The Handbook of Brain Theory and Neural Networks

Second Edition

EDITED BY
Michael A. Arbib

The handbook of brain theory and neural networks / edited by Michael A. Arbib;
editorial advisory board, Shun-ichi Amari . . . [et al.]; editorial assistant, Prudence H. Arbib.
p. cm.
"A Bradford book."
Includes bibliographical references and index.
ISBN 0-262-01197-2
1. Neural networks (Neurobiology)—Handbooks, manuals, etc.
2. Neural networks (Computer science)—Handbooks, manuals, etc.
I. Title: Brain theory and neural networks. II. Arbib, Michael A.
QP363.3.H36 2002
612.8'2—dc21
2002038664

A Bradford Book
THE MIT PRESS
Cambridge, Massachusetts
London, England

satisfies the properties of a Lyaponov function; it is bounded from below and always decreases with time when using the dynamics specified by Equation 23. This can be seen by observing that $dE/dt = -\Sigma_i(\lambda I_i z_i/\tau)(\partial E/\partial z_i)^2$ and recalling that the zs are constrained to be positive by definition. Hence the system converges to a minimum of E.

To understand the global convergence of the system, we examine the Hessian of E, the matrix with components $H_{ij} = \partial^2 E/(\partial z_i \partial z_j)$. We see that on the diagonal we have $H_{ii} = 1/(\lambda I_i z_i)$ and off the diagonal $H_{jk} = \prod_{i \neq j,k} z_i$. From Equation 21 we see that the x_i are always positive, and so the z_i lie in the range [0, 1]. Thus the diagonal elements are all greater than $1/(\lambda I_{max})$ and the off-diagonal elements are all less than 1. By making $\lambda \geq (N-1)/I_{max}$ we can ensure that the Hessian is positive definite; hence E is convex, and so there is a single solution that the system converges to. It was shown, in the large λ limit, that this corresponds to the winner-take-all solution (Yuille and Grzywacz, 1989).

Discussion

Winner-take-all is a special case of softmax. Both problems can be formulated in terms of energy minimization, and both can be solved by a number of continuous-time and discrete-time dynamical systems. Some of these systems can be implemented by VLSI circuits or by biologically plausible mechanisms.

These systems can be generalized in a straightforward way to systems of competitive memories or optimization problems. In these cases only convergence to locally optimal solutions is guaranteed.

Finally, recent results on the large computational power of winner-take-all networks, and their need for only positive weights, are very exciting. They ensure that winner-take-all networks will continue to be a major research topic for computation, biology, and VLSI.

Road Map: Dynamic Systems
Background: Computing with Attractors
Related Reading: Modular and Hierarchical Learning Systems; Optimization, Neural

References

Amari, S., and Arbib, M., 1977, Competition and cooperation in neural nets, in *Syst. Neurosci.* (J. Metzler, Ed.), San Diego: Academic Press, pp. 119–165. ◆

Bridle, J., 1989, Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, in *Neuro-computing: Algorithms, Architectures* (F. Fogelman-Soulie and J. Hefault, Eds.), New-York: Springer-Verlag.

Dev, P., 1975, Perception of depth surfaces in random-dot stereograms, *Int. J. ManMach. Stud.*, 7:511–528.

Elfadel, I. M., 1995, Convex potentials and their conjugates in analog mean-field optimization, *Neural Computat.*, 7:1079–1104.

Elias, S. A., and Grossberg, S., 1975, Pattern formation, contrast control, and oscillations in the short term memory of shunting on-center off-surround networks, *Biol. Cybern.*, 20:69–98.

Hertz, J., Krogh, A., and Palmer, R. G., 1991, *Introduction to the Theory of Neural Computation*, Redwood City, CA: Addison-Wesley. ◆

Horiuchi, T. K., Morris, T. G., Koch, C., and DeWeerth, S. P., 1997, Analog VLSI circuits for attention-based visual tracking, in *Advances in Neural Information Processing Systems 9*, San Mateo, CA: Morgan Kaufmann, pp. 706–712.

Indiveri, G., 1999, Neuromorphic analog VLSI sensor for visual tracking: Circuits and application examples, *IEEE Trans. Circuits and Syst. II Analog Digital Signal Process.*, 46:1337–1347.

Jordan, M. I., and Jacobs, R. A., 1992, Hierarchies of adaptive experts, in *Advances in Neural Information Processing Systems* (J. Moody, S. Hanson, and R. Lippmann, Eds.), San Mateo, CA: Morgan Kaufmann, pp. 985–993.

Lazzaro, J., Ryckebusch, S., Mahowald, M. A., and Mead, C. A., 1989, Winner-take-all networks of O(N) complexity, in *Advances in Neural Information Processing Systems* (D. S. Touretsky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 703–711.

Maass, W., 2000, On the computational power of winner-take-all, *Neural Computat.*, 12:2519–2535.

Marr, D., and Poggio, T., 1977, Cooperative computation of stereo disparity, *Science*, 195:283–328.

Rangarajan, A., 2000, Self-annealing and self-annihilation: Unifying deterministic annealing and relaxation labeling, *Pattern Recogn.*, 33:635–649.

Waugh, F., and Westervelt, R., 1993, Analog neural networks with local competition: I. Dynamics and stability, *Phys. Rev. E*, 47:4524–4536.

Yuille, A. L., and Grzywacz, N. M., 1989, A winner-take-all mechanism based on presynaptic inhibition, *Neural Computat.*, 1:334–347.

Ying-Yang Learning

Lei Xu

Introduction

This article addresses the issue of simultaneously building (1) a bottom-up pathway for encoding a pattern in the observation space into its representation in a representation space and (2) a top-down pathway for decoding or reconstructing a pattern from an inner representation back to a pattern in the observation space. This approach has been widely adopted in the literature of modeling a perception system for decades. A typical example is the ADAPTIVE RESONANCE THEORY (q.v.), developed by Grossberg and Carpenter starting in the 1970s. In the past decade, this approach has been widely adopted in various studies of brain theory and neural networks. Typical examples include Mumford's integrated theory for the corticothalamic and the corticocortical feedback (see THALAMUS), Kawato's theory on the CEREBELLUM AND MOTOR CONTROL (q.v.), and Hinton and colleagues' HELMHOLTZ MACHINES AND SLEEP-WAKE LEARNING (q.v.). Moreover, the LMSER self-organizing rule proposed by Xu in 1991 (reference in Xu, 2001a) also uses a bidirectional architecture for statistical unsupervised learning.

The basic spirit of LMSER self-organizing was further developed into the Bayesian ying-yang (BYY) harmony learning in the mid-1990s. BYY harmony learning formulates the two pathways in a general statistical framework. First, a so-called BYY system is proposed for modeling the two pathways in a coordinated fashion via two complementary Bayesian representations of the joint distribution on the observation space and representation space. As a result, a number of existing major learning problems and learning methods are revisited as special cases from a unified perspective. Second, after further developments in the past several years, a harmony learning theory has been developed from which not only new regularization techniques (see GENERALIZATION AND REGULARIZATION IN NONLINEAR LEARNING SYSTEMS) are obtained from a systematic perspective, but also using this theory on the BYY sys-

tem results in an easily implemented approach for model selection that is made either automatically during parameter learning or sequentially after parameter learning via a new class of criteria. Third, application of the first two achievements to various specific BYY systems with typical structures led to three major learning paradigms, namely unsupervised learning, supervised learning, and temporal modeling, with new insights and a number of new results.

This article provides an introduction to the fundamentals of BYY harmony learning and outlines the major results. Further details are given in Xu (2001a, 2001b, 2002a, 2002b). Moreover, the ability of BYY harmony learning for regularization and model selection is explained from an information-theoretic perspective. A comparative discussion is made to clarify how it differs not only from the minimum message length (MML) and minimum description length (MDL) (Wallace and Dowe, 1999; Rissanen, 1999; also see MINIMUM DESCRIPTION LENGTH ANALYSIS), as well as Bayesian approach (Mackey, 1992), but also from information geometry theory (Csiszar and Tusnady, 1984; see NEUROMANIFOLDS AND INFORMATION GEOMETRY) and from HELMHOLTZ MACHINES AND SLEEP-WAKE LEARNING (q.v.).

The Bayesian Ying-Yang System

We consider a world $\mathbf{X}$ with each object in an observation represented by an $\mathbf{x} \in \mathbf{X}$. Corresponding to each $\mathbf{x}$, there is an inner representation $\mathbf{y} \in \mathbf{Y}$ in the representation domain $\mathbf{Y}$ of a learning system. We consider the joint distribution of $\mathbf{x}$, $\mathbf{y}$, which can be understood from two complementary perspectives.

On the one hand, we can interpret each $\mathbf{x}$ as generated from an invisible inner representation $\mathbf{y}$ via a backward path distribution $q(\mathbf{x}|\mathbf{y})$, called *a generative model* $q(\mathbf{x}) = \int q(\mathbf{x}|\mathbf{y})q(\mathbf{y})d\mathbf{y}$, that maps from an inner distribution $q(\mathbf{y})$. On the other hand, we can interpret each $\mathbf{x}$ as being mapped into an invisible inner representation $\mathbf{y}$ via a forward path distribution $p(\mathbf{y}|\mathbf{x})$, called a *representative model* $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$, that matches the inner density $q(\mathbf{y})$.

The two perspectives reflect the two types of Bayesian decomposition of the joint density $q(\mathbf{x}|\mathbf{y})q(\mathbf{y}) = q(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$ on $\mathbf{X} \times \mathbf{Y}$. Without any constraints, the two decompositions should be theoretically identical. However, in a real consideration, the four components $p(\mathbf{y}|\mathbf{x})$, $p(\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$, $q(\mathbf{y})$ should all be subject to certain structural constraints according to the nature of the learning task. Thus, we usually have two different but complementary Bayesian representations:

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

where (with compliments to the ancient Chinese ying-yang philosophy) $p(\mathbf{x}, \mathbf{y})$ is called the *yang machine*, which consists of the observation space (*yang space*) $p(\mathbf{x})$ and the forward pathway (*yang pathway*) $p(\mathbf{y}|\mathbf{x})$; and $q(\mathbf{x}, \mathbf{y})$ is called the *ying machine*, which consists of the invisible state space (*ying space*) $q(\mathbf{y})$ and the *ying* (or backward) *pathway* $q(\mathbf{x}|\mathbf{y})$. Such a pair of ying-yang models is called a *Bayesian ying-yang (BYY) system.*

From a set χ of samples from the observed world $\mathbf{X}$, the distribution $p(\mathbf{x})$ is given either by an empirical density $p(\mathbf{x}|\chi)$ or a nonparametric estimate $p(\mathbf{x}|\chi, h^2)$ with a unknown smoothing parameter h, as will be further specified later by Equations 8 and 11. The task of learning on a BYY system consists of specifying all the aspects of $p(\mathbf{y}|\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$, $q(\mathbf{y})$ as well as h (if any).

First, we need to design the structure of $q(\mathbf{y})$, which depends on learning tasks that are closely related to the complexity of the world $\mathbf{X}$ that we observe. One typical example is a world $\mathbf{X} = \{X, L\}$ that consists of a number of individual objects to observe, with L denoting a set of labels and each $\ell \in L$ denoting an object. In this case, each $\mathbf{x} = \{x, \ell\}$ contains a feature vector $x = [x^{(1)}, \ldots, x^{(d)}]^T$ observed from the object ℓ, subject to a joint underlying distribution $p(\mathbf{x}) = p(x, \ell)$. Correspondingly, we consider a representation domain $\mathbf{Y} = \{Y, L\}$, subject to a parametric structure of $q(\mathbf{y}) = q(y, \ell)$ that describes the vector $\mathbf{y}$ and the label ℓ jointly. This $q(\mathbf{y})$ is specified by three ingredients. The first consists of a set $\mathbf{k} = \{k, \{m_\ell\}\}$, with k denoting the number of labels in L and m_ℓ being the dimension of either a binary or a real vector $\mathbf{y}$ that corresponds to $\ell \in L$. We call both k, m_ℓ the scales of the representation domain $\mathbf{Y}$. The second ingredient is the functional form of $q(\mathbf{y})$, which is usually prespecified according to the nature of learning task. The third consists of a set θ_y of parameters in this given function form.

Second, we need to design the structures of $p(\mathbf{y}|\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$ that specify the mapping capacity of $x \to y$ and $y \to x$, respectively. Each of the two can be either parametric or structure free. We say $p(u|v)$ is structure free if $p(u|v)$ can be any function that satisfies $\int p(u|v) = 1$, $p(u|v) \geq 0$. A structure-free distribution is actually specified in learning. Given its functional form, a parametric $p(u|v, \theta_{u|v})$ is structured by a set $\theta_{u|v}$ of unknown parameters.

Putting this all together, the nature of a BYY system depends on the structure of $q(\mathbf{y})$ for describing the representation domain $\mathbf{Y}$, and the architecture of a BYY system is featured by a combination of the specific structures of $p(\mathbf{y}|\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$. Discarding a useless architecture where both $p(\mathbf{y}|\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$ are structure free, there remain three choices for a meaningful BYY architecture:

- *Backward architecture* (*B-architecture*): $p(\mathbf{y}|\mathbf{x})$ is structure-free and $q(\mathbf{x}|\mathbf{y})$ is parametric.
- *Forward architecture* (*F-architecture*): $q(\mathbf{x}|\mathbf{y})$ is structure-free and $p(\mathbf{y}|\mathbf{x})$ is parametric.
- *Bidirectional architecture* (*BI-architecture*): Both $p(\mathbf{y}|\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$ are parametric.

Generally speaking, a learning task includes two subtasks. One is called *parameter learning* and is for determining a value of the set θ that consists of all the unknown parameters in $p(\mathbf{y}|\mathbf{x})$, $q(\mathbf{x}|\mathbf{y})$, $q(\mathbf{y})$ as well as h (if any). The other subtask is called *model selection* and is for selecting the scales of representation, since a collection of specific BYY systems with different scales in $\mathbf{k}$ corresponds to a family of specific models that share the same system configuration but with different scales.

Harmony Learning

We consider learning in a broad sense that starts from two $p(u)$, $q(u)$, with each of them having certain unknown parts, in either or both scales and parameters. The task of learning is to specify all the unknowns from the known parts. Our *fundamental learning principle* is to make $p(u)$, $g(u)$ have the *best harmony in* a twofold sense:

- The difference between the resulting $p(u)$, $g(u)$ should be minimized.
- The resulting $p(u)$, $g(u)$ should be of the least complexity.

Mathematically, we use a functional $H(p\|q)$ to measure the *degree of harmony* between $p(u)$ and $q(u)$. When both $p(u)$, $q(u)$ are discrete densities in the following form:

$$q(u) = \sum_{t=1}^{N} q_t \delta(u - u_t), \; \sum_{t=1}^{N} q_t = 1 \tag{2}$$

with $\delta(u)$ being a δ-function, we can simply use the following cross entropy

$$H(p\|q) = \sum_{t=1}^{N} p_t \ln q_t \tag{3}$$

as a typical example of such a measure. The maximization of $H(p\|q)$ has two interesting natures:

- *Matching nature*: With p fixed, $\max_q H(p\|q)$ pushes q toward

$$q_t = p_t, \text{ for all } t \tag{4}$$

- *Least complexity nature*: $\max_p H(p\|q)$ with q fixed pushes p toward its simplest form

$$p(u) = \delta(u - u_\tau), \text{ with } \tau = \arg\max_t g_t \tag{5}$$

or equivalently $p_\tau = 1$, and $p_t = 0$ for other t, which is of least complexity from the statistical perspective (Xu, 2001a).

Thus, the maximization of this functional indeed implements the above harmony purpose mathematically. As shown by Xu (2001a), we can further represent either a discrete or continuous density $q(u)$ in the form of Equation 2 by considering its sample points $\{u_t\}_{t=1}^N$ via the following normalization:

$$\hat{q}_t = q(u_t)/z_q, \; z_q = \sum_{t=1}^N q(u_t) \tag{6}$$

Putting this into Equation 3, we can get a general form of the harmony measure:

$$H(p\|q) = \int p(u) \ln q(u) du - \ln z_q \tag{7}$$

which reduces to Equation 3 when $q(u)$, $p(u)$ are discrete (as in Equation 2), with u enumerated from $u_1, \ldots, u_N$ deterministically.

Moreover, when $p(u)$ is given by its empirical density (Devroye et al., 1996),

$$p_0(u) = \frac{1}{N} \sum_{t=1}^N \delta(u - u_t) \tag{8}$$

a crude approximation $z_q = 1$ will make $H(p\|q)$ in Equation 7 become the likelihood

$$L(\theta) = \sum_{t=1}^N \ln q(u_t) \tag{9}$$

Thus, finding $\max_q H(p\|q)$ becomes equivalent to conventional maximum likelihood (ML) learning.

Generally, the term $\ln z_q$ imposes a regularization on ML learning. Two typical examples are given as follows:

1. *Normalization learning*: With $p(u)$ given by Equation 8, we approximate either a discrete or continuous $q(u)$ by Equation 6 and get Equation 7 in the form

$$H(p\|q) = L(\theta) - \ln z_q, \; z_q = \sum_{t=1}^N q(u_t) \tag{10}$$

with $\ln z_q$ imposing a de-learning on the ML learning, which avoids $q(u)$ overfitting a finite size data set (Xu, 2001a).

2. *Data smoothing learning*: Consider $p(u)$ given by a Parzen window estimate (Devroye et al., 1996)

$$p(u) = p_h(u) = \frac{1}{N} \sum_{t=1}^N G(u|u_t, h^2 I) \tag{11}$$

where, as hereafter in this article, $G(u|\mu, \Sigma)$ denotes a Gaussian density with mean vector μ and covariance matrix Σ. Under a weak constraint $\sum_{t=1}^N p(u_t) \approx \sum_{t=1}^N q(u_t) = z_q$, we can approximately get (Xu, 2001a)

$$z_q \approx \frac{z_q^N(h, k)}{N(2\pi h^2)^{k/2}}, \; z_q^N(h, k) = \sum_{\tau=1}^N \sum_{t=1}^N e^{-0.5(\|u_t - u_\tau\|^2/h^2)} \tag{12}$$

where k is the dimension of u. Thus, Equation 7 becomes

$$H(p\|q) = \int p_h(u) \ln q(u) du + 0.5k \ln (2\pi h^2) + \ln N - \ln z_q^N(h, k) \tag{13}$$

The first term regularizes ML learning by smoothing each likelihood $\ln q(u_t)$ in the near-neighborhood of u_t, and thus is referred to as *data smoothing*. The role of h^2 is equivalent to the hyperparameter in Tikhonov-type regularization (Bishop, 1995), but with a new feature that the other terms balance the first term such that an appropriate h is learned together with θ (Xu, 2001a).

BYY Harmony Learning

The fact that $\max_\theta \int p_0(u) \ln q(u|\theta) du$ leads to ML learning is well known in the literature. Moreover, $\max_\theta \int p^*(u) \ln q(u|\theta) du$, with $p^*(u)$ being the true distribution of samples, has also been studied in developing the Akaike information criterion (AIC) for model selection (Akaike, 1974). However, the least complexity nature of Equation 5 has rarely been studied because it is regarded as useless in a conventional sense. In contrast, least complexity plays an essential role that enables the harmony learning on a BYY system to implement model selection.

To be specific, we put $p(u) = p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, $q(u) = q(\mathbf{x}, \mathbf{y}) = q(\mathbf{x}|\mathbf{y})q(\mathbf{y})$ into Equation 7, and get

$$H(p\|q) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \ln [q(\mathbf{x}|\mathbf{y})q(\mathbf{y})] d\mathbf{x} d\mathbf{y} - \ln z_q \tag{14}$$

Again, the term $-\ln z_q$ imposes regularization on learning either by normalization similar to Equation 10 or by data smoothing similar to Equation 12. This regularization may be simply ignored by setting $z_q = 1$. The details are given in Xu (2000, 2001a). For example, similar to Equation 10, we can simply get

$$z_q = \sum_{t=1}^N q(\mathbf{x}_t|\mathbf{y}_t)q(\mathbf{y}_t) \tag{15}$$

on a set of samples $\chi = \{\mathbf{x}_t\}_{t=1}^N$, where $\mathbf{y}_t$ is estimated during learning as an inner representation of $\mathbf{x}_t$.

Mathematically, harmony learning is implemented by

$$\max_{\theta, \mathbf{k}} H(\theta, \mathbf{k}), \text{ where } H(\theta, \mathbf{k}) = H(p\|q) \tag{16}$$

Unlike the case of Equation 7, the least complexity nature of a BYY system makes selecting $\mathbf{k}$ possible, because now only $p(\mathbf{x})$ is fixed as a nonparametric estimate, while $p(\mathbf{y}|\mathbf{x})$ is not fixed but able to be pushed into its least complexity form during learning. In a B-architecture, $p(\mathbf{y}|\mathbf{x})$ is free and thus will be determined by $\max_{p(\mathbf{y}|\mathbf{x})} H(p\|q)$, resulting in

$$p(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y} - \hat{\mathbf{y}}), \; \hat{\mathbf{y}} = \arg\max_{\mathbf{y}} [q(\mathbf{x}|\mathbf{y})q(\mathbf{y})] \tag{17}$$

In turn, the matching nature of harmony learning will further push $q(\mathbf{x}|\mathbf{y})$ and $q(\mathbf{y})$ toward their corresponding least complexity forms. In a BI-architecture, the learning will similarly push $p(\mathbf{y}|\mathbf{x})$ into its least complexity form, e.g., $p(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y} - f_j(\mathbf{x}, W_{\mathbf{y}|\mathbf{x}}))$ (Xu, 2001a).

As for $q(\mathbf{y}) = q(y, \ell) = q(y|\ell)q(\ell)$, it is not difficult to observe that letting $p(\ell)$ be zero is equivalent to reducing k by one, and that letting the variance of every $q(y^{(j)}|\ell)$, for all $\ell \in L$, be zero is equivalent to removing the jth dimension (i.e., reducing the dimension m by one). In other words, making θ take a specific value is equivalent to forcing k, m_ℓ to be reduced effectively to appropriate scales. So, model selection may come into effect either in parallel with parameter learning or sequentially after making pa-

rameter learning via enumerating k on an appropriate range. That is, we have the following two types of learning implementation:

- *Parameter learning with automated model selection*: We set k, $\{m_\ell\}$ in $\mathbf{k}$ large enough and then implement harmony learning by

$$\max_{\theta} H(\theta),\ H(\theta) = H(\theta, \mathbf{k}) \tag{18}$$

  The least complexity nature Equation 5 will let θ take a specific value such that $\mathbf{k} = \{k, \{m_\ell\}\}$ are effectively reduced to appropriate scales, i.e., model selection is made automatically in parallel with parameter learning.
- *Parameter learning followed by model selection*: Alternatively, we can make parameter learning and model selection sequentially in two steps. In the first step, we enumerate k, m_ℓ from small values incrementally, and at each specific k, m_ℓ we perform parameter learning Equation 18, to get the best parameter value θ^*. Moreover, to simplify the implementation, we can even assume $q(\ell) = 1/k$ and $q(y|\ell)$ comes from a family that satisfies certain constraint (Xu, 2002a). Then in the second step, we select a best k^*, m_ℓ^* by

$$\min_{k,m_\ell} J(k, m_\ell), \text{ where } J(k, m_\ell) = -H(\theta^*, \mathbf{k}) \tag{19}$$

  If there is more than one solution for which $J(k, m_\ell)$ gets the same minimum, we take one with the smallest values on k, $\{m_\ell\}$.

This, two-step implementation can be modified with the first step replaced by alternatives. One is to replace Equation 18 by minimizing the Kullback divergence (see LEARNING AND STATISTICAL INFERENCE)

$$\min_{\theta} KL(\theta) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \ln \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x}|\mathbf{y})q(\mathbf{y})}\, d\mathbf{x}d\mathbf{y} \tag{20}$$

as in the initial work of the BYY learning made in 1995 (reference in Xu, 2002a) and other early studies. In this situation, the first step leads us to a number of existing learning models that are based on the maximum likelihood principle or its equivalents. Another alternative is to replace $H(\theta)$ in Equation 18 by $H(\theta) - \lambda KL(\theta)$ with $\lambda > 0$ gradually reducing toward zero from a given value. Both alternatives can reduce the local minimum effect caused by the winner-take-all mechanism of Equation 17, but at the cost of greater difficulty in handling the integral of $\mathbf{y}$ in Equation 20.

Information-Theoretic Perspective, MML/MDL, and the Bayesian Approach

Alternatively, we can understand harmony learning from an information-theoretic perspective. We consider the transfer of the information in $\mathbf{x}$ from a sender via a communication line to a receiver. Instead of directly encoding $\mathbf{x}$ for transmission, $\mathbf{x}$ is mapped to its inner representation $\mathbf{y}$, and then $\mathbf{y}$ is encoded and sent to the receiver. The receiver then decodes $\mathbf{y}$ to reconstruct $\mathbf{x}$.

Without losing generality, we consider the BYY system with $p(\mathbf{y}|\mathbf{x}) = p(y|x, \theta_{y|x})$, $q(\mathbf{x}|\mathbf{y}) = p(x|y, \theta_{x|y})$, and $q(\mathbf{y}) = q(y|\theta_y)$, as shown in Figure 1. On the sender side, x is mapped into its code y via the yang passage $p(\mathbf{y}|\mathbf{x}) = p(y|x, \theta_{y|x})$ and then y is encoded and sent to the receiver side. On the receiver side, a parametric regression function $\hat{x} = g(y, \theta_g)$ is used to construct x with an error $\varepsilon = x - \hat{x}$. Assuming the functional form of $g(y, \cdot)$ is known at the receiver end, in order to get the original x we need to know not only y but also ε and θ_g, which should be decided at the sender end and then transferred via the communication line, too.

For this purpose, the reconstruction process at the receiver end is simulated by the ying machine at the sender end. First, the yang

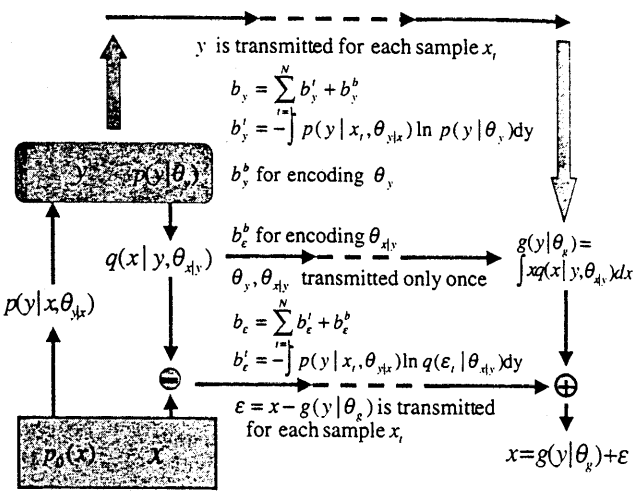

Figure 1. Bayesian ying-yang harmony learning from an information-theoretic perspective.

passage $p(y|x, \theta_{y|x})$ is assumed to come from a known parametric family but with an unknown set of parameters $\theta_{y|x}$. Second, the mapped y is assumed to be exactly described by a distribution of a known parametric family $q(y|\theta_y)$ but with an unknown set of parameters θ_y. Third, we attempt to reconstruct x by the regression $g(y, \theta_g)$ with the residual $\varepsilon = x - g(y, \theta_g)$ that comes from a known parametric family $q(\varepsilon|\theta_\varepsilon)$ but with an unknown set of parameters θ_ε. That is, we have $p(x|y, \theta_{x|y}) = q(x - g(y|\theta_g)|\theta_\varepsilon)$ with $\theta_{x|y} = \{\theta_g, \theta_\varepsilon\}$ and $g(y|\theta_g) = \int xp(x|y, \theta_{x|y})dx$.

We consider the building of the above BYY system from the perspective that the transmission of y, ε, and θ_g is made most efficiently, comparing this approach with the minimum message length (MML) and minimum description length (MDL) approached (Wallace and Dowe, 1999; Rissanen, 1986, 1999), which can be regarded as specific implementations of the more general algorithmic complexity exemplified by the celebrated Kolmogorov complexity (Gammerman and Vovk, 1999).

If we know the true distribution $q(y|\theta_y^*)$ with an exact value of θ_y^*, it follows that the number of bits for encoding y is $b_y^s = \sum_{t=1}^{N} b'_y$, with b'_y being the bits that are needed to encode y for each sample x_t. Since the probability of using a particular y as a code of x_t is $p(y|x_t, \theta_{y|x})dy$, this b'_y should be the expected number of bits to be used at x_t, i.e., $b'_y = -\int p(y|x_t, \theta_{y|x}) \ln q(y|\theta_y^*)dy - c_y$, where $c_y = 0$ when $q(y|\theta_y^*)$ is a discrete probability distribution and $c_y = \ln \delta_y$ when $q(y|\theta_y^*)$ is a continuous density, with δ_y being a quantization resolution. Usually c_y is omitted in the MML/MDL literature, since it is regarded as a constant.

However, on a set of finite samples $\{x_t\}_{t=1}^N$, instead of getting exactly θ_y^* we can obtain only an estimate θ_y that is itself a random variable. Thus, the bits for encoding y consist of the above b_y^s plus b_y^b, i.e., $b_y = b_y^s + b_y^b$, where b_y^b is the number of bits for encoding an estimate θ_y, which does not depend on each individual sample but on the entire data set $\{x_t\}_{t=1}^N$, or equivalently the distribution of θ_y.

Similarly, the number of bits for encoding ε also consist of two parts $b_\varepsilon = \sum_{t=1}^{N} b'_\varepsilon + b^b_\varepsilon$, where $b'_\varepsilon = -\int p(y|x_t, \theta_{y|x}) \ln q(\varepsilon|\theta_\varepsilon)dy = -\int p(y|x_t, \theta_{y|x}) \ln q(x_t|y, \theta_{x|y})dy$ for each sample x_t, and b^b_ε for encoding θ_ε. Moreover, we use b_g to count the bits for encoding θ_g, and then use $b^b_{x|y} = b^b_\varepsilon + b_g$ to denote the total bits for encoding $\theta_{x|y} = \{\theta_g, \theta_\varepsilon\}$, which again does not depend on each individual sample but on the entire batch of data.

Summing up, the total description length is $L_T = Nb_s + b_\theta$ with $b_s = (1/N)\sum_{t=1}^{N}(b_y^t + b_\varepsilon^t)$ being the average number of bits that is needed for each sample x_t and $b_\theta = b_y^b + b_{x|y}^b = b_y^b + b_\varepsilon^b + b_g$ being the total number of bits for encoding $\theta_{x|y}$, θ_y. Thus, the average unit length for each sample x_t is $L_U = b_s + (b_\theta/N)$. Since b_θ does not depend on the size N, (b_θ/N) decreases toward zero as N increases.

It further follows that $b_y^t + b_\varepsilon^t = -\int p(y|x_t, \theta_{y|x}) \ln [q(x_t|y, \theta_{x|y})q(y|\theta_y)]dy$ and thus that

$$b_s = -\int p(y|x_t, \theta_{y|x})p_0(x) \ln [q(x_t|y, \theta_{x|y})q(y|\theta_y)]dxdy$$
$$p_0(x) = \frac{1}{N}\sum_{t=1}^{N}\delta(x - x_t) \quad (21)$$

Comparing this with Equation 14, we have exactly $H(p\|q) = -b_s - \ln z_q^{-1}$, where the second term depends on the specific value of parameters $\theta_{x|y}$, θ_y and the complexity of y, but does not depend on each individual sample x_t. Also, it tends to zero as $N \to \infty$, which is consistent with the role of b_θ/N that tends to zero as $N \to \infty$. So, BYY harmony learning relates closely to the MML/MDL spirit in that both have in common minimizing b_s, i.e., the part of the description length for each sample. For this reason, as well as shown experimentally (Xu, 2001a), BYY harmony learning has model selection ability qualitatively similar to that of MML/MDL. This ability can be understood from the interaction between the two parts in $b_y^t + b_\varepsilon^t$. As the representation scale increases, b_y^t increases while b_ε^t decreases. The minimization of the two parts trades off an appropriate scale for representing y.

However, BYY harmony learning differs from MML/MDL on the specific form for encoding θ. One obvious advantage of BYY harmony learning is that using $\ln z_q^{-1}$ instead of b_θ/N is easy in implementation. Usually, b_θ is difficult to compute and has to be replaced by some rough approximation or bound, with consequent poor actual performance. Further studies are warranted to explore the quantitative relation between $\ln z_q^{-1}$ and b_θ/N to see whether the features of the two can be combined.

In the literature on neural networks learning, it has been widely regarded that the Bayesian approach is equivalent to the MDL approach. However, the situation is not so simple, but depends on how the MDL and the Bayesian approaches are implemented.

One typical Bayesian implementation is for parameter regularization, i.e., a priori density on parameters is assumed such that parameters are determined based on the joint distribution of the parameters and the observed sample data. When MDL was first proposed (Rissanen, 1986), it was implemented basically in equivalence to the MML approach (Wallace and Dowe, 1999), which shares the same spirit of Bayesian regularization in that its first part encodes the fitting residuals and its second part corresponds to the a priori density on parameters. According to the original authors of MML, however, MML actually uses an improper prior density if we insist on relating it to the Bayesian perspective.

Another typical Bayesian implementation is the so-called evidence-based Bayesian approach (Mackey, 1992; see also BAYESIAN METHODS AND NEURAL NETWORKS) using what is called the BIC model selection criterion in the literature of statistics. This one has in principle the closet agreement with the MDL principle that considers an average of all the MML code lengths for all distributions in a family instead of a single MML code length (Rissanen, 1999). In various actual implementations, however, it usually degenerates to be identical to MML after selecting a noninformative uniform prior and approximating the integral of the marginal density via considerable simplification. Interestingly, the latest implementation of MDL uses a so-called normalized maximum likelihood model as the universal model (Rissanen, 1999), which leads to improved code length and becomes different from both MML and the evidence-based Bayesian approach.

BYY harmony learning shares the common spirit of MML as well as Bayesian regularization in the general sense that z_q^{-1} can be regarded as another type of improper prior density on parameters in the BYY system such that the term $-\ln z_q^{-1}$ imposes further regularization on parameter learning, while the interaction between the two parts $b_y^t + b_\varepsilon^t$ in b_s makes model selection implemented either automatically during parameter learning or subsequently after parameter learning via a new class of model selection criteria. It is also possible to further extend BYY harmony learning to share the sprit of MDL and the evidence-based Bayesian approach by normalizing z_q^{-1} into an a priori density $p(\theta)$ and then maximizing an average harmony measure $\int H(\theta)p(\theta)d\theta$. However, this has as extra cost the difficulty in implementing the integral.

Examples of Applications

Applying Equations 18 and 19 to specific BYY systems for various learning tasks, we have obtained not only new insights but also a number of new results. The details are given in Xu (2001a, 2001b, 2002a, 2002b). In the following, we briefly introduce several examples of unsupervised learning.

- *The MSE clustering, number of clusters, and RPCL learning*: Considering a simple B-architecture with $p(\mathbf{y}) = p(y, \ell) = \delta(y - \mu_\ell)/k$ and $p(\mathbf{x}|\mathbf{y}) = p(x|y, \ell) = G(x|y, \sigma^2 I)$, it follows that Equation 18 with the term $-\ln z_q^{-1}$ ignored (i.e., with $z_q = 1$) becomes equivalent to

$$\hat{\ell} = \arg\min_{\ell} \|x_t - \mu_\ell\|^2, \min_{all\ \ell} E_2, E_2 = \sum_{t=1}^{N}\|x_t - \mu_{\hat{\ell}}\|^2 \quad (22)$$

  This is exactly conventional least square clustering or vector quantization, which leads to the well-known k-means algorithm and classical COMPETITIVE LEARNING (q.v.). Moreover, we can get several new results. First obtained by Xu in 1997 (reference in Xu, 2002a), it follows that Equation 19 becomes the following criterion for the best number of clusters

$$k^* = \arg\min_{k} J(k), \text{ with } J(k) = 0.5d \ln E_2 + \ln k \quad (23)$$

  Second, with z_q given by the normalization Equation 15 and $p(\mathbf{y}) = p(y, \ell) = \delta(y - m_\ell)p(\ell)$, not only can we get a modified version of Equation 23 from Equation 19, but it also follows that Equation 18 in parallel implementation becomes equivalent to rival penalized competitive learning (RPCL), proposed by Xu, Krzyzak, and Oja in 1993 (reference in Xu, 2002a), that is able to find the correct number of clusters automatically during learning. Third, these results can also be extended to regularized versions (Xu, 2001b).
- *PCA and Gaussian factor analysis (FA)*: For a B-architecture with $p(\mathbf{y}) = G(y|0, I)$ and $p(\mathbf{x}|\mathbf{y}) = G(x|Ay, \sigma^2 I)$, it follows that Equation 18 with $z_q = 1$ becomes equivalent to PRINCIPAL COMPONENT ANALYSIS (PCA) (q.v.) and Equation 19 becomes

$$m^* = \arg\min_{m} J(m),$$
$$J(m) = 0.5d \ln \sigma^{*2} + 0.5m(\ln 2\pi + 1) \quad (24)$$

  for the principal subspace dimension m. Moreover, Equation 18 with $z_q = 1$ becomes equivalent to Gaussian FA when $p(\mathbf{x}|\mathbf{y}) = G(x|Ay, \Sigma)$, with an adaptive EM algorithm developed for its implementation. Also, Equation 18 with z_q given by the normalization Equation 15 will lead to RPCL-type learning that is able to automatically determine the dimension m.
- *Elliptic RPCL, Gaussian mixture, and local PCA*: For a B-architecture with $p(\mathbf{y}) = p(\ell)\ G(y|\mu_\ell, I)$ and $p(\mathbf{x}|\mathbf{y}) = G(x|A_\ell y,$

$\sigma_\ell^2 I$), from Equation 18 we can obtain (1) both the batch and adaptive EM-type algorithms for either elliptic clustering or ML learning on Gaussian mixtures, (2) an elliptic RPCL algorithm with automated selection on cluster number during learning, (3) extensions to local PCA, and (4) other extensions. Moreover, we can use Equation 19 for selecting both k and the dimensions $\{m_\ell\}$ of local subspaces, which simplifies to

$$\begin{aligned}[k^*, \{m_\ell^*\}] &= \arg \min_{k,\{m_\ell\}} J(k, \{m_\ell\}) \\ J(k, \{m_\ell\}) &= 0.5 \sum_{\ell=1}^{k} p(\ell)[\ln |\Sigma_\ell| \\ &\quad + m_\ell(\ln 2\pi + 1)] - \sum_{\ell=1}^{k} p(\ell) \ln p(\ell), \\ \Sigma_\ell &= A_\ell A_\ell^T + \sigma_\ell^2 I \end{aligned} \tag{25}$$

- *Binary FA, non-Gaussian FA, and local extensions*: For a B-architecture with $p(\mathbf{y}) = \prod_{j=1}^{m} p(y_j)$ where each $p(y_j)$ is a scalar finite mixture (e.g., Gaussian mixture) and $p(\mathbf{x}|\mathbf{y}) = G(\mathbf{x}|Ay, \sigma^2 I)$, from Equation 18 we have obtained both the adaptive EM-type and RPCL-type algorithms for implementing either binary FA when each y_j is binary or non-Gaussian real FA when each y_j is real which was previously studied under the name of Bayesian Ying Yang Kullback dependence reduction in 1998 (reference in Xu, 2000) and further developed with the name changed into the current one. Moreover, from Equation 19 we get criteria for selecting the number m of factors. Furthermore, these results can also be extended to localized versions by considering $p(\mathbf{y}) = p(\ell)\prod_{j=1}^{m_\ell} p_\ell(y_j)$ and $p(\mathbf{x}|\mathbf{y}) = G(\mathbf{x}|A_\ell y, \sigma_\ell^2 I)$.
- *ICA and competitive ICA*: For an F-architecture with $p(\mathbf{y}) = \prod_{j=1}^{m} p(y_j)$ as above and $p(\mathbf{y}|\mathbf{x}) = \delta(y - Wy)$, from Equation 18 we can revisit (1) the *learned parametric mixture–based ICA* algorithm, first proposed by Xu, Yang, and Amari in 1996 (reference in Xu, 2001a), that works not only on cases where some components of y are super-Gaussian and others are sub-Gaussian, but also on cases where W is not invertible but $|WW^T| \neq 0$. Moreover, it has been further extended to a localized version via competition by considering $p(\mathbf{y}) = p(\ell)\prod_{j=1}^{m_\ell} p_\ell(y_j)$ and $p(\mathbf{y}|\mathbf{x}) = p(\ell|x)\delta(y - W_\ell x)$ (Xu, 2002a).
- *LMSER learning, principal ICA, and local extensions*: For a BI-architecture with $p(\mathbf{y}) = \prod_{j=1}^{m} p(y_j)$, $p(\mathbf{x}|\mathbf{y}) = G(\mathbf{x}|Ay, \sigma^2 I)$, and $p(\mathbf{y}|\mathbf{x}) = \delta(y - s(Wx))$, Equation 18 leads us not only to revisit LMSER learning that was first proposed by Xu in 1991 and then directly adopted by Karhunen and Joutsensalo (1994) to implement ICA under the name of nonlinear PCA, but also various extensions, including a so-called principal ICA that corresponds to the direct extension of PCA to ICA. Moreover, from Equation 19 we get criteria for selecting the dimension m. Furthermore, these results have also been extended to localized versions (Xu, 2001b, 2002a).

A number of new results have also obtained on supervised learning and temporal modeling.

For supervised learning, new understandings are obtained on three-layer feedforward nets with backpropagation learning, on the popular mixture-of-experts (ME) model with the corresponding EM algorithm (see MODULAR AND HIERARCHICAL LEARNING SYSTEMS), and on the alternative ME model (Xu, Jordan, and Hinton in Xu, 2002a) as well as the normalized radial basis function (NRBF) network and its extensions. Moreover, various adaptive EM-type learning algorithms are developed from both Equation 18 and Equation 20 since 1998. New criteria have been derived from Equation 19 for deciding the number of hidden units, the number of experts, and the number of basis functions. Also, we get an alternative approach for deciding the set of supporting vectors in SUPPORT VECTOR MACHINES (q.v.). For further details see (Xu, 2001b, 2002b).

Temporal BYY harmony learning has been developed as a general state space approach for modeling data that has temporal relationship among samples, and provides not only a unified point of view on Kalman filter (see KALMAN FILTERING: NEURAL IMPLICATIONS) and HIDDEN MARKOV MODELS (q.v.), but also several new results, such as higher-order HMMs, independent HMMs, temporal ICA, temporal factor analysis, temporal extension of competitive ICA and LMSER learning, and more, with adaptive algorithms for implementation and criteria for selecting the number of states or sources. Further details are supplied in (Xu, 2000, 2001a).

Discussion

Conventional ML learning, as in maximizing $L(\theta)$ in Equation 9, is widely used for estimating θ for a parametric density $q(u|\theta)$ directly on a set $\{u_t\}_{t=1}^N$ of samples. For many practical problems, such as perception, u consists of two parts, $u = (x, y)$, with y invisible. What can be observed is a sample set $\{x_t\}_{t=1}^N$, and thus ML learning is not directly applicable to $q(u|\theta)$. In such cases, ML learning is usually implemented on the marginal density

$$q(x|\theta) = \int q(u|\theta)dy = \int q(x_t|y, \theta_{x|y})q(y|\theta_y)dy \tag{26}$$

which is usually called the factor model or latent variable model or *generative model* in the literature.

However, a direct implementation of ML learning on $q(x|\theta)$ is usually not computationally effective. The problem is solved by two closely related approaches. One is the popular EM algorithm, developed under incomplete data theory (IDT) (Dempster, Laird, and Rubin, 1977). The other is the well-known alternative minimization (Csiszar and Tusnady, 1984), also called the *em* algorithm, developed under information geometry theory (IGT) (see NEUROMANIFOLDS AND INFORMATION GEOMETRY). The approaches work on a class of problems where y takes finite discrete values such that either Equation 26 is a finite mixture or the integral in Equation 26 can be analytically solved. However, in the implementation, we have to compute

$$\begin{aligned} p(y|x) &= \frac{q(x|y, \theta_{x|y})q(y|\theta_y)}{\int q(x|y, \theta_{x|y})q(y|\theta_y)dy}, \text{ and } \max_\theta Q(\theta) \text{ with} \\ Q(\theta) &= \int p(y|x)p_0(x) \ln [q(x|y, \theta_{x|y})q(y|\theta_y)]dxdy \end{aligned} \tag{27}$$

When y is a binary vector of many bits or a non-Gaussian real vector, Equation 27 must be computed either by an exhaustive enumeration or by Monte Carlo approximation, and both those are computationally very expensive. The well-known Helmholtz machine tackles this problem by using a parametric $p(y|x, \theta_{y|x})$ in Equation 20 to avoid the computation on $p(y|x)$, and then minimizes the Helmholtz energy in place of maximizing $Q(\theta)$.

As discussed in the initial work by Xu in 1995 (reference in Xu, 2002a), the BYY system together with implementing Equation 20 provides a unified perspective on understanding not only the above ML learning-related approaches but also a class of information-theoretic approaches. First, $\min_{p(y|x)} KL(\theta)$ in Equation 20 with a free Yang pathway $p(y|x)$ and the empirical density $p_0(x)$ will lead to Equation 27, as well as the equivalence of Equation 20 to ML learning on Equation 26. In other words, the above IDT-and IGT-based approaches are revisited from this new perspective. Second, given a parametric $p(y|x, \theta_{y|x})$, Equation 20 becomes equivalent to minimizing the Helmholtz energy, and a specific design of $p(y|x,$

$\theta_{y|x}$), $q(x|y, \theta_{x|y})$, $q(y|\theta_y)$ will lead us to revisit Helmholtz machine learning. Third, given a parametric $p(y|x, \theta_{y|x})$ but with $q(x|y, \theta_{x|y})$ free, Equation 20 becomes equivalent to minimizing

$$KL_y(\theta_{y|x}, \theta_y) = \int p(y|\theta_{y|x}) \ln \frac{p(y|\theta_{y|x})}{q(y|\theta_y)} dy$$
$$p(y|\theta_{y|x}) = \int p(y|x, \theta_{y|x}) p_0(x) dx \quad (28)$$

which consists of a class of information-theoretic approaches, including both the minimum mutual information approach and the INFOMAX approach for INDEPENDENT COMPONENT ANALYSIS (q.v.).

Moreover, BYY harmony learning goes beyond the approaches discussed above. First, in addition to using Equation 20 for parameter learning, the second step in the two-step implementation of BYY harmony learning provides model selection via a new class of criteria given by Equation 19, sharing a feature similar to the MML/MDL/AC and Bayeisian approaches. Second, the parallel implementation of BYY harmony learning as discussed for Equation 18 provides an easily implementable approach for model selection that is made automatically during parameter learning. Third, the architecture of the BYY system and the term $-\ln z_q^{-1}$ in the harmony function provide new regularization techniques from a systematic perspective. In contrast, learning parameters via minimizing the Kullback divergence is the sole target of the approaches discussed above, while the issues of regularization and model selection are outside the scope of their studies. Even focusing on parameter learning via minimizing Kullback divergence alone, the studies are made from different perspectives with different purposes.

Road Map: Learning in Artificial Networks
Background: Bayesian Methods and Neural Networks; Helmholtz Machines and Sleep-Wake Learning
Related Reading: Adaptive Resonance Theory; Generalization and Regularization in Nonlinear Learning Systems; Learning and Statistical Inference; Model Validation

References

Akaike, H., 1974, A new look at the statistical model identification, *IEEE Trans. Autom. Control*, 19:714–723.

Bishop, C. M., 1995, Training with noise is equivalent to Tikhonov regularization, *Neural Computat.*, 7:108–116.

Csiszar, I., and Tusnady, G., 1984, Information geometry and alternating minimization procedures, *Statist. Decisions*, Suppl. 1, pp. 205–237.

Dempster, A. P., Laird, N. M., and Rubin, D. B., 1977, Maximum likelihood from incomplete data via the EM algorithm, *J. R. Statist. Soc.*, B39:1–38.

Devroye, L., Györfi, L., and Lugosi, G., 1996, *A Probability Theory of Pattern Recognition*, New York: Springer-Verlag.

Gammerman, A., and Vovk, V., Eds. 1999, *Kolmogorov Complexity* (special issue), *Computer J.*, 42(4).

Karhunen, J., and Joutsensalo, J., 1994, Representation and separation of signals using nonlinear PCA type learning, *Neural Netw.*, 7:113–127.

Mackey, D., 1992, A practical Bayesian framework for backpropagation, *Neural Computat.*, 4:448–472.

Rissanen, J., 1999, Hypothesis selection and testing by the MDL principle, *Computer J.*, 42:260–269.

Wallace, C. S., and Dowe, D. R., 1999, Minimum message length and Kolmogorov complexity, *Computer J.*, 42:270–280.

Xu, L., 2000, Temporal BYY learning for state space approach, hidden Markov model and blind source separation, *IEEE Trans. Sign. Process.*, 48:2132–2144.

Xu, L., 2001a, BYY harmony learning, independent state space and generalized APT financial analyses, *IEEE Trans. Neural Netw.*, 12:822–849. ◆

Xu, L., 2001b, Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM models, *Int. J. Neural Syst.*, 11:43–69. ◆

Xu, L., 2002a, BYY harmony learning, structural RPCL, and topological self-organizing on mixture models, *Neural Netw.*, in press.

Xu, L., 2002b, BYY learning, regularized implementation, and model selection on modular networks with one hidden layer of binary units, *Neurocomputing*, in press.