J Glob Optim (2010) 47:369–401
DOI 10.1007/s10898-008-9364-0

Machine learning problems from optimization perspective

Lei Xu

Received: 19 September 2008 / Accepted: 3 October 2008 / Published online: 14 November 2008

Abstract Both optimization and learning play important roles in a system for intelligent tasks. On one hand, we introduce three types of optimization tasks studied in the machine learning literature, corresponding to the three levels of inverse problems in an intelligent system. Also, we discuss three major roles of convexity in machine learning, either directly towards a convex programming or approximately transferring a difficult problem into a tractable one in help of local convexity and convex duality. No doubly, a good optimization algorithm takes an essential role in a learning process and new developments in the literature of optimization may thrust the advances of machine learning. On the other hand, we also interpret that the key task of learning is not simply optimization, as sometimes misunderstood in the optimization literature. We introduce the key challenges of learning and the current status of efforts towards the challenges. Furthermore, learning versus optimization has also been examined from a unified perspective under the name of Bayesian Ying-Yang learning, with combinatorial optimization made more effectively in help of learning.

Keywords Three levels of inverse problems · Parameter learning · Model selection · Local convexity · Convex duality · Learning versus optimization · Convex programming · Bayesian Ying-Yang learning · Automatic model selection · Learning based combinatorial optimization

1 Introduction

Optimization takes an essential part in an intelligent system. Associating with three different levels of inverse problems, there are three types of nested optimization tasks.

The first type of optimization (shortly, Type-1 optimization) is associated with tasks of inverse inference. An observation x is regarded as either generated from an inner representation

L. Xu (✉)
Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong, China
e-mail: lxu@cse.cuhk.edu.hk

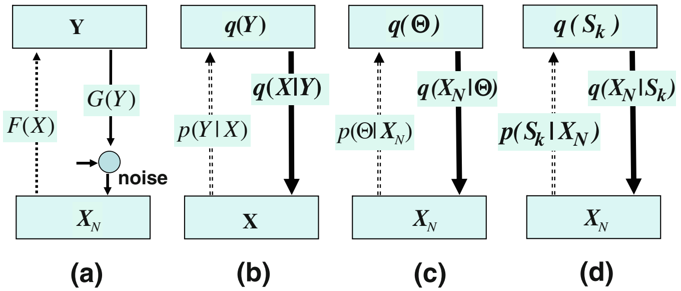


Fig. 1 Optimizations for inverse inference and probabilistic perspective

y or a consequence from a cause y via a given mapping $G : y \rightarrow x$. The task is inversely inferring y. Examples of such tasks include memory association, classification, decision making, input encoding, reasoning, etc. If $G : y \rightarrow x$ is one-to-one and its inverse one-to-one mapping $F : x \rightarrow y$ is analytically solvable, we can directly compute $y = F(x)$. In other cases, we need to enumerate every $y \in \mathcal{D}_y$ and check whether it is mapped to the observation x by $G : y \rightarrow x$, where $\mathcal{D}_y$ consists of all the possible values that y may take.

This is not a simple task due to uncertainties. One uncertainty is incurred by noises in observation, as shown in Fig. 1a, for which we need an objective measure with noise in consideration and search a best y via an optimization. Also, uncertainty is involved because the mapping $G : y \rightarrow x$ is many-to-one or infinite many to one, for which we need an additional measure that regularizes the optimization such that a most regular or reasonable y can be selected among many possible solutions. In the framework of probability theory, the first uncertainty is described by a distribution $q(x|y)$ for a probabilistic mapping $y \rightarrow x$ while the second uncertainty is considered by a distribution $q(y)$ for every $y \in \mathcal{D}_y$ on its chance to be a reasonable cause or inner representation, as shown in Fig. 1b. Then the uncertainty of corresponding inverse mapping is described by a distribution $p(y|x)$ for a probabilistic inverse map $x \rightarrow y$, summarized in the 1st column of Table 1 are four typical ways.

The first choice is Bayesian inference (BI) that provides a distribution $p(y|x)$ via combining evidences from $q(x|y, \theta_{x|y})$ and $q(y|\theta_y)$ in a normalized way, which involves an integral with a computational complexity that is usually too high to be practical. The difficulty is tackled by seeking a most probable mapping $x \rightarrow y$ in a sense of the largest probability $p(y|x)$, called the maximum Bayes (MB) or MAximum Posteriori (MAP). It further degenerates into $y^* = arg \max_y q(x|y, \theta_{x|y})$ when there is no knowledge about $q(y|\theta_y)$. In some cases, making maximization may also be computationally expensive. Instead, the last choice is to Learn a Parametric Distribution (LPD) $p(y|x, \theta_{y|x})$ by which an inverse mapping $x \rightarrow y$ can be fast implemented. To get this $p(y|x, \theta_{y|x})$, we need its structure pre-specified and then learn the parameter set $\theta_{y|x}$ from samples either based on $q(x|y, \theta_{x|y})$ and $q(y|\theta_y)$ or in help of a teacher who teaches a desired response to each sample. Actually, this LPD is a special case of the following second type of inverse problems.

The above studies base on knowing $G : y \rightarrow x$ or $q(x|y)$ and $q(y)$, while they are usually unknown. What we can actually base on is only a set of observation samples $\mathcal{X}_N = \{x_t\}_{t=1}^N$. Provided that $q(x|y)$ and $q(y)$ come from two families of parametric functions $q(x|y, \theta_{x|y})$ and $q(y|\theta_y)$ with their corresponding function structures pre-specified but two sets $\theta_{x|y}, \theta_y$ of unknown continuous parameters. As illustrated in Fig. 1c, the task is getting an inverse mapping $\mathcal{X}_N \rightarrow \Theta$, referred by the term *estimation* or *parameter learning* for Θ. This Θ

Table 1 Typical methods for three levels of inverse problems

	(a) Inverse inference on y	(b) Parameter learning	(c) Model selection
BI	$p(y \mid x) = \frac{q(x \mid y, \theta_{x \mid y}) q(y \mid \theta_y)}{q(x \mid \theta)}$ $q(x \mid \theta) = \int q(x \mid y, \theta_{x \mid y}) q(y \mid \theta_y) dy$	$p\,(\theta \mid X_N) = \frac{q(X_N \mid \theta) q(\theta)}{q(X_N \mid S)}$ $q\,(X_N \mid S) = \int q\,(X_N \mid \theta)\, q\,(\theta)\, d\theta$	$p\,(k \mid X_N) = \frac{q(X_N \mid S_k) q(k)}{q(X_N \mid \aleph)}$ $q\,(X_N \mid \aleph) = \sum_k q\,(X_N \mid S_k)\, q\,(k)$
MB	$\max_y \left[q\left(x \mid y, \theta_{x \mid y}\right) q\left(y \mid \theta_y\right) \right]$	$\max_\theta \left[q\,(X_N \mid \theta)\, q\,(\theta) \right]$	$\max_k \left[q\,(X_N \mid S_k)\, q\,(k) \right]$
ML	$\max_y q\left(x \mid y, \theta_{x \mid y}\right)$	$\max_\theta q\,(X_N \mid \theta)$	$\max_k q\,(X_N \mid S_k)$
LPD	$p\left(y \mid x, \theta_{y \mid x}\right)$	$p\,(\theta \mid X_N)$	$p\,(k \mid X_N)$

BI—Bayesian Inference, MB—Maximum Bayes or called Maximum Posteriori (MAP), ML–Maximum Likelihood or Marginal Likelihood, LPD—Learned Parametric Distribution

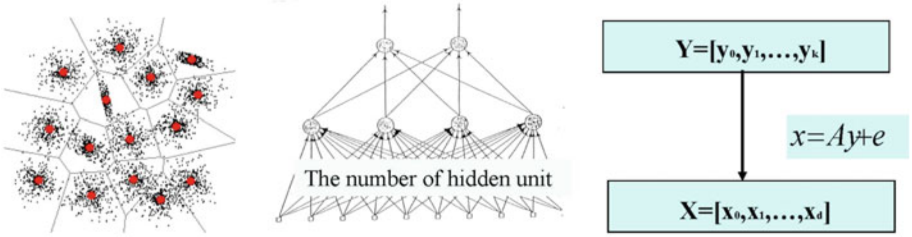


Fig. 2 A combination of a series of individual simple structures

consists of $\theta_{x|y}$, θ_y, as well as $\theta_{y|x}$ if the above LPD is considered together. Similar to Fig. 1b, we consider two distributions $q(\mathcal{X}_N|\Theta)$ and $q(\Theta)$ for uncertainties.

Usually, $q(\mathcal{X}_N|\Theta)$ is described by $q(\mathcal{X}_N|\Theta) = \int q(\mathcal{X}_N|\mathcal{Y}_N, \theta_{x|y})q(\mathcal{Y}_N|\theta_y)d\mathcal{Y}_N$, while it is difficult to get an appropriate $q(\Theta)$, which needs a priori knowledge that we may not have. The simplest and most widely studied one is the maximum likelihood (ML) learning $\max_\Theta q(\mathcal{X}_N|\Theta)$, with the role of $q(\Theta)$ ignored. Also, various efforts were made on considering $q(\Theta)$. One is the choice at Table 1—MB, i.e., $\max_\Theta[q(\mathcal{X}_N|\Theta)q(\Theta)]$. The second type of optimization (shortly, Type-2 optimization) is encountered during implementing both the ML and MB learning, featured by a continuous optimization that usually involves many local optimal solutions. Extensive studies have been made under different names [22,26,34], referred collectively in term of Bayesian school. Related efforts also include those made under *Tikhonov regularization* [21,30] or regularization approaches.

Conceptually, we may also consider the choice of Table 1b—BI for a probabilistic inverse mapping by a distribution $p(\Theta|\mathcal{X}_N)$. Getting $p(\Theta|\mathcal{X}_N)$ encounters a difficult integral over Θ. An alternative is using a particularly designed parametric structure in place of $p(\Theta|\mathcal{X}_N)$, i.e., the choice LPD in the 2nd column of Table 1. Except of some special cases, even the integral over $\mathcal{Y}_N$ for $q(\mathcal{X}_N|\Theta)$ encounters either a summation or a numerical integral, both of which involve huge computing costs. Efforts have been made in the Helmholtz free energy based learning [6,9], BYY Kullback learning [37], and BYY harmony learning [37,50] for avoiding these integrals and getting $p(y|x, \theta_{y|x})$ and $p(\Theta|\mathcal{X}_N)$ for the LPD choices in Table 1. Detailed discussions are referred to Sects.3.3 and 4.1.

Usually, we do not know how to pre-specify the structure of $q(x|y, \theta_{x|y})$ and $q(y|\theta_y)$. We are facing a problem of inversely determining them from $\mathcal{X}_N = \{x_t\}_{t=1}^N$ too, for which we consider a family of infinite many structures $\{S_{\mathbf{k}}(\Theta_{\mathbf{k}})\}$ via combining a set of individual simple structures (or simply called units) via a simple combination scheme, as shown in Fig. 2. Every unit can be simply one point, one dimension in a linear space, or one simple computing unit. The types of the basic units and the combination scheme jointly act as a seed or meta structure $\aleph$ that grows into a family $\{S_{\mathbf{k}}(\Theta_{\mathbf{k}})\}$ with each $S_{\mathbf{k}}$ sharing a same configuration but in different scales, each of which is labeled by a scale parameter $\mathbf{k}$ in term of one integer or a set of integers. That is, each specific $\mathbf{k}$ corresponds to one candidate model with a specific complexity. We can enumerate each candidate via enumerating[1] $\mathbf{k}$ and evaluate each candidate by a selection criterion $J(\mathbf{k})$.

As shown in Fig. 1d, the third level of inverse problems considers selecting an appropriate $\mathbf{k}^*$ based on $\mathcal{X}_N = \{x_t\}_{t=1}^N$ only, usually referred as *model selection*. That is, the third type optimization (shortly, Type-3 optimization) belongs discrete optimization. However, it is

[1] We say that $\mathbf{k}_1$ proceeds $\mathbf{k}_2$ or $\mathbf{k}_1 \prec \mathbf{k}_2$ if $S_{\mathbf{k}_1}$ is a part (or called a substructure) of $S_{\mathbf{k}_2}$. When $\mathbf{k}$ consists of only one integer, $\mathbf{k}_1 \prec \mathbf{k}_2$ becomes simply $\mathbf{k}_1 < \mathbf{k}_2$.

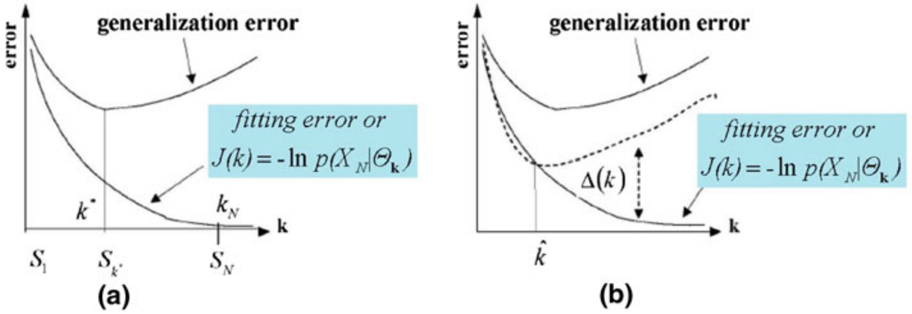


Fig. 3 Model selection: fitting performance versus generalization performance

not simply a conventional discrete optimization. We can not simply use $J(\mathbf{k}) = -\max_\Theta \ln p(X_t|\Theta_\mathbf{k})$ for this purpose, as illustrated in Fig. 3a. For a finite N, this $J(k)$ will keep decreasing as k increases and reach zero at a value k_N that is usually much larger than the appropriate one $\mathbf{k}^*$. Though a $S_\mathbf{k}(\Theta_\mathbf{k})$ with $\mathbf{k}^* \prec \mathbf{k}$ can get a low value $J(\mathbf{k})$ that means fitting $\mathcal{X}_N$ well, it has a poor generalization performance, i.e., performing poorly on new samples with the same regularity underlying $\mathcal{X}_N$. This is usually called *over-fitting* problem that makes the key challenge of learning is not same as seeking a global optimal solution. As shown in Fig. 3a, seeking the global minimum of $-\ln p(X_t|\Theta_\mathbf{k})$ tends to reduce generalization error for a small $\mathbf{k}$, though this tendency weakens as $\mathbf{k}$ decreases. On the other hand, as $\mathbf{k}$ goes beyond $\mathbf{k}^*$, not only seeking a global minimum is no longer difficult because a global minimum can be reached at infinite many values of $\Theta_\mathbf{k}$, but also reaching the global minimum of $-\ln p(X_t|\Theta_\mathbf{k})$ is no longer helpful to reduce generalization error.

Moreover, this Type-3 optimization is nested with a series of implementations of Type-2 optimization for estimating a best $\Theta^*_\mathbf{k}$ at each $\mathbf{k}$, which usually incurs a huge computing cost, while many practical applications demand that learning is made adaptively upon each sample comes. Efforts are also demanded on tackling this computational challenge too.

In the rest of this paper, several typical problems for three optimization types are introduced in Sect. 2. Also, we interpret that the key task of learning is not simply optimization, via introducing the key challenges of learning and the current status of efforts towards the challenges. Then, a unified perspective is provided in Sect. 3 to show how the three types interact and implemented in one learning system, under guidance of the *Bayesian Ying Yang harmony* theory. Moreover, comparisons are made on relations and differences with typical existing learning theories. In Sect. 4, learning versus optimization is further elaborated. The roles of convexity in machine learning are further discussed, featured by approximately transferring a difficult problem into a tractable one. Also, a learning based approach is introduced for making combinatorial optimization more effectively. Finally concluding remarks are made in Sect. 5.

2 Learning problems: three types of optimizations

2.1 Type 1: optimizations for inverse inference

We start at several typical examples of inverse inference by the choice Table 1a—MB (the choice MB in the first column of Table 1), i.e., the following optimization problem

$$y^* = arg \max_y [q(x|y)q(y)], \tag{1}$$

with $q(x|y)$ and $q(y)$ in certain specific forms. Usually, observation noises are regarded as from Gaussian,[2] that is,

$$q(x|y) = G(x|\mu_y, \Sigma_y),\ \mu_y = \int xq(x|y)dx,\ \Sigma_y = \int (x-\mu_y)(x-\mu_y)^T q(x|y)dx. \quad (2)$$

Listed in Table 2 are its typical structures and several structures of $q(y)$.

For $q(x|y)$ of Table 2c (A)&(1) (i.e., Type A & case (1) of $q(x|y)$ in Table 2c) and $q(y)$ of Table 2a (2) + Table 2b (1), Eq. 1 is solved analytically by a linear function and thus the mapping $x \to y$ is computable directly. We proceed to the combination with $q(y)$ of Table 2a (1) and $q(x|y)$ of Table 2c (A)&(2), at which Eq. 1 becomes

$$\ell^* = arg\max_{\ell}[G(x|\mu_\ell, \Sigma_\ell)\alpha_\ell]. \quad (3)$$

The mapping $x \to \ell^*$ is called maximum posteriori (MAP) classification, and is solved simply by enumeration. The situation becomes more complicated if we change $q(y)$ into Table 2a (2) + Table 2b (2), which leads to the problem of binary factor analysis [40,42,43]. With each element of $y = [y^{(1)}, \ldots, y^{(m)}]^T$ in binary value, Eq. 1 becomes equivalently a combinatorial optimization problem as follows:

$$\min_{y^{(1)},\ldots,y^{(m)}} \left\{ 0.5e^T\Sigma^{-1}e - \sum_{j=1}^{m}\left[y^{(j)}\ln q_j + (1-y^{(j)})\ln(1-q_j)\right]\right\},\ e = x - Ay - \mu,$$
$$\text{subject to}\ \ y^{(j)} = 1 \text{ or } 0,\ \ j = 1, \ldots, m, \quad (4)$$

for which an exhaustive enumeration will grow exponentially with m. Usually, a heuristic iterative algorithm is used [40,42,43] without a guarantee of a global optimal solution.

We further go to the combination with $q(x|y)$ of Table 2c (B)&(2) and with $q(y)$ of Table 2a (2) + Table 2b (3) or (4), which leads to the problem of nonGaussian factor analysis [40,42,43]. It follows that Eq. 1 becomes

$$\min_{\{y_g^{(j)}, i^{(j)}\}_{j=1}^m} \left\{0.5e^T\Sigma^{-1}e - \sum_{j=1}^{m}\ln\left[\beta^{(j,i^{(j)})}G(y_g^{(j)}|\nu^{(j,i^{(j)})}, \lambda^{(j,i^{(j)})})\right]\right\},\ \text{for Table 2b (3)},$$
$$\min_{y_g^{(1)},\ldots,y_g^{(m)}} \left\{0.5e^T\Sigma^{-1}e - \sum_{j=1}^{m}\ln\sum_{i}\beta^{(j,i)}G(y_g^{(j)}|\nu^{(j,i)}, \lambda^{(j,i)})\right\},\ \text{for Table 2b (4)}, \quad (5)$$

with $e = x - (Ay + \mu)$, which is an even difficult task of global continuous optimization.

Interestingly, a classical combinatorial optimization problem may also be re-examined from the perspective of Eq. 1. We start at one typical example that considers an $N \times N$ symmetrical matrix X that represents an attribute graph in consideration and a $N \times N$ symmetrical matrix A as a reference. The task is matching two graphs X, A via searching various permutations of their vertices, which is usually referred as attribute graph matching (AGM) [8,31,54,55]. A mismatch is considered via the following matrix norm:

$$E_o(Y, X) = \|X - Y^TAY\|^2, \quad (6)$$

and the task is formulated into a combinatorial optimization as follows

$$\min_{Y\in\mathbf{\Pi}_N} E_o(Y, X),\ \text{where } \mathbf{\Pi}_N \text{ consists of all the } N\times N \text{ permutation matrices.} \quad (7)$$

[2] In this paper, $G(u|\mu, \Sigma)$ denotes a Gaussian density of u with a mean μ and a covariance matrix Σ.

Table 2 Typical structures of $q(y) = q(y|\theta_y)$ and $q(x|y) = q(x|y, \theta_{x|y}) = G(x|\mu_y, \sum_y)$

(*a*)

y	(1) $\{l\}$	(2) $\{y\}$	(3) $\{y, l\}$
$q(y\|\theta_y)$	$q(l) = \alpha_l \geqslant 0, \sum_{l=1}^{k} \alpha_l = 1$	$q(y) = \mathbf{\Pi}_{j=1}^{m} q(y^{(j)})$	$q(y, l) = q(y\|l)\alpha_l, q(y\|l) = \prod_{j=1}^{m_l} q(y^{(j)}\|l)$

(*b*)

	(1) Gaussian	(2) Bernoulli	(3) $y^{(j)} = \{y_g^{(j)}, i^{(j)}\}$	(4) Mixture
$q(y^{(j)})$	$G(y^{(j)}\|0, \lambda^{(j)})$	$q_j^{y^{(j)}} (1 - q_j)^{1-y^{(j)}}$	$G(y_g^{(j)}\|\nu^{(j,i)}, \lambda^{(j,i)})\beta^{(j,i)}$	$\sum_{i=1}^{\kappa^{(j)}} \beta^{(j,i)} q^{(i)}(y_g^{(j)})$
$q(y^{(j)}\|l)$	$G(y^{(j)}\|0, \lambda_l^{(j)})$	$q_{l,j}^{y^{(j)}} (1 - q_{l,j})^{1-y^{(j)}}$	$G(y_g^{(j)}\|\nu_l^{(j,i)}, \lambda_l^{(j,i)})\beta_l^{(j,i)}$	$\sum_{i=1}^{\kappa^{(j)}} \beta_l^{(j,i)} q_l^{(i)}(y_g^{(j)})$

(*c*)

Type	A	B	C			(1)	(2)	(3)	(4)
y	$\{l\}$	$\{y\}$	$\{y, l\}$	$q(x\|y, \theta_{x\|y})$ $= G(x\|\mu_y, \Sigma_y)$	Relation of noise to y	none of $\{y, l\}$	only $\{l\}$	only $\{y\}$	both $\{y, l\}$
μ_y	μ_l	$Ay + \mu$	$A_l y + \mu_l$		Σ_y	Σ	Σ_l	Σ_y	$\Sigma_{y,l}$

Note: For Choice (b)(3) we have $q(y^{(j)}|l) = q(y_g^{(j)}, i^{(j)}|l) = q(y_g^{(j)}|i^{(j)}, l)q(i^{(j)}|l)$ with $q(y_g^{(j)}|i^{(j)}, l) = G(y_g^{(j)}|\nu_l^{(j,i)} \lambda_l^{(j,i)})$, $q(i^{(j)}|l) = \beta_l^{(j,i)} \geqslant 0, \sum_{i=1}^{\kappa_l^{(j)}} \beta_l^{(j,i)} = 1$

We encounter a nonlinear mapping $G : Y \rightarrow X$ by $Y^T AY \rightarrow X$ that can be put into the framework as shown in Fig. 1b, with

$$q(Y) \text{ on } \mathbf{\Pi}_N, \quad q(X|Y) = G(X|Y^T AY, \sigma^2 I_{N\times N}). \tag{8}$$

With x, y replaced by X, Y respectively, it follows that Eq. 1 becomes equivalent to Eq. 7 if $q(Y)$ is uniform over $\mathbf{\Pi}_N$.

Another example is the well known traveling salesman problem (TSP), i.e., a salesman visits every city only once and returns to the staring city, with this looping trip in a shortest distance. The locations of N cities can be represented by an $N \times 2$ matrix X with a loop path from the 1st row to the last row. Enumerating all the possible loops in $YX, \forall Y \in \mathbf{\Pi}_N$, the TSP is formulated as Eq. 7 with

$$E_o(Y, X) = \|D^T YX\|^2, \; D = [d_1, \ldots, d_N]. \tag{9}$$

For $1 \le i \le N-1$, each N dimensional vector d_i consists of zeros except that its ith element is 1 and its $i+1$-th element is -1, while the elements of d_N are zeros except that its N-th element is 1 and its 1st element is -1. Thus, $D^T YX$ calculates the location differences of two subsequent cities and its norm is thus the total distance of the corresponding loop.

The TSP problem, as well as a class of combinatorial optimization problems formulated as Eq. 7, can also be revisited from the perspective of Eq. 1 with

$$q(Y) \text{ on } \mathbf{\Pi}_N, \quad q(X|Y) = e^{-\frac{1}{\lambda}E_o(Y,X)}/Z_\lambda(X), \quad Z_\lambda(X) = \sum_Y e^{-\frac{1}{\lambda}E_o(Y,X)}. \tag{10}$$

Again, Eq. 1 becomes equivalent to Eq. 7 if $q(Y)$ is uniform over $\mathbf{\Pi}_N$.

This perspective not only provides new insights on such a class of classical combinatorial optimization problems, but also a different road for making combinatorial optimization more effectively in help of approaches developed in the literature of machine learning. This issue will be further discussed later in Sect. 4.

2.2 Type 2: optimizations for parameter learning

Given a set $\Xi = \{\xi_t\}$ with its elements being identically and independently distributed (i.i.d.), we have the following product form

$$p(\Xi) = \prod_t p(\xi_t). \tag{11}$$

When the samples in $\mathcal{X}_N = \{x_t\}_{t=1}^N$ are i.i.d., the choice Table 1b—ML becomes

$$\max_\Theta \ln q(\mathcal{X}_N|\Theta), \quad \ln q(\mathcal{X}_N|\Theta) = \sum_{t=1}^N \ln q(x_t|\Theta). \tag{12}$$

Also, Θ usually contains a part of parameters that should satisfy some constraints, e.g., any covariance matrix should be nonnegative definite, and a discrete probability satisfies

$$\sum_{j=1}^n a_j = 1, \quad 1 \ge a_j \ge 0. \tag{13}$$

Therefore, a Type-2 optimization is usually a constrained continuous optimization.

We further discuss several typical problems of Type-2 optimizations for parameter learning, again with $q(x|y, \theta_{x|y})$ and $q(y|\theta_y)$ in those structures of Table 2. The simplest case is the combination with $q(x|y)$ of Table 2c (A)&(1) and $q(y)$ of Table 2a (2)+Table 2b (1), by which Eq. 12 is solved analytically without involving an optimization. However, another simple situation, featured by the combination with $q(y)$ of Table 2a (1) and $q(x|y)$ of Table 2c (A)&(2), becomes quite complicated already. That is, considering the following Gaussian mixture

$$q(x_t|\Theta) = \sum_{\ell=1}^{k} \alpha_\ell G(x|\mu_\ell, \Sigma_\ell), \tag{14}$$

the ML learning by Eq. 12 is already a nonlinear multivariate continuous optimization, with constraints that α_ℓ should satisfy Eq. 13 and every Σ_ℓ should be nonnegative definite. There are many local optimal solutions, and there is no algorithm available to guarantee a global optimal solution. A widely used algorithm for implementing Eq. 12 is an iterative algorithm called expectation-maximization (EM), which guarantees the satisfaction of all the constraints during iterations and a convergence to a local optimal solution.

There are certain studies on convergence rate of the EM algorithm [53]. However, the picture about its computational complexity is quite complicated. To get some insights, we consider a degenerated case with $\Sigma_\ell = \sigma^2 I$ and $\alpha_\ell = 1/k$. In this case, a Type-1 optimization by Eq. 1 becomes equivalent to $\ell^* = arg\min_\ell \|x_t - \mu_\ell\|^2$ that classifies a sample x_t to the cluster represented by μ_ℓ. Thus, the set $\mathcal{X}_N$ of samples is divided into k clusters represented by k center points $\mu_\ell, \ell = 1, \ldots, k$, which is the widely encountered problem called *clustering analysis*. A typical formulation is that the representation of each x_t by its cluster center is measured by the corresponding square error and we expect the total square error over the entire set $\mathcal{X}_N$ is minimized, i.e.,

$$\begin{aligned} &\min_\Theta J_N(\Theta), \quad J_N(\Theta) = \sum_{t=1}^{N}\sum_{\ell=1}^{k} y_{\ell,t}\|x_t - \mu_\ell\|^2, \quad \Theta = \{\mu_\ell\}_{\ell=1}^k, \\ &\text{subject to } y_{\ell,t} = \begin{cases} 1, & \text{if } \ \ell = arg\min_\ell \|x_t - \mu_\ell\|^2, \\ 0, & \text{otherwise;} \end{cases} \end{aligned} \tag{15}$$

where $y_{\ell,t}$ indicates a classification of x_t to the cluster represented by μ_ℓ. From this problem, we observe two features. First, a Type-1 optimization $\min_\ell \|x_t - \mu_\ell\|^2$ is actually nested within the optimization by Eq. 15 for learning Θ. Second, this apparently simple task is actually a typical NP-hard combinatorial problem.

Next, we proceed to consider a problem called binary factor analysis [40,42,43], featured by the combination with $q(y)$ of Table 2a (2)+Table 2b (2) and $q(x|y)$ of Table 2c (B)&(2). It follows from previous discussions that Eq. 1 is alone a combinatorial optimization problem already. Moreover, $q(x|\Theta)$ at the choice Table 1a—BI takes the following specific form

$$q(x|\Theta) = \sum_y G(x|Ay + \mu, \Sigma) \prod_{j=1}^{m} q_j^{y^{(j)}}(1 - q_j)^{1-y^{(j)}}, \quad \Theta = \{A, \Sigma, \{q_j\}\}. \tag{16}$$

Comparing with Eq. 14, this $q(x|\Theta)$ is actually a mixture of 2^m Gaussian components that is enumerated via a binary vector y. Thus, there is also an EM algorithm for implementing its corresponding Eq. 12. However, not only the computing cost increases considerably because evaluating $q(x|\Theta)$ at each iteration needs to compute 2^m terms, but also it becomes much more vulnerable to fall a poor local solution since it becomes a nonlinear optimization problem with a considerably increased number of local optimal solutions.

We further consider the problem of nonGaussian factor analysis, featured by the combination with $q(x|y)$ of Table 2c (B)&(1) and with $q(y)$ of Table 2a (2)+Table 2b (3) or (4).

In this case, a Type-1 optimization by Eq. 1 has again a computational complexity that grows exponentially with m, while the implementation of a Type-2 optimization by Eq. 12 becomes even more difficult. We observe $q(y)$ of Table 2a (2) + Table 2b (3), with $q(x|\Theta)$ at the choice Table 1b—ML

$$q(x|\Theta) = \sum_{i^{(j)}=1, j=1}^{\kappa^{(j)}, m} \int G(x|Ay_g + \mu, \Sigma)[\beta^{(j,i^{(j)})} G(y_g^{(j)}|\nu^{(j,i^{(j)})}, \lambda^{(j,i^{(j)})})] dy_g, \quad (17)$$

which is actually a mixture of $\prod_{j=1}^{m} \kappa^{(j)}$ Gaussian components. Similarly, its corresponding Eq. 12 can be made by an EM algorithm, while we encounter a difficult nonlinear optimization with a large number of local optimal solutions. For the case with $q(y)$ of Table 2a (2) + Table 2b (4), by turning the product $\prod_{j=1}^{m} \sum_i \beta^{(j,i)} G(y_g^{(j)}|\nu^{(j,i)}, \lambda^{(j,i)})$ into a summation of $\prod_{j=1}^{m} \kappa^{(j)}$ terms [18] we encounter a situation similar to the above Eq. 17 though it is apparently that the corresponding integral over y_g can not be made analytically.

2.3 Type 3: optimizations for model selection

A Type-3 optimization is a discrete optimization via enumerating $\mathbf{k}$ and evaluating a criterion $J(\mathbf{k})$. As discussed after Fig. 3, the key challenge is how to get a $J(\mathbf{k})$ for a good approximation on the generalization performance of a learning model, only based on a finite size of samples in $\mathcal{X}_N$. In the past 30 or 40 years, several learning principles or theories have been proposed and studied for an appropriate $J(\mathbf{k})$, roughly along three directions.

Those measures summarized in Table 1 are featured by the most probable principle based on probability theory. The efforts of the first direction can be summarized under this principle. As discussed previously, the ML choice of the 2nd column in Table 1 can not serve as $J(\mathbf{k})$. Studies on the BI choice of the 2nd column, i.e., $J(\mathbf{k}) = -\max_{\Theta}[q(\mathcal{X}_N|\Theta)q(\Theta)]$, have been made under the name of *minimum message length* (MML) [34]. It can provide an improved performance over $J(\mathbf{k}) = -\max_{\Theta} q(\mathcal{X}_N|\Theta)$ but is sensitive to whether an appropriate $q(\Theta)$ is pre-specified, which is difficult. Studies on the BI choice of the third column in Table 1 have also been conducted widely in the literature. Usually assuming that $q(\mathbf{k})$ is equal for every $\mathbf{k}$, we are lead to the ML (marginal likelihood) choice of the third column, i.e., $J(\mathbf{k}) = -\ln q(\mathcal{X}_N|S_k)$, by which the effect of $q(\Theta)$ has been integrated out. However, the integral over Θ is difficult to compute and thus is approximately tackled by turning it into the following format:

$$J(k) = -\max_{\Theta} \ln q(\mathcal{X}_N|\Theta) + \Delta(\mathbf{k}), \quad (18)$$

where the term $\Delta(\mathbf{k})$ is resulted from a rough approximation such that it is computable. Differences on $q(\Theta)$ and on techniques for approximating the integral result in different specific forms. Typical efforts include those under the names of *Bayesian Information Criterion* [20,28], Bayes Factors [14], the evidence or the marginal likelihood [16], etc. Also, the *Akaike Information Criterion (AIC)* can be obtained as a special case though it was originally derived from a different perspective [1,2].

The second direction follows the well known principle of Ockham Razor, i.e., seeking a most economic model that represents $\mathcal{X}_N$. It is implemented via minimizing a two part coding length. One is for encoding the residuals or errors incurred by the model in representing $\mathcal{X}_N$, which actually corresponds to the first term in Eq. 18. The other is for encoding the model itself, which actually corresponds to the second term in Eq. 18. Different specific forms maybe obtained due to differences on what measure is used for the length and on how to evaluate the measure, which is usually difficult, especially for the second part coding. Studies have been made under the names of MML [34], *minimum description length* (MDL) [23], best information transfer, etc. After this or that type of approximation, the resulted criteria turn out closely related to or even same as those obtained along the above first direction.

Another direction is towards estimating the generalization performance directly. One typical approach is called *cross-validation* (CV). $\mathcal{X}_N$ is randomly and evenly divided into $D_i, i = 1, \ldots, m$ parts, each D_i is used to measure the performance of $S_{\mathbf{k}}$ with its $\Theta_{\mathbf{k}}$ determined from the rest samples in $\mathcal{X}_N$ after taking D_i away. Then we use the average performances of m times as an estimation of $J(\mathbf{k})$ [24,29]. One other approach is using the VC dimension based learning theory [32] to estimate a bound of generalization performance via theoretical analysis. A rough bound can be obtained for some special cases, e.g., a Gaussian mixture [35]. Generally, such a bound is difficult to get because it is very difficult to estimate the VC dimension of a learning model.

For all the above studies, we handle a discrete optimization nested with a series of implementations of Type-2 optimization for estimating a best $\Theta^*_{\mathbf{k}}$ at each $\mathbf{k}$. The task usually incurs a huge computing cost, while many practical applications demand that learning is made adaptively upon each sample comes. Moreover, the parameter learning performance deteriorates rapidly as $\mathbf{k}$ increases, which makes the value of $J(\mathbf{k})$ evaluated unreliably. Efforts have been made on tackling this challenge along two directions. One is featured by incremental algorithms that attempts to incorporate as much as possible what learned as $\mathbf{k}$ increases step by step, focusing on learning newly added parameters. Such an incremental implementation can save computing costs in certain extent. However, one Type-2 optimization has to be made by enumerating each value of $\mathbf{k}$, and computing costs are still very high. Also, it usually leads to suboptimal performance because not only those newly added parameters but also the old parameter set $\Theta_{\mathbf{k}}$ have to be re-learned. Another type of efforts has been made on a category that consists of individual substructures, e.g., a Gaussian mixture by Eq. 14. A local error criterion is used to check whether a new sample x is classified to one substructure. If x is regarded as not belonging to anyone of substructures, an additional substructure is added to accommodate this new x. This incremental implementation is much faster but very vulnerable to be trapped into a poor performance.

The other direction consists of learning algorithms that start with $\mathbf{k}$ at a large value and decrease $\mathbf{k}$ step by step, with extra parameters discarded and the remaining parameter updated. These algorithms are further classified into two types. One is featured by decreasing $\mathbf{k}$ step by step, based on evaluating the value of $J(\mathbf{k})$ at each $\mathbf{k}$. The other is called automatic model selection, with extra structural parts removed automatically during parameter learning. One early effort is Rival Penalized Competitive Learning (RPCL) [56] for a model that consists of k individual substructures, featured by a penalized mechanism that discards those extra substructures and makes model selection automatically during learning. Various extensions have been made in the past one decade and half. Readers are referred to a recent encyclopedia paper [57].

3 A Unified perspective: BYY learning

3.1 Bayesian Ying-Yang system and best harmony learning

Firstly proposed in 1995 [37] and developed in the past decade, Bayesian Ying-Yang (BYY) learning acts as a unified statistical framework featured by a Bayesian Ying-Yang (BYY) system with all the unknowns learned under a best harmony theory [42–44,50].

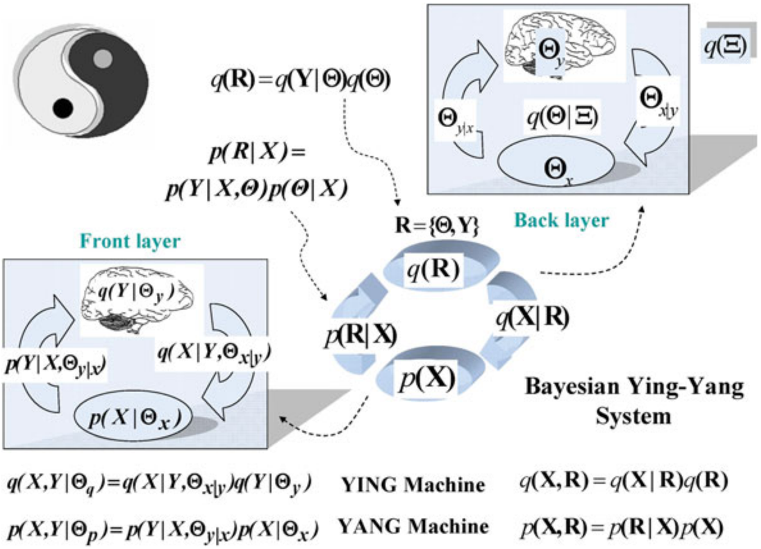


Fig. 4 Bayesian Ying-Yang system

As shown in Fig. 4, a unified scenario of Fig. 1 is considered by regarding that the observation set $\mathbf{X} = \{x\}$ are generated via a top-down path from its inner representation $\mathbf{R} = \{\mathbf{Y}, \Theta\}$. Given a system architecture, the parameter set Θ collectively represents the underlying structure of $\mathbf{X}$, while one element $y \in \mathbf{Y}$ is the corresponding inner representation of one element $x \in \mathbf{X}$. A mapping $\mathbf{R} \to \mathbf{X}$ and an inverse mapping $\mathbf{X} \to \mathbf{R}$ are jointly considered via the joint distribution of $\mathbf{X}$ and $\mathbf{R}$ in two types of Bayesian decomposition as shown at the right-bottom of Fig. 4. In a compliment to the famous ancient Ying-Yang philosophy, the decomposition of $p(\mathbf{X}, \mathbf{R})$ coincides the Yang concept with a visible domain $p(\mathbf{X})$ for a Yang space and a forward pathway by $p(\mathbf{R}|\mathbf{X})$ as a Yang pathway. Thus, $p(\mathbf{X}, \mathbf{R})$ is called Yang machine. Similarly, $q(\mathbf{X}, \mathbf{R})$ is called Ying machine with an invisible domain $q(\mathbf{R})$ for a Ying space and a backward pathway by $q(\mathbf{X}|\mathbf{R})$ as a Ying pathway. Such a Ying-Yang pair is called *Bayesian Ying-Yang (BYY) system*.

As shown in Fig. 4, the system is further divided into two layers. The front layer is actually the one shown in Fig. 1b, with a parametric Ying-Yang pair at the left-bottom of Fig. 4, which consists of four components with each associated with a subset of parameters $\Theta = \{\Theta_p, \Theta_q\}$, where $\Theta_p = \{\theta_{y|x}, \theta_x\}$ and $\Theta_q = \{\theta_y, \theta_{x|y}\}$. This Θ is accommodated on the back layer with a priori structure $q(\Theta|\Xi_q)$ to back up the front layer, the back layer may be further modulated by a meta knowledge from a meta layer $q(\Xi)$. Correspondingly, an inference on Θ is given by $p(\Theta|\mathbf{X}, \Xi_p)$ that integrates information from both the front layer and the meta layer. Putting together, we have

$$q(\mathbf{X}, \mathbf{R}) = q(\mathbf{X}|\mathbf{Y}, \theta_{x|y})q(\mathbf{Y}|\theta_y)q(\Theta|\Xi_q), \;\; p(\mathbf{X}, \mathbf{R}) = p(\Theta|\mathbf{X}, \Xi_p)p(\mathbf{Y}|\mathbf{X}, \theta_{y|x})p(\mathbf{X}|\theta_x). \tag{19}$$

The external input is a set of samples $\mathcal{X}_N = \{x_t\}_{t=1}^N$ of $\mathbf{X} = \{x\}$, based on which we form an estimate of $p(\mathbf{X}|\theta_x)$ either directly or with a unknown scalar parameter $\theta_x = h$. Based on this very limited knowledge, the goal of building up the entire system is too ambitious to pursuit. We need to further specify certain structures of $p(\mathbf{X}, \mathbf{R})$ and $q(\mathbf{X}, \mathbf{R})$. Similar to the discussions made at Sect. 1, the Ying Yang system is also featured by a given meta structure $\aleph$ that grows into a family $\{S_{\mathbf{k}}(\Theta_{\mathbf{k}})\}$ with each $S_{\mathbf{k}}$ sharing a same configuration but in different scales of $\mathbf{k}$. The meta structure $\aleph$ consists $\aleph_q$, $\aleph_p$ for the Ying machine and the Yang machine respectively, from which we get the structures of $q(\mathbf{X}, \mathbf{Y}|\Theta_q)$ and $p(\mathbf{X}, \mathbf{Y}|\Theta_p)$ in different scales.

Different structures of the Ying machine $q(\mathbf{X}, \mathbf{Y}|\Theta_q)$ are considered to accommodate the world knowledge and different types of dependencies encountered in various learning tasks. First, an expression format is needed for each inner representation $\mathbf{Y}$. The general case is $\mathbf{Y} = \{\mathbf{Y}_v, \mathbf{L}\}$ with $\mathbf{Y}_v = \{y\}$, $\mathbf{L} = \{\ell\}$, see Table 2a & (b) for some examples.[3] Each ℓ takes a finite number of integers to denote one of several labels for tasks of pattern classification, choice decision, and clustering analyses, etc, while each y is a vector that acts as an inner coding or a cause for observations. Moreover, $q(\mathbf{Y}_v|\theta_y)$ describes the structure dependence among a set of values that $\mathbf{Y}_v$ may take. Second, $q(\mathbf{X}|\mathbf{Y}_v, \mathbf{L}, \theta_{x|y})$ describes the knowledge about the dependence relation from inner representation to observation. The simplest and widely studied example is Gaussian based linear regression, see Table 2c for some examples. Third, in addition to these structures, the knowledge is also represented by Θ jointly, which may be further confined by a background knowledge via a priori structure $q(\Theta|\Xi)$ with a unknown parameter set Ξ_q, for which readers are referred to several choices discussed in [52].

The Yang machine $p(\mathbf{X}, \mathbf{Y}|\Theta_p)$ consists of $p(\mathbf{X}|\theta_x)$ as the input to the system and $p(\mathbf{R}|\mathbf{X})$ that takes the inverse mapping roles in Fig. 1b and c. Performing the roles best and fast are two purposes that compete each other, and a structure that best trades off the two purposes is considered for $p(\mathbf{R}|\mathbf{X})$. An analogy of this Ying Yang system to the ancient Ying-Yang philosophy motivates to determine the unknowns under a best harmony principle, which is mathematically implemented by maximizing the following harmony measure

$$\max_{\{\mathbf{k},\ \Xi,\ p(\Theta|\mathbf{X},\Xi)\}} H(p\|q, \mathbf{k}, \Xi), \ \ H(p\|q, \mathbf{k}, \Xi) = \int p(\mathbf{R}|\mathbf{X})p(\mathbf{X}) \ln [q(\mathbf{X}|\mathbf{R})q(\mathbf{R})] d\mathbf{X}d\mathbf{R}$$
$$= \int p(\Theta|\mathbf{X}, \Xi) H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi) d\Theta, \quad (20)$$
$$H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi) = \sum_L \int p(L|\mathbf{X}, \theta_{y|x}) p(\mathbf{Y}_v|\mathbf{X}, L, \theta_{y|x}) p(\mathbf{X}|\theta_x)$$
$$\times \ln [q(\mathbf{X}|\mathbf{Y}_v, L, \theta_{x|y}) q(\mathbf{Y}_v|L, \theta_y) q(L|\theta_L) q(\Theta|\Xi_q)] d\mathbf{Y}_v d\mathbf{X}.$$

Maximizing $H(p\|q)$ forces $q(\mathbf{X}|\mathbf{R})q(\mathbf{R})$ to match $p(\mathbf{R}|\mathbf{X})p(\mathbf{X})$. In other words, $q(\mathbf{X}|\mathbf{R})q(\mathbf{R})$ attempts to describe the data $p(\mathbf{X})$ in help of $p(\mathbf{R}|\mathbf{X})$, which actually uses $q(\mathbf{X})$ in the next equation Eq. 21 to fit $p(\mathbf{X})$ not in a maximum likelihood sense but with a promising model selection nature. Due to a finite size of samples $\mathcal{X}_N = \{x_t\}_{t=1}^N$ and structural constraint of $p(\mathbf{R}|\mathbf{X})$, this matching aims at but may not really reach a perfect matching $p(\mathbf{R}|\mathbf{X})p(\mathbf{X}) = q(\mathbf{X}|\mathbf{R})q(\mathbf{R})$. Still we get a trend at this equality by which $H(p\|q)$ becomes the negative entropy that describes the complexity of system, and thus its further maximization is actually minimizing the complexity of system, which consequently provides a model selection nature on $\mathbf{k}$.

This model selection nature can also be observed on a differential level from a updating flow for maximizing $H(p\|q, \mathbf{k}, \Xi)$ via $dH(p\|q, \mathbf{k}, \Xi) = \int [p(\mathbf{R}|\mathbf{X})dL(\mathbf{X}, \mathbf{R}) + L(\mathbf{X}, \mathbf{R})dp(\mathbf{R}|\mathbf{X})]\ p(\mathbf{X})d\mathbf{X}d\mathbf{R}$ with $L(\mathbf{X}, \mathbf{R}) = \ln [q(\mathbf{X}|\mathbf{R})q(\mathbf{R})]$. Consider a Bayesian structure

$$p(\mathbf{R}|\mathbf{X}) = \frac{q(\mathbf{X}|\mathbf{R})q(\mathbf{R})}{q(\mathbf{X})}, \quad q(\mathbf{X}) = \int q(\mathbf{X}|\mathbf{R})q(\mathbf{R})d\mathbf{R}, \quad (21)$$

the first term of $dH(p\|q, \mathbf{k}, \Xi)$ actually leads to the updating flow of the M step in the EM algorithm for the maximum likelihood learning [17], i.e., the gradient flow $dL(\mathbf{X}, \mathbf{R})$ under

[3] y is simply denoted as y wherever it does not cause a confusion.

all possible choices of **R** is integrated via the weighting of $p(\mathbf{R}|\mathbf{X})$. This updating flow is modified by the second term of $dH(p\|q, \mathbf{k}, \Xi)$ in a same format but with $p(\mathbf{R}|\mathbf{X})dL(\mathbf{X}, \mathbf{R})$ replaced by $\delta_L(\mathbf{R}, \mathbf{X})dL(\mathbf{X}, \mathbf{R})$. That is, we have

$$\begin{aligned}&\delta_L(\mathbf{R}, \mathbf{X}) = L(\mathbf{X}, \mathbf{R}) - \textstyle\int p(\mathbf{R}|\mathbf{X})L(\mathbf{X}, \mathbf{R})d\mathbf{R},\\ &dH(p\|q, \mathbf{k}, \Xi) = \textstyle\int [p(\mathbf{R}|\mathbf{X})[1 + \delta_L(\mathbf{R}, \mathbf{X})]dL(\mathbf{X}, \mathbf{R})p(\mathbf{X})d\mathbf{X}d\mathbf{R}.\end{aligned} \tag{22}$$

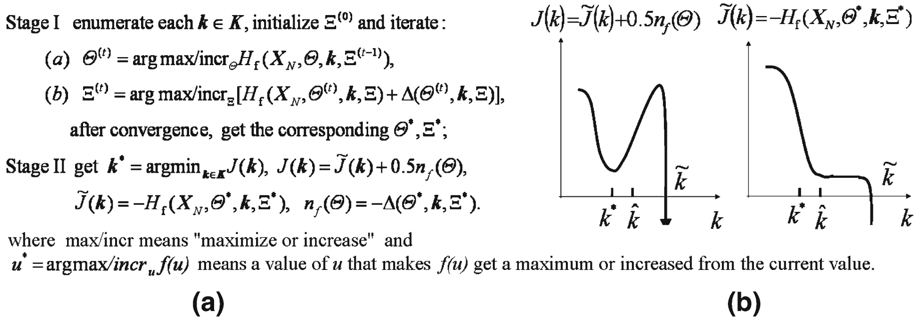
Noticing that $L(\mathbf{X}, \mathbf{R})$ describes the fitness of an inner representation **R** on the observation **X**, we observe that $\delta_L(\mathbf{R}, \mathbf{X})$ indicates whether the considered **R** fits **X** better than the average of all the possible choices of **R**. Each gradient flow $dL(\mathbf{X}, \mathbf{R})$ is integrated via weighting not just by $p(\mathbf{R}|\mathbf{X})$ but also by a modification of a relative fitness measure $1 + \delta_L(\mathbf{R}, \mathbf{X})$. If $\delta_L(\mathbf{R}, \mathbf{X}) > 0$, updating goes along the same direction of the EM learning even with an increased strength. If $0 > \delta_L(\mathbf{R}, \mathbf{X}) > -1$, i.e., the fitness is worse than the average and the current **R** is doubtful, updating still goes along the same direction of the EM learning but with a reduced strength. When $-1 > \delta_L(\mathbf{R}, \mathbf{X})$, updating reverses the direction of the EM learning and actually becomes de-learning. Therefore, $\delta_L(\mathbf{R}, \mathbf{X})$ provides a mechanism to seek appropriate inner representations for **R** and thus the corresponding complexity **k**. Readers are referred to the end of Sect. 3.3 for a further insight on how this mechanism relates to and improves RPCL learning with no need for pre-specifying a de-learning strength.

In implementation, it follows from the local convexity based derivation in the next subsection that $H(p\|q, \mathbf{k}, \Xi)$ in Eq. 20 can be approximately turned into the following format:

$$H(p\|q, \mathbf{k}, \Xi) = H_f(\mathcal{X}_N, \Theta^*, \mathbf{k}, \Xi) + \Delta(\Theta^*, \mathbf{k}, \Xi), \quad \Theta^* = \max_{\Theta} H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi), \tag{23}$$

where $\Delta(\Theta^*, \mathbf{k}, \Xi)$ either involves no integral over Θ or an integral over a subset of Θ that is analytically solvable. Thus, a best Ying Yang harmony by maximizing $H(p\|q, \mathbf{k}, \Xi)$ can be made via a two stage implementation in Fig. 5a, which is a process of Type-3 optimization nested with a series of Type-2 optimizations implemented at Stage I.

Though it is difficult to precisely define, the scale **k** of an entire system is featured by the scale or complexity for representing R, which is roughly regarded as consisting of the scale $\mathbf{k}_Y$ for representing Y. Actually, the model selection problem in many typical learning tasks [45,48] can be reformulated into a BYY system for selecting merely this $\mathbf{k}_Y$ part. Interestingly, the $\mathbf{k}_Y$ part associates with a subset $\tilde{\theta}_y \subset \theta_y$ of parameters in $q(\mathbf{Y}|\theta_y)$ in a sense that a parameter $\vartheta \in \tilde{\theta}_y$ becoming zero indicates that its associated contribution to $\mathbf{k}_Y$ can be discarded, and thus $\mathbf{k}_Y$ effectively reduces by one. The contribution of such a parameter $\vartheta \to 0$ to $H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi)$ is either 0 or $-\infty$, and a number of such parameters make $\tilde{J}(\mathbf{k})$ in Fig. 5b either get a flat (i.e., no change) range $[\hat{\mathbf{k}}, \tilde{\mathbf{k}}]$ or tend to $-\infty$ beyond $\tilde{\mathbf{k}}$ (i.e., $H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi)$



Stage I enumerate each $\boldsymbol{k} \in \boldsymbol{K}$, initialize $\Xi^{(0)}$ and iterate:

(*a*) $\Theta^{(t)} = \arg\max/\mathrm{incr}_{\Theta} H_f(X_N, \Theta, \boldsymbol{k}, \Xi^{(t-1)})$,

(*b*) $\Xi^{(t)} = \arg\max/\mathrm{incr}_{\Xi}[H_f(X_N, \Theta^{(t)}, \boldsymbol{k}, \Xi) + \Delta(\Theta^{(t)}, \boldsymbol{k}, \Xi)]$,

after convergence, get the corresponding Θ^*, Ξ^*;

Stage II get $\boldsymbol{k}^* = \mathrm{argmin}_{\boldsymbol{k}\in\boldsymbol{K}} J(\boldsymbol{k})$, $J(\boldsymbol{k}) = \tilde{J}(\boldsymbol{k}) + 0.5 n_f(\Theta)$,

$\tilde{J}(\boldsymbol{k}) = -H_f(X_N, \Theta^*, \boldsymbol{k}, \Xi^*)$, $n_f(\Theta) = -\Delta(\Theta^*, \boldsymbol{k}, \Xi^*)$.

where max/incr means "maximize or increase" and

$u^* = \mathrm{argmax}/incr_u f(u)$ means a value of u that makes $f(u)$ get a maximum or increased from the current value.

$J(k) = \tilde{J}(k) + 0.5 n_f(\Theta)$
$\tilde{J}(k) = -H_f(X_N, \Theta^*, k, \Xi^*)$
$\tilde{k}$
k^*
$\hat{k}$
k
$\tilde{k}$
k^*
$\hat{k}$
k
(a)
(b)

Fig. 5 Implementation of Bayesian Ying-Yang Learning

becomes singular). Initially, each parameter $\vartheta \in \tilde{\theta}_y$ is set at a value away from 0. Promisingly, a gradient based updating flow that pushes $H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi)$ to increase will yield a force that pushes each parameter in $\tilde{\theta}_y$ towards 0. If one is really pushed to zero, its associated contribution is regarded as extra and thus can be discarded. In such a way, as long as $\mathbf{k}$ is initialized at one big enough value, $\hat{\mathbf{k}}$ in Fig. 5b can be determined as an upper bound estimate of $\mathbf{k}^*$ during parameter learning on Ξ^* and Θ^* by only implementing Stage I in Fig. 5a, which can significantly reduce the computational cost needed for a two stage implementation [52]. However, the performance of this automatic model selection will deteriorate as the sample size N reduces. In such a case, we implement both the two stages in Fig. 5a with a computational cost similar to those conventional two stage implementations of model selection. Still differently, the contribution of $\mathbf{k}_Y$ to $H(p\|q, \mathbf{k}, \Xi)$ is usually able to be addressed more accurately and thus its corresponding $J(\mathbf{k})$ provides an improvement over those typical model selection criteria, though the contribution featured by the rest part of $\mathbf{k}$ is still roughly estimated along a line similar to those typical criteria by the number $n_f(\Theta)$ of free parameters in Θ.

3.2 Nested optimizations: local convexity based learning and local information conservation

With $p(\mathbf{X}|\theta_x)$ given empirically from $\mathcal{X}_N$, i.e., $\delta(\mathbf{X} - \mathcal{X}_N)$, it follows from Eq. 20 that

$$\begin{aligned}
H_f(\mathcal{X}_N, \Theta, \mathbf{k}, \Xi) &= \sum_L \int p(L|\mathcal{X}_N, \theta_{y|x}) p(\mathbf{Y}_v|\mathcal{X}_N, L, \theta_{y|x}) \\
&\quad \times \mathcal{L}_L(\mathcal{X}_N, \mathbf{Y}, \Theta_q) d\mathbf{Y}_v - Z(\Theta|\Xi_q), \\
\mathcal{L}_L(\mathcal{X}_N, \mathbf{Y}_v, \Theta_q) &= \ln\left[q(\mathcal{X}_N|\mathbf{Y}_v, L, \theta_{x|y}) q(\mathbf{Y}_v|L, \theta_y) q(L|\theta_L)\right], \\
Z(\Theta|\Xi_q) &= -\ln q(\Theta|\Xi_q).
\end{aligned} \tag{24}$$

There still remains an integral over $\mathbf{Y}_v$. Maximizing $H(p\|q)$ with respect to a $p(\mathbf{Y}_v|\mathbf{X}, L, \theta_{y|x})$ that is free of structure leads to

$$\begin{aligned}
&p(\mathbf{Y}_v|\mathbf{X}, L, \theta_{y|x}) = \delta(\mathbf{Y}_v - \mathbf{Y}^*_{vL}(\Theta_q)), \quad \mathbf{Y}^*_{vL}(\Theta_q) = \max_{\mathbf{Y}_v} \mathcal{L}_L(\mathcal{X}_N, \mathbf{Y}_v, \Theta_q), \\
&H_f(\mathcal{X}_N, \Theta, \mathbf{k}, \Xi) = \sum_L p(L|\mathcal{X}_N, \theta_{y|x}) \mathcal{L}_L(\mathcal{X}_N, \mathbf{Y}^*_{vL}(\Theta_q), \Theta_q) - Z(\Theta|\Xi_q).
\end{aligned} \tag{25}$$

The computational difficulty incurred by the integral over $\mathbf{Y}_v$ has been avoided. As a result, learning Θ is nested with a series of Type-1 optimization for $\mathbf{Y}^*_{vL}$.

However, it also incurs two problems. First, the above $\mathbf{Y}^*_{vL}(\Theta_q)$ may not have a differentiable expression with respect to Θ_q, or it even has no analytical expression. Thus, a gradient based algorithm for $\max_\Theta H(p\|q, \Theta)$ can not take the relation $\mathbf{Y}^*_{vL}(\Theta_q)$ in consideration, which makes learning fragile to local optimal performance. Second, the mapping from a set $\mathcal{X}_N$ of random samples to the corresponding inner representations is probabilistic while $\delta(\mathbf{Y}_v - \mathbf{Y}^*_{vL}(\Theta_q))$ can not take this uncertainty in consideration, since it only takes over from the Ying machine the information of the first order statistics. To improve, we consider a structure of $p(\mathbf{Y}_v|\mathbf{X}, L, \theta_{y|x})$ in help of local convexity for a best Ying Yang harmony in the front layer via approximately considering the second order statistics.

Considering a Taylor expansion of $Q(\xi)$ around $\bar{\xi} = \int \xi p(\xi) d\xi$ up to the second order, we approximately have

$$\int p(\xi)Q(\xi)d\xi \approx Q(\bar{\xi}) + 0.5Tr[\Sigma H_Q(\bar{\xi})], \quad H_Q(\xi) = \nabla^2_{\xi} Q(\xi),$$
$$\Sigma = \int p(\xi)(\xi - \bar{\xi})(\xi - \bar{\xi})^T d\xi, \tag{26}$$

where $\nabla_u f(u) = \partial f(u)/\partial u$ and $\nabla^2_u f(u) = \partial^2 f(u)/\partial u \partial u^T$.

With $\mathbf{Y}_v$ as ξ and $\mathcal{L}(\mathcal{X}_N, \mathbf{Y}_v, \Theta_q)$ as $Q(\xi)$, from Eq. 24 we approximately have

$$\begin{aligned} &H_f(\mathcal{X}_N, L, \Theta, \mathbf{k}, \Xi) \approx \sum_L [p(L|\mathcal{X}_N, \theta_{y|x})\mathcal{L}_L(\mathcal{X}_N, \bar{\mathbf{Y}}_{vL}(\theta_{y|x}), \Theta_q) \\ &\quad + 0.5 d_Y(L, \Theta)] - Z(\Theta|\Xi_q), \\ &d_Y(L, \Theta) = Tr[\Gamma_L(\theta_{y|x}) H_L(\bar{\mathbf{Y}}_{vL}(\theta_{y|x}), \Theta_q)], \; H_L(\mathbf{Y}_v, \Theta_q) = \nabla^2_{\mathbf{Y}_v} \mathcal{L}_L(\mathcal{X}_N, \mathbf{Y}_v, \Theta_q), \\ &\bar{\mathbf{Y}}_{vL}(\theta_{y|x}) = \int \mathbf{Y}_v p(\mathbf{Y}_v|\mathcal{X}_N, L, \theta_{y|x}) d\mathbf{Y}_v, \\ &\Gamma_L(\theta_{y|x}) = \int [\mathbf{Y}_v - \bar{\mathbf{Y}}_{vL}(\theta_{y|x})][\mathbf{Y}_v - \bar{\mathbf{Y}}_{vL}(\theta_{y|x})]^T p(\mathbf{Y}_v|\mathcal{X}_N, L, \theta_{y|x}) d\mathbf{Y}_v, \end{aligned} \tag{27}$$

which is handled via $\bar{\mathbf{Y}}_{vL}(\theta_{y|x})$ and $\Gamma_L(\theta_{y|x})$, with no need of a structure for $p(\mathbf{Y}_v|\mathcal{X}_N, L, \theta_{y|x})$.

The integral for $\bar{\mathbf{Y}}(\theta_{y|x})$ can be removed via a differentiable parametric function $\bar{\mathbf{Y}}(\mathcal{X}, \theta_{y|x})$, e.g., a linear function of $\mathcal{X}$ followed by a nonlinear scale function in a variable by variable manner. One example will be introduced later in Table 3.

In some applications, e.g., the Bernoulli case in Table 3(b), $\Gamma_L(\theta_{y|x})$ is simply a parametric function

$$\Gamma_L(\theta_{y|x}) = \Gamma_L[\bar{\mathbf{Y}}(\mathcal{X}, \theta_{y|x})] \tag{28}$$

that is specified directly by a given parametric function $\bar{\mathbf{Y}}(\mathcal{X}, \theta_{y|x})$. In general, instead of explicitly computing $\Gamma_L(\theta_{y|x})$, we consider $d_Y(L, \Theta)$ in Eq. 27 by the following instantaneous estimate:

$$d_Y(L, \Theta) = e^T_{vL}(\Theta) H_L(\bar{\mathbf{Y}}(\mathcal{X}, \theta_{y|x}), \Theta_q) e_{vL}(\Theta), \; e_{vL}(\Theta) = \mathbf{Y}^*_{vL}(\Theta_q) - \bar{\mathbf{Y}}_{vL}(\theta_{y|x}). \tag{29}$$

with $\mathbf{Y}^*_{vL}(\Theta_q)$ obtained by Eq. 25 via Type-1 optimization.

Alternatively, we can also get Eq. 26 with $\bar{\xi}$ replaced by $\xi^* = arg \max Q(\xi)$. Similar to Eq. 27, it follows from $\mathbf{Y}_v - \mathbf{Y}^*_{vL}(\Theta_q) = \mathbf{Y}_v - \bar{\mathbf{Y}}_{vL}(\theta_{y|x}) + \bar{\mathbf{Y}}_{vL}(\theta_{y|x}) - \mathbf{Y}^*_{vL}(\Theta_q)$ that we get

$$\begin{aligned} &H_f(\mathcal{X}_N, L, \Theta, \mathbf{k}, \Xi) \approx \sum_L p(L|\mathcal{X}_N, \theta_{y|x})\{\mathcal{L}_L(\mathcal{X}_N, \mathbf{Y}^*_{vL}(\theta_{y|x}), \Theta_q) + 0.5 d^*_Y(L, \Theta) + 0.5 e^T_{vL} \\ &\quad \times (\Theta) H_L(\mathbf{Y}^*_{vL}(\Theta_q), \Theta_q) e_{vL}(\Theta)\}, \; 0.5 d^*_Y(L, \Theta) = Tr[\Gamma_L(\theta_{y|x}) H_L(\mathbf{Y}^*_{vL}(\Theta_q), \Theta_q)], \end{aligned} \tag{30}$$

where $d^*_Y(L, \Theta) = d_Y = Tr[I_Y]$ becomes the dimension of $\mathbf{Y}_v$ under a local information conservation constraint that $\Gamma_L(\theta_{y|x}) = -H^{-1}_L(\mathbf{Y}^*_{vL}, \Theta_q)$. Moreover, L is discrete, instead of basing on Hessian matrix this local information conservation becomes simply $p(L|X_N, \theta_{y|x}) \propto \ln L_L(X_N, \mathbf{Y}^*_v \theta_{y|x})$, i.e., being proportional to the likelihood described by the Ying machine, referring Table 3 (its 2nd line) for a detailed example.

Though the above studies come from Eq. 26 that involves the differentiable concept that is applicable to Θ of real parameters and $\mathbf{Y}_v$ of real variables. It deserves a particular remark that these studies are still applicable to the cases with $\mathbf{Y}_v$ consisting of vectors in binary variables, simply regarding $\mathbf{Y}_v$ as real when considering $\partial/\partial \mathbf{Y}_v$ and $\partial^2/\partial \mathbf{Y}_v \partial \mathbf{Y}^T_v$. Strict mathematical proofs can also be obtained from a different perspective [52].

Next we return to Eq. 20 with $p(\mathbf{X}|\theta_x) = \delta(\mathbf{X} - \mathcal{X}_N)$, i.e., $\int p(\Theta|\mathcal{X}_N, \Xi) H_f(\mathcal{X}_N, \Theta, \mathbf{k}, \Xi) d\Theta$. Regarding Θ as ξ and $H_f(\mathcal{X}_N, \Theta, \mathbf{k}, \Xi)$ as $Q(\xi)$, from Eq. 26 we approximately get

Table 3 An adaptive learning algorithm

$$H_t(\Theta_\ell) = L_\ell(x_t, \eta_{t,\ell}(\theta_{y|x,\ell}), \Theta_\ell) - \frac{1}{2}Tr[h_x^2\Sigma_j^{-1} + \prod_\ell^{yy}\Gamma_{t,\ell}] + \frac{1}{N}\ln q(h|X_N), L_\ell(x, y, \Theta_\ell) = \ln[\alpha_j G(x|A_j y + \mu_j, \Sigma_j)q(y|\ell, \theta_{y,\ell})\alpha_\ell]$$

$$p(\ell|x_t) = \frac{e^{\pi_t(\Theta_\ell)}}{\sum_{j=1}^k e^{\pi_t(\Theta_j)}}, \pi_t(\Theta_\ell) = \begin{cases} \ln[\alpha_\ell G(x_t|\mu_\ell, A_\ell\Lambda_\ell A_\ell^T + \Sigma_\ell)], & \text{for } i_L = 0, \\ L_\ell(x_t, \eta_{t,\ell}(\theta_{y|x,\ell}), \Theta_\ell), & \text{for } i_L = 1. \end{cases} \quad \delta_{j,\ell} = \begin{cases} 1 & \text{if } j = \ell, \\ 0, & \text{otherwise.} \end{cases}$$

$$d\sum_{j=1}^k p(j|x_t)H_t(\Theta_j) = \sum_{j=1}^k [p(j|x_t)dH_t(\Theta_j) + H_t(\Theta_j)dp(j|x_t)], dp(j|x_t) = p(j|x_t)\sum_{\ell=1}^k[\delta_{j,\ell} - p(\ell|x_t)]d\pi_t(\Theta_\ell).$$

Type	$q(y\|\ell, \theta_{y,\ell})$	A_ℓ	$\eta(x_t, \theta_{y\|x,\ell})$	$\prod_\ell^{\mathrm{yy}}$	$\Gamma_{t,j}$
$i_Y = 1$	Gaussian $G(y\|0, \Lambda_\ell)$	$A_\ell^{\mathrm{T}}A_\ell = I$	$W_\ell x_t + w_\ell$	Λ_ℓ^{-1}	$\varepsilon_\ell(x_t)\varepsilon_\ell^T(x_t), \varepsilon_\ell(x_t) = y_{t,\ell}^* - \eta(x_t, \theta_{y\|x,\ell})$
$i_Y = 0$	Binary $\prod_{\mathrm{i}} q_j^{(\mathrm{i})\mathrm{y}^{(\mathrm{i})}}(1 - q_{\mathrm{j}}^{(\mathrm{i})})^{1-\mathrm{y}^{(\mathrm{i})}}$	In general	$s(W_\ell x_t + w_\ell)$	0	$diag[s(\bar{y}_{t,\ell}^{(i)})(1 - s(\bar{y}_{t,\ell}^{(i)}))], \bar{y}_{t,\ell} = W_\ell x_t + w_\ell$

where $y_{t,\ell}^* = \arg\max_y L(x, y, \Theta_\ell), s(u) = vec[s(u^{(i)})], s(r) = 1/(1 + e^{-r}), vec[u^{(j)}]$ is a vector with $u^{(i)}$ being the ith element, and diag $[u^{(i)}]$ is a diagonal matrix with $u^{(i)}$ being the ith diagonal element

The algorithm consists of iterating the Yang step and Ying step until converged:

Yang STEP Get $y_{t,\ell}^* = (A_j^T\Sigma_j^{-1}A_j + \Lambda_j^{-1})^{-1}A_j^T\Sigma_j^{-1}(x_t - \mu_j)$ and $p_{j,t} = p_t(j|\Theta^{old})$, let $\delta h_{j,t} = H_t(\Theta_j^{old}) - \sum_{\ell=1}^k p_{j,t}H_t(\Theta_\ell^{old})$
(*Note*: $\delta h_{j,t}$ is a simplified case of $\delta_{\mathrm{L}}(R, X)$ in Eq. 22, especially Eq. 52)

Ying STEP In the following, $\Delta\theta \propto \mathrm{g}_\theta$ means make a updating along the direction of g_θ with a small step size:

(a) $\Delta\mathrm{c} \propto (I - \alpha 1^T)g_\alpha$, with $\mathrm{c} = vec[c_\ell], \alpha = vec[\alpha_\ell], 1 = vec[1], g_\alpha = diag[p_{\ell,t}(1 + \delta h_{\ell,t})], \alpha_j^{new} = e^{c_j^{old} + \Delta c_j} \Big/ \sum_\ell e^{c_\ell^{old} + \Delta c_\ell}$.

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

(b) For Type $i_Y = 1, \Lambda_j = D_j D_j^T, D_j = diag[d_j^{(i)}], D_j^{new} = D_j^{old} + \Delta D_j, \Delta\mathrm{D}_j \propto p_{j,t}\Lambda_j^{old}G_{\Lambda_j}\Lambda_j^{old}D_j^{old}, G_{\Lambda_j} = \Lambda_j^{-old}diag[(1 + i_L\delta h_{j,t})(y_{j,t}^* y_{j,t}^{*T} - \Lambda_j^{old}) + \Gamma_{t,j}]\Lambda_j^{-old} + (1 - i_L)\delta h_{j,t}diag[A_j^{oldT}\delta\Sigma_j^x A_j^{old}]$ with $\Gamma_{t,j}$ given as in the above table and $\delta\Sigma_j^x = (\Sigma_j^{xold})^{-1}(e_{j,t}^x e_{j,t}^{xT} - \Sigma_j^{xold})(\Sigma_j^{xold})^{-1}, e_{j,t}^x = x_t - \mu_j^{old}, \Sigma_j^{xold} = \Sigma_j^{old} + A_j^{old}\Lambda_j^{old}A_j^{oldT}$.

For Type $i_Y = 0, q_j^{(i)} = 1/(1 + e^{-\beta_j^{(i)new}}), \beta_j^{new} = \beta_j^{old} + \delta\beta_j, \delta\beta_j \propto p_{j,t}(1 + \delta h_{t,j})[s(\bar{y}_t) - q_j]$. Let $\lambda_j^{(i)} = q_j^{(i)}(1 - q_j^{(i)})$. If $\lambda_j^{(i)} \to 0$, discard the dimension $y_j^{(i)}$ and its corresponding subset of parameters in Θ_j

Table 3 continued

(c) $\mu_j^{new} = \mu_j^{old} + \Delta\mu_j, \Delta\mu_j \propto p_{j,t}\{(1+i_L\delta h_{j,t})e_t^{x|y} + (1-i_L)\delta h_{j,t}\Sigma_j^{old}(\Sigma_j^{xold})^{-1}\ e_t^x\}, e_t^{x|y} = x_t - \hat{x}_{j,t}, \hat{x}_{j,t} = A_j^{old}y_{j,t}^* + \mu_j^{old}, A_j^{new} = A_j^{old} + \Delta A_j, \Delta A_j \propto p_{j,t}\Sigma_j^{old}G_{A_j}(I - A_j^{old}A_j^{old\mathrm{T}})$, (for$A_j^{\mathrm{T}}A_j = I$). $\Sigma_j^{old}G_{A_j} = \{(1+i_L\delta h_{j,t})(e_t^{x|y}y_{j,t}^{*\mathrm{T}} - A_j^{old}\Gamma_{t,j}) + (1-i_L)\delta h_{j,t}\Sigma_j^{old}\delta\Sigma_j^x A_j^{old}\Lambda_j^{old}\}$, with $\Gamma_{t,j}$ given as in the above table. $\Sigma_j^{new} = S_jS_j^{\mathrm{T}}, S_j^{new} = S_j^{old} + \Delta S_j, \Delta S_j \propto p_{j,t}G_{\Sigma_j}S_j^{old} = p_{j,t}\Sigma_j^{old}G_{\Sigma_j}\Sigma_j^{old}S_j^{old-\mathrm{T}}, \Sigma_j^{old}G_{\Sigma_j}\Sigma_j^{old} = (1+i_L\delta h_{j,t})\Sigma_j^{-old}(e_{j,t}^{x|y}e_{j,t}^{x|yT} - \Sigma_j^{old} + A_j^{old}\Gamma_{t,j}A_j^{oldT} + 0.5h^2I)\Sigma_j^{-old} + (1-i_L)\delta h_{j,t}\delta\Sigma_j^x;$

(d) $W_j^{new} = W_j^{old} + \Delta W_j, w_j^{new} = w_j^{old} + \Delta w_j$ with $\Delta W_j \propto p_{j,t}g_{y,j}x_t^T, \Delta w_j \propto p_{j,t}g_{y,j}, g_{y,j} = g_{y,j}^{\pi} + g_{y,j}^R$

For Type $i_Y = 1, g_{y,j}^{\pi} = p_{j,t}(1+\delta h_{j,t})(A_j^T\Sigma_j^{-1}e_t^{x|y} + \Lambda_j^{-1}y), e_t^{x|y} = x_t - A_j^{old}\bar{y}_t - \mu_j^{old}, g_{y,j}^R = (A_j^T\Sigma_j^{-1}A_j + \Lambda_j^{-1})\varepsilon_\ell(x_t).$

For Type $i_Y = 0, g_{y,j}^{\pi} = p_{j,t}(1+\delta h_{j,t})D_s(A_j^T\Sigma_j^{-1}e_t^{x|y} + \delta_{q_j}), e_t^{x|y} = x_t - A_j^{old}s(\bar{y}_t) - \mu_j^{old},$

$g_{y\cdot j}^R = p_{j,t}D_s(diag[s(\bar{y}_t^{(1)}), \ldots, s(\bar{y}_t^{(m_j)})] - 0.5I)A_j^T\Sigma_j^{-1}A_j1.$

where $\bar{y}_t = W_jx_t + w_j, \delta_{q_j} = [\delta_{q_j}^{(1)}, \ldots, \delta_{q_j}^{(1)}]^T, \delta_{q_j}^{(i)} = ln[q_j^{(i)}\Big/(1-q_j^{(i)})], D_s = diag[s'(\bar{y}_t^{(1)}), \ldots, s'(\bar{y}_t^{(m_j)})], s'(r) = ds(r)/dr.$

(e) $h^{new} = h^{old} + \Delta h$ with $\Delta h \propto \{\frac{1}{N}q'(h|X_N) - h\sum_{j=1}^k p_{j,t}Tr[\Sigma_j^{-1}]\}$, where $f'(r) = \frac{df(r)}{dr}$.

$$H(p\|q,\mathbf{k},\Xi)=H_f(\mathcal{X}_N,\bar{\Theta},\mathbf{k},\Xi)+0.5Tr[\Sigma H_H(\bar{\Theta})],\quad H_H(\Theta)=\nabla^2_{\Theta}H_f(\mathcal{X}_N,\Theta,\mathbf{k},\Xi),$$
$$\bar{\Theta}=\int\Theta p(\Theta|\mathcal{X}_N)d\Theta,\quad \Sigma=\int(\Theta-\bar{\Theta})(\Theta-\bar{\Theta})^T p(\Theta|\mathcal{X}_N)d\Theta. \tag{31}$$

Again, we do not need to specify a structure for $p(\Theta|\mathcal{X}_N)$. In fact, getting to know $p(\Theta|\mathcal{X}_N)$ is the aim of learning. Instead, for any learning process $\mathcal{X}_N \to \Theta$, the resulted estimate $\Theta*$ can be regarded as the above $\bar{\Theta}$ with Σ measuring the randomness of this estimation due to a finite size of samples in $\mathcal{X}_N$. One typical example is to let $\bar{\Theta}=\Theta^*=\max_{\Theta} H_f(\mathbf{X},\Theta,\mathbf{k},\Xi)$ from which we get Eq. 23 with $\Delta(\Theta^*,\mathbf{k},\Xi)=0.5Tr[\Sigma H_H(\Theta^*)]$ in two choices. One is considering $\Sigma=H_H(\Theta^*)^{-1}$ and thus getting

$$\Delta(\Theta^*,\mathbf{k},\Xi)=-0.5d_{\mathbf{k}},\quad d_{\mathbf{k}}=n_f(\Theta), \tag{32}$$

where $n_f(\Theta)$ is the number of free parameters in Θ [44,49,50]. The other is considering $\bar{\Theta}$ and Θ^* approximately by $\Theta^{(t)}$, $\Theta^{(t-1)}$ obtained in Fig. 5a, from which we get

$$\Delta(\Theta^{(t)},\mathbf{k},\Xi)=-0.5d_{\mathbf{k}},\quad d_{\mathbf{k}}=n_f(\Theta)+(\Theta^{(t)}-\Theta^{(t-1)})^T H_H(\Theta^{(t)})(\Theta^{(t)}-\Theta^{(t-1)}). \tag{33}$$

We further proceed to consider $p(\mathbf{X}|\theta_x)$ estimated from $\mathcal{X}_N=\{x_t\}_{t=1}^N$ with a unknown scalar parameter $\theta_x=h$. Replace $\mathbf{X}$ by $\mathbf{X}$, h and notice that only $\mathbf{X}$ relates to h, we have

$$p(\mathbf{X},h)=p(\mathbf{X}|h)p(h),\quad p(\mathbf{X}|h)=G(\mathbf{X}|\mathcal{X}_N,h^2I)=\prod_{t=1}^N G(x|x_t,h^2I)$$
$$p(\mathbf{R}|\mathbf{X},h)=p(\mathbf{R}|\mathbf{X}),\quad q(\mathbf{X},h|\mathbf{R})=q(h|\mathbf{X},\mathbf{R})q(\mathbf{X}|\mathbf{R}),\quad q(h|\mathbf{X},\mathbf{R})=q(h|\mathbf{X}), \tag{34}$$

with $q(\mathbf{R})$ remains unchanged. Put it into Eq. 20, we have

$$H(p\|q,\mathbf{k},\Xi)=\int p(h)p(\Theta|\mathbf{X},\Xi)H_f(\mathbf{X},\Theta,h,\mathbf{k},\Xi)d\Theta d\mathbf{X}dh \tag{35}$$

Maximizing $H(p\|q,\mathbf{k},\Xi)$ with a $p(h)$ that is free of constraint, we get

$$p(h)=\delta(h-h^*),\quad h^*=arg\max_h=arg\max_h H_f(\mathbf{X},\Theta,h,\mathbf{k},\Xi), \tag{36}$$
$$H_f(\mathbf{X},\Theta,h,\mathbf{k},\Xi)=\sum_L\int p(L|\mathbf{X},\theta_{y|x})p(\mathbf{Y}_v|\mathbf{X},L,\theta_{y|x})G(\mathbf{X}|\mathcal{X}_N,h^2I)$$
$$\times\ln[q(\mathbf{X}|\mathbf{Y}_v,L,\theta_{x|y})q(\mathbf{Y}_v|L,\theta_y)q(L|\theta_L)q(\Theta|\Xi_q)]d\mathbf{Y}_v d\mathbf{X}.$$

Similar to Eq. 26 we also have $\int G(\xi|\mu,\Sigma)Q(\xi)d\xi\approx Q(\xi)_{\xi=\mu}+0.5Tr[\Sigma\nabla^2_{\xi}Q(\xi)]_{\xi=\mu}$. Regarding $\mathbf{X}$ as ξ, and $\mathbf{X}$ as $G(\mathbf{X}|\mathcal{X}_N,h^2I)$ as $Q(\xi)$, $H_f(\mathbf{X},\Theta,h,\mathbf{k},\Xi)$ in Eq. 36 becomes

$$H_f(\mathbf{X},\Theta,h,\mathbf{k},\Xi)=H_f(\mathcal{X}_N,\Theta,\mathbf{k},\Xi)+0.5h^2Tr[\Sigma(\mathcal{X}_N)]-Z(h),$$
$$Z(h)=-\ln q(h|\mathcal{X}_N),\quad \Sigma(\mathbf{X})=\nabla^2_{\mathbf{X}}\sum_L p(L|\mathcal{X}_N,\theta_{y|x})\ln q(\mathbf{X}|\mathbf{Y}_v,L,\theta_{x|y}), \tag{37}$$

from which we modify the two stage implementation in Fig. 5a via replacing all the appearances of Θ with $\{\Theta,h\}$. With an appropriate h^* learned together with Θ^*, a considerable improvement can be obtained to reduce the deterioration caused by a small sample size N.

For a further insight via a detailed expression, we consider the cases that the elements of $\mathbf{X}=\{x\}$ are i.i.d. by Eq. 11 and thus have $q(\mathbf{X}|\mathbf{Y},\theta_{x|y})=\prod q(x|y,\ell,\theta_{x|y,\ell})$, $q(\mathbf{Y}|\theta_y)=\prod q(y,\ell|\theta_y)$, $q(y,\ell|\theta_y)=q(y|\ell,\theta_{y,\ell})\alpha_\ell$, $\alpha_\ell=q(\ell)$, $p(\mathbf{Y}|\mathbf{X},\theta_{y|x})=\prod p(y|x,\ell,\theta_{y|x,\ell})$ $p(\ell|x,\theta_{\ell|x})$. With $\Delta(\Theta^{(t)},\mathbf{k},\Xi)$ given by Eq. 33 and $H_f(\mathcal{X}_N,\Theta,\mathbf{k},\Xi)$ by $H_f(\mathcal{X}_N,\Theta,h,\mathbf{k},\Xi)$, further considering $q(\Theta|\Xi_q)=q(h)\prod_\ell q(\Theta_\ell|\Xi_q)$, it follows from Eq. 37 and Eq. 27 that we get

$$
\begin{aligned}
H_f(\mathcal{X}_N, h, \Theta, \mathbf{k}, \Xi) &= \sum_t \sum_\ell p(\ell|x_t)[H_t(\Theta_\ell) + N^{-1} \ln q(\Theta_\ell|\Xi_q)], \qquad (38)\\
H_t(\Theta_\ell) &= L_\ell(x_t, \eta_\ell(x_t, \theta_{y|x;\ell}), \Theta_\ell) + 0.5Tr[h^2 \textstyle\prod_\ell^x + \prod_\ell^y \Gamma_{t,\ell}]\\
&\quad + N^{-1} \ln q(h|\mathcal{X}_N),\\
L_\ell(x, y, \Theta_\ell) &= \ln [q(x|y, \ell, \theta_{x|y,\ell}) q(y|\ell, \theta_{y,\ell}) \alpha_\ell],\\
\textstyle\prod_\ell^y &= \nabla_y^2 \ln [q(x|y, \ell, \theta_{x|y,\ell}) q(y|\ell, \theta_{y,\ell})],\\
\textstyle\prod_\ell^x &= \nabla_x^2 \ln q(x|y, \ell, \theta_{x|y,\ell}),\\
\Gamma_{t,\ell} &= \begin{cases} \Gamma_\ell[\eta_\ell(x, \theta_{y|x,\ell})], & \text{for } i_\prod = 0 \text{ \& from Eq. 28},\\ \varepsilon_\ell(x_t)\varepsilon_\ell^T(x_t), & \text{for } i_\prod = 1 \text{ \& from Eq. 29}. \end{cases}\\
\varepsilon_\ell(x_t) &= y^*_{t,\ell} - \eta_\ell(x_t, \theta_{y|x,\ell}),\ q(x|\theta_\ell) = \int q(x|y, \ell, \theta_{x|y,\ell}) q(y|\ell, \theta_y) dy,\\
p(\ell|x_t) &= \frac{e^{-\pi_t(\Theta_\ell)}}{\sum_j e^{-\pi_t(\Theta_j)}},\ \pi_t(\Theta_\ell) = \begin{cases} \ln [\alpha_\ell q(x_t|\theta_\ell)], & \text{for } i_L = 0,\\ L_\ell(x_t, \eta_\ell(x_t, \theta_{y|x,\ell}), \Theta_\ell), & \text{for } i_L = 1, \end{cases}
\end{aligned}
$$

where $\eta_\ell(x_t, \theta_{y|x;\ell})$ is a parametric function, a typical example given in Table 3 is quasi-linear function that consists of a linear function of x_t followed by a nonlinear scale function in an element by element manner.

In the implementation as in Fig. 5a, Stage II is a discrete optimization. Both Stages I(a) & I(b) are featured by continuous optimization, based on $\nabla_\Theta H_f(\mathcal{X}_N, \Theta^{(t)}, \mathbf{k}, \Xi^{(t-1)})$ and $\nabla_\Xi[H_f(\mathcal{X}_N, \Theta^{(t)}, \mathbf{k}, \Xi) + \Delta(\Theta^{(t)}, \mathbf{k}, \Xi)]$. To get an insight on how Stage I(a) is implemented with an automatic model selection on $\mathbf{k}_Y$, we give an adaptive algorithm in Table 3 for the previously examples $q(x|y, \ell, \theta_{x|y})$ of Table 2c and $q(y|\theta_y) = q(y|\ell, \theta_{y,\ell})q(\ell)$ of Table 2a (3) plus Table 2b (1)&(2). The algorithm bases on getting the differential updating flow $d\sum_\ell p(\ell|x_t, \theta_{\ell|x})[H_t(\Theta_\ell) + N^{-1} \ln q(\Theta_\ell|\Xi_q)]$. For simplicity, we ignore a priori $q(\Theta_\ell|\Xi_q)$. Also, interested readers are further referred to algorithms for extensions of supervised learning tasks (e.g., function approximation, pattern recognition) and temporal modeling tasks [51].

In a summary, implementing the best harmony learning by Eq. 20 is a Type-3 optimization that involves an integral over Θ and an optimization on searching a $p(\Theta|\mathbf{X}, \Xi)$. With the help of local convexity, the problem is handled by searching $\{\Theta_k\}$ via a series of Type-2 optimizations that involve an integral over $\mathbf{Y}$ and an optimization on searching a $p(\mathbf{Y}_v|\mathbf{X}, L, \theta_{y|x})$. Again with the help of local convexity based local information conservation, the problem is handled by searching a series values of $\mathbf{Y}^*_{vL}$ via a series of Type-3 optimizations.

3.3 Best harmony, best matching, and related approaches

We start at observing how the best harmony learning degenerates as a BYY system degenerates to a conventional model $q(\mathbf{X}|\Theta)$. We consider $\mathbf{R} = \{\Theta\}$ without an inner representation part $\mathbf{Y}$, which leads us back to Fig. 1c and simplifies $H(p\|q, \mathbf{k}, \Xi)$ in Eq. 20 into

$$
H(p\|q) = \int p(\Theta|\mathbf{X}) p(\mathbf{X}) \ln [q(\mathbf{X}|\Theta) q(\Theta)] d\mathbf{X} d\Theta. \qquad (39)
$$

For $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$, maximizing $H(p\|q)$ with respect to a free $p(\Theta|\mathbf{X})$ leads to the MB type in Table 1, i.e., $\max_\Theta \ln [q(\mathcal{X}_N|\Theta)q(\Theta)]$, while $J(\mathbf{k})$ in Fig. 5 becomes

$$
\mathbf{k}^* = arg \min_{\mathbf{k}} J(\mathbf{k}), \quad J(\mathbf{k}) = -\max_\Theta \ln [q(\mathcal{X}_N|\Theta) q(\Theta)] + 0.5 d_{\mathbf{k}}, \qquad (40)
$$

which is a Bayesian learning based extension of AIC. For a non-informative $q(\Theta)$, it further degenerates to exactly AIC [1,2]. For a general case with $p(\mathbf{X}, h)$ by Eq. 34, it follows from Eq. 37 that Eq. 39 is extended into

$$H(p\|q) = \int p(h)p(\Theta|\mathcal{X}_N)H_h(p\|q,\Theta)d\Theta \approx \max_{\Theta,h} H_h(p\|q,\Theta) - 0.5d_{\mathbf{k}},$$
$$H_h(p\|q,\Theta) = \ln[q(\mathcal{X}_N|\Theta)q(\Theta)] + 0.5h^2Tr[\nabla^2_{\mathbf{X}} \ln q(\mathbf{X}|\Theta)] + \ln q(h|\mathcal{X}_N). \quad (41)$$

With $p(\Theta|\mathbf{X})$ in a given structure, the BYY harmony learning is different from the conventional Bayesian learning. E.g., we consider $p(\Theta|\mathbf{X})$ with the BI structure in Table 1 and rewrite Eq. 39 into $H(p\|q) = \int p(\Theta|\mathbf{X})p(\mathbf{X}) \ln p(\Theta|\mathbf{X})d\mathbf{X}d\Theta + \int p(\mathbf{X}) \ln q(\mathcal{X}|S)d\mathbf{X}$. Particularly, for $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$ it further becomes

$$H(p\|q) = \int p(\Theta|\mathcal{X}_N) \ln p(\Theta|\mathcal{X}_N)d\Theta + \ln q(\mathcal{X}_N|S). \quad (42)$$

The maximization of its second term is exactly the MI (marginal likelihood) choice in Table 1. As already discussed in Sect. 1, it has been previously studied under various names [14,16, 20,28]. Additionally, the first term in Eq. 42 is the negative entropy of $p(\Theta|\mathcal{X}_N)$ and its maximization is seeking an inverse inference $\mathcal{X}_N \to \Theta$ with a least uncertainty.

Also, we let the structure S in place of $\mathbf{R}$, and get a generalization of Eq. 39 as follows:

$$H(p\|q) = \sum_S p(S|\mathbf{X})p(\mathbf{X}) \ln[q(\mathbf{X}|S)q(S)]. \quad (43)$$

When $p(S|\mathbf{X})$ is free of structure, maximizing $H(p\|q)$ with respect to $p(S|\mathbf{X})$ leads to $\max_S \ln[q(\mathcal{X}_N|S)q(S)]$ for model selection, i.e., the BI choice in Table 1. In the special case that $q(S)$ is equal for each candidate S, it degenerates to $\max_S \ln q(\mathcal{X}_N|S)$, i.e., the ML choice in Table 1. Moreover, a generalized counterpart of Eq. 42 becomes

$$H(p\|q) = \sum_S p(S|\mathcal{X}_N) \ln p(S|\mathcal{X}_N) + \ln q(\mathcal{X}_N), \quad q(\mathcal{X}_N) = \sum_S q(\mathcal{X}_N|S)q(S).$$

For a BYY system, in addition to making the best harmony learning by Eq. 20, an alternative has also been proposed and studied in [37] under the name of Bayesian Kullback Ying Yang (BKYY) learning that performs the following *best matching* principle:

$$\begin{aligned}\min KL(p\|q), \quad KL(p\|q) &= \int p(\mathbf{R}|\mathbf{X})p(\mathbf{X}) \ln \frac{p(\mathbf{R}|\mathbf{X})p(\mathbf{X})}{q(\mathbf{X}|\mathbf{R})q(\mathbf{R})}d\mathbf{X}d\mathbf{R} \\ &= \int p(\Theta|\mathbf{X})\{\int p(\mathbf{Y}|\mathbf{X},\theta_{y|x})p(\mathbf{X}) \\ &\quad \times \ln \frac{p(\Theta|\mathbf{X})p(\mathbf{Y}|\mathbf{X},\theta_{y|x})p(\mathbf{X})}{q(\mathbf{X}|\mathbf{Y},\theta_{x|y})q(\mathbf{Y}|\theta_y)q(\Theta)}d\mathbf{X}d\mathbf{Y}\}d\Theta,\end{aligned} \quad (44)$$

which reaches to the best matching $KL(p\|q) = 0$ if and only if $p(\mathbf{R}|\mathbf{X})p(\mathbf{X}) = q(\mathbf{X}|\mathbf{R})q(\mathbf{R})$.

As a BYY system degenerates to a conventional model $q(\mathbf{X}|\Theta)$, the above Eq. 44 is simplified into the following counterpart of Eq. 39:

$$\min KL(p\|q), \; KL(p\|q) = \int p(\Theta|\mathbf{X})p(\mathbf{X}) \ln \frac{p(\Theta|\mathbf{X})p(\mathbf{X})}{q(\mathbf{X}|\Theta)q(\Theta)}d\mathbf{X}d\Theta. \quad (45)$$

Minimizing $KL(p\|q)$ with respect to a $p(\Theta|\mathbf{X})$ that is free of structure leads to $p(\Theta|\mathbf{X}) = q(\mathbf{X}|\Theta)\, q(\Theta)/q(\mathcal{X}|S)$ and $q(\mathcal{X}|S) = \int q(\mathbf{X}|\Theta)q(\Theta)\mu(d\Theta)$. As a result, Eq. 45 becomes

$$\min KL(p\|q), \; KL(p\|q) = \int p(\mathbf{X}) \ln[p(\mathbf{X})/q(\mathcal{X}|S)]d\mathbf{X}, \quad (46)$$

or max $\int p(\mathbf{X}) \ln q(\mathcal{X}|S) d\mathbf{X}$ if $p(\mathbf{X})$ is an input irrelevant to $q(\mathcal{X}|S)$ and $q(\Theta)$. For $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$, it further becomes equivalent to the MI (marginal likelihood) choice in Table 1. For a more general case with $p(\mathbf{X}, h)$ by Eq. 34, Eq. 46 provides a data smoothing extension with not only Θ but also h learned.

Alternatively we may also consider $\min_{q(\Theta)} KL(p\|q)$ when $q(\Theta)$ is free of constraint, which leads to $q(\Theta) = p(\Theta|\mathbf{X})$ and $KL(p\|q) = \int p(\Theta|\mathbf{X}) p(\mathbf{X}) \ln [p(\mathbf{X})/q(\mathbf{X}|\Theta)] d\mathbf{X} d\Theta$. When $p(\mathbf{X})$ is an input irrelevant to $q(\mathbf{X}|\Theta)$, it is equivalent to $\max \int p(\Theta|\mathbf{X})$ $p(\mathbf{X}) \ln q(\mathbf{X}|\Theta) d\mathbf{X} d\Theta$, and further becomes $\max \int p(\Theta|\mathcal{X}_N) \ln q(\mathcal{X}_N|\Theta) d\Theta$ if $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$. A further maximization with a structural free $p(\Theta|\mathbf{X})$ leads to the classical ML learning again. Moreover, in help of Eq. 26, we are lead to Eq. 40, i.e., the ML learning based AIC [1,2].

Next, we return to Eq. 44 with its inner representation $\mathbf{Y}$ in consideration. When $p(\mathbf{Y}|\mathbf{X}, \theta_{y|x})$ is free of constraint, $\min_{p(\mathbf{Y}|\mathbf{X},\theta_{y|x})} KL(p\|q)$ leads again to Eq. 45 with

$$p(\mathbf{Y}|\mathbf{X}, \theta_{y|x}) = q(\mathbf{X}|\mathbf{Y}, \theta_{x|y}) q(\mathbf{Y}|\theta_y)/q(\mathbf{X}|\Theta), \; q(\mathbf{X}|\Theta) = \int q(\mathbf{X}|\mathbf{Y}, \theta_{x|y}) q(\mathbf{Y}|\theta_y) d\mathbf{Y}. \tag{47}$$

On the other hand, $\min_{q(\Theta)} KL(p\|q)$ with a free $q(\Theta)$ results in $q(\Theta) = p(\Theta|\mathbf{X})$ and also

$$\min KL(p\|q) = \int p(\Theta|\mathbf{X}) KL(p\|q, \Theta) d\Theta \geq \min KL(p\|q, \Theta).$$

$$KL(p\|q, \Theta) = \int p(\mathbf{Y}|\mathbf{X}, \theta_{y|x}) p(\mathbf{X}) \ln \frac{p(\mathbf{Y}|\mathbf{X}, \theta_{y|x}) p(\mathbf{X})}{q(\mathbf{X}|\mathbf{Y}, \theta_{x|y}) q(\mathbf{Y}|\theta_y)} d\mathbf{X} d\mathbf{Y}. \tag{48}$$

This $\min KL(p\|q, \Theta)$ was originally proposed in 1995 under the name Bayesian Kullback Ying Yang (BKYY) learning [37]. From $\min_{p(\mathbf{Y}|\mathbf{X},\theta_{y|x})} KL(p\|q, \Theta)$, we are lead to the above discussed Eq. 47 again.

The difference between the best Ying Yang matching by Eq. 44 and the best Ying Yang harmony learning by Eq. 20 can be better understood from the following relation:

$$KL(p\|q) = \int p(\mathbf{R}|\mathbf{X}) p(\mathbf{X}) \ln [p(\mathbf{R}|\mathbf{X}) p(\mathbf{X})] d\mathbf{X} d\mathbf{R} - H(p\|q). \tag{49}$$

In addition to maximizing $H(p\|q)$, minimizing $KL(p\|q)$ also includes minimizing the first term that is the negative entropy of the Yang representation, which cancels out the least complexity nature that was discussed after Eq. 20. Recalling Eq. 22, we may also observe this difference on the level of differential flow. It follows that Eq. 44 consists of getting $p(\mathbf{R}|\mathbf{X})$ by Eq. 21, then fixing it and maximizing $\int p(\mathbf{R}|\mathbf{X}) p(\mathbf{X}) \ln [q(\mathbf{X}|\mathbf{R}) q(\mathbf{R})] d\mathbf{X} d\mathbf{R}$ via a differential flow $\int [p(\mathbf{R}|\mathbf{X}) dL(\mathbf{X}, \mathbf{R}) p(\mathbf{X}) d\mathbf{X} d\mathbf{R}$ [17]. As previously discussed after Eq. 22, the best Ying Yang harmony learning by Eq. 20 is made via the differential flow by Eq. 22 with $p(\mathbf{R}|\mathbf{X})$ replaced by $p(\mathbf{R}|\mathbf{X})[1 + \delta_L(\mathbf{R}, \mathbf{X})]$ that provides a model selection mechanism.

A summary of the BYY learning related approaches is provided in Fig. 6. The common part of all the approaches is the shadowed center area, featured by using a probabilistic model to best match a data set $\mathcal{X}_N$ via determining three levels of its unknowns. The first two levels are the ML learning for unknown parameter learning and model selection shown in the ML row of Table 1, which has been widely studied from various perspective as previously discussed in Sect. 1 [14,16,20,28]. The third level is evaluating or selecting an appropriate meta structure $\aleph$ via $q(\mathcal{X}_N|\aleph)$, i.e., the second term in Eq. 44, for which few studies have been made yet but may deserve to explore.

Outbound from this shadowed center we have two directions. One is to the left-side. Priori probabilities are taken in consideration for determining three levels of its unknowns.

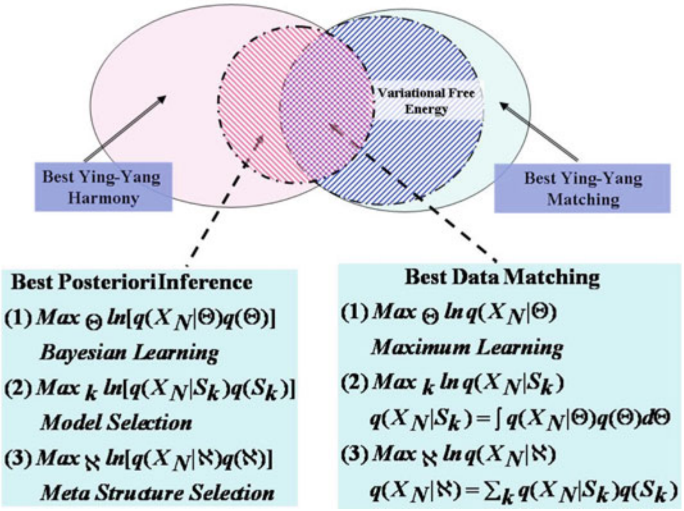


Fig. 6 Best harmony, Best matching, and Typical learning approaches

The first two levels are the MB choices for parameter learning and model selection in Table 1. As discussed in Sect. 1, studies have made under the name of Bayesian learning or Bayesian approach [22,26], as well as MML [34]. The third level is again evaluating an appropriate meta structure $\aleph$ via $q(\mathcal{X}_N|\aleph)q(\aleph)$ with a priori $q(\aleph)$ in consideration. Moving forward even left, we are lead to those areas of the best Ying Yang harmony learning by Eq. 20, which includes but goes beyond the areas of the ML and MB approaches, as discussed earlier.

The second direction goes the right-side, the domain of the best Ying Yang matching by Eq. 44. Out of the shadowed center, we enter the common area shared with the approach of *variational free energy* or the Helmholtz machine [6,19]. Moving right still, we proceed beyond and lead to a number of other cases, as discussed earlier.

The last but not least, we discuss a more detailed relation of the best Ying Yang harmony learning by Eq. 20 to an early effort called RPCL, mentioned previously in Sect. 2.3. RPCL was proposed heuristically [56] with a rival penalized mechanism for automatic model selection. Considering k individual substructures with each described by a parameter set θ_ℓ, as one new sample x_t comes, each individual has a fitness measure $L(x_t, \theta_\ell)$. A updating $\theta_\ell^{new} = \theta_\ell^{old} + \Delta\theta_\ell$ is made as follows [49]:

$$\Delta\theta_\ell \propto p_{\ell,t}\nabla_{\theta_\ell} L(x_t,\theta_\ell), \quad p_{\ell,t} = \begin{cases} 1, & \text{if } \ell = \ell^* \text{with } \ell^* = arg\max_\ell L(x_t,\theta_\ell), \\ -\gamma, & \text{if } \ell^* = arg\max_{\ell \neq \ell^*} L(x_t,\theta_\ell), \\ 0, & \text{otherwise.} \end{cases} \tag{50}$$

where $\gamma > 0$ is a small number. This is, the most fit one learns x_t, while the second winner (rival) de-learns, such that those extra substructures have been gradually discarded during an iterating process of Eq. 50. In the sequel, we try to link Eq. 50 to $H_f(\mathbf{X}, \Theta, \mathbf{k}, \Xi)$ by Eq. 20 at $p(\mathbf{X}|\theta_x) = \delta(\mathbf{X} - \mathcal{X}_N)$ via rewriting it into $H_L(\Theta, \mathbf{k}) + H(\theta_{y|x}) + \ln q(\Theta|\Xi_q)$ with

$$H_L(\Theta,\mathbf{k}) = \sum_L p(L|\mathcal{X}_N, \theta_{y|x}) \ln [q(\mathcal{X}_N|L, \theta_{xy}) q(L|\theta_L)],$$

$$
\begin{aligned}
H(\theta_{y|x}) &= \sum_L p(L|\mathcal{X}_N, \theta_{y|x}) \left[\int p(\mathbf{Y}_v|\mathcal{X}_N, L, \theta_{y|x}) \ln p(\mathbf{Y}_v|\mathbf{X}, L, \theta_{y|x}) d\mathbf{Y}_v \right], \\
q(\mathbf{X}|L, \theta_{xy}) &= \int q(\mathbf{X}|\mathbf{Y}_v, L, \theta_{x|y}) q(\mathbf{Y}_v|L, \theta_y) d\mathbf{Y}_v. \qquad (51)
\end{aligned}
$$

Following the line of Eq. 22, we get the differential updating flow of $H_L(\Theta, \mathbf{k})$ as follows:

$$
\begin{aligned}
& dH_L(\Theta, \mathbf{k}) = \sum_L p(L|\mathcal{X}_N, \theta_{y|x})[1 + \delta_L(\mathcal{X}_N)] dL(\mathcal{X}_N, \Theta_L), \\
& L(\mathcal{X}_N, \Theta_L) = \ln [q(\mathcal{X}_N|L, \theta_{xy}) q(L|\theta_L)], \\
& \delta_L(\mathcal{X}_N) = L(\mathcal{X}_N, \Theta_L) - \sum_L p(L|\mathcal{X}_N, \theta_{y|x}) L(\mathcal{X}_N, \Theta_L); \\
& \text{For the i.i.d. case by Eq. 38, it becomes } \ L(x, \theta_\ell) = \ln [q(x|\theta_\ell)\alpha_\ell], \\
& dH_L(\Theta, \mathbf{k}) = \sum_t \sum_\ell p(\ell|x_t)[1 + \delta_\ell(x_t)] dL(x_t, \theta_\ell), \quad p(\ell|x_t) = \frac{e^{L(x,\theta_\ell)}}{\sum_j e^{L(x,\theta_j)}}, \qquad (52) \\
& \delta_\ell(x_t) = L(x_t, \theta_\ell) - \sum_j p(\ell|x_t, \theta_{\ell|x}) L(x_t, \theta_j), \\
& q(x|\theta_\ell) = \int q(x|y, \ell, \theta_{x|y,\ell}) q(y|\ell, \theta_y) dy.
\end{aligned}
$$

Similar to the discussion after Eq. 22, $p(\ell|x_t)[1 + \delta_\ell(x_t)]$ may enhance learning, make de-learning, and proceed with reservation, in comparison with the ML learning on a finite mixture $\sum_\ell \alpha_\ell q(x|\theta_\ell)$. Instead of getting $p_{\ell,t}$ via winner-take-all competition in Eq. 50, $p(\ell|x_t)$ in Eq. 52 involves a soft competition while $p(\ell|x_t)[1+\delta_\ell(x_t)]$ further uses the average fitness as a reference.

In general, the updating flow also has contributions from $dH(\theta_{y|x})$ and $d \ln q(\Theta|\Xi_q)$, which provides certain regularization or modification on $dH_L(\Theta, \mathbf{k})$ in Eq. 52. In the special case that $\mathbf{Y}$ consists of only $\mathbf{L}$ without the part $\mathbf{Y}_v$, we have simply $H(\theta_{y|x}) = 0$.

4 Learning versus optimization: cross fertilization

4.1 Roles of convexity in learning

Now we are ready to go over the main points about the relations between learning and optimization. *First*, the purpose of learning is building up a learning system to represent regularity or dependence structure underlying training samples. The goal is achieved via an optimization process. *Second*, learning is not just optimization. It consists of at least three basic tasks, namely, getting an appropriate structure as the hardware of a learning system, finding a good learning theory to guide a learning process for determining an appropriate structural complexity of a learning system (called model selection) and all unknown parameters (called parameter learning) in the learning system, and then implementing model selection and parameter learning effectively by an optimization process. *Third*, a major computational difficulty comes from computing integrals. The integrals are usually avoided after being turned into a summation plus optimizations in help of certain technique. *Fourth*, optimizations in a learning process are hierarchically nested. The outmost one is a discrete optimization for model selection on $\mathbf{k}$, during which each step consists of computations plus a continuous optimization for parameter learning on Θ. Moreover, each parameter learning process may also be nested with either or both of a discrete optimization and a continuous optimization for inferring inner representation Y. Beyond these, the interaction between learning and optimization can also be viewed from studies on the nature of convexity. In the sequel, we discuss this issue roughly from three directions.

The first direction is featured by a fact that a learning theory is actually implemented via optimizing a cost function with certain convexity and learning algorithm is developed with help of local convexity. The simplest one is minimizing a least square error $\sum_{t=1}^{N} \|y_t - f(x_t, \theta)\|^2$, which is convex with respect to f but is not necessary with respect to Θ. The least square error is actually a special case of the likelihood function $\ln q(\mathcal{X}_N|\Theta)$, which is again convex with respect to q but not necessary with respect to Θ. The latter depends on the function form of $q(x_t|\Theta)$ and types of parameters within the set Θ. Also, the harmony measure $H(p\|q)$ by Eq. 20 and the Kullback measure $KL(p\|q)$ by Eq. 44 are convex with respect to either or both of p, q. Still, whether $H(p\|q)$ or $KL(p\|q)$ are convex with respect to $\mathbf{R}$ (thus $\mathbf{Y}, \Theta$) depends on the specific function forms of p, q.

Moreover, use of convexity takes an important role in avoiding computational difficulty of integrals and in developing an effective learning algorithms. Recalling Sect. 3.2, local convexity acts as a major tool that makes the best harmony learning by Eq. 20 become computationally implementable. In addition to such a use of local convexity, efforts have also been made on avoiding computational difficulty of integrals by using convexity in a global level. Maximizing the likelihood function $q(\mathbf{X}|\Theta) = \int q(\mathbf{X}|\mathbf{Y}, \theta_{x|y}) q(\mathbf{Y}|\theta_y) d\mathbf{Y}$ is suggested to be replaced by maximizing one of its lower bound via the *Helmholtz free energy or variational free energy* [6,19], which can also be understood from the formulation of Eq. 47, that is, $\max_\Theta q(\mathbf{X}|\Theta)$ is replaced by maximizing the following cost

$$\begin{aligned} F &= -\int p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x}) \ln \frac{p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x})}{q(\mathcal{X}_N|\mathbf{Y}, \Theta) q(\mathbf{Y}|\theta_y)} d\mathbf{Y} \\ &= -\int p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x}) \ln \frac{p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x})}{q(\mathbf{Y}|\mathcal{X}_N, \Theta)} d\mathbf{Y} + \ln q(\mathcal{X}_N|\Theta) \le \ln q(\mathcal{X}_N|\Theta), \\ q(\mathbf{Y}|\mathcal{X}_N, \Theta) &= q(\mathcal{X}_N|\mathbf{Y}, \theta_{x|y}) q(\mathbf{Y}|\theta_y)/q(\mathcal{X}_N|\Theta). \end{aligned} \tag{53}$$

Instead of computing $q(\mathcal{X}_N|\Theta)$ and $q(\mathbf{Y}|\mathcal{X}_N, \Theta)$, a pre-specified parametric model is considered for $p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x})$, and learning is made for determining the unknown parameters $\theta_{y|x}$ together with Θ via maximizing F.

Actually, maximizing F by Eq. 53 is equivalent to $\min_\Theta KL(p\|q, \Theta)$ by Eq. 48 with $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$. In other words, two approaches coincide in this situation, while they were motivated from two different perspectives. Maximizing F by Eq. 53 directly aims at approximating the ML learning on $q(\mathcal{X}_N|\Theta)$, with an approximation gap that trades off computational efficiency via a pre-specified parametric $p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x})$. This gap disappears if $p(\mathbf{Y}|\mathcal{X}_N, \theta_{y|x})$ is able to reach the posteriori $q(\mathbf{Y}|\mathcal{X}_N, \Theta)$. However, minimizing $KL(p\|q, \Theta)$ by Eq. 48 is not motivated from a purpose of approximating the ML learning though it was also shown in [37] that $\min_{p(\mathbf{Y}|\mathbf{X}, \theta_{y|x})} KL(p\|q, \Theta)$ for a $p(\mathbf{Y}|\mathbf{X}, \theta_{y|x})$ free from of constraints makes $\min_\Theta KL(p\|q, \Theta)$ become the ML learning when $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$. Instead, the motivation is determining all the unknowns in the Ying-Yang pair to make the pair best matched. The approaches of the shadowed center in Fig. 6 are special cases of minimizing the *Helmholtz free energy* $-F$ by Eq. 53 and of minimizing $KL(p\|q, \Theta)$ by Eq. 48. In addition to being equivalent to the ML learning and approximating the ML learning, studies on $\min_\Theta KL(p\|q, \Theta)$ by Eq. 48 further covers not only extensions to $p(\mathbf{X}, h)$ by Eq. 34, but also the problems of $\min_{q(\mathbf{X}|\mathbf{Y}, \theta_{x|y})} KL(p\|q, \Theta)$ with respect to a free $q(\mathbf{X}|\mathbf{Y}, \theta_{x|y})$, which leads to

$$\min \int p(\mathbf{Y}|\Theta_p) \ln \frac{p(\mathbf{Y}|\Theta_p)}{q(\mathbf{Y}|\theta_y)} d\mathbf{Y}, \quad p(\mathbf{Y}|\Theta_p) = \int p(\mathbf{Y}|\mathbf{X}, \theta_{y|x}) p(\mathbf{X}) d\mathbf{X}. \tag{54}$$

Particularly, when $q(\mathbf{Y}|\theta_y)$ is independent among its components and $p(\mathbf{Y}|\mathbf{X}, \theta_{y|x})$ has a simple post-linear structure, Eq. 54 further becomes equivalent to the minimum mutual information (MMI) base ICA learning [3]. The details are referred to [38,42,44,45].

The second direction is directly applying the existing results obtained from convex optimization literature, tailored for a particular learning task. Typical ones are those featured by support vectors [32]. One example is the following convex optimization:

$$\begin{aligned}
&\max_{\{\alpha, C^*\}} W(\alpha, C^*),\ \alpha = \{\alpha_t,\ t = 1, 2, \ldots, N\},\\
&\qquad W(\alpha, C^*) = \sum_{t=1}^{N} \alpha_t - 0.5 \sum_{t=1}^{N} \sum_{\tau=1}^{N} \alpha_t \alpha_\tau y_t y_\tau (x_t \cdot x_\tau) - 0.5 c_n C^*,\\
&\text{subject to } \sum_{t=1}^{N} \alpha_t y_t = 0,\ 1 \geq \alpha_t \geq 0,\ t = 1, 2, \ldots, N,\ C^* \geq 0, \qquad (55)
\end{aligned}$$

where $c_n > 0$ is a given constant, $\{x_t, y_t\}$ are a given set of paired samples, with x_t being a real vector and y_t taking either -1 or 1. The optimal solution is used to build up a hyperplane that classifies each sample into one of two classes. Also, extensions have been made with the inner product $(x_t \cdot x_\tau)$ replaced by a kernel function $K(x_t, x_\tau)$ that satisfies Mercer condition [33].

In addition to this classification, studies have also been made on making an ϵ-insensitive robust regression $y = \sum_{t=1}^{N} \beta_t K(x, x_t)$ by $\beta_t = (\alpha_t^* - \alpha_t)/C^*$, with $\alpha_t^*, \alpha_t, C^*$ obtained from the following optimization problem:

$$\begin{aligned}
&\max_{\{\alpha, \alpha^*, C^*\}} W(\alpha, \alpha^*, C^*),\ \alpha = \{\alpha_t,\ t = 1, 2, \ldots, N\},\ \alpha^* = \{\alpha_t^*,\ t = 1, 2, \ldots, N\},\\
&\qquad W(\alpha, \alpha^*, C^*) = -\epsilon \sum_{t=1}^{N} (\alpha_t^* + \alpha_t) + \sum_{t=1}^{N} y_t (\alpha_t^* - \alpha_t) - 0.5 c_n C^*\\
&\qquad\qquad + \frac{1}{2C^*} \sum_{t=1}^{N} \sum_{\tau=1}^{N} (\alpha_t^* - \alpha_t)(\alpha_\tau^* - \alpha_\tau) K(x_t, x_\tau),\\
&\text{subject to } \sum_{t=1}^{N} \alpha_t^* = \sum_{t=1}^{N} \alpha_t,\ 1 \geq \alpha_t^* \geq 0,\ 1 \geq \alpha_t \geq 0,\\
&\qquad t = 1, 2, \ldots, N,\ C^* \geq 0, \qquad (56)
\end{aligned}$$

where $c_n > 0, \epsilon > 0$ are given constants.

The third direction is featured by those extensive studies under the name of variational approximation methods [12,13], which further puts the basic idea of the *Helmholtz free energy or variational free energy* [6,19] in a general framework of approximation methods rooting from techniques in the calculus of variations, and in a wide variety of uses such as finite element analysis, quantum mechanics, statistical mechanics, and statistics [27]. The key idea is turning a complex problem into a simpler one, featured by a decoupling of the degrees of freedom in the original problem. This decoupling is achieved via an expansion of the problem to include additional parameters (called variational parameters), in help of convex duality [25].

We start at the following example [13]:

$$\ln(x) = \min_y (yx - \ln y - 1) \leq yx - \ln y - 1, \qquad (57)$$

which transfers a nonlinear function into a bound in a linear function, in help of a free parameter y. We can recover the exact value of logarithm for the optimal choice of y.

This type of variational transformation can be handled systematically in help of convex duality studied in the optimization literature [11,25]. That is, a convex function $f(x)$ can be represented via a conjugate or dual function as follows:

$$f(x) = \max_y [y^T x - f^*(y)], \text{ with } f^*(y) = \max_x [y^T x - f(x)], \tag{58}$$

which includes the above Eq. 57 as a special case.

In general, the convex duality has been applied to the machine learning literature to obtain upper or lower bounds on a function of interest such that the original computation is replaced by a tractable approximation. If the function is already convex then we simply calculate the conjugate function. If a function is not convex, an invertible transformation is sought such that the function becomes convex after transformed. Then, the conjugate function is calculated in the transformed space and transform back. Furthermore, it has been shown in Sect. 6 of [13] that Eq. 53 can be reached directly in help of the convex duality by Eq. 58, which renders the Helmholtz machine type learning [6,19] as a typical example of the variational methods for probability distributions.

4.2 Combinatorial optimization from a BYY learning perspective

It is interesting to further discuss that learning may help to make optimization more effectively too. Recalling Fig. 1b, Eq. 8 and 10, we may reexamine the combinatorial optimization problems such as AGM, TSP from the perspective of inferring a best Y^* based on the Ying machine $q(X|Y)$ and $q(Y)$. Particularly, we need an appropriate $q(Y)$ that is able to limit each Y to distribute within the permutation matrix family $\mathbf{\Pi}_N$, while $q(X|Y)$ describes a specific combinatorial optimization problem $E_o(Y, X)$, which can be generally described as in Eq. 10. Without a priori preference, $q(Y)$ is equal for each $Y \in \mathbf{\Pi}_N$. We infer Y^* by Eq. 1 or equivalently $Y^* = arg \max_{Y \in \mathbf{\Pi}_N} \ln q(X|Y)$, where $\ln q(X|Y) = -\frac{1}{\lambda} E_o(Y, X) - \ln Z_\lambda(X)$. That is, it is equivalent to the original combinatorial optimization problem $\min_{Y \in \mathbf{\Pi}_N} E_o(Y, X)$. As discussed in Sect. 1, the probabilistic version of Eq. 1 is $P(Y|X) = q(X|Y)q(Y)/\sum_Y q(X|Y)q(Y)$. The summation should goes over $Y \in \mathbf{\Pi}_N$ and thus has a computational complexity same as the original problem.

Recalling the first column of Table 1, an effort towards this computational difficulty is LPD. That is, we let $P(Y|X)$ to be expressed by a parametric model $P(Y|X, \theta_p)$ to simplify the computation. Then, the problem is turned to learn θ_p. Since the dimension of Y is known and fixed by a specific problem, there is no need on model selection. We consider the best Ying Yang matching by min $KL(p\|q, \theta)$ in Eq. 48. Since the input X is fixed and $Z_\lambda(X)$ is irrelevant to learning, it follows from Eq. 10 that min $KL(p\|q, \Theta)$ is further simplified into

$$\min_\Theta KL(p\|q, \Theta), \;\; KL(p\|q, \Theta) = \sum_Y P(Y|X, \theta_p) \ln \frac{P(Y|X, \theta_p)}{q(X|Y)q(Y \; on \; \mathbf{\Pi}_N, \theta_q)}, \tag{59}$$

$$\text{or} \min_\Theta \Big[\sum_Y P(Y|X, \theta_p) \ln P(Y|X, \theta_p) - \sum_Y P(Y|X, \theta_p) \ln e^{-\frac{1}{\lambda} E_o(Y,X) + \ln q(Y \; on \; \mathbf{\Pi}_N, \theta_q)} \Big].$$

where λ controls the strength of the role by $q(Y \; on \; \mathbf{\Pi}_N, \theta_q)$. To make Eq. 59 works, we need an appropriate $q(Y \; on \; \mathbf{\Pi}_N, \theta_q)$ and an appropriate scheme for controlling the value of λ. Then, we can alternatively make $\min_{\theta_p} KL(p\|q, \Theta)$ and $\min_{\theta_q} KL(p\|q, \Theta)$ such that the iteration converges to a minimum as globally as possible.

In the sequel, we introduce some previous studies [4,5,36,38,47], actually on a special case of the above Eq. 59. Considering $q(Y \; on \; \mathbf{\Pi}_N, \theta_q)$ is equal for every $Y \in \mathbf{\Pi}_N$ and noticing $q(X|Y)$ by Eq. 10 with $\sum_Y q(X|Y) = \sum_Y e^{-\frac{1}{\lambda}E_o(Y,X)}/Z_\lambda(X) = 1$, we can turn Eq. 59 into

$$\min_{\Theta,\ s.t.\ Y\ on\ \mathbf{\Pi}_N} KL(p,q),\ KL(p,q) = \sum_Y p(Y|X,\theta_p)\ln\frac{p(Y|X,\theta_p)}{q(Y|X,\lambda)},$$
$$q(Y|X,\lambda) = \frac{q(X|Y)q(Y\ on\ \mathbf{\Pi}_N,\theta_q)}{\sum_Y q(X|Y)q(Y\ on\ \mathbf{\Pi}_N,\theta_q)} = \frac{e^{-\frac{1}{\lambda}E_o(Y,X)}}{Z_\lambda(X)}, \tag{60}$$

for which we have an intuitive interpretation. It is difficult to get the peak Y^* since $E_o(Y,X)$ has many local minimums or $q(Y|X,\lambda)$ has many local maximums. We use a simple distribution $p(Y|X,\theta_p)$ to approximate $q(Y|X,\lambda)$ such that the global peak of $p(Y|X,\theta_p)$ becomes easier to find and that $q(Y|X,\lambda)$ and $p(Y|X,\theta_p)$ share the same peak Y^*.

Moreover, we consider a series of domains of $q(Y|X,\lambda)$:

$$D_\varepsilon(\lambda) = \{Y : q(Y|X,\lambda) > \varepsilon,\ a \text{ small constant } \varepsilon > 0\} \tag{61}$$

under the control of a parameter λ. For a sequence $\lambda_0 > \lambda_1 \cdots > \lambda_t$, we have $D_\varepsilon(\lambda_t) \subset \ldots D_\varepsilon(\lambda_1) \subset D_\varepsilon(\lambda_0)$ that keep the global minimization solution of $E_o(Y,X)$ included, since the equivalence of $\max_Y q(Y|X,\lambda)$ to $\min_Y E_o(Y,X)$ is irrelevant to λ. Therefore, we can find a sequence $p_0(Y|X,\theta_p), p_1(Y|X,\theta_p),\ldots,p_t(Y|X,\theta_p)$ that approximates $q(Y|X,\lambda)$ on the shrinking domain $D_\varepsilon(\lambda)$. For a large λ_t, $q(Y|X,\lambda)$ has a large support and thus $p(Y|X,\theta_p)$ adapts the overall configuration of $q(Y|X,\lambda)$ in a big domain $D_\varepsilon(\lambda)$. As λ_t reduces, $p_t(Y|X,\theta_p)$ becomes more and more concentrating on adapting the detailed configuration of $q(Y|X,\lambda)$ around the global peak solution $Y^* \in D_\varepsilon$. As long as λ_0 is large enough and λ reduces slowly enough towards to zero, we can finally find the global minimization solution of $E_o(Y,X)$. In implementation, this process is still made by Eq. 60 simply with $\sum_Y$ replaced by $\sum_{Y\in D_Y}$. Alternatively, we can also consider the case with the positions of p,q swapped $\min_p\ KL(p,q)$, which leads us to a class of Metropolis sampling based mean field approaches. Details are referred to Sect. II(B) in [41].

Here, we focus on Eq. 60 and consider $p(Y|X,\theta_p)$ in the following simple forms:

$$p_1(Y|X,\theta_p) = Z_1^{-1}\prod_{i,j} e^{y_{ij}\ln p_{ij}},\quad 0 \le p_{ij},\ Z_1 = \sum_{i,j}\prod_{i,j} e^{y_{ij}\ln p_{ij}},$$
$$p_2(Y|X,\theta_p) = \prod_{i,j} p_{ij}^{y_{ij}}(1-p_{ij})^{1-y_{ij}},\quad 0 \le p_{ij} \le 1; \tag{62}$$

and from the constraints of $Y \in \mathbf{\Pi}_N$ we have

$$C^c:\ \sum_{i=1}^N \langle y_{ij}\rangle = 1, j = 1,\ldots,N,\ C^r:\ \sum_{j=1}^N \langle y_{ij}\rangle = 1, i = 1,\ldots,N;$$
$$\langle y_{ij}\rangle = \begin{cases} p_{ij}\frac{Z_{ij}}{Z_1}, & \text{for} p_1(Y|X,\theta_p), \\ p_{ij}, & \text{for} p_2(Y|X,\theta_p), \end{cases}\quad Z_{ij} = \textstyle\sum_{k\ne i,l\ne j}\prod_{k,l} e^{y_{kl}\ln q_{kl}}; \tag{63}$$

where $\langle y\rangle$ denotes the expectation of the random variable y. When N is large, we have $Z_{ij} \approx Z_1$, and thus $\langle y_{ij}\rangle \approx p_{ij}$ for the case of $p_1(Y|X,\theta_p)$.

Putting Eq. 62 into Eq. 60, we have $KL(p,q) = H_{Y|X} - L_{X|Y}$ with

$$H_{Y|X} = \sum_Y p(Y|X,\theta_p)\ln p(Y|X,\theta_p) = \begin{cases} \sum_{ij} p_{ij}\ln p_{ij}, & \text{for} p_1, \\ \sum_{ij}[p_{ij}\ln p_{ij} + (1-p_{ij})\ln(1-p_{ij})], & \text{for} p_2. \end{cases}$$

$$L_{X|Y} = \sum_Y p(Y|X, \theta_p) \ln q(Y|X, \beta) = -\frac{1}{\lambda} \sum_Y p(Y|X, \theta_p) E_o(Y, X) - \ln Z_\lambda(X). \quad (64)$$

Further noticing $\sum_Y p(Y|X, \theta_p) E_o(Y, X) \approx E_o(\{p_{ij}\}, X)$, Eq. 60 becomes equivalent to

$$\min_{p_{ij}} E(\{p_{ij}\}, X), \quad \text{subject to Eq. 63},$$

$$E(\{p_{ij}\}, X) = \frac{1+\gamma}{\lambda} E_o(\{p_{ij}\}, X) + (1-\gamma) \begin{cases} \sum_{ij} p_{ij} \ln p_{ij}, & \text{for } p_1(Y|X, \theta_p), \\ \sum_{ij} [p_{ij} \ln p_{ij} + (1-p_{ij}) \ln (1-p_{ij})], & \text{for } p_2(Y|X, \theta_p), \end{cases} \quad (65)$$

where $\gamma = 0$ but it will be extended to take other values later in Eq. 67. The case for $p_1(Y|X, \theta_p)$ interprets the Lagrange-Transform approach with the barrier term $\sum_{i,j} y_{ij} \ln y_{ij}$ in [36] and justifies the intuitive treatment of simply regarding the discrete y_{ij} as an analog variable between the interval [0, 1]. These analog variables are actually the parameters of a simple distribution that we use to approximate the Gibbs distribution induced from the cost $E(\{p_{ij}\}, X)$ of the discrete variables.

Similarly, the case for $p_2(Y|X, \theta_p)$ interprets and justifies the Lagrange-Transform or Lagrange-Barrier approach with the barrier $\sum_{i,j} [y_{ij} \ln y_{ij} + (1 - y_{ij}) \ln (1 - y_{ij})]$ in [38], where this barrier is intuitively argued to be better than the barrier $\sum_{i,j} y_{ij} \ln y_{ij}$ because it gives a U-shape curve. Here, this intuitive preference can also be justified from Eq. 63 since there is an approximation $Z_{ij} \approx Z_1$ used for $p_1(Y|X, \theta_p)$, but no approximation for $p_2(Y|X, \theta_p)$. Moreover, both the barriers are respectively the special cases (a) $S(y_{ij}) = y_{ij}$ and (b) $S(y_{ij}) = y_{ij}/(1-y_{ij})$ of a family of barrier functions that are equivalent to minimizing the leaking energy in the classical Hopfield network [38].

In the implementation of Eq. 65, a set of iterative updating equations in a parallel implementation is obtained in [38] from $\frac{\partial E(\{p_{ij}\}, X)}{\partial p_{ij}} = 0$ to replace those dynamic equations in the Hopfield network [10], with an improved stability. Moreover, the constraints C^c, C^r are handled by updating the Lagrange coefficients in help of another set of dynamic equations for maximization or iterative equations [4,5,47].

The solution from the above approaches will be $\{p_{ij}\}$ with $0 \le p_{ij} \le 1$, instead of a permutation matrix $Y \in \mathbf{\Pi}_N$. We can approximately turn p_{ij} into either 1 if p_{ij} is larger than a threshold or 0 otherwise.

Alternatively, we may consider the best harmony learning by Eq. 20. The counterpart of Eq. 59 is

$$\max H(p\|q, \Theta), \; H(p\|q, \Theta) = \sum_Y P(Y|X, \theta_p) \ln [q(X|Y) q(Y \; on \; \mathbf{\Pi}_N, \theta_q)]. \quad (66)$$

For $P(Y|X, \theta_p)$ given by Eq. 62, max $H(p\|q, \Theta)$ will push p_{ij} to be 1 or 0, which avoids the above problem. However, this max $H(p\|q, \Theta)$ is vulnerable due to many local maximums.

As previously suggested in [45] and further elaborated in Sect. 23.4.2 of [39], one solution is minimizing a combination $KL(p\|q, \Theta) - \gamma H(p\|q, \Theta)$. We start at $\gamma = 0$ or gradually increase γ in certain way similar to simulated annealing [15]. Also, we may approximately regard $q(Y \; on \; \mathbf{\Pi}_N, \theta_q)$ is free, and making max $H(p\|q, \Theta)$ with respect to it results in $q(Y \; on \; \mathbf{\Pi}_N, \theta_q) = P(Y|X, \theta_p)$ and $H(p\|q, \Theta) = -H_{Y|X} - L_{X|Y}$. It follows from Eq. 64 that

$$KL(p\|q, \Theta) - \gamma H(p\|q, \Theta) = (1-\gamma) H_{Y|X} - (1+\gamma) L_{X|Y}. \quad (67)$$

Therefore, it is implemented still by Eq. 65 as above discussed with γ starting from 0 and gradually increasing. As γ goes beyond 1, max $H(p\|q, \Theta)$ dominates and p_{ij} will be pushed to be either 1 or 0.

In addition to the Lagrange type algorithms [4,5,38,47], recently it is proposed in [46,47] that the constraints $C^c : \sum_{i=1}^{N} p_{ij} = 1, j = 1, \ldots, N, C^r : \sum_{j=1}^{N} p_{ij} = 1, i = 1, \ldots, N$, and $0 \le p_{ij} \le 1$ are jointly considered by letting $y_{ij} = r_{ij}^2$, which induces a matrix $R = \{r_{ij}\}_{i=1,j=1}^{i=N,j=N}$ that satisfies $RR^T = I$. As a result, the problem $\min_{p_{ij}} E(\{p_{ij}\}, X)$ in Eq. 65 is relaxed into $\min_{RR^T=I} E(\{r_{ij}^2\}, X)$, handled by a gradient flow on the Stiefel manifold $RR^T = I$ with all the constraints guaranteed automatically. One example of such a flow is $R^{new} = R^{old} + \Delta R$ with $\Delta R \propto G_R(I - R^T R)$ and $G_R = \nabla_V E(\{p_{ij}\}, X)_{p_{ij}=r_{ij}^2} \circ R$ for $E(\{p_{ij}\}, X)$ in Eq. 65, where the notation $\circ$ denotes the Hadamard product. A general technique for optimization on the Stiefel manifold was elaborated in [7] and can be adopted for our purpose.

5 Concluding remarks

Based on a set of evidences or samples, the purpose of learning consists of providing a learning system with a pre-specified structure in an appropriate scale or complexity, and then determining all the unknown parameters within this structure, as well as enabling the learning system to perform inference and to make various reactions upon observed evidences or samples. Aiming at this purpose, we need one learning theory to evaluate what is an appropriate structure and which performance is good. As introduced in this paper, such a theory could be developed from one of two major principles. One is called *best matching*. That is, making the learning system best match a given set of samples, in a sense of $\max_\Theta \ln q(\mathcal{X}_N|\Theta)$ or $\max_\Theta \ln [q(\mathcal{X}_N|\Theta)q(\Theta)]$ for determining parameters Θ in a pre-specified structure and in a sense of $\max \ln q(\mathcal{X}_N),\ q(\mathbf{X}) = \int q(\mathbf{X}|\Theta)q(\Theta)\mu(d\Theta)$ for selecting a best structure as well. The other is called *best harmony*, associated with the BYY system. Instead of simply considering the learning system as a part to match the observed data as the other part, we consider a learning system that consists of a Ying machine and a Yang machine, with data considered as a part of the Yang machine. Not only all the unknown parameters but also the scales of structures are determined such that the Yang-Yang pair reaches a best harmony in a sense of a best matching with structures in a least complexity, which is different from *best matching* in nature. It degenerates to becoming equivalent to *best matching* in two special cases. One is that the Yang machine degenerates into consisting of data only while the Ying machine degenerates into a single structure $q(\mathbf{X}|\Theta)$ without considering any inner representations. The other case is that the Yang machine consists of data $p(\mathbf{X}) = \delta(\mathbf{X} - \mathcal{X}_N)$ and a Yang pathway that is free to be determined in a best matching by Eq. 44 or 45.

For whatever a theory, an effective implementation is needed, which is usually featured by two competing mechanisms. One is integrating evidences from all the possible scenarios in a broad scope via integrals of types $\int[\cdot]d\Theta$ and $\int[\cdot]d\mathbf{Y}$, while the other mechanism is optimal searching with focuses on one or more best values of Θ and $\mathbf{Y}$. Optimization approaches take their roles not only in an effective implementation of optimal searching, but also in an approximate implementation of integrating evidences to avoid the difficulty of handling integrals. Corresponding to the problems of determining three levels of unknowns in a learning system, there are three types of optimization tasks as previously introduced in Sect.2. The nature of convexity takes important roles in machine learning, either directly towards a convex programming or approximately transferring a difficult problem into a tractable one in help of local convexity or convex duality. Therefore, new developments from the optimization literature will always thrust the advances of machine learning. Furthermore, learning versus optimization has also been examined from a Ying-Yang perspective, with combinatorial optimization made more effectively.

Acknowledgements The work described in this paper was fully supported by a grant from the Research Grant Council of the Hong Kong SAR (Project No: CUHK4173/06E).

References

1. Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **19**, 714–723 (1974)
2. Akaike, H.: Likelihood of a model and information criteria. J. Econom. **16**, 3–14 (1981)
3. Amari, S., Cichocki, A., Yang, H.: A New Learning Algorithm for Blind Signal Separation. Advances in NIPS, 8, pp. 757–763. MIT Press (1996)
4. Dang, C., Xu, L.: A globally convergent Lagrange and barrier function iterative algorithm for the traveling salesman problem. Neural Netw. **14**(2), 217–230 (2001)
5. Dang, C., Xu, L.: A Lagrange multiplier and Hopfield-type barrier function method for the traveling salesman problem.. Neural Comput. **14**(2), 303–324 (2001)
6. Dayan, P., Hinton, G.E., Neal, R.M., Zemel, R.S.: The Helmholtz machine. Neural Comput. **7**(5), 889–904 (1995)
7. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**, 303–353 (1998)
8. Eshera, E., Fu, K.S.: A graph distance measure for image analysis. IEEE Trans. SMC **14**(3), 396–408 (1984)
9. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length and Helmholtz free energy. Adv. NIPS **6**, 3–10 (1994)
10. Hopfield, J.J., Tank, D.W.: Neural computation of decisions in optimization problems. Biol. Cybern. **52**, 141–152 (1985)
11. Horst, R., Pardalos, P.M.: Handbook of Global Optimization, Nonconvex Optimization and its Applications, vol. 2. Kluwer (1995)
12. Jaakkola, T.S.: Tutoiral on variational approximation methods. In: Opper, M., Saad, D. (eds.) Advanced Mean Field Methods: Theory and Pratice, pp. 129–160. MIT press (2001)
13. Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: Introduction to variational methods for graphical models. Mach. Learn. **37**, 183–233 (1999)
14. Kass, R.E., Raftery, A.E.: Bayes factors. J. Am. Stat. Assoc. **90**, 773–795 (1995)
15. Kirkpatrick, S., Gelatt, C.G. Jr., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
16. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003)
17. McLachlan, G.J., Geoffrey, J.: The EM Algorithms and Extensions, Wiley (1997)
18. Moulines, E., Cardoso, J., Gassiat, E.: Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. Proc. ICASSP97, pp. 3617–3620 (1997)
19. Neal, R., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 355–368. MIT Press, Cambridge, MA (1999)
20. Neath, A.A., Cavanaugh, J.E.: Regression and time series model selection using variants of the schwarz information criterion. Commun. Stat. A **26**, 559–580 (1997)
21. Poggio, T., Girosi, F.: Networks for approximation and learning. Proc. IEEE **78**, 1481–1497 (1990)
22. Press, S.J.: Bayesian statistics: principles, models, and applications. Factors. Wiley (1989)
23. Rissanen, J.: Stochastic Complexity in Statistical Inquiry. World Scientific, Singapore (1989)
24. Rivals, I., Personnaz, L.: On cross validation for model selection. Neural Comput. **11**, 863–870 (1999)
25. Rockafellar, R.: Convex Analysis. Princeton University Press (1972)
26. Ruanaidh, O., Joseph, J.K.: Numerical Bayesian methods applied to signal processing. Springer-Verlag, New York (1996)
27. Rustagi, J.: Variational Method in Statistics. Academic Press, New York (1976)

28. Schwarz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 461–464 (1978)
29. Stone, M.: Cross-validation: a review. Math. Operat. Stat. **9**, 127–140 (1978)
30. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-posed Problems. Winston and Sons (1977)
31. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. IEEE Trans. Pattern Anal. Mach. Intell. **10**(5), 695–703 (1988)
32. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer (1995)
33. Vapnik, V.N.: Estimation of Dependences Based on Empirical Data. Springer (2006)
34. Wallace, C.S., Boulton, D.M.: An information measure for classification. Comput. J. **11**, 185–194 (1968)
35. Wang, L., Feng, J.: Learning Gaussian mixture models by structural risk minimization. Proc. 2005 Int. Conf. Machine Learning and Cybernetics (ICMLC), pp. 4858–4863. 19–21 Aug 2005, Guangzhou, China (2005)
36. Xu, L.: Combinatorial optimization neural nets based on a hybrid of Lagrange and transformation approaches. Proc. 1994 World Congress Neural Networks, pp. 399–404. SanDiego (1994)
37. Xu, L.: Bayesian-Kullback coupled YING-YANG machines: unified learnings and new results on vector quantization. Proc. ICONIP95, pp. 977–988. Beijing (1995)
38. Xu, L .: On the hybrid LT combinatorial optimization: new U-shape barrier, sigmoid activation, least leaking energy and maximum entropy. Proc. ICONIP'95, pp. 309–312. Beijing (1995)
39. Xu, L.: Bayesian Ying-Yang system and theory as a unified statistical learning approach (I): unsupervised and semi-unsupervised learning. In: Amari, K. (ed.) Brain-like Computing and Intelligent Information Systems, pp. 241–274. Springer-Verlag (1997)
40. Xu, L.: BYY harmony learning, independent state space and generalized APT financial analyses. IEEE Trans. Neural Netw. **12**, 822–849 (2001)
41. Xu, L.: Distribution approximation, combinatorial optimization, and Lagrange-barrier. Proc. Intl. Joint Conf. on Neural Networks 2003, July 20–24, pp. 2354–2359. Portland (2003)
42. Xu, L.: Independent component analysis and extensions with noise and time: A Bayesian Ying-Yang learning perspective. Neural Inf. Process. Lett. Rev. **1**, 1–52 (2003)
43. Xu, L.: Temporal BYY encoding, Markovian state spaces, and space dimension determination. IEEE Trans. Neural Netw. **15**, 1276–1295 (2004)
44. Xu, L.: Advances on BYY harmony learning: information theoretic perspective, generalized projection geometry, and independent factor auto-determination. IEEE Trans. Neural Netw. **15**, 885–902 (2004)
45. Xu, L.: Bayesian Ying Yang learning (I): a unified perspective for statistical modeling. In: Zhong, L. (ed.) Intelligent Technologies for Information analysis, pp. 615–659. Springer (2004)
46. Xu, L.: One-Bit-Matching ICA Theorem, convex–concave programming, and combinatorial optimization. Lecture Notes in Computer Science, Advances in Neural Networks, vol. 3496, pp. 5–20. Springer-Verlag (2005)
47. Xu, L.: One-Bit-Matching theorem for ICA, convex-concave programming on polyhedral set, and distribution approximation for combinatorics. Neural Comput. **19**, 546–569 (2007)
48. Xu, L.: A trend on regularization and model selection in statistical learning: A Bayesian Ying Yang learning perspective. In: Duch, M. (ed.) Challenges for Computational Intelligence, pp. 365–406. Springer-Verlag (2007)
49. Xu, L.: A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving. Pattern Recognit. **40**, 2129–2153 (2007)
50. Xu, L.: Bayesian Ying Yang learning. Scholarpedia **2**(3), 1809 (2007). http://scholarpedia.org/article/BayesianYingYangLearning
51. Xu L.: From normalized RBF networks to subspace based functions. In: Soria, E., Martín, J.D., Magdalena, R., Martínez, M., Serrano, A.J. (eds.) To Appear in Handbook of Research on Machine Learning Applications. IGI Global (formerly Idea Group Publishing) (2008a)
52. Xu, L.: Bayesian Ying Yang system, best harmony learning, and Gaussian manifold based family. In: Zurada, J.M. (ed.) Computational Intelligence: Research Frontiers, WCCI2008 Plenary/Invited Lectures, LNCS5050, pp. 48–78 (2008b)
53. Xu, L., Jordan, M.I.: On convergence properties of the EM algorithm for Gaussian mixtures. Neural Comput. **8**(1), 129–151 (1996)
54. Xu, L., King, I.: A PCA approach for fast retrieval of structural patterns in attributed graphs. IEEE Trans. Syst. Man Cybernet. B **31**(5), 812–817 (2001)
55. Xu, L., Klasa, S.: A PCA like rule for pattern classification based on attributed graph. Proc. 1993 Intl. Joint Conf. on Neural Networks, Oct. 1993, pp. 1281–1284. Nagoya, Japan (1993)

56. Xu, L., Krzyzak, A., Oja, E.: Rival penalized competitive learning for clustering analysis, RBF net and curve detection. IEEE Trans. on Neural Netw. **4**, 636–649. Its early version on Proc. of 11th ICPR92, vol. I, pp. 672–675 (1992 and 1993)
57. XU, L.: Rival penalized competitive learning. Scholarpedia **2**(8), 1810 Retried from http://www.scholarpedia.org/article/Rival_penalized_competitive_learning