Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach (II): From Unsupervised Learning to Supervised Learning and Temporal Modeling *

Lei Xu

Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China
Fax 852 2603 5024, Email lxu@cs.cuhk.hk, http://www.cse.cuhk.edu.hk/~lxu/

Invited paper, in K.M. Wong, I. King and D.Y. Yeung eds, **Theoretical Aspects of Neural Computation** : A Multidisciplinary Perspective (Hong Kong International Workshop TANC'97), Springer, pp25-42.

Abstract. A unified statistical learning approach called *Bayesian Ying-Yang (BYY) system and theory* has been developed by the present author in recent years. In a sister paper [1], this BYY system and theory has been shown to function as a general theory for unsupervised learning and its extension called semi-unsupervised learning, such that not only several existing popular unsupervised learning approaches are included as special cases, but also a number of new theories and models are provided for unsupervised pattern recognition and cluster analysis, factorial encoding, data dimension reduction, and independent component analysis. In this paper, the basic system and theory in [1] is further theoretically justified and extended into a general system and theory with a general implementation technique such that not only those results [1] are kept as special cases still, but also it works for supervised learning and temporal modeling on *parameter learning, regularization, structural scale or complexity selection, and architecture design*. Particularly, temporal modeling and regression based on Hidden Markov Model (HMM) and the linear and nonlinear state space model are discussed in detail, with an adaptive algorithm proposed for various specific variants of HMM model and state space models. Moreover, the criteria for deciding the number of hidden states in HMM and the order of state space are also proposed. In another sister paper [2] of this proceeding, several specific models and algorithms as well as model selection criteria will be given for dependence reduction, data dimension reduction, independent component analysis, supervised classification and regression. In addition, the

* Supported by the HK RGC Earmarked Grants CUHK250/94E and CUHK 339/96E and by Ho Sin-Hang Education Endowment Fund for Project HSH 95/02. The basic ideas of the BYY learning in my previous papers started the first year of my returning to HK. As HK in transition to China, this work was in transition to its current shape. The preliminary version of this paper and its sister papers [1, 2] are all completed in the first month that HK returned to China and thus I formally returned to my motherland as well. I would like to use this work as a memory of this historic event.

relation of the BYY learning system and theory to a number of existing learning models and theories has been discussed in [1].

1 Basic Bayesian Ying-Yang System and Theory

As shown in Fig.1, the BYY system consists of seven components. The first four components form the core. The other three surrounding components are added for the purposes of supervised learning. The core itself functions as a general framework for unsupervised learning, as shown in [1].

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

$$p_{M_1} = p_{M_1}(x, y) = p_{M_{y|x}}(y|x)p_{M_x}(x), \quad p_{M_2} = p_{M_2}(x, y) = p_{M_{x|y}}(x|y)p_{M_y}(y). \quad (1)$$

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

The task of specifying a Ying-Yang system is called *learning* in a broad sense, which consists of the following four levels of specifications:

Item 1.1 According to the nature of the perception task, the *Representation Domain Y and Its Complexity k* are designed. For example, we have either $y \in R^k$ or a binary vector $y = [y^{(1)}, \cdots, y^{(k)}]^T, y^{(j)} \in \{0, 1\}$.

Item 1.2 Based on the given set of training samples, some previous knowledge, assumption and heuristics, *Architecture Design* is made by specifying the architectures of four components $p_{M_x}(x)$, $p_{M_{y|x}}(y|x)$, $p_{M_{x|y}}(x|y)$ and $p_{M_y}(y)$. First, with a given set $D_x = \{x_i\}_{i=1}^N$ from an original density $p(x)$, $p_{M_x}(x)$ is fixed at some parametric or nonparametric empirical density estimation of $p(x)$,

[2] It should be "Yin" in the Mainland Chinese spelling system. However, I prefer to use "Ying" for the beauty of symmetry. Furthermore, strictly speaking we should use $P(u)$ to replace $p(u)$ when the corresponding random variable u is discrete. However, we simply use $p(.)$ for both the cases. Readers may identify the difference according to whether the involved variable is real or discrete.

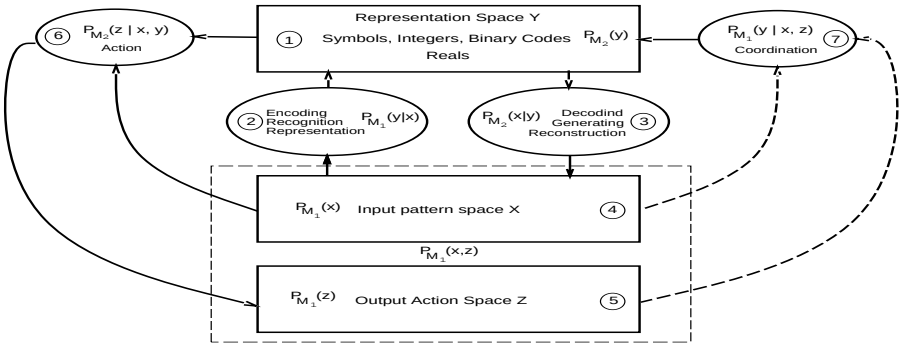

Fig. 1. The Bayesian YING-YANG System

e.g., $p_{M_x}(x) = p_{h_x}(x)$ given by a kernel estimate:

$$p_{h_x}(x) = \frac{1}{\#D_x} \sum_{x_i \in D_x} K_{h_x}(x - x_i),\ K_{h_x}(r) = \frac{1}{h_x^d} K(\frac{r}{h_x}), \tag{2}$$

with a prefixed kernel function $K(.)$ and a prefixed smoothing parameter h_x. Next, for the other three components, each $p_{M_a}(a), a \in \{x|y,\ y|x,\ y\}$ can be designed in two ways. One is called *Free*. It implies a totally unspecified density or probability function in the form $p(a)$ without any constraint. Thus, it is free to change such that it can be indirectly specified through other components. The other is called *Parameterized Architecture*. It means that $p_{M_a}(a), a \in \{x|y,\ y|x,\ y\}$ is either a simple parametric density, e.g., a Gaussian $p_{M_{x|y}}(x|y) = G(x, m_{x|y}, \Sigma_{x|y})$ with mean $m_{x|y}$ and variance matrix $\Sigma_{x|y}$, or a compounded parametric density with some of its parameters defined by a complicated function with a given parametric architecture consisting of a number of elementary units that are organized in a given structure. Taking a three layer perceptron as an example, we have

$$S(x, W) = [m_{y_1}, \cdots, m_{y_{k_r}}], \quad m_{y_i} = s(\sum_{j=1}^{k^b} w_{i,j}^{(2)} s(x^T W_j^{(1)} + w_{j,0}^{(1)})), \tag{3}$$

as $m_{y|x}$ in $p_{M_{y|x}}(y|x) = G(y, m_{y|x}, \sigma^2 I)$, with $W = \{w_{i,j}^{(2)}, W_j^{(1)}, w_{j,0}^{(1)}\}$, $s(u)$ is a sigmoid function, and k^b represents the scale or complexity of the parametric architecture. Therefore, the design of a parameterized architecture consists of

(i) Specification of density function form $p_a(a)$, For example, we have $p_{M_{y|x}}(y|x) = G(y, S(x, W), \sigma^2 I)$ for eq.(3);

(ii) Specification of one or several types of elementary units in a fixed basic structure with a complexity k_a^b. For example, it can be a simple sigmoid neuron or a gaussian unit with k_a^b ignored, or it can be a m_{y_i} given by eq.(3) with a complexity k_a^b;

(iii) Specification of a structure on how to organize those elementary units into the architecture. For example, by the cascade organization of sigmoid neurons, we can get a three layer perceptron eq.(3).

Item 1.3 We also need to select the set of scale parameters $k = \{k_r, q_r, \{k_a^b\}\}$ with each element defined as above, which is called *Structural Scale Selection* or *Model Selection*.

Item 1.4 After the above three levels of specifications, the unspecified part for each component $p_{M_a}(a)$, $a \in \{x|y,\ y|x,\ y\}$ is a set θ_a of parameters in certain domains. Putting them together, we get the parameter set $\Theta = \{\theta_{x|y}, \theta_{y|x}, \theta_y\}$, which we call *Parameter Learning*. In the literature, this task is also often simply called *learning* in a narrow sense.

Our basic theory is that the specifications of an entire Ying-Yang system p_{M_1}, p_{M_2} by eq.(1) in the above four levels best enhances the so called *Ying-Yang Harmony or Marry*, through minimizing a harmony measure called *separation functional*:

$$F_s(M_1, M_2) = F_s(p_{M_1}, p_{M_2}) \geq 0,\ F_s(M_1, M_2) = 0,\ iff\ p_{M_1} = p_{M_2}, \tag{4}$$

which describes the harmonic degree of the Ying-Yang pair. Such a learning theory is called *Bayesian Ying-Yang (BYY) Learning Theory*. As shown in [1], this theory functions as a general theory for unsupervised learning and its semi-unsupervised extension. One important feature is that it provides a new perspective, i.e., the interaction of learning on the complement Ying and Yang structures, for tackling the learning and model selection problems on a training set of finite samples with an improved generalization, as will be further addressed in Sec.2, especially by Item 2.2 and Item 2.4.

Three categories of separation functionals, namely *Convex Divergence, L_p Divergence, and De-correlation Index*, have been suggested [1]. Particularly, the *Convex Divergence* is defined as [3]

$$\begin{aligned} &F_s(M_1, M_2) = CV(p_{M_1}||p_{M_2}), \quad F_s(M_2, M_1) = CV(p_{M_2}||p_{M_1}), \\ &CV(p||q) = f(1) - \int p(u) f(q(u)/p(u)) du, \quad f(v) \text{ is strictly convex on } (0, +\infty). \end{aligned} \tag{5}$$

In this case, the BYY learning is called Bayesian Convex YING-YANG (BCYY) learning. Particularly, when $f(u) = \ln u$, eq.(5) becomes the well known Kullback Divergence:

$$\begin{aligned} &F_s(M_1, M_2) = KL(p_{M_1}||p_{M_2}),\ or\ F_s(M_2, M_1) = KL(p_{M_2}||p_{M_1}), \\ &KL(p||q) = \int p(u) \ln \frac{p(u)}{q(u)} du. \end{aligned} \tag{6}$$

In this special case, the BYY learning is called Bayesian-Kullback YING-YANG (BKYY) learning. It should be noted that the asymmetry feature of $CV(p||q)$ and $KL(p||q)$ and thus BCYY and BKYY learning have two different variants. In papers [1, 2] and this paper, we focus on the case of $F_s(M_1, M_2), KL(M_1, M_2)$ only. In [3], we will particularly discuss $F_s(M_2, M_1)$ and $KL(M_2, M_1)$.

---

[3] For convenience, in this whole paper we adopt such a convention that (a) $\int_u$ denotes the integral operation when u is known to be a real ; (b) in general $\int_u$ denotes either the integral operation for a real u or the summation operation for a discrete u; (c) we explicitly use $\sum_u$ if u is already known to be discrete.

The above proposed basic form of the BYY system and theory is mainly for unsupervised learning (i.e., knowing $D_x = \{x_i\}_{i=1}^N$) and its semi-unsupervised extension (i.e., knowing a hybrid data set $D_H = \{D_{x,y}, D_x\}$ with $D_{x,y} = \{x_i, y_i\}_{i=1}^{N'}$). However, for supervised learning, the above basic form covers only the ordinary maximum likelihood learning by $KL(M_2, M_1)$[5].

2 BYY Learning System and Theory

We further consider the complete system in Fig.1 with the last three components joined in. One is the *Output Action Space* $z \in Z$ with its distribution $p_{M_z}(z)$. The other is the *Action Terminal (AT)* described by a distribution $p_{M_{z|x,y}}(z|x,y)$ for the mapping $x \to z$ that is modulated by the internal representation y. The another one is the *Coordination Terminal (CT)* described by a distribution $p_{M_{y|x,z}}(y|x,z)$, which lets the invisible representation Y be in coordination with the two visible spaces X, Z. Here, with respect to the invisible space Y, the model $p_{M_1^L}(x,z) = p_{M_{z|x}}(z|x)p_{M_x}(x)$ for the joint X, Z forms a large *Yang model*; while with respect to the visible output action space Z, the model $p_{M_2^L}(x,y) = p_{M_2}(x,y) = p_{M_{x|y}}(x|y)p_{M_y}(y)$ for the joint X, Y forms a large *Ying model*. The action terminal $p_{M_{z|x,y}}(z|x,y)$ is the passage from the large Ying space to the Yang space Z. The coordination terminal $p_{M_{y|x,z}}(y|x,z)$ is the passage from the large Yang space to the Ying space Y. Thus, we have another YING-YANG pair

$$\begin{aligned} p_{M_1^L} &= p_{M_1^L}(x,y,z) = p_{M_{y|x,z}}(y|x,z)p_{M_{z|x}}(z|x)p_{M_x}(x), \\ p_{M_2^L} &= p_{M_2^L}(x,y,z) = p_{M_{z|x,y}}(z|x,y)p_{M_2^L}(x,y) = p_{M_{z|x,y}}(z|x,y)p_{M_{x|y}}(x|y)p_{M_y}(y), \end{aligned} \tag{7}$$

with the old Ying-Yang pair in eq.(1) to form an enlarged Ying-Yang system.

Therefore, in addition to specify the old Ying-Yang pair, we need also to specify the new Ying-Yang pair. First, $p_{M_1^L}(z,x)$ is specified via $D_{x,z} = \{x_i, z_i\}_{i=1}^N$ such that $p_{M_x}(x) = p_{h_x}(x)$ is still the same as before and $p_{M_{z|x}}(z|x) = p_{h_z}(z|x)$ for a pair $(x,z') \in D_{x,z}$:

$$p_{h_z}(z|x) = \begin{cases} \delta_d(z-z'), & z \text{ is discrete}, \\ \frac{1}{h_z^{d_z}} K(\frac{z-z'}{h_z}), & z \text{ is real}, \end{cases}, \quad \delta_d(z) = \begin{cases} 1 & \text{for } z = 0, \\ 0, & \text{for } z \neq 0. \end{cases} \tag{8}$$

So, only two new components $p_{M_{y|x,z}}(y|x,z)$ and $p_{M_{z|x,y}}(z|x,y)$ need to join the previously discussed core for being specified through *Architecture Design*, *Structural Scale Selection*, and *Parameter Learning*.

Again, our basic theory is that all the specifications should best enhance the *Ying-Yang Harmony* for both the Ying-Yang pairs, through minimizing:

$$\begin{aligned} & F_{two}(M_1, M_2) = F_s(p_{M_1}, p_{M_2}) + F_s(p_{M_1^L}, p_{M_2^L}), \\ \textit{Particularly}, \quad & KL_{two}(M_1, M_2) = KL(p_{M_1}||p_{M_2}) + KL(p_{M_1^L}||p_{M_2^L}), \\ \textit{or} \quad & KL_{two}(M_2, M_1) = KL(p_{M_2}||p_{M_1}) + KL(p_{M_2^L}||p_{M_1^L}). \end{aligned} \tag{9}$$

It can be noticed that eq.(9) will degenerate into eq.(6) when $z = x$ which makes $KL(p_{M_2^L}, p_{M_1^L}) = KL(p_{M_2}, p_{M_1})$ and $KL_{two}(M_1, M_2) = 2KL(M_1, M_2)$. In other words, the extended BKYY learning indeed includes the basic BKYY

learning eq.(6) as a special case. For some other separation functionals, we also have $F_{two}(M_1, M_2) = 2F_s(M_1, M_2)$ when $z = x$.

We use S_a, k_a and θ_a to denote respectively the architecture, structural scale, and parameters of each component $p_{M_a}(a), a \in \{x|y,\ y|x,\ y,\ (y|x,z), z|x,y)\}$. Putting them together, we have the following parts to be specified:

$$\begin{aligned} S &= \{S_{x|y}, S_{y|x}, S_y, S_{z|x,y}, S_{y|x,z}\}, \quad \textit{for Architecture Design}, \\ k &= \{k_r, k^b_{x|y}, k^b_{y|x}, k^b_y, k^b_{z|x,y}, k^b_{y|x,z}\}, \quad \textit{for Structural Scale Selection}, \\ \Theta &= \{\theta_{x|y}, \theta_{y|x}, \theta_y, \theta_{z|x,y}, \theta_{y|x,z}\}, \quad \textit{for Parameter Learning}. \end{aligned} \tag{10}$$

It should be noticed that this is the most general notations. A part of these parameters can be prespecified in many specific cases.

For BKYY learning, we also often use the following decompositions:

$$\begin{aligned} &KL(p_{M_1}||p_{M_2}) = Kl(p_{M_1}||p_{M_2}) - H(p_{M_x}(x)), \quad H(p) = -\int p(u)\ln p(u)du, \\ &Kl(p_{M_1}||p_{M_2}) = H_C(p_{M_1}||p_{M_2}) - E_{p_{M_x}(x)}[H(p_{M_{y|x}}(y|x)], \\ &H_c(p||q) = -\int p(u)\ln q(u)du, \quad E_{p(u)}[g(u)] = \int p(u)g(u)du, \\ &KL(p_{M_1^L}||p_{M_2^L}) = Kl(p_{M_1^L}||p_{M_2^L}) - H(p_{M_1^L}(z,x)), \\ &Kl(p_{M_1^L}||p_{M_2^L}) = H_C(p_{M_1^L}||p_{M_2^L}) - E_{p_{M_1^L}(z,x)}[H(p_{M_{y|x,z}}(y|x,z)]. \end{aligned} \tag{11}$$

Finally, we summarize the major roles of the BYY learning theory as follows:

Item 2.1 Parameter estimation or learning. That is, we determine

$$\Theta^* = arg\min_{\Theta} F(\Theta, S, k), \quad \text{given } S \text{ and } k \text{ fixed}, \tag{12}$$

where $F(\Theta, S, k)$ denotes $F_s(p_{M_1}, p_{M_2})$ for eq.(4) and $F_{two}(M_1, M_2)$ for eq.(9), in the general cases, and denotes $KL(p_{M_1}||p_{M_2})$ for eq.(6) and $KL_{two}(M_1, M_2)$ for eq.(9) for BKYY learning. For the latter case, $KL(.||.)$ can also simply replaced by $Kl(.||.)$ given by eq.(11) since the terms $H(p_{M_x}(x))$, $H(p_{M_1^L}(z,x))$ are irrelevant to Θ, S, k.

Item 2.2 Structural scale selection, or model selection. We determine

$$\begin{aligned} &k^* = \min_k \mathcal{K}, \quad \mathcal{K} = \{j : J(j) = \min_k J(k)\}, \quad J(k) = \begin{cases} J_1(k), \\ J_2(k) \end{cases}, \text{ given } S; \\ &J_1(k) = F(\Theta^*, k, S), \quad J_2(k) = \begin{cases} H_C(p^*_{M_1}||p^*_{M_2}), & \text{for eq.(6)}, \\ H_C(p^*_{M_1^L}||p^*_{M_2^L}), & \text{for eq.(9)}; \end{cases} \end{aligned} \tag{13}$$

where "*" denotes the case with the parameter Θ^* given by eq.(12) with $F(\Theta, S, k)$ being the same as in eq.(12) too.

When k is small, the discrepancy of the learned Ying and Yang models can not cancel and thus $J_1(k)$ is high, as k increases to the best k^*, both the Ying and Yang models fit the samples well and the discrepancy of the learned two becomes the minimum and thus $J_1(k)$ reaches its minimum and then keeps this minimum thereafter.

From eq.(11), we can also get

$$H_C(p_{M_1}||p_{M_2}) = Kl(p_{M_1}||p_{M_2}) + E_{p_{M_x}(x)}[H(p_{M_{y|x}}(y|x)],$$

$$H_C(p_{M_1^L}||p_{M_2^L}) = Kl(p_{M_1^L}||p_{M_2^L}) + E_{p_{M_1^L}(z,x)}[H(p_{M_{y|x,z}}(y|x,z)]. \quad (14)$$

After ignoring the terms $H(p_{M_x}(x))$, $H(p_{M_1^L}(z,x))$ that are irrelevant to Θ, S, k, we see that $J_2(k)$ is obtained from $J_1(k)$ with an additional term — the conditional entropy or the uncertain complexity of the passage from the Yang space to Ying space. The more complicated the passage is, the larger the term. It is small when k is small, but the discrepancy of the learned Ying and Yang models will be large. When k is larger than the best k^*, the discrepancy will keep at its minimum and the complexity or the uncertainty due to the passage will be increase as k. Thus, the minimization of $J_2(k)$ select a model with the scale k^* that minimizes both the discrepancy due to an over-determined Ying-Yang pair and the uncertainty due to a under-determined passage from Yang to Ying.

We can also further justify $J_1(k), J_2(k)$ more formally as follows.

We first consider the following two points:

(1) We use $J_1^o(k), J_2^o(k)$ to denote the limit of $J_1(k), J_2(k)$ by letting $p_{M_x}(x)$ to be replaced by $p^o(x, \theta_x^o, k^o)$ for $F = F_s(M_1, M_2)$ given by eq.(4) or $p_{M_1^L}(z,x)$ to be replaced by $p^o(x, z, \theta_{x,z}^o, k^o)$ for $F = F_{two}(M_1, M_2)$ given by eq.(9), where x comes from $p^o(x, \theta_x^o, k^o)$ with it structural scale being k^o or the pair z, x comes from $p^o(x, z, \theta_{x,z}^o, k^o)$. This limit, i.e., the consistency of $J_1(k), J_2(k)$, can be reached as the sample number $N \to \infty$ under the condition that the estimate $p_{M_x}(x), p_{M_1^L}(z,x)$ are consistent (e.g., given by eq.(2) and eq.(8)) and that the densities in the Ying and Yang parts satisfy some mild regular condition.

(2) We use $\mathcal{D}_k$ to denote the domain of $\Theta_k = \Theta$ at k. We consider a class of so called *incremental architectures*. That is, as the scale increases from $k-1$ to k, from $\Theta_k = \{\Theta_{k-1}, \Phi_k\}$ with Φ_k being not empty and also not a subset of Θ_{k-1} we have $\mathcal{D}_{k-1} \subset \mathcal{D}_k$ since the case of $k-1$ can be regarded as a special case of k where at least some parameters in Φ_k are fixed at their specific values, e.g., for a finite mixture $p(x, \Theta_k) = \sum_{j=1}^k \alpha_j p(x|\theta_j)$, discussed in [1], $\mathcal{D}_{k-1}$ can be regarded as a subset of $\mathcal{D}_k$ with one fixed $\alpha_j = 0$.

Based on the two points, we can prove that:

Theorem 2.1 For a Ying -Yang pair with incremental architecture, $J_1^o(k) > J_1^o(k^o)$ for $k < k^o$ and $J_1^o(k) = J_1^o(k^o)$ for $k \geq k^o$. Moreover, as k increases from 1 to k^o, $J_1^o(k)$ is monotonically non-increasing and reaches its minimum at $J_1^o(k^o)$.

Proof θ_x^o or $\theta_{x,z}^o$ is only contained in $\mathcal{D}_k, k \geq k^o$ and $\mathcal{D}_1 \subset \cdots \subset \mathcal{D}_{k^o-1} \subset \mathcal{D}_{k^o}$, thus the minimization eq.(12) is made on the domain that becomes bigger and bigger until it contains θ_x^o or $\theta_{x,z}^o$ after $k \geq k^o$. **Q.E.D.**

The theorem justifies $J_1(k)$ for the cases of a large sample number N. For the case of finite samples, as $k > k^o$ increases, both the Ying and Yang models further enhance their fits to the finite samples since they have excess freedom, $J_1(k)$ may still decrease slowly as k passes k^o and some modification can be added to help the detection of k^o (e.g., by hypothesis testing).

We further look into $J_2^o(k)$. For $k < k^o$, it is actually a measure that decreases as the discrepancy between the pair $p_{M_1}^*, p_{M_2}^*$ or the pair $p_{M_1^L}^*, p_{M_2^L}^*$ reduces and reaches its minimum at $k = k^o$ with $p_{M_1}^* = p_{M_2}^*$ or $p_{M_1^L}^* = p_{M_2^L}^*$. After $k \geq k^o$, this equality will keep and thus $J_2^o(k)$ actually becomes the entropy or the uncertain

complexity of $p^*_{M_2}$ or $p^*_{M_2^L}$.

For an incremental architecture, from $\Theta_k = \{\Theta_{k^o}, \Phi_k\}$ we see that $p^*_{M_1} = p^*_{M_2}$ or $p^*_{M_1^L} = p^*_{M_2^L}$ means the fact that only Θ_{k^o} is independently optimized in eq.(12) and actually the parameters in Φ_k are dependently fixed at their specific values by the parameters in Θ_{k^o}. For the cases with k and k^o produce the equivalent performances, we like to select the smallest k^o for saving the complexity. For example, for a finite mixture $p^o(x, \theta^o_x, k^o) = \sum_{j=1}^{k^o} \alpha^o_j p(x|\theta^o_j)$, we have $\sum_{j=1}^{k} \alpha_j p(x|\theta_j) = \sum_{j=1}^{k^o} \alpha^o_j p(x|\theta^o_j)$ for $k > k^o$, which can be true only when for each j there is at least one i such that $\theta_i = \theta^o_j$ and for some j there are more than one $i_1, \cdots i_m$ such that $\theta_{i_1} = \cdots = \theta_{i_m} = \theta^o_j$ with $\sum_{r=1}^{m} \alpha_{i_r} = \alpha^o_j$. In this case, m moldes of $\alpha_{i_m} p(x|\theta^o_j)$ equivalently represents one $\alpha^o_j p(x|\theta^o_j)$, and the cases for k, k^o are equivalent. For some algorithms, the minimization eq.(12) forces Φ_k to take values independently on the whole domain $\mathcal{D}_k$ and deviate from the correct value, resulting in $J^o_2(k) > J^o_2(k^o)$ automatically. For example, for the above finite mixture with the constraint $\alpha_j = 1/k$ (e.g., using the well known K-means algorithm for clustering on gaussian mixture) or even simply each $\alpha_j > 0$, then we have $\sum_{j=1}^{k} \alpha_j p(x|\theta_j) \neq \sum_{j=1}^{k^o} \alpha^o_j p(x|\theta^o_j)$ for $k > k^o$, and thus $J^o_2(k) > J^o_2(k^o)$. This property makes $J_2(k)$ better in the case of finite samples to alleviate the problem that $J_1(k)$ may decrease slowly as k passes k^o.

Item 2.3 Architecture evaluation. Similar to eq.(13), we can also select a set of architecture $\mathcal{S} = \{S^{(i)}, i = 1, \cdots, N_s\}$ by $i^* = arg\min_i J(S^{(i)})$ with

$$J_1(S^{(i)}) = F(\Theta^*, S^{(i)}),\ J_2(S^{(i)}) = \begin{cases} H_C(p^*_{M_1}||p^*_{M_2}), & \text{for eq.(6)}, \\ H_C(p^*_{M_1^L}||p^*_{M_2^L}), & \text{for eq.(9)}. \end{cases} \quad (15)$$

Item 2.4 Regularization. The BYY learning theory can improve generalizations from the following aspects:

(a) Given finite samples and a structural scale k that may be too large with excess freedom. The key reason of poor generalization is that the modeling problem in such cases are under-determined with many models that can fit the finite samples well but more uncertain to fit future samples. When only one model is used for fitting the samples, the solution can be any one in an under-determined domain and thus may be far from the desired one in this domain. Hence, it may fit these samples well but perform poorly for a testing set; when we use both the Ying and Yang models to fit these samples, each is still under-determined. However, due to the complement structures of the two, each has a different under-determined domain. The learning eq.(12) aims at minimizing the discrepancy of the two that both fit these samples, therefore the under-determined domain reduces into the intersection of the original two under-determined domains, with the true solution kept in this intersection since the Ying and Yang are equal naturally at the true solution. This process equivalently regularizes the ill-posed learning and thus improves the generalization.

(b) The generalization can also be improved by the selection criteria given in Item 2.2 and Item 2.3.

(c) The generalization can also be obtained via selecting smooth parameters h_x, h_z in estimating $p_{M_x}(x)$ by eq.(2) and $p_{h_z}(z|x)$ by eq.(8) according to

$$\{h_x^*, h_z^*\} = arg \min_{h_x, h_z} J(h_x, h_z), \quad J(h_x, h_z) = \begin{cases} F(\Theta^*, k^*, S^{(i^*)}, h_x, h_z), \\ H_C(p^*_{M_1}||p^*_{M_2}), \\ H_C(p^*_{M_1^L}||p^*_{M_2^L}). \end{cases} \tag{16}$$

The above approaches are obviously different from the existing regularization or generalization error up-bound methods (e.g., VC dimension) that introduce an extra penalty term to the original error cost for being minimized together. They are also considerably different from the existing Bayesian approach that introduce a priori density on the parameters, at least in the two aspects:

Item 2.5 The BYY system and theory only bases on the two complement Bayesian representations, there is no use of a priori on the parameters. Instead, a priori can be embedded via the designs of the two complement architectures.

Item 2.6 In addition, the two Bayesian representations may not be equal, i.e., the Bayesian rule may not be exactly true but only approximately hold. Therefore, our approach should not be confused with the existing Bayesian approach. It is a new type of *Structural and Relaxed* Bayesian approach.

3 A General Technique for Implementation

Let F be either $F_s(M_1, M_2)$ in eq.(4) or $F_{two}(M_1, M_2)$ in eq.(9). $\min_{M_1, M_2} F$ can be implemented by alternatively repeating the following Step (a) and (b):

$$a.\ Fix\ M_2 = M_2^{old},\ M_1^{new} = arg \min_{M_1} F;\ b.\ Fix\ M_1 = M_1^{old},\ M_2^{new} = arg \min_{M_2} F, \tag{17}$$

which guarantees to reduce F until converged to one local minimum, where F is the same as discussed in eq.(12) and Item 2.1.

For BKYY learning, the integral operations in computing KL or Kl may be simplified into some implementable forms. For example, for discrete y with the design $p_{M_y}(y) = \alpha_y > 0,\ p_{M_{y|x}}(y|x) = p(y|x) \geq 0,\ p_{M_{x|y}}(x|y) = G(x, m_y, \Sigma_y)$, and $p_{M_x}(x) = p_{h_x}(x)$ by eq.(2), the above eq.(17) becomes the following *Smoothed EM* algorithm for gaussian mixtures:

$$\begin{aligned} E\ Step: &\quad get\ p^*(y|x_i) = \alpha_y G(x, m_y, \Sigma_y) / \sum_{y=1}^{k} \alpha_y G(x, m_y, \Sigma_y); \\ M\ Step: &\quad \alpha_y = \frac{1}{N} \sum_{i=1}^{N} p^*(y|x_i), \quad m_y^{new} = \frac{1}{\alpha_y N} \sum_{i=1}^{N} p^*(y|x_i) x_i, \\ &\quad \Sigma_y^{new} = h^2 I_d + \frac{1}{\alpha_y N} \sum_{i=1}^{N} p^*(y|x_i)(x_i - m_y^{new})(x_i - m_y^{new})^T. \end{aligned} \tag{18}$$

which is actually a smoothed variant of the well known EM algorithm for gaussian mixture, as discussed in details by [4]. Moreover, the number k can be selected by $J_1(k)$, $J_2(k)$ with the same detailed formula given in [4]. Furthermore, h can also be selected by eq.(16).

However, in the most general case, we will still encounter the implementation difficulty for dealing with these integral operations. Here, we propose a general stochastic sampling technique for implementation. First, we represent the separation functional by the following general form:

$$F(M_1, M_2) = \int_u f_{\{M_1,M_2\}}(u)du = \int_u p_r(u)\frac{1}{p_r(u)}f_{\{M_1,M_2\}}(u)du, \quad p_r(u) \neq 0, \quad (19)$$

where $p_r(u)$ is a given known smooth density function, called *Sampling Reference*, e.g, if the integral is made on a compact support S, $p_r(u)$ can be a uniform density on this S. Moreover, u is x, y for eq.(5) and eq.(6), x, y, z for eq.(9), and $f_{\{M_1,M_2\}}(u)$ is the part to be integrated in the integrals of eq.(5), eq.(6), and eq.(9). E.g.,

$$f_{\{M_1,M_2\}}(u) = \begin{cases} p_{M_{y|x}}(y|x)p_{M_x}(x)\ln\frac{p_{M_{y|x}}(y|x)p_{M_x}(x)}{p_{M_{x|y}}(x|y)p_{M_y}(y)}, \\ p_{M_{y|x,z}}(y|x,z)p_{M_1^L}(z,x)\ln\frac{p_{M_{y|x,z}}(y|x,z)p_{M_1^L}(z,x)}{p_{M_{z|x,y}}(z|x,y)p_{M_2^L}(x,y)}. \end{cases} \quad (20)$$

We make randomly sampling according $p_r(u)$ and get $\{u_t\}_{t=1}^N$, then use the empirical estimation $p(u) = \frac{1}{N}\sum_{t=1}^N \delta(u - u_t)$ in eq.(19) and get

$$F(M_1, M_2) = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{p_r(u_t)}f_{\{M_1,M_2\}}(u_t), \quad (21)$$

which is a stochastic approximation of eq.(19).

Since we already have some samples, e.g, $D_x = \{x_i\}_{i=1}^N$, $D_{x,z} = \{x_i, z_i\}_{i=1}^N$. We propose to use them in either one of the following two ways:

(a) Let $p_{M_x}(x)$ be given by eq.(2) and $p_{M_{z|x}}(z|x)$ by eq.(8) with $h_x \neq 0, h_z \neq 0$, and thus $p_{M_1^L}(z,x) = p_{M_x}(x)p_{M_{z|x}}(z|x)$. Then we directly use eq.(21).

(b) Let $p_r(u) = p_r(y)p_{M_x}(x)$ or $p_r(u) = p_r(y)p_{M_{z|x}}(z|x)p_{M_x}(x)$, make randomly sampling according to $p_r(y)$ and get $\{y_t\}_{t=1}^{N'}$, then simplify eq.(21) into

$$F(M_1, M_2) = \frac{1}{N}\sum_{t=1}^{N'}\sum_{\tau=1}^{N}\frac{1}{p_r(y_t)}f_{\{M_1,M_2\}}(y_t, u_\tau),$$

$$f_{\{M_1,M_2\}}(y,u) = \begin{cases} p_{M_{y|x}}(y|x)\ln\frac{p_{M_{y|x}}(y|x)}{p_{M_{x|y}}(x|y)p_{M_y}(y)}, & \text{for } u = x \\ p_{M_{y|x,z}}(y|x,z)\ln\frac{p_{M_{y|x,z}}(y|x,z)}{p_{M_{z|x,y}}(z|x,y)p_{M_2^L}(x,y)}, & \text{for } u = (x,z). \end{cases} \quad (22)$$

As long as N is large enough, we can implement $\min_{M_1,M_2} F(M_1, M_2)$ via the *Alternative Minimization* procedure eq.(17).

We can also simply adjust M_1, M_2 respectively by

$$M_1^{new} = M_1^{old} - \eta G_{M_1}/p_r(u_t), \quad M_2^{new} = M_2^{old} - \eta G_{M_2}/p_r(u_t), \quad (23)$$

once we get a sample u_t, where G_{M_1}, G_{M_2} are the gradient descent direction of $f_{\{M_1,M_2\}}(u_t)$ with respect to M_1, M_2 at M_1^{old}, M_2^{old} respectively. That is, we get a so called adaptive algorithm for implementation.

4 BYY Supervised Learning Family

The BYY system given in Fig. 1 and eq.(10), in its nature, applies to various tasks of $x \to z$ association type, including *classification, regression, function approximation, control action, ..., etc.* Supervised learning is used for learning this $x \to z$ association. As shown by Fig. 1 and eq.(10), we have five components to specify. There may be several choices for the specification of each one, which create a family that consists of quite a number of variants. Here, we propose a hierarchy to organize them and then concentrate on some of them for further introduction.

Item 3.1 *Bayesian Kullback Ying-Yang (BKYY) family* and *Bayesian Non-Kullback Ying-Yang (BNKYY) family* are obtained according to whether the Kullback divergence is used as the separation functional.

Item 3.2 We further divide BKYY learning into two big branches, according to whether $p_{M_{y|x,z}}(y|x,z)$ is free. If it is free such that it can be determined by $\min_{M_1,M_2} F_{two}(M_1,M_2)$ without any constraint, we call that the coordination terminal is of full capacity for coordination, and we call the corresponding BKYY learning as *Full Coordinated BKYY Learning*; otherwise if $p_{M_{y|x,z}}(y|x,z)$ comes from some parametric family with certain constraints, we call the corresponding learning as *Constrained Coordinated BKYY Learning*.

For the *Full Coordinated BKYY Learning*, we have the following theorem:

Theorem 3.1 *With* $p_{M_{y|x,z}}(y|x,z)$ *free, both* $\min_{p_{M_{y|x,z}}(y|x,z)} KL(p_{M_1^L}||p_{M_2^L})$ *and* $\min_{p_{M_{y|x,z}}(y|x,z)} KL_{two}(M_1,M_2) \Leftrightarrow p_{M_{y|x,z}}(y|x,z) = \frac{p_{M_{z|x,y}}(z|x,y)p_{M_2^L}(x,y)}{p_{M_2}(x,z)}$,
$p_{M_2}(x,z) = \int_y p_{M_{z|x,y}}(z|x,y)p_{M_2^L}(x,y)dy = \int_y p_{M_{z|x,y}}(z|x,y)p_{M_{x|y}}(x|y)p_{M_y}(y)dy$,
$\min_{p_{M_{y|x,z}}(y|x,z)} KL_{two}(M_1,M_2) = KL(p_{M_1^L}(z,x)||p_{M_2}(x,z)) + KL(p_{M_1}||p_{M_2})$.

This theorem basically tells us two points. One is that $p_{M_{y|x,z}}(y|x,z)$ will be fixed at the Bayesian posterior probability if it is free. The second is that after $p_{M_{y|x,z}}(y|x,z)$ settled, the specification of the remaining parts in $KL^L(M_1,M_2)$ only relates to a mixture joint density $p_{M_2}(x,z)$.

Item 3.3 The BKYY learning can be also described in different types, according to the relative status of the Ying $p_{M_2^L}(x,y) = p_{M_{x|y}}(x|y)p_{M_y}(y)$ and Yang $p_{M_1^L}(x,y) = p_{M_{y|x}}(y|x)p_{M_x}(x)$. If $\min_{p_{M_{y|x,z}}(y|x,z)} KL_{two}(M_1,M_2)$ gives

$$p_{M_1^L}(x,y) = p_{M_{y|x}}(y|x)p_{M_x}(x) = p_{M_{x|y}}(x|y)p_{M_y}(y) = p_{M_2^L}(x,y), \tag{24}$$

or if we force it to hold, we have $KL(p_{M_1}||p_{M_2})$. We call it *Fully Matched* BKYY learning. Otherwise, if we cannot have $KL(p_{M_1}||p_{M_2}) = 0$, it means that $p_{M_1^L}(x,y) \neq p_{M_2^L}(x,y)$ and thus both parts actually impose constraints on each other via $\min_{p_{M_{y|x,z}}(y|x,z)} KL_{two}(M_1,M_2)$, we call it *Partially Matched* BKYY learning. Here, we show three examples with $KL(p_{M_1}||p_{M_2}) = 0$. First, if the data set D_x comes from the true distribution

$$p^o(x,y,\Theta^o) = p^o(y|x,\theta^o_{y|x})p^o(x,\theta^o_x) = p^o(x|y,\theta^o_{x|y})p^o(y,\theta^o_y), \tag{25}$$

and we exactly put $p_{M_{x|y}}(x|y)$, $p_{M_y}(y)$ and $p_{M_{y|x}}(y|x)$ on their counterparts respectively. Second, $p_{M_{x|y}}(x|y)$ and $p_{M_y}(y)$ are both free such that the Ying

model $p_{M_2^L}(x,y)$ can freely follow any Yang model. Third, $p_{M_{y|x}}(y|x), p_{M_y}(y)$ are free and $\mathcal{F}_1 \cap \mathcal{F}_2 \neq \emptyset$, where $\mathcal{F}_1$ is the set of all the functions represented in the form $p(y|x)/p(y)$, and $\mathcal{F}_2$ is the set of all the functions represented by $p_{M_{x|y}}(x|y)/p_{M_x}(x)$ with $p_{M_x}(x)$ prefixed and the structure of $p_{M_{x|y}}(x|y)$ prefixed.

Item 3.4 A parametric component is actually modeled by a physical device, and a free component is indirectly defined through the physical devices for other components, we call a BYY system a *Yang* based system when $p_{M_{x|y}}(x|y)$ is free and $p_{M_{y|x}}(y|x)$ is parametric, a *Ying* based system when $p_{M_{x|y}}(x|y)$ is parametric and $p_{M_{y|x}}(y|x)$ is free, and a *Ying-Yang Tuned* system when $p_{M_{x|y}}(x|y)$ and $p_{M_{y|x}}(y|x)$ both parametric. Furthermore, for a *Fully Matched* BKYY learning , it follows from eq.(24) that $p_{M_2^L}(x,y)$ is actually modeled by

$$p_{M_2^L}(x,y) = \begin{cases} p_{M_{x|y}}(x|y)p_{M_y}(y), & \text{a } \textit{Ying} \text{ based system,} \\ p_{M_{y|x}}(y|x)p_{M_x}(x), & \text{a } \textit{Yang} \text{ based system.} \end{cases} \tag{26}$$

Item 3.5 We may classify the architectures of BKYY learning according to the feature of the component $p_{M_{z|x,y}}(z|x,y)$. One type is the special case that $p_{M_{z|x,y}}(z|x,y) = p_{M_{z|y}}(z|y)$. The $p_{M_2}(x,z)$ given in Theorem 3.1 becomes

$$\begin{aligned} p_{M_2}(x,z) &= \int_y p_{M_{z|y}}(z|y)p_{M_2^L}(x,y)dy \\ &= \begin{cases} \int_y p_{M_{z|y}}(z|y)p_{M_{x|y}}(x|y)p_{M_y}(y)dy, & \textit{Ying} \text{ based or Ying-Yang Tuned,} \\ \int_y p_{M_{z|y}}(z|y)p_{M_{y|x}}(y|x)p_{M_x}(x)dy, & \text{Fully Matched } \textit{Yang} \text{ based,} \end{cases} \end{aligned} \tag{27}$$

which links x, z via a *cascade architecture* $x \to y \to z$ or $x \leftarrow y \to z$ and thus x, z are independent of each other when y is known. The architecture $x \to y \to z$ is usually called *Three-Layer Feedforward Net* or *Three Layer Perceptron*. The architecture $x \leftarrow y \to z$ can be used as a hypothesis testing model with z, x generated from the hypothesis y to be tested by the observed data. The other type of architectures is that $p_{M_{z|x,y}}(z|x,y) \neq p_{M_{z|y}}(z|y)$, and $p_{M_2}(x,z)$ is given by Theorem 3.1. In this case, each $p_{M_{z|x,y}}(z|x,y)$ builds a direct link $x \to z$ itself with the link gated via the internal variable y such that a weighted mixture $p_{M_2}(x,z)$ is formed in Theorem 3.1 in a *parallel architecture*. This architecture is usually called *Localized Architecture* or *Mixture of Experts*.

The following theorems on eq.(9) and eq.(6) are helpful for further understanding the above discussed various BKYY supervised learning systems.

Theorem 3.2 When $p_{M_{y|x}}(y|x)$ is free, both $\min_{p_{M_{y|x}}(y|x)} KL(p_{M_1}||p_{M_2})$ and $\min_{p_{M_{y|x}}(y|x)} KL_{two}(M_1, M_2) \Leftrightarrow p_{M_{y|x}}(y|x) = \frac{p_{M_{x|y}}(x|y)p_{M_y}(y)}{p_{M_2}(x)}$, with $p_{M_2}(x) = \int_y p_{M_{x|y}}(x|y)p_{M_y}(y)dy$, and $\min_{p_{M_{y|x}}(y|x)} KL(p_{M_1}||p_{M_2}) = KL(p_{M_x}(x)||p_{M_2}(x))$.

This theorem also tells us two points. One is that $p_{M_{y|x}}(y|x)$ will be fixed at the Bayesian posterior probability if it is free. The second is that after $p_{M_{y|x}}(y|x)$ is settled in this way, the specification of the remaining parts in $KL(p_{M_1}||p_{M_2})$ is the maximum likelihood learning of the mixture $p_{M_2}(x)$.

Theorem 3.3 When $p_{M_y}(y)$ is free, $\min_{p_{M_y}(y)} KL_{two}(M_1, M_2)$ gives $p_{M_y}(y) = 0.5(p_{M_1^L}(y) + p_{M_1}(y))$, with $p_{M_1}(y) = \int_x p_{M_{y|x}}(y|x)p_{M_x}(x)dx$ and

$p_{M_1^L}(y) = \int_{x,z} p_{M_{y|x,z}}(y|x,z) p_{M_1^L}(z,x) dx dz$. For a *Fully Matched* BKYY, we have $p_{M_y}(y) = p_{M_1}(y) = p_{M_1^L}(y)$.

Theorem 3.4 When $p_{M_{x|y}}(x|y)$ and $p_{M_y}(y)$ are both free, $\min_{\{p_{M_{x|y}}(x|y), p_{M_y}(y)\}} KL_{two}(M_1, M_2)$ is reached at $p_{M_{x|y}}(x|y) p_{M_y}(y) = 0.5(p_{M_1^L}(x,y) + p_{M_{y|x}}(y|x) p_{M_x}(x))$, $p_{M_1^L}(x,y) = \int_z p_{M_{y|x,z}}(y|x,z) p_{M_1^L}(z,x) dz$. Particularly, with $P_{M_1}(y)$ given by Theorem 3.3, for a *Fully Matched* BKYY learning we have $p_{M_{x|y}}(x|y) p_{M_y}(y) = p_{M_1^L}(x,y) = p_{M_{y|x}}(y|x) p_{M_x}(x)$, with $p_{M_y}(y) = p_{M_1}(y) = p_{M_1^L}(y)$, $p_{M_{x|y}}(x|y) = p_{M_1^L}(x|y) = p_{M_1}(x|y)$, and $p_{M_1^L}(x|y) = p_{M_1^L}(x,y)/p_{M_1^L}(y)$, $p_{M_1}(x|y) = p_{M_{y|x}}(y|x) p_{M_x}(x)/p_{M_1}(y)$.

5 Temporal BKYY Modeling System and Theory

Given time series $\{x_1, x_2, \cdots, x_T\}$ and $\{y_1, y_2, \cdots, y_T\}$, it is not enough to consider each pair (x_t, y_t) instantaneously as we did in the previous sections, since the order of samples conveys important serial information. Although we can simply make a temporal modeling by using a Time-Delay (TD) line as an input vector, i.e. $x = [x_{t-1}, x_{t-2}, \cdots, x_{t-p}]$, into the previous instantaneous model, it will not meet well our need on nonstationary time series. In [6], BKYY learning system and theory has been extended to act as a general system and theory for temporal modeling, which not only includes and extends the existing major temporal models, such as Hidden Markov Model (HMM), ARMA and AR models, but also provides some interesting new models. In this section, we will more systematically summarize and develop the results given in [6].

Let $\mathcal{X}_T = \{x_1, x_2, \cdots, x_T\}$ and $\mathcal{Y}_T = \{y_1, y_2, \cdots, y_T\}$, where each x_t, y_t can be a real scalar, a vector, and a discrete number, depending on the specific problem that we consider. In eq.(4), by replacing x in all its occurrences by $\mathcal{X}_T$ and y in all its occurrences by $\mathcal{Y}_T$, we can directly use the previous BYY learning system and theory as a general starting point for temporal modeling.

We consider the cases that $\mathcal{X}_T$, $\mathcal{Y}_T$ have the $p > 1$, $q > 1$ order Markov property, and get

$$p_{M_y}(\mathcal{Y}_T) = \prod_{t=1}^{T} p_{M_y}(y_t|\mathcal{Y}_{t-1}^{t-q}), \quad p_{M_{x|y}}(\mathcal{X}_T|\mathcal{Y}_T) = \prod_{t=1}^{T} p_{M_{x|y}}(x_t|y_t, \mathcal{X}_{t-1}^{t-p}, \mathcal{Y}_{t-1}^{t-q}),$$
$$p_{M_{y|x}}(\mathcal{Y}_T|\mathcal{X}_T) = \prod_{t=1}^{T} p_{M_{y|x}}(y_t|x_t, \mathcal{Y}_{t-1}^{t-q}, \mathcal{X}_{t-1}^{t-p}). \tag{28}$$

where we use the notations:

$$\mathcal{U}_t = \{u_1, \cdots, u_t\}, \mathcal{U}_t^\tau = \{u_\tau, \cdots, u_t\}, \ \mathcal{U}_0 = \{\emptyset\}, \mathcal{U}_t^\tau = \begin{cases} \mathcal{U}_t, & \tau \le 1, \\ \mathcal{U}_t^\tau = \{\emptyset\}, & \tau > t \end{cases}. \tag{29}$$

From putting these equations, we have

$$Kl(M_1, M_2) = \int p_{M_{y|x}}(\mathcal{Y}_T|\mathcal{X}_T) p_{M_x}(\mathcal{X}_T) \ln \frac{p_{M_{y|x}}(\mathcal{Y}_T|\mathcal{X}_T)}{p_{M_{x|y}}(\mathcal{X}_T|\mathcal{Y}_T) p_{M_y}(\mathcal{Y}_T)} d\mathcal{X}_T d\mathcal{Y}_T, \tag{30}$$

from which we may get various cases of the Temporal BKYY modeling system and theory. Several such cases have been discussed in [6].

We focus on the most typical case that $q = 1, p = 0$, that is

$$Kl(M_1, M_2) = \sum_{t=1}^{T} \int Kl_t(M_1, M_2|y_{t-1}) p_{M_y}(y_{t-1}) dy_{t-1}, \tag{31}$$

$$Kl_t(M_1, M_2|y_{t-1}) = \int p_{M_{y|x}}(y_t|x_t, y_{t-1}) p_{M_x}(x_t|y_{t-1}) \ln \frac{p_{M_{y|x}}(y_t|x_t, y_{t-1})}{p_{M_{x|y}} p_{M_y}(y_t|y_{t-1})} dx_t dy_t$$

$$p_{M_y}(y_t) = \int p_{M_y}(y_t|y_{t-1}) p_{M_y}(y_{t-1}) dy_{t-1}, \quad p_{M_{x|y}} = \begin{cases} p_{M_{x|y}}(x_t|y_t, y_{t-1}, \Theta_{M_{x|y}}), \\ p_{M_{x|y}}(x_t|y_t, \Theta_{M_{x|y}}) \end{cases}.$$

y_t is the invisible state variable. The parametric $p_{M_y}(y_t|y_{t-1}) = p(y_t|y_{t-1}, \Theta_{M_y})$ describes the state transient, and the parametric $p_{M_{x|y}}$ describes how x_t is generated from or related to the current and immediate pass states. While $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ describes the encoding or state discovery from the current x_t and the previous state y_{t-1}. It can be either free or parametric. For the special case that it is free, the minimization of $Kl(M_1, M_2)$ and thus $Kl_t(M_1, M_2|y_{t-1})$ will let it indirectly be specified by

$$p_{M_{y|x}}(y_t|x_t, y_{t-1}) = \frac{p_{M_{x|y}} p_{M_y}(y_t|y_{t-1})}{p_{M_{x|y,y}}(x_t|y_{t-1})}, \; p_{M_{x|y,y}}(x_t|y_{t-1}) = \int p_{M_{x|y}} p_{M_y}(y_t|y_{t-1}) dy_t,$$

$$Kl_t(M_1, M_2) = -\int p_{M_x}(x_t) \ln p_{M_{x|y,y}}(x_t|y_{t-1}) dx_t, \; p_{M_x}(x_t|y_{t-1}) = p_{M_x}(x_t). \tag{32}$$

In this case, the minimization of $Kl_t(M_1, M_2|y_{t-1})$ is equivalent to the maximization of a weighted likelihood.

The minimization of $Kl(M_1, M_2)$ can be made by the *Alternative Minimization* procedure eq.(17). However, here we should start at $t = 0$ with $p_{M_y}(y_0)$ specified, update $p_{M_y}(y_{t-1})$ into $p_{M_y}(y_t)$ by eq.(31) as a weight in the next $Kl_t(M_1, M_2), t > 1$. In some special case, we can directly deal the integral or summation operations involved.

In analogue to getting eq.(22) and eq.(23), from eq.(31) we propose a general adaptive algorithm based on stochastic approximation.

Given $p_{M_{x|y}}$ by eq.(31) and $p_{M_y}(y_0)$ with a random sample y_0. With a *Sampling Reference* density $p_r(y_t)$, starting from $t = 1$ we repeat the following steps:

Step 0: Get an observation x_t, randomly get one sample y_t from $p_r(y_t)$.

Step 1: With $p_{M_{x|y}}$, $p_{M_y}(y_t|y_{t-1})$ fixed, for $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ free, update it by eq.(32); for $p_{M_{y|x}}(y_t|x_t, y_{t-1}) = p(y_t|x_t, y_{t-1}, \Theta_{M_{y|x}})$ parametric, update

$$\Theta_{M_{y|x}}^{new} = \Theta_{M_{y|x}}^{old} - \eta \frac{p_{M_y}(y_{t-1})}{p_r(u_t)} \frac{\partial kl(\Theta_{M_{y|x}}, \Theta_{M_y}, \Theta_{M_{x|y}})}{\partial \Theta_{M_{y|x}}}|_{\Theta_{M_{y|x}} = \Theta_{M_{y|x}}^{old}},$$

$$kl(\Theta_{M_{y|x}}, \Theta_{M_y}, \Theta_{M_{x|y}}) = p(y_t|x_t, y_{t-1}, \Theta_{M_{y|x}}) \ln \frac{p(y_t|x_t, y_{t-1}, \Theta_{M_{y|x}})}{p_{M_{x|y}} p(y_t|y_{t-1}, \Theta_{M_y})} \tag{33}$$

Step 2: With $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ fixed, update

$$\Theta_{M_{x|y}}^{new} = \Theta_{M_{x|y}}^{old} + \eta \frac{p_{M_y}(y_{t-1}) p_{M_{y|x}}(y_t|x_t, y_{t-1})}{p_r(u_t)} \frac{\partial \ln p_{M_{x|y}}}{\partial \Theta_{M_{x|y}}}|_{\Theta_{M_{x|y}} = \Theta_{M_{x|y}}^{old}}, \tag{34}$$

$$\Theta_{M_y}^{new} = \Theta_{M_y}^{old} + \eta \frac{p_{M_y}(y_{t-1}) p_{M_{y|x}}(y_t|x_t, y_{t-1})}{p_r(u_t)} \frac{\partial \ln p(y_t|y_{t-1}, \Theta_{M_y})}{\partial \Theta_{M_y}}|_{\Theta_{M_y} = \Theta_{M_y}^{old}}.$$

Step 3: Update $p_{M_y}(y_t)$ by eq.(31).

Note: we can use an adpative choice $p_r(y_t) = p(y_t|y_{t-1}, \Theta_{M_y}^{old})$ in the above.

After the above modeling, we can use (a) $p(y_t|y_{t-1}, \Theta_{M_y})$ and $p_{M_{x|y}}$ by eq.(31) to predict x_t at $t-1$; (b) $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ to encode or recognize x_t into y_t; (c) $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ and $p_{M_{x|y}}$ to filter the current x_t.

In the following, we further look into two typical examples:

Item 4.1 BYY Hidden Markov Model (HMM) and learning theory When both x_t, y_t are discrete label, $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ is free and $p_{M_{x|y}} = p_{M_{x|y}}(x_t|y_t)$, we have that $\mathcal{Y}_T$ is a hidden Markov chain and $p_{M_y}(y_t|y_{t-1})$ corresponds to a $k \times k$ Markov transfer probability matrix. From eq.(32), we see that the minimization of $Kl(M_1, M_2)$ given by eq.(31) is equivalent to the maximum likelihood learning on a HMM. Also, from eq.(32) and eq.(31), via the *Alternative Minimization* eq.(17), we can get the well known Baum algorithm for HMM. Furthermore, we can get several new results for HMM as follows:

(a) The above stochastic approximation adaptive algorithm for HMM.

(b) Various extensions by different choices, e.g., x_t is real, $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ is parametric, $p_{M_{x|y}} = p_{M_{x|y}}(x_t|y_t, y_{t-1})$.

(c) The criteria for deciding the number k of hidden states by using eq.(13) on eq.(31), i.e., after getting $\Theta^*_{M_{x|y}}, \Theta^*_{M_y}$, $p^*_{M_{y|x}}(y_t|x_t, y_{t-1})$, and $p^*_{M_x}(x_t|y_{t-1}) = \int p(y_t|y_{t-1}, \Theta^*_{M_y}) p_{M_{x|y}}|_{\Theta_{M_{x|y}} = \Theta^*_{M_{x|y}}} dy_t$ via learning, we have

$$J_1(k) = Kl(M_1, M_2)|_{\{\Theta^*_{M_{x|y}}, \Theta^*_{M_y}, p^*_{M_{y|x}}(y_t|x_t, y_{t-1})\}}, \quad J_2(k) = J_1(k) - \tag{35}$$

$$\sum_{t=1}^{T} \int p_{M_y}(y_{t-1}) p^*_{M_{y|x}}(y_t|x_t, y_{t-1}) p^*_{M_x}(x_t|y_{t-1}) \ln p^*_{M_{y|x}}(y_t|x_t, y_{t-1}) dx_t dy_t dy_{t-1}.$$

which can also be computed incrementally as t increases.

(d) Other types of extensions from eq.(28) or $KL(M_2, M_1)$ or eq.(5).

Item 4.2 BYY State Space model and theory Considering

$$y_t = E(y_{t-1}, \Theta_g) + e_t, \; x_t = H(y_t, \Theta_h) + v_t, \text{ with white noise } e_t, v_t, \tag{36}$$

where e_t is independent of y_{t-1}, v_t is independent of y_t, e_t. $E(., \Theta_g), H(., \Theta_h)$ are deterministic linear or nonlinear mappings. When they are linear, eq.(36) reduces to the conventional linear state space model. In addition, $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ is either free or given by the parametric model

$$y_t = F(x_t, y_{t-1}, \Theta_f) + n_t, n_t \; is \; independent \; of \; x_t, y_{t-1}, e_t, v_t. \tag{37}$$

where $H(., \Theta_h)$ is a deterministic linear or nonlinear mapping.

From gaussians $G(e_t, 0, \Sigma_e)$, $G(v_t, 0, \Sigma_v)$ and $G(n_t, 0, \Sigma_n)$, we have

$$p_{M_{y|x}}(y_t|x_t, y_{t-1}) = \begin{cases} free, \\ G(y_t, F(x_t, y_{t-1}, \Theta_f)\Sigma_n) \end{cases}, \quad \Theta_{M_{y|x}} = \{\Theta_f, \Sigma_n\},$$
$$p_{M_y}(y_t|y_{t-1}) = G(y_t, E(y_{t-1}, \Theta_e), \Sigma_e), \; p_{M_{x|y}}(x_t|y_t) = G(x_t, H(y_t, \Theta_h), \Sigma_v),$$
$$\Theta_{M_y} = \{\Theta_e, \Sigma_e\}, \quad \Theta_{M_{x|y}} = \{\Theta_v, \Sigma_v\}, \quad kl(\Theta_{M_{y|x}}, \Theta_{M_y}, \Theta_{M_{x|y}}) =$$

$$G(y_t, F(x_t, y_{t-1}, \Theta_f), \Sigma_n) \ln \frac{G(y_t, F(x_t, y_{t-1}, \Theta_f), \Sigma_n)}{G(x_t, H(y_t, \Theta_h), \Sigma_v) G(y_t, E(y_{t-1}, \Theta_e), \Sigma_e)}, \tag{38}$$

$$\ln p_{M_{x|y}} = -0.5[\ln |\Sigma_v| + (x_t - H(y_t, \Theta_h))^T \Sigma_v^{-1} (x_t - H(y_t, \Theta_h))] + const,$$

$$\ln p_{M_y}(y_t|y_{t-1}) = -0.5[\ln |\Sigma_e| + (y_t - E(y_{t-1}, \Theta_e))^T \Sigma_e^{-1} (y_t - E(y_{t-1}, \Theta_e))] + const.$$

In the special case that $E(., \Theta_g), H(., \Theta_h)$ are linear and $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ is free, by putting eq.(38) into eq.(32) and eq.(31), we can analytically solve the integrals and get an accurate iterative algorithm by *Alternative Minimization* eq.(17), which is closely related to the well known Kalman filter algorithm for linear state space. Moreover, we also get several new results:

(a) Putting these conditions in eq.(33) and eq.(34), we have the above adaptive algorithm for both linear and nonlinear state space models.

(b) Variants by parametric $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ or $p_{M_{x|y}} = p_{M_{x|y}}(x_t|y_t, y_{t-1})$.

(c) The criteria for deciding the order k of the state space by the same one as eq.(35), which can be simplified considerably by inserting in the above eq.(38).

6 Temporal BKYY Regression System and Theory

For certain practical applications, we observe two time series $\{x_1, x_2, \cdots, x_T\}$ and $\{z_1, z_2, \cdots, z_T\}$, we are interested not only in modeling each of them but also in their regression or mapping relationship. In these cases, we should consider the general BYY learning system and theory discussed in Sec.3.

For simplicity, we consider the fully matched BKYY learning of the cases given by eq.(26) and eq.(27). That is, $KL(M_1, M_2) = 0$ and thus we have $Kl^L(M_1, M_2) =$

$$\int p_{M_{y|x,z}}(\mathcal{Y}_T|\mathcal{X}_T, \mathcal{Z}_T) p_{M_1^L}(\mathcal{X}_T, \mathcal{Z}_T) \ln \frac{p_{M_{y|x,z}}(\mathcal{Y}_T|\mathcal{X}_T, \mathcal{Z}_T)}{p_{M_{z|x,y}}(\mathcal{Z}_T|\mathcal{X}_T, \mathcal{Y}_T) p_{M_2^L}(\mathcal{X}_T, \mathcal{Y}_T)} d\mathcal{X}_T d\mathcal{Y}_T d\mathcal{Z}_T,$$

$$p_{M_2^L}(\mathcal{X}_T, \mathcal{Y}_T) = \begin{cases} p_{M_{x|y}}(\mathcal{X}_T|\mathcal{Y}_T) p_{M_y}(\mathcal{Y}_T), & \text{a } \textit{Ying} \text{ based system,} \\ p_{M_{y|x}}(\mathcal{Y}_T|\mathcal{X}_T) p_{M_x}(\mathcal{X}_T), & \text{a } \textit{Yang} \text{ based system,} \end{cases} \tag{39}$$

where we still use the assumption in eq.(28), and we assume that $\mathcal{Z}_T$ has the $p > 1$ order Markov property too. Generally speaking, from eq.(39) we may get various cases of the Temporal BKYY regression system and theory. Also, we can discard $-\int p_{M_{y|x,z}}(\mathcal{Y}_T|\mathcal{X}_T, \mathcal{Z}_T) p_{M_1^L}(\mathcal{X}_T, \mathcal{Z}_T) \ln p_{M_x}(\mathcal{X}_T) d\mathcal{X}_T d\mathcal{Y}_T d\mathcal{Z}_T$ for a Yang based system, because it depends on only the training data.

We again focus on the typical case that $q = 1, p = 0$, that is

$$Kl_L(M_1, M_2) = \sum_{t=1}^{T} \int Kl_L(t)(M_1, M_2) p_{M_y}(y_{t-1}) dy_{t-1}, \quad Kl_L(t)(M_1, M_2)$$

$$= \int p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1}) p_{M_1^L}(x_t, z_t) \ln \frac{p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1})}{p_{M_{z|x,y}} T_{t,t-1}} dx_t dz_t dy_t,$$

$$p_{M_{z|x,y}} = \begin{cases} p_{M_{z|x,y}}(z_t|x_t, y_t), \\ p_{M_{z|x,y}}(z_t|x_t, y_t, y_{t-1}); \end{cases} \quad p_{M_{x|y}} = \begin{cases} p_{M_{x|y}}(x_t|y_t), \\ p_{M_{x|y}}(x_t|y_t, y_{t-1}) \end{cases},$$

$$T_{t,t-1} = \begin{cases} p_{M_{x|y}} p_{M_y}(y_t|y_{t-1}), & \text{a } \textit{Ying} \text{ based system,} \\ p_{M_{y|x}}, \ p_{M_{y|x}} = \begin{cases} p_{M_{y|x}}(y_t|x_t,), \\ p_{M_{y|x}}(y_t|x_t, y_{t-1}), \end{cases} & \text{a } \textit{Yang} \text{ based system.} \end{cases} \tag{40}$$

Similar to Sec.5, we can use the *Alternative Minimization* procedure eq.(17) to get an accurate implementation algorithm or to obtain a stochastic approximation adaptive algorithm as in Sec.5.

In comparison with the model eq.(31), we have $p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1})$ instead of $p_{M_{y|x}}(y_t|x_t, y_{t-1})$ describes encoding or state discovering from the current x_t, z_t together and the previous state y_{t-1}, and we also have $p_{M_{z|x,y}}$ for the mapping or regression from $x_t \to z_t$ under the status of y_t or y_t, y_{t-1}. Similarly, $p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1})$ can be either free or parametric. When it is free, the minimization of $Kl_L(M_1, M_2)$ lets it be indirectly specified by

$$
\begin{aligned}
p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1}) &= \begin{cases} \frac{p_{M_{x|y}}(x_t|y_t, y_{t-1}) p_{M_y}(y_t|y_{t-1})}{p_{M_{z|x,y},M_2}(z_t, x_t, y_{t-1})}, & \text{a } \textit{Ying} \text{ based system,} \\ \frac{p_{M_{x|y}}(x_t|y_t, y_{t-1}) p_{M_{y|x}}(y_t|x_t, y_{t-1})}{p_{M_{z|x,y},M_2}(z_t|x_t, y_{t-1})}, & \text{a } \textit{Yang} \text{ based system;} \end{cases} \\
p_{M_{z|x,y},M_2}(z_t, x_t, y_{t-1}) &= \int p_{M_{x|y}}(x_t|y_t, y_{t-1}) p_{M_y}(y_t|y_{t-1}) dy_t, \qquad \textit{Ying}, \\
p_{M_{z|x,y},M_2}(z_t|x_t, y_{t-1}) &= \int p_{M_{x|y}}(x_t|y_t, y_{t-1}) p_{M_{y|x}}(y_t|x_t, y_{t-1}) dy_t, \quad \textit{Yang}; \qquad (41) \\
Kl_L(t)(M_1, M_2) &= -\int p_{M_1^L}(x_t, z_t) \begin{cases} \ln p_{M_{z|x,y},M_2}(z_t, x_t, y_{t-1}) dz_t dx_t, & \textit{Ying}, \\ \ln p_{M_{z|x,y},M_2}(z_t|x_t, y_{t-1}) dz_t dx_t, & \textit{Yang}. \end{cases}
\end{aligned}
$$

From here, we may get the maximum likelihood learning for various generalized state space model or HMM based regression. Moreover, we get various cases of BKYY learning for them when $p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1})$ is parametric.

It is interesting to observe that for both the cases that $p_{M_{y|x,z}}(y_t|x_t, z_t, y_{t-1})$ is free and parametric, we have:

(a) In a Yang based system, the transient from y_{t-1} to y_t is controlled by $p_{M_{y|x}}$ depending on x_t. Particularly, for a cascade structure $p_{M_{z|x,y}}(z_t|x_t, y_t, y_{t-1}) = p_{M_{z|x,y}}(z_t|y_t, y_{t-1})$, we actually get a general recurrent feed-forward BYY system, which not only includes the conventional three layer recurrent net as special case, but also provides new recurrent models. For $p_{M_{z|x,y}}(z_t|x_t, y_t, y_{t-1}) \neq p_{M_{z|x,y}}(z_t|y_t, y_{t-1})$, we get a model of recurrent mixture of experts.

(a) In a Ying based system, for a cascade structure, $p_{M_{z|x,y}}(z_t|x_t, y_t, y_{t-1}) = p_{M_{z|x,y}}(z_t|y_t, y_{t-1})$ and $p_{M_{x|y}}(x_t|y_t, y_{t-1})$ describes how the observation x_t, z_t are coordinately generated from the hidden states. For $p_{M_{z|x,y}}(z_t|x_t, y_t, y_{t-1}) \neq p_{M_{z|x,y}}(z_t|y_t, y_{t-1})$, we get a general recurrent alternative mixture expert model, with the mapping by multiple experts gated by $p_{M_{x|y}}(x_t|y_t, y_{t-1}) p_{M_y}(y_t|y_{t-1})$.

References

1. Xu, L., "Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (I) Unsupervised and Semi-Unsupervised Learning", Invited paper, S. Amari and N. Kassabov eds., *Brain-like Computing and Intelligent Information Systems*, 1997, Springer-Verlag, pp241-274.
2. Xu, L., "Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (III) Models and Algorithms for ICA, Supervised Learning Networks and Temporal Processing", in the same proceedings, pp43-60. .

3. Xu, L., "Bayesian Ying-Yang System and Theory as A Unified Statistical Learning Approach: (IV) The Ying Dominated Theory, Models and Algorithms", to appear.
4. Xu, L., "Bayesian Ying-Yang Machine, Clustering and Number of Clusters", *Pattern Recognition Letters*, in press, 1997.
5. Xu, L., "YING-YANG Machine: a Bayesian-Kullback scheme for unified learnings and new results on vector quantization", Keynote talk, *Proc. Intl Conf. on Neural Information Processing (ICONIP95)*, Oct 30 - Nov. 3, pp977-988(1995).
6. Xu, L., "YING-YANG Machine for Temporal Signals", Keynote talk, *Proc IEEE Intl. Conf. Neural Networks & Signal Processing*, Nanjing, Dec.10-13, Vol. I, pp644-651(1995).

Appendix: Proofs of Theorems

Proof of Theorem 3.1 Since $p_{M_{y|x,z}}(y|x,z)$ is irrelevant to $KL(M_1,M_2)$, we have that $\min_{p_{M_{y|x,z}}(y|x,z)}\ KL_{two}(M_1,M_2)$ is equivalent to $\min_{p_{M_{y|x,z}}(y|x,z)}\ KL^L(M_1,M_2)$. From eq.(9), $KL^L(M_1,M_2)=\int p_{M_1^L}(z,x)KL(p_{M_{y|x,z}}(y|x,z)||p_{M_2}(y|x,z))dxdz+$ $KL(p_{M_1^L}(z,x)||p_{M_2}(x,z))$, $p_{M_2}(y|x,z)=\frac{p_{M_{z|x,y}}(z|x,y)p_{M_2^L}(x,y)}{p_{M_2}(x,z)}$, with $p_{M_2}(x,z)$ given by Theorem 3.1.

Proof of Theorem 3.2 Since $p_{M_{y|x}}(y|x)$ is irrelevant to $KL^L(M_1,M_2)$, $\min_{p_{M_{y|x}}(y|x)}\ KL_{two}(M_1,M_2)$ is equivalent to $\min_{p_{M_{y|x}}(y|x)}\ KL(M_1,M_2)$. Via eq.(6),
$KL(M_1,M_2)=\int_x p_{M_x}(x)KL(p_{M_{y|x}}(y|x)||p_{M_2}(y|x))dx+KL(p_{M_x}(x)||p_{M_2}(x))$,
with $p_{M_2}(y|x)=\frac{p_{M_{x|y}}(x|y)p_{M_y}(y)}{p_{M_2}(x)}$ and $p_{M_2}(x)$ given by Theorem 3.2. The second term is irrelevant to $p_{M_{y|x}}(y|x)$, thus the first term is minimized at $p_{M_{y|x}}(y|x)=p_{M_2}(y|x)$.

Proof of Theorem 3.3 From eq.(9) and eq.(6), in $KL_{two}(M_1,M_2)$ the part relevant to $p_{M_y}(y)$ is $T=-\int p_{M_{y|x,z}}(y|x,z)p_{M_1^L}(z,x)\ln p_{M_y}(y)dxdydz-\int p_{M_{y|x}}(y|x)p_{M_x}(x)$ $\ln p_{M_y}(y)dxdy=-\int_y p_{M_1^L}(y)\ln p_{M_y}(y)dy-\int_y p_{M_1}(y)\ln p_{M_y}(y)dy$, where $p_{M_1^L}(y)$ and $p_{M_1}(y)$ are given in Theorem 3.3. Furthermore, the minimization of T with respect to $p_{M_y}(y)$ will not be affected when we add into T the following terms
$T+\int_y\frac{p_{M_1^L}(y)+p_{M_1}(y)}{2}\ln\frac{p_{M_1^L}(y)+p_{M_1}(y)}{2}dy=\int_y\frac{p_{M_1^L}(y)+p_{M_1}(y)}{2}\ln\frac{p_{M_1^L}(y)+p_{M_1}(y)}{2p_{M_y}(y)}dy$,
that are irrelevant to $p_{M_y}(y)$. This T reaches its minimum at $p_{M_y}(y)=0.5(p_{M_1^L}(y)$ $+p_{M_1}(y))$. For a *Fully Matched* BKYY learning, by integrating both sides of eq.(24), we have $p_{M_y}(y)=p_{M_1}(y)$ and thus $p_{M_1^L}(y)=2p_{M_y}(y)-p_{M_1}(y)=p_{M_y}(y)$ too.

Proof of Theorem 3.4 Similar to the proof of Theorem 3.3, we can get that in $KL_{two}(M_1,M_2)$ the part that is relevant to $p_{M_{x|y}}(x|y),p_{M_y}(y)$ is $T=-\int_{x,y}p_{M_1^L}(x,y)$ $\ln[p_{M_{x|y}}(x|y)p_{M_y}(y)]dxdy-\int_{x,y}p_{M_{y|x}}(y|x)p_{M_x}(x)\ln[p_{M_{x|y}}(x|y)p_{M_y}(y)]dxdy$,
$T+\int_{x,y}\frac{p_{M_1^L}(x,y)+p_{M_{y|x}}(y|x)p_{M_x}(x)}{2}\ln\frac{p_{M_1^L}(x,y)+p_{M_{y|x}}(y|x)p_{M_x}(x)}{2}dxdy$
$=\int_{x,y}\frac{p_{M_1^L}(x,y)+p_{M_{y|x}}(y|x)p_{M_x}(x)}{2}\ln\frac{p_{M_1^L}(x,y)+p_{M_{y|x}}(y|x)p_{M_x}(x)}{2p_{M_{x|y}}(x|y)p_{M_y}(y)}dxdy$. Thus, we get the first equation in Theorem 3.2. Particularly, for a *Fully Matched* BKYY learning, we have $p_{M_{x|y}}(x|y)p_{M_y}(y)=p_{M_{y|x}}(y|x)p_{M_x}(x)$ given by eq.(24) and thus we get the first equation in Theorem 3.4. From Theorem 3.3, we have $p_{M_y}(y)=p_{M_1}(y)=p_{M_1^L}(y)$ and then use them to divide each terms in that first equation, we get the rest of Theorem 3.4 proved.