Front. Electr. Electron. Eng. China 2011, 6(2): 215–244
DOI 10.1007/s11460-011-0153-z

RESEARCH ARTICLE

Lei SHI, Shikui TU, Lei XU

Learning Gaussian mixture with automatic model selection: A comparative study on three Bayesian related approaches

Abstract Three Bayesian related approaches, namely, variational Bayesian (VB), minimum message length (MML) and Bayesian Ying-Yang (BYY) harmony learning, have been applied to automatically determining an appropriate number of components during learning Gaussian mixture model (GMM). This paper aims to provide a comparative investigation on these approaches with not only a Jeffreys prior but also a conjugate Dirichlet-Normal-Wishart (DNW) prior on GMM. In addition to adopting the existing algorithms either directly or with some modifications, the algorithm for VB with Jeffreys prior and the algorithm for BYY with DNW prior are developed in this paper to fill the missing gap. The performances of automatic model selection are evaluated through extensive experiments, with several empirical findings: 1) Considering priors merely on the mixing weights, each of three approaches makes biased mistakes, while considering priors on all the parameters of GMM makes each approach reduce its bias and also improve its performance. 2) As Jeffreys prior is replaced by the DNW prior, all the three approaches improve their performances. Moreover, Jeffreys prior makes MML slightly better than VB, while the DNW prior makes VB better than MML. 3) As the hyper-parameters of DNW prior are further optimized by each of its own learning principle, BYY improves its performances while VB and MML deteriorate their performances when there are too many free hyper-parameters. Actually, VB and MML lack a good guide for optimizing the hyper-parameters of DNW prior. 4) BYY considerably outperforms both VB and MML for any type of priors and whether hyper-parameters are optimized. Being different from VB and MML that rely on appropriate priors to perform model selection, BYY does not highly depend on the type of priors. It has model selection ability even without priors and performs already very well with Jeffreys prior, and incrementally improves as Jeffreys prior is replaced by the DNW prior. Finally, all algorithms are applied on the Berkeley segmentation database of real world images. Again, BYY considerably outperforms both VB and MML, especially in detecting the objects of interest from a confusing background.

Keywords Bayesian Ying-Yang (BYY) harmony learning, variational Bayesian (VB), minimum message length (MML), empirical comparison, Gaussian mixture model (GMM), automatic model selection, Jeffreys prior, Dirichlet, joint Normal-Wishart (NW), conjugate distributions, marginalized student's T-distribution

1 Introduction

Gaussian mixture model (GMM) [1,2] has been widely applied to clustering, object detection, image segmentation, marketing analysis, speaker identification, and optical character recognition, etc. [3–5]. Learning a GMM includes parameter learning for estimating all the unknown parameters and model selection for determining the number k of Gaussian components. Parameter learning is usually implemented under the maximum likelihood principle by an expectation-maximization (EM) algorithm [2,6,7]. A conventional model selection approach is featured by a two-stage implementation. The first stage enumerates k to get a set $\mathcal{M}$ of candidate models with the unknown parameters of each candidate estimated by the EM algorithm. In the second stage, one best candidate is selected by a model selection criterion. Examples of such criteria include Akaike's information criterion (AIC) [8], Bayesian inference criterion (BIC) [9], minimum description length (MDL) criterion [10,11] (which stems from another viewpoint but coincides with BIC when it is simplified to an analytically computable criterion), etc. However, this two-stage implementation

Received April 21, 2011; accepted April 30, 2011

Lei SHI, Shikui TU, Lei XU (✉)
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China
E-mail: lxu@cse.cuhk.edu.hk

suffers from a huge computation because it requires parameter learning for each $k \in \mathcal{M}$. Moreover, a larger k often implies more unknown parameters, and then parameter estimation becomes less reliable so that the criterion evaluation reduces its accuracy (see Sect. 2.1 in Ref. [12] for a detailed discussion).

One road to tackle the problems is referred to automatic model selection that automatically determines k during parameter learning. An early effort is rival penalized competitive learning (RPCL) [13–15] with the number k automatically determined during learning. Moreover, three Bayesian related approaches can be implemented with a nature of automatic model selection. One is Bayesian Ying-Yang (BYY) learning, proposed in Ref. [16] and systematically developed in the past decade and a half, which provides a general statistical learning framework that can handle both parameter learning and model selection under a best harmony principle. BYY is capable of automatic model selection even without imposing any priors on the parameters, and its performance can be further improved with appropriate priors incorporated according to a general guideline [12,17].

With the help of appropriate priors, efforts have been made on minimum message length (MML) [18] and variational Bayes (VB) [19–21] for learning GMM with automatic model selection. MML approach minimizes a two-part message for a statement of model and a statement of data encoded by that model, which involves a Fisher-term computing the determinant of Fisher information matrix [22,23]. Figueiredo and Jain in Ref. [18] developed such an MML algorithm for GMM with a prior that is the product of independent Jeffreys priors on the mixing weights and the parameters in Gaussian components individually, and the Fisher information matrix was approximated by a block-diagonal matrix. VB tackles the difficulty in computing the marginal likelihood with a lower bound by means of variational method. The existing VB algorithms for GMM [20,24] are featured by a Dirichlet prior on mixing weights and an *independent* Normal-Wishart (NW) prior on each Gaussian component's parameters.

Still, there is lack of a systematic comparative investigation on the relative strengths and weaknesses of the above three Bayesian related approaches in term of their automatic model selection performances. This paper aims to make such a systematic comparison. We consider GMM with two types of priors on its parameters. One is the Jeffreys prior on all GMM parameters, which is a non-informative prior defined to be proportional to the square root of the determinant of the Fisher information, via approximately using a block-diagonal complete data Fisher information given in Ref. [18]. The other is a parametric conjugate prior [25], which imposes a Dirichlet prior on the mixing weights and a *joint* Normal-Wishart prior, shortly denoted as DNW.

The algorithm for MML is adopted from Ref. [18], either used directly or with some modification for DNW prior. The algorithm for VB with DNW prior is a variant of the one in Ref. [20]. Instead of the *independent* NW prior, this variant algorithm considers the *joint* NW prior suggested in Ref. [25] to take the advantage of conjugate distributions. Moreover, the BYY algorithm for Jeffreys prior can be regarded as a special case of the unified Ying-Yang learning procedure introduced in Ref. [12] (see its Sect. 3.1 and especially Fig. 7). Still, there are no available algorithms yet in the existing literature for implementing VB with Jeffreys prior and implementing BYY with DNW prior. The corresponding algorithms have been developed in this paper.

Without any priors on the parameters, maximum a posteriori (MAP) approach, VB and MML all degenerate to maximum likelihood (ML) learning, and BYY is still capable of automatic model selection. Further considering priors merely on the mixing weights, we have several observations via simulation study. VB is better than MML and MAP for a Jeffreys prior, but VB deteriorates to be inferior to both MML and MAP with the Jeffreys replaced by a Dirichlet prior. For either Jeffreys or Dirichlet prior, BYY considerably outperforms MAP, VB and MML. For the Jeffreys, MAP and MML bias to oversized models, while VB and BYY bias to undersized models. For the Dirichlet, all the approaches incline to oversized models. Moreover, optimizing the hyper-parameters of the Dirichlet prior further improves the performances of BYY, but brings down the other approaches.

Next, we consider a full prior on all parameters. The performances of automatic model selection are evaluated through extensive experiments on a wide range of randomly generated data sets, via controlling the hardness of tasks performed by varying the dimension d of data, the number N of samples, the number k^* of Gaussian components, and the overlap degree β of Gaussian components. It is empirically found that considering priors on all the parameters of GMM makes each approach reduce its bias and also improve its performance. Comparing with Jeffreys priors, this performance increment is more obvious as the Dirichlet is added with a joint NW, such that a full DNW prior outperforms a full Jeffreys prior for most approaches. The Jeffreys prior makes MML slightly better than VB, while the DNW priors make VB better than MML. BYY considerably outperforms the rest approaches for any type of priors and whether or not hyper-parameters are optimized.

As the hyper-parameters of DNW prior are optimized by each of its own learning principle, BYY further improves its performance considerably and outperforms the others significantly, which concurs with the nature that learning hyper-parameters is a part of the entire BYY harmony learning (see the learning procedure shown

in Fig. 6(a) in Ref. [17] or in Fig. 5(a) in Ref. [26]). Also, MAP improves its counterpart with the hyper-parameters pre-fixed too, which could be understood from the relation that MAP can be regarded as a degenerated case of the BYY harmony learning (see Eq. (A.4) in Ref. [17] or Eq. (39) in Ref. [26]). However, both VB and MML deteriorate when there are too many free hyper-parameters, especially the performance of VB drops drastically, becoming even inferior to MAP. The reason is that VB and MML maximize the marginal likelihood via variational approximation and Laplace approximation respectively. Maximizing the marginal likelihood with respect to a free priori $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ makes it tend to the maximum likelihood, which is not good for optimizing the hyper-parameters, see Sect. 5.2. In other words, VB and MML lack a good guide for optimizing the hyper-parameters of DNW prior.

Finally, we apply all the algorithms to unsupervised image segmentation on the Berkeley segmentation database of real world images. The segmentation performances are evaluated by the probabilistic Rand (PR) index [27]. Again, BYY outperforms VB and MML considerably, with a better ability to detect the objects of interest that are even highly confused with the background. Still, the DNW prior results in a better performance than the Jeffreys prior for all the three approaches, while BYY-DNW always performs the best.

The remainder of this paper is organized as follows. Section 2 introduces GMM, model selection, and two types of priors. Section 3 considers learning algorithms of BYY, MML, and VB with Jeffreys prior. Section 4 further proceeds to learning algorithms for VB, MML, and BYY with the Dirichlet prior on the mixing weights and the DNW prior on all the parameters of GMM. Section 5 is devoted to experimentally evaluate the performances of automatic model selection by these algorithms through a wide range of synthetic data sets and the Berkeley segmentation database. Finally, concluding remarks are made in Sect. 6.

2 Gaussian mixture, model selection, and using priors

2.1 Gaussian mixture model and EM algorithm

GMM [1] assumes that an observation $\boldsymbol{x} \in \mathcal{R}^d$ is distributed as a linear mixture of k Gaussian distributions, i.e.,

$$\begin{aligned} q(\boldsymbol{x}|\boldsymbol{\Theta}) &= \sum_{i=1}^{k} \alpha_i q(\boldsymbol{x}|\boldsymbol{\theta}_i), \\ q(\boldsymbol{x}|\boldsymbol{\theta}_i) &= G(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1}) \\ &= \frac{|\boldsymbol{T}_i|^{1/2}}{(2\pi)^{d/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{T}_i (\boldsymbol{x}-\boldsymbol{\mu}_i)\right], \end{aligned} \quad (1)$$

with parameters $\boldsymbol{\Theta} = \boldsymbol{\alpha} \cup \{\boldsymbol{\theta}_i\}_{i=1}^k$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_k]^{\mathrm{T}}$, and $\boldsymbol{\theta}_i = (\boldsymbol{\mu}_i, \boldsymbol{T}_i)$. Therein α_i is the mixing weight for the ith component with each $\alpha_i \geqslant 0$ and $\sum_{i=1}^k \alpha_i = 1$, and $G(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{T}^{-1})$ denotes a Gaussian density with a mean $\boldsymbol{\mu}$ and a precision (inverse covariance) matrix $\boldsymbol{T}$. Each $q(\boldsymbol{x}|\boldsymbol{\theta}_i)$ is named as a component, and k refers to the component number. Here and throughout this paper, $q(\cdot)$ is used to denote *a generative distribution, likelihood or prior*, while $p(\cdot)$ refers to *a posterior distribution*.

GMM can be also regarded as a latent variable model by introducing a binary latent vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_k]^{\mathrm{T}}$, subject to $y_i \in \{0, 1\}, \forall i$, and $\sum_{i=1}^k y_i = 1$. The generative process of an observation $\boldsymbol{x}$ is interpreted as follows: 1) $\boldsymbol{y}$ is sampled from a multinomial distribution with probabilities $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_k]^{\mathrm{T}}$; 2) $\boldsymbol{x}$ is randomly generated by the ℓth Gaussian components $q(\boldsymbol{x}|\boldsymbol{\theta}_\ell)$ with $y_\ell = 1$. Therefore, we have

$$\begin{aligned} q(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\Theta}) &= \prod_{i=1}^{k} \left[G\left(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1}\right)\right]^{y_i}, \\ q(\boldsymbol{y}|\boldsymbol{\Theta}) &= \prod_{i=1}^{k} \alpha_i^{y_i}. \end{aligned} \quad (2)$$

The likelihood $q(\boldsymbol{x}|\boldsymbol{\Theta})$ described in Eq. (1) can also be computed by marginalizing $q(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\Theta}) = q(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\Theta}) q(\boldsymbol{y}|\boldsymbol{\Theta})$ over $\boldsymbol{y}$, i.e., $q(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{\boldsymbol{y}} q(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\Theta})$. For a set of i.i.d. samples $\boldsymbol{X}_N = \{\boldsymbol{x}_t\}_{t=1}^N$ from GMM, there is correspondingly a set of latent variables $\boldsymbol{Y} = \{\boldsymbol{y}_t\}_{t=1}^N$ and $q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta}) = \prod_{t=1}^N q(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{\Theta}) q(\boldsymbol{y}_t|\boldsymbol{\Theta}) = \prod_{t=1}^N \prod_{i=1}^k [\alpha_i G(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1})]^{y_{it}}$.

Given $\boldsymbol{X}_N$, there are three levels of inverse problems in GMM modeling [28]. The first level is the inference of the component labels $\boldsymbol{y}$. The second is the estimation of the parameters $\boldsymbol{\Theta}$ given a fixed k, which is usually implemented by maximizing the likelihood $q(\boldsymbol{X}_N|\boldsymbol{\Theta}) = \prod_{t=1}^N q(\boldsymbol{x}_t|\boldsymbol{\Theta})$ with the help of the following well known EM algorithm [2,6,7]:

E-step: for $i = 1, 2, \ldots, k$, and $t = 1, 2, \ldots, N$, let $\alpha_i^p = \alpha_i^{\mathrm{old}}$ and get:

$$p_{it} = \frac{\alpha_i^p G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_i^{\mathrm{old}}, \boldsymbol{T}_i^{\mathrm{old}\ -1}\right)}{\sum_{j=1}^k \alpha_j^p G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_j^{\mathrm{old}}, \boldsymbol{T}_j^{\mathrm{old}\ -1}\right)}, \ n_i = \sum_{t=1}^{N} p_{it}. \quad (3)$$

M-step: Let $s_i = n_i$ and $n_0 = 0$, update $\boldsymbol{\Theta} = \boldsymbol{\alpha} \cup \{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^k$:

$$\begin{aligned} \alpha_i^{\mathrm{new}} &= \frac{s_i}{\sum_{j=1}^k s_j}, \\ \boldsymbol{\mu}_i^{\mathrm{new}} &= \frac{1}{n_i} \sum_{t=1}^{N} p_{it} \boldsymbol{x}_t, \\ \boldsymbol{T}_i^{\mathrm{new}\ -1} &= \frac{1}{n_i - n_0} \sum_{t=1}^{N} p_{it} (\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}})(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}})^{\mathrm{T}}, \end{aligned} \quad (4)$$

where n_0 is a given constant for a regularization purpose, with $n_0 = 0$ for the maximum likelihood learning.

Instead of $\alpha_i^p = \alpha_i^{\text{old}}$, other options of α_i^p also lead to different types of learning, as to be introduced in the rest of this paper.

The third level is the determination of an appropriate k. These three levels of inverse problems are hierarchically nested in order, with the third level being the outmost.

2.2 Automatic model selection and three Bayesian related approaches

Shortly, automatic model selection means to automatically determine an appropriate k during parameter learning. An early effort is RPCL [13–15]. The key idea is that not only the winning Gaussian component moves a little bit to adapt the current sample but also the rival (i.e., the second winner) Gaussian component is repelled a little bit from this sample to reduce a duplicated information allocation. As a result, an extra Gaussian component is driven far away from data, or equivalently its mixing weight or proportion of samples that are allocated to this component is driven towards zero. In general, RPCL is applicable to any model that consists of k individual substructures, with extra substructures discarded by a rival penalized mechanism and thus model selection made automatically [29–33].

According to its general formulation (see the last part of Sect. 2.1 in Ref. [12] or pages 65–67 in Ref. [34]), automatic model selection is a nature of learning a mixture of k individual substructures with k initialized large enough, by a learning rule or principle with the following two features. First, there is an indicator on a subset of parameters that actually represents a particular structural component, and the component is effectively discarded if its corresponding indicator becomes zero. Second, in implementation of this algorithm or principle, there is an intrinsic mechanism that drives such an indicator towards zero if the corresponding structure is redundant and thus can be effectively discarded. As such, three Bayesian related approaches can be implemented with such a nature of automatic model selection.

MML approach [22,23] is mathematically equivalent to a maximum a posteriori method that improves the likelihood function $q(\boldsymbol{X}|\boldsymbol{\Theta}, k)$ by an improper prior that modifies a proper prior $q(\boldsymbol{\Theta}|k)$ into becoming proportional to $q(\boldsymbol{\Theta})/|\boldsymbol{I}(\boldsymbol{\Theta})|^{1/2}$ by the Fisher information matrix $\boldsymbol{I}(\boldsymbol{\Theta})$. Figueiredo and Jain in Ref. [18] proposed such an MML algorithm for GMM with a Jeffreys prior (MML-Jef), where $\boldsymbol{I}(\boldsymbol{\Theta})$ is approximated by a block-diagonal matrix. VB tackles the difficulty in computing the marginal $q(\boldsymbol{X}|k) = \int q(\boldsymbol{X}|\boldsymbol{\Theta}, k) q(\boldsymbol{\Theta}|k)\, \mathrm{d}\boldsymbol{\Theta}$, with a computable lower bound by means of variational method. The existing VB algorithms for GMM [19–21] are featured by a Dirichlet prior on mixing weights α_j, $j = 1, 2, \ldots, k$, and an *independent* Normal-Wishart (NW) prior on each Gaussian component's parameters.

BYY learning on typical structures leads to new model selection criteria, new techniques for implementing regularization and a class of algorithms that implement automatic model selection during parameter learning. Details are referred to Refs. [12,17,31–33,35]. Both MML and VB perform automatic model selection based on appropriate priors $q(\boldsymbol{\Theta}|k)$, as well as an approximated expression of $\boldsymbol{I}(\boldsymbol{\Theta})$ by an analytical function of parameters. Favorably, BYY is capable of automatic model selection even without imposing any priors on the parameters, and its performance can be further improved as appropriate priors are incorporated according to a general guideline [12,17].

Still, there is lack of a systematic comparative investigation on the relative strengths and weaknesses of the above three Bayesian related approaches in term of their automatic model selection performances. This paper aims to make such a systematic comparison.

2.3 Jeffreys prior and Dirichlet-Normal-Wishart prior

We consider GMM in Eq. (1) with two types of priors $q(\boldsymbol{\Theta})$ on its parameters $\boldsymbol{\Theta} = \{\alpha_j, \boldsymbol{\mu}_j, \boldsymbol{T}_j\}$. One is the Jeffreys prior, which is a non-informative prior defined to be proportional to the determinant of the Fisher information. The other is a parametric conjugate prior [25], which imposes a Dirichlet prior on the mixing weights α_j, a *joint* Normal-Wishart prior that consists of a Normal distribution on $\boldsymbol{\mu}_j$ conditional on the Wishart distributed $\boldsymbol{T}_j$, and thus we shortly denote this conjugate prior as DNW.

The Jeffreys prior is computed as

$$q(\boldsymbol{\Theta}) \propto \sqrt{|\boldsymbol{I}(\boldsymbol{\Theta})|},$$
$$\boldsymbol{I}(\boldsymbol{\Theta}) = -\mathrm{E}\left[\frac{\partial^2 \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta})}{\partial \mathrm{vec}(\boldsymbol{\Theta}) \partial \mathrm{vec}(\boldsymbol{\Theta})^{\mathrm{T}}}\right], \tag{5}$$

where $\boldsymbol{I}(\boldsymbol{\Theta})$ is the Fisher information matrix. Due to the difficulty of getting an exact analytical expression of $\boldsymbol{I}(\boldsymbol{\Theta})$, it is usually computed approximately.

We follow Ref. [18] to approximate $\boldsymbol{I}(\boldsymbol{\Theta})$ by the following block-diagonal complete-data Fisher information matrix $\boldsymbol{I}_c(\boldsymbol{\Theta})$ such that $\boldsymbol{I}_c(\boldsymbol{\Theta}) - \boldsymbol{I}(\boldsymbol{\Theta})$ is positive definite:

$$\boldsymbol{I}_c(\boldsymbol{\Theta}) = N \times \text{Block-Diag}[\alpha_1 \boldsymbol{I}_c(\boldsymbol{\theta}_1), \alpha_2 \boldsymbol{I}_c(\boldsymbol{\theta}_2), \ldots, \alpha_k \boldsymbol{I}_c(\boldsymbol{\theta}_k), \boldsymbol{I}_c(\boldsymbol{\alpha})], \tag{6}$$

$$|\boldsymbol{I}_c(\boldsymbol{\theta}_i)| \propto |\boldsymbol{T}_i|^{-d}, \quad |\boldsymbol{I}_c(\boldsymbol{\alpha})| \propto \prod_{i=1}^{k} \alpha_i^{-1}, \tag{7}$$

where $\boldsymbol{I}_c(\boldsymbol{\theta}_i)$ is the Fisher information of the ith Gaussian component, and $\boldsymbol{I}_c(\boldsymbol{\alpha})$ is the Fisher information of the mixing weights. It follows from Eqs. (5)–(7) that

$$|\boldsymbol{I}(\boldsymbol{\Theta})| \propto N^{k(\rho+1)} \prod_{i=1}^{k} \alpha_i^{\rho-1} \prod_{i=1}^{k} |\boldsymbol{T}_i|^{-d},$$

$$q(\boldsymbol{\Theta}) \propto N^{\frac{k(\rho+1)}{2}} \prod_{i=1}^{k} \alpha_i^{\frac{\rho-1}{2}} \prod_{i=1}^{k} |\boldsymbol{T}_i|^{-\frac{d}{2}}, \quad (8)$$

where $\rho = d + d(d+1)/2$ is the number of free parameters in each Gaussian component. Readers are referred to Appendix B for an alternative perspective with Eq. (8) derived from a block-diagonal approximation to the Fisher information matrix of a lower bound of the likelihood function.

The DNW prior is expressed by

$$q(\boldsymbol{\Theta}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi) \prod_{i=1}^{k} G(\boldsymbol{\mu}_i|\boldsymbol{m}_i, \boldsymbol{T}_i^{-1}/\beta) \mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi},\gamma), \quad (9)$$

$$\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi) = \frac{\Gamma(\xi)}{\prod_{i=1}^{k}\Gamma(\xi\lambda_i)} \left(\prod_{i=1}^{k} \alpha_i^{\xi\lambda_i - 1} \right),$$

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_k]^{\mathrm{T}}, \ \sum_{i=1}^{k} \lambda_i = 1, \ \lambda_i \geqslant 0, \xi > 0, \quad (10)$$

$$\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi},\gamma) = \frac{|\boldsymbol{\Phi}|^{\frac{\gamma}{2}} |\boldsymbol{T}_i|^{\frac{\gamma-d-1}{2}} 2^{-\frac{d\gamma}{2}}}{\Gamma_d(\frac{\gamma}{2})} \exp\left\{ -\frac{1}{2}\mathrm{Tr}(\boldsymbol{T}_i\boldsymbol{\Phi}) \right\},$$

$$\beta > 0, \gamma > d-1, \quad (11)$$

where $\Gamma(\cdot)$ is the Gamma function, and $\Gamma_d(\cdot)$ is the generalized Gamma function with $\Gamma_d(a/2) = \pi^{d(d-1)/4} \prod_{j=1}^{d} \Gamma(\frac{a-j+1}{2})$.

It follows from Ref. [25] that the *independent* NW prior used in Ref. [20] is not conjugate to Gaussian likelihood when both mean and precision of a Gaussian are unknown. To take the advantage of conjugate distributions, this paper considers the *joint* NW prior suggested in Ref. [25]. For the Normal priori $G(\boldsymbol{\mu}_i|\boldsymbol{m}_i, \boldsymbol{T}_i^{-1}/\beta)$ on the mean vectors $\boldsymbol{\mu}_i$, this paper considers two alternatives. Following Refs. [20,24,25], one considers the same $\boldsymbol{m}_i = \boldsymbol{m}$ for all the mean vectors $\boldsymbol{\mu}_i$. Also, we relax each $\boldsymbol{m}_i$ to be freely adapted instead of being bundled together.

3 Algorithms with Jeffreys priors

3.1 BYY harmony learning and BYY-Jef algorithms

Firstly proposed in Ref. [16] and systematically developed over a decade and half, BYY harmony learning theory is a general statistical learning framework that provides not only new model selection criteria but also automatic model selection algorithms, under a best harmony principle [17]. Readers are referred to Ref. [12] for a latest systematical introduction about BYY harmony learning.

Briefly, a BYY system consists of Yang machine and Ying machine, respectively corresponding to two types of decomposition, namely, Yang $p(\boldsymbol{R}|\boldsymbol{X})p(\boldsymbol{X})$ and Ying $q(\boldsymbol{X}|\boldsymbol{R})q(\boldsymbol{R})$, where the data $\boldsymbol{X}$ is regarded as generated from its inner representation $\boldsymbol{R} = \{\boldsymbol{Y}, \boldsymbol{\Theta}\}$ that consists of latent variables $\boldsymbol{Y}$ and parameters $\boldsymbol{\Theta}$, supported by a hyper-parameter set $\boldsymbol{\Xi}$ that consists of both the hyper-parameters $\boldsymbol{\Xi}_q$ in the distributions of Ying machine and the hyper-parameters $\boldsymbol{\Xi}_p$ in Yang machine. The harmony measure is mathematically expressed as follows [12,17]:

$$H(p||q,\boldsymbol{\Xi}) = \int p(\boldsymbol{R}|\boldsymbol{X})p(\boldsymbol{X}) \ln[q(\boldsymbol{X}|\boldsymbol{R})q(\boldsymbol{R})] \, \mathrm{d}\boldsymbol{X}\mathrm{d}\boldsymbol{R}. \quad (12)$$

Maximizing the above $H(p||q,\boldsymbol{\Xi})$ leads to not only a best matching between the Ying-Yang pair, but also a compact model with a least complexity. Such an ability can be observed from several perspectives (see Sect. 4.1 in Ref. [12]).

Being different from MML and VB that both base on an appropriate prior $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ to make model selection, the BYY harmony learning by Eq. (12) bases on $q(\boldsymbol{R}) = q(\boldsymbol{Y}|\boldsymbol{\Theta})q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ to make model selection, with $q(\boldsymbol{Y}|\boldsymbol{\Theta})$ in a role that is not only equally important to $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ but also easy computing, while $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ is still handled in a way similar to MML and VB. As addressed in Sect. 2.2 of Ref. [12], the BYY harmony learning leads to improved model selection via either or both of improved model selection criteria and learning algorithms with automatic model selection.

Maximizing $H(p||q,\boldsymbol{\Xi})$ is implemented with the help of the general two-stage iterative procedure shown by Fig. 6(a) in Ref. [17] (also see Eqs. (6) and (7) in Ref. [28] and Fig. 5(b) in Ref. [12]). The first stage estimates $\boldsymbol{\Xi}$ (usually via estimating $\boldsymbol{\Theta}$) by an optimization of continuous variables, while the second stage involves a discrete optimization on one or several integers that index candidate models. This paper only considers the first stage where automatic model selection actually performs, though the second stage may be also considered to further improve the model selection performance with much more computing costs.

For GMM in Eq. (1), $H(p||q,\boldsymbol{\Theta},\boldsymbol{\Xi})$ takes a specific expression as given by Eq. (10) in Ref. [12], which is rewritten as below:

$$H(p||q,\boldsymbol{\Theta},\boldsymbol{\Xi}) = \sum_{i=1}^{k}\sum_{t=1}^{N} p(i|\boldsymbol{x}_t,\boldsymbol{\Theta}) \ln\left[\alpha_i G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1}\right)\right] + \sum_{i=1}^{k} R\left(h,\theta_i\right),$$

$$p(i|\boldsymbol{x}_t,\boldsymbol{\Theta}) = \frac{\alpha_i G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1}\right)}{\sum_{j=1}^{k} \alpha_j G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_j, \boldsymbol{T}_j^{-1}\right)},$$

$$R\left(h,\theta_j\right) = \ln\left[q\left(h\,|\boldsymbol{X}_N\right) q\left(\theta_j\right)\right] - \frac{1}{2}\mathrm{Tr}\left(h^2\boldsymbol{T}_j\right),$$

$$\theta_i = \{\alpha_i, \mu_i, \boldsymbol{T}_i\}, \quad (13)$$

which comes from Eq. (12) on the following BYY system:

$$Ying: \ q(\boldsymbol{X},\boldsymbol{R}) = q(\boldsymbol{X}|\boldsymbol{R})q(\boldsymbol{R}), \ q(\boldsymbol{R}) = q(\boldsymbol{Y}|\boldsymbol{\Theta})q(\boldsymbol{\Theta}),$$
$$q(\boldsymbol{Y}|\boldsymbol{\Theta}) = \prod_{t=1}^{N} q(\boldsymbol{y}_t|\boldsymbol{\Theta}),$$

$$
\begin{aligned}
& q(\boldsymbol{X}|\boldsymbol{R}) = \prod_{t=1}^{N} q(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{\Theta}), \\
Yang: \; & p(\boldsymbol{R}, \boldsymbol{X}) = p(\boldsymbol{R}|\boldsymbol{X}) p(\boldsymbol{X}), \\
& p(\boldsymbol{R}|\boldsymbol{X}) = p(\boldsymbol{\Theta}|\boldsymbol{X}) p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta}), \\
& p_h(\boldsymbol{X}) = G\left(\boldsymbol{X}|\boldsymbol{X}_N, h^2 \boldsymbol{I}\right), \\
& p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta}) = \prod_{t=1}^{N} \prod_{i=1}^{k} p(i|\boldsymbol{x}_t, \boldsymbol{\Theta})^{y_{it}}, \qquad (14)
\end{aligned}
$$

where $q(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{\Theta})$, $q(\boldsymbol{y}_t|\boldsymbol{\Theta})$ of the Ying machine are given by Eq. (2). For the Yang machine, $p_h(\boldsymbol{X})$ is a smoothed expression of a sample set $\boldsymbol{X}_N = \{\boldsymbol{x}_t\}$ with $p_0(\boldsymbol{X})$ at $h = 0$ being the empirical distribution. $p(\boldsymbol{R}|\boldsymbol{X})$ is designed as a functional of the Ying-machine according to the variety preservation (VP) principle, see Sect. 4.2 and especially Eq. (28) in Ref. [12]. One typical example is the Bayesian posterior of the Ying-machine. In this paper, $p(i|\boldsymbol{x}_t, \boldsymbol{\Theta})$ is designed via the Bayesian posterior by Eq. (13), while $p(\boldsymbol{\Theta}|\boldsymbol{X})$ is simply considered as a free structure that maximizes the harmony measure $H(p||q, \boldsymbol{\Xi})$ in Eq. (12), which makes the maximization of $H(p||q, \boldsymbol{\Xi})$ become equivalent to

$$
p(\boldsymbol{\Theta}|\boldsymbol{X}) = \delta(\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}), \; \hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} H(p||q, \boldsymbol{\Theta}, \boldsymbol{\Xi}). \qquad (15)
$$

Putting all the above settings into Eq. (12) leads us to Eq. (16). Details are referred to Sect. 3.1 of Ref. [12] and especially its Fig. 7 for a unified Ying-Yang learning procedure for implementing the maximization of $H(p||q, \boldsymbol{\Theta}, \boldsymbol{\Xi})$ via iterating the Yang-step and Ying-step alternatively.

Ignoring $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ and letting $h = 0$, the Ying-step has the same expression as the M-step in Eq. (4) of the EM algorithm, while the Yang-step is different from the E-step of the EM algorithm in that p_{it} by Eq. (3) is changed into

$$
\begin{aligned}
& p_{it} = p(i|\boldsymbol{x}_t) + \Delta_{it}, \; p(i|\boldsymbol{x}_t) = p\left(i|\boldsymbol{x}_t, \boldsymbol{\Theta}^{\text{old}}\right), \\
& \Delta_{it} = \Delta_{it}\left(\boldsymbol{\Theta}^{\text{old}}\right), \Delta_{it}(\boldsymbol{\Theta}) = p(i|\boldsymbol{x}_t, \boldsymbol{\Theta}) \delta_{it}(\boldsymbol{\Theta}), \\
& \delta_{it}(\boldsymbol{\Theta}) = \ln p(i|\boldsymbol{x}_t, \boldsymbol{\Theta}) - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t, \boldsymbol{\Theta}) \ln p(j|\boldsymbol{x}_t, \boldsymbol{\Theta}) \\
& \qquad = \ln\left[\alpha_i G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1}\right)\right] \\
& \qquad\quad - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t, \boldsymbol{\Theta}) \ln\left[\alpha_j G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_j, \boldsymbol{T}_j^{-1}\right)\right], \qquad (16)
\end{aligned}
$$

which actually modifies the M-step of the EM algorithm via δ_{it}. For a sample $\boldsymbol{x}_t$, $\delta_{it} > 0$ means that the ith component is better than the average of all the components to represent $\boldsymbol{x}_t$ and thus the ith component is further updated towards $\boldsymbol{x}_t$ with an enhanced strength $p_{it} > 0$. If $0 > \delta_{it} > -1$, i.e., the ith component is below the average for representing $\boldsymbol{x}_t$ but not too far away, the ith component is slightly updated towards $\boldsymbol{x}_t$ with a much reduced strength $p_{it} > 0$. Moreover, when $-1 > \delta_{it}$, we get $p_{it} < 0$ that reverses the direction to become de-learning, somewhat similar to updating the rival in RPCL learning that automatically determines k during parameter learning [13,13–15,31]. In fact, this nature of automatic model selection comes from the role of $q(\boldsymbol{Y}|\boldsymbol{\Theta})$, as previously introduced after Eq. (12).

To compare MML and VB that both base on an appropriate prior $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ to make model selection, we also include $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ for the BYY harmony learning. For simplicity, we still let $h = 0$ and consider

$$
\begin{aligned}
& q(\boldsymbol{\Theta}) \propto \prod_{i=1}^{k} [\alpha_i^{\frac{\rho-1}{2}} |\boldsymbol{T}_i|^{-\frac{i_T d}{2}}], \\
& R(h, \theta_j) = \frac{1}{2}[(\rho - 1) \ln \alpha_i - i_T d \ln |\boldsymbol{T}_i|], \qquad (17)
\end{aligned}
$$

which is put into Eq. (13). Then, simplified version of the unified Ying-Yang learning procedure in Fig. 7 in Ref. [12] is obtained for maximizing $H(p||q, \boldsymbol{\Xi})$. Given in Table 1, this simplification is shortly denoted as Algorithm BYY-Jef. For convenience, we use an indicator i_T, with $i_T = 1$ considering the Jeffreys prior on all parameters $\boldsymbol{\Theta}$, with $i_T = 0$ and $\rho = 0$ considering the Jeffreys prior merely on the mixing weights $\boldsymbol{\alpha}$ via removing from p_{it} a background effect $0.5/N$ in Table 1. Moreover, letting $i_T = 1$ imposes Jeffreys prior on the precision matrices via adding a background $0.5\rho/N$ to p_{it} and a diagonal background d/N. As Yang-step and Ying-step iterate, the ith component's mixing weight α_i will be pushed to zero and then discarded if it is extra. Thus, the number of Gaussians is automatically determined.

3.2 VB and VB-Jef algorithms

The Bayesian framework allows proper incorporation of prior knowledge on the model parameters. Bayesian model selection is implemented to choose a model complexity k (e.g., the number of Gaussian components of GMM) with the maximum marginal likelihood, which is obtained by integrating out the latent variables $\boldsymbol{Y}$ and the parameters $\boldsymbol{\Theta}$, i.e., $q(\boldsymbol{X}_N) = \int q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta}) q(\boldsymbol{\Theta}) \mathrm{d}\boldsymbol{Y} \mathrm{d}\boldsymbol{\Theta}$. However, the involved integration is usually very difficult. Variational Bayesian [19,20] tackles this difficulty via constructing a computable lower bound for the log marginal likelihood by means of variational methods, and an EM-like algorithm is employed to optimize this lower bound. More precisely, the lower bound is given as follows:

$$
\begin{aligned}
& J_{\text{VB}}(\boldsymbol{\Xi}^*, \boldsymbol{\Xi}) \\
& \quad = \int p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{\Xi}^*) \ln \frac{q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta}) q(\boldsymbol{\Theta}|\boldsymbol{\Xi})}{p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{\Xi}^*)} \mathrm{d}\boldsymbol{Y} \mathrm{d}\boldsymbol{\Theta} \\
& \quad = -\int p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{\Xi}^*) \ln \frac{p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{\Xi}^*)}{p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{X}_N, \boldsymbol{\Xi})} \mathrm{d}\boldsymbol{Y} \mathrm{d}\boldsymbol{\Theta} \\
& \qquad + \ln q(\boldsymbol{X}_N|\boldsymbol{\Xi}), \qquad (18)
\end{aligned}
$$

where $\boldsymbol{Y}$ represents all hidden variables, e.g., the label $\boldsymbol{y}$ in Eq. (2) for GMM, $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ is a prior on

Table 1 BYY-Jef for learning GMM, with $i_T = 1$ and $\rho = d + 0.5d(d+1)$ considering Jeffreys priors on both mixing weights and the precision matrices, and with $i_T = 0$ and $\rho = 0$ considering Jeffreys prior merely on mixing weights, and with $i_T = 0$ and $\widetilde{\gamma} = 0$ considering no priors, respectively

Initialization: Randomly initialize GMM with a large enough component number k; set $\tau = 0, i_\alpha = 1$ and the initial harmony measure $J_{\mathrm{BYY}}(\tau) = -\infty$;

repeat

Randomly pick a sample $\boldsymbol{x}_t$ from the dataset $\boldsymbol{X}_N$;

Yang-step: For $i = 1, 2, \ldots, k$, get p_{it} by Eq. (16);

Ying-step: Update $\alpha_i^{\mathrm{new}} = (1-\eta)\alpha_i^{\mathrm{old}} + \eta \frac{p_{it} + \widetilde{\gamma}}{\sum_{j=1}^k (p_{it} + \widetilde{\gamma})}$, and update $\{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^k$ by

$$\widetilde{\gamma} = \begin{cases} 0.5(\rho - 1)/N, & \text{BYY-Jef ,} \\ 0, & \text{BYY without priors;} \end{cases}$$

$$\boldsymbol{\mu}_i^{\mathrm{new}} = \boldsymbol{\mu}_i^{\mathrm{old}} + \eta p_{it}(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}});$$

$$\boldsymbol{T}_i^{-1\ \mathrm{new}} = \boldsymbol{S}_i^{\mathrm{new}} \boldsymbol{S}_i^{\mathrm{new\ T}},\ \boldsymbol{S}_i^{\mathrm{new}} = \mathbf{S}_i^{\mathrm{old}}(\boldsymbol{I}_d + \eta \boldsymbol{G}_i),$$

$$\boldsymbol{G}_i = p_{it}[\boldsymbol{T}_i^{\mathrm{old}}(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}})(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}})^{\mathrm{T}} - \boldsymbol{I}_d] + i_T \frac{d}{N} \boldsymbol{I}_d;$$

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\alpha_i \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another* $5N$ *runs have passed* **then** let $\tau = \tau + 1$; calculate $J_{\mathrm{BYY}}(\tau)$ by Eqs. (13) and (17);

until $J_{\mathrm{BYY}}(\tau) - J_{\mathrm{BYY}}(\tau - 1) < \epsilon J_{\mathrm{BYY}}(\tau - 1)$, *with* $\epsilon = 10^{-5}$;

parameters $\boldsymbol{\Theta}$ with hyper-parameters $\boldsymbol{\Xi}$, and $p(\boldsymbol{\Theta}, \boldsymbol{Y})$ is a variational posterior with hyper-parameters $\boldsymbol{\Xi}^*$ to approximate the exact Bayesian posterior $p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{X}_N, \boldsymbol{\Xi}) \propto q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta}) q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$. It follows from Eq. (18) that the lower bound $J_{\mathrm{VB}}(\boldsymbol{\Xi}^*, \boldsymbol{\Xi})$ is tight to $\ln q(\boldsymbol{X}_N|\boldsymbol{\Xi})$ when $p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{\Xi}^*) = p(\boldsymbol{\Theta}, \boldsymbol{Y}|\boldsymbol{X}_N, \boldsymbol{\Xi})$.

For computational convenience, $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ is usually chosen to be conjugate priors, and $p(\boldsymbol{\Theta}, \boldsymbol{Y})$ is usually assumed to be a factorized form $p(\boldsymbol{\Theta}, \boldsymbol{Y}) = p(\boldsymbol{Y}) \prod_i p(\boldsymbol{\Theta}_i)$ with $\boldsymbol{\Theta} = \cup_i \boldsymbol{\Theta}_i$, so that maximizing J_{VB} makes the variational posterior $p(\boldsymbol{\Theta})$ to be in the same form as the corresponding prior.

To the best of our knowledge, there is still no VB algorithm yet on GMM with the Jeffreys prior $q(\boldsymbol{\Theta})$ in Eq. (8). We develop one on the assumption that the variational posterior is factorized as follows:

$$\begin{aligned}
&p(\boldsymbol{\Theta}, \boldsymbol{Y}) = p(\boldsymbol{Y}) p(\boldsymbol{\Theta}), \quad p(\boldsymbol{Y}) = \prod_{t=1}^N p(\boldsymbol{y}_t), \\
&p(\boldsymbol{y}_t) = \prod_{i=1}^k {p_{it}}^{y_{it}}, \\
&p(\boldsymbol{\Theta}) = G(\mathrm{vec}(\boldsymbol{\Theta})|\mathrm{vec}(\boldsymbol{\Theta}^*), \boldsymbol{\Pi}), \\
&\text{with} \quad \boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} u(\boldsymbol{\Theta}), \quad u(\boldsymbol{\Theta}) = \ln q(\boldsymbol{X}_N, \boldsymbol{\Theta}), \\
&\qquad \boldsymbol{\Pi} = -\left[\frac{\partial^2 u(\boldsymbol{\Theta})}{\partial \mathrm{vec}(\boldsymbol{\Theta}) \partial \mathrm{vec}(\boldsymbol{\Theta})^{\mathrm{T}}} \right]^{-1}, \qquad (19)
\end{aligned}$$

where $p(\boldsymbol{y})$ shares the same form with $q(\boldsymbol{y}|\boldsymbol{\Theta})$ by Eq. (2) due to its conjugate nature. The inverse of the matrix $\boldsymbol{\Pi}$ is conceptually different from the Fisher information $\boldsymbol{I}(\boldsymbol{\Theta})$ in Eq. (5). More specifically, $\boldsymbol{I}(\boldsymbol{\Theta})$ concerns the expected Hessian of the log-*conditional*-distribution $\ln q(\boldsymbol{X}_N|\boldsymbol{\Theta})$, while the inverse of $\boldsymbol{\Pi}$ concerns the Hessian of the log-*joint*-distribution $\ln q(\boldsymbol{X}_N, \boldsymbol{\Theta})$. Because the Jeffreys prior $q(\boldsymbol{\Theta})$ by Eq. (17) is not a conjugate one, the exact variational posterior $p(\boldsymbol{\Theta})$ is difficult to compute and thus approximated by a Gaussian distribution around the apex $\boldsymbol{\Theta}^*$ [17]. Here and throughout this paper, vec$(\cdot)$ refers to the vectorization operator.

Since the quantity $u(\boldsymbol{\Theta})$ and the Hessian $\boldsymbol{\Pi}$ are still difficult to be tackled directly, we approximate $u(\boldsymbol{\Theta})$ by the following variational lower-bound $u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\})$:

$$u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\}) = \sum_{\boldsymbol{Y}} p(\boldsymbol{Y}) \ln \frac{q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta}) q(\boldsymbol{\Theta})}{p(\boldsymbol{Y})}, \quad (20)$$

and with $q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta}) = \prod_{t=1}^N q(\boldsymbol{x}_t, \boldsymbol{y}_t|\boldsymbol{\Theta})$, it follows from Eq. (2) that

$$\begin{aligned}
u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\}) = &\sum_{t=1}^N \sum_{i=1}^k p_{it} \ln \left[\alpha_i G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1}\right) \right] \\
&+ \sum_{i=1}^k \left[\frac{\rho - 1}{2} \ln \alpha_i - \frac{i_T d}{2} \ln |\boldsymbol{T}_i| \right] \\
&- \sum_{t=1}^N \sum_{i=1}^k p_{it} \ln p_{it} + \frac{k(\rho+1)}{2} \ln N. \quad (21)
\end{aligned}$$

Accordingly, the matrix $\boldsymbol{\Pi}$ in Eq. (19) is approximated by the following block-diagonal:

$$\begin{aligned}
\boldsymbol{\Pi}(\boldsymbol{\Theta}^*, \{p_{it}\}) \triangleq\ &\text{Block-Diag}[\boldsymbol{\Pi}^{\boldsymbol{\alpha}}, \boldsymbol{\Pi}^{\boldsymbol{\mu}_1}, \ldots, \boldsymbol{\Pi}^{\boldsymbol{\mu}_k}, \\
&\boldsymbol{\Pi}^{\boldsymbol{T}_1}, \ldots, \boldsymbol{\Pi}^{\boldsymbol{T}_k}], \\
\boldsymbol{\Pi}^{\boldsymbol{\alpha}} = &-\mathrm{diag}\left[\frac{\partial^2 u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}^{\mathrm{T}}} \right]^{-1}, \\
|\boldsymbol{\Pi}^{\boldsymbol{\alpha}}| \propto &\prod_{i=1}^k \frac{\alpha_i^2}{\sum_{t=1}^N p_{it} + (\rho - 1)/2}, \\
\boldsymbol{\Pi}^{\boldsymbol{\mu}_i} = &-\left[\frac{\partial^2 u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\})}{\partial \boldsymbol{\mu}_i \partial \boldsymbol{\mu}_i^{\mathrm{T}}} \right]^{-1}, \\
|\boldsymbol{\Pi}^{\boldsymbol{\mu}_i}| \propto &\left(\sum_{t=1}^N p_{it} \right)^{-d} |\boldsymbol{T}_i|^{-1}, \\
\boldsymbol{\Pi}^{\boldsymbol{T}_i} = &-\left[\frac{\partial^2 u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\})}{\partial \mathrm{vec}(\boldsymbol{T}_i) \partial \mathrm{vec}(\boldsymbol{T}_i)^{\mathrm{T}}} \right]^{-1},
\end{aligned}$$

$$|\mathbf{\Pi}^{\boldsymbol{T}_i}| \propto \left(\sum_{t=1}^{N} p_{it} - d\right)^{-d(d+1)/2} |\boldsymbol{T}_i|^{d+1}. \quad (22)$$

Putting all the above into Eq. (18), solving the VB problem is approximated as follows:

$$\begin{aligned}
&\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} u^{\mathrm{lb}}(\mathbf{\Theta}, \{p_{it}^*\}),\\
&\{p_{it}^*\} = \arg\max_{\{p_{it}\}} J_{\mathrm{VB}}^{\mathrm{Jef}}(\mathbf{\Theta}^*, \{p_{it}\}),\\
&J_{\mathrm{VB}}^{\mathrm{Jef}}(\mathbf{\Theta}^*, \{p_{it}\})\\
&= u^{\mathrm{lb}}(\mathbf{\Theta}^*, \{p_{it}\}) + \frac{1}{2}\ln|\mathbf{\Pi}(\mathbf{\Theta}^*, \{p_{it}\})| + \mathrm{const}\\
&= u^{\mathrm{lb}}(\mathbf{\Theta}^*, \{p_{it}\}) - \frac{1}{2}\sum_{i=1}^{k}\ln\left(\sum_{t=1}^{N} p_{it} + \frac{\rho-1}{2}\right)\\
&\quad - \frac{i_T}{2}\sum_{i=1}^{k}\left[d\ln\left(\sum_{t=1}^{N} p_{it}\right) + \frac{d(d+1)}{2}\ln\left(\sum_{t=1}^{N} p_{it} - d\right)\right]\\
&\quad + \sum_{i=1}^{k}\left[\ln\alpha_i + \frac{i_T d}{2}\ln|\boldsymbol{T}_i|\right] + \mathrm{const}, \qquad (23)
\end{aligned}$$

where we consider Jeffreys prior merely on mixing weights by setting $i_T = 0$ and $\rho = 0$, and further consider Jeffreys priors on both mixing weights and Gaussian parameters by setting $i_T = 1$ and $\rho = d + 0.5d(d+1)$.

The corresponding VB algorithm, shortly denoted as VB-Jef, is listed in Table 2. Moreover, when $i_T = 0$ and fixing $\rho = 0$, we have its special case VB-Jef($\boldsymbol{\alpha}$) with Jeffreys prior merely on mixing weights. Also, VB-Jef($\boldsymbol{\alpha}$) is shared by MML-Jef [18] via some options (see the next subsection for details). As briefly summarized in Table 3, VB-Jef has the same M-step as MAP-Jef, which extends the M-step of the EM algorithm by considering the Jeffreys prior $q(\mathbf{\Theta})$ that affects updating $\boldsymbol{\alpha}$ by $s_i = n_i - 0.5(1-\rho)$ and also updating the precision matrices by a regularization $n_0 = d$ in Table 2. Comparing with the E-step of MAP-Jef, VB-Jef has an additional term $\frac{1}{2}\ln|\mathbf{\Pi}(\mathbf{\Theta}^*, \{p_{it}\})|$ in its objective function, which makes its E-step become different from the E-step of both MAP-Jef and the standard EM, with p_{it} obtained by Eq. (3) via a modified α_i^p. Also, this p_{it} differs from the Yang-step by Eq. (16) for Algorithm BYY-Jef. During learning by Table 2, the ith Gaussian component is discarded if $\alpha_i \to 0$, and thus the number of Gaussians is automatically determined after convergence.

3.3 MML and MML-Jef algorithms

Proposed by Wallace and Boulton [22,23], MML is an information theoretic restatement of Occam's Razor. Among different models, the one generating the shortest overall code length is regarded most likely to be

Table 2 Learning GMM with Jeffreys prior: MML-Jef [18] with $J(\tau) = J_{\mathrm{MML}}^{\mathrm{Jef}}(\mathbf{\Theta})$ by Eq. (25), and VB-Jef with $J(\tau) = J_{\mathrm{VB}}^{\mathrm{Jef}}$ by Eq. (23) plus its special case VB-Jef($\boldsymbol{\alpha}$) with Jeffreys prior merely on mixing weights

Initialization: Randomly initialize GMM with a large enough component number k; set $\tau = 0$ and the initial objective $J(\tau) = -\infty$;

repeat

E-step: For $i = 1, 2, \ldots, k$, get

$$\alpha_i^p = \begin{cases} \alpha_i^{\mathrm{old}}, & \text{MML-Jef},\\ \alpha_i^{\mathrm{old}} \exp\left[-\dfrac{1}{2(N\alpha_i^{\mathrm{old}} - 1/2)}\right], & \text{VB-Jef}(\boldsymbol{\alpha}),\\ \alpha_i^{\mathrm{old}} \exp\left[-\dfrac{1}{2(N\alpha_i^{\mathrm{old}} + (\rho-1)/2)} - \dfrac{d}{2N\alpha_i^{\mathrm{old}}} - \dfrac{d(d+1)}{4(N\alpha_i^{\mathrm{old}} - d)}\right], & \text{VB-Jef}; \end{cases}$$

Then, get p_{it} and n_i by Eq. (3);

M-step: Get

$$s_i = n_i - \frac{1}{2}(1 - \delta s_i),\ \delta s_i = \begin{cases} 1-\rho,\ \rho = d + d(d+1)/2, & \text{MML-Jef},\\ \rho, & \text{VB-Jef and VB-Jef}(\boldsymbol{\alpha}), \end{cases}$$

$$n_0 = \begin{cases} 0, & \text{MML-Jef and VB-Jef}(\boldsymbol{\alpha}),\\ d, & \text{VB-Jef}; \end{cases}$$

Update $\boldsymbol{\alpha} \cup \{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^k$ by Eq. (4);

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\alpha_i \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another five runs have passed* **then** let $\tau = \tau + 1$; calculate the objective $J(\tau)$;

until $J(\tau) - J(\tau - 1) < \epsilon J(\tau - 1)$, *with* $\epsilon = 10^{-5}$;

Table 3 The EM implementations of MAP-Jef, VB-Jef, and MML-Jef: common and different points

	E-step: $\{p_{it}^*\} = \arg\max_{\{p_{it}\}} f^{\mathrm{E}}(\mathbf{\Theta}^*, \{p_{it}\})$	M-step: $\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} f^{\mathrm{M}}(\mathbf{\Theta}, \{p_{it}^*\})$
MAP-Jef	$f^{\mathrm{E}} = u^{\mathrm{lb}}(\mathbf{\Theta}^*, \{p_{it}\})$	$f^{\mathrm{M}} = u^{\mathrm{lb}}(\mathbf{\Theta}, \{p_{it}^*\})$
VB-Jef	$f^{\mathrm{E}} = u^{\mathrm{lb}}(\mathbf{\Theta}^*, \{p_{it}\}) + \frac{1}{2}\ln\lvert\mathbf{\Pi}(\mathbf{\Theta}^*, \{p_{it}\})\rvert$	$f^{\mathrm{M}} = u^{\mathrm{lb}}(\mathbf{\Theta}, \{p_{it}^*\})$
MML-Jef	$f^{\mathrm{E}} = u^{\mathrm{lb}}(\mathbf{\Theta}^*, \{p_{it}\})$	$f^{\mathrm{M}} = u^{\mathrm{lb}}(\mathbf{\Theta}, \{p_{it}^*\}) - \frac{1}{2}\ln\lvert\boldsymbol{I}(\mathbf{\Theta})\rvert$

the best, where the code length is $\text{Length}(\boldsymbol{X},\boldsymbol{\Theta}) = \text{Length}(\boldsymbol{X}|\boldsymbol{\Theta}) + \text{Length}(\boldsymbol{\Theta})$, which consists of a statement of the model and a statement of data encoded by using that model. Stemming from a similar philosophy, MDL [11] is another closely related model selection criterion that is commonly used in a two-stage procedure too [11,36]. Detailed comparisons on MML and MDL are referred to Refs. [23,36–38].

As introduced in Refs. [22,23], MML seeks a model $\boldsymbol{\Theta}$ by maximizing the objective function:

$$J_{\text{MML}}(\boldsymbol{\Theta}) = \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) + \ln q(\boldsymbol{\Theta}) - \frac{1}{2}\ln|\boldsymbol{I}(\boldsymbol{\Theta})|, \quad (24)$$

where $\boldsymbol{I}(\boldsymbol{\Theta})$ is the Fisher information matrix given in Eq. (5). It is mathematically equivalent to an MAP method given a prior that is proportional to $q(\boldsymbol{\Theta})/|\boldsymbol{I}(\boldsymbol{\Theta})|^{1/2}$.

Using the Jeffreys prior $q(\boldsymbol{\Theta}) \propto \sqrt{|\boldsymbol{I}(\boldsymbol{\Theta})|}$ in Eq. (5), $J_{\text{MML}}(\boldsymbol{\Theta})$ in Eq. (24) degenerates to be the ML principle since $\ln q(\boldsymbol{\Theta}) - \frac{1}{2}\ln|\boldsymbol{I}(\boldsymbol{\Theta})| = 0$. To avoid this situation, Figueiredo and Jain [18] considered

$$\begin{aligned} J^{\text{Jef}}_{\text{MML}}(\boldsymbol{\Theta}) &= \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) \\ &\quad - \frac{1}{2}\sum_{i=1}^{k}(\ln\alpha_i + d\ln|\boldsymbol{T}_i|) - \frac{1}{2}\ln|\boldsymbol{I}(\boldsymbol{\Theta})| \\ &\approx \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) - \frac{\rho}{2}\sum_{i=1}^{k}\ln\alpha_i - \frac{k(\rho+1)}{2}\ln N, \end{aligned} \quad (25)$$

which comes from Eq. (24) with a decoupled approximation $q(\boldsymbol{\Theta}) \propto \prod_{i=1}^{k}\left[\alpha_i^{-\frac{1}{2}}|\boldsymbol{T}_i|^{-\frac{d}{2}}\right]$ as a counterpart of $\boldsymbol{I}_c(\boldsymbol{\Theta})$ by Eq. (6) as $\boldsymbol{I}(\boldsymbol{\Theta})$. The MML algorithm in Ref. [18], shortly denoted as MML-Jef and restated in Table 2, was derived to equivalently implement the maximization of $J^{\text{Jef}}_{\text{MML}}(\boldsymbol{\Theta})$.

In the special case of considering the Jeffreys prior merely on $\boldsymbol{\alpha}$, we have

$$q(\boldsymbol{\Theta}) \propto \prod_{i=1}^{k}\alpha_i^{-\frac{1}{2}}, \ \boldsymbol{I}_c(\boldsymbol{\Theta}) \propto \prod_{i=1}^{k}\alpha_i^{-1}, \quad (26)$$

which also makes $J_{\text{MML}}(\boldsymbol{\Theta})$ in Eq. (24) degenerate to be the ML principle. Correspondingly, the algorithm MML-Jef in Table 2, denoted by MML-Jef($\boldsymbol{\alpha}$), degenerates to implement the ML learning.

4 Algorithms with Dirichlet prior and DNW priors

4.1 Algorithms BYY-Dir($\boldsymbol{\alpha}$), VB-Dir($\boldsymbol{\alpha}$), and MML-Dir($\boldsymbol{\alpha}$)

We consider the prior on the mixing weights $q(\boldsymbol{\alpha})$ replaced by the Dirichlet prior $\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi)$ in Eq. (10), under the framework of VB, MML, and BYY respectively. Different from the Jeffreys prior, the Dirichlet prior has a set of hyper-parameters $\boldsymbol{\Xi} = \{\boldsymbol{\lambda},\xi\}$, which are unknown either to be specified before learning or to be simultaneously determined during learning.

We start at BYY-Dir($\boldsymbol{\alpha}$). Although there is a general guideline for implementing the maximization of $H(p||q,\boldsymbol{\Xi},\boldsymbol{\Xi}^*)$ with priors on the parameters [12,17], there are still no algorithm developed for implementing BYY harmony learning with either the Dirichlet prior or the DNW prior yet. Here, we propose such an algorithm featured by an alternative decomposition of the Yang machine part $p(\boldsymbol{R}|\boldsymbol{X})$ that differs from its counterpart in Eq. (14). Specifically, we consider a Bayesian Ying-Yang system with its Ying machine still given by Eq. (14), together with the Dirichlet prior

$$q(\boldsymbol{\Theta}) = q(\boldsymbol{\alpha}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi), \quad (27)$$

while $p(\boldsymbol{R}|\boldsymbol{X})$ in Eq. (14) is replaced by

$$\begin{aligned} &p(\boldsymbol{R}|\boldsymbol{X}) = p(\boldsymbol{\Theta}|\boldsymbol{Y},\boldsymbol{X}_N)p(\boldsymbol{Y}|\boldsymbol{X}), \\ &p(\boldsymbol{\Theta}|\boldsymbol{Y},\boldsymbol{X}_N) \approx p(\boldsymbol{\Theta}|\boldsymbol{Y}), \\ &p(\boldsymbol{\Theta}|\boldsymbol{Y}) = p(\boldsymbol{\alpha}|\boldsymbol{Y}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}^*,\xi+N), \\ &p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{t=1}^{N}\prod_{i=1}^{k}p(i|\boldsymbol{x}_t)^{y_{it}}. \end{aligned} \quad (28)$$

Still in accordance with the variety preservation principle (see Sect. 4.2 in Ref. [12]), $p(\boldsymbol{\alpha}|\boldsymbol{Y})$ comes from the Bayesian posterior of $\prod_{t=1}^{N}q(\boldsymbol{y}_t|\boldsymbol{\Theta})q(\boldsymbol{\alpha})$. With the help of the conjugate property, $p(\boldsymbol{\alpha}|\boldsymbol{Y})$ is also a Dirichlet $\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}^*,\xi+N)$ with a degree $(\xi+N)$. Moreover, $p(i|\boldsymbol{x}_t)$ comes from considering the Bayesian posterior

$$\begin{aligned} p(i|\boldsymbol{x}) &\propto \int \alpha_i G\left(\boldsymbol{x}|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1}\right)q(\alpha_i)\mathrm{d}\alpha_i \\ &= \lambda_i G\left(\boldsymbol{x}|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1}\right). \end{aligned} \quad (29)$$

It follows from Eq. (12) that the harmony measure is rewritten as

$$\begin{aligned} &H^{\text{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta},\boldsymbol{\Xi}) = \\ &\quad \sum_{i=1}^{k}\sum_{t=1}^{N}p(i|\boldsymbol{x}_t,\boldsymbol{\Theta},\boldsymbol{\lambda})\ln\left[\alpha_i^p(\xi+N,\boldsymbol{\lambda}^*)G\left(\boldsymbol{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1}\right)\right] \\ &\quad + \sum_{i=1}^{k}R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i), \\ &p(i|\boldsymbol{x}_t,\boldsymbol{\Theta},\boldsymbol{\lambda}) = \frac{\lambda_i G(\boldsymbol{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})}{\sum_j\lambda_j G(\boldsymbol{x}_t|\boldsymbol{\mu}_j,\boldsymbol{T}_j^{-1})}, \\ &\alpha_i^p(\xi+N,\boldsymbol{\lambda}^*) = \exp[\delta\Psi((\xi+N)\lambda_i^*)], \\ &R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i) = \frac{1}{k}\ln\Gamma(\xi) - \ln\Gamma(\xi\lambda_i) \\ &\qquad\qquad +(\xi\lambda_i-1)\delta\Psi((\xi+N)\lambda_i^*), \\ &\delta\Psi((\xi+N)\lambda_i^*) = \Psi((\xi+N)\lambda_i^*) - \Psi(\xi+N), \end{aligned} \quad (30)$$

where $\boldsymbol{\Xi} = \{\boldsymbol{\lambda}^*,\xi,\boldsymbol{\lambda}\}$, and $\Psi(x) = (\mathrm{d}/\mathrm{d}x)\ln\Gamma(x)$ is the digamma function. This above $H^{\text{Dir}(\boldsymbol{\alpha})}$ is maximized via a gradient based updating algorithm summarized in Table 4, which is shortly denoted as BYY-Dir($\boldsymbol{\alpha}$). Being different from the one in Table 1, the H-step is added for updating the hyper-parameters $\boldsymbol{\Xi}$ via maximizing

$H^{\mathrm{Dir}(\boldsymbol{\alpha})}$ with respect to $\boldsymbol{\Xi}$. Instead of α_i, here λ_i is pushed towards zero during learning and then the ith Gaussian component is discarded, which performs automatic model selection on determining the number of Gaussian components. Readers are referred to Appendix A for the detailed derivations about $H^{\mathrm{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta}, \boldsymbol{\Xi})$ and the Algorithm BYY-Dir($\boldsymbol{\alpha}$).

We observe a similar format to $H(p||q, \boldsymbol{\Theta}, \boldsymbol{\Xi})$ in Eq. (13) but with three different points. First, α_i of $p(i|\boldsymbol{x}_t, \boldsymbol{\Theta})$ in Eq. (13) is replaced by λ_i of $p(i|\boldsymbol{x}_t, \boldsymbol{\Theta}, \boldsymbol{\lambda})$ in Eq. (30). Second, α_i of $\ln[\alpha_i G(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1})]$ in Eq. (13) is replaced by α_i^p in Eq. (30) that is a parametric function of ξ, λ_i^*. Third, the regularization term $R(h, \theta_j)$ in Eq. (13) is replaced by $R_{(\xi+N,\lambda_i^*)}(\xi, \lambda_i)$ in Eq. (30) that provides certain regularizing role on learning $\xi, \lambda_i^*, \lambda_i$.

Next, we introduce VB-Dir($\boldsymbol{\alpha}$). With the following pair of conjugate Dirichlet priors:

$$q(\boldsymbol{\alpha}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}, \xi),\ p(\boldsymbol{\alpha}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}^*, \xi^*), \tag{31}$$

where $\{\boldsymbol{\lambda}^*, \xi^*\}$ are posterior hyper-parameters. Accordingly, the variational lower bound J_{VB} in Eq. (18) becomes

$$\begin{aligned} &J_{\mathrm{VB}}^{\mathrm{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta}, \boldsymbol{\Xi}) \\ &= \int \sum_{\boldsymbol{Y}} p(\boldsymbol{\alpha}) p(\boldsymbol{Y}) \ln \frac{q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\alpha}) q(\boldsymbol{\alpha})}{p(\boldsymbol{Y}) p(\boldsymbol{\alpha})} \mathrm{d}\boldsymbol{\alpha} \\ &= \sum_{t=1}^{N} \sum_{i=1}^{k} p_{it} \{\ln[\alpha_i^p(\xi^*, \boldsymbol{\lambda}^*) G(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{T}_i^{-1})] - \ln p_{it}\} \\ &\quad + \sum_{i=1}^{k} [R_{(\xi^*,\lambda_i^*)}(\xi, \lambda_i) - R_{(\xi^*,\lambda_i^*)}(\xi^*, \lambda_i^*)], \end{aligned} \tag{32}$$

where $\alpha_i^p(\xi^*, \boldsymbol{\lambda}^*)$ and $R_{(\xi^*,\lambda_i^*)}(\xi, \lambda_i)$ take the same forms as in Eq. (30). Maximizing this objective leads to a VB algorithm listed also in Table 5, shortly denoted as VB-Dir($\boldsymbol{\alpha}$).

Being different from the algorithm in Table 2, not only α_i^* in the E-step depends on the prior hyper-parameters $\boldsymbol{\Xi}$, but also there is the H-step added in Table 5 that updates the hyper-parameters $\boldsymbol{\Xi}$, via the gradient of $J_{\mathrm{VB}}^{\mathrm{Dir}(\boldsymbol{\alpha})}$ by Eq. (32) with respect to $\boldsymbol{\Xi}$. Moreover, for both VB-Dir($\boldsymbol{\alpha}$) and MML-Dir($\boldsymbol{\alpha}$) , the ith Gaussian component is discarded if $\lambda_i \to 0$. That is, automatic model selection on the number of Gaussian components is made also via the prior hyper-parameters instead of checking whether $\alpha_i \to 0$.

Finally, we move to MML-Dir($\boldsymbol{\alpha}$). With the Dirichlet prior $q(\boldsymbol{\Theta}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}, \xi)$, the objective MML criterion by Eq. (24) becomes

$$\begin{aligned} J_{\mathrm{MML}}^{\mathrm{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta}) &= \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) + \sum_{i=1}^{k} (\xi\lambda_i - 1) \ln \alpha_i \\ &\quad + \ln \Gamma(\xi) - \sum_{i=1}^{k} \ln \Gamma(\xi\lambda_i) - \frac{1}{2} \ln |\boldsymbol{I}(\boldsymbol{\alpha})|, \end{aligned} \tag{33}$$

where $|\boldsymbol{I}(\boldsymbol{\alpha})| = |N\boldsymbol{I}_c(\boldsymbol{\alpha})| = N^k \prod_{i=1}^{k} \alpha_i^{-1}$ as given by

Table 4 BYY-Dir($\boldsymbol{\alpha}$) algorithm for GMM with Dirichlet prior on mixing weights

Initialization: Randomly initialize GMM with a large enough component number k; set $\tau = 0$ and the initial harmony measure $J_{\mathrm{BYY}}(\tau) = -\infty$;

repeat

Randomly pick a sample $\boldsymbol{x}_t$ from the dataset $\boldsymbol{X}_N$;

Yang-step: Get $p(i|\boldsymbol{x}_t) = p(i|\boldsymbol{x}_t, \boldsymbol{\Theta}^{\mathrm{old}}, \boldsymbol{\lambda}^{\mathrm{old}})$ and $\alpha_i = \alpha_i^p(\xi^{*\mathrm{old}}, \boldsymbol{\lambda}^{*\mathrm{old}})$ by Eq. (30);

Further get $\pi_{it} = \ln[\alpha_i G(\boldsymbol{x}_t|\boldsymbol{\mu}_i^{\mathrm{old}}, \boldsymbol{T}_i^{\mathrm{old}\ -1})]$ and

$$\delta_{it} = \pi_{it} - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t)\pi_{jt},\ p_{it} = p(i|\boldsymbol{x}_t)(1 + \delta_{it});$$

Ying-step: Denote $\xi^{*\mathrm{old}} = \xi^{\mathrm{old}} + N$, and update the hyper-parameters $\boldsymbol{\lambda}^*$:

$$\lambda_i^{*\mathrm{new}} = \frac{\lambda_i^{*\mathrm{old}} + \eta\Delta\lambda_i(\boldsymbol{\Xi}^{\mathrm{old}})}{\sum_{j=1}^{k}(\lambda_i^{*\mathrm{old}} + \Delta\lambda_j(\boldsymbol{\Xi}^{\mathrm{old}}))},\ \Delta\lambda_i(\boldsymbol{\Xi}) = \varsigma_{it}\xi^*\lambda_i^*[\Psi'(\xi^*\lambda_i^*) - \Psi'(\xi^*)],$$

where $\varsigma_{it} = p(i|\boldsymbol{x}_t) + (\xi^{\mathrm{old}}\lambda_i^{\mathrm{old}} - 1)/N$, and $\Psi'(x) = (\mathrm{d}/\mathrm{d}x)\Psi(x)$ is the trigamma function;

Then, update the parameters $\{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^k$ by

$$\begin{aligned} &\boldsymbol{\mu}_i^{\mathrm{new}} = \boldsymbol{\mu}_i^{\mathrm{old}} + \eta p_{it}(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}}), \\ &\boldsymbol{T}_i^{-1\ \mathrm{new}} = \boldsymbol{S}_i^{\mathrm{new}} \boldsymbol{S}_i^{\mathrm{new}\ \mathrm{T}},\ \boldsymbol{S}_i^{\mathrm{new}} = \boldsymbol{S}_i^{\mathrm{old}}(\boldsymbol{I}_d + \eta\boldsymbol{G}_i), \\ &\boldsymbol{G}_i = p_{it}[\boldsymbol{T}_i^{\mathrm{old}}(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}})(\boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}})^{\mathrm{T}} - \boldsymbol{I}_d]; \end{aligned}$$

H-step: Get $\nu_i = \Psi(\xi^{\mathrm{old}}) - \Psi(\xi^{\mathrm{old}}\lambda_i^{\mathrm{old}}) - \Psi(\xi^*) + \Psi(\xi^*\lambda_i^{*\ \mathrm{old}})$ and update :

$$\begin{aligned} &\lambda_i^{\mathrm{new}} = (\lambda_i^{\mathrm{old}} + \eta\delta\lambda_i)/\textstyle\sum_{j=1}^{k}(\lambda_j^{\mathrm{old}} + \eta\delta\lambda_j), \quad \delta\lambda_i = p(i|\boldsymbol{x}_t)\delta_{it} + \frac{\xi^{\mathrm{old}}}{N}\upsilon_i, \\ &\xi^{\mathrm{new}} = \xi^{\mathrm{old}} + \eta\delta\xi, \quad \delta\xi = \textstyle\sum_{i=1}^{k}\{\lambda_i^{\mathrm{old}}\upsilon_i + N\varsigma_{it}[\lambda_i^{*\mathrm{old}}\Psi'(\xi^{*\mathrm{old}}\lambda_i^{*\mathrm{old}}) - \Psi'(\xi^{*\mathrm{old}})]\}; \end{aligned}$$

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\lambda_i \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another* $5N$ *runs have passed* **then** let $\tau = \tau + 1$; calculate $J_{\mathrm{BYY}}(\tau) = H^{\mathrm{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta}^{\mathrm{new}}, \boldsymbol{\Xi}^{\mathrm{new}})$ by Eq. (30);

until $J_{\mathrm{BYY}}(\tau) - J_{\mathrm{BYY}}(\tau - 1) < \epsilon J_{\mathrm{BYY}}(\tau - 1)$, *with* $\epsilon = 10^{-5}$;

Table 5 Learning GMM with Dirichlet prior on mixing weights : MML-Dir($\boldsymbol{\alpha}$) with $J_{\mathrm{MML}}(\tau)$ by Eq. (33) versus VB-Dir($\boldsymbol{\alpha}$) with $J_{\mathrm{VB}}(\tau)$ by Eq. (32)

Initialization: Randomly initialize GMM with a large enough component number k; set $\tau = 0$ and the initial objective $J(\tau) = -\infty$;

repeat

E-step: Let $\xi^* = \xi^{\mathrm{old}} + N$ and get

$$\alpha_i^p = \begin{cases} \alpha_i^{\mathrm{old}}, & \text{MML-Dir}(\boldsymbol{\alpha}), \\ \exp[\Psi(\xi^* \lambda_i^{*\mathrm{old}}) - \Psi(\xi^*)], & \text{VB-Dir}(\boldsymbol{\alpha}); \end{cases}$$

Then, get p_{it} and n_i by Eq. (3);

M-step: For VB-Dir($\boldsymbol{\alpha}$), get $\lambda_i^{*\mathrm{new}} = \dfrac{\xi^{\mathrm{old}} \lambda_i^{\mathrm{old}} + n_i}{\xi^*}$;

For MML-Dir($\boldsymbol{\alpha}$), get $\alpha_i^{\mathrm{new}} = \dfrac{s_i}{\sum_{j=1}^k s_j}$ with $s_i = n_i + \xi^{\mathrm{old}} \lambda_i^{\mathrm{old}} - \dfrac{1}{2}$;

Update $\{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^k$ by Eq. (4) with $n_0 = 0$;

H-step: Update the prior hyper-parameters $\{\boldsymbol{\lambda}, \xi\}$ as follows:

$$\lambda_i^{\mathrm{new}} = \frac{\lambda_i^{\mathrm{old}} + \eta \delta \lambda_i}{\sum_{j=1}^k (\lambda_j^{\mathrm{old}} + \eta \delta \lambda_j)},$$

$$\delta \lambda_i = \begin{cases} \ln \alpha_i^{\mathrm{old}} - \Psi(\xi^{\mathrm{old}} \lambda_i^{\mathrm{old}}) + \Psi(\xi^{\mathrm{old}}), & \text{MML-Dir}(\boldsymbol{\alpha}), \\ \Psi(\xi^{*\mathrm{old}} \lambda_i^{*\mathrm{old}}) - \Psi(\xi^{*\mathrm{old}}) - \Psi(\xi^{\mathrm{old}} \lambda_i^{\mathrm{old}}) + \Psi(\xi^{\mathrm{old}}), & \text{VB-Dir}(\boldsymbol{\alpha}), \end{cases}$$

$$\xi^{\mathrm{new}} = \xi^{\mathrm{old}} + \eta \delta \xi, \quad \delta \xi = \sum_{i=1}^k \lambda_i^{\mathrm{old}} \delta \lambda_i;$$

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\lambda_i \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another* 5 *runs have passed* **then** let $\tau = \tau + 1$; calculate the variational function as $J(\tau)$;

until $J(\tau) - J(\tau - 1) < \epsilon J(\tau - 1)$, *with* $\epsilon = 10^{-5}$;

Eqs. (6) and (7). An algorithm is developed in Table 5 to maximize $J_{\mathrm{MML}}^{\mathrm{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta})$, shortly denoted as MML-Dir($\boldsymbol{\alpha}$). Its E-step is the same as the E-step of the EM algorithm given in Sect. 2.1, while its M-step is similar to the one in Table 2 with s_i extended to include the prior hyper-parameters $\boldsymbol{\lambda}$ and ξ in consideration. Moreover, $J_{\mathrm{MML}}^{\mathrm{Dir}(\boldsymbol{\alpha})}$ consists of unknown hyper-parameters $\boldsymbol{\Xi}$, the H-step is added in Table 5 to adjust the prior hyper-parameters $\boldsymbol{\Xi}$ by a gradient method to maximize $J_{\mathrm{MML}}^{\mathrm{Dir}(\boldsymbol{\alpha})}(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Xi}$.

4.2 Algorithms with DNW priors

We proceed to consider the DNW priors in Eq. (9) on all the parameters. For BYY harmony learning, we have the Ying machine still given by Eq. (14), and the Yang machine still given by Eq. (28) but with the following modifications:

$$p(\boldsymbol{\alpha}|\boldsymbol{Y}) = \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}^*, \xi + N),$$
$$p(\boldsymbol{\Theta}|\boldsymbol{Y}) = p(\boldsymbol{\alpha}|\boldsymbol{Y}) \prod_{i=1}^k \left[G\left(\boldsymbol{\mu}_i | \boldsymbol{m}_i^*, \frac{\boldsymbol{T}_i^{-1}}{\beta + \sum_{t=1}^N y_{it}}\right) \cdot \mathcal{W}\left(\boldsymbol{T}_i | \boldsymbol{\Phi}_i^*, \gamma + \sum_{t=1}^N y_{it}\right)\right]. \tag{34}$$

The conjugate posteriori $p(\boldsymbol{\Theta}|\boldsymbol{Y})$ comes from the Bayesian posterior of $\prod_{t=1}^N q(\boldsymbol{y}_t|\boldsymbol{\Theta}) q(\boldsymbol{\Theta})$, in accordance with the variety preservation principle (see Sect. 4.2 in Ref. [12]). Moreover, $p(i|\boldsymbol{x}_t)$ comes from considering the Bayesian posterior

$$p(i|\boldsymbol{x}) \propto \int \alpha_i G(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{T}_i) q(\boldsymbol{\theta}_i|\boldsymbol{\Xi}) \mathrm{d}\boldsymbol{\theta}_i = \lambda_i \mathcal{T}(\boldsymbol{x}|\tau, \boldsymbol{m}_i, \boldsymbol{\Upsilon}), \tag{35}$$

where $\mathcal{T}(\boldsymbol{x}|\tau, \boldsymbol{m}_i, \boldsymbol{\Upsilon})$ is a multivariate Student's T-distribution with a degree-of-freedom τ, a mean $\boldsymbol{m}_i$ and a scatter matrix $\boldsymbol{\Upsilon}$ [39], that is, we have

$$\mathcal{T}(\boldsymbol{x}|\tau, \boldsymbol{m}_i, \boldsymbol{\Upsilon}) = \frac{\Gamma(\frac{\tau+d}{2})}{(\pi\tau)^{\frac{d}{2}} \Gamma(\frac{\tau}{2})} \cdot \frac{1}{|\boldsymbol{\Upsilon}|^{\frac{1}{2}} [1 + \frac{1}{\tau}(\boldsymbol{x} - \boldsymbol{m}_i)^{\mathrm{T}} \boldsymbol{\Upsilon}^{-1} (\boldsymbol{x} - \boldsymbol{m}_i)]^{\frac{\tau+d}{2}}},$$
$$\text{with} \quad \tau = \gamma - d + 1, \ \boldsymbol{\Upsilon} = \frac{\beta + 1}{\beta \tau} \boldsymbol{\Phi}. \tag{36}$$

Putting the above terms into Eq. (12), the harmony measure becomes

$$H^{\mathrm{DNW}}(\boldsymbol{\Xi}) = \int \sum_{\boldsymbol{Y}} p(\boldsymbol{\Theta}|\boldsymbol{Y}) p(\boldsymbol{Y}|\boldsymbol{X}) p(\boldsymbol{X}) \ln[q(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{\Theta}) q(\boldsymbol{Y}|\boldsymbol{\Theta}) q(\boldsymbol{\Theta}|\boldsymbol{\Xi})] \mathrm{d}\boldsymbol{X} \mathrm{d}\boldsymbol{\Theta} > H^{\mathrm{DNW}}(\boldsymbol{\Xi}, \boldsymbol{\Xi}^*),$$
$$H^{\mathrm{DNW}}(\boldsymbol{\Xi}, \boldsymbol{\Xi}^*) = \sum_{i=1}^k \sum_{t=1}^N p(i|\boldsymbol{x}_t, \boldsymbol{\Xi}, \boldsymbol{\lambda}) \ln \left[\alpha_i^p (\xi + N, \boldsymbol{\lambda}^*) G\left(\boldsymbol{x}_t | \boldsymbol{m}_i^*, \frac{\boldsymbol{\Phi}_i^*}{\gamma_i^* + 1 - p(i|\boldsymbol{x}_t, \boldsymbol{\Xi}, \boldsymbol{\lambda})}\right)\right]$$

$$+\sum_{i=1}^{k}[R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i)-R(\Xi_i)]+R(\Phi,\beta,\gamma)-R(k)+\text{const},$$

$$p(i|\boldsymbol{x}_t,\Xi,\boldsymbol{\lambda})=\frac{\lambda_i[1+\frac{\beta}{\beta+1}(\boldsymbol{x}_t-\boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{\Phi}^{-1}(\boldsymbol{x}_t-\boldsymbol{m}_i)]^{-\frac{\gamma+1}{2}}}{\sum_{j=1}^{k}\lambda_j[1+\frac{\beta}{\beta+1}(\boldsymbol{x}_t-\boldsymbol{m}_j)^{\mathrm{T}}\boldsymbol{\Phi}^{-1}(\boldsymbol{x}_t-\boldsymbol{m}_j)]^{-\frac{\gamma+1}{2}}},$$

$$R(\Xi_i)=\frac{1}{2}\left(d\sum_{t=1}^{N}\ln(\gamma_i^*+1-p(i|\boldsymbol{x}_t,\Xi,\boldsymbol{\lambda}))\right)+\frac{1}{2}(\gamma-d)\ln|\boldsymbol{\Phi}_i^*|$$
$$+\frac{1}{2}\beta\gamma_i^*(\boldsymbol{m}_i-\boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i-\boldsymbol{m}_i^*)+\frac{1}{2}\gamma_i^*\mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}),$$

$$R(\Phi,\beta,\gamma)=\frac{k}{2}(\gamma\ln|\boldsymbol{\Phi}|+d\ln\beta)-\sum_{j=1}^{d}\left[k\ln\Gamma\left(\frac{\gamma+1-j}{2}\right)-\frac{N+k(\gamma-d)}{2}\Psi\left(\frac{\gamma+1-j}{2}\right)\right],$$

$$R(k)=\frac{k}{2}\left[d(d+1)\ln 2+\frac{d(d+1)}{2}\ln\pi+d\right],$$

$$\Xi=\{\boldsymbol{\lambda}^*,\xi,\boldsymbol{\lambda},\{\boldsymbol{m}_i,\boldsymbol{m}_i^*,\boldsymbol{\Phi}_i^*\}_{i=1}^k,\beta,\gamma,\boldsymbol{\Phi}\},\quad \gamma_i^*=\gamma+\sum_{t=1}^{N}p(i|\boldsymbol{x}_t,\Xi,\boldsymbol{\lambda}). \tag{37}$$

Therein $\alpha_i^p(\xi+N,\boldsymbol{\lambda}^*)$ and $R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i)$ are the same as in Eq. (30). An algorithm is developed in Table 6 to maximize the lower bound of harmony measure $H^{\mathrm{DNW}}(\Xi,\Xi^*)$. Readers are referred to Appendix A for the detailed derivations about $H^{\mathrm{DNW}}(\Xi,\Xi^*)$ and this algorithm.

Again, we observe a similar format to $H(p||q,\Theta,\Xi)$ in Eq. (13) and $H^{\mathrm{Dir}(\boldsymbol{\alpha})}(\Theta,\Xi)$ in Eq. (30) but with three different points. First, $p(i|\boldsymbol{x}_t,\Theta,\boldsymbol{\lambda})$ in Eq. (30) is replaced by $p(i|\boldsymbol{x}_t,\Xi,\boldsymbol{\lambda})$ in Eq. (37) that has a longer tail and thus increases its attention on the overlapped regions. Second, $\ln[\alpha_i^p G(\boldsymbol{x}|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})]$ in Eq. (30) is replaced by $\ln[\alpha_i^p(\xi,\boldsymbol{\lambda})G(\boldsymbol{x}|\boldsymbol{m}_i^*,\boldsymbol{\Phi}_i^*/\gamma_i^*)]$ in Eq. (37) with the mean and covariance of each Gaussian component estimated by the regularized hyper-parameters. Third, the regularization term $R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i)$ in Eq. (30) is replaced by $R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i)-R(\Xi_i)+[R(\Phi,\beta,\gamma)-R(k)]/k$ that provides the regularizing role on all the parameters.

Also, we can further extend Eqs. (24) and (6) to consider the following $J_{\mathrm{MML}}(\Theta)$ with the DNW prior given in Eq. (9):

$$J_{\mathrm{MML}}^{\mathrm{DNW}}(\Theta,\Xi)=\ln q(\boldsymbol{X}_N|\Theta)+\ln\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi)$$
$$+\sum_{i=1}^{k}\ln G(\boldsymbol{\mu}_i|\boldsymbol{m}_i,\boldsymbol{T}_i^{-1}/\beta)$$
$$+\sum_{i=1}^{k}\ln\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi},\gamma)-\frac{1}{2}\ln|\boldsymbol{I}(\Theta)|. \tag{38}$$

Compared with $J_{\mathrm{MML}}^{\mathrm{Dir}(\boldsymbol{\alpha})}$ in Eq. (33), the third and fourth terms are added due to the Normal-Wishart priors on the Gaussian parameters, and last term comes from Eq. (6). The algorithm to maximize this $J_{\mathrm{MML}}^{\mathrm{DNW}}(\Theta,\Xi)$ is sketched in Appendix B, shortly denoted as MML-DNW. Actually, a special case of this MML-DNW algorithm returns to be the same as the MML-Jef algorithm given in Ref. [18].

Moreover, we can also extend $J_{\mathrm{VB}}^{\mathrm{Dir}(\boldsymbol{\alpha})}(\Theta,\Xi)$ in Eq. (32) into a general variational lower bound J_{VB} with the DNW prior by Eq. (9). The corresponding posterior is factorized as

$$p(\boldsymbol{Y},\Theta)=p(\boldsymbol{Y})p(\boldsymbol{\alpha})\prod_{i=1}^{k}p(\boldsymbol{\mu}_i,\boldsymbol{T}_i),$$
$$p(\boldsymbol{\mu}_i,\boldsymbol{T}_i)=G(\boldsymbol{\mu}_i|\boldsymbol{m}_i^*,\boldsymbol{T}_i^{-1}/\beta_i^*)\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi}_i^*,\gamma_i^*), \tag{39}$$

where $p(\boldsymbol{Y})$ and $p(\boldsymbol{\alpha})$ remain the same as in Eq. (31), while $p(\boldsymbol{\mu}_i,\boldsymbol{T}_i)$ is a Normal-Wishart distribution resulted from the conjugate property. It follows that $J_{\mathrm{VB}}^{\mathrm{DNW}(\boldsymbol{\alpha})}$ given by Eq. (32) is extended into

$$J_{\mathrm{VB}}^{\mathrm{DNW}}(\Xi^*,\Xi)=\int\prod_{i=1}^{k}p(\boldsymbol{\mu}_i,\boldsymbol{T}_i)\left\{J_{\mathrm{VB}}^{\mathrm{DNW}(\boldsymbol{\alpha})}\right.$$
$$\left.+\ln\frac{\prod_i[G(\boldsymbol{\mu}_i|\boldsymbol{m}_i,\boldsymbol{T}_i^{-1}/\beta)\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi},\gamma)]}{\prod_i p(\boldsymbol{\mu}_i,\boldsymbol{T}_i)}\right\}\mathrm{d}\Theta,$$
$$\Xi^*=\{\boldsymbol{\lambda}^*,\xi^*\}\cup\{\boldsymbol{m}_i^*,\beta_i^*,\boldsymbol{\Phi}_i^*,\gamma_i^*\}_{i=1}^k, \tag{40}$$

which is maximized by the VB learning algorithm introduced in Appendix C, shortly denoted as the algorithm VB-DNW.

5 Empirical analysis

5.1 Performances of BYY, VB, and MML: A quick look

Based on a synthetic dataset illustrated in Fig. 1(a) with five Gaussian components, we start from comparing the performances of automatic model selection by VB, MML, and BYY, as well as MAP, with no priors or priors $q(\boldsymbol{\alpha})$ merely on the mixing weights. Learning starts from random initializations on one GMM with 20 Gaussian components as illustrated in Fig. 1(b). Each algorithm is implemented for 500 independent trials. The results are shown in Figs. 2–4.

Table 6 BYY algorithm on GMM with a DNW prior (BYY-DNW)

Initialization: Randomly initialize the model with a large enough number k of components; set $\tau = 0$ and the harmony measure $J_{\mathrm{BYY}}(\tau) = -\infty$;

repeat

Randomly pick a sample $\boldsymbol{x}_t$ from the dataset $\boldsymbol{X}_N$;

Yang-step: Get $p(i|\boldsymbol{x}_t) = p(i|\boldsymbol{x}_t, \boldsymbol{\Xi}^{\mathrm{old}}, \boldsymbol{\lambda}^{\mathrm{old}})$ by Eq. (37), $\alpha_i = \alpha_i^p(\xi^{*\mathrm{old}}, \boldsymbol{\lambda}^{*\mathrm{old}})$ by Eq. (30),

$$w_{it} = \left[1 + \frac{\beta^{\mathrm{old}}}{\beta^{\mathrm{old}}+1}\boldsymbol{e}_{it}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{old}\,-1}\boldsymbol{e}_{it}\right]^{-1},\ \boldsymbol{e}_{it} = \boldsymbol{x}_t - \boldsymbol{m}_i^{\mathrm{old}},\ \boldsymbol{e}_{it}^* = \boldsymbol{x}_t - \boldsymbol{m}_i^{*\mathrm{old}},\ \boldsymbol{\varepsilon}_i = \boldsymbol{m}_i^{*\mathrm{old}} - \boldsymbol{m}_i^{\mathrm{old}}, \text{ and}$$

$$\delta_{it} = \delta_{it}^{(1)} + \delta_{it}^{(2)} + \delta_{it}^{(3)},\ \ \delta_{it}^{(u)} = \pi_{it}^{(u)} - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t)\pi_{jt}^{(u)},\ u = 1, 2, 3;$$

$$\pi_{jt}^{(1)} = \ln\left[\alpha_i G(\boldsymbol{x}_t|\boldsymbol{m}_i^{*\ \mathrm{old}}, \boldsymbol{\Phi}_i^{*\ \mathrm{old}}/(\gamma_i^{*\ \mathrm{old}} + 1 - p(i|\boldsymbol{x}_t)))\right],$$

$$\pi_{it}^{(2)} = -\frac{1}{2}\sum_{\tau\neq t} p(i|\boldsymbol{x}_\tau)\boldsymbol{e}_{i\tau}^{*\mathrm{T}}\boldsymbol{\Phi}_i^{*\mathrm{old}\,-1}\boldsymbol{e}_{i\tau}^*,\ \ \pi_{it}^{(3)} = -\frac{\beta^{\mathrm{old}}}{2}\boldsymbol{\varepsilon}_i^{\mathrm{T}}\boldsymbol{\Phi}_i^{*\mathrm{old}\,-1}\boldsymbol{\varepsilon}_i - \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Phi}^{\mathrm{old}}\boldsymbol{\Phi}_i^{*\mathrm{old}\,-1}\right);$$

Ying-step: Get $\xi^* = \xi^{\mathrm{old}} + N$, and update the hyper-parameters $\boldsymbol{\lambda}^*$:

$$\lambda_i^{*\mathrm{new}} = \frac{\lambda_i^{*\mathrm{old}} + \eta\Delta\lambda_i(\boldsymbol{\Xi}^{\mathrm{old}})}{\sum_{j=1}^{k}(\lambda_i^{*\mathrm{old}} + \Delta\lambda_j(\boldsymbol{\Xi}^{\mathrm{old}}))},\ \Delta\lambda_i(\boldsymbol{\Xi}) = \varsigma_{it}\xi^*\lambda_i^*[\Psi'(\xi^*\lambda_i^*) - \Psi'(\xi^*)],$$

where $\varsigma_{it} = p(i|\boldsymbol{x}_t) + (\xi^{\mathrm{old}}\lambda_i^{\mathrm{old}} - 1)/N$, and $\Psi'(x) = (\mathrm{d}/\mathrm{d}x)\Psi(x)$ is the trigamma function;

Then, get $\gamma_i^* = \gamma^{\mathrm{old}} + \sum_{t=1}^{N} p(i|\boldsymbol{x}_t)$ for each i, update $\{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^k$ by

$$\boldsymbol{m}_i^{*\mathrm{new}} = \boldsymbol{m}_i^{*\mathrm{old}} + \eta p(i|\boldsymbol{x}_t)[\gamma_i^{*\mathrm{old}} + 1 - p(i|\boldsymbol{x}_t)]\boldsymbol{e}_{it}^* - \beta^{\mathrm{old}}\gamma_i^{*\mathrm{old}}\boldsymbol{\varepsilon}_i/N,$$

$$\boldsymbol{\Phi}_i^{*\mathrm{new}} = \boldsymbol{S}_i^{\mathrm{new}}\boldsymbol{S}_i^{\mathrm{new}\mathrm{T}},\quad \text{with}\quad \boldsymbol{S}_i^{\mathrm{new}} = \boldsymbol{S}_i^{\mathrm{old}}\left\{\boldsymbol{I}_d + \eta\left[\frac{\gamma_i^{*\ \mathrm{old}}\boldsymbol{\Phi}_i^{*\mathrm{old}\ -1}}{\gamma_i^{*\ \mathrm{old}} - d}\boldsymbol{G}_i(\boldsymbol{\Xi}_i^{\mathrm{old}}) - \boldsymbol{I}_d\right]\right\},$$

$$\boldsymbol{G}_i(\boldsymbol{\Xi}_i) = \frac{\gamma_i^{*\mathrm{old}} + 1 - p(i|\boldsymbol{x}_t)}{\gamma_i^{*\mathrm{old}}}Np(i|\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{m}_i^*)(\boldsymbol{x}_t - \boldsymbol{m}_i^*)^{\mathrm{T}} + \beta\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^{\mathrm{T}} + \boldsymbol{\Phi};$$

H-step: Get $\nu_i = \lambda_i^{\mathrm{old}}[\Psi(\xi^{\mathrm{old}}) - \Psi(\xi^{\mathrm{old}}\lambda_i^{\mathrm{old}}) - \Psi(\xi^*) + \Psi(\xi^*\lambda_i^{*\ \mathrm{old}})]$, and update:

$$\lambda_i^{\mathrm{new}} = (\lambda_i^{\mathrm{old}} + \eta\delta\lambda_i)/\sum_{j=1}^{k}(\lambda_j^{\mathrm{old}} + \eta\delta\lambda_j),\quad \delta\lambda_i = p(i|\boldsymbol{x}_t)\delta_{it} + \frac{\xi^{\mathrm{old}}}{N}\nu_i,$$

$$\xi^{\mathrm{new}} = \xi^{\mathrm{old}} + \eta\delta\xi,\quad \delta\xi = \sum_{i=1}^{k}\{\nu_i + N\varsigma_{it}[\lambda_i^{*\mathrm{old}}\Psi'(\lambda_i^{*\mathrm{old}}\xi^{*\mathrm{old}}) - \Psi'(\xi^{*\mathrm{old}})]\},$$

$$\boldsymbol{m}_i^{\mathrm{new}} = \boldsymbol{m}_i^{\mathrm{old}} + \eta\delta\boldsymbol{m}_i(\boldsymbol{\Xi}_i^{\mathrm{old}}),\quad \delta\boldsymbol{m}_i(\boldsymbol{\Xi}_i) = \frac{1}{\beta+1}p(i|\boldsymbol{x}_t)\delta_{it}w_{it}\boldsymbol{e}_{it} + \frac{\gamma_i^*}{(\gamma+1)N}\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}\boldsymbol{\varepsilon}_i,$$

$$\beta^{\mathrm{new}} = \beta^{\mathrm{old}} + \eta\left[\frac{kd}{\beta^{\mathrm{old}}} - \delta\beta(\boldsymbol{\Xi}^{\mathrm{old}})\right],$$

$$\delta\beta(\boldsymbol{\Xi}) = \sum_{i=1}^{k}\gamma_i^*\boldsymbol{\varepsilon}_i^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}\boldsymbol{\varepsilon}_i + \frac{N(\gamma+1)}{\beta+1}\sum_{i=1}^{k}p(i|\boldsymbol{x}_t)\delta_{it}w_{it}(1 + \boldsymbol{e}_{it}^{\mathrm{T}}\boldsymbol{\Phi}^{-1}\boldsymbol{e}_{it}),$$

$$\boldsymbol{\Phi}^{\mathrm{new}} = \boldsymbol{S}^{\mathrm{new}}\boldsymbol{S}^{\mathrm{new}\mathrm{T}},\quad \text{with}\quad \boldsymbol{S}^{\mathrm{new}} = \boldsymbol{S}^{\mathrm{old}}\left\{\boldsymbol{I}_d + \eta\left[\frac{1}{k\gamma}\boldsymbol{G}(\boldsymbol{\Xi}^{\mathrm{old}}) - \boldsymbol{I}_d\right]\right\},$$

$$\boldsymbol{G}(\boldsymbol{\Xi}) = \sum_{i=1}^{k}\left[\gamma_i^*\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1} - \frac{\beta N(\gamma+1)}{\beta+1}p(i|\boldsymbol{x}_t)\delta_{it}w_{it}\boldsymbol{e}_{it}\boldsymbol{e}_{it}^{\mathrm{T}}\boldsymbol{\Phi}^{-1}\right],$$

$$\gamma^{\mathrm{new}} = \gamma^{\mathrm{old}} + \eta\left[N\sum_{i=1}^{k}p(i|\boldsymbol{x}_t)\delta_{it}\ln w_{it} + \frac{N + k(\gamma^{\mathrm{old}} - d)}{2}\sum_{j=1}^{d}\Psi'\left(\frac{\gamma^{\mathrm{old}} + 1 - j}{2}\right) + \delta\gamma(\boldsymbol{\Xi}^{\mathrm{old}})\right],$$

$$\delta\gamma(\boldsymbol{\Xi}) = k\ln|\boldsymbol{\Phi}| - \sum_{i=1}^{k}\left[\ln|\boldsymbol{\Phi}_i^*| + \mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}) + \beta\boldsymbol{\varepsilon}_i^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}\boldsymbol{\varepsilon}_i\right];$$

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\lambda_i \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another* $5N$ *runs have passed* **then** let $\tau = \tau + 1$; calculate $J_{\mathrm{BYY}}(\tau) = H^{\mathrm{DNW}}(\boldsymbol{\Xi}, \boldsymbol{\Xi}^*)$ by Eq. (37);

until $J_{\mathrm{BYY}}(\tau) - J_{\mathrm{BYY}}(\tau-1) < \epsilon J_{\mathrm{BYY}}(\tau-1)$, *with* $\epsilon = 10^{-5}$;

The first column of Fig. 2 shows the results of MAP, VB, MML, and BYY, without considering any priors on the parameters. MAP, VB, and MML degenerate to maximum likelihood (ML) learning and thus they all largely bias to oversized models, while BYY is still capable of model selection with a bias towards undersized models.

The second column of Fig. 2 shows the results by MAP-Jef($\boldsymbol{\alpha}$), VB-Jef($\boldsymbol{\alpha}$), MML-Jef($\boldsymbol{\alpha}$), and BYY-Jef($\boldsymbol{\alpha}$). It can be observed that VB and BYY bias to undersized models while MAP and MML bias to oversized models. To be specific, MML-Jef($\boldsymbol{\alpha}$) degenerated to ML according to Eq. (26), VB-Jef($\boldsymbol{\alpha}$) performs better than MAP-Jef($\boldsymbol{\alpha}$), and BYY-Jef($\boldsymbol{\alpha}$) further outperforms VB-Jef($\boldsymbol{\alpha}$).

The third column of Fig. 2 compares MAP-Dir($\boldsymbol{\alpha}$), VB-Dir($\boldsymbol{\alpha}$), MML-Dir($\boldsymbol{\alpha}$) and BYY-Dir($\boldsymbol{\alpha}$) based on Dirichlet prior with the prior hyper-parameters pre-fixed at $\lambda_i = 1/k$ ($\forall i$) and $\xi = k/4$. As the Jeffreys is replaced by a Dirichlet, all algorithms improve while VB deteriorates. Also, all approaches incline to oversized models.

It is natural to wonder whether the values of the prior

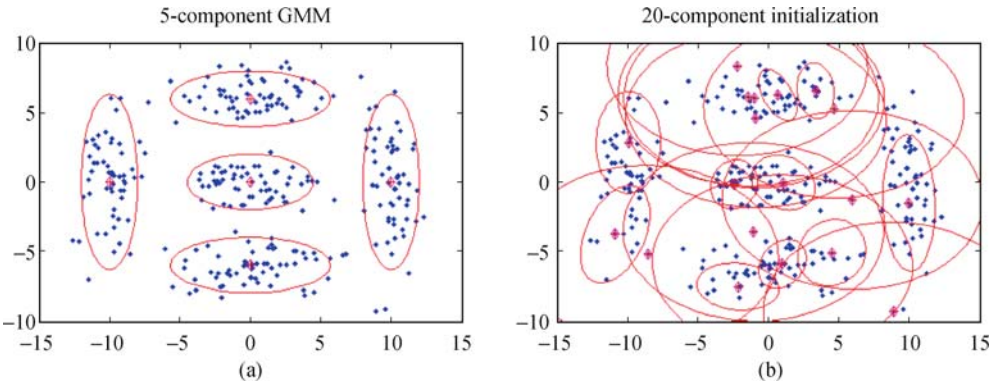


Fig. 1 Synthetic data and initialization. (a) A dataset of 300 samples is generated from a 2-dimensional 5-component GMM, with equal mixing weights 1/5; (b) all the three algorithms start from randomly initialized 20-component GMMs and shown here is an example (Each red curve indicates to a contour of equal probability density per component, and the red diamonds indicate the Gaussian means.)

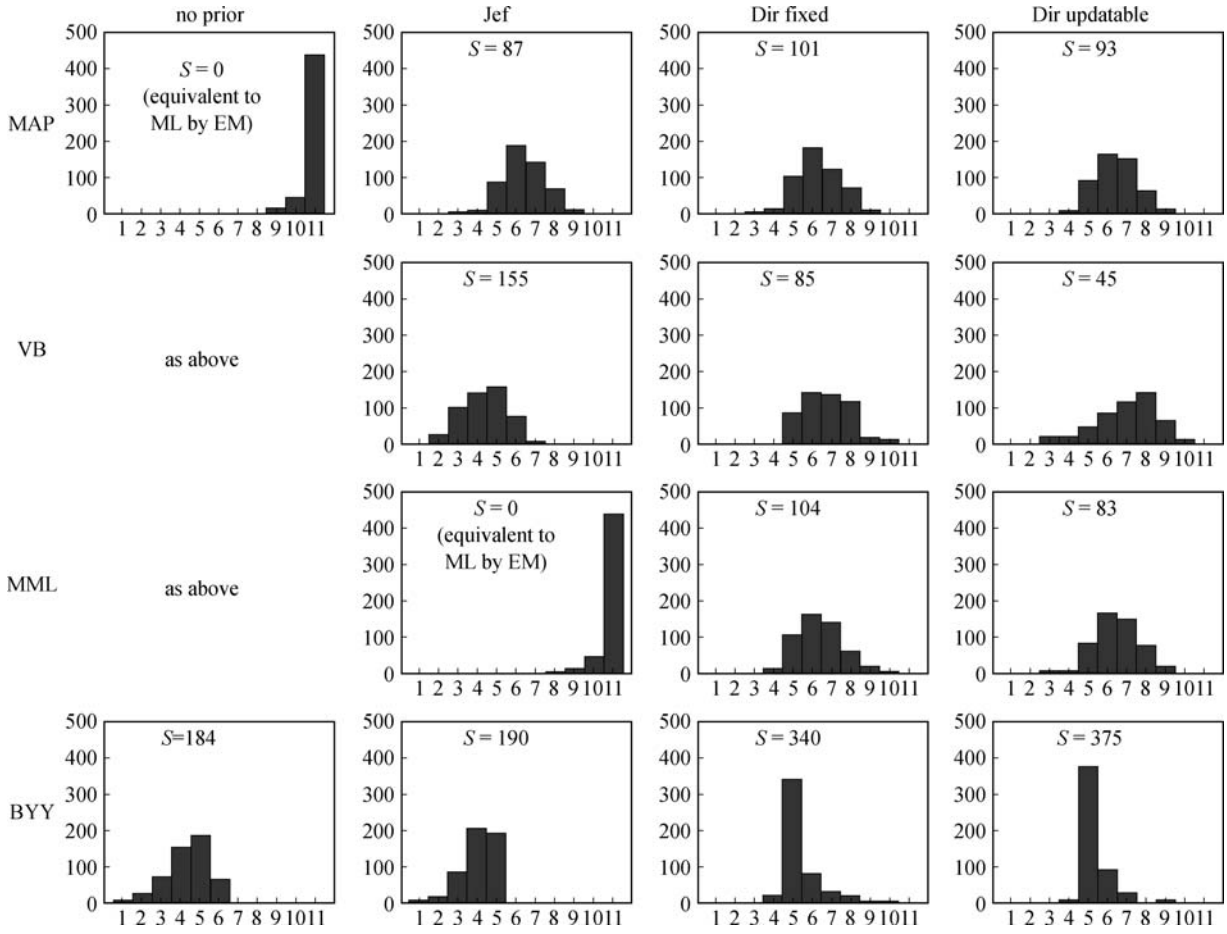


Fig. 2 Model selection results on a synthetic dataset illustrated in Fig. 1(a) by algorithms with priors $q(\boldsymbol{\alpha})$ (Each algorithm is implemented for 500 independent trials. The histograms show the frequencies f_i of the estimated component number being i, and the values of S that stands for the times of successfully selecting the true component number $k^* = 5$. When there is no prior, MAP, VB, and MML all degenerate to the maximum likelihood learning, with its resulted histogram actually stretching beyond the rightmost boundary (the rightmost bar being the sum of all the bars beyond the boundary for a rough visualization).)

hyper-parameters λ_i and ξ affect the performances of model selection. For this purpose, the fourth column of Fig. 2 compares MAP-Dir($\boldsymbol{\alpha}$), VB-Dir($\boldsymbol{\alpha}$), MML-Dir($\boldsymbol{\alpha}$) and BYY-Dir($\boldsymbol{\alpha}$) with the prior hyper-parameters λ_i and ξ optimized under each of its own learning principle. The results show that VB deteriorates considerably, and MAP and MML decline slightly, while BYY obtains a further improvement.

Next, we move to examine the performances of three algorithms on the same synthetic dataset illustrated in Fig. 1(a) but with priors on all parameters. As shown in Fig. 4, considering a full prior improves all the

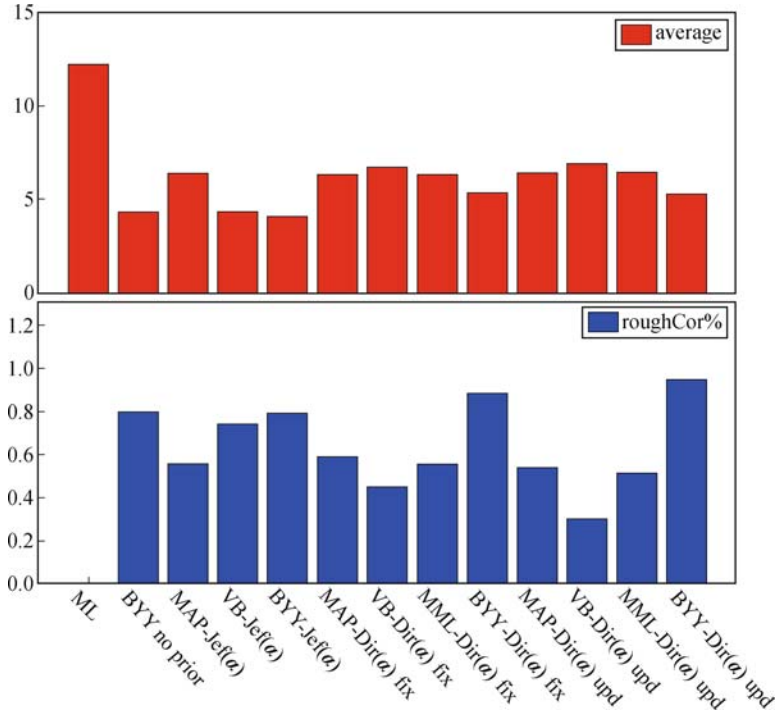


Fig. 3 Statistics of model selection results in Fig. 2 (The "average"(red) bar indicates $\bar{k} = \frac{1}{500\times 5}\sum_i (f_i \cdot i)$, and the "roughCor" (blue) bar indicates a rough correct selection rate by $\bar{f} = (f_4 + f_5 + f_6)/500$. For both types of bars, the better a method is, the more closer its "average" is to 5 and its "roughCor" is to 1.)

approaches except MAP with merely priors on the mixing weights.

As shown by the first column of Fig. 4, considering the Jeffreys prior on all the parameters of GMM makes each approach reduce its bias shown by its counterpart of Fig. 2, except that a full prior increases the bias of MAP to oversized models. MAP-Jef is the worst, MML-Jef gets $S = 146$ that is bigger than $S = 136$ by VB-Jef, while VB-Jef performs slightly better than MML-Jef in an average sense. Moreover, BYY-Jef outperforms both VB and MML much significantly, with its errors mainly skewed to oversized models.

The second column compares MAP-DNW, VB-DNW, MML-DNW and BYY-DNW, with the hyper-parameters of the Dirichlet prior still pre-fixed at $\lambda_i = 1/k$ $(\forall i)$ and $\xi = k/4$ and with the hyper-parameters of the Normal-Wishart (NW) priors fixed at $\beta = 1$, $\boldsymbol{\Phi} = \boldsymbol{I}_d$, and $\gamma = d + 1$, as well as $\boldsymbol{m}_i = \boldsymbol{m}$ and $\boldsymbol{m} = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{x}_t$. In implementation, the H-step in each algorithm is skipped. From Jeffreys to DNW, one observes that, in contrast to MAP and VB, both MML and BYY improve their performances, and the DNW prior makes VB incline obviously to oversized models. Still, BYY outperforms both MML and VB significantly.

Next, we consider to optimize the hyper-parameters under each of its own learning principle. Shown in the third column are the results by considering the DNW prior in Eq. (9), featured with each $\boldsymbol{m}_i$ to be freely adapted, as introduced in Sect. 2.3. It can be observed that BYY further improves its performance considerably and outperforms the others significantly, which concurs with the nature that BYY best harmony provides a guideline to optimize the hyper-parameters [17,26] and learning hyper-parameters is a part of the entire learning implementation (e.g., see the learning procedure shown in Fig. 6(a) in Ref. [17] or in Fig. 5(a) in Ref. [26]). Also, it is observed that MAP improves its counterpart with the hyper-parameters pre-fixed too, which could be understood from the relation that MAP can be regarded as a degenerated case of the BYY harmony learning (see Eq. (A.4) in Ref. [17] or Eq. (39) in Ref. [26]).

However, it also follows from the third column that both VB and MML deteriorate, especially the performance of VB drops drastically, becoming even inferior to MAP. The reason is that VB and MML maximize the marginal likelihood $q(\boldsymbol{X}_N|\boldsymbol{\Xi}) = \int q(\boldsymbol{X}_N|\boldsymbol{\Theta})q(\boldsymbol{\Theta}|\boldsymbol{\Xi})\mathrm{d}\boldsymbol{\Theta}$, via variational approximation and Laplace approximation respectively. Maximizing $q(\boldsymbol{X}_N|\boldsymbol{\Xi})$ with respect to a free $q(\boldsymbol{\Theta}|\boldsymbol{\Xi})$ tends to becoming $\max_{\boldsymbol{\Theta}} q(\boldsymbol{X}_N|\boldsymbol{\Theta})$ and $q(\boldsymbol{\Theta}|\boldsymbol{\Xi}) = \delta(\boldsymbol{\Theta} - \boldsymbol{\Theta}^*)$ with $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} q(\boldsymbol{X}_N|\boldsymbol{\Theta})$, i.e., towards maximizing the likelihood function. In other words, the learning principle of VB and MML, i.e., maximizing the marginal likelihood, may not be good for optimizing the hyper-parameters. To get a further insight on this observation, we consider the DNW prior in Eq. (9) with every $\boldsymbol{m}_i = \boldsymbol{m}$ constrained to be the same. This constraint reduces the number of free parameters and prevents a deep optimization with respect to the hyper-parameters. Observed from the fourth column of Fig. 4, the performances of MAP, VB and MML are actually

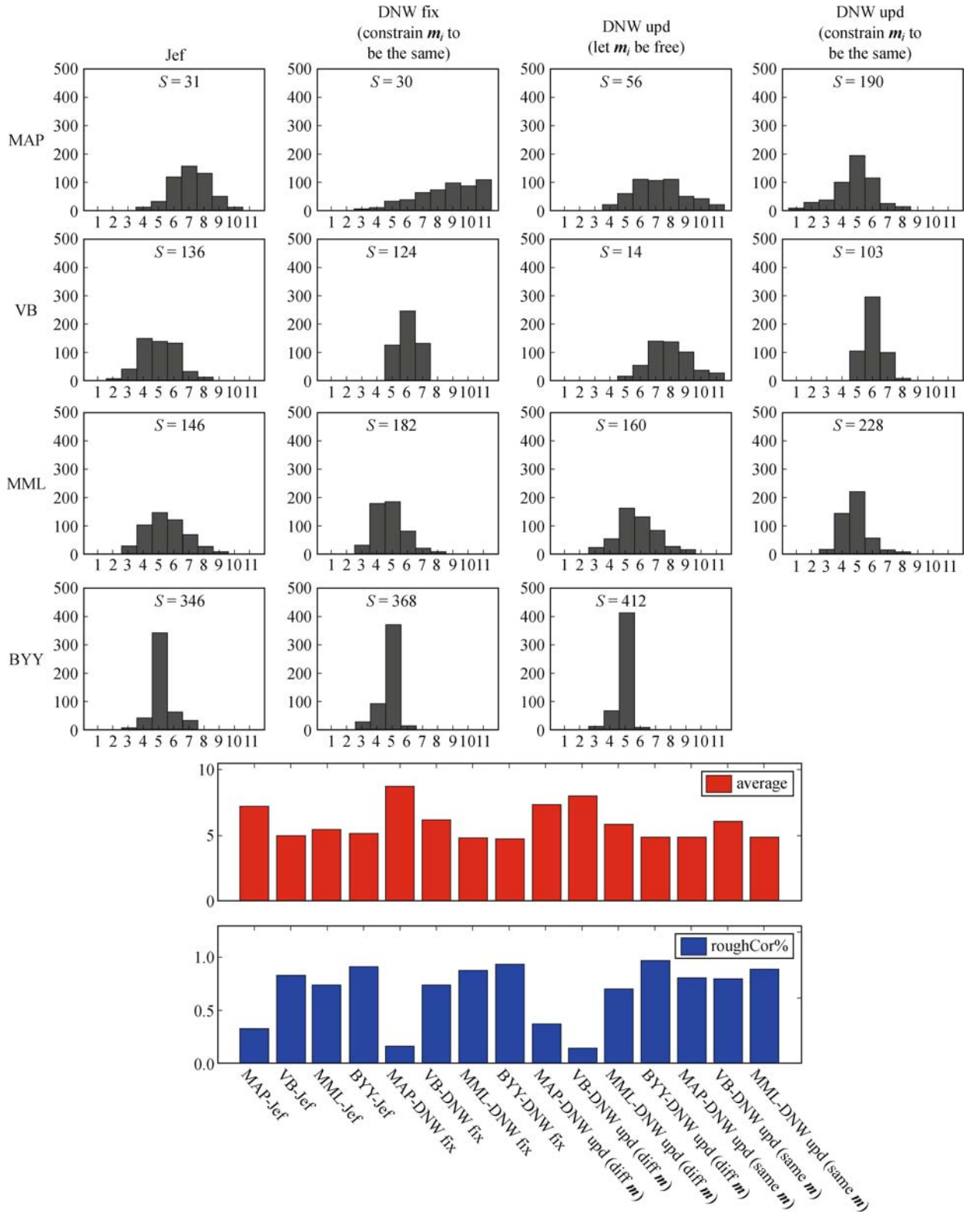


Fig. 4 Model selection results on a synthetic dataset illustrated in Fig. 1(a) by three algorithms with priors on all parameters

improved considerably. In other words, the learning principle of VB and MML with certain constraints also provide some guide for optimizing the hyper-parameters. But it is not comparable to BYY that still outperforms MAP, VB and MML considerably.

Summarizing the above observations, we iterate the following key points:

1) Considering a full prior improves all the approaches with merely priors on the mixing weights.

2) With a full prior, BYY consistently improves from the first column to the last column, and significantly outperforms all the other approaches.

3) As the hyper-parameters of DNW prior are optimized by each of its own learning principle, BYY and MAP improve their performances while VB and MML deteriorate their performances when there are too many free hyper-parameters.

4) Lacking a good guide for optimizing the hyper-parameters, VB performs no obvious improvements as Jeffreys prior is replaced by DNW prior. Also, for DNW prior with the hyper-parameters optimized, MAP outperforms VB considerably.

5) Better than nothing, the learning principle of VB and MML with certain constraints also provide some guide for optimizing the hyper-parameters.

5.2 VB, MML, and BYY with full priors: Extensive comparisons

Based on the major observations at the end of the previous subsection, we make further investigations on a wide range of synthetic datasets. Though the performances of VB are inferior to MAP with the hyper-parameters of DNW prior optimized, MAP is not further considered since it can be regarded as a degenerated case of BYY. We compare the performances of VB, MML, and BYY, with priors fully on all parameters.

The datasets are repeatedly generated from GMM with $\{\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^{k^*}$ by Eq. (1) or Eq. (2), where k^* denotes the true component number, $\{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^{k^*}$ is randomly generated according to the joint Normal-Wishart distribution $G(\boldsymbol{\mu}_i|\boldsymbol{m}_i, \boldsymbol{T}_i^{-1}/\beta)\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi}, \gamma)$ given in Eq. (9), with the hyper-parameters set at $\forall\alpha_i = 1/k^*$, $\gamma = 50$ and $\boldsymbol{\Phi} = \boldsymbol{I}_d$, as well as $\boldsymbol{m}_i = \boldsymbol{m}$ and $\boldsymbol{m} = \boldsymbol{0}$.

To cover a wide range of experimental conditions, we vary the values of the sample size N, the data dimensionality d, the true component number k^*, and the overlap degree β of Gaussian components, where the hyper-parameter β increases from small to large indicates that the degree of separation of the Gaussian components changes from large to small. Generally speaking, the model selection task becomes more and more difficult as N decreases, or d increases, or k^* increases, or β increases.

We consider four series of experiments specified in Table 7. Starting from a same point in the 4-dimensional factor space, each series varies one factor of (N, d, k^*, β) while the remaining three are fixed. For each specific setting of (N, d, k^*, β), 500 datasets are generated independently, and all the algorithms are implemented with a same initial component number 25.

Table 7 Four series of experiments

starting case	$(N, d, k^*, \beta) = (300, 5, 3, 0.02)$
series 1	vary $N \in \{300, 290, 280, \ldots, 30\}$ and fix d, k^*, β
series 2	vary $d \in \{5, 6, 7, \ldots, 30\}$ and fix N, k^*, β
series 3	vary $k^* \in \{3, 4, 5, \ldots, 20\}$ and fix N, d, β
series 4	vary $\beta \in \{0.1, 0.2, 0.3, \ldots, 2.0\}$ and fix N, d, k^*

The performances of model selection by VB, MML, and BYY on the four series are shown in Fig. 5 with the Jeffreys prior on all parameters. Jeffreys prior makes MML better than VB, and further makes BYY outperform MML except for a large sample size N in series 1, where MML is the best. Comparing with Fig. 2 that considers Jeffreys priors merely on the mixing weights, adding Jeffreys priors on $\{\boldsymbol{\mu}_j, \boldsymbol{T}_j\}$ makes MML improved more than VB does.

We further compare VB, MML, and BYY with DNW priors on all parameters. As previously discussed on the results shown in the 3rd column of Fig. 4, the learning principle of VB and MML is not good for guiding the searching of the hyper-parameters. Also, it is difficult to search hyper-parameters in a try-and-test manner for a series of experimental settings. Being different from Fig. 4 where fixing $\boldsymbol{\Xi}$ results in better performances than updating $\boldsymbol{\Xi}$ under VB and MML, heuristically fixing $\boldsymbol{\Xi}$ usually make the performances of VB and MML not good enough. Taking the datasets of series 1 as an example, shown in Fig. 6 are the performances of VB, MML and BYY with the hyper-parameters $\boldsymbol{\Xi}$ fixed at the same values as those in Fig. 4, which are actually worse than their counterparts that optimize the hyper-parameters under each of its own learning principle. Better than nothing, as previously discussed on the results shown in the 4th column of Fig. 4, the learning principle of VB and MML with every $\boldsymbol{m}_i = \boldsymbol{m}$ constrained to be same still provide some guide for optimizing the hyper-parameters. With this consideration, we compare the performances of model selection by VB, MML, and BYY with the DNW prior, via optimizing the hyper-parameters under each of its own learning principle and also under the constraints $\boldsymbol{m}_i = \boldsymbol{m}$ for VB and MML. Shown in Fig. 7 are results on the four series. Also, the performances of those with the Jeffreys prior versus the DNW prior are re-plotted together in Fig. 8 for an easy comparison. We obtain further observations as follows:

1) BYY considerably outperforms both VB and MML for both types of priories, except the large values of N in series 1 where MML is the best.

2) Jeffreys prior makes MML slightly better than VB, while the DNW prior makes VB better than MML.

3) All the three approaches improve their performances as Jeffreys is replaced by DNW, as shown in Fig. 8. The Wishart prior on the precision matrix with appropriate hyper-parameters can effectively avoid the singularity of the covariance matrix caused by a very small sample size, and thus improve the model selection performance.

4) VB is more sensitive than MML to which type of prior is used, as Jeffreys prior makes VB inferior to MML while DNW prior makes VB better than MML. BYY is much robust than VB and MML for either Jeffreys or

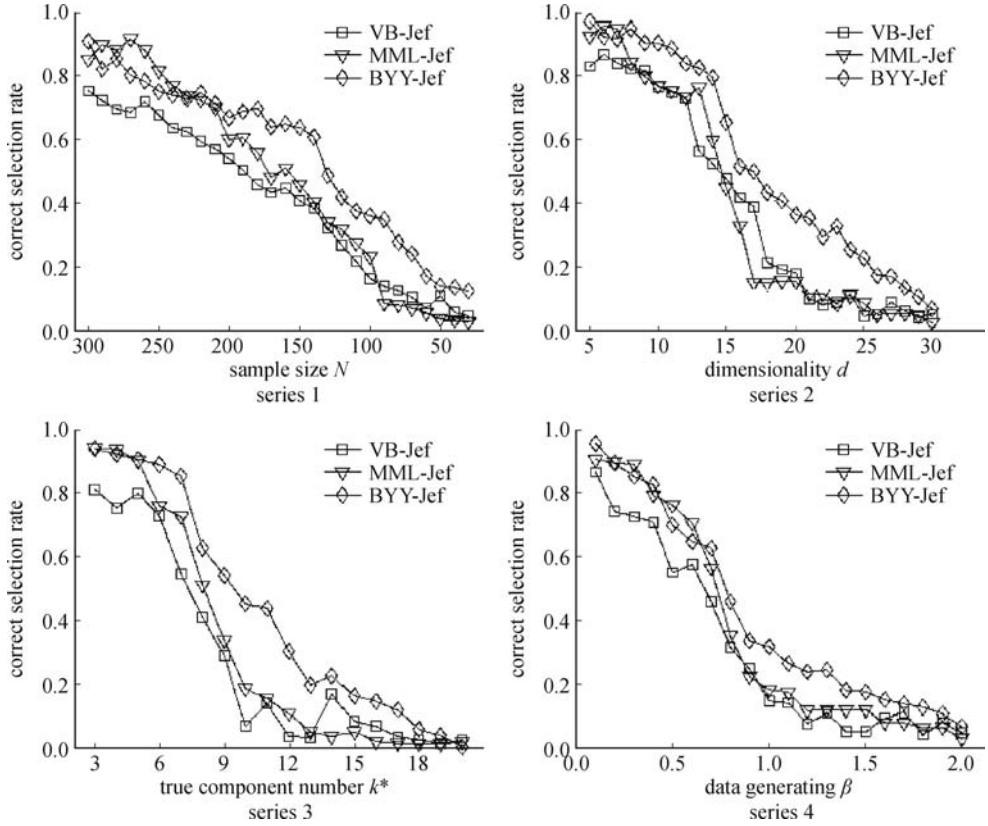


Fig. 5 Performances by Algorithms VB-Jef, MML-Jef and BYY-Jef in the four series of experiments

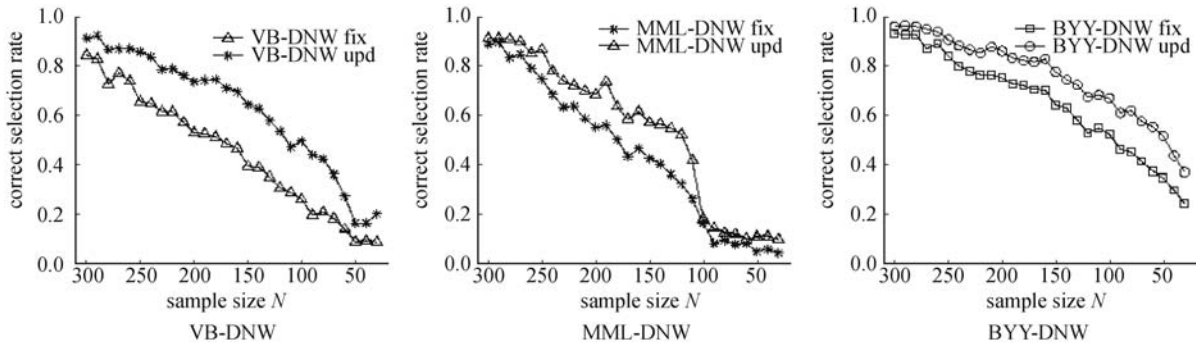


Fig. 6 Performances of VB-DNW, MML-DNW and BYY-DNW on the dataset of series 1: hyper-parameters fixed (at $\lambda_i = 1/k$ $(\forall i)$, $\xi = k/4$, $\beta = 1$, $\boldsymbol{\Phi} = \boldsymbol{I}_d$, $\gamma = d + 1$, $\boldsymbol{m}_i = \boldsymbol{m}$ $(\forall i)$ and $\boldsymbol{m} = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{x}_t$) versus hyper-parameters optimized

DNW prior.

5.3 Application on image segmentation

We further apply the algorithms to unsupervised image segmentation on the Berkeley segmentation database of real world images (http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/). We choose the features proposed by Varma and Zisserman [40], shortly denoted as VZ features, which has been used in image segmentation with promising results [41]. Specifically, a VZ feature is constructed for each pixel by picking a $w \times w$-sized window centered around the pixel, vectorizing the color information of all pixels in this window, and then performing dimensionality reduction to v dimensions by principal component analysis (PCA). The VZ features have considered the neighborhood information and are insensitive to noise because of PCA. Following Refs. [40,41], we start from the LAB color space, and set $w = 7$ and $v = 8$ to construct the VZ features.

For every image, each of the learning algorithms outputs a GMM model that assigns each image pixel to the cluster (represented by a Gaussian component) by the maximum posterior probability. Since the true component number is unknown, we evaluate the resulted segmentations from the GMM clusters by PR index [27], which takes values between 0 and 1. A higher PR score indicates a better segmentation with a higher percentage of pixel pairs in the segmentation having the same labels

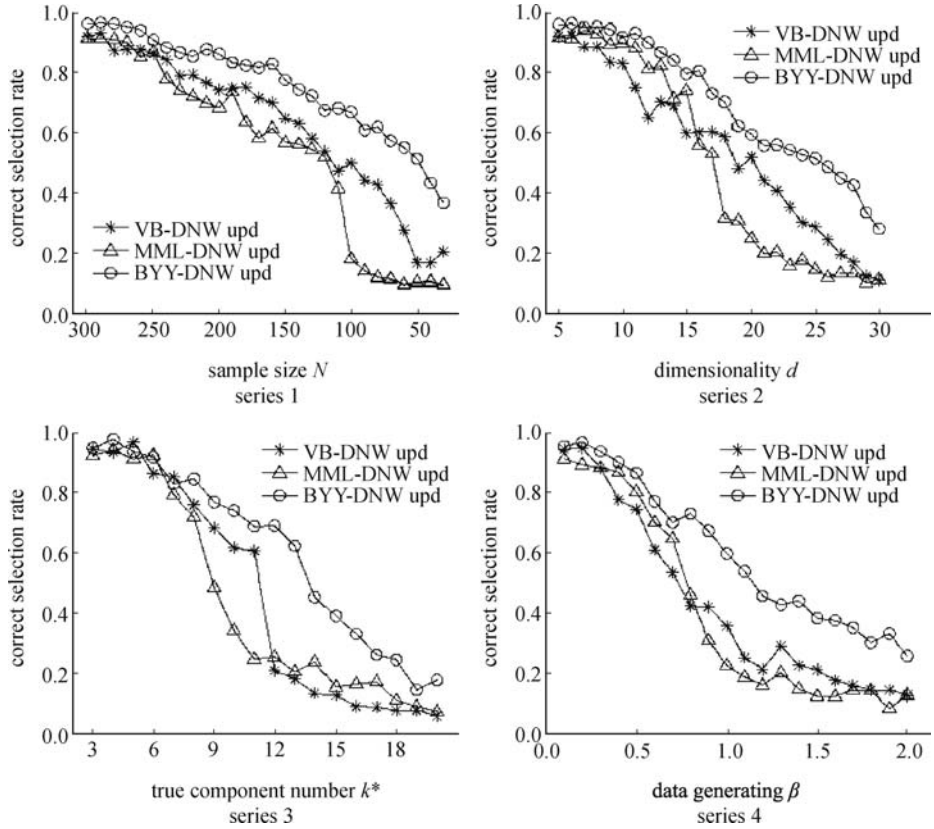


Fig. 7 Performances by Algorithms VB-DNW upd, MML-DNW upd and BYY-DNW upd in the four series of experiments

as in the ground truth segmentation provided by the database. A good model selection performance is closely related to, but not necessarily implies, a high PR score. To investigate the performance of each algorithm, we do not conduct further post-processing operations such as region merging and graph cut [42,43], although they would further improve the segmentation results.

The component number is initialized as 10 for each algorithm in experiments. Table 8 gives the average PR scores of five runs on all of the 100 testing images of the Berkeley image segmentation database. The segmentation performances are consistent with their model selection performances. Again, BYY-DNW obtains the highest average PR scores.

Shown in Fig. 9 are two examples chosen from the database. It can be observed that BYY-DNW is able to detect the objects of interest from a confusing background. Interestingly, the average PR score of VB-DNW on all 100 images is even lower than that of BYY-Jef, as given in Table 8, though the PR scores of VB-DNW on the above two images are higher than those of BYY-Jef.

6 Concluding remarks

This paper has presented a comparative investigation on the relative strengths and weaknesses of MML, VB and BYY in automatic model selection for determining the component number of a GMM with either a Jeffreys or a Dirichlet-Normal-Wishart (DNW) prior. The algorithm for MML is adopted from Ref. [18], either used directly for Jeffreys prior or with some modification for DNW prior. The algorithm for VB with DNW prior comes from modifying the one in Ref. [20] via the NW prior suggested in Ref. [25]. Moreover, the BYY algorithm for Jeffreys prior is a special case of the unified Ying-Yang learning procedure introduced in Ref. [12]. Furthermore, the algorithm for VB with Jeffreys prior and the algorithm for BYY with DNW priors have been developed in this paper. Through synthetic experiments, we have the following empirical findings.

Considering priors only on the mixing weights, VB is better than MML and MAP for the Jeffreys prior, but

Table 8 Average PR scores of five runs on the 100 testing images of the Berkeley image segmentation database (without post-processing operations)

VB-Jef	MML-Jef	BYY-Jef	VB-DNW upd	MML-DNW upd	BYY-DNW upd
0.772	0.752	0.816	0.803	0.788	0.851

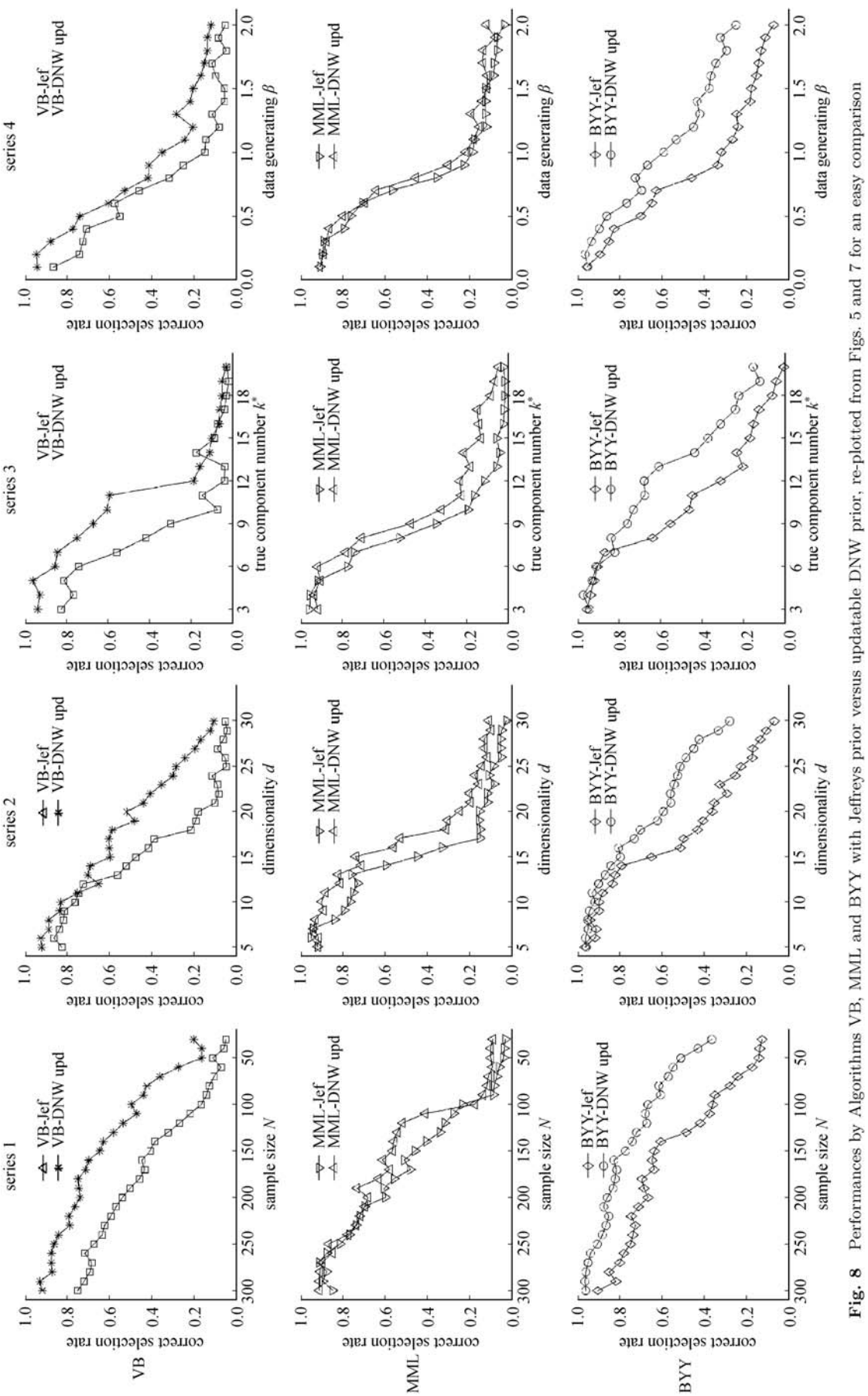


Fig. 8 Performances by Algorithms VB, MML and BYY with Jeffreys prior versus updatable DNW prior, re-plotted from Figs. 5 and 7 for an easy comparison

VB deteriorates to be inferior to both MML and MAP with the Jeffreys replaced by the Dirichlet prior. For either Jeffreys or Dirichlet prior, BYY considerably outperforms MAP, VB and MML. For the Jeffreys, MAP and MML bias to oversized models, while VB and BYY bias to undersized models. For the Dirichlet, all the approaches incline to oversized models. Moreover, optimizing the hyper-parameters of the Dirichlet prior further improves the performances of BYY, but brings down the other approaches.

Considering a full prior on all parameters of GMM makes each approach reduce its bias and also improve its performance. Jeffreys prior makes MML slightly better than VB, while the DNW prior makes VB better than MML. BYY considerably outperforms VB, MAP and MML for any type of priors and whether or not hyper-parameters are optimized. Being different from VB and MML that rely on appropriate priors to perform model selection, BYY does not highly depend on type of priors. It has model selection ability even without priors and performs already very good with Jeffreys prior, and incrementally improves as Jeffreys prior is replaced by the DNW prior.

As Jeffreys prior is replaced by DNW prior with appropriate hyper-parameters, all the approaches improve their performances. As the hyper-parameters of DNW prior are optimized by each of its own learning principle, BYY and MAP improve their performances while VB and MML deteriorate their performances when there are too many free hyper-parameters. Via DNW prior with

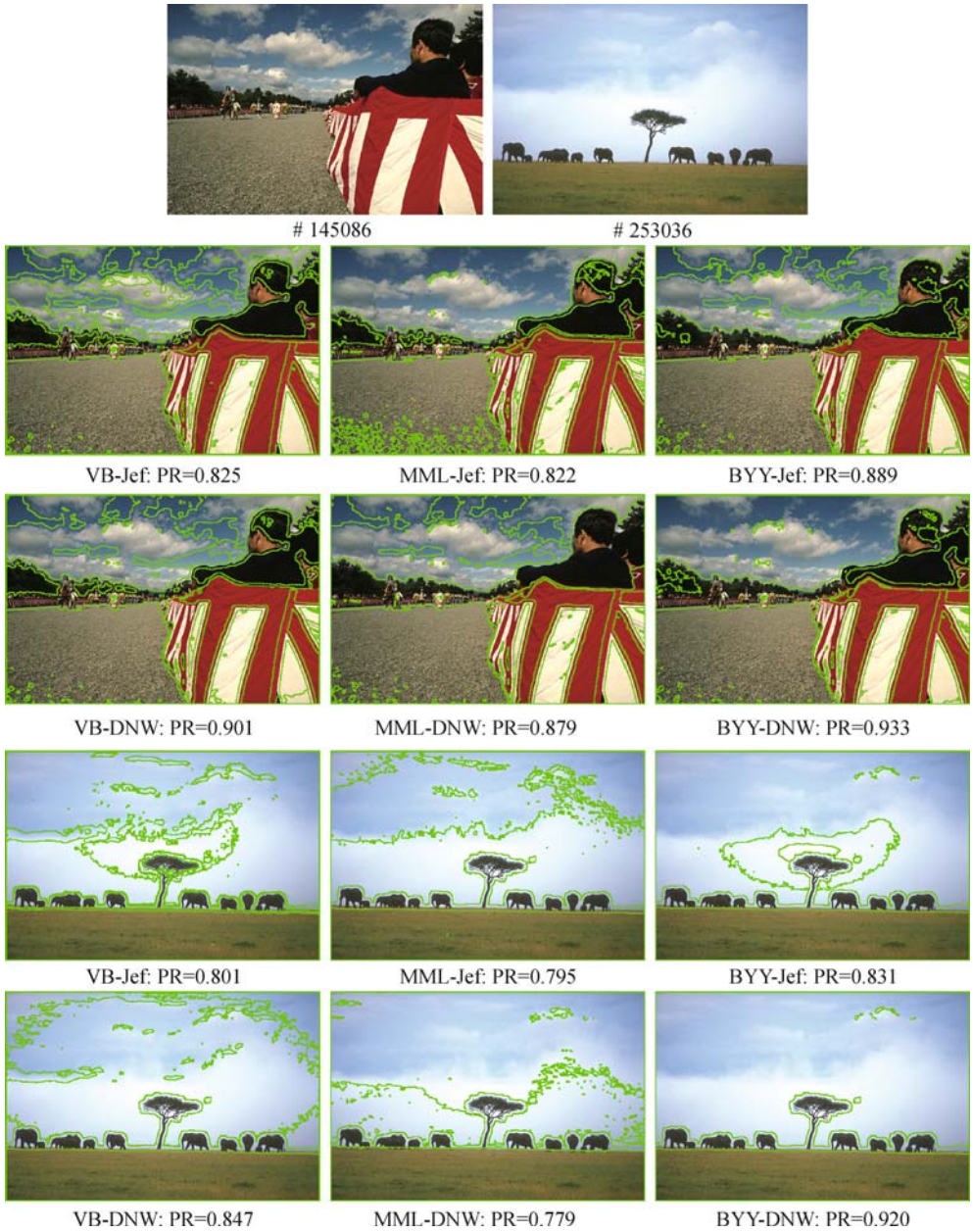


Fig. 9 Two image segmentation examples from Berkeley segmentation database (Segmentation results by each algorithm are illustrated by the highlighted green curves.)

the hyper-parameters optimized, MAP may also outperform VB considerably, while VB and MML lack a good guide for optimizing the hyper-parameters of DNW prior. Better than nothing, the learning principle of VB and MML with certain constraints also provide some guide for optimizing the hyper-parameters.

Furthermore, all algorithms are applied to a real world database for image segmentation. BYY is again confirmed to considerably outperform VB and MML, especially BYY-DNW is superior for its robustness in detecting the objects of interest from a confusing background.

Appendix A Derivations about BYY-DNW

Here we provide further details on the derivations of $H^{\mathrm{DNW}}(\Xi)$ and the BYY-DNW algorithm. Its special case $H^{\mathrm{Dir}(\boldsymbol{\alpha})}(\Theta,\Xi)$ and the algorithm BYY-Dir($\boldsymbol{\alpha}$) can be obtained by similar derivations via simply considering $\Xi=\{\boldsymbol{\lambda},\xi\}$ and $\Xi^*=\{\boldsymbol{\lambda}^*\}$.

We rewrite the harmony measure in Eq. (37) as follows:

$$\begin{aligned} H^{\mathrm{DNW}}(\Xi) &= \sum_{\boldsymbol{Y}} p(\boldsymbol{Y}|\mathbf{X}_N)\left\{\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\boldsymbol{X}_N,\boldsymbol{Y}|\Theta)]+\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\Theta)]\right\} \\ &= \mathrm{E}_{\boldsymbol{Y}|\mathbf{X}_N}\left\{\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\boldsymbol{X}_N,\boldsymbol{Y}|\Theta)]+\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\Theta)]\right\}, \end{aligned} \tag{A.1}$$

where $\mathrm{E}_{\Theta|\boldsymbol{Y}}[\cdot]$ denotes expectation with respect to $p(\Theta|\boldsymbol{Y})$ given by Eq. (34), and similarly we use $\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[\cdot]$ to denote expectation with respect to $p(\boldsymbol{Y}|\boldsymbol{X})$ given by Eq. (28) and $p(i|\boldsymbol{x}_t,\Xi,\boldsymbol{\lambda})$ by Eq. (37). Specifically, we get

$$\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\Theta)] = \mathrm{E}_{\Theta|\boldsymbol{Y}}\ln\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi)+\sum_{i=1}^{k}\mathrm{E}_{\Theta|\boldsymbol{Y}}\ln\left[G(\boldsymbol{\mu}_i|\boldsymbol{m}_i,\boldsymbol{T}_i^{-1}/\beta)\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi},\gamma)\right], \tag{A.2}$$

$$\begin{aligned} &\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda},\xi)] = \sum_{i=1}^{k}\{(\xi\lambda_i-1)\,\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln\alpha_i]\}+\ln\Gamma(\xi)-\sum_{i=1}^{k}\ln\Gamma(\xi\lambda_i), \\ &\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln G(\boldsymbol{\mu}_i|\boldsymbol{m}_i,\boldsymbol{T}_i^{-1}/\beta)] = -\frac{d}{2}\ln(2\pi)+\frac{d}{2}\ln\beta+\frac{1}{2}\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln|\boldsymbol{T}_i|]-\frac{\beta}{2}\mathrm{Tr}\{\mathrm{E}_{\Theta|\boldsymbol{Y}}\left[\boldsymbol{T}_i(\boldsymbol{\mu}_i-\boldsymbol{m}_i)(\boldsymbol{\mu}_i-\boldsymbol{m}_i)^{\mathrm{T}}\right]\}, \\ &\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi},\gamma)] = \frac{\gamma}{2}\ln|\boldsymbol{\Phi}|-\frac{d\gamma}{2}\ln 2+\frac{\gamma-d-1}{2}\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln|\boldsymbol{T}_i|]-\frac{1}{2}\mathrm{Tr}\{\mathrm{E}_{\Theta|\boldsymbol{Y}}[\boldsymbol{T}_i]\boldsymbol{\Phi}\}-\ln\Gamma_d\left(\frac{\gamma}{2}\right), \end{aligned}$$

where if we denote $\xi^*=\xi+N$, then

$$\begin{aligned} &\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln\alpha_i] = \Psi(\xi^*\lambda_i^*)-\Psi(\xi^*), \quad \mathrm{E}_{\Theta|\boldsymbol{Y}}[\boldsymbol{\mu}_i]=\boldsymbol{m}_i^*, \quad \mathrm{E}_{\Theta|\boldsymbol{Y}}[\boldsymbol{T}_i]=\left(\gamma+\sum_{t=1}^{N}y_{it}\right)\boldsymbol{\Phi}_i^{*-1}, \\ &\mathrm{E}_{\Theta|\boldsymbol{Y}}[\boldsymbol{T}_i(\boldsymbol{\mu}_i-\boldsymbol{m}_i)(\boldsymbol{\mu}_i-\boldsymbol{m}_i)^{\mathrm{T}}] = \frac{1}{\beta+\sum_t y_{it}}\boldsymbol{I}_d+\left(\gamma+\sum_t y_{it}\right)\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i^*-\boldsymbol{m}_i)(\boldsymbol{m}_i^*-\boldsymbol{m}_i)^{\mathrm{T}}, \\ &\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln|\boldsymbol{T}_i|] = \sum_{j=1}^{d}\Psi\left(\frac{\gamma+\sum_t y_{it}+1-j}{2}\right)-\ln|\boldsymbol{\Phi}_i^*|+d\ln 2. \end{aligned} \tag{A.3}$$

Putting the above details into Eq. (A.2), we have

$$\begin{aligned} \mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\Theta)] = {} & \ln\Gamma(\xi)-\sum_{i=1}^{k}\ln\Gamma(\xi\lambda_i)+\sum_{i=1}^{k}(\xi\lambda_i-1)(\Psi(\xi^*\lambda_i^*)-\Psi(\xi^*)) \\ & +\sum_{i=1}^{k}\Bigg[-\frac{d}{2}\ln(2\pi)+\frac{d}{2}\ln\beta-\frac{\beta(\gamma+\sum_{t=1}^{N}y_{it})}{2}(\boldsymbol{m}_i-\boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i-\boldsymbol{m}_i^*)-\frac{\beta d}{2(\beta+\sum_{t=1}^{N}y_{it})} \\ & +\frac{\gamma}{2}\ln|\boldsymbol{\Phi}|+\frac{\gamma-d}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma+\sum_{t=1}^{N}y_{it}+1-j}{2}\right)-\frac{d^2}{2}\ln 2-\frac{\gamma-d}{2}\ln|\boldsymbol{\Phi}_i^*| \\ & -\frac{\gamma+\sum_{t=1}^{N}y_{it}}{2}\mathrm{Tr}\left(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}\right)-\frac{d(d-1)}{4}\ln\pi-\sum_{j=1}^{d}\ln\Gamma\left(\frac{\gamma+1-j}{2}\right)\Bigg]. \end{aligned} \tag{A.4}$$

Also, we get

$$\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln q(\boldsymbol{X}_N,\boldsymbol{Y}|\Theta)] = \mathrm{E}_{\Theta|\boldsymbol{Y}}\left[\sum_{t=1}^{N}\sum_{i=1}^{k}y_{it}\ln[\alpha_i G(\boldsymbol{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})]\right]=\sum_{t=1}^{N}\sum_{i=1}^{k}y_{it}\mathrm{E}_{\Theta|\boldsymbol{Y}}\left[\ln\left[\alpha_i G(\boldsymbol{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})\right]\right], \tag{A.5}$$

$$\begin{aligned} &\mathrm{E}_{\Theta|\boldsymbol{Y}}\left[\ln[\alpha_i G(\mathbf{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})]\right] = \mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln\alpha_i]+\mathrm{E}_{\Theta|\boldsymbol{Y}}\left[\ln[G(\boldsymbol{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})]\right], \\ &\mathrm{E}_{\Theta|\boldsymbol{Y}}\left[\ln G(\boldsymbol{x}_t|\boldsymbol{\mu}_i,\boldsymbol{T}_i^{-1})\right] = -\frac{d}{2}\ln(2\pi)+\frac{1}{2}\mathrm{E}_{\Theta|\boldsymbol{Y}}[\ln|\boldsymbol{T}_i|]-\frac{1}{2}\mathrm{Tr}\{\mathrm{E}_{\Theta|\boldsymbol{Y}}[\boldsymbol{T}_i(\mathbf{x}_t-\boldsymbol{\mu}_i)(\boldsymbol{x}_t-\boldsymbol{\mu}_i)^{\mathrm{T}}]\}, \end{aligned}$$

$$\mathrm{E}_{\boldsymbol{\Theta}|\boldsymbol{Y}}[\mathbf{T}_i(\boldsymbol{x}_t-\boldsymbol{\mu}_i)(\boldsymbol{x}_t-\boldsymbol{\mu}_i)^{\mathrm{T}}]=\frac{\boldsymbol{I}_d}{\beta+\sum_t y_{it}}+\left(\gamma+\sum_t y_{it}\right)\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_t-\boldsymbol{m}_i^*)(\boldsymbol{x}_t-\boldsymbol{m}_i^*)^{\mathrm{T}}. \tag{A.6}$$

With the help of the above equations, Eq. (A.5) is further expressed as

$$\begin{aligned}\mathrm{E}_{\boldsymbol{\Theta}|\boldsymbol{Y}}[\ln q(\boldsymbol{X}_N,\boldsymbol{Y}|\boldsymbol{\Theta})]=\sum_{i=1}^{k}\sum_{t=1}^{N}y_{it}\Bigg\{&\Psi(\xi^*\lambda_i^*)-\Psi(\xi^*)-\frac{d}{2}\ln(2\pi)+\frac{1}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma+\sum_{t=1}^{N}y_{it}+1-j}{2}\right)+\frac{d}{2}\ln 2\\&-\frac{1}{2}\ln|\boldsymbol{\Phi}_i^*|-\frac{\gamma+\sum_{t=1}^{N}y_{it}}{2}(\boldsymbol{x}_t-\boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_t-\boldsymbol{m}_i^*)-\frac{d}{2(\beta+\sum_{t=1}^{N}y_{it})}\Bigg\}.\end{aligned} \tag{A.7}$$

Adding Eqs. (A.4) and (A.7) and combining some terms, we get

$$\begin{aligned}&\mathrm{E}_{\boldsymbol{\Theta}|\boldsymbol{Y}}[\ln q(\boldsymbol{X}_N,\boldsymbol{Y}|\boldsymbol{\Theta})]+\mathrm{E}_{\boldsymbol{\Theta}|\boldsymbol{Y}}[\ln q(\boldsymbol{\Theta})]\\&=\sum_{i=1}^{k}\sum_{t=1}^{N}y_{it}\left\{\Psi(\xi^*\lambda_i^*)-\Psi(\xi^*)-\frac{1}{2}\ln|\boldsymbol{\Phi}_i^*|-\frac{\gamma+\sum_{t=1}^{N}y_{it}}{2}(\boldsymbol{x}_t-\boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_t-\boldsymbol{m}_i^*)\right\}+\ln\Gamma(\xi)\\&\quad+\sum_{i=1}^{k}\Bigg\{-\ln\Gamma(\xi\lambda_i)+(\xi\lambda_i-1)(\Psi(\xi^*\lambda_i^*)-\Psi(\xi^*))-\frac{\gamma-d}{2}\ln|\boldsymbol{\Phi}_i^*|\\&\quad-\frac{\beta(\gamma+\sum_{t=1}^{N}y_{it})}{2}(\boldsymbol{m}_i-\boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i-\boldsymbol{m}_i^*)-\frac{\gamma+\sum_{t=1}^{N}y_{it}}{2}\mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1})\Bigg\}\\&\quad+\frac{k\gamma}{2}\ln|\boldsymbol{\Phi}|+\frac{kd}{2}\ln\beta-\frac{kd(d+1)}{2}\ln 2-\frac{2Nd+kd(d+1)}{4}\ln\pi-\frac{dk}{2}-k\sum_{j=1}^{d}\ln\Gamma\left(\frac{\gamma+1-j}{2}\right)\\&\quad+\sum_{i=1}^{k}\frac{\gamma+\sum_{t=1}^{N}y_{it}-d}{2}\left(\sum_{j=1}^{d}\Psi\left(\frac{\gamma+\sum_{t=1}^{N}y_{it}+1-j}{2}\right)\right),\end{aligned} \tag{A.8}$$

in which the term $\beta+\sum_{t=1}^{N}y_{it}$ in the denominator has been canceled out according to

$$\sum_{i=1}^{k}\sum_{t=1}^{N}y_{it}\left\{-\frac{d}{2(\beta+\sum_{t=1}^{N}y_{it})}\right\}-\sum_{i=1}^{k}\frac{d\beta}{2(\beta+\sum_{t=1}^{N}y_{it})}=-\frac{dk}{2}. \tag{A.9}$$

It is still difficult to compute Eq. (A.1) by directly using Eq. (A.8). For computational convenience, we further approximate Eq. (A.8) by

$$\mathrm{E}_{\boldsymbol{\Theta}|\boldsymbol{Y}}[\ln q(\boldsymbol{X}_N,\boldsymbol{Y}|\boldsymbol{\Theta})]+\mathrm{E}_{\boldsymbol{\Theta}|\boldsymbol{Y}}[\ln q(\boldsymbol{\Theta})]>\cdots+\textstyle\sum_{i=1}^{k}\frac{\gamma+\sum_{t=1}^{N}y_{it}-d}{2}\left(\sum_{j=1}^{d}\Psi(\frac{\gamma+1-j}{2})\right), \tag{A.10}$$

in which we have applied the monotonic increasing property of the digamma function $\Psi(\cdot)$ on the last term of Eq. (A.8), i.e.,

$$\Psi\left(\frac{\gamma+\sum_{t=1}^{N}y_{it}+1-j}{2}\right)>\Psi\left(\frac{\gamma+1-j}{2}\right). \tag{A.11}$$

Next, we compute the details of Eq. (A.1) by taking expectation $\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}(\cdot)$ on Eq. (A.10) for which we need to calculate the following terms:

$$\begin{aligned}&\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]=p(i|\boldsymbol{x}_t)=p(i|\boldsymbol{x}_t,\boldsymbol{\Xi},\boldsymbol{\lambda}),\\&\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}\left[y_{it}\left(\gamma+\sum_{\tau=1}^{N}y_{i\tau}\right)\right]=\gamma\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]+\sum_{\tau\neq t}\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{i\tau}]+\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}^2]\\&\qquad=\gamma\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]+\sum_{\tau=1}^{N}\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{i\tau}]-\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]+\mathrm{E}_{\boldsymbol{Y}|\boldsymbol{X}}[y_{it}]\\&\qquad=p(i|\boldsymbol{x}_t,\boldsymbol{\Xi},\boldsymbol{\lambda})\left[\gamma+\sum_{\tau=1}^{N}p(i|\boldsymbol{x}_\tau,\boldsymbol{\Xi},\boldsymbol{\lambda})+1-p(i|\boldsymbol{x}_t,\boldsymbol{\Xi},\boldsymbol{\lambda})\right].\end{aligned} \tag{A.12}$$

Based on the above equations, the lower bound of harmony measure given by Eq. (37) is obtained as follows:

$$\begin{aligned}H^{\mathrm{DNW}}(\boldsymbol{\Xi}^*,\boldsymbol{\Xi})=&\sum_{i=1}^{k}\sum_{t=1}^{N}p(i|\boldsymbol{x}_t,\boldsymbol{\Xi},\boldsymbol{\lambda})\ln\left[\alpha_i^p(\xi+N,\boldsymbol{\lambda}^*)G\left(\boldsymbol{x}_t|\boldsymbol{m}_i^*,\frac{\boldsymbol{\Phi}_i^*}{\gamma_i^*+1-p(i|\boldsymbol{x}_t,\boldsymbol{\Xi},\boldsymbol{\lambda})}\right)\right]\\&+\sum_{i=1}^{k}[R_{(\xi+N,\lambda_i^*)}(\xi,\lambda_i)-R(\boldsymbol{\Xi}_i)]+R(\Phi,\beta,\gamma)-R(k)+\mathrm{const},\end{aligned} \tag{A.13}$$

with all notation details listed in Eq. (37).

The learning task is to maximize $H^{\mathrm{DNW}}(\boldsymbol{\Xi}^*, \boldsymbol{\Xi})$ by Eq. (A.13) with respect to $\boldsymbol{\Xi}^* = \{\boldsymbol{\lambda}^*\} \cup \{\boldsymbol{m}_i^*, \boldsymbol{\Phi}_i^*\}_{i=1}^k$ and $\boldsymbol{\Xi} = \{\boldsymbol{\lambda}, \xi, \beta, \boldsymbol{\Phi}, \gamma\} \cup \{\boldsymbol{m}_i\}$. A gradient method is adopted for this maximization. Generally, we update each $\vartheta \in \boldsymbol{\Xi}^* \cup \boldsymbol{\Xi}$ as follows:

$$\vartheta^{\mathrm{new}} = \vartheta^{\mathrm{old}} + \eta \nabla_\vartheta H^{\mathrm{DNW}}, \quad \nabla_\vartheta H^{\mathrm{DNW}} = \left.\frac{\partial H^{\mathrm{DNW}}}{\partial \vartheta}\right|_{\boldsymbol{\Xi}^*=\boldsymbol{\Xi}^{*\mathrm{old}}, \boldsymbol{\Xi}=\boldsymbol{\Xi}^{\mathrm{old}}}. \tag{A.14}$$

The gradient $\nabla_\vartheta H^{\mathrm{DNW}}$ is calculated by

$$\begin{aligned}\nabla_\vartheta H^{\mathrm{DNW}} = &\sum_{t=1}^{N}\sum_{i=1}^{k}\left\{p(i|\boldsymbol{x}_t)\delta_{it}^{(1)} \cdot \nabla_\vartheta \widetilde{\pi}_{it} + p(i|\boldsymbol{x}_t) \cdot \nabla_\vartheta \pi_{it}^{(1)}\right\} + \sum_{i=1}^{k} \nabla_\vartheta R_{(\xi+N),\lambda_i^*}(\xi, \lambda_i) \\ &- \sum_{i=1}^{k} \nabla_\vartheta R(\boldsymbol{\Xi}_i) + \nabla_\vartheta R(\boldsymbol{\Phi}, \beta, \gamma) - \nabla_\vartheta R(k),\end{aligned} \tag{A.15}$$

where the gradient of the first term of H^{DNW} in Eq. (A.13) is similar to Eq. (16), i.e.,

$$\sum_{t=1}^{N}\sum_{i=1}^{k} \nabla_\vartheta p(i|\boldsymbol{x}_t) \cdot \pi_{it}^{(1)} = \sum_{t=1}^{N}\sum_{i=1}^{k} p(i|\boldsymbol{x}_t)\delta_{it}^{(1)} \cdot \nabla_\vartheta \widetilde{\pi}_{it}, \tag{A.16}$$

and

$$\begin{aligned}&\delta_{it}^{(1)} = \pi_{it}^{(1)} - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t)\pi_{jt}^{(1)}, \quad \pi_{it}^{(1)} = \ln\left[\alpha_i^p(\xi+N, \boldsymbol{\lambda}^*) G\left(\boldsymbol{x}_t|\boldsymbol{m}_i^*, \frac{\boldsymbol{\Phi}_i^*}{\gamma_i^* + 1 - p(i|\boldsymbol{x}_t)}\right)\right], \\ &w_{it} = \left[1 + \frac{\beta}{\beta+1}(\boldsymbol{x}_t - \boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{\Phi}^{-1}(\boldsymbol{x}_t - \boldsymbol{m}_i)\right]^{-1}, \quad \widetilde{\pi}_{it} = \ln\left[\lambda_i w_{it}^{\frac{\gamma+1}{2}}\right].\end{aligned} \tag{A.17}$$

For $\vartheta = \lambda_i^* \in \boldsymbol{\Xi}^*$, $\boldsymbol{\Xi}^* = \{\boldsymbol{\lambda}^*\} \cup \{\boldsymbol{m}_i^*, \boldsymbol{\Phi}_i^*\}_{i=1}^k$, we have

$$\begin{aligned}\nabla_{\lambda_i^*} H^{\mathrm{DNW}} &= \sum_{j=1}^{k}\sum_{t=1}^{N} p(j|\boldsymbol{x}_t)\nabla_{\lambda_i^*}\pi_{jt}^{(1)} + \sum_{j=1}^{k}[\nabla_{\lambda_i^*} R_{(\xi+N,\lambda_j^*)}(\xi, \lambda_i)] \\ &\propto \left(\sum_{t=1}^{N} p(i|\boldsymbol{x}_t) + \xi\lambda_i - 1\right)\xi^*[\Psi'(\xi^*\lambda_i^*) - \Psi'(\xi^*)],\end{aligned} \tag{A.18}$$

with $R_{(\xi+N,\lambda_i^*)}(\xi, \lambda_i) = \ln\Gamma(\xi) - \ln\Gamma(\xi\lambda_i) + (\xi\lambda_i - 1)(\Psi((\xi+N)\lambda_i^*) - \Psi(\xi+N))$.

It should be noted that we calculate $\nabla\lambda_i^*\Psi(\xi^*)$ in the above equation by

$$\nabla_{\lambda_i^*}\Psi(\xi^*) = \nabla_{\lambda_i^*}\Psi\left(\xi^* \cdot \sum_{i=1}^{k}\lambda_i^*\right) = \xi^*\Psi'(\xi^*), \quad \text{where } \xi^* = \xi + N, \tag{A.19}$$

which provides a term $\Psi'(\xi^*)$ of the same scale as $\Psi'(\xi^*\lambda_i^*)$ to get a stable gradient by $\Psi'(\xi^*\lambda_i^*) - \Psi'(\xi^*)$. Similarly, we get

$$\begin{aligned}&\nabla_{\boldsymbol{m}_i^*} H^{\mathrm{DNW}} \propto \sum_{t=1}^{N} p(i|\boldsymbol{x}_t)[\gamma_i^* + 1 - p(i|\boldsymbol{x}_t)](\boldsymbol{x}_t - \boldsymbol{m}_i^*) + \beta\gamma_i^*(\boldsymbol{m}_i - \boldsymbol{m}_i^*), \\ &\nabla_{\boldsymbol{\Phi}_i^*} H^{\mathrm{DNW}} \propto \frac{\gamma_i^*}{\gamma_i^* - d}\left[\sum_{t=1}^{N} p(i|\boldsymbol{x}_t)\frac{\gamma_i^* + 1 - p(i|\boldsymbol{x}_t)}{\gamma_i^*}(\boldsymbol{x}_t - \boldsymbol{m}_i^*)(\boldsymbol{x}_t - \boldsymbol{m}_i^*)^{\mathrm{T}} + \beta(\boldsymbol{m}_i - \boldsymbol{m}_i^*)(\boldsymbol{m}_i - \boldsymbol{m}_i^*)^{\mathrm{T}} + \boldsymbol{\Phi}\right] - \boldsymbol{\Phi}_i^*.\end{aligned} \tag{A.20}$$

For $\vartheta = \lambda_i \in \boldsymbol{\Xi}$, $\boldsymbol{\Xi} = \{\boldsymbol{\lambda}, \xi, \beta, \boldsymbol{\Phi}, \gamma\} \cup \{\boldsymbol{m}_i\}$, we have

$$\begin{aligned}\nabla_{\lambda_i} H^{\mathrm{DNW}} &= \sum_{t=1}^{N}\{p(i|\boldsymbol{x}_t)\delta_{it}\nabla_{\lambda_i}\widetilde{\pi}_{it}\} + \nabla_{\lambda_i}\sum_{j=1}^{k}[R_{(\xi+N,\lambda_i^*)}(\xi, \lambda_i) - R(\boldsymbol{\Xi}_j)] \\ &\propto \sum_{t=1}^{N} p(i|\boldsymbol{x}_t)\delta_{it} + \xi\lambda_i[\Psi(\xi) - \Psi(\xi\lambda_i) + \Psi(\xi^*\lambda_i^*) - \Psi(\xi^*)], \\ &\text{with } \delta_{it} = \delta_{it}^{(1)} + \delta_{it}^{(2)} + \delta_{it}^{(3)},\end{aligned} \tag{A.21}$$

where $\delta_{it}^{(2)}$ comes from taking the derivatives of $(\gamma_i^* + 1 - p(i|\boldsymbol{x}_t, \boldsymbol{\Xi}, \boldsymbol{\lambda}))$ in $\pi_{it}^{(1)}$, and $\delta_{it}^{(3)}$ comes from taking the derivatives of γ_i^* in $R(\boldsymbol{\Xi}_i)$. Particularly, they can be obtained by rewriting

$$\sum_{t=1}^{N}\sum_{j=1}^{k} p(j|\boldsymbol{x}_t) \cdot \nabla_{\lambda_i}\pi_{jt}^{(1)} = \sum_{t=1}^{N}\sum_{j=1}^{k} \nabla_{\lambda_i} p(j|\boldsymbol{x}_t) \cdot \pi_{jt}^{(2)} + \cdots, \qquad -\sum_{j=1}^{k} \nabla_\vartheta R(\boldsymbol{\Xi}_j) = \sum_{t=1}^{N}\sum_{j=1}^{k} \nabla_{\lambda_i} p(j|\boldsymbol{x}_t) \cdot \pi_{jt}^{(3)} + \cdots, \tag{A.22}$$

and then the Eq. (A.21) follows from Eq. (A.16) with

$$\delta_{it}^{(2)} = \pi_{it}^{(2)} - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t)\pi_{jt}^{(2)}, \quad \pi_{it}^{(2)} = -\frac{1}{2}\sum_{\tau\neq t} p(i|\boldsymbol{x}_\tau)(\boldsymbol{x}_\tau - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_\tau - \boldsymbol{m}_i^*),$$
$$\delta_{it}^{(3)} = \pi_{i}^{(3)} - \sum_{j=1}^{k} p(j|\boldsymbol{x}_t)\pi_{j}^{(3)}, \quad \pi_{i}^{(3)} = -\frac{\beta}{2}(\boldsymbol{m}_i - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i - \boldsymbol{m}_i^*) - \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}). \tag{A.23}$$

Similar to Eq. (A.19), we calculate $\nabla_{\lambda_i}\ln\Gamma(\xi)$ in the above equation by

$$\nabla_{\lambda_i}\ln\Gamma(\xi) = \nabla_{\lambda_i}\ln\Gamma\left(\xi\cdot\sum_{j=1}^{k}\lambda_j\right) = \Psi(\xi)\xi\cdot\nabla_{\lambda_i}\left(\sum_{j=1}^{k}\lambda_j\right) = \xi\Psi(\xi), \tag{A.24}$$

which provides a term $\Psi(\xi)$ of the same scale as $\Psi(\xi\lambda_i)$ to get a stable gradient by $\Psi(\xi) - \Psi(\xi\lambda_i)$.

Similarly, we get

$$\begin{aligned}
\nabla_\xi H^{\mathrm{DNW}} &\propto \sum_{i=1}^{k}\lambda_i[\Psi(\xi) - \Psi(\xi\lambda_i) + \Psi(\xi^*\lambda_i^*) - \Psi(\xi^*)] + \sum_{i=1}^{k}\left(\sum_{t=1}^{N} p(i|\boldsymbol{x}_t) + \xi\lambda_i - 1\right)[\lambda_i^*\Psi'(\xi^*\lambda_i^*) - \Psi'(\xi^*)],\\
\nabla_{\boldsymbol{m}_i} H^{\mathrm{DNW}} &\propto \frac{\gamma+1}{\beta+1}\sum_{t=1}^{N} p(i|\boldsymbol{x}_t)\delta_{it}w_{it}\boldsymbol{\Phi}^{-1}(\boldsymbol{x}_t - \boldsymbol{m}_i) + \gamma_i^*\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i^* - \boldsymbol{m}_i),\\
\nabla_\beta H^{\mathrm{DNW}} &\propto \frac{kd}{\beta} - \frac{\gamma+1}{\beta+1}\sum_{t=1}^{N}\sum_{i=1}^{k} p(i|\boldsymbol{x}_t)\delta_{it}w_{it}[1 + (\boldsymbol{x}_t - \boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{\Phi}^{-1}(\boldsymbol{x}_t - \boldsymbol{m}_i)] - \sum_{i=1}^{k}\gamma_i^*(\boldsymbol{m}_i - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i - \boldsymbol{m}_i^*),\\
\nabla_{\boldsymbol{\Phi}} H^{\mathrm{DNW}} &\propto \frac{\beta(\gamma+1)}{\beta+1}\sum_{t=1}^{N}\sum_{i=1}^{k} p(i|\boldsymbol{x}_t)\delta_{it}w_{it}\boldsymbol{\Phi}^{-1}(\boldsymbol{x}_t - \boldsymbol{m}_i)(\boldsymbol{x}_t - \boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{\Phi}^{-1} + \gamma k\boldsymbol{\Phi}^{-1} - \sum_{i=1}^{k}\gamma_i^*\boldsymbol{\Phi}_i^{*-1},\\
\nabla_\gamma H^{\mathrm{DNW}} &\propto \sum_{t=1}^{N}\sum_{i=1}^{k} p(i|\boldsymbol{x}_t)\delta_{it}\ln w_{it} + k\ln|\boldsymbol{\Phi}| + \frac{N + k(\gamma-d)}{2}\sum_{j=1}^{d}\Psi'\left(\frac{\gamma+1-j}{2}\right)\\
&\quad - \sum_{i=1}^{k}[\ln|\boldsymbol{\Phi}_i^*| + \mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}) + \beta(\boldsymbol{m}_i - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i - \boldsymbol{m}_i^*)],
\end{aligned} \tag{A.25}$$

where the term $\delta_{it}\ln w_{it}$ in the equation for γ comes from the part $\frac{\partial w_{it}^{(\gamma+1)/2}}{\partial\gamma}$, and is different from the former three terms $\delta_{it}w_{it}$ in the equations for $\boldsymbol{m}_i$, β and $\boldsymbol{\Phi}$.

Appendix B MML

One major difficulty in implementing MML lies in the computation of the determinant of the Fisher information matrix $\boldsymbol{I}(\boldsymbol{\Theta})$. In Eq. (25), the determinant of $\boldsymbol{I}(\boldsymbol{\Theta})$ is calculated by Eq. (7), where $\boldsymbol{I}(\boldsymbol{\Theta})$ is approximated by a block-diagonal complete-data Fisher information matrix $\boldsymbol{I}_c(\boldsymbol{\Theta})$ in Eq. (6) which is taken from Ref. [18]. Alternatively, we provide another equivalent derivation of Eq. (8) by using the following lower-bound trick similar to Eqs. (21) and (22) for tackling $\boldsymbol{\Pi}$ in the VB-Jef algorithm:

$$\begin{aligned}
&\ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) \approx u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\}) - \ln q(\boldsymbol{\Theta}),\\
&\boldsymbol{I}(\boldsymbol{\Theta}) \approx \text{Block-Diag}[\boldsymbol{I}(\boldsymbol{\alpha}), \boldsymbol{I}(\boldsymbol{\mu}_1), \boldsymbol{I}(\boldsymbol{\mu}_2), \ldots, \boldsymbol{I}(\boldsymbol{\mu}_k), \boldsymbol{I}(\boldsymbol{T}_1), \boldsymbol{I}(\boldsymbol{T}_2), \ldots, \boldsymbol{I}(\boldsymbol{T}_k)],\\
&\boldsymbol{I}(\boldsymbol{\alpha}) = -\mathrm{E}\left[\frac{\partial^2[u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\}) - \ln q(\boldsymbol{\Theta})]}{\partial\boldsymbol{\alpha}\partial\boldsymbol{\alpha}^{\mathrm{T}}}\right], \quad |\boldsymbol{I}(\boldsymbol{\alpha})| = \prod_{i=1}^{k}\frac{N}{\alpha_i},\\
&\boldsymbol{I}(\boldsymbol{\mu}_i) = -\mathrm{E}\left[\frac{\partial^2[u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\}) - \ln q(\boldsymbol{\Theta})]}{\partial\boldsymbol{\mu}_i\partial\boldsymbol{\mu}_i^{\mathrm{T}}}\right], \quad |\boldsymbol{I}(\boldsymbol{\mu}_i)| = (N\alpha_i)^d|\boldsymbol{T}_i|,\\
&\boldsymbol{I}(\boldsymbol{T}_i) = -\mathrm{E}\left[\frac{\partial^2[u^{\mathrm{lb}}(\boldsymbol{\Theta}, \{p_{it}\}) - \ln q(\boldsymbol{\Theta})]}{\partial\mathrm{vec}(\boldsymbol{T}_i)\partial\mathrm{vec}(\boldsymbol{T}_i)^{\mathrm{T}}}\right], \quad |\boldsymbol{I}(\boldsymbol{T}_i)| = (N\alpha_i)^{d(d+1)/2}|\boldsymbol{T}_i|^{-(d+1)}.
\end{aligned} \tag{B.1}$$

The computational details of MML-Jef to maximize $J_{\mathrm{MML}}^{\mathrm{Jef}}(\boldsymbol{\Theta})$ by Eq. (25) are listed in Table 2. It should be noted that the MML objective used in Ref. [18] is $J_{\mathrm{MML}}^{\mathrm{Jef}}(\boldsymbol{\Theta}) + k\rho(1 - \ln 12)$, with an additional term (coming from an optimal quantization [18]) which is irrelevant to $\boldsymbol{\Theta}$ and thus takes no effect on automatic model selection. Therefore, the Eq. (25) is equivalent to the MML objective in Ref. [18] for automatic model selection.

Moreover, if we replace the Jeffreys prior with the DNW prior, the MML objective function in Eq. (24) becomes the one in Eq. (38), which is expressed in details as follows:

$$J_{\mathrm{MML}}^{\mathrm{DNW}}(\boldsymbol{\Theta}, \boldsymbol{\Xi}) = \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) + \ln\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\lambda}, \xi) + \sum_{i=1}^{k}\ln G(\boldsymbol{\mu}_i|\boldsymbol{m}_i, \boldsymbol{T}_i^{-1}/\beta) + \sum_{i=1}^{k}\ln\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi}, \gamma) - \frac{1}{2}\ln|\boldsymbol{I}_c(\boldsymbol{\Theta})|$$

$$= \ln q(\boldsymbol{X}_N|\boldsymbol{\Theta}) + \sum_{i=1}^{k}\left(\xi\lambda_i - \frac{\rho+1}{2}\right)\ln\alpha_i + \frac{\gamma}{2}\sum_{i=1}^{k}\ln|\boldsymbol{T}_i| - \frac{\beta}{2}\sum_{i=1}^{k}(\boldsymbol{\mu}_i-\boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{T}_i(\boldsymbol{\mu}_i-\boldsymbol{m}_i) - \frac{1}{2}\sum_{i=1}^{k}\mathrm{Tr}(\boldsymbol{T}_i\boldsymbol{\Phi})$$
$$+ \ln\Gamma(\xi) - \sum_{i=1}^{k}\ln\Gamma(\xi\lambda_i) + \frac{kd}{2}\ln\beta - \frac{kd}{2}\ln(2\pi) + \frac{k\gamma}{2}\ln|\boldsymbol{\Phi}| - \frac{kd\gamma}{2}\ln 2 - k\ln\Gamma_d\left(\frac{\gamma}{2}\right) - \frac{k\rho}{2}\ln N. \quad \text{(B.2)}$$

The $J_{\mathrm{MML}}^{\mathrm{DNW}}(\boldsymbol{\Theta},\boldsymbol{\Xi})$ by Eq. (B.2) is different from $J_{\mathrm{MML}}^{\mathrm{Jef}}(\boldsymbol{\Theta},\boldsymbol{\Xi})$ by Eq. (25) in the term $\ln q(\boldsymbol{\Theta})$ with $q(\boldsymbol{\Theta})$ being DNW prior instead of Jeffreys prior. Then, the algorithm (shortly denoted as MML-DNW) to maximize Eq. (B.2) is obtained by modifying MML-Jef accordingly with details given in Table B1 Specifically, the MML-DNW algorithm becomes the same as the MML-Jef algorithm in Ref. [18], if we skip the H-step and fix the hyper-parameters as $\xi = k/2$, each $\lambda_i = 1/k$, $\beta = 0$, $\boldsymbol{\Phi} = \boldsymbol{0}$ and $\gamma = 0$.

Table B1 MML algorithm on GMM with a Dirichlet-Normal-Wishart prior (MML-DNW) (Its implementation is equivalent to MML-Jef in Ref. [18] when we skip H-step and fix $\xi = k/2$, $\boldsymbol{\lambda} = (1/k,\ldots,1/k)^{\mathrm{T}}$, $\beta = 0$, $\boldsymbol{\Phi} = \boldsymbol{0}$ and $\gamma = 0$.)

Initialization: Randomly initialize the model with a large enough number k of components; set $\tau = 0$ and the MML objective function $J_{\mathrm{MML}}(\tau) = -\infty$;

repeat

E-step: Get $\alpha_i^p = \alpha_i^{\mathrm{old}}$ for $i = 1, 2, \ldots, k$, then get p_{it} and n_i by Eq. (3);

M-step: Update the parameters $\boldsymbol{\Theta} = \boldsymbol{\alpha} \cup \{\boldsymbol{\mu}_i, \boldsymbol{T}_i\}_{i=1}^{k}$:

$$\alpha_i^{\mathrm{new}} = \frac{s_i}{\sum_{j=1}^{k} s_j}, \qquad s_i = \sum_{t=1}^{N} p_{it} + \xi^{\mathrm{old}}\lambda_i^{\mathrm{old}} - \frac{\rho+1}{2}, \qquad \rho = d + \frac{d(d+1)}{2},$$
$$\boldsymbol{\mu}_i^{\mathrm{new}} = \frac{1}{n_i + \beta^{\mathrm{old}}}\left[\sum_{t=1}^{N} p_{it}\boldsymbol{x}_t + \beta^{\mathrm{old}}\boldsymbol{m}_i^{\mathrm{old}}\right], \quad \boldsymbol{e}_{it} = \boldsymbol{x}_t - \boldsymbol{\mu}_i^{\mathrm{old}},$$
$$\boldsymbol{T}_i^{-1\,\mathrm{new}} = \frac{1}{n_i + \gamma^{\mathrm{old}}}\left[\sum_{t=1}^{N} p_{it}\boldsymbol{e}_{it}\boldsymbol{e}_{it}^{\mathrm{T}} + \beta^{\mathrm{old}}(\boldsymbol{\mu}_i^{\mathrm{old}} - \boldsymbol{m}_i^{\mathrm{old}})(\boldsymbol{\mu}_i^{\mathrm{old}} - \boldsymbol{m}_i^{\mathrm{old}})^{\mathrm{T}} + \boldsymbol{\Phi}^{\mathrm{old}}\right];$$

H-step: Update the prior hyper-parameters $\{\boldsymbol{m}_i\}$ in the following two cases:

1) General case (each $\boldsymbol{m}_i$ is free): $\boldsymbol{m}_i^{\mathrm{new}} = \boldsymbol{\mu}_i^{\mathrm{old}}$;

2) Special case (constrain each $\boldsymbol{m}_i = \boldsymbol{m}$): $\forall i$, $\boldsymbol{m}_i^{\mathrm{new}} = (\sum_{i=1}^{k}\boldsymbol{T}_i^{\mathrm{old}})^{-1}(\sum_{i=1}^{k}\boldsymbol{T}_i^{\mathrm{old}}\boldsymbol{\mu}_i^{\mathrm{old}})$;

Then update the prior hyper-parameters $\{\boldsymbol{\lambda}, \xi, \beta, \boldsymbol{\Phi}, \gamma\}$:

$$\lambda_i^{\mathrm{new}} = \frac{\lambda_i^{\mathrm{old}} + \eta\delta\lambda_i}{\sum_{j=1}^{k}(\lambda_j^{\mathrm{old}} + \eta\delta\lambda_j)}, \quad \delta\lambda_i = \ln\alpha_i^{\mathrm{old}} - \Psi(\xi^{\mathrm{old}}\lambda_i^{\mathrm{old}}) + \Psi(\xi^{\mathrm{old}}),$$
$$\xi^{\mathrm{new}} = \xi^{\mathrm{old}} + \eta\delta\xi, \qquad \delta\xi = \sum_{i=1}^{k}\lambda_i^{\mathrm{old}}\delta\lambda_i,$$
$$\beta^{\mathrm{new}} = \frac{kd}{\sum_{i=1}^{k}(\boldsymbol{\mu}_i^{\mathrm{old}} - \boldsymbol{m}_i^{\mathrm{old}})^{\mathrm{T}}\boldsymbol{T}_i^{\mathrm{old}}(\boldsymbol{\mu}_i^{\mathrm{old}} - \boldsymbol{m}_i^{\mathrm{old}})},$$
$$\boldsymbol{\Phi}^{\mathrm{new}} = k\gamma^{\mathrm{old}}\left(\sum_{i=1}^{k}\boldsymbol{T}_i^{\mathrm{old}}\right)^{-1}, \quad \gamma^{\mathrm{new}} = \gamma^{\mathrm{old}} + \eta\delta\gamma(\boldsymbol{\Theta}^{\mathrm{old}}, \boldsymbol{\Xi}^{\mathrm{old}}),$$
$$\delta\gamma(\boldsymbol{\Theta},\boldsymbol{\Xi}) = \sum_{i=1}^{k}\ln|\boldsymbol{T}_i| + k\ln|\boldsymbol{\Phi}| - kd\ln 2 - k\sum_{j=1}^{d}\Psi\left(\frac{\gamma+1-j}{2}\right);$$

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\alpha_i \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another* 5 *runs have passed* **then** let $\tau = \tau + 1$; calculate the MML objective function as $J_{\mathrm{MML}}(\tau)$ by Eq. (B.2);

until $J_{\mathrm{MML}}(\tau) - J_{\mathrm{MML}}(\tau-1) < \epsilon J_{\mathrm{MML}}(\tau-1)$, *with* $\epsilon = 10^{-5}$ *in our implementation*;

Appendix C VB

The variational lower bound $J_{\mathrm{VB}}^{\mathrm{DNW}}$ given in Eq. (40) with the DNW prior can be rewritten as follows:

$$J_{\mathrm{VB}}^{\mathrm{DNW}}(\boldsymbol{\Xi}^*, \boldsymbol{\Xi}) = \mathrm{E}_{p_{\boldsymbol{Y}}}\mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta})] - \mathrm{E}_{p_{\boldsymbol{Y}}}[\ln p(\boldsymbol{Y})] + \mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln q(\boldsymbol{\Theta})] - \mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln p(\boldsymbol{\Theta})], \quad \text{(C.1)}$$

where $\mathrm{E}_{p_{\boldsymbol{\Theta}}}[\cdot]$ and $\mathrm{E}_{p_{\boldsymbol{Y}}}[\cdot]$ denotes expectations with respect to the variational posteriors $p(\boldsymbol{\Theta})$ and $p(\boldsymbol{Y})$ in Eq. (39).

Similar to Eqs. (A.7) and (A.4), we have

$$\mathrm{E}_{p_{\boldsymbol{Y}}}\mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln q(\boldsymbol{X}_N, \boldsymbol{Y}|\boldsymbol{\Theta})] = \sum_{i=1}^{k}\sum_{t=1}^{N} p_{it}\left\{\Psi(\xi^*\lambda_i^*) - \Psi(\xi^*) - \frac{d}{2}\ln(2\pi) + \frac{1}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma_i^*+1-j}{2}\right)\right.$$
$$\left. + \frac{d}{2}\ln 2 - \frac{1}{2}\ln|\boldsymbol{\Phi}_i^*| - \frac{\gamma_i^*}{2}(\boldsymbol{x}_t - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_t - \boldsymbol{m}_i^*) - \frac{d}{2\beta_i^*}\right\}, \quad \text{(C.2)}$$

$$\mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln q(\boldsymbol{\Theta})] = \ln\Gamma(\xi) - \sum_{i=1}^{k}\ln\Gamma(\xi\lambda_i) + \sum_{i=1}^{k}(\xi\lambda_i - 1)(\Psi(\xi^*\lambda_i^*) - \Psi(\xi^*)) + \sum_{i=1}^{k}\Bigg[- \frac{d}{2}\ln(2\pi) + \frac{d}{2}\ln\beta - \frac{\beta d}{2\beta_i^*} + \frac{\gamma}{2}\ln|\boldsymbol{\Phi}|$$
$$+\frac{\gamma - d}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma_i^* + 1 - j}{2}\right) - \frac{d^2}{2}\ln 2 - \frac{\beta\gamma_i^*}{2}(\boldsymbol{m}_i - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i - \boldsymbol{m}_i^*) - \frac{\gamma - d}{2}\ln|\boldsymbol{\Phi}_i^*|$$
$$-\frac{\gamma_i^*}{2}\mathrm{Tr}(\boldsymbol{\Phi}\boldsymbol{\Phi}_i^{*-1}) - \frac{d(d-1)}{4}\ln\pi - \sum_{j=1}^{d}\ln\Gamma\left(\frac{\gamma + 1 - j}{2}\right)\Bigg]. \tag{C.3}$$

Since $p(\boldsymbol{\Theta})$ and $q(\boldsymbol{\Theta})$ are both DNW distributions with different parameters $\boldsymbol{\Xi}^* = \{\boldsymbol{\lambda}^*, \xi^*, \boldsymbol{m}_i^*, \beta_i^*, \boldsymbol{\Phi}_i^*, \gamma_i^*\}$ and $\boldsymbol{\Xi} = \{\boldsymbol{\lambda}, \xi, \{\boldsymbol{m}_i\}, \beta, \boldsymbol{\Phi}, \gamma\}$, $\mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln p(\boldsymbol{\Theta})]$ is obtained by replacing $\boldsymbol{\Xi}$ with $\boldsymbol{\Xi}^*$ in Eq. (C.3) appropriately:

$$\mathrm{E}_{p_{\boldsymbol{\Theta}}}[\ln p(\boldsymbol{\Theta})] = \ln\Gamma(\xi^*) - \sum_{i=1}^{k}\ln\Gamma(\xi^*\lambda_i^*) + \sum_{i=1}^{k}(\xi^*\lambda_i^* - 1)(\Psi(\xi^*\lambda_i^*) - \Psi(\xi^*)) + \sum_{i=1}^{k}\Bigg[- \frac{d}{2}\ln(2\pi) + \frac{d}{2}\ln\beta_i^* - \frac{\beta_i^* d}{2\beta_i^*} + \frac{\gamma_i^*}{2}\ln|\boldsymbol{\Phi}_i^*|$$
$$+\frac{\gamma_i^* - d}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma_i^* + 1 - j}{2}\right) - \frac{d^2}{2}\ln 2 - \frac{\beta_i^*\gamma_i^*}{2}(\boldsymbol{m}_i^* - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i^* - \boldsymbol{m}_i^*) - \frac{\gamma_i^* - d}{2}\ln|\boldsymbol{\Phi}_i^*|$$
$$-\frac{\gamma_i^*}{2}\mathrm{Tr}(\boldsymbol{\Phi}_i^*\boldsymbol{\Phi}_i^{*-1}) - \frac{d(d-1)}{4}\ln\pi - \sum_{j=1}^{d}\ln\Gamma\left(\frac{\gamma_i^* + 1 - j}{2}\right)\Bigg], \tag{C.4}$$

$$\mathrm{E}_{p_{\boldsymbol{Y}}}[\ln q(\boldsymbol{Y})] = \sum_{t=1}^{N}\sum_{i=1}^{k}\mathrm{E}_{p_{\boldsymbol{Y}}}[y_{it}]\ln p_{it} = \sum_{t=1}^{N}\sum_{i=1}^{k}p_{it}\ln p_{it}, \tag{C.5}$$

where $\{p_{it}\}$ in Eq. (C.5) are the parameters of the variational posterior $p(\boldsymbol{Y})$ as given in Eq. (19).

Putting the details of Eqs. (C.2)–(C.5) into Eq. (C.1) and combining like terms, we have

$$J_{\mathrm{VB}}^{\mathrm{DNW}} = \sum_{t=1}^{N}\sum_{i=1}^{k}p_{it}\Bigg[- \frac{1}{2}\ln|\boldsymbol{\Phi}_i^*| + \frac{1}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma_i^* + 1 - j}{2}\right) - \frac{\gamma_i^*}{2}(\boldsymbol{x}_t - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_t - \boldsymbol{m}_i^*) - \frac{d}{2\beta_i^*}$$
$$+\Psi(\xi^*\lambda_i^*) - \ln p_{it}\Bigg] + \sum_{i=1}^{k}\Bigg[\frac{d}{2}\ln\frac{\beta}{\beta_i^*} - \frac{\beta\gamma_i^*}{2}(\boldsymbol{m}_i^* - \boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{m}_i^* - \boldsymbol{m}_i) + \frac{\gamma}{2}(\ln|\boldsymbol{\Phi}| - \ln|\boldsymbol{\Phi}_i^*|)$$
$$-\frac{d\beta}{2\beta_i^*} + \frac{\gamma - \gamma_i^*}{2}\sum_{j=1}^{d}\Psi\left(\frac{\gamma_i^* + 1 - j}{2}\right) - \sum_{j=1}^{d}\ln\frac{\Gamma(\frac{\gamma+1-j}{2})}{\Gamma(\frac{\gamma_i^*+1-j}{2})} - \frac{\gamma_i^*}{2}\mathrm{Tr}(\boldsymbol{\Phi}_i^{*-1}\boldsymbol{\Phi}) + \frac{\gamma_i^* d}{2}\Bigg] - \frac{Nd}{2}\ln\pi$$
$$-(N + \xi - \xi^*)\Psi(\xi^*) + \frac{kd}{2} + \ln\frac{\Gamma(\xi)}{\Gamma(\xi^*)} - \sum_{i=1}^{k}\ln\frac{\Gamma(\xi\lambda_i)}{\Gamma(\xi^*\lambda_i^*)} + \sum_{i=1}^{k}(\xi\lambda_i - \xi^*\lambda_i^*)\Psi(\xi^*\lambda_i^*), \tag{C.6}$$

an algorithm maximizes which is shortly denoted as VB-DNW and listed in Table C1.

Table C1 VB algorithm on GMM with a DNW prior (VB-DNW)

Initialization: Randomly initialize the model with a large enough number k of components; set $\tau = 0$ and the variational function $J_{\mathrm{VB}}(\tau) = -\infty$;

repeat

E-step: Estimate the component responsibilities p_{it} for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, N$:

$$p_{it} = \frac{r_{it}^{1/2}}{\sum_{j=1}^{k} r_{jt}^{1/2}}, \qquad \text{with}$$
$$r_{it} = \exp\left[- \ln|\boldsymbol{\Phi}_i^*| + \sum_{j=1}^{d}\Psi\left(\frac{\gamma_i^* + 1 - j}{2}\right) - \gamma_i^*(\boldsymbol{x}_t - \boldsymbol{m}_i^*)^{\mathrm{T}}\boldsymbol{\Phi}_i^{*-1}(\boldsymbol{x}_t - \boldsymbol{m}_i^*) - \frac{d}{\beta_i^*} + 2\Psi(\xi^*\lambda_i^*)\right];$$

M-step: Update the posterior hyper-parameters $\boldsymbol{\Xi}^* = \{\boldsymbol{\lambda}^*, \xi^*\} \cup \{\boldsymbol{m}_i^*, \beta_i^*, \boldsymbol{\Phi}_i^*, \gamma_i^*\}_{i=1}^k$:

$$\lambda_i^{*\mathrm{new}} = \frac{\xi^{\mathrm{old}}\lambda_i^{\mathrm{old}} + \sum_{t=1}^{N} p_{it}}{\xi^{\mathrm{old}} + N}, \qquad \xi^{*\mathrm{new}} = \xi^{\mathrm{old}} + N,$$
$$\boldsymbol{m}_i^{*\mathrm{new}} = \frac{\sum_{t=1}^{N} p_{it}\boldsymbol{x}_t + \beta^{\mathrm{old}}\boldsymbol{m}_i^{\mathrm{old}}}{\sum_{t=1}^{N} p_{it} + \beta^{\mathrm{old}}}, \qquad \beta_i^{*\mathrm{new}} = \beta^{\mathrm{old}} + \sum_{t=1}^{N} p_{it}, \quad \boldsymbol{e}_{it} = \boldsymbol{x}_t - \boldsymbol{m}_i^{*\mathrm{old}},$$
$$\boldsymbol{\Phi}_i^{*\mathrm{new}} = \sum_{t=1}^{N} p_{it}\boldsymbol{e}_{it}\boldsymbol{e}_{it}^{\mathrm{T}} + \beta(\boldsymbol{m}_i^{\mathrm{old}} - \boldsymbol{m}_i^{*\mathrm{old}})(\boldsymbol{m}_i^{\mathrm{old}} - \boldsymbol{m}_i^{*\mathrm{old}})^{\mathrm{T}} + \boldsymbol{\Phi}^{\mathrm{old}}, \quad \gamma_i^{*\mathrm{new}} = \gamma^{\mathrm{old}} + \sum_{t=1}^{N} p_{it};$$

H-step: Update the prior hyper-parameters $\{\boldsymbol{m}_i\}$ in the following two cases:

1) General case (each $\boldsymbol{m}_i$ is free): $\boldsymbol{m}_i^{\mathrm{new}} = \boldsymbol{m}_i^{*\mathrm{old}}$;

2) Special case (constrain each $\boldsymbol{m}_i = \boldsymbol{m}$):

$\forall i,\ \boldsymbol{m}_i^{\mathrm{new}} = (\sum_{i=1}^{k}\gamma_i^{*\mathrm{old}}\boldsymbol{\Phi}_i^{*\mathrm{old}\,-1})^{-1}(\sum_{i=1}^{k}\gamma_i^{*\mathrm{old}}\boldsymbol{\Phi}_i^{*\mathrm{old}\,-1}\boldsymbol{m}_i^{*\mathrm{old}})$;

Table C1 VB algorithm on GMM with a DNW prior (VB-DNW) (Continued)

Then update the prior hyper-parameters $\Xi = \{\boldsymbol{\lambda}, \xi, \beta, \boldsymbol{\Phi}, \gamma\}$:

$$\lambda_i^{\text{new}} = \frac{\lambda_i^{\text{old}} + \eta\delta\lambda_i}{\sum_{j=1}^{k}(\lambda_j^{\text{old}} + \eta\delta\lambda_j)}, \quad \delta\lambda_i = \Psi(\xi^{*\text{old}}\lambda_i^{*\text{old}}) - \Psi(\xi^{*\text{old}}) - \Psi(\xi^{\text{old}}\lambda_i^{\text{old}}) + \Psi(\xi^{\text{old}}),$$

$$\xi^{\text{new}} = \xi^{\text{old}} + \eta\delta\xi, \quad \delta\xi = \sum_{i=1}^{k}\lambda_i^{\text{old}}\delta\lambda_i,$$

$$\beta^{\text{new}} = \frac{kd}{\sum_{i=1}^{k}[\gamma_i^{*\text{old}}(\boldsymbol{m}_i^{*\text{old}} - \boldsymbol{m}_i^{\text{old}})^{\text{T}}\boldsymbol{\Phi}_i^{*\text{old}\,-1}(\boldsymbol{m}_i^{*\text{old}} - \boldsymbol{m}_i^{\text{old}}) + d/\beta_i^{*\text{old}}]},$$

$$\boldsymbol{\Phi}^{\text{new}} = k\gamma^{\text{old}}\left(\sum_{i=1}^{k}\gamma_i^{*\text{old}}\boldsymbol{\Phi}_i^{*\text{old}\,-1}\right)^{-1},$$

$$\gamma^{\text{new}} = \gamma^{\text{old}} + \eta\sum_{i=1}^{k}\left\{\ln|\boldsymbol{\Phi}^{\text{old}}| - \ln|\boldsymbol{\Phi}_i^{*\text{old}}| + \sum_{j=1}^{d}\left[\Psi\left(\frac{\gamma_i^{*\text{old}} + 1 - j}{2}\right) - \Psi\left(\frac{\gamma^{\text{old}} + 1 - j}{2}\right)\right]\right\};$$

$\odot$ **for** $i = 1, 2, \ldots, k$ **do if** $\lambda_i^* \to 0$ **then** discard component i, let $k = k - 1$ and **continue**;

if *another* 5 *runs have passed* **then** let $\tau = \tau + 1$; calculate the variational function as $J_{\text{VB}}(\tau)$ by Eq. (40);

until $J_{\text{VB}}(\tau) - J_{\text{VB}}(\tau - 1) < \epsilon J_{\text{VB}}(\tau - 1)$, *with* $\epsilon = 10^{-5}$ *in our implementation*;

With the prior and posterior on $\boldsymbol{\theta}_i$ being both joint Normal-Wishart suggested in Ref. [25], this VB-DNW algorithm is different from the VB algorithm (shortly denoted as VB-iDNW) on GMM studied in Ref. [20], where both $q(\boldsymbol{\theta}_i)$ and $p(\boldsymbol{\theta}_i)$ were assumed as *independent* Normal-Wishart. Particularly, the independent NW prior takes $q(\boldsymbol{\theta}_i) = G(\boldsymbol{\mu}_i|\boldsymbol{m}_i, \boldsymbol{A})\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi}, \gamma)$ and therein $\boldsymbol{\mu}_i$ is independent from $\boldsymbol{T}_i$, and such an independence also exists in the posterior $p(\boldsymbol{\theta}_i) = G(\boldsymbol{\mu}_i|\boldsymbol{m}_i^*, \boldsymbol{A}_i^*)\mathcal{W}(\boldsymbol{T}_i|\boldsymbol{\Phi}_i^*, \gamma_i^*)$. Comparing the implementations of VB-DNW and VB-iDNW, they have one key difference that all computation formulas related to $p(\boldsymbol{\theta}_i)$ in VB-iDNW are modified accordingly as the independent NW prior is replaced by the joint NW prior. Especially, to update the posterior $p(\boldsymbol{\theta}_i)$ in the M-step, updating $\{\boldsymbol{m}_i^*, \boldsymbol{A}_i^*, \boldsymbol{\Phi}_i^*, \gamma_i^*\}$ in VB-iDNW is modified to updating $\{\boldsymbol{m}_i^*, \beta_i^*, \boldsymbol{\Phi}_i^*, \gamma_i^*\}$ in VB-DNW with the number of free hyper-parameters greatly reduced.

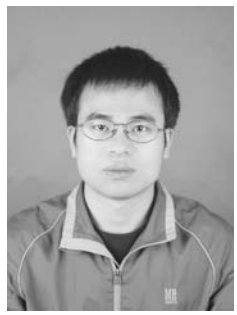
Acknowledgements The work described in this paper was supported by a grant of the General Research Fund (GRF) from the Research Grant Council of Hong Kong SAR (Project No. CUHK418011E).

References

1. Constantinopoulos C, Titsias M K. Bayesian feature and model selection for Gaussian mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(6): 1013–1018
2. Redner R, Walker H. Mixture densities, maximum likelihood and the EM algorithm. SIAM Review, 1984, 26(2): 195–239
3. Engel A, den Broeck C P L V. Statistical Mechanics of Learning. New York: Cambridge University Press, 2001
4. Constantinopoulos C, Likas A. Unsupervised learning of Gaussian mixtures based on variational component splitting. IEEE Transactions on Neural Networks, 2007, 18(3): 745–755
5. Verbeek J J, Vlassis N, Krose B. Efficient greedy learning of Gaussian mixture models. Neural Computation, 2003, 15(2): 469–485
6. Xu L, Jordan M I. On convergence properties of the EM algorithm for Gaussian mixtures. Neural Computation, 1996, 8(1):129–151
7. Mclachlan G J, Krishnan T. The EM Algorithm and Extensions (Wiley Series in Probability and Statistics). New York: Wiley-Interscience, 2007
8. Akaike H. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 1974, 19(6): 716–723
9. Schwarz G. Estimating the dimension of a model. The Annals of Statistics, 1978, 6(2): 461–464
10. Barron A R, Rissanen J, Yu B. The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory, 1998, 44(6): 2743–2760
11. Rissanen J. Modelling by the shortest data description. Automatica, 1978, 14(5): 465–471
12. Xu L. Bayesian Ying-Yang system, best harmony learning, and five action circling. Frontiers of Electrical and Electronic Enginnering in China, 2010, 5(3): 281–328
13. Xu L, Krzyzak A, Oja E. Unsupervised and supervised classifications by rival penalized competitive learning. In: Proceedings of the 11th International Conference on Pattern Recognition. 1992, I: 672–675
14. Xu L, Krzyzak A, Oja E. Rival penelized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Transactions on Neural Networks, 1993, 4(4): 636–649
15. Xu L. Rival penalized competitive learning, finite mixture, and multisets clustering. In: Proceedings of IEEE Intentional Joint Conference on Neural Networks. 1998, 2: 2525–2530
16. Xu L. Bayesian-Kullback coupled Ying-Yang machines: unified learnings and new results on vector quantization. In: Proceedings of International Conference on Neural Information Processing. 1995, 977–988
17. Xu L. Bayesian Ying Yang learning. Scholarpedia, 2007, 2(3): 1809
18. Figueiredo M A F, Jain A K. Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(3): 381–396
19. Neal R, Hinton G E. A view of the EM algorithm that justifies incremental, sparse, and other variants. Learning in Graphical Models. Norwell: Kluwer Academic Publishers, 1998, 355–368

20. Corduneanu A, Bishop C M. Variational Bayesian model selection for mixture distributions. In: Proceedings of the 8th International Conference on Artificial Intelligence and Statistics. 2001, 27–34
21. Jaakkola T, Jordan M. Bayesian parameter estimation via variational methods. Statistics and Computing, 2000, 10(1): 25–37
22. Wallace C, Boulton D. An information measure for classification. The Computer Journal, 1968, 11(2): 185–194
23. Wallace C S, Dowe D L. Minimum message length and Kolmogorov complexity. The Computer Journal, 1999, 42(4): 270–283
24. Attias H. A variational Bayesian framework for graphical models. Advances in Neural Information Processing Systems, 2000, 12: 209–215
25. Gelman A, Carlin J B, Stern H S, Rubin D B. Bayesian Data Analysis. 2nd ed. Texts in Statistical Science. Boca Raton: Chapman & Hall/CRC, 2003
26. Xu L. Machine learning problems from optimization perspective. Journal of Global Optimization, 2010, 47(3): 369–401
27. Unnikrishnan R, Pantofaru C, Hebert M. Toward objective evaluation of image segmentation algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6): 929–944
28. Xu L. Bayesian Ying Yang system, best harmony learning and Gaussian manifold based family. In: Zurada J, Yen G, Wang J, eds. Computational Intelligence: Research Frontiers. Berlin-Heidelberg: Springer-Verlag, 2008, 5050: 48–78,
29. Xu L. Multisets modeling learning: a unified theory for supervised and unsupervised learning. In: Proceedings of IEEE Intentional Joint Conference on Neural Networks. 1994, I: 315–320
30. Xu L. A unified learning framework: multisets modeling learning. In: Proceedings of World Congress on Neural Networks. 1995, 1: 35–42
31. Xu L. BYY harmony learning, structural RPCL, and topological self-organizing on unsupervised and supervised mixture models. Neural Networks, 2002, 15(8–9): 1125–1151
32. Xu L. Data smoothing regularization, multi-sets-learning, and problem solving strategies. Neural Networks, 2003, 16(5–6): 817–825
33. Xu L. A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving. Pattern Recognition, 2007, 40(8): 2129–2153
34. Xu L. Learning algorithms for RBF functions and subspace based functions. Handbook of Research on Machine Learning, Applications and Trends: Algorithms, Methods and Techniques. Hershey: IGI Global, 2009, 60–94
35. Xu L. Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, ME-RBF models and three-layer nets. International Journal of Neural Systems, 2001, 11(1): 3–69
36. Bartlett P L, Boucheron S, Lugosi G. Model selection and error estimation. Machine Learning, 2002, 48(1–3): 85–113
37. Kearns M, Mansour Y, Ng A Y, Ron D. An experimental and theoretical comparison of model selection methods. Machine Learning, 1997, 27(1): 7–50
38. Wallace C S, Dowe D L. Refinements of MDL and MML coding. The Computer Journal, 1999, 42(4): 330–337
39. Kotz S, Nadarajah S. Multivariate t Distributions and Their Applications. Cambridge: Cambridge University Press, 2004
40. Varma M, Zisserman A. Texture classification: are filter banks necessary? In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2003, 2: 691–698
41. Nikou C, Likas A, Galatsanos N. A Bayesian framework for image segmentation with spatially varying mixtures. IEEE Transactions on Image Processing, 2010, 19(9): 2278–2289
42. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888–905
43. Rother C, Kolmogorov V, Blake A. "GrabCut": interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics, 2004, 23(3): 309–314

Lei SHI obtained his B.Eng. degree in Computer Science and Technology from University of Science and Technology of China, in 2005. He is currently a Ph.D student of Department of Computer Science and Engineering in The Chinese University of Hong Kong. His research interests include statistical learning and neural computing.



Shikui TU is a Ph.D candidate of 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, chair professor of The Chinese University of Hong Kong (CUHK), Fellow of IEEE (2001–), Fellow of International Association for Pattern Recognition (2002–), and Academician of European Academy of Sciences (2002–). He completed his Ph.D thesis at Tsinghua University by the end of 1986, became postdoc at Peking University in 1987, then promoted to associate professor in 1988 and a professor in 1992. During 1989–1993 he was

research associate and postdoc in Finland, Canada and USA, including Harvard and MIT. He joined CUHK as senior lecturer in 1993, professor in 1996, and chair professor in 2002. He published several well-cited papers on neural networks, statistical learning, and pattern recognition, e.g., his papers got over 3400 citations (SCI) and over 6300 citations by Google Scholar (GS), with the top-10 papers scored over 2100 (SCI) and 4100 (GS). One paper scored 790 (SCI) and 1351 (GS). He served as a past governor of International Neural Network Society (INNS), a past president of APNNA, and a member of Fellow Committee of IEEE CI Society. He received several national and international academic awards (e.g., 1993 National Nature Science Award, 1995 INNS Leadership Award and 2006 APNNA Outstanding Achievement Award).