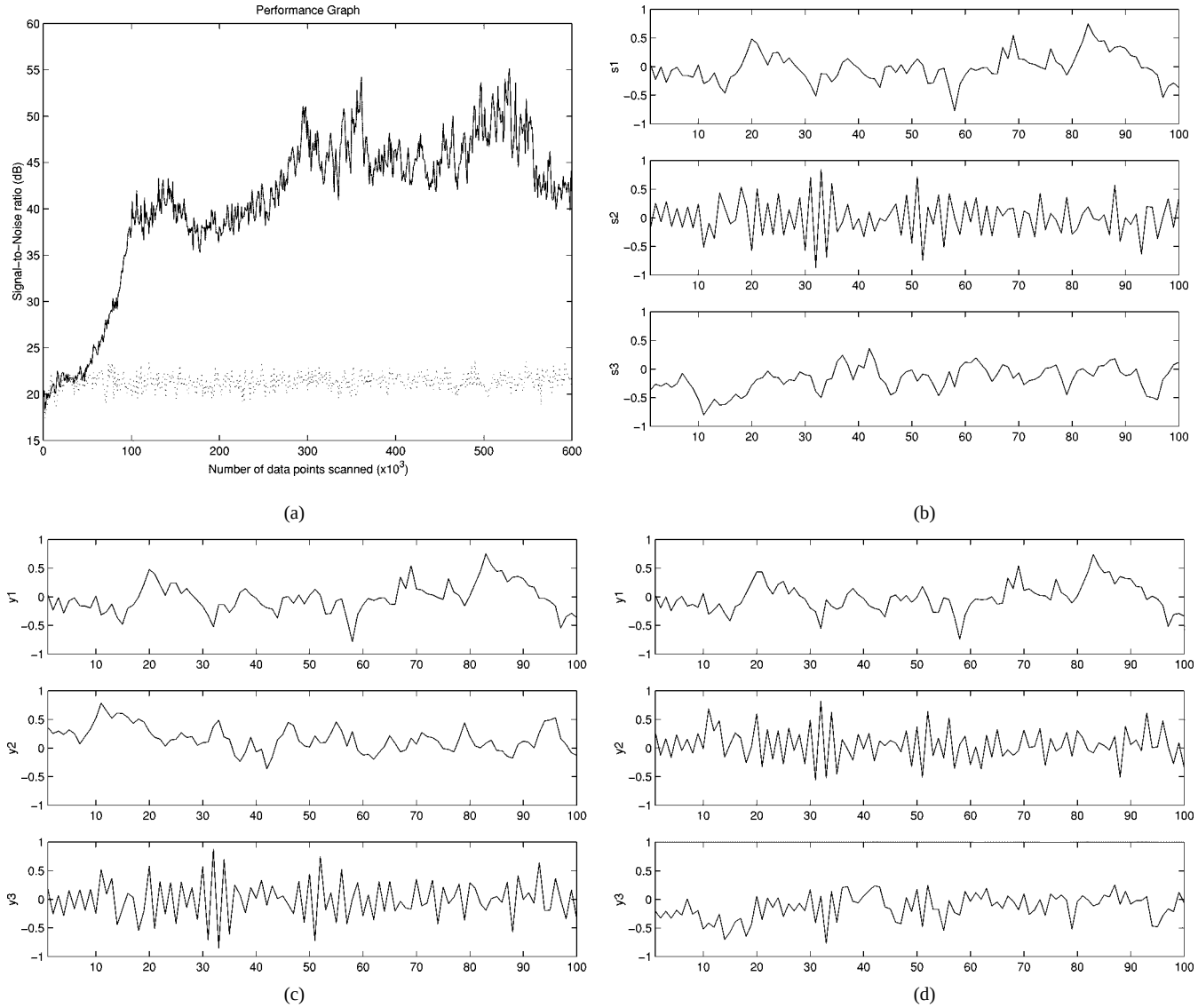

Fig. 1. Comparison of TICA and LPM-ICA on the Data-Set-1. (a) Changes of the average SNR as time t goes, with the solid line for TICA and the dotted line for LPM-ICA. (b) Segment of each of the three sources. (c) Recovered source segments by TICA. Specifically, the first row recovers the first row in (b), the second row recovers the third row in (b) up to a negative sign -1, and the third row recover the second row in (b), also up to a negative sign -1. (d) Recovered source segments by LPM-ICA, the first, second, and third rows recover the first, second, and third rows in (b), respectively. For the second row, there is also a change of negative sign -1.

and variance $0.01I_3$, where for both the data sets, we arbitrarily set

$$A = \begin{pmatrix} 1.6016 & 0.1890 & -0.1712 \\ 0.1748 & -2.3698 & -0.7649 \\ -0.3020 & -0.6645 & 0.1454 \end{pmatrix}$$
$$B = \begin{pmatrix} 1.2041 & 0.2742 & 0.8284 \\ -0.6920 & 1.1862 & 0.2091 \\ -0.7020 & 0.6810 & 0.2944 \end{pmatrix}. \quad (52)$$

The Data-Set-1 is designed for the BSS problems on a noisy instantaneous mixing model with sources being not i.i.d. but having serial dependence. The Data-Set-2 is designed for the BSS problems on a convolution mixing model with i.i.d. source samples which can even be i.i.d. Gaussians.

The learning is *real time*, i.e., it is made at each t as each sample comes, and each sample is used only once without any repeated or periodical sampling.

To evaluate the results, we need a measure for the error between an original source and its recovered counterpart. The measure should be invariant to scaling and permutation. The permutation issue is simply handled by manually pairing an original source and its recovered counterpart. To handle the scaling issue, we normalize the j-th original source and its recovered counterpart into the range $[-1, 1]$ and calculate the corresponding mean square error MSE_j. Then, we use the average signal-to-noise ratio $\mathrm{SNR} = (\mathrm{SNR}_1 + \mathrm{SNR}_2 + \mathrm{SNR}_3)/3$ as a measure for the source recovering performance such that the larger the SNR is, the better the performance is, where $\mathrm{SNR}_j = 10\log_{10}(\mathrm{Var}_j/\mathrm{MSE}_j)$ with Var_j being the variance of the j-th normalized original source. Since Var_j is irrelevant to source recovering, we can further ignore it and simply use $\mathrm{SNR}_j = -10\log_{10}\mathrm{MSE}_j$.

In Fig. 1, the results on the Data-Set-1 demonstrate that TICA outperforms LPM-ICA considerably, which can be observed

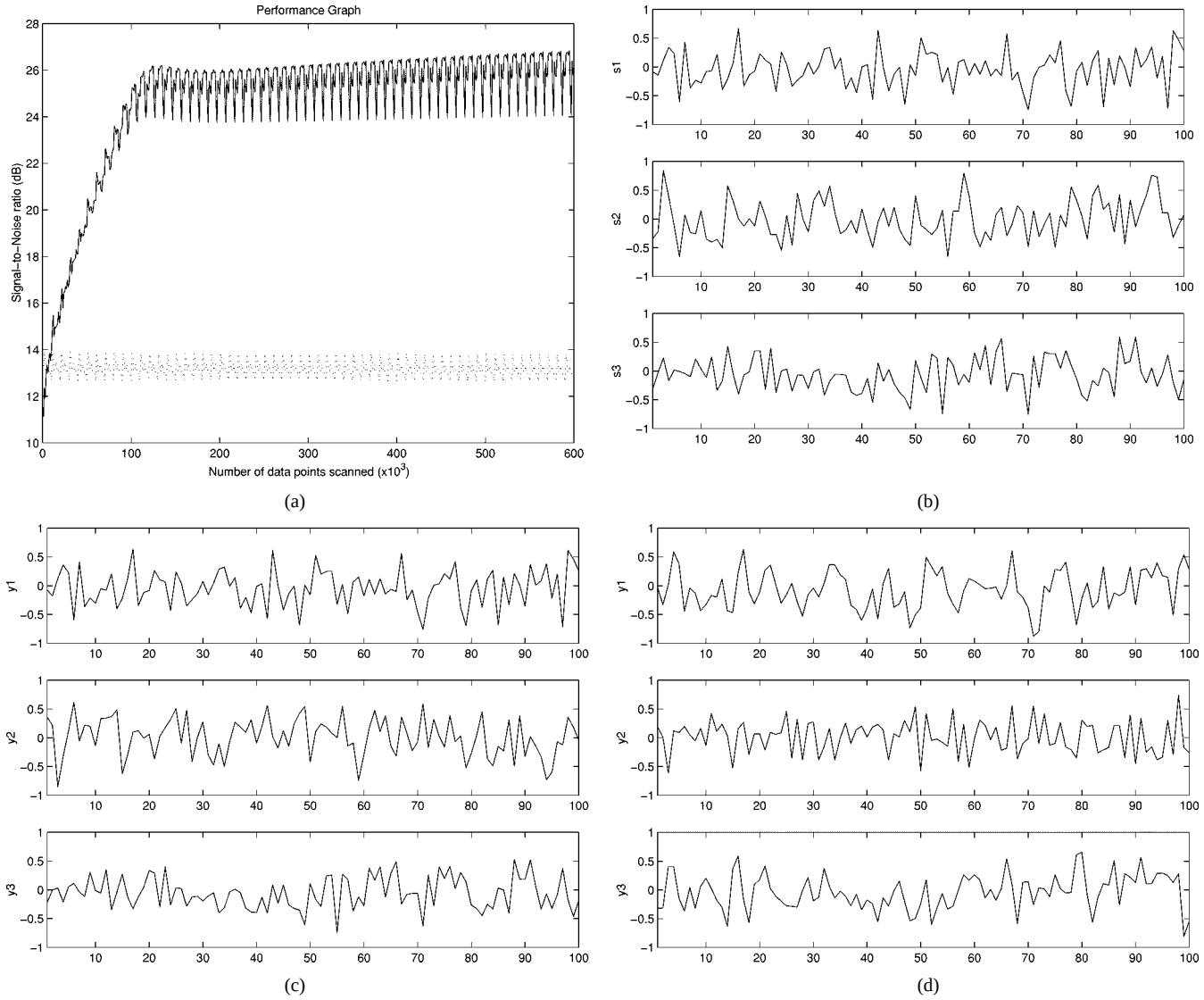

Fig. 2. Comparison of TICA and LPM-ICA on the Data-Set-2. (a) and (b) Same items as in Fig. 1. (c) and (d) Recovered source segments by TICA and LPM-ICA, respectively, and the first, second, and third rows recover the first, second, and third d rows in (b), respectively. Moreover, in (c) the second row recovers the second row in (b) with a change of sign -1.

from either the obtained SNR ratios or the recovered waveforms by ignoring the reversals due to a negative sign -1. Thus, it is really useful to take into consideration the serial relations among the sources.

In Fig. 2, ignoring the reversals due to a negative sign -1 we can observe that the results on the Data-Set-2 demonstrate that on a convolution mixing model TICA can still work well even on sources that are i.i.d. Gaussians, as predicted by conclusion 3 in Section IV-A. In contrast, it is well known that any i.i.d. Gaussians via an instantaneous mixing model are not separable [20]. Moreover, we also notice that LPM-ICA performs poorly since it takes no consideration on the contribution of the time delay part By_{t-1}.

VI. Conclusion

TBYY learning is a new general state space approach for modeling signal, which provides a new perspective not only for a unified understanding on Kalman filtering, HMM, ICA and BSS, but also a general guideline for various variants and further developments. Particularly, algorithms are developed for solving both real and binary BSS problems with temporal dependence and observation noise in consideration, and criteria are provided for selecting an appropriate number of sources. Moreover, theorems are given on the conditions for source separation by linear and nonlinear TICA, and it is shown that not only non-Gaussian but also Gaussian sources can also be separated by TICA when temporal dependence is explored. Some experiments have demonstrated that the TICA algorithm obtained from this framework outperform considerably an existing counterpart algorithm.

Acknowledgment

The author thanks Y. M. Cheung for experiments.

References

[1] T. W. Anderson and H. Rubin, "Statistical inference in factor analysis," in *Proc. Berkeley Symp. Math. Statist. Prob.*, vol. 5, 1956, pp. 111–150.

[2] S.-I. Amari, A. Cichocki, and H. Yang *et al.*, "A new learning algorithm for blind separation of sources," in *Advances in Neural Information Processing*, D. S. Touretzky *et al.*, Eds. Cambridge, MA: MIT Press, 1996, vol. 8, pp. 757–763.
[3] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind de-convolution," *Neural Comput.*, vol. 7, pp. 1129–1159, 1995.
[4] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York: Wiley, 1997.
[5] J. F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017–3030, 1996.
[6] A. Cichocki, S. C. Douglas, and S. Amari, "Robust techniques for independent component analysis (ICA) with noisy data," *Neurocomput.*, vol. 22, no. 1–3, pp. 113–129, 1998.
[7] P. Comon, "Independent component analysis—A new concept?," *Signal Process.*, vol. 36, pp. 287–314, 1994.
[8] L. Devroye *et al.*, *A Probability Theory of Pattern Recognition*. Berlin, Germany: Springer, 1996.
[9] *Neurocomput., Special Issue Independence Artif. Neural Networks*, , vol. 22, no. 1–3, 1998.
[10] M. Gaeta and J.-L. Lacounme, "Source separation without a priori knowledge: The maximum likelihood solution," in *Proc. Eur. Signal Processing Conf.*, 1990, pp. 621–624.
[11] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[12] "Invited special session on blind signal separation," in *Proc. 1997 Eur. Symp. Artificial Neural Networks*, Apr. 16–18, 1997, pp. 243–297.
[13] C. Jutten and J. Herault, "Independent component analysis versus principal component analysis," in *Proc. Eur. Signal Processing Conf.*, 1988, pp. 643–646.
[14] J. Karhunen and J. J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks*, vol. 7, pp. 113–127, 1994.
[15] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, London, U.K.: Chapman & Hall, 1983.
[16] J.-P. Nadal and N. Parga, "Nonlinear neurons in the low-noise limit: A factorial code maximizes information transfer," *Network*, vol. 5, pp. 565–581, 1994.
[17] E. Oja and A. Hyvarinen, "Blind signal separation by neural networks," in *Proc. Int. Conf. Neural Information Processing (ICONIP 96)*, Singapore, Sept. 24–27, 1996, pp. 7–14.
[18] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[19] A. Taleb and C. Jutten, "Nonlinearity source separation: The post-nonlinear mixtures," in *Proc. 1997 Eur. Symp. Artificial Neural Networks*, Apr. 16–18, 1997, pp. 279–284.
[20] L. Tong, Y. Inouye, and R. Liu, "Waveform-perseving blind estimation of multiple independent sources," *IEEE Trans. Signal Processing*, vol. 41, pp. 2461–2470, July 1993.
[21] L. Xu, "Bayesian Ying-Yang theory for empirical learning, regularization and model selection: General formulation, and data mining, unsupervised learning and Bayesian Ying-Yang theory," in *Proc. IJCNN99*, vol. 1, Washington, DC, July 1999, pp. 552–557.
[22] ——, "Temporal BYY learning and its applications to extended Kalman filtering, hidden Markov model, and sensor-motor integration," in *Proc. IJCNN99*, vol. 2, Washington, DC, July 1999, pp. 949–954.
[23] ——, "Temporal Bayesian Ying-Yang dependence reduction, blind source separation and principal independent components," in *Proc. IJCNN99*, vol. 2, Washington, DC, July 1999, pp. 1071–1076.
[24] ——, "RBF nets, mixture experts, and Bayesian Ying-Yang learning," *Neurocomput.*, vol. 19, no. 1–3, pp. 223–257, 1998.
[25] ——, "Bayesian Ying-Yang system and theory as a unified statistical learning approach: (V) Temporal modeling for perception and control," in *Proc. ICONIP'98*, vol. 2, Kitakyushu, Japan, Oct. 21–23, 1998, pp. 877–884.
[26] ——, "Bayesian Kullback Ying-Yang dependence reduction theory," *Neurocomput.*, vol. 22, no. 1–3, pp. 81–112, 1998.
[27] L. Xu, C. C. Cheung, and S.-I. Amari, "Learned parametric mixture based ICA algorithm," *Neurocomput.*, vol. 22, no. 1–3, pp. 69–80, 1998.
[28] L. Xu, "Bayesian Ying-Yang system and theory as a unified statistical learning approach: (I) Unsupervised and semi-unsupervised learning," in *Brain-Like Computing and Intelligent Information Systems*, S. Amari and N. Kassabov, Eds. Berlin, Germany: Springer-Verlag, 1997, pp. 241–274.
[29] ——, "Bayesian Ying-Yang learning based ICA models," in *Proc. 1997 IEEE Signal Processing Soc. Workshop*, Sept. 24–26, 1997, pp. 476–485.
[30] L. Xu, C. C. Cheung, J. Ruan, and S.-I. Amari, "Nonlinearity and separation capability: Further justification for the ICA algorithm with a learned mixture of parametric densities," in *Proc. 1997 Eur. Symp. Artificial Neural Networks*, Apr. 16–18, 1997, pp. 291–296.
[31] L. Xu *et al.*, "A unified learning scheme: Bayesian-Kullback YING-YANG machine," in *Advances in Neural Information Processing Systems*, D. S. Touretzky *et al.*, Eds. Cambridge, MA: MIT Press, 1996, vol. 8, pp. 444–450. A part of its preliminary version on *Proc. ICONIP95*, pp. 977–988, Oct. 30–Nov. 3, 1995.
[32] ——, "Least mean square error reconstruction for self-organizing neural-nets," *Neural Networks*, vol. 6, pp. 627–648, 1993. Its early version on *Proc. 1991 Int. Joint Conf. Neural Networks*, pp. 2363–2373, 1991.

Lei Xu (SM'94) received the Ph.D. degree from Tsinghua University, China, in 1986.

He is currently a Professor with the Department of Computer Science and Engineering, Chinese University of Hong Kong (CUHK), and the Director of the Peking University Joint Center on Intelligence Engineering, CUHK. He is also a Full Professor with Peking University and a Guest Full Professor with the University of Paisley, Frisley, U.K. He was a Postdoc with the Department of Mathematics, Peking University, in 1987, and became one of ten exceptionally promoted young Associate Professors of Peking University in 1988. From 1989 to 1993, he was a Postdoc or Senior Research Associate with several universities in Finland, Canada, and the United States, including Harvard University, Cambridge, MA, and Massachusetts Institute of Technology, Cambridge. He has published more than 200 academic papers and given more than ten keynote/invited/tutorial talks and served as program committee members and session chairs in international neural networks conferences including WCNN, ENNS-ICANN, ICONIP, IJCNN, and NNCM. Also, he was a program committee chair of ICONIP'96 and a general chair of IDEAL'98. He is a past president of Asian-Pacific Neural Networks Assembly. He is an associate editor for six international journals on neurocomputing including *Neural Networks*.

Dr. Xu has received several Chinese academic awards including the National Nature Science Award and the State Education Council FOK YING TUNG Award. He has received international awards including an 1995 International Neural Networks Society Leadership Award. He is an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS.