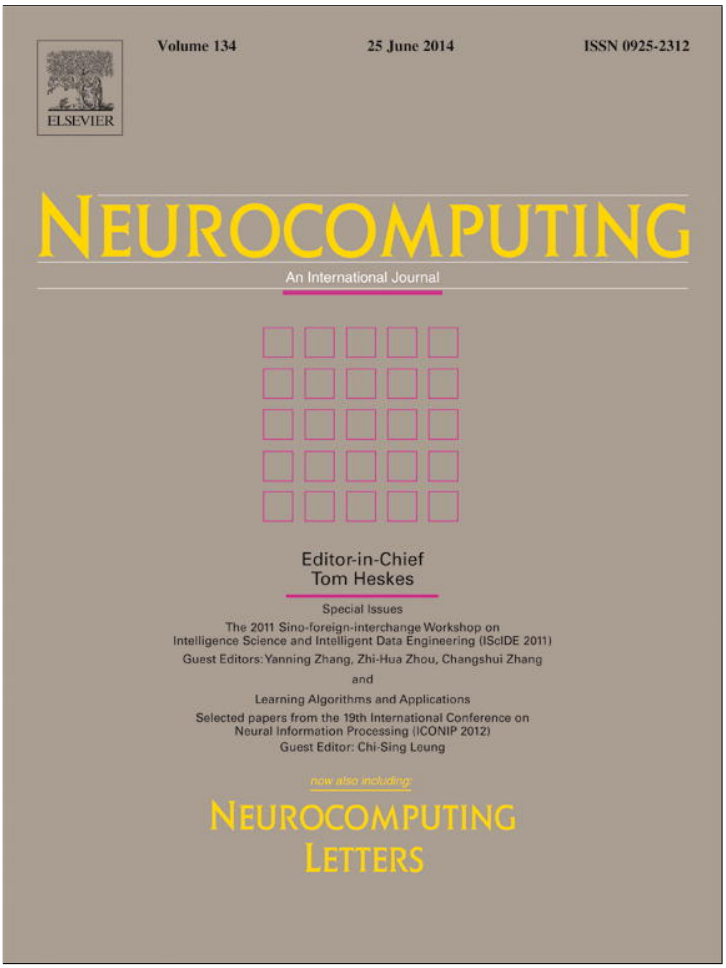


Neurocomputing 134 (2014) 149–158

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning binary factor analysis with automatic model selection

Shikui Tu, Lei Xu *

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

ARTICLE INFO

Article history:
Received 4 June 2012
Received in revised form
12 December 2012
Accepted 31 December 2012
Available online 6 February 2014

Keywords:
Automatic model selection
Binary Factor Analysis
Variational Bayes
Bayesian Ying-Yang

ABSTRACT

Binary Factor Analysis (BFA) uncovers the independent binary information sources from observations with wide applications. BFA learning hierarchically nests three levels of inverse problems, i.e., inference of binary code for each observation, parameter estimation and model selection. Under Bayesian Ying-Yang (BYY) framework, the first level becomes an intractable Binary Quadratic Programming (BQP) problem, while model selection can be conducted automatically during parameter learning. We conduct extensive experiments to reveal that the performance order of four BQP methods is reversed from making BQP optimization to making BYY automatic model selection, which implies that learning is not merely optimization. Moreover, the BFA learning algorithm is further developed with priors over parameters to improve the performance. Finally, based on BFA, we empirically compare BYY with Variational Bayes (VB) and Bayesian information criterion (BIC).

1. Introduction

Binary Factor Analysis (BFA) explores latent binary structures of data. Unlike the conventional factor analysis where the latent factor is assumed to be Gaussian, BFA traces the observation to independent Bernoulli information sources. Research on BFA has been focused on analysis of binary data, such as social research questionnaires and market basket data, with the aid of Boolean algebra [1], and also on the discovery of binary factors in continuous data, [2–4], taking advantage of the representational capacity of the underlying binary structure. When considering all the random variables to be binary, factor analysis becomes the restricted Boltzmann machine which is the building block of the deep belief network [5]. This paper considers the same BFA model as in [4,2], under Bayesian Ying-Yang (BYY) harmony learning [6,7], in a comparison with Variation Bayes (VB) [8] and Bayesian information criterion (BIC) [9]. Rissanen's Minimum Description Length (MDL) stems from another viewpoint but coincides with BIC when it is simplified to a simple computable criterion [10].

The hierarchy of all unknowns in a learning system makes the learning process not just an optimization but a series of hierarchically nested continuous or discrete optimizations. As summarized in [7], there are three levels of inverse problems, i.e., inverse inference from observation to inner representation, parameter learning, and model selection. In terms of BFA, the first level of inverse problems in BFA is the inference of an m-bit inner binary code $\boldsymbol{y}(\boldsymbol{x})$ or a 2^m-point posterior distribution $p(\boldsymbol{y}|\boldsymbol{x})$ for each observation $\boldsymbol{x}$, given the parameters and the coding length of $\boldsymbol{y}$, i.e., $m=\dim(\boldsymbol{y})$. It is difficult due to its combinatorial complexity. Under BYY, maximizing the objective functional turns this problem into a Binary Quadratic Programming (BQP) problem that searches an optimal binary code $\boldsymbol{y}(\boldsymbol{x})$ for each training sample $\boldsymbol{x}$. A preliminary study in [11] compared four BQP methods and suggested that some amount of error in BQP optimization is not always a bad thing but instead provides a helpful regularization for the learning process. Conventionally, the second and the third level are implemented by a two-phase procedure, i.e., parameter learning (usually maximum likelihood learning) is conducted for each m in a candidate set $\mathcal{M}$, one of which is then selected by a model selection criterion, e.g., BIC [9]. However, this two-phase implementation suffers from a huge computation, because it requires parameter learning that is nested with a BQP for each $m\in\mathcal{M}$. Moreover, a larger m often implies more unknown parameters, and thus parameter estimation becomes less reliable so that the criterion evaluation reduces its accuracy, see Section 2.1 in [12] for a detailed discussion.

This paper further investigates the four BQP methods in [11] used for the BYY learning on BFA. One is the exact BQP solver by enumeration (shortly denoted as **enum**). The other three are approximate methods, i.e., the **greedy** method in [13], the **cdual** method derived from the canonical duality theory [14], and the **round** method by relaxing the binary $\boldsymbol{y}$ to a continuous one and rounding the optimal solution back to binary [15]. Their BQP optimization performances follow an order: **round** < **cdual** < **greedy** < **enum**. Extensive experiments show that **cdual** and **round** are fast and more effective in discarding extra factors, and lead to much better model selection

* Corresponding author.
E-mail address: lxu@cse.cuhk.edu.hk (L. Xu).

http://dx.doi.org/10.1016/j.neucom.2012.12.069
0925-2312

performances than **greedy** and **enum**. Actually, some amount of error in BQP provides a helpful learning regularization with a gain on both computational efficiency and model selection performance.

Moreover, automatic model selection is adopted to save the computation of two-phase implementation by starting from a large enough m and then discarding redundant binary factors during parameter learning. We further develop BFA learning algorithms by considering prior distributions over parameters, which play a role of Bayesian regularization. With the help of priors, **enum** and **greedy** improve their automatic model selection performances, but are still inferior to **cdual** and **round**.

Finally, we empirically investigate the performance between BYY, VB, and BIC. Such comparisons have been made on factor analysis in [16] and Gaussian mixture model in [17], but not on BFA yet. We simplify the VB-ICA algorithm [18,19] to obtain a VB algorithm on BFA. The results reveal that BYY is the best for most configurations, while BIC is more robust than VB. VB is good only when both training sample size N is large and noise is small, and declines drastically when N reduces and noise increases. Moreover, applied to the problem of blind binary image separation, the results again show that BYY outperforms VB.

The rest of this paper is organized as follows. BFA model is introduced in Section 2. BYY harmony learning is briefly reviewed in Section 3, and a BYY-BFA algorithm is derived with priors over the parameters. Section 4 introduces VB and BIC for an empirical analysis in Section 5, while concluding remarks are given in Section 6.

2. Binary Factor Analysis

In Binary Factor Analysis (BFA), an n-dimensional observed variable $\boldsymbol{x}$ is modeled as

$$\boldsymbol{x}=\boldsymbol{A}\boldsymbol{y}+\boldsymbol{c}+\boldsymbol{\varepsilon}, \tag{1}$$

where the hidden factor vector $\boldsymbol{y}\in\{-1,1\}^m$ is an internal binary code with each element being either -1 or 1 drawn from a Bernoulli distribution, and $\boldsymbol{y}$ is independent of the Gaussian noise $\boldsymbol{\varepsilon}$. This model has been studied previously from different perspectives [15,4,2].

The BFA can also be mathematically formalized by the following probabilistic distributions:

$$q(\boldsymbol{y}|\boldsymbol{\Theta})=\prod_{i=1}^{m}\beta_i^{(1+y_i)/2}(1-\beta_i)^{(1-y_i)/2},\quad q(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\Theta})=G(\boldsymbol{x}|\boldsymbol{A}\boldsymbol{y}+\boldsymbol{c},\boldsymbol{\Sigma}_e), \tag{2}$$

where $\boldsymbol{\beta}=[\beta_1,\ldots,\beta_m]$, $0<\beta_i<1$, $i=1,2,\ldots,m$, $\boldsymbol{\Sigma}_e$ is a positive definite diagonal matrix, and $G(\cdot|\boldsymbol{\mu},\boldsymbol{\Psi})$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Psi}$, and $\boldsymbol{\Theta}=\{\boldsymbol{A},\boldsymbol{\beta},\boldsymbol{c},\boldsymbol{\Sigma}_e\}$ is the set of parameters.

Similar to [20,18], we consider the joint prior distribution on the parameters $\boldsymbol{\Theta}=\{\boldsymbol{A},\boldsymbol{\Sigma}_e,\boldsymbol{\beta},\boldsymbol{c}\}$ to be a product of distributions on each parameter independently:

$$q(\boldsymbol{\Theta}|\boldsymbol{\Xi})=q(A)q(\boldsymbol{\beta})q(\boldsymbol{c})q(\boldsymbol{\Sigma}_e), \tag{3}$$

where $\boldsymbol{\Xi}$ is the set of hyperparameters. Each column $\mathbf{a}_i$ of A is independently distributed according to a Gaussian distribution with its covariance controlled by a precision parameter α_i which is further assumed to follow a Gamma distribution

$$q(A)=\prod_{i=1}^{m}G\left(\mathbf{a}_i|0,\frac{1}{\alpha_i}\mathbf{I}_n\right),\quad q(\alpha_i)=\Gamma(\alpha_i|a^{\alpha},b^{\alpha}), \tag{4}$$

where $\Gamma(x|a,b)=(b^a/\Gamma(a))x^{a-1}e^{-bx}$ denotes the Gamma density. A Dirichlet distribution is appropriate for each β_i which satisfies $\beta\in[0,1]$:

$$q(\boldsymbol{\beta})=\prod_{i=1}^{m}\mathcal{D}(\beta_i|\boldsymbol{\lambda}_i,\xi_i)=\prod_{i=1}^{m}\frac{\Gamma(\xi_i)\cdot\beta_i^{\xi_i\lambda_{i1}-1}(1-\beta_i)^{\xi_i\lambda_{i2}-1}}{\Gamma(\xi_i\beta_i)\Gamma(\xi_i(1-\beta_i))}. \tag{5}$$

Usually, $q(\boldsymbol{\mu})$ is assumed to be a Gaussian with zero mean, i.e., $G(\boldsymbol{\mu}|\boldsymbol{\mu}_0,\lambda_0^{\mu}\mathbf{I}_n)$. Moreover, the case of isotropic noise is considered, i.e., $\boldsymbol{\Sigma}_e=\varphi^{-1}\mathbf{I}_n$, and a Gamma distribution is imposed on the noise precision parameter φ:

$$q(\boldsymbol{\Sigma}_e)=q(\varphi)=\Gamma(\varphi|a^{\varphi},b^{\varphi}). \tag{6}$$

3. Bayesian Ying-Yang (BYY) harmony learning

Firstly proposed in [6] and systematically developed over a decade and half [12,21], the Bayesian Ying-Yang harmony learning theory is a unified statistical learning framework under a best harmony principle, which leads to a new family of algorithms that performs automatic model selection during parameter learning. The best harmony is mathematically to maximize the following general harmony functional [12,7]:

$$H(p\|q)=\int p(X)p(R|X)\ln[q(X|R)q(R)]dR\,dX \tag{7}$$

$$H(p\|q)=\int p(\boldsymbol{\Theta}|X)H(p\|q,\boldsymbol{\Theta})d\boldsymbol{\Theta}, \tag{8}$$

$$H(p\|q,\boldsymbol{\Theta})=\int p(Y|X,\boldsymbol{\Theta})p(X)\ln[q(X|Y,\boldsymbol{\Theta})q(Y|\boldsymbol{\Theta})]dY\,dX+\ln q(\boldsymbol{\Theta}|\boldsymbol{\Xi}), \tag{9}$$

where the observation X is regarded to be generated from its inner representation $R=\{Y,\boldsymbol{\Theta}\}$ with latent variable Y and parameters $\boldsymbol{\Theta}$. As interpreted in [7], maximizing $H(p\|q)$ forces $q(X|R)q(R)$ to match $p(R|X)p(X)$. Due to a finite sample size and practical constraints on $p(R|X)$, this matching aims at but may not really reach a perfect matching $p(R|X)p(X)=q(X|R)q(R)$. Still, we get a trend at this equality which turns $H(p\|q)$ into a negative entropy that describes the complexity of system, and thus further maximizing it leads to a least complexity. Hence, this matching is not in a maximum likelihood sense but with a promising model selection nature. Readers are referred to not only a summary of nine aspects on the novelty and favorable natures of BYY harmony learning, made at the end of Section 4.1 in [12], but also the roadmap shown in Fig. A2 in [12], as well as to a systematic outline on the 13 topics about best harmony learning in Section 7 in [21].

The model selection performance of not only BYY criterion but also BYY automatic model selection on BFA has been comparatively investigated in [4], in comparison with existing typical model selection criteria, including Bayesian Information Criterion (BIC) [9] etc., which are implemented in a two-phase procedure that first trains a set of candidate models and then selects the one with the minimum criterion value. This two-stage implementation suffers from a huge computation because it requires parameter learning for each candidate model scale. Moreover, a larger model scale often implies more unknown parameters, and thus parameter estimation becomes less reliable so that the criterion evaluation reduces its accuracy, see Section 2.1 in [12] for a detailed discussion. This paper focuses on BYY based automatic model selection, incorporated with appropriate prior distributions on parameters.

Specifically, we consider the BFA model by Eq. (2) with independently and identically distributed (i.i.d.) samples in $X_N=\{\boldsymbol{x}_t\}_{t=1}^N$, from which we have

$$q(X|Y,\boldsymbol{\Theta})=\prod_t q(\boldsymbol{x}_t|\boldsymbol{y}_t,\boldsymbol{\Theta}),\ q(Y|\boldsymbol{\Theta})=\prod_t q(\boldsymbol{y}_t|\boldsymbol{\Theta}), \tag{10}$$

where $q(\boldsymbol{x}_t|\boldsymbol{y}_t,\boldsymbol{\Theta})$ and $q(\boldsymbol{y}_t|\boldsymbol{\Theta})$ are given by Eq. (2). Moreover, we consider the empirical density $p(X)=\delta(X-X_N)$ and no constraints for both $p(Y|X,\boldsymbol{\Theta})$ and $p(\boldsymbol{\Theta}|X)$, then maximizing $H(p\|q)$ in Eq. (8) with respect to $p(Y|X,\boldsymbol{\Theta})$ and $p(\boldsymbol{\Theta}|X)$ becomes

$$\max_{\boldsymbol{\Theta}} H(p\|q), \quad H(p\|q)\approx \sum_{t=1}^{N} \ln[q(\boldsymbol{x}_t|\hat{\boldsymbol{y}}_t,\boldsymbol{\Theta})q(\hat{\boldsymbol{y}}_t|\boldsymbol{\Theta})]+\ln q(\boldsymbol{\Theta}|\Xi), \tag{11}$$

where $\hat{\boldsymbol{y}}_t$ is obtained through the following binary quadratic programming (BQP) problem:

$$\hat{\boldsymbol{y}}_t=\arg\max_{\boldsymbol{y}\in\{-1,1\}^m} \ln[q(\boldsymbol{x}_t|\hat{\boldsymbol{y}}_t,\boldsymbol{\Theta})q(\boldsymbol{y}_t|\boldsymbol{\Theta})]=\arg\min_{\boldsymbol{y}\in\{-1,1\}^m}\left\{\frac{1}{2}\boldsymbol{y}^T\boldsymbol{Q}_y\boldsymbol{y}-\boldsymbol{f}_y\boldsymbol{y}\right\}, \tag{12}$$

with

$$\boldsymbol{Q}_y=\boldsymbol{A}^T\boldsymbol{\Sigma}_e^{-1}\boldsymbol{A}, \quad \boldsymbol{f}_y=\tfrac{1}{2}[\ln\boldsymbol{\beta}-\ln(1-\boldsymbol{\beta})]+\boldsymbol{A}^T\boldsymbol{\Sigma}_e^{-1}(\boldsymbol{x}_t-\boldsymbol{c}), \tag{13}$$

and $\ln\boldsymbol{\beta}=[\ln\beta_1,\ldots,\ln\beta_m]$ and $\ln(1-\boldsymbol{\beta})=[\ln(1-\beta_1),\ldots,\ln(1-\beta_m)]$.

Iteratively implementing Eqs. (11) and (12) actually takes a specific form of Ying-Yang alternative procedure with convergence guaranteed [12]. The details of the obtained BYY-BFA algorithm are given in Algorithm 1. It should be noted that the effects of $q(\boldsymbol{\Theta})$ are shut-off when the indicator $\tau=0$, at which BYY-BFA degenerates back to the one in [11]. Moreover, Eq. (11) is merely a rough approximation, while more advanced treatments on $p(Y|X,\boldsymbol{\Theta})$ are possible, e.g., see Eq. (20) and Section 4.3 in [12].

Algorithm 1. A BYY harmony learning algorithm for BFA with automatic model selection (BYY-BFA).

Input : A set $X_N=\{\boldsymbol{x}_t\}_{t=1}^N\subset\Re^n$ of observations
Output: estimate of $\dim(\boldsymbol{y})$ and $\boldsymbol{\Theta}=\{\boldsymbol{\beta},\boldsymbol{A},\boldsymbol{c},\boldsymbol{\Sigma}_e(\varphi)\}$

Initialize $\dim(\boldsymbol{y})$ with a large integer m_{init};
$\boldsymbol{\Theta}_0=\{\boldsymbol{\beta}(\gamma_0),\boldsymbol{A}_0,\boldsymbol{c}_0,\boldsymbol{\Sigma}_0\}$;
repeat
  Encode X_N into $\{\hat{\boldsymbol{y}}(\boldsymbol{x}_t):\boldsymbol{x}_t\in X_N\}$ with a binary encoder ;
  Update $\boldsymbol{\Theta}$ along the gradient flow of $H(p\|q)$ by Eq.(8) ;
  $\boldsymbol{\beta}\leftarrow 1/[1+e^{-\boldsymbol{\gamma}}], \quad \boldsymbol{\gamma}=[\gamma_1,\ldots,\gamma_m]$;
  $\gamma_i\leftarrow\gamma_i+\eta\cdot\left\{\sum_{t=1}^N[\frac{1+\hat{y}_{it}}{2}-\beta_i]+\tau\cdot\partial^q\beta_i\cdot\beta_i(1-\beta_i)\right\}$;
  $\partial^q\beta_i=-\xi_i[\psi(\xi_i\beta_i)-\psi(\xi_i(1-\beta_i))]+\frac{\xi_i\lambda_{i1}-1}{\beta_i}-\frac{\xi_i\lambda_{i2}-1}{1-\beta_i}$;
  $\boldsymbol{\varepsilon}(\boldsymbol{x}_t)=\boldsymbol{x}_t-\boldsymbol{A}\hat{\boldsymbol{y}}(\boldsymbol{x}_t)-\boldsymbol{c}$;
  $\boldsymbol{c}\leftarrow\boldsymbol{c}+\eta\cdot\{\sum_t\varphi\boldsymbol{\varepsilon}(\boldsymbol{x}_t)-\tau\cdot(\boldsymbol{\mu}-\boldsymbol{\mu}_0)/\lambda_0^{\mu}\}$;
  $\boldsymbol{A}\leftarrow\boldsymbol{A}+\eta\cdot\left\{\sum_t\left[\boldsymbol{\Sigma}_e^{-1}\boldsymbol{\varepsilon}(\boldsymbol{x}_t)\hat{\boldsymbol{y}}(\boldsymbol{x}_t)^T\right]-\tau\cdot\boldsymbol{A}\cdot diag[\boldsymbol{\alpha}]\right\}$;
  $\alpha_i\leftarrow\alpha_i+\eta\cdot\left\{\frac{n}{2\alpha_i}-\frac{\mathbf{a}_i^T\mathbf{a}_i}{2}+\frac{a^{\alpha}-1}{\alpha_i}-b_i^{\alpha}\right\}$;
  $\varphi\leftarrow\varphi+\eta\cdot\left\{\frac{Nn}{2\varphi}-\frac{1}{2}\sum_t\boldsymbol{\varepsilon}^T(\boldsymbol{x}_t)\boldsymbol{\varepsilon}(\boldsymbol{x}_t)+\tau\cdot\left(\frac{a^{\varphi}-1}{\varphi}-b^{\varphi}\right)\right\}$;
  $\boldsymbol{\Sigma}_e\leftarrow\varphi^{-1}\mathbf{I}_n$, $\mathbf{I}_n$ is an $n\times n$ unit vector ;
  $\forall i$, **if** $\beta_i<\epsilon$ *or* $\beta_i>1-\epsilon$ *or* $\|\mathbf{a}_i\|_2^2<\delta\sqrt{\mathbf{a}_i^T\boldsymbol{\Sigma}_e\mathbf{a}_i}$, *where* $\mathbf{a}_i$ *is the* i*-th column of* $\boldsymbol{A}$, **then**
    Discard the i'th dimension of $\boldsymbol{y}$; update $\boldsymbol{\Theta}$ accordingly;
until $H(p\|q)$ *has reached convergence* ;

4. BIC and Variational Bayes (VB)

Bayesian model selection is to compute the marginal likelihood $q(X_N|m,\Xi)=\int q(X_N|\boldsymbol{\Theta},m)q(\boldsymbol{\Theta}|m,\Xi)\,d\boldsymbol{\Theta}$ given a data set X_N and then to select the model scale by maximizing the likelihood, i.e., $m^*=\arg\max_m q(X_N|m,\Xi)$. Since the computation involves a usually high dimensional integral over all parameters $\boldsymbol{\Theta}$, it is difficult to obtain the exact value of $q(X_N|m,\Xi)$. Then, approximation plays important roles. BIC [9] is one widely used approximation by Laplace method, and it is simplified as follows:

$$\mathcal{J}_{bic}(\boldsymbol{\Theta},m)=-\ln q(X_N|\boldsymbol{\Theta},m)+\frac{\ln N}{2}d_m, \tag{14}$$

where N and d_m denote the sample size and the number of free parameters under model scale m, respectively. Since the last term in Eq. (14) is irrelevant to $\boldsymbol{\Theta}$, parameter estimation based on $\mathcal{J}_{bic}(\boldsymbol{\Theta},m)$ degenerates back to maximum likelihood (ML) learning, which is not good for automatic model selection. Usually, BIC is implemented via the following two-phase procedure:

$$m^*=\arg\min_m \mathcal{J}_{bic}(\hat{\boldsymbol{\Theta}}^{ML},m), \quad \hat{\boldsymbol{\Theta}}^{ML}=\arg\max_{\boldsymbol{\Theta}}\ln q(X_N|\boldsymbol{\Theta},m). \tag{15}$$

Developed recently, Variational Bayes (VB) [8] tackles the integration by means of variation methods to approximate $\ln q(X_N|m,\Xi)$ with a lower bound:

$$\begin{aligned}\mathcal{F}(p(\boldsymbol{\Theta}),p(Y),m,\Xi)&=\int p(\boldsymbol{\Theta})p(Y)\ln\left[\frac{q(X_N,Y|\boldsymbol{\Theta})q(\boldsymbol{\Theta}|m,\Xi)}{p(\boldsymbol{\Theta})p(Y)}\right]dY\,d\boldsymbol{\Theta}\\&=\ln q(X_N|m,\Xi)-KL(p(\boldsymbol{\Theta})p(Y)\|q(Y,\boldsymbol{\Theta}|X_N,m,\Xi)),\end{aligned} \tag{16}$$

with Y representing hidden variables, where $q(\boldsymbol{\Theta}|m,\Xi)$ is a prior distribution over parameters $\boldsymbol{\Theta}$ given hyperparameters Ξ and model scale m. Moreover, $KL(p\|q)=\int p\ln(p/q)\geq 0$ is the KL-divergence, and $q(Y,\boldsymbol{\Theta}|X_N,m,\Xi)\propto q(X_N,Y|\boldsymbol{\Theta})q(\boldsymbol{\Theta}|m,\Xi)$. The variational posterior is usually assumed to be further factorized as $p(\boldsymbol{\Theta})p(Y)=\prod_i p(\theta_i)\prod_t p(\boldsymbol{y}_t)$ for a computable lower bound $\mathcal{F}$. It is straightforward to show [8] that the optimum form for each component of variational posterior distribution is

$$p(\vartheta_i)\propto\exp\{\langle\ln[q(X_N,Y|\boldsymbol{\Theta})q(\boldsymbol{\Theta}|m,\Xi)]\rangle_{\prod_{j\neq i}p(\vartheta_j)}\}, \tag{17}$$

where $\vartheta_i\in\boldsymbol{\Theta}\cup Y$, and $\langle\cdot\rangle_p$ denotes expectation with respect to p.

For BFA, by putting Eqs. (2), (3) and (10) into Eq. (16), a VB-BFA algorithm is obtained to maximize $\mathcal{F}$ by iteratively computing Eq. (17). Notice that the Bernoulli distribution $q(\boldsymbol{y}|\boldsymbol{\Theta})$ in Eq. (2) can be regarded as a special case of

$$q(\boldsymbol{y}|\boldsymbol{\Theta})=\prod_{i=1}^{m}\sum_{j_i=1}^{k_i}\beta_{i,j_i}G(y_i|\mu_{i,j_i},\sigma_{i,j_i}^2), \tag{18}$$

under the conditions $\forall i,\ k_i=2,\ \mu_{i,1}=-1,\ \mu_{i,2}=1,\ \sigma_{i,1}^2=\sigma_{i,2}^2=0$. Generally, Eq. (1) with $q(\boldsymbol{y}|\boldsymbol{\Theta})$ by Eq. (18) is called non-Gaussian factor analysis (NFA) [22] or independent factor analysis (IFA) [23], which relaxes the noise-free assumption of independent component analysis (ICA) [20,18]. A VB algorithm for noisy ICA or actually NFA was proposed in [20], but it did not consider priors on the parameters in Eq. (18), while the parameters were treated with proper priors in [18,19] where Eq. (5) is extended for general $k_1,\ldots,k_m$.

Actually, the VB-BFA algorithm discussed here can be regarded a special implementation of the VB-ICA algorithm in [18,19].

5. Empirical analysis

5.1. How error in solving BQP affects model selection

In BFA learning, there are three levels of inverse problems [7], including inferring $\boldsymbol{y}$ for every $\boldsymbol{x}$, estimating the parameters, and selecting an appropriate coding length m. Three problems are corresponding to three types of optimization tasks, i.e., a discrete optimization over $\{-1,1\}^m$, a continuous optimization over the

Table 1
Algorithms for solving the BQP in Eq. (12).

Name	Description
enum	Exhaustively enumerate $\boldsymbol{y}\in\{-1,1\}^m$, which was used in BFA [4]
greedy	The greedy BQP algorithm on page 203 [13]
cdual	The canonical dual approach to the BQP (see [14] or Algorithm 2 in [11])
round	Round $\tilde{\boldsymbol{y}}=\boldsymbol{Q}_y^{-1}\boldsymbol{f}_y(\boldsymbol{x})$ to the nearest binary vector in $\{-1,1\}^m$, which was proposed [15, Table II, p. 836] for BFA learning under the name "fixed posteriori approximation", and was shown to be the least accurate one of four BQP methods in [11]

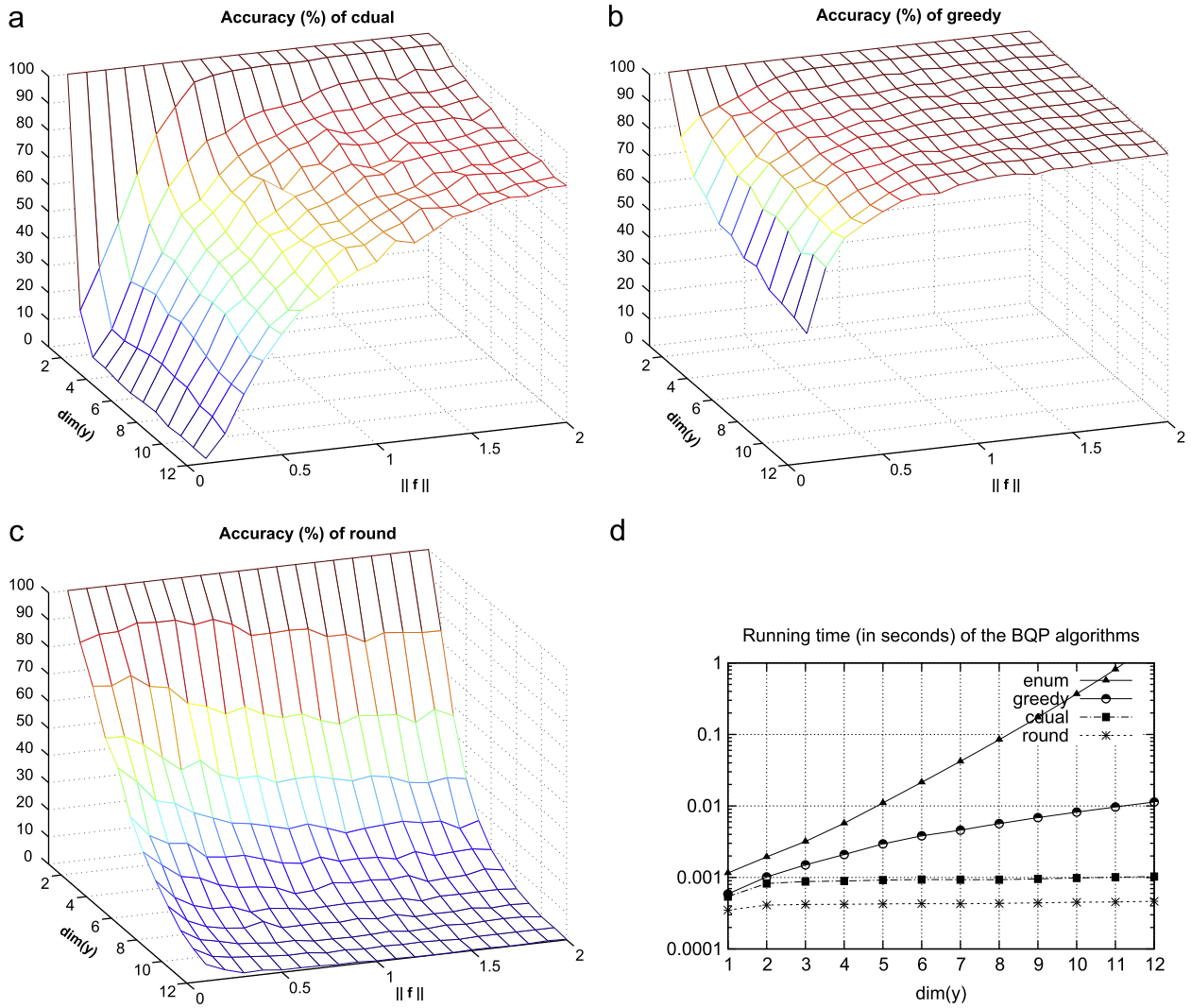


Fig. 1. Percentage of correct solutions (out of 100 runs) by (a) **cdual**, (b) **greedy**, and (c) **round** over a 10×10 ($\dim(\boldsymbol{y}) \times \|\boldsymbol{f}_y(\boldsymbol{x})\|$) configurations with $\mathrm{tr}(\boldsymbol{Q}) = 1.0$ by $\boldsymbol{Q} \leftarrow \boldsymbol{Q}/\mathrm{tr}[\boldsymbol{Q}]$ and $\boldsymbol{f}_y \leftarrow \boldsymbol{f}_y/\mathrm{tr}[\boldsymbol{Q}]$ (where $\mathrm{tr}(M)$ denotes the trace of a matrix M); (d) computation cost in seconds with time axis in log-scale.

parameter space, and a discrete optimization over a set of candidate scales $\mathcal{M} = \{1, \ldots, m_{init}\}$ where m_{init} is a positive integer given beforehand. The three optimizations are hierarchically nested to learn all the unknowns based on a set of observations $\{\boldsymbol{x}_t\}_{t=1}^N$.

Encoding a binary $\boldsymbol{y}$ for every $\boldsymbol{x}$ is formulated as an NP-hard BQP problem in Eq. (12). A preliminary study in [11] investigated the impact of four BQP solvers on BYY based automatic model selection, and suggested that some amount of error in solving BQP improves model selection performance. In this paper, the four BQP methods, as restated in Table 1, are further investigated via extensive experiments.

According to Eqs. (12) and (13), we devise and carry out a synthetic experiment by varying the coding length $\dim(\boldsymbol{y})$ and the relative size of $\|\boldsymbol{f}_y(\boldsymbol{x})\|$. Fig. 1[1] shows the accuracy and the efficiency of the approximate BQP algorithms listed in Table 1. **round** is the fastest but its accuracy degrades rapidly as $\dim(\boldsymbol{y})$ increases; **greedy** is more accurate than **round** and **cdual** but suffers from $O(\dim^3(\boldsymbol{y}))$ computation [13]. As $\|\boldsymbol{f}_y(\boldsymbol{x})\|_2$ turns small, **round** becomes slightly more accurate while the error of **cdual** and **greedy** rises up significantly.

To examine how BQP accuracy affects parameter learning, BYY-BFA is implemented without model selection (i.e., ignoring lines 13–14 in Algorithm 1) by fixing $\dim(\boldsymbol{y})$ at a pre-specified m_{init}. A synthetic data set is randomly generated according to Eq. (2), where $\dim(\boldsymbol{x}) = 10$, the true $\dim(\boldsymbol{y})$ is $m^* = 5$, and $\boldsymbol{y}$ evenly takes values from the 2^{m^*} points in $\{-1, 1\}^{m^*}$, and $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}$, $\boldsymbol{U}$ is orthogonal, $\boldsymbol{\Lambda} = diag[\lambda_1, \ldots, \lambda_m]$, $\lambda_i = 1 (\forall i)$, and $\boldsymbol{\Sigma}_e = \sigma^2 \mathbf{I}_n$. Fig. 2 shows the learning trajectory of minimum norm of columns of $\boldsymbol{A}$ with the same initialization and $m_{init} = 6$. When one extra binary factor is added, the corresponding column of $\boldsymbol{A}$ is compressed into a much smaller norm by using **cdual** or **round**. A stronger shrinkage is better for automatic model selection, as will be justified subsequently.

If $\dim(\boldsymbol{y})$ in BFA is also unknown to be determined, then model selection problem is encountered. The central goal of model selection is to obtain a compact model that minimizes generalization error. Conventionally, model selection is tackled by a two-stage implementation like Eq. (15). To avoid the expensive enumeration for each $m \in \mathcal{M}$, we activate lines 13–14 in Algorithm 1. Starting from a large coding length m_{init}, the gradient flow of $H(p\|q)$ will make the extra binary factors discarded in lines 13–14 during learning.

In experiments, synthetic data are generated according to Eq. (2) for BFA with each configuration (m^*, N, σ), where m^* is the underlying true $\dim(\boldsymbol{y})$, N is the training sample size, and λ_i is

[1] All experiments in this paper are implemented with GNU Octave 3.0.3 on a Intel Core 2 Duo 2.13 GHz with 1 GB RAM running FreeBSD 7.0.

uniformly randomly generated from the interval [1,2] so that the scale $\|\mathbf{a}_i\|_2$ corresponding to each binary factor varies, where $\mathbf{a}_i$ is the ith column of $\boldsymbol{A}$. All configuration triples (m^*, N, σ) are given in $\{(m^*, N, \sigma) | m^* \in \{3\},\ N \in V_N,\ \sigma \in V_\sigma\}$, where $V_N = \{200, 150, 100, 50, 40, 30, 20\}$, and $V_\sigma = \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. In the experiments, Algorithm 1 is implemented with $\tau = 0$, $\delta = 2$, $m_{init} = 2m^* - 1$.

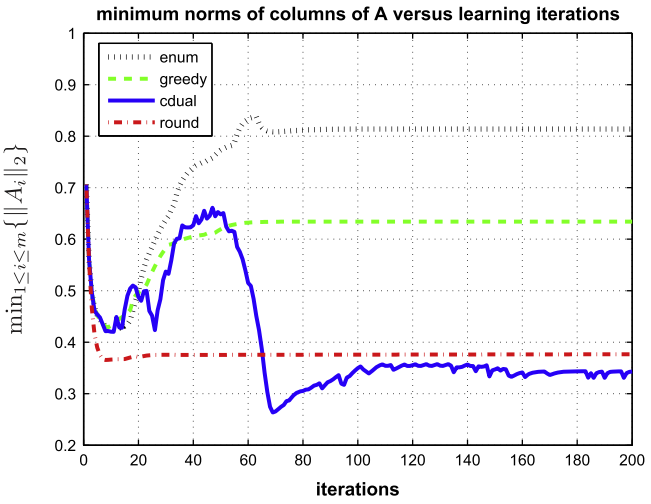


Fig. 2. Learning without dimension deduction on a synthetic data set from $\{\dim(\boldsymbol{x}) = 10,\ \text{true } \dim(\boldsymbol{y})m^* = 5, \sigma = 0.3,\ \text{sample size } N = 50\}$.

The model selection accuracies on all configurations are reported in Figs. 3(a) and (b), and 4(a) and (b). Generally, the accuracies decline as N reduces. From the aspect of noise levels σ, the accuracies first increase as σ grows from 0.01 to around 0.25, and then decrease when σ proceeds to 1.0. Among all the implementations of Algorithm 1 nested with one of the four BQP based binary encoders, the model selection accuracy shows an order of $\{$**cdual**, **round**$\}$ > **greedy** > **enum**, which reverses the BQP optimization performance order. For a detailed comparison between **cdual** and **round**, the differences of the accuracies by **cdual** minus those by **round** are represented by a heatmap in Fig. 5(a). **round** is superior in the configurations around $\sigma = 0.25$, i.e., $\{(m^*, N, \sigma) | \sigma \in U_\delta(0.25)\}$ where $U_\delta(0.25) = (0.25 - \delta, 0.25 + \delta)$, $\delta > 0$, and δ becomes small as N goes small. In contrast, **cdual** has an advantage for very small or medium noise levels.

5.2. Priors over parameters affect model selection

BYY is capable of automatic model selection without any priors over the parameters of BFA, as demonstrated in Figs. 3(a) and (b), and 4(a) and (b). Moreover, according to Eqs. (8) and (9), proper

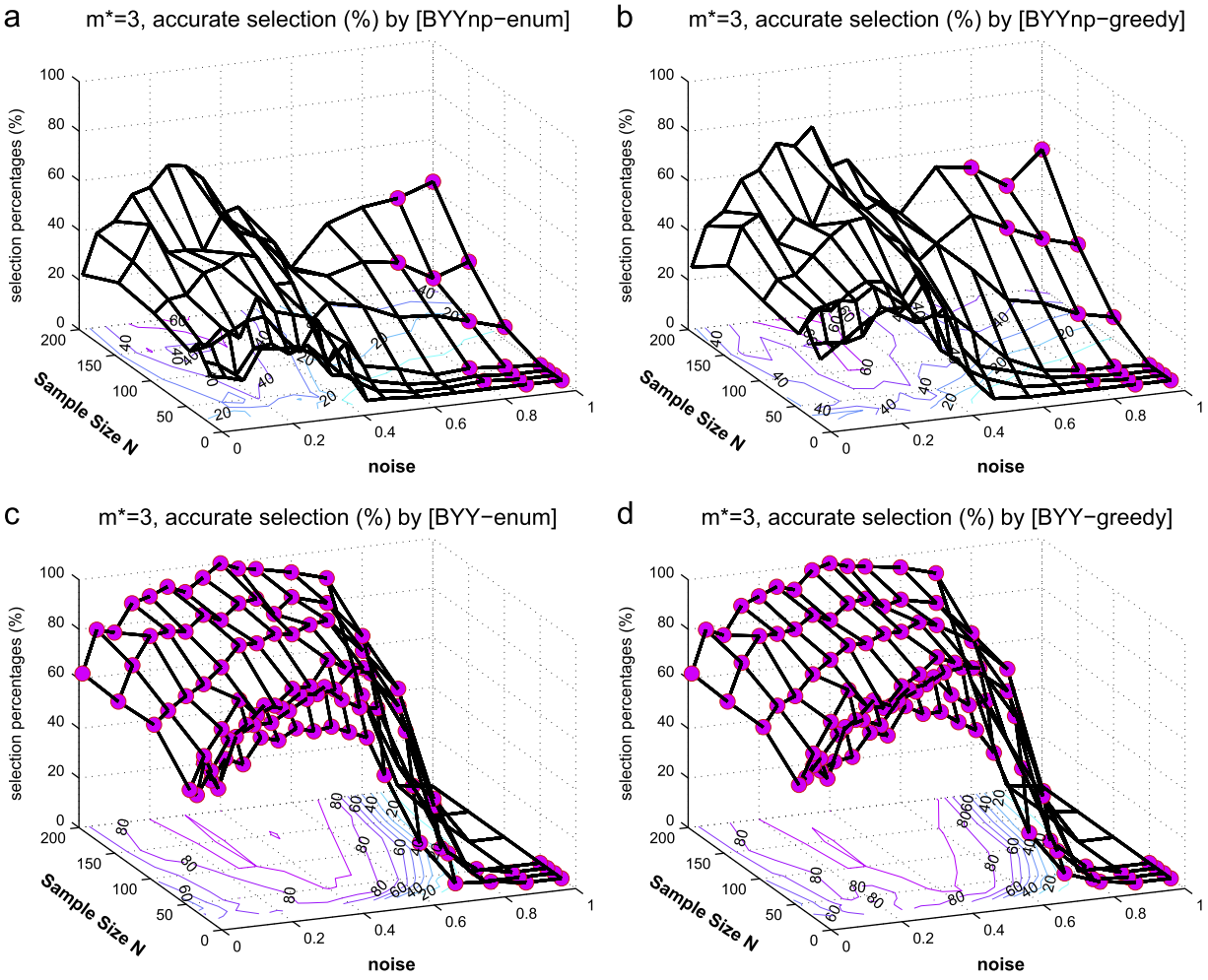


Fig. 3. Automatic model selection accuracies of BYY-BFA under various configurations: (a) **enum** without prior; (b) **greedy** without prior; (c) **enum** with prior; (d) **greedy** with prior. A red ball indicates a higher accuracy by BYY-BFA with or without prior distributions over parameters, for each BQP method. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

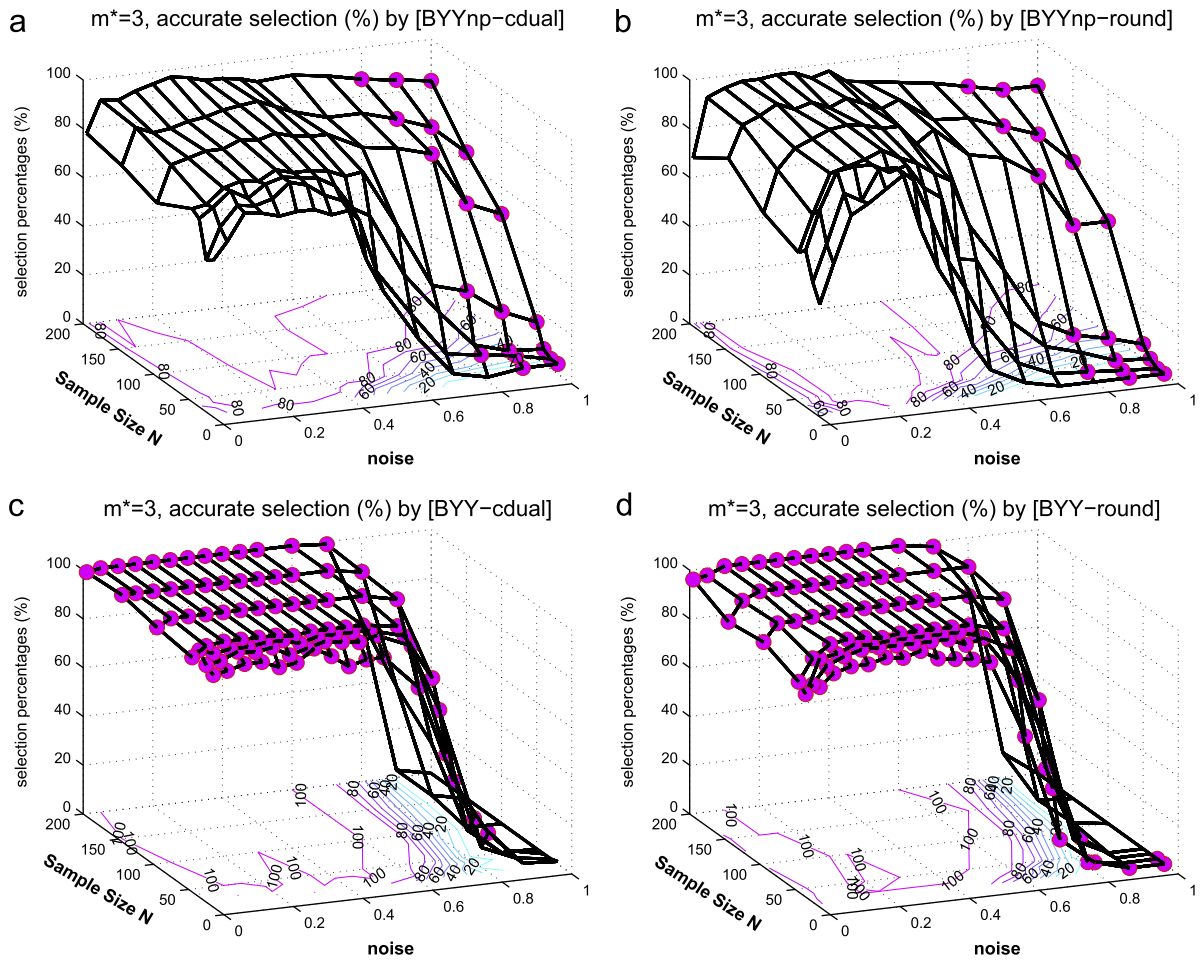


Fig. 4. Automatic model selection accuracies of BYY-BFA under various configurations: (a) **cdual**, without prior (i.e., $\tau = 0$ in Algorithm 1); (b) **round**, without prior; (c) **cdual**, with prior (i.e., $\tau = 1$ in Algorithm 1); (d) **round**, with prior.

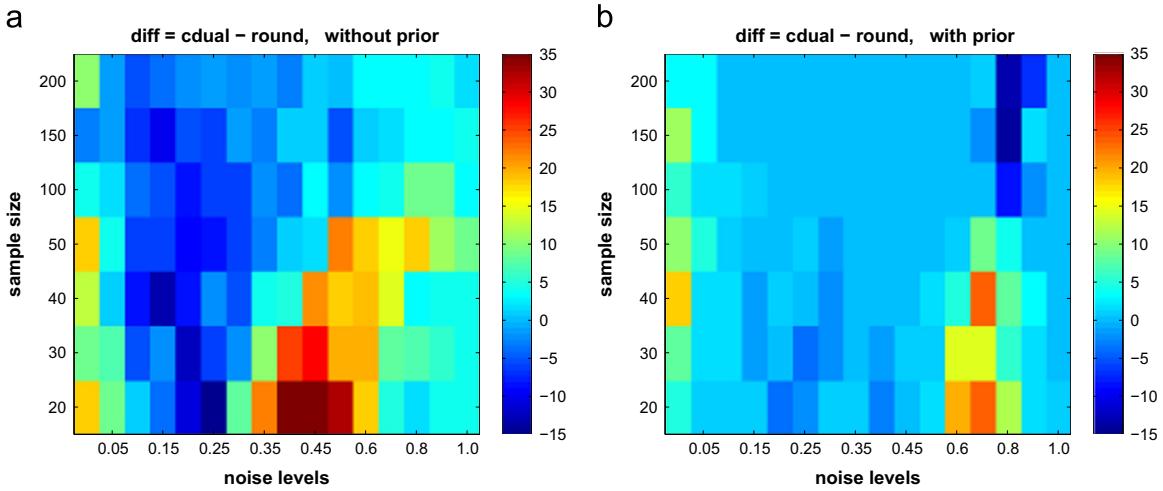


Fig. 5. Comparison of model selection performances by BYY embedded with **cdual** or **round**, by the heatmap of differences (a) between Fig. 4(a) and (b), and (b) between Fig. 4(c) and (d).

priors can be incorporated under a general guideline of BYY learning in [12]. Such efforts have been made on factor analysis in [16], Gaussian mixture model in [17]. The prior term $\ln q(\Theta|\Xi)$ in Eq. (11), which plays a regularization role, takes effect in Algorithm 1 by setting $\tau = 1$, and the hyperparameters Ξ are set according to [18,19].

The model selection accuracies of BYY-BFA aided with priors on parameters are reported in Figs. 3(c) and (d), and 4(c) and (d). The

BYY-BFA algorithm becomes improved for each BQP method, especially when N is small. Some exceptions are located at the cases of very large noise. Moreover, the gain from the incorporation of priors improves **enum** and **greey** a lot in model selection by Fig. 3(c) and (d), but still inferior to **cdual** and **round** by Fig. 4(c) and (d). As shown by the heatmap in Fig. 5(b), the difference between **cdual** and **round** is narrowed down by the benefits from priors, except when σ is around 0.7, **cdual**/**round** has a relative advantage for a small/large sample size, respectively.

5.3. Comparisons among BYY, VB, and BIC

As discussed in Section 4, the VB-BFA algorithm is a special case of the VB-ICA algorithm proposed in [18,19]. In the experiments,

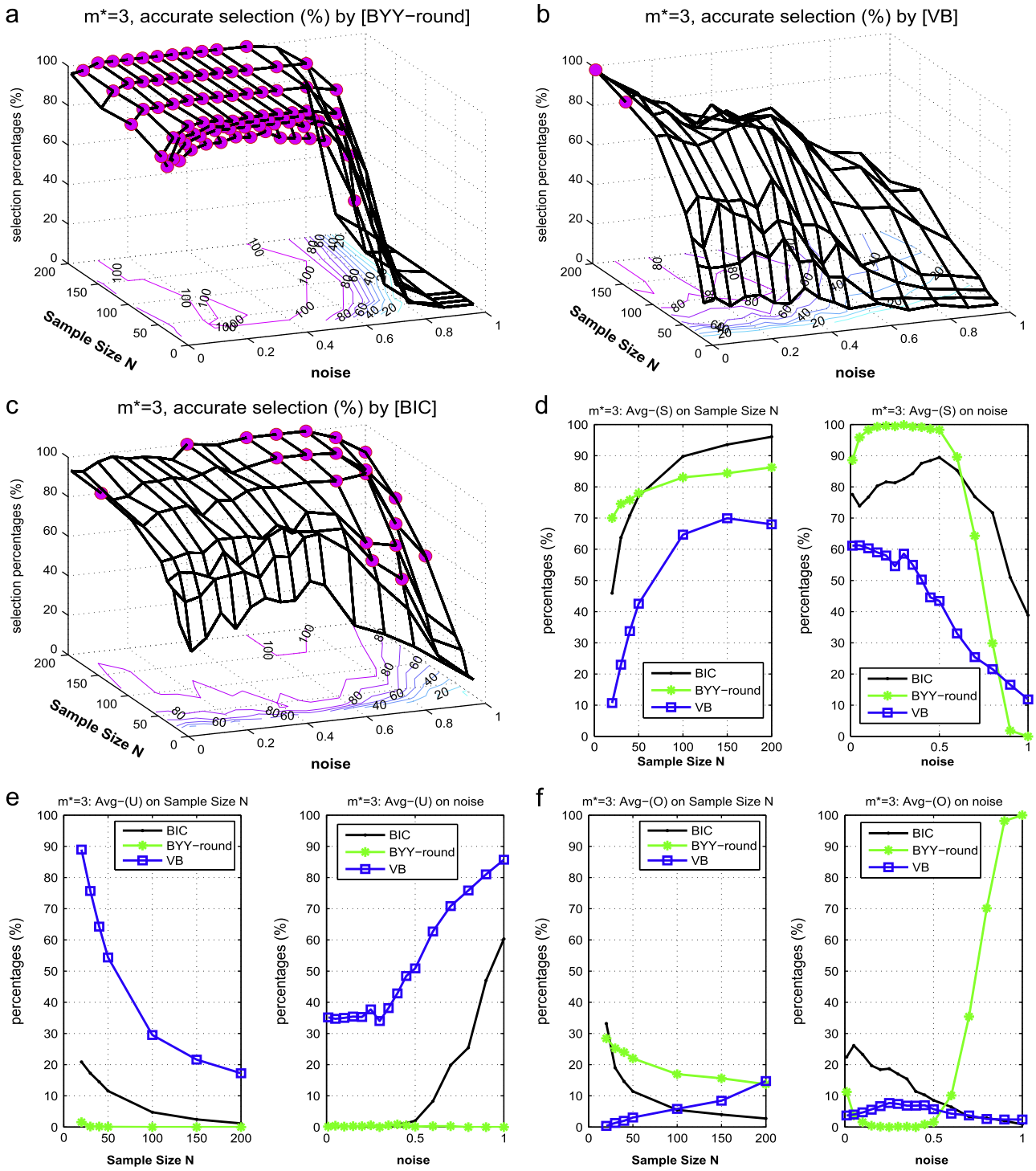
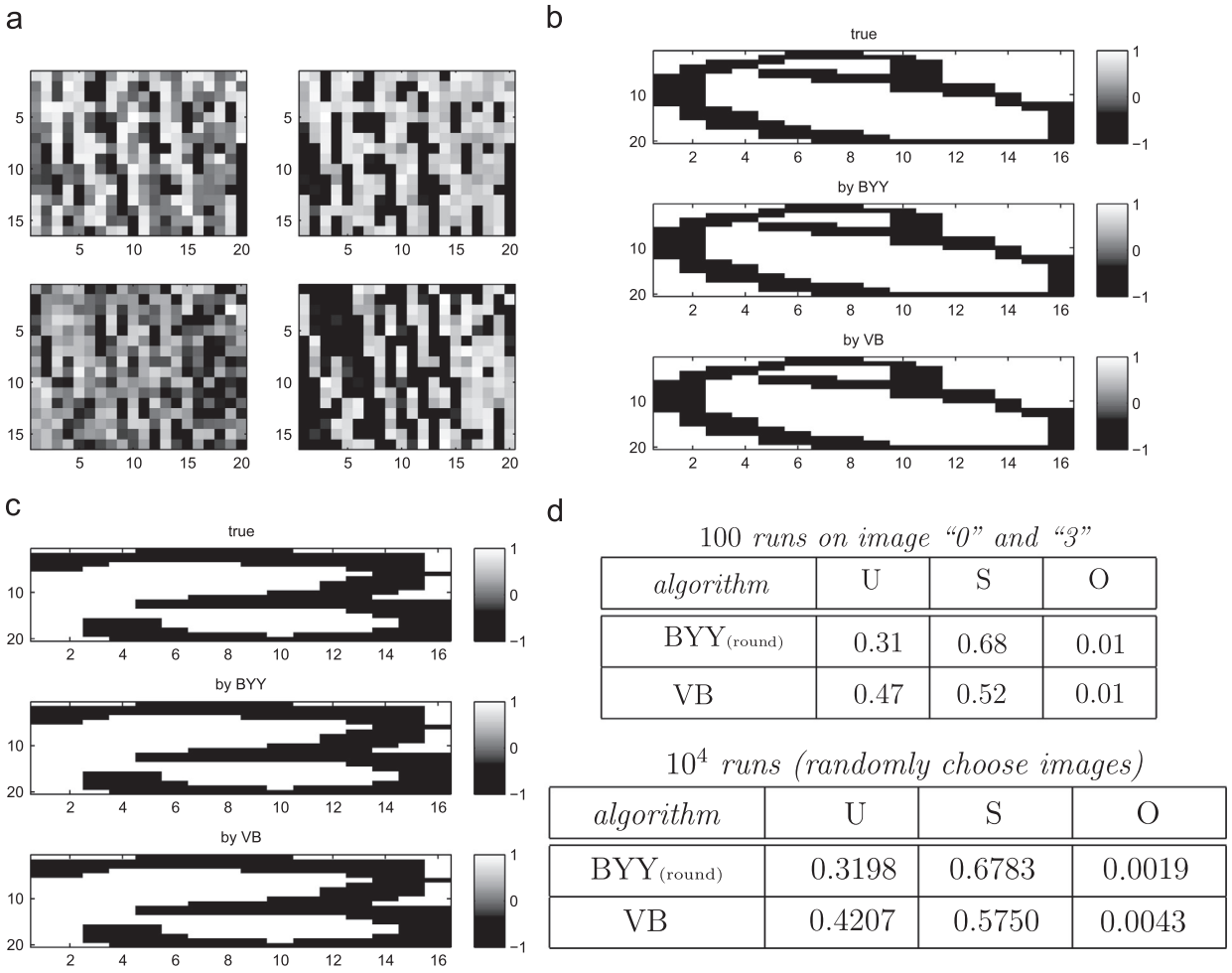


Fig. 6. Comparison of model selection performances: (a) automatic model selection by BYY with **round**; (b) automatic model selection by VB; (c) two-phase model selection by BIC. A red ball is drawn to indicate a highest accuracy ($>50\%$) by BYY, VB, and BIC, and also contours of accuracies are given in the plane of (N, *noise*). The results of success-selection (S), underestimation (U), and overestimation (O) are averaged along one axis and then projected to the other axis in (d), (e), and (f), respectively. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



100 *runs on image "0" and "3"*

algorithm	U	S	O
BYY (round)	0.31	0.68	0.01
VB	0.47	0.52	0.01

10^4 *runs (randomly choose images)*

algorithm	U	S	O
BYY (round)	0.3198	0.6783	0.0019
VB	0.4207	0.5750	0.0043

Fig. 7. Results of recovering binary images. (a) Mixed images corrupted by noise; (b) the true (*top*) binary image of "0" and the recovered ones by BYY (*middle*) or VB (*bottom*); (b) the true (*top*) binary image of "3" and the recovered ones by BYY (*middle*) or VB (*bottom*); (d) percentages of underestimation (U), success-selection (S), overestimation (O) out of: (*top*) 100 independent runs when using image "0" and "3", or (*bottom*) 10^4 runs when the images are randomly picked as binary sources.

we accordingly simplify the algorithm[2] in [19] to implement VB-BFA, and conduct automatic model selection according to lines 13–14 in Algorithm 1. For a reference, we also implement BIC in a two-phase procedure by Eq. (15), where the well-known Expectation-Maximization (EM) algorithm is adopted to estimate the parameters (see e.g., Eqs. (6)–(9) in [4] for the EM algorithm of BFA). VB and BIC are implemented on the same synthetic data used in Fig. 3.

The model selection results are presented in Fig. 6, where BYY-BFA with **round**, i.e., Fig. 4(d), is selected for comparison because it is fast and the best of all (or at least comparable to the one with **cdual**) as shown in Figs. 3 and 4. Fig. 6(a)–(c) shows that BYY is generally the best at most cases. BIC is more robust than VB, while VB performs well only when N is large and noise is mall, and deteriorates drastically as N reduces or noise increases.

Moreover, the results of underestimation (U) $\hat{m} < m^*$, success-selection (S) $\hat{m} = m^*$ and overestimation (O) $\hat{m} > m^*$, are separately averaged along one axis and then projected to the other, helping to explore the marginalized effect of either of noise level and N, where $\hat{m}$ is an estimate of the true factor number m^*. Fig. 6(d) confirms the above results that BYY is robust for a small sample size whereas BIC is robust for a large noise. It can be observed from Fig. 6(e) and (f) that BYY suffers from overestimation at high noise levels, while VB and BIC mainly suffer from underestimation.

5.4. Recovering binary images

We apply BYY-BFA and VB-BFA to bind separation of binary sources. The algorithms are demonstrated on the data set "Binary Alphadigits",[3] which consists of binary images of size 20×16 of handwritten digits "0" through "9" and capital "A" through "Z" with 39 instances for each class.

Two binary images of handwritten digits "0" and "3", as shown in the top of Fig. 7(b) and (c), are selected for example. They are mixed by a randomly generated 4×2 matrix $\boldsymbol{A}$, and then added by a Gaussian noise with variance 0.1. The noise corrupted mixed images are shown in Fig. 7(a), which appears without evident digital patterns.

Both BYY-BFA and VB-BFA are implemented in Fig. 7(a) by initializing $m_{init} = 3$. As shown in Fig. 7(b) and (c), the two binary digits can be correctly recovered with the extra binary dimension automatically discarded during learning. However, if the estimated number $\hat{m}$ of digits is wrong, then the binary images cannot be recovered accurately. An example of $\hat{m} = 1$ in Fig. 8 demonstrates

[2] In [19], two factorizations to approximate the posterior were considered with two VB algorithms, "vbICA1" and "vbICA2", derived correspondingly. Here, "vbICA2" is simplified to implement VB-BFA, because it was shown in [19] to be more robust. The matlab package is available from http://www.robots.ox.ac.uk/~parg/projects/ica/riz/code.html.

[3] The data set can be downloaded from http://www.cs.nyu.edu/~roweis/data.html.

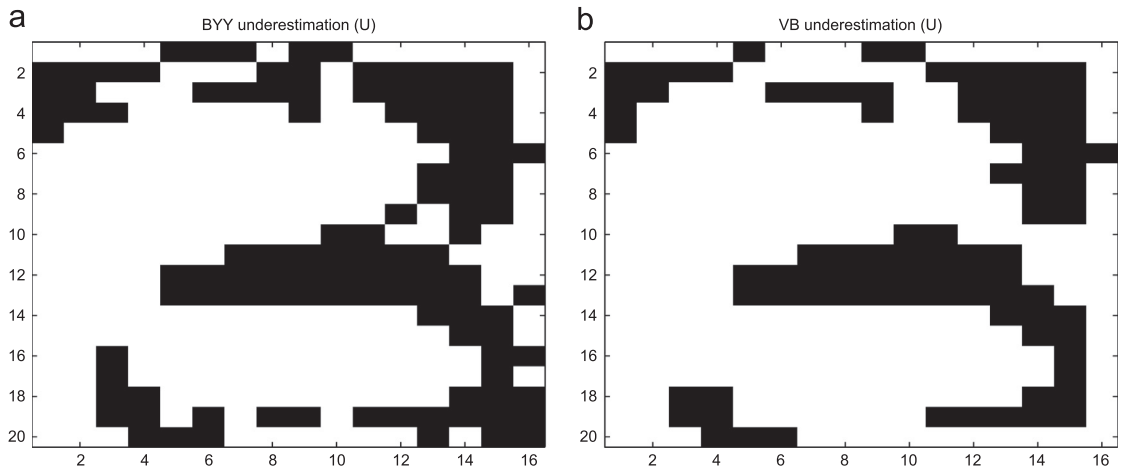


Fig. 8. An example of both BYY and VB detecting only one image "3". It shows a case of underestimation (U) with $\hat{m} < m^* = 2$, where $\hat{m}$ is the estimate of m^* by BYY or VB. (a) BYY, $\hat{m} = 1$. (b) VB, $\hat{m} = 1$.

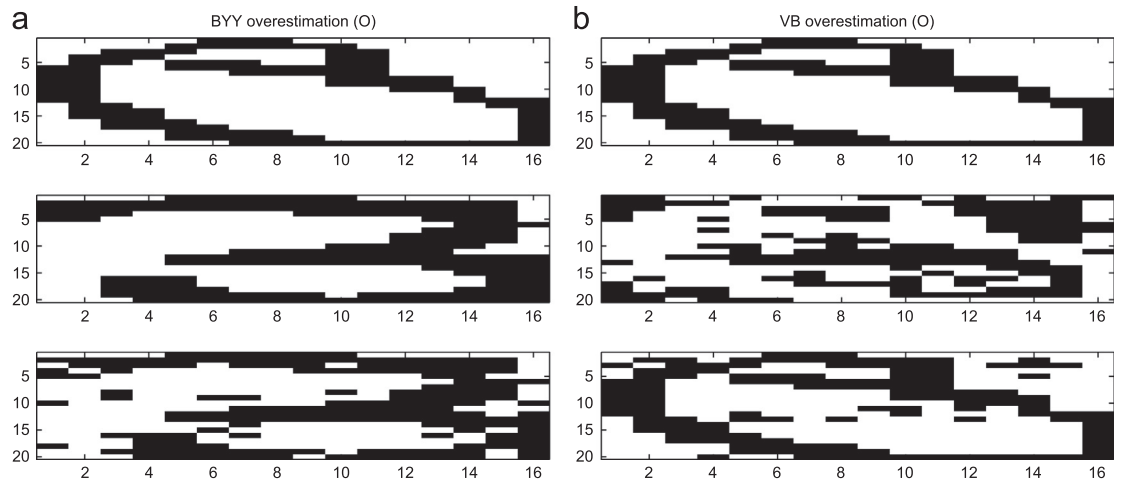


Fig. 9. An example of both BYY and VB detecting one extra image. It shows a case of overestimation (O) with $\hat{m} > m^* = 2$, where $\hat{m}$ is the estimate of m^* by BYY or VB. (a) BYY, $\hat{m} = 3$. (b) VB, $\hat{m} = 3$.

that only the digit "3" is approximately detected, while an example of $\hat{m} = 3$ in Fig. 9 shows that the extra one may duplicate or even disturb the reconstruction of the images "0" and "3".

We repeatedly generate 100 noise corrupted mixed images from image "0" and "3". The automatic model selection results, given in the top table of Fig. 7(d), show that BYY is more accurate than VB, which is consistent with the observations from Fig. 6.

For a fair selection on image classes, we randomly pick two image classes, and then randomly pick an instance within each class, and then mix them with noise in the same way as the above process. The results in the bottom table of Fig. 7(d) again justify that BYY outperforms VB.

6. Concluding remarks

This paper has investigated the performance of BYY based automatic model selection which automatically discards extra binary factors during parameter learning. A BQP optimization problem is embedded in the BFA learning process to encode a binary code for each observation. Experiments showed that some amount of error in solving the BQP may produce a learning regularization with gains not only computational efficiency but also model selection accuracy. Moreover, the BFA learning algorithm has been developed with appropriate prior distributions on parameters and it further improved model selection performance. For comparisons, we also implemented VB to perform automatic model selection, together with BIC in a two-phase implementation as a reference. Empirical analysis indicated that BYY is superior for most configurations with different training sample sizes and noise levels. BIC is more robust than VB, while VB is good only for a large sample size and low noise but declines quickly as the sample size goes down and as the noise grows large.

The harmony functional by Eq. (11) is merely a rough approximation of Eq. (8), and it ignores a term that involves the number of free parameters as given in Eq. (29) in [24]. The term is not helpful to automatic model selection but contributes to the model selection criterion in a two-phase implementation similar to BIC by Eqs. (14) and (15). With the help of this contribution, BYY may improve the performance for the cases of large noise in Fig. 6(a). Moreover, the performance may be further improved by exploring the co-dimension nature of the matrix pair A and Y with a composite indicator given in Eq. (36) of [24] to indicate whether a hidden dimension should be discarded. All these possible improvements are left for the future work.

Acknowledgments

The work described in this paper was fully supported by RGC Direct Grant project 2050502.

References

[1] A. Keprt, V. Snásel, Binary factor analysis with help of formal concepts, in: V. Snásel, R. Belohlávek (Eds.), CLA, CEUR Workshop Proceedings, vol. 110, CEUR-WS.org, 2004, pp. 90–101.

[2] G.W. Taylor, G.E. Hinton, S.T. Roweis, Modeling human motion using binary latent variables, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), NIPS, MIT Press, Cambridge, MA, 2007, pp. 1345–1352.

[3] B.L. Zhang, L. Xu, M. Fu, Learning multiple causes by competition enhanced least mean square error reconstruction, Int. J. Neural. Syst. 7 (3) (1996) 223–236.

[4] Y. An, X. Hu, L. Xu, A comparative investigation on model selection in independent factor analysis, J. Math. Model. Algorithms 5 (2006) 447–473.

[5] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[6] L. Xu, A unified learning scheme: Bayesian–Kullback Ying-Yang machines, in: D.S. Touretzky, M. Mozer, M.E. Hasselmo (Eds.), NIPS, MIT Press, pp. 444–450 (a preliminary version in Proceedings of ICONIP 95, Publishing House of Electronics Industry, Beijing, 1995, pp. 977–988).

[7] L. Xu, Machine learning problems from optimization perspective, a special issue for CDGO 07, J. Global Optim. 47 (2010) 369–401.

[8] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, Mach. Learn. 37 (2) (1999) 183–233.

[9] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.

[10] J. Rissanen, Basics of estimation, Front. Electr. Electron. Eng. China 5 (2010) 274–280.

[11] K. Sun, S. Tu, D.Y. Gao, L. Xu, Canonical dual approach to binary factor analysis, in: T. Adali, C. Jutten, J.M.T. Romano, A.K. Barros (Eds.), Independent Component Analysis and Signal Separation, Lecture Notes in Computer Science, vol. 5441, Springer, Berlin Heidelberg, 2009, pp. 346–353.

[12] L. Xu, Bayesian Ying-Yang system, best harmony learning and five action circling, Front. Electr. Electron. Eng. China 5 (3) (2010) 281–328.

[13] P. Merz, B. Freisleben, Greedy and local search heuristics for unconstrained binary quadratic programming, J. Heuristics 8 (2) (2002) 197–213.

[14] S.-C. Fang, D.-Y. Gao, R.-L. Shue, S.-Y. Wu, Canonical dual approach for solving 0–1 quadratic programming problems, J. Ind. Manag. Optim. 4 (1) (2008) 125–142.

[15] L. Xu, BYY harmony learning, independent state space and generalized APT financial analyses, IEEE Trans. Neural Netw. 12 (4) (2001) 822–849.

[16] S. Tu, L. Xu, Parameterizations make different model selections: empirical findings from factor analysis, Front. Electr. Electron. Eng. China 6 (2011) 256–274.

[17] L. Shi, S. Tu, L. Xu, Learning Gaussian mixture with automatic model selection: a comparative study on three Bayesian related approaches, Front. Electr. Electron. Eng. China 6 (2011) 215–244.

[18] R.A. Choudrey, S.J. Roberts, Variational mixture of Bayesian independent component analyzers, Neural Comput. 15 (1) (2003) 213–252.

[19] R.A. Choudrey, Variational methods for Bayesian independent component analysis (Ph.D. thesis), Oxford University, Oxford, 2002.

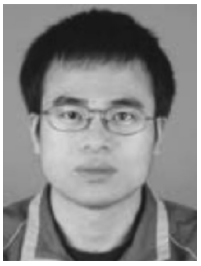
[20] N.D. Lawrence, C.M. Bishop, Variational Bayesian Independent Component Analysis, Technical Report, University of Cambridge, 1999.

[21] L. Xu, On essential topics of BYY harmony learning: current status, challenging issues, and gene analysis applications, Front. Electr. Electron. Eng. China 7 (2012) 147–196 (A special issue on Machine Learning and Intelligence Science: IScIDE (C)).

[22] L. Xu, Advances on BYY harmony learning: information theoretic perspective, generalized projection geometry, and independent factor autodetermination, IEEE Trans. Neural Netw. 15 (4) (2004) 885–902.

[23] H. Attias, Independent factor analysis, Neural Comput. 11 (1999) 803–851.

[24] L. Xu, 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 (2011) 86–119.



Dr. Shikui TU is currently a post-doc in University of Massachusetts Medical School. He got his Ph.D degree in 2012 from the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He obtained his Bachelor degree from School of Mathematical Science, Peking University, in 2006. His research interests include statistical learning, pattern recognition, and bioinformatics.

Lei Xu is chair professor of Chinese Univ Hong Kong (CUHK). He completed his Ph.D thesis at Tsinghua Univ by the end of 1986, became postdoc at Peking Univ in 1987, then promoted to associate professor in 1988 and professor in 1992. During 1989–1993 he made post-doctorial researches in Finland, Canada and USA, including Harvard and MIT. He joined CUHK as a Senior Lecturer in 1993, Professor in 1996, and Chair Professor in 2002. He has published about 100 journal papers, with a number of well-cited papers on neural networks, statistical learning, and pattern recognition, e.g., his papers got over 4500 SCI-citations, with the ones of the top-10 papers over 2600 and the citation numbers become more than doubled by Google Scholar. Prof. Xu served as a Governor of international Neural Network Society (INNS), a Past President of APNNA, and a Member of Fellow committee of IEEE Computational Intelligence Society. Prof. Xu is the Editor-in-chief of Springer open access Journal of Applied Informatics. He is also current or past Associate Editors for eight academic journals. Prof. Xu has received several national and international academic awards (e.g., 1993 National Nature Science Award, 1995 INNS Leadership Award and 2006 APNNA Outstanding Achievement Award). Prof. Xu is a Fellow of IEEE (2001–), Fellow of International Association for Pattern Recognition (2002–), member of European Academy of Sciences (2002–).