In N. Zhong and J. Liu (eds)
Intelligent Technologies for Information Analysis, Springer, pp661-706, 2004

23. Bayesian Ying Yang Learning (II): A New Mechanism for Model Selection and Regularization

Lei Xu

Chinese University of Hong Kong, Hong Kong

Abstract

Efforts toward a key challenge of statistical learning, namely making learning on a finite size of samples with model selection ability, have been discussed in two typical streams. Bayesian Ying Yang (BYY) harmony learning provides a promising tool for solving this key challenge, with new mechanisms for model selection and regularization. Moreover, not only the BYY harmony learning is further justified from both an information theoretic perspective and a generalized projection geometry, but also comparative discussions are made on its relations and differences from the studies of minimum description length (MDL), the bit-back based MDL, Bayesian approach, maximum likelihood, information geometry, Helmholtz machines, and variational approximation. In addition, bibliographic remarks are made on the advances of BYY harmony learning studies.

23.1 Introduction: A Key Challenge and Existing Solutions

A key challenge to all the learning tasks is that learning is made on a finite size set $\mathcal{X}$ of samples from the world $\mathbf{X}$, while our ambition is to get the underlying distribution such that we can apply it to as many as possible new samples coming from $\mathbf{X}$.

Helped by certain pre-knowledge about $\mathbf{X}$ a learner, $\mathcal{M}$ is usually designed via a parametric family $p(x|\theta)$, with its density function form covering or being as close as possible to the function form of the true density $p_*(x|\cdot)$. Then, we obtain an estimator $\hat{\theta}(\mathcal{X})$ with a specific value for θ such that $p(x|\hat{\theta}(\mathcal{X}))$ is as close as possible to the true density $p_*(x|\theta_o)$, with the true value θ_o. This is usually obtained by determining a specific value of $\hat{\theta}(\mathcal{X})$ that minimizes a cost functional

$$\mathcal{F}(p(x|\theta), \mathcal{X}) \text{ or } \mathcal{F}(p(x|\theta), q_{\mathcal{X}}(x)), \tag{23.1}$$

where $q_{\mathcal{X}}$ is an estimated density of x from $\mathcal{X}$, e.g., given by the empirical density:

$$p_0(x) = \frac{1}{N}\sum_{t=1}^{N}\delta(x - x_t), \quad \delta(x) = \begin{cases} \lim_{\delta\to 0}\frac{1}{\delta^d}, & x = 0, \\ 0, & x \neq 0, \end{cases} \tag{23.2}$$

where d is the dimension of x and $\delta > 0$ is a small number. With a given *smoothing parameter* $h > 0$, $q_{\mathcal{X}}$ can also be the following non-parametric Parzen window density estimate [23.21]:

$$p_h(x) = \frac{1}{N}\sum_{t=1}^{N} G(x|x_t, h^2 I), \tag{23.3}$$

When $p(x|\theta) = p_0(x)$, given by Eq. (23.2), a typical example of Eq. (23.1) is

$$\min_{\theta} \; -\mathcal{F}(p(x|\theta), \mathcal{X}) = -\int p_0(x) \ln p(x|\theta)\mu(dx), \tag{23.4}$$

where $\mu(.)$ is a given measure. It leads to the maximum likelihood (ML) estimator $\hat{\theta}(\mathcal{X})$. For a fixed N, we usually have $\hat{\theta}(\mathcal{X}) \neq \theta_o$ and $p(x|\hat{\theta}(\mathcal{X})) \neq p_*(x|\theta_o)$. Thus, though $p(x|\hat{\theta}(\mathcal{X}))$ best matches the sample set $\mathcal{X}$ in the sense of Eq. (23.1) or Eq. (23.4), $p(x|\hat{\theta}(\mathcal{X}))$ may not well apply to new samples from the same world $\mathbf{X}$.

However, if there is an oracle who tells us the function form of $p_*(x|\cdot)$, we can conveniently use it as the function form of $p(x|\cdot)$. In this case, it follows from the large number law in probability theory that the ML estimator $\hat{\theta}(\mathcal{X}) \to \theta_o$ and $p_*(x|\theta) \to p_*(x|\theta_o)$ as $N \to \infty$. Shortly, the estimator $\hat{\theta}(\mathcal{X})$ is said to be statistically consistent. Actually, this large number law can be regarded as the mathematical formalization of a fundamental philosophy or principle of modern science that a truth about the world exists independent of our perception, and that we will tend to and finally approach the truth as long as the evidences we collected about the truth become infinitely many. However, assuming knowing the true density form of $p_*(x|\cdot)$ implies actually a knowledge on a major structure of the world $\mathbf{X}$ and what to be precisely discovered are remaining details. In many realistic problems we have no such an oracle to tell us the knowledge on the true function form of $p_*(x|\cdot)$ and, thus, in these cases the large number law may fail even as $N \to \infty$.

To avoid the problem, we consider a family $\mathcal{F}$ of density function forms $p(x|\theta_j), j = 1, \cdots, k, \cdots$ with each sharing a same configuration but its structural scale increasing with k such that $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \cdots \mathcal{P}_k \subset \cdots$, where $\mathcal{P}_j = \{p(x|\theta_j)|\forall\theta_j \in \Theta_j\}$. The task of learning is to decide both a best j^* and the corresponding best $\theta^*_{j^*}$ for best describing the true $p_*(x|\theta_o)$. For a finite size set $\mathcal{X}$ of samples, $\mathcal{F}(p(x|\theta_j), \mathcal{X})$ by Eq. (23.1) will monotonically decrease and finally reach 0 as j increases. With a much larger scale, a $p(x|\theta_j)$ that reaches $\mathcal{F}(p(x|\theta_j), \mathcal{X}) = 0$ is usually far from $p_*(x|\theta_o)$. The smaller the sample size, the worse the situation is. Still, as $N \to \infty$ the resulted $p(x|\theta_j)$ will approach $p_*(x|\theta_o)$ if it is included in the family $\mathcal{F}$.

Unfortunately, this ML-type principle is challenged by the fact that the purpose of learning is to guide a learner $\mathcal{M}$ to interact with the world that is usually not only stochastic but also in changing dynamically. Thus, we are not able to collect enough samples either because not a plenty of resources or because not an enough speed to catch the dynamic changing of world.

Therefore, what $\mathcal{M}$ encounters is usually finite number N of samples and, thus, the large number law does not apply.

In past decades, many efforts have been made toward this critical challenge, forming roughly two main streams.

23.2 Existing Solutions

23.2.1 Efforts in the First Stream

By insisting in that there is a true underlying density $p_*(x|\theta_o)$ that apply to all the samples, we desire a best estimate by minimizing $\mathcal{F}(p(x|\theta), p_*(x|\theta_o))$. Unfortunately, this is not directly workable, since $p_*(x|\theta_o)$ is not known. Alternatively, a classic idea is to quantitatively estimate the discrepancy between $\mathcal{F}(p(x|\theta), p_*(x|\theta_o))$ and $\mathcal{F}(p(x|\theta), \mathcal{X})$ such that we have

$$\mathcal{F}(p(x|\theta), p_*(x|\theta_o)) = \mathcal{F}(p(x|\theta), \mathcal{X}) + \Delta(\theta, \theta_o, \mathcal{X}), \tag{23.5}$$

where $\Delta(\theta, \theta_o, \mathcal{X})$ is an estimate of $\mathcal{F}(p(x|\theta), p_*(x|\theta_o)) - \mathcal{F}(p(x|\theta), \mathcal{X})$. This is usually difficult to accurately estimate without knowing $p_*(x|\theta_o)$. In the literature, Δ is usually an estimate on certain bounds of this discrepancy, which may be obtained from $\mathcal{X}$ and the structural features of $p(x|\theta)$, helped by some structural knowledge about $p_*(x|\theta_o)$. Using the bounds, we implement either one or both of the following two types of corrections on estimates from Eq. (23.1):

(a) *Model Selection* We consider a number of candidate models $M_j, j = 1, \cdots, k$, each having its own density function $p(x|\theta_j)$. We estimate each bound Δ_j for the discrepancy between $\mathcal{F}(p(x|\theta_j, M_j), p_*(x|\theta_o))$ and $\mathcal{F}(p(x|\theta_j, M_j), \mathcal{X})$. Over all candidate models, we select the j^*-th model by

$$j^* = arg\min_j [\mathcal{F}(p(x|\theta_j, M_j), \mathcal{X}) + \Delta_j], \tag{23.6}$$

which is referred as *Model Selection*. In the literature, model selection is usually made up of two stages. At the first stage, parameter learning takes place on determining θ_j^* by empirically minimizing $\mathcal{F}(p(x|\theta), \mathcal{X})$. At the second stage, selection of the best j^* takes place by Eq. (23.6). The estimated correcting term Δ_j relies on the complexity of the model M_j, while it does not contain any unknown variables of θ_j.

(b) *Regularization* If we are able to estimate a tighter bound $\Delta(\theta)$ that varies with θ, we can directly get a corrected value θ^* by

$$\theta^* = arg\min_\theta [\Delta(\theta) + \mathcal{F}(p(x|\theta), \mathcal{X})]. \tag{23.7}$$

Such a type of effort is usually referred to as *regularization*, since it regularizes certain singularities caused by a finite number N of samples. Given a model with large enough scale, such a value θ^* makes the model act effectively as

one with a reduced scale. This effective model may neither be identical to that resulted from the above model selection among a number of candidate models $M_j, j = 1, \cdots, k$, nor necessarily lead to $p_*(x|\theta_o)$. However, we have no particular reason to insist on which one is a true density $p_*(x|\theta_o)$ for a small size of samples that actually can be described by many models.

Several approaches have been developed in this stream. One typical example is the VC dimension-based learning theory [23.74], which considers $\mathcal{F}$ as the error or loss of performing a discrete nature task, such as classification or decision, on a set $\mathcal{X}$ of samples, with Δ_j estimated based on a complexity measure of the structure of M_j. The second type of example is AIC [23.1], as well as its extensions AICB, CAIC, etc. [23.2, 23.3, 23.67, 23.11, 23.12, 23.35, 23.36, 23.13], which usually consider a regression or modeling task, with Δ_j estimated as a bias of the likelihood $\int p_0(x) \ln p(x|\theta)\mu(dx)$ to the information measure $\int p_*(x|\theta_o) \ln p(x|\theta)\mu(dx)$. Another typical example is the so-called cross validation [23.64, 23.65, 23.66, 23.55]. Instead of estimating a bound of Δ_j, it targets at estimating $\mathcal{F}(p(x|\theta_j, M_j), p_*(x|\theta_o))$ via splitting $\mathcal{X}$ into a training subset $\mathcal{X}_t$ and a validation subset $\mathcal{X}_v$. First, one gets an estimate $\hat{\theta}_j$ by minimizing $\mathcal{F}(p(x|\theta_j, M_j), \mathcal{X}_t)$ and then estimates $\mathcal{F}(p(x|\theta_j, M_j), p_*(x|\theta_o))$ via jointly considering $\mathcal{F}(p(x|\hat{\theta}_j, M_j), \mathcal{X}_t)$ and $\mathcal{F}(p(x|\hat{\theta}_j, M_j), \mathcal{X}_v)$. Moreover, studies on cross validation relate closely to Jackknife and bootstrap techniques [23.24, 23.25].

Most of studies on these typical approaches are conducted on model selection only, since a rough bound Δ may already be able to give a correct selection among a series of individual models that are discretely different from each other, and, thus, have certain robustness on errors. In contrast, a rough bound $\Delta(\theta)$ usually makes $\min_\theta \Delta(\theta) + \mathcal{F}(p(x|\theta), \mathcal{X})$ lead to a poor performance. However, to get an appropriate bound, $\Delta(\theta)$ requires more knowledge on the true $p_*(x|\theta_o)$, which is usually difficult.

23.2.2 Efforts in the Second Stream

Instead of taking a true underlying density $p_*(x|\theta_o)$ as the target of considerations, the well known Ockham's principle of economy is used as the learning principle. If there are a number of choices for getting a model to fit a set $\mathcal{X}$ of samples, we use the one such that $p(x|\theta)$ not only matches $\mathcal{X}$ well but also has minimum complexity. This principle can be intuitively well understood. When $\mathcal{X}$ consists of a finite number N of samples, we can have infinite choices on $p(x|\theta)$ that describe or accommodate $\mathcal{X}$ well, or better as the complexity of $p(x|\theta)$ increases after satisfying a minimum requirement. That is, learning is a typical ill-posed problem, with intrinsic indeterminacy on its solution. The indeterminacy depends on how large the complexity of $p(x|\theta)$ is. The larger it is, the lower is the chance of getting the true underlying density $p_*(x|\theta_o)$, and, thus, the more likely that the learned choice generalizes poorly beyond the N samples in $\mathcal{X}$. Therefore, we choose the choice with the min-

imum complexity among all those that are able to describe $\mathcal{X}$ sufficiently well.

Based on this principle, approaches have been developed for both regularization and model selection.

(a) One example consists of various efforts either under the name 'regularization' or via certain equivalent techniques. One of most popular one is the well known Tikhonov regularization theory [23.30, 23.68], which minimizes $\mathcal{F}(p(x|\theta), \mathcal{X})$, with a so-called stabilizer that describes the irregularity or non-smoothness of $p(x|\theta)$. In the literature of both statistics and neural networks, there are many efforts that minimize $\mathcal{F}(p(x|\theta), \mathcal{X})$, with a penalty term in various forms. These heuristics take a role similar to that of the Tikhonov stabilizer [23.60, 23.22]. One critical weak point of these efforts is the lack of a systematic or quantitative way to guide how to choose the added term and to control the strength of the term in minimization. In the literature of statistics, the role of the added term is alternatively interpreted as controlling a tradeoff between bias and variance for an estimator [23.27, 23.73].

(b) The second type of efforts for implementing the Ockham's principle consists of those studies based on Bayesian approach. There are three major versions [23.42]. One is called maximum a posteriori probability (MAP), since it maximizes the posteriori probability

$$p(M_j, \theta_j|\mathcal{X}) = p(\mathcal{X}|\theta_j, M_j)p(\theta_j|M_j)p(M_j)/p(\mathcal{X}). \tag{23.8}$$

Specifically, its maximization with respect to θ_j is equivalent to maximizing $\ln[p(\mathcal{X}|\theta_j,\ M_j)\ p(\theta_j|M_j)] = \ln p(\mathcal{X}|\theta_j, M_j) + \ln p(\theta_j|M_j)$, with the first term being a specific example of $\mathcal{F}(p(x|\theta_j), \mathcal{X})$, and $\ln p(\theta_j|M_j)$ acting as a regularization term. That is, it provides a perspective that determines the Tikhonov stabilizer via a priori density $p(\theta_j|M_j)$. Moreover, model selection can be made by selecting j^* with the corresponding $p(\mathcal{X}|\hat{\theta}_j, M_j)p(\hat{\theta}_j|M_j)p(M_j)$ being the largest, where each a priori $p(M_j)$ is usually set uniformly and, thus, ignored, and $\hat{\theta}_j$ is given by either the above MAP regularization or an ML estimator, which is equivalent to using non-informative uniform prior as $p(\theta_j|M_j)$. However, an improperly selected $p(\theta_j|M_j)$ usually leads to a poor performance.

Instead of basing on a special value $\hat{\theta}_j$ of θ_j, the other version of the Bayesian approach makes model selection by selecting j^* as the largest of

$$\begin{aligned} &p(M_j|\mathcal{X}) = p(\mathcal{X}|M_j)p(M_j)/p(\mathcal{X}), \\ &p(\mathcal{X}|M_j) = \textstyle\int p(\mathcal{X}|\theta_j, M_j)p(\theta_j|M_j)d\mu(\theta_j), \end{aligned} \tag{23.9}$$

or, simply, the largest $p(\mathcal{X}|M_j)$, with $p(M_j)$ being regarded as uniform and, thus, ignored. The term $p(\mathcal{X}|M_j)$ is called the evidence (EV) or marginal likelihood, and, thus, it is also referred to as the EV approach. Typical studies include not only the so-called BIC and variants [23.59, 23.38, 23.48] that were proposed as a competitor of AIC and variants in the literature of statistics since the late 1970's, but also those renewed interests in the literature of

neural networks in the last decade, exemplified by the study of [23.45, 23.46, 23.19].

Another version of the Bayesian approach is to use the Bayesian factor (BF)

$$BF_{ij} = p(\mathcal{X}|M_i)/p(\mathcal{X}|M_j) \ , \tag{23.10}$$

i.e., the ratio of evidences, for model comparison via hypothesis testing [23.26, 23.50, 23.40].

A common key problem in all three versions of Bayesian studies is how to get a priori density $p(\theta|M_j)$. Its choice reflects how much a priori knowledge is used. One widely used example is the Jeffery priori or a non-informative uniform priori [23.37, 23.9, 23.45, 23.46, 23.48, 23.42]. Moreover, the EV approach and the BF approach have the problem of how to compute the evidence accurately and efficiently, since it involves an integral. Stochastic simulation techniques such as the importance sampling approach and MCMC are usually used for implementations [23.48, 23.49, 23.14]. Certain comparisons are referred to [23.48, 23.23]. Recently, the Variational Bayes (VB) method has also been proposed in the literature of neural networks as an alternative way for efficient implementation [23.72, 23.28, 23.56].

The third type of efforts is made toward the implementation of Ockham's principle directly. One typical example is called the minimum message length (MML) theory [23.69, 23.70, 23.71], which was first proposed in the late 1960s' as an information measure for classification. The message length is defined via a two part message coding method. First, one needs a length for coding a hypothesis H (or equivalently called a model), described by $\log_2 P(H)$. Second, one needs a length for coding the residuals of using H to fit or interpret the observed set $\mathcal{X}$, described by $\log_2 P(\mathcal{X}|H)$. The two part message length

$$M_L = -\log_2 P(H) - \log_2 P(\mathcal{X}|H) \tag{23.11}$$

is minimized, which is conceptually equivalent to the posterior probability $P(H)P(\mathcal{X}|H)$, where H denotes either a specific parameter θ with a known probability function or a model M. The MML theory closely relates to the MAP approach Eq. (23.8) but actually has a difference. The MML theory considers the coding length of probability instead of considering density in the MAP approach [23.71].

The other typical example is the Minimum Description Length (MDL) theory [23.32, 23.52, 23.53, 23.54]. The basic idea is to represent a family of densities with an unknown parameter set θ, but a given density via a universal model that is able to imitate any particular density in the family. Such a universal model is described by a single probability distribution. Via the fundamental Kraft inequality, one constructs a code, e.g., a prefix code, for such a probability distribution, and, conversely, such a code defines a probability distribution. In this way, we can compare and select among different families by the code length of each family, which explains the name MDL.

A specific implementation of the MDL theory depends on how the code length is described. In the early stage, this length is actually specified via a two part coding method similar to the MML, and, thus, the corresponding implementation of the MDL is basically the same as the MML. Later, the mixture $p(\mathcal{X}|M_j)$ in Eq. (23.9) is used as the universal model for the family of M_j, and, thus, $\ln p(\mathcal{X}|M_j)$ is used as the code length. In this case, the corresponding implementation of the MDL is basically equivalent to the EV or BIC approach, as in Eq. (23.9). However, by selecting a non-informative uniform prior $p(\theta|M_j)$, and approximating the integral in getting the mixture $p(\mathcal{X}|M_j)$ via simplification, an MML code length and the average of all the MML code lengths for all distributions in a family become no different. Thus, this MDL implementation usually becomes identical to the MML. In the latest implementation of the MDL, a so-called normalized maximum likelihood (NML) model is used as the universal model, which leads to an improved code length and becomes different from both the MML and the EV/BIC approach [23.54]. Such a NML is also used to get a new estimate on the BF factor for model comparison.

Both the MML and the MDL can be regarded as specific implementations of more general algorithmic complexity, addressed by the celebrated Kolmogorov complexity. The connections discussed above between MML/MDL and MAP/EV actually reveal the deep relations between the fields of statistics, information theory, and computational complexity theory. Moreover, relations between the first two main types of efforts have also been explored in the past two decades, particularly on typical examples of the first type, such as AIC and cross validation, versus typical examples of the second type, such as MAP, EV/BIC, BF, etc. [23.62, 23.63, 23.64, 23.65, 23.66, 23.7, 23.15]. Furthermore, various applications of all the studies discussed above can be found in the literature, including linear regression, time series modeling, Markov chain [23.41], as well as complicated neural network modeling problems.

23.3 Bayesian Ying Yang Harmony Learning

The Bayesian Ying Yang (BYY) harmony learning was proposed in [23.109], and systematically developed in past years [23.84, 23.85, 23.82, 23.83, 23.80, 23.81, 23.76, 23.77]. This BYY harmony learning acts as a general statistical learning framework not only for understanding various dependence structures, such as generative structures, transform or mapping structures, finite mixture structures, temporal structures, and topological map structures, but also for tackling the key challenge, previously discussed, with a new learning mechanism that makes model selection either *automatically* implemented during parameter learning or *subsequently implemented after* parameter learning via a new class of model selection criteria obtained from this mechanism. Jointly with this BYY harmony learning, new types of regularization have also been proposed, namely a data smoothing technique that provides a new

solution on the hyper-parameter in a Tikinov-like regularization [23.68], a normalization with a new conscience de-learning mechanism that has a nature similar to that of the rival penalized competitive learning (RPCL) [23.114], and a structural regularization imposing certain constraints by designing a specific forward structure in a BYY system, as well as a f-regularization by replacing $\ln(r)$ with a convex function $f(r)$. The details of the f-regularization has been introduced in the previous chapter in this book. The other three regularization approaches will be introduced in the following sections.

23.3.1 Bayesian Ying Yang Harmony Learning

As shown in Fig. 23.1, a BYY system considers coordinately learning two complement representations of the joint distribution $p(x, y)$:

$$p(u) = p(x, y) = p(y|x)p(x),\ q(u) = q(x, y) = q(x|y)q(y), \tag{23.12}$$

basing on $p(x)$ that is estimated from a set of samples $\{x_t\}_{t=1}^N$, while $p(y|x)$, $q(x|y)$ and $q(y)$ are unknowns but subject to certain pre-specified structural constraints. In a compliment to the famous Chinese ancient Ying-Yang philosophy, the decomposition of $p(x, y)$ coincides the Yang concept with the visible domain by $p(x)$ regarded as a Yang space and the forward pathway by $p(y|x)$ as a Yang pathway. Thus, $p(x, y)$ is called Yang machine. Similarly, $q(x, y)$ is called Ying machine with the invisible domain by $q(y)$ regarded as a Ying space and the backward pathway by $q(x|y)$ as a Ying path.

On one hand, we can interpret that each x is generated from an invisible inner representation y via a backward path distribution $q(x|y)$ or called a generative model

$$q(x) = \int q(x|y)q(y)\mu(dy) \tag{23.13}$$

that maps from an inner distribution $q(y)$. In this case, $p(y|x)$ is not explicitly specified or said being free to be specified, while two pre-specified parametric models $q(x|y)$ and $q(y)$ form a backward path to fix the observations of x. We say that the Ying-Yang system in this case has a backward architecture (shortly B-architecture).

On the other hand, we can interpret that each x is represented as being mapped into an invisible inner representation y via a forward path distribution $p(y|x)$ or called a representative model

$$p(y) = \int p(y|x)p(x)\mu(dx) \tag{23.14}$$

that matches the inner density $q(y)$. In this case, $q(x|y)$ is not explicitly specified or said being free to be specified. Forming a forward path, $p(x)$ is estimated from a given set of samples and then is mapped via pre-specified parametric model $p(y|x)$ into $p(y)$ by Eq. (23.14) to match a pre-specified parametric model $q(y)$. We say that the Ying-Yang system in this case has a forward architecture (shortly F-architecture).

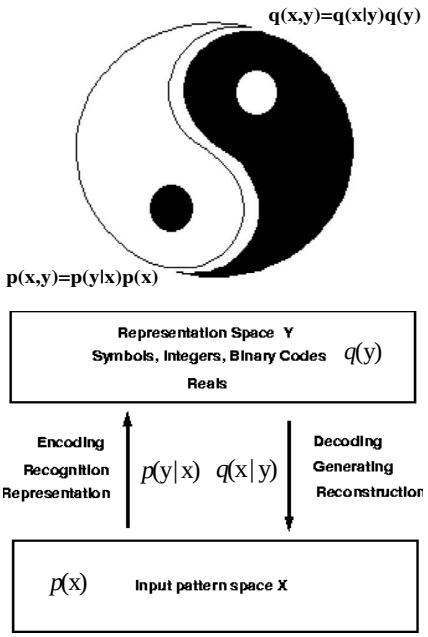


Fig. 23.1. Bayesian Ying Yang System

Moreover, the above two architectures can be combined with $p(y|x)$, $q(x|y)$ and $q(y)$ are all pre-specified parametric models. In this case, we say that the Ying-Yang system in this case has a Bi-directional architecture (shortly BI-architecture).

As discussed in [23.80] and in the previous chapter of this book, types of representation space of y specify types of learning functions that the BYY system can implement, while the above three architectures characterize the performances and computing costs of implementation.

The name of BYY system not just came for the above direct analogy between Eq. (23.12) and the Ying-Yang concept, but also is closely related to that the principle of making learning on Eq. (23.12) is motivated from the well known harmony principle of the Ying-Yang philosophy, which is different from making $p(x)$ by Eq. (23.13) fit a set of samples $\{x_t\}_{t=1}^N$ under the ML principle [23.57] or its approximation [23.58, 23.31, 23.17, 23.18] as well as simply the least mean square error criterion [23.115], and also different from making $q(y)$ by Eq. (23.15) satisfy certain pre-specified properties such as maximum entropy [23.8] or matching the following independent density [23.6]:

$$q(y) = \prod_{j=1}^{m} q(y^{(j)}). \tag{23.15}$$

Under this harmony principle, the Ying-Yang pair by Eq. (23.12) is learned coordinately such that the pair is matched in a compact way as the Ying-Yang

sign shown in Fig. 23.1. In other words, the learning is made in a twofold sense that

- The difference between the two Bayesian representations in Eq. (23.12) should be minimized.
- The resulted entire BYY system should be of the least complexity.

Mathematically, this principle can be implemented by [23.109, 23.82, 23.80]

$$\max_{\theta,m} H(\theta,m), \tag{23.16}$$
$$H(\theta,m) = H(p\|q) = \int p(y|x)p(x)\ln[q(x|y)q(y)]\mu(dx)\mu(dy) - \ln z_q,$$

where θ consists of all the unknown parameters in $p(y|x)$, $q(x|y)$, and $q(y)$ as well as $p(x)$ (if any), while m is the scale parameter of the inner representation y. The task of determining θ is called *parameter learning*, and the task of selecting m is called *model selection* since a collection of specific BYY systems by Eq. (23.12) with different values of m corresponds to a family of specific models that share a same system configuration but in different scales. Furthermore, the term Z_q imposes regularization on learning [23.77, 23.80, 23.83], via two types of implementation. One is called data smoothing that provides a new solution to the hyper-parameter for a Tikinov-like regularization [23.68], and the other is called normalization that causes a new conscience de-learning mechanism similar to that of the rival penalized competitive learning (RPCL) [23.114, 23.83, 23.80].

As described in the previous chapter, considering the harmony measure

$$H(p\|q) = \int p(u)\ln q(u)\mu(du) - \ln z_q. \tag{23.17}$$

Least complexity nature means that $\max_p H(p\|q)$ with q fixed pushes p toward its simplest form

$$p(u) = \delta(u - u_\tau). \tag{23.18}$$

Now, only $p(x)$ is fixed at a non-parametric estimate but $p(y|x)$ is either free in a B-architecture or a parametric form in a BI-architecture and, thus, will be pushed into its least complexity form due to the least complexity nature by Eq. (23.18). For example, in a B-architecture $p(y|x)$ will be determined by $\max_{p(y|x)} H(p\|q)$, resulting in the following least complexity form:

$$p(y|x) = \delta(y - y(x)), \quad y(x) = arg\max_y [q(x|y)q(y)]. \tag{23.19}$$

On the other hand, the matching nature of harmony learning will further push $q(x|y)$ and $q(y)$ toward their corresponding least complexity forms. In other words, the least complexity nature and the matching nature collaborate to make model selection possible such that m is appropriately determined.

As described in [23.80], Eq. (23.16) introduces a new mechanism that makes model selection implemented

– either automatically during the following parameter learning with m initialized large enough:

$$\max_{\theta} \; H(\theta), \quad H(\theta) = H(\theta, m), \tag{23.20}$$

which makes θ take a specific value such that m is effectively reduced to an appropriate one.

– or via the following type of model selection criteria obtained from this mechanism:

$$\min_{m} J(m), \quad J(m) = -H(\theta^*, m), \tag{23.21}$$

which is implemented via parameter learning for θ^* at each value of m that is enumerated from a small value incrementally.

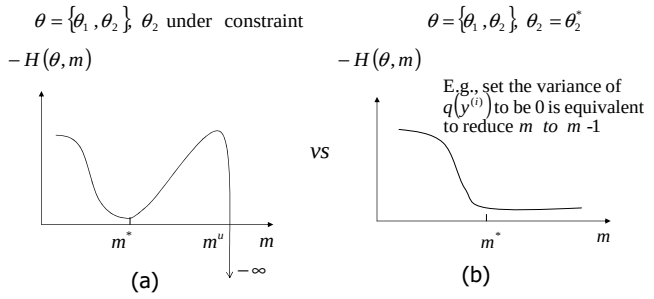


Fig. 23.2. (a) Model selection made after parameter learning on every m in a given interval $[m_d, m_u]$, (b) Automatic model selection with parameter learning on a value m of large enough.

The above feature is not shared by the existing approaches in literature. By the conventional approaches, parameter learning and model selection are made in a two-phase style. First, parameter learning is made usually under the maximum likelihood principle. Then, model selection is made by a different criterion, e.g., AIC, MDL, etc. These model selection criteria are usually not good for parameter learning, while the maximum likelihood criterion is not good for model selection, especially on a small size of training samples.

Specifically, the above parameter learning for getting θ^* can be implemented in help of either Eq. (23.20) or the following Kullback divergence based parameter learning:

$$\min_{\theta} \; KL(\theta) = \int p(y|x)p(x) \ln \frac{p(y|x)p(x)}{q(x|y)q(y)} \mu(dx)\mu(dy). \tag{23.22}$$

Moreover, the implementation of both Eq. (23.20) and Eq. (23.22) can be made by alternatively performing the following two steps:

$$\begin{array}{ll} \text{Ying step:} & \text{fixing } p(x,y), \text{update unknowns in } q(x,y), \\ \text{Yang step:} & \text{fixing } q(x,y), \text{update unknowns in } p(x,y), \end{array} \tag{23.23}$$

which is called the Ying-Yang alternative procedure. It is guaranteed that either of $-H(\theta)$ and $KL(\theta)$ gradually decreases until becomes converged. The details are referred to [23.80, 23.75].

As above discussed, parameter learning by Eq. (23.20) usually leads to an automatic model selection and, thus, there is no need to implement the selection by Eq. (23.21). However, for certain learning tasks, the inner representation is pre-specified to be uniform across both different objects and different dimensions [23.80]. In these cases, automatic model selection will not happen during learning by Eq. (23.20) in the first phase, we need to implement Eq. (23.21) in the second phase.

Particularly, on a B-architecture, the minimization of the above $KL(\theta)$ with respect to a free $p(y|x)$ will result in

$$\begin{aligned} &p(y|x) = \frac{q(x|y)q(y)}{q(x)},\ q(x) = \int q(x|y)q(y)\mu(dy), \\ &KL(\theta) = \int p(x) \ln \frac{p(x)}{q(x)} \mu(dx), \end{aligned} \tag{23.24}$$

which becomes equivalent to ML learning on $q(x)$ when $p(x) = p_0(x)$ is given by Eq. (23.3) [23.109]. In this case, we actually implement the ML learning in the first phase and then model selection by Eq. (23.21) in the second phase.

Without the least complexity nature by Eq. (23.18), the implementation of Eq. (23.22) will not lead to a case of Fig. 23.2(b), and, thus, there is no need to impose the assumption that $q(y)$ comes from a family with equal variances among components.

23.3.2 Structural Inner Representations

The inner representation by Eq. (23.15) is a typical example but not an only example. Actually it is a degenerated case of the multiple modular inner representation discussed by Eq. (1.16) in the previous chapter. That is,

$$\begin{aligned} q(\mathbf{y}) = q(y,\ell) = q(y|\ell)q(\ell),\ q(\ell) = \sum_{j=1}^{k} \alpha_j \bar{\delta}(\ell - j),\ \alpha_\ell \geq 0,\ \sum_{\ell=1}^{k} \alpha_\ell = 1, \\ q(y|\ell) = \textstyle\prod_{j=1}^{m_\ell} q(y^{(j)}|\ell),\ y = [y^{(1)}, \cdots, y^{(m_\ell)}]^T, \\ \text{where } \bar{\delta}(u) = \begin{cases} 1, if \text{u=0}, \\ 0, otherwise; \end{cases} \end{aligned} \tag{23.25}$$

from which we return to Eq. (23.15) when $k = 1$.

When $k \geq 2$, the above Eq. (23.25) also include the following two useful special cases:

– $q(\mathbf{y}) = \int q(y|\ell)q(\ell)dy = q(\ell)$ In this case, we have that $q(x)$ by Eq. (23.13) becomes the following finite mixture

$$q(x) = \sum_{\ell=1}^{k} q(\ell) q(x|\ell). \tag{23.26}$$

which is a weighted sum of k different component densities of $q(x|\ell)$. Moreover, $\max_{\theta,m}\ H(\theta, m)$ by Eq. (23.16) includes directly maximizing $\ln q(y) = \ln q(\ell)$ that not only contains the information of making k selected in a way similar to Eq. (23.21) but also can drive an extra α_ℓ toward zero such that an appropriate k can be automatically decided during learning.

– $q(y) = \sum_{\ell=1}^{k} q(y|\ell) q(\ell)$ In this case, we still have that $q(x)$ by Eq. (23.13) takes the format of Eq. (23.26) but now with

$$q(x|\ell) = \int q(x|y) q(y|\ell) dy. \tag{23.27}$$

In other words, the k different component densities share a common part $q(x|y)$ but with differences in $q(y|\ell)$. Now $\ln q(y)$ still contains the information that can select k in a way similar to Eq. (23.21), e.g., we have $\ln q(y) = -\ln k + \ln \sum_{\ell=1}^{k} q(y|\ell)$ when $q(\ell) = 1/k$. However, maximizing $\ln q(y)$ will not necessarily drive an extra individual α_ℓ toward zero. Thus, k will not be automatically decided during learning.

Generally, Eq. (23.25) covers a representation space with k modules and each module locally consists of m_ℓ independent components. As discussed in [23.80] and also the previous chapter in this book, $\max_{\theta,m}\ H(\theta, m)$ by Eq. (23.16) not only let $\{k, m_\ell\}$ to be selected in a way similar to Eq. (23.21) but also can drive both an extra α_ℓ and the variance of an extra component in $q(y|\ell)$ toward zero such that appropriate $\{k, m_\ell\}$ can be automatically decided during learning.

Specifically, when y is real and non-Gaussian, each component density can be modeled by a Gaussian mixture [23.103, 23.105, 23.82]

$$q(y^{(j)}|\ell) = \sum_{i=1}^{\kappa_{j,\ell}} \beta_{ji\ell} G(y^{(j)}|\mu_{ji\ell}, \lambda_{ji\ell}),\ \sum_{i=1}^{\kappa_{j,\ell}} \beta_{ji\ell} = 1, 0 \le \beta_{ji\ell} \le 1. \tag{23.28}$$

In this case, the information about $\kappa_{j,\ell}$ is contained in $\beta_{ji\ell}$ and Eq. (23.21) can be used for selecting $\kappa_{j,\ell}$. For example, when $q(\ell) = 1/k$, and $\beta_{ji\ell} = 1/\kappa_{j,\ell} = \kappa_\ell$, we have $\ln q(\mathbf{y}) = \ln q(y, \ell)$ with

$$\ln q(y, \ell) = -\ln k - \sum_{j=1}^{m_\ell} \ln \kappa_\ell + \sum_{j=1}^{m_\ell} \ln [\sum_{i=1}^{\kappa} G(y^{(j)}|\mu_{ji\ell}, \lambda_{ji\ell})], \tag{23.29}$$

which does include $-\ln k - \sum_{j=1}^{m_\ell} \ln \kappa_\ell$ that is in favor of smaller size of k, m_ℓ, κ_ℓ. However, maximizing $\ln q(y)$ is made via maximizing $\ln [\sum_{i=1}^{\kappa_{j,\ell}} \beta_{ji\ell} G(y^{(j)}|\mu_{ji\ell}, \lambda_{ji\ell})]$ that will not necessarily drive an individual $\beta_{ji\ell}$ toward zero and, thus, $\kappa_{j,\ell}$ will not be automatically decided during learning.

Automatic selection on $\kappa_{j,\ell}$ can be made by introducing random variables $z_j = 1, \cdots, \kappa_{j,\ell}, j = 1, \cdots, m_\ell$ and re-organizing the structure of the inner representation space as follows

$$q(\mathbf{y}) = q(\ell) \prod_{j=1}^{m_\ell} q(y^{(j)}|\ell, z_j = i) q(z_j = i|\ell),$$
$$q(y^{(j)}|\ell, z_j = i) = G(y^{(j)}|\mu_{ji\ell}, \lambda_{ji\ell}),\ q(z_j = i|\ell) = \beta_{ji\ell}. \tag{23.30}$$

In this case, maximizing $\ln q(\mathbf{y})$ consists of directly maximizing $\ln q(z_j = i|\ell) = \ln \beta_{ji\ell}$ such that each $\kappa_{j,\ell}$ can be selected either via Eq. (23.21) or automatically driving an extra $\beta_{ji\ell}$ toward zero during learning.

It should be noted that different representation spaces also lead to differences on implementing Eq. (23.19). To get a further insight, we here focus on a special case that $k = 1$ and $q(x|y) = G(x|Ay, \Sigma)$, i.e., a linear factor model $x = Ay + e$ from real independent factors by Eq. (23.25) of nonGaussians. This is a typical model for the so-called noisy ICA [23.78]. It follows from Eq. (23.28) that $q(y^{(j)}) = \sum_{i=1}^{\kappa_j} \beta_{ji} G(y^{(j)}|\mu_{ji}, \lambda_{ji})$ and the problem of Eq. (23.19) is a continuous nonlinear optimization problem that has to be tackled by an iterative algorithm [23.82, 23.78]. While it follows from Eq. (23.30) and Eq. (23.19) that $arg\max_{\mathbf{y}}[q(x|\mathbf{y})q(\mathbf{y})] = arg\max_{\mathbf{y}}[\ln G(x|Ay, \Sigma) + \ln q(\mathbf{y})]$ and $\ln q(\mathbf{y}) = \sum_{j=1}^{m}[\ln G(y^{(j)}|\mu_{jz_j}, \lambda_{jz_j}) + \ln \beta_{jz_j}]$. Thus, Eq. (23.19) becomes

$$\max_z \{\sum_{j=1}^{m} \ln \beta_{jz_j} + \max_y L_z(x, y)\}$$
$$\max_y L_z(x, y) = [\ln G(x|Ay, \Sigma) + \sum_{j=1}^{m} \ln G(y^{(j)}|\mu_{jz_j}, \lambda_{jz_j})], \tag{23.31}$$

where $z = [z_1, \cdots, z_m]^T$.

With x and $z_1, \cdots, z_m$ fixed, $\max_y L_z(x, y)$ is a quadratic optimization that can be analytically solved as follow:

$$y_z(x) = [\Lambda_z^{-1} + A^T \Sigma^{-1} A]^{-1}[A^T \Sigma^{-1} x + \Lambda_z^{-1} \mu_z],$$
$$\mu_z = [\mu_{1z_1}, \cdots, \mu_{mz_m}]^T,\ \Lambda_z = diag[\lambda_{1z_1}, \cdots, \lambda_{mz_m}]. \tag{23.32}$$

Then, we can implement the following discrete optimization:

$$z^* = \max_z [L_z(x, y_z(x)) + \sum_{j=1}^{m} \ln \beta_{jz_j}]. \tag{23.33}$$

As a result, the solution of Eq. (23.19) is simply $y_{z^*}(x), z^*$.

Being different from an iterative algorithm [23.82, 23.78], the solution by Eq. (23.33) can be computed analytically. A direct implementation of Eq. (23.33) needs a number $\prod_{j=1}^{m} \kappa_j$ of comparisons. However, this cost can be further reduced by exploring the function structure of $L_z(x, y) + \sum_{j=1}^{m} \ln \beta_{jz_j}$.

23.4 Regularization Versus Model Selection

23.4.1 ML, HL, and Z-Regularization

As discussed in Sect. 23.1, regularization and model selection are two different strategies for tackling the problem of a finite size of samples. Model selection prefers a model that has least complexity for which a compact inner representation is aimed at such that extra representation space can be released. In contrast, regularization is imposed on a model that has a fixed scale of representation space with its complexity larger than needed such that inner representation can spread as uniformly as possible over all the representation space with a distribution that is as simple as possible, which, thus, becomes equivalent to a model with a reduced complexity.

The harmony learning by Eq. (23.20) attempts to compress the representation space via the least complexity that is demonstrated with a winner-take-all (WTA) competition by Eq. (23.19). This type of parameter learning aims at a compact inner representation with an automatic model selection by discarding extra representation space during parameter learning. However, there is no free lunch. The WTA operation by Eq. (23.19) locally per sample will make the learning become sensitive to the initialization of parameters and the way that samples are presented, resulting in that samples are over-aggregated in a small representation space. It usually leads to a local maximum solution for Eq. (23.20). Pre-specifying a uniform inner representation can regularizes the WTA operation. However, the feature of automatic model selection is also lost since the representation space scale is already fixed. Thus, model selection should be made by Eq. (23.21) in the second phase.

With a soft competition by Eq. (23.24) in place of the WTA competition by Eq. (23.19), the ML learning, or equivalently the Kullback divergence based learning (shortly KL learning) by Eq. (23.22) with a B-architecture and an empirical input density by Eq. (23.3), provides a more spread inner representation that improves the local maximum problem. Again, there is no free lunch since its model selection ability is weak, especially on a small size of samples. Thus, making model selection by Eq. (23.21) is needed in the second phase too.

Instead of the above two phase style, regularization to the WTA by Eq. (23.19) may also be imposed to the harmony learning (Shortly HL) by Eq. (23.20) such that automatic model selection still occurs via either some external help or certain internal mechanism.

Externally, we can combine the KL learning by Eq. (23.22) with the harmony learning by Eq. (23.20), in the following three ways:

- The simplest way is to make the KL learning by Eq. (23.22) with the resulted parameters as the initialization of the harmony learning by Eq. (23.20).
- The other way suggested in [23.83] is to let $H(\theta)$ in Eq. (23.20) replaced with $(1-\lambda)H(\theta) - \lambda KL(\theta)$ in help of an appropriate λ.

– Moreover, with $\lambda > 0$ gradually reducing toward zero from a given value such that the regularization role by Eq. (23.22) takes effect at the beginning and then gradually decades as learning goes. That is, we combine KL and HL in a simulated annealing way such that KL is implemented in an early period of learning and is gradually switched to HL as learning goes [23.83, 23.77]. The disadvantage is that the computing cost is very high since parameter learning has to be repeatedly conducted.

Internally, regularization to the WTA by Eq. (23.19) is imposed during the HL learning by Eq. (23.20) via either a BI-architecture or the role of z_q.

Instead of letting $p(y|x)$ free to be decided by Eq. (23.19), we consider a BI-architecture with $p(y|x)$ designed in a structure such that it is not able to become the WTA by Eq. (23.19). Generally, for $p(y|x)$ in the form of

$$\begin{aligned} &p(u|v) = \sum_{j=1}^{n} \beta_j(v) p_j(u|v),\ \sum_{j=1}^{n} \beta_j(v) = 1,\ \beta_j(v) \geq 0, \\ &p_j(u|v) = G(u|f_j(v|\theta_{u|v,j}), \Sigma_{u|v,j}), \end{aligned} \tag{23.34}$$

the harmony learning by Eq. (23.20) will push it toward the following form of least complexity [23.84, 23.82, 23.80]

$$p(u|v) = \sum_{j=1}^{n} \beta_j(v) \delta(u - f_j(v, \theta_{u|v,j})),\ \sum_{j=1}^{n} \beta_j(v) = 1,\ \beta_j(v) = 0,\ or\ 1,$$

unless extra constraints are imposed to prevent $\Sigma_{u|v,j} \to \varepsilon^2 I$ and ε^2 tends to zero. Moreover, Eq. (23.19) is simplified into

$$\begin{aligned} &p(y|x) = \delta(y - y(x)),\ y(x) = f_{j^*(x)}(x|\theta_{y|x,j^*(x)}), \\ &j^*(x) = arg \max_j [q(x|y)q(y)]_{y=f_j(x|\theta_{y|x,j})}, \end{aligned} \tag{23.35}$$

where the maximum is searched by simply enumerating n possibilities. Thus, regularization can also be observed from the perspective that the number of local maxima considerably reduces in comparison with Eq. (23.19). However, there is no free lunch too. The problem is transferred to the difficulty of per-specifying the function form of each $y = f_j(x|\theta_{y|x,j})$. If each function is too simple, the representation ability of $p(y|x)$ is limited and is far from the optimal one. If it is too complicated with too much free parameters, it creates certain problems that need regularization to be imposed too.

Regularization to the WTA by Eq. (23.19) may also be imposed via the so-called z-regularization. This type of regularization can be implemented either by data smoothing or by normalization.

For data smoothing regularization, one simple way is only considering smoothing on x via $p(x) = p_{h_x}(x)$ by Eq. (23.3) with $z_q = \sum_{t=1}^{N} p_{h_x}(x_t)$. As discussed in [23.82, 23.83, 23.80], the regularization is made via $h_x^2 > 0$ while h is determined in help of $-\ln z_q$. Moreover, a smoothing can be imposed on y via modifying $p(y|x) = \delta(y - y(x))$ in Eq. (23.19). For example, in

the case of only one object (i.e., $k = 1$), we let $p(y|x) = G(y|y_t, h_y^2 I)$ and $z_q = (2\pi h_y)^{-0.5m} \sum_{t=1}^N p_h(x_t)$.

For normalization regularization, we have also different implementations.

When y takes either a discrete value $1, \cdots, k$ or is a binary vector $y = [y^{(1)}, \cdots, y^{(m)}]$, and also when $q(x|y)$ and $q(y)$ are both Gaussian, we can consider the constraint $\sum_{t=1}^N \int \frac{q(x_t|y)q(y)}{z_q} \mu(dy) = 1$ because the integral over y is either a summation or analytically solvable. Thus, we have

$$z_q = \sum_{t=1}^N q(x_t), \quad q(x_t) = \int q(x_t|y)q(y)\mu(dy). \tag{23.36}$$

In other cases, this integral over y is difficult to compute. Even when it becomes a computable summation for a binary vector $y = [y^{(1)}, \cdots, y^{(m)}]$, the computing cost will increase exponentially with m.

One solution is to let the integral over the entire domain of y to be approximated by a summation on a set Y_t that consists of a few number of samples y_τ as follows:

$$q(x_t) = \gamma_t \sum_{y_\tau \in Y_t} q(x|y_\tau)q(y_\tau), \ \gamma_t = 1/\sum_{y_\tau \in Y_t} q(y_\tau), \tag{23.37}$$

where γ_t makes $q(y_\tau)/\sum_{y_\tau \in Y_t} q(y_\tau)$ represent discrete probabilities that weight $q(x|y_\tau)$ such that $q(x_t)$ is closer to a marginal density.

One other solution is consider $\sum_{t=1}^N \sum_{y_\tau \in Y_t} \frac{q(x_t|y_\tau)q(y_\tau)}{z_q} \mu(dy) = 1$, which results in

$$z_q = \sum_{t=1}^N \sum_{y_\tau \in Y_t} q(x_t|y_\tau)q(y_\tau). \tag{23.38}$$

The set Y_t can be obtained according to $p(y|x)$. One way is randomly picking a set of samples of y according to $p(y|x)$. The other way is getting only one $y_t = y(x_t)$ for each x_t via either the peak point (e.g., by Eq. (23.19)) or the mean point (e.g., $y(x)$ by Eq. (23.35)) of $p(y|x)$.

In the cases that there is only one sample y_t in Y_t, it follows from Eq. (23.36) and Eq. (23.38) that

$$z_q = \begin{cases} \sum_{t=1}^N q(x_t|y_t)q(y_t), & \text{(a) by Eq. (23.38)}, \\ \sum_{t=1}^N q(x_t|y_t), & \text{(b) by Eq. (23.36)}. \end{cases} \tag{23.39}$$

Further with $p(x) = p_{h_x}(x)$ given by Eq. (23.3), $H(p\|q)$ by Eq. (23.17) either on a B-architecture with Eq. (23.19) or on a BI-architecture with Eq. (23.35) can be unified into the following representation

$$\begin{aligned} &H(p\|q) = \\ &\tfrac{1}{N} \sum_{t=1}^N \int \delta(y - y(x_t)) \ln [q(x_t|y)q(y)]\mu(dy) - \ln z_q + 0.5{h_x}^2 \pi_q, \end{aligned} \tag{23.40}$$

$$\begin{aligned} &\pi_q = \\ &\tfrac{1}{N} \sum_{t=1}^N Tr[\tfrac{\partial^2 \ln q(x|y_t)}{\partial x \partial x^T}]_{x=x_t}, \end{aligned}$$

and we have the following gradient

$$\nabla_\theta H(p\|q) = 0.5{h_x}^2 \nabla_\theta \pi_q$$

$$+\frac{1}{N}\sum_{t=1}^{N}\int[\delta(y-y(x_t))-\eta_t(y)]\nabla_\theta \ln[q(x_t|y)q(y)]\mu(dy),$$

$$\eta_t(y)=\begin{cases}0, & z_q=1,\\ \bar{\delta}(h_x)\sum_{y_\tau\in Y_t}\frac{q(x_t|y_\tau)q(y_\tau)}{z_q}\delta(y-y_\tau), & z_q \text{ by Eq. (23.36)},\\ \bar{\delta}(h_x)\sum_{y_\tau\in Y_t}\gamma_t\frac{q(x_t|y_\tau)q(y_\tau)}{z_q}\delta(y-y_\tau), & z_q \text{ by Eq. (23.37)},\end{cases}$$

$$\bar{\delta}(h_x)=\begin{cases}1, & h_x=0,\\ 0, & h_x>0.\end{cases} \tag{23.41}$$

For $h_x>0$, we have $\bar{\delta}(h_x)=0$ and $\eta_t(y)=\delta(y-y(x_t))$ for all the cases. In this case, the data smoothing regularization is imposed via $0.5h_x^2\nabla_\theta\pi_q$ and an appropriate regularization strength h_x^2 is determined via maximizing $-\ln z_q+0.5{h_x}^2\nabla_\theta\pi_q$ with $z_q=\sum_{t=1}^{N}p_{h_x}(x_t)$. The details are referred to [23.84, 23.85, 23.82, 23.83, 23.80, 23.81, 23.76, 23.77].

For $h_x=0$, we have $0.5{h_x}^2\nabla_\theta\pi_q=0$ and $\bar{\delta}(h_x)=1$. In this case, the normalization regularization is imposed via $-\ln z_q$, which can be observed via the difference of $\eta_t(y)$ in Eq. (23.41). It introduces a degree of conscience de-learning on each updating direction $\nabla_\theta \ln[q(x_t|y)q(y)]$ to avoid over-fitting on each sample pair x_t, y_τ. With and without $-\ln z_q$ in action, $\nabla_\theta H(p\|q)$ takes the same format, and also adaptive updating can be made in the form of $\eta_t(y)\nabla_\theta \ln[q(x_t|y)q(y)]$ per sample x_t.

23.4.2 KL-λ-HL Spectrum

The KL learning by Eq. (23.22) on a BYY system is not limited to just the ML learning. Even on a B-architecture with $p(y|x)$ determined by Eq. (23.24), letting $p(x)=p_h(x)$ by Eq. (23.3) will make the KL learning by Eq. (23.22) perform a regularized ML learning.

Moreover, the KL learning by Eq. (23.22) on a BI-architecture was suggested in [23.109] with $p(y|x)$ in a given parametric family $\mathcal{P}^S_{y|x}$. If the posteriori estimation by Eq. (23.24) is contained in this family $\mathcal{P}^S_{y|x}$, the situation will be equivalent to the KL learning by Eq. (23.22) with a B-architecture; if not, the posteriori estimation by Eq. (23.24) will be approximated by the closest one within the family $\mathcal{P}^S_{y|x}$. This architecture leads to an advantage that the computing difficulty on the integral in $p(y|x)$ by Eq. (23.24) is avoided by an easy implementing parametric model. In its sprit, this is equivalent to those approaches called variational approximation to the ML learning on $q(x)$ [23.58].

Beyond the approximation purpose, studies on the KL learning by Eq. (23.22) on a BI-architecture were also made along two directions since 1996 [23.110]. One is to design a parametric model that makes the inner representation more spreading than that of $p(y|x)$ by Eq. (23.24) such that ML learning is further regularized. The other is to design a parametric model that makes the inner representation more concentrated such that it tends to facilitate automatic model selection. One family of such designs is as follows:

$$p(y|x) = \frac{\psi(q(x|y), q(y))}{\int \psi(q(x|y), q(y))\mu(dy)}, \tag{23.42}$$

which returns to $p(y|x)$ by Eq. (23.24) for the ML learning when $\psi(\xi, \eta) = \xi\eta$. It makes the inner representation either more spreading for a regularized ML learning, e.g., when $\psi(\xi, \eta) = \lambda_1 \ln \xi + \lambda_2 \ln \eta, \lambda_1 \geq 0, \lambda_2 \geq 0$, or more concentrated to facilitate model selection, e.g., when $\psi(\xi, \eta) = e^{\lambda_1 \xi} + e^{\lambda_2 \eta}, \lambda_1 \geq 0, \lambda_2 \geq 0$ that will make $p(y|x)$ tend to Eq. (23.19) as $\lambda_1 = \lambda_2 \to \infty$. A simply form can even be $\psi(\xi, \eta) = (\xi\eta)^\lambda$ which varies from spreading cases to concentrated cases as λ increases from 0 to ∞.

As discussed in the previous subsection, the HL learning with the WTA by Eq. (23.19) on a B-architecture will also be regularized by a BI-architecture with $p(y|x)$ in a more spreading representation. Except those extreme cases that become equivalent to the WTA by Eq. (23.19), e.g., when $\psi(\xi, \eta) = (\xi\eta)^\lambda$ with $\lambda \to \infty$, $p(y|x)$ by Eq. (23.42) generally leads to a regularized harmony learning. Even when $\psi(\xi, \eta) = \xi\eta$, we will not be lead to the ML learning but to a regularized HL learning in a B-architecture with a free $p(y|x)$ replaced by a posteriori estimation by Eq. (23.24).

From the above discussion, we observe that the KL learning by Eq. (23.22) and the HL learning by Eq. (23.20) become closely related via appropriately designing $p(y|x)$. The difference lays in the following term E_p:

$$E_p = -\int p(y|x)p(x) \ln [p(y|x)p(x)]\mu(dx)\mu(dy) + \ln z_p. \tag{23.43}$$

When $p(y|x) = \delta(y - y(x))$ is deterministic and $p(x) = p_0(x)$ given by Eq. (23.2), we have $E_p = 0$. That is, the KL learning by Eq. (23.22) and the HL learning by Eq. (23.20) becomes equivalent on this special BI-directional architecture. When $p(y|x)$ is free to be determined via learning, the difference is that the HL learning by Eq. (23.20) automatically results in a deterministic type $p(y|x)$ by Eq. (23.19) while the KL learning by Eq. (23.22) will result in a non- deterministic type $p(y|x)$ by Eq. (23.24).

Moreover, the KL learning by Eq. (23.22) and the HL learning by Eq. (23.20) are also related for those $p(y|x)$ such that $E_p = c \neq 0$ becomes a constant irrelevant to any unknown parameters in θ, e.g., with $p(x) = p_0(x)$ given by Eq. (23.2) we have

$$E_p = 0.5m \ln (2\pi h_y^2) - \ln N, \ \ for\ p(y|x) = G(y|y(x), h_y^2 I). \tag{23.44}$$

The KL learning by Eq. (23.22) and the HL learning by Eq. (23.20) are no longer equivalent when h_y, m are unknown to be determined. In the special case that h_y, m are prefixed in advance, the KL learning by Eq. (23.22) further becomes equivalent to

$$\max_{\theta,\ s.t.\ E_p = c \neq 0} H(\theta). \tag{23.45}$$

The above discussions also apply to BYY systems with a F-architecture, with a free $p(x|y)$ decided by

$$p(x|y) = p(y|x)p(x)/p(y),\ p(y) = \int p(y|x)p(x)\mu(dx), \tag{23.46}$$

such that we have

$$KL(\theta) = \int p(y) \ln \frac{p(y)}{q(y)} \mu(dy), \;\; H(\theta) = -KL(\theta) - E_p, \tag{23.47}$$

with E_p given in Eq. (23.43). When $q(y)$ is a uniform distribution, minimizing this $KL(\theta)$ becomes equivalent to maximizing the entropy, which maximizes the information transfer from input data to its inner representation via the forward path. Generally, $-KL(\theta)$ describes the incremental of information contained in the representation of y after this information transfer. Thus, minimizing this $KL(\theta)$ is equivalent to making this incremental maximized via maximizing this information transfer against upon the information already in the inner representation. Particularly, when $p(y|x) = \delta(y - Wx)$, $q(y)$ by Eq. (23.15), and $p(x) = p_0(x)$ by Eq. (23.2), both the KL learning by Eq. (23.22) and the HL learning by Eq. (23.20) become equivalent to the minimum mutual information approach for ICA that was previously discussed after Eq. (23.15). All these cases are featured by maximum information transfer and, thus, shortly called as the Max-Inform approach.

Generally, the KL learning by Eq. (23.22) and the HL learning by Eq. (23.20) are different for those $p(y|x)$ that do not satisfy $E_p = c$. This difference can also be observed from the learning results in those cases that the KL learning by Eq. (23.22) results in only $p(x, y) = q(x, y)$, but does not the least complexity nature, while the HL learning by Eq. (23.20) results in not only $p(x, y) = q(x, y)$ but also a minimized entropy

$$\begin{array}{l} H_q = -\int q(x, y) \ln q(x, y) \mu(dx) \mu(dy) \; \textit{or equivalently} \\ H_p - \int p(x, y) \ln p(x, y) \mu(dx) \mu(dy), \end{array} \tag{23.48}$$

which makes model toward a least complexity.

In a summary, the family of KL learning by Eq. (23.22) and the family of HL learning by Eq. (23.20) do share an intersection that consists of interesting models. However, two families are different with each containing useful models outside this intersection. The union of the two families consists of a spectrum of learning models, ranging from regularized ML or Max-Inform versions to the original ML or Max-Inform versions, and then reaching regularized versions of HL learning and finally to the HL learning. In addition, as discussed previously in Sect. 23.4.1, regularized versions of ML or Max-Inform and the HL learning are also obtainable by the role of z_p and z_q via either data smoothing or normalization [23.77, 23.83, 23.84, 23.94].

This spectrum can be extended via a convex combination $\lambda KL(\theta) + (1 - \lambda)H(\theta), \;\; 0 \leq \lambda \leq 1$. Its minimization is equivalent to the KL learning when $\lambda = 1$ and then tends to the HL learning as λ decreases from 1 to 0. As λ varies from 0 to 1, the HL learning is regularized toward to the KL learning.

The combination may go beyond the above spectrum, which can be observed by considering a B-architecture with $p(y|x)$ free. It follows from $H(\theta) = -KL(\theta) - E_p$ that

$$\lambda KL(\theta) - (1-\lambda)H(\theta) = \lambda[E_p + \frac{1}{\lambda}H(\theta)]. \quad (23.49)$$

Ignoring the regularization role of z_p and z_q by setting $z_p = 1, z_q = 1$, we can further get

$$\begin{aligned} E_p + \frac{1}{\lambda}H(\theta) &= \int p(y|x)p(x) \ln \frac{p(y|x)p(x)}{[q(x|y)q(y)]^{\frac{1}{\lambda}}} \mu(dx)\mu(dy) \\ &= \int p(y|x)p(x) \ln \frac{p(y|x)}{p_Q(y|x)} \mu(dx)\mu(dy) + \int p(x) \ln \frac{p(x)}{\hat{q}(x)} \mu(dx)\mu(dy), \\ p_Q(y|x) &= [q(x|y)q(y)]^{\frac{1}{\lambda}}/\hat{q}(x),\ \hat{q}(x) = \int [q(x|y)q(y)]^{\frac{1}{\lambda}} \mu(dy). \end{aligned} \quad (23.50)$$

which was firstly proposed in [23.101]. Its minimization with respect to a free $p(y|x)$ will lead to

$$\begin{aligned} p(y|x) &= p_Q(y|x) = \frac{[q(x|y)q(y)]^{\frac{1}{\lambda}}}{\hat{q}(x)}, \\ E_p + \frac{1}{\lambda}H(\theta) &= \int p(x) \ln \frac{p(x)}{\hat{q}(x)} \mu(dx), \end{aligned} \quad (23.51)$$

where $p(y|x)$ here is a special case of Eq. (23.42) with $\psi(\xi, \eta) = (\xi\eta)^{\frac{1}{\lambda}}$ that makes the inner representation more concentrated than $p(y|x)$ by Eq. (23.24). Minimizing $\int p(x) \ln \frac{p(x)}{\hat{q}(x)} \mu(dx)$ is different from both the KL learning by Eq. (23.22) with $p(y|x) = p_Q(y|x)$ and from the ML learning on $\hat{q}(x)$ since $\int \hat{q}(x)\mu(dx) \neq 1$.

The spectrum can be further extended by considering a linear combination $\lambda KL(\theta) - (1-\lambda)H(\theta)$ with $\lambda > 1$, which is no longer a convex combination since $1 - \lambda < 0$. However, it is still meaningful by observing $E_p + \frac{1}{\lambda}H(\theta)$, and Eq. (23.51) still applies. The difference is that $1/\lambda < 1$ makes the inner representation more spreading than that of the ML learning, with the regularization strength increasing as λ increases. However, as λ becomes too large, a too strong regularization will make the system finally loose the ability of adapting input data.

23.5 An Information Transfer Perspective

In the past decade, extensive studies have been made on the minimum description length (MDL) [23.52, 23.54]. Sharing the common sprit of the minimum message length (MML) [23.69, 23.71], the BIC model selection criterion and variants [23.59, 23.48], and the celebrated Kolmogorov complexity [23.29], the key idea is to implement the well known Ockham's principle of economy to code a set of samples $\{x_t\}_{t=1}^N$ for being transferred from a sender to a receiver via a two part coding. One is the amount of bits for coding the residuals of using a parametric model $p(x|\theta)$ to fit a set of samples $\{x_t\}_{t=1}^N$. The second part is the amount of bits for coding the parameter set θ, provided that the function form of $p(x|\theta)$ has already known at the receiver and,

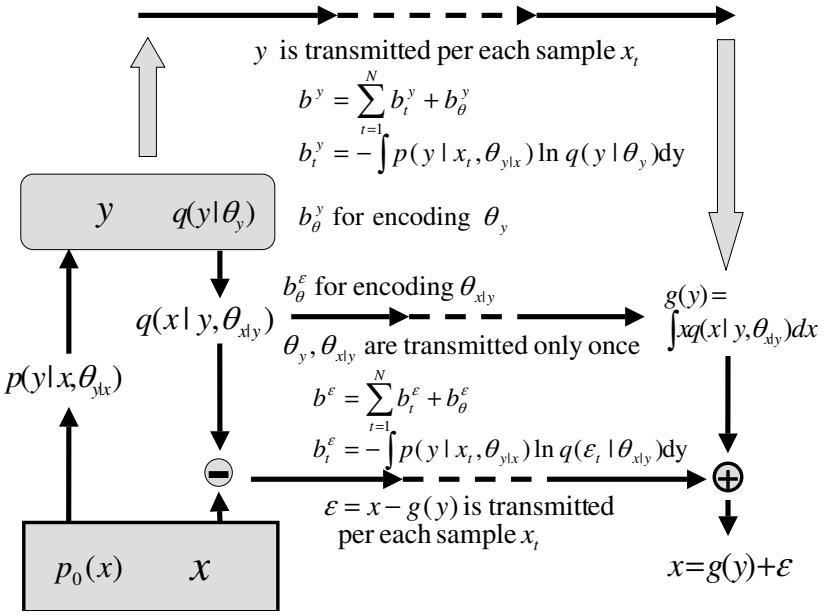


Fig. 23.3. BYY harmony learning from an Information-theoretic Perspective

thus, no need for being encoded. A best information transfer is reached when the bits for both the parts are minimized.

In the existing literature, given a density model $p(x|\theta)$ for a d dimensional real random vector x, the amount of bits per sample x_t to be transmitted is described by $b_t^\varepsilon = -\ln p(x_t|\theta) - d \ln \delta$, where $\delta > 0$ is a pre-specified constant resolution and usually ignored. The total amount of bits for the first part is $b^\varepsilon = \sum_{t=1}^{N} b_t^\varepsilon$. The amount b_θ^ε of bits for the second part is common to every sample of x_t, and, thus, only needs to be transmitted one time in advance. Thus, the average amount of bits to be transmitted is $\frac{1}{N}\sum_{t=1}^{N} b_t^\varepsilon + \frac{b_\theta^\varepsilon}{N}$. For a large size N of samples, the second term becomes very small and can be ignored. The minimization of the first term is actually equivalent to the ML learning. However, this term does not contain enough information to select an appropriate complexity (e.g., the number of parameters in θ) for $p(x|\theta)$.

In a contrary, for a finite size N of samples, we encounter a so-called over-fitting effect that the larger the complexity is, the smaller the residual of using $p(x|\theta)$ to fit the set $\{x_t\}_{t=1}^N$ is, and, thus, the smaller of the first term is. The second term takes its role that balances off the over-fitting effect since b_θ^ε increases as the complexity increases. However, b_θ^ε is described by $-\ln p(\theta)$. The priori distribution $p(\theta)$ is usually not available and can only be very roughly estimated, e.g., by a non-informative uniform prior or Jeffery priori [23.42, 23.37]. Instead of coding x_t directly for transmission, the MDL implementation with a bits-back strategy in [23.33, 23.32] maps x to y and

then code y for transmission. However, as to be further discussed in the next subsection, this bits-back based MDL is actually equivalent to the ML learning and, thus, is still not good for model selection.

The BYY harmony learning by Eq. (23.16) can also be understood from an information transfer perspective, with a new insight on its ability for model selection and regularization. As shown in Fig. 23.3, we consider a system in which x is mapped to an inner representation y that is encoded and sent to the receiver, and the receiver then decodes y to reconstruct x. Learning is made to obtain the encoder $p(y|x)$ for getting y from x, the distribution $q(y)$ for the codes on y, and the decoder $q(x|y)$ for getting x from y.

Provided that the function form of $q(y|\theta_y)$ is already known at the receiver, the average amount of bits to be transmitted is $\frac{1}{N}\sum_{t=1}^{N} b_t^y + \frac{b_\theta^y}{N}$, with b_t^y being the amount of bits per sample for coding y and b_θ^y being the amount of bits for coding θ_y. To reconstruct x_t, one also needs the decoder $q(x|y)$ that should also be coded at the sender and then sent to the receiver. The decoder is also coded in two parts. One is coding the residual between the original x_t and its reconstruction by the decoder, and the amount of bits per sample is b_t^ε. The other part is the amount b_θ^ε of bits to code the parameter set $\theta_{x|y}$ of $q(x|y)$. Provided that the function form of $q(x|y)$ is already known at the receiver, the average amount of bits for $q(x|y)$ is $\frac{1}{N}\sum_{t=1}^{N} b_t^\varepsilon + \frac{b_\theta^\varepsilon}{N}$.

As a result, referred to [23.75], the entire amount of bits is $N[\frac{b_\theta^y + b_\theta^\varepsilon}{N} - H(\theta, m)]$, with $H(\theta, m)$ given in Eq. (23.16). That is, the BYY harmony learning by Eq. (23.16) attempts to maximizing the information transfer in a sense of minimizing the total coding bits after approximately ignoring $b_\theta^y + b_\theta^\varepsilon$.

Being different from the above discussed conventional MDL that degenerates back to he ML learning after discarding the bits b_θ/N, the harmony measure $-H(\theta, m)$ by Eq. (23.16) without b_θ^y/N and b_θ^ε/N will not disable the model selection ability. The role of b_θ has now been jointly shared by the bits b^y for encoding the inner representation y of x and the bits $b_\theta^y + b_\theta^\varepsilon$ as a counterpart of b_θ. Not only carrying the information about x, the bits b^y also encode the scales of representation that either indicates model complexity directly or includes the core part of model complexity. This provides an alternative insight on why the BYY harmony learning can make model selection.

The above difference also leads to an important difference in implementing model selection. To avoid an inappropriately chosen $q(\theta)$ to deteriorate learning considerably, only a non-informative uniform prior is used as $q(\theta)$ in MDL and thus has no effect on parameter learning for determining θ, which is still made by a ML learning as the first step. The MDL criterion comes in effect at the second step for model selection. This two step implementation costs heavily since parameters learning on getting θ has to be made on all the candidate models in consideration. By the BYY harmony learning, the job of model selection is also performed via a family of densities $q(y|\theta_y)$ with a given parametric structure but unknown parameters θ_y that is determined

during learning process, which is a significant relaxation from solely relying on a priori density $q(\theta)$. As a result, not only parameter learning is performed more accurately but also model selection is made via the scale parameters of y that are determined automatically during learning parameters in θ_y.

The regularization role of Z_q in $-H(\theta, m)$ by Eq. (23.16) can also be understood from a more precise perspective of information transfer. Instead of considering a quantization by a pre-specified constant resolution $\delta > 0$ that is currently widely adopted in the MDL literature.

23.6 BYY Harmony Learning Versus Related Approaches

23.6.1 Relation and Difference to the Bits-Back Based MDL and Bayesian Approaches

The above information transfer perspective shares certain common part with the bits-back based MDL proposed in [23.33, 23.32]. However, there are two key differences.

First, the term Z_q replaces the role of a pre-fixed quantization resolution δ that is currently widely adopted in the MDL literature. Without considering what type of data distribution it is, manually setting a constant δ is simply because there is no a better solution available but it is clearly not a good solution. In the BYY harmony learning by Eq. (23.16), the term Z_q provides a better solution. In the data smoothing implementation, Z_q takes the input data distribution in consideration via the Parzen window estimator by Eq. (23.3) with a smoothing parameter h. This h takes a role similar to a quantization resolution δ, but now it is also learned to adapt the set of samples $\{x_t\}_{t=1}^N$. In the normalization implementation, Z_q takes the input data distribution in consideration indirectly via the learned parametric densities $q(x|y)$ and $q(y)$ as well as their values on the a set of samples $\{x_t\}_{t=1}^N$.

Second, an even fundamental difference is that BYY harmony learning does not adopt the bits-back strategy [23.33, 23.32]. Considering the dependence among the inner codes generated by $p(y|x)$, it has been argued in [23.33, 23.32] that the total amount of bits should be subtracted by the following amount of bits

$$H(\theta_{y|x}) = \int p(y|x)p(x)\ln p(y|x)\mu(dx)\mu(dy). \tag{23.52}$$

With this amount claimed back, the total amount of bits that has been considered by [23.33, 23.32] is actually equivalent to the Kullback divergence $KL(\theta)$ by Eq. (23.22), after discarding a term $H_x = \int p(x)\ln p(x)dx$ that is irrelevant to learning when $p(x) = p_0(x)$ by Eq. (23.2). In other words, the bits-back based MDL [23.33, 23.32] actually provides an interpretation to the Kullback learning by Eq. (23.22) from a information transfer perspective. In contrast, without including $H(\theta_{y|x})$ by Eq. (23.52), the discussion

in Sect. 23.5 provides an interpretation to the BYY harmony learning by Eq. (23.16). As to be further discussed in the next subsection, the Kullback learning by Eq. (23.22) is equivalent to implementing parameter learning under the ML principle or its certain regularized variants in lack of model selection ability, while BYY harmony learning provides a new mechanism that makes model selection either after or during parameter learning.

An insight can also be obtained by further observing the role of the bits-back amount $-H(\theta_{y|x})$ by Eq. (23.52). With the dimension of y fixed, the Kullback learning by Eq. (23.22) implements a stochastic encoding by $p(y|x)$ that allows certain dependence among the resulted codes. This dependence generates a redundant amount $-H(\theta_{y|x})$ of bits that is suggested in [23.33, 23.32] to be subtracted from computing the total amount of bits. In a contrast, aiming at seeking an appropriate dimension for y, the BYY harmony learning by Eq. (23.16) actually minimizes [23.75]

$$-H(\theta, k) = KL(\theta) - H(\theta_{y|x}) + C_y \ . \tag{23.53}$$

Where $-H(\theta_{y|x}) + C_y \geq 0$. That is, $-H(\theta, k) \geq KL(\theta)$ is an upper bound of the total bits considered in [23.33, 23.32].

When $p(y|x)$ is free, $\max_{p(y|x)} H(p\|q)$ results in $p(y|x)$ as in Eq. (23.19). It happens similarly when $p(y|x)$ is parametric either directly in a form of $\delta(y - y(x))$ or tends to be pushed into this form via $\max_{p(y|x)} H(p\|q)$. In these cases, $-H(\theta_{y|x}) + C_y$ reaches its minimum value 0. Thus, the BYY harmony learning achieves the minimum total number of bits instead of one upper bound.

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

Though such a redundant coding makes information transfer more reliable, allowing redundancy in the representation space of y already means that this representation space is not in its minimum complexity.

Furthermore, the BYY harmony learning may also be related to Bayesian approaches by replacing y with a parameter set θ in that Eq. (23.19) becomes equivalent to the Bayesian learning by Eq. (23.8). Ignoring $-\ln z_q$, $H(\theta, m)$ by Eq. (23.16) is actually the MML description length by Eq. (23.11), while $-\ln z_q \neq 0$ provides a type of regularization similar to that discussed in

Sect. 23.4.1. Also, Eq. (23.13) becomes equivalent to the evidence given by Eq. (23.9) and it follows from Eq. (23.24) that $KL(\theta)$ becomes equivalent to the description length based on this evidence. The difference between the MML description length and the evidence based description length is actually the bits-back part by Eq. (23.52) with y replaced by θ. As discussed in Sect. 23.2.2, knowing a priori $q(\theta)$ is a difficult task and a rough estimate $q(\theta)$ may seriously affect the MAP solution by Eq. (23.8). Thus, the description length based on Eq. (23.9) is usually regarded as an improvement over that by Eq. (23.11) since the integral over θ can regularize in a certain extent the discrepancy caused by $q(\theta)$.

However, the BYY harmony learning is different from the above MML description length and the evidence based description length in that an inner representation y takes the place of θ to avoid the difficulty of getting $q(\theta)$, which brings us the following advantages:

- Instead of specifying a density $q(\theta)$, the BYY harmony learning only needs to specify a family of densities $q(y|\theta)$ with a given parametric structure but unknown parameters θ, while learning further specifies one among the family. Therefore, the difficulty of requiring a detailed priori knowledge has been relaxed significantly. Moreover, the above superiority of the evidence based description length due to the bits-back type regularization disappears. On a contrary, as discussed in the early part of this subsection, the bits-back type regularization actually weaken the model selection ability and lost the nature of automatic model selection.
- Instead of considering all the parameters in the description length, the BYY harmony learning focuses only at those useful scale parameters m, k, etc., via the structures of the inner representation space of y which avoids to handle the difficulty of and saves the computing costs on estimating those complexities that are unnecessary for determining m, k, etc.
- As discussed in above, the BYY harmony learning is able to make model selection automatically during learning parameters. In contrast, using the evidence based description length for model selection has to be made via a two stage implementation since the evidence based description length has to be estimated after parameter learning.

23.6.2 Relations to Information Geometry, Helmholtz Machine and Variational Approximation

The minimization of $KL(\theta)$ by Eq. (23.22) with respect to a free $p(y|x)$ will result in Eq. (23.24) and becomes equivalent to the ML learning on $q(x)$ when $p(x) = p_0(x)$ by Eq. (23.2) [23.109]. This case relates to the information geometry theory (IGT) [23.16, 23.4, 23.5] that is also equivalent to the ML learning on $q(x)$ by Eq. (23.13). Moreover, the well known EM algorithm [23.20, 23.51, 23.47] is reached by the em algorithm obtained in IGT.

Making parameter learning by Eq. (23.22) also relates to the Helmholtz machine learning (HML) when $p(x) = p_0(x)$ is given by Eq. (23.2) and both $p(y|x)$ and $q(x|y)$ are both given by the conditional independent densities by Eq. (23.54) as used in [23.31, 23.18]. That is, the densities are given with the following format

$$p(u|v) = \prod_{j=1}^{m} \pi_j(v)^{u^{(j)}} (1 - \pi_j(v))^{1-u^{(j)}}, \tag{23.54}$$
$$\pi(v) = [\pi_1(v), \cdots, \pi_m(v)]^T = S(Wv + c),$$
$$S(y) = [s(y^{(1)}), \cdots, s(y^{(m)})]^T,\ 0 \le s(r) \le 1 \text{ is a sigmoid function},$$

where u is a binary vector. In this case, making parameter learning by Eq. (23.22) actually becomes equivalent to an one layer HML. Also, the well known wake-sleep algorithm for HML can be regarded as a simplified adaptive form of Eq. (23.23). With a general insight via Eq. (23.23), other specific algorithms for implementing the HML may also be developed.

It is also deserve to notice that making parameter learning by Eq. (23.22) with a parametric $p(y|x) \in \mathcal{P}_{y|x}(\theta_{y|x})$ is different from that a free $p(y|x) \in \mathcal{P}^0_{y|x}$ in that a parametric family $\mathcal{P}_{y|x}(\theta_{y|x})$ is a subset of the family $\mathcal{P}^0_{y|x}$ that consists of all the density functions in the form $p(y|x)$. Thus, we always have $\min_{p(y|x)\in\mathcal{P}_{y|x}(\theta_{y|x})} KL \ge \min_{p(y|x)\in\mathcal{P}^0_{y|x}} KL$. When $p(x) = p_0(x)$ is given by Eq. (23.2), it follows from Eq. (23.24) that the latter becomes equivalent to the ML learning on $q(x)$ by Eq. (23.13). In other words, making parameter learning by Eq. (23.22) with a parametric $p(y|x)$ actually implements a type of constrained ML learning on $q(x)$, which is also called a variational approximation to the ML learning on $q(x)$ [23.58, 23.56].

The BYY harmony learning is different from three existing approaches as follows. First, the BYY harmony learning minimizes the harmony measure $-H(p\|q)$ instead of the Kullback divergence $KL(p\|q)$ in Eq. (23.22), not only for parametric learning but also for model selection. Even using the Kullback learning by Eq. (23.22) for parameter learning, it is still followed by model selection via Eq. (23.21). In contrast, parameter learning via minimizing the Kullback divergence is the only target in IGT, HML, and variational approximation, while the issues of regularization and model selection are out of the scope of their studies.

Second, as discussed later in Eq. (23.66), the harmony learning may also be regarded as implementing a type of constrained ML learning, especially when $p(y|x) \in \mathcal{P}_{y|x}(\theta_{y|x})$ is parametric. However, it is different from the above discussed constrained ML learning via variational approximation [23.57, 23.56]. An additional constraint should be imposed on both types of learning to make them become equivalent.

Third, even focusing on the common part, i.e., parameter learning via minimizing Kullback divergence for implementing parameter learning, these studies are conducted from different perspectives with different purposes.

IGT studies the general properties possessed by Eq. (23.22) and alternative minimization for two general p and q from the perspectives of geometry structure [23.16] and differential geometry structure [23.4, 23.5]. HML and variational approximation consider developing efficient algorithms for implementing empirical parameter learning on a forward-backward net via an approximation of the ML learning on the marginal density $q(x)$ in Eq. (23.13). In contrast, the BYY learning studies two distributions in the two complementary Bayesian representations in Eq. (23.12) by systematically investigating not only three typical architectures for different learning tasks, but also regularization by either a conscience de-learning type via normalization or a Tikhonov-type via data smoothing with its smoothing parameter h estimated in sample way. While IGT, HML, and variational approximation have neither explicitly and systematically considered the two complementary representations in Eq. (23.12) nor the regularization of two such types.

23.6.3 A Projection Geometry Perspective

Projection Geometry in Vector Space. Through obtaining a quasi Pythagorean relation under the Kullback divergence

$$KL(p\|q) = \int p(u) \ln \frac{p(u)}{q(u)} du \geq 0, \; KL(p\|q) = 0, iff \; p(u) = q(u) \quad (23.55)$$

This divergence based learning has been further theoretically studied from the perspective of ordinary geometry and differential geometry under the name of information geometry [23.16, 23.4, 23.5]. Actually, neither the harmony measure by Eq. (23.17) nor the Kullback divergence by Eq. (23.55) satisfies all the properties of the conventional metric measure. Moreover, the harmony measure by Eq. (23.17) even does not satisfies a quasi Pythagorean relation that the Kullback divergence satisfies. In this section, we suggest to investigate both the harmony measure based learning and the Kullback divergence based learning by lowering down from a metric level to an even basic level, namely, a level of projection geometry.

We start at reviewing some basic properties in the conventional vector space R^d. We denote $U_c = \{u : u \in R^d \; and \; \|u\|^2 = c^2, \text{ for a constant } c > 0\}$, which is a sphere shell with the radius c.

As shown in Fig. 23.4, for $u = ce^{\theta_u} \in U_c, v = c'e^{\theta_v} \in U_{c'}$, their inner product is

$$u^T v = cc' \cos(\theta_v - \theta_u), \quad (23.56)$$

which is symmetric to v and u and leads to a norm $\|u\|^2$ that further leads to the metric $\|u - v\|$.

Imposing the constraint $\|u\| = 1$, the inner product returns back to the projection of v on u as follows:

$$\Pi_u^v = c' \cos(\theta_v - \theta_u), \quad (23.57)$$

which has the following properties:

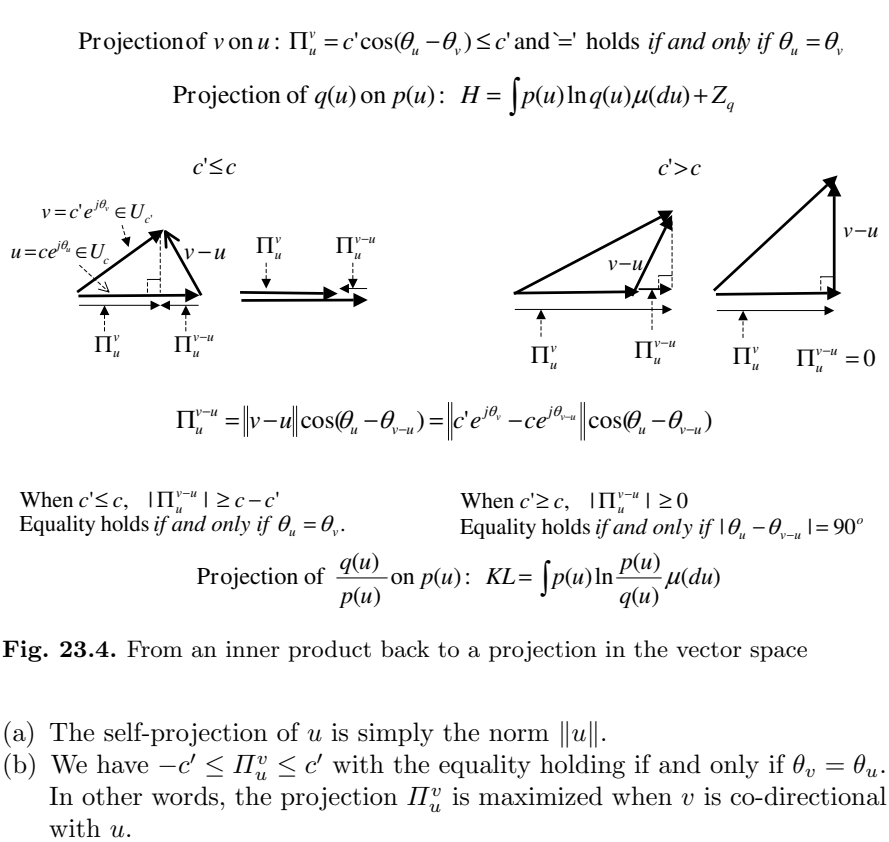


Fig. 23.4. From an inner product back to a projection in the vector space

(a) The self-projection of u is simply the norm $\|u\|$.
(b) We have $-c' \le \Pi_u^v \le c'$ with the equality holding if and only if $\theta_v = \theta_u$. In other words, the projection Π_u^v is maximized when v is co-directional with u.
(c) The projection Π_u^v reaches its minimum 0 when $\theta_v - \theta_u = 0.5\pi$, which is said that v is orthogonal to u.
(d) When $c = c'$, $\theta_v = \theta_u$ implies $v = u$. That is, the maximal projection is equivalent to the equality $v = u$, when v, u are on the same shell U_c.
(e) $\theta_v = \theta_u$ can be achieved by rotating the directions of both v and u or the direction of either v or u. That is, the projection $v^T u$ has the symmetry property.

The error or residual $v - u = \|v - u\|e^{-\theta_{v-u}}$ also has a projection on u:

$$\Pi_u^{v-u} = \|v - u\|\cos(\theta_{v-u} - \theta_u), \tag{23.58}$$

with the following properties:

(f) As shown in Fig. 23.4, when $c' > c$, this residual projection $|\Pi_u^{v-u}|$ reaches its minimum 0 when $\theta_{v-u} - \theta_u = 0.5\pi$, with $\|v - u\| \neq 0$. In this case, the residual $v - u$ is said to be orthogonal to u, where the norm of u and the projection v on u becomes the same, i.e., $\Pi_u^v = c$ or $\cos(\theta_v - \theta_u) = c/c'$.

(g) When $\|v\| \leq \|u\|$, this residual projection $|\Pi_u^{v-u}|$ reaches its minimum $c-c'$ if and only if $\theta_v = \theta_u$. When $c = c'$, we have $u = v$ and the minimum value is 0.

In a summary, we have

> *When v, u locate on a same shell U_c, the concepts of maximizing the projection v to u, minimizing the residual projection $(v-u)$ to u, of making residual $v-u$ being orthogonal to u, and the equality $v = u$ are all the same thing.* (23.59)

Projection Geometry in a Functional Space. In an analogy, we consider a functional space

$$\mathcal{Q} = \{q(u) : q(u) \geq 0 \text{ and } \textstyle\int q(u)\mu(du) < \infty\}, \tag{23.60}$$

where $u \in S_u \subseteq R^d$ and μ is a given measure on the support S_u, and $\mu(du)$ only relates to du but to neither u nor $q(u)$. A useful subspace $\mathcal{P}_c \subset \mathcal{Q}$ is

$$\mathcal{P}_c = \{p(u) : p(u) \geq 0,\ \textstyle\int p(u)\mu(du) = c, \text{ for a constant } c > 0\}. \tag{23.61}$$

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

Given $p(u) \in \mathcal{P}_c, q(u) \in \mathcal{P}_{c'}$, we define the projection of $q(u)$ on $p(u)$ by

$$\begin{aligned} H(p\|q) &= \textstyle\int p(u)\mu(du) \ln (q(u)\mu(du)) = \int p(u) \ln q(u)\mu(du) - Z_q, \\ Z_q &= -\textstyle\int p(u) \ln \mu(du)\mu(du) = -\ln \mu(du), \end{aligned} \tag{23.62}$$

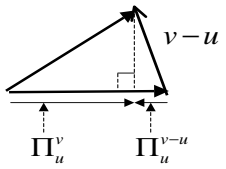
which can be regarded as the counterpart of Eq. (23.57) as shown in Fig. 23.4. It can be observed that Eq. (23.62) becomes the same as Eq. (23.16) and Z_q takes the same role as $-\ln z_q$, when

$$p(u) = p(x, y) = p(y|x)p(x), q(u) = q(x, y) = q(x|y)q(y). \tag{23.63}$$

Considering $\int p(u)\mu(du) = 1$ and $p(u) = p_0(u)$ is the empirical density by Eq. (23.3) when $h_u = 0$, it follows that $\mu(du) = 1/\sum_{t=1}^N p(u_t)$ and $z_q = \sum_{t=1}^N q(u_t)$ from which and Eq. (23.63), we also get Eq. (23.36).

In correspondence to Eq. (23.57), we have the following properties:

(1) The self-projection of $p(u)$ is $H(p\|p) = \int p(u)\mu(du) \ln [p(u)\mu(du)]$, which can be regarded as a type of norm of p and it becomes the negative entropy of the probability distribution $p(u)\mu(du)$ when $p(u) \in \mathcal{P}_1$ is a density.
(2) $H(p\|q)$ is maximized if and only if $q(u) = \frac{c'}{c}p(u)$, i.e., $q(u)$ has the same shape as $p(u)$, because we have $\int \hat{p}(u) \ln \hat{q}(u)\mu(du) \leq \int \hat{p}(u) \ln \hat{p}(u)\mu(du)$ with $c\hat{p}(u) = p(u)$, $c'\hat{q}(u) = q(u)$ and $\hat{p}(u), \hat{q}(u) \in \mathcal{P}_1$.
(3) When $c = c'$, $H(p\|q)$ is maximized if and only if $q(u) = p(u)$.
(4) When $p(u)$ is free to be any choice in $\mathcal{P}_c$, the maximization of $H(p\|q)$ will also let $p(u)$ to become $c\delta(u - u^*)$, where $u^* = arg\max_u\ q(u)$.



$c = 1, c' = 1$

$\Pi_u^v = \cos(\theta_u - \theta_v)$

$\Pi_u^v = \max$ *if and onl y if* $\theta_u = \theta_v$.

$\Leftrightarrow$

$\Pi_u^{v-u} = \|v-u\|\cos(\theta_u - \theta_{v-u})$
$= \left\|e^{j\theta_v} - e^{j\theta_{v-u}}\right\|\cos(\theta_u - \theta_{v-u})$

$\Pi_u^{v-u} = 0$ if and only if $\theta_u = \theta_v$.

$\int p(u)\mu(du) = 1, \int q(u)\mu(du) = 1$

$H = \int p(u)\ln q(u)\mu(du) + Z_q$

$KL = \int p(u)\ln\frac{p(u)\mu(du)}{q(u)\mu(du)}\mu(du)$
$= \int p(u)\ln\frac{p(u)}{q(u)}\mu(du)$

$\max_q \ H$ results in $q(u) = p(u)$
$\max_p \ H$ results in $p(u) = \delta(u - u^*)$ $\not\Leftrightarrow$
$\max_{p,q} H$ results in $q(u) = p(u) = \delta(u - u^*)$ $\not\Leftrightarrow$

$\min_q \ KL$ results in $q(u) = p(u)$
$\min_p \ KL$ results in $p(u) = q(u)$
$\min_{p,q} KL$ results in $q(u) = p(u)$

Fig. 23.5. Unit norm based projection: from the vector space to a functional space

In comparison with the situation of Eq. (23.57), there are three differences. One is that each density represents a point of infinite dimension. Second, each component is constrained to be nonnegative. Third, the constraint $\int p(u)\mu(du) = c$ is a first order linear constrained, instead of the quadratic constraint $\|u\|^2 = c^2$. These differences result in that the maximization of $H(p\|q)$ makes not only that $p(u)$ and $q(u)$ has a same shape in the sense $q(u) = \frac{c'}{c}p(u)$ but also that $p(u)$ prefers to have a simplest shape $c\delta(u - u^*)$. When $p(u)$ is free to be any choice in $\mathcal{P}_c$ and $q(u)$ is free to be any choice in $\mathcal{P}_{c'}$, the maximization of $H(p\|q)$ will finally let that both $p(u)$ and $q(u)$ become impulse functions. When $p(u) \in P, q(u) \in Q$ are constrained to be unable to become impulse functions, the maximization of $H(p\|q)$ will make that $p(u)$ and $q(u)$ become close in a shape of a least complexity but not able completely equal. Therefore, the maximization of $H(p\|q)$ on a BYY system Eq. (23.63) indeed implements the harmony principle given at the beginning of Sect. 23.3.1, while the maximization of the projection u to v only ensures u and v become co-directional but does not have such a least complexity.

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

Moreover, if we use $p(u) \in \mathcal{P}_c$ to represent $q(u) \in \mathcal{P}_{c'}$ and define the discrepancy or residual [1] by $p(u) \ominus q(u) = p(u)\mu(du)/[q(u)\mu(du)] = p(u)/q(u)$,

[1] Under this definition, $p(u) \ominus q(u)$ is generally not guaranteed to still remain in $\mathcal{Q}$. For a subset $\mathcal{Q}_q \subset \mathcal{Q}$ with $\mathcal{Q}_q = \{q(u) : q(u) \in Q, \int_{D_u} q^2(u)\mu(du) <$

we let $q(u)\mu(du)$ in Eq. (23.62) to be replaced by the residual in this representation and get that this residual projection on $p(u)$ as follows

$$R(p\|q) = \int p(u) \ln [p(u)/q(u)]\mu(du) = H(p\|p) - H(p\|q). \tag{23.64}$$

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

$$\begin{aligned} &R(p\|q) = c[KL(\hat{p}\|\hat{q}) + \ln \frac{c}{c'}], \\ &KL(\hat{p}\|\hat{q}) = \int \hat{p}(u) \ln [\hat{p}(u)/\hat{q}(u)]\mu(du). \end{aligned} \tag{23.65}$$

From which we can observe the following properties:

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

From the above discussions, we see that the concepts of maximizing $H(p\|q)$ and of minimizing the residual projection $R(p\|q)$ are related, but not equivalent. Even when $c = c' = 1$, we do not have the equivalence between Π_u^v and Π_u^{v-u} as given in Eq. (23.59) for Eq. (23.57) and Eq. (23.58). This provides a geometry perspective on why and how the maximization of $H(p\|q)$ on a BYY system Eq. (23.63), which is a generalization of maximizing the projection for the co-directionality, is different from the minimization of Kullback divergence $KL(p\|q)$ on a BYY system Eq. (23.63) or equivalently the maximum likelihood learning, which is a generalization of minimizing the residual projection. Moreover, the latter does not have the least complexity nature that enables the former to make model selection.

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

$$\begin{aligned} &\max_{p\in P, q\in Q,\ s.t.\ H(p\|p)=H_0} H(p\|q) \text{ is equivalent to} \\ &\min_{p\in P, q\in Q,\ s.t.\ H(p\|p)=H_0} KL(p\|q). \end{aligned} \tag{23.66}$$

$\infty, \int_{D_u} q^{-2}(u)\mu(du) < \infty, \int_{D_u} \mu(du) < \infty\}$, we can define the addition by $r(u) = p(u) \oplus q(u) = p(u)q(u)$ and have $r(u) \in \mathcal{Q}_q$. Also, we have the unit $1 = p(u)p^{-1}(u) \in \mathcal{Q}_q$ for $u \in S_u$ and the inverse $p^{-1}(u) = 1/p(u) \in \mathcal{Q}_q$.

In this case, it follows that the induced minus operation $p(u) \ominus q(u) = p(u)/q(u)$ is still in $\mathcal{Q}_q$. That is, we get $\mathcal{Q}_q$ as an Abel group. Moreover, on an appropriate subset $\mathcal{Q}_l$ we can further define the dot product $\alpha \circ p(u) = p(u)^\alpha \in \mathcal{Q}_l$ for $\alpha \in R$ and, thus, get $\mathcal{Q}_l$ as a linear functional space. Furthermore, we can introduce the geometrical concepts of the projection Eq. (23.62), the residual projection Eq. (23.64) and the corresponding orthogonality to $\mathcal{Q}_q, \mathcal{Q}_l$.

With $p(x)$ given by Eq. (23.3), the constraint $H(p\|p) = H_0$ means certain constraint imposed on $p(y|x)$. In other words, Eq. (23.66) happens on a class of BI-directional architectures, and can also be regarded as implementing a type of constrained ML learning, which is different from those of variational approximation [23.57, 23.56] that implements $\min_{p\in P, q\in Q} KL(p\|q)$ with $p(y|x)$ in a constrained structure but without requiring the constraint $H(p\|p) = H_0$.

In addition, the above discussions on the geometry properties of $p(u) \in \mathcal{P}_c$ and $q(u) \in \mathcal{P}_{c'}$ with $c \neq 1, c' \neq 1$ may also be extended beyond probability densities. Also, with $R(p\|q) = 0$ we can get the concept of the orthogonality of the residual $p(u) \ominus q(u)$ to $p(u)$.

23.7 Bibliographic Remarks

In the previous chapter of the present book, main results of using BYY system and harmony learning on typical learning problems have been summarized. Also, bibliographic remarks have been made on the progress of these studies from both the aspect of BYY system with the KL learning and the aspect of computing techniques for implementing BYY learning. In this section, further bibliographic remarks will be made on the progress from the model selection and regularization aspects of BYY harmony learning.

23.7.1 On BYY Harmony Learning (I): Model Selection Criteria vs. Automatic Model Selection

As discussed in Sect. 23.3.1, maximizing the harmony measure by Eq. (23.16) that makes model selection either automatically during parameter learning by Eq. (23.20) or via a selection criterion Eq. (23.21) after parameter learning.

In help of the so-called 'hard-cut' treatment of posteriori probabilities, this harmony measure with $z_q = 1$ was firstly obtained in 1995 both at its special case of both Gaussian mixture (see Eq. (20) and (22) in [23.109], Eq. (13) and (14) in [23.113], and Eq. (13) in [23.112]), and at a special case of finite mixture (see Eq. (7) in [23.112]). Companying with this measure, two types of detailed studies were conducted as follows:

- One is called two-phase style learning in Sect. 23.3.1. That is, model selection is made via $\min_k J(k)$ after parameter learning. This $J(k)$ is a simplified version of the harmony measure after discarding irrelevant terms. Typical examples include $J(k)$ by Eq. (24) in [23.109] and $J(k)$ by Eq. (13) in [23.111].
- The other type of studies is on parameter learning with automatic model selection by Eq. (23.16) in Sect. 23.3.1. It was suggested (see Sect. 5.2 and the footnote on page 986 in [23.109], also see Sect. 3 in [23.111] and the second part of Sect. 5 in [23.112]) that an appropriate value of k can be

automatically determined during parameter learning by the so-called hard-cut EM algorithm (see the algorithm after Eq. (20) in [23.109], also see the algorithm after Eq. (15) in [23.112]), via discarding a Gaussian component $\alpha_j G(x|m_j, \Sigma_j)$ if either or both of Σ_j and $\alpha_j = P(y = j)$ become zero.

It should also be noticed that studies of BYY system with the harmony measure based learning by Eq. (23.16) and the Kullback divergence based learning by Eq. (23.22) were conducted jointly, with the following relations found:

- The additive relationship $KL = -H - E_p + D$ by
$$KL(p\|q) = -H(p\|q) - E_p, E_p = -\int p(u) \ln p(u)\mu(du) + \ln z_p. \quad (23.67)$$
or equivalently $H = -KL - E_p + D$ by Eq. (23.47) that was firstly presented in [23.109], where D is a term that is only related to the smoothing parameter h for a $p(x)$ given by Eq. (23.3).
- The term D becomes irrelevant to learning when $h = 0$ or equivalently $p(x)$ is given by Eq. (23.2). In these cases, D can be discarded and we can simply consider $KL = -H - E_p$ or equivalently $H = -KL - E_p$.
- The inequality relation $KL \leq -H$ was also firstly observed in [23.112]. The equality $KL = -H$ holds when $E_p = 0$, where the KL learning and the harmony learning become equivalent as discussed in Sect. 23.4.2.
- As discussed in Sect. 23.6.1, the harmony learning is different from the KL learning in that the minimization of $-H = KL + E_p$ attempts to push $E_p \geq 0$ toward its minimum $E_p = 0$ such that a minimum coding length is reached via minimizing the model complexity.

23.7.2 On BYY Harmony Learning (II): Model Selection Criteria

After the initial results obtained in 1995 [23.109], various specific $J(k)$ forms of Eq. (23.21) have been subsequently obtained from Eq. (23.16) for model selection in typical learning models, with main progresses summarized as follows:

(1) Not only $J(k)$ by Eq. (24) in [23.109] and by Eq. (13) in [23.111] was further studied experimentally in 1996 [23.108] and both theoretically and experimentally in 1997 [23.104], but also Eq. (7) in [23.112] (i.e, the special form of Eq. (23.16) with $z_q = 1$ on a finite mixture) is reiterated via Eq. (10) in [23.107] and Eq. (18) in [23.104], and then applied to multi-sets mixture learning by Eq. (15) in [23.107]).
(2) Started from 1997, the harmony measure by Eq. (23.16) with $z_q = 1$ is further suggested under the notation $J_2(k)$ as a general criterion for model selection (see Eqs. (3.8) and (3.9) in [23.101], Eqs. (13) and (15) in [23.102], and Eq. (12) in [23.104]). Recently, the superiority of this criterion has been further supported by experiments made in comparison with classic criteria including AIC, CAIC and MDL [23.34].

(3) In 1998, extending the relation $-H = KL + E_p$, not only the weighted sum by Eq. (23.49) was firstly suggested (see Eq. (48) in [23.92]), but also its variant $KL+\lambda E_p$ was also suggested (see Eq. (8) in [23.90], Eq. (22) in [23.91], Eq. (7c) in [23.100], Eqs. (17) and (18) in [23.93], Eq. (6f) in [23.94], as well as Eq. (8b) in [23.96]). The form $KL + \lambda E_p$ returns to $-H = KL + E_p$ when $\lambda = 1$. This form makes it possible to be further extended with the function ln(.) replaced by a general convex function $f(.)$ (see Eq. (15) and (10) in [23.97] and Sect. 4 in [23.87], also Sect. II(B) in [23.84] and Sect. 2.5 in [23.85]).

(4) Also started from 1997, typical forms of the harmony measure by Eq. (23.16) with $z_q = 1$ and $p(x)$ given by Eq. (23.2) have also been developed as model selection criteria for the following learning models:

- PCA and FA (see Eq. (9.13) in [23.101], Eq. (56) in [23.92], Eq. (33) and (37) in [23.93], as well as Eq. (13) and (18b) in [23.100]).
- Principal ICA that extends ICA to noise situation (see Eq. (10.9) in [23.101], Eq. (11) in [23.89], Eq. (56) in [23.86] and Eq. (55) in [23.85]).
- Binary LMSER (see Eq. (8.10) and (8.13) in [23.101]).
- Logistic LMSER (see Eq. (38) in [23.76]).
- Regularized LMSER via minimizing the variances of hidden units (see Eq. (8.10) and (8.13) in [23.101], Eq. (40) and (41) in [23.93], Eq. (20b) and (20c) in [23.100]).
- Mixture of experts with an approximated criterion proposed firstly (see Table 6(5) in [23.103], Sect. 4.3(1)&(2) in [23.92]), and then a much improved version (see Eq. (84) in [23.83]).
- Alternative ME (see Table 6(5) in [23.103], Sect. 4.3(1)&(2) in [23.92]);
- RBF nets (see Table 7(4) in [23.103], Sect. 4.3(3) in [23.92]).
- Three layer networks, with not only some approximate criteria proposed for both binary stochastic hidden units [23.103, 23.94, 23.95] (e.g., see Eq. (56) in [23.92]) and deterministic real hidden units (see Eqn(88) & (89) in [23.86]), but also improved versions for binary stochastic hidden units (see Type (b) of Eq. (67) in [23.83], also see Eq. (47) in [23.76]), stochastic Gaussian hidden units (see the real y case of Eq. (9a) in [23.99]), and deterministic real hidden units (see Eqn(139) in [23.78]).
- Temporal factor analysis (see Eq. (23b) in [23.90], Eq. (49) in [23.82], the case (a) of Eq. (82) & Eq. (83) in [23.80]).
- Hidden Markov model (HMM) (see Eq. (34) in [23.90]).
- Independent HMM (see Eq. (51) in [23.82], Eq. (85) and (86) in [23.80]).
- Temporal extensions of binary LMSER (see Eq. (46) in [23.82], Eq. (93) in [23.80]).

(5) Started from 2001, the above criteria have been further extended via z-regularization with z_q as discussed in Sect. 23.4.1 and $p(x)$ by Eq. (23.3).

- In [23.83], we got criteria for model selection by Eq. (41) for Gaussian mixture and various special cases, by Eq. (67) for three layer networks, by

Eq. (84) for mixture-of-experts and alternative ME, as well as by Eq. (85) for RBF nets.
- In [23.82], we got criteria for model selection given by the cases (b) & (c) of Eq. (82) & (83) for temporal factor analysis.
- In [23.80], Table 3 provides a systematic summary of criteria, ranging from empirical learning to z-regularization for model selection on various Gaussian mixtures and Gaussian mixture of experts, and Table 4 provides a systematic summary of criteria, also ranging from empirical learning to z-regularization for model selection on various and non-Gaussian mixtures.
- In [23.76], criteria for model selection are also given by Eq. (33) on modular binary factor analyses, and by Eq. (48) on modular binary LMSER.

(6) The last but not least, the relation between the status of observation noise and model selection has been elaborated. For a backward model $x = Ay$ with no noise, the dimension m of y can be determined via the rank of the covariance matrix x. For a forward model $y = Wx$, the dimension m of y is actually pre-given instead of being decided by model selection. In other words, model selection are necessary only for a B-architecture and a BI-architecture, where an observation noise is considered via its backward or generative path (see pp841-843 of [23.82] and pp 1148-1149 of [23.80]).

23.7.3 On BYY Harmony Learning (III): Automatic Model Selection

As discussed in Sect. 23.7.1, also started from 1995 on Gaussian mixture, an appropriate value of k is automatically determined via discarding a Gaussian component $\alpha_j G(x|m_j, \Sigma_j)$ if either or both of Σ_j and $\alpha_j = P(y = j)$ become zero, during implementing a hard-cut EM algorithm for the maximization of harmony measure by Eq. (23.16) [23.109, 23.111, 23.112]. Main progresses along this direction are summarized as follows:
(1) This hard-cut EM algorithm based automatic model selection was not only experimentally demonstrated [23.108] but also further extended to learning on alternative mixture of experts (see Sect. 3.3 in [23.107]).
(2) An adaptive version of this hard-cut EM algorithm was linked to the winner-take-all (WTA) competitive learning. An adaptive version of the EM algorithm was also heuristically proposed and shown to demonstrate a type of rival penalized competitive learning (RPCL) [23.114] mechanism (see Sect. 6 in [23.109]).
(3) Not only adaptive version of the hard-cut EM algorithm is used in implementing learning, but also the rival penalized competitive learning (RPCL) [23.114] is used in place of the hard-cut EM algorithm such that the advantage of RPCL on learning with automatic model selection is adopted. Moreover, the original RPCL learning has been further extended into two types of general forms (i.e., TYPE A and Type B) for learning on Gaussian mixture, multisets modeling (local PCA, local MCA, local subspace, etc.),

mixture of experts, and RBF net (see Sect. 5 in [23.106], Sect. 4.3 in [23.107], and [23.98]).

(4) Started from 1999, the general form of using the harmony measure for both parameter learning and model selection, as shown in Eq. (23.16), has been studied (see Eq. (11) and (12) in [23.86] and [23.85]). Moreover, making parameter learning with automatic model selection by Eq. (23.20) was further made systematically in 2001 [23.83]. Not only the role of the least complexity nature Eq. (23.18) in model selection has been understood, but also the side-effect of the WTA competition by Eq. (23.19), i.e., making the maximization of Eq. (23.20) easy to be trapped at local maximums, is tackled by introducing certain regularization (see Sect. 2.5 in [23.83]). Moreover, four types of regularization have been proposed, as summarized into the following two groups:

(a) Harmony measure + regularization term, that is, a regularization is introduced additively. Specifically, the regularization term can be one of the following three choices:
   - The normalization term as discussed in Sect. 23.4.1 was proposed as regularization term (see the second part on page 52 in [23.83]). It was first time revealed that the harmony learning by Eq. (23.20) with normalization regularization acts as a general RPCL learning framework that implements a floating RPCL learning mechanism (see Sect. 3.2 in [23.83]), which not only justifies the heuristically proposed RPCL learning from the BYY harmony learning perspective but also provides a guide for automatically controlling the ratio of learning and de-learning that was a difficult task in the original RPCL learning [23.114].
   - The normalization term is λE_p, that is, we have $H + \lambda E_p$ that returns to the harmony measure H alone when $\lambda = 0$ and becomes $-KL = H + E_p$ when $\lambda = 1$ (see Eq. (42) and (43) in [23.83]). We can simply choose one appropriate value for λ or let λ to decrease from 1 to 0 gradually in a simulated annealing way.

(b) A regularization can also be introduced in a non-additive way. As will be further discussed in the next subsection, we have two typical techniques as follows (see Sect. 3.2 in [23.83]):
   - regularization is structural and introduced via a BI-architecture,
   - regularization is introduced via data smoothing.

(5) The above four types of combining the roles of the harmony measure and regularization can also be understood from the perspective of competitive learning [23.79]. The nature by Eq. (23.18) encourages a WTA competition by Eq. (23.19), while each of them acts in different manners. Data smoothing penalizes the winner, while both the λE_p and the structural regularization penalize the winner but compensates other participants. However, all these competition-penalty mechanisms makes the WTA effect weaken but encourage gain diversification among participants in competition.

(6) Similarly, the detailed forms of the two groups were also proposed for implementing the harmony learning Eq. (23.20) on mixture of experts, alternative ME, RBF nets, three layer net, as well as SVM type kernel regression (see Sect. 4 in [23.83]).

In the past two years, BYY harmony learning on various Gaussian/ non-Gaussian mixture and mixture-of-experts as well as modular networks with one hidden layer have been systematically studied in [23.80] and [23.76], respectively, with the following main results:

- A systematic summary and further elaboration of BYY harmony learning and RPCL learning on various details of Gaussian mixture and Gaussian mixture of experts, including MSE clustering, elliptic clustering, subspace clustering, alternative ME, RBF nets with automatic model selection (see Sect. 3 in [23.80]).
- BYY harmony learning algorithms for learning with automatic hidden factor determination on modular binary FA, local LMSER, competitive ICA (see Sect. 4 in [23.80] and Sect. 4 in [23.76]), as well as on three layer networks (see Sect. 5 in [23.76]).
- Extension of the harmony learning by Eq. (23.20) to the so-called f-harmony learning (see Sect. 2.3.2 in [23.80]).

23.7.4 On Regularization Methods

Several regularization methods have also been developed during the studies on BYY learning. Not only each of them can improve the learning performances on a BYY system in the case of a small size of samples, but also some of the methods remain useful even being independent of BYY system. Main results are summarized as follows:

Data smoothing regularization, which came from replacing the empirical density $p_0(x)$ by Eq. (23.2), that is equivalent to directly use a set of training samples, via a Parzen window density $p_h(x)$ by Eq. (23.3) with a smoothing parameter $h > 0$. The idea started from suggesting the use of $p_h(x)$ by Eq. (23.3) in a BYY system (see Eq. (5) in [23.109] and Eq. (1) in [23.111], also see Sect. 1 in [23.106] and Eq. (1) in [23.107]). In 1997, it was further proposed under the name of data smoothing (see Eq. (16) in [23.107] and Eq. (3.10) in [23.101]) that an appropriate h is also learned via implementing the KL learning by Eq. (23.22), which becomes equivalent to

$$\min_{\theta,h} KL(\theta,h), \quad KL(\theta,h) = \int p_h(x) \ln \frac{p_h(x)}{q(x|\theta)} \mu(dx), \tag{23.68}$$

which was firstly presented by Eq. (7) in [23.103]. In a BYY system, $q(x|\theta) = \int q(x|y)q(y)dy$ is the marginal density represented by the Ying machine. Generally, being independent of BYY system, $q(x|\theta)$ can be any parametric model for density estimation. Also in [23.102], the data smoothing regularization is suggested on $q(z|x,\theta_{z|x})$ for supervised learning of three layer forward net and mixture of experts.

Data smoothing introduces a Tikhonov-type regularization [23.10] into parameter learning, with the role h^2 being equivalent to the hyper-parameter in a Tikhonov regularization. What is new here is that an appropriate h can be learned via an easy implementation. Several advances have been made on implementing data smoothing since 1997, including

- A smoothed EM algorithm from learning on Gaussian mixture (see Eq. (18) in [23.102]).
- Three techniques for computing the integral $\int G(x|x_t, h^2 I)F(x)\mu(dx)$, namely stochastic approximation and mean-field [23.94] as well as the following second order approximation (see Sect. 2.4 in [23.88] and Sect. 2.3 in [23.86]):

$$\int G(x|x_t, h^2 I)F(x)\mu(dx) \approx F(x_t) + 0.5h^2 Tr[H_F],$$
$$\text{with the Hessian matrix } H_F = \frac{\partial^2 F(x)}{\partial x \partial x^T}]_{x=x_t}. \tag{23.69}$$

- Four approaches for solving h, i.e., quantization based enumeration, stochastic approximation, iterative updating, and solving a second order algebraic equation [23.94, 23.86, 23.85, 23.82, 23.80].
- In independent factor model of non-Gaussian real factors, mixture of experts, alternative ME, RBF nets, and three layer networks, different smoothing parameters are provided for input data, output data, and inner representation, respectively [23.94, 23.86, 23.85, 23.82, 23.80, 23.76].
- Two types of data smoothing mechanisms are provided, with one for the KL learning and the other for the harmony learning (see Sect. II(A) & (B) in [23.82] and Sect. 2 in [23.80]).

Details are further referred to [23.85, 23.82, 23.80] as well as a recent summary given in [23.77].

Normalization regularization, which came from the normalization term z_q. Firstly proposed in [23.82, 23.83], this normalization term causes a conscience de-learning that not only introduces a regularization to the ML learning, but also makes BYY harmony learning behave similar to the RPCL learning [23.114]. The details of the normalization role and its implementation on Gaussian mixture can be found in Sect. 3.2 of [23.83] and Sect. II(E) of [23.82]. Further results on Gaussian mixture, Gaussian mixture of experts, non-Gaussian mixture of experts, as well as modular networks with one hidden layer of binary units can be found in [23.76] and [23.78].

Structural regularization, which happens in a BYY system where certain regularization to a B-architecture or a F-architecture is imposed via its free part being replaced with an appropriately chosen parametric model. This was firstly suggested in 1997 (see Item 3.4 in [23.101], also see Item 2.5 and Item 2.6 in [23.102]). Typical examples are $p(y|x)$ given by Eq. (23.24) for a B-architecture and $p(x|y)$ given by Eq. (23.46). For example,

- It was suggested (see Sect. 2.5 in [23.83]) that the local maximum side-effect of the WTA competition by Eq. (23.19) with a B-architecture can be

regularized with an appropriate parametric $p(y|x)$ by Eq. (23.24). Recently it have been experimentally shown in [23.43] that such a regularization makes BYY harmony learning on Gaussian mixture also demonstrate a RPCL mechanism with automatic selection on k.
- The previously discussed principal ICA that extends ICA to noise situation (see Eq. (10.9) in [23.101], Eq. (11) in [23.89]) can also be regarded as that an ICA $y = Wx$ is regularized by $G(x|Wy, \sigma^2 I)$.
- In comparison with the above first two types of regularization, one major advantage of structural regularization is easy to be implemented via an adaptive algorithm. However, we can not avoid computational difficulty of the integral in $p(y|x)$ by Eq. (23.24) when y is real and non-Gaussian. Moreover, choosing a parametric model instead of $p(y|x)$ by Eq. (23.24) is not easy if there is not enough a priori knowledge.

Annealing Procedure As discussed in Sect. 23.4.2, the KL learning can be regarded as a regularized version of the HL learning. The advantage of two can be combined by Eq. (23.49) with the regularization strength gradually decreasing as λ decreases in a simulated annealing procedure. As discussed in the previous subsection, the local maximum side-effect of the WTA competition by Eq. (23.19) can be solved via such a simulated annealing (see Eq. (42) and (43) in [23.83]), which has been further supported by experiments on Gaussian mixture [23.44].

f-function regularization can also be imposed with $\ln(r)$ replaced by a convex function f, which has also been supported by experimental demonstrations on Gaussian mixture [23.104] and ICA problems [23.105]. Readers are referred to a detailed introduction provided in the previous chapter in this same book.

23.8 Conclusions

Efforts of making learning on a finite size of samples have been discussed in three typical streams. BYY harmony learning provides a new mechanisms for model selection and regularization, which has been further justified from both an information theoretic perspective and a generalized projection geometry. Further insights have also been obtained via discussions on its relations and differences from major existing approaches.

References

23.1 H. Akaike: A new look at the statistical model identification, IEEE Tr. Automatic Control, 19, 714-723 (1974)
23.2 H. Akaike: Likelihood of a model and information criteria, Journal of Econometrics, 16, 3-14 (1981)
23.3 H. Akaike: Factor analysis and AIC, Psychometrika, 52, 317-332 (1987)

23.4 S. Amari: Differential geometry methods in statistics, Lecture Notes in Statistics 28, Springer (1985)
23.5 S. Amari: Information geometry of the EM and em algorithms for neural networks, Neural Networks, 8, No. 9, 1379-1408 (1995)
23.6 SI. Amari, A. Cichocki, HH. Yang: A new learning algorithm for blind separation of sources. In: DS. Touretzky et al. (eds.), *Advances in Neural Information Processing 8*, MIT Press, 757-763 (1996)
23.7 AC. Atkinson: Likelihood ratios, posterior odds and information criteria, Journal of Econometrics, 16, 15-20 (1981)
23.8 A. Bell, T. Sejnowski: An information maximization approach to blind separation and blind deconvolution, Neural Computation, 17, 1129-1159 (1995)
23.9 J. Berger: *Statistical Decision Theory and Bayesian Analyses* (Springer-Verlag, New York) (1985)
23.10 CM. Bishop: Training with noise is equivalent to Tikhonov regularization, Neural Computation, 7, 108-116 (1995)
23.11 H. Bozdogan: Model Selection and Akaike's Information Criterion: The general theory and its analytical extension, Psychometrika, 52, 345-370 (1987)
23.12 H. Bozdogan, DE. Ramirez: FACAIC: Model selection algorithm for the orthogonal factor model using AIC and FACAIC, Psychometrika, 53 (3), 407-415 (1988)
23.13 JE. Cavanaugh: Unifying the derivations for the Akaike and corrected Akaike information criteria, Statistics and Probability Letters, 33, 201-208 (1997)
23.14 S. Chib: Marginal likelihood from the Gibbs output, Journal of the American Statistical Association, 90 (432), 1313-1321 (1995)
23.15 GC. Chow: A comparison of the information and posterior probability criteria for model selection, Journal of Econometrics, 16, 21-33 (1981)
23.16 I. Csiszar, G. Tusnady: Information geometry and alternating minimization procedures, Statistics and Decisions, Supplementary Issue, No. 1, 205-237 (1984)
23.17 P. Dayan, GE. Hinton: The Helmholtz machine, Neural Computation 7, No. 5, 889-904 (1995)
23.18 P. Dayan, GE. Hinton: Varieties of Helmholtz machine, Neural Networks, 9, No. 8, 1385-1403 (1996)
23.19 G. Cooper, E. Herskovitz: A Bayesian method for the induction of probabilistic networks from data, Machine Learning, 9, 309-347 (1992)
23.20 AP. Dempster, NM. Laird, DB. Rubin: Maximum-likelihood from incomplete data via the EM algorithm, J. Royal Statistical Society, 39, 1-38 (1977)
23.21 PA. Devijver, J. Kittler: Pattern Recognition: A Statistical Approach (Prentice-Hall) (1982)
23.22 L. Devroye et al.: *A Probability Theory of Pattern Recognition* (Springer) (1996)
23.23 TJ. DiCiccio et al.: Computing Bayes factors by combining simulations and asymptotic Approximations, J. American Statistical Association, 92 (439), 903-915 (1997)
23.24 B. Efron: Estimating the error rate of a prediction rule: Improvement on cross-validation, J. American Statistical Association, 78, 316-331 (1983)
23.25 B. Efron, R. Tibshirani: *An Introduction to the Bootstrap* (Chaoman and Hall, New York) (1993)
23.26 AE. Gelfand, DK. Dey: Bayesian model choice: Asymptotic and exact calculations, Journal of the Royal Statistical Society B, 56 (3), 501-514 (1994)
23.27 S. Geman, E. Bienenstock, R. Doursat: Neural Networks and the bias-variance dilemma, Neural Computation, 4, 1-58 (1992)

23.28 Z. Ghahramani, MJ. Beal: Variational inference for Bayesian mixture of factor analysis. In: SA. Solla, TK. Leen, KR, Muller, (eds.), *Advances in Neural Information Processing Systems 12*, Cambridge, MA: MIT Press, 449-455 (2000)
23.29 A. Gammerman, V. Vovk: Kolmogorov complexity, Computer Journal, 42 (4) (1999)
23.30 F. Girosi et al.: Regularization theory and neural architectures, Neural Computation, 7, 219-269 (1995)
23.31 GE. Hinton, P. Dayan, BJ. Frey, RN. Neal: The wake-sleep algorithm for unsupervised learning neural networks, Science, 268, 1158-1160 (1995)
23.32 GE. Hinton, RS. Zemel: Autoencoders, minimum description length and Helmholtz free energy, Advances in NIPS, 6, 3-10 (1994)
23.33 GE. Hinton, D. van Camp: Keeping neural networks simple by minimizing the description length of the weights, *Sixth ACM Conference on Computational Learning Theory*, Santa Cruz, July, 1993 (1993)
23.34 XL. Hu, L. Xu: A Comparative Study of Several Cluster Number Selection Criteria, *Proc. of IDEAL03*, Lecture Notes in Computer Science 2690, Springer-Verlag, pp. 195-202 (2003)
23.35 CM. Hurvich, CL. Tsai: Regression and time series model in samll samples, Biometrika, 76, 297-307 (1989)
23.36 CM. Hurvich, CL. Tsai: A corrected Akaike information criterion for vector autoregressive model selection, J. of Time Series Analysis, 14, 271-279 (1993)
23.37 H. Jeffreys: *Theory of Probability* (Clarendon Press, Oxford 1939)
23.38 RL. Kashyap: Optimal choice of AR and MA parts in autoregressive and moving-average models, IEEE Trans. PAMI, 4, 99-104 (1982)
23.39 RE. Kass, AE. Raftery: Bayes factors, Journal of the American Statistical Association, 90 (430), 773-795 (1995)
23.40 RE. Kass, L. Wasserman: The selection of prior distributions by formal rules, J. American Statistical Association, 91 (435), 1343-1370 (1996)
23.41 RW. Katz: On some criteria for estimating the order of a Markov chain, Technometrics, 23 (3), 243-249 (1981)
23.42 P. Kontkanen et al.: Bayesian and Information-Theoretic priors for Bayeisan network parameters, Machine Learning: ECML-98, Lecture Notes in Artificial Intelligence, Vol. 1398, 89-94, Springer-Verlag (1998)
23.43 J. Ma, T. Wang, L. Xu: A gradient BYY harmony learning rule on Gaussian mixture with automated model selection, in press, Neurocomputing (2003)
23.44 J. Ma, T. Wang, L. Xu: The Annealing EM Algorithm for Gaussian Mixture with Automated Model Selection (submitted) (2003)
23.45 D. Mackey: A practical Bayesian framework for back-propagation, Neural Computation, 4, 448-472 (1992)
23.46 D. Mackey: Bayesian Interpolation, Neural Computation, 4, 405-447 (1992)
23.47 GJ. McLachlan, T. Krishnan: *The EM Algorithm and Extensions* (John Wiley and Son, INC. 1997)
23.48 AA. Neath, JE. Cavanaugh: Regression and Time Series model selection using variants of the Schwarz information criterion, Communications in Statistics A, 26, 559-580 (1997)
23.49 MA. Newton, AE. Raftery: Approximate Bayesian inference with the weighted likelihood Bootstrap, J. Royal Statistical Society B, 56 (1), 3-48 (1994)
23.50 A. O'Hagan: Fractional Bayes factors for model comparison, J. Royal Statistical Society B, 57 (1), 99-138 (1995)
23.51 RA. Redner, HF. Walker: Mixture densities, maximum likelihood, and the EM algorithm, SIAM Review, 26, 195-239 (1984)

23.52 J. Rissanen: Stochastic complexity and modeling, Annals of Statistics, 14 (3), 1080-1100 (1986)
23.53 J. Rissanen: *Stochastic Complexity in Statistical Inquiry* (World Scientific, Singapore 1989)
23.54 J. Rissanen: Hypothesis selection and testing by the MDL principle, Computer Journal, 42 (4), 260-269 (1999)
23.55 I. Rivals, L. Personnaz: On Cross Validation for Model Selection, Neural Computation, 11, 863-870 (1999)
23.56 M. Sato: Online model selection based on the vairational Bayes, Neural Computation, 13, 1649-1681 (2001)
23.57 E. Saund: A multiple cause mixture model for unsupervised learning, Neural Computation, Vol. 7, pp. 51-71 (1995)
23.58 L. Saul, MI. Jordan: Exploiting tractable structures in intractable Networks, *Advances in NIPS 8*, MIT Press, 486-492 (1995)
23.59 G. Schwarz: Estimating the dimension of a model, Annals of Statistics, 6, 461-464 (1978)
23.60 SL. Sclove: Application of model-selection criteria to some problems in multivariate analysis, Psychometrika, 52 (3), 333-343 (1987)
23.61 C. Spearman: General intelligence domainively determined and measured, Am. J. Psychol. 15, 201-293 (2004)
23.62 M. Stone: Cross-validatory choice and assessment of statistical prediction, *J. Royal Statistical Society B*, 36, 111-147 (1974)
23.63 M. Stone: Asymptotics for and against cross-validation, *Biometrika*, 64 (1), 29-35 (1977)
23.64 M. Stone: An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, *J. Royal Statistical Society B*, 39 (1), 44-47 (1977)
23.65 M. Stone: Cross-validation: A review, *Math. Operat. Statist.* , 9, 127-140 (1978)
23.66 M. Stone: Comments on model selection criteria of Akaike and Schwartz. *J. Royal Statistical Society B*, 41 (2), 276-278 (1979)
23.67 N. Sugiura: Further analysis of data by Akaike's infprmation criterion and the finite corrections, *Communications in Statistics A*, 7, 12-26 (1978)
23.68 AN. Tikhonov, VY. Arsenin: *Solutions of Ill-posed Problems*, Winston and Sons (1977)
23.69 CS. Wallace, DM. Boulton: An information measure for classification, *Computer Journal*, 11, 185-194 (1968)
23.70 CS. Wallace, PR. Freeman: Estimation and inference by compact coding, *J. of the Royal Statistical Society*, 49 (3), 240-265 (1987)
23.71 CS. Wallace, DR. Dowe: Minimum message length and Kolmogorov complexity, *Computer Journal*, 42 (4), 270-280 (1999)
23.72 S. Waterhouse et al.: Bayesian method for mixture of experts. In: DS. Touretzky et al. (eds.), Advances in NIPS 8, 351-357 (1996)
23.73 DH. Wolpert: On Bias Plus Variance, Neural Computation, 9 (1997)
23.74 VN. Vapnik: *The Nature Of Statistical Learning Theory* (Springer-Verlag) (1995)
23.75 L. Xu: Advances on BYY Harmony Learning: Information Theoretic Perspective, Generalized Projection Geometry, and Independent Factor Autodetermination, in press, IEEE Trans on Neural Networks (2004)
23.76 L. Xu: BYY Learning, Regularized Implementation, and Model Selection on Modular Networks with One Hidden Layer of Binary Units, *Neurocomputing*, Vol. 51, pp. 227-301 (2003)
23.77 L. Xu: Data smoothing regularization, multi-sets-learning, and problem solving strategies, *Neural Networks, Vol. 15, No. 5-6*, 817-825 (2003)

23.78 L. Xu: Independent Component Analysis and Extensions with Noise and Time: A Bayesian Ying Yang Learning Perspective, *Neural Information Processing - Letters and Reviews*, Vol. 1, No. 1, pp1-52 (2003)
23.79 L. Xu: Data-Smoothing Regularization, Normalization Regularization, and Competition-Penalty Mechanism for Statistical Learning and Multi-Agents, *Proc. IJCNN '03*, July 20-24, 2003, Portland, Oregon, pp. 2649-2654 (2003)
23.80 L. Xu: BYY Harmony Learning, Structural RPCL, and Topological Self-Organizing on Mixture Models, *Neural Networks, Vol. 15, No. 8-9*, 1125-1151 (2002)
23.81 L. Xu: Bayesian Ying Yang Harmony Learning, *The Handbook of Brain Theory and Neural Networks*, Second edition, (MA Arbib, Ed.), Cambridge, MA: The MIT Press, pp. 1231-1237 (2002)
23.82 L. Xu: BYY Harmony Learning, Independent State Space and Generalized APT Financial Analyses, *IEEE Tr on Neural Networks*, 12 (4), 822-849 (2001)
23.83 L. Xu: Best Harmony, Unified RPCL and Automated Model Selection for Unsupervised and Supervised Learning on Gaussian Mixtures, Three-Layer Nets and ME-RBF-SVM Models, *Intl J of Neural Systems*, 11 (1), 43-69 (2001)
23.84 L. Xu: Temporal BYY Learning for State Space Approach, Hidden Markov Model and Blind Source Separation, *IEEE Tr on Signal Processing 48*, 2132-2144 (2000)
23.85 L. Xu: BYY Learning System and Theory for Parameter Estimation, Data Smoothing Based Regularization and Model Selection, *Neural, Parallel and Scientific Computations*, Vol. 8, pp. 55-82 (2000)
23.86 L. Xu: Bayesian Ying Yang Unsupervised and Supervised Learning: Theory and Applications, *Proc. of 1999 Chinese Conference on Neural Networks and Signal Processing*, pp. 12-29, Shantou, China, Nov. 1999 (1999)
23.87 L. Xu: Bayesian Ying Yang Theory for Empirical Learning, Regularization and Model Selection: General Formulation, *Proc. IJCNN'99*, DC, USA, July 10-16, 1999, Vol. 1 of 6, pp. 552-557 (1999)
23.88 L. Xu: BYY Data Smoothing Based Learning on A Small Size of Samples, *Proc. IJCNN'99*, DC, USA, July 10-16, 1999, Vol. 1 of 6, pp. 546-551 (1999)
23.89 L. Xu: Data Mining, Unsupervised Learning and Bayesian Ying Yang Theory, *Proc. IJCNN'99*, DC, USA, July 10-16, 1999, Vol. 4 of 6, pp. 2250-2525 (1999)
23.90 L. Xu: Bayesian Ying Yang System and Theory as a Unified Statistical Learning Approach:(V) Temporal Modeling for Temporal Perception and Control, *Proc. ICONIP'98*, Oct.21-23, 1998, Kitakyushu, Japan, Vol. 2, pp. 877-884 (1998)
23.91 L. Xu: Bayesian Kullback Ying-Yang Dependence Reduction Theory, Neurocomputing, 22 (1-3), 81-112 (1998)
23.92 L. Xu: RBF Nets, Mixture Experts, and Bayesian Ying Yang Learning, *Neurocomputing*, Vol. 19, No. 1-3, 223-257 (1998)
23.93 L. Xu: Bayesian Ying Yang Learning Theory For Data Dimension Reduction and Determination, *Journal of Computational Intelligence in Finance*, Finance & Technology Publishing, Vol. 6, No. 5, pp, 6-18 (1998)
23.94 L. Xu: Bayesian Ying Yang System and Theory as A Unified Statistical Learning Approach (VII): Data Smoothing, *Proc. ICONIP'98*, Oct. 21-23, 1998, Kitakyushu, Japan, Vol. 1, pp. 243-248 (1998)
23.95 L. Xu: BKYY Three Layer Net Learning, EM-Like Algorithm, and Selection Criterion for Hidden Unit Number, *Proc. ICONIP'98*, Oct. 21-23, 1998, Kitakyushu, Japan, Vol. 2, pp. 631-634 (1998)
23.96 L. Xu: Bayesian Ying Yang Dependence Reduction Theory and Blind Source Separation on Instantaneous Mixture, *Proc. Intl ICSC Workshop I&ANN'98*, Feb.9-10, 1998, Tenerife, Spain, pp. 45-51 (1998)

23.97 L. Xu: Bayesian Ying Yang System and Theory as A Unified Statistical Learning Approach: (VI) Convex Divergence, Convex Entropy and Convex Likelihood, *Proc. IDEAL98*, Hong Kong, pp. 1-12 (1998)
23.98 L. Xu: Rival Penalized Competitive Learning, Finite Mixture, and Multisets Clustering, *Proc. of IJCNN98*, Anchorage, Alaska, Vol. 2, pp. 2525-2530 (1998)
23.99 L. Xu: Bayesian Ying Yang System and Theory as A Unified Statistical Learning Approach: (IV) Further Advances, *Proc. IJCNN98*, Anchorage, Alaska, Vol. 2, pp. 1275-1270 (1998)
23.100 L. Xu: BKYY Dimension Reduction and Determination, *Proc. IJCNN98*, Anchorage, Alaska, Vol. 3, pp. 1822-1827 (1998)
23.101 L. Xu: Bayesian Ying Yang System and Theory as A Unified Statistical Learning Approach: (I) Unsupervised and Semi-Unsupervised Learning. In: S. Amari, N. Kassabov (eds.), *Brain-like Computing and Intelligent Information Systems*, Springer-Verlag, pp. 241-274 (1997)
23.102 L. Xu: Bayesian Ying Yang System and Theory as A Unified Statistical Learning Approach (II): From Unsupervised Learning to Supervised Learning and Temporal Modeling. In: KM. Wong et al. (eds.), *Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective*, Springer-Verlag, pp. 25-42 (1997)
23.103 L. Xu: Bayesian Ying Yang System and Theory as A Unified Statistical Learning Approach (III): Models and Algorithms for Dependence Reduction, Data Dimension Reduction, ICA and Supervised Learning. In: KM. Wong et al. (eds.), *Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective*, Springer-Verlag, pp. 43-60 (1997)
23.104 L. Xu: Bayesian Ying Yang Machine, Clustering and Number of Clusters, *Pattern Recognition Letters 18*, No. 11-13, 1167-1178 (1997)
23.105 L. Xu: Bayesian Ying Yang Learning Based ICA Models, *Proc. IEEE NNSP97*, Sept. 24-26, 1997, Florida, pp. 476-485 (1997)
23.106 L. Xu: Bayesian-Kullback YING-YANG Learning Scheme: Reviews and New Results, *Proc. ICONIP96*, Sept.24-27, 1996, Hong Kong, Vol. 1, 59-67 (1996)
23.107 L. Xu: Bayesian-Kullback YING-YANG Machines for Supervised Learning, *Proc. WCNN96*, Sept.15-18, 1996, San Diego, CA, pp. 193-200 (1996)
23.108 L. Xu: How Many Clusters?: A YING-YANG Machine Based Theory for A Classical Open Problem in Pattern Recognition, *Proc IEEE ICNN96*, June 2-6, 1996, DC, Vol. 3, pp. 1546-1551 (1996)
23.109 L. Xu: Bayesian-Kullback Coupled YING-YANG Machines: Unified Learnings and New Results on Vector Quantization, *Proc. ICONIP95*, Oct 30-Nov.3, 1995, Beijing, China, pp. 977-988 (1995)
23.110 L. Xu: YING-YANG Machine for Temporal Signals, Keynote Talk, *Proc. IEEE Intl Conf. on NNSP95*, Dec.10-13, 1995, Nanjing, Vol. 1, pp. 644-651 (1995)
23.111 L. Xu: New Advances on The YING-YANG Machine, *Proc. Intl. Symp. on Artificial Neural Networks*, Dec.18-20, 1995, Taiwan, ppIS07-12 (1995)
23.112 L. Xu: Cluster Number Selection, Adaptive EM Algorithms and Competitive Learnings, Invited Talk, *Proc. IEEE Intl Conf.. on NNSP95*, Dec.10-13, 1995, Nanjing, Vol. 2, pp. 1499-1502 (1995)
23.113 L. Xu: New Advances on The YING-YANG Machine, Invited Talk, *Proc. of 1995 Intl Symp. on Artificial Neural Networks*, Dec.18-20, 1995, Taiwan, ppIS07-12 (1995)
23.114 L. Xu, A. Krzyzak, E. Oja: Rival Penalized Competitive Learning for Clustering Analysis, RBF net and Curve Detection, IEEE Tr. on Neural Networks, 4, 636-649 (1993)

23.115 L. Xu: Least mean square error reconstruction for self-organizing neural-nets, Neural Networks, 6, 627-648, 1993. Its early version on *Proc. IJCNN91'Singapore*, 2363-2373 (1991)