in "Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques", eds. by Olivas, Guerrero, Sober, Benedito, & Lopez, IGI Global publication, pp60-94.

Chapter 3
Learning Algorithms for RBF Functions and Subspace Based Functions

Lei Xu
Chinese University of Hong Kong and Beijing University, PR China

ABSTRACT

Among extensive studies on radial basis function (RBF), one stream consists of those on normalized RBF (NRBF) and extensions. Within a probability theoretic framework, NRBF networks relates to nonparametric studies for decades in the statistics literature, and then proceeds in the machine learning studies with further advances not only to mixture-of-experts and alternatives but also to subspace based functions (SBF) and temporal extensions. These studies are linked to theoretical results adopted from studies of nonparametric statistics, and further to a general statistical learning framework called Bayesian Ying Yang harmony learning, with a unified perspective that summarizes maximum likelihood (ML) learning with the EM algorithm, RPCL learning, and BYY learning with automatic model selection, as well as their extensions for temporal modeling. This chapter outlines these advances, with a unified elaboration of their corresponding algorithms, and a discussion on possible trends.

BACKGROUND

The renaissance of neural network and then machine learning since the 1980's is featured by two streams of extensive studies, one on multilayer perceptron and the other on radial basis function networks or shortly RBF net. While a multilayer perceptron partitions data space by hyperplanes and then making subsequent processing via nonlinear transform, a RBF net partitions data space into modular regions via local structures, called radial or non-radial basis functions. After extensive studies on multilayer perceptron, studies have been turned to RBF net since the late 1980's and the early 1990's, with a wide range applications (Kardirkamanathan, Niranjan, & Fallside, 1991; Chang & Yang, 1997; Lee, 1999;

DOI: 10.4018/978-1-60566-766-9.ch003

Mai-Duy & Tran-Cong, 2001; Er, 2002; Reddy & Ganguli, 2003; Lin & Chen 2004; Isaksson, Wisell, & Ronnow, 2005; Sarimveis, Doganis, & Alexandridis, 2006; Guerra & Coelho, 2008; Karami & Mohammadi, 2008).

In the literature of machine learning, advances on RBF net can be roughly divided into two streams. One stems from the literature of mathematics on multivariate function interpolation and spline approximation, as well as Tikhonov type regularization for ill-posed problems, which were brought to neural networks learning by (Powell, 1987; Broomhead & Lowe, 1998; Poggio & Girosi, 1990; Yuille & Grzywacz, 1989) and others. Actually, it is shown that RBF net can be naturally derived from Tikhonov regularization theory, i.e. least square fitting subjected to a rotational and translational constraint term by a different stabilizing operator (Poggio & Girosi, 1990; Yuille & Grzywacz, 1989). Moreover, RBF net has also been shown to have not only the universal approximation ability possessed by multilayer perceptron (Hartman, 1989; Park & Sandberg,1993), but also the best approximation ability (Poggio & Girosi, 1990) and a good generalization property (Botros & Atkeson, 1991).

The other stream can be traced back to Parzen Window estimator proposed in the early 1960's. Studies of this stream progress via active interactions between the literature of statistics and the literature of machine learning. During 1970's-80's, Parzen window has been widely used for estimating probability density, especially for estimating the class densities that are used for Bayesian decision based classification in the literature of pattern recognition. Also, it has been introduced into the literature of neural networks under the name of probabilistic neural net (Specht, 1990). By that time, it has not yet been related to RBF net since it does not directly relates to the above discussed function approximation purpose.

In the literature of neural networks and machine learning, Moody & Darken (1989) made a popular work via Gaussian bases as follows:

$$f(x)=\sum_{j=1}^{k} w_j \varphi_j(x-c_j,\Sigma_j), \quad \varphi_j(x-c_j,\Sigma_j)=e^{-0.5(x-c_j)^T\Sigma_j^{-1}(x-c_j)} \Big/ \sum\nolimits_{j=1}^{k} e^{-0.5(x-c_j)^T\Sigma_j^{-1}(x-c_j)}. \tag{1}$$

This normalized type of RBF net is featured by

$$\sum_{j=1}^{k} \varphi_j(x-c_j,\Sigma_j)=1, \tag{2}$$

and thus usually called Normalized RBF (RBF). Moody & Darken (1989) actually considered a special case $\Sigma_j=\sigma_j^2 I$. Unknowns are learned from a set of samples via a two stage implementation. First, a clustering algorithm (e.g., k-means) is used to estimate c_i as each cluster's center. Second, $\varphi_j(x)$ is calculated for every sample and unknowns w_j are estimated by a least square linear fitting. Nowlan (1990) proceeded along this line with Σ_j considered in its general form such that not only the receptive field of each base function can be elliptic instead of radial symmetrical, but also the well known EM algorithms (Redner & Walker, 1984) are adopted for estimating c_i and Σ_j with better performances than that by a clustering algorithm.

In the nonparametric statistics literature, extensive studies have also been made towards regression tasks under the name of kernel regression estimator, which is an extension of Parzen window estimator from density estimation to statistical regression (Devroye, 1981&87). In (Xu, Krzyzak, & Yuille,

1992&94), kernel regression estimators are shown to be special cases of NRBF net. In help of this connection, theoretical results available on kernel regression have been adopted, which cast new lights on the theoretical analysis of NRBF net in several aspects, e.g., universal approximation extended to statistical consistency, convergence extended to convergence rate, etc.

Another line of studies in the early 1990 is featured by using the mixtures-of-experts (ME) (Jacobs, Jordan, Nowlan, & Hinton, 1991) for regression:

$$f(x) = \sum_{j=1}^{k} g_j(x,\phi) f_j(x,\theta_j), \tag{3}$$

where each $g_j(x,\phi)$ is a function implemented by a three layer forward networks, and $0 \le g_j(x,\phi) \le 1$ is called gating net. This $f(x) = \sum_{j=1}^{k} g_j(x,\phi) f_j(x,\theta_j)$ is actually a regression equation of a mixture of conditional distribution as follows:

$$q(z \mid x) = \sum_{j=1}^{k} g_j(x,\phi) q(z \mid x, j), \text{ with } f(x) = E_{q(z|x)} z \text{ and } f_j(x,\theta_j) = E_{q(z|x,j)} z. \tag{4}$$

One typical example is $q(z \mid x, j) = G(z \mid f_j(x,\theta_j), \Gamma_j)$, where $G(u \mid \mu, \Sigma)$ denotes a Gaussian density with mean vector μ and covariance matrix Σ. All the unknowns in this mixture are estimated via the maximum likelihood (ML) learning by a generalized EM algorithm (Jordan & Jacobs, 1994; Jordan & Xu, 1995).

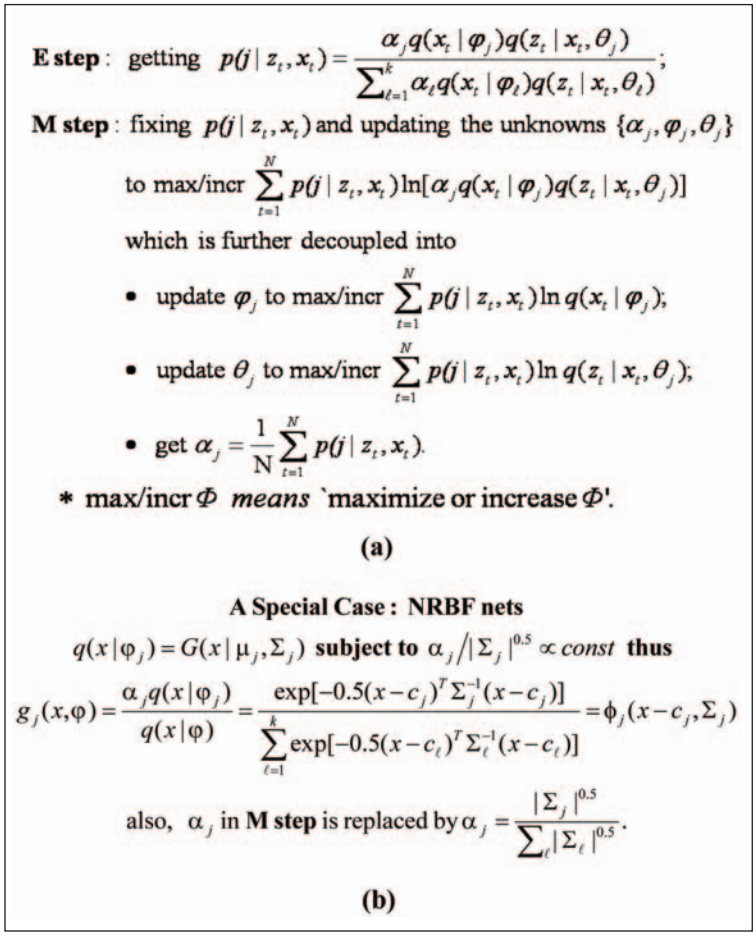
Furthermore, a connection has been built between this ME model and NRBF net such that the EM algorithm has been adopted for estimating jointly all the parameters in a NRBF net to improve suboptimal performances due to a two stage implementation. In sequel, we start at this EM algorithm, and then introduce further advances on NRBF studies. Not only basis functions is extended from radial to subspace based functions, and further to temporal modeling, but also studies further proceed to tackle model selection problem, i.e., how to determine the number *k* and the subspace dimensions.

THE EM ALGORITHM AND BEYOND: ALTERNATIVE ME VERSUS ENRBF

Xu, Jordan & Hinton (1994&95) proposed an alternative mixtures-of-experts (AME) via a different gating net. Considering each expert $G(z \mid f_j(x,\theta_j), \Gamma_j)$ supported on a Gaussian $G(x \mid \mu_j, \Sigma_j)$ or generally an expert $q(z \mid x, \theta_j)$ supported on a corresponding $q(x \mid \phi_j)$, we have that $p(z \mid x)$ is supported on a finite mixture $q(x \mid \phi)$ with its corresponding posteriori as the gating net:

$$q(z \mid x) = \sum_{j=1}^{k} g_j(x,\phi) q(z \mid x, \theta_j), \quad g_j(x,\phi) = \frac{\alpha_j q(x \mid \phi_j)}{q(x \mid \phi)}, \quad q(x \mid \phi) = \sum_{j=1}^{k} \alpha_j q(x \mid \phi_j), \ 0 \le \alpha_j, \ \sum_{j=1}^{k} \alpha_j = 1. \tag{5}$$

Figure 1. Algorithm 1: EM algorithm for alternative mixture of experts and NRBF nets



Given a training set $\{z_t, x_t\}_{t=1}^{N}$, parameter learning is made by the ML learning on the joint distribution $q(z \mid x)q(x \mid \phi) = \sum_{j=1}^{k} \alpha_j q(x \mid \phi_j) q(z \mid x, \theta_j)$, which is implemented by the EM algorithm, i.e., Algorithm 1(a). Details are referred to (Xu, Jordan & Hinton, 1995).

We observe that the M step decouples the updating in parts. The one for updating θ_j of each expert is basically same as in (Jacobs, Jordan, Nowlan, & Hinton, 1991) except here we get a different $p(j \mid z_t, x_t)$ as the E step. The ones for updating the unknowns α_j, ϕ_j of the gate can be made via letting $p(j \mid z_t, x_t)$ to take the role of $p(j \mid x_t)$ in the M step of the EM algorithm on a finite mixture or Gaussian mixture (Redner & Walker, 1984; Nowlan, 1990).

Alternatively, we can use the resulted $p(j \mid z_t, x_t)$ to replace $g_j(x, \phi)$ as the gate in eqn.(5), i.e.,

$$q(z_t \mid x_t) = \sum_{\ell=1}^{k} p(\ell \mid z_t, x_t) q(z_t \mid x_t, \theta_\ell) \ . \tag{6a}$$

When z_t is discrete, this equation is still directly computable on testing samples. However, when z_t takes real values that are usually not available on testing samples, we can not directly use $p(j \mid z_t, x_t)$ to

replace $g_j(x,\phi)$ as the gate in eqn.(4). Instead, we get an estimate $f_j(x_t,\theta_j) = E_{q(z|x,j)}z$ as $\hat{z}_t$, from which we usually have either $q(\hat{z}_t \mid x_t,\theta_j)$ in a constant or $q(\hat{z}_t \mid x_t,\theta_j) = \varphi(\theta_j)$, e.g., $\varphi(\theta_j) = (2\pi)^{-0.5d} \mid \Gamma_j \mid^{-0.5}$ for $q(z \mid x,\theta_j) = G(z \mid f_j(x,\theta_j),\Gamma_j)$. Thus, we get

$$p(j \mid x,z) = \frac{\alpha_j q(x \mid \phi_j) q(z \mid x,\theta_j)}{\sum_{j=1}^{k} \alpha_j q(x \mid \phi_j) q(z \mid x,\theta_j)}, \quad \text{for an integer z,}$$
$$p(j \mid x,\hat{z}_t) = \frac{\alpha_j q(x \mid \phi_j) \varphi(\theta_j)}{\sum_{j=1}^{k} \alpha_j q(x \mid \phi_j) \varphi(\theta_j)}, \qquad \text{for a real z.} \tag{6b}$$

which takes the place of $g_j(x,\phi)$ as the gate in eqn.(4).

Furthermore, NRBF net is obtained as a special case, as shown in Algorithm I(b). In this case, the regression function in eqn. (3) becomes

$$f(x) = \sum_{j=1}^{k} f_j(x,\theta_j)\varphi_j(x - c_j,\Sigma_j), \tag{7}$$

which is actually an extension of NRBF net with a constant w_j generalized to a general regression $f_j(x,\theta_j)$. That is, we are lead back to eqn.(2) when $f_j(x,\theta_j) = w_j$ and the so called Extended NRBF (ENRBF) net (Xu, Jordan & Hinton, 1994&95; Xu, 1998) when $f_j(x,\theta_j) = W_j x + w_j$. Straightforwardly, we can use the EM algorithm in Algorithm I to update all the unknowns, from which we can also obtain different versions of the EM algorithms at various particular cases (Xu, 1998). For an example, it improves the two stage implementation of learning NRBF by (Nowlan, 1990) in a sense that the updating of c_i, Σ_j also takes w_j in consideration via $p(j \mid z_t, x_t)$, while the updating of w_j takes c_i and Σ_j in consideration via this $p(j \mid z_t, x_t)$ too.

It is interesting to further elaborate the connection between RBF net and the ME models from a dual perspective. For the RBF net by eqn.(2), we seek a combination of a series of basis functions $\varphi_j(x - c_j,\Sigma_j)$ via linear weights of w_j. For the ME model by eqn.(3), we seek a convex combination of a series of functions $f_j(x,\theta_j)$ weighted by $g_j(x,\phi)$ that actually takes the role of the basis function $\varphi_j(x - c_j,\Sigma_j)$ in eqn.(2). That is, there is a dual perspective that can swap the roles of the weighting coefficients and what are weighted. The ME perspective provides extensions of RBF net from classical radial basis functions $\varphi_j(x - c_j,\Sigma_j)$ to $g_j(x,\phi) = \alpha_j G(x \mid \mu_j,\Sigma_j) \Big/ \sum_{j=1}^{k} \alpha_j G(x \mid \mu_j,\Sigma_j)$ and further to $p(j \mid z_t, x_t)$ in eqn.(6a) and eqn.(6b). Moreover, in addition to NRBF net with $f_j(x,\theta_j) = w_j$ and ENRBF net with $f_j(x,\theta_j) = W_j x + w_j$, $f_j(x,\theta_j)$ can also be implemented by a multilayer networks (Jacobs, Jordan, Nowlan, & Hinton, 1991), which actually combines the features of both NRBF net and three layer forward networks.

TWO STAGE IMPLEMENTATION VERSUS AUTOMATIC MODEL SELECTION

Using a structure by a RBF net or a ME model to accommodate a mapping $x \rightarrow z$, one needs a learning algorithm that estimates unknown parameters in this structure. The performance of a resulted RBF or ME can be measured via a set of testing samples by either a square error $\left\|z_t - f(x_t)\right\|^2$ for function approximation or classification error for pattern recognition, as well as the likelihood $\ln p(z_t \mid x_t)$ for distribution approximation. Correspondingly, an algorithm is guided by one of these purposes via a set of training set. The classic RBF net studies considers minimizing the square error $\left\|z_t - f(x_t)\right\|^2$ on a set $\mathbf{X}_N = \{x_t\}_{t=1}^N$ of training samples. The studies (Jacobs, Jordan, Nowlan, & Hinton, 1991; Jordan & Jacobs, 1995) on the ME model considers maximizing the likelihood $\ln p(z_t \mid x_t)$ on $\mathbf{X}_N$, while the alternative ME considers maximizing the likelihood $\ln p(z_t, x_t)$ on $\mathbf{X}_N$ (Xu, Jordan & Hinton, 1994&95). However, a good performance on a training set is not necessarily good on a testing set, especially when the training set consists of a small size of samples. The reason is too many free parameters to be determined. Studies towards this problem have been widely studied along two directions in the literature of statistics and machine learning.

One is called regularization that adds some constraint or regularity to the unknown parameters, the selected structure, and the training samples (Powell, 1987; Poggio, & Girosi, 1990). Readers are referred to (Xu, 2007c) for a summary of typical regularization techniques studied in the literature of learning. One example is smoothing $\mathbf{X}_N$ by a Gaussian kernel with a zero mean and a unknown smoothing parameter h as its variance, i.e., instead of directly using $\mathbf{X}_N$ for training we consider $p(\mathbf{X} \mid \mathbf{X}_N, h) = G(\mathbf{X} - \mathbf{X}_N \mid 0, h^2 I)$.

Actually, maximizing $\ln p(z_t, x_t)$ by the alternative ME can be regarded as maximizing $\ln p(z_t \mid x_t)$ plus a term $\ln q(x \mid \phi)$ that adds certain constrain on samples of x_t. This is an example of a family of typical regularization techniques, featured by the format $\ln p(z_t \mid x_t)$ or $\ln p(z_t \mid x_t) + \lambda\Omega(\theta, z_t, x_t)$, where $\Omega(\theta)$ or $\Omega(\theta, z_t, x_t)$ is usually called stabilizer or regularizor, and λ is called regularization strength.

However, we encounter two difficulties to impose a regularization. One is the difficulty of appropriately controlling a regularization strength λ, which is usually able to be roughly estimated only for a rather simple structure via either handling the integral of marginal density or in help of cross validation (Stone, 1978), but with very extensive computing costs. The other is the difficulty of choosing an appropriate $\Omega(\theta)$ or $\Omega(\theta, x_t)$ based on a priori knowledge that we are usually not available and difficult to get. Instead, an isotropic or nonspecific stabilizer is usually imposed, e.g.,. $\Omega(\theta) = \left\|\theta\right\|^2$. This type of isotropic or nonspecific regularization faces a dilemma. We need a regularization on a structure $\mathbf{S}_\mathbf{k}(\Theta_\mathbf{k})$ with extra parts, while an isotropic or nonspecific regularization can not discard the extra parts but still let them in action to blur those useful parts. For an example, given a set of samples $\{z_t, x_t\}$ that come from a curve $z = x^2 + 3x + 2$, if we use a polynomial $z = \sum_{i=0}^k a_i x^i$ of an order $k > 2$ to fit the data set, we desire to force all the parameters $\{a_i, i \geq 3\}$ to be zero, while minimizing $\left\|\theta\right\|^2 = \sum_{i=0}^k a_i^2$ fails to treat the parameters $\{a_i, i \geq 3\}$ differently from the parameters $\{a_i, i \leq 2\}$.

To tackle the problems, we turn to consider the other direction that consists of those efforts made under the name of model selection. It refers to select an appropriate one among a family of infinite many candidate structures $\{\mathbf{S}_\mathbf{k}(\Theta_\mathbf{k})\}$ with each $\mathbf{S}_\mathbf{k}(\Theta_\mathbf{k})$ in a same configuration but in different scales, each of which is labeled by a scale parameter $\mathbf{k}$ in term of one integer or a set of integers. Selecting an

appropriate **k** means getting a structure that consists of an appropriate number of free parameters. For a structure by a RBF net or a ME model, this **k** is simply the number *k* of bases or experts. Usually, a maximum likelihood (ML) principle based learning is not good for model selection. The EM algorithm in Algorithm I works well with a pre-specified number *k* of bases or experts. The performance will be affected by whether an appropriate *k* is selected, while how to determine *k* is a critical problem.

Many efforts have been made towards model selection for over three decades in past. The classical one is making a two stage implementation. First, enumerate a candidate set **K** of **k** and estimate the unknown set Θ_k of parameters by ML learning for a solution $\Theta_{\mathbf{k}}^*$ at each $\mathbf{k} \in \mathbf{K}$. Second, use a model selection criterion $J(\Theta_{\mathbf{k}}^*)$ to select a best $\mathbf{k}^*$. Several criteria are available for the purpose, such as AIC, CAIC, BIC, cross validation, etc (Stone, 1978; Akaike, 1981; Bozdogan, 1987; Wallace& Freeman, 1987; Cavanaugh, 1997; Rissanen, 1989; Vapnik, 1995&2006). Also, readers are referred to (Xu, 2007c) for a recent elaboration and comparison. Some of these criteria (e.g., AIC, BIC) have also been adopted for selecting the number **k** of basis functions in RBF networks (Konishi, Ando, & Imoto, 2004). Unfortunately, anyone of these criteria usually provides a rough estimate that may not yield a satisfactory performance. Even with a criterion $J(\Theta_k)$ available, this two stage approach usually incurs a huge computing cost. Still, the parameter learning performance deteriorates rapidly as **k** increases, which makes the value of $J(\Theta_k)$ evaluated unreliably.

One direction that tackles this challenge is featured by incremental algorithms that attempts to incorporate as much as possible what learned as **k** increases step by step. Its focus is on learning newly added parameters, e.g., the studies made on mixture of factor analyses (Ghahramani & Beal, 2000; Salah and & Alpaydin, 2004). Such an incremental implementation can save computing costs. However, it usually leads to suboptimal performance because not only those newly added parameters but also the old parameter set Θ_k actually have to be re-learned. This suboptimal problem is lessened by a decremental implementation that starts with **k** at a large value and reduces **k** step by step. At each step, one takes a subset out of Θ_k with the remaining parameter updated, and discard the subset with a biggest decreasing of $J(\Theta_{\mathbf{k}}^*)$ after trying a number of such subsets. Such a procedure can be formulated as a tree searching. The initial parameter set Θ_k is the root of the tree, and discarding one subset moves to one immediate descendent. A depth-first searching suffers from a suboptimal performance seriously, while a breadth-first searching suffers a huge combinatorial computing cost. Usually, a trade off between the two extremes is adopted.

One other direction of studies is called automatic model selection. An early effort of this direction is Rival Penalized Competitive Learning (RPCL) (Xu, Krzyzak, & Oja, 1992&93) for adaptively learning the centers c_i and Σ_j of radial basis in NRBF networks (Xu, Krzyzak & Oja, 1992&93; Xu,1998a), with the number *k* automatically determined during learning. The key idea is that not only the winner c_w moves a little bit to adapt the current sample x_t but also the rival (i.e., the second winner) c_r is repelled a little bit from x_t to reduce a duplicated information allocation. As a result, an extra c_j will be driven far away from data with its corresponding $\alpha_j \to 0$ and $Tr[\Sigma_j] \to 0$. Moreover, RPCL learning algorithm has been proposed for jointly learning not only c_i and Σ_j but also $W_j x + w_j$ (Sec.4.4 & Table 2, Xu, 2001&02). In general, RPCL is applicable to any $\mathbf{S}_{\mathbf{k}}(\Theta_{\mathbf{k}}\}$ that consists of *k* individual substructures. With *k* initially at a value larger enough, a coming sample x_t is allocated to one of the *k* substructures via competition, and the winner adapts this sample by a little bit, while the rival is de-learned a little bit to reduce a duplicated allocation. This rival penalized mechanism will discard those extra substructures, making model selection automatically during learning. Various extensions have been made in the

past decades (Xu,2007d). Instead of the heuristic mechanism embedded in RPCL algorithm, Bayesian Ying-Yang (BYY) harmony learning was proposed in (Xu,1995) and systematically developed in the past decade, which leads us to not only new model selection criteria but also algorithms with automatic model selection ability via maximizing a Ying Yang harmony functional.

In general, this automatic model selection direction demonstrates a quite difference nature from the usual incremental or decremental implementation that bases on evaluating the change $J(\Theta_{\mathbf{k}}) - J(\Theta_{\mathbf{k}} \cup \theta_{\Delta})$ as a subset θ_{Δ} of parameters is added or removed. Thus, automatic model selection is associated with a learning algorithm or a learning principle with the following two features:

- There is an indicator $\rho(\theta_{\mathrm{r}})$ on a subset $\theta_{\mathbf{r}} \in \Theta_{\mathbf{k}}$, we have $\rho(\theta_{\mathbf{r}}) = 0$ if θ_{r} consists of parameters of a redundant structural part.
- In implementation of this learning algorithm or optimizing this learning principle, there is an mechanism that automatically drives $\rho(\theta_{\mathbf{r}}) \to 0$ and θ_{r} towards a specific value. Thus, the corresponding redundant structural part is effectively discarded.

Shown in Algorithm 2 (displayed in Figure 2) is a summary of three types of adaptive algorithms for alternative ME, including NRBF and ENRBF as special cases. The implementation is almost the same for three types of learning, except taking different values of $\eta_{j,t}$. Considering *k* individual substructures, each has a local measure $H_t(\Theta_j)$ on its fitness to a coming sample x_t and the parameters Θ_j are updated by updating rules with same format to three types of learning. The updating rules are obtained from the gradient $\nabla_{\Theta_j} H_t(\Theta_j)$ either directly or in a modified direction that has a positive projection on this gradient, as explained in Figure 3(E). According to Figure 3(A), each updating direction is modified by $\eta_{j,t}$ on its direction and step size.

We get a further insight at a special case that $h_x = 0,\ \mathrm{h}_{\mathrm{z}} = 0$ (thus Step (d) discarded). The first allocating scheme is $\eta_{j,t} = p_{j,t} = p_t(j \mid \Theta) = p(j \mid z_t, x_t)$, i.e., same as the E step in Algorithm I. Actually, Algorithm II at this setting is an adaptive EM algorithm for ML learning (Xu, 1998). The second scheme only takes values at the winner and the rival, while the negative value of $\eta_{j,t} = p_{j,t} = -\gamma$ reverses the updating direction such that the rival is de-learned a little bit to reduce its fitness $H_t(\Theta_r)$ to the current sample x_t. It extends the RPCL learning for NRBF from a two stage implementation (Xu, Krzyzak, &Oja, 1992&93) to an improved implementation that updates all the unknowns jointly (Xu, 1998). One problem to the RPCL learning is how to choose an appropriate $\gamma > 0$. The third scheme $\eta_{j,t} = p_{j,t}(1 + \delta h_{t,j})$ shares the features of both the EM learning and RPCL learning without needing a pre-specified $\gamma > 0$, which is derived from the BYY harmony learning to be introduced in the sequel.

BYY HARMONY LEARNING: FUNDAMENTALS

As shown in the left –top corner of Figure 4, a set $\mathbf{X} = \{x\}$ of samples are regarded as generated via a top-down path from its inner representation $\mathbf{R} = \{\mathbf{Y}, \Theta\}$, with a long term memory Θ that is a collection of all unknown parameters in the system for collectively representing the underlying structure of $\mathbf{X}$, and with a short term memory $\mathbf{Y}$ that each element $y \in \mathbf{Y}$ is the corresponding inner representation of one element $x \in X$. A mapping $\mathbf{R} \to \mathbf{X}$ and an inverse $\mathbf{X} \to \mathbf{R}$ are jointly considered via the joint

Figure 2. Algorithm 2: Adaptive learning algorithm for alternatve ME and NRBF nets

$H_t(\Theta_j) = \pi_t(\Theta_j) - \frac{1}{2} Tr[h_x^2(\Pi_j^x + \Pi_j^{z|x}) + h_z^2 \Pi_j^z] + \frac{1}{N} \ln\{q(h_x) q(h_z)\}]$, $\Pi_j^x = -\nabla_x^2 \ln q(x|\varphi_j)$,
$\pi_t(\Theta_j) = \ln[\alpha_j q(x_t|\varphi_j) q(z_t|x_t,\theta_j)]$, $\Pi_{z,j}^x = -\nabla_x^2 \ln q(z|x,\theta_j), \Pi_{z,j}^z = -\nabla_z^2 \ln q(z|x,\theta_j)$,
where $\nabla_x^2 f(x) = \partial^2 \ln f(x) / \partial x \partial x^T$, $Tr[A]$ denotes the trace of a matrix A

(a)

We maximize $\sum_{j=1}^k p_t(j|\Theta) H_t(\Theta_j)$ by gradient based rules in Fig. 3(E) via
$dH_0(p \| q, \Theta, \mathbf{k}, \Xi) = \sum_{j=1}^k p_{j,t}(1 + \delta h_{t,j}) d\pi(x_t, \Theta_j)$,
$\delta h_{t,j} = \pi(x_t, \Theta_j) - \sum_{\ell=1}^k p_{\ell,t} \pi(x_t, \Theta_\ell)$, $p_t(j|\Theta) = \exp[\pi_t(\Theta_j)] / \sum_{\ell=1}^k \exp[\pi_t(\Theta_\ell)]$.

(b)

This maximizati on is made by iterating the following two steps untill converged :

YING STEP Choose one allocating scheme in Fig.3(A) and
get $p_{j,t}, \delta h_{j,t}$ based on the current value of Θ.

YANG STEP Let $\eta_{j,t} = p_{j,t}(1 + \delta h_{j,t})$, update :

(a) $\alpha_j^{new} = e^{c_j^{old} + \Delta c_j} / \sum_\ell e^{c_\ell^{old} + \Delta c_\ell}$, $\mathbf{c} = [c_1, \ldots, c_k]^T$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_k]^T$, $\mathbf{1} = [1, \ldots, 1]^T$,
$\Delta \mathbf{c} \propto (I - \boldsymbol{\alpha} \mathbf{1}^T) g_{\boldsymbol{\alpha}}$, $g_{\boldsymbol{\alpha}} = diag[p_{1,t}(1 + \delta h_{1,t}), \ldots, p_{k,t}(1 + \delta h_{k,t})]$,
$(\because \sum_{j=1}^k p_{j,t}(1 + \delta h_{j,t}) d \ln \alpha_j = Tr[g_{\boldsymbol{\alpha}}{}^T d\boldsymbol{\alpha}] = Tr[g_{\boldsymbol{\alpha}}{}^T (I - \mathbf{1}\boldsymbol{\alpha}^T) d\mathbf{c}]$)
or simply $\alpha_j = |\Sigma_j|^{0.5} / \sum_\ell |\Sigma_\ell|^{0.5}$ for NRBF or ENRBF.
If a $\alpha_j \to 0$, discard the corresponding structure and its Θ_j.

(b) $\varphi_j + \Delta\varphi_j \in D_{\varphi_j}$ with $\Delta\varphi_j \propto \eta_{j,t} \nabla_{\varphi_j \in D_{\varphi_j}} \ln q(x_t|\varphi_j)$;

(c) $\theta_j + \Delta\theta_j \in D_{\theta_j}$ with $\Delta\theta_j \propto \eta_{j,t} \nabla_{\theta_j \in D_{\theta_j}} \ln q(z_t|x_t,\theta_j)$,
where $u + \Delta u \in D_u$ means `updating within the domain D_u of u'.

(d) $h_z^{new} = h_z^{old} + \Delta h_z$ with $\Delta h_z \propto \{\frac{1}{N} q'(h_z) - h_z \sum_{j=1}^k p_{j,t} Tr[\Pi_j^z]\}$, $f'(r) = \frac{df(r)}{dr}$,
$h_x^{new} = h_x^{old} + \Delta h_x$ with $\Delta h_x \propto \{\frac{1}{N} q'(h_z) - h_x \sum_{j=1}^k p_{j,t} Tr[\Pi_j^x + \Pi_j^{z|x}]\}$.

(c)

distribution of **X**, **R** in two types of Bayesian decomposition. In a compliment to the famous ancient Ying-Yang philosophy, the decomposition of $p(\mathbf{X}, \mathbf{R})$ coincides the Yang concept with a visible domain $p(\mathbf{X})$ for a Yang space and a forward pathway by $p(\mathbf{R} \mid \mathbf{X})$ as a Yang pathway. Thus, $p(\mathbf{X}, \mathbf{R})$ is called Yang machine. Also, $q(\mathbf{X}, \mathbf{R})$ is called Ying machine with an invisible domain $q(\mathbf{R})$ for a Ying space and a backward pathway by $q(\mathbf{X} \mid \mathbf{R})$ as a Ying pathway. Such a Ying-Yang pair is called Bayesian Ying-Yang (BYY) system. It further consists of two layers. The front layer is itself a Ying-Yang pair for $\mathbf{X} \to \mathbf{Y}$ and $\mathbf{Y} \to \mathbf{X}$. The back layer supports the front layer by a priori $q(\Theta \mid \Xi)$, while $p(\mathbf{R} \mid \mathbf{X})$ consists of the posteriori $p(\Theta \mid \mathbf{X}, \Xi)$ that transfers the knowledge from observations to the back layer.

The input to the Ying Yang system is through $p(\mathbf{X} \mid \Theta_x) = p(\mathbf{X} \mid \mathbf{X}_N, h)$ obtained from a sample set $\mathbf{X}_N = \{x_t\}_{t=1}^N$, e.g., $p(\mathbf{X} \mid \mathbf{X}_N, h) = G(\mathbf{X} - \mathbf{X}_N \mid 0, h^2 I)$. To build up an entire system, we need to

Figure 3. Allocating schemes, typical examples, and gradient based updating

EM	RPCL	BYY
$\delta h_{t,j} = 0$	$\delta h_{t,j} = 0$	$\delta h_{t,j} = \pi(x_t, \Theta_j) - \sum_{\ell=1}^{k} p_t(j \mid \Theta)\pi(x_t, \Theta_j)$
$p_{j,t} = p_t(j \mid \Theta)$	$p_{j,t} = \overline{\delta}_{j,\ell^*} - \gamma\overline{\delta}_{j,r^*}$	$p_{j,t} = p_t(j \mid \Theta)$

$$p_t(j \mid \Theta) = \frac{e^{\pi_t(\Theta_j)}}{\sum_{\ell=1}^{k} e^{\pi_t(\Theta_\ell)}}, \ \overline{\delta}_{i,\ell} = \begin{cases} 1, & i = \ell, \quad \ell^* = \arg\max_j \pi(x_t, \Theta_j); \\ 0, & i \neq \ell, \quad r^* = \arg\max_{j \neq \ell^*} \pi(x_t, \Theta_j). \end{cases}$$

(A)

$$q(x \mid \varphi_j) = G(x \mid \mu_j, \Sigma_j), \Pi_j^x = \Sigma_j^{-1}, \ \eta_{j,t} = p_{j,t}(1 + \delta\, h_{t,j}), \Delta\mu_j \propto \eta_{j,t} e_{j,t}, \ e_{j,t} = x_t - \mu_j,$$

$$\Sigma_j = S_j S_j^T, G_{\Sigma_j} = \Sigma_j^{-1}(e_{j,t}e_{j,t}^T + 0.5h_x^2 I - \Sigma_j)\Sigma_j^{-1}, \Delta S_j \propto \eta_{j,t}(e_{j,t}e_{j,t}^T + 0.5h_x^2 I - \Sigma_j)S_j^{-T}$$

(B)

$$q(z \mid x, \theta_j) = G(z \mid W_j^{z|x} x + w_j^{z|x}, \Sigma_j^{z|x}), \ \Pi_j^z = \Sigma_j^{z|x\,-1}, \ \ \Pi_j^{z|x} = W_j^{z|x} \Sigma_j^{z|x\,-1} W_j^{z|x}$$

$$\Delta w_j^{z|x} \propto \eta_{j,t}\varepsilon_{j,t}, \ \ \varepsilon_{j,t} = z_t - (W_j^{z|x} x_t + w_j^{z|x}), \ \ \Delta W_j^{z|x} \propto \eta_{j,t}\varepsilon_{j,t} x_t^T - h_z^2 W_j^{z|x}, \ \ \Sigma_j^{z|x} = \Gamma_j \Gamma_j^T,$$

$$G_{\Sigma_j^{z|x}} = \Sigma_j^{z|x\,-1}(\varepsilon_{j,t}\varepsilon_{j,t}^T + h_z^2 I - \Sigma_j^{z|x})\Sigma_j^{z|x\,-1},$$

$$\Delta\Gamma_j \propto \eta_{j,t}\Sigma_j^{z|x} G_{\Sigma_j^{z|x}} \Gamma_j = \eta_{j,t}(\varepsilon_{j,t}\varepsilon_{j,t}^T + h_z^2 I - \Sigma_j^{z|x})\Gamma_j^{-T}.$$

(C)

$$q(z \mid x, \theta_j) = q(z = \ell \mid x, \theta_j) = e^{\overline{z}_j^{(\ell)}} / \sum_\ell e^{\overline{z}_j^{(\ell)}} = q_j^{(\ell)}, \ \ \overline{z}_j = [\overline{z}_j^{(1)}, \ldots, \overline{z}_j^{(k)}]^T$$

$$\mathbf{q}_j = [q_j^{(1)}, \ldots, q_j^{(k)}]^T, \ \ \overline{z}_j = W_j^{z|x} x + w_j^{z|x} - \tfrac{1}{2}[x^T V_{j,1}^{z|x} x, \ldots, x^T V_{j,k}^{z|x} x]^T$$

$$\Delta w_j^{z|x} \propto \eta_{j,t}\varepsilon_{j,t}, \ \ \varepsilon_{j,t} = [\overline{\delta}_{1,\ell}, \ldots, \overline{\delta}_{k,\ell}]^T - \mathbf{q}_{j,t}, \ \ q_{j,t}^{(\ell)} = q(z_t = \ell \mid x_t, \theta_j)$$

$$\Delta V_{j,i}^{z|x} \propto -\eta_{j,t}\varepsilon_{j,t}^{(i)} x_t x_t^T, \ \ \Delta W_j^{z|x} \propto \eta_{j,t}\{\varepsilon_{j,t} x_t^T - h_z^2(\mathrm{diag}[\mathbf{q}_{j,t}] - \mathbf{q}_{j,t}\mathbf{q}_{j,t}^T) W_j^{z|y}\}$$

(D)

Max/incr $f(\theta, \Sigma)$ with a vector θ and a positive definite matrix Σ via

$$\mathrm{g}_\theta = \nabla_\theta f, \ \mathrm{G}_\Sigma = \nabla_\Sigma f. \quad We \ \ update \ \ \theta^{new} = \theta^{old} + \Delta\theta \ \ with$$

- Gradient updating: $\Delta\theta \propto \mathrm{g}_\theta$ which means $\Delta\theta = \gamma \mathrm{g}_\theta$ with a small scalar γ.
- Projected Gradient : $\Delta\theta \propto P\mathrm{g}_\theta$ by a positive definite P, with $\mathrm{g}_\theta^{\mathrm{T}} P \mathrm{g}_\theta > 0$.

  $update \ \ \Sigma = SS^T \ \ via \ \ S^{new} = S^{old} + \Delta S \ \ to$ keep Σ positive definite.
- Gradient updating: $\Delta S \propto \mathrm{G}_S, \mathrm{G}_S = \mathrm{G}_\Sigma S$, since $Tr[G_\Sigma^T d\Sigma] = 2Tr[(\mathrm{G}_\Sigma S)^T dS]$.
- Projected Gradient : $\Delta S \propto P\mathrm{G}_S$ or $\Delta S \propto \mathrm{G}_S P$ by a positive definite P.

  $(\because Tr[\mathrm{G}_S{}^T P \mathrm{G}_S] = \mathrm{vec}(\mathrm{G}_S)^T (I \otimes P) vec(\mathrm{G}_S) > 0)$
- Bi-projected Gradient : $\Delta S \propto P\mathrm{G}_S P$ by a positive definite P.

  $(\because Tr[\mathrm{G}_S{}^T P \mathrm{G}_S P] = \mathrm{vec}(\mathrm{G}_S)^T (P \otimes P) vec(\mathrm{G}_S) > 0)$

Remarks : (a) $vec(A)$ stacks all the colums of A into a vector, and $\otimes$ is Kronecker product.

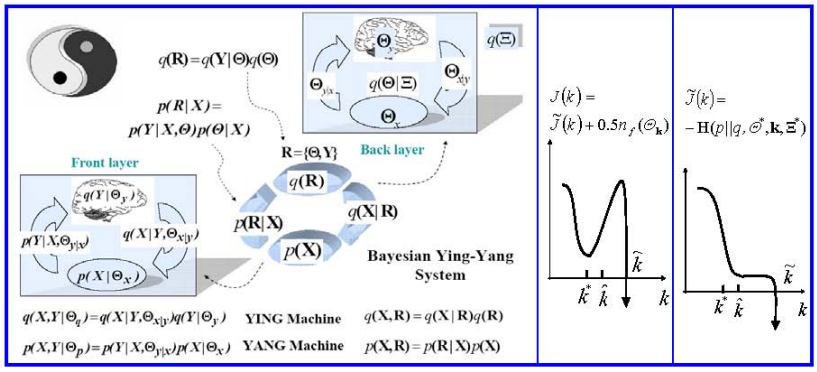
(b) P is chosen to simplify computation, and may speed up convergence too.

(E)

design appropriate structures for the rests in the system. Different designs perform different learning tasks and thus typical learning models are unified under a general framework. As shown in Figure 4, designs are guided by the following three principles, respectively:

- **Least Redundancy Principle** Subject to the nature of learning tasks, $q(\mathbf{R})$ should be in a structure for the inner representation of $\mathbf{X}_N$ encoded with a redundancy as least as possible. First, the number of variables and parameters of $\mathbf{R} = \{\mathbf{Y}, \Theta\}$ should be as less as possible, which is itself the model selection task that is beyond the task of design. Second, the dependences among these

Figure 4. Bayesian Ying-Yang system and best harmony learning



(a) **(b)**

Distributive Measure		Collective Measure (entropy)	
Likelihood based	$U(p)\begin{cases}\propto \ln p(u)+c, & \text{(a)}\\ =\ln p(u), & \text{(b)}\end{cases}$	**Renyi**	$U(p)=\frac{1}{1-\alpha}\log\int p^{\alpha}(u)du$
Hessian based	$U(p)\begin{cases}\propto \nabla^2_{\mathbf{u}\mathbf{u}^T}\ln p(u), & \text{(a)}\\ =\nabla^2_{\mathbf{u}\mathbf{u}^T}\ln p(u), & \text{(b)}\end{cases}$	**Shannon**	$U(p)=-\int p(u)\ln p(u)du$

where "$\propto$" means proportional, and $\nabla^2_{uu^T}f(u)=\partial^2 f(u)/\partial u\partial u$ is applicable to a twice differentiable $f(u)$. Both the notations will be used subsequently.

Remarks:

1. "$\propto$" is not meaningful to a collective measure since it is just a real scalar but provides a relaxation on a distributive measure.
2. $U(p)=U(q)$ means $p=aq^b$ with $\int p=1$ for Likelihood (a), and we get $a=1,b=1$ for Likelihood (b).
3. In a Hessian measure, $U(p)=U(q)$ at point u^* means that p,q get a local uncertainty conversation in a neighbor of u^* in a sense of equal local convexity.
4. Likelihood based implies Hessian based if p,q are twice differentiable. Also, Renyi implies Shannon, and Renyi reduces to Shannon for $\alpha=1$.

(c)

variables should be as least as possible, which is possibly or at least partly to be considered. E.g., when **Y** consists of multiple components $\mathbf{Y}=\{Y^{(j)}\}$, we design $q(\mathbf{Y}\mid\Theta)=\prod_j q(Y^{(j)}\mid\theta_y^{(j)})$.

- **Divide and Conquer Principle** Subject to the representation formats of **X** and **Y**, a complicated mapping $\mathbf{Y}\to\mathbf{X}$ is modeled by designing $q(\mathbf{X}\mid\mathbf{R})=q(\mathbf{X}\mid\mathbf{Y},\Theta)$ via a mixture of a number of simple structures. E.g., we may design $q(\mathbf{X}\mid\mathbf{Y},\Theta)$ via a mixture of a number of linear regression featured by Gaussians of **Y** conditional on **X**. In some situation, it is not necessary to design $q(\mathbf{R})$ and $q(\mathbf{X}\mid\mathbf{R})$ separately. Instead, we design $q(\mathbf{X},\mathbf{R})$ (especially $q(\mathbf{X},\mathbf{Y},\Theta_q)$) via a single parametric model as a whole but still attempting to follow the above two principles. One example is a product Gaussian mixture, as the one encountered in Type B of Rival penalized competitive learning (see Xu, 2007b). In general, we may consider an integrable measure $M(\mathbf{X},\mathbf{Y},\Theta_q)$ by

$$q(\mathbf{X},\mathbf{Y},\Theta_q) = M(\mathbf{X},\mathbf{Y},\Theta_q) \,/\, \int M(\mathbf{X},\mathbf{Y},\Theta_q) d\mathbf{X} d\mathbf{Y}$$

- **Uncertainty Conversation Principle** In a compliment to the Ying-Yang philosophy, Ying is primary and thus is designed first, while Yang is relatively secondary and thus is designed basing on Ying. Moreover, as illustrated by the Ying-Yang sign located at the top-left of Figure 4, the room of varying or dynamic range of Yang should balance with that of Ying, which motivates to design $p(\mathbf{X}, \mathbf{R})$ (in fact, only $p(\mathbf{R} \mid \mathbf{X})$ because we have $p(\mathbf{X} \mid \Theta_x) = p(\mathbf{X} \mid \mathbf{X}_N, h)$ already) under a principle of uncertainty conversation between Ying-Yang. In other words, Yang machine preserves a varying room or dynamic range that is appropriate to accommodate uncertainty or information contained within the Ying machine. That is $U(p(\mathbf{X},\mathbf{R})) = U(q(\mathbf{X},\mathbf{R}))$ under a uncertainty measure $U(p)$, in one of the choices within the table in Figure 4. Since a Yang machine consists of two components $p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X})$, we may consider one or both of the uncertainty conversations as follows:

$$U(p(\mathbf{R} \mid \mathbf{X})) = U(q(\mathbf{R} \mid \mathbf{X})), \quad q(\mathbf{R} \mid \mathbf{X}) = q(\mathbf{X},\mathbf{R}) \,/\, q(\mathbf{X}) \text{ and}$$
$$U(p(\mathbf{X})) = U(q(\mathbf{X})), \; q(\mathbf{X}) = \int q(\mathbf{X},\mathbf{R}) d\mathbf{R}$$

The above uncertainty conservation may occur at three different levels. One is likelihood based conservation that the structure of Yang machine gets a strong link to the ones of Ying machine. Considering $\sum_L p(L \mid \mathbf{X}, \theta_{L|X}) = 1$, the Yang structure is actually given by the Bayesian inverse of Ying machine $p(L \mid \mathbf{X}, \theta_{L|X}) = q(\mathbf{X},\mathbf{R}) \,/\, \sum_L q(\mathbf{X},\mathbf{R})$, further examples are referred to Tab.2 of (Xu, 2008a). A less constrained link is provided by a Hessian based conversation, i.e., a conservation on local convexity, which is applicable to $\mathbf{Y}, \Theta$ of real variables. This is a type of local information conservation, e.g., Fisher information conservation. The most relaxed conservation is based on a collective measure instead of details, e.g., Shannon entropy.

Again, we use $S_{\mathbf{k}}(\Theta_{\mathbf{k}})$ to denote a family of system specifications, with $\mathbf{k}$ featured by the scale or complexity for representing $\mathbf{R}$, which is contributed by the scale $\mathbf{k}_{\mathbf{Y}}$ for representing $\mathbf{Y}$ and an effective number standing for free parameters in $\Theta_{\mathbf{k}}$. An analogy of this Ying Yang system to the Chinese ancient Ying-Yang philosophy motivates to determine $\mathbf{k}$ and $\Theta_{\mathbf{k}}$ under **the best harmony principle**, mathematically to maximize

$$H(p \,||\, q, \mathbf{k}, \Xi) = \int p(\mathbf{R} \mid \mathbf{X}) p(\mathbf{X} \mid \mathbf{X}_N, h) \ln[q(\mathbf{X} \mid \mathbf{R}) q(\mathbf{R})] d\mathbf{X} d\mathbf{R}. \tag{8}$$

On one hand, the maximization forces $q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})$ to match $p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X} \mid \mathbf{X}_N, h)$. In other words, $q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})$ attempts to describe the data $p(\mathbf{X} \mid \mathbf{X}_N, h)$ in help of $p(\mathbf{R} \mid \mathbf{X})$, which uses actually $q(\mathbf{X}) = \int q(\mathbf{X} \mid \mathbf{R}) q(\mathbf{R}) d\mathbf{R}$ to fit $p(\mathbf{X} \mid \mathbf{X}_N, h)$ not in a maximum likelihood sense but with a promising model selection nature. Due to a finite size of $\mathbf{X}_N$ and structural constraint of $p(\mathbf{R} \mid \mathbf{X})$, this matching aims at (but may not really reach) $q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R}) = p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X} \mid \mathbf{X}_N, h)$. Still we get a trend at this equality by which $H(p \,||\, q, \mathbf{k}, \Xi)$ becomes the negative entropy, and its further maximization is minimizing the system complexity, which consequently provides a model selection nature on $\mathbf{k}$.

At the first glance, one may feel the formulae eqn.(8) somewhat familiar. In fact, its degenerated case with **R** vanished leads to $H(p \,||\, q) = \int p(\mathbf{X}) \ln q(\mathbf{X}) d\mathbf{X}$. With $p(\mathbf{X}) = p(\mathbf{X} \mid \mathbf{X}_N, h)$ at $h = 0$ and $q(\mathbf{X}) = q(\mathbf{X} \mid \Theta)$, maximizing this $H(p \,||\, q)$ becomes $\max_\Theta \ln q(\mathbf{X}_N \mid \Theta)$, i.e., maximum likelihood (ML) learning. The situation with $p(\mathbf{X})$ beyond $p(\mathbf{X} \mid \mathbf{X}_N, h)$ was also explored in the signal processing literature, via $\min_{q(\mathbf{X})} - H(p \,||\, q)$ under the name of Minimum Cross-Entropy (Minxent). It was noticed that $\min_{p(\mathbf{X})} - H(p \,||\, q)$ leads to a singular result that $p(\mathbf{X}) = \delta(\mathbf{X} - \mathbf{X}^*)$, $\mathbf{X}^* = \arg\max_{\mathbf{X}} \ln q(\mathbf{X} \mid \Theta)$, which was regarded as irregular and not useful. Instead, efforts were turned to minimizing the classic Kullback–Leibler divergence $KL(p \,||\, q) = H(p \,||\, p) - H(p \,||\, q)$ with respect to p, which pushes p to match q and thus the above singular result is avoided. Moreover, if p is given, $\min_{q(\mathbf{X})} KL(p \,||\, q)$ is still equivalent to $\min_{q(\mathbf{X})} - H(p \,||\, q)$. Thereafter, Minimum Cross-Entropy is usually used to refer this $\min KL(p \,||\, q)$.

Interestingly, with an inner representation **R** considered in the BYY system, the scenario becomes different from the above classic situation. With $p(\mathbf{X}) = p(\mathbf{X} \mid \mathbf{X}_N, h)$ fixed, $\max_{p(\mathbf{X})} H(p \,||\, q)$ is made with respect to only $p(\mathbf{R} \mid \mathbf{X})$, which is no longer useless but responsible to the promising least complexity nature discussed after eqn.(8). In other words, maximizing $\max H(p \,||\, q)$ with respect to unknowns not only in Ying part $q(\mathbf{X}, \mathbf{R})$ but also in Yang part $p(\mathbf{X}, \mathbf{R})$ makes the Ying-Yang system become a best harmony. Alternatively, we may regard such a mathematical formulation as an information theoretic interpretation of the ancient Ying Yang philosophy.

In implementation, $H(p \,||\, q, \mathbf{k}, \Xi) = \int p(\Theta \mid \mathbf{X}_N) H(p \,||\, q, \Theta, \mathbf{k}, \Xi) d\Theta$ can be approximated via a Taylor expansion of $H(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ around Θ^* up to the second order as follows:

$$H(p \,||\, q, \mathbf{k}, \Xi) = \int p(\Theta \mid \mathbf{X}_N) H(p \,||\, q, \Theta, \mathbf{k}, \Xi) d\Theta \approx \mathrm{H}(p \,||\, q, \Theta^*, \mathrm{k}, \Xi) + 0.5 d_{\mathbf{k}}(\Xi),$$
$$\Theta^* = \arg\max_\Theta H(p \,||\, q, \Theta, \mathbf{k}, \Xi), \quad d_k(\Xi) = Tr[\Gamma\ \Omega(\Theta^*)] + (\zeta - \Theta^*)^T \Omega(\Theta^*, \Xi)(\zeta - \Theta^*),$$

$$H(p \,||\, q, \Theta, \mathbf{k}, \Xi) = H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi) + \ln q(\Theta \mid \Xi),$$

$$H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi) = \int p(\mathbf{Y} \mid \mathbf{X}, \Theta_{\mathrm{y|x}}) p(\mathbf{X} \mid \mathbf{X}_N, h) \ln [q(\mathbf{X} \mid \mathbf{Y}, \Theta_{\mathrm{x|y}}) q(\mathbf{Y} \mid \Theta_{\mathrm{y}})]\ d\mathbf{X} d\mathbf{Y} + \ln q(h),$$

$$\Omega(\Theta\ \Xi) = \nabla^2_\Theta\ H(p \,||\, q, \Theta, k, \Xi),\ \zeta = \int \Theta p(\Theta \mid \mathbf{X}_N)\ d\Theta,\ \Gamma = \int (\Theta - \zeta)(\Theta - \zeta)^T p(\Theta \mid \mathbf{X}_N)\, d\Theta. \tag{9}$$

The maximization of $H(p \,||\, q, \mathbf{k}, \Xi)$ consists of an inner maximization that seeks the best harmony among the front layer Ying-Yang supported by a priori $q(\Theta \mid \Xi)$ and an outer maximization that seeks the best harmony of the entire Ying Yang system with $\mathrm{d_k}(\Xi)$ taken in consideration for the interaction between the front and back layers.

BYY HARMONY LEARNING: CHARACTERISTICS AND IMPLEMENTATIONS

Systematically considering two pathways and two domains coordinately as a BYY system under a probability theoretic ground and designing three system components under the principles of *least redundancy*,

divide-conquer, and *uncertainty conversation*, respectively. BYY harmony learning is different not only from cognitive science motivated adaptive resonance theory (Grossberg, 1976; Grossberg & Carpenter, 2002) and the least mean square error reconstruction based auto-association (Bourlard & Kamp, 1988) and LMSER self-organization (Xu, 1991 & 93), but also from those probability theoretic approaches that either merely consider a bottom-up pathway as the inverse of a top-down pathway (e.g., Bayesian approaches), or approximate such inverses for tackling intractable computations (e.g., Helmholtz machine, varianional Bayes, and extensions). Mathematically implementing the Ying-Yang harmony philosophy, in a sense that Ying and Yang not only matches each other but also seeks a best match in a most compact way, by determining all unknowns in a BYY system via maximizing the functional of $H(p \,||\, q, \mathbf{k}, \Xi)$, with the model selection nature explained after eqn.(8).

Readers are further referred to (Xu, 2007a) for a systematical overview on relations of Bayesian Ying Yang learning to a number of typical approaches for either or both of parameter learning and model selection, covering the following aspects:

- **Best Data Matching perspective,** for modeling a latent or generative model and involving ML, Bayesian, AIC, BIC, MDL, MML, marginal likelihood, etc;
- **Best Encoding perspective,** for encoding inner representation by a bottom-up pathway (or called a transformation / recognition / representative model) and involving INFOMAX, MMI based ICA approaches and extensions;
- **Two Pathway Perspective,** involving information geometry based em-algorithm, Helmholtz Machine, variational approximation, and bits-back based MDL, etc.
- **Optimal Information Transfer Perspective**, involving MDL, MML, and bits-back based MDL, etc.

The model selection nature of maximizing $H(p \,||\, q, \mathbf{k}, \Xi)$ can also be observed from its gradient follow with the Yang path in a Bayesian structure $p(\mathbf{R} \mid \mathbf{X}) = q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R}) \,/\, q(\mathbf{X})$, $q(\mathbf{X}) = \int q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})d\mathbf{R}$. It follows that we have

$$dH(p \,||\, q, \mathbf{k}, \Xi) = \int p(\mathbf{R} \mid \mathbf{X})[1 + \delta_L(\mathbf{X}, \mathbf{R})]dL(\mathbf{X}, \mathbf{R})p(\mathbf{X} \mid \mathbf{X}_N, h)d\mathbf{X}d\mathbf{R},$$
$$\delta_L(\mathbf{X}, \mathbf{R}) = L(\mathbf{X}, \mathbf{R}) - \int p(\mathbf{R} \mid \mathbf{X})L(\mathbf{X}, \mathbf{R})d\mathbf{R}, \text{ and } \; L(\mathbf{X}, \mathbf{R}) = \ln[q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})] \tag{10a}$$

Noticing that $L(\mathbf{X}, \mathbf{R})$ describes the fitness of an inner representation $\mathbf{R}$ on the observation $\mathbf{X}$, we observe that $\delta_L(\mathbf{X}, \mathbf{R})$ indicates whether the considered $\mathbf{R}$ fits $\mathbf{X}$ better than the average of all the possible choices of $\mathbf{R}$. Letting $\delta_L(\mathbf{X}, \mathbf{R}) = 0$, the rest of $dH(p \,||\, q, \mathbf{k}, \Xi)$ is actually the updating flow of the M step in the EM algorithm for the maximum likelihood learning (McLachlan & Geoffrey,1997). Usually $\delta_L(\mathbf{X}, \mathbf{R}) \neq 0$, i.e., the gradient flow $dL(\mathbf{X}, \mathbf{R})$ under all possible choices of $\mathbf{R}$ is integrated via weighting not just by $p(\mathbf{R} \mid \mathbf{X})$ but also by a modification of a relative fitness measure $1 + \delta_L(\mathbf{X}, \mathbf{R})$. If $\delta_L(\mathbf{X}, \mathbf{R}) > 0$, updating goes along the same direction of the EM learning with an increased strength. If $0 > \delta_L(\mathbf{X}, \mathbf{R}) > -1$, i.e., the fitness is worse than the average and the current $\mathbf{R}$ is doubtful, updating still goes along the same direction of the EM learning but with a reduced strength. When $-1 > \delta_L(\mathbf{X}, \mathbf{R})$, updating reverses the direction of the EM learning and actually becomes de-learning. In other words, the BYY harmony learning shares the same nature of RPCL learning but with an improvement that

there is no need on a pre-specified de-learning strength $\gamma > 0$, as previously discussed in Figure 3(A) on a special case of NRBF & ME with $\mathbf{R} = \{j, \Theta\}$, where $p(\mathbf{R} \mid \mathbf{X})[1 + \delta_L(\mathbf{X}, \mathbf{R})]$ is simplified into $p_{j,t}(1 + \delta h_{t,j})$ and $\delta_L(\mathbf{X}, \mathbf{R})$ into $\delta h_{t,j}$.

More specifically, the model selection nature of Bayesian Ying Yang learning possess the following favorable promising features.

- The conventional model selection approaches aim at model complexity conveyed at either or both of the level of structure S_k and the level of parameter set Θ. This task is difficult to estimate, usually resulting in some rough bounds. Bayesian Ying Yang learning considers not only the levels of S_k and Θ but also the level of short memory representation **Y** by the front layer of BYY system in Figure 4(a). That is, the scale **k** of the BYY system is considered with the part $\mathbf{k}_Y$ for representing **Y**, while the rest part, contributed by S_k and Θ, is estimated via ln $q(\Theta|\Xi)$ and $d_{\mathbf{k}}(\Xi)$ in eqn.(9), which is along a line similar to the above conventional approaches. Interestingly, the part $\mathbf{k}_Y$ is modeled via $q(\mathrm{Y}|\Theta_y)$ in $H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi)$, which is estimated more accurately than the rest part. Promisingly, the model selection problems of many typical learning tasks can be reformulated into selecting merely the $\mathbf{k}_Y$ part in a BYY system (Xu, 2005). Therefore, the resulted BYY harmony criterion $J(\mathbf{k})$ shown by the left one of Figure 4(b) may considerably improve the performances by typical model selection approaches, which has been shown by in experiments (Shi, 2008).
- Even interestingly, this $\mathbf{k}_Y$ part associates with a set $\tilde{\Theta}_y \subseteq \Theta_y$ of parameters on which there exists an indicator $\rho(\tilde{\Theta}_y)$. Maximizing $H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ will exert a force that pushes $\rho(\tilde{\Theta}_y) \to 0$, which means that its associated contribution to $\mathbf{k}_Y$ can be discarded. E.g., each j of k values associates with one α_j, we can discard a structure if its correspondent $\rho(\alpha_j) = \alpha_j \to 0$. As a result, k effectively reduces to $k - 1$. As illustrated on the right of Figure 4(b), $\alpha_j \to 0$ means its contribution to $\tilde{J}(k)$ is 0, and a number of such parameters becoming 0 result in that $\tilde{J}(k)$ has effectively no change on a range $[\hat{k}, \tilde{k}]$. Also, each dimension $y^{(i)}$ of $q(y \mid \theta_j^y)$ associates with its variance λ_j and this dimension can be discarded if $\lambda_j \to 0$. As illustrated beyond $\tilde{k}$ on the right of Figure 4(b), such a parameter becoming 0 contributes to $\tilde{J}(k)$ by $-\infty$. As long as $\mathbf{k}_Y$ is initialized at a big enough value, $\hat{\mathbf{k}}$ can be found as an upper bound estimate of $\mathbf{k}^*$. That is, an automatic model selection is incurred during parameter learning. Details are referred to Sec.2.3 of (Xu, 2008a&b).
- The separated consideration of $\mathbf{k}_Y$ from the rest of **k** also provides a general framework that integrates the roles of regularization and model selection, such that not only the automatic model selection mechanism on $\mathbf{k}_Y$ can avoid the previously mentioned disturbance by a regularization with an inappropriate priori $q(\Theta \mid \Xi)$, but also imprecise approximations caused by handling the integrals may be alleviated via regularization. Specifically, model selection is made via $q(\mathbf{Y} \mid \Theta_y)$ in Ying machine, while regularization is imposed in Yang machine via designing its structure under a uncertainty conservation principle given at the bottom of Figure 4 and making *data smoothing regularization* via $p(\mathbf{X}) = p(\mathbf{X} \mid \mathbf{X}_N, h)$ with $h \neq 0$ that takes a role similar to regularization strength, while the difficulty of the conventional regularization approaches on controlling this strength has been avoided because an appropriate $h \neq 0$ is also determined during maximizing $H(p \,||\, q, \Theta, \mathbf{k}, \Xi)$.

At a first glance, a scenario with $q(\mathbf{Y} \mid \Theta_{y})$ is seemly involved also in several typical learning approaches, especially those with an EM like two pathway implementation, such as the EM algorithm implemented ML learning (Redner & Walker, 1984), information geometry based em-algorithm (Amari, 1995), Helmholtz Machine (Hinton, Dayan, Frey, & Neal, 1995; Dayan, 2002), variational approximation (Jordan, Ghahramani, Jaakkola, & Saul, 1999), the bits-back based MDL (Hinton & Zemel, 1994), etc. Actually, these studies have neither put $q(\mathbf{Y} \mid \Theta_{y})$ in a role for describing $\mathbf{k}_{Y}$ nor sought for the above nature of automatic model selection. Instead, the role of $q(\mathbf{Y} \mid \Theta_{y})$ in these studies is estimating $q(\mathbf{X}) = \int q(\mathbf{X} \mid \mathbf{Y}, \Theta_{x|y}) q(\mathbf{Y} \mid \Theta_{y}) d\mathbf{Y}$ in a ML sense or approximately, which is similar to the one in Bayesian Kullback Ying Yang (BKYY) learning that not only accommodates a number of typical statistical learning approaches (including these studies) as special cases but also provides a bridge to understand the relation and difference from the best harmony learning by maximizing $H(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ in eqn.(8). Proposed in (Xu,1995), BKYY learning performs a best Ying Yang matching by minimizing:

$$KL(p \,||\, q, \mathbf{k}, \Xi) = \int p(\mathbf{R} \mid \mathbf{X}) p(\mathbf{X} \mid \mathbf{X}_{N}, h) \ln \frac{p(\mathbf{R} \mid \mathbf{X}) p(\mathbf{X} \mid \mathbf{X}_{N}, h)}{q(\mathbf{X} \mid \mathbf{R}) q(\mathbf{R})} d\mathbf{X} d\mathbf{R} - H(p \,||\, q, \mathbf{k}, \Xi), \tag{10b}$$

that is, a best Ying Yang harmony includes not only a best Ying Yang matching as a part but also minimizing the entropy or the complexity of a Yang machine. In other words, it seeks a Yang machine that not only best matches the Ying machine but also keeps itself in a least complexity.

Considering a learning system in a Ying-Yang pair naturally motivates to implement the maximization of $H(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ or the minimization of $KL(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ by an alternative iteration of

- **Yang step**: fixing all the unknowns in the Ying machine, we update the rest of the unknowns in the Yang machine (after excluding those common unknowns shared by the Ying machine);
- **Ying step**: fixing those just updated unknowns in the Yang step, we update all the unknowns in the Ying machine.

Not only this iteration is guaranteed to converge, but also it includes the well known EM algorithm and provides a general perspective for developing other EM-like algorithms.

Recalling eqn.(9), the maximization of $H(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ can be approximately decoupled into an inner maximization for the best harmony among the front layer supported by a priori $q(\Theta \mid \Xi)$ and an outer maximization for interaction between the front and back layers. Thus, we have the following two stage implementation:

$$\begin{aligned}
&\text{Stage I} \quad \text{enumerate each } \mathbf{k} \in \mathbf{K}, \text{ initialize } \Xi^{(0)} \text{ and iterate :} \\
&\quad (a) \quad \Theta^{(t)} = \arg\max/\mathrm{incr}_{\Theta} H(p \,||\, q, \Theta, \mathbf{k}, \Xi^{(t-1)}), \\
&\quad (b) \quad \Xi^{(t)} = \arg\max/\mathrm{incr}_{\Xi}[\mathrm{H}(p \,||\, q, \Theta^{(t)}, \mathbf{k}, \Xi) + d_{\mathbf{k}}(\Xi)], \quad \Delta\Theta_{\tau}^{(t)} = \Theta^{(t)} - \Theta^{(t-\tau)}, \quad \tau \geq 1, \\
&\qquad d_{\mathbf{k}}(\Xi) = -n_{f}(\Theta_{\mathbf{k}}) + \Delta\Theta_{\tau}^{(t)T} \Omega(\Theta^{(t)}, \Xi) \Delta\Theta_{\tau}^{(t)}, \quad \text{after convergence we get} \quad \Theta^{*}, \Xi^{*}; \\
&\text{Stage II} \quad \mathbf{k}^{*} = \mathrm{argmin}_{\mathbf{k}} J(\mathbf{k}), \quad J(\mathbf{k}) = \tilde{J}(\mathbf{k}) + 0.5 n_{f}(\Theta_{\mathbf{k}}).
\end{aligned} \tag{11a}$$

during which a previous estimate $\Theta^{(t-\tau)}$ is used as ζ with $\Gamma = -\Omega^{-1}(\Theta^*, \Xi)$, and thus $d_{\mathbf{k}}(\Xi)$ is simplified into $\Delta\Theta_{\tau}^{(t)T}\Omega(\Theta^{(t)}, \Xi)\Delta\Theta_{\tau}^{(t)}$, where an integer $n_f(\Theta_{\mathbf{k}})$ denotes the number of free parameters in $\Theta_{\mathbf{k}}$. Being different from a classic two stage implementation, not only model selection is made at Stage II, but also automatic model selection occurs during implementing Stage I that is actually implemented by one Ying Yang iteration as above, from which we can further get detailed algorithms, e.g., Algorithm 2 in the previous section and Algorithm 3 in the next section, as further discussed in sequel.

When samples $\mathbf{X}_N = \{x_t\}_{t=1}^N$ are independent and identically distributed (i.i.d.), from $\mathbf{Y}_N = \{y_t, j_t\}_{t=1}^N$ and $q(\Theta \mid \Xi) = q(h)\prod_{j=1}^k q(\Theta_j \mid \Xi)$, $H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ in eqn.(11a) becomes $\sum_{t=1}^N \sum_{j=1}^k p(j \mid x, \theta^{y|x})\{\int p(y \mid x,\ \theta_j^{y|x})G(x \mid x_t, h_x^2 I)\ln[q(x \mid y, \theta_{x|y}^j)q(y \mid \theta_j^y)\alpha_j]dxdy\}$ with $\alpha_j = q(j)$. By a Taylor expansion of $\ln[q(x \mid y, \theta_{x|y}^j)q(y \mid \theta_j^y)\alpha_j]$ with respect to *x*, *y* around x_t and $\eta(x_t \mid \theta_j^{y|x})$, respectively, from which we further get

$$H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi) \approx \sum_{t=1}^N \sum_{j=1}^k p(j \mid x_t, \theta^{y|x}) H_t(\Theta_j), \tag{11b}$$

$$H_t(\Theta_j) = \pi(x_{t,}\ \eta(x_t \mid \theta_j^{y|x}), \Theta_j) - 0.5Tr[h^2\Pi_j^x + \Gamma_j^{y|x}\Pi_j^y] + \tfrac{1}{N}\ln q(h),$$

$$\Pi_j^x = -\nabla_x^2\ \ln q(x \mid y, \theta_j^{x|y}),\quad \Pi_j^y = -\nabla_y^2\ \ \pi_t(y, \Theta_j)\quad \pi_t(y, \Theta_j) = \ln[q(x_t \mid y, \theta_j^{x|y})q(y \mid \theta_j^y)\alpha_j],$$

$$\eta(x_t \mid \theta_j^{y|x}) = \int y p(y \mid x_t, \theta_j^{y|x})dy,\quad \Gamma_j^{y|x} = \int p(y \mid x_t, \theta_j^{y|x})[y - \eta(x_t \mid \theta_j^{y|x})][y - \eta(x_t \mid \theta_j^{y|x})]^T dy.$$

In the special case $\mathbf{Y}_N = \{j_t\}_{t=1}^N$, i.e., $\{y, j\} \to j$ without *y*, $H_t(\Theta_j)$ becomes $\pi(x, \Theta_j) - 0.5Tr[h^2\nabla_x^2 \ln q(x \mid \theta_j)] + \frac{1}{N}\ln q(h)$, $\pi(x, \Theta_j) = \ln[q(x \mid \theta_j)\alpha_j]$. Further considering the substitution $x \to x$, z and thus $q(x \mid \theta_j) \to q(x \mid \phi_j)q(z \mid x, \theta_j)$ with $G(x \mid x_t, h^2 I) \to G(x \mid x_t, h_x^2 I)G(z \mid z_t(x_t), h_z^2 I)$ and $q(h) \to q(h_x)q(h_z)$, $H_t(\Theta_j)$ in eqn.(11b) is simplified into the one same as in Algorithm 2(a).

We consider the design of $p(j \mid x_t, \theta_x)$ according to a principle of uncertainty conversation between Ying-Yang, under the likelihood based measure given in Figure 4(c). From $p(j \mid x, \theta_x) \propto q(x \mid \phi_j)q(z \mid x, \theta_j)\alpha_j$ and the constraint $\sum_j p(j \mid x, \theta_x) = 1$, we have $p(j \mid x, \theta_x) = q(x \mid \phi_j)q(z \mid x, \theta_j)\alpha_j \,/\, \sum_j q(x \mid \phi_j)q(z \mid x, \theta_j)\alpha_j = p_t(j \mid \Theta)$, which concurs with the one in Algorithm 2(a) and Figure 3(a). In this setting, we further have

- $dH_0(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ as in Algorithm 2 (b), based on which we update $\Theta^{new} = \Theta^{old} + \Delta\Theta$, $\Delta\Theta \propto dH_0(p \,||\, q, \Theta, \mathbf{k}, \Xi) \,/\, d\Theta$ with $q(\Theta_j \mid \Xi)$ ignored by letting $\ln q(\Theta_j \mid \Xi) = 0$, which leads to Algorithm 2 and Figure 3(a).
- $\eta_{j,t} = p_{j,t}(1 + \delta h_{t,j})$ modifies the EM learning by $1 + \delta h_{t,j}$ that either enhances or weaken $p_{j,t}$ depending on whether or not its fitness is better than the average. Even this $\eta_{j,t} = p_{j,t}(1 + \delta h_{t,j})$ will become negative if its fitness falls far below the average, and thus leads to de-learning, which acts as an improvement of RPCL learning.
- Maximization of $H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ pushes $\sum_{t=1}^N \sum_{j=1}^k p(j \mid x_t, \theta^{y|x}) \ln \alpha_j \to N\sum_{j=1}^k \alpha_j \ln \alpha_j$ with $N\alpha_j = \sum_{j=1}^k p(j \mid x_t, \theta^{y|x})$. Further more, $N\sum_{j=1}^k \alpha_j \ln \alpha_j \le 0$ approaches its upper bound

with some $\alpha_j \to 0$ if the *j*-th basis function or expert is redundant, and its contribution $N\alpha_j \ln \alpha_j$ becomes 0. In other word, we get an example that concurs with the previously discussed feature, i.e., $\tilde{J}(k)$ remains unchanged on $[\hat{k}, \tilde{k}]$, as illustrated on the right of Figure 4(b).

The above scenario can be directly extended to the case $\mathbf{Y}_N = \{y_t, j_t\}_{t=1}^N$ with

$$q(x \mid \phi_j) = \int q(x \mid y, \theta_j^{x|y}) q(y \mid \theta_j^y) dy \tag{11c}$$

put into $H_t(\Theta_j)$ in Algorithm 2(a).

In fact, the resulted model conceptually has no big difference from NRBF, ENRBF, ME, etc. The difference lays in that the basis function bases on the marginal density $q(x \mid \phi_j)$.

More generally, we can put the substitution $x \to x,z$ and its induced substitutions $q(x \mid y, \theta_j^{x|y}) \to q(x, z \mid y, \theta_j^{x,z|y}) = q(z \mid x, y, \theta_j^{z|x,y}) q(x \mid y, \theta_j^{x|y})$, $G(x \mid x_t, h^2 I) \to G(x \mid x_t, h_x^2 I) G(z \mid z_t(x_t), h_z^2 I)$ and $q(h) \to q(h_x) q(h_z)$ directly into eqn.(11b), resulting in

$$H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi) \approx \sum_{t=1}^{N} \sum_{j=1}^{k} p(j \mid x_t, z_t, \theta_{y|x}) H_t(\Theta_j), \tag{11d}$$

$$H_t(\Theta_j) = \pi_t(\Theta_j) - \tfrac{1}{2} Tr[h_x^2(\Pi_j^x + \Pi_j^{z|x}) + h_z^2 \Pi_j^z + \Gamma_j^{y|x} \Pi_j^y] + \tfrac{1}{N} \ln\{q(h_x) q(h_z)\},$$
$$\Pi_j^{z|x} = -\nabla_x^2 \ln q(z \mid \xi,\ \theta_j^{z|\xi}), \quad \Pi_j^z = -\nabla_z^2 \ln q(z \mid \xi, \theta_j^{z|\xi}),\ \xi = \{x, y\},$$

$$\pi_t(\Theta_j) = \pi_t(\eta(x_t \mid \theta_j^{y|x}), \Theta_j), \quad \pi_t(y, \Theta_j) = \ln[q(z_t \mid \xi, \theta_j^{z|\xi}) q(x_t \mid \boldsymbol{y}, \theta_j^{x|y}) q(y \mid \theta_j^y) \alpha_j].$$

Π_j^x, $\eta(x_t \mid \theta_j^{y|x})$ and $\Gamma_j^{y|x}$ are same as in eqn.(11b), from which and eqn.(11c) we can derive Algorithm 3 to be introduced in the next section. Moreover, in additional to the feature of pushing an extra $\alpha_j \to 0$, the maximization of $H_0(p \,||\, q, \Theta, \mathbf{k}, \Xi)$ in eqn.(11d) pushes $\pi_t(\Theta_j)$ to increase, which then pushes $\ln q(y \mid \theta_j^y)$ to increase. If there is a redundant dimension $y^{(i)}$, the corresponding $q(y^{(i)} \mid \theta_j^y)$ is pushed towards $\delta(y^{(i)} - c^{(i)})$ with its variance $\lambda_j \to 0$, which contributes to $\tilde{J}(k)$ by $-\infty$. In other word, we also get an specific example that concurs with the previously discussed feature that $\tilde{J}(k)$ tends $-\infty$, as illustrated beyond $\tilde{k}$ on the right of Figure 4(b).

SUBSPACE BASED FUNCTIONS AND CASCADED EXTENSIONS

In many practices, there is only a finite size of training samples distributed in small dimensional subspaces, instead of scattering over all the dimensions of the observation space. These subspace structures can not be well captured by considering $q(x \mid \phi_j) = G(x \mid \mu_j, \Sigma_j)$ only. Moreover, there are too many free parameters in Σ_j, which usually leads to poor performances. Towards the problems, we consider $q(x \mid \phi_j)$ via subspace by independent factor analysis as shown in Figure 5, where observed samples are regarded as generated from a subspace with independent factors distributed along each coordinate of an inner *m* dimensional representation $y = [y^{(1)} \ldots, y^{(m)}]$. Typically, we consider $q(y^{(i)} \mid \theta_j^y)$ of a Gaussian

for a real $y^{(i)}$ and of a Bernoulli for a binary $y^{(i)}$.

By eqn. (10c) we get $q(x \mid \phi_j)$ to be put into eqn.(5) for extensions of basis functions or gating nets $g_j(x,\phi)$, which leads to subspace based extensions of ME and RBF networks. Correspondingly, $f(x)$ in eqn.(3) has been extended from a combination of functions supported on radial bases to a combination of functions supported on subspace bases, which are thus called subspace based functions (SBF). As discussed after eqn.(11c), learning of SBF can still be made by either Algorithm 1 or 2, with the updating formulae for $q(x \mid \phi_j)$ revised accordingly. Also, to facilitate computing we get $\alpha_j q(x \mid \phi_j) = \rho_t(\Theta_j)$ approximately by

$$\rho_t(\Theta_j) = \int e^{\pi_t(y,\Theta_j)} dy \approx (2\pi)^{0.5 m_j} \mid \Pi_j^y \mid^{0.5} \exp[\pi(\eta(x_t \mid \theta_j^{y|x}),\Theta_j)], \;\; \Pi_j^y \text{ as in eqn.(11b)} , \tag{12}$$

where `≈` becomes `=` when $q(x \mid y,\theta_j^{x|y})$ and $q(y \mid \theta_j^y)$ are both Gaussian.

As before, an appropriate number k can be determined by automatic model selection during implementing Algorithm 2. In addition to k, we need an appropriate dimension m_ℓ for each subspace too, which can not be performed by Algorithm II. Though the problem may be handled in help of a two stage implementation, the effect of a finite number of samples will more seriously deteriorate the evaluation on the values of $J(\Theta_k)$ because we need to determine $k + 1$ integers (i.e., k plus $m_\ell, \ell = 1,\ldots,k$).

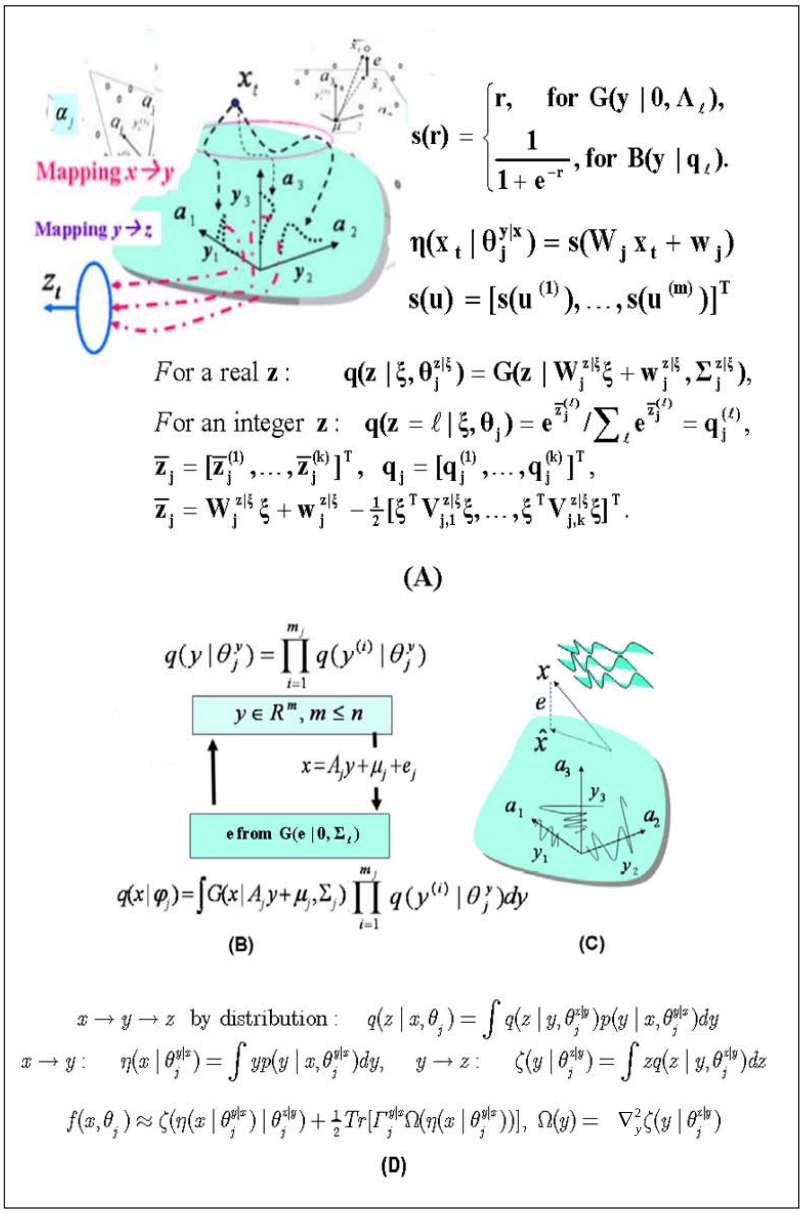
Alternatively, instead of implementing a direct mapping $x \rightarrow z$ by $q(z_t \mid x_t,\theta_j^{z|x})$ or $f_j(x,\theta_j^{z|x})$, extensions can also be made in help of the hidden factors y by $q(z_t \mid \xi,\theta_j^{z|\xi}) = q(z_t \mid y,\theta_j^{z|y})$, as shown in Figure 5. Two cascade mappings $x \rightarrow y$ and $y \rightarrow z$ replace the direct map $x \rightarrow z$ by $f_j(x,\theta_j^{z|x})$ or $q(z_t \mid x_t,\theta_j^{z|x})$. The mapping $x \rightarrow y$ acts as feature extraction, and thus the cascade $x \rightarrow y \rightarrow z$ can discard redundant parts, which is called cascaded subspace based functions. In a regression task, the cascaded $x \rightarrow y \rightarrow z$ is exactly a compound function of the regressions for $x \rightarrow y$ and $y \rightarrow z$ when the latter is linear. In other cases, an approximation is obtained from a Taylor expansion of $\zeta(y \mid \theta_j^{z|y})$ around $\eta(x \mid \theta_j^{y|x})$ up to the second order.

However, it is no longer applicable to use eqn.(11c) to get $q(x \mid \phi_j)$ in eqn.(5) for $g_j(x,\phi)$. Instead, we consider $q(x \mid \phi_j)q(z \mid x,\theta_j) = \int q(x \mid y,\theta_j^{x|y})q(y \mid \theta_j^y)q(z \mid \xi,\theta_j^{z|\xi})dy$ and get $p(j \mid x,z_t)$ from eqn. (6b). To facilitate computing, we can also get $\alpha_j q(x \mid \phi_j)q(z \mid x,\theta_j) = \rho_t(\Theta_j)$ approximately by eqn. (12) with $\pi_t(y,\Theta_j)$ extended accordingly. That is, we have

$$p(j \mid x,z) = \rho_t(\Theta_j) \,/\, \textstyle\sum_{j=1}^k \rho_t(\Theta_j), \quad \text{with } \rho_t(\Theta_j) \text{ by eqn.(12) and } \pi_t(y,\Theta_j) \text{ by eqn.(11d).} \tag{13}$$

Learning is made to maximize $H_0(p \,||\, q,\Theta,\mathbf{k},\Xi)$ in eqn.(11d), from which we obtain an adaptive algorithm as summarized in Algorithm 3, for implementing learning with an appropriate number k and an appropriate dimension m_ℓ of each subspace determined automatically. According to three typical cases of $q(z \mid \xi,\theta_j^{z|\xi})$, Algorithm 3 performs one of the tasks shown in Figure 6.

Figure 5. Subspace based independent factor analyses and extensions to cascaded subspace based functions



- Let $q(z \mid \xi, \theta_j^{z|\xi}) = 1$ (i.e., ignored), it degenerates into the adaptive algorithms for learning on either a local factor analysis (LFA) given in Table 3 in (Xu, 2008) or a local binary factor analysis (BFA).
- When $q(z \mid \xi, \theta_j^{z|\xi}) = G(z \mid W_j^{z|\xi}\xi + w_j^{z|\xi}, \Sigma_j^{z|\xi})$, it performs a regression that consists of a number of local functions. For the SBF case with $q(z \mid x, \theta_j^{z|x}) = G(z \mid W_j^{z|x}x + w_j^{z|x}, \Sigma_j^{z|x})$, each local linear function is $z = W_j^{z|x}(x - \mu_j) + w_j^{z|x}$, and the mapping $x \to z$ is $\sum_{j=1}^k p(j \mid x_t, \Theta)[W_j^{z|x}(x_t - \mu_j) + w_j^{z|x}]$. In this case, Algorithm 3 shares with Algorithm 2 the

Figure 6. Typical terms in Algorithm 3

	Gaussian	**Discrete** $z = 1, \cdots, k$
	$\mathbf{q}(\mathbf{z} \mid \boldsymbol{\xi}, \boldsymbol{\theta}_j^{z\mid\xi}) = \mathbf{G}(\mathbf{z} \mid \mathbf{W}_j^{z\mid\xi}\boldsymbol{\xi} + \mathbf{w}_j^{z\mid\xi}, \boldsymbol{\Sigma}_j^{z\mid\xi})$	$\mathbf{q}(\mathbf{z} = \ell \mid \boldsymbol{\xi}, \boldsymbol{\theta}_j) = \mathbf{e}^{\overline{\mathbf{z}}_j^{(\ell)}} / \sum_\ell \mathbf{e}^{\overline{\mathbf{z}}_j^{(\ell)}} = \mathbf{q}_j^{(\ell)}$, $\overline{\mathbf{z}}_j = \mathbf{W}_j^{z\mid\xi}\boldsymbol{\xi} + \mathbf{w}_j^{z\mid\xi} - \frac{1}{2}[\boldsymbol{\xi}^{\mathrm{T}}\mathbf{V}_{j,1}^{z\mid\xi}\boldsymbol{\xi}, \ldots, \boldsymbol{\xi}^{\mathrm{T}}\mathbf{V}_{j,k}^{z\mid\xi}\boldsymbol{\xi}]^{\mathrm{T}}$
$\mathbf{g}_j^{z\mid\xi} = \nabla_\xi \mathbf{lnq}(\mathbf{z} \mid \boldsymbol{\xi}, \boldsymbol{\theta}_j^{z\mid\xi})$	$\mathbf{W}_j^{z\mid\xi\,\mathrm{T}}\boldsymbol{\Sigma}_j^{z\mid\xi\,-1}\mathbf{e}_t^{z\mid\xi}$, $\mathbf{e}_t^{z\mid\xi} = \mathbf{z}_t - \mathbf{W}_j^{z\mid y}\boldsymbol{\xi}$	$\mathbf{W}_j^{z\mid\xi\,\mathrm{T}}(\mathbf{e}_\ell - \mathbf{q}_{j,t}) - (\mathbf{V}_{j,\ell}^{z\mid\xi} - \sum_k \mathbf{q}_{j,t}^{(k)}\mathbf{V}_{j,k}^{z\mid\xi})\boldsymbol{\xi}$
$\boldsymbol{\Pi}_j^{z\mid\xi} = -\nabla_\xi^2 \mathbf{lnq}(\mathbf{z} \mid \boldsymbol{\xi}, \boldsymbol{\theta}_j^{z\mid\xi})$	$\mathbf{W}_j^{z\mid\xi\,\mathrm{T}}\boldsymbol{\Sigma}_j^{z\mid\xi\,-1}\mathbf{W}_j^{z\mid\xi}$	**approximately** $\mathbf{W}_j^{z\mid\xi\,\mathrm{T}}(\mathrm{diag}[\mathbf{q}_{j,t}] - \mathbf{q}_{j,t}\mathbf{q}_{j,t}^{\mathrm{T}})\mathbf{W}_j^{z\mid\xi} + \mathbf{V}_{j,\ell}^{z\mid\xi} - \sum_k \mathbf{q}_{j,t}^{(k)}\mathbf{V}_{j,k}^{z\mid\xi}$
$\boldsymbol{\Pi}_j^{z} = -\nabla_z^2 \mathbf{lnq}(\mathbf{z} \mid \boldsymbol{\xi}, \boldsymbol{\theta}_j^{z\mid\xi})$	$\boldsymbol{\Sigma}_j^{z\mid\xi\,-1}$	$\mathbf{0}$
Data smoothing	h_z^2 **estimated**	**simply setting** $h_z^2 = 0$

(A)

$\mathbf{q}(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}_j^{z\mid y})$	**Gaussian** $\mathbf{G}(\mathbf{z} \mid \mathbf{W}_j^{z\mid y}\mathbf{y} + \mathbf{w}_j^{z\mid y}, \boldsymbol{\Sigma}_j^{z\mid y})$	**Discrete** $\mathbf{q}(\mathbf{z} \mid \mathbf{y}, \boldsymbol{\theta}_j) = \mathbf{q}(\mathbf{z} = \ell \mid \mathbf{y}, \boldsymbol{\theta}_j)$
$\mathbf{e}_t^{z\mid y}$	$\mathbf{z}_t - (\mathbf{W}_j^{z\mid y\ \mathrm{old}}\mathbf{y}_{j,t}^{*} + \mathbf{w}_j^{z\mid y\ \mathrm{old}})$	$\mathbf{e}_\ell - \mathbf{q}_{j,t}$
$\boldsymbol{\Psi}_q$	I	$\mathrm{diag}[\mathbf{q}_{j,t}] - \mathbf{q}_{j,t}\mathbf{q}_{j,t}^{T}$
$\boldsymbol{\Sigma}_j^{z\mid y}$ $\mathbf{V}_{j,i}^{z\mid y}$ $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{k}$	$\boldsymbol{\Sigma}_j^{z\mid y} = \mathbf{S}_j^{z\mid y}\mathbf{S}_j^{z\mid y\,\mathrm{T}}$, $\mathbf{S}_j^{z\mid y\ \mathrm{new}} = \mathbf{S}_j^{z\mid y\ \mathrm{old}} + \boldsymbol{\Delta}\mathbf{S}_j^{z\mid y}$, $\boldsymbol{\Delta}\mathbf{S}_j^{z\mid y} \propto \boldsymbol{\eta}_{j,t}\mathbf{G}_j^{z\mid y}\mathbf{S}_j^{z\mid y\ \mathrm{old}\,-\mathrm{T}}$, $\mathbf{G}_j^{z\mid y} = \mathbf{e}_{j,t}^{z\mid y}\mathbf{e}_{j,t}^{z\mid y\mathrm{T}} - \boldsymbol{\Sigma}_j^{z\mid y\ \mathrm{old}} + \mathbf{W}_j^{z\mid y\ \mathrm{old}}\boldsymbol{\Gamma}_{j,t}^{y\mid x}\mathbf{W}_j^{z\mid y\ \mathrm{old}\,\mathrm{T}} + \mathbf{0.5h}_z^2\mathbf{I}$	$\mathbf{V}_{j,i}^{z\mid y} = \mathbf{Q}_{j,i}^{z\mid y}\mathbf{Q}_{j,i}^{z\mid y\ \mathrm{T}}$, $\mathbf{Q}_{j,i}^{z\mid y\ \mathrm{new}} = \mathbf{Q}_{j,i}^{z\mid y\ \mathrm{old}} + \boldsymbol{\Delta}\mathbf{Q}_{j,i}^{z\mid y}$, $\boldsymbol{\Delta}\mathbf{Q}_{j,i}^{z\mid y} \propto -\boldsymbol{\eta}_{j,t}(\boldsymbol{\delta}_{i\ell} - \mathbf{q}_{j,t}^{(k)})[\boldsymbol{\Gamma}_{j,t}^{y} + (\boldsymbol{\delta}_{i\ell} - \mathbf{q}_{j,t}^{(k)})\boldsymbol{\Gamma}_{j,t}^{y\mid x}]\mathbf{Q}_{j,i}^{z\mid y\ \mathrm{old}\,-\mathrm{T}}$
$\mathbf{w}_j^{z\mid y\ \mathrm{new}} = \mathbf{w}_j^{z\mid y\ \mathrm{old}} + \boldsymbol{\Delta}\mathbf{w}_j^{z\mid y}$, $\boldsymbol{\Delta}\mathbf{w}_j^{z\mid y} \propto \boldsymbol{\eta}_{j,t}\mathbf{e}_t^{z\mid y}$, $\mathbf{W}_j^{z\mid y\ \mathrm{new}} = \mathbf{W}_j^{z\mid y\ \mathrm{old}} + \boldsymbol{\Delta}\mathbf{W}_j^{z\mid y}$, $\boldsymbol{\Delta}\mathbf{W}_j^{z\mid y} \propto \boldsymbol{\eta}_{j,t}(\mathbf{e}_t^{z\mid y}\mathbf{y}_{j,t}^{*\,\mathrm{T}} - \boldsymbol{\Psi}_q^{\mathrm{old}}\mathbf{W}_j^{z\mid y}\boldsymbol{\Gamma}_{j,t}^{y\mid x})$		

(B)

same equations for updating $q(z_t \mid x_t, \theta_j^{z\mid x})$ by Figure 3(C)&(D). For a cascaded SBF case with $q(z \mid y, \theta_j^{z\mid y}) = G(z \mid W_j^{z\mid y}y + w_j^{z\mid y}, \Gamma_j^{z\mid y})$, a local mapping is made by $\eta(x \mid \theta_j^{y\mid x}) = s(W_j\, x\ + w_j)$ that performs a local dimension reduction and a regression $z = W_j^{z\mid y}y + w_j^{z\mid y}$, with $x \to z$ by $\sum_{j=1}^{k} p(j \mid x_t, \Theta)[W_j^{z\mid y}s(W_j\, x\ + w_j) + w_j^{z\mid y}]$. Being different from Algorithm 2 that makes automatic selection only on *k*, learning by Algorithm 3 on the part for $y \to\ \ z$ is coupled with the part for $x \to y$, during which automatic model selection is obtained on the dimension of each subspace, as previously explained after eqn.(11d).

- When $q(z \mid y, \theta_j) = q(z = \ell \mid y, \theta_j)$, $\ell = 1, \ldots, \mathrm{k}$, it performs a classification $x \to z$ into one of labels. In help of those terms in Figure 6, the task is made by a group of classifiers with a local

feature extraction by $\eta(x \mid \theta_j^{y|x}) = s(W_j\, x \ + w_j)$.

Simplified variants of Algorithm 3 may be considered with degenerated settings on one or more of $h_x = 0$, $h_z = 0$, $\Gamma_j^{y|x} = 0$ that actually remove certain regularization on learning. Specifically, h_x = 0 shuts down smoothing regularization on samples of *x*, and h_z = 0 shuts down the one on samples of *z*, while $\Gamma_j^{y|x} = 0$ shuts down a regularization on the domain of *y*. Also, Algorithm 3 may degenerate into EM or RPCL with a degenerated setting h_x = 0 by choosing its corresponding allocating scheme in Figure 3(A).

The cases that the variable $y^{(i)}$ on each coordinate of a subspace is binary by a Bernoulli are usually encountered in practical problems that encodes each input *x* into binary codes for subsequent processing (Heinen, 1996; Treier & Jackman, 2002). The cascaded mapping $x \to y \to z$ implements $f_j(x, \theta_j^{z|x}) = W_j^{z|y} s(Wx + w_j) + w_j^{z|y}$ that is a three layer forward net. In other words, the special case *k* = 1 leads us to a classic three layer forward network. Being different from both the least square back propagation learning (Rumelhart, Hinton, & Williams, 1986) and the EM algorithm based ML learning (Ma, Ji, & Farmer, 1997), there is favorably an automatic model selection on the number m_ℓ of hidden units during implementing Algorithm 3 (shown in Figure 7). Moreover, a general case *k* > 1 leads to $f(x, \theta^{z|x}) = \sum_{j=1}^{k} p(j \mid x_t, \Theta) f_j(x, \theta_j^{z|x})$, i.e., a mixture of experts with each expert in a three layer forward networks. Sharing a hidden representation *y* in Figure 5(A) by experts and its gate, the number of free parameters is reduced. This mixture of experts is different from the conventional way of using one three layer forward networks as each expert in either (Jacobs, Jordan, Nowlan, & Hinton, 1991) or (Xu, Jordan & Hinton, 1995), where $q(z \mid \xi, \theta_j^{z|\xi}) = q(z \mid x, \theta_j^{z|x})$ is implemented via a three layer forward networks as a whole.

In the cases with a binary vector *y*, the integral $\int dy$ becomes a summation $\sum_y$ over all the 2^{m_ℓ} terms, which becomes impractical computationally. Conceptually, the technique of using Taylor expansion in eqn.(11b) and eqn.(11d) is not applicable to *y* of a binary vector. Still, we can get eqn. (11b) and eqn.(11d) by rewriting $\pi_t(y, \Theta_j)$ in a format $b_0 + b_1[y - Ey] + b_2[y - Ey][y - Ey]^T$, and we replace $\int dy$ by $\sum_y$ to get $\Gamma_j^{y|x}$ in eqn.(11b) and $\rho_t(\Theta_j)$ in eqn.(13) . Moreover, the task of getting $y_{j,t}^* = \arg\max_y \pi_t(y, \Theta_j)$ is also conceptually a NP hard 0-1 programming. In Algorithm 3, it is approximately handled as suggested in (Table II, Xu, 2001b) under the name of fixed posteriori approximation, i.e., solving $\nabla_y \pi_t(y, \Theta_j) = 0$ as if each $y^{(i)}$ is real and then is rounded into binary. With an extra computing for solving a nonlinear equation of a scalar variable, a further improvement can be obtained by adopting the canonical dual approach (Fang, Gao, Shu, & Wu, 2008; Sun, Tu, Gao, & Xu, 2009).

In addition to getting $\Gamma_j^{y|x}$ in eqn.(11b) by $\Gamma_j^{y|x} = \varepsilon_{j,t} \varepsilon_{j,t}^T$ in Algorithm 3, we also have a degenerated choice and a generalized choice as follows:

- a degenerated choice is considering $\Gamma_j^{y|x} = diag[\gamma_{j,t}^{(1)}, \ldots, \gamma_{j,t}^{(m_j)}]$, $\gamma_{j,t}^{(i)} = s(\overline{y}_t^{(i)})(1 - s(\overline{y}_t^{(i)}))$, which is equivalently considering a conditional independent distribution $q(y \mid x, \theta_j^{y|x}) = \prod_{i=1}^{m_j} s(\overline{y}^{(i)})^{y^{(i)}} \left(1 - s(\overline{y}^{(i)})\right)^{1-y^{(i)}}$ $\overline{y}_t = W_j^{y|x} x_t + w_j$
- a generalized choice is estimating:

Figure 7. Algorithm 3: Adaptive algorithms for LFA, cascade SBF and their binary variants

With $q(x_t \mid y,\theta_{x|y,j}) = G(x_t \mid A_j y + \mu_j, \Sigma_j)$ **and** $q(z_t \mid \xi,\theta_j^{z|\xi})$ **in one of choices in Fig.6(A), eq.(10d) becomes**

$$H_0(p \,\|\, q,\Theta,k,\Xi) \approx \sum_{t=1}^{N}\sum_{j=1}^{k} p(j \mid x_t,\theta^{y|x}) H_t(\Theta_j),$$

$$H_t(\Theta_j) = \pi(x_t, \eta(x_t \mid \theta_j^{y|x}), \Theta_j) - \tfrac{1}{2}\mathrm{Tr}[\Gamma_j^{y|x}\Pi_j^{y} + h_x^2(\Sigma_j^{-1} + \Pi_j^{z|x}) + h_z^2\Pi_j^{z}]$$
$$+ \tfrac{1}{N}\ln[q(h_x)q(h_z)],\ \pi_t(y,\Theta_j) = \ln[\alpha_j G(x_t \mid A_j y + \mu_j, \Sigma_j) q(y \mid \theta_j^{y}) q(z_t \mid \xi, \theta_j^{z|\xi})]_{\xi = y \text{ or } x}$$

$$\Pi_j^{y} = \nabla_y^2 \pi_t(y,\Theta_j) = A_j^{T}\Sigma_j^{-1}A_j + \Pi_j^{z|y} + \Delta_{\Pi}^{y,\ell},$$ **where we have**

- $q(y \mid \theta_\ell^{y}) = \begin{cases} G(y \mid \mu_\ell^{y}, \Lambda_\ell), & \Lambda_\ell \text{ is diagonal, Gaussian,} \\ B(y \mid q_\ell) = \prod_{i=1}^{m_\ell} (1-q_\ell^{(i)})^{1-y^{(i)}} q_\ell^{(i)\, y^{(i)}}, & \text{Bernoulli.} \end{cases}$ $\quad \Delta_{\Pi}^{y,\ell} = \begin{cases} \Lambda_\ell^{-1}, & \text{for } G(y \mid \mu_\ell^{y}, \Lambda_\ell), \\ 0, & \text{for } B(y \mid q_\ell). \end{cases}$
- $\Gamma_j^{y|x} = \varepsilon_{j,t}\varepsilon_{j,t}^{T}$, $\varepsilon_{j,t} = \eta(x_t \mid \theta_j^{y|x}) - y_{j,t}^{*}$, $y_{j,t}^{*} = \arg\max_y \pi_t(y,\Theta_j)$, **from the rationale that the possible upmost bound of** $H_0(p \,\|\, q,\Theta,k,\Xi)$ **is got at** $p(y \mid x_t, z_t, j) = \delta(y - y_{j,t}^{*})$.
- $p(j \mid x_t,\theta^{y|x})$, $\eta(x_t \mid \theta_j^{y|x})$, $\Pi_j^{z|}$, Π_j **are given in Fig.6(A).**

Maximization of $H_0(p \,\|\, q,\Theta,k,\Xi)$ **is implemented via its gradient flow**

$$\nabla_\Theta H_0(p \,\|\, q,\Theta,k,\Xi) = \nabla_{[h_x,h_z]} \ln[q(h_x)q(h_z)] + \sum_{t=1}^{N}\sum_{j=1}^{k} p_{j,t}(1+\delta h_{j,t})\nabla_{\Theta_j}\pi(x_t, \eta(x_t \mid \theta_j^{y|x}), \Theta_j) -$$
$$\tfrac{1}{2}\sum_{t=1}^{N}\sum_{j=1}^{k} p_{j,t}\{\nabla_{\Theta_j}\mathrm{Tr}[\Gamma_j^{y|x}\Pi_j^{y} + h_x^2(\Sigma_j^{-1} + \Pi_j^{z|x}) + h_z^2\Pi_j^{z}] + \delta h_{j,t}\nabla_{\Theta_j}\ln|\Pi_j^{y}|\}$$

by iterating the following two steps until its convergence:

YANG STEP: get $p_{j,t} = p(j \mid z_t, x_t)$ **and** $\delta h_{j,t} = H_t(\Theta_j) - \sum_{\ell=1}^{k} p_{\ell,t} H_t(\Theta_\ell)$,

- **Solve** $\nabla_y \pi_t(y,\Theta_j) = 0$, **i.e.** $A_j^{T}\Sigma_j^{-1}(x_t - A_j y + \mu_j) + \Lambda_j^{-1}(y - \mu_j^{y}) + g_j^{z|y} = 0$ **with** $g_j^{z|y}$ **in Fig.6(A), which is linear of** y **(or approximately for a discrete** z**).**
- **update** $W_j^{new} = W_j^{old} + \Delta W_j$, $w_j^{new} = w_j^{old} + \Delta w_j$, $\Delta W_j \propto p_{j,t}\varepsilon_{j,t}x_t^{T}$, $\Delta w_j \propto \varepsilon_{j,t}$.

YING STEP get $\alpha_j^{new} = e^{c_j^{old} + \Delta c_j} / \sum_\ell e^{c_\ell^{old} + \Delta c_\ell}$, **same as Step (a) in Algorithm II,**

If a $\alpha_j \to 0$, **discard the corresponding structure and its** Θ_j.

(*a*) $\mu_j^{new} = \mu_j^{old} + \Delta\mu_j$, $\Delta\mu_j \propto p_{j,t}(1+\delta h_{j,t})e_t^{x|y}$, $e_t^{x|y} = x_t - (A_j^{old}y_{j,t} + \mu_j^{old})$,

$A_j^{new} = A_j^{old} + p_{j,t}\Delta A_j (I - A_j^{old}A_j^{old\,T})$, $\Delta A_j \propto (1+\delta h_{j,t})e_t^{x|y}y_{j,t}^{T} - A_j^{old}(\Gamma_{j,t}^{y|x} + \delta h_{j,t}\Pi_j^{y\,-1})$,

$\Sigma_j^{new} = S_j S_j^{T}$, $S_j^{new} = S_j^{old}(I + p_{j,t}S_j^{-old}\Sigma_j^{-old}\Delta S_j \Sigma_j^{-old}S_j^{old\,-T})$,

$\Delta S_j \propto (1+\delta h_{j,t})(e_{j,t}^{x|y}e_{j,t}^{x|y\,T} - \Sigma_j^{old}) + A_j^{old}(\Gamma_{j,t}^{y|x} + \delta h_{j,t}\Pi_j^{y\,-1})A_j^{old\,T} + h_x^2 I$.

(*b*) **Get** $y_{j,t} = \eta(x_t \mid \theta_j^{y|x})$, **update** $G(y \mid \mu_j^{y}, \Lambda_j)$ **by** $\Lambda_j = D_j D_j^{T}$, $D_j^{new} = (I + \Delta D_j)D_j^{old}$,

$\Delta D_j \propto p_{j,t}\mathrm{diag}[(1+\delta h_{j,t})(e_{j,t}^{y}e_{j,t}^{y\,T} - \Lambda_j^{old}) + \Gamma_{j,t}^{y|x}]$, $e_{j,t}^{y} = y_{j,t} - \mu_j^{y}$ with $\mu_j^{y} = 0$;

Update $B(y \mid q_j)$ **by** $q_\ell^{(i)} = 1/(1+e^{-\beta_\ell^{(i)new}})$, $\beta_\ell^{new} - \beta_\ell^{old} \propto p_{\ell,t}(1+\delta h_{t,\ell})(y_{j,t} - q_\ell)$.

If $\lambda_\ell^{(i)} \to 0$ *or* $q_\ell^{(i)}(1-q_\ell^{(i)}) \to 0$, **discard the corresponding dimension** $y_\ell^{(i)}$.

(*c*) $\eta_{j,t} = 1 + \delta h_{j,t}$, **update** $q(z \mid y, \theta_j^{z|y})$ **by Fig.6(B) or** $q(z \mid x, \theta_j^{z|x})$ **by Fig.3(C) & (D).**

(*d*) $h_z^{new} = h_z^{old} + \Delta h_z$, $h_x^{new} = h_x^{old} + \Delta h_x$ **same as Step (d) in Algorithm II.**

$$\Gamma_j^{y|x} = \frac{1}{\#N(y_{j,t}^*)} \sum_{y \in N(y_{j,t}^*)} [y - \eta(x_t \mid \theta_j^{y|x})][y - \eta(x_t \mid \theta_j^{y|x})]^T, \quad y_{j,t}^* = \arg\max_y \pi_t(y, \Theta_j), \tag{14}$$

where $N(y_{j,t}^*)$ consists of $\mathbf{y}_{\mathbf{j,t}}^*$ and those values with a κ bit distance away, e.g., $\kappa = 1$.

FUTURE TRENDS

A further direction is extending SBF models in order to model temporal relations among data, via providing $q(y^{(i)}|\omega^{(i)})$ with a temporal structure such that temporal structure underlying samples of x can be projected to the temporal structures of y, as shown Figure 5(C). There have been three types of efforts. The straightforward one is to let each $f_j(x, \theta_j)$ being a regression of past samples $f_j(\{x_{t-\tau}\}_{\tau=1}^{\kappa}, \theta_j^{z|x})$, which has no difference from $f_j(x, \theta_j)$ by regarding $\{x_{t-\tau}\}_{\tau=1}^{\kappa}$ as a vector. The second is embedding temporal structure

$$\alpha_t = Q\alpha_{t-1}, \quad \alpha_t = [\alpha_{t,1}, \ldots, \alpha_{t,k}]^T, \ \mathrm{Q} = [q_{ji}], \quad 0 \le q_{ji} \le 1, \ \sum_{\mathrm{i}=1}^{k} q_{ji} = 1, \tag{15}$$

into each priori $0 \le \alpha_j, \ \sum_{\mathrm{j}} \alpha_j = 1$. From *t-1* to *t*, a new α_t is computed from its past α_{t-1}, and modulates the gate $g_j(x, \phi)$ in eqn.(5) to vary as time goes. For learning Q, we modify Step (a) of Algorithm 2&3, in either of the two ways as follows.

Recursive Updating

$$\begin{gathered}\text{Update} \quad Q \ \text{ by } \ q_{ji} = e^{c_{ji}} \Big/ \sum_{\ell} e^{c_{j\ell}} \ \text{ via } \ C = [c_{ji}] \ \text{ by } \ C^{new} = C^{old} + \Delta C, \quad \Delta C \propto \alpha_{t-1} g_\alpha^T - Q^T diag[g_\alpha], \\ \text{which comes from } \ Tr[g_\alpha^T d\alpha_t] \ \text{ via } \ d\alpha_t = dQ\alpha_{t-1} \ \text{ and } \ \mathrm{dQ^T} = dC - \mathbf{1}[\mathbf{q}_1^T d\mathbf{c}_1, \ldots, \mathbf{q}_k^T d\mathbf{c}_k], \\ \mathrm{Q^T} = [\mathbf{q}_1, \ldots, \mathbf{q}_k], \ \ \mathbf{q}_j = [q_{j1}, \ldots, q_{jk}]^T, \ \text{ and from} \\ Tr[g_\alpha^T d\alpha_t] = Tr[g_\alpha^T dQ\alpha_{t-1}] = Tr[g_\alpha \alpha_{t-1}^T dQ^T] = Tr[g_\alpha \alpha_{t-1}^T dC - diag[g_\alpha] Q dC].\end{gathered} \tag{16a}$$

Asymptotic Updating

As $t \to \infty$, $\alpha_t \to \alpha$ we have $\alpha = Q\alpha$, thus $(\mathrm{I\text{-}Q})d\alpha = (dQ)\alpha$,

$$\begin{gathered}\text{or } d\alpha = \ (\mathrm{I\text{-}Q})^{-1}(dQ)\alpha, \ \text{ from } \ Tr[g_\alpha^T (I-Q)^{-1} dQ\alpha] \ = Tr[g_\alpha^Q \alpha^T dC - diag[g_\alpha^Q] Q dC], \\ \text{we update } \ C^{new} = C^{old} + \Delta C, \quad \Delta C \propto \alpha^{\mathrm{old}} g_\alpha^{QT} - Q^{old\ T} diag[g_\alpha^Q], \ \ g_\alpha^Q = (I - Q^{old})^{-T} g_\alpha.\end{gathered} \tag{16b}$$

Putting eqn.(15) into eqn.(5), we get an extension of alternative ME to a temporal one gated by a Hidden Markov chain. Initially, let $\alpha_0 = [\alpha_{0,1}, \ldots, \alpha_{0,k}]^T$ we can get α_t from either $\alpha_t = Q^t \alpha_0$ or via $Q^t \alpha_{t-1}$ step by step, which makes the gate varies as time.

Figure 8. Modified updating equations for temporal subspaces

$$q(x_t \mid y_t, \varphi_j) = G(x_t | A_j y + \mu_j, \Sigma_j), \quad A_j^{\mathrm{T}} A_j = I$$

$q(y|\theta_j^y) = \prod_{i=1}^{m_j} q_j^{(i)y^{(i)}}(1-q_j^{(i)})^{1-y^{(i)}}$

At step (b) of Algorithm III, $q_j^{(i)}$ is replaced via modifying its learning accordingly.

$$\begin{bmatrix} 1-q_j^{(i)} \\ q_j^{(i)} \end{bmatrix}_t = \pi \begin{bmatrix} 1-q_j^{(i)} \\ q_j^{(i)} \end{bmatrix}_{t-1}, \quad 0 \le \pi_{j0}^{(i)}, \pi_{j1}^{(i)} \le 1, \quad \pi = \begin{bmatrix} \pi_{j0}^{(i)}, & 1-\pi_{j0}^{(i)} \\ 1-\pi_{j1}^{(i)}, & \pi_{j1}^{(i)} \end{bmatrix}$$

updating π in a same way as $\boldsymbol{\alpha}_t = Q\boldsymbol{\alpha}_{t-1}$ in eqn.(15)

$$q_j^{(i)} = s(\sum\nolimits_\tau b_{j\tau}^{(i)} y_{t-\tau}^{(i)}) = s(\mathbf{b}_j^{(i)T} \mathbf{y}_{t:\tau}^{(i)}),$$

$$s(r) = 1/(1+e^{-r}), \ \mathbf{b}_j^{(i)} = [b_{j1}^{(i)}, \ldots, b_{j\tau}^{(i)}]^{\mathrm{T}}, \mathbf{y}_{t:\tau}^{(i)} = [y_{t-1}^{(i)}, \ldots, y_{t-\tau}^{(i)}]^{\mathrm{T}}$$

$$\mathbf{b}_j^{(i)new} = \mathbf{b}_j^{(i)old} + \Delta \mathbf{b}_j^{(i)}, \Delta \mathbf{b}_j^{(i)} \propto \mathbf{g}_{q_j}^{(i)} s'(\mathbf{b}_j^{(i)T} \mathbf{y}_{t:\tau}^{(i)}) \mathbf{y}_{t:\tau}^{(i)}.$$

$$\because \mathbf{g}_{q_j}^{(i)} dq_j^{(i)} = \mathbf{g}_{q_j}^{(i)} s'(\mathbf{b}_j^{(i)T} \mathbf{y}_{t:\tau}^{(i)}) \mathbf{y}_{t:\tau}^{(i)\mathrm{T}} d\mathbf{b}_j^{(i)}$$

$q(y|\theta_j^y) = G(y \mid \mu_j^y, \Lambda_j)$ at step (b) of Algorithm III is replaced by $G(y_t \mid B_j y_{t-1}, \Lambda_j^y)$

$y_t = B_j y_{t-1} + e_t^y$, $\Lambda_j^y = E e_t^y e_t^{yT} \neq \mathrm{I}$ but diagonal still, $E e_t^y = 0$, $E e_t^y y_{t-1}^T = 0$, $B_j = \Lambda_j^{y\,0.5} W_j \Lambda_j^{y\,-0.5}$, $W_j = \phi_j D_j \phi_j^T$, $\phi_j^T \phi_j = I$.

$D_j = diag[\frac{e^{d_j^{(i)}} - e^{-d_j^{(i)}}}{e^{d_j^{(i)}} + e^{-d_j^{(i)}}}]_{i=1}^{m_j}$ for a real $-\infty < d_j^{(i)} < +\infty$.

Regression parameterization $\mu_{j,t}^y = B_j \mu_{j,t-1}^y$, $\mu_{j,t}^y = E y_t$

For learning, we modify Step (b) of Algorithm III with

$$\Lambda_j = \Lambda_j^y, \quad \mu_j^y = B_j \mu_{j,t-1}^y, \quad B_j^{new} = B_j^{old} + \Delta B_j, \ \Delta B_j \propto p_{j,t} e_{j,t}^y.$$

Marginalization equivalently $\Lambda_{j,t} = B_j \Lambda_{j,t-1} B_j^T + \Lambda_j^y$, $\Lambda_{j,t} = E y_t y_t^T$

Initially $\Lambda_{j,0} = \Lambda_j^y$, we have $\Lambda_{j,t} = \Lambda_j^y + \sum_{\tau=1}^{t} B_j^\tau \Lambda_j^y (B_j^\tau)^T =$

$$\Lambda_j^y + \sum_{\tau=1}^{t} \Lambda_j^{y\,0.5} W_j^\tau \Lambda_j^{y\,-0.5} \Lambda_j^y \Lambda_j^{y\,-0.5} W_j^\tau \Lambda_j^{y\,0.5} = \Lambda_j^y + \Lambda_j^{y\,0.5} [\sum_{\tau=1}^{t} W_j^{2\tau}] \Lambda_j^{y\,0.5} =$$

$$\Lambda_j^y + \Lambda_j^{y\,0.5} \phi_j [\sum_{\tau=1}^{t} D_j^{2\tau}] \phi_j^T \Lambda_j^{y\,0.5}, \text{ from which } \Lambda_{j,t} = \Lambda_j^{y\,0.5} \Sigma_j^D \Lambda_j^{y\,0.5} + \Lambda_j^y,$$

$$\Sigma_j^D = \phi_j D_j^\Sigma \phi_j^T, D_j^\Sigma = [I - (i_\infty - 1) D_j^{2(t+1)}](I - D_j^2)^{-1},$$

where $i_\infty = 1$ if interested on an asymptotic behavior, otherwise, $i_\infty = 0$.

Modifying Step (b) of Algorithm III with $\mu_j^y = 0$, by G_{Λ_j} we update

$$\phi_j^{new} = \phi_j^{old} + \Delta\phi_j, \ \Delta\phi_j \propto p_{j,t} G_{\phi_j} (I - \phi_j^{old} \phi_j^{old\,T}), \ G_{\phi_j} = \Delta D_j \phi_j D_j^\Sigma,$$

$$\Lambda_j^{y\,new} = \Lambda_j^{y\,\mathrm{old}} + \Delta\Lambda_j^y, \ \Delta\Lambda_j^y \propto p_{j,t} G_{\Lambda_j^y}, \ G_{\Lambda_j^y} = \Delta D_j (\Lambda_j^y + \Lambda_{j,t}),$$

$$D_j^{*new} - D_j^{*old} \propto p_{j,t} G_{D_j^*}, \ G_{D_j^*}^T = [D_j^\Sigma D - i_\infty (t+1) D_j^{2t-1}] \phi_j^T \Delta D_j \phi_j, \text{which}$$

comes from putting $d\Lambda_{j,t} = d(\Lambda_j^{y\,0.5} \Sigma_j^D \Lambda_j^{y\,0.5} + \Lambda_j^y)$ into $\mathrm{Tr}[G_{\Lambda_{j,t}}^T d\Lambda_{j,t}]$, we get $\mathrm{Tr}[G_{\Lambda_{j,t}}^T d\Lambda_{j,t}] = \mathrm{Tr}[G_{\Lambda_j^y}^T d\Lambda_j^y] + \mathrm{Tr}[G_{\phi_j}^T d\phi_j] + \mathrm{Tr}[G_{D_j^*}^T dD_j^*]$.

If one $\lambda_j^{y(i)} \to 0$ in $\Lambda_j^y = diag[\lambda_j^{y(1)}, \ldots, \lambda_j^{y(m_j)}]$, discard the dimension $y_j^{(i)}$ and its corresponding subset of parameters in Θ_j.

As shown in Figure 5(C), the third way is embedding a temporal structure $q(y_t \mid \omega_t)$ into the distribution of $y^{(i)}$ in each subspace, via a regression parameterization $\omega_t = f(\sum_\tau \vartheta_{i,\tau} y_{t-\tau})$ on past samples of $y_t^{(i)}$ or estimating $q(y_t^{(i)})$ as a marginal distribution $q(y_t) = \sum_{y_{t-1}} q(y_t \mid y_{t-1}) q(y_{t-1})$ or $q(y_t) = \int q(y_t \mid y_{t-1}) q(y_{t-1}) dy_{t-1}$ via the distributions of past samples. Shown in Figure 8 are detailed equations. For subspaces of Gaussian $y_t^{(i)}$, $y_t = B_j y_{t-1} + e_t^y$ is embedded into subspaces of local factor analysis, and thus has been studied under the name of local temporal factor analysis (TFA) (see Sec. IV, Xu, 2004).

Traced back to the early 1960's in the literature of nonparametric statistics, studies on RBF networks started from simple kernels $k(x, x_i)$ located at each sample and combined by linear weights that are simply samples of z. Then, studies proceeded along a direction not only with $k(x, x_i)$ extended to learning various structures in different subspaces and for different temporal dependences, but also with those combining weights estimated from samples or further extended to a learning gating structure.

Interestingly, a reversed direction of trends has also become popularized in recent years. One example is that SBF seeks to use simple linear subspaces instead of experts in a sophisticated structure. Another example is the widely studied support vector machine (SVM). It returns to considering locating each base or kernel $k(x, x_i)$ at each sample x_i. In help of convex optimization techniques (Rockafellar, 1972; Vapnik, 1995), learning is made both on weighting parameters for linear combination and on selecting a small subset of samples for locating kernels $k(x, x_i)$. In the past decade, efforts are made not only beyond the classic SVM limitation of only considering two category classification, but also on how to estimate an appropriate structure for kernels $k(x, x_i)$.

Cross-fertilization of two directions deserves be explored too. The first direction aims at modeling samples by seeking its distribution or structure, based on which classification and also other problem solving tasks are handled. While a crucial nature of SVM is seeking a discriminating strip such that two classes become more apart from each other, merely based on samples around the boundary of two classes. To classifying samples with a more sophisticated structure, studies on the second direction further proceed to estimate an appropriate structure for kernels, for which it is helpful to recall RBF studies that have experienced a path from simple kernels to various structures. On the other hand, studies of the first direction may also get a help from studies of the second direction by enhancing efforts on seeking a more robust boundary structure, e.g., in Algorithms II or III, with $\ell^* = \arg\max_\ell q(z = \ell \mid x, \theta_j)$ we require $q(z = \ell^* \mid x, \theta_j) - \max_{j \neq \ell^*} q(z = j \mid x, \theta_j)$ to be larger than a pre-specified threshold.

The last but not the least, its mathematical relation of Bayesian Ying Yang learning to the classical generalization error bound still remains an open issue. We recall that the concept of generalization error comes from measuring the discrepancy of a model estimated on a training set $\mathbf{X}_N = \{x_t\}_{t=1}^N$ from the true regularity of samples underlying $\mathbf{X}_N$. Actually this concept involves a truth-learner philosophy. That is, there are a truth or regularity and a learner. The learner can not directly see the truth or regularity but can get a set $\mathbf{X}_N$ of samples from the truth or regularity subject to some disturbances or noises. The learner attempts to describe this $\mathbf{X}_N$ as well as future samples from the same truth or regularity. It describes $\mathbf{X}_N$ via measuring an error that the learner fits $\mathbf{X}_N$ (e.g., those minimum fitting error based approaches) or how likely $\mathbf{X}_N$ really comes from a model described by the learner (.e.g., the maximum likelihood based methods). More generally, a generalization error attempts to measure an error that the learner fits not only $\mathbf{X}_N$ but also any samples from the same truth or regularity, which has a prediction nature and thus is difficult to obtain accurately.

In a contrast, a Ying Yang harmony involves a relative philosophy about two matters or systems that interact each other, while the concept of learner-seeking- truth is extended to a concept about how close the two systems are, which may be observed from two general aspects. One is externally observing how the two systems describe $\mathbf{X}_N$, which leads to either a correlation type concept if we are interested in whether two descriptions share certain common points or an indifference concept (a relaxed version of equivalence concept) if we are further interested in how close or how different the two descriptions are. This scenario includes the truth-learner type as a special case that a system is simply the one that generates $\mathbf{X}_N$. Still, we may need to consider the concepts of correlation and indifference about how the

two systems describe samples beyond $\mathbf{X}_N$ but from the same truth or regularity. This lacks study yet but likely involves a trading-off between a minimum fitting error and a least model complexity.

The other aspect is observing how the two systems are, both externally on describing $\mathbf{X}_N$ and internationally on the inner representation $\mathbf{R}$. Not only concepts of correlation and matching should be considered for two systems with respect to both $\mathbf{X}$ and $\mathbf{R}$, but also each system $M(\mathbf{X},\mathbf{R})$ may have two different architectures, e.g., we get two complement systems $p(\mathbf{X} \mid \mathbf{R})p(\mathbf{R})$ as Ying and $p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X})$ as Yang for a joint distribution $p(\mathbf{X},\mathbf{R})$, or generally $M(\mathbf{X} \mid \mathbf{R})M(\mathbf{R})$ and $M(\mathbf{R} \mid \mathbf{X})M(\mathbf{X})$ even beyond a probability theoretic framework. That is, we have the interaction between two complement systems, the one $M(\mathbf{X} \mid \mathbf{R})M(\mathbf{R})$ models or describes the external data $\mathbf{X}_N$, while the other $M(\mathbf{R} \mid \mathbf{X})M(\mathbf{X})$ consists of $M(\mathbf{R} \mid \mathbf{X})$ for perceiving $\mathbf{X}_N$ and $M(\mathbf{X})$ that comes from the data $\mathbf{X}_N$ directly or after smoothing. On such two complement systems that describe both an external data $\mathbf{X}_N$ and the inner representation $\mathbf{R}$, a correlation type concept is further developed into a harmony concept under certain conservation principle (e.g. $\int p(\mathbf{X},\mathbf{R})d\mathbf{X}d\mathbf{R} = 1$), which combines the concepts of a minimum fitting error and a least model complexity while not facing the difficulty of seeking an appropriate trade-off. In other word, this $\mathbf{X}_N$ based Ying Yang harmony closely relates to the classical concept of generalization error that bases on future samples with a difficult prediction nature.

CONCLUSION

Studies on RBF networks have been reviewed along the streams of its developments in past decades. Backtracked to the early 1960's in the literature of nonparametric statistics on Parzen Window estimator and subsequently on kernel regression estimator, advances are featured not only by the era from nonparametric based simple kernels to parameter estimation based normalized RBF networks, but also by the era of extending simple kernels into certain structures, from studies on mixture of experts and its alternatives to studies on subspace based functions (SBF), as well as further extensions for exploring temporal structures among samples. Moreover, different types of typical learning algorithms have also been summarized under the Bayesian Ying Yang learning framework for learning not only normalized RBF, ME and alternatives, but also SBF and temporal extensions.

ACKNOWLEDGMENT

The work described in this paper was supported by a grant from the Research Grant Council of the Hong Kong SAR (Project No: CUHK4173/06E), and also supported by the program of Chang Jiang Scholars, Chinese Ministry of Education for Chang Jiang Chair Professorship in Peking University.

REFERENCES

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 714–723.

Akaike, H. (1981). Likelihood of a model and information criteria. *Journal of Econometrics*, *16*, 3–14.

Amari, S. I., Cichocki, A., & Yang, H. (1996), A new learning algorithm for blind separation of sources, In Touretzky, Mozer, & Hasselmo (Eds.), *Advances in Neural Information Processing System 8*, MIT Press, 757-763.

Bell, A., & Sejnowski, T. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, *7*, 1129–1159.

Botros, S. M., & Atkeson, C. G. (1991), Generalization properties of radial basis function, In Lippmann, Moody, & Touretzky (eds), *Advances in Neural Information Processing System 3*, Morgan Kaufmann Pub., 707-713.

Bozdogan, H. (1987). Model Selection and Akaike's Information Criterion: The general theory and its analytical extension. *Psychometrika*, *52*, 345–370.

Broomhead, D. S., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, *2*, 321–323.

Cavanaugh, J. (1997). Unifying the derivations for the Akaike and corrected Akaike information criteria. *Statistics & Probability Letters*, *33*, 201–208.

Chang, P. R., & Yang, W. H. (1997). Environment-adaptation mobile radio propagation prediction using radial basis function neural networks. *IEEE Transactions on Vehicular Technology*, *46*, 155–160.

Chen, S., Cowan, C. N., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for Radial basis function networks. *IEEE Transactions on Neural Networks*, *2*, 302–309.

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The Helmholtz machine. *Neural Computation*, *7*(5), 889–904.

Devroye, L. (1981). On the almost everywhere convergence of nonparametric regression function estimates. *Annals of Statistics*, *9*, 1310–1319.

Devroye, L. (1987). *A Course in Density Estimation*. Boston: Birkhauser.

Er, M. J. (2002). Face recognition with radial basis function (RBF) neural networks. *IEEE Transactions on Neural Networks*, *13*(3), 697–710.

Fang, S. C., Gao, D. Y., Shue, R. L., & Wu, S. Y. (2008). Canonical dual approach for solving 0-1 quadratic programming problems. *Journal of Industrial and Management Optimization*, *4*(1), 125–142.

Ghahramani, Z., & Beal, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. In Solla, Leen, & Muller (eds), *Advances in Neural Information Processing Systems 12*, MIT Press, 449-455.

Girosi, F., & Poggio, T. (1990). Networks and the best approximation property. *Biological Cybernetics*, *63*(3), 169–176.

Guerra, F. A., & Coelho, L. S. (2008). Multi-step ahead nonlinear identification of Lorenz's chaotic system using radial basis neural network with learning by clustering and particle swarm optimization. *Chaos, Solitons, and Fractals*, *35*(5), 967–979.

Hartman, E. J., Keeler, J. D., & Kowalski, J. M. (1990). Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation*, *2*, 210–215.

Heinen, T. (1996). *Latent class and discrete latent trait models: Similarities and differences*. Housand Oaks, California: Sage.

Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. N. (1995). The wake-sleep algorithm for unsupervised learning neural networks. *Science*, *268*, 1158–1160.

Hinton, G. E., & Zemel, R. S. (1994), Autoencoders, minimum description length and Helmholtz free energy, In Cowan, Tesauro, & Alspector (eds), *Advances in Neural Information Processing Systems 6*, Morgan Kaufmann Pub., 449-455.

Isaksson, M., Wisell, D., & Ronnow, D. (2005). Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks. *IEEE Transactions on Microwave Theory and Techniques*, *53*(11), 3422–3428.

Jaakkola, T. S. (2001), Tutorial on variational approximation methods, in Opper & Saad (eds), *Advanced Mean Field Methods: Theory and Practice*, MIT press, 129-160.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation, 3*, 79–87.

Jordan, M., Ghahramani, Z., Jaakkola, T., & Saul, L. (1999). Introduction to variational methods for graphical models. *Machine Learning*, *37*, 183–233.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation, 6*, 181–214.

Jordan, M. I., & Xu, L. (1995). Convergence Results for The EM Approach to Mixtures of Experts Architectures. *Neural Networks*, *8*, 1409–1431.

Karami, A., & Mohammadi, M. S. (2008). Radial basis function neural network for power system load-flow. *International Journal of Electrical Power & Energy Systems*, *30*(1), 60–66.

Kardirkamanathan, V., Niranjan, M., & Fallside, F. (1991), Sequential adaptation of Radial basis function neural networks and its application to time-series prediction, In Lippmann, Moody, & Touretzky (eds), Advances in Neural Information Processing System 3, Morgan Kaufmann Pub., 721-727.

Konishi, S., Ando, T., & Imoto, S. (2004). Bayesian information criteria and smoothing parameter selection in radial basis function networks. *Biometrika*, *91*(1), 27–43.

Lee, J. (1999). A practical radial basis function equalizer. *IEEE Transactions on Neural Networks, 10*, 450–455.

Lin, G. F., & Chen, L. H. (2004). A non-linear rainfall-runoff model using radial basis function network. *Journal of Hydrology (Amsterdam), 289*, 1–8.

Ma, S., Ji, C., & Farmer, J. (1997). An Efficient EM-based Training Algorithm for Feedforward Neural Networks. *Neural Networks, 10*(2), 243–256.

MacKay, D. (2003), Information Theory, Inference, and Learning Algorithms, Cambridge University Press.

Mackey, D. (1992). A practical Bayesian framework for backpropagation. *Neural Computation*, *4*, 448–472.

Mai-Duy, N., & Tran-Cong, T. (2001). Numerical solution of differential equations using multiquadric radial basis function networks . *Neural Networks*, *14*(2), 185–199.

McLachlan, G. J., & Geoffrey, J. (1997), The EM Algorithms and Extensions, Wiley.

Mel, B. W., & Omohundro, S. M. (1991), How receptive field parameters affect neural learning. In Lippmann, Moody, & Touretzky (eds), Advances in Neural Information Processing System 3, Morgan Kaufmann Pub., 757-763.

Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units . *Neural Computation*, *1*, 281–294.

Neath, A. A., & Cavanaugh, J. E. (1997). Regression and Time Series model selection using variants of the Schwarz information criterion. *Communications in Statistics A, 26*, 559–580.

Nowlan, S. J. (1990), Max likelihood competition in RBF networks, TR. CRG-Tr-90-2, U. of Toronto, Dept. of Computer Science.

Park, J., & Sandberg, I. W. (1993). Universal approximation using radial-basis-function networks. *Neural Computation*, *5*, 305–316.

Poggio, T., & Girosi, F. (1990). Networks for approximation and learning, Networks for approximation and learning. *Proceedings of the IEEE*, *78*, 1481–1497.

Powell, M. J. D. (1987), Radial basis functions for multivariable interpolation: a review, in Mason & Cox (Eds.), Algorithms for Approximation, Oxford: Clarendon Press.

Reddy, R., & Ganguli, R. (2003). Structural damage detection in a helicopter rotor blade using radial basis function neural networks. *Smart Materials and Structures*, *12*, 232–241.

Redner, R. A., & Walker, H. F. (1984). Mixture densities, maximum likelihood, and the EM algorithm. *SIAM Review*, *26*, 195–239.

Rissanen, J. (1986). Stochastic complexity and modeling. *Annals of Statistics*, *14*(3), 1080–1100.

Rissanen, J. (1989), Stochastic Complexity in Statistical Inquiry, World Scientific: Singapore.

Rockafellar, R. (1972), Convex Analysis, Princeton University Press.

Rumelhart, D. E., Hintont, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Salah, A. A., & Alpaydin, E. (2004), Incremental mixtures of factor analyzers, Proc. the 17th International Conference on Pattern Recognition, 23-26 Aug. 2004, Cambridge, UK, Vol.1, 276-279.

Sarimveis, H., Doganis, P., & Alexandridis, A. (2006). classification technique based on radial basis function neural networks. *Advances in Engineering Software*, *37*(4), 218–221.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Shi, L. (2008), Bayesian Ying-Yang harmony learning for local factor analysis: a comparative investigation, In Tizhoosh & Ventresca (eds), Oppositional Concepts in Computational Intelligence, Springer-Verlag, 209-232.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*, 109–118.

Stone, M. (1978). Cross-validation: A review. *Math. Operat. Statist.*, *9*, 127–140.

Sun, K., Tu, S. K., Gao, D. Y., & Xu, L. (2009), Canonical Dual Approach to Binary Factor Analysis. In T. Adali et al. (Eds.), ICA 2009 (LNCS 5441), pp 346-353.

Treier, S., & Jackman, S. (2002), Beyond factor analysis: modern tools for social measurement. Presented at the 2002 Annual Meetings of the Western Political Science Association and the Midwest Political Science Association.

Vapnik, V. (1995), The Nature Of Statistical Learning Theory, Springer.

Vapnik, V. (2006), Estimation of Dependences Based on Empirical Data, Springer.

Wallace, C., & Freeman, P. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society. Series A (General)*, *49*(3), 240–265.

Wallace, C. S., & Boulton, D. M. (1968). An information measure for classification. *The Computer Journal*, *11*, 185–194.

Xu, L. (1995), Bayesian-Kullback coupled YING-YANG machines: unified learning and new results on vector quantization, Proc.ICONIP95, Oct 30-Nov.3, 1995, Beijing, pp 977-988. A further version in NIPS8, D.S. Touretzky, et al (Eds.), MIT Press, 444–450.

Xu, L. (1998). RBF nets, mixture experts, and Bayesian Ying-Yang learning. *Neurocomputing*, *19*(1-3), 223–257.

Xu, L. (2001). Best Harmony, Unified RPCL and Automated Model Selection for Unsupervised and Supervised Learning on Gaussian Mixtures, ME-RBF Models and Three-Layer Nets. *International Journal of Neural Systems*, *11*(1), 3–69.

Xu, L. (2001). BYY harmony learning, independent state space and generalized APT financial analyses. *IEEE Transactions on Neural Networks*, *12*(4), 822–849.

Xu, L. (2002). BYY harmony learning, structural RPCL, and topological self-organizing on unsupervised and supervised mixture models. *Neural Networks*, *15*, 1125–1151.

Xu, L. (2004a). Advances on BYY harmony learning: information theoretic perspective, generalized projection geometry, and independent factor auto-determination. *IEEE Transactions on Neural Networks*, *15*, 885–902.

Xu, L. (2004b). Temporal BYY encoding, Markovian state spaces, and space dimension determination. *IEEE Transactions on Neural Networks*, *15*, 1276–1295.

Xu, L (2005), Fundamentals, Challenges, and Advances of Statistical Learning for Knowledge Discovery and Problem Solving: A BYY Harmony Perspective, Keynote talk. *Proc. Of Intl. Conf. on Neural Networks and Brain*, Oct. 13-15, 2005, Beijing, China, Vol. 1, 24-55.

Xu, L. (2007a), Bayesian Ying Yang Learning, Scholarpedia 2(3):1809, Retrieved from http://scholarpedia.org/article/Bayesian_Ying_Yang_learning.

Xu, L. (2007b), Rival penalized competitive learning, Scholarpedia 2(8):1810, Retrieved from http://www.scholarpedia.org/article/Rival_penalized_competitive_learning

Xu, L. (2007c), A trend on regularization and model selection in statistical learning: a Bayesian Ying Yang learning perspective, In Duch & Mandziuk (eds.), *Challenges for Computational Intelligence*, Springer-Verlag, 365-406.

Xu, L. (2007d). A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving. *Pattern Recognition*, *40*, 2129–2153.

Xu, L. (2008a), Bayesian Ying Yang System, Best Harmony Learning, and Gaussian Manifold Based Family, In Zurada et al (eds.) Computational Intelligence: Research Frontiers, WCCI2008 Plenary/Invited Lectures, LNCS5050, 48–78.

Xu, L. (2008b). (in press). Machine learning problems from optimization perspective, A special issue for CDGO 07. *Journal of Global Optimization*.

Xu, L. (2008c), Independent Subspaces, in Ramón, Dopico, Dorado & Pazos (Eds.), *Encyclopedia of Artificial Intelligence*, IGI Global (IGI) publishing company, 903-912.

Xu, L., Jordan, M. I., & Hinton, G. E. (1994), A Modified Gating Network for the Mixtures of Experts Architecture, Proc. of WCNN94, San Diego, CA, 405-410.

Xu, L., Jordan, M. I., & Hinton, G. E. (1995), An Alternative Model for Mixtures of Experts, In Tesauro, Touretzky & Leen (eds), Advances in Neural Information Processing Systems 7, MIT Press, 633-640.

Xu, L., Krzyzak, A., & Oja, E. (1993). Rival Penalized Competitive Learning for Clustering Analysis, RBF net and Curve Detection. *IEEE Trans. Neural Networks*, *4*(4), 636-649. An early version on Proc. 1992 IJCNN, Nov.3-6, 1992, Beijing, 665-670.

Xu, L., Krzyzak, A., & Yuille, A. L. (1994). On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates and Receptive Field Size. *Neural Networks*, *7*(4), 609–628.

Yuille, A. L., & Grzywacz, N. M. (1989). A mathematical analysis of the motion coherence theory. *International Journal of Computer Vision*, *3*, 155–175.

KEY TERMS AND THEIR DEFINITIONS

Normalized Radial Basis Function (NRBF) and Extended NRBF: A radial basis function (RBF) $f(x) = \sum_{j=1}^{k} w_j \varphi_j(x - c_j, \theta_j)$ is a linear combination of a series of simple function $\{\varphi_j(x - c_j, \theta_j)\}_{j=1}^{k}$, with each called a base that is located at c_j. Each base is classically radial symmetrical from its center c_j, while it becomes unnecessary presently. A RBF is called normalized RBF (NRBF) if we have $\sum_{j=1}^{k} \varphi_j(x - c_j, \theta_j) = 1$, and is further called Extended NRBF when each constant w_j is extended into a function $f_j(x)$. Typically, we consider the cases that $f_j(x) = W_j x + w_j$ is a linear function.

Mixtures-of-Experts (ME) and Alternative ME (AME): Initially, a mixture-of-experts is a weighted combination $f(x) = \sum_{j=1}^{k} g_j(x, \phi) f_j(x, \theta_j)$ of a number of functions with each $f_j(x, \theta_j)$ called expert that is weighted by $g_j(x, \phi) = e^{v_j(x,\phi)} \Big/ \sum_{j=1}^{k} e^{v_j(x,\phi)}$ that is called a gating net with each $v_j(x, \phi)$ implemented by a three layer forward net. Generally, this weighted combination is actually the regression equation of a mixture of conditional distributions, i.e., $q(z \mid x) = \sum_{j=1}^{k} g_j(x, \phi) q(z \mid x, \theta_j)$. All the unknowns are estimated via the maximum likelihood (ML) learning on $q(z \mid x)$ by a generalized Expectation and Maximization (EM) algorithm. Moreover, alternative ME is a particular ME family with $p(z \mid x)$ supported on a finite mixture $q(x \mid \phi) = \sum_{j=1}^{k} \alpha_j q(x \mid \phi_j), \quad \sum_{j=1}^{k} \alpha_j = 1, \quad \alpha_j \geq 0,$ and with its corresponding posteriori as the gating net $g_j(x, \phi) = \alpha_j q(x \mid \phi_j) \Big/ q(x \mid \phi)$. All the unknowns are estimated via the ML learning on $q(z \mid x) q(x \mid \phi) = \sum_{j=1}^{k} \alpha_j q(x \mid \phi_j) q(z \mid x, \theta_j)$ by the EM algorithm.

Subspace Based Functions and Cascaded Extensions: Considering $q(x \mid \phi_j)$ as generated from a subspace with independent factors distributed along each coordinate of an inner *m* dimensional representation $y = [y^{(1)} \ldots, y^{(m)}]$, we get basis functions or the gating net by $g_j(x, \phi) = \alpha_j q(x \mid \phi_j) \Big/ q(x \mid \phi)$ based on different subspaces, resulting in subspace based extensions of ME and RBF networks. That is, we get $f(x) = \sum_{j=1}^{k} g_j(x, \phi) f_j(x, \theta_j)$ as a combination of functions supported on subspace bases, which is thus called subspace based functions (SBF). Moreover, a direct mapping $x \rightarrow \quad z$ by $f_j(x, \theta_j)$ can be replaced by a mapping $x \rightarrow y$ as feature extraction or dimension reduction and then followed by another mapping $y \rightarrow z$, from which we get a cascade $x \rightarrow y \rightarrow z$ mapping. This cascaded extension will discard redundant parts with further improvements on performance.

Model Selection and Two Stage Implementation: It refers to select an appropriate one among a family of infinite many candidate structures $\{\mathbf{S}_\mathbf{k}(\Theta_\mathbf{k})\}$ with each $\mathbf{S}_\mathbf{k}(\Theta_\mathbf{k})$ in a same configuration but in different scales, each of which is labeled by a scale parameter **k** in term of one integer or a set of integers. Selecting an appropriate **k** means getting a structure consisting of an appropriate number of free parameters. Usually, a maximum likelihood (ML) learning is not good for model selection. Classically, model selection is made in a two stage implementation. First, enumerate a candidate set **K** of **k** and estimate the unknown set Θ_k of parameters by ML learning for a solution $\Theta_\mathbf{k}^*$ at each $\mathbf{k} \in \mathrm{K}$. Second, use a model selection criterion $J(\Theta_\mathbf{k}^*)$ to select a best $\mathbf{k}^*$. Several criteria are available for the purpose, such as AIC, CAIC, BIC, cross validation, etc.

Rival Penalized Competitive Learning (RPCL): It is a further development of competitive learning in help of an appropriate balance between participating and leaving mechanisms, such that an appropriate number *k* of individual substructures will be allocated to learn multiple structures underlying observations. With *k* initially at a value larger enough, the participating mechanism is featured by that

a coming sample x_t is allocated to one of the *k* substructures via competition, and the winner adapts this sample by a little bit, while the leaving mechanism is featured by that the rival is de-learned a little bit to reduce a duplicated allocation, which will discard extra substructures, with model selection made automatically during learning.

Bayesian Ying-Yang System: A set $\mathbf{X} = \{x\}$ of samples and its inner representation $\mathbf{R}$ in an intelligent system are jointly considered by their joint distribution in two types of Bayesian decomposition. In a compliment to the famous ancient Ying-Yang philosophy, one decomposition $p(\mathbf{X}, \mathbf{R}) = p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X})$ coincides the Yang concept with a visible domain $p(\mathbf{X})$ as a Yang space and a forward pathway by $p(\mathbf{R} \mid \mathbf{X})$ as a Yang pathway. Thus, $p(\mathbf{X}, \mathbf{R})$ is called Yang machine. Also, $q(\mathbf{X}, \mathbf{R}) = q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})$ is called Ying machine with an invisible domain $q(\mathbf{R})$ as a Ying space and a backward pathway by $q(\mathbf{X} \mid \mathbf{R})$ as a Ying pathway. Such a Ying-Yang pair is called Bayesian Ying-Yang system. The input to the Ying Yang system is through $p(\mathbf{X}) = p(\mathbf{X} \mid \mathbf{X}_N, h)$ directly from a training sample set $\mathbf{X}_N = \{x_t\}_{t=1}^N$, while the inner representation $q(\mathbf{R}) = q(\mathbf{Y}, \Theta) = q(\mathbf{Y} \mid \Theta)q(\Theta \mid \Xi)$ describes both a long term memory Θ that is a collection of all unknown parameters in the system and a short term memory $\mathbf{Y}$ with each $y \in \mathbf{Y}$ corresponding to one element $x \in X$. To build up an entire system, we need to design appropriate structures for each component. Specifically, the structure of $q(\mathbf{Y} \mid \Theta)$ is designed subject to the nature of learning tasks and a principle of least representation redundancy, the structure of $q(\mathbf{X} \mid \mathbf{R})$ is designed to suit the mapping $\mathbf{Y} \to \mathbf{X}$ under a principle of divide and conquer so that a complicated mapping is realized by a number of simple ones, while the structure of $p(\mathbf{R} \mid \mathbf{X})$ is designed for an inverse map $\mathbf{X} \to \mathbf{Y}$ under a principle of uncertainty conversation between Ying-Yang, i.e., Yang machine preserves a room or varying range that is appropriate to accommodate uncertainty or information contained in the Ying machine.

Bayesian Ying Yang Learning: Named in a compliment to the famous ancient Chinese Ying-Yang philosophy, it refers to a general statistical learning framework that formularizes learning tasks in a two pathway featured intelligent system via two complementary Bayesian representations of the joint distribution on the external observation and its inner representation, with all unknowns in the system determined by a principle that two Bayesian representations become best harmony. This system is called Bayesian Ying Yang system, mathematically described by $q(\mathbf{X}, \mathbf{R}) = q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})$ and $p(\mathbf{X}, \mathbf{R}) = p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X})$. This best harmony is mathematically implemented by maximizing $H(p \,||\, q) = \int p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X}) \ln[q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})]d\mathbf{X}d\mathbf{R}$, called Bayesian Ying Yang harmony learning. It follows from $H(p \,||\, q) = -KL(p \,||\, q) + \int p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X}) \ln[p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X})]d\mathbf{X}d\mathbf{R}$ that this best Ying Yang harmony principle includes not only a best Ying Yang matching by minimizing the Ying Yang divergence $KL(p \,||\, q) = \int p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X}) \ln[p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X}) \big/ q(\mathbf{X} \mid \mathbf{R})q(\mathbf{R})]d\mathbf{X}d\mathbf{R}$, but also minimizing the entropy $-\int p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X}) \ln[p(\mathbf{R} \mid \mathbf{X})p(\mathbf{X})]d\mathbf{X}d\mathbf{R}$ of the Yang machine. In other words, a best Ying Yang harmony seeks a Yang machine as an inverse of Ying machine such that it best matches the Ying machine and also keeps a least complexity. Moreover, this best Ying Yang matching provides a general perspective that unifies a number of typical statistical learning approaches.

Automatic Model Selection: Being different from a usual incremental or decremental model selection that bases on evaluating the change $J(\Theta_{\mathbf{k}}) - J(\Theta_{\mathbf{k}} \cup \theta_{new})$ as a subset θ_{new} of parameters is added or removed, automatic model selection is associated with not only a learning algorithm or a learning principle but also an indicator $\rho(\theta_{\mathbf{r}})$ on a subset $\theta_{\mathbf{r}} \in \Theta_{\mathbf{k}}$. If $\theta_{\mathbf{r}}$ consists of parameters of a redundant structural part, learning via either implementing this learning algorithm or optimizing this learning principle will

drive $\rho(\theta_{\mathbf{r}}) \to 0$ and θ_{r} towards a specific value, such that the corresponding redundant structural part is effectively removed. One example of such a learning algorithm is Rival Penalized Competitive Learning, while one example of such a learning principle is Bayesian Ying-Yang Harmony Learning.