Advances on BYY Harmony Learning: Information Theoretic Perspective, Generalized Projection Geometry, and Independent Factor Autodetermination

Lei Xu, *Fellow, IEEE*

***Abstract*—The nature of Bayesian Ying–Yang harmony learning is reexamined from an information theoretic perspective. Not only its ability for model selection and regularization is explained with new insights, but also discussions are made on its relations and differences from the studies of minimum description length (MDL), Bayesian approach, the bit-back based MDL, Akaike information criterion (AIC), maximum likelihood, information geometry, Helmholtz machines, and variational approximation. Moreover, a generalized projection geometry is introduced for further understanding such a new mechanism. Furthermore, new algorithms are also developed for implementing Gaussian factor analysis (FA) and non-Gaussian factor analysis (NFA) such that selecting appropriate factors is automatically made during parameter learning.**

***Index Terms*—Automatic model selection, Bayesian, bit-back, Bayesian Ying–Yang (BYY) system, factor analysis, harmony learning, information theoretic, minimum description length, non-Gaussian factors, projection geometry.**

I. Introduction

The sprit of simultaneously building up two pathways, i.e., a bottom-up pathway for encoding an observed pattern into a representation space and a top-down pathway for decoding or reconstructing a pattern from an inner representation back to a pattern in the observation space, has been widely adopted in various studies of brain theory and neural networks. Typical examples include ART theory [10], Kawato's theory on cerebellum and motor control [26], Helmholtz machines and wake-sleep learning [12], [13], [17]. Moreover, the least mean square error reconstruction (LMSER) self-organizing learning proposed in 1991 [62] is also an effort that uses a bidirectional architecture for unsupervised learning. The basic sprit of the LMSER learning has been further developed into the Bayesian Ying–Yang (BYY) harmony learning [59], which is firstly proposed in 1995 and then systematically developed in past years [45]–[50], [52].

The BYY harmony learning formulates the two pathway sprit as shown in Fig. 1. This paper considers coordinately learning two complement representations of the joint distribution $p(x,y)$

$$p(x,y)=p(y|x)p(x), \quad q(x,y)=q(x|y)q(y) \tag{1}$$

basing on $p(x)$ that is estimated from a set of samples $\{\bar{x}_t\}_{t=1}^N$, while $p(y|x)$, $q(x|y)$ and $q(y)$ are unknowns but subject to certain prespecified structural constraints. The pair forms a so called BYY system [59], in a compliment to the famous Chinese ancient Ying–Yang philosophy. Interestingly, the decomposition of $p(x,y)$ coincides the Yang concept with the visible domain by $p(x)$ regarded as a Yang space and the forward pathway by $p(y|x)$ as a Yang pathway. Thus, $p(x,y)$ is called Yang machine. Similarly, $q(x,y)$ is called Ying machine with the invisible domain by $q(y)$ regarded as a Ying space and the backward pathway by $q(x|y)$ as a Ying path.

This BYY system can lead us to a number of existing major learning models as special cases from a unified perspective, including

- those so called predictive/forward models by

$$p(y)=\int p(y|x)p(x)dx. \tag{2}$$

One major type of examples is a deterministic mapping $y=Wx$ that performs either principal component analysis (PCA) for a Gaussian y or independent component analysis (ICA) for a non-Gaussian y, through making that $p(y)$ becomes maximum entropy [7], [16] or matches the following independent density [4]:

$$q(y)=\prod_{j=1}^{m} q\left(y^{(j)}\right) \tag{3}$$

- those so called generative/backward models by

$$q(x)=\int q(x|y)q(y)dy. \tag{4}$$

One type of examples corresponds to the cases with $y=1,\cdots,m$ and $q(x|y)$ being Gaussian. In these cases, (4) is a Gaussian mixture that is either directly used for density estimation via the maximum likelihood (ML) learning with the expectation and minimization (EM) algorithm [14], [30] or further simplified into the mean square error (MSE) clustering and the elliptic clustering [48], [50], [57]. One other type corresponds to the cases with $y=[y^{(1)},\cdots,y^{(m)}]^T$ that satisfies (3). Typical examples are multiple cause models, Gaussian factor analysis (FA), binary FA, and non-Gaussian FA as well as their extensions [45], [49], [52], [53].

- those bidirectional models that trade off the features of the above two. One is the Helmholtz machines that is motivated by a fast approximation of the ML learning on the

Manuscript received March 15, 2003; revised November 1, 2003. This work was supported by a grant from the Research Grant Council of the Hong Kong SAR, Project No: CUHK4169/00E.

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

Digital Object Identifier 10.1109/TNN.2004.828767

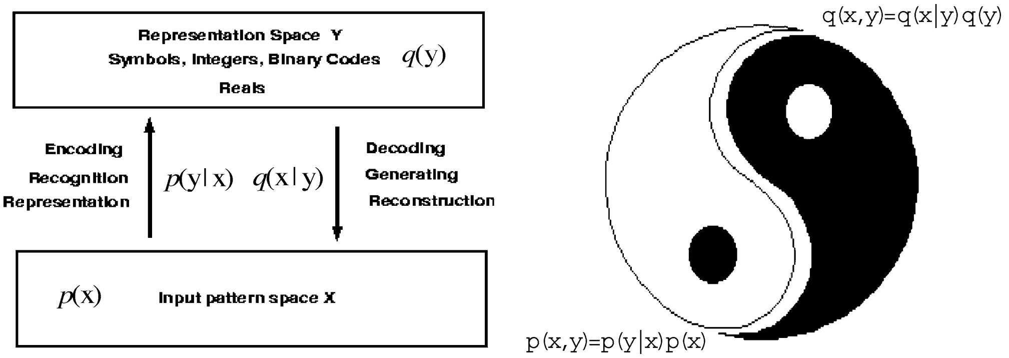


Fig. 1. Bayesian Ying–Yang system.

generative model (4) via adding in a forward path $p(y|x)$ [12], [13], [17]. The other is the LMSER learning [62] that was experimentally found to implement ICA firstly in [25] under the name of nonlinear PCA, and was further found to actually implement a principal ICA that can be regarded as either an extension of ICA with noise or a regularized binary factor analysis [45], [48], [49], [53], [56].

Beyond all the above discussed as well as other variants of similar types, this BYY system also leads to both typical supervised learning models such as mixture-of-experts (ME) [20], [22], [23], the alternative ME model [60], radial basis function nets and extensions [45], [50], [54], three layer nets [48], [50], [54], and typical temporal models such as Kalman filter [9], [24], Hidden Markov model and extensions [33], temporal FA, temporal ICA and temporal LMSER, etc. [47], [49].

The name of BYY system not just came for the above direct analogy between (1) and the Ying–Yang concept, but also is closely related to that the principle of making learning on (1) is motivated from the well known harmony principle of the Ying–Yang philosophy, which is different both from making $p(x)$ by (4) fit a set of samples $\{\bar{x}_t\}_{t=1}^N$ under the ML principle [37] or its approximation [12], [13], [17], [38] as well as simply the least MSE criterion [62], and from making $q(y)$ by (2) satisfy certain prespecified properties such as maximum entropy or matching (3), [4], and [7]. Under this harmony principle, the Ying–Yang pair by (1) is learned coordinately such that the pair is matched in a compact way as shown in Fig. 1. In other words, the learning is made in a twofold sense that

- The difference between the two Bayesian representations in (1) should be minimized.
- The resulted entire BYY system should be of the least complexity.

Mathematically, this principle can be implemented by [48], [49], and [59]

$$\max_{\theta,\mathbf{m}} H(\theta,\mathbf{m}), H(\theta,\mathbf{m}) = H(p\|q,\theta) = \int p(y|x)p(x)\ln\left[q(x|y)q(y)\right] \times \mu(dx)\mu(dy) - Z_q \tag{5}$$

where $\mu(.)$ is a given measure, θ consists of all the unknown parameters in $p(y|x)$, $q(x|y)$, and $q(y)$ as well as $p(x)$ (if any), while $\mathbf{m}$ is the scale parameter of the inner representation y. It is simply m for the case by (3). In a general case, $\mathbf{m}$ is a set of integers that acts as different types of scale parameters. The task of determining θ is called *parameter learning*, and the task of selecting $\mathbf{m}$ is called *model selection* since a collection of specific BYY systems by (1) with different scale values corresponds to a family of specific models that share a same system configuration but in different scales.

As described in [48] and (5) introduces a new mechanism that makes model selection implemented.

- either automatically during the following parameter learning with scale parameters in $\mathbf{m}$ initialized large enough:

$$\max_{\theta} H(\theta), \quad H(\theta) = H(\theta,\mathbf{m}) = H(p\|q,\theta) \tag{6}$$

  which makes θ take a specific value that is equivalent to make $\mathbf{m}$ reduced to an appropriate one in its effect. On a Gaussian mixture case of (4), it means that the correct number of Gaussian components or clusters is automatically determined during learning. It was firstly implemented in 1995 [59] by the so called hard-cut EM algorithm without the regularization role of Z_q. Further improvements were subsequently obtained with regularization imposed via either a Z_q in a normalization term or a particularly designed parametric $p(y|x)$ [50], which leads to a nature similar to the rival penalized competitive learning (RPCL) learning [48], [50], [61]. Moreover, both the hard-cut EM and RPCL type algorithms have been developed for implementing multisets mixture learning [46] and binary FA [45].
- or after implementing parameter learning for θ^* at each of candidates of $\mathbf{m}$ via enumerating scale parameters in $\mathbf{m}$ incrementally to large upper bounds, we select a best $\mathbf{m}^*$ via the following type of model selection criteria:

$$\min_{\mathbf{m}} J(\mathbf{m}), \; J(\mathbf{m}) = -H(\theta^*,\mathbf{m}) = -H(p\|q,\theta^*). \tag{7}$$

Making model selection by (7) is necessary in the cases that (6) becomes not applicable. One case is, as to be discussed on the case by (47) in Section IV-A, that certain constraint has to be imposed on a part of θ during learning θ^* via (6). The other case is that θ^* is obtained not by (6) but by the following Kullback divergence based parameter learning:

$$\min_{\theta} KL(\theta) = \int p(y|x)p(x)\ln\frac{p(y|x)p(x)}{q(x|y)q(y)}\mu(dx)\mu(dy) \tag{8}$$

1. Gaussian Mixture[46,48] $\sum_{l=1}^{m} \alpha_l G(x \mid \mu_l, \Sigma_l)$, $J(m) = 0.5\sum_{l=1}^{m} \alpha_l (\ln|\Sigma_l| + 0.5h^2 Tr[\Sigma_l^{-1}]) - \sum_{l=1}^{m} \alpha_l \ln \alpha_l$, $m_\theta = k(0.5d^2 + 0.5d + 1)$

2. Factor Analysis[45,47,48, 49] $q(x \mid \theta) = \int G(x \mid Ay, \sigma^2 I) q(y \mid \theta_y) dy$, $J(\mathbf{m}) = J_{\mathbf{x}|\mathbf{y}}(\mathbf{m}) + J_{\mathbf{y}}(\mathbf{m})$, $J_{\mathbf{x}|\mathbf{y}}(\mathbf{m}) = 0.5d(\ln \sigma^2 + \frac{h^2}{\sigma^2})$

	$q(y \mid \theta_y)$	$J_{\mathbf{y}}(m)$	m_θ
(a) FA[47,49]	$G(y \mid 0, I)$	$0.5m[1 + \ln(2\pi)]$	md
(b) BFA[45,47,53]	$\prod_{j=1}^{m} q_j^{y^{(j)}} (1 - q_j)^{1 - y^{(j)}}$	$-\sum_{j=1}^{m} [q_j \ln q_j + (1 - q_j)\ln(1 - q_j)]$	$md + m$
(c) IFA[47,49]	$\prod_{j=1}^{m} \sum_{i=1}^{\kappa_j} \beta_{ji} G(y^{(j)} \mid \mu_{ji}, \sigma_{ji}^2)$	$0.5m[1 + \ln(2\pi)] + \sum_{j=1}^{m} \sum_{i=1}^{k_j} \beta_{ji}[0.5 \ln \sigma_{ji}^2 - \ln \beta_{ji}]$	$md + 3\Sigma_{j=1}^{m} (\kappa_j - 1)$
(d) pTFA[43,47,49]	$G(y \mid B\xi, I)$	$0.5m[1 + \ln(2\pi)]$	$md + m$
(e) iTFA[43,47,49]	$G(y \mid 0, B\Lambda B + I)$,	$0.5m[1 + \ln(2\pi)] + 0.5 \sum_{j=1}^{m} \ln(1 - b_j^2)$	$md + 2m$

$B = diag\,[b_1, \cdots, b_m]$ and $\Lambda = diag\,[\lambda_1^2, \cdots, \lambda_m^2]$

3. RBF net[47,49,53] $q(z \mid x) = \sum_{l=1}^{m} p(l \mid x) G(z \mid A_l x + b_l, \sigma_z^2 I)$, $J(m) = 0.5 d_z \ln \sigma_z^2 + \widetilde{J}(m) - \sum_{l=1}^{m} \alpha_l \ln \alpha_l$

(a) $p(l \mid x) = \frac{\alpha_l G(x \mid m_l, \Sigma_l)}{\sum_{l=1}^{m} \alpha_l G(x \mid m_l, \Sigma_l)}$: $\widetilde{J}(m) = 0.5\sum_{l=1}^{m} \alpha_l (\ln|\Sigma_l| + 0.5h^2 Tr[\Sigma_l^{-1}])$, $m_\theta = m[\dim(\Sigma_l) + d_x + d_x d_z + d_z + 2]$

(b) $p(l \mid x) = p(l \mid x, \phi)$: $\alpha_l = \frac{\Sigma_{t=1}^{N} p(l \mid x, \phi)}{N}$, use $\frac{\alpha_l G(x \mid m_l, \Sigma_l)}{\sum_{l=1}^{m} \alpha_l G(x \mid m_l, \Sigma_l)}$ to approximate $p(l \mid x, \phi)$, $m_\theta = \dim(\phi) + m(d_x d_z + d_z + 1)$

4. Three Layer Net[45,47,50,54] $q(z \mid x) = G(z \mid As(Wx + d) + b, \sigma_z^2 I)$, $m_\theta = m(1 + d_x + d_z)$

$$J(m) = 0.5 d_z \ln \sigma_z^2 - \sum_{j=1}^{m} [q_j \ln q_j + (1 - q_j)\ln(1 - q_j)] - 0.5 \ln|W^T W| - \frac{\Sigma_{t=1}^{N} \ln[s(\mathbf{w}_j x_t + d_j)(1 - s(\mathbf{w}_j x_t + d_j))]}{N},$$

$$q_j = \frac{\Sigma_{t=1}^{N} s(\mathbf{w}_j x_t + d_j)}{N}, \quad s(y) = [s(y^{(1)}), \cdots, s(y^{(m)})] \text{ for } y = [y^{(1)}, \cdots, y^{(m)}] \text{ and } s(r) = (e^r - e^{-r})/(e^r + e^{-r}).$$

Fig. 2. Typical examples of $J(\mathbf{m})$ and m_θ, where d, d_z are dimensions of x, z, and $dim(\phi)$ denotes the number of free parameters in ϕ. Each criterion, e.g., the one for IFA, is referred by the text via the notation "*Eqn.2(c) in Fig. 2*".

which has been systematically investigated in the early study of the BYY learning and actually shown being equivalent to the ML learning on $q(x)$ by (4) when $p(y|x)$ is free [59].

In the above two cases, $J(\mathbf{m})$ has a type of U-shape as to be shown later in Fig. 8. When θ^* is obtained via (6) without any constraints on θ, it is not necessary to use (7) since $\mathbf{m}^*$ is determined automatically during implementing (6). Actually, $J(\mathbf{m})$ in this case will have a L-shape. As scale parameters in $\mathbf{m}$ grows, $J(\mathbf{m})$ first reaches $\mathbf{m}^*$ decreasingly and then remains unchanged as $\mathbf{m}$ further varies. In the cases that N is very small, the model selection can be improved with $J(\mathbf{m})$ in (7) replaced by

$$J_G(\mathbf{m}) = J(\mathbf{m}) + \frac{m_f + m_{E_X}}{N} \tag{9}$$

with the details refereed to (20) that will be encountered later in Section II-B. The first application of (7) was given in 1995 [59] for the number of Gaussians in a Gaussian mixture and of clusters after clustering by the K-means algorithm. Subsequently, specific criteria have been derived for various models of supervised, unsupervised, and temporal learning [45], [47]–[49], [52]. Several examples will be given in Fig. 2 in Section II-B.

The implementation of either (6) or (8) can be made by alternating the following two steps:

$$\begin{aligned} \text{Ying-step}: \quad & \text{fixing } p(x,y) \\ & \text{update unknowns in } q(x,y) \\ \text{Yang-step}: \quad & \text{fixing } q(x,y) \\ & \text{update unknowns in } p(x,y) \end{aligned} \tag{10}$$

which is called the Ying–Yang alternative procedure. It is guaranteed that either of $-H(\theta)$ and $KL(\theta)$ gradually decreases until becomes converged. The details are referred to [44] and [48].

In this paper, the model selection ability of the BYY harmony learning and the regularization role of Z_q are further explored. In Section II, new justification is provided from an information theoretic perspective, with comparative discussions made on its relations to and differences from the studies on not only minimum message length (MML) [41], [42], minimum description length (MDL) [34], Bayesian approach, and the bit-back based MDL [18], [19], but also Akaike information criterion (AIC), maximum likelihood, information geometry [2], [3], [11], Helmholtz machines [12], [13], [17] and variational approximation [36], [38]. In Section III, a generalized projection geometry is introduced for further understanding such a mechanism of model selection and regularization. Then, new algorithms are developed for implementing Gaussian FA and non-Gaussian FA in Section IV, such that appropriate factors are selected automatically during learning. Before giving conclusions, experiments are demonstrated in Section V.

II. An Information Transfer Perspective

A. MDL and Bayesian Approach

In the past decade, extensive studies have been made on the MDL [34]. Sharing the common sprit of the MML [41], [42], the BIC model selection criterion [32], [39], and the celebrated Kolmogorov complexity [42], the key idea is to implement the well known Ockham's principle of economy to code a set of samples $\{\bar{x}_t\}_{t=1}^N$ for being transferred from a sender to a receiver via a two-part coding. One is the amount of bits for coding the residuals of using a parametric model $p(x|\theta)$ to fit this set of samples $\{\bar{x}_t\}_{t=1}^N$. The 2nd part is the amount of bits for coding the parameter set θ, provided that the function form of $p(x|\theta)$ has already known at the receiver and thus no need for being encoded. A best information transfer is reached when the bits for both the parts are minimized.

In the existing literature, given a density model $p(x|\theta)$ for a d dimensional real random vector x, the amount of bits per sample $\bar{x}_t$ to be transmitted is described by $b_t^\varepsilon = -\ln p(\bar{x}_t|\theta) - d\ln\delta$, where $\delta > 0$ is a prespecified constant resolution and thus its role is usually ignored. The total amount of bits for the first part is $b^\varepsilon = \sum_{t=1}^N b_t^\varepsilon$. The amount b_θ^ε of bits for the second part is common to every sample, and thus only needs to be transmitted one time in advance. Thus, the average amount of bits to be transmitted is $(1/N)\sum_{t=1}^N b_t^\varepsilon + (b_\theta^\varepsilon/N)$. For a large size N of samples, the second term becomes very small and thus can be ignored. The minimization of the first term is actually equivalent to the ML learning. However, this term does not contain enough information to select an appropriate complexity (e.g., the number of parameters in θ) for $p(x|\theta)$. On a contrary, for a finite size N of samples we encounter a so called over-fitting effect that the larger the complexity is, the smaller the residual of using $p(x|\theta)$ to fit the set $\{\bar{x}_t\}_{t=1}^N$ is, and thus the smaller the first term is. The second term b_θ^ε described by $-\ln p(\theta)$ takes its role that balances off the over-fitting effect since b_θ^ε increases as the complexity increases.

This two part coding $b^\varepsilon + b_\theta^\varepsilon$ was firstly suggested under the name of the MML for clustering analysis [41], [42]. It has been also studied from an equivalent perspective that maximizes

$$\sum_{t=1}^N \ln q(\bar{x}_t|\theta) + \ln q(\theta) \tag{11}$$

under the name of Bayesian learning [15]. It is also called the maximum posteriori estimate (MAP) when used for determining θ only. One key problem is that the priori $q(\theta)$ is usually not available and thus is estimated very roughly, e.g., by a noninformative uniform prior or Jeffery priori [21]. The learning performance can be considerably deteriorated by an inappropriate $q(\theta)$.

Under the name of the MDL [34], an improvement is proposed by encoding x subject to the marginal distribution

$$q(x) = \int q(x|\theta)q(\theta)\mu(d\theta). \tag{12}$$

The total bits in this way is short than that by the MML. The total bits by the MML contain a redundant amount of bits for encoding θ since a part of bits underlying

$$p(\theta|x) = \frac{q(x|\theta)q(\theta)}{\int q(x|\theta)q(\theta)\mu(d\theta)} \tag{13}$$

is already contained in the bits for encoding x subject to $q(x|\theta)$. Though obtained from a different perspective, the MDL is actually equivalent to the Bayesian information criterion (BIC) that was proposed a decade earlier [32], [39], but recently widely studied in the literature of machine learning under the name of the evidence based or marginal Bayesian approach [27], [28], featured by the maximization of $\sum_{t=1}^N q(x_t)$.

In spite of the differences in concept, all the above approaches actually all crash into a same criterion in implementation after $q(\theta)$ is over-simplified into a noninformative uniform prior, as will be further discussed at the end of Section II-D.

B. BYY Harmony Learning With Full Representation

The BYY harmony learning can also be understood from an information transfer perspective, with a new insight on its ability for model selection and regularization. For this purpose, we start at considering a generalized case of BYY harmony learning.

Instead of only considering y as an inner representation x in (1), the parameter set θ is also partly a representation of the entire sample set but not any individual sample $\bar{x}_t$ alone. Generally, we can extend (1) into

$$p(X,R) = p(R|X)p(X), q(X,R) = q(X|R)q(R) \tag{14}$$

for the joint distribution of the observation X and its inner representation R as follows:

- X consist of a set of random vectors $\{x_t\}$ that may be linked via certain topological relations [44]. The simplest one can be a line topology, that is, $X = x_1 \cdots x_t \cdots x_N$ denotes a sequence. A BYY system on this type of observation is called temporal BYY system on which studies have been made in [43], [49], [53] with both new insights and new results. In the simplest case, $x_1, \cdots, x_t, \cdots, x_N$ are mutually *independent* and *identically distributed* (i.i.d.), which is the main focus of this paper.
- $R = \{Y, \theta\}$ with Y consisting of i.i.d. $y_1, \cdots y_t, \cdots, y_N$. Moreover, the parameter set θ is randomly taken according to either *a priori* distribution $q(\theta)$ before observing anything or a posteriori distribution $p(\theta|X)$ after observing instances of X.

Thus, (14) is added with the following details:

$$p(X) = \prod_{t=1}^N p(x_t)$$
$$p(x_t) = G(x_t|\bar{x}_t, h^2 I)$$

$$\begin{aligned} p(R|X) &= p(Y|X,\theta)p(\theta|X) \\ p(Y|X,\theta) &= \prod_{t=1}^{N} p(y_t|x_t,\theta) \\ q(X|R) &= q(X|Y,\theta) \\ &= \prod_{t=1}^{N} q(x_t|y_t,\theta) \\ q(Y) &= q(Y|\theta)q(\theta) \\ &= q(\theta)\prod_{t=1}^{N} q(y_t|\theta) \end{aligned} \qquad (15)$$

where $G(x|m,\Sigma)$ denotes a Gaussian density with mean vector m and covariance matrix Σ, and $\bar{X} = \bar{x}_1,\cdots,\bar{x}_t,\cdots,\bar{x}_N$ is a specific value of X in a sense that each x_t takes a specific value $\bar{x}_t$.

In an analogy to the process from (1) to $H(p\|q,\theta)$ by (5), it follows from (14) and (15) that:

$$\begin{aligned} H(p\|q) &= \int p(R|X)p(X) \\ &\quad \times \ln\left[q(X|R)q(R)\right]\mu(dX)\mu(dR) - Z_q \\ &\approx N\int p(\bar{X})p(\theta|\bar{X})H(p\|q,\bar{X},\theta) \\ &\quad \times\mu(d\theta)\mu(d\bar{X}) + \frac{E_X}{N}\Bigg] \\ E_X &= \int p(\bar{X})p(\theta|\bar{X})\ln q(\theta)\mu(d\theta)\mu(d\bar{X}) \\ H(p\|q,\bar{X},\theta) &= \int p(y_t|x_t,\theta)p(x_t) \\ &\quad \times \ln\left[q(x_t|y_t,\theta)q(y_t|\theta)\right]\mu(dx_t)\mu(dy_t) - Z_q \\ with\, p(x_t) &= G(x_t|\bar{x}_t,h^2 I). \end{aligned} \qquad (16)$$

The above approximation is justified when $h > 0$ in $G(x_t|\bar{x}_t,h^2 I)$ is small and there will be no approximation when $h = 0$. Actually, $H(p\|q,\bar{X},\theta)$ is same as that in (5) and the above $H(p\|q)$ is equivalent to [51, eq. (27)]. $H(p\|q)$ differs from $H(p\|q,\bar{X},\theta)$ by taking in a consideration on the randomness of θ and $\bar{X}$.

Usually, an appropriate *a priori* $q(\theta)$ is not available. One way is given as follows [51]:

$$\begin{aligned} q(\theta) &= \frac{Z_q(\theta)}{\int Z_q(\theta)\mu(d\theta)} \\ p(\theta|\bar{X}) &= \frac{q(\bar{x}_t|\bar{y}_t,\theta)q(\bar{y}_t|\theta)}{\int Z_q(\theta)\mu(d\theta)} \\ Z_q(\theta) &= \sum_{t=1}^{N} q(\bar{x}_t|\bar{y}_t,\theta)q(\bar{y}_t|\theta) \end{aligned}$$

which can be inserted into (16) for a further study. However, its computing is tedious.

In the existing literature [28], [32], [34], [39], a so called improper noninformative uniform prior $q(\theta) = 1$ is used, in a consideration that we naturally like that $q(\theta)$ is uniform to any values of θ when there is no any *a priori* knowledge available. However, this $q(\theta) = 1$ is not a proper uniform density on an infinite domain R^{m_f}. However, directly adopting $q(\theta) = 1$ leads to $E_X = 0$ in (16), which has no bias on parameter learning for θ but also no help on model selection for $\mathbf{m}$. A better trick is considering the integral over (12) via simply setting $q(\theta) = 1$, which leads to an additional term for helping model selection [28], [32], [34], [39].

This paper alternatively suggests to consider two extreme cases of $q(\theta)$. One extreme case is that $q(\theta)$ is free and thus determined via maximizing $H(p\|q)$ or equivalently maximizing E_X in (16), which leads to $q(\theta) = p(\theta|\bar{X})$. To get a specific form of $p(\theta|\bar{X})$, we consider an estimator $\theta = T(\bar{X})$ on a sample set $\bar{X}$, e.g., θ^* via the parameter learning by (6) can be denoted as $\theta^* = T(\bar{X})$. Assuming that $\theta = T(\bar{X})$ is unbiased to the true value θ^o, in help of the celebrated Cramer-Rao inequality we can let $p(\theta|\bar{X})$ to be given by the asymptomatic form of the best unbiased estimator θ on a size N of samples. That is

$$\begin{aligned} p(\theta|\bar{X}) &= G\left(\theta|\theta^o,\frac{F^{-1}(\theta^o)}{N}\right) \\ F(\theta) &= \int \frac{\partial \ln p(x|\theta)}{\partial\theta} \\ &\quad \times \frac{\partial \ln p(x|\theta)}{\partial\theta^T}p(x|\theta)dx \text{ or equivalently} \\ F(\theta) &= -\int \frac{\partial^2 \ln p(x|\theta)}{\partial\theta\partial\theta^T}p(x|\theta)dx \end{aligned} \qquad (17)$$

where $F(\theta)$ is known as Fisher-information matrix. Thus, we have

$$\begin{aligned} E_X &= \int p(\bar{X})p(\theta|\bar{X})\ln p(\theta|\bar{X})\mu(d\theta)\mu(d\bar{X}) \\ &\approx -0.5m_\theta \ln\frac{2\pi e\sigma_f^2}{N} \\ \sigma_f^2 &= |F(\theta^o)|^{\frac{-1}{m_\theta}} \end{aligned} \qquad (18)$$

where m_θ denotes the number of free parameters in θ and σ_f^2 acts effectively like a variance. Moreover, it follows from θ^* by (6) and $\nabla_\theta H(p\|q,\bar{X},\theta)|_{\theta=\theta^*} = 0$ that:

$$\begin{aligned} H(p\|q,\bar{X},\theta) &\approx H(p\|q,\bar{X},\theta^*) \\ &\quad -0.5(\theta-\theta^*)^T h_{\bar{X}}(\theta^*)(\theta-\theta^*) \\ h_{\bar{X}}(\theta) &= -\frac{\partial^2 H(p\|q,\bar{X},\theta)}{\partial\theta\partial\theta^T} \\ \frac{1}{N}H(p\|q) &= \int p(\bar{X})p(\theta|\bar{X}) \\ &\quad \times H(p\|q,\bar{X},\theta)\mu(d\theta)\mu(d\bar{X}) + \frac{E_X}{N} \\ &\quad \int p(\bar{X})p(\theta|\bar{X}) \\ &\quad \times H(p\|q,\bar{X},\theta)\mu(d\theta)\mu(d\bar{X}) \\ &= H(p\|q,\bar{X},\theta^*) \\ &\quad -0.5Tr\left[\left(\Sigma_H + \frac{F^{-1}(\theta^o)}{N}\right)h_{\bar{X}}(\theta^*)\right] \end{aligned} \qquad (19)$$

where $Tr[A]$ is the trace of the matrix A and $\Sigma_H = E(\theta^* - \theta^o)(\theta^* - \theta^o)^T$. Further assuming that $\theta^* = T(\bar{X})$ is also unbiased, we can approximately let the above $F(\theta^o)$ replaced by $F(\theta^*)$ and let Σ_H given by the covariance matrix $E(\theta^* - E\theta^*)(\theta^* - E\theta^*)^T$. This covariance matrix may be estimated via cross validation with an expensive computing cost.

Moreover, we can also simply let Σ_H replaced by the covariance matrix of the best estimator as in (17). Together with (18), it follows from (16) and $-(1/N)H(p\|q)$ that we get:

$$\begin{aligned} J_G(\mathbf{m}) \approx & -H(p\|q, \bar{X}, \theta^*) + \frac{m_f + m_{E_X}}{N} \\ m_f =& Tr\left[F^{-1}(\theta^*)h_{\bar{X}}(\theta^*)\right] \\ m_{E_X} =& 0.5 m_\theta \left[\ln \frac{e}{N} + \ln\left(2\pi\sigma_f^2\right)\right] \end{aligned} \tag{20}$$

where m_f can be regarded as an effective number of free parameters.

Another extreme case is considering a prior $q(\theta) = G(\theta|\theta^o, N\sigma^2 I)$ that becomes a noninformative uniform as $N \to \infty$. It follows from (16) that:

$$\begin{aligned} E_X =& -0.5 m_\theta \left[\ln N + \ln(2\pi\sigma^2)\right] - c_N \\ m_{E_X} =& 0.5 m_\theta \left[\ln N + \ln(2\pi\sigma^2)\right] \end{aligned} \tag{21}$$

where $c_N = 0.5\sigma^{-2}Tr[F^{-1}(\theta^o)]/N^2$ can be ignored as N increases. One way is let $\sigma^2 = \sigma_f^2$ given as in (18). In this case, m_{E_X} in (20) and (21) indicates two extreme settings. We may also take their average as follows:

$$m_{E_X} = 0.5 m_\theta \left[0.5 + \ln\left(2\pi\sigma_f^2\right)\right]. \tag{22}$$

Thus, model selection by (7) can be further improved into that by (9) via $J_G(\mathbf{m})$ by (20) with m_{E_X} by either of (20), (21) and (22), especially when N is quite small. Typical examples are listed in Fig. 2. For simplicity, we even may let $m_f = m_\theta$ after approximately regarding $F(\theta^*) = h_{\bar{X}}(\theta^*)$.

C. An Information Transfer Perspective

As shown in Fig. 3, we consider a system in which x is mapped to an inner representation y that is encoded and sent to the receiver, and the receiver then decodes y to reconstruct x. Learning is made to obtain $p(y|x, \theta_{y|x})$ for getting y from x, the distribution $q(y|\theta_y)$ for the codes on y, and the decoder $q(x|y, \theta_{x|y})$ for getting x from y, under assumption that the function forms of $q(y|\theta_y)$ and $q(x|y, \theta_{x|y})$ are already known at the receiver.

We consider the problem of transferring a set $\bar{X} = \{\bar{x}_t\}_{t=1}^N$ of known samples from $p(x)$. The probability of getting $\bar{x}_t$ for mapping is approximately $p(\bar{x}_t)\mu(\delta_p(\bar{x}_t))$, where $\delta_p(\bar{x}_t)$ is a small volume centering at $\bar{x}_t$ such that we have $\sum_{\bar{x}_t \in \bar{X}} p(\bar{x}_t)\mu(\delta_p(\bar{x}_t)) \approx 1$ as an approximation of $\int p(x)\mu(dx) = 1$. Since there are N such small volumes, one above constraint is not enough to fix them and extra constraints should be imposed. The simplest one is assuming that all of them are same, i.e., $\delta_p(\bar{x}_t) = \delta_p(x)$ for every t. Thus,

$$\mu\left(\delta_p(x)\right) \approx \frac{1}{\sum_{t=1}^N p(\bar{x}_t)}. \tag{23}$$

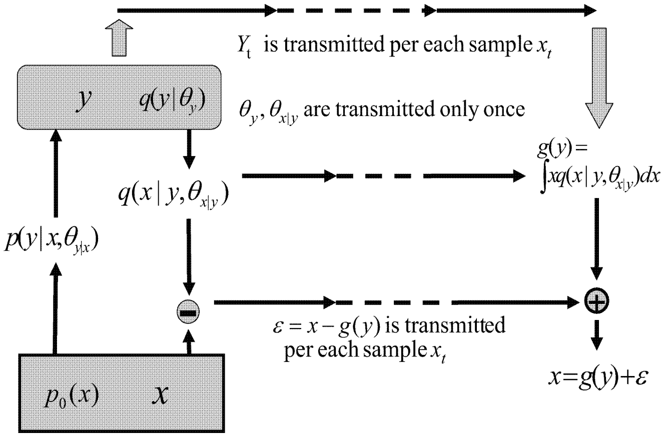

Y_t is transmitted per each sample x_t
y
$q(y|\theta_y)$
$\theta_y, \theta_{x|y}$ are transmitted only once
$g(y) = \int xq(x|y, \theta_{x|y})dx$
$q(x|y, \theta_{x|y})$
$p(y|x, \theta_{y|x})$
$\varepsilon = x - g(y)$ is transmitted per each sample x_t
$x = g(y) + \varepsilon$
$p_0(x)$
x

Fig. 3. Bayesian Ying–Yang harmony learning from an information-theoretic perspective.

Moreover, each sample $\bar{x}_t$ may be encoded via a set $\bar{Y}_t$ that may consist of one sample $\bar{y}_t$ or a finite number of samples of y. The probability of getting both $\bar{x}_t$ and $\bar{y} \in \bar{Y}_t$ is approximately $p(\bar{y}|\bar{x}_t, \theta_{y|x})\mu(\delta_p(y)) \times p(\bar{x}_t)\mu(\delta_p(x))$ subject to $\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu(\delta_p(y))\mu(\delta_p(x)) \approx 1$. Similar to the discussion in Section II-A, we can encode $\bar{y}$ via a two part encoding subject to $q(\bar{y}|\theta_y)$. One is an amount of bits for coding θ_y. The other is the amount of bits for coding the residual information of $\bar{y}$ that is not included in θ_y. The amount for one sample $\bar{y}$ is $-\ln[q(\bar{y}|\theta_y)\mu(\delta_q(y))]$ and thus in total is given as follows:

$$\begin{aligned} b^y =& -\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu\left(\delta_p(x)\right) \\ & \times \mu\left(\delta_p(y)\right) \ln\left[q(\bar{y}|\theta_y)\mu\left(\delta_q(y)\right)\right] \\ =& b_1^y + b_2^y \\ b_1^y =& -\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu\left(\delta_p(x)\right) \\ & \times \mu\left(\delta_p(y)\right) \ln q(\bar{y}|\theta_y), \\ b_2^y =& -\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu\left(\delta_p(x)\right) \\ & \times \mu\left(\delta_p(y)\right) \ln \mu\left(\delta_q(y)\right) \\ \approx & -\ln \mu\left(\delta_q(y)\right). \end{aligned} \tag{24}$$

To reconstruct $\bar{x}_t$ from $\bar{y} \in Y_t$ at the receiver, we need to code an amount of bits for coding $\theta_{x|y}$ to get $g(\bar{y}) = \int xq(x|\bar{y}, \theta_{x|y})dx$ at the receiver. Also, we need to code the residual $\varepsilon_t = \bar{x}_t - g(\bar{y})$ with an amount of bits as follows:

$$\begin{aligned} b^\varepsilon =& -\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu\left(\delta_p(x)\right) \\ & \times \mu\left(\delta_p(y)\right) \ln\left[q(\bar{x}_t|\bar{y}, \theta_{x|y})\mu\left(\delta_q(x)\right)\right] \\ =& b_1^\varepsilon + b_2^\varepsilon \\ b_1^\varepsilon =& -\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu\left(\delta_p(x)\right) \\ & \times \mu\left(\delta_p(y)\right) \ln q(\bar{y}|\theta_y) \\ b_2^\varepsilon =& -\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} p(\bar{y}|\bar{x}_t, \theta_{y|x})p(\bar{x}_t)\mu\left(\delta_p(x)\right) \\ & \times \mu\left(\delta_p(y)\right) \ln \mu\left(\delta_q(x)\right) \\ \approx & -\ln \mu\left(\delta_q(x)\right). \end{aligned} \tag{25}$$

Moreover, $(b_1^y + b_1^\varepsilon)/N$ can be approximated by its limit as the sizes of samples in $\bar{X}$ and $\bar{Y}_t$ tend to infinite large and $\delta_p(x) \to$

dx, $\delta_p(y) \to dy$, which leads to the first term of the harmony measure $-H(p\|q,\theta)$ in (5). Also, we have

$$Z_q = -\ln \mu\left(\delta_q(y)\right) - \ln \mu\left(\delta_q(x)\right) = -\ln\left[\mu\left(\delta_q(y)\right)\mu\left(\delta_q(x)\right)\right]. \tag{26}$$

In other words, $b^y + b^\varepsilon$ leads to the harmony measure $-H(p\|q,\theta)$ in (5). Similar to (23), it follows from $\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} q(\bar{x}_t|\bar{y},\theta_{x|y})\ q(\bar{y}|\theta_y)\mu(\delta_q(y))\mu(\delta_q(x)) \approx 1$ that

$$\mu\left(\delta_q(y)\right)\mu\left(\delta_q(x)\right) \approx \frac{1}{\sum_{\bar{x}_t \in \bar{X}} \sum_{\bar{y} \in \bar{Y}_t} q(\bar{x}_t|\bar{y},\theta_{x|y})q(\bar{y}|\theta_y)} \tag{27}$$

which introduces a type of regularization studied under the name of normalization in [48], [50].

In the case that $p(x)$ is estimated by the following Parzen window estimate:

$$p_h(x) = \frac{1}{N}\sum_{t=1}^{N} G(x|\bar{x}_t, h^2 I) \tag{28}$$

with a small $h \neq 0$, $H(p\|q,\theta)$ in (5) becomes

$$\begin{aligned} H(p\|q,\theta) &= \frac{1}{N}\sum_{t=1}^{N}\int p(y|\bar{x}_t,\theta_{y|x}) \\ &\quad \times \ln\left[q(\bar{x}_t|y,\theta_{x|y})q(y)\right]\mu(dy) - h^2 Tr[\pi_q] - Z_q \\ \pi_q &= -\frac{1}{N}\sum_{t=1}^{N}\int p(y|\bar{x}_t,\theta_{y|x})\frac{\partial^2 \ln q(\bar{x}_t|y,\theta_{x|y})}{\partial x \partial x^T}\mu(dy). \end{aligned} \tag{29}$$

Moreover, with $p_h(x)$ replacing $q(x|y,\theta_{x|y})$ in (27) we correspondingly get that

$$\mu\left(\delta_q(y)\right)\mu\left(\delta_q(x)\right) \approx \frac{1}{\sum_{\bar{x}_t \in \bar{X}} p_h(\bar{x}_t) \sum_{\bar{y} \in \bar{Y}_t} q(\bar{y}|\theta_y)} \tag{30}$$

which introduces another type of regularization that was previously studied under the name of data-smoothing [46], [50]. Also, when $\bar{Y}_t$ is not limited to each t but consists of samples in all t we have

$$Z_q \approx \ln\left[\sum_{\bar{x}_t \in \bar{X}} p_h(\bar{x}_t)\right] + \ln\left[\sum_{\bar{y} \in \bar{Y}} q(\bar{y}|\theta_y)\right] \tag{31}$$

which provides an alternative implementation of normalization regularization.

Further considering that the amount of bits $b_\theta^y + b_\theta^\varepsilon$ for the parameter set $\{\theta_y, \theta_{x|y}\} = \theta$ actually encodes the information that is described by $p(\theta|\bar{X})$ but has not been covered by samples $\bar{X} = \{x_1, \cdots, x_t, \cdots, x_N\}$, we have $-\int p(\theta|\bar{X})\ln p(\theta)\mu(d\theta)$ that is equivalent to $-E_X$ in (16). Therefore, $-NH(p\|q,\theta) + b_\theta^y + b_\theta^\varepsilon$ finally leads to $-H(p\|q)$ by (16). In other words, the BYY harmony learning with $H(p\|q)$ by (16) attempts to maximizing the best information transfer in a sense of the minimum expected coding bits, while the BYY harmony learning with $-H(p\|q,\theta)$ by (5) implements this goal approximately by ignoring the bits $b_\theta^y + b_\theta^\varepsilon$.

D. Relation and Difference to MDL, Bayesian Approach, Akaike Information Criterion, and Minimum Entropy

The above information transfer perspective shares the same sprit of minimizing the coding bits of information with the MDL approach discussed previously in Section II-A. However, there are two differences.

One key difference is that the two coding parts by the previously discussed MDL have been replaced with the three coding parts by the BYY harmony learning. By MDL, as discussed in Section II-A, model selection is enabled via the balance between the bits b_θ and the bits b^ε. Discarding the bits b_θ, the MDL degenerates back to the ML learning, with its model selection ability disabled. By the BYY harmony learning, in addition to the bits b^ε for encoding the residual part (i.e., the bits of x that is unable to be described by the BYY system in consideration), the role of b_θ has now been jointly shared by the bits b^y for encoding the inner representation y of x and by the bits $b_\theta^y + b_\theta^\varepsilon$ as a counterpart of b_θ. Not only carrying the information about x, the bits b^y also encode the scales of representation that either indicates model complexity directly or includes the core part of model complexity. Thus, discarding the bits $b_\theta^y + b_\theta^\varepsilon$ will not disable the model selection ability, though it may weaken the performance of model selection when N is rather small.

The above difference also leads to an important difference in implementing model selection. To avoid an inappropriately chosen $q(\theta)$ to deteriorate learning considerably, only a noninformative uniform prior is used as $q(\theta)$ in MDL and thus has no effect on parameter learning for determining θ, which is still made by a ML learning as the first step. The MDL criterion comes in effect at the second step for model selection. This two step implementation costs heavily since parameters learning on getting θ has to be made on all the candidate models in consideration. By the BYY harmony learning, the job of model selection is also performed via a family of densities $q(y|\theta_y)$ with a given parametric structure but unknown parameters θ_y that is determined during learning process, which is a significant relaxation from solely relying on *a priori* density $q(\theta)$. As a result, not only parameter learning is performed more accurately but also model selection is made via the scale parameters of y that are determined automatically during learning parameters in θ_y.

Another difference is that the term Z_q replaces the role of a prefixed quantization resolution δ that is currently widely adopted in the MDL literature. Without considering what type of data distribution it is, manually setting a constant δ is simply because there is no a better solution available but it is clearly not a good solution. In the BYY harmony learning by (5), the term Z_q provides a regularization role [43], [46], [48]. In the data smoothing implementation, $Z_q = Z_q(h)$ takes the input data distribution in consideration via the Parzen window estimator by (28) with a smoothing parameter h. This h takes a role similar to a quantization resolution δ, but now it is also learned to adapt the set of samples $\{x_t\}_{t=1}^N$. In the normalization implementation, $Z_q = Z_q(\theta_{x|y}, \theta_y)$ takes the input data distribution in consideration indirectly via the learned parametric densities $q(x|y,\theta_{x|y})$ and $q(y|\theta_y)$ on the a set of samples $\{x_t\}_{t=1}^N$.

It is interesting to further observe that $H(p||q,\theta)$ degenerates back to the likelihood function after removing away both the role of Z_q by setting $Z_q = 0$ and the inner representation y in Section II-B (i.e., having $R = \theta$ only). Moreover, $-(1/N)H(p||q)$ degenerates into $-(1/N)\sum_{t=1}^N \ln p(x_t|\theta) + (m_f + m_{E_X})/N)$, which is different from both AIC and BIC. In addition, with $Z_q \neq 0$, $H(p||q,\theta)$ will degenerate to a likelihood function with a type of regularization similar to that discussed in Section II-A-2 of [48].

It also deserves to mention that $-\int p(R|X)p(X)\ln[q(X|R)q(R)]\mu(dR)\mu(dX)$ becomes equivalent to an entropy when $p(R|X)p(X) = q(X|R)q(R)$. However, maximizing harmony is generally different from minimizing entropy. First, $p(R|X)p(X) = q(X|R)q(R)$ will not happen for making learning on a finite size N of samples in $\bar{X}$. Second, the minimization of neither the entropy $-\int q(X|R)q(R)\ln[q(X|R)q(R)]\mu(dR)\mu(dX)$ nor the entropy $-\int p(R|X)p(X)\ln[p(R|X)p(X)]\mu(dR)\mu(dX)$ can provide a meaningful tool for loading the information in the sample set $\bar{X}$ to $p(R|X)$ or $q(X|R)$ and $q(R)$. However, after the parameters of $q(X|R)$ and $q(R)$ have been determined via another learning principle (e.g., maximum likelihood), a same specific criteria $J(\mathbf{m})$ in (7) can be obtained for model selection from either $-H(p||q,\theta^*)$ or $-\int q(X|R)q(R)\ln[q(X|R)q(R)]\mu(dR)\mu(dX)$.

E. Relation and Difference to the Bits-Back Based MDL

Both the MDL implementation with a bits-back strategy in [18], [19] and the BYY harmony learning share a common feature that x is mapped to y and then y is coded for transmission, instead of coding x_t directly for transmission. However, there are differences again.

Similar to Section II-D, one difference is that the BYY harmony learning uses the term Z_q to replace the role of a prefixed quantization resolution δ that is still adopted in the bits-back based MDL. Also, the bits $b_\theta^y/N + b_\theta^\varepsilon/N$ in $H(p||q)$ have not been taken in consideration by the bits-back based MDL. Another even fundamental difference is that BYY harmony learning does not adopt the bits-back strategy that is the key feature of the bits-back based MDL [18], [19].

Considering the dependence among the inner codes generated by $p(y|x)$, it has been argued in [18], [19] that the total amount of bits to be transferred should be subtracted by the following amount of bits:

$$H(\theta_{y|x}) = \int p(y|x,\theta_{y|x})p(x)\ln p(y|x,\theta_{y|x})\mu(dx)\mu(dy). \tag{32}$$

With this amount claimed back, the total amount of bits that has been considered by [18], [19] is actually equivalent to the Kullback divergence $KL(\theta)$ by (8), after discarding a term $H_x = \int p(x)\ln p(x)dx$ that is irrelevant to learning when $p(x)$ is given by (28) with $h = 0$. In other words, the bits-back based MDL [18], [19] actually provides an interpretation to the Kullback learning by (8) from a information transfer perspective. In a contrast, without including $H(\theta_{y|x})$ by (32), the discussion in Section II-C provides an interpretation to the BYY harmony learning by (5) and (20). As further discussed in the next subsection, the Kullback learning by (8) is equivalent to implementing parameter learning under the ML principle or its certain regularized variants in lack of model selection ability, while BYY harmony learning provides a new mechanism that makes model selection either after or during parameter learning.

An insight can also be obtained via observing the role of the bits-back amount $-H(\theta_{y|x})$ by (32). With the dimension of y fixed, the Kullback learning by (8) implements a stochastic encoding by $p(y|x,\theta_{y|x})$ that allows certain dependence among the resulted codes. This dependence generates a redundant amount $-H(\theta_{y|x})$ of bits that is suggested in [18] and [19] to be subtracted from computing the total amount of bits. In a contrast, aiming at seeking appropriate representation scales for y, the BYY harmony learning by (5) with Z_q by (26) actually minimizes

$$\begin{aligned}-H(p||q,\theta) = KL(\theta) - H(\theta_{y|x}) - H_x \\ - \ln\mu(\delta_q(x)) - \ln\mu(\delta_q(y)).\end{aligned} \tag{33}$$

Moreover, we have $-H(\theta_{y|x}) - \ln\mu(\delta_q(y)) \geq 0$ and $-H_x - \ln\mu(\delta_q(x)) = 0$ when $p(x)$ is given by (28) with $h = 0$. Thus, $-H(\theta,k) \geq KL(\theta)$ is an upper bound of the total bits considered in [18] and [19].

When $p(y|x)$ is free, $\max_{p(y|x)} H(p||q,\theta)$ results in [48] and[50]:

$$p(y|x) = \delta(y - y(x)), \quad y(x) = arg\max_y [q(x|y)q(y|\theta_y)]. \tag{34}$$

It happens similarly when $p(y|x)$ is parametric either directly in a form of $\delta(y - y(x))$ or tends to be pushed into this form via $\max_{p(y|x)} H(p||q,\theta)$. In these cases, $-H(\theta_{y|x}) - \ln\mu(\delta_q(y))$ reaches its minimum value 0. Thus, the BYY harmony learning will achieve the minimum total number of bits instead of acting as one upper bound.

In other words, the BYY harmony learning reaches the optimal coding bits both by learning unknown parameters and by squeezing out any stochastic redundancy that comes from allowing one x to share more than one inner codes of y. As a result, all the inner codes will occupy a representation space as compact as possible. That is, model selection occurs automatically during the process of approaching the optimal coding bits. On a contrary, the dimension for the inner codes of y is prespecified for a bits-back based MDL case, and the task is learning unknown parameters under this fixed dimension (usually assumed to be large enough for what needed). Due to there is certain redundancy in the representation space, it is allowed that one x may be redundantly represented by more than one inner codes. Instead of squeezing out this dependence, the redundant bits of $-H(\theta_{y|x})$ by a stochastic $p(y|x)$ is not zero but discounted in counting the total amount of bits. Though such a redundant coding makes information transfer more reliable, allowing redundancy in the representation space of y already means that this representation space is not in its minimum complexity.

F. Relation and Difference to Information Geometry, Helmholtz Machine, and Variational Approximation

The minimization of $KL(\theta)$ by (8) with respect to a free $p(y|x)$ will result in

$$\begin{aligned} p(y|x) &= \frac{q(x|y)q(y)}{q(x)} \\ q(x) &= \int q(x|y)q(y)\mu(dy) \\ KL(\theta) &= \int p(x)\ln\frac{p(x)}{q(x)}\mu(dx) \end{aligned} \tag{35}$$

which is equivalent to the ML learning on $q(x)$ when $p(x)$ is given by (28) with $h = 0$ [59]. This case relates to the information geometry theory (IGT) [2], [3], [11] that is also equivalent to the ML learning on $q(x)$ by (4), and the well known EM algorithm [14], [30] is reached by the em algorithm in IGT.

Making parameter learning by (8) also relates to the Helmholtz machine learning (HML) when $p(x)$ is given by (28) with $h = 0$ and both $p(y|x)$ and $q(x|y)$ are both given by the conditional independent densities based on the sigmoid layered networks as used in [13], [17]. That is, the densities are given with the following format

$$\begin{aligned} p(u|v) &= \prod_{j=1}^{m} \pi_j(v)^{u^{(j)}} \left(1-\pi_j(v)\right)^{1-u^{(j)}} \\ \pi(v) &= [\pi_1(v), \cdots, \pi_m(v)]^T \\ &= s(Wv+c), \\ s(y) &= \left[s\left(y^{(1)}\right), \cdots, s\left(y^{(m)}\right)\right]^T \\ &\qquad 0 \le s(r) \le 1 \text{ is a sigmoid function} \end{aligned} \tag{36}$$

where u is a binary vector. In this case, making parameter learning by (8) actually becomes equivalent to an one layer HML. Also, the well known wake-sleep algorithm for HML can be regarded as a specific adaptive form of (10). With a general insight via (10), other specific algorithms for implementing the HML may also be developed.

It is also deserve to notice that making parameter learning by (8) with a parametric $p(y|x) \in \mathcal{P}_{y|x}(\theta_{y|x})$ is different from a free $p(y|x) \in \mathcal{P}^0_{y|x}$ in that a parametric family $\mathcal{P}_{y|x}(\theta_{y|x})$ is a subset of the family $\mathcal{P}^0_{y|x}$ containing all the density functions in the form $p(y|x)$. Thus, we always have $\min_{p(y|x)\in\mathcal{P}_{y|x}(\theta_{y|x})} KL \ge \min_{p(y|x)\in\mathcal{P}^0_{y|x}} KL$. When $p(x)$ is given by (28) with $h = 0$, it follows from (35) that the latter becomes equivalent to the ML learning on $q(x)$ by (4). In other words, making parameter learning by (8) with a parametric $p(y|x)$ actually implements a type of constrained ML learning on $q(x)$, which is also called a variational approximation to the ML learning on $q(x)$ [36], [38].

The BYY harmony learning are different from the three existing approaches as follows.

First, the BYY harmony learning minimizes the harmony measure $-H(p\|q,\theta)$ instead of a Kullback divergence $KL(p\|q)$ in (8), not only for parametric learning but also for model selection. Even using the Kullback learning by (8) for parameter learning, it is still followed by making model selection via (7) or (9). In contrast, making parameter learning via minimizing Kullback divergence is the only target in IGT, HML, and variational approximation, while the issues of regularization and model selection are out of the scopes of their studies.

Second, as discussed later in (45), the harmony learning may also be regarded as implementing a type of constrained ML learning, especially when $p(y|x) \in \mathcal{P}_{y|x}(\theta_{y|x})$ is parametric. However, it is different from the above discussed constrained ML learning via variational approximation [36], [37]. As shown in (45), an additional constraint should be imposed on both types of learning to make them become equivalent.

Third, even focusing on the common part, i.e., making parameter learning via minimizing Kullback divergence for implementing parameter learning, these studies are made from different perspectives with different purposes. IGT studies the general properties possessed by (8) and alternative minimization for two general p, q from the perspectives of geometry structure [11] and differential geometry structure [2], [3]. HML and variational approximation consider developing efficient algorithms for implementing empirical parameter learning on a forward-backward net via an approximation of the ML learning on the marginal density $q(x)$ in (4). In contrast, the BYY learning studies two distributions in the two complementary Bayesian representations in (1) by systematically investigating not only three typical architectures for different learning tasks, but also regularization by either a conscience de-learning type via normalization or a Tikhonov-type via data smoothing with its smoothing parameter h estimated in an easy implementing way. While IGT, HML and variational approximation have neither explicitly and systematically considered the two complementary representations in (1) nor the regularization of such two types.

III. A Projection Geometry Perspective

Through obtaining a quasi-Pythagorean relation under the Kullback divergence by (8), this divergence based learning has been further theoretically studied from the perspective of both the ordinary geometry and differential geometry under the name of information geometry [2], [3], [11]. Actually, neither the harmony measure by (5) nor the Kullback divergence by (8) satisfies all the properties of the conventional metric measure. Moreover, the harmony measure by (5) even does not satisfy a quasi-Pythagorean relation that the Kullback divergence satisfies. In this section, we suggest to investigate both the harmony measure based learning and the Kullback divergence based learning from a geometry perspective, relaxed from a metric level to a projection level.

We denote $U_c = \{u : u \in R^d \ and \ \|u\|^2 = c^2, \text{ for a constant } c > 0\}$, which is a sphere shell with the radius c. As illustrated in Fig. 4, from the concept of the inner product $u^T v$ at its special case $\|u\| = 1$ we can get a concept of the projection Π^v_u of v on u. Moreover, a residual vector $v - u$ also has a projection Π^{v-u}_u on u. Furthermore, we have the following interesting nature

$$\begin{aligned} &When\ v,\ u\ locate\ on\ a\ same\ shell\ U_c, \\ &the\ concepts\ of\ maximizing\ the\ projection\ v\ to\ u, \end{aligned}$$

$$\begin{aligned}&minimizing\ the\ residual\ projection\ (v-u)\ to\ u,\\&of\ making\ residual\ v-u\ being\ orthogonal\ to\ u,\\&and\ the\ equality\ v=u\ are\ all\ the\ same\ thing.\end{aligned}\tag{37}$$

In an analogy, we consider a functional space

$$\mathcal{Q}=\left\{q(u):q(u)\geq 0\,\text{and}\int q(u)\mu(du)<\infty\right\}\tag{38}$$

where $u\in S_u\subseteq R^d$ and μ is a given measure on the support S_u. A useful subspace $\mathcal{P}_c\subset\mathcal{Q}$ is

$$\mathcal{P}_c=\left\{p(u):p(u)\geq 0,\int p(u)\mu(du)=c,\ \text{for a constant}\ c>0\right\}.\tag{39}$$

Particularly, when $c=1$, $\mathcal{P}_1$ is the probability density space.

Given $p(u)\in\mathcal{P}_c$, $q(u)\in\mathcal{P}_{c'}$, we define the projection of $q(u)$ on $p(u)$ under the constraint of a set of samples $\{u_t\}_{t=1}^N$ as follows:

$$\begin{aligned}H(p\|q)&=\int p(u)\ln q(u)\mu(du)-Z_q,\\Z_q&=-\sum_{t=1}^{N}p(u_t)\mu\left(\delta_p(u)\right)\ln\mu\left(\delta_q(u)\right)\\&\approx\ln\mu\left(\delta_q(u)\right)\ subject\ to\ \sum_{t=1}^{N}p(u_t)\mu\left(\delta_p(u)\right)\\&\approx 1,\end{aligned}\tag{40}$$

which acts as a counterpart of Π_u^v as shown in Fig. 4. Specifically, from $\sum_{t=1}^N q(u_t)\mu(\delta_q(u))\approx 1$ we have

$$\mu\left(\delta_q(u)\right)=\frac{1}{\sum_{t=1}^N q(u_t)},\quad Z_q=-\ln\sum_{t=1}^{N}q(u_t).\tag{41}$$

Alternatively, we can also let $\mu(\delta_q(u))\approx\mu(\delta_p(u))$ and get from $\sum_{t=1}^N p(u_t)\mu(\delta_p(u))\approx 1$ that

$$\begin{aligned}\mu\left(\delta_q(u)\right)\approx\mu\left(\delta_p(u)\right)&=\frac{1}{\sum_{t=1}^N p(u_t)}\\Z_q\approx Z_p&=-\ln\sum_{t=1}^{N}p(u_t).\end{aligned}\tag{42}$$

Also, it can be observed that (40) leads to (5) when $p(u)=p(x,y)$, $q(u)=q(x,y)$ and to (16) when $p(u)=p(R|X)p(X)$, $q(u)=q(X|R)q(R)$.

Extending the property that the self-projection of u is simply the norm $\|u\|$, the self-projection of $p(u)$ is $H(p\|p)=\sum_{t=1}^N p(u_t)\mu(\delta_p(u_t))\ln[p(u_t)\mu(\delta_p(u_t))]\approx\int p(u)\ln p(u)\mu(du)-\ln\mu(\delta_p(u))$, which can be regarded as a type of norm of p and it represents the negative entropy of the probability distribution $p(u)\mu(\delta_p(u))$ when $p(u)\in\mathcal{P}_1$ is a density. Also, extending the property that the projection Π_u^v is maximized when v is co-directional with u, $H(p\|q)$ is maximized if and only if $q(u)=(c'/c)p(u)$, i.e., $q(u)$ may not be equal to $p(u)$ but has the same shape as $p(u)$, which can be observed from $\int\hat{p}(u)\ln\hat{q}(u)\mu(du)\leq\int\hat{p}(u)\ln\hat{p}(u)\mu(du)$ with $c\hat{p}(u)=p(u)$, $c'\hat{q}(u)=q(u)$ and $\hat{p}(u),\hat{q}(u)\in\mathcal{P}_1$.

However, there are three differences in comparison with the situation of Π_u^v. One is that each density represents a point of infinite dimension. Second, each component is constrained to be nonnegative. Third, the constraint $\int p(u)\mu(du)=c$ is a first-order linear constraint, instead of the quadratic constraint $\|u\|^2=c^2$.

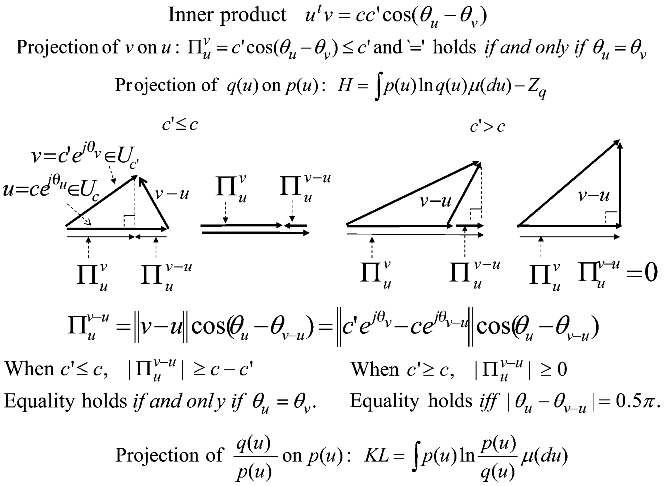

Inner product $u^tv=cc'\cos(\theta_u-\theta_v)$
Projection of v on u: $\Pi_u^v=c'\cos(\theta_u-\theta_v)\leq c'$ and '=' holds *if and only if* $\theta_u=\theta_v$
Projection of $q(u)$ on $p(u)$: $H=\int p(u)\ln q(u)\mu(du)-Z_q$
$c'\leq c$
$c'>c$
$v=c'e^{j\theta_v}\in U_{c'}$
$u=ce^{j\theta_u}\in U_c$
$v-u$
Π_u^v
Π_u^{v-u}
$v-u$
$v-u$
Π_u^v
Π_u^{v-u}
Π_u^v
Π_u^{v-u}
Π_u^v
$\Pi_u^{v-u}=0$
$\Pi_u^{v-u}=\|v-u\|\cos(\theta_u-\theta_{v-u})=\|c'e^{j\theta_v}-ce^{j\theta_{v-u}}\|\cos(\theta_u-\theta_{v-u})$
When $c'\leq c$, $|\Pi_u^{v-u}|\geq c-c'$
Equality holds *if and only if* $\theta_u=\theta_v$.
When $c'\geq c$, $|\Pi_u^{v-u}|\geq 0$
Equality holds *iff* $|\theta_u-\theta_{v-u}|=0.5\pi$.
Projection of $\frac{q(u)}{p(u)}$ on $p(u)$: $KL=\int p(u)\ln\frac{p(u)}{q(u)}\mu(du)$

Fig. 4. From an inner product back to a projection in the vector space.

Due to these differences, the maximization of $H(p\|q)$ makes not only that $p(u)$ and $q(u)$ has a same shape in the sense $q(u)=(c'/c)p(u)$ but also that $p(u)$ prefers to have a simplest shape $c\delta(u-u^*)$, where $u^*=arg\max_u q(u)$. When $p(u)$ is free to be any choice in $\mathcal{P}_c$ and $q(u)$ is free to be any choice in $\mathcal{P}_{c'}$, the maximization of $H(p\|q)$ will finally make that both $p(u)$ and $q(u)$ become impulse functions. When $p(u)\in P$, $q(u)\in Q$ are constrained to be unable to become impulse functions, the maximization of $H(p\|q)$ will make that $p(u)$ and $q(u)$ become close in a shape of a least complexity but not able completely equal. Therefore, the maximization of $H(p\|q)$ on a BYY system (1) indeed implements the harmony principle as described in the introduction section, while the maximization of the projection u to v only ensures u and v become co-directional but does not have such a least complexity.

In addition, $H(p\|q)$ does not share the symmetry that possessed by Π_u^v at $\|v\|=\|u\|$. If exchanging the positions of p, q, though $\max H(p\|q)$ still makes that $p(u)$ and $q(u)$ have a same shape, it is different in a sense that $q(u)$ but not $p(u)$ is now pushed to a shape of $c'\delta(u-u^*)$.

Moreover, if we use $p(u)\in\mathcal{P}_c$ to represent $q(u)\in\mathcal{P}_{c'}$ and define the discrepancy or residual[1] by $p(u)\ominus q(u)=p(u)\delta_p(u)/q(u)\delta_q(u)\approx p(u)/q(u)$ under $\delta_p(u)\approx\delta_q(u)$, we get that this residual projection on $p(u)$ as follows:

$$\begin{aligned}R(p\|q)&=\int p(u)\ln\left[\frac{p(u)\delta_p(u)}{q(u)\delta_q(u)}\right]\mu(du)\\&=H(p\|p)-H(p\|q)\\&\approx\int p(u)\ln\left[\frac{p(u)}{q(u)}\right]\mu(du).\end{aligned}\tag{43}$$

Since $p(u)=c\hat{p}(u)$, $q(u)=c'\hat{q}(u)$ with $\hat{p}(u),\hat{q}(u)\in\mathcal{P}_1$, it follows that:

$$R(p\|q)\approx c\left[KL(\hat{p}\|\hat{q})+\ln\frac{c}{c'}\right]$$

[1] Under this definition, $p(u)\ominus q(u)$ is not guaranteed to still remain in $\mathcal{Q}$, a further discussion is referred to [44].

$$KL(\hat{p}||\hat{q}) = \int \hat{p}(u) \ln \left[\frac{\hat{p}(u)}{\hat{q}(u)}\right] \mu(du). \tag{44}$$

From which we can observe the following properties:

- Minimizing $R(p||q)$ is equivalent to both minimizing the self-projection of $p(u)$ and maximizing the projection of $q(u)$ on $p(u)$. When the self-projection $H(p||p)$ is fixed at a constant, minimizing the residual projection is equivalent to maximizing $H(p||q)$.
- The residual $p(u) \ominus q(u)$ is said to be orthogonal to $p(u)$ when the residual projection $R(p||q)$ becomes 0 that happens when the norm of p and the projection of q on p become the same, i.e., $H(p||p) = H(p||q)$.
- When $c = c'$, the minimum value of $R(p||q)$ is 0 which is reached if and only if $p(u) = q(u)$. Moreover, when $c = c' = 1$, $p(u)$ and $q(u)$ are densities and $R(p||q) = KL(p||q)$.

From the above discussions, we see that the concepts of maximizing $H(p||q)$ and of minimizing the residual projection $R(p||q)$ are related, but not equivalent. Even when $c = c' = 1$, we do not have the equivalence that exists between Π_u^v and Π_u^{v-u} as given in (37), as illustrated in Fig. 5. This provides a geometry perspective on why and how the maximization of $H(p||q)$ on a BYY system (1), which is a generalization of maximizing the projection for the co-directionality, is different from the minimization of $KL(p||q)$ on a BYY system (1) or equivalently the maximum likelihood learning, which is a generalization of minimizing the residual projection. Moreover, the latter does not have the least complexity nature that enables the former to make model selection.

However, imposing an additional constraint that $H(p||p)$ is fixed at a constant H_0, we have

$$\max_{p\in P, q\in Q,\ s.t.\ H(p||p)=H_0} H(p||q) \text{ is equivalent to } \min_{p\in P, q\in Q,\ s.t.\ H(p||p)=H_0} KL(p||q). \tag{45}$$

With $p(x)$ given by (28), the constraint $H(p||p) = H_0$ means certain constraint imposed on $p(y|x)$. In these cases, (45) can also be regarded as implementing a type of constrained ML learning, which is different from those of variational approximation [36], [37] that also implements $\min_{p\in P, q\in Q} KL(p||q)$ with $p(y|x)$ in a constrained structure but without requiring the constraint $H(p||p) = H_0$.

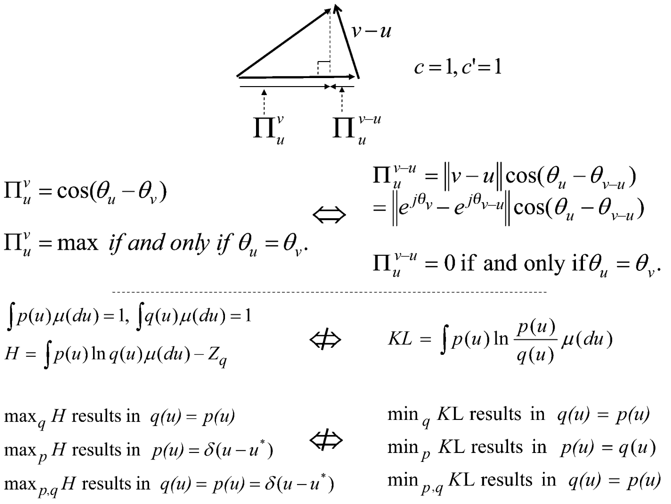

$v-u$
$c=1, c'=1$
Π_u^v
Π_u^{v-u}
$\Pi_u^v = \cos(\theta_u - \theta_v)$
$\Pi_u^v = \max$ *if and only if* $\theta_u = \theta_v$.
$\Leftrightarrow$
$\Pi_u^{v-u} = \|v-u\|\cos(\theta_u - \theta_{v-u})$
$= \|e^{j\theta_v} - e^{j\theta_{v-u}}\|\cos(\theta_u - \theta_{v-u})$
$\Pi_u^{v-u} = 0$ if and only if $\theta_u = \theta_v$.
$\int p(u)\mu(du) = 1, \int q(u)\mu(du) = 1$
$H = \int p(u) \ln q(u)\mu(du) - Z_q$
$\not\Leftrightarrow$
$KL = \int p(u) \ln \frac{p(u)}{q(u)} \mu(du)$
$\max_q H$ results in $q(u) = p(u)$
$\max_p H$ results in $p(u) = \delta(u-u^*)$
$\max_{p,q} H$ results in $q(u) = p(u) = \delta(u-u^*)$
$\not\Leftrightarrow$
$\min_q KL$ results in $q(u) = p(u)$
$\min_p KL$ results in $p(u) = q(u)$
$\min_{p,q} KL$ results in $q(u) = p(u)$

Fig. 5. Unit norm based projection: from the vector space to a functional space.

IV. BYY Independence Learning and Factor Autodetermination

A. BYY Independence Learning on Linear Real Factor Model

Under the constraint by (3), the BYY harmony learning by (5) is called the BYY independence learning and can be further classified into various types according to the specific differences in $p(y|x)$, $q(y|x)$, $q(y)$, and $p(x)$ as well as Z_q.

In this paper, we concentrate on a special class of the BYY independence learning that bases on a free Yang path and a linear Ying path $x = Ay + e$ with y being real and e being a Gaussian noise. That is

$$q(x|y) = G(x|Ay, \Sigma), \quad \textit{and} \quad p(y|x) \text{ is free.} \tag{46}$$

Moreover, an additional constraint on the scaling of y should be imposed since (3) remains satisfied and $q(x)$ by (4) remains unchanged under any scaling transform $\tilde{y} = Dy$ with a diagonal matrix D. This indeterminacy can be removed by imposing the unit variance constraint $E(yy^T) = I$ with $E(y) = 0$ because $E(\tilde{y}\tilde{y}^T) = D^2 \neq I$ for any $D \neq I$. Given in (47) at the bottom of the page are three typical examples: where the Gaussian case leads to the classic factor analysis (FA) [5], [29], while the other cases can be called non-Gaussian FA [47], [49]. Also in (47), $H_n(y^{(j)})$ is the nth-order Chebyshev-Hermite polynomials, ρ_j is the third-order moment of $y^{(j)}$, and $\kappa_j^2 \geq 0$ is the fourth-order moment of $y^{(j)}$, and θ_y consists of a set of unknown parameters

$q(y)$ is given by eq. (3) with

$$q\left(y^{(j)}|\theta_y\right) = \begin{cases} G\left(y^{(j)}|0,1\right) & \text{Gaussian} \\ G\left(y^{(j)}|0,1\right)\left[1 + \frac{\rho_j}{6} H_3\left(y^{(j)}\right) + \frac{\kappa_j^2 - 3}{24} H_4\left(y^{(j)}\right)\right] & \text{Gram-Charlier expansion} \\ \sum_{i=1}^{k_j} \beta_{ji} G\left(y^{(j)}|\mu_{ji}, \sigma_{ji}^2\right),\ s.t.\ E\left(y^{(j)}\right)^2 = 1 & \text{Gaussian mixture} \end{cases} \tag{47}$$

shown in (48) at the bottom of the pageSince a free $p(y|x)$ is actually determined from $\max_{p(y|x)} H(p\|q)$ as in (34), we get one $y_t = y(x_t)$ per each sample x_t. For an example, at the Gaussian case in (47), we have simply

$$y_t = \left[I + A^T\Sigma^{-1}A\right]^{-1} A^T\Sigma^{-1}(x_t - \mu) \qquad \text{where } \mu = 0 \text{ for eq. } (46). \tag{49}$$

For other cases, $y_t = y(x_t)$ is obtained via a nonlinear optimization that can be made by anyone of existing iterative algorithms, denoted as

$$y^{\text{new}}(x_t) = ITER\left(y^{\text{old}}(x_t)\right). \tag{50}$$

Specific algorithms of this type are proposed [44], [48], [49]. For the Gaussian mixture case in (47), the detailed form of (50), under the name of the fixed posterior approximation in [49], consists of two steps:

$$\begin{aligned} \text{Step (a)} : p_{jr} &= \frac{\beta_{jr} G\left(y^{(j)}|\mu_{jr}, \sigma^2_{jr}\right)}{\sum_{i=1}^{k_j} \beta_{ji} G\left(y^{(j)}|\mu_{ji}, \sigma^2_{ji}\right)} \\ b_j &= \sum_{r=1}^{k_j} \frac{p_{jr}}{\sigma^2_{jr}},\ d_j = \sum_{r=1}^{k_j} \frac{p_{jr}\mu_{jr}}{\sigma^2_{jr}} \\ \text{Step (b)} : y_t^{\text{new}} &= \left(A^T\Sigma^{-1}A + \text{diag}[\text{b}_1, \cdots, \text{b}_\text{m}]\right)^{-1} \\ &\quad \times (A^T\Sigma^{-1}x_t + d). \end{aligned} \tag{51}$$

Summarizing all the above discussions and discarding certain irrelevant constant terms, and considering $p(x)$ given by (28), $H(p\|q,\theta)$ by (5) takes the following simplified form [44], [47]:

$$\begin{aligned} H(\theta, m) &= -0.5\ln|\Sigma| - 0.5h^2 Tr[\Sigma^{-1}] - J_y \\ J_y &= -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{m} \ln q\left(y_t^{(j)}|\theta_y\right) \\ \Sigma &= h^2 + \frac{1}{N}\sum_{t=1}^{N} e_t e_t^T \\ e_t &= x_t - Ay_t,\ y_t = y(x_t) \end{aligned} \tag{52}$$

either without regularization via $h = 0$ or with data-smoothing regularization via $h \neq 0$.

Learning by (6) can be made per each pair x_t, y_t to increase this $H(\theta, m)$ via updating $q(x|y)$ by (46) and $q(y)$ by (47). Specifically, $q(x|y)$ is updated via a least mean square like algorithm as follows:

$$\begin{aligned} A^{\text{new}} &= A^{\text{old}} + \eta e_t y_t^T \\ e_t &= x_t - A^{\text{old}} y_t \\ \Sigma^{\text{new}} &= (1-\eta)\Sigma^{\text{old}} + \eta\left(h^2 + e_t e_t^T\right) \end{aligned} \tag{53}$$

where $\eta > 0$ is a given learning step size. Though in a same notation, η is usually different for updating different types of parameters, e.g., it may be chosen differently for updating A, Σ, respectively. In the rest of paper, the notation η is always used in such a sense.

While for $q(y)$, no updating on θ_y is made for the Gaussian case. For the other two cases, updating on θ_y is made to increase $\ln q(y^{(j)})$ as follows:

- For the Gram-Charlier expansion case, we can simply do it in a gradient ascent way

$$\begin{aligned} g_j &= 1 + \frac{\rho_j^{old}}{6} H_3\left(y^{(j)}\right) + \frac{\kappa_j^{2\ old} - 3}{24} H_4\left(y^{(j)}\right) \\ \rho_j^{new} &= \rho_j^{old} + \eta \frac{H_3\left(y^{(j)}\right)}{6g_j} \\ \kappa_j^{2\ new} &= \kappa_j^{2\ old} + \eta \frac{2\kappa_j^{old} - 3}{24 g_j} H_4\left(y^{(j)}\right). \end{aligned} \tag{54}$$

- For the Gaussian mixture case, we have an EM-like updating as follows [49]:

$$\begin{aligned} p_{jr} &= \frac{\beta_{jr} G\left(y^{(j)}|m_{jr}, \sigma^2_{jr}\right)}{\sum_{i=1}^{k_j} \beta_{ji} G\left(y^{(j)}|m_{ji}, \sigma^2_{ji}\right)}, \\ \beta_{jr}^{\text{new}} &= (1-\eta_0)\beta_{jr}^{\text{old}} + \eta_0 p_{jr} \\ \hat{m}_{jr}^{\text{new}} &= m_{jr}^{\text{old}} + \eta_0 p_{jr}\left(y^{(j)} - m_{jr}^{\text{old}}\right) \\ \hat{\sigma}_{jr}^{2\ \text{new}} &= (1-\eta_0 p_{jr})\sigma_{jr}^{2\ \text{old}} + \eta_0 p_{jr}\left(y^{(j)} - m_{jr}^{\text{old}}\right)^2 \\ m_j &= \sum_{r=1}^{k_j} \beta_{jr}^{\text{new}} \hat{m}_{jr}^{\text{new}} \\ \sigma_j^2 &= \sum_{r=1}^{k_j} \beta_{jr}^{\text{new}} \hat{\sigma}_{jr}^{2\ \text{new}}. \end{aligned} \tag{55}$$

In order to insure $Ey^{(j)} = 0$, $E(y^{(j)})^2 = 1$, a normalization is followed as below:

$$m_{jr}^{\text{new}} = \frac{\left(\hat{m}_{jr}^{\text{new}} - m_j\right)}{\sigma_j} \quad \sigma_{jr}^{2\ \text{new}} = \frac{\hat{\sigma}_{jr}^{2\ \text{new}}}{\sigma_j^2}. \tag{56}$$

In a summary, the Ying–Yang alternative procedure by (10) takes the following detailed form given by (57) at the bottom of the next page. For a large size N of samples, regularization is not necessary and thus the above Yang-step (b) can be disabled via simply setting $h = 0$. For a small size N of samples, a data-smoothing regularization takes its role with Yang-step (b) implemented as follows:

$$\begin{aligned} g_h &= \frac{h}{d} Tr[\Sigma^{-1}] + \frac{1}{h} + \frac{h_0^2}{h^3} \\ h_0^2 &= \frac{1}{d}\sum_{t=1}^{N}\sum_{\tau=1}^{N} p_{t,\tau}\|x_t - x_\tau\|^2 \end{aligned}$$

$$\theta_y = \begin{cases} \text{empty} & \text{Gaussian} \\ \left\{\rho_j, \kappa_j^2\right\} & \text{Gram-Charlier expansion} \\ \left\{\beta_{ji}, \mu_{ji}, \sigma^2_{ji}\right\},\ s.t.\ \sum_{i=1}^{k_j}\beta_{ji} = 1,\ 0 \le \beta_{ji} \le 1 & \text{Gaussian mixture} \end{cases}. \tag{48}$$

$$p_{t,\tau} = \frac{e^{-\frac{\|x_t - x_\tau\|^2}{2h^{2\ old}}}}{\sum_{t=1}^{N}\sum_{\tau=1}^{N} e^{-\frac{\|x_t - x_\tau\|^2}{2h^{2\ old}}}} \tag{58}$$

$$h^2 \text{ is given by either } h^2 = \frac{2h_0^2}{1+\sqrt{1+4h_0^2 d^{-1} Tr[\Sigma^{-1}]}}$$

$$or\ h^{2\ old} + \eta_h g_h \text{ with a step size } \eta > 0$$

where d is the dimension of x. The details on this updating are referred to Section 2 in [46] and [48].

It should be noticed that the mechanism of automatically selecting m during learning by (6) has been disabled due to the constraint $E(yy^T) = I$ that actually fixes the dimension m. With the learning by (57) made at each value of m that is enumerated from a small value incrementally, an appropriate m can be selected by either (7) or (9) with $J(m)$ and m_f given by Eqn.2.(a) in Fig. 2.

B. Decorrelated FA and Independence Embedded Decorrelated FA

The constraint $E(yy^T) = I$ removes the scaling indeterminacy, but unfavorably disables the mechanism of automatic selection on m too. As a result, we have to compute $J(m)$ with a very expensive cost. What we really want is to remove the scaling indeterminacy but keep the automatic selection mechanism. This can be achieved via reconsidering the nature of (4) and the difference between the BYY harmony learning and the ML learning. Actually, the ML learning or equivalently the Kullback learning by (8) remains equivalent under any linear transform $\tilde{y} = By$ since $q(x)$ by (4) remains unchanged. However the situation of the BYY harmony learning will be quite different.

We consider the following singular value decomposition

$$A = UDV^T = \sum_{j=1}^{m} d_j u_j v_j^T, \quad U = [u_1, \cdots, u_m]$$

$$V = [v_1, \cdots, v_m], \quad U^T U = I, \quad VV^T = I \tag{59}$$

where u_j is a d-dimension vector and v_j is a m-dimension vector. It can be observed that $d_j = 0$ means that the term $u_j v_j^T$ has no contribution to A and thus is effectively equivalent to reduce the dimension of m by 1. However, neither the ML learning on $q(x)$ by (4) nor the BYY harmony learning via maximizing $\ln G(x|Ay, \Sigma)$ can drive an extra d_j toward 0 for this purpose.

Considering $\tilde{y} = DV^T y$, we have $Ay = UDV^T y = U\tilde{y}$ and

$$E\tilde{y}\tilde{y}^T = DV^T VD = D^2. \tag{60}$$

Moreover, $q(x|y)$ by (46) and $q(y)$ by (47) have been mapped into

$$q(x|\tilde{y}) = G(x|U\tilde{y}, \Sigma), \quad q(\tilde{y}) = |D|^{-1} q(VD^{-1}\tilde{y}|\theta_y). \tag{61}$$

on which $q(x)$ by (4) remains unchanged and thus its ML learning or equivalently the Kullback learning by (8) also remains unchanged. In the existing literature, there are efforts that preprocess input data of x via prewhitening such that $E(xx^T) = I$ and thus a linear model $x = Uy + e$, $U^T U = I$ or equivalently $G(x|Uy, \Sigma)$ can be considered. However, in this way there is still no driving force that pushes an extra d_j toward 0 since the ML learning is still made in these studies.

On a contrary, the BYY harmony learning on (61) will push an extra d_j toward zero via maximizing $\ln q(\tilde{y}) = -\ln|D| + \ln q(VD^{-1}\tilde{y}|\theta_y) = -\sum_{j=1}^{m} \ln|d_j| + \ln q(y|\theta_y)$. In other words, model selection is made automatically during making parameter learning by (6).

Except for the case that $q(\tilde{y})$ is derived from $q(y)$ by (47) in the Gaussian case, the constraint by (3) becomes broken for all the other cases. What still remains to be satisfied is the constraint by (60). In other words, the components of $\tilde{y}$ remains decorrelated but are not guaranteed to be independent. Precisely, we should use the name of decorrelated FA instead of the name of independent FA on the level of $\tilde{y}$. This decorrelated FA still falls in the paradigm of the conventional FA [29]. Interestingly, this decorrelated FA with a non-Gaussian $q(y)$ by (47) implies that components of $y = VD^{-1}\tilde{y}$ will become independent. That is, an independent FA on the level of y is actually embedded in this decorrelated FA on the level of $\tilde{y}$. So, we call it Independence Embedded Decorrelated FA.

Because the mapping $VD^{-1}y$ is invertible, the joint density $q(x|y)q(y)$ differs from $q(x|\tilde{y})q(\tilde{y})$ only in $|D|$. There is no difference on getting $\tilde{y}_t = DV^T y$ either via y_t by (49) and (50) or directly by (49) and (50) in term of $\tilde{y}$ with

$$\tilde{y}_t = (\Lambda^{-1} + U^T\Sigma^{-1}U)^{-1}U^T\Sigma^{-1}(x_t - \mu)$$

$$\text{where } \mu = 0 \text{ for eq. } (46). \tag{62}$$

Corresponding to $H(p||q)$ by (52), now we have

$$H(\theta, m) = -0.5\ln|\Sigma| - 0.5h^2 Tr[\Sigma^{-1}]$$

$$-\ln|D| + \frac{1}{N}\sum_{t=1}^{N} \ln q(VD^{-1}\tilde{y}_t)$$

$$\Sigma = h^2 + \frac{1}{N}\sum_{t=1}^{N} e_t e_t^T$$

$$\begin{aligned}
&\text{Ying Step}: (a) \text{ updating } q(x|y) \text{ by eq. } (53)\\
&\qquad (b) \text{ updating } q(y) \text{ by } \begin{cases} \text{do nothing,} & \text{Gaussian}\\ \text{by eq. } (54) & \text{Gram-Charlier expansion}\\ \text{by eqs. } (55)\&(56) & \text{Gaussian mixture.}\end{cases}\\
&\text{Yang Step}: (a) \text{ getting } y_t \text{ by eq. } (49) \text{ or eq. } (50)\\
&\qquad (b)\ h^2 = \begin{cases} 0, & \text{without regularization}\\ \text{updated} & \text{data-smoothing regularization.}\end{cases}
\end{aligned} \tag{57}$$

$$e_t = x_t - U\tilde{y}_t. \tag{63}$$

Instead of (53), $q(x|y)$ is now updated subject to the constraint $U^TU = I$ by

$$\begin{aligned} e_t &= x_t - U^{old}\tilde{y}_t \\ g_U &= \frac{\partial \ln G(x|U\tilde{y}, \Sigma)}{\partial U} \\ &= \tilde{y}_t e_t^T \Sigma^{old\,-1} \\ U^{\text{new}} &= U^{old} + \eta_0 \left(g_U^T - U^{old} g_U U^{old}\right) \\ \Sigma^{\text{new}} &= (1-\eta_0)\Sigma^{old} + \eta_0 e_t e_t^T. \end{aligned} \tag{64}$$

While for $q(\tilde{y})$, the updating on θ_y of $q(y|\theta_y)$ remains the same as by (54) and (55), plus D and V being updated as follows:

$$\begin{aligned} V^{\text{new}} &= V^{\text{old}} + \eta \left(g_V - V^{\text{old}} g_V^T V^{\text{old}}\right) \\ D^{\text{new}} &= \big\{(1-\eta_0)I - \eta_0 \big(\text{diag}\left[V^T\phi(y_t)y_t^T V\right] \big\} D^{\text{old}} \\ g_V &= \frac{\partial \ln q(VD^{-1}\tilde{y}_t|\theta_y)}{\partial V} \\ &= \phi(y_t) y_t^T V \\ y_t &= VD^{-1}\tilde{y}_t \\ \phi(y_t) &= \frac{\partial \ln q(y|\theta_y)}{\partial y}. \end{aligned} \tag{65}$$

Similar to U, the updating on V is made under the constraint $V^TV = I$. In addition to updating as in (64) and (65), U, V may also be updated by other orthogonal flow algorithms [47]. The above updating on D comes from $D^{-1}g_D D^{-1} = D^{-1}(\partial \ln q(VD^{-1}\tilde{y}_t|\theta_y)/\partial D)D^{-1} = -diag[V^T\phi(y_t)\tilde{y}_t^T]$ and $D^{new} = D^{old} - \eta_0(diag[V^T\phi(y_t)y_t^T V] + I)D^{old}$.

In a summary, the Ying–Yang alternative procedure by (57) now takes the following detailed form:

$$\begin{aligned} \text{Ying Step}: &(a) \text{ updating } q(x|y) \text{ by eq. (64)} \\ &(b) \text{ updating } q(y) \text{ same as in eq. (57)} \\ &\quad \text{and update } V \text{ and } D \text{ by eq. (65)} \\ &(c) \text{ If } d_j^2 \text{ tends to 0 constantly, discard } y^{(j)} \\ &\quad \text{and all the parameters related to } y^{(j)}. \\ \text{Yang Step}: &(a)\&(b) \text{ same as in eq. (57).} \end{aligned} \tag{66}$$

Being different from the procedure by (57) and those existing FA algorithms [29], [35], [55], the procedure by (66) makes parameter learning with appropriate factors selected automatically via Ying-step (c) in help of the least complexity nature of the BYY harmony learning that pushes the corresponding variance d_j^2 toward zero.

In some special cases, we have $V = I$ and the procedure by (66) can be simplified correspondingly.

Finally, it should be noted that it is y_t but not $\tilde{y}_t$ acts as the recovered independent factors. That is, the mapping $x_t \to y_t$ performs the ICA under the noise e_t.

C. Determination of Module Number of Each Dimensional Representation

For the first two cases of $q(y^{(j)}|\theta_y)$ in (47), the representation scale of y is fully specified by the determination of the dimension m. However, for the third case, the determination of the dimension m has only completed a part of model selection task, since the number k_j of modules in each dimension, i.e., the number k_j of Gaussians in each scalar Gaussian mixture, has not be discussed yet. These $\{k_j\}$ can also be determined via considering the structure of the representation space of y.

The inner representation by (3) can be regarded as a degenerated case of a joint real-discrete inner representation $q(y, \ell) = q(\ell)\prod_{j=1}^{m_\ell} q(y^{(j)}|\ell)$ in place of y, with ℓ taking a finite number of integers. That is, we have a $\Sigma - \Pi$ type of $q(y) = \sum_{\ell=1}^{k} q(\ell)\prod_{j=1}^{m_\ell} q(y^{(j)}|\ell)$. As shown in [48], this type of y-representation will lead us to various local extensions of independent analysis.

The inner representation by (3) with $q(y^{(j)}|\theta_y)$ being the third case in (47) can be regarded as a degenerated case of the following joint real-discrete inner representation:

$$\begin{aligned} q(y, \ell) &= \prod_{j=1}^{m} q\left(y^{(j)}, \ell^{(j)}\right) \\ q\left(y^{(j)}, \ell^{(j)}\right) &= q\left(y^{(j)}|\ell^{(j)}\right) q\left(\ell^{(j)}\right) \\ y &= \left[y^{(1)}, \cdots, y^{(m)}\right] \\ \ell &= \left[\ell^{(1)}, \cdots, \ell^{(m)}\right] \\ \ell^{(j)} &= 1, \cdots, k_j. \end{aligned} \tag{67}$$

That is, we have a $\Pi - \Sigma$ type of

$$q(y) = \prod_{j=1}^{m} \sum_{\ell^{(j)}=1}^{k_j} q\left(y^{(j)}|\ell^{(j)}\right) q\left(\ell^{(j)}\right) \tag{68}$$

which returns to (3) with $q(y^{(j)}|\theta_y)$ being the third case in (47) when

$$\begin{aligned} q\left(y^{(j)}|\ell^{(j)}\right) &= G\left(y^{(j)}|\mu_{ji}, \sigma_{ji}^2\right) \\ q\left(\ell^{(j)} = i\right) &= \beta_{ji}, \quad \beta_{ji} \geq 0, \quad \sum_{i=1}^{k_j} \beta_{ji} = 1. \end{aligned} \tag{69}$$

With the above $q(y, \ell)$ replacing $q(y)$ in (5), it follows from (34) and (52) that:

$$\begin{aligned} J_y &= -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{m} \ln q\left(y_t^{(j)}, \ell_t^{(j)}\right) \\ &= -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{m} \left[\ln q\left(y_t^{(j)}|\ell_t^{(j)}\right) + \ln q\left(\ell_t^{(j)}\right)\right] \\ y_t &= \left[y_t^{(1)}, \cdots, y_t^{(m)}\right] \\ \ell_t &= \left[\ell_t^{(1)}, \cdots, \ell_t^{(m)}\right] \\ [y_t, \ell_t] &= arg\max_{\{y,\ell\}} \{\ln q(x_t|y) \\ &\qquad + \sum_{j=1}^{m}\left[\ln q\left(y^{(j)}|\ell^{(j)}\right)\right. \\ &\qquad\qquad \left.+ \ln q\left(\ell^{(j)}\right)\right]\} \\ y_t &= y_{\ell_t}(x_t) \\ y_\ell(x_t) &= arg\max_{y}\left[\ln q(x_t|y) + \sum_{j=1}^{m} \ln q\left(y^{(j)}|\ell^{(j)}\right)\right] \end{aligned}$$

$$\begin{aligned}\ell_t = arg\max_{\ell}\{&\ln q\left(x_t|y_\ell(x_t)\right)\\ &+\sum_{j=1}^{m}\left[\ln q\left(y_\ell^{(j)}(x_t)|\ell^{(j)}\right)\right.\\ &\left.\left.+\ln q\left(\ell^{(j)}\right)\right]\right\}. \qquad (70)\end{aligned}$$

For (69), instead of using an iterative algorithms as in (50), the above maximization for $y_\ell(x_t)$ is a quadratic optimization that can be analytically solved as follow:

$$\begin{aligned}y_\ell(x_t) &= \left[\Lambda_\ell^{-1} + A^T\Sigma^{-1}A\right]^{-1}\left[A^T\Sigma^{-1}x_t + \Lambda_\ell^{-1}\mu_\ell\right]\\ \mu_\ell &= \left[\mu_{1\ell^{(1)}},\cdots,\mu_{m\ell^{(m)}}\right]^T\\ \Lambda_\ell &= \mathrm{diag}\left[\sigma^2_{1\ell^{(1)}},\cdots,\sigma^2_{m\ell^{(m)}}\right]. \qquad (71)\end{aligned}$$

Then, ℓ_t is given via the discrete optimization by the last line in (70).

The learning is still made by (57) with

$$\begin{aligned}&for\ j = 1,\cdots,m,\\ &p_{ji} = \begin{cases}1, & \text{if } i = \ell_t^{(j)},\\ 0, & \text{otherwise},\end{cases}\\ &\quad\text{and update all of } \beta_{ji},\mu_{ji},\sigma^2_{ji}\\ &\quad\text{by eq. (55) and eq. (56)} \qquad (72)\end{aligned}$$

which is implemented at each setting of m, $\{k_j\}_{j=1}^m$ that are enumerated from small values incrementally, then an appropriate $\mathbf{m} = \{m,\{k_j\}\}$ can be selected by (7) or (9) with $J(\mathbf{m})$ and m_f given by Eqn.2.(c) in Fig. 2.

To make learning with automatic selection on m, $\{k_j\}_{j=1}^m$, corresponding to (61) with $y = VD^{-1}\tilde{y}$ we can also get (63) modified into

$$\begin{aligned}H\left(\theta,m,\{k_j\}_{j=1}^m\right) &= -0.5\ln|\Sigma| - 0.5h^2Tr[\Sigma^{-1}]\\ &+\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{m}\ln\left[\beta_{j\ell_t^{(j)}}G\left(\tilde{y}_t^{(j)}|\tilde{\mu}_{j\ell_t^{(j)}},\tilde{\sigma}^2_{j\ell_t^{(j)}}\right)\right] \qquad (73)\end{aligned}$$

with Σ as in (63) and y_t and ℓ_t as in (70).

In implementation, we modify (66) as follows:

$$\begin{aligned}\text{Ying Step}: &(a)\ U,\Sigma \text{ are still updated by eq. (64)}\\ &(b) \text{ with } p_{ji} \text{ by eq. (72), update all of } \beta_{ji}\\ &\quad \mu_{ji},\sigma^2_{ji} \text{ by eq. (55) and eq. (56), update}\\ &\quad V \text{ still by eq. (65) with } \phi(y_t) \text{ taking a}\\ &\quad \text{simple form } \phi(y_t) = -\tilde{\Lambda}_{\ell_t}^{-1}(y_t - \tilde{\mu}_{\ell_t})\\ &(c) \text{ If } \beta_{ji} \text{ tends to 0 constantly, discard the}\\ &\quad \text{corresponding } G\left(\tilde{y}^{(j)}|\tilde{\mu}_{ji},\tilde{\sigma}^2_{ji}\right), \text{ If}\\ &\quad \beta_{ji} = 1 \text{ and } \tilde{\sigma}^2_{ji} \text{ tends to 0 constantly,}\\ &\quad \text{discard the dimension } y^{(j)} \text{ and all the}\\ &\quad \text{parameters related to } y^{(j)}.\\ \text{Yang Step}: &\quad \text{get } \ell_t \text{ by the last line in eq. (70) and}\\ &\quad \text{get } y_\ell(x_t) \text{ by eq. (71) then make } (b) \text{ sameas in}\\ &\quad \text{eq. (57).} \qquad (74)\end{aligned}$$

V. Experiments

Experiments are made on comparing the non-Gaussian factor analysis (NFA) and the EM based exact ML learning algorithm [31] that is called independent factor analysis (IFA) in [6].

By the IFA, the product of $q(y^{(j)})$ by a Gaussian mixture as in (47) is used via introducing a set of random variable $z^{(j)}$, $j = 1,\cdots,m$ such that the product of m summations in (3) is equivalently exchanged into a summation of $\prod_j n_j$ products. As a result, the integral in (4) becomes a summation of the $\prod_j n_j$ analytically computable integrals on Gaussians and thus $q(x)$ becomes a mixture of $\prod_j n_j$ Gaussians. Thus, they were able to implement the ML learning on $q(x)$ with the EM algorithm. At each step, however, a summation of $\prod_j n_j$ terms has to be computed. The complexity increases exponentially with the number m of factors, i.e., $O(n^m)$ with $n = \max_j n_j$.

In contrast, for the NFA by (57) on a Gaussian mixture, the integral is replaced by finding y_t via a nonlinear optimization by (50) and (51) (NFA-O) with its complexity being considerably less, in comparison of making the integral in getting $q(x)$. Moreover, we usually need only a few iterations by (50) instead of waiting it to converge.

We consider data sets from a model $x = Ay + e$ with y consisting of four, six, and eight factors, respectively. One half of factors are from uniform distribution (sub-gaussian) while the other half are from standard log-normal distribution (super-gaussian). Also, e is randomly generated from $G(e|0,0.04I)$, where I denotes a unit matrix.

With each Gaussian mixture by (47) consisting of three gaussian components, both the NFA-O and IFA work well. Shown in Fig. 6 are the recovered factors of a 4-factor model in comparison with the corresponding original sources, respectively. Shown in Fig. 7(a) are the MSEs between the recovered factors and the original factors. Moreover, shown in Fig. 7(b) are the corresponding time complexities by NFA-O and IFA. As the number m of factors increasing from 4 to 6, the time used by NFA-O increases from 36.63 s to 58.86 s (about $(6/4) = 1.5$ times) while that consumed by IFA increases from 76.25 s to 646.4 s (about $3^{(6-4)}$ times). We get a similar situation as m increases from 6 to 8. That is, we found empirically that the time complexity of NFA-O increases linearly with the number m of factors while IFA increases exponentially with m. In other words, with a similar or even improved performance, NFA-O outperforms IFA significantly in the aspect of computing complexity.

Furthermore, on a set of data that consists of 50 samples of five-dimension generated from two sources of uniform distributions (sub-gaussian) and two sources of log Gaussian distribution (super-gaussian), with noise e generated from a gaussian $G(e|0,0.01*I)$. We perform the NFA learning by (57) on the Gaussian mixture case with $h = 0$ and m increased from 1 to 6. Shown in Fig. 8(a) is the obtained $J(m)$ given by Eqn.2.(c) in Fig. 2. We observe that the minimum corresponds to 4. That is, the correct number of factors has been detected. In contrast, the number m of factors has to be pregiven for the IFA learning [6], [31].

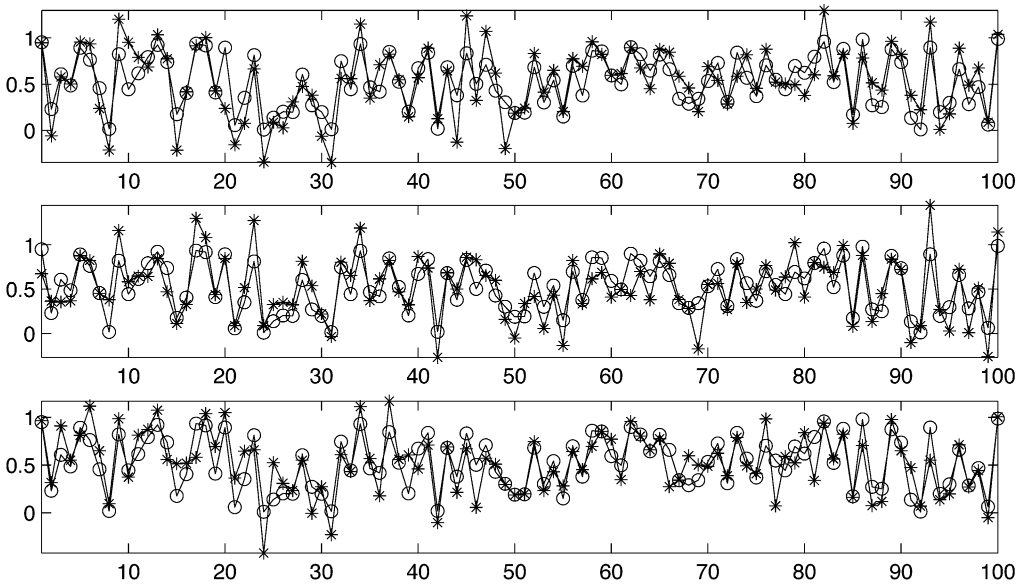
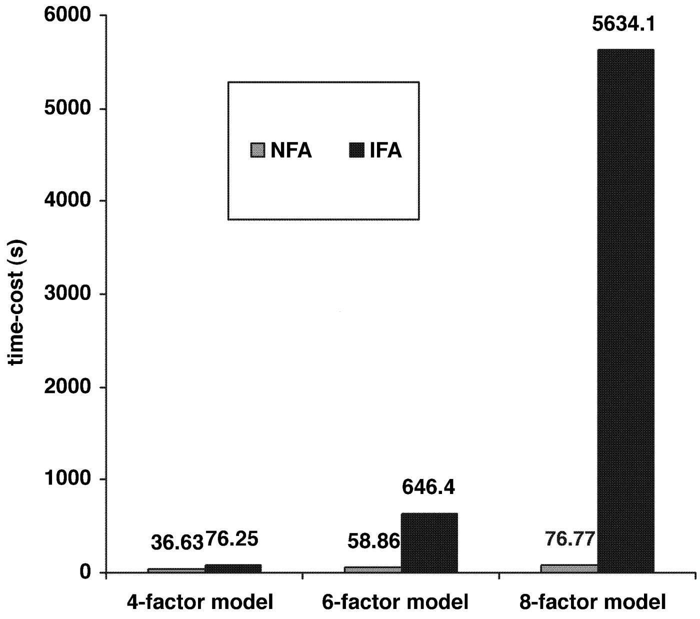


Fig. 6. Snapshots of one of the four factors, with "$*$" for the recovered and "o" for the original. The top for IFA, the middle for NFA by (57) via optimization in help of (50) and (51), (NFA-O); and the bottom for NFA by (57) via (71) analytically in help of (70) and (72)(NFA-A).

Factor	NFA-A	NFA-O	IFA
1	.0369	.0379	.0426
2	.0401	.0299	.0292
3	.0446	.0336	.0400
4	.0189	.0245	.0344
Mean	.0351	.0315	.0366

(a)



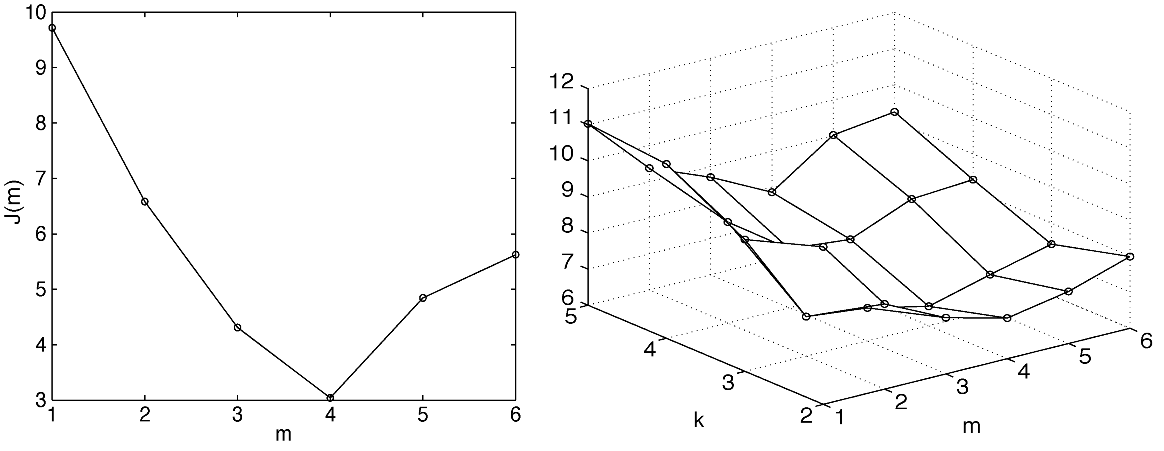
(b)

Fig. 7. Comparisons between NFA and IFA. (a) On the MSEs between the recovered factors and the original factors. (b) On time complexity.

Also, we perform the NFA learning by (66) on the Gaussian mixture case with $h = 0$ and $k_j = 3$ for all j, for parameter learning with automatic model selection. Initially, we set $D^2 = \text{diag}[6.99, .89, .62, .18, .12]$. As learning tends to converged, we get $D^2 = diag[7.84, .85, .42, .36, .002]$) such that a correct number of 4 factors has been automatically determined during learning.

For the NFA learning by (57) with $q(y)$ given by (3) and (47) as in Section IV-A, each k_j has to be known and prefixed, which however is usually difficult to know in advance. We can also select appropriate $\mathbf{m} = \{m, \{k_j\}\}$ by (7) with $J(\mathbf{m})$ given by Eqn.2.(c) in Fig. 2. Illustrated in Fig. 8(b) is an example that is an counterpart of Fig. 8(a), under the setting $h = 0$ and $k = k_1 = k_2 = \ldots = k_m$ for simplicity. It can be observed that the minimum is correctly found at $m = 4$ and $k = 3$. If we do not impose $k = k_1 = k_2 = \ldots = k_m$, the cost of searching a minimum of $J(m, \{k_j\})$ will be very expensive and also increases exponentially with K_R, where K_R is the maximum upper bound that every k_j has to be enumerated.

The problem is tackled by the NFA learning by (57) with $q(y, \ell)$ given by (67) and (69) as in Section IV-C. In implementation, the process of finding y_t be made via (71) analytically in help of (70) and (72) (NFA-A). To get an insight on this replacement, shown in Fig. 6 are the results of NFA-A in a comparison with the results of NFA-O. It can be observed that a similar or slightly improved performance is achieved by NFA-A.



(a) (b)

Fig. 8. Model selection via criteria. (a) On the factor number; (b) on both the number of factors and the number of Gaussian components for each factor.

Learning has also be implemented by (66) with automatic model selection. We initialize $m = 5$ and $k_j = 4,\ j = 1,\ldots,5$ with the initial $D^2 = \mathrm{diag}[8.37, 0.86, 0.40, 0.21, 0.11]$. As the learning converged, we get $D^2 = \mathrm{diag}[7.96, 0.87, 0.38, 0.34, 0.001]$ such that the number m is automatically determined as 4, during which k_1, k_2, and k_4 are automatically determined as 3 by observing one β_{ji} tending to 0 constantly while k_3 is automatically determined as 2 with two β_{3_i} tending to 0 during the learning. The results are consistent with that obtained in Fig. 8(b).

VI. Conclusion

The ability of the BYY harmony learning for model selection and regularization is reexamined from both an information theoretic perspective and a generalized projection geometry perspective. Comparative discussions are made on its relations and differences from the studies on MML/MDL, Bayesian approach, the bit-back based MDL, maximum likelihood, AIC, information geometry, Helmholtz machine, and variational approximation. Moreover, new algorithms are proposed for implementing Gaussian FA and non-Gaussian FA, such that appropriate factors is determined during learning.

Acknowledgment

The author would like to thank Z. Y. Liu for experiments.

References

[1] H. Akaike, "New look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. 19, pp. 714–723, Dec. 1974.

[2] S. Amari, *Differential Geometry Methods in Statistics*. New York: Springer-Verlag, 1985, Lecture Notes in Statistics 28.

[3] S. Amari and H. Nagaoka, *Methods of Information Geometry*. London, U.K.: Oxford Univ. Press, 2000.

[4] S. Amari *et al.*, "A new learning algorithm for blind separation of sources," in *Advances in Neural Information Processing 8*, D. S. Touretzky *et al.*, Eds. Cambridge, MA: MIT Press, 1996, pp. 757–763.

[5] T. W. Anderson and H. Rubin, "Statistical inference in factor analysis," in *Proc. 3rd Berkeley Symp. Mathematical Statistical Problems*, Berkeley, CA, 1956, pp. 111–150.

[6] H. Attias, "Independent factor analysis," *Neural Computat.*, vol. 11, pp. 803–851, 1999.

[7] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind de-convolution," *Neural Computat.*, vol. 7, pp. 1129–1159, 1995.

[8] H. Bozdogan, "Model selection and Akaike's information criterion: the general theory and its analytical extension," *Psychometrika*, vol. 52, pp. 345–370, 1987.

[9] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. New York: Wiley, 1997.

[10] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis., Graph. Image Process.*, vol. 37, pp. 54–115.

[11] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, no. 1, pp. 205–237, 1984.

[12] P. Dayan *et al.*, "The Helmholtz machine," *Neural Computat.*, vol. 7, pp. 889–904, 1995.

[13] P. Dayan and G. E. Hinton, "Varieties of Helmholtz machine," *Neural Netw.*, vol. 9, pp. 1385–1403, 1996.

[14] A. P. Dempster *et al.*, "Maximum- likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

[15] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[16] C. Fyfe, "Introducing asymmetry into interneuron learning," *Neural Computat.*, vol. 7, pp. 1167–1181, 1995.

[17] G. E. Hinton *et al.*, "The wake-sleep algorithm for unsupervised learning neural networks," *Science*, pp. 1158–1160, 1995.

[18] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," *Advances NIPS*, vol. 6, pp. 3–10, 1994.

[19] G. E. Hinton and D. van Camp, "Keeping neural networks simple by minimizing the description length of the weights," in *Proc. 6th ACM Conf. Computational Learning Theory*, Santa Cruz, CA, July 1993.

[20] R. A. Jacobs *et al.*, "Adaptive mixtures of local experts," *Neural Computat.*, vol. 3, pp. 79–87, 1991.

[21] H. Jeffreys, *Theory of Probability*. Oxford, U.K.: Clarendon Press, 1939.

[22] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computat.*, vol. 6, pp. 181–214, 1994.

[23] M. I. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts," *Neural Netw.*, vol. 8, pp. 1409–1431, 1995.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, pp. 35–45, Mar. 1960.

[25] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Netw.*, vol. 7, pp. 113–127, 1994.

[26] M. Kawamoto, "Cerebellum and motor cobtrol," in *The Handbook of Brain Theory and Neural Networks*, 2nd ed, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 2002, pp. 190–195.

[27] P. Kontkanen *et al.*, "Bayesian and information-theoretic priors for Bayeisan network parameters," in *Machine Learning: ECML-98*. New York: Springer-Verlag, 1998, vol. 1398, Lecture Notes in AI, pp. 89–94.

[28] D. Mackey, "A practical Bayesian framework for backpropagation," *Neural Computat.*, vol. 4, pp. 448–472, 1992.

[29] R. McDonald, *Factor Analysis and Related Techniques*. Hillsdale, NJ: Lawrence Erlbaum, 1985.

[30] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[31] E. Moulines, J. Cardoso, and E. Gassiat, "Maximum likelihood for blind separation and deconvolution of noisy signals using mixturemodels," in *Proc. ICASSP '97*, Munich, Germany, Apr. 1997, pp. 3617–3620.

[32] A. A. Neath and J. E. Cavanaugh, "Regression and time series model selection using variants of the Schwarz information criterion," *Commun. Statist. A*, vol. 26, pp. 559–580, 1997.

[33] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[34] J. Rissanen, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, pp. 1080–1100, 1986.

[35] D. Rubi and D. Thayer, "EM algorithm for ML factor analysis," *Psychometrika*, vol. 57, pp. 69–76, 1976.

[36] M. Sato, "Online model selection based on the vairational Bayes," *Neural Computat.*, vol. 13, pp. 1649–1681, 2001.

[37] E. Saund, "A multiple cause mixture model for unsupervised learning," *Neural Computat*, vol. 7, pp. 51–71, 1995.

[38] L. Saul and M. I. Jordan, "Exploiting tractable structures in intractable networks," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1995, pp. 486–492.

[39] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.

[40] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*: V.H. Winston and Sons., 1977.

[41] C. S. Wallace and D. M. Boulton, "An information measure for classification," *Comput. J.*, vol. 11, pp. 185–194, 1968.

[42] C. S. Wallace and D. R. Dowe, "Minimum message length and Kolmogorov complexity," *Comput. J.*, vol. 42, pp. 270–280, 1999.

[43] L. Xu, "Temporal BYY encoding, markovian state spaces, and space dimension determination," *IEEE Trans. Neural Networks*, 2004.

[44] ——, "Bayesian Ying Yang Learning (I): A Unified Perspective for Statistical Modeling," in *Intelligent Technologies for Information Analysis*, N. Zhong and J. Liu, Eds. New York: Springer-Verlag, 2004, pp. 607–652.

[45] ——, "BYY learning, regularized implementation, and model selection on modular networks with one hidden layer of binary units," *Neurocomput.*, vol. 51, pp. 227–301, 2003.

[46] ——, "Data smoothing regularization, multi-sets-learning, and problem solving strategies," *Neural Netw.*, vol. 15, pp. 817–825, 2003.

[47] ——, "Independent component analysis and extensions with noise and time: a Bayesian Ying-Yang learning perspective," *Neural Inform. Processing Lett. Rev.*, vol. 1, pp. 1–52, 2003.

[48] ——, "BYY harmony learning, structural RPCL, and topological self-organizing on mixture models," *Neural Netw.*, vol. 15, pp. 1125–1151, 2002.

[49] ——, "BYY harmony learning, independent state space and generalized apt financial analyzes," *IEEE Trans. Neural Networks*, vol. 12, pp. 822–849, July 2001.

[50] ——, "Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on gaussian mixtures, ME-RBF models and three-layer nets," *Int. J. Neural Syst.*, vol. 11, pp. 3–69, 2001.

[51] ——, "BYY harmony learning, model selection, and information approach: further results," in *Proc. 2001 Int. Conf. Neural Information Processing (ICONIP)*, vol. I, Shanghai, Nov. 14–18, 2001, pp. 30–37.

[52] ——, "Temporal BYY learning for state space approach, hidden Markov model and blind source separation," *IEEE Trans. on Signal Processing*, vol. 48, pp. 2132–2144, 2000.

[53] ——, "Bayesian Kullback Ying-Yang dependence reduction theory," *Neurocomput.*, vol. 22, pp. 81–112, 1998.

[54] ——, "RBF nets, mixture experts, and Bayesian Ying-Yang learning," *Neurocomput.*, vol. 19, pp. 223–257, 1998.

[55] ——, "Bayesian Ying-Yang learning theory for data dimension reduction and determination," *J. Computat. Intell. Finance*, vol. 6, pp. 6–18, 1998.

[56] L. Xu, C. C. Cheung, and S.-I. Amari, "Learned parametric mixture based ICA algorithm," *Neurocomput.*, vol. 22, pp. 69–80, 1998.

[57] L. Xu, "Bayesian Ying-Yang machine, clustering and number of clusters," *Pattern Recognit. Lett.*, vol. 18, pp. 1167–1178, 1997.

[58] ——, "Bayesian-Kullback Ying-Yang learning scheme: reviews and new results," in *Proc. 1996 Int. Conf. Neural Information Processing (ICONIP)*, vol. 1, Hong Kong, China, Sept. 24–27, 1996, pp. 59–67.

[59] ——, "Bayesian-Kullback Coupled YING-YANG Machines: Unified Learning and New Results on Vector Quatization," in *Proc. Int. Conf. Neural Information Processing (ICONIP96)*, Oct.–Nov. 1995, pp. 977–988.

[60] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," in *Advances in Neural Information Processing Systems 7*, J. D. Cowan, Ed. Cambridge, MA: MIT Press, 1995, pp. 633–640.

[61] L. Xu, A. Krzyzak, and E. Oja, "Rival penalized competitive learning for clustering analysis, RBF net and curve detection," *IEEE Trans. Neural Networks*, vol. 4, pp. 636–649, July 1993.

[62] L. Xu, "Least mean square error reconstruction for self-organizing neural-nets," *Neural Netw.*, pp. 627–648, 1993.

Lei Xu (SM'94–F'01) received the Ph.D. degree from Tsinghua University, Beijing, China, in 1987.

He is a Chair Professor with the Department of Computer Science and Engineering, the Chinese University of Hong Kong (CUHK), Hong Kong. He joined the National Key Lab on Machine Perception, Peking University, Beijing, China, in 1987, where he became one of ten university-level exceptionally promoted young Associate Professors in 1988 and was exceptionally promoted to a Full Professor in 1992. From 1989 to 1993, he worked at several universities in Finland, Canada, and the United States, including Harvard University and the Massachusetts Institute of Technology, both in Cambridge, MA. He joined CUHK in 1993 as a Senior Lecturer, became a Professor in 1996 and then took the current Chair Professor position in 2002. He has published over 100 academic journal papers, with several well cited contributions to pattern recognition and statistical learning for neural networks. He has given a number of keynote/plenary/invited/tutorial talks in international major neural networks (NN) conferences, such as WCNN, IEEE-ICNN, IJCNN, ICONIP, etc.

Prof. Xu is one of the past Governors of the International Neural Networks Society, a past President of Asia-Pacific NN Assembly, a past Chair of the Computational Finance Technical Committee of the IEEE NN Society, and an Associate Editor for six international journals on NN, including Neural Networks and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1994 to 1998. He was an ICONIP'96 Program Committee Chair, a Joint-ICANN-ICONIP'03 Program Committee Co-Chair and a General Chair of IDEAL'98, IDEAL'00, and IEEE CIFER'03. He has served as one of the program committee members in international major NN conferences over the past decade, including the International Joint Conference on Neural Networks, the World Conference on Neural Networks, and the IEEE-International Conference on Neural Networks. He has received several Chinese national prestigious academic awards, including the National Nature Science Prize, as well as international awards, including the 1995 INNS Leadership Award. He is a Fellow of the International Association on Pattern Recognition and a Member of the European Academy of Sciences.