Xu *Applied Informatics* (2015) 2:5
DOI 10.1186/s40535-015-0008-4

Applied Informatics
a SpringerOpen Journal

RESEARCH Open Access

Further advances on Bayesian Ying-Yang harmony learning

Lei Xu[1,2]

Correspondence:
lxu@cse.cuhk.edu.hk
[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China
[2] Department of Computer Science and Engineering, The Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, SEIEE Building 3, 800 Dongchuan Road, Minhang District, 200240 Shanghai, China

Abstract

After a short tutorial on the fundamentals of Bayes approaches and Bayesian Ying-Yang (BYY) harmony learning, this paper introduces new progresses. A generic information harmonising dynamics of BYY harmony learning is proposed with the help of a Lagrange variety preservation principle, which provides Lagrange-like implementations of Ying-Yang alternative nonlocal search for various learning tasks and unifies attention, detection, problem-solving, adaptation, learning and model selection from an information harmonising perspective. In this framework, new algorithms are developed to implement Ying-Yang alternative nonlocal search for learning Gaussian mixture and several typical exemplars of linear matrix system, including factor analysis (FA), mixture of local FA, binary FA, nonGaussian FA, de-noised Gaussian mixture, sparse multivariate regression, temporal FA and temporal binary FA, as well as a generalised bilinear matrix system that covers not only these linear models but also manifold learning, gene regulatory networks and the generalised linear mixed model. These algorithms are featured with a favourable nature of automatic model selection and a unified formulation in performing unsupervised learning and semi-supervised learning. Also, we propose a principle of preserving multiple convex combinations, which leads alternative search algorithms. Finally, we provide a chronological outline of the history of BYY learning studies.

Keywords: Automatic model selection; Lagrange; Variety preservation; Ying-Yang alternation; De-noised Gaussian mixture; Factor analysis; Local factors; Binary factors; nonGaussian factors; Temporal factors; Multivariate regression; Bilinear matrix system; Linear mixed model

Background

Bayes approach and automatic model selection

Learning in an intelligent system is featured by three levels of inverse problems, for which details are referred to Sect. 1 of (Xu 2010a,b). To be self-contained, we make a brief overview on typical learning tasks and approaches from such a viewpoint, with help of the illustration in Figure 1.

Learning tasks associated with the front level can be viewed from a perspective of learning a mapping $x \to y$, called representative model, by which an observed sample x in a visible domain X is mapped into its corresponding encoding y as a signal or inner code to perform a task of problem solving, such as abstraction, classification, inference and control. Existing learning methods for a representative model can be roughly divided into two groups as follows:

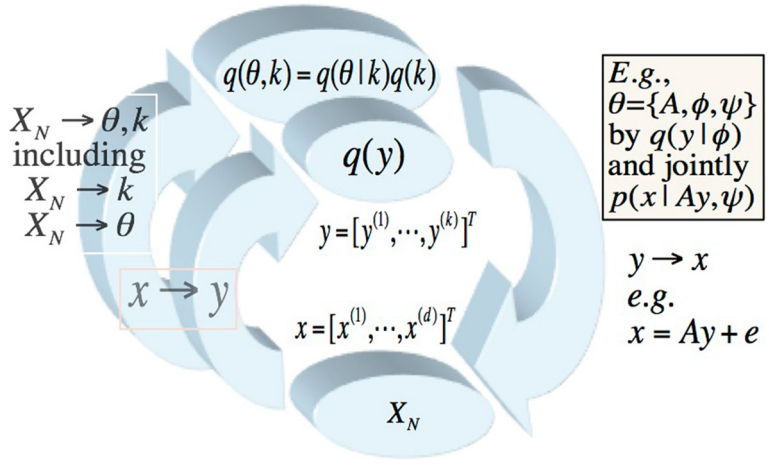

Figure 1 Learning studies from a viewpoint of three levels of inverse problems.

(1) One is featured by learning a mapping $x \to y$ according to whether a principle is satisfied by the resulted inner encodings of y, while not explicitly taking the other directional mapping $y \to x$ in consideration. One exemplar family is featured by a linear mapping $y = Wx$ that transforms x into y of independent components, such as principal component analysis (PCA) and independent component analyses (ICA) (Xu 2003a). The other widely studied family is supervised learning by a linear or a nonlinear mapping that makes samples of y to approach the desired target samples.

(2) The other group is featured by learning a mapping $x \to y$ as an inverse of a given mapping $y \to x$ that describes how observed samples are generated. Some efforts aim at that the cascade of $x \to y$ and $y \to x$ implement a unitary transform $x \to x$, as often encountered in adaptive control. Most of studies consider $y \to x$ in a probabilistic sense by $q(x|y)$ together with y described by $q(y)$. Accordingly, $x \to y$ is either directly the Bayesian inverse of $q(x|y)q(y)$ or its certain approximation.

Typically, the mapping $y \to x$ in the front level is unknown and should be learned from a given set $X_N = \{x_t\}_{t=1}^N$ of samples, which is also called generative learning. Usually, the corresponding distribution structure (or called generative models) is designed according to types of applications. One widely studied structure is the linear system shown in Figure 1. As to be further addressed in the subsequent sections, this structure not only covers subspace methods, Gaussian mixture, factor analysis and its extensions to binary or nonGaussian factors but also can be further generalised to many others.

The generative learning task is estimating $\theta = \{\psi, \phi\}$ in the pre-designed distributions $q(x|y,\psi)$ and $q(y|\phi)$. One most widely used principle is the maximum likelihood, that is,

$$\theta^* = arg\max_{\theta} L(\theta),\ L(\theta) = \sum_{t=1}^{N} \ln q(x_t|\theta),\ q(x|\theta) = \int q(x|y,\psi)q(y|\phi)dy. \tag{1}$$

Though it can be implemented directly by a gradient-based algorithm, an effective alternative is called the expectation-maximisation (EM) algorithm (Dempster et al. 1977) that alternatively implements its E step for $x \to y$ by the following Bayes inverse:

$$p(y|x) = p(y|x,\theta^{old}),\ p(y|x,\theta) = \frac{q(x|y,\psi)q(y|\phi)}{q(x|\theta)}, \tag{2}$$

and its M step that updates θ by

$$\theta^{new} = arg\max_\theta L(\theta, \theta^{old}), \tag{3}$$
$$L(\theta, \theta^{old}) = \sum_{t=1}^{N} \int p(y|x_t, \theta^{old}) \ln [\, q(x_t|y, \psi) q(y|\phi)] dy.$$

This EM iteration is guaranteed to converge to a local maximum of $L(\theta)$ without requiring any learning stepsize, while the gradient-based algorithm needs an appropriate learning stepsize that results in learning instability if the size is too big or a very slow convergence if the size is too small. Moreover, the EM algorithm keeps the constraints of Gaussian mixture satisfied and demonstrates a super-linear convergence rate, with further details referred to Xu and Jordan (1996).

In many applications, the computation of $p(y|x, \theta)$ is intractable. The variational method is proposed to maximise a lower bound of $L(\theta)$ (Dayan et al. 1995; Jordan et al. 1999). Precisely, estimating θ by Equation 1 or Equation 2 implements another inverse problem $X_N \to \theta$ in the second level shown in Figure 1, on which all the levels share the same $q(x|y, \psi)$ that maps y, ψ (a part of θ), and also inclusively k all together to describe how observed samples of x are generated. Similarly, we may consider $X_N \to \theta$ by a Bayes inverse $p(\theta|X_N)$ of $q(x|\theta)q(\theta|k)$. However, its computation is intractable. Instead, we get $X_N \to \theta$ by the following maximum posterior (MAP) (or called the classic or naive Bayes learning):

$$\theta^* = arg\max_\theta [\, L(\theta) + \ln q(\theta|k)] \,. \tag{4}$$

How to use a priori $q(\theta|k)$ is a topic that has a long history and has been considered from several aspects. The classic Bayes school uses different parametric distributions on different parts of θ according to the natures of learning tasks and empirical experiences. Typical examples are those of conjugate priors (Diaconis and Ylvisaker 1979; Ntzoufras and Tarantola 2013). Extensive studies along this line have been made in the machine learning literature, especially on Dirichlet-multinomial for Gaussian mixture. Related studies also include those on multivariate linear regression and extensions. When Gaussian priori is used on each regression coefficient, learning by Equation 4 implements the ridge regression (Hoerl 1985) and Tikhonov regularisation (Tikhonov et al. 1995). When Laplace priori is used on each regression coefficient, learning by Equation 4 implements LASSO regression (Tibshirani 1996) or called sparse learning.

Another Bayes school prefers to use a non-informative priori. For a parameter varies on a compact support, such a priori is simply a uniform distribution. However, there is no such a uniform distribution on an infinite large support. Typically, a non-informative improper distribution $q(\theta|k)$ is used under the name of Jeffery priori (Jeffreys 1946), which has been widely used in the machine learning literature too. Also, there are some efforts that attempt to blend the two schools, e.g. the Jeffery priori is jointly used with a proper priori by the minimum message length (MML) method (Figueiredo and Jain 2002; Wallace and Dowe 1999). Moreover, there is also one effort called induced bias cancellation (IBC), by which the use of a priori is to cancel an implicit prior induced from using a learning model on a finite size of samples, e.g. see Eqs (20) and (21) in Xu (2000a) and also Sect. 3.4.3 in Xu (2007a). Interestingly, as addressed on page 304 of Xu (2010a), this IBC may be regarded as a degenerated but easy computing approximation of the normalised maximum likelihood (NML) that is obtained from a mini-max principle (Barron et al. 1998), which takes a key role in the recent developments of the MDL encoding.

One critical weak point of learning by Equation 4 is prone to a bad priori because $q(\theta|k)$ takes an important position that is equal to the empirical estimator via $L(\theta)$. To mitigate such a bad effect, the up-to-date Bayes studies prefer to consider the following one:

$$k* = argmax_k L(X_N, k),\ L(X_N, k) = \sum_{t=1}^{N} \ln q(x_t|k),$$
$$q(x|k) = \int q(x|\theta)q(\theta|k)d\theta. \tag{5}$$

It actually implements the third level inverse $X_N \rightarrow k$ (In Figure 1 there are merely two levels because the 2nd and 3rd levels are merged in a consideration of automatic model selection to be addressed after Equation 7). This task is usually called model selection. However, the integral over θ is computationally intractable, which is typically handled with help of some approximating technique. One classical one is made by the Bayesian information criterion (BIC) (Schwarz 1978) that approximately turns $L(X_N, k)$ into

$$L(X_N, k) \approx L(X_N, \theta^*) - 0.5klnN, \tag{6}$$

by which learning is made via a two-stage implementation. The first stage enumerates all possible numbers of k to obtain a set of candidate models featured by different values of k, and estimates θ^* by Equation 1 for each candidate. At the second stage, we select the best candidate by Equation 5 with $L(X_N, k)$ given by Equation 6. In implementation, the minimum description length (MDL) (Rissanen 1978) is actually equivalent to this BIC. There are also a number of other variants of $L(X_N, k)$ available in the literature, e.g. another classic one is Akaike's information criterion (AIC) (Akaike 1974, 1987).

However, a two-stage implementation suffers from a huge computation because it requires parameter learning for each candidate. Also, estimating θ^* by Equation 1 will become less reliable when the component number k is large and thus incurs for more free parameters.

This problem is tackled by considering a learning process or principle with a nature of automatic model selection, e.g. discarding extra hidden dimensions of y in Figure 1. With k initialised large enough, a learning principle demonstrates such a nature with the following two features:

- there is an indicator $\Psi_\pi(\theta)$ on θ or its subset, based on which a particular subset π can be effectively discarded if we have

$$\Psi_\pi(\theta) \rightarrow 0, \tag{7}$$

  e.g. $\Psi_\pi(\theta)$ is the variance of $y^{(i)}$ in Figure 1.
- in learning implementation there is an intrinsic mechanism that leads to Equation 7 when the corresponding structure is redundant and thus can be effectively discarded.

Such automatic model selection is actually made during implementing the inverse problem $X_N \rightarrow \theta$. Thus, we merge the corresponding two levels in Figure 1 because it combines both the inverse problem $X_N \rightarrow \theta$ and the inverse problem $X_N \rightarrow k$.

For the existing studies, there are three roads towards automatic model selection. One is a heuristic road, featured by an early effort called Rival Penalised Competitive Learning (RPCL) made in the early 1990s (Xu et al. 1992, 1993), which gets an appropriate number k of clusters automatically determined during learning.

The second road is getting an aid from appropriate priories. For examples, learning by Equation 4 demonstrates such a nature by using either a Laplace priori in sparse learning (Tibshirani 1996) or jointly the Jeffery priori and a proper priori by the minimum message length (MML) (Figueiredo and Jain 2002). Another example is the Variational Bayes (VB) (Corduneanu and Bishop 2001; McGrory and Titterington 2007) that approximately maximises a lower bound of $L(X_N, k)$ in Equation 5 via learning the hyper parameters in both a priori $q(\theta|k)$ and an approximate posteriori $p(\theta|X_N)$.

The third road is the following BYY harmony learning (BYY) that was firstly proposed in 1995 (Xu 1995) and subsequently developed systematically, which provides a general framework for learning $X_N \to \theta$ and $X_N \to k$ under the BYY best harmony principle.

Bayesian Ying-Yang harmony learning

We reformulate Figure 1 into a general probabilistic formulation, resulting in Figure 2. We use $R = \{Y, \{\theta\}\}$ to summarise three levels of inner representation and $P(R|X)$ for the mapping $X \to R$ that consists of the mappings $X \to Y$ and $X \to \theta$. On the one hand, we have $p(X, R) = p(R|X)p(X)$ to describe the joint distribution of X, R, which is featured by a visible domain X (called Yang according to Chinese ancient philosophy) and a transformation from samples of observations into inner codes (in a function like a male animal and thus also called Yang according to the same Chinese philosophy). Jointly we called $p(X, R) = p(R|X)p(X)$ a Yang structure or machine. On the other hand, we have a Ying structure or machine $q(X, R) = q(X|R)q(R)$ to describe also the joint distribution of X, R, which is featured by $q(R)$ to describe the invisible (thus called Ying) domain R for inner representation and a transformation from inner codes to observations (in a function like a female animal and thus called Ying). The paired Ying-Yang structures formulates a system called Bayesian Ying-Yang (BYY), in a tribute to the Chinese ancient philosophy.

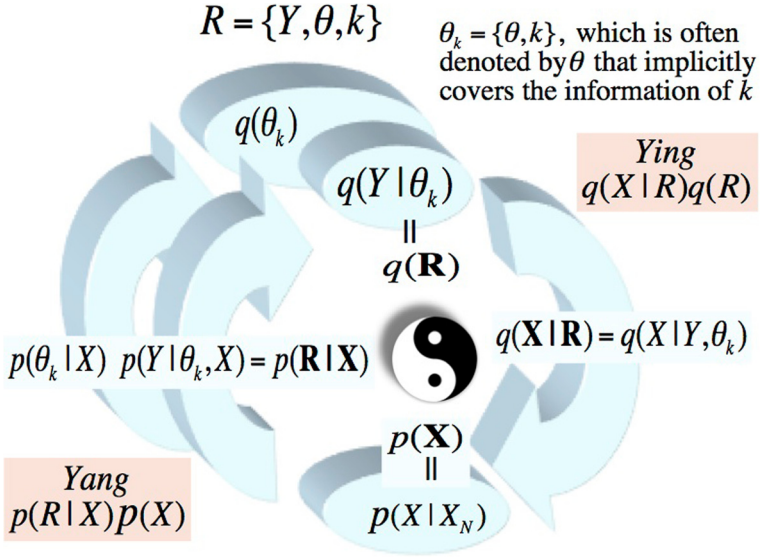

Figure 2 Bayesian Ying-Yang system.

The task of learning a BYY system starts from its structure design. That is, we need to give each of four component distributions a specific mathematical structure. Usually, $p(X)$ comes from a given set X_N of samples as follows:

$$p_h^N(X) = \prod_{t=1}^{N} G(x|x_t, h^2 I), \text{ especially } p_0^N(X) = \delta(X - X_N), \tag{8}$$

where $G(x|\mu, \Sigma)$ denotes a Gaussian density with the mean vector μ and the covariance matrix Σ.

For the rest of the three components, we start at designing the structures of $q(X|R)$ and $q(R)$, based on which we further design the structure of $p(R|X)$ that is typically a sort of an inverse of the Ying $q(X|R)q(R)$ machine. This is consistent to the Ying-Yang philosophy, according to which Ying is primary and comes first, while the Yang is secondary and bases on the Ying.

The design of each component is guided by the corresponding one of the following three principles (Xu 2009):

- A principle of least redundant representation for $q(R)$.
- A principle of divide-conquer for $q(X|R)$.
- A principle of Ying-Yang uncertainty conversation or variety preservation for $p(R|X)$.

Further details are referred to Sect.4.2 of (Xu 2010a) and Sect.3.2 of (Xu 2012a). The first two principles are adopted from the existing studies, while the third is specific to the BYY system. In a compliment to the Yin-Yang philosophy, it requires that Yang machine preserves a dynamic range to appropriately accommodate uncertainty or information contained in the Ying machine. That is, we have $U(p(X,R)) = U(q(X,R))$ under a uncertainty measure $U(p)$ as shown within the table of Figure four(a) in Xu (2009).

Given a BYY system designed, the unknown values of all variables in $R = \{Y, \{\theta\}\}$ are learnt according to a Ying-Yang best harmony principle. Mathematically, it is equivalent to make $p(R|X)p(X)$ and $q(X|R)q(R)$ become a best matching pair in a most compact form with a least complexity, which is achieved via maximising the following harmony functional:

$$\begin{aligned} H(p||q) &= \int p(R|X)p(X) \ln[\, q(X|R)q(R)]\, dXdR, \\ &\text{subject to } U(p(X,R)) = U(q(X,R)). \end{aligned} \tag{9}$$

On the one hand, maximising $H(p||q)$ forces the Ying $q(X|R)q(R)$ to match the Yang $p(R|X)p(X)$. There are always certain structural constraints imposed on the Ying-Yang structures and also a constraint comes from $p(X) = p_h^N(X)$ by Equation 8 on a finite size of samples, because of which a perfect equality $q(X|R)q(R) = p(R|X)p(X)$ may not be really reached but still be approached as close as possible. At this equality, $H(p||q)$ becomes the negative entropy that describes the complexity of the BYY system. Further maximising it will decrease the system complexity and thus provides an ability for determining an appropriate k.

As addressed in Sect.4.1 of Xu (2010a), this principle is spelled as Ying-Yang best harmony from a perspective that Ying and Yang both adapt each other to reach the best agreement in a most tacit way (consuming a least amount of effort made in information communication), which can be better understood by rewriting Equation 9 into

$$\begin{aligned} H(p||q) &= H_{R|X} - KL(p(R|X)p(X)\|q(X|R)q(R)), \\ H_{R|X} &= \int p(R|X)p(X)\ln[\,p(R|X)p(X)]\,dXdR, \\ &\text{where } KL(p\|q) = \int p(u)\ln\frac{p(u)}{q(u)}du. \end{aligned} \tag{10}$$

Maximising $H(p||q)$ consists of minimising the second term for a best matching or agreement between the Ying-Yang pair and of minimising the first term for a least amount of information to be communicated from the Yang to the Ying towards an agreement.

The novelty and salient features of Equation 9 may also be observed from other aspects. Further details are referred to Sect. 4.1 in Xu (2010a) and Sect. 4.2.3 in Xu (2012a). Shown in Table 1 are recent applications and empirical studies of the BYY harmony learning.

Currently, the implementation of BYY harmony learning may suffer a dilemma of suboptimal solution versus learning instability. It is this dilemma that motivates the progresses introduced in this paper, which are outlined as follows:

- A Lagrange implementation of the principle of variety preservation is proposed for learning the Yang structure, with a new Ying-Yang alternation nonlocal search obtained and the abovementioned dilemma removed.
- An information harmonising perspective for BYY harmony learning such that the tasks of attention, detection, problem-solving, adaptation, learning and model selection are integrated in a concise formulation.
- Learning algorithms that implement Ying-Yang alternative nonlocal search for learning GMM, FA, local FA, binary FA, nonGaussian FA, de-noised GMM, temporal FA, temporal binary FA and sparse multivariate regression, as well as a generalised bilinear matrix system that covers not only these linear models but also manifold learning, gene regulatory networks and the generalised linear mixed model, with a favourable nature of automatic model selection and a unified formulation in performing unsupervised and semi-supervised learning.
- A principle of preserving multiple convex combinations for implementing BYY harmony learning, which leads another type of Ying-Yang alternative nonlocal search algorithms.

Finally, at the end of this paper, a chronological outline is given on the innovative time points in the history of BYY harmony learning studies.

Methods

BYY harmony learning: Lagrange Ying-Yang alternation

Ignoring a priori $q(\theta)$, we simplify the best harmony of $H(p||q)$ by Equation 9 into

$$\begin{aligned} &\max_\theta H(\theta)\ _{subject\ to\ U(p(X,Y))=U(q(X,Y))}, \\ &H(\theta) = \int p(Y|X)p_h^N(X)\ln[\,q(X|Y,\theta)q(Y|\theta)]\,dY\,dX, \end{aligned} \tag{11}$$

Table 1 Recent BYY applications and empirical studies

Papers	Outcomes
Shi et al. (2011a)	A comparative investigation has been made on three Bayesian related approaches, namely, variational Bayesian (VB), minimum message length (MML) and BYY harmony learning, through the task of learning Gaussian mixture model (GMM) with an appropriate number of components automatically determined. On not only simulated GMM data sets but also the Berkeley segmentation database of real world images, extensive experiments have shown that BYY harmony learning considerably outperforms both MML and VB regardless whether a Jeffreys prior or a conjugate Dirichlet-Normal-Wishart (DNW) prior is used and whether the hyper-parameters of DNW prior are further optimised.
Tu and Xu (2011a)	A further comparison has been made on factor analysis (FA) with an appropriate number of factors determined, and extensive experiments have shown that not only BYY and VB outperform AIC, BIC and DNLL but also BYY outperforms VB considerably. Moreover, using VB to optimise the hyper-parameters of priors deteriorates the performances while using BYY for this purpose can improve the performances.
Tu and Xu (2011b)	Empirical comparisons have also been made on factor selection performances of AIC, BIC, Bozdogan's AIC, Hannan-Quinn criterion, Minka's (MK) criterion, Kritchman & Nadler's hypothesis tests (KN), Perry & Wolfe's MiniMax rank (MM) and BYY harmony learning, by varying signal-to-noise ratio (SNR) and training sample size N. It has been shown that AIC and BYY harmony learning, as well as MK, KN and MM, are relatively more robust than the others against decreasing N and SNR, and BYY is superior for a small size N.
Shi et al. (2014); Tu and Xu (2014)	Extension of FA has been made to binary FA with automatic factor selection. Again, it is empirically shown that BYY outperforms VB and BIC. Also, efforts of (Shi et al. 2014) extend the studies of (Shi et al. 2011a) and two FA parameterizations in (Tu and Xu 2011a) into Mixture of Factor Analyzers (MFA) and Local Factor Analysis (LFA) for the problem of automatically determining the component number and the number of factors of each FA. On not only a wide range of synthetic experiments but also real applications of face recognition, handwritten digit image clustering and unsupervised image segmentation, it has been also shown that BYY outperforms VB reliably on both MFA and LFA.
Chen et al. (2014)	Further developments of (Shi et al. 2011a) have also been made to avoid some learning instability (see *Remarks* at the bottom of this table), an implementation of BYY harmony learning by either a projection-embedded algorithm or the algorithm by Table 3 in this paper needs no priori but outperforms not only MML with Jeffreys prior and VB with Dirichlet-Normal-Wishart prior but also BYY with these priors given in (Shi et al. 2011a). On the Berkeley segmentation data set, the semantic image segmentation performances have shown that BYY outperforms not only MML, VB, BYY-Jef and BYY-DNW but also three leading image segmentation algorithms, namely gPb-owt-ucm, MN-Cut and Mean Shift.

Remarks.
For the first three items above, the BYY harmony learning is implemented via one of two techniques as follows:
(a) Gradient-based local search that needs a small step size to be pre-specified. If this step size is too small, learning is too slow and easy to get stuck at a local optimal solution. If this step size is too big, learning becomes unstable.
(b) Ying-Yang nonlocal search that consists of an expectation-maximisation (EM) like two steps, with no learning stepsize but a correcting δ in E step. For GMM, it follows from Eq. (11) in (Xu L 2010a) that E step of the EM algorithm that allocates x_t to the ℓ th Gaussian by $p(\ell|x_t,\theta^{old})$ is replaced by $p(\ell|x_t,\theta^{old})+\delta(\theta^{old})$ with an approximation that may cause learning instability, also see Equations 88 and 89 for details. □

where the above constraint is a simplification of the counterpart in Equation 9. One example is considered in Sect. 4.1 in Xu (2010a) and Sect. 4.2.3 in Xu (2012a), featured with the following counterpart without considering the component $p(X)$:

$$p(Y|X)=q(Y|\theta,X),\ q(Y|\theta,X)=\frac{q(X|Y,\theta)q(Y|\theta)}{q(X|\theta)},$$
$$q(X|\theta)=\int q(X|Y,\theta)q(Y|\theta)dY. \tag{12}$$

Even earlier in 2007, another example is given by Eq.(72) in Xu (2007a), under the name of equal covariance with $U(p(X,Y))=U(q(X,Y))$ denoting that the Yang preserves the covariance of $q(X,Y)$.

The existing algorithms for $\max_\theta H(\theta)$ directly impose the constraint $U(p(X,Y)) = U(q(X,Y))$, which makes learning suffer a dilemma of either local optimal solution or some learning instability, see the remarks in Table 1.

In this paper, we indirectly consider a relaxation of $U(p(X,Y)) = U(q(X,Y))$ via considering $KL(p(X,Y)\| q(X,Y)) = 0$ as a Lagrange constraint (since $KL(p\|q) \geq 0$ becomes zero at the target $p = q$), resulting in the following augmented maximisation:

$$\max_\theta H_L(\theta),\ H_L(\theta) = H(\theta) - \eta KL(p(Y|X)p(X)\|q(X|Y,\theta)q(Y|\theta)) \leq H(\theta), \tag{13}$$

where $\eta > 0$ is a Lagrange coefficient. A nonzero value η will relax the target $KL(p(X,Y)\|q(X,Y)) = 0$. The smaller the value η is, it becomes more relaxed, or vice versa.

Moreover, Equation 13 can be rewritten into

$$\begin{aligned} &\max_\theta H_L(\theta),\ H_L(\theta) = (1+\eta)H(\theta) + \eta[E_{Y|X} + E_X(h)], \\ &E_{Y|X} = -\int p(Y|X)p_h^N(X)\ln p(Y|X)dY\ dX,\ E_X(h) = -\int p_h^N(X)\ln p_h^N(X)dX. \end{aligned} \tag{14}$$

Given $p(Y|X) = p_{Y|X}^{old}$ fixed, $\max_\theta H_L(\theta)$ becomes

$$\theta^{new} = arg\max_\theta H(\theta)_{p_{Y|X}=p_{Y|X}^{old}},\ h^{new} = arg\max_h H_L(\theta)_{p_{Y|X}=p_{Y|X}^{old}}, \tag{15}$$

with $H_L(\theta^{new}) \geq H_L(\theta^{old})$.

Given $\theta = \theta^{new}, h = h^{new}$, maximising $H_L(\theta)$ subject to $\int p(Y|X)dY = 1$ with respect to a free $p(Y|X)$ results in

$$p_{Y|X}^{new} = \frac{[q(X|Y,\theta^{new})q(Y|\theta^{new})]^{(1+1/\eta)}}{\int [q(X|Y,\theta^{new})q(Y|\theta^{new})]^{(1+1/\eta)}dY}, \tag{16}$$

which keeps $H_L(\theta)$ to be nondecreasing too.

Therefore, alternatively updating Equations 15 and 16 makes $H_L(\theta)$ monotonically nondecrease and finally converge. That is, learning stability is guaranteed.

Given h fixed, the term $E_X(h)$ can be ignored because it is irrelevant to updating θ and $p(Y|X)$. With help of $E_X(h)$, an appropriate h can be estimated in a way similar to ones summarised in Sect.2 of (Xu L 2003b).

Without losing generality, we consider Equation 14 at the special case $h = 0$ and get

$$\begin{aligned} &\max_\theta H_L(\theta),\ H_L(\theta) = (1+\eta)H(\theta) + \eta E_{Y|X}, \\ &H(\theta) = \int p(Y|X_N)\ln[q(X_N|Y,\theta)q(Y|\theta)]\,dY, \\ &E_{Y|X} = -\int p(Y|X_N)\ln p(Y|X_N)dY, \end{aligned} \tag{17}$$

from which we get two types of detailed implementation according to the types of variables in Y.

When the variables in Y are discrete valued, the integral over Y becomes summation. It follows from Equations 15 and 16 that we are led to the general procedure for Ying-Yang alternative implementation given in Algorithm 1.

Algorithm 1 Ying-Yang alternative procedure (A)

Require: initialise $\theta^{old}, \eta^{old}, p^{old}_{Y|X_N}$ in equal probability over the domain of Y.

Repeat the following two steps **until** converged:

***Ying*-Step:** *get* $\theta^{new} = arg\max_\theta H(\theta)_{p_{Y|X}=p^{old}_{Y|X}}$.

trimming: descard a subset $\pi \subset \theta$ if $\Psi_\pi(\theta) \to 0$.

***Yang*-Step**: get $p^{new}_{Y|X} = \frac{[q(X|Y,\theta^{new})q(Y|\theta^{new})]^{(1+1/\eta)}}{\sum_Y [q(X|Y,\theta^{new})q(Y|\theta^{new})]^{(1+1/\eta)}}$.

Remarks:

(a) The Ying step shares a same format of the M-step of the popular EM (expectation and maximisation) algorithm. Ignoring the part of *trimming*, we may simply obtain the M-step of the EM algorithm for the maximum likelihood learning.

(b) The part of *trimming* is associated with automatic model selection nature. As previously addressed about Equation 7, such a nature makes $\Psi_\pi(\theta) \to 0$ and thus the corresponding subset π can be discarded.

(c) The difference of this algorithm from the EM lies in the Yang step, featured by η, which makes $p^{new}_{Y|X_N}$ become more selective for automatic model selection. When $\eta = \infty$, the Yang step will degenerate into the E-step.

(d) η is controlled as described by Equation 27 and the discussions thereafter.

When the variables in Y are real valued, the integral over Y becomes intractable, for which we seek the help of the following Taylor expansion around u^* up to the second order :

$$\begin{aligned} \max_{\eta_u} \int p(u)Q(u)du \approx Q(u^*) - \tfrac{1}{2} Tr[\,\Gamma_u \Pi_{u^*}\,], \\ u^* = \arg\max_u Q(u),\ \Pi_u = -\tfrac{\partial^2 Q(u)}{\partial u \partial u^{\mathrm{T}}}, \end{aligned} \tag{18}$$

where η_u, Γ_u are the mean and the covariance of $p(u)$.

From Equations 17 and 18, we approximately have

$$\begin{aligned} H(\theta) \approx \pi(X_N, Y_*, \theta) - \tfrac{1}{2} Tr[\,\Gamma^Y_{X_N} \Pi^Y_{X_N}\,], \\ \pi(X_N, Y, \theta) = \ln[\,q(X_N|Y,\theta)q(Y|\theta)\,], \\ Y_* = arg\max_Y \pi(X_N, Y, \theta),\ \Pi^Y_{X_N} = -\tfrac{\partial^2 \pi(X_N,Y,\theta)}{\partial vec(Y) \partial vec(Y)^T}, \\ \Gamma^Y_X = Cov_{p(vec(Y)|X)} vec(Y), \end{aligned} \tag{19}$$

where $Cov_{p(u)}u$ denotes the covariance matrix of $p(u)$ and $vec(A)$ denotes the vector obtained by stacking the column vectors of A one by one.

Maximising the above $H(\theta)$, we get another type of Ying-Yang alternative implementation, as summarised in Algorithm 2.

Given $Y_* = Y^{old}_*$, $\Gamma^Y_{X_N} = \Gamma^{Y\,old}_{X_N}$, the counterpart of Equation 15 becomes simply

$$\theta^{new} = arg\max_\theta H(\theta), \tag{20}$$

which acts as the Ying step of Algorithm 2.

Algorithm 2 Ying-Yang alternative procedure (B)

Require: initialise θ^{old}, η^{old}, Y^{*old}, $\Gamma_{X_N}^{Y\,old}$.

Repeat the following two steps **until** converged:

***Ying*-Step:** perform Equation 20 with $H(\theta)$ by Equation 19, i.e.

$\theta^{new} = arg\max_\theta H(\theta)$,

trimming: descard a subset $\pi \subset \theta$ if $\Psi_\pi(\theta) \to 0$.

***Yang*-Step**: perform Equation 21, i.e. get

$Y_*^{new} = arg\max_Y \pi(X_N, Y, \theta^{new})$, $\Gamma_X^{Y\,new} = \frac{\eta}{1+\eta}\Pi_X^{Y\,new\,-1}$,

with η^{new} controlled as Remark (d) in Algorithm 1.

Remarks: Its relation to the EM algorithm is similar to Algorithm 1. Again, η makes a difference via sharping the covariance $\Gamma_{X_N}^{Y\,new}$ to become more selective.

Given $\theta = \theta^{new}$ and Y_*^{new}, the counterpart of Equation 16 become simply

$$\Gamma_X^{Y\,new} = arg\max_\theta[\,(1+\eta)H(\theta) + \eta E_{Y|X}] = \frac{\eta}{1+\eta}\Pi_X^{Y\,new\,-1}, \tag{21}$$

where $E_{Y|X} \approx 0.5 d_Y \ln(2\pi e) + 0.5\ln|\Gamma_X^Y|$ is obtained by approximately regarding it as the entropy of a Gaussian density with a covariance matrix $\Gamma_{X_N}^Y$.

Another insight on Equation 13 comes from observing $q(Y|\theta,X)q(X|\theta) = q(X|Y,\theta)q(Y|\theta)$ from Equation 12, by which $KL(p(Y|X)p(X)\|q(X|Y,\theta)q(Y|\theta))$ becomes

$$\begin{aligned} &KL(p(Y|X)p(X)\|q(Y|\theta,X)q(X|\theta)) \\ &= \int p(X)KL(p(Y|X)\|q(Y|\theta,X))dX + KL(p(X)\|q(X|\theta)). \end{aligned} \tag{22}$$

With $p(X) = p_0^N(X)$ by Equation 8 and with $q(Y|\theta,X)$, $q(X|\theta)$ by Equation 12, we can rewrite Equation 13 into

$$\max_\theta H_L(\theta),\ \ H_L(\theta) = H(\theta) - \eta KL(p(Y|X_N)\|q(Y|\theta,X_N)) + \eta\ln q(X_N|\theta), \tag{23}$$

from which we observe that the maximisation of $H_L(\theta)$ consists of not only a best Ying-Yang harmony but also a degree η of jointly a top-down maximum likelihood learning and a bottom-up best matching between the posteriors $p(Y|X_N)$ and $q(Y|\theta,X_N)$.

The maximisation of the above second and third terms is exactly what has been widely called variational learning (Corduneanu and Bishop 2001; Jordan et al. 1999; McGrory and Titterington 2007), which is equivalent to the Ying-Yang best matching, as previously pointed out in Xu (2010a) (especially see the roadmap in its Figure A2). The sum of two terms may be simply observed from

$$\begin{aligned} -[\,H(\theta) + E_{Y|X}] &= -KL(p(Y|X)\|q(X_N|Y,\theta)q(Y|\theta)) \\ &= \ln q(X_N|\theta) - KL(p(Y|X)\|q(Y|\theta,X)) \le \ln q(X_N|\theta), \end{aligned} \tag{24}$$

which is a degenerated case that does not have the harmonising information flow $H(\theta)$ in the centre of Figure 3.

Next, we consider to drop off the last term in Equation 23, resulting in

$$\begin{aligned} &\max_\theta H_G(\theta), \\ &H_G(\theta) = H(\theta) - \eta KL(p(Y|X_N)\|q(Y|\theta,X_N)) \\ &= -\eta\ln q(X_N|\theta) + (1+\eta)H(\theta) + \eta E_{Y|X} \le H(\theta), \end{aligned}$$

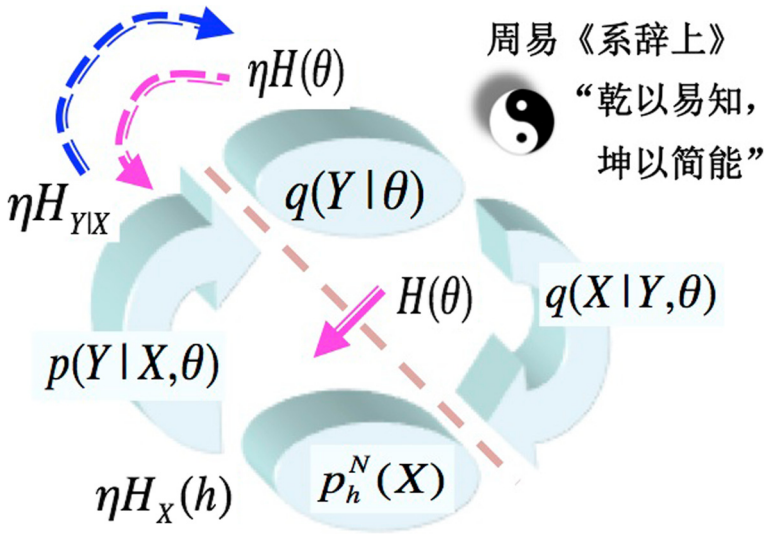

Figure 3 Information harmonising formulation.

which may also be obtained from considering the constraint by Equation 12 in a Lagrange. At the special case $\eta = 1$, we may regard it as counterpart of Equation 24, with a difference in that $H(\theta)$ replaces $\ln q(X_N|\theta)$.

On the other hand, we may also generalise Equation 24 by a Lagrange as follows:

$$\begin{aligned} &\ln q(X_N|\theta) - \eta KL(p(Y|X)\|q(Y|\theta,X)) \\ &= (1-\eta)\ln q(X_N|\theta) + \eta H(\theta) + \eta E_{Y|X} \geq H(\theta) + E_{Y|X}, \end{aligned}$$

which becomes the counterpart of Equation 24 generally instead of only at $\eta = 1$. Alternatively, we may reach a tighter lower bound by an appropriate value of η.

Last but not least, maximising $H_L(\theta)$ by Equations 14 and 17 relates closely to some previous efforts summarised in Table 2.

Table 2 Related studies: KL-η-HL spectrum

Year	Outcomes
1998	The following convex combination with $0 \leq \eta \leq 1$ is heuristically proposed $(1-\eta)KL(p(Y\|X)p(X)\|q(Y\|R)q(Y)) - \eta H(\theta), \quad (A)$ as a criterion for model selection, e.g. see Eq. (49) in Xu (1998a) and Eq. (22) in Xu (1998b). The above equation (A) can be rewritten into a format that is exactly equivalent to $H_L(\theta) = (1+\eta)H(\theta) + \eta E_{Y\|X}$ in Equation 17.
2000	It is further proposed to make $\max_\theta H_L(\theta)$ with $\eta > 0$ monotonically decreased from a big value (i.e. remove the constraint $\eta \leq 1$), see Eq. (23) in Xu (2000a), which is further addressed for learning Gaussian mixture in Xu (2001a), e.g. see paragraphs around its Eq. (42) and Eq. (43).
2003	The above equation (A) has been also reexamined from a perspective of the KL-η-HL spectrum, with details referred to Eqs. (62-64) in (Xu 2003a).

Remarks.
(a) This family is further investigated in 2012 from a perspective of the Yang structure, see Sect. 3.4.2 in Xu (2012a) and especially the parts around its Eq. (46) on a family of the Yang structures. Each of such structures corresponds an inverse of Ying machine in a range from superBayes ($\eta > 0$) to Bayes ($\eta = 0$).
(b) What was discussed in Xu (2012a) is actually a range that also includes a subBayes inverse of Ying machine coming from ($\eta < 0$), that is, superBayes→Bayes→subBayes.
(c) The symbol η was actually λ in the above mentioned studies.
(d) The concept of superBayes versus subBayes may be understood from Equation 16. The two factors of $q(X|Y,\theta)q(Y|\theta)$ are mutually linear for Bayes, superlinear for superBayes and sublinear for subBayes. □

Information harmonising dynamics

According to the Ying-Yang philosophy placed at the upper right corner of Figure 3, the Ying and Yang constitutes a harmony system surviving in an environment, by which the Ying is primary while the Yang has not only a nature of variety but also a good adaptability to both the Ying and its environment. We may not only understand Equations 13, 14 and 17 from a classic perspective but also get new insight on how the Ying and Yang interact dynamically.

The status of Ying-Yang harmony is jointly featured by $H(p||q)$ and the Lagrange quantity η, where $H(p||q)$ is given in Equation 9 or simply $H(p||q) = H(\theta)$ in Equation 11, while η is given in Equations 13 an 14, reflecting an agreement of balance between Ying and Yang in one of the following aspects:

- **(a)** Balance within the Yang domain, i.e. seeking a match between $p_h^N(X)$ by Equation 8 and $q(X_N|\theta) = \int q(X|Y,\theta)q(Y|\theta)dY$, measured by a divergence $-KL(p_h^N(X)\|q(X_N|\theta))$ or equivalently a likelihood $L(\theta) = \ln q(X_N|\theta)$.
- **(b)** Balance along the Yang pathway, i.e. to satisfy the constraint by Equation 12, e.g. measured by $-KL(p(Y|X_N)\|q(Y|\theta,X_N))$.
- **(c)** Balance between Ying-Yang, i.e. both (a) and (b), measured by $KL(p(Y|X)p(X)\|q(X|Y,\theta)q(Y|\theta))$, as in Equation 13.

Here, we focus on the standard cases, i.e., Ying dominated models or the Ying is primary. For some exceptional cases that the Yang is primary, e.g. forward architecture (see Sect.II(C) in Xu (2001b)), we may consider a balance within the Yang domain and a balance via the Yang pathway.

Typically, η could be a monotonically increasing function of a goodness that measures such a balance, while a best Ying-Yang harmony is reached at a balance that the Ying-Yang system has a least complexity.

Quantitatively, the harmonising dynamics remains to be an open topic that demands further investigation. Qualitatively, this dynamics may be roughly depicted via the dynamics of η as follows.

We start at considering two extreme cases. One happens at a bad Ying-Yang balance, featured by

$$\eta \text{ takes a very small value around 0.} \tag{25}$$

The dynamics of maximising $H_L(\theta)$ focuses at maximising $H(\theta)$ that makes $p(Y|\theta,X_N) = \delta(Y - Y^*)$ with $Y^* = arg\max_Y \pi(X_N,Y,\theta)$ become mostly focused and least flexible in order to rapidly satisfy the most urgent need of Ying, that is, the BYY harmony learning degenerates to one special case that is an extension of competitive learning. Though it still works when the resulted $H(\theta)$ is used as a model selection criterion, e.g. see Eq.(10a) in Xu (1996), it becomes prone to an initialisation and poor in automatic model selection because of the winner-take-all (WTA) competition among the inner representations of Y. Therefore, we should not let η_t always stay at a too small value.

The other extreme happens when the Ying-Yang balances well, featured by

$$\eta \text{ takes a very large value} \tag{26}$$

such that $\eta \approx 1 + \eta$. In such cases, maximising $H_L(\theta)$ by Equation 17 actually focuses at maximising $\eta[H(\theta) + E_{Y|X}]$, or equivalently minimising the Kullback divergence $\eta KL(p(Y|X)\|q(X_N|Y,\theta)q(Y|\theta))$ for a Ying-Yang best matching, which makes $p(Y|\theta, X_N)$ tend to Equation 12 and thus enjoy a larger varying range or a big flexibility to cope with new samples. However, the harmonising information $H(\theta)$ in the centre of Figure 3 becomes neglectable, i.e. becoming weak in reducing the system complexity. In such a case, Algorithm 1 and Algorithm 2 become equivalent to the EM algorithm for the maximum likelihood, which is poor in model selection too. This means that the dynamics is approaching an equilibrium as η tends a big value, during which model selection or structure changing is gradually shut off while parameters may still be refined.

In the beginning, a BYY system is given with a pre-designed Ying-Yang structure and usually with all the unknown parameters initialized either randomly or according to a priori knowledge. Thus, the BYY system fits a given set X_N of samples badly, resulting in a poor Ying-Yang balance with a small η value in a way similar to the first extreme case. The dynamics focuses on not only adjusting the structure but also updating the parameters towards a balance with η quickly growing up, which gradually tends to an equilibrium with X_N well described by a Ying-Yang structure in an appropriate complexity.

Surviving in an environment, the BYY system typically stays at one equilibrium of its harmonising dynamics. As the environment changes, the dynamics is featured by performing the following actions:

(A) *Equilibrium and attention* When the system feels familiar with its observations, the dynamics stays at one equilibrium with a big value of η. An unexpected environmental change will make η drop. A large drop will trigger the system's attention to detect environmental novelty. In other words, there is an attention mechanism associated with η.

(B) *Detection and problem-solving* A small drop of η is associated with a deviation from one equilibrium, which causes an incremental of KL. This incremental is associated with actions of detecting objects, recognising patterns and solving problems (e.g. inference or control) by the mapping $X \to Y$ via $p(Y|\theta, X_N)$.

(C) *Adaptation and learning* When the two opposed changes of η and of KL are not big enough such that the value of ηKL may not change considerably, learning will not be triggered and $H_L(\theta)$ by Equation 17 approximately stays unchanged. However, maximising $H_L(\theta)$ will start to minimise KL when the incremental of KL becomes large while η remains a high value, i.e. becoming close to the second extreme case by Equation 26. In this case, the learning made by Algorithm 1 or Algorithm 2 becomes closer to the maximum likelihood learning that merely updates the parameters in the system without a big structural change, that is, no model selection occurs.

(D) *Model selection and structure pruning* A big drop of η will happen when the BYY system faces a largely different environment, i.e. becoming the extreme case $\eta = 0$, the dynamics has to not only adjust the structure but also update the parameters towards a new equilibrium with η brought up quickly.

In a summary, the above actions are featured by a feedback signal η as follows:

$$\eta = g(v),\ v = f(d_M, d_D, d_U),\ \frac{dg(v)}{dv} < 0,\ \frac{\partial f}{\partial d_u} > 0,\ u = M, D, U. \tag{27}$$

Conceptually, η monotonically decreases with a vigilance signal ν, and this ν monotonically increases with d_M, d_D and d_U, where d_M reflects the discrepancy between data X and its counterpart $\hat{X}$ reconstructed by the model, e.g. measured by the negative log-likelihood $-\ln q(X_N|\theta)$ or $KL(p_h^N(X)\|q(X_N|\theta))$, while d_D reflects the deviation of an inner representation Y from the desired Y_d, e.g. measured by the square error Y and its corresponding $\hat{Y}$. Moreover, d_U is a measure that reflects salient occurrences that attract attentions. Further investigation is needed on the detailed forms of d_M, d_D and d_U, as well as the specific form of $g(f(\cdot,\cdot,\cdot))$, which may be considered by nonlinear regression.

As illustrated in Figure 3, the strength η controls the flexibility and adaptability that Yang enjoys, described by an entropy gain $-\eta\int p(Y|X)p_h^N(X)\ln p(Y|X)p_h^N(X)dYdX = \eta[E_{Y|X} + E_X(h)]$. Transferring this information from the Yang to the Ying, the Ying attempts to harmonise the information by updating parameters and modifying its structure to increase an amount of negative entropy $\eta H(\theta)$. Therefore, a net amount of harmonising information $(1+\eta)H(\theta) + \eta[E_{Y|X} + E_X(h)]$ is maximized, by which we are led to Equations 14 and 17.

For a large η, the Yang enjoys a large flexibility to avoid an overfitting of samples and to prepare an adaptability for possible environmental changes. The more flexibility (i.e. $\eta E_{Y|X}$) that the Yang currently enjoys, the larger amount of negative entropy (i.e. $\eta H(\theta)$) is needed for the Ying to manage. When it becomes difficult to manage, the Ying-Yang balance will deteriorate and thus incur for a drop of η to reduce the flexibility of Yang. In other words, there is a negative feedback mechanism that stabilises the dynamics of information harmonising, as illustrated in Figure 4.

Learning Gaussian mixture and learning factor analysis

We start at considering Gaussian mixture as follows

$$q(x,y|\theta) = \prod_{\ell=1}^{k} q(x,\theta_\ell)^{y(\ell)}, q(x,\theta_\ell) = \alpha_\ell G(x|\mu_\ell,\Sigma_\ell), \tag{28}$$

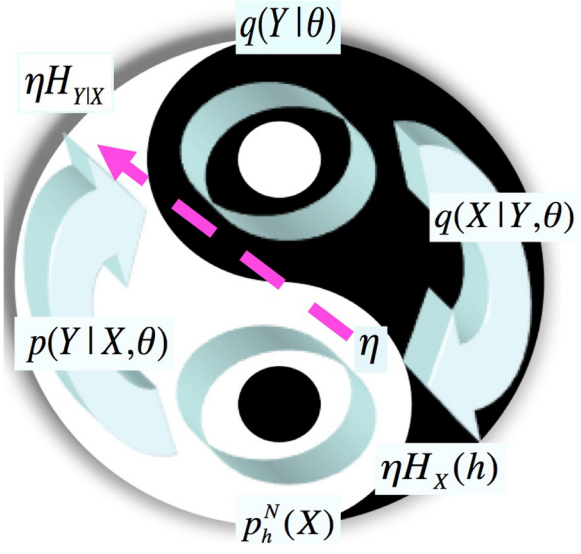

Figure 4 Negative feedback stabilises dynamics.

with $x \in R^d$ and $\theta_j = \{\alpha_j, \mu_j, \Sigma_j\}$, where $y = [y^{(1)}, \cdots, y^{(k)}]^T$ satisfies

$$\sum_{\ell=1}^{k} y^{(\ell)} = 1, y^{(\ell)} \text{ takes either 0 or 1.}$$

Given $X_N = \{x_t\}_{t=1}^N$ of i.i.d. samples, its corresponding samples $Y = \{y_t\}_{t=1}^N$ are also i.i.d. Accordingly, $p_{Y|X_N}^{new}$ in Equation 16 becomes simplified into

$$p_{Y|X_N}^{new} = \prod_{t=1}^{N} p(y_t|x_t, \theta^{new}, \eta^{new}),\ p(y|x,\theta,\eta) = \frac{q(x,y|\theta)^{\frac{1+\eta}{\eta}}}{\sum_y q(x,y|\theta)^{\frac{1+\eta}{\eta}}},$$

$$p(\ell|x,\theta) = p(y^{(\ell)} = 1, y^{(j)} = 0, \forall j \neq \ell|x,\theta,\eta) = \frac{[\alpha_\ell G(x|\mu_\ell,\Sigma_\ell)]^{\frac{1+\eta}{\eta}}}{\sum_{j=1}^{k}[\alpha_j G(x|\mu_j,\Sigma_j)]^{\frac{1+\eta}{\eta}}}, \tag{29}$$

from which the Yang step of Algorithm 1 is turned into the Yang step of a new Ying-Yang alternating algorithm for learning Gaussian mixture, summarised in Algorithm 3. Its Ying step is obtained by maximising $H_L(\theta)$ in Equation 17 with

$$H(\theta) = \sum_{t=1}^{N} \sum_{y_t} p(y_t|x_t,\theta,\eta) \ln q(x_t, y_t|\theta). \tag{30}$$

Algorithm 3 BYY learning for Gaussian mixture

Require: initialise θ^{old}, η^{old}, and let $p_{\ell,t} = 1/k, \eta$ varies as described by Remark (d) of Algorithm 1.

Repeat the following two steps **until** converged:

***Ying*-Step:** *get* $\alpha_\ell^{new}, \mu_\ell^{new}, \Sigma_\ell^{new}$ as follows:

$$n_\ell = \sum_{t=1}^{N} p_{\ell,t},\ \alpha_\ell^{new} = \frac{n_\ell}{\sum_{j=1}^{k} n_j},\ \mu_\ell^{new} = \frac{1}{n_\ell}\sum_{t=1}^{N} p_{\ell,t} x_t,$$

$$\Sigma_\ell^{new} = \frac{1}{n_\ell}\sum_{t=1}^{N} p_{\ell,t}(x_t - \mu_\ell^{new})(x_t - \mu_\ell^{new})^T,$$

trimming:

if $\alpha_i^{new} \to 0$ or $\alpha_i^{new} Tr[\Sigma_i^{new}] \to 0$, discard the ith Gaussian, let k=k-1.

***Yang*-Step**: for $t = 1, \cdots, N$ and $\ell = 1, \cdots, k$, get $p_{\ell,t} = p(\ell|x_t, \theta^{new})$ by Equation 29.

Remarks: When $\eta = \infty$, this algorithm degenerates to the EM algorithm (Redner and Walker 1984) for the maximum likelihood learning on Gaussian mixture. A finite value η makes it differ from the EM algorithm in that $p_{\ell,t}$ becomes more selective for automatic model selection.

Next, we consider one popular linear system as follows:

$$x = Ay + e,\ \ q(y|\phi) = G(y|\nu, \Lambda),\ \Lambda = diag[\lambda_1, \cdots, \lambda_k],$$
$$Eey^T = 0 \text{ or } q(e|y,\psi) = q(e) = G(e|0,\Sigma), \tag{31}$$

which leads to what is typically called factor analysis (FA), where Σ is a nonnegative diagonal matrix.

Classically, the name FA is used to refer the model Equation 31 with $\Lambda = I$. In this paper, we use FA-a to shortly denote this classical FA, and use FA-b to refer the one by Equation 31 with a diagonal matrix $\Lambda \neq I$ together with the following orthogonal constraint

$$A^T A = I. \tag{32}$$

For the maximum likelihood learning, FA-a and FA-b are equivalent. However, FA-b becomes much more favourable by using a learning algorithm with a nature of automatic model selection. Readers are referred to Sect.2.2 in Xu (2011 and Tu and Xu (2011a) for further studies on FA-b versus FA-a.

Given $X_N = \{x_t\}_{t=1}^N$ of i.i.d. samples and its corresponding $Y = \{y_t\}_{t=1}^N$, $H_L(\theta)$ in Equation 19 and $H(\theta)$ in Equation 17 become simplified into

$$\begin{gathered} H(\theta) \approx \sum_{t=1}^N H(\theta|x_t),\ H_L(\theta) \approx \sum_{t=1}^N H_L(\theta|x_t), \\ H(\theta|x_t) = \pi(x_t, y_t, \theta) - \tfrac{1}{2} Tr[\,\Gamma_{y|x}\Pi_{y|x}\,], \\ \pi(x, y, \theta) = \ln[\,G(x|Ay + \mu, \Sigma)G(y|\nu, \Lambda)],\ \Pi_{y|x} = A^{\mathrm{T}}\Sigma^{-1}A + \Lambda^{-1}. \\ H_L(\theta|x_t) = (1+\eta)H(\theta|x_t) + \eta\frac{\ln|\Gamma_{y|x}| + m\ln(2\pi e)}{2} \\ y_t = arg\max_y\ \pi(x_t, y, \theta) = Wx_t + w,\ W = \Gamma_{y|x}A^T\Sigma^{-1},\ w = \Lambda^{-1}\nu - W\mu, \end{gathered} \tag{33}$$

Usually, ν is set to be 0. Here we use ν to denote a constant vector for convenience of a further extension in Algorithm 14.

We update Σ^{new}, Λ^{new} by Equation 20 via solving them analytically as follows:

$$\begin{gathered} y_t = W^{old}x_t + w^{old},\ e_t = x_t - \mu - A^{\,old}(y_t - \nu), \\ \Sigma^{new} = A^{\,old}\Gamma_{y|x}^{old}A^{\,old\,\mathrm{T}} + \tfrac{1}{N}\sum_t e_t e_t^T,\ \Lambda^{new} = \Gamma_{y|x}^{old} + \tfrac{1}{N}\sum_t (y_t - \nu)(y_t - \nu)^T. \end{gathered} \tag{34}$$

Moreover, for updating A^{new} we can get

$$A = R^{xy}\Lambda^{new\,-1},\ R_{xy} = \tfrac{1}{N}\sum_t e_t(y_t - \nu)^T. \tag{35}$$

For updating FA-a, the above obtained A can be directly used as A^{new}. However, it can not be directly used as A^{new} for updating FA-b because there is also the orthogonal constraint by Equation 32 to be satisfied, for which we let

$$A^{new} = G_S[\,R_{xy}\Lambda^{new\,-1}\,], \tag{36}$$

where $G_S[A]$ denotes a Gram-Schmidt operator that orthogonalizes A. Even simply, we may make a gradient-based local search

$$A^{new} = A^{old} + \gamma_A \Delta A, \tag{37}$$

where $\gamma_A > 0$ is a small learning stepsize, and ΔA is a projection $\nabla_A H(\theta)$ onto Equation 32, e.g. for A given by Equation 35 we simply get

$$\Delta A = (I - AA^T)\nabla_A H(\theta^{old}),\ \nabla_A H(\theta) = \Sigma^{new\,-1}[\,R_{xy} - \Lambda^{new}A\,]. \tag{38}$$

The orthogonal constraint by Equation 32 also takes a role of removing a scale indeterminacy of the linear system by Equation 31, because an arbitrary diagonal matrix $D \neq I$ will make Equation 32 break though we may have $Ay = (AD)(D^{-1}y) = A^*y^*$ with y^* still from $G(y|\nu, \Lambda)$. Further details are referred to Sect.2.2 in Xu (2011).

Also, there are alternative constraints in place of Equation 32, e.g. see Eqs. (33) and (34) in Xu (2011).

One weak point by the above Equations 37 and 38 is that an appropriate γ_A is needed; otherwise, it may cause learning instability. Alternatively, we may replace Equation 32 by the following easy computing one:

$$Tr[A^TA] = 1. \tag{39}$$

Shortly, the notation **FA-c** is used to refer such a type of FA, namely, the one by Equation 31 not only with a diagonal $\Lambda \neq I$ but also with Equation 39. Then, we consider $\nabla_A H_\gamma(\theta)$ via Lagrange $H_\gamma(\theta) = H(\theta) - \gamma(Tr[A^T A] - 1)$, resulting in

$$\Sigma^{new} \nabla_A H_\gamma(\theta) = R_{xy} - \Lambda^{new} A - \gamma \Sigma^{new} A, \tag{40}$$

which is solved as follows

$$A^{new} = A_{\gamma^*},\ A_\gamma = R_{xy}(\Lambda^{new} + \gamma \Sigma^{new})^{-1},\ \gamma^* \text{ is the root of } Tr[A_\gamma], \tag{41}$$

where γ^* is obtainable by any one-variate iterative algorithm, e.g. Newton.

In summary, we can turn Algorithm 2 into Algorithm 4 for learning factor analyses, via modifying the Ying step, that is, we update $\Sigma^{new}, \Lambda^{new}$ based on Equation 34 and then update A^{new} according to a choice of possible constraints on A.

When $\Sigma = \sigma^2 I$, we also get an alternative algorithm for learning Principal Component Analysis (PCA) with automatic model selection on the number of principal components. Further details about PCA versus FA are referred to Sect.3.2 of (Xu 2010a).

Algorithm 4 BYY learning for FA

Require: Given $\{x_t\}$, get $\mu = \frac{\sum_t x_t}{N}$ and v.

initialise y_t randomly from $G(y|v, I)$, let $\theta = \{A, \Sigma, \Lambda, W, w\}, \Gamma_{y|x}$, and $\Lambda = diag[\lambda_1, \cdots, \lambda_m]$.

Repeat the following two steps **until** converged:

***Ying*-Step:** get Λ^{new}, Σ_e^{new} by Equation 34 and get

$$A^{new} \begin{cases} \text{by Equation 35,} & \text{for FA-a,} \\ \text{by Equation 37 together with Equation 38,} & \text{for FA-b,} \\ \text{by Equation 41,} & \text{for FA-c.} \\ \text{by Equation 57 plus Equations 59 and 60,} & \text{see Equation 61.} \end{cases}$$

trimming: for $i = 1, 2 \ldots, k$, discard the ith column of A and the ith element of y if $\lambda_i^{new} \to 0$, let $k = k - 1$.

***Yang*-Step**:

$$\Gamma_{y|x}^{new} = \frac{\eta}{1+\eta}(A^{newT} \Sigma^{new-1} A^{new} + \Lambda^{new-1})^{-1}.$$
$$W^{new} = \Gamma_{y|x}^{new} A^{new\,T} \Sigma^{new\,-1},\ w^{new} = \Lambda^{new} v - W^{new} \mu.$$

Remarks: It degenerates to the EM algorithm (Rubin and Thayer 1982; Tipping and Bishop 1999; Xu 1998c,d), by simply letting $\frac{\eta}{1+\eta}$ replaced by 1.

Learning local factor analysis

We can combine factor analysis by Equation 31 and Gaussian mixture by Equation 28 into the following general one:

$$q(y, \ell|\phi) = G(y|v_\ell, \Lambda_\ell) q(\ell|\alpha),\ q(\ell|\alpha) = \sum_{j=1}^k \alpha_\ell \delta_{\ell,j},\ \sum_{j=1}^k \alpha_j = 1,\ 1 \geq \alpha_j \geq 0,$$
$$\pi(x, y, \ell, \theta) = \ln[\, G(x|A_\ell y + \mu_\ell, \Sigma_\ell) q(y, \ell|\phi) q(A_\ell)], \tag{42}$$

where $\delta_{i,j}$ is the Kronecker delta with $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ otherwise, which actually describes i.i.d. samples $X_N = \{x_t\}_{t=1}^N$ by a mixture of *local factor analysis* or *local subspaces* at a special case $\Sigma_\ell = \sigma_\ell^2 I$.

Accordingly, $H_L(\theta)$ in Equation 23 is rewritten into

$$\begin{gathered} H_L(\theta) = \sum_{t=1}^{N} \sum_{\ell=1}^{k} p(\ell|x_t,\theta)[\,(1+\eta)H_L(\theta|\ell,x_t) - \eta \ln p(\ell|x_t,\theta)\,], \\ H_L(\theta|\ell,x_t) = H(\theta|\ell,x_t) + \tfrac{\eta}{1+\eta} E_{y|\ell,x_t}, \\ E_{y|\ell,x_t} = -\int p(y|\ell,x_t,\theta) \ln p(y|\ell,x_t,\theta) dy, \\ H(\theta|\ell,x_t) = \int p(y|\ell,x_t,\theta)\pi(x_t,y,\ell,\theta) dy. \end{gathered} \tag{43}$$

Similar to $H_L(\theta|x_t)$ in Equation 33, we further get

$$\begin{gathered} H_L(\theta|\ell,x_t) = \pi(x_t,y_{t,\ell},\ell,\theta) + \tfrac{0.5\eta}{1+\eta}[\,\ln|\Gamma_{\ell,y|x}| + \ln(2\pi)^{m_\ell}\,], \\ E_{y|x_t} = 0.5[\,\ln|\Gamma_{\ell,y|x}| + m_\ell \ln(2\pi e)\,], \\ H(\theta|\ell,x_t) = \pi(x_t,y_{t,\ell},\ell,\theta) - \tfrac{1}{2} Tr[\,\Gamma_{\ell,y|x}\Pi_{\ell,y|x}\,], \\ \Pi_{\ell,y|x} = A_\ell^T \Sigma_\ell^{-1} A_\ell + \Lambda_\ell^{-1},\ \Gamma_{\ell,y|x}^{new} = \tfrac{\eta}{\eta+1} \Pi_{\ell,y|x}^{old\,-1}, \\ y_{t,\ell} = arg\max_y\ \pi(x_t,y,\ell,\theta) = W_\ell x_t + w_\ell, \\ W_\ell = \Gamma_{\ell,y|x} A_\ell^T {\Sigma_\ell}^{-1},\ w_\ell = \Lambda_\ell^{-1} \nu_\ell - W_\ell \mu_\ell, \end{gathered} \tag{44}$$

from which we further get θ^{new} via maximising $\sum_{t=1}^{N} \sum_{\ell=1}^{k} p(\ell|x_t,\theta) H(\theta|\ell,x_t)$, resulting in the Ying step of a new Ying-Yang alternating algorithm for learning a mixture of *local factor analysis*, as in Algorithm 5. Actually, this Ying step combines the Ying of Algorithm 3 and the Ying of Algorithm 4.

Algorithm 5 BYY learning for local factor analysis

Require: initialise $\theta = \{A_\ell, \Sigma_\ell, W_\ell, w_\ell, \Lambda_\ell = diag[\,\lambda_{\ell,1}, \cdots, \lambda_{\ell,m_\ell}\,]\}$, $\Gamma_{y|x}$, and η, initialise $p_{\ell,t} = 1/k$, and get y_t randomly from $G(y|0,I)$.

Repeat the following two steps **until** converged:

***Ying*-Step:** *get* $\alpha_\ell^{new}, \mu_\ell^{new}, A_\ell^{new}, \Sigma_\ell^{new}, \nu_\ell^{new}, \Lambda_\ell^{new}$ by

$\alpha_\ell^{new} = \frac{1}{N} \sum_{t=1}^{N} p_{\ell,t}$, $\mu_\ell^{new} = \frac{1}{N\alpha_\ell^{new}} \sum_{t=1}^{N} p_{\ell,t} x_t$,

$y_{t,\ell} = W_\ell^{old} x_t + w_\ell^{old}$, $e_{t,\ell} = x_t - \mu_\ell^{new} - A_\ell^{old} y_{t,\ell}$, $\nu_\ell^{new} = \frac{1}{N\alpha_\ell^{new}} \sum_{t=1}^{N} p_{\ell,t} y_{t,\ell}$,

$\Sigma_\ell^{new} = A_\ell^{old} \Gamma_{\ell,y|x}^{old} A_\ell^{old\,\mathrm{T}} + \frac{1}{N\alpha_\ell^{new}} \sum_{t=1}^{N} p_{\ell,t} e_{t,\ell} e_{t,\ell}^T$,

$\Lambda_\ell^{new} = \Gamma_{\ell,y|x}^{old} + \frac{1}{N\alpha_\ell^{new}} \sum_{t=1}^{N} p_{\ell,t} y_{t,\ell} y_{t,\ell}^T$, $R_\ell^{xy} = \frac{1}{N\alpha_\ell^{new}} \sum_{t=1}^{N} p_{\ell,t} e_{t,\ell} y_{t,\ell}^T$.

For FA-b, we get A_ℓ^{new} by one of the four choices in the Ying step of Algorithm 4 with all the involved symbols getting the corresponding subscript $\ell = 1, \ldots, k$ attached.

TRIMMING:

if one $\lambda_{\ell,i}^{new}$ of tends to 0, discard the ith column of A_ℓ, let $m_\ell = m_\ell - 1$.

if $\alpha_i^{new} \to 0$ or $\alpha_i^{new} Tr[\,\Sigma_i^{new}] \to 0$, discard $G(x|A_i y + \mu_i, \Sigma_i)$ and $G(y|\nu_i, \Lambda_i)$, let $k = k - 1$.

***Yang*-Step**: We get $p_{\ell,t} = p(\ell|x_t,\theta^{new})$ by Equation 45. It follows from Equation 44 that we also get

$\Gamma_{\ell,y|x}^{new} = \frac{\eta}{1+\eta} (A_\ell^{newT} {\Sigma_\ell}^{new-1} A_\ell^{new} + {\Lambda_\ell}^{new-1})^{-1}$.

$W_\ell^{new} = \Gamma_{\ell,y|x}^{new} A_\ell^{new\,T} {\Sigma_\ell}^{new\,-1}$, $w_\ell^{new} = {\Lambda_\ell}^{new\,-1} \nu_\ell^{new} - W_\ell^{new} \mu_\ell^{new}$.

Maximising $H_L(\theta)$ with respect to $p(\ell|x_t,\theta)$ yields

$$p(\ell|x_t,\theta) = \frac{e^{\frac{\eta+1}{\eta}\pi(x_t,y,\ell,\theta)+\frac{1}{2}\ln[|\Gamma_{\ell,y|x}|(2\pi)^{m_\ell}]}}{\sum_{\ell=1}^{k} e^{\frac{\eta+1}{\eta}\pi(x_t,y,\ell,\theta)+\frac{1}{2}\ln[|\Gamma_{\ell,y|x}|(2\pi)^{m_\ell}]}} = \frac{[\,\alpha_\ell G(x_t|\mu_\ell, A_\ell\Lambda_\ell A_\ell^T+\Sigma_\ell)]^{\frac{\eta+1}{\eta}}}{\sum_{\ell=1}^{k}[\,\alpha_\ell G(x_t|\mu_\ell, A_\ell\Lambda_\ell A_\ell^T+\Sigma_\ell)]^{\frac{\eta+1}{\eta}}}, \tag{45}$$

from which and together with Equation 44, we see that the Yang step of Algorithm 5 actually combines the Yang of Algorithm 3 and the Yang of Algorithm 4.

This algorithm degenerates back to not only Algorithm 4 with $k = 1$ but also Algorithm 3 with $y = 0$ and $A_\ell = 0$ for each ℓ.

Learning binary factor analysis

We consider another setting of the linear system, with each $y^{(\ell)}$ taking either 0 or 1 and $q(y|\phi)$ in Equation 31 being a multivariate Bernoulli distribution as follows:

$$q(y|\phi) = \textstyle\prod_i \alpha_i^{y^{(i)}}(1-\alpha_i)^{1-y^{(i)}},\ q(x|y,\psi) = G(x|Ay+\mu,\Sigma), \tag{46}$$

which is called binary factor analysis (BFA).

Together with adding the constraint on y in Equation 28, we are lead to an equivalent form of Equation 28. In other words, learning BFA may be regarded as a relaxation or extension of learning Gaussian mixture.

Putting this setting into Equation 17, we get its simplified version as follows:

$$\begin{aligned} H_L(\theta) &= (1+\eta)H(\theta) + \eta E_{Y|X},\ H(\theta) = \textstyle\sum_{t=1}^{N}\sum_{y\in\mathcal{C}_{tf}} p(y|x_t,\theta)\pi(x_t,y,\theta), \\ \pi(x,y,\theta) &= \ln[\,G(x|Ay+\mu,\Sigma)\textstyle\prod_i \frac{\alpha_i^{y^{(i)}}}{(1-\alpha_i)^{y^{(i)}-1}}], \\ E_{Y|X} &= -\textstyle\sum_{t=1}^{N}\sum_{y\in\mathcal{C}_{tf}} p(y|x_t,\theta)\ln p(y|x_t,\theta). \end{aligned} \tag{47}$$

For a small k, $\mathcal{C}_{tf}$ can be the entire set that consists of all the possible values of y. For a large k, such an entire set could be huge, instead we consider one $\mathcal{C}_{tf}$ that merely consists of one subset of values that we focus on. One choice is given by

$$\mathcal{C}_{tf} = \{y:\ \text{differing from } y_t^* \text{ by less than } \kappa \text{ bits}\}, \tag{48}$$
$$\text{where } y_t^* = arg\max_y \pi(x_t,y,\theta^{old}) \text{ and } \kappa \text{ is a small number, e.g. } \kappa = 1 \text{ or } 2.$$

One example was given by Eq. (20) in Xu (2010a) for binary FA, and the other example may also be found in Sect. 2.1.5 of Xu (2012a) on learning Gaussian mixture.

Given y_t and $\mathcal{C}_{tf}, t = 1,\ldots,N$ by Equation 48, we maximise $H_L(\theta)$ with respect to $p(y|x,\theta)$, resulting in

$$p(y|x,\theta) = \frac{exp[\frac{1+\eta}{\eta}\pi(x,y,\theta)]}{\sum_{y\in\mathcal{C}_{tf}} exp[\frac{1+\eta}{\eta}\pi(x,y,\theta)]}, \tag{49}$$

from which we get the Yang step of Algorithm 6 for binary factor analyses, similar to getting the Yang step of Algorithm 3 from the Yang step of Algorithm 1.

Algorithm 6 BYY learning for binary FA

Require: Given $\{x_t\}$, get $\mu = \frac{1}{N}\sum_t x_t$ and get each $\mathcal{C}_{tf}$ by randomly picking n_κ values of y.

initialise $p_{y|x_t} = 1/n_\kappa,\ y \in \mathcal{C}_{tf}$ and $\theta^{old} = \{A^{old}, \alpha^{old}, \Sigma^{old}\}$.

Repeat the following two steps **until** converged:

***Ying*-Step:** get $N_p = \sum_{t=1}^N \sum_{y\in\mathcal{C}_{tf}} p_{y|x_t}$, $\alpha^{new} = \frac{1}{N_p}\sum_{t=1}^N \sum_{y\in\mathcal{C}_{tf}} p_{y|x_t} y_t$,

$e_t = x_t - \mu - A^{old} y_t$, $\Sigma^{new} = \frac{1}{N_p}\sum_{t=1}^N \sum_{y\in\mathcal{C}_{tf}} p_{y|x_t} e_t e_t^T$,

$\Lambda^{new} = \frac{1}{N_p}\sum_{t=1}^N \sum_{y\in\mathcal{C}_{tf}} p_{y|x_t} yy^T$, $R^{xy} = \frac{1}{N_p}\sum_{t=1}^N \sum_{y\in\mathcal{C}_{tf}} p_{y|x_t} e_t y^T$,

$$A^{new} = \begin{cases} R^{xy}\Lambda^{new\,-1}, & \text{with no priori,} \\ \text{by Equation 57 plus Equations 59 and 60,} & \text{see Equation 61.} \end{cases}$$

trimming: for $i = 1, 2 \ldots, k$, discard the ith column of A and the ith element of y if $\alpha_i^{new}(1-\alpha_i^{new}) \to 0$, let $k = k-1$.

***Yang*-Step**: for $t = 1, \cdots, N$, get $\mathcal{C}_{tf}$ by Equation 48 and then get $p_{y|x_t} = p(y|x_t, \theta^{new})$, *for* $y \in \mathcal{C}_{tf}$, by Equation 49.

With $p(y|x,\theta)$ fixed, we get θ^{new} by maximising $H(\theta)$, resulting in the Ying step of Algorithm 6.

Imposing the constraint on y in Equation 29 and letting $\mathcal{C}_{tf}$ to cover the entire domain y, this algorithm degenerates to Algorithm 3 for Gaussian mixture when $\Sigma_\ell = \Sigma$.

The summation over $\mathcal{C}_{tf}$ will incur for a high computing cost when $\mathcal{C}_{tf}$ consists of many elements. Alternatively, we may assume that $p(y|x_t,\theta) = \prod_i p(y^{(i)}|x_t,\theta)$ with $0 \le \xi_{y|x_t}^{(i)} = \int y^{(i)} p(y^{(i)}|x_t,\theta)dy^{(i)} \le 1$, and we simplify $H(\theta)$ and $E_{Y|X}$ into

$$\begin{aligned} H(\theta) &= \sum_{t=1}^N \pi(x_t, y, \theta)_{y=[\xi_{y|x_t}^{(1)},\ldots,\xi_{y|x_t}^{(k)}]^T}, \\ E_{Y|X} &= -\sum_{t=1}^N \sum_{i=1}^k \xi_{y|x_t}^{(i)} \ln \xi_{y|x_t}^{(i)}, \end{aligned} \tag{50}$$

from which we get Algorithm 7 with a simplified Ying step, but its Yang step needs to get $\xi_{y|x_t}$ by solving a constrained quadratic optimisation via one of typical existing techniques (Fang et al. 1997; Floudas and Visweswaran 1995).

Algorithm 7 Another algorithm for binary FA

Require: Given $\{x_t\}$, get $\mu = \frac{1}{N}\sum_t x_t$, initialise $\theta^{old} = \{A^{old}, \alpha^{old}, \Sigma^{old}\}$ and $\xi_{y|x_t} = [\frac{1}{k}, \ldots, \frac{1}{k}]^T$.

Repeat the following two steps **until** converged.

***Ying*-Step:** $\alpha^{new} = \frac{\sum_{t=1}^N \xi_{y|x_t}}{N}$, $\xi_{y|x_t} = [\xi_{y|x_t}^{(1)}, \ldots, \xi_{y|x_t}^{(k)}]^T$, $e_t = x_t - \mu - A^{old}\xi_{y|x_t}$,

$\Sigma^{new} = \frac{1}{N}\sum_{t=1}^N e_t e_t^T$, $\Lambda^{new} = \frac{1}{N}\sum_{t=1}^N \xi_{y|x_t}\xi_{y|x_t}^T$, $R^{xy} = \frac{1}{N}\sum_{t=1}^N e_t \xi_{y|x_t}^T$,

$$A^{new} = \begin{cases} R^{xy}\Lambda^{new\,-1}, & \text{with no priori,} \\ \text{by Equation 57 plus Equations 59 and 60,} & \text{see Equation 61.} \end{cases}$$

trimming: for $i = 1, 2 \ldots, k$, discard the ith column of A and the ith element of y if $\alpha_i^{new}(1-\alpha_i^{new}) \to 0$, let $k = k-1$.

***Yang*-Step**: for $t = 1, \cdots, N$, get $\xi_{y|x_t}$ to maximise $H_L(\theta)$ by a constrained quadratic optimisation (Fang et al. 1997; Floudas and Visweswaran 1995).

Learning nonGaussian factor analysis

We progress to consider an even general case, called nonGaussian factor analysis (NFA), with each independent component of y being nonGaussian, e.g. from a mixture of univariate Gaussians. Here, we consider the following setting:

$$\begin{aligned} q(y,z|\phi) &= \textstyle\prod_i \{G(y^{(i)}|\nu^{(i)}_{z^{(i)}}, \lambda^{(i)}_{z^{(i)}}) q(z^{(i)}|\alpha)\}, \\ z &= [\, z^{(1)}, \ldots, z^{(k)}]^T,\ z^{(i)} = 1, \ldots, m_i \ with\ m_i \geq 1, \\ q(z^{(i)}|\alpha) &= \textstyle\sum_{j=1}^{m_i} \alpha_j^{(i)} \delta_{j,z^{(i)}},\ \sum_{j=1}^{m_i} \alpha_j^{(i)} = 1, \\ \pi(x,y,z,\theta) &= \ln [\, G(x|Ay+\mu, \Sigma) q(y,z|\phi) q(A)]. \end{aligned} \tag{51}$$

where $1 \geq \alpha_j^{(i)} \geq 0$. Putting it into Equation 23, similar to Equation 43 we get

$$\begin{aligned} H_L(\theta) &= \textstyle\sum_{t=1}^N \sum_{z \in \mathcal{C}_{tf}} p(z|x_t,\theta)[\,(1+\eta) H_L(\theta|z,x_t) + \ln p(z|x_t,\theta)]\,, \\ H_L(\theta|z,x_t) &= H(\theta|z,x_t,) + \tfrac{\eta}{1+\eta} E_{y|z,x_t}(\theta), \\ E_{y|z,x_t}(\theta) &= -\textstyle\int p(y|z,x_t,\theta) \ln p(y|z,x_t,\theta) dy, \\ H(\theta|z,x_t) &= \textstyle\int p(y|z,x_t,\theta) \pi(x_t,y,z,\theta) dy. \end{aligned}$$

Similar to Equation 49, maximising $H_L(\theta)$ gets $p(z|x_t,\theta)$ in the Yang step. Similar to Equation 44, we also get $y_{z,t} = [\, y^{(1)}_{z,t}, \ldots, y^{(k)}_{z,t}]^T = arg\max_y \pi(x_t,y,z,\theta)$ and $\Gamma_{z,y|x} = arg\max_{\Gamma_{z,y|x}} H_L(\theta|z,x_t)$.

We maximise $H_L(\theta)$ to update θ, resulting in the Ying step of Algorithm 8 for learning NFA. The Ying step consists of the first part for updating each component $\alpha_j^{(i)} G(y^{(i)}|\nu^{(i)}_{z^{(i)}}, \lambda^{(i)}_{z^{(i)}})$ and the second part for updating $G(x|Ay+\mu, \Sigma)$. Also, the role of $\delta_{j,z^{(i)}}$ is picking those components that have contributions to the corresponding $\alpha_j^{(i)}, \nu_j^{(i)}, \lambda_j^{(i)}$ according to whether $z = j$. The number m_i of the components is determined via trimming off $G(y^{(i)}|\nu^{(i)}_{z^{(i)}}, \lambda^{(i)}_{z^{(i)}})$ if $(\alpha_j^{(i)} \lambda_j^{(i)})^{new} \to 0$.

Unsupervised vs semi-supervised

Instead of knowing i.i.d. samples $X_N = \{x_t\}_{t=1}^N$, there maybe a subset $X_s \subset X_N$ in which each $x_t \in X_s$ is associated with a supervision sample y_t^*. The problem is called unsupervised learning when X_s is an empty set, and called supervised learning when $X_s = X_N$. Generally, the problem is called semi-supervised learning as X_s is between the two extreme cases.

For the BYY harmony learning, unsupervised, semi-supervised and supervised learning are all expressed in a same formulation. There are two types of implementation according to whether y is discrete or real.

When y is discrete, we modify $H_L(\theta)$ by Equation 17 into

$$\begin{aligned} \max_\theta H_{L,S}(\theta) &= H_L(\theta) + \gamma H_S(\theta), \\ H_S(\theta) &= \textstyle\sum_{x_t \in X_s} \sum_{y_t} \delta_{y_t, y_t^*} p(y_t|x_t,\theta,\eta) \ln q(x_t,y_t|\theta), \end{aligned}$$

where y_t^* is the teaching label associated with X_s, and $\gamma > 0$ is a confidence factor. The bigger the $\gamma > 0$ is, the higher our confidence is on the supervision sample.

Accordingly, maximising $H_{L,S}(\theta)$ results in

$$p(y|x,\theta) = \frac{q(x,y|\theta)^{[\gamma \delta_{y,y_t^*} + 1 + \eta]/\eta}}{\sum_y q(x,y|\theta)^{[\gamma \delta_{y,y_t^*} + 1 + \eta]/\eta}}. \tag{52}$$

Algorithm 8 BYY learning for nonGaussian FA

Require: Initialise $\theta = \{A, \Sigma, \{\alpha_j^{(i)}\}, \{\nu_j^{(i)}\}, \{\lambda_j^{(i)}\}\}$. Get $\mathcal{C}_{tf}$ by randomly picking n_κ values of z, let $\mu = \frac{1}{N}\sum_t x_t$. For $z \in \mathcal{C}_{tf}$, get $p_{z|x_t} = \frac{1}{n_\kappa}$ and get $y_{z,t}$ randomly from $G(y|0, I)$.

Repeat the following two steps **until** converged:

***Ying*-Step:** For $j = 1, \ldots, k_i; i = 1, 2 \ldots, m$, we get

$$n^{(i)} = \sum_{j=1}^{m_i}\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}\delta_{j,z^{(i)}},\ \alpha_j^{(i)new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}\delta_{j,z^{(i)}}}{n^{(i)}},$$

$$\delta_{x,y} = \begin{cases} 1, & x = y, \\ 0, & x \neq y; \end{cases} \quad \nu_j^{(i)new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}\delta_{j,z^{(i)}} y_{z,t}^{(i)}}{n^{(i)}},$$

where $y_{z,t} = [\, y_{z,t}^{(1)}, \cdots, y_{z,t}^{(m)}]^T$ is computed in the Yang step.

$$\lambda_j^{(i)new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}\delta_{j,z^{(i)}}[(y_{z,t}^{(i)} - \nu_j^{(i)new})^2 + \rho_{z,x_t}^{(i)}]}{n^{(i)}}.$$

where $\rho_{z,x_t}^{(i)}$ is the ith diagonal element of $\Gamma_{z,y|x}^{old}$.

$$N_p = \sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t},\ \Sigma^{new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}[e_t e_t^T + A^{old}\Gamma_{z,y|x}^{old} A^{old\ T}]}{N_p},$$

$$e_t = x_t - \mu - A^{old} y_{z,t},\ \Lambda^{new} = \frac{1}{N_p}\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t} diag[\, \lambda_{z^{(1)}}^{(1)new}, \ldots, \lambda_{z^{(k)}}^{(k)new}],$$

$$R^{xy} = \frac{1}{N_p}\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t} e_t y_{z,t}^T,$$

$$A^{new} = \begin{cases} R^{xy}\Lambda^{new\,-1}, & \text{with no priori,} \\ \text{by Equation 57 plus Equations 59 and 60,} & \text{see Equation 61.} \end{cases}$$

Trimming: if $(\alpha_j^{(i)}\lambda_j^{(i)})^{new} \to 0$, discard $\nu_j^{(i)}, \lambda_j^{(i)}$, let $m \to m-1$; discard the ith column of A and the ith element of y if $m_i = 1$ and $(\lambda^{(i)})^{new} \to 0$, let $k = k-1$. where $\lambda^{(i)}$ is the sum of $\lambda_j^{(i)}$'s that are not discarded yet.

***Yang*-Step**: for $t = 1, \cdots, N$, get $\mathcal{C}_{tf}$ by Equation 48 and get

$$p_{z|x_t} = p(z|x_t, \theta^{new}),\ for\ z \in \mathcal{C}_{tf},$$

$$\text{where } p(z|x_t, \theta) = \frac{e^{\frac{\eta+1}{\eta}\pi(x_t,y,z,\theta) + \frac{1}{2}\ln|\Gamma_{z,y|x}|}}{\sum_{y\in\mathcal{C}_{tf}} e^{\frac{\eta+1}{\eta}\pi(x_t,y,z,\theta) + \frac{1}{2}\ln|\Gamma_{z,y|x}|}},$$

$$\Gamma_{z,y|x}^{new} = \Gamma_{z,y|x}(\theta^{new}),\ y_{z,t} = y(z, x_t, \theta^{new}),\ z \in \mathcal{C}_{tf},$$

$$\Gamma_{z,y|x}(\theta) = \frac{\eta}{1+\eta}(A^T\Sigma^{-1}A^+\Lambda_z^{-1})^{-1},$$

$$y(z, x_t, \theta) = \Gamma_{z,y|x}A^T\Sigma^{-1}(x_t - \mu + \Lambda_z^{-1}\nu_z),$$

$$\nu_z = [\, \nu_{z^{(1)}}^{(1)}, \ldots, \nu_{z^{(k)}}^{(k)}]^T,\ \Lambda_z = [\, \lambda_{z^{(1)}}^{(1)}, \ldots, \lambda_{z^{(k)}}^{(k)}].$$

Remarks: (a) It returns to Algorithm 4 when $m_i = 1, \nu_1^{(i)} = 0, \forall i$.

(b) If $m_i = 2, \nu_1^{(i)} = 0, \nu_2^{(i)} = 1$ for all i, it learns a noisy binary FA as $\lambda_j^{(i)}, j = 1, 2$ are fixed at a small constant. It further degenerates to learning binary FA by letting $\lambda_j^{(i)} = 0$.

(c) Generally, we may set $\nu_1^{(i)} = 0$ to simplify computation.

(d) To save storage, $\Gamma_{z,y|x}^{new}$ and $y_{z,t}$ may be computed during Ying step.

Fixing $p(y|x, \theta)$, we further update θ via maximising $H_{L,S}(\theta)$. From Equations 30 and 52, we have

$$H_{L,S}(\theta) = \sum_{t=1}^{N}\sum_{y_t} p(y_t|x_t)\ln q(x_t, y_t|\theta),$$
$$p(y_t|x_t) = p(y_t|x_t, \theta)(\eta + 1 + \gamma\delta_{y_t, y_t^*}), \tag{53}$$

from which we modify Algorithm 3 into Algorithm 9, with the Ying step kept unchanged while the Yang step modified into Algorithm 9.

Algorithm 9 Semi-supervised BYY learning for Gaussian mixture

Require: & **Repeat** : same as in Algorithm 3.

***Ying*-Step:** same as in Algorithm 3.

***Yang*-Step**: for $t = 1, \cdots, N$ and $\ell = 1, \cdots, k$, get $p_{\ell,t} = (\eta + 1 + \gamma\delta_{\ell,\ell_t^*})p_{\ell|x_t}(\theta^{new})$,

$$p_{\ell|x_t}(\theta) = \frac{[\alpha_\ell G(x_t|\mu_\ell,\Sigma_\ell)]^{[\gamma\delta_{\ell,\ell_t^*}+1+\eta]/\eta}}{\sum_{j=1}^k [\alpha_j G(x_t|\mu_j,\Sigma_j)]^{[\gamma\delta_{j,\ell_t^*}+1+\eta]/\eta}}.$$

Remarks:

(a) For each sample x_t, $\delta_{\ell,\ell_t^*} = 0$ if it has no teaching label while $\delta_{\ell,\ell_t^*} = 1$ when ℓ is equal to a given teaching label ℓ_t^*.

(b) The bigger the $\gamma > 0$ is, the higher the supervision strength is. We may let $\gamma > 0$ to start at a high value and gradually decrease towards a pre-specified value.

We can always assign a teaching label ℓ_t^* to each sample x_t. If there is no teaching label, we assign ℓ_t^* to be a number larger than k and thus always have $\delta_{\ell,\ell_t^*} = 0$. Otherwise, we let ℓ_t^* to be its teaching label and have $\delta_{\ell,\ell_t^*} = 1$ when $\ell = \ell_t^*$.

Similarly, we modify Algorithm 6 for learning binary FA into a semi-supervised version, i.e. Algorithm 10. Whether or not there is a teaching sample y_t^* for x_t, we may always assign one y_t^* to each sample x_t. If there is no teaching sample, we assign y_t^* to be out of $\mathcal{C}_{tf}$ and thus have $\delta_{y,y_t^*} = 0$. Otherwise, we let y_t^* to be its teaching sample and have $\delta_{y,y_t^*} = 1$ when $y = y_t^*$.

Algorithm 10 Semi-supervised BYY for binary FA

Require: & **Repeat** : same as in Algorithm 6.

***Ying*-Step:** same as Algorithm 6.

***Yang*-Step:** for $t = 1, \cdots, N$, get $\mathcal{C}_{tf}$ by Equation 48 and then

$$p_{y|x_t}^{new} = (\eta + 1 + \gamma\delta_{y,y_t^*})p_{y|x_t}(\theta^{new}), \forall y \in \mathcal{C}_{tf},\ p_{y|x_t}(\theta) = \frac{exp[\frac{\gamma\delta_{y,y_t^*}+1+\eta}{\eta}\pi(x_t,y,\theta)]}{\sum_{y\in\mathcal{C}_{tf}} exp[\frac{\gamma\delta_{y,y_t^*}+1+\eta}{\eta}\pi(x_t,y,\theta)]}.$$

Remarks: For a sample x_t, $\delta_{y,y_t^*} = 0$ if it has no teaching label and $\delta_{y,y_t^*} = 1$ when y is equal to the teaching label y_t^*.

When y is real valued, teaching samples will not affect the Yang step, while updating θ by the Ying step becomes maximising

$$H_S(\theta) = \sum_{t=1}^N [\,(1+\eta)\pi(x_t,y_t,\theta) + \gamma I_t \pi(x_t,y_t^*,\theta)\,],\ y_t = arg\max_y \pi(x_t,y,\theta), \qquad (54)$$

where I_t is an indicator explained by the remark given in Algorithm 11. It follows from Equation 54 that Algorithm 4 for learning FA can be modified into Algorithm 11 with some changes in the Ying step.

Moreover, we may combine Equations 53 and 54 to modify Algorithm 8 for learning NFA into Algorithm 12. Similar to Algorithm 10, we may always assign one discrete vector $z_t^* = [\,z_t^{(1)*}, \cdots, z_t^{(m)*}]$ to each sample x_t. If there is no teaching information, we assign z_t^* to take a value that is out of our consideration, e.g. letting every $z_t^{(i)*}$ to be a big number,

Algorithm 11 Semi-supervised BYY learning for FA

Require: & **Repeat** : same as in Algorithm 4.

***Ying*-Step:** get $y_t = W^{old}x_t + w^{old}$, then get $e_t = x_t - \mu - A^{old}y_t$, $e_t^* = x_t - \mu - A^{old}y_t^*$,
$e_te_t^T$ is replaced by $\frac{(\eta+1)e_te_t^T+\gamma I_te_t^*e_t^{*T}}{\eta+1+\gamma I_t}$ in updating Σ^{new}.
$y_ty_t^T$ is replaced by $\frac{(\eta+1)y_ty_t^T+\gamma I_ty_t^*y_t^{*T}}{\eta+1+\gamma I_t}$ in updating Λ^{new}.
$e_ty_t^T$ is replaced by $\frac{(\eta+1)e_ty_t^T+\gamma I_te_t^*y_t^{*T}}{\eta+1+\gamma I_t}$ in updating R_{xy}.
the other parts of Yang step are same as in Algorithm 4.

***Yang*-Step**: same as in Algorithm 4.

Remarks: I_t is an indicator with $I_t = 1$ when x_t is associated with a teaching sample y_t^* and with $I_t = 0$ when x_t is not associated with a teaching sample.

we always have $\delta_{z,z_t^*} = 0$ for $z \in \mathcal{C}_{tf}$. Otherwise, we let z_t^* to be its teaching label about z_t, and use $\delta_{z,z_t^*} = 1$ to indicate $z = z_t^*$.

Similar to the Yang step of Algorithm 9 and of Algorithm 10, we get $p_{z|x_t}(\theta)$ with a difference that $p_{z|x_t} = p_{z|x_t}(\theta^{new})$ is not globally rescaled by a factor. Instead, a rescaling is distributed among each updating in the Ying step. Another difference from Algorithm 10 lies in that each z_t is also associated with another real valued vector $y_t = [y_t^{(1)}, \cdots, y_t^{(m)}]$.

Algorithm 12 Semi-supervised BYY for NFA

Require: & **Repeat** : same as in Algorithm 8.

***Ying*-Step:** For $j = 1, \ldots, k_i$ and $i = 1, 2 \ldots, m$, we get

$$n^{(i)} = \sum_{j=1}^{m_i}\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}[\,(1+\eta)\delta_{j,z^{(i)}} + \gamma\delta_{j,z_t^{(i)*}}],$$

$$\alpha_j^{(i)new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}[(1+\eta)\delta_{j,z^{(i)}}+\gamma\delta_{j,z_t^{(i)*}}]}{n^{(i)}},$$

$$\nu_j^{(i)new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}[(1+\eta)\delta_{j,z^{(i)}}y_{z,t}^{(i)}+\gamma\delta_{j,z_t^{(i)*}}y_{z,t}^{(i)*}]}{n^{(i)}},$$

$$\Delta\lambda_j^{(i)} = \gamma\delta_{j,z_t^{(i)*}}(y_{z,t}^{(i)*} - \nu_j^{(i)new})^2 + (1+\eta)\delta_{j,z^{(i)}}[\,\rho_{z,x_t}^{(i)} + (y_{z,t}^{(i)} - \nu_j^{(i)new})^2].$$

$$\lambda_j^{(i)new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}\delta_{j,z^{(i)}}\Delta\lambda_j^{(i)}}{n^{(i)}}.$$

where $\rho_{z,x_t}^{(i)}$ is the ith diagonal element of $\Gamma_{z,y|x}^{old}$.

$$N_p = \sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}(\eta + 1 + \gamma\delta_{z,z_t^*}),$$

$$e_t = x_t - \mu - A^{old}y_{z,t},\ e_t^* = x_t - \mu - A^{old}y_{z,t}^*,\ \Delta\Sigma_z = e_te_t^T + A^{old}\Gamma_{z,y|x}^{old}A^{old\,T},$$

$$\Sigma^{new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}[(\eta+1)\Delta\Sigma_z+\gamma\delta_{z,z_t^*}e_t^*e_t^{*T}]}{N_p},$$

$$\Lambda^{new} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}diag[\lambda_{z^{(1)}}^{(1)new},\ldots,\lambda_{z^{(k)}}^{(k)new}]}{N_p},$$

$$R^{xy} = \frac{\sum_{t=1}^{N}\sum_{z\in\mathcal{C}_{tf}} p_{z|x_t}[(\eta+1)e_ty_{z,t}^T+\gamma\delta_{z,z_t^*}e_t^*y_{z,t}^{*T}]}{N_p},$$

$$A^{new} = \begin{cases} R^{xy}\Lambda^{new\,-1}, & \text{without a priori,} \\ \text{by Equation 57 plus Equations 59 and 60,} & \text{see Equation 61.} \end{cases}$$

Trimming: same as in Algorithm 8.

***Yang*-Step:** for $t = 1, \cdots, N$, get $\mathcal{C}_{tf}$ by Equation 48 and get

$$p_{z|x_t} = p_{z|x_t}(\theta^{new}), \forall z \in \mathcal{C}_{tf},\ p(z|x_t,\theta) = \frac{e^{\frac{\gamma\delta_{z,z_t^*}+\eta+1}{\eta}\pi(x_t,y,z,\theta)+\frac{1}{2}\ln|\Gamma_{z,y|x}|}}{\sum_{y\in\mathcal{C}_{tf}} e^{\frac{\gamma\delta_{z,z_t^*}+\eta+1}{\eta}\pi(x_t,y,z,\theta)+\frac{1}{2}\ln|\Gamma_{z,y|x}|}},$$

$$y_{z,t} = y(z,x_t,\theta^{new}),\ y_{z,t}^* = \begin{cases} [y_{z,t}^{(1)*},\cdots,y_{z,t}^{(m)*}]^T, & \text{given,} \\ y(z_t^*,x_t,\theta^{new}), & \text{unknown and estimated.} \end{cases}$$

the other parts of Yang step are same as in Algorithm 8.

For each teaching label z_t^*, we may have two situations. One is that the corresponding teaching vector $y_{z,t}^* = [y_{z,t}^{(1)*}, \cdots, y_{z,t}^{(m)*}]^T$ is given together with z_t^*. The other is that we have z_t^* only and need to estimate $y_{z,t}^*$.

Also, the situation is different from getting $y_{z,t} = y(z_t, x_t, \theta^{new})$ in Algorithm 8 where we only have x_t without knowing both z_t^* and $y_{z,t}^*$. Here, we estimate $y_{z,t}^* = y(z_t^*, x_t, \theta^{new})$ based on given the teaching signal z_t^*.

Still, it relates to updating A, Σ, R^{xy} in Algorithm 11 in that δ_{z,z_t^*} takes a role of I_t though the situation becomes more complicated due to the role of z_t^* and a scalar Gaussian mixture of each component $y_t^{(i)}$.

BYY harmony sparse learning : a dual view

In all the previous sections, the BYY harmony learning implements the maximisation of $H(\theta)$ in Equation 11 without considering a priori $q(\theta)$. In this section, we show that learning performance can be further improved by a priori aided learning from a dual perspective.

We still consider the linear system in Figure 1, where the pair A, Y is observed from a dual view, or called co-dimensional (shortly co-dim) perspective (see Sect.2 in Xu (2011)). Considering a priori $q(A|\rho)$ while ignoring priories on other parameters, we rewrite $H(p||q)$ by Equation 9 into

$$\begin{aligned} H(p||q) = H(\theta, \phi, \rho) = \int p(A|X)[\, H(\theta) + \ln q(A|\rho)]\, dA \\ = \int p(A|X)p(Y|X)p_h^N(X) \ln[\, q(X|AY, \psi)q(Y|\phi)q(A|\rho)]\, dAdY\, dX, \\ = \int p(Y|X)[\, H_d(\theta) + \ln q(Y|\phi)]\, dY \end{aligned} \tag{55}$$

from which we observe that $p(A|X), p(Y|X)$ take a same position in the first line and the last line, respectively, and that $H_d(\theta)$ is actually a dual counterpart of $H(\theta)$ in Equation 11 as follows

$$\begin{aligned} H(\theta) = \int p(Y|X)p_h^N(X) \ln[\, q(X|AY, \psi)q(Y|\phi)]\, dY\, dX, \\ H_d(\theta) = \int p(A|X)p_h^N(X) \ln[\, q(X|AY, \psi)q(A|\rho)]\, dAdX. \end{aligned}$$

This dual view motivates to improve the learning via not only updating A aided with a priori $q(A|\rho)$ but also maximising $H_d(\theta)$.

First, it follows from the second line in Equation 55 with help of Equation 18 that maximising $H(p||q)$ is approximately turned into

$$\begin{aligned} \{A_*, \rho_*\} = argmax_{A,\rho} H(A, \rho, \theta^-), \\ H(p||q) = H(A_*, \rho_*, \theta^-) - \tfrac{1}{2} Tr[\, \Gamma_{X_N}^A \Pi_X^A], \\ \Gamma_X^A = Cov_{p(vec[A]|X)} vec[A]\,,\ \Pi_X^A = -\tfrac{\partial^2 \pi(X_N, AY, \theta)}{\partial vec[A] \partial vec[A]^T}, \\ H(A, \rho, \theta^-) = H(\theta) + \ln q(A|\rho),\ \theta_*^- = argmax_{\theta_*^-} H(\theta), \end{aligned}$$

where θ^- is resulted from removing A, ρ from θ. Anyone of the algorithms introduced in the previous sections can implement the maximisation of the last line above.

Here, we consider the maximisation of the first line, for which we start at

$$\begin{aligned} H(A, \rho, \theta^-) = \int p(Y|X_N)\pi(X_N, AY, \theta) dY, \\ \pi(X_N, AY, \theta) = \ln[\, q(X_N|AY, \theta)q(A|\rho)q(Y|\theta)]\,. \end{aligned}$$

Given $X_N = \{x_t\}_{t=1}^N$ that consists of i.i.d. column vectors, we consider the settings

$$\begin{aligned} q(A|\rho) &= \textstyle\prod_j G(a_j|0, \Sigma_j^a) \\ \ln q(A|\rho) &= -0.5md \ln(2\pi) - 0.5 \textstyle\sum_j \ln|\Sigma_j^a| - 0.5 \sum_j a_j^T \Sigma_j^{a\,-1} a_j, \\ q(X_N|AY,\theta) &= \textstyle\prod_t G(x_t|Ay_t, \Sigma),\ Ay_t = \sum_i a_i y_t^{(i)}, \\ \ln q(X_N|AY,\theta) &= -0.5N[\, d\ln(2\pi) + \ln|\Sigma|] - 0.5 \textstyle\sum_t (x_t - Ay_t)^T \Sigma^{-1}(x_t - Ay_t), \end{aligned} \tag{56}$$

from which we can get

$$\begin{aligned} \nabla_{a_j} H(A,\rho,\theta^-) &= -\Sigma_j^{a\,-1} a_j + N\Sigma^{-1}[\, \mathbf{r}_{xy}^{(j)} - \textstyle\sum_i a_i \lambda_{ij}], \\ \Lambda &= [\, \lambda_{ij}],\ R_{xy} = [\, \mathbf{r}_{xy}^{(1)}, \cdots, \mathbf{r}_{xy}^{(m)}], \end{aligned}$$

where θ^- is obtained from implementing the maximisation of the last line in Equation 56, which are available by the algorithms introduced in the previous sections.

From $\nabla_{a_j} H(A,\rho,\theta^-) = 0, j = 1, \cdots, m$, A is solved by the following equation:

$$\mathcal{B}\, vec(A) = vec(R_{xy}),\ \mathcal{B} = I_{d\times d} \otimes \Lambda + \frac{diag[\Sigma\Sigma_1^{a\,-1}, \cdots, \Sigma\Sigma_m^{a\,-1}]}{N}, \tag{57}$$

where $\otimes$ is the Kronecker product. This equation is equivalent to Eq. (51) in Xu (2011), i.e. the problem of solving a Sylvester matrix equation (Bartels and Stewart 1972; Miyajima 2013).

From $\nabla_{a_j} H(A,\rho,\theta^-)$, we further get the second order derivative as follows

$$\begin{aligned} -\nabla^2_{a_j a_\ell^T} H(A,\rho,\theta^-) &= \Sigma_j^{a\,-1}\delta_{j\ell} + N\Sigma^{-1}\lambda_{j\ell}, \\ \Pi_X^A &= N\Sigma^{-1} \otimes \Lambda + diag[\, \Sigma_1^{a\,-1}, \cdots, \Sigma_m^{a\,-1}]. \end{aligned} \tag{58}$$

Putting it into $H(A,\rho,\theta^-)$ and fixing $\Gamma_{X_N}^A$, we get $\nabla_\rho H(A,\rho,\theta^-)$ and its root as follows

$$\rho^{new} = diag[\, \Sigma_1^a, \cdots, \Sigma_m^a]^{new} = \Gamma_{X_N}^{A\ new} + vec(A^{new})vec(A^{new})^T. \tag{59}$$

Similar to Equation 33, it follows from $H_L(p||q) = (1+\eta)H(p||q) + 0.5\eta \ln|\Gamma_{y|x}|$ that we get

$$\Gamma_{X_N}^{A\ new} = \tfrac{\eta}{1+\eta}\Pi_X^{A\ new}, \tag{60}$$

which is put in the above Equation 59 for updating ρ^{new}.

Computations of $\mathcal{B}, \Gamma_{X_N}^A, \Pi_X^A$ are rather simple since $\Sigma_j^a, \Lambda, \Sigma$ are typically diagonal matrices, and even $\Sigma = \sigma^2 I$. Such uncorrelated structures facilitate learning featured with the nature of automatic model selection, see Sect.2.2 of Xu (2012a) and Sect.2.2 of Xu (2010a), that pushes redundant elements of A towards zeros via pushing its corresponding variances towards zeros. As a result, learning leads to a sparse matrix A.

Such a BYY harmony sparse learning comes from $q(A|\rho)$ that takes a dual role of $q(Y|\phi)$. Being different from the existing sparse learning studies (Shi et al. 2011a, 2014; Tu and Xu 2011a; Xu 2012b) that consider either $q(A|\rho)$ in a long tail distribution with extensive computing cost or $q(A|\rho)$ in Equation 56 with help of one additional $q(\rho)$ (see Sect.III of Xu (2012b)), here the updating by Equation 59 is made by $q(A|\rho)$ in Equation 56 without considering such a priori $q(\rho)$.

Of course, we may progress to consider a priori $q(\rho)$ and also some priories about Λ, Σ, which will lead to another layer of integral about $q(\rho), \Lambda, \Sigma$. Readers are referred to Sect.2.3 in Xu (2011) for the details of implementation.

We may improve all the algorithms introduced in the previous sections, simply with its counterpart of solving A replaced by

$$\text{updating } A \text{ by Equation 57 together with Equations 59 and 60.} \tag{61}$$

which has been already listed in Algorithm 4, Algorithm 5, Algorithm 6, Algorithm 7, Algorithm 8 and Algorithm 12 as one alternative of $A^{new} = R^{xy}\Lambda^{new\,-1}$ in the Ying step.

The implementation of maximising the first line in Equation 55 is featured by the order of integrals $\int [\cdot]\, dYdA$. In a dual view, we may also swap the order to consider maximising the last line in Equation 55. The detailed implementation will be quite similar. Moreover, we may alternatively conduct the two implementations.

De-noise Gaussian mixture

The Gaussian mixture by Equation 28 may also be viewed from a perspective of one specific linear system in Figure 1, with $x_t \in R^d$ generated as follows

$$\begin{gathered} x = Ay + e,\ y = [y^{(1)}, \cdots, y^{(k)}]^T, y^{(j)} = 0 \text{ or } 1,\ \textstyle\sum_{j=1}^{k} y^{(j)} = 1, \\ q(y|\phi) = \textstyle\prod_{\ell=1}^{k} \alpha_\ell^{y^{(\ell)}},\ \phi = \{\alpha_\ell\},\ \textstyle\sum_{\ell=1}^{k} \alpha_\ell = 1, \text{ with } \alpha_\ell \geq 0. \\ q(e|\psi) = G(e|0, \sigma_e^2 I),\ q(A|\rho) = \textstyle\prod_j G(a_j|\mu_j, \Sigma_j), \end{gathered} \tag{62}$$

as proposed in Sect.3.1 of Xu (2011). We have

$$\begin{gathered} q(x|\theta) = \textstyle\sum_y \int q(x|Ay, \psi) q(y|\phi) q(A|\rho) dA \\ = \textstyle\sum_j \alpha_j \int G(x|a_j, \sigma_e^2 I) G(a_j|\mu_j, \Sigma_j) da_j = \textstyle\sum_j \alpha_j G(x|\mu_j, \sigma_e^2 I + \Sigma_j). \end{gathered}$$

That is, we get a Gaussian mixture with each covariance matrix added with the variance of a common noise e. Given $y^{(j)} = 1$, we see that $\hat{x} = x - e = a_j$ comes from $G(a_j|\mu_j, \Sigma_j)$ and provides a de-noised version of observed sample x. Since $y^{(j)}$ takes 1 by a probability α_j, the de-noised $\hat{x}$ actually comes from a mixture $\sum_j \alpha_j G(x|\mu_j, \Sigma_j)$. Thus, this study is called, in Sect.3.1 of Xu (2011), learning *de-noised Gaussian mixture* or shortly de-noised GM.

Somewhat similar to Equation 43, we can rewrite $H_L(\theta)$ in Equation 23 into

$$\begin{gathered} H_L(\theta) = \textstyle\sum_{t=1}^{N} \sum_{\ell=1}^{k} p(\ell|x_t, \theta) [\,(1+\eta) H_L(\theta|\ell, x_t) - \eta \ln p(\ell|x_t, \theta)\,], \\ H_L(\theta|\ell, x_t, \eta) = H(\theta|\ell, x_t) + \tfrac{\eta}{1+\eta} E_{a_\ell|x_t}, E_{a_\ell|x_t} = -\textstyle\int p(a_\ell|x_t, \theta) \ln p(a_\ell|x_t, \theta) da_\ell, \\ H(\theta|\ell, x_t) = \textstyle\int p(a_\ell|x_t, \theta) \pi(x_t, a_\ell, \theta) da_\ell, \\ \pi(x, a_\ell, \theta) = \ln [\, G(x|a_\ell, \sigma_e^2 I) G(a_\ell|\mu_\ell, \Sigma_\ell) \alpha_\ell]. \end{gathered} \tag{63}$$

Similar to Equation 44, we further get

$$\begin{gathered} H(\theta|\ell, x_t) = \pi(x_t, a_\ell, \theta) - \tfrac{1}{2} Tr[\,\Gamma_{a_\ell|x} \Pi_{a_\ell|x}],\ H_{a_\ell|x_t} = 0.5[\ln |\Gamma_{a_\ell|x}| + d \ln (2\pi e)], \\ a_{t,\ell} = arg\max_{a_\ell}\ \pi(x_t, a_\ell, \theta) = W_\ell x_t + w_\ell,\ a_{t,\ell} = [\,\sigma_e^2 I + \Sigma_\ell]^{-1} (\Sigma_\ell x_t + \sigma_e^2 \mu_\ell), \\ W_\ell = [\,\sigma_e^2 I + \Sigma_\ell]^{-1} \Sigma_\ell,\ w_\ell = [\,\sigma_e^2 I + \Sigma_\ell]^{-1} \sigma_e^2 \mu_\ell, \\ \Pi_{a_\ell|x} = (\sigma_e^2)^{-1} I + \Sigma_\ell^{-1},\ \Gamma_{a_\ell|x}^{new} = \tfrac{\eta}{\eta+1} \Pi_{a_\ell|x}^{old\,-1} = \tfrac{\eta}{\eta+1} [\,\sigma_e^2 I + \Sigma_\ell]^{-1} \sigma_e^2 \Sigma_\ell, \\ H_L(\theta|\ell, x_t, \eta) = \pi(x_t, a_\ell, \theta) + \tfrac{0.5\eta}{1+\eta} \ln \tfrac{|\sigma_e^2 \Sigma_\ell|}{|\sigma_e^2 I + \Sigma_\ell|} + c_\eta, \end{gathered} \tag{64}$$

where c_η is a constant that does not relate to θ, ℓ.

Maximising $H_L(\theta)$ with respect to $p(\ell|x_t,\theta)$ yields

$$p(\ell|x_t,\theta)=\frac{e^{[\frac{\eta+1}{\eta}\pi(x_t,a_\ell,\theta)+0.5\ln\frac{|\sigma_e^2\Sigma_\ell|}{|\sigma_e^2 I+\Sigma_\ell|}]}}{\sum_{j=1}^k e^{[\frac{\eta+1}{\eta}\pi(x_t,a_j,\theta)+0.5\ln\frac{|\sigma_e^2\Sigma_j|}{|\sigma_e^2 I+\Sigma_j|}]}}. \tag{65}$$

Then, we maximise $\sum_{t=1}^N\sum_{\ell=1}^k p(\ell|x_t,\theta)H(\theta|\ell,x_t)$ to get θ^{new}, resulting in

$$\begin{aligned}\sigma_e^{2new}&=\frac{Tr[\Gamma_{a_\ell|x}^{old}]}{d}+\frac{\sum_{\ell=1}^k\sum_t p(\ell|x_t,\theta)(x_t-a_{t,\ell})^T(x_t-a_{t,\ell})}{Nd},\\ \alpha_\ell^{new}&=\frac{\sum_t p(\ell|x_t,\theta)}{N},\ \mu_\ell^{new}=\frac{\sum_t p(\ell|x_t,\theta)a_{t,\ell}}{\sum_t p(\ell|x_t,\theta)}.\\ \Sigma_\ell^{new}&=\Gamma_{a_\ell|x}^{old}+\frac{\sum_t p(\ell|x_t,\theta)(a_{t,\ell}-\mu_\ell^{new})(a_{t,\ell}-\mu_\ell^{new})^T}{\sum_t p(\ell|x_t,\theta)}.\end{aligned} \tag{66}$$

Putting the above Equations 66, 65 and 64 into Algorithm 13, we get a new Ying-Yang alternating algorithm for learning de-noise GM, which improves its counterpart in Sect.3.1 of Xu (2011) in that the Lagrange technique used in Algorithm 3 is used to help the Ying-Yang alternative implementation. Also, $p(\ell|x_t,\theta)$ in Equation 65 has been extended to cover semi-supervised learning in the same way as in Algorithm 9.

Algorithm 13 BYY learning for de-noise GM

Require: initialise θ,η, let $p_{\ell,t}=1/k,\Gamma_{a_\ell|x}=0,a_{t,\ell}=x_t$.

Repeat the following two steps **until** converged:

***Ying*-Step:** *get* $\alpha_\ell^{new},\mu_\ell^{new},\Sigma_\ell^{new}$by Equation 66 as follows:,

$\sigma_e^{2\ new}=\frac{Tr[\Gamma_{a_\ell|x}^{old}]}{d}+\frac{\sum_{\ell=1}^k\sum_t p(\ell|x_t,\theta)(x_t-a_{t,\ell})^T(x_t-a_{t,\ell})}{Nd}$,

$n_\ell=\sum_{t=1}^N p_{\ell,t},\ \alpha_\ell^{new}=\frac{n_\ell}{\sum_{j=1}^k n_j},\ \mu_\ell^{new}=\frac{1}{n_\ell}\sum_{t=1}^N p_{\ell,t}a_{t,\ell}$,

$\Sigma_\ell^{new}=\Gamma_{a_\ell|x}^{old}+\frac{1}{n_\ell}\sum_{t=1}^N p_{\ell,t}(a_{t,\ell}-\mu_\ell^{new})(a_{t,\ell}-\mu_\ell^{new})^T$.

trimming:

if $\alpha_i^{new}\to 0$ or $\alpha_i^{new}Tr[\Sigma_i^{new}]\to 0$, discard the ith Gaussian, let k=k-1.

***Yang*-Step**: for $t=1,\cdots,N$ and $\ell=1,\cdots,k$, get $p_{\ell,t}=(\eta+1+\gamma\delta_{\ell,\ell_t^*})p_{\ell|x_t}(\theta^{new})$with

$$p(\ell|x_t,\theta)=\frac{e^{[\frac{\gamma\delta_{\ell,\ell_t^*}+1+\eta}{\eta}\pi(x_t,a_\ell,\theta)+0.5\ln\frac{|\sigma_e^2\Sigma_\ell|}{|\sigma_e^2 I+\Sigma_\ell|}]}}{\sum_{j=1}^k e^{[\frac{\gamma\delta_{j,\ell_t^*}+1+\eta}{\eta}\pi(x_t,a_j,\theta)+0.5\ln\frac{|\sigma_e^2\Sigma_j|}{|\sigma_e^2 I+\Sigma_j|}]}},$$

$\pi(x,a_\ell,\theta)=\ln[\,G(x|a_\ell,\sigma_e^2 I)G(a_\ell|\mu_\ell,\Sigma_\ell)\alpha_\ell]$.

Also, get$a_{t,\ell}=[\,\sigma_e^{2\ new}I+\Sigma_\ell^{new}]^{-1}\,(\Sigma_\ell^{new}x_t+\sigma_e^{2\ new}\mu_\ell^{new})$,

$\Gamma_{a_\ell|x}^{new}=\frac{\eta}{\eta+1}[\,\sigma_e^{2\ new}I+\Sigma_\ell^{new}]^{-1}\,\sigma_e^{2\ new}\Sigma_\ell^{new}$.

Remarks:

(a) When $\sigma_e^2=0$, this algorithm degenerates to become the same as Algorithm 9 and further to Algorithm 3 if also $\gamma=0$.

(b) For each sample x_t, $\delta_{\ell,\ell_t^*}=0$ if it has no teaching label and $\delta_{\ell,\ell_t^*}=1$ when ℓ is equal to a given teaching label ℓ_t^*.

(c) $a_{t,\ell}$ outcomes the de-noised samples for each cluster. Also, classification of a sample x_t can be made by $\ell^*=arg\max_\ell p_{\ell,t}$ and thus a_{t,ℓ^*} is treated as the de-noised sample of x_t.

Conventionally, noises are filtered by a preprocess (if needed) with help of a standard noise filtering method. In many applications, however, the problems of filtering noises and making clustering or density estimation are actually two coupled tasks. Instead, the de-noise GM provides a model to consider both in a same learning process, while Algorithm 13 provides a useful tool that implements both the tasks. Moreover, we can include some knowledge (e.g. teaching labels) in an easy way. One example is its potential application to image segmentation. Applying to a noisy image, $a_{t,\ell}$ outcomes de-noised pixels for each segmented region, while pixel classification can be made by $\ell^* = arg\max_\ell p_{\ell,t}$. For a sharpen image, we may merely use a_{t,ℓ^*} as the de-noised pixels of each segmented region.

Sparse linear and logistic regression

When we are given a set of paired samples $\{x_t, y_t^*\}$, the FA model by Equation 31 actually performs a multiple linear regression for the following mapping $y \to x$:

$$x = Ay + \mu + e,\ y = [\, y^{(1)}, \cdots, y^{(k)}]^T, q(y|\phi) = G(y|0, \Lambda),\ q(e) = G(e|0, \sigma^2 I). \quad (67)$$

Though we may directly use Algorithm 11 for learning, it is difficult to trim off the redundant elements of y via checking whether $\lambda_i \to 0$ in the Ying step of Algorithm 4. In this case, the contribution of $\{y_t^*\}$ will make none λ_i in Λ in Equation 34 tend to zero. In contrast, learning by Algorithm 4 and Algorithm 11 is still able to push redundant elements of A towards zero when updating A is made by Equation 57 together with Equations 59 and 60.

For clarity, we simplify Algorithm 4 and Algorithm 11 into Algorithm 14. Its Yang step is directly Equation 60. The Ying step is a simplification of Equation 57 together with Equation 59 at $\Sigma = \sigma^2 I$, plus a new equation for updating σ^2. All the updating aims to maximise $H(p||q)$ of Equation 56 in a simplification as follows

$$H(p||q) = \ln[\, q(X_N|AY, \theta) q(A)] - \frac{1}{2} Tr[\, \Gamma_{X_N}^A \Pi_X^A],$$

with $q(A) = q(A|\rho)$ given by Equation 56. Also, from Equation 67 we have $q(X_N|AY, \theta) = \prod_t G(x_t|Ay_t + \mu, \Sigma)$.

To get a further insight, we observe a special case that x_t is simply univariate, i.e. $d = 1$, at which A becomes a vector a^T and Equation 67 actually becomes the widely studied linear regression problem, for which Algorithm 14 is simplified into Algorithm 15. It differs from the ordinary linear regression in that Λ is corrected by a term $\frac{\sigma^2}{N} \Sigma^{a\,-1}$ for solving a.

Algorithm 14 BYY harmony sparse learning for multi-dimensional regression

Require: get $\{x_t, y_t^*\}$ with $\nu = \frac{\sum_t y_t^*}{N}$, let $\Lambda = \frac{\sum_t (y_t^* - \nu)(y_t^* - \nu)^T}{N}$. Initialise $A = 0, \Gamma = 0$.

Repeat the following two steps **until** converged:

***Ying*-Step:** get

$$e_t = x_t - A^{old} y_t,\ \mu^{new} = \frac{1}{N} \sum_t e_t,\ \sigma^{2\ new} = \frac{Tr[\Gamma^{old}(I \otimes \Lambda)]}{d} + \frac{\sum_t (e_t - \mu^{new})(e_t - \mu^{new})^T}{Nd},$$

$$R_{xy} = \frac{1}{N} \sum_t (e_t - \mu^{new})(y_t - \nu)^T,\ diag[\, \Sigma_1^a, \cdots, \Sigma_m^a] = \Gamma^{old} + vec(A^{old}) vec(A^{old})^T,$$

$$\Pi_X^{A\ new} = N\sigma^{-2\ new} I \otimes \Lambda + diag[\, \Sigma_1^a, \cdots, \Sigma_m^a]^{-1},$$

get A^{new} by solving $\mathcal{B}\, vec(A) = vec(R_{xy}),\ \mathcal{B} = I_{d \times d} \otimes \Lambda + \sigma^{2\ new} \frac{diag[\Sigma_1^a, \cdots, \Sigma_m^a]^{-1}}{N}$.

***Yang*-Step**: $\Gamma^{new} = \frac{\eta}{1+\eta} (\Pi_X^{A\ new})^{-1}$.

Algorithm 15 BYY harmony sparse learning for linear regression $x = y^T a + \mu + e$

Require: get $\{x_t, y_t^*\}$ with $\nu = \frac{\sum_t y_t^*}{N}$, let $\Lambda = \frac{\sum_t (y_t^* - \nu)(y_t^* - \nu)^T}{N}$. Initialise $a = 0, \Gamma = 0$.

Repeat the following two steps **until** converged:

***Ying*-Step:** get

$e_t = x_t - y_t^T a^{old},\ \mu^{new} = \frac{1}{N}\sum_t e_t,\ \Sigma^a = \Gamma^{old} + a^{old} a^{old\ T},$

$\sigma^2 = \frac{Tr[\Gamma^{old}\Lambda]}{d} + \frac{\sum_t (e_t - \mu^{new})(e_t - \mu^{new})^T}{Nd},\ R_{yx} = \frac{\sum_t (y_t - \nu)(e_t - \mu^{new})^T}{N},$

$a^{new} = [\Lambda + \frac{\sigma^2}{N}\Sigma^{a\,-1}]^{-1} R_{yx}.$

***Yang*-Step**: $\Gamma^{new} = \frac{\eta}{1+\eta}(\frac{N\Lambda}{\sigma^2} + \Sigma^{a\,-1})^{-1}$.

Also, we may maximise $H(p||q)$ to make sparse learning for a multiple logistic regression by

$$\begin{gathered}
\ln q(X_N|AY,\theta) = \textstyle\sum_t \ln q(x_t|\hat{A}y_t + \mu), \\
\ln q(x_t|Ay_t + \mu) = \textstyle\sum_{i=1}^d [\, x_t^{(i)} \ln s(\hat{x}_t^{(i)}) + (1 - x_t^{(i)}) \ln (1 - s(\hat{x}_t^{(i)}))], \ \hat{x}_t = Ay_t + \mu, \\
q(A|\rho) = \textstyle\prod_j G(a_j|0, \Sigma_j^a),\ q(\mu) = G(\mu|0, \Sigma^\mu),
\end{gathered} \tag{68}$$

where $0 \le s(r) \le 1$ is a sigmoid function, e.g. simply

$$s(r) = 1/(1 + e^{-r}). \tag{69}$$

We further get

$$\begin{gathered}
\nabla_{a_j} \ln q(A) = -\Sigma_j^{a\,-1} a_j,\ \nabla_\mu \ln q(\mu) = -\Sigma^{\mu\,-1}, \\
\delta a_j = \nabla_{a_j} \ln q(X_N|AY,\theta) = \textstyle\sum_t \xi_t^{(j)} s'(\hat{x}_t^{(j)}) y_t, \\
where\ s'(r) = \tfrac{ds(r)}{dr},\ \xi_t^{(i)} = \tfrac{x_t^{(i)}}{s(\hat{x}_t^{(i)})} - \tfrac{1 - x_t^{(i)}}{1 - s(\hat{x}_t^{(i)})}, \\
\delta\mu = \nabla_\mu \ln q(X_N|AY,\theta) = [\textstyle\sum_t \xi_t^{(1)} s'(\hat{x}_t^{(1)}), \cdots, \sum_t \xi_t^{(d)} s'(\hat{x}_t^{(d)})]^T, \\
\Pi_{a_j} = -\nabla_{a_j a_j^T} \ln q(X_N|AY,\theta) = \textstyle\sum_t w_t^{(j)} y_t y_t^T, \\
\Pi_\mu = -\nabla_{\mu\mu^T} \ln q(X_N|AY,\theta) = diag[\textstyle\sum_t w_t^{(1)}, \cdots, \sum_t w_t^{(d)}], \\
w_t^{(i)} = \xi_t^{(i)} s''(\hat{x}_t^{(i)}) + \tfrac{x_t^{(i)} s'^2(\hat{x}_t^{(i)})}{s^2(\hat{x}_t^{(i)})} + \tfrac{(1 - x_t^{(i)}) s'^2(\hat{x}_t^{(i)})}{(1 - s(\hat{x}_t^{(i)}))^2},\ with\ s''(r) = \tfrac{d^2 s(r)}{d^2 r},
\end{gathered} \tag{70}$$

from which we get Algorithm 16 to make the BYY sparse learning for logistic regression. Similar to Equation 60, we get its Yang step. Similar to Equations 58 and 59, we have

$$\begin{gathered}
\Pi_X^{a_j} = \Pi_{a_j} + \Sigma_j^{a\,-1},\ \Pi_X^\mu = \Pi_\mu + \Sigma^{\mu\,-1}, \\
\Sigma_j^a = \Gamma_{X_N}^{a_j\ new} + a_j a_j^T,\ \Sigma^\mu = \Gamma_{X_N}^{\mu\ new} + \mu\mu^T,
\end{gathered} \tag{71}$$

which is put into the Ying step of Algorithm 16. Being different from Algorithm 14, there is no need to consider $\Sigma = \sigma^2 I$, while updating A, μ is made by gradient ascending instead of solving nonlinear equation.

Algorithm 16 Sparse learning for logistic regression

Require: get $\{x_t, y_t^*\}$ with $\nu = \frac{\sum_t y_t^*}{N}, \Lambda = \frac{\sum_t (y_t^*-\nu)(y_t^*-\nu)^T}{N}$. Initialise $A = 0, \Gamma_{X_N}^{a_j} = 0, \Gamma_{X_N}^{\mu} = 0$.

Repeat the following two steps **until** converged:

***Ying*-Step:** for $j = 1, \cdots, m$, get

$\Sigma_j^a = \Gamma_{X_N}^{a_j\ old} + a_j^{old} a_j^{old\ T},\ \Pi_X^{a_j} = \Pi_{a_j} + \Sigma_j^{a\ -1},\ a_j^{new} = a_j^{old} + \zeta(\delta a_j^{new} - \Sigma_j^{a\ -1} a_j^{old}),$

$\Sigma^{\mu} = \Gamma_{X_N}^{\mu\ old} + \mu^{old}\mu^{old\ T},\ \Pi_X^{\mu} = \Pi_{\mu} + \Sigma^{\mu\ -1},\ \mu^{new} = \mu^{old} + \zeta(\delta\mu^{new} - \Sigma^{\mu\ -1}\mu^{old}),$

with $\Pi_X^{a_j}, \Pi_X^{\mu}$ in Equation 71 and $\delta a_j, \delta\mu$ in Equation 70, where $\zeta >$ is a small stepsize.

***Yang*-Step**: *for* $j = 1, \cdots, m$, get $\Gamma_{X_N}^{a_j\ new} = \frac{\eta}{1+\eta}(\Pi_X^{a_j\ new})^{-1},\ \Gamma_{X_N}^{\mu\ new} = \frac{\eta}{1+\eta}(\Pi_X^{\mu\ new})^{-1}$.

Temporal FA and temporal binary FA

The FA model by Equation 31 has been extended to modelling temporal dependence in (Xu 1999a,2001b,2004a) by adding the following vector based auto-regression

$$y_t = By_{t-1} + \varepsilon_t,\ q(\varepsilon_t|\phi) = G(\varepsilon_t|0, \Lambda). \tag{72}$$

The joint modelling by Equations 31 and 72 is called temporal factor analysis, shortly temporal FA or TFA.

Learning TFA can be implemented by maximising $H(p||q)$ as follows:

$$\begin{aligned} H(p||q) &= H_1(p||q) + H_2(p||q) - \textstyle\sum_t \ln G(y_t|By_{t-1}, \Lambda) \\ H_1(p||q) &= \textstyle\sum_t \pi_1(x_t, Ay_t, \theta) - \frac{1}{2} Tr[\,\Gamma_{X_N}^A \Pi_X^A] - \frac{1}{2} Tr[\,\Gamma_{y|x}\Pi_{Y|X}]\,, \\ H_2(p||q) &= \textstyle\sum_t \pi_2(y_t, By_{t-1}, \psi) - \frac{1}{2} Tr[\,\Gamma_{X_N}^B \Pi^{B_X}]\,, \\ \pi_1(x_t, Ay_t, \theta) &= \ln[\,q(x_t|Ay_t, \Sigma)G(y_t|By_{t-1}, \Lambda)] + \tfrac{1}{N}\ln q(A|\rho_A), \\ \pi_2(y_t, By_{t-1}, \psi) &= \ln[\,G(y_t|By_{t-1}, \Lambda)G(y_{t-1}|0, \Omega_{t-1})] + \tfrac{1}{N}\ln q(B|\rho_B). \end{aligned} \tag{73}$$

Given $\nu = By_{t-1}$ fixed, maximising $H_1(p||q)$ is decoupled from $H_2(p||q) - \sum_t \ln G(y_t|By_{t-1}, \Lambda)$ and thus is handled exactly by learning FA, as summarised in Part-A of Algorithm 17. With Λ fixed, maximising $H_2(p||q)$ is decoupled from $H_1(p||q) - \sum_t \ln G(y_t|By_{t-1}, \Lambda)$ too. Also, samples of $\{y_t, y_{t-1}\}$ are available from implementing Part-A. The problem of maximising $H_2(p||q)$ is equivalent to the special case of a multiple linear regression at $\mu = 0, \nu = 0$ and $d = m$, and thus B can be learned by Algorithm 14, as summarised in Part-B of Algorithm 17.

One additional issue needs to be handled. The implementation of Part-A needs to know the covariance matrix Ω_{t-1} of y_{t-1}, which takes the position of Λ in Algorithm 14. It follows from $y_t = By_{t-1} + \varepsilon_t$ that we have the following equation as a constraint:

$$\Omega_t = B\Omega_{t-1}B^T + \Lambda, \tag{74}$$

which may be recursively updated from Ω_0 after Λ updated in Part-A and B updated in Part-B.

Algorithm 17 BYY harmony sparse learning for TFA

Require: consider $x_t = Ay_t + e_t, y_t = By_{t-1} + \varepsilon_t$. Initialise $A = 0, B = 0, \Omega_0 = I$.

Repeat the following three parts **until** converged:

***Part*-A:** update A by Equation 57 together with Equations 59 and 60; update Σ, Λ, ρ_A and get $Y = \{y_t\}$ by Algorithms 4 and 11.

***Part*-B:** with $Y = \{y_t\}$ and Λ obtained from the above,
get Ω_{t-1} from Part-C below, update B, ρ_B by Algorithm 14 with the following substitutions:
y_t in the place of x_t, and y_{t-1} in the place of y_t,
Λ in the place of Σ, and B in the place of A,
Ω_{t-1} in the place of Λ, and ρ_B in the place of ρ_A.

***Part*-C:** get Λ in Part-A and B in Part-B, update Ω by using Equation 74 or solving Equation 75.

When $\{y_t\}$ is a stationary process with $\Omega_{t-1} \to \Omega$ as $t \to \infty$, Equation 74 becomes $\Omega = B\Omega B^T + \Lambda$ or

$$[I - (B \otimes B)]\, vec(\Omega) = vec(\Lambda), \tag{75}$$

from which we may get Ω by solving this equation.

In a similar way, we may also extend the binary FA by Equation 46 to modelling temporal dependence by the following modification

$$q(x_t|y_t, \psi) = G(x|Ay + \mu, \Sigma),\ q(y_t|y_{t-1}, \phi) = \textstyle\prod_i \alpha_i^{y_t^{(i)}} (1 - \alpha_i)^{1-y_t^{(i)}},$$
$$\alpha_i = s(\hat{y}_t^{(i)}),\ [\hat{y}_t^{(1)}, \cdots, \hat{y}_t^{(m)}]^T = \hat{y}_t = By_{t-1} + \nu,$$

where $y_t^{(i)}$ takes either 0 or 1, and $s(r)$ is a sigmoid function, e.g. by Equation 69. This model is called temporal binary factor analysis, shortly temporal BFA.

Similar to Equation 73, learning temporal BFA can be implemented by maximising $H(p||q)$ with help of maximising $H_1(p||q)$ by Algorithm 6 for learning BFA with α_i fixed, as summarised in Part-A of Algorithm 18, and with help of maximising $H_2(p||q)$ by Algorithm 16 to learn B, ν for logistic regression $y_{t-1} \to y_t$, as summarised in Part-A of Algorithm 18.

Algorithm 18 BYY sparse learning for TBFA

Require: consider $x = Ay + \mu + e, y_t = By_{t-1} + \varepsilon_t$. Initialise $A = 0, B = 0, \Omega_0 = I$.

Repeat the following two parts **until** converged:

***Part*-A:** update A by Equation 57 together with Equations 59 and 60;
update Σ, Λ, ρ_A and get $Y = \{y_t\}$ by Algorithm 6.

***Part*-B:** with $Y = \{y_t\}$ above, update B, ν, ρ_B by Algorithm 16 with the following substitutions:
y_t in the place of x_t and y_{t-1} in the place of y_t,
B in the place of A and ρ_B in the place of ρ_A, μ in the place of μ.

Bi-linear matrix system and manifold learning

Putting samples $x_t, e_t, y_t = 1, \cdots, N$ into their corresponding matrix formats $X, E \in R^{d\times N}, Y \in R^{k\times N}$, respectively, we extend the FA model $x = Ay + e$ in Equation 31 into the following generalised bi-linear matrix system (BMS)

$$X = \mu(AY) + E,\ E = \left[e_t^{(i)}\right],\ q(X|Y,\theta) = q(X - \mu(AY)) = q(E|Y), \tag{76}$$
$$q(E|Y) = \prod_{t=1}^{N}\prod_{i=1}^{d} q(e_t^{(i)}|Y),$$

where $\mu(AY)$ is an inverse link function of AY that is linear to either one of A, Y with the other fixed, and $\mu(\Omega) = [\,\mu(\omega_{i,j})]$ for a matrix $\Omega = [\,\omega_{i,j}]$.

It can be used as a general formulation for existing typical linear models, classified by whether one or more of the following three natures are possessed.

Additive noise E The BMS is called additive or non-additive based on whether or not $q(E|Y) = q(E)$. One typical additive family is that elements of E are independent Gaussian noises, i.e.

$$q\left(e_t^{(i)}|Y\right) = G\left(e_t^{(i)}|0,\sigma_t^{(i)\,2}\right). \tag{77}$$

Independent factors Y We get a BMS, featured by whether we have

$$q(Y|\theta) = \prod_{t=1}^{N}\prod_{j=1}^{k} q\left(y_t^{(j)}\right). \tag{78}$$

Link function μ The BMS is called bi-linear according to whether

$$\mu(\xi) = \xi. \tag{79}$$

The special cases of the BMS featured with Equations 77, 78 and 79 all satisfied include FA, BFA, NFA and others. Also, their corresponding implementations of BYY harmony learning are previously introduced by Algorithms 4 to 16. The special cases with only Equations 78 and 79 satisfied were addressed in Sect. 2 of Xu (2011).

Beyond Equation 78, the generalised BMS models with Equations 77 and 79 held were also previously addressed in Sect.II of Xu (2012b) and Sect.5 of Xu (2012a). One type is temporal learning featured by autoregression across columns of Y, e.g. by Equation 72, rather extensively studied since 2000 (Xu 2000b, 2001b, 2004a). A recent summary about TFA studies is referred to Sect.5.2 of Xu (2012a).

The other type is manifold learning featured by that Y comes from the following matrix normal distribution (MND) (Dutilleul 1999; Gupta and Nagar 1999; Xu 2012b):

$$N(U|C,\Omega,\Sigma) = \frac{e^{-0.5Tr[\Omega^{-1}(U-C)^T\Sigma^{-1}(U-C)]}}{(2\pi)^{0.5kN}|\Sigma|^{0.5k}|\Omega|^{0.5N}},$$

where a matrix Ω describes the cross-column dependence of the matrix variate U, and a matrix Σ describes the cross-row dependence of U. This matrix distribution is equivalent to a multivariate Gaussian distribution $G(vec(U)|vec(C), \Sigma \otimes \Omega)$.

One example is $q(Y|\theta) = N(Y|0, L^{-1}, I)$ with L given by the graph Laplacian, which was firstly considered by Eq.(27) in Xu (2012b) and led to a BYY harmony based manifold learning. Such an insight may be observed from Equation 19. Maximisation of $H(\theta)$ subject to Equation 21 consists of maximising $\pi(X,Y,\theta) = \ln q(X|Y,\theta) - 0.5(kNln(2\pi) - \ln|L|) - 0.5T_n$ that includes to minimise

$$T_n = Tr[\,YLY^T\,], \tag{80}$$

which is a key term in the Laplacian eigenmaps for preserving topologically the neighbourhood relation in manifold learning (Belkin and Niyogi 2003). Differently, the BYY

harmony learning obtains $Y_* = arg\max_Y \pi(X_N, Y, \theta)$ in place of learning an approximate linear mapping $Y \approx WX$.

The other example is $q(Y|\theta) = N(Y|0, L^{-1}, \Lambda)$ that was firstly given by Eq.(107) in Xu (2012a), which is featured by a diagonal matrix Λ that is added in as free parameters to be adjusted. Accordingly, Equation 80 is modified into

$$T_n = \ln|\Lambda| + Tr[\,\Lambda^{-1} YLY^T\,]. \tag{81}$$

This Λ takes a role similar to the one in Algorithm 4. Actually, this situation can be regarded as an extended counterpart of FA-b while the situation with T_n by Equation 80 can be regarded as an extended counterpart of FA-a. The BYY harmony learning helps to learn Λ for determining an appropriate manifold dimension k, i.e. the row dimension of Y. Following the schematic Algorithm 2, we can develop one detailed BYY harmony learning algorithm for implementing the BMS by Equation 76.

Conceptually, there are also other choices beyond Equation 78, which can be very diversified. The next subsection further examines a family of choices featured with certain decoupled parts of Y.

Decoupled BMS, regulatory networks and LMM model

We may narrow our consideration on the dependence among the parts of Y by considering linear dependence within Y by a linear product of a matrix B for describing dependence and a matrix with mutually independent elements, that is, we consider

$$X = \mu(AYB^T) + E, \text{with Equation 78 satisfied.} \tag{82}$$

This formulation extends those previous models for independent factor analyses into their counterparts in a BMS formulation.

From the perspective of making the maximum likelihood learning on the parametric distribution of X, the formulation by Equation 82 includes the ones by both Equations 80 and 81 as its special cases with

$$\eta(r) = r \text{ and } q(Y|\theta) = N(Y|0, \Lambda_c, \Lambda_r), \tag{83}$$

where both Λ_c, Λ_r are diagonal matrices.

It follows from Theorem 2.3.10 in (Gupta and Nagar 1999) that we have $N(Y_B|0, B\Lambda_c B^T, \Lambda_r)$ with $YB^T = Y_B$. Let $L^{-1} = BB^T$ and E by Equation 76, we are led to Equation 80 when $\Lambda_r = I$ and to Equation 81 when $\Lambda_r \neq I$. Generally, we may also consider Equation 78 with elements from other Gaussian and nonGaussian distributions.

The above relations do not hold for the BYY harmony learning even when Equation 79 holds. Instead, the formulation by Equation 82 is more preferred than its counterpart in Equation 76. During the implementation of the BYY harmony learning, either $q(Y|\theta) = N(Y|0, L^{-1}, I)$ or $q(Y|\theta) = N(Y|0, L^{-1}, \Lambda)$ takes a role of to controlling the compleixty of Y, i.e. the row dimension and also the matrix sparsity.

Usually, B is not learned but provided from or designed based on a sample set X, e.g. $L^{-1} = BB^T$. Sometimes, B is learned subject to the following constraint

$$B = B_o D_B, D_B \text{ is diagonal}, B_o^T B_o = I \text{ with elements being either 0 or 1.} \tag{84}$$

Gene regulatory networks (TRN) takes an important role in biology networks and modelling TRN based on gene expression data is one of major topics in the studies of computational genomics (Bar-Joseph et al. 2012; Karlebach and Shamir 2008; Morris and

Mattick 2014). In the previous efforts (Tu et al. 2011, 2012a,b), the BFA and NFA have been applied to model gene transcriptional regulation, which leads to improvements of networks component analysis (NCA) (Liao et al. 2003). Still, Equations 82 and 83 with $\mu(r) = r$ jointly also provide a new TRN model. Instead of pre-specifying the topology of A according to some priori knowledge (Liao et al. 2003), we get the topological information underlying the samples of X by the graph Laplacian L and then get B by $L^{-1} = BB^T$, while A is obtained via learning with or without pre-specifying its topology. Also, we may consider a priori of A with help of Equation 56. During the implementation of the BYY harmony learning, an appropriate number of transcription factors may be determined via learning the diagonal matrix Λ_r.

We may further partition $Y = [Y_s, F]$ and correspondingly $B = [B_s, Z]$, with elements of Y_s being stochastic variables and elements of F being unknown constants, where F and Z could be empty when all the elements of Y are stochastic variables. By this partition, we get $AYB^T = AY_sB_s^T + AFZ^T$. For simplicity, we drop the subscript s and still use F to denote the unknown constant matrix product AF instead of further decomposing it into two parts. Similarly, we also partition Z and get a constant offset term C. As a result, Equation 82 is rewritten into

$$X = \mu(AYB^T + FZ^T + C) + E, \text{ together with Equation 78,} \tag{85}$$

which returns back to Equation 82 simply with $F = 0$.

Let $AY = Y_A$, we have $E(Y_A Y_A^T) = AE(YY^T)A^T$. When the columns of Y are i.i.d. from a Gaussian with a zero mean and a diagonal covariance matrix Λ_r, Equation 85 becomes $X = Y_A B^T + FZ^T + E$ with Y_A denoting random effects and F denoting fixed effects; that is, we are led to the linear mixture model (LMM) when $\mu(r) = r$ and generalised LMM (GLMM) when $\mu(r) \neq r$.

Unknowns in LMM or GLMM may be estimated by one of the algorithms developed in the literature of statistics under the principle of the least square error or maximum likelihood (Demidenko 2013). Both LMM and GLMM have been applied for modeling various associations in the studies of biology and recently in the studies of computational genomics (Yang et al. 2014; Zhou and Stephens 2014; Zou et al. 2014). The BYY harmony learning provides one alternative method for estimating the unknowns in LMM or GLMM, with one advantage of determining an appropriate row dimension of Y and a sparse matrix A. Conventionally, B and Z are design matrices that are usually pre-specified based on given samples and priori knowledge. Also, either or both of B and Z may consist of partially given elements and partially unknowns to be estimated via learning. One example is shown in Figure four in Xu (2011).

Following the schematic Algorithm 2, we can further develop the detailed BYY harmony learning algorithm for learning Equation 85. Here, we consider the special case $\mu(r) = r$ to learn Ψ that consists of all the unknowns (i.e. A, C, Σ_c, Σ_r and the rest unknowns). Noticing that $vec(AYB^T + FZ^T + C) = vec(AYB^T) + vec(FZ^T) + vec(C)$, $vec(AYB^T) = (B \otimes A)vec(Y) = (BY^T \otimes I)vec(A)$ and $vec(FZ^T) = (Z \otimes I)vec(F)$, it follows from Equations 19 and 21 that we learn Ψ by

$$\max_{\Psi,Y,F} \pi(\Psi, Y, F), \text{ where}$$
$$\pi(\Psi, Y, F) = \ln [N(E|0, \Sigma_c, \Sigma_r)N(Y|0, \Lambda_c, \Lambda_r)N(A|0, D_c, D_r)N(F|0, K_c, K_r)],$$

$$E = X - \mu(AYB^T + FZ^T + C),\ \Sigma_E = \Sigma_c \otimes \Sigma_r,$$

subject to (86)

$$vec(Y_*) = arg\max_Y \pi(\Psi, Y, F) = \Gamma_X^Y (B \otimes A)^T \Sigma_E^{-1} vec(X - FZ^T - C),$$

$$vec(A_*) = arg\max_A \pi(\Psi, Y, F) = \Gamma_X^A (YB^T \otimes I)^T \Sigma_E^{-1} vec(X - FZ^T - C),$$

$$vec(F_*) = arg\max_F \pi(\Psi, Y, F) = \Gamma_X^F (Z \otimes I)^T \Sigma_E^{-1} vec(X - AYB^T - C),$$

$$\Gamma_X^Y = \frac{\eta}{\eta+1}\Pi_X^{Y\ -1},\ \Gamma_X^A = \frac{\eta}{\eta+1}\Pi_X^{A\ -1},\ \Gamma_X^F = \frac{\eta}{\eta+1}\Pi_X^{F\ -1},$$

$$\Pi_X^Y = -\frac{\partial^2 \pi(\Psi, Y, F)}{\partial vec(Y) \partial vec(Y)^T} = (B \otimes A)^T \Sigma_E^{-1} (B \otimes A) + (\Lambda_c \otimes \Lambda_r)^{-1}.$$

$$\Pi_X^A = -\frac{\partial^2 \pi(\Psi, Y, F)}{\partial vec(A) \partial vec(A)^T} = (YB^T \otimes I)^T \Sigma_E^{-1} (YB^T \otimes I) + (D_c \otimes D_r)^{-1}.$$

$$\Pi_X^F = -\frac{\partial^2 \pi(\Psi, Y, F)}{\partial vec(F) \partial vec(F)^T} = (Z \otimes I)^T \Sigma_E^{-1} (Z \otimes I) + (K_c \otimes K_r)^{-1}.$$

A preservation principle of multiple convex combination

We observe the following estimators for the sample mean and sample covariance:

$$\mu = \frac{1}{N}\sum_{t=1}^N x_t,\ \Sigma = \frac{1}{N}\sum_{t=1}^N (x_t - \mu)(x_t - \mu)^T,$$

each of which is featured by a convex combination of a number of individual statistics x_t or $(x_t - \mu)(x_t - \mu)^T$. Also, we observe the Ying step of Algorithm 3 and find that μ_ℓ, Σ_ℓ are such convex combinations too. Actually, such convex combinations can be found in also the algorithms introduced in the previous sections.

Moreover, the harmony functional $H(p||q)$ by Equation 9 is an estimation function that comes from a convex combination of an infinite many of individual estimation function featured by the Ying machine $q(X|R)q(R)$ at an infinite many individuals of R, weighted by the Yang machine $p(R|X)p(X)$.

The above examples are all the explicit combinations of explicit individual statistics or estimation functions. Even generally, such a convex combination applies to many implicit functions. For example, we examine the following convex combination

$$f(\mu) = \sum_t a_t f_t(\mu),\ f_t(\mu) = \|x_t - \mu\|^2,\ \sum_t a_t = 1, a_t \geq 0, \tag{87}$$

from which we observe the following natures:

(a) The gradient field $\nabla_\mu f(\mu)$ is a convex combination of the gradient fields $\{\nabla_\mu f_t(\mu)\}_{t=1}^N$.

(b) The root of $\nabla_\mu f(\mu) = 0$ is also a convex combination of the roots of $\{\nabla_\mu f_t(\mu) = 0\}_{t=1}^N$.

(c) The minimum of $f(\mu)$ is a convex combination of the minimums of $\{f_t(\mu)\}_{t=1}^N$ too.

These natures are closely related to the first order derivative or the gradient field of estimation functions. The nature (a) describes a global feature of the gradient fields of estimation functions, and the nature (b) describes features within some important local areas (e.g. around the sinks) of these gradient fields. While the nature (c) is equivalent to the nature (b) if $\{f_t(\mu)\}_{t=1}^N$ have gradient fields. Generally, the nature (c) may even apply to those individual estimation functions that do not have gradient fields.

In Equation 87, a convex combination of individual convex functions implies or induces all the above three natures. Given a convex combination of individual convex functions, if it also preserves at least one of the three natures above, we say that it preserves a nature of *multiple convex combination* (MCC). The classic maximum likelihood learning preserves such a MCC nature too, because both $(1/N)\sum_{t=1}^{N}\ln q(x_t|\theta)$ and $(1/N)\sum_{t=1}^{N}\nabla_\theta \ln q(x_t|\theta)$ are convex combinations.

Such a nature is not implied everywhere. One example is the BYY harmony learning subject to Equation 12 as follows

$$H(\theta)=\int p(Y|\theta,X_N)\pi(X_N,Y,\theta)dY,\ \text{s.t.}\ p(Y|\theta,X_N)=q(Y|\theta,X_N),$$

which is a special case of Equation 9 and thus is still a convex combination of an infinite many of individual estimators $\pi(X_N,Y,\theta)$ at an infinite many individual values of Y, weighted by the Yang machine $p(Y|\theta,X_N)$. But, considering the gradient field directly may not preserve the MCC nature.

As shown in Eq.(25) of Xu (2010a), we get such a gradient field as follows:

$$\begin{aligned}\nabla_\varphi H(\theta)&=\int p_\delta(Y|\theta,X_N)\nabla_\varphi\pi(X_N,Y,\theta)dY,\\ p_\delta(Y|\theta,X_N)&=p(Y|\theta,X_N)[\,1+\Delta\pi(X_N,Y,\theta)\,],\\ \Delta\pi(X_N,Y,\theta)&=\pi(X_N,Y,\theta)-\int p(Y|\theta,X_N)\pi(X_N,Y,\theta)dY,\end{aligned}\tag{88}$$

based on which we may develop a gradient based local search algorithm.

However, it suffers a problem of pre-specifying an appropriate learning stepsize. One alternative considers combining the roots of $\nabla_\varphi\pi(X_N,Y,\theta)=0$ at individual values of Y to approximate the root of $\nabla_\varphi H(\theta)=0$. One example is given by Eq. (11) in Xu (2010a) for learning Gaussian mixture, that is, letting $p^{new}_{\ell|x_t}$ in Algorithm 3 to be replaced by

$$p^{new}_{\ell|x_t}=p_{\ell t}(\theta^{new})[\,1+\delta_t^{(i)}(\theta^{new})\,].\tag{89}$$

Similarly, one other example can be found in Algorithm 2 and Eq. (10a) in Xu (2009) for learning radial basis functions (RBF) and extensions.

Still, this type of implementation may cause learning instability because the resulted $p^{new}_{\ell|x_t}$ may break the constraint $0\le p^{new}_{\ell|x_t}\le 1$.

The above observation motivates another *preservation principle of multiple convex combinations*. We consider an estimator via making $\max_\theta f(\theta)$, $f(\theta)=\sum_t a_t f_t(\theta)$, s.t. $\sum_t a_t=1, a_t\ge 0$, where each individual $f_t(\theta)$ possesses more than one of the natures $\xi^{(j)}(f_t), j=1,\ldots,c,\ with, c\ge 1$. The problem can be further modified into the following one:

$$\begin{gathered}\max_\theta f(\theta),\ f(\theta)=\textstyle\sum_t a_t f_t(\theta),\\ \text{subject to not only } \textstyle\sum_t a_t=1, a_t\ge 0, \text{but also}\\ \text{each corresponding nature } \xi^{(j)}(f) \text{ is a convex combination } \textstyle\sum_t b_t^{(j)}\xi^{(j)}(f_t),\end{gathered}\tag{90}$$

where the weights $\sum_t b_t^{(j)}=1, b_t^{(j)}\ge 0$ may be different for a different j and also may not be necessarily same as the weights $\sum_t a_t=1, a_t\ge 0$.

As an example, we modify Equation 9 to explicitly satisfy the principle of preserving one MCC nature as follows:

$$\begin{aligned} & \max_\theta H(\theta),\ H(\theta) = \int p(Y|\theta, X_N)\pi(X_N, Y, \theta)dY, \\ \text{subject to} \quad & p(Y|\theta, X_N) = q(Y|\theta, X_N),\ \nabla_\varphi H(\theta) = \int p_Y \nabla_\varphi \pi(X_N, Y, \theta)dY, \\ & p_Y \in \mathcal{C}_p = \{p_Y : 0 \le p_Y \le 1, \int p_Y dY = 1\}, \end{aligned} \tag{91}$$

where $\varphi \subseteq \theta$ is a subset of parameters to be estimated in our consideration. It can be the entire set of θ or a part of θ. Under this setting, we get the root of $\nabla_\varphi H(\theta) = 0$ by a convex combination of the roots of $\nabla_\varphi \pi(X_N, Y, \theta) = 0$.

Actually, Algorithm 1, Algorithm 3, Algorithm 5, Algorithm 6 and Algorithm 8, as well as their corresponding EM algorithms, are all the examples that pursuit along this direction. The Yang step or the E step actually gets such a $p_Y \in \mathcal{C}_p$ while the Ying step or the M step estimates the root of $\nabla_\varphi H(\theta) = 0$ by a convex combination of the roots of $\nabla_\varphi \pi(X_N, Y, \theta) = 0$.

Comparing $\nabla_\varphi H(\theta)$ in Equation 88 and $\nabla_\varphi H(\theta)$ in Equation 91, we get an alternative implementation that consists of two steps as follows:

(1) Get $\mathcal{P}_\delta$ that consists of $p_\delta(Y|\theta, X_N)$ by Equation 88 at all the possible values of Y.
(2) Project the set $\mathcal{P}_\delta$ to the convex set $\mathcal{C}_p$ under a nearest principle.

There are two key issues to be handled as follows:

- One is to be the nearest in what a sense? in a square or L_1 distance?
- The other is an effective algorithm to find such a projection.

Another important issue is a theoretical guarantee on whether $H(\theta)$ keeps increasing or nondecreasing such that learning convergence is guaranteed.

Results and discussion

The results of BYY harmony learning implementations are summarized in Tables 3 & 4 for those made before 2010 and in Table 1 for those made after 2010. Most of the fundamentals and major implementing techniques of the BYY harmony learning are developed in the period of 1995 to 2001, for which we provide an outline chronologically in Table 3 featured by the time points at which the major innovative studies started at. In particular, the threads of 1995(c), 1997(a), 1999(a) and 2000(a) reach the present formulation $H(p||q)$ in Equation 9 though these were considered on merely $R = \{Y\}$ in $\int [\cdot]\, dR$. Subsequent developments in the next decade are then further outlined in Table 4. Also, further details are referred to the following recent overviews:

- *Theoretical aspects and relations to other methods* see Sect.4.1, Appendix A and B in Xu (2010a), and Sects.4.1 and 4.2 in Xu (2012a).
- *Algorithms and applications* see the roadmaps in Figure three and Figure eleven of Xu (2010a), also in Figure one of Xu (2011) and Sect.5 of Xu (2012a), plus recent applications in (Pang et al. 2013; Shi et al. 2011a,b,c, 2014; Tu and Xu 2011a; Tu et al. 2011, 2012a,b; Tu and Xu 2014; Wang et al. 2011).
- *Outlines on major topics in Xu (2012a)* see Sect.7 for 3 topics on statistical learning in general, 8 topics on BYY system, 13 topics on best harmony learning and 4 topics on implementation, as well as 15 topics on exemplar learning tasks and algorithms.

Table 3 A foundation period of BYY studies (1995 to 2001)

Year	Outcomes	
1995		The following fundamental points of BYY harmony learning were firstly proposed in Xu (1995):
	(a)	The BYY system is proposed as a unified perspective for statistical learning.
	(b)	Under the name of BKYY learning, the Ying-Yang best matching by the minimisation of $KL(p(Y\|X)p(X)\|q(Y\|R)q(Y))$ has been proposed for learning parameters θ.
	(c)	One simplified version of $H(\theta)$ is proposed to get a hard-cut version of EM algorithm, see its Eqs. (19) and (20) and a criterion for selecting the number of components in Gaussian mixture (i.e. the cluster number), see Eqs. (22) and (24) in Xu (1995).
	(d)	One preliminary version of the BYY harmony learning based automatic model selection was presented, see its Sect. 5.2.
	(e)	The relationship $H(p\|\|q) = H_{R\|X} - KL(p\|\|q)$ by Equation 10 was also firstly identified, see Eqs. (8), (11) and (12) in Xu (1995).
1996		Points (c)(d) were verified experimentally in Xu (1996).
1997		Four progresses are made as follows:
	(a)	Beyond 1995(d), suggested $H(\theta)$ in a general expression as model selection criterion, see Eq. (12) in Xu (1997a) and Eq. (3.8) in Xu (1997b). Also, addressed its special cases on Gaussian mixture.
	(b)	Proposed to use $p_h^N(X)$ by Equation 8 and learn h for regularisation, see Eq. (3.10) in Xu (1997b). A smoothed EM is proposed for Gaussian mixture, see Eq. (18) in Xu (1997c).
	(c)	Proposed semi-supervised EM algorithm for Gaussian mixture, see Eq.(7.14) in XU (1997b).
	(d)	Extended BKYY to BCYY by replacing Kullback divergence with its convex counterpart, see Sect.5 in Xu (1997a) and Eqs.(19)-(23) in Xu (1997c).
1998		The following progresses are made:
	(a)	Proposed equation (A) in Table 2 as a criterion for model complexity, e.g. see Eq. (49) in Xu (1998a) and Eq. (22) in Xu (1998b).
	(b)	As an exemplar of 1997(a), derived model selection criteria for three-layer net and RBF net (see Eq. (56) and Eqs (61)-(64) in Xu (1998a)) and also for FA (see Eqs. (37)(43) in Xu (1998b)).
	(c)	Beyond 1995(c), developed adaptive EM algorithms for learning RBF net (see Sect.3.2) and FA (see Sect.4.2.4) in Xu (1998b) and Sect.3.2 in Xu (1998c).
1999		Further efforts are made, among which major ones are as follows:
	(a)	Beyond 1997(a), proposed a general form for parameter learning and model selection, see Sect.2 in Xu (1999b), Sect.2.2 in Xu (1999a), and Sect.2.2 in Xu (1999c).
	(b)	Beyond 1997(b), systematically studied data smoothing regularisation in Xu (1999d), with an approximation technique in Equation 18 and estimating techniques for h.
	(c)	Proposed Taylor expansion approximation by Equation 18 to remove the integral in BYY implementation, see Eq. (90) and Eq. (91) in Xu (1999e), later in the journal papers (Xu 2000c, 2001b).
2000		In Xu (2000d,2000a), $H(\theta)$ based harmony learning has been elaborated into its present formulation, supported by mathematical analysis on Ying-Yang best harmony versus Ying-Yang best matching, and featured with three innovative points:
	(a)	Beyond 1999(a), proposed a general form of $\max_\theta H(\theta)$ with automatic model selection, see Eq. (29) in Xu (2000d) and Sect.4 in Xu (2000a).
	(b)	Proposed Eq (23) in Xu (2000a) to implement equation (A) in Table 2 by learning θ with automatic model selection.
	(c)	Also proposed normalisation regularisation in parallel to data smoothing regularisation in the above 1998(b), see Sect. 2 and Sect.3 in Xu (2000a) and Eq. (21) in Xu (2000d).
2001		Further progresses are made as follows:
	(a)	Used $p(Y\|\theta, X_N) = q(Y\|\theta, X_N)$ in Equation 12 to get Yang structure for $\max_\theta H(\theta)$, see Eq. (40) in Xu (2001a), Eqs. (24)(27) in Xu (2001c).
	(b)	Developed a BYY harmony learning algorithm for Kernel regression and support vectors, see Sec.4.5 and Table seven in Xu (2001a).
	(c)	Understood $H(\theta)$ in its general form from an information transferring aspect via three layer encoding, see Sect.4.3 in Xu (2001c).
	(d)	Beyond 1998(b), derived model selection criteria for local PCA, see Eq. (23) in Xu (2001c), and local ICA, see Eq. (33) in Xu (2001d).

Table 4 Further advances of BYY studies (2002 to 2013)

Year		BYY harmony learning formulation
2004	(a)	$H(p\|\|q)$ in Equation 9 with $R = \{Y, \theta\}$ is proposed in Sect.II(B) of Xu (2004b), not only integrating the thread of data smoothing regularisation via 1997(b) and 1999(b) and of normalisation regularisation via 2000(c) into a specific formulation of a priori; but also covering a usual priori $q(\theta)$ as a component.
	(b)	Subsequent elaborations are referred to Sect.3.4 in Xu (2007a), Sect.3.4 in Xu (2007b), Sect.2 in Xu (2008), Eq. (8) in Xu (2009), especially to Sect.4 in Xu (2010a) and Sect.4 in Xu (2012a) for recent surveys.
2007		Beyond 2001(a), efforts on designing the structure of $p(R\|X)$ based on $q(X\|R)$ and $q(R)$ progress from the early concept of bi-directional architecture further towards.
	(a)	Either a preservation principle $p(Y\|\theta, X_N) = q(Y\|\theta, X_N)$, e.g. by Eq. (40) in Xu (2001a), Eq. (24), and Eq. (27) in Xu (2001c);
	(b)	Or that $p(Y\|\theta, X_N)$ preserves certain statistics of $q(Y\|\theta, X_N)$, e.g. equal covariance by Eqs. (72)(73) in Xu (2007a), which are elaborated under the name of uncertainty conversation or variety preservation between Ying and Yang, see pp69-72 in Xu (2009), with details referred to Sect.4.2 in Xu (2010a) and Sect.3.2.2 in Xu (2012a).
2008		Learning tasks are summarised into three levels of inverse problems and integrated into a unified representation of BYY system, see Xu (2008, 2009), and an introduction in Sect.1 of Xu (2010a).
	(a)	Radon-Nikodym derivative based formulation of Ying-Yang harmony information was proposed, with degenerated cases covering Shannon information and Kullback Leibler information. Details are referred to Sect.4.1 in Xu (2010a) and an overview in Figure five of Xu (2012a).
	(b)	Hierarchical temporal BYY harmony learning was developed in Sect. 5 of Xu (2010a), see Figures twelve and fourteen in Xu (2010a) and Figure eleven in Xu (2012a).
	(c)	BYY system provides an all-in-one formulation for unsupervised, supervised and semi- supervised learning , see Sect.4.4 in Xu (2010a) and Table two in Xu (2012a).
2011		Co-dim matrix pair formulation and a hierarchy of co-dim matrix pairs for BYY harmony learning have been proposed, with details referred to Sect.2.2, Sect.4 and Figure three in Xu (2011). Its special cases cover not only several typical learning models but also de-noised Gaussian mixture (see Algorithm 13), manifold learning as previously discussed about Equation 80, and the dual formulation as previously introduced in Equations 55 and 56.
Type		**BYY system design**
3-A		Started from the very beginning in 1995 Xu (1995), BYY system was classified into three architectures (3-A), i.e. forward architecture with $q(X\|R)$ in a free structure, backward architecture with $p(R\|X)$ in a free structure, and bi-directional architecture with both $q(X\|R)$ and $p(R\|X)$ in parametric structures, rather thoroughly examined before the mid of 2000th (Xu 2000c, 2001a,e, 2002, 2003a, 2004a,c).
3-P		Focuses are turned to three principles (3-P) for designing the structures of each component in a BYY system, i.e. the principle of least redundancy for $q(Y)$, the principle of divid-and-conquer for $q(X\|R)$, and the principle of uncertainty conversation or variety preservation for $p(Y\|X)$, as stated above by the item 2007(a) and (b). An overview is referred to Figure three in Xu (2012a).

Readers are also referred to Sect.3.2 and Sect.3.4 on topics and demanding issues about BYY system design, to Sect.4.2.3 on novelty and features of best harmony theory.

Before closing this paper, we continue the previous discussion made on Figure 4. As illustrated in Figure 5 and also referred to Appendix B(2) of Xu (2010a), learning is featured by a dynamic process of implementing learning theory to learn from what it observes and to adapt its environment, which may also be understood from a famous ancient Chinese TCM WuXing theory. A learning process is featured by repeatedly circling of five actions or states. For each circling, the first action A-1 gathers samples and information as the system's input; A-2 transfers the input into inner candidate assumptions or suggestions; A-3 integrates or regulates evidences about candidates that comes from A-2; A-4 selects good candidates or trims off bad ones; and A-5 interprets or manages the environment.

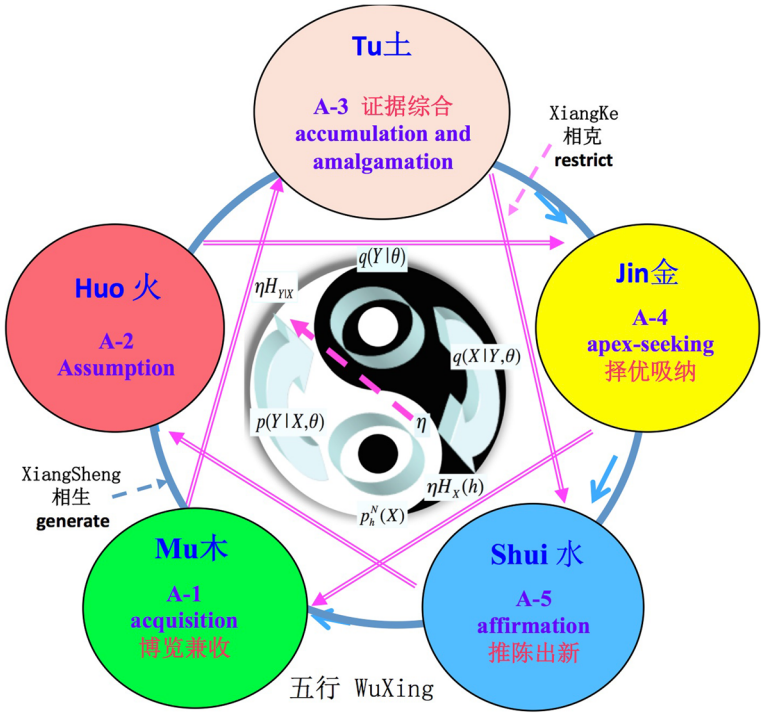

Figure 5 Negative feedback stabilises dynamics.

The harmonising dynamics discussed previously in Figure 3 and the corresponding subsection may also be observed from this perspective. At the centre of Figure 5, the bottom of the Yin-Yang logo has a black centre, which is usually called fish eye. This indicates the output of A-5, while its surrounding white ring indicates the Yang domain. The starting part of the Yang arrow indicates A-1 for picking samples in the Yang domain to get $p_h^N(X)$, and the arrow ends at the white fish eye on the top, implementing A-2 by $p(Y,\theta|X)$. On the other hand, the surrounding black ring of the white fish eye indicates the Ying domain that collects all the candidates as well as the associated evidences. The starting part of the Ying arrow indicates A-4 for choosing good candidates probabilistically via $q(X|Y,\theta)$, and the arrow ends at the bottom black fish eye and completes one circling.

As addressed by Equation 27 and the discussions thereafter, the signal η is measured at two fish eyes and also modulated by the inner attention of the system. A small η reflects either a bad Ying-Yang mutual agreement (a big mismatch to the desire) in the top fish eye or a bad fitting in the bottom fish eye.

A poor performance incurred from a poor selection of Y at A-4, resulting in a small value η that is feedback to A-2 to harmonise the attempts of updating θ. In such a negative feedback mechanism, the dynamics of information harmonising is stabilised. Interestingly, such a mechanism is executed in a pattern 'A-2/Huo modulates A-4/Jin', which complies with the classic 'XiangKe' principle of the Chinese TCM WuXing theory. In other word, the 'XiangKe' principle can be regarded as an ancient negative feedback principle.

Conclusions

Based on Lagrange variety preservation of Yang structure, this paper proposes a generic framework of dynamic BYY harmony learning, which not only unifies attention, detection, problem-solving, adaptation, learning and model selection from an information harmonising perspective but also provides a new type of Ying-Yang alternative nonlocal search to overcome a dilemma of suboptimal solution versus learning instability typically suffered by the existing Ying-Yang alternative nonlocal search. Algorithms are developed for learning Gaussian mixture, factor analysis (FA), mixture of local FA, binary FA, nonGaussian FA, de-noised Gaussian mixture, sparse multivariate regression, temporal FA and temporal binary FA, as well as a generalised bilinear matrix system that covers not only these linear models but also manifold learning, gene regulatory networks and the generalised linear mixed model. These algorithms are featured with not only a favourable nature of automatic model selection but also a unified formulation in performing unsupervised learning and semi-supervised learning. Moreover, a principle of preserving multiple convex combinations is also proposed to improve the BYY harmony learning, which leads another type of Ying-Yang alternative nonlocal search.

Competing interests

The author declares no competing interests.

Acknowledgements

This work was supported by a CUHK Direct grant project 4055025 and a starting-up for the Zhi-Yuan chair professorship by Shanghai Jiao Tong University.

Received: 18 July 2014 Accepted: 21 April 2015
Published online: 13 June 2015

References

Akaike H (1974) A new look at the statistical model identification. Automatic Control IEEE Trans 19(6):716–723
Akaike H (1987) Factor analysis and aic. Psychometrika 52(3):317–332
Barron A, Rissanen J, Yu B (1998) The minimum description length principle in coding and modeling. Inf Theory IEEE Trans 44(6):2743–2760
Bartels RH, Stewart G (1972) Solution of the matrix equation ax+ xb= c. Commun ACM 15(9):820–826
Bar-Joseph Z, Gitter A, Simon I (2012) Studying and modelling dynamic biological processes using time-series gene expression data. Nature Rev Genet 13(8):552–564
Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396
Chen G, Heng P-A, Xu L (2014) Projection-embedded byy learning algorithm for gaussian mixture-based clustering. Appl Inf 1(2):1–20
Corduneanu A, Bishop CM (2001) Variational bayesian model selection for mixture distributions. In: Artificial Intelligence and Statistics. Morgan Kaufmann Waltham, MA Vol. 2001. pp 27–34
Dayan P, Hinton GE, Neal RM, Zemel RS (1995) The helmholtz machine. Neural Comput 7(5):889–904
Dempster AP, Laird NM, Rubin DB, et al. (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc 39(1):1–38
Demidenko E (2013) Mixed Models: Theory and Applications with R. John Wiley & Sons, Hoboken, New Jersey
Diaconis P, Ylvisaker D, et al. (1979) Conjugate priors for exponential families. Ann Stat 7(2):269–281
Dutilleul P (1999) The mle algorithm for the matrix normal distribution. J Stat Comput Simul 64(2):105–123
Fang S-C, Rajasekera JR, Tsao H-SJ (1997) Entropy Optimization and Mathematical Programming, Vol. 8. Springer, New York
Figueiredo MAF, Jain AK (2002) Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell 24:381–396
Floudas CA, Visweswaran V (1995) Quadratic optimization. In: Handbook of Global Optimization. Springer, New York. pp 217–269
Gupta AK, Nagar DK (1999) Matrix Variate Distributions, Vol. 104. CRC Press, Chapman & Hall, Boca Raton, Florida
Hoerl RW (1985) Ridge analysis 25 years later. Am Stat 39(3):186–192
Jeffreys H (1946) An invariant form for the prior probability in estimation problems. Proc R Soc Lond. Series A. Math Phys Sci 186(1007):453–461
Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Mach Learn 37(2):183–233
Karlebach G, Shamir R (2008) Modelling and analysis of gene regulatory networks. Nat Rev Mol Cell Biol 9(10):770–780
Liao JC, Boscolo R, Yang Y-L, Tran LM, Sabatti C, Roychowdhury VP (2003) Network component analysis: reconstruction of regulatory signals in biological systems. Proc Natl Acad Sci 100(26):15522–15527

McGrory CA, Titterington DM (2007) Variational approximations in bayesian model selection for finite mixture distributions. Comput Stat Data Anal 51:5352–5367
Miyajima S (2013) Fast enclosure for solutions of sylvester equations. Linear Algebra Appl 439(4):856–878
Morris KV, Mattick JS (2014) The rise of regulatory rna. Nature Rev Genet 15(6):423–437
Ntzoufras I, Tarantola C (2013) Conjugate and conditional conjugate bayesian analysis of discrete graphical models of marginal independence. Comput Stat Data Anal 66:161–177
Pang Z, Tu S, Wu X, Xu L (2013) Discriminative gmm-hmm acoustic model selection using two-level bayesian ying yang harmony learning. In: Intelligent Science and Intelligent Data Engineering. Springer, Berlin Heidelberg. pp 719–726
Redner RA, Walker HF (1984) Mixture densities, maximum likelihood and the em algorithm. SIAM Rev 26(2):195–239
Rissanen J (1978) Modeling by shortest data description. Automatica 14(5):465–471
Rubin DB, Thayer DT (1982) Em algorithms for ml factor analysis. Psychometrika 47(1):69–76
Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464
Shi L, Tu S, Xu L (2011a) Learning gaussian mixture with automatic model selection: A comparative study on three bayesian related approaches. Front Electrical Electronic Eng China 6(2):215–244
Shi L, Tu SK, Xu L (2011b) Learning gaussian mixture with automatic model selection: a comparative study on three bayesian related approaches. Front Electr Electron Eng China 6:215–244. A special issue on Machine Learning and Intelligence Science: IScIDE2010 (B)
Shi L, Wang P, Liu H, Xu L, Bao Z (2011c) Radar hrrp statistical recognition with local factor analysis by automatic bayesian ying-yang harmony learning. Signal Process IEEE Trans 59(2):610–617
Shi L, Liu Z-Y, Tu S, Xu L (2014) Learning local factor analysis versus mixture of factor analyzers with automatic model selection. Neurocomputing 139:3–14
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc B 58:267–288
Tikhonov A, Goncharsky A, Stepanov V, Yagola A (1995) Numerical methods for the solution of ill-posed problems. Kluwer Academic, Netherlands
Tipping ME, Bishop CM (1999) Probabilistic principal component analysis. J R Stat Soc: Series B (Statistical Methodology) 61(3):611–622
Tu SK, Xu L (2011a) Parameterizations make different model selections : empirical findings from factor analysis. Front Electr Electron Eng China 6:256–274. A special issue on Machine Learning and Intelligence Science: IScIDE2010 (B)
Tu S, Xu L (2011b) An investigation of several typical model selection criteria for detecting the number of signals. Front Electr Electron Eng China 6(2):245–255
Tu SK, Chen RS, Xu L (2011) A binary matrix factorization algorithm for protein complex prediction. Proteome Sci 9(Suppl 1):18
Tu S, Chen R, Xu L (2012a) Transcription network analysis by a sparse binary factor analysis algorithm. J Integrative Bioinformatics 9(2):198
Tu S, Luo D, Chen R, Xu L (2012b) A non-gaussian factor analysis approach to transcription network component analysis. In: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium On. IEEE. pp 404–411
Tu S, Xu L (2014) Learning binary factor analysis with automatic model selection. Neurocomputing 134:149–158
Wallace CS, Dowe DL (1999) Minimum message length and kolmogorov complexity. Comput J 42(4):270–283
Wang P, Shi L, Du L, Liu H, Xu L, Bao Z (2011) Radar hrrp statistical recognition with temporal factor analysis by automatic bayesian ying-yang harmony learning. Front Electr Electron Eng China 6(2):300–317
Xu L, Krzyzak A, Oja E (1992) Unsupervised and supervised classifications by rival penalized competitive learning. In: Pattern Recognit, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference On. IEEE, New Jersey. pp 496–499
Xu L, Krzyzak A, Oja E (1993) Rival penalized competitive learning for clustering analysis, rbf net, and curve detection. Neural Netw IEEE Trans 4(4):636–649
Xu L (1995) Bayesian-kullback coupled ying-yang machines: Unified learnings and new results on vector quantization. In: Proc. Int. Conf. Neural Information Process (ICONIP '95). Publishing House of Electronics Industry, Beijing. pp 977–988
Xu L (1996) How many clusters?: A ying-yang machine based theory for a classical open problem in pattern recognition. In: Neural Netw, 1996., IEEE International Conference On. IEEE, New Jersey Vol. 3. pp 1546–1551
Xu L, Jordan MI (1996) On convergence properties of the em algorithm for gaussian mixtures. Neural Comput 8(1):129–151
Xu L (1997a) Bayesian ying–yang machine, clustering and number of clusters. Pattern Recognit Lett 18(11):1167–1178
Xu L (1997b) Bayesian ying yang system and theory as a unified statistical learning approach:(i) unsupervised and semi-unsupervised learning. In: Brain-like Computing and Intelligent Information Systems. Springer-Verlag, Berlin Heidelberg. pp 241–274
Xu L (1997c) Bayesian ying yang system and theory as a unified statistical learning approach (ii): from unsupervised learning to supervised learning and temporal modeling. In: Proceedings of Theoretical Aspects of Neural Computation: A Multidisciplinary Perspective. Springer, Berlin Heidelberg. pp 25–42
Xu L (1998a) Rbf nets, mixture experts, and bayesian ying–yang learning. Neurocomputing 19(1-3):223–257
Xu L (1998b) Bayesian kullback ying–yang dependence reduction theory. Neurocomputing 22(1):81–111
Xu L (1998c) Bayesian ying-yang dimension reduction and determination. J Comput Intell Finance 6(5):11–16
Xu L (1998d) Bkyy dimension reduction and determination. In: Neural Netw Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference On. IEEE, New Jersey Vol. 3. pp 1822–1827
Xu L (1999a) Temporal byy learning and its applications to extended kalman filtering, hidden markov model, and sensor-motor integration. In: Neural Netw, 1999. IJCNN'99. International Joint Conference On. IEEE, New Jersey Vol. 2. pp 949–954
Xu L (1999b) Bayesian ying yang theory for empirical learning, regularisation and model selection: general formulation. In: Neural Netw, 1999. IJCNN'99. International Joint Conference On. IEEE, New Jersey Vol. 1. pp 552–557
Xu L (1999c) Bayesian ying yang supervised learning, modular models, and three layer nets. In: Neural Netw, 1999. IJCNN'99. International Joint Conference On. IEEE, New Jersey Vol. 1. pp 540–545

Xu L (1999d) Byy data smoothing based learning on a small size of samples. In: Neural Netw, 1999. IJCNN'99. International Joint Conference On. IEEE, New Jersey Vol. 1. pp 546–551
Xu L (1999e) Byy ying yang unsupervised and supervised learning: theory and applications. In: Neural Netw and Signal Processing, Proceedings of 1999 Chinese Conference On. Publishing house of Electronic industry, Beijing. pp 112–29
Xu L (2000a) Byy prod-sum factor systems and harmony learning. invited talk. In: Proceedings of International Conference on Neural Information Processing (ICONIP'2000), KAIST, Taejon Vol. 1. pp 548–558
Xu L (2000b) Temporal byy learning for state space approach, hidden markov model, and blind source separation. Signal Process IEEE Trans 48(7):2132–2144
Xu L (2000c) Byy learning system and theory for parameter estimation, data smoothing based regularisation and model selection. Neural Parallel Sci Comput 8(1):55–83
Xu L (2000d) Best harmony learning. In: Intelligent Data Engineering and Automated Learning (IDEAL 2000). Data Mining, Financial Engineering, and Intelligent Agents. Springer, Berlin Heidelberg. pp 116–125
Xu L (2001a) Best harmony, unified rpcl and automated model selection for unsupervised and supervised learning on gaussian mixtures, three-layer nets and me-rbf-svm models. Int J Neural Syst 11(01):43–69
Xu L (2001b) Byy harmony learning, independent state space, and generalised apt financial analyses. Neural Netw IEEE Trans 12(4):822–849
Xu L (2001c) Byy harmony learning, model selection, and information approach: Further results. In: Neural Information Processing (ICONIP'2001), 2001. Proceedings International Joint Conference On. APPNA, Shanghai Vol. I. pp 30–37
Xu L (2001d) Byy harmony learning, local independent analyses, and apt financial applications. In: Neural Netw, 2001. Proceedings. IJCNN'01. International Joint Conference On. IEEE, New Jersey Vol. 3. pp 1817–1822
Xu L (2001e) An overview on unsupervised learning from data mining perspective. In: Advances in Self-Organising Maps. Springer, Berlin Heidelberg. pp 181–209
Xu L (2002) Byy harmony neural networks, structural rpcl, and topological self-organizing on mixture models. Neural Netw 15:1125–1151
Xu L (2003a) Independent component analysis and extensions with noise and time: a bayesian ying-yang learning perspective. Neural Inf Process Lett Rev 1:1–52
Xu L (2003b) Data smoothing regularization, multi-sets-learning, and problem solving strategies. Neural Netw 16:817–825
Xu L (2004a) Temporal byy encoding, markovian state spaces, and space dimension determination. Neural Netw IEEE Trans 15(5):1276–1295
Xu L (2004b) Advances on byy harmony learning: information theoretic perspective, generalized projection geometry, and independent factor autodetermination. Neural Netw IEEE Trans 15(4):885–902
Xu L (2004c) Bi-directional byy learning for mining structures with projected polyhedra and topological map. In: Proceedings of IEEE ICDM2004 Workshop on Foundations of Data Mining. ICDM, Brighton. pp 2–14
Xu L (2007a) A unified perspective and new results on rht computing, mixture based learning, and multi-learner based problem solving. Pattern Recognit 40:2129–2153
Xu L (2007b) A trend on regularization and model selection in statistical learning: A bayesian ying yang learning perspective. In: Challenges for Computational Intelligence. Springer, Berlin Heidelberg. pp 365–406
Xu L (2008) Bayesian ying yang system, best harmony learning, and gaussian manifold based family. In: Computational Intelligence: Research Frontiers. Springer, Berlin Heidelberg. pp 48–78
Xu L (2009) Learning algorithms for rbf functions and subspace based functions. In: E S Olivas e.a. (ed) Handbook of Research on Machine Learning, Applications and Trends: Algorithms, Methods and Techniques. IGI Global, Hershey, PA. pp 60–94
Xu L (2010a) Bayesian ying-yang system, best harmony learning, and five action circling. Front Electr Electron Eng China 5:281–328. A special issue on Emerging Themes on Information Theory and Bayesian Approach
Xu L (2010b) Machine learning problems from optimization perspective. J Global Optimization 47(3):369–401
Xu L (2011) Codimensional matrix pairing perspective of byy harmony learning: hierarchy of bilinear systems, joint decomposition of data-covariance, and applications of network biology. Front Electr Electron Eng China 6:86–119. A special issue on Machine Learning and Intelligence Science: IScIDE2010 (A)
Xu L (2012a) On essential topics of byy harmony learning: current status, challenging issues, and gene analysis applications. Front Electr Electron Eng China 7:147–196
Xu L (2012b) Semi-blind bilinear matrix system, byy harmony learning, and gene analysis applications. In: Proceedings of The 6th International Conference on New Trends in Information Science, Service Science and Data Mining. AICIT, Taipei. pp 661–666
Yang J, Zaitlen NA, Goddard ME, Visscher PM, Price AL (2014) Advantages and pitfalls in the application of mixed-model association methods. Nat Genet 46(2):100–106
Zhou X, Stephens M (2014) Efficient multivariate linear mixed model algorithms for genome-wide association studies. Nat Methods 11(4):407–409
Zou J, Lippert C, Heckerman D, Aryee M, Listgarten J (2014) Epigenome-wide association studies without the need for cell-type composition. Nat Methods 11(3):309–311