NAIST-IS-DD0461031

Doctoral Dissertation

Probabilistic approach to unsupervised representation learning in dynamic environments

Jun-ichiro Hirayama

February 20, 2007

Department of Bioinformatics and Genomics
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Jun-ichiro Hirayama

Thesis Committee:

Professor Shin Ishii	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Associate Professor Tomohiro Shibata	(Co-supervisor)

Probabilistic approach to unsupervised representation learning in dynamic environments[*]

Jun-ichiro Hirayama

Abstract

Learning useful representation of multivariate data is one of the principal themes of unsupervised learning. Basic techniques includes the principal component analysis (PCA) and the independent component analysis (ICA), and there have also been proposed more advanced ones often extending these basic techniques. In this thesis, I investigate particular extensions of such “representation learning” techniques to be capable of difficult non-stationary contexts, particularly focusing on specific problems in which some difficulties arises for previous methods due to dynamic natures of environment.

In the former part of this thesis, I propose a new extension of probabilistic/noisy independent component analysis (ICA), so as to address a difficult non-stationarity that may occur in real-world contexts of blind source separation (BSS) problem, such that each source signal abruptly appears or disappears with time. I employ a special type of hidden Markov and hidden semi-Markov models to realize a dynamic switching of source variables in the framework of probabilistic ICA. I derive an effective approximation of Bayesian inference for this model based on the variational Bayes (VB) method. In simulation experiments using artificial and realistic source signals, I demonstrate the superior performance of the proposed method to existing ones, especially in uncertain situations.

[*] Doctoral Dissertation, Department of Bioinformatics and Genomics, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0461031, February 20, 2007.

In the latter part, I investigate an online feature extraction problem, particularly focusing on such situations that the environment is not stationary but dynamically changing, sometimes even abruptly. In this difficult non-stationary context, an appropriate control of adaptability/stability of a learning model is the key for rapid adaptation to the environmental changes. Focusing on feature extraction task by means of the probabilistic PCA (PPCA), I propose two online learning schemes that have such a character, based on the previously-proposed online variational Bayes (VB) method: One is based on an explicit formulation of probabilistic novelty detection by a mixture of PPCA model, the other is a principled approach using the hierarchical Bayes method based on a new interpretation of the online VB presented in this thesis. I demonstrate their availabilities in simulation experiments. In addition, I also discuss the biological implication of the proposed learning models, especially with hypothesizing their possible realization in brain.

While the methods proposed in this thesis have been developed as general statistical techniques to analyze and process stochastic data that inherently have dynamic natures, their high performances demonstrated in the simulation experiments indicate the future availabilities for specific real-world problems. Thus, I finally discuss the potential applications of the proposed methods with also discussing open issues remained for future developments of these methods.

Keywords:

Representation learning, dynamic environment, blind source separation, online feature extraction, independent component analysis, online Bayesian learning

Contents

Chapter 1

Introduction

1. What is representation learning?

In many data analysis or processing tasks emerged in the field of machine learning, observations often have such property that their complex, redundant or noisy natures prevent an efficient processing or a direct interpretation of meaningful signals therein. Learning systems should, then, transform the original data into another representation that have a simple and compact character and/or can reveal the hidden signals. This is sometimes a principal purpose of that task by itself and, if not so, is often a key requirement for such systems to make the task successful. If one assume that no additional information other than those data (inputs) is used to learn the representations, this becomes an issue of *unsupervised learning*. Unsupervised learning is a type of learning which in general aims at finding underlying structure of inputs, by distinguishing from noise, particularly without further receiving any target outputs (as in *supervised learning*) or rewards for the system's outputs from the environment (as in *reinforcement learning*). Although many tasks of unsupervised learning, such as clustering, visualization, data compression, density estimation and outlier/novelty detection, can be regarded as to learn another useful representation of original data in a general meaning, in this thesis I ordinary mean the term "representation" in a limited sense as the transformed quantitative values that varies on a continuous domain [1] — thus, for example, clustering is not the method to find data representation in this sense, since it transforms the data into discrete qualitative values, i.e, class labels — and I refer to such types of learning to obtain quantitative representation of data, particularly in the framework of unsupervised learning, as "representation learning." Those issues within supervised or reinforcement learning schemes are also interesting, but is not considered in this thesis.

Basic techniques of representation learning includes factor analysis (FA), principal component analysis (PCA) and independent component analysis (ICA) [52]. Both FA and PCA are standard multivariate statistical methods. ICA is rather a new technique but recently have been quickly popularized. These basic methods commonly seek to find a set of basis vectors in the input domain (sometimes implicitly), that is, to describe an input as a linear combination of basis vectors. Both the basis vectors and their coefficients for each input are determined from data; the coefficients are then the new representation of each input. These basic techniques (and also the other advanced ones) have recently been understood in a unifying framework of probabilistic generative modeling, especially with *latent variable models* [83, 13]. Observations often have such a character that they can be well described by some unobserved quantities, which can be regarded as certain instances of random variables, called latent variables. These variables sometimes correspond to actual physical values, but also can be defined in a virtual or an abstract sense. Representing such a data-generating process involving latent variables by means of probabilistic latent variable models, one can achieve learning and inference on unknown quantities in a consistent manner. FA was originally this type of methods, and PCA and ICA have also been formulated as latent variable models [92, 52]. Importantly, furthermore, this unifying framework have facilitated the recent advances in representation learning. There have recently been proposed extensive kinds of techniques in this area, including many variants and extensions of the basic techniques (e.g., [73, 57, 94, 51, 61, 23, 46, 16]).

[1] But sometimes I use this term in a more general sense to denote the total way to describe the original data according to the transformed quantities.

2. Motivation and overview

Previous methods of representation learning, such as PCA and ICA, are mainly static techniques ignoring any dynamic aspect of the inputs. However, in cases of time-series inputs in real-world problems, the statistical characteristics often change dynamically. Consider the learning systems such as humans, animals, or well-sophisticated intelligent robots that successfully operates in the real world. The statistical property of sensory inputs to these systems varies with time as reflecting the dynamic nature of environment, while they will be able to accommodate the changes to obtain appropriate internal representations of sensory inputs. It is not straightforward, however, to address such situations by standard, static techniques of representation learning. This principally motivate the studies in this thesis. In an engineering viewpoint, developing dynamic kinds of representation learning have basal importance to build a learning system that is adequate for operating in the real world. Furthermore, in a biological viewpoint, theoretical advances in such directions would facilitate the understanding of the function and the implementation of sensory representational systems in brain, as is often the way of theoretical/computational neuroscience [27, 81, 32]. As an initial step to build and understand a learning system that is adequate for operating in the real world, this thesis investigates dynamic kinds of representation learning methods.

In this thesis, I focus on two specific application domains, i.e., blind source separation and feature extraction, in each of which a particular difficulty arises from dynamic nature of the real-world environment. I propose two kinds of learning models respectively for the two problem domains, and demonstrate them within the rather specific context. It should be noted that, however, the scope of these learning models is is not necessarily limited to these particular problems. Below I briefly describe these problems and the specific difficulties arising in dynamic contexts with providing the overview of this thesis. In each of the two problem domains, the latent variables behind inputs will have meanings as source signals to be separated from their noisy mixtures, or feature vectors to be extracted from original inputs, respectively. The term *latent variable* is used as a general term, while it will be rephrased like *source* or *feature* depending on its context.

2.1 Blind source separation with non-stationary source appearances

Among the recent advances in representation learning, ICA and its related methods are of particular importance for their theoretical significance and broad applicability. While the scope of ICA is fairly general and thus it has been successfully applied to various engineering problems, the central motivation for developing ICA have historically been from a specific engineering problem in the field of signal processing. This is known as the blind source separation (BSS), which is a problem of recovering unknown source signals from their observed mixtures, where the detail of mixing process is unknown as the term 'blind' suggests. A popular illustration of BSS is the "cocktail-party problem." In a cocktail party, where many persons are simultaneously speaking, one can only observe the mixtures of original speech signals that is of most interests (with often involving other noises). The task here is to separate the original source signals only from the mixture signals recorded in multiple microphones, without knowing the actual mixing process. Assuming mutual independence among the source signals, standard ICA try to find a basis representation of the observed mixtures so that the coefficients become mutually independent and thus can be regarded as recovered source signals.

Source signals emerged in real-world problems often have such a difficult non-stationarity that each source signal abruptly appears or disappears, so that the sources being active at a moment dynamically changes with time. Most ICA methods, however, assume that a fixed set of source signals consistently exists throughout the time-series to be examined. The performance of source separation by the previous ICA methods thus will be degrade in such kinds of situations, especially when the situation is noisy and thus uncertain. To overcome this problem, in Chap. 2, I propose a new extension of ICA which automatically switches active subset of sources to accommodate the non-stationary appearances of source signals. A particular difficulty of this problem is due to the character that the problem structure changes with time in effect, in that the latent variables of interest are different for each context among all the potential ones. This requires a dynamic kind of model selection (or variable selection, more specifically) to be realized in a dynamic manner. The approach in this thesis is to incorporate a principled way of dynamic variable selection into a previous probabilistic formulation of (noisy) ICA.

2.2 Online feature extraction with accommodating environmental changes

In the fields of pattern recognition, a preliminary task to transform original data into another representation that is useful for subsequent recognition task is called the feature extraction [12], where the resultant quantities are called the features. Since the character of features strongly affect the recognition performance, finding appropriate features is an important issue for recognition systems. In practice, feature extraction is sometimes achieved by hand based on expert knowledges about the data-generating process and also about what the appropriate features are for the specific problem. However, one often do not have enough knowledges to achieve it, and it is not necessarily straightforward to obtain appropriate features from primitive form of data even if one has enough knowledges.

Representation learning can provide a systematic solution of feature extraction in unsupervised manners, which is important for such cases that one do not have enough knowledge or ability to extract appropriate features faithfully by hand. In particular, PCA has conventionally been applied for feature extraction (see, for example, [93]), especially as a technique of dimensionality reduction, i.e., to obtain a small number of basis vectors to describe the inputs, with retaining original information as much as possible. In addition, ICA also have recently been used for feature extraction, which try to find features that are mutually as independent as possible. Although the capability of these methods are sometimes limited due to the lack of specific knowledges depending on the problems, they have significant importance for their broad applicability.

When one consider about the real-world applications in dynamic contexts, an important issue of feature extraction is to achieve it in an online manner, that is, to process the inputs incrementally at each time point without retaining the past inputs. Such types of learning are called online learning. Online learning has a practical advantage especially when the environment has a non-stationary character, since it has a potential to address environmental changes if one reasonably set a meta-parameter that determines the speed of adaptation to new inputs. How to control the meta-parameter is, however, quite a difficult question. The changes occurred in realistic environments are sometimes not gradual but abrupt. Even if features have been appropriately extracted in a certain period that was regarded as almost stationary, an abrupt change at the next time point would quickly makes the features obsolete. For such difficult non-stationarities, it becomes crucial for real-world online learning systems to control the meta-parameter with appropriately reflecting the environmental changes. In the latter part of this thesis, focusing on a effective realization of online Bayesian learning, online variational Bayes (VB) [87], I propose two schemes to control the meta-parameter therein to address changing environments (Chapter. 3). One is based on an explicit formulation of probabilistic novelty detection by means of probabilistic mixture model, the other is a principled approach using hierarchical Bayes method, specifically with a new interpretation of online VB framework. I employ a probabilistic version of PCA as a specific instance of representation learning model, and validate it in simulation experiments of online feature extraction. It should be noted that, however, the issue of meta-parameter control in online VB is rather a general issue and is not necessarily limited to representation learning.

A high ability to learn sensory representation, i.e., features of sensory inputs, is a characteristic property of biological learning systems such as humans and animals. The original motivation of the study of online representation learning above is actually in understanding the principles of brain representation learning. While this thesis is described mainly from an engineering viewpoint, its connection to biological systems is also an interesting topic and can be a important contribution for the neuroscience field. Thus, I will also discuss the biological implication of the proposed dynamic learning scheme in Chapter. 3, especially with hypothesizing its potential implementation realized in brain.

Chapter 2

Switching ICA

1. Introduction

There have been proposed many types of extensions of standard ICA which originally assumes that the mixing process is linear, involving no noise, and sources have no temporal structures. These extensions were supposed, for example, to incorporate noisy [67, 22, 50, 7, 59, 61] and/or nonlinear [57, 91, 2, 61] situations, or to employ temporal information about the source signals. Such extensions have recently attracted attention for their potential capabilities of performing source separation effectively even when the standard ICA assumption does not hold.

To the best of our knowledge, however, there have been no studies of ICA focusing on such situations that the sources being active at a moment dynamically changes with time, which is one of the principal issues of this thesis. One exception is that Amari et al. [4] have reported the applicability of their method, natural-gradient-based ICA with a nonholonomic constraint, in a closely-related situation. The learning rule for this method was not affected by abrupt changes in the average magnitudes of source signals. This would allow source signals to have zero magnitude for a certain period, i.e., the sources being temporally inactive for the period. The performance of this method, however, can be poor in the presence of noise. One reason is that this method does not explicitly assume the existence of noise. Another reason is that the method ignores the signal property, i.e., temporal continuity of active/inactive periods, since the method was not designed for such situations. On the other hand, the present study is motivated by this particular case. In this chapter, we propse a new ICA method, *Switching ICA*, to address such temporal switching of active sources especially with the robustness against noise.

Several studies on noisy ICA previously have utilized temporal information to improve BSS for non-stationary source signals [75, 9, 78, 21, 95, 89], many of which employ the hidden Markov model (HMM) [79] to represent the temporal structure. These existing studies have tried to incorporate general non-stationarity that may exist in source signals, whereas this thesis focuses on addressing the special type of non-stationary situations, i.e., where the appearances of sources changes with time. This is not very general but quite important in dealing with various real-world signals. The new ICA method propsed in this thesis then employs a special type of HMM in order to incorporate such prior knowledge that the source may abruptly appear or disappear with time. This special setting of the HMM then provides an effect of *variable selection* in a dynamic way. This is the key difference of the new method from the previous HMM-ICA, which has a more general structure but no effect of variable selection. The new model is expected to improve the reconstruction of source signals when they are actually switched on and off temporally, and when there exists a certain amount of noise. It is also expected that the improved estimation of source signals will lead to more accurate estimation of the mixing matrix. Furthermore, we also investigate the use of another temporal structure, a hidden semi-Markov model (HSMM) [36, 84, 58, 54], as a potentially better model than the HMM to represent the duration of the presence/absence of source signals.

The proposed method is formulated as a noisy ICA that is based on generative models as in the previous studies [67, 22, 48, 7, 59, 97], with incorporating the dynamic variable selection mechanism by means of HMM/HSMM as mentioned above. Furthermore, as we employ Bayesian inference to estimate the new model, the resultant Switching ICA algorithm is a kind of Bayesian ICA, which has recently been investigated [66, 20, 19, 85, 21] for its several advantages over the conventional maximum likelihood (ML), such as the ability to avoid over-learning, a drawback of ML-based ICA [86]. The VB method [10] have also been used in the previous studies of noisy ICA [66, 20, 19] including HMM-ICA [21].

2. Model

2.1 Generative model

Let $x_{j,t}$ $(j = 1, \ldots, d)$ denote observations from d channels at time step t that are linear mixtures of unknown and mutually independent source signals from n channels, $s_{i,t}$ $(i = 1, \ldots, n)$, plus Gaussian noise. The probabilistic generative model of an observation vector $\boldsymbol{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{d,t})^T$ is then given by

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{\varepsilon}_t = \sum_{i=1}^{n} s_{i,t}\boldsymbol{a}_i + \boldsymbol{\varepsilon}_t, \tag{2.1}$$

where $\boldsymbol{s}_t = (s_{1,t}, s_{2,t}, \ldots, s_{n,t})^T$ is the source vector, a $d\times n$ matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$ is the mixing matrix, and $\boldsymbol{\varepsilon}_t$ is the noise vector, which is assumed in this study to be distributed as an isotropic Gaussian with a mean of zero and a variance of β^{-1}. The superscript ^{T} denotes the transpose. The d-dimensional vector $\boldsymbol{a}_i$ is referred to as an independent component-loading vector or a basis vector of observations.

Consider that each source is in either an *active* state or an *inactive* state, indicating whether the source is present or absent at each time step, respectively. The source signal $s_{i,t}$ is then represented as

$$s_{i,t} = z_{i,t}o_{i,t}, \tag{2.2}$$

where $z_{i,t}$ is an indicator variable, called a *switching variable*, which takes zero or one, and $o_{i,t}$ is a random variable representing the original signal of source i. If $z_{i,t} = 0$, then the source $s_{i,t}$ is inactive and consistently zero. In contrast, if $z_{i,t} = 1$, implying $s_{i,t} = o_{i,t}$, then the source is active and provides the original signal $o_{i,t}$.

Let $\boldsymbol{\zeta}_h$ $(h = 1, \ldots, 2^n)$ represent each of 2^n realizations of the indicator vector [1], $\boldsymbol{z}_t = (z_{1,t}, z_{2,t}, \ldots, z_{n,t})^T$. For a specific realization of $\boldsymbol{z}_t = \boldsymbol{\zeta}_h$, the generative model in Eq. (2.1) is written as

$$\boldsymbol{x}_t = \boldsymbol{A}^h\boldsymbol{s}_t^h + \boldsymbol{\varepsilon}_t = \sum_{j=1}^{n^h} s_{j,t}^h\boldsymbol{a}_j^h + \boldsymbol{\varepsilon}_t, \tag{2.3}$$

[1] The manner of indexing each realization by h can be chosen arbitrarily.

where $\boldsymbol{s}_t^h = (s_{1,t}^h, s_{2,t}^h, \ldots, s_{n^h,t}^h)^T$ is a collective vector of n^h active sources, $s_{j,t}^h$ is the j-th active source for $\boldsymbol{z}_t = \boldsymbol{\zeta}_h$, and $\boldsymbol{A}^h = (\boldsymbol{a}_1^h, \boldsymbol{a}_2^h, \ldots, \boldsymbol{a}_{n^h}^h)$ is a $d \times n^h$ matrix, the j-th column of which is the column vector in $\boldsymbol{A}$ corresponding to the j-th active source $s_{j,t}^h$. In this chapter, the observations are assumed to be preliminarily normalized as $\bar{\boldsymbol{x}}_t = \mathbf{0}$ without loss of generality. There are now 2^n generative models, each of which has a different subset (and different number) of sources as independent components. This is the basis of Switching ICA.

2.2 Modeling original signal of sources

The original signal of the i-th source, $o_{i,t}$, as an MoG, is modeld as in previous studies [67, 7, 97, 20, 66, 19], which is mainly for the existence of an analytical solution for the posterior distribution. Although an MoG can represent various kinds of distributions, including super-Gaussian and sub-Gaussian distributions, in this study we employ the simplest version, a scale mixture [6] of two Gaussian distributions, to well represent super-Gaussian signals. The parameters of scale-MoG are determined simultaneously with the estimation of the mixing matrix and the noise variance as in [67, 7]. The two-components scale-MoG density function of $o_{i,t}$ is then given as

$$p(o_{i,t}) = \alpha_i \mathrm{N}(o_{i,t} \mid 0, \gamma_{1i}^{-1}) + (1 - \alpha_i)\mathrm{N}(o_{i,t} \mid 0, \gamma_{0i}^{-1}), \tag{2.4}$$

where α_i is the mixing rate, and γ_{i0} and γ_{i1} are inverse variances of the two Gaussians. The means are assumed to be zero. Although the use of this form is for the sake of simplicity, an extension that employs a general form of MoG (as in [7]) can be performed when such source distributions should be considered.

The distribution of source signals, conditional on the switching variable, is given as follows. When $z_{i,t} = 1$, the source $s_{i,t}$ is equal to $o_{i,t}$, suggesting the original signal distribution given in Eq. (2.4):

$$p(s_{i,t} \mid z_{i,t} = 1) = \alpha_i \mathrm{N}(s_{i,t} \mid 0, \gamma_{1i}^{-1}) + (1 - \alpha_i)\mathrm{N}(s_{i,t} \mid 0, \gamma_{0i}^{-1}). \tag{2.5}$$

For all sources $i = 1, 2, \ldots, n$, we collectively write $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ and $\boldsymbol{\gamma} = \{\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1\}$, where $\boldsymbol{\gamma}_0 = (\gamma_{01}, \gamma_{02}, \ldots, \gamma_{0n})^T$ and $\boldsymbol{\gamma}_1 = (\gamma_{11}, \gamma_{12}, \ldots, \gamma_{1n})^T$. Eq. (2.5) is also represented in a hierarchical form by introducing a latent variable $y_{i,t} \in \{0, 1\}$

that indicates the two Gaussian components:

$$p(y_{i,t}) = \alpha_i^{y_{i,t}}(1-\alpha_i)^{1-y_{i,t}}, \tag{2.6a}$$

$$p(s_{i,t} \mid y_{i,t}, z_{i,t}=1) = \mathrm{N}(s_{i,t} \mid 0, \gamma_{i,t}^{-1}), \tag{2.6b}$$

where $\gamma_{i,t} = \gamma_{0i}$ (if $y_{i,t}=0$) or γ_{1i} (if $y_{i,t}=1$). On the other hand, the source $s_{i,t}$ is consistently zero when $z_{i,t}=0$. The conditional distribution is thus represented as Dirac's Delta distribution:

$$p(s_{i,t} \mid z_{i,t}=0) = \delta(s_{i,t}), \tag{2.7}$$

where $\delta(\cdot)$ denotes Dirac's Delta function. Because of the independence assumption, the joint distribution of all of the n sources is given as a product of scale-MoGs (Eq. (2.5)) for active sources and Dirac's Delta distributions (Eq. (2.7)) for inactive sources, for a given indicator allocation. Let $\bar{\boldsymbol{s}}_t^h = (\bar{s}_{1,t}^h, \bar{s}_{2,t}^h, \ldots, \bar{s}_{\bar{n}^h,t}^h)^T$ represent the collective vector of $\bar{n}^h$ inactive sources, where $\bar{s}_{k,t}^h$ is the k-th inactive source for $\boldsymbol{z}_t = \boldsymbol{\zeta}_h$. We also define $\boldsymbol{y}_t = (y_{1,t}, y_{2,t} \ldots, y_{n,t})^T$ and $\boldsymbol{y}_t^h = (y_{1,t}^h, \ldots, y_{n^h,t}^h)^T$ corresponding to $\boldsymbol{s}_t$ and $\boldsymbol{s}_t^h$, respectively. Furthermore, $\boldsymbol{\gamma}_0^h$, $\boldsymbol{\gamma}_1^h$ and $\boldsymbol{\alpha}^h$ are defined to represent the MoG parameters of active sources. The joint distribution of $\boldsymbol{s}_t$ conditional on $\boldsymbol{z}_t = \boldsymbol{\zeta}_h$ is then given by

$$p(\boldsymbol{s}_t \mid \boldsymbol{z}_t = \boldsymbol{\zeta}_h) = \sum_{\boldsymbol{y}_t^h} \alpha_{\boldsymbol{y}_t^h} \mathrm{N}_{n^h}(\boldsymbol{s}_t^h \mid \boldsymbol{0}, \boldsymbol{V}_{\boldsymbol{y}_t^h}) \prod_{k=1}^{\bar{n}^h} \delta(\bar{s}_{k,t}^h), \tag{2.8}$$

where

$$\alpha_{\boldsymbol{y}_t^h} = \prod_{j=1}^{n^h} (\alpha_j^h)^{y_{j,t}^h}(1-\alpha_j^h)^{1-y_{j,t}^h}, \tag{2.9a}$$

$$\boldsymbol{V}_{\boldsymbol{y}_t^h}^{-1} = \mathrm{diag}\left(\gamma_{1,t}^h, \gamma_{2,t}^h, \ldots, \gamma_{n^h,t}^h\right). \tag{2.9b}$$

Here, diag$(\cdot)$ denotes the diagonal matrix having specified values as its diagonal elements, and $\gamma_{j,t}^h = \gamma_{0j}^h$ (if $y_{j,t}^h = 0$) or γ_{1j}^h (if $y_{j,t}^h = 1$).

2.3 Markov assumption on switching dynamics

For real-world data, it is natural to assume a temporal continuity on each of the active and inactive states. One practical way to incorporate such temporal dependence into the model is to employ a Markov process to model the dynamics of the switching variables. In particular, in this study, strictly following the assumption of mutual independence among sources, the Markov processes for n sources are assumed to be independent of each other. The initial and transition probabilities of the switching variables are then given by $p(\boldsymbol{z}_1) = \prod_{i=1}^{n} p(z_{i,1})$ and $p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}) = \prod_{i=1}^{n} p(z_{i,t} \mid z_{i,t-1})$, respectively, where we assume

$$p(z_{i,1}) = \begin{cases} 1-\pi_i & \text{if } z_{i,1}=0 \\ \pi_i & \text{if } z_{i,1}=1 \end{cases}, \tag{2.10a}$$

$$p(z_{i,t} \mid z_{i,t-1}) = \begin{cases} \rho_{0i} & \text{if } (z_{i,t}, z_{i,t-1}) = (0,0) \\ 1-\rho_{0i} & \text{if } (z_{i,t}, z_{i,t-1}) = (1,0) \\ 1-\rho_{1i} & \text{if } (z_{i,t}, z_{i,t-1}) = (0,1) \\ \rho_{1i} & \text{if } (z_{i,t}, z_{i,t-1}) = (1,1) \end{cases}. \tag{2.10b}$$

Here, π_i is the probability for the initial presence of the i-th source, and ρ_{0i}, $\rho_{1i} \in [0,1]$ are the probabilities that the i-th source switches from active to inactive and that the i-th source switches from inactive to active, respectively. Figure 2.1 shows the transition diagram of this setting.

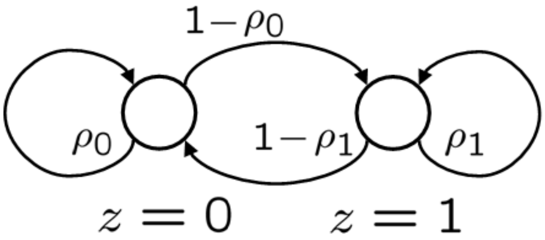

Figure 2.1. Transition diagram of active ($z=1$) and inactive ($z=0$) states.

The temporal structure assumed on the switching variables is the key aspect of the Switching ICA model. By assuming the Markov dynamics on switching variables, estimation for the state of the corresponding source signal, active or inactive, is expected to be improved for such real-world signals that a source signal continues to exist for a certain time after appearing and continues not to exist for a certain time after disappearing. The parameters $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)^T$

and $\boldsymbol{\rho} = \{\boldsymbol{\rho}_0, \boldsymbol{\rho}_1\}$, where $\boldsymbol{\rho}_1 = (\rho_{11}, \rho_{12}, \ldots, \rho_{1n})^T$ and $\boldsymbol{\rho}_0 = (\rho_{01}, \rho_{02}, \ldots, \rho_{0n})^T$, are unknown and must be estimated. The self-transition probabilities, ρ_{0i} and ρ_{1i}, are automatically estimated within the range of $[0.5, 1]$, under the assumption that $z_{i,t}$ tends to remain at the same value, and the source signal then remains active or inactive for a certain time. It should be noted that the Markov process in Eq. (2.10) includes a special case of the lack of dynamics. That is, if $\rho_{0i} + \rho_{1i} = 1$, then $z_{i,t}$ are independently distributed as $p(z_{i,t} = 0) = \rho_{0i}$ and $p(z_{i,t} = 1) = \rho_{1i} = 1 - \rho_{0i}$ for $t = 1, 2, \ldots, \tau$; in contrast, if $(\pi_i, \rho_{0i}) = (0, 1)$ or $(\pi_i, \rho_{1i}) = (1, 1)$, then $z_{i,t}$ is constant at zero or one, respectively, for $t = 1, 2, \ldots, \tau$, which corresponds to the conventional assumption of stationary ICA.

To summarize the above description of our probabilistic generative model, Fig. 2.2 illustrates a graphical model of the Switching ICA.

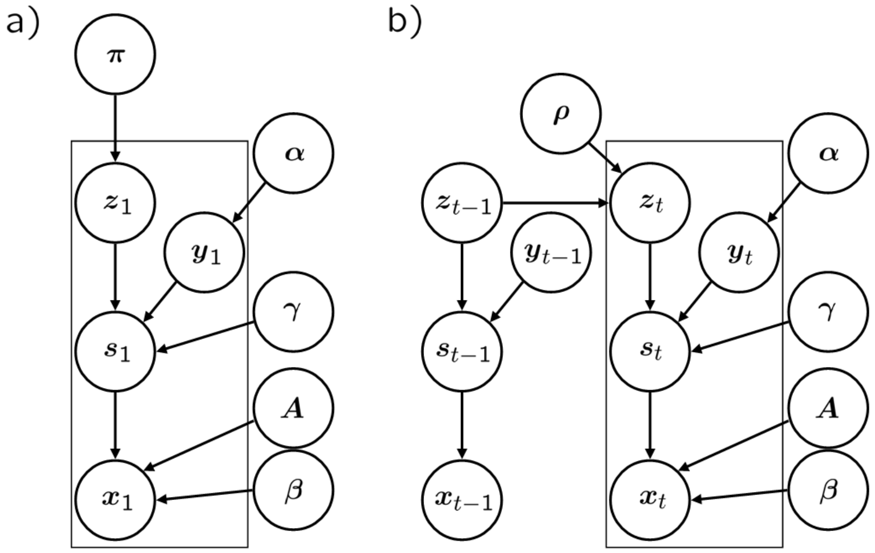

Figure 2.2. Graphical models. a) At the initial time step $t = 1$. b) Transition from time step $t - 1$ to t. The solid square represents one time slice.

3. Bayesian inference

In this section, we describe the Bayesian inference for the Switching ICA model defined in the previous section, the algorithm of which follows the variational Bayes (VB) method [8, 10].

3.1 Prior distribution

Let $\boldsymbol{\theta} = \{\boldsymbol{A}, \beta\}$, $\boldsymbol{\phi} = \{\boldsymbol{\alpha}, \boldsymbol{\gamma}\}$ and $\boldsymbol{\omega} = \{\boldsymbol{\pi}, \boldsymbol{\rho}\}$ represent the model parameters involved in the observation process, scale-MoG source models and dynamical processes, respectively. We assume that their prior distributions are given as the following conjugate forms, $p_0(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) = p_0(\boldsymbol{\theta})p_0(\boldsymbol{\phi})p_0(\boldsymbol{\omega})$:

$$p_0(\boldsymbol{\theta}) = p_0(\boldsymbol{A}, \beta) = \mathrm{N}_{d\times n}\left(\boldsymbol{A} \mid \boldsymbol{M}_0, \beta^{-1}\boldsymbol{I}_d, \boldsymbol{G}_0^{-1}\right)\mathrm{Ga}(\beta \mid \kappa_0, \lambda_0), \tag{2.11a}$$

$$p_0(\boldsymbol{\phi}) = p_0(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = \prod_{i=1}^{n} \mathrm{Be}(\alpha_i \mid u_{\alpha_i}, w_{\alpha_i})\mathrm{Ga}(\gamma_{0i} \mid u_{\gamma_{0i}}, w_{\gamma_{0i}})\mathrm{Ga}(\gamma_{1i} \mid u_{\gamma_{1i}}, w_{\gamma_{1i}}), \tag{2.11b}$$

$$p_0(\boldsymbol{\omega}) = p_0(\boldsymbol{\pi}, \boldsymbol{\rho}) = \prod_{i=1}^{n} \mathrm{Be}(\pi_i \mid u_{\pi_i}, w_{\pi_i})\mathrm{Be}(\rho_{0i} \mid u_{\rho_{0i}}, w_{\rho_{0i}})\mathrm{Be}(\rho_{1i} \mid u_{\rho_{1i}}, w_{\rho_{1i}}), \tag{2.11c}$$

where $\mathrm{N}_{d\times n}(\cdot \mid \cdot, \cdot, \cdot)$, $\mathrm{Ga}(\cdot \mid \cdot, \cdot)$ and $\mathrm{Be}(\cdot \mid \cdot, \cdot)$ denote the matrix normal distribution, the Gamma Distribution, and the Beta distribution, respectively (for definitions, see Appendix A).

3.2 Variational Bayes method

The VB method reformulates the Bayesian inference as a functional optimization problem. The objective function, called the variational free energy, is defined as a functional of a probability distribution q referred to as the trial distribution:

$$\mathcal{F}[q(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})] = \left\langle \log \frac{p(\boldsymbol{X}, \boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})}{q(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})} \right\rangle_{\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}}, \tag{2.12}$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$ denotes the observation matrix, and $\boldsymbol{\xi} = \{\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{Z}\}$ is the set of latent variables, with $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_\tau)$, $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\tau)$ and

$\boldsymbol{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_\tau)$. Here, $\langle \cdot \rangle_\varphi$ denotes the expectation with respect to the trial distribution, $q(\varphi)$ [2]. Equation (2.12) can be written as

$$\mathcal{F}[q(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})] = \log p(\boldsymbol{X}) - \mathrm{KL}\left[q(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \parallel p(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega} \mid \boldsymbol{X})\right], \tag{2.13}$$

where $\mathrm{KL}[\cdot \| \cdot]$ is the Kullbuck-Leibler (KL) divergence. Eq. (2.13) indicates that the exact maximization of $\mathcal{F}$ with respect to q is equivalent to the minimization of the KL divergence between q and the true posterior p, since the first term in Eq. (2.13) does not depend on q.

The solution of the exact maximization of Eq. (2.12), i.e., $q = p$, however, involves intractable integration (or summation) in many cases. To avoid this, the VB method introduces a constraint on q, and then finds the q that best approximates p. The constraint is usually given as a factorization of q that assumes partial independence of unknown variables. In this study, we use the constraint as

$$q(\boldsymbol{\xi}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \approx q(\boldsymbol{\xi}) q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}), \tag{2.14}$$

which allows us to obtain the closed-form solutions of $q(\boldsymbol{\xi})$ and $q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})$. The maximization of Eq. (2.12) is then approximately performed by alternative maximization with respect to each of the two functions. According to [10], the two solutions are given as

$$q(\boldsymbol{\xi}) \propto \exp\left(\langle \log p(\boldsymbol{X}, \boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \rangle_{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}}\right), \tag{2.15a}$$

$$q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \propto \exp\left(\langle \log p(\boldsymbol{X}, \boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \rangle_{\boldsymbol{\xi}}\right) p_0(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}), \tag{2.15b}$$

where the normalization terms are omitted. Equations (2.15a) and (2.15b) can also be written as follows, by employing the conditional independence among variables assumed in the generative model. Equation (2.15a) can be represented as

$$q(\boldsymbol{\xi}) = q(\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{Z}) = \frac{1}{C_{\boldsymbol{\xi}}} \psi_\theta\left(\boldsymbol{X}, \boldsymbol{S}\right) \psi_\phi\left(\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{Z}\right) \psi_\omega\left(\boldsymbol{Z}\right), \tag{2.16}$$

where $\psi_\theta\left(\boldsymbol{X}, \boldsymbol{S}\right)$, $\psi_\phi\left(\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{Z}\right)$ and $\psi_\omega\left(\boldsymbol{Z}\right)$ are defined as $\exp\left(\langle \log p(\boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{\theta}) \rangle_{\boldsymbol{\theta}}\right)$, $\exp(\langle \log p(\boldsymbol{S}, \boldsymbol{Y} \mid \boldsymbol{Z}, \boldsymbol{\phi}) \rangle_{\boldsymbol{\phi}})$ and $\exp\left(\langle \log p(\boldsymbol{Z} \mid \boldsymbol{\omega}) \rangle_{\boldsymbol{\omega}}\right)$, respectively. $C_{\boldsymbol{\xi}}$ is the normalization term. Equation (2.15b) can be further factorized into the form $q(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) =$

[2] Hereinafter, this notation will sometimes be omitted when it is clear from the context.

$q(\boldsymbol{\theta})q(\boldsymbol{\phi})q(\boldsymbol{\omega})$ rigorously, where

$$q(\boldsymbol{\theta}) = \frac{1}{C_{\boldsymbol{\theta}}}\psi_S\left(\boldsymbol{X}, \boldsymbol{\theta}\right)p_0(\boldsymbol{\theta}), \tag{2.17a}$$

$$q(\boldsymbol{\phi}) = \frac{1}{C_{\boldsymbol{\phi}}}\psi_{S,Y,Z}\left(\boldsymbol{\phi}\right)p_0(\boldsymbol{\phi}), \tag{2.17b}$$

$$q(\boldsymbol{\omega}) = \frac{1}{C_{\boldsymbol{\omega}}}\psi_Z\left(\boldsymbol{\omega}\right)p_0(\boldsymbol{\omega}). \tag{2.17c}$$

Here, $\psi_S\left(\boldsymbol{X}, \boldsymbol{\theta}\right)$, $\psi_{S,Y,Z}\left(\boldsymbol{\phi}\right)$ and $\psi_Z\left(\boldsymbol{\omega}\right)$ are defined as $\exp\left(\langle \log p(\boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{\theta})\rangle_{\boldsymbol{S}}\right)$, $\exp(\langle \log p(\boldsymbol{S}, \boldsymbol{Y} \mid \boldsymbol{Z}, \boldsymbol{\phi})\rangle_{\boldsymbol{S},\boldsymbol{Y},\boldsymbol{Z}})$ and $\exp\left(\langle \log p(\boldsymbol{Z} \mid \boldsymbol{\omega})\rangle_{\boldsymbol{Z}}\right)$, respectively. $C_{\boldsymbol{\theta}}$, $C_{\boldsymbol{\phi}}$ and $C_{\boldsymbol{\omega}}$ are the normalization terms.

Equations (2.16) and (2.17) mutually interact through the calculation of expectations. The expectations with respect to the model parameters, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and $\boldsymbol{\omega}$, are easily calculated as conjugate forms of prior distributions in Eq. (2.11) (See Appendix B.2 for details). In contrast, the expectation with respect to the latent variables, $\boldsymbol{\xi}$, is calculated in a more complicated manner, which is described in the rest of this section. The alternate calculation of Eqs. (2.16) and (2.17) is called the VB-EM algorithm [10] mainly as an analogy from the EM algorithm [28, 70], where Eq. (2.16) is called the VB-E step and Eq. (2.17) the VB-M step. The VB-EM algorithm converges to a local maximum of the variational free energy, Eq. (2.12). In practice, we can avoid poor local maxima by using the standard (noisy) ICA method to set initially the trial distributions and/or by comparing multiple results from different initial conditions to select the best one with the largest value of the variational free energy.

3.3 Inference on sources under a specific switching assumption

Equation (2.16) shows that $\boldsymbol{s}_t$ and $\boldsymbol{y}_t$ at a specific time step t are conditionally independent from those of the other time steps, given a specific indicator vectors $\boldsymbol{z}_t$, since the temporal dependence is involved only in $\psi_\omega(\boldsymbol{Z})$, which is cancelled out in the calculation of $q(\boldsymbol{S}, \boldsymbol{Y} \mid \boldsymbol{Z}) = q(\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{Z}) / \sum_{\boldsymbol{Y}} \int d\boldsymbol{S} q(\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{Z})$. Because of this conditional independence, we can write $q(\boldsymbol{S}, \boldsymbol{Y} \mid \boldsymbol{Z}) = \prod_{t=1}^{T} q(\boldsymbol{s}_t, \boldsymbol{y}_t \mid \boldsymbol{z}_t) =$

$\prod_{t=1}^{T} q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t) q(\boldsymbol{y}_t \mid \boldsymbol{z}_t)$, where

$$q(\boldsymbol{s}_t, \boldsymbol{y}_t \mid \boldsymbol{z}_t) = \frac{\psi_\theta(\boldsymbol{x}_t, \boldsymbol{s}_t)\psi_\gamma(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)\psi_\alpha(\boldsymbol{y}_t, \boldsymbol{z}_t)}{\sum_{\boldsymbol{y}_t} \int d\boldsymbol{s}_t \, \psi_\theta(\boldsymbol{x}_t, \boldsymbol{s}_t)\psi_\gamma(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)\psi_\alpha(\boldsymbol{y}_t, \boldsymbol{z}_t)}, \tag{2.18}$$

and

$$q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t) = \frac{q(\boldsymbol{s}_t, \boldsymbol{y}_t \mid \boldsymbol{z}_t)}{\int d\boldsymbol{s}_t \, q(\boldsymbol{s}_t, \boldsymbol{y}_t \mid \boldsymbol{z}_t)} = \frac{\psi_\theta(\boldsymbol{x}_t, \boldsymbol{s}_t)\psi_\gamma(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)}{l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)}, \tag{2.19a}$$

$$q(\boldsymbol{y}_t \mid \boldsymbol{z}_t) = \int d\boldsymbol{s}_t \, q(\boldsymbol{s}_t, \boldsymbol{y}_t \mid \boldsymbol{z}_t) = \frac{l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)\psi_\alpha(\boldsymbol{y}_t)}{\sum_{\boldsymbol{y}_t} l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)\psi_\alpha(\boldsymbol{y}_t)}, \tag{2.19b}$$

with the notations: $\psi_\theta(\boldsymbol{x}_t, \boldsymbol{s}_t) = \exp(\langle \log p(\boldsymbol{x}_t \mid \boldsymbol{s}_t, \boldsymbol{\theta})\rangle_{\boldsymbol{\theta}})$, $\psi_\gamma(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t) = \exp(\langle \log p(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t, \boldsymbol{\gamma})\rangle_{\boldsymbol{\gamma}})$ and $\psi_\alpha(\boldsymbol{y}_t, \boldsymbol{z}_t) = \exp(\langle \log p(\boldsymbol{y}_t \mid \boldsymbol{z}_t, \boldsymbol{\alpha})\rangle_{\boldsymbol{\alpha}})$. The normalization term in Eq. (2.19a), $l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)$, is defined as

$$l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t) = \int d\boldsymbol{s}_t \, \psi_\theta(\boldsymbol{x}_t, \boldsymbol{s}_t)\psi_\gamma(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t), \tag{2.20}$$

which also acts as the marginal likelihood of $\boldsymbol{y}_t$ given $\boldsymbol{x}_t$ and $\boldsymbol{z}_t$ in Eq. (2.19b). The integral in Eq. (2.19b) can be analytically performed (see Appendix B.1). Further calculation of Eq. (2.19a) yields (see Appendix B.1 for derivation)

$$q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t = \boldsymbol{\zeta}_h) = q(\boldsymbol{s}_t \mid \boldsymbol{y}_t^h, \boldsymbol{z}_t = \boldsymbol{\zeta}_h) = \mathrm{N}_{n^h}(\boldsymbol{s}_t^h \mid \hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h}, \hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h}) \prod_{k=1}^{\bar{n}^h} \delta(\bar{s}_{k,t}^h), \tag{2.21}$$

where the hyperparameters are given by

$$\hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h} = \left(\langle \beta (\boldsymbol{A}^h)^T \boldsymbol{A}^h \rangle + \left\langle \boldsymbol{V}_{\boldsymbol{y}_t^h}^{-1} \right\rangle \right)^{-1}, \tag{2.22a}$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h} = \hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h} \left\langle \beta \boldsymbol{A}^h \right\rangle^T \boldsymbol{x}_t. \tag{2.22b}$$

Thus, the approximate posterior of source vector $\boldsymbol{s}_t$ conditional on $\boldsymbol{z}_t$ is given in the same form as Eq. (2.8):

$$q(\boldsymbol{s}_t \mid \boldsymbol{z}_t = \boldsymbol{\zeta}_h) = \sum_{\boldsymbol{y}_t^h} q(\boldsymbol{y}_t^h \mid \boldsymbol{z}_t = \boldsymbol{\zeta}_h) q(\boldsymbol{s}_t \mid \boldsymbol{y}_t^h, \boldsymbol{z}_t = \boldsymbol{\zeta}_h) \tag{2.23a}$$

$$= \sum_{\boldsymbol{y}_t^h} \hat{\alpha}_{\boldsymbol{y}_t^h} \mathrm{N}_{n^h} \left(\boldsymbol{s}_t^h \mid \hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h}, \hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h} \right) \prod_{i=1}^{\bar{n}^h} \delta(\bar{s}_{i,t}^h), \tag{2.23b}$$

where

$$\hat{\alpha}_{\boldsymbol{y}_t^h} = q(\boldsymbol{y}_t^h \mid \boldsymbol{z}_t = \boldsymbol{\zeta}_h) = \sum_{\bar{\boldsymbol{y}}_t^h} q(\boldsymbol{y}_t \mid \boldsymbol{z}_t = \boldsymbol{\zeta}_h). \tag{2.24}$$

Note here that all of the conditional moments appearing in Eq. (2.22), $\langle \boldsymbol{V}_{\boldsymbol{y}_t^h}^{-1} \rangle$, $\langle \beta(\boldsymbol{A}^h)^T \boldsymbol{A}^h \rangle$ and $\langle \beta \boldsymbol{A}^h \rangle$ for a given h, are obtained from the complete moments, $\langle \boldsymbol{V}_{\boldsymbol{y}_t}^{-1} \rangle$ ($\equiv \mathrm{diag}(\langle \gamma_{1,t} \rangle, \langle \gamma_{2,t} \rangle, \ldots, \langle \gamma_{n,t} \rangle)$), $\langle \beta \boldsymbol{A}^T \boldsymbol{A} \rangle$ and $\langle \beta \boldsymbol{A} \rangle$, for the case where all of the n components exist. That is, $\langle \boldsymbol{V}_{\boldsymbol{y}_t^h}^{-1} \rangle$ and $\langle \beta(\boldsymbol{A}^h)^T \boldsymbol{A}^h \rangle$ are obtained respectively from $\langle \boldsymbol{V}_{\boldsymbol{y}_t}^{-1} \rangle$ and $\langle \beta \boldsymbol{A}^T \boldsymbol{A} \rangle$ by eliminating the rows and columns that correspond to inactive components in the h-th indicator allocation. Similarly, $\langle \beta \boldsymbol{A}^h \rangle$ is also obtained by eliminating certain columns of $\langle \beta \boldsymbol{A} \rangle$.

Calculation of the approximate posterior for $\boldsymbol{\theta} = \{\boldsymbol{A}, \beta\}$, Eq. (2.17a), requires only the first and second posterior moments of $\boldsymbol{s}_t$, $\langle \boldsymbol{s}_t \rangle$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle$. To calculate the expectations with respect to $\boldsymbol{s}_t$, first the conditional expectation with respect to $\boldsymbol{s}_t$ given $\boldsymbol{z}_t$, $\langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ is evaluated, and then the expectation with respect to the approximate posterior, $q(\boldsymbol{z}_t)$, is taken, as presented below. Note that, given a specific $\boldsymbol{z}_t = \boldsymbol{\zeta}_h$ that involves $z_{i,t} = 0$, the corresponding entries of the conditional expectations (the i-th element of $\langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ and the i-th row and column of $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$) become zero. Thus, we need only the calculations of the non-zero entries thereof, which are given by the elements of the conditional expectations with respect to $\boldsymbol{s}_t^h$:

$$\langle \boldsymbol{s}_t^h \rangle = \sum_{\boldsymbol{y}_t^h} \hat{\alpha}_{\boldsymbol{y}_t^h} \langle \boldsymbol{s}_t^h \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t, \boldsymbol{z}_t = \boldsymbol{\zeta}_h} = \sum_{\boldsymbol{y}_t^h} \hat{\alpha}_{\boldsymbol{y}_t^h} \hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h}, \tag{2.25a}$$

$$\langle \boldsymbol{s}_t^h (\boldsymbol{s}_t^h)^T \rangle = \sum_{\boldsymbol{y}_t^h} \hat{\alpha}_{\boldsymbol{y}_t^h} \langle \boldsymbol{s}_t^h (\boldsymbol{s}_t^h)^T \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t, \boldsymbol{z}_t = \boldsymbol{\zeta}_h} = \sum_{\boldsymbol{y}_t^h} \hat{\alpha}_{\boldsymbol{y}_t^h} \left(\hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h} \hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h}^T + \hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h} \right). \tag{2.25b}$$

3.4 Forward-Backward inference on switching variables

To complete the calculation of expected sufficient statistics in the VB-E step, we need the marginal distributions, $q(\boldsymbol{z}_t)$ and $q(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$. Let $e(\boldsymbol{x}_t, \boldsymbol{z}_t)$ be the marginal likelihood of $\boldsymbol{z}_t$, which is given by the denominator in Eq. (2.19b). Then, the approximate posterior of $\boldsymbol{Z}$ is given by

$$q(\boldsymbol{Z}) = q(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T) = \frac{1}{C_{\boldsymbol{\xi}}} e(\boldsymbol{x}_1, \boldsymbol{z}_1) \psi_\pi(\boldsymbol{z}_1) \prod_{t=2}^{\mathcal{T}} e(\boldsymbol{x}_t, \boldsymbol{z}_t) \psi_\rho(\boldsymbol{z}_t, \boldsymbol{z}_{t-1}), \tag{2.26}$$

where $\psi_\pi(\boldsymbol{z}_1) \equiv \exp(\langle p(\boldsymbol{z}_1 \mid \boldsymbol{\pi})\rangle_{\boldsymbol{\pi}})$ and $\psi_\rho(\boldsymbol{z}_t, \boldsymbol{z}_{t-1}) \equiv \exp(\langle \log p(\boldsymbol{z}_t \mid \boldsymbol{z}_{t-1}, \boldsymbol{\rho})\rangle_{\boldsymbol{\rho}})$. Here, $C_{\boldsymbol{\xi}}$ is the normalization term that appeared in Eq. (2.16). Equation (2.26) is equivalent to the standard form of the HMM. Thus, the marginals of interest, $q(\boldsymbol{z}_t)$ for $t = 1, \ldots, \tau$ and $q(\boldsymbol{z}_{t+1}, \boldsymbol{z}_t)$ for $t = 1, \ldots, \tau - 1$, can be calculated exactly by using the Forward-Backward algorithm [3] [79]. The expectations in the terms $\psi_\pi(\boldsymbol{z}_1)$ and $\psi_\rho(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$ are calculated from $q(\boldsymbol{\pi}, \boldsymbol{\rho})$, whose derivations are described in Appendix B.2. Using the resultant marginal distribution, $q(\boldsymbol{z}_t)$, the expectations about the source vector, $\langle \boldsymbol{s}_t \rangle$ and $\left\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \right\rangle$, are obtained as explained above, that is, the weighted means of conditional moments are:

$$\left\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \right\rangle = \sum_{\boldsymbol{z}_t} q(\boldsymbol{z}_t) \langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}, \tag{2.27a}$$

$$\langle \boldsymbol{s}_t \rangle = \sum_{\boldsymbol{z}_t} q(\boldsymbol{z}_t) \langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}. \tag{2.27b}$$

The pairwise marginal, $q(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$, is used for calculating expectations that are necessary for the trial distribution of dynamics parameters, $q(\boldsymbol{\omega})$ (see Appendix B.2).

3.5 Final estimates of unknown variables

After the convergence of the VB-EM algorithm, the estimates of all unknown variables are obtained from the resultant approximate posterior, q. Although many types of estimators can be obtained from the posterior distribution, according to a variety of loss criteria, we conveniently take the expected values as the estimators for unknowns, because they are calculated in the VB-EM procedure. Only for the switching variables, the maximum *a posteriori* (MAP) estimator is obtained:

$$\hat{\boldsymbol{z}}_t = \mathrm{argmax}_{\boldsymbol{z}_t} q(\boldsymbol{z}_t). \tag{2.28}$$

Since the MAP estimate $\hat{\boldsymbol{z}}_t$ is a binary vector, as is the true switching vector, it is easier to interpret than the expected vector, which may take analog values.

[3] C_ξ is also obtained by the Forward-Backward algorithm, practically as the product of the scaling constant introduced to avoid any numerical underflow [79].

4. Simulations

In this section, we compare the performance of Switching ICA (SwICA) with those by three other methods. The first one is the natural-gradient ICA with a nonholonomic constraint [4] (NG-N). The other two methods are modified versions of SwICA. One is without temporal dynamics on the switching variables, referred to as NoDyna; the other is without the Dirac's Delta prior on the sources, which is almost equivalent to the existing (Bayesian) HMM-ICA [21] and thus referred to as HMM-ICA. In NoDyna, the switching variables, $z_{i,t}$ for $t = 1, 2, \ldots, \tau$, were considered as independent Bernoulli samples $p(z_{i,t} = 1) = 1 - p(z_{i,t} = 0) = \rho_{zi}$ instead of obeying the Markov process in Eq. (2.10). That is, the n parameters, ρ_{zi} for $i = 1, 2, \ldots, n$, were simply estimated using a conjugate Beta distribution in a similar manner to learning the dynamics parameters in the Markov process. In HMM-ICA, the Dirac's Delta conditional source model was replaced by the scale-MoG in Eq. (2.4), where the MoG parameters therein were also assumed to be unknown and were estimated.

To evaluate the performance, we use the average Source-to-Distortion Ratio (aSDR) and Amari's Performance Index (PI) [5], which measure the estimation performance of source signals and the mixing matrix, respectively. Let $s_i^\star$ and $\hat{s}_i$ represent the true and estimated source variables, not specifying the time index. Then, aSDR is defined as

$$\text{aSDR(dB)} = 10 \log_{10} \left(\frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{|\text{corr}[\hat{s}_i, s_i^\star]|} - 1 \right)^{-1} \right), \tag{2.29}$$

where corr$[\cdot, \cdot]$ denotes the sample correlation coefficient. In this calculation, the estimated sources were permuted so that the average of absolute correlations was the largest. This measure is a slight modification of the Source-to-Distortion Ratio proposed in [42]. When the absolute correlation for every i is high, aSDR becomes high. In contrast, when the absolute correlation for every i is low, aSDR becomes low. Next, let $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}$ be the true and estimated mixing matrices, respectively. Then, PI is defined as

$$\text{PI} = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{|u_{ij}|}{\max_k |u_{ik}|} - 1 \right) + \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \frac{|u_{ij}|}{\max_k |u_{kj}|} - 1 \right), \tag{2.30}$$

where $\boldsymbol{U} = [u_{ij}] = \hat{\boldsymbol{A}}^{\dagger}\boldsymbol{A}^{\star}$ and † denotes the pseudo-inverse. This measure becomes zero if the estimated mixing matrix $\hat{\boldsymbol{A}}$ correctly recovers the true mixing matrix, except for the scale and the order of columns, otherwise it has a positive value. In addition, we compute the signal-to-noise ratio (SNR) of the observations as

$$\mathrm{SNR(dB)} = 10\log_{10}\left(\frac{1}{d}\sum_{i=1}^{d}\frac{\mathrm{var}[\sum_{j=1}^{n} A_{ij}^{\star}s_j^{\star}]}{(\beta^{\star})^{-1}}\right), \tag{2.31}$$

where $\beta^{\star}$ is the true inverse variance of Gaussian noise, and var[·] denotes the sample variance.

4.1 Artificially-generated sources

We first examined the case of artificially generated source signals. The numbers of mixtures and potential source signals were set as $d = n = 3$, and the time-series length was $\tau = 1000$. The switching variables and source signals are shown in Fig. 2.3. The bar graph at the top of each panel shows the value of the true switching variables, $z_{i,t}^{\star}$ for $t = 1, 2, \ldots, \tau$, where the white bands indicate $z_{i,t}^{\star} = 1$ (active) and the black bands $z_{i,t}^{\star} = 0$ (inactive). The active source signals were generated from the scale-MoG in Eq. (2.5). The source signals were then artificially mixed with a mixing matrix, $\boldsymbol{A}^{\star}$, plus Gaussian noises with various noise levels to correspond to the SNRs of 0, 4, 8, 12, and 16. In the following, $\boldsymbol{A}^{\star}$ is given as

$$\boldsymbol{A}^{\star} = \begin{pmatrix} -0.8321 & 0.4851 & 0.0316 \\ 0.5547 & 0.4851 & -0.9482 \\ 0 & 0.7276 & 0.3161 \end{pmatrix}, \tag{2.32}$$

where the norm of each column vector is normalized to be 1 for simplicity. To conduct the VB inference for NoDyna, HMM-ICA and SwICA, we set the prior hyperparameters at $\boldsymbol{M}_0 = \boldsymbol{0}$, $\boldsymbol{G}_0 = 0.01\boldsymbol{I}_n$, $\kappa_0 = \lambda_0 = 1\times10^{-3}$, and $u_{\cdot} = w_{\cdot} = 0.5$, and iterated the VB-E and M steps 500 times. To initialize the trial distributions in HMM-ICA and SwICA, we ran a noisy ICA without the latent dynamics or the Dirac's Delta prior, which was constructed by eliminating temporal dynamics from HMM-ICA, and then assigned the resultant approximate posteriors as the initial distributions. Each of the four algorithms was executed over 50 runs for

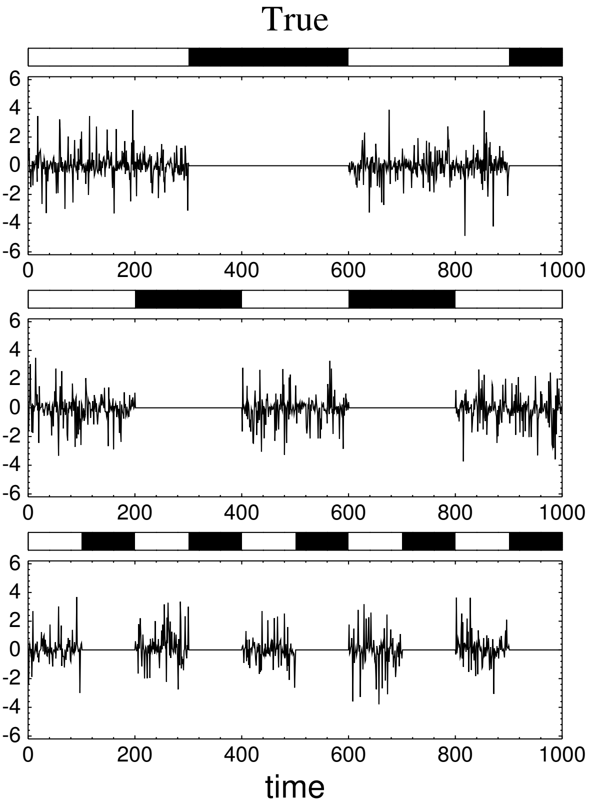

Figure 2.3. The true source signals and switching variables. The bar graph at the top of each panel indicates the value of the true switching variable $z^{\star}_{i,t}$, where the white bands represent $z^{\star}_{i,t} = 1$ (active) and the black bands represent $z^{\star}_{i,t} = 0$ (inactive).

each noise level, with application of different Gaussian noises for each run. The simulation times by SwICA for one run was about 160 min on average, with the Matlab program running on a Linux computer with a XEON 2.40-GHz CPU and a memory of 1 GB.

Figure 2.4 shows examples of recovered source signals by the four algorithms with SNR=8, where the compared result here is of the highest aSDRs for each algorithm, each out of the 50 runs. Figure 2.5 also shows the variational free energy for each algorithm except for NG-N. In Fig. 2.4, the bar graph illustrates the MAP estimates of the switching variables, where the white and black bands indicate active and inactive, respectively. These panels show that SwICA could effectively recover the original source signals that abruptly appeared or disappeared with time, compared to the other algorithms. The reconstruction of inactive periods by NG-N was heavily influenced by noise. This is also the case with HMM-ICA, though the switching variables (which indicate either of the two scale-MoG components in this case) were estimated successfully. NoDyna avoided such unexpected reconstruction in the inactive periods – in comparison to NG-N or HMM-ICA. The estimated signals in active periods, however, frequently have smaller values than the true ones, and there still exist some artifacts in the inactive periods. In contrast, Switching ICA successfully recovered the original source signals as well as avoided artifacts in the inactive periods. It should be noted that, however, the estimation of switching variables by SwICA still includes much failure, especially in the active periods. In Sec. 5, we will present a modification to improve it.

Figure 2.6 shows a comparison of the four algorithms' performance at different noise levels. The left panels show the aSDRs, and the right ones the logarithm of PIs. This figure illustrates that: 1) The performance of NG-N is lower than that of SwICA, even at the highest SNR (at 16) in this experiment; 2) the performance of HMM-ICA is comparable to that of SwICA at the higher SNRs (at 12 and 16), while it degenerates with increasing noise (SNR=4 and 8); 3) the performance of NoDyna is lower than SwICA at high SNRs (at 8, 12 and 16), but it becomes comparable with a lower SNR (at 4); and finally, 4) SwICA exhibits a consistently good performance both in source recovery and mixing matrix estimation, except for the lowest SNR (at 0), where the four algorithms show comparable low-level

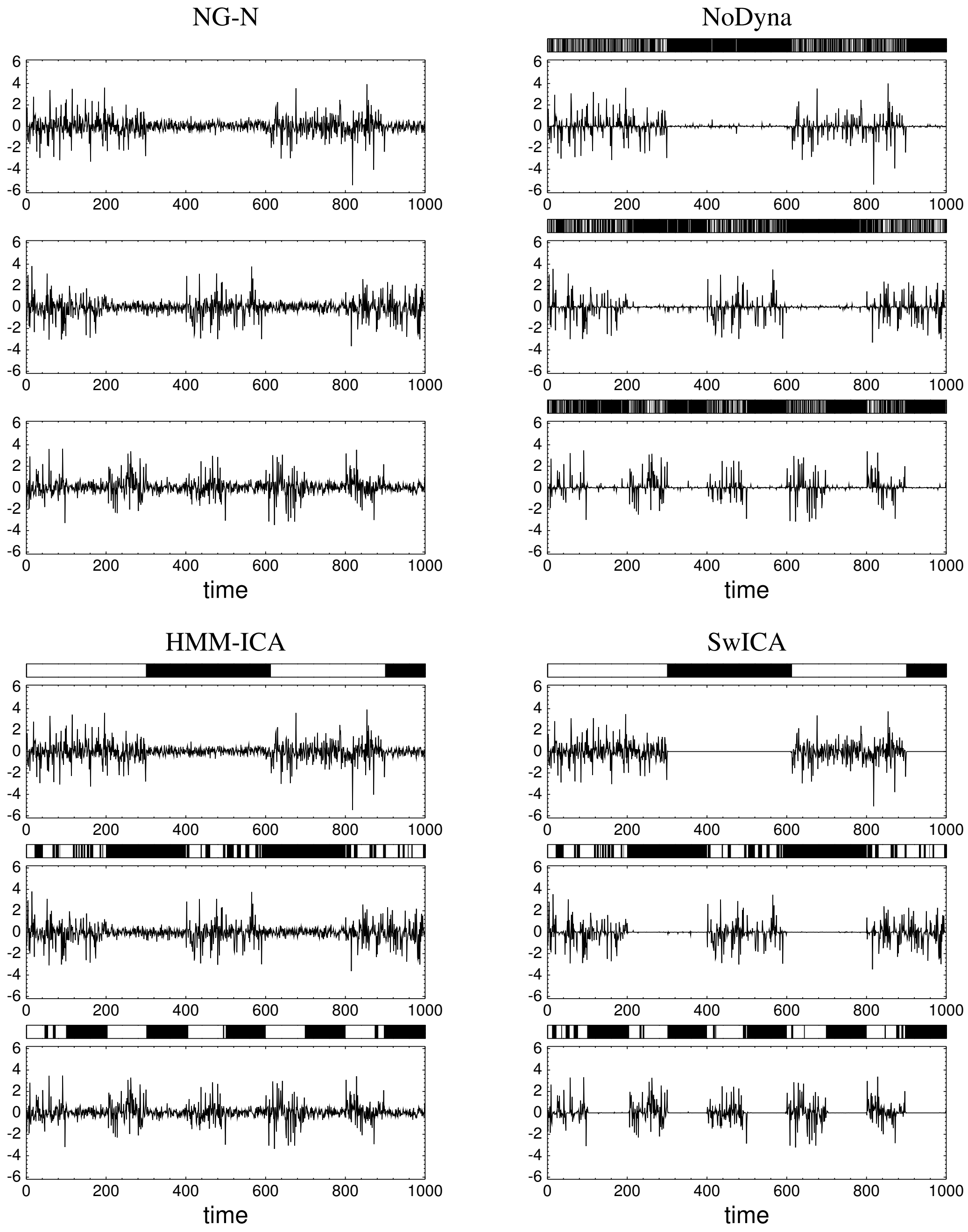

Figure 2.4. Examples of recovered source signals (and the switching variables except for NG-N) by the four algorithms, where each is of the highest aSDR among 50 runs. The bar graph denotes the MAP estimates of the switching variables, where the black and white bands represent zero (inactive) signals and one (active) signals, respectively. The order and the sign of each source were appropriately adjusted, and each source was scaled such that the corresponding column vector of the mixing matrix has a unit norm.

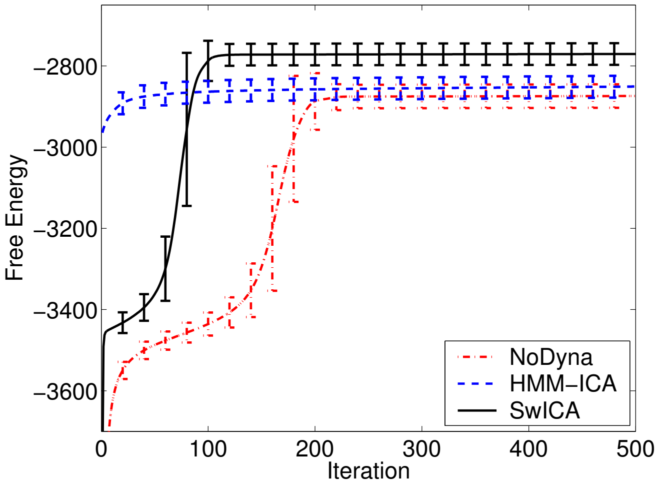

Figure 2.5. The variational free energy of the three algorithms, NoDyna, HMM-ICA and SwICA, along 500 learning steps. An error bar represents the standard deviation.

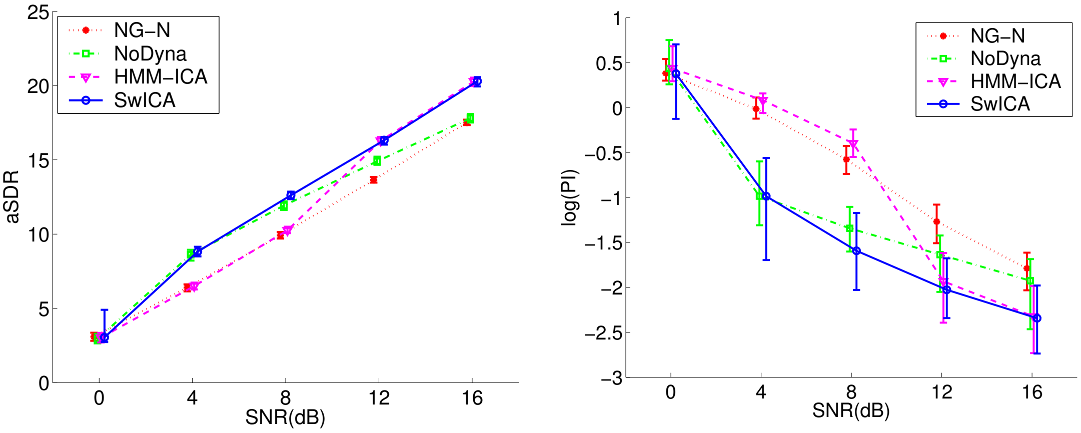

Figure 2.6. The performances at different noise levels for artificial source signals. The left panel shows the aSDR and the right one the logarithm of PI, by the four algorithms. An error bar represents the range between 10 percentile (lower) and the 90 percentile (upper).

performances.

A case with an unknown number of sources

In the experiment above, it was assumed that the total number of sources presented in the given time-series is known. The total number of sources, however, is not necessarily known in real-world problems. In our SwICA, such a situation would be handled by preparing a model with a larger n, where some redundant sources are expected to be estimated as inactive for all $t = 1, 2, \ldots, \tau$. To examine this nature, we prepared mixture signals by using only the first two source signals (and the first two column vectors in the mixing matrix of Eq. (2.32)) in the previous experiment, while the model assumed $n = 3$ (overestimated) instead of $n = 2$ (true). The two algorithms HMM-ICA and SwICA were performed along $1,000$ learning steps. The other settings were the same as in the previous one. Figure 2.7 shows the results of the highest aSDR. While HMM-ICA accidentally recovered artifact signals in the inactive periods, SwICA could successfully avoid them; actually, it completely switched off the third signals, $s_{3,t}$ for all $t = 1, 2, \ldots, \tau$, in this result. The mixing matrices estimated by the two algorithms were:

$$\hat{\boldsymbol{A}} = \begin{pmatrix} -0.8312 & 0.4605 & -0.4447 \\ 0.5559 & 0.4979 & -0.5682 \\ -0.0108 & 0.7349 & 0.6923 \end{pmatrix} \quad \text{(HMM-ICA)}, \tag{2.33a}$$

$$\hat{\boldsymbol{A}} = \begin{pmatrix} -0.8328 & 0.4969 & 0 \\ 0.5535 & 0.4888 & 0 \\ -0.0086 & 0.7171 & 0 \end{pmatrix} \quad \text{(SwICA)}, \tag{2.33b}$$

where the third column was effectively suppressed by SwICA. Figure 2.8 shows the histograms of the variances of recovered $s_{i,\cdot}$ over 20 runs. This result indicates that SwICA successfully and robustly suppresses irrelevant sources and hence recovers the original number of sources in this experiment.

An overcomplete case

We also investigated the case in which the number of active sources is larger than that of the mixture signals ($n > d$). In such an overcomplete case, the BSS

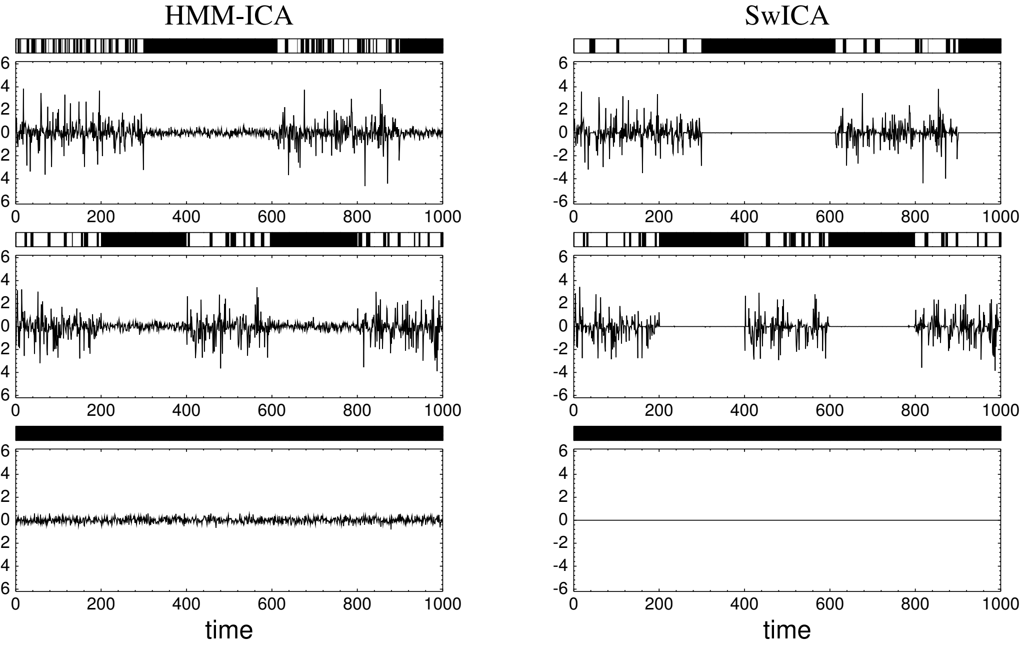

Figure 2.7. Examples of recovered source signals and the estimated switching variables by HMM-ICA and SwICA in the case that the number of true source signals is two, but the model over-estimates it *a priori* as three.

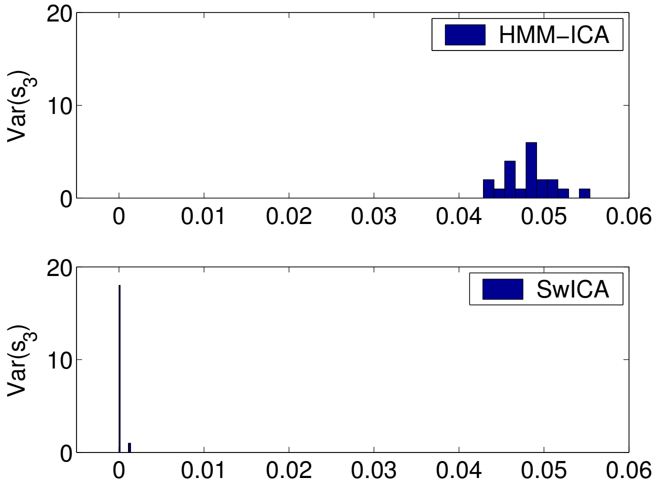

Figure 2.8. Histograms of the variances of the third source signals, $s_{3,\cdot}$, which were estimated by HMM-ICA (upper panel) and SwICA (lower panel). In this simulation, the number of true source signals was two, and thus the third source signal should be estimated as being zero or close to zero. In the lower panel, the bin width is set smaller than in the upper one, to make it easier to see that the variance by SwICA is likely the zero value almost exactly.

becomes quite difficult. We examined a simple setting here, in which the source signals ($n = 3$) in Fig. 2.3 was mixed into two observation signals ($d = 2$), with

$$\boldsymbol{A}^{\star} = \begin{pmatrix} -0.8321 & 0.7071 & 0.0333 \\ 0.5547 & 0.7071 & -0.9994 \end{pmatrix}. \tag{2.34}$$

We compared the two algorithms NoDyna and SwICA, each of which was repeated 20 times with the application of different Gaussian noises. The SNR was set at 16. Figure 2.9 shows a typical example of source and switching variables estimated by the two algorithms. Figure 2.10 shows the correlation coefficients between the true and estimated sources (as posterior expectations) and the accuracy of the MAP estimates of switching variables, by the two algorithms. These results indicate that the dynamics model employed in SwICA, rather than that in NoDyna is effective at providing prior knowledge to recover the original signals and their presence/absence even in this difficult overcomplete case.

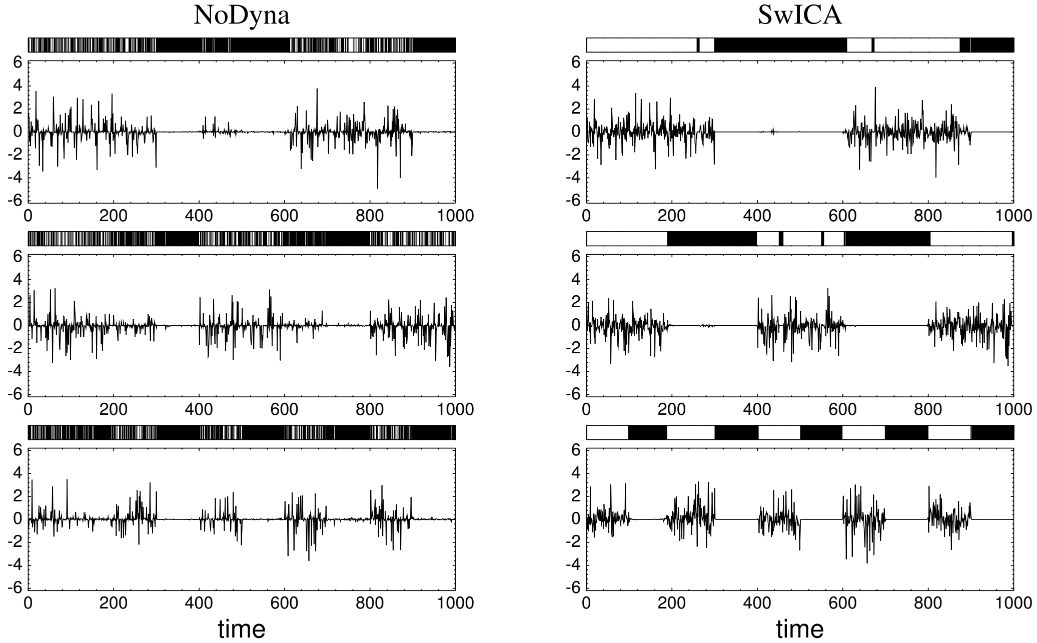

Figure 2.9. Examples of source signals and the switching variables estimated by NoDyna and SwICA in the overcomplete case.

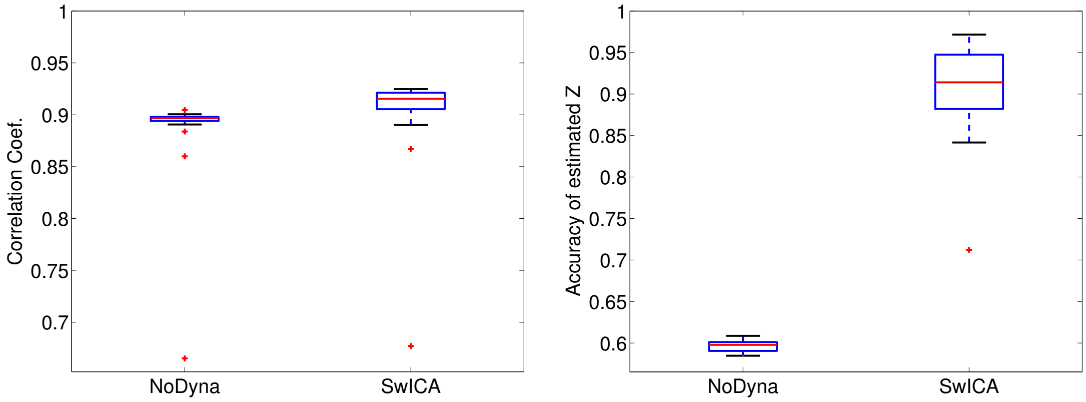

Figure 2.10. The overcomplete case. Left: The correlation coefficient between the true and the estimated source signals. Right: The accuracy of MAP estimates of the switching variables.

4.2 Realistic audio sources

Next, we evaluate the performance of Switching ICA for more realistic source signals, where the scale-MoG source model does not necessarily represent well the true source-generation process. Consider a situation in which two persons are in conversation with music playing in the background. Most of the time, the number of signals at any moment is one or two because one person is usually silent while the other person is speaking. However, three simultaneous source signals may sometimes occur like when the two speakers speak over each other. To simulate such a situation, we prepared the source signals as depicted in the left-hand panel of Fig. 2.11. The three waveforms denote two speech signals and one music signal (a singing voice). The original data were taken from the SpEAR database [96]. The sampling rate was reduced from 16 kHz to 0.5 kHz to reduce the sample size in order to produce a shorter simulation time. Although such an extremely low sampling rate may not be realistic, it is still sufficient for testing the fundamental performance of the proposed method. The total length of the down-sampled signals was $\tau = 1,000$. Some part of the original signals (the last 400 steps of the first signal and the first 400 steps of the second signal) were explicitly set at zero to simulate the above situation. They were also pre-processed such that each source signal in the active periods had a zero mean and unit variance.

The source signals were then artificially mixed into a three-channel observation time-series ($d = n = 3$) using the mixing matrix in Eq. (2.32) while applying Gaussian noises at various SNRs. The other experimental settings were the same as those in Sec. 4.1.

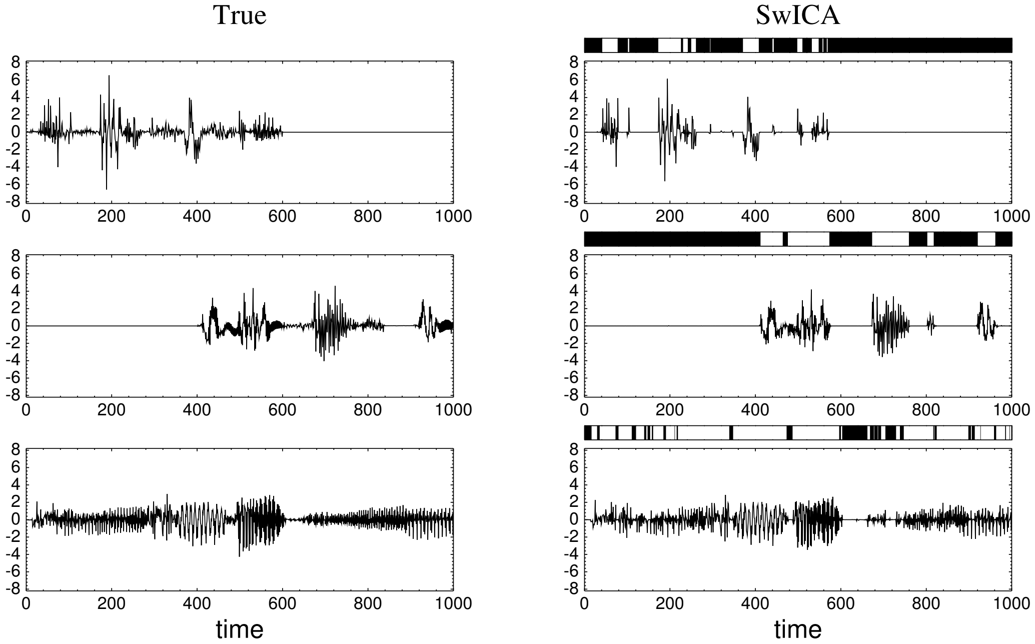

Figure 2.11. The case of realistic audio signals. Left: True source signals. Right: Estimated source signals and their presence/absence, estimated by SwICA.

Figure 2.12 compares the results obtained by the four algorithms over 50 runs. With low SNRs such as 0, 4 and 8, SwICA showed superior performance over the other three algorithms. NoDyna no longer showed comparable results, as seen in Fig. 2.6. With a high SNR (= 16), however, HMM-ICA performed better than SwICA, which is a major difference from Fig. 2.6. One possible reason for this difference is the discrepancy between the assumed and true models for active sources. With a high SNR, where the model discrepancy becomes large, the limited representation capacity of SwICA (which is a special model of HMM-ICA) may have led to such degenerate performance.

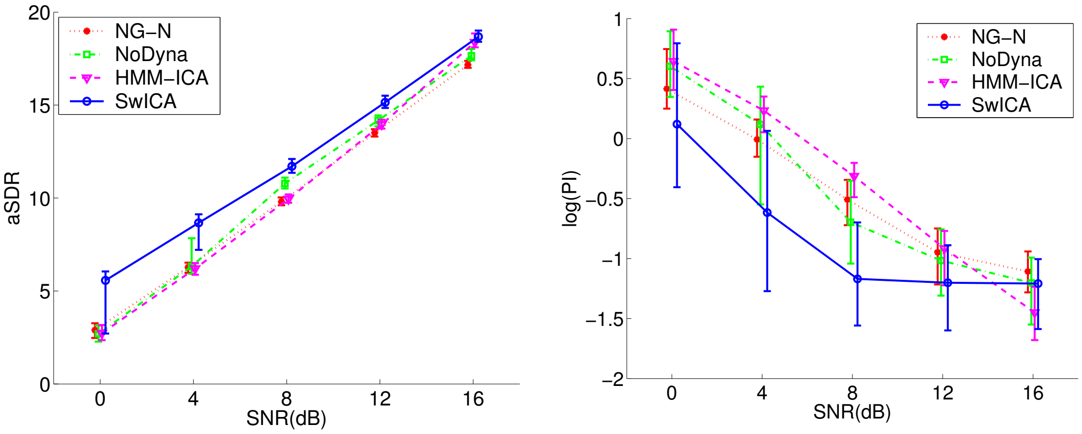

Figure 2.12. The performance at different noise levels in the case of realistic audio source signals. The left panel shows the aSDR and the right one the logarithm of PI, by the four algorithms. An error bar represents the range between the 10 percentile (lower) and the 90 percentile (upper).

5. Extension to semi-Markov switching

The Markov setting on the switching variables (Sec. 2.3) implicitly assumes that the duration of each active or inactive state is distributed as geometric [4], i.e., the probability of staying in a same state exponentially decreases with time. Although the Markov setting provides an effective inference procedure based on the Forward-Backward algorithm, the geometric assumption is not very consistent with many real-world phenomena, in which a source signal is likely to keep active or inactive for a certain time period once it has turned into one of them. The simulation results in Sec. 4.1 actually showed that the estimated states of each source tend to switch more frequently between active and inactive than the true sources do. Thus, we expect that the estimation of active/inactive states of true source signals would improve (with some increase in the computational cost) by employing a temporal structure that can accommodate another types of duration distribution representing such a situation more appropriately. In this section,

[4] The duration for a specific state in a Markov process is distributed as the geometric distribution: $p(d) = (1-\rho)\rho^{d-1}$, where d denotes the duration and ρ denotes the self-transition probability of the state.

we describe a simple extension of our Markov-based Switching ICA, and present simulation results.

A hidden semi-Markov model (HSMM) [36, 84, 58, 54] is effective in avoiding the geometric duration of HMMs. Johnson [54] reviewed several types of HSMM that had been proposed mainly in the field of speech recognition. One conventional way to construct an HSMM is to employ an explicit model of durations via parametric distribution, such as multinomial or Gamma distributions. However, this often incurs a heavy computational cost. Another practical way, which is employed in this section, is to control the duration distribution of single states implicitly by expanding a state of HMM into a Markov-chain of multiple states. This type of HSMM is computationally tractable, while it has been empirically shown to have sufficient ability to represent realistic duration distributions [54]. Along this line, we refine the Markov dynamics described in Sec. 2.3 to have a semi-Markov property by the following simple modification. We first introduce a *phase variable*, $m_{i,t}$, that takes a discrete value, and define an augmented switching variable as $\tilde{z}_{i,t} = (z_{i,t}, m_{i,t})$. If we newly consider a Markov chain on $\tilde{z}_{i,t}$, the duration distribution of $z_{i,t} = 0$ or 1 (irrespective to $m_{i,t}$) is no longer geometric. For simplicity, we assume $m_{i,t}$ takes a binary value, 0 or 1, and the Markov process on $\tilde{z}_{i,t}$ is assumed to have a limited number of parameters [5]. Figure 2.13 shows the transition diagram of the four states of $\tilde{z}_{i,t}$. The initial probability is given as

$$p(\tilde{z}_{i,1}) = p(z_{i,1}, m_{i,1}) = \begin{cases} 0.5(1-\pi_i) & \text{if } z_{i,t} = 0 \\ 0.5\pi_i & \text{if } z_{i,t} = 1 \end{cases}. \tag{2.35a}$$

The transition probabilities are given as follows. Let $\rho_i(z) = \rho_{0i}^{1-z}\rho_{1i}^{z}$, then, if $m_{i,t-1} = 0$,

$$p(z_{i,t}, m_{i,t} \mid z_{i,t-1}, m_{i,t-1} = 0) = \begin{cases} \rho_i(z_{i,t-1}) & \text{if } m_{i,t} = 0,\ z_{i,t} = z_{i,t-1} \\ 1 - \rho_i(z_{i,t-1}) & \text{if } m_{i,t} = 1,\ z_{i,t} = z_{i,t-1} \\ 0 & \text{otherwise} \end{cases}, \tag{2.36}$$

[5] In the current setting, the number of parameters is actually the same as in the original Markov dynamics.

and if $m_{i,t-1} = 1$,

$$p(z_{i,t}, m_{i,t} \mid z_{i,t-1}, m_{i,t-1} = 1) = \begin{cases} \rho_i(z_{i,t-1}) & \text{if } m_{i,t} = 1,\ z_{i,t} = z_{i,t-1} \\ 1 - \rho_i(z_{i,t-1}) & \text{if } m_{i,t} = 0,\ z_{i,t} \neq z_{i,t-1} \\ 0 & \text{otherwise} \end{cases} . \quad (2.37)$$

Figure 2.14 shows an example of duration distributions for the Markov (geometric) and the semi-Markov processes [6]. In the semi-Markov model, the duration probability does not simply decrease with time, but increases with time to a peak and then falls with a heavy tail.

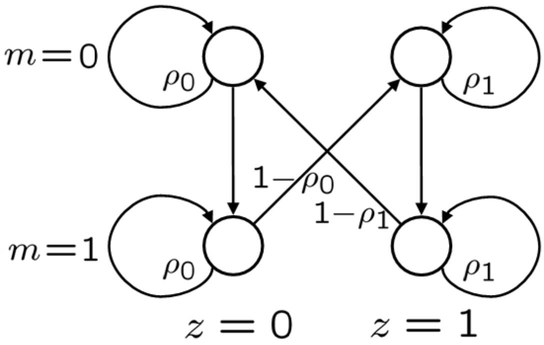

Figure 2.13. Transition diagram of the augmented switching variable $\tilde{z} = (z, m)$ in the semi-Markov model. The transition from active ($z = 1$) to inactive ($z = 0$) or from inactive to active, occurs only if the phase variable is $m = 1$.

The simple setting of this semi-Markov model alters the original algorithm only in the following points. First, we now carry out the Forward-Backward calculation on $\tilde{\boldsymbol{z}}_t$ instead of on $\boldsymbol{z}_t$, assigning the same evidence $e(\boldsymbol{x}_t, \boldsymbol{z}_t)$ for $\tilde{\boldsymbol{z}}_t$ that have the same $\boldsymbol{z}_t$ realization. We note that the computational cost in this calculation can be reduced to some extent by carefully considering the structure

[6] The probability distribution of durations in the semi-Markov model is given by

$$p(d) = \begin{pmatrix} 0 \\ 1-\rho \end{pmatrix}^T \begin{pmatrix} \rho & 0 \\ 1-\rho & \rho \end{pmatrix}^{d-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

where d and ρ respectively denote the duration period and the self-transition parameter.

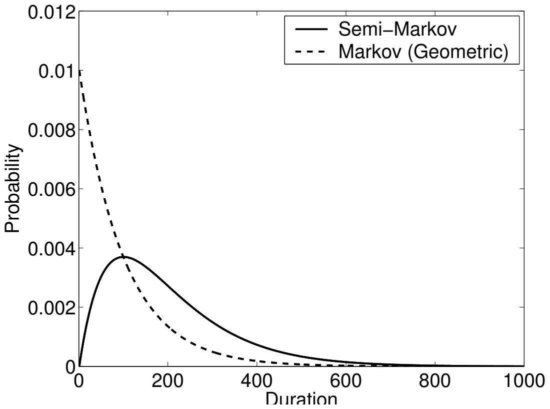

Figure 2.14. Example of duration distributions. The dotted and solid lines correspond to the Markov and semi-Markov dynamics, respectively, with the same self-transition parameter, $\rho = 0.99$.

of Markov chains in Fig. 2.13, in which the large parts of the transition matrix have zero values. Second, after the Forward-Backward calculation, the marginals, $q(\boldsymbol{z}_t)$ and $q(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$, should be calculated by marginalizing $\boldsymbol{m}_t$. Finally, the variational free energy should be modified appropriately so as to correspond to the semi-Markov chain. Other parts of the learning algorithm are the same as in the original SwICA.

Figure 2.15 shows estimation results by the semi-Markov modification of Switching ICA (SwICA-SM). In this simulation, SwICA-SM is initialized based on the result by Markov-SwICA (SwICA-M). The other simulation setting was the same as in Sec. 4.1. The result is of the highest aSDR out of 50 runs, at SNR $= 8$. By comparing it with the result by SwICA-M in Fig. 2.4, the switching variables were recovered more clearly. Figure 2.16 shows that the estimation of switching variables was actually improved from that by SwICA-M at various SNR levels (such as 4, 8 and 12). Figure 2.17 shows the estimation performance of source signals and mixing matrix by SwICA-SM in comparison to that by SwICA-M. Although the performance was at a comparable or slightly inferior level to that of SwICA, the performance was still higher than those of the other algorithms such as NoDyna and HMM-ICA.

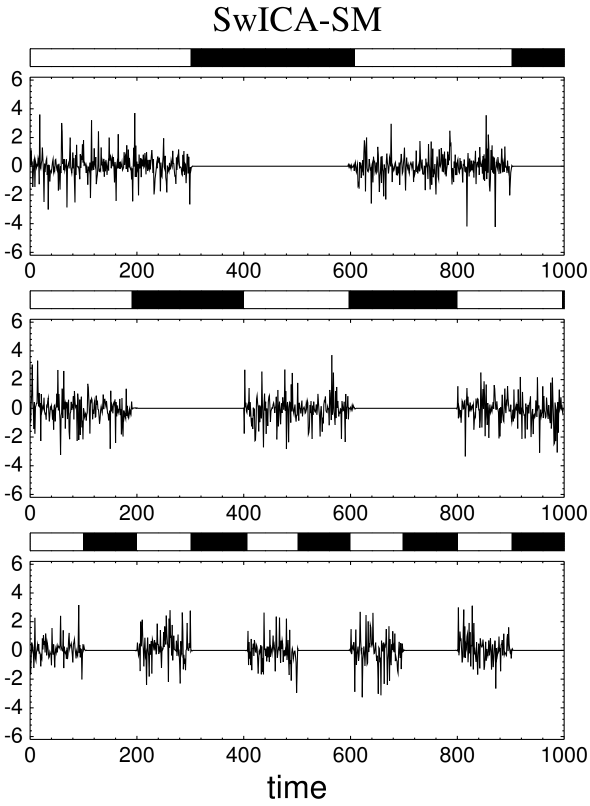

Figure 2.15. Examples of recovered source signals and the switching variables by the semi-Markov extension of SwICA. The result in this figure can be compared with the results by the other algorithms in Fig. 2.4.

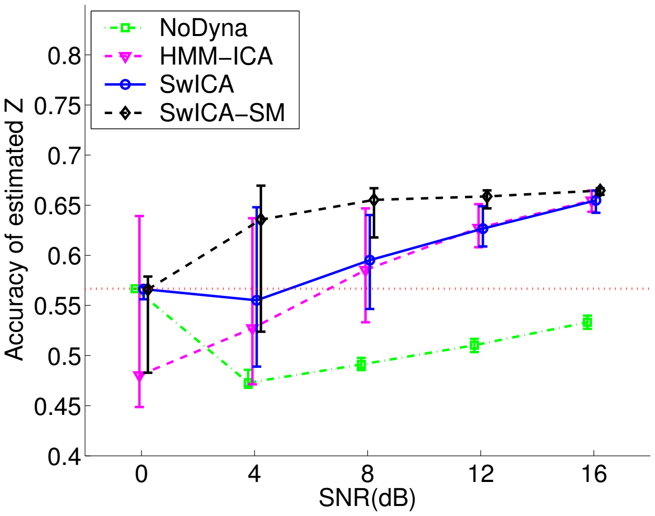

Figure 2.16. The accuracy of MAP estimates of switching variables by NoDyna, HMM-ICA, SwICA(-M) and SwICA-SM. The dotted horizontal line denotes the accuracy when the sources are estimated as always being active.

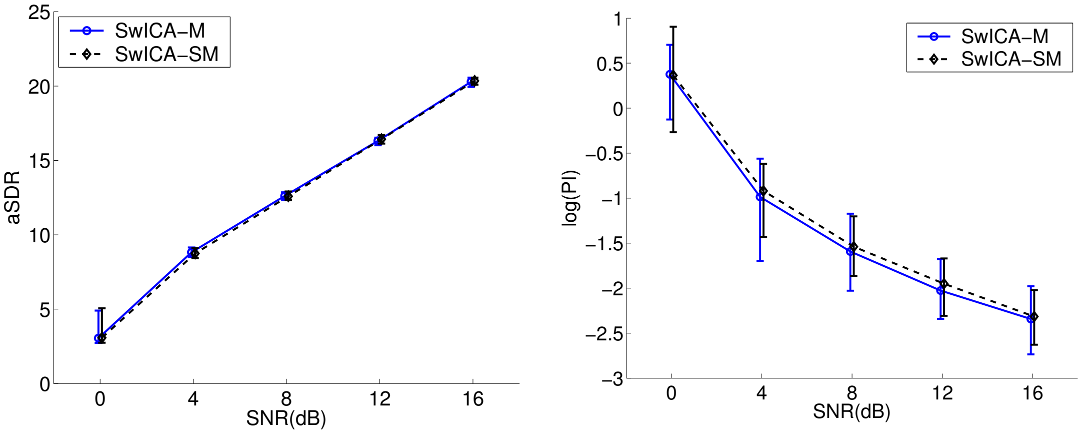

Figure 2.17. The performance of SwICA-SM at different noise levels compared to that of SwICA. The left panel shows the aSDR and the right one the logarithm of PI, by the four algorithms. An error bar represents the range between the 10 percentile (lower) and the 90 percentile (upper). These panels show that the performance of SwICA-SM is comparable to or slightly inferior to that of SwICA.

6. Discussion

In our experiments, the proposed Switching ICA exhibited high performance in the situations where sources dynamically switch on and off with time, especially when some amount of noise was present. The results in Sec. 4.4 showed that the Dirac's Delta prior in conjunction with the Markov dynamics is useful for robust estimation of source signals and their presence/absence. That method was shown to be robust also in the cases when the total number of sources was overestimated or actually larger than that of the mixtures. The two algorithms, the ones without Dirac's Delta (HMM-ICA) and without dynamics (NoDyna), were found to be ineffective in our experiments. Our results revealed that the Dirac's Delta prior is effective at relatively low SNRs for suppressing unnecessary reconstruction of source signals in inactive periods, while the temporal dynamics is needed at higher SNRs where the difference between the assumed (i.i.d.) and the true (non-i.i.d.) processes would matter. This is typically observed when comparing the switching variables estimated by NoDyna and SwICA in Fig. 2.4. In addition, we showed that the semi-Markov extension is also beneficial for more accurate estimation of switching variables, with a comparable performance to the original Markov model in source separation and mixing matrix estimation. The accurate estimation of active/inactive states would be useful for application to recognition tasks such as speaker identification from audio signals.

The Markov version of our Switching ICA can be regarded as a special case of HMM-ICA, such that the variance of one conditional source model is set to the zero limit. Thus, at least in principle, HMM-ICA should be capable of performing comparably with SwICA if the model parameter could approach an ideal one. In our experiments, however, the performance of HMM-ICA was found to be inferior to that of SwICA, except for the case of audio sources in the presence of a relatively high SNR. One potential reason for this degradation is that the learning process of HMM-ICA could be highly variable due to the redundant model capacity (which leads to a large parameter space to be explored). The solution would then be affected more than SwICA by, for instance, noise, small sample sizes, or inappropriate initialization conditions. If we have sufficient amount of data or a more effective initialization scheme, then the performance of HMM-ICA may improve to a level comparable with that of SwICA. The results may be further improved by longer simulations based on a more careful convergence judgment, even though each trial could be seen as roughly converged in Fig. 2.5. If we cannot prepare such improved situations in practice, however, the special SwICA method is much more advantageous, as shown in our experiments, by avoiding the potential variability and non-robustness of the HMM-ICA method. The original version of HMM-ICA is, on the other hand, effective at incorporating a more general non-stationarity of (active) source signals. Accordingly, solid contributions of the present study are to incorporate the inactive source model into the general HMM-ICA model and to introduce the semi-Markov dynamics into the non-stationary ICA context. We also note that a similar approach based on Markov-Delta setting has recently been investigated to model audio signals particularly in the time-frequency domain [35].

Switching ICA also has a close relationship to other existing principles. First, the use of Dirac's Delta prior to effectively suppress irrelevant parts of the model is actually an essential idea of the Bayesian variable selection [37, 40, 38, 17] usually employed in multiple regression problems, which use a similar prior (usually a mixture of a Gaussian and a Dirac's Delta) on the regression coefficient. The SwICA model can be regarded as a dynamic version of the Bayesian variable selection, by incorporating the Markov (or semi-Markov) property into the prior. We note that the variational free energy criterion [10], which is a conventional method of model selection in the VB framework, cannot be utilized for this purpose, since it does not accommodate the case in which the model structure may change within a single dataset. Second, if we assume no dynamics on the switching variables (which is exactly the case of NoDyna), the source prior in SwICA becomes a kind of sparse prior, which has a high density at zero and a heavy tail, utilized in problems of learning overcomplete representations [73, 59, 74]. The difficulty with such overcomplete situations is the existence of indeterminacy in recovering source signals, then its resolution requires appropriate prior knowledge of the original sources. In Sec. 4.1, it was shown that the temporal information, as well as the sparseness, can be useful for recovering source signals in overcomplete cases. Although previous studies of the sparse representation usually consider i.i.d. data instead of time-series signals, the proposed method will be useful when considering, for example, sequential images like video images, since sparseness-based methods have been successfully applied to static images without temporal structure.

Although our Switching ICA was shown to be effective in various cases considered in our experiments, there still can be situations where its BSS performance may be insufficient and there is room for improvement. First, the Switching ICA model assumes each source signal have no temporal dependence in active periods. When active sources by themselves have clear temporal structures, the performance of Switching ICA would not be enough and could be improved by carefully addressing the temporal structure. One potential approach to this issue is to model the active sources by HMM with a general structure; in this case, the model is an extension of usual HMM-ICA which incorporates the inactive source model (with or without the semi-Markov dynamics). Second, the switching variables are assumed to be mutually independent from each other. This is a natural setting in the context of ICA, while it is an interesting issue to incorporate a higher-order correlation among the sources (as in [51]), by introducing a mutual dependence among the switching variables. In audio cases, for example, the appearance of each speaker is often not independent from those of the other speakers as in conversation of multiple persons. In such cases, the performance of our Switching ICA would degrade, especially in noisy situations. To investigate extensions of Switching ICA to such situations is remained for our future study.

A major drawback of the Switching ICA is its high computational cost. In this study, we employed a rather naive implementation of computing posterior distributions in the VB-E step. The cost will then grow exponentially with the number of potential sources, n, and so is intractable when n is large. The reason is twofold. First, the joint posterior distribution of all of the n sources, Eq. (2.23b), will have an intractable number of components when n^h is large. Second, the Forward-Backward calculation for the switching vector $\boldsymbol{z}_t$, which possibly takes one of 2^n states, also becomes intractable for a large n. The same difficulty also arises in the semi-Markov model. A popular and accessible way to reduce such an exponential cost of joint posterior computations is to employ the naive mean-field approximation for the source posterior as in [7], which make the cost to a polynomial order. A more advanced mean-field approach investigated in [47] would also be useful to improve the accuracy of naive mean-field approximation. Alternatively, Monte Carlo techniques may be employed, as reported in the Bayesian variable selection literature [37, 40, 38, 17], specifically by using the sequential Monte Carlo [30] for estimating the underlying dynamic processes. In addition to the naive implementation of VB-E step, there is an another cause that may slower the computational speed. As reported recently in [77], the convergence of VB-EM often becomes quite slow in low-noise situations, due to the strong posterior dependence between latent variables and model parameters. To overcome this problem, some techniques that have been proposed to improve the convergence of EM, such as a gradient-based optimization or a heuristic procedure (both suggested in [77]), would be available.

Chapter 3

Balancing plasticity and stability of online Bayesian learning

1. Introduction

Online learning is a framwork of learning, in which the learning model attempts to adapt to new inputs incrementally without retaining the series of past inputs. This is in contrast to batch learning which is executed after all the inputs are given and retaining the past inputs in the memory. Online learning thus requires less memory than batch schemes, and learning can be started even when only part of the data has been observed, both of which are important properties in practice. Recently, Sato [87] have proposed an online variational Bayes (VB) method that is an effective online learning scheme based on Bayesian inference. The Bayesian framework naturally incorporates a principled way of model selection and potentially avoids overlearning phenomena that may degrade the learning performance. Although an exact implementation of Bayesian inference is usually intractable, VB methods [8, 71], which were originally developed as a batch-type learning scheme, provide an effective approximation. The VB methods also can naturally accommodate probabilistic models that involve latent variables. The online VB method is an alternative to the standard VB within online learning scenarios.

Besides the above basic advantages of online learning, an important character is its potential to adapt to changing environments by properly adjusting a meta-parameter that controls the balance of plasticity and stability[1] of the learning model. In an early stage after an environmental change, the learning model should exhibit high plasticity (and low stability) to accelerate the learning to quickly assimilate the new inputs; in contrast, it should shift to lower plasticity (and higher stability) in the subsequent stage to gradually decrease the learning speed to stabilize it and realize a proper stochastic approximation. Although a number of studies have concentrated on such adaptive control mechanisms of online learning [3, 25, 90, 69, 88, 68], no study has paid special attention to online VB learning. In this chapter, we propose two methods to control the balance of plasticity and stability of learning model in the online VB framework.

Online VB involves a meta-parameter, called *forgetting factor*, which can be regarded as to modulate the weights or cofidence on past inferences about latent variables. In the standard formulation of the online VB method, only the expected sufficient statistics are explicitly maintained at each time step to estimate model parameters, where these values are incrementally updated according to a new datum. The forgetting factor regulates the speed of this updates indirectly, and thus determine the balance between the plasticity and stability. To address environmental changes, the forgetting factor at each time step should be appropriately scheduled with reflecting the changes, while the occurrences of changes are explicitly unknown in usual.

The two methods proposed in this chapter are both to control the forgetting factor in adaptive manners, with estimating environmental changes, but are realized by two different architectures. In the first one, dynamic control of the forgetting factor is realized by two steps: First, probabilistic novelty detection is performed to evaluate the posterior probability of environmental changes; and then, the forgetting factor at the moment is determined based on the posterior probability, i.e., evaluated degree of novelty, in a simple way. In Sec. 3, we describe the proposed method of probabilistic novelty detection based on a simple mixture model, and also a specific scheduling scheme of the forgetting factor according to the novelty. In the second one, a hierarchical Bayes technique naturally integrate the two steps in the first approach within a theoretically-consistent framework. We show that the online VB learning can be interpreted as a special

[1] These terms follow Grossberg's "plasticity/stability dilemma" [43].

type of incremental Bayes updating in Sec. 4, with treating the forgetting factor as a hyperparameter of prior belief at each time step. The scheduling of forgetting factor is then acheived by a hierarchical Bayes inference based on this new fact.

The two framework proposed in this chapter are quite general and can be potentally applied to many kinds of probabilistic unsupervised models. The principal interest in this chapter, however, is in representation learning, particularly in its application to online feature extraction in dynamic environments. We thus employ one simple probabilistic model for representation learning, the probabilistic principal component analysis (PPCA) model [92]. The two online learning models are validated through computer simulations using both artificial and realistic datasets, focusing on the task of feature extraction from sequential inputs with accommodating environmental changes. Furthermore, while this chapter describe these issues mainly from an engineering viewpoint, the work in this chapter is originally started from a question how a brain learn appropriately internal representations, or the features, of sensory inputs in changing environments. Thus we also discuss briefly the biological implication of the proposed dynamic learning scheme especially with hypothesizing its potential implementation realized in brain.

2. Online VB for probabilistic PCA

2.1 Probabilistic PCA

PPCA is a probabilistic generative model with latent variables, such that its maximum likelihood (ML) estimation is equivalent to the usual PCA [92]. PPCA for an n-dimensional observed variable $\boldsymbol{x}_t \in \Re^n$ is given by

$$\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{y}_t + \boldsymbol{\mu} + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}_n\left(\boldsymbol{\xi}_t \mid \boldsymbol{0}, \sigma_x^2 \boldsymbol{I}_n\right), \tag{3.1}$$

where t denotes the discrete time or the sample index. $\boldsymbol{y}_t \equiv (y_{t,1}, \cdots, y_{t,m})^T \in \Re^m$ $(m \leq n)$ is a latent variable corresponding to a principal component score, which is generated independently at each time step from a standard Gaussian distribution. The superscript ^{T} denotes the transpose. $\boldsymbol{\xi}_t \in \Re^n$ is a white noise and $\mathcal{N}_p(\cdot \mid \cdot, \cdot)$ denotes a p-dimensional Gaussian density function [2]. $\boldsymbol{I}_n$ is an $n \times n$ identity matrix, and σ_x^2 ($\sigma_x^2 > 0$) is an observation noise variance which is assumed to be a known constant for simplicity. $\boldsymbol{W} \equiv (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m) \in \Re^{n \times m}$ is the principal component loading matrix, where $\boldsymbol{w}_j \in \Re^n (j = 1, \cdots, m)$ is a principal component vector. $\boldsymbol{\mu} \in \Re^n$ is the expected observation. With the notations: $\boldsymbol{\Theta} \equiv (\boldsymbol{W}, \boldsymbol{\mu}) \in \Re^{n \times (m+1)}$ and $\tilde{\boldsymbol{y}}_t \equiv \left(\boldsymbol{y}_t^T, 1\right)^T \in \Re^{(m+1)}$, Eq. (3.1) is rewritten as

$$\boldsymbol{x}_t = \boldsymbol{\Theta}\tilde{\boldsymbol{y}}_t + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}_n\left(\boldsymbol{\xi}_t \mid \boldsymbol{0}, \sigma_x^2 \boldsymbol{I}_n\right). \tag{3.2}$$

2.2 Online VB learning

The model parameter $\boldsymbol{\Theta}$ can be inferred using an online variational Bayes (VB) method [87]. Let $(X_{1:t}, Y_{1:t}) \equiv \{(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau) \mid \tau = 1, \ldots, t\}$ be a sequence of observations and corresponding latent variables. The objective of the Bayesian inference is to obtain a posterior distribution of unknown variables, $p(Y_{1:t}, \boldsymbol{\Theta} \mid X_{1:t})$, when given observation variables $X_{1:t}$. For this purpose, an online variational free energy with a time-dependent forgetting factor $\lambda(s) \in [0, 1]$ $(s = 1, \ldots, t)$ is defined by

$$F^\lambda\left[q\right](t) = T^\lambda(t) L^\lambda(t) - H(t) \tag{3.3a}$$

$$L^\lambda(t) = \eta\left(t\right) \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda\left(s\right) \right) E\left[\log \frac{p\left(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta}\right)}{q_\tau\left(\boldsymbol{y}_\tau \mid \boldsymbol{x}_\tau\right)}\right] \tag{3.3b}$$

$$H(t) = E\left[\log \frac{q_\theta\left(\boldsymbol{\Theta} \mid X_{1:t}\right)}{p\left(\boldsymbol{\Theta}\right)}\right], \tag{3.3c}$$

where $q\left(Y_{1:t}, \boldsymbol{\Theta} \mid X_{1:t}\right)$ is a trial distribution to approximate the true posterior distribution $p(Y_{1:t}, \boldsymbol{\Theta} \mid X_{1:t})$, and $E[\cdot]$ denotes the expectation over the trial distribution q. Online VB usually assumes a factorized form of

Furthermore, $p(\boldsymbol{\Theta})$ is the prior distribution of $\boldsymbol{\Theta}$. $T^\lambda(t) \equiv \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda\left(s\right)\right)$ is an effective data number and $\eta(t) \equiv 1/T^\lambda(t)$ is the normalization term. The online VB method for PPCA is derived as a sequential maximization process of the variational free energy (3.3). When a datum $\boldsymbol{x}_t$ is observed at time t, F^λ is

[2] $\mathcal{N}_p\left(\boldsymbol{x} \mid \boldsymbol{m}, \boldsymbol{\Sigma}\right) \equiv (2\pi)^{-p/2} \left|\boldsymbol{\Sigma}\right|^{-1} \exp\left[-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{m}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x} - \boldsymbol{m}\right)\right]$, where $\boldsymbol{x} \in \Re^p$ is a random vector. $\boldsymbol{m} \in \Re^p$ and $\boldsymbol{\Sigma} \in \Re^{p \times p}$ are a mean vector and a covariance matrix, respectively.

maximized with respect to q_t in the online VB-Estep while q_τ $(\tau = 1, \dots, t-1)$ and q_θ are fixed. In the next step, called the online VB-Mstep, F^λ is maximized with respect to q_θ while q_τ $(\tau = 1, \dots, t)$ is fixed. These two steps are executed every time a new datum is observed. The solutions of the two steps at time t can be obtained as closed forms:

$$q_t(\boldsymbol{y}_t \mid \boldsymbol{x}_t) = \frac{\exp\bigl(E_{\boldsymbol{\Theta}}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta})\right]\bigr)}{\int d\boldsymbol{y}_t \exp\bigl(E_{\boldsymbol{\Theta}}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta})\right]\bigr)}, \tag{3.4a}$$

$$q_\theta(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\Bigl(\sum_{\tau=1}^{t}\left(\prod_{s=\tau+1}^{t}\lambda\left(s\right)\right)E_{\boldsymbol{y}_\tau}\left[\log p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta})\right]\Bigr)p_0(\boldsymbol{\Theta})}{\int d\boldsymbol{\Theta}\exp\Bigl(\sum_{\tau=1}^{t}\left(\prod_{s=\tau+1}^{t}\lambda\left(s\right)\right)E_{\boldsymbol{y}_\tau}\left[\log p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta})\right]\Bigr)p_0(\boldsymbol{\Theta})}, \tag{3.4b}$$

where $E_{\boldsymbol{\Theta}}[\cdot]$ and $E_{\boldsymbol{y}_\tau}[\cdot]$ denote expectations over trial distributions $q(\boldsymbol{\Theta})$ and $q(\boldsymbol{y}_\tau)$, respectively.

2.3 Forgetting factor and learning rate

The original VB method [8, 14, 71] can be regarded as a special case of the online VB method, in which $\lambda(s) = 1$ for all s and the E and M steps are applied in a batch manner after all data are observed. Instead of storing all observed data, however, the online VB method needs only to maintain the expected sufficient statistics; this scheme is more natural for learning by animals than the batch one. The expected sufficient statistics are defined by

$$\left\langle f(\boldsymbol{x}, \boldsymbol{y})\right\rangle(t) = \eta(t)\sum_{\tau=1}^{t}\left(\prod_{s=\tau+1}^{t}\lambda\left(s\right)\right)E\left[f(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau)\right], \tag{3.5}$$

where $f(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is given by a quadratic function of $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{y}}_t$ in the case of PPCA (see Appendix C.1 for details). This calculation can be done incrementally as

$$\left\langle f(\boldsymbol{x}, \boldsymbol{y})\right\rangle(t) = (1 - \eta(t))\left\langle f(\boldsymbol{x}, \boldsymbol{y})\right\rangle(t-1) + \eta(t)E\left[f(\boldsymbol{x}_t, \boldsymbol{y}_t)\right], \tag{3.6}$$

where the normalization term $\eta(t)$ acts as the learning rate to control the speed of updating $\left\langle f(\boldsymbol{x}, \boldsymbol{y})\right\rangle(t)$. $\eta(t)$ can also be calculated incrementally, because its reciprocal $T^\lambda(t)$ is given by the following step-wise equation

$$T^\lambda(t) = 1 + \lambda(t)T^\lambda(t-1). \tag{3.7}$$

Even if such a step-wise calculation is used, it is shown that the online VB method achieves a stochastic approximation of the Bayesian inference if scheduling like $\lambda(s) \overset{s\to\infty}{\longrightarrow} 1$ is used [87].

Forgetting factor $\lambda(t)$ controls the balance between plasticity and stability of online VB learning. According to Eq. (3.7), a large $\lambda(t)$ allows the effective data number $T^{\lambda}(t)$ to increase, after which the learning rate $\eta(t)$ becomes small. In contrast, a small $\lambda(t)$ makes $T^{\lambda}(t)$ small, and as a result, $\eta(t)$ becomes large. The learning rate $\eta(t)$ regulates the updating speed of sufficient statistics as shown in Eq. (3.6), so that it directly balances the plasticity and stability of the online learning process. In a dynamic environment, then, $\lambda(t)$ should be a rather small values so as to make $\eta(t)$ large (high plasticity) when the environment changes, while it should be close to one during stationary periods so as to make $\eta(t)$ small (high stability).

3. Novelty-based scheduling using mixture model

3.1 A mixture of PPCA

Let $z_t \in \{0, 1\}$ be an indicator variable regarded as a latent variable. The probabilistic generative model of the simple version of MPPCA is given by

$$\boldsymbol{x}_t = \boldsymbol{\Theta}\tilde{\boldsymbol{y}}_t + \boldsymbol{\xi}_t + z_t\boldsymbol{\zeta}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}_n\left(\boldsymbol{\xi}_t \mid \boldsymbol{0}, \sigma_x^2\boldsymbol{I}_n\right), \quad \boldsymbol{\zeta}_t \sim \mathcal{N}_n\left(\boldsymbol{\zeta}_t \mid \boldsymbol{0}, \sigma_\zeta^2\boldsymbol{I}_n\right). \tag{3.8}$$

The third term is the additional noise, where σ_ζ^2 is a constant noise variance and is known. The joint probability distribution for a triplet $(\boldsymbol{x}_t, \boldsymbol{y}_t, z_t)$ is given by

$$p\left(\boldsymbol{x}_t, \boldsymbol{y}_t, z_t = 0 | \boldsymbol{\Theta}, m\right) = (1-r)\mathcal{N}_m\left(\boldsymbol{y}_t | \boldsymbol{0}, \boldsymbol{I}_m\right)\mathcal{N}_n\left(\boldsymbol{x}_t | \boldsymbol{\Theta}\tilde{\boldsymbol{y}}_t, \sigma_x^2\boldsymbol{I}_n\right), \tag{3.9a}$$

$$p\left(\boldsymbol{x}_t, \boldsymbol{y}_t, z_t = 1 | \boldsymbol{\Theta}, m\right) = r\mathcal{N}_m\left(\boldsymbol{y}_t | \boldsymbol{0}, \boldsymbol{I}_m\right)\mathcal{N}_n\left(\boldsymbol{x}_t | \boldsymbol{\Theta}\tilde{\boldsymbol{y}}_t, \sigma_\epsilon^2\boldsymbol{I}_n\right), \tag{3.9b}$$

here $\sigma_\epsilon^2 = \sigma_x^2 + \sigma_\zeta^2$. Here, the principal component dimensionality m is explicitly expressed. The prior probability for the index variable, $P(z_t = 1) = 1 - P(z_t = 0) = r$, is assumed to be known such to represent the *a priori* knowledge of the occurrence probability of environmental changes.

Since the two PPCA components, (3.9a) and (3.9b), have the same parameter $\boldsymbol{\Theta}$ except for different Gaussian noises, an environmental change can be detected according to the following principle. Consider a situation where the model parameter $\boldsymbol{\Theta}$ has been estimated from the previous observations $\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots$ and then a new observation $\boldsymbol{x}_t$ is given. If $\boldsymbol{x}_t$ is generated in the current environment, it can be described sufficiently by the component (3.9a) with the regular noise variance σ_x^2, thus the posterior probability of $z_t = 1$ becomes small. If $\boldsymbol{x}_t$ is generated in a novel environment, it can be regarded as an outlier in the component (3.9a). In this case, the observation can be described better by the component (3.9b) having a larger noise variance σ_ϵ^2, thus the posterior probability of $z_t = 1$ becomes large. Accordingly, the posterior probability of $z_t = 1$ can be viewed as a confidence of environmental change between time steps $t-1$ and t.

3.2 Novelty-based scheduling of forgetting factors

Based on the posterior probability of z_t, which informs of the occurrence of environmental changes, our online learning is able to regulate the learning dynamics by scheduling $\lambda(t)$ as

$$\lambda\left(t\right) = \left(1-\alpha\right)\lambda\left(t-1\right) + \alpha(1 - q_t(z_t = 1)), \tag{3.10}$$

where $q_t(z_t = 1)$ denotes the posterior probability of $z_t = 1$ at time t. α $(0 < \alpha < 1)$ is a smoothing constant to reduce an excessive sensitivity to outliers that may appear even in a static environment. When an environment changes, a temporal increase of $q_t(z_t = 1)$ results in the decrease of $\lambda(t)$. This means that an environmental change induces an increase of the ACh level, and facilitates the learning by placing more weight on the recent data than the previous data. $q_t(z_t = 1) = 0$ for any t implies $\lambda(t) \overset{t\to\infty}{\longrightarrow} 1$ from any $\lambda(0)$, so that the online VB learning achieves stochastic approximation of the Bayesian inference if the environment continues to be stationary.

Although the scheduling by Eq. (3.10) makes $\lambda(t)$ low after an environmental change is detected, subsequent learning requires that $\lambda(t)$ gradually increases in order to conduct proper online learning in the new environment. However, $\lambda(t)$ often fails to recover and remains low because $q_t(z_t = 1)$ is apt to be high due to the unfaithful model that exists at the beginning of the new environment, which becomes serious especially in a high-dimensional case. We therefore introduce a refractory period (RP) into the scheduling, in order that $\lambda(t)$ recovers after dropping to almost zero in response to an environmental change; if $\lambda(t)$ is below a threshold ϕ at time $t = t_0$, $q_t(z_t = 1)$ in Eq. (3.10) is explicitly replaced by 0 during $t = t_0 + 1, \ldots, t_0 + \nu$, where ϕ and ν are constant parameters. The effect of RP is shown in the simulation in Section 3.4.

3.3 A criterion for online model selection

Since the indicator variable z_t is added to the set of latent variables, our online VB learning for the MPPCA is modified into a sequential maximization of the following online variational free energy $F_m^\lambda[q]$ (see Appendix C.1):

$$F_m^\lambda[q](t) = T_m^\lambda(t) L_m^\lambda(t) - H_m(t) \tag{3.11a}$$

$$L_m^\lambda(t) = \eta_m(t) \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda_m(s) \right) E\left[\log \frac{p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau, z_\tau \mid \boldsymbol{\Theta}, m)}{q_\tau(\boldsymbol{y}_\tau, z_\tau | \boldsymbol{x}_\tau, m)} \right] \tag{3.11b}$$

$$H_m(t) = E\left[\log \frac{q_\theta(\boldsymbol{\Theta} \mid X_{1:t}, m)}{p(\boldsymbol{\Theta} \mid m)} \right], \tag{3.11c}$$

where $Z_{1:t} \equiv \{z_\tau \mid \tau = 1, \ldots, t\}$. The trial distribution, $q(Y_{1:t}, Z_{1:t}, \boldsymbol{\Theta} \mid X_{1:t}, m)$, is factorized as $q_\theta(\boldsymbol{\Theta} \mid X_{1:t}, m) \prod_{\tau=1}^{t} q_\tau(\boldsymbol{y}_\tau, z_\tau \mid \boldsymbol{x}_\tau, m)$. The other notations are the same as those in Section 2.1, except that the dependence on the principal component dimensionality m is explicitly noted here.

When the principal component dimensionality m is unknown, our online VB learning can estimate it within the framework of hierarchical Bayesian inference. In the general online VB learning [87], an MAP estimator of m is approximately obtained by $m^* = \mathrm{argmax}_m F_m^\lambda[q]$ after applying the VB learning to models with $m = 1, \cdots, n$, under the assumption that the prior distribution $p(m)$ $(m = 1, \cdots, n)$ is non-informative. In our online VB scheme, however, the online variational free energy $F_m^\lambda[q]$ is dependent on the effective data number $T_m^\lambda(t)$, which may vary among $m = 1, \ldots, n$, because $T_m^\lambda(t)$ is dependent on $\lambda(t)$ (see Eq. (3.7)) and $\lambda(t)$ is dependent on the model's representation ability. To conduct an appropriate model selection, the influence of the effective data number in each model, $m = 1, \ldots, n$, should be normalized:

$$\tilde{F}_m^\lambda[q](t) = F_m^\lambda[q](t) / T_m^\lambda(t) = L_m^\lambda(t) - H_m(t) / T_m^\lambda(t). \tag{3.12}$$

Although determining the principal component dimensionality based on this criterion is heuristic, it works well as can be seen in the next section.

3.4 Simulations

Our learning model introduced in the previous section was evaluated using two types of computer simulations, which employed synthesized and real data sets.

Synthesized data

The basic features of our approach were examined by using simple two-dimensional synthesized data. A two-dimensional vector $\boldsymbol{x}_t$ was generated according to Eq. (3.2) at each time step t, where the actual parameter $\boldsymbol{\Theta} = (\boldsymbol{W}, \boldsymbol{\mu})$ was usually fixed but occasionally changed. The known constants were set as follows: $\sigma_x^{-2} = 10^{-2}, \sigma_\zeta^{-2} = 10^{-6}, r = 0.001, \alpha = 0.05$, and $\gamma = 0.001$. In this simulation, the RP described in Section 3.2 was not used.

First, we assumed a situation in which the principal component dimensionality m of actual data is known as $m = 1$. The actual parameters at each time step were: $\boldsymbol{\Theta} = \begin{pmatrix} 5 & 10 \\ -1 & 10 \end{pmatrix}$ for $t = 1, \ldots, 200$, $\begin{pmatrix} 1 & -10 \\ 5 & 10 \end{pmatrix}$ for $t = 201, \ldots, 400$, and $\begin{pmatrix} -3 & -10 \\ 3 & -10 \end{pmatrix}$ for $t = 401, \ldots, 600$. Figure 3.7 shows learning processes under the following three conditions: 1) the forgetting factor was fixed at $\lambda(t) = 1$ for any t; 2) fixed at $\lambda(t) = 0.8$ for any t; and, 3) $\lambda(t)$ was scheduled by Eq. (3.10). Only the direction of the estimated principal component vector, represented as the angle from the x_1-axis, is shown in this figure. The estimator of the learning model is given as the expectation [3] of the model parameter over the trial posterior distribution. When the forgetting factor $\lambda_m(s)(s = 1, \ldots, T)$ was set at constant of 1, the estimator closely approached the true value in a stationary environment, but it could not follow environmental changes. When the forgetting factor was set at a smaller constant of 0.8, the estimator could alter its value in response to environmental changes, but a high variance remained. Because of this variance, the estimator could not be improved in a stationary environment even when the

[3] In our case, the posterior distribution is Gaussian, thus the parameter expectation is identical to its mean.

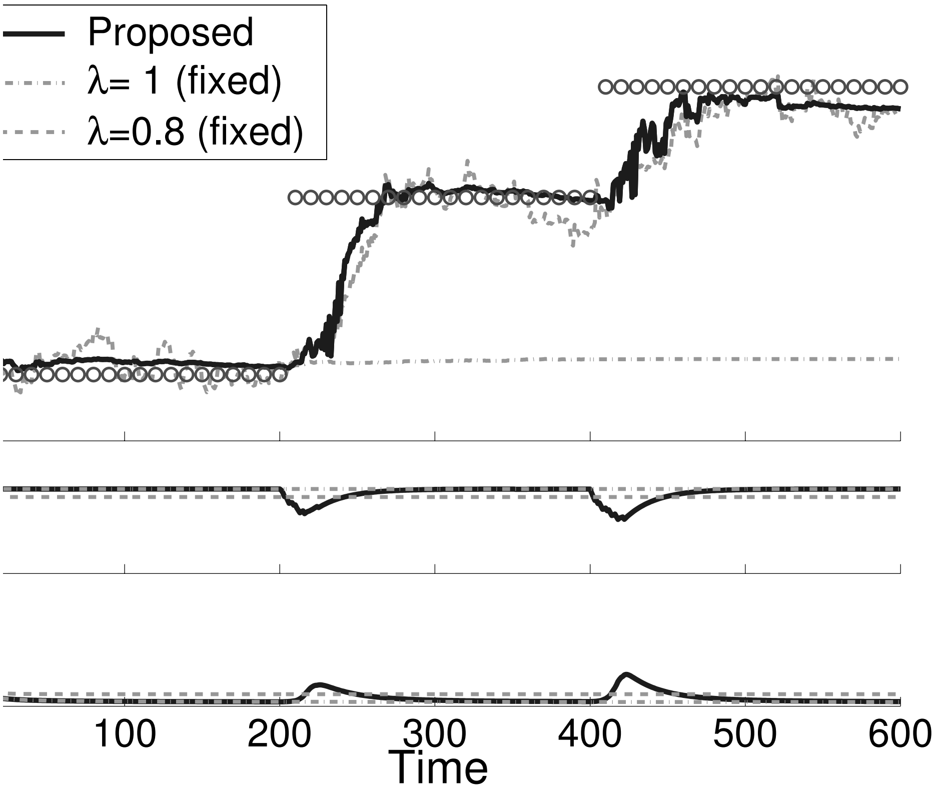

Figure 3.1. The direction of the estimated principal component vector and modulatory variables. The horizontal axis denotes the time step t. In the top panel, the estimator is shown in the three cases of $\lambda(t) = 1.0$, $\lambda(t) = 0.8$, and $\lambda(t)$ is controlled by the proposed scheduling scheme. Only the direction of the principal component vector, the angle from the x_1-axis, is shown. A mark '$\circ$' represents a real value in each time step. The middle and bottom panels show the time series of $\lambda(t)$ and $\eta(t)$, respectively.

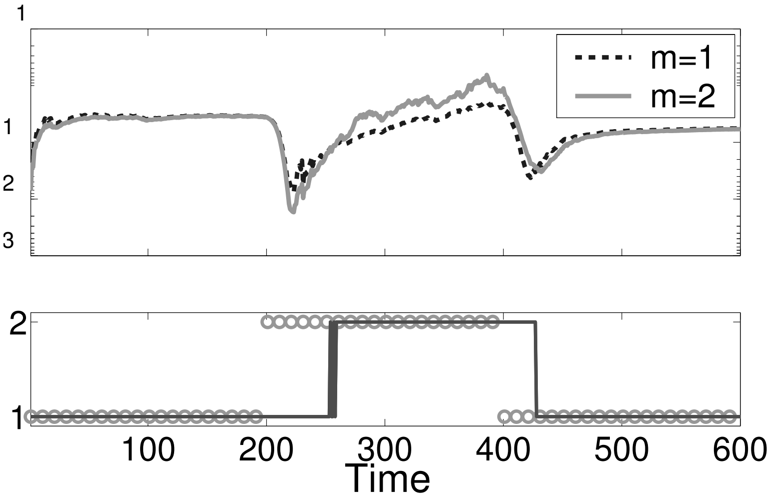

Figure 3.2. Normalized variational free energy $\tilde{F}_m^{\lambda}(t)$ (upper panel, in a log scale) and the MAP estimator m^* (lower panel) in each time step. A mark '$\circ$' and the solid line in the lower panel denote the actual principal component dimensionality and the MAP estimator, respectively.

time elapsed. When the forgetting factor was regulated by our method, on the other hand, the inference exhibited high performance. Namely, the estimator could alter its value just after the environment change, while it was improved in a stationary environment as the number of observed data increased. The figure also shows the modulation of $\lambda(t)$ and $\eta(t)$, which makes the learning flexible as described above.

Next, we assumed a situation in which the actual principal component dimensionality m was unknown. The dimensionality was usually fixed but occasionally changed such that $m = 1$ for $t = 1, \ldots, 200$ and $t = 401, \ldots, 600$, and $m = 2$ for $t = 201, \ldots, 400$. The actual parameter at $t = 201, \ldots, 400$ was set at $\boldsymbol{\Theta} = \begin{pmatrix} 1 & -5 & -10 \\ 5 & 1 & 10 \end{pmatrix}$ and those at the other time steps were the same as above. Figure 3.8 shows the normalized online variational free energy $\tilde{F}_m^{\lambda}(t)$ for the models with $m = 1$ and $m = 2$, and the MAP estimator m^*. Note that $\tilde{F}_m^{\lambda}(t)$ is shown on a log scale here. The figure indicates that the normalized online variational free energy $\tilde{F}_m^{\lambda}(t)$ is a suitable criterion for model selection in

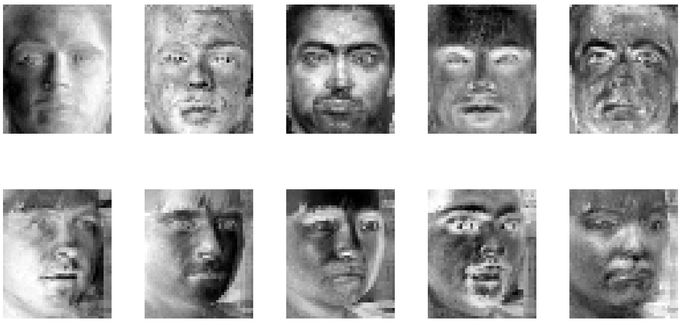

Figure 3.3. The first five eigenfaces (first to fifth from left to right) extracted by the standard PCA from the frontal (upper row) and half-profile (lower row) face images.

this case, and hence the principal component dimensionality can be estimated appropriately in an online fashion.

Real data: face images

Assuming that a representational system is used for our recognition of images, our approach was evaluated also by using a data set of realistic face images. The data set used here consists of 100 gray-scale photographs of frontal faces and 100 of half-profile faces, registered in Yale Face Database B [39]. The subjects in this data set were six males and one female in various lighting conditions. We standardized the images such that all the images contained 49×41 pixels, and the centers of eyes for frontal views and the centers of faces for half-profile views took the same coordinate. The pixel values were normalized to be within $[0, 1]$, thus each image was represented as a 2,009-dimensional vector of normalized pixel values. Basis vectors (principal components) extracted from a set of face images using PCA were called "eigenfaces" [93]. Figure 3.3 shows the first five eigenfaces extracted from the frontal and half-profile face images, and Figure 3.4 shows the largest 20 eigenvalues in the two subsets.

The learning process was divided into two phases. In the first phase, 100 observations were randomly selected from the frontal faces and sequentially provided to the learning model. This phase is called the "frontal condition." In the next phase, called the "half-profile condition," 100 observations were selected

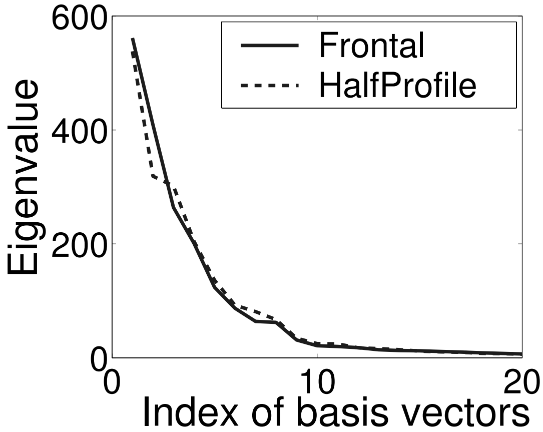

Figure 3.4. The eigenvalues of the frontal and half-profile face images, which were obtained by the standard PCA. The largest 20 (out of 2,009) are shown in descending order. The horizontal axis denotes indices for basis vectors.

from half-profile faces. The known constants were set as: $\sigma_x^{-2} = 60, \sigma_\zeta^{-2} = 5, r = 0.001, \alpha = 0.02$, and $\gamma = 0.001$. The scheduling of the forgetting factor described in Section 3.2 was used with or without the RP, where $\phi = 0.05$ and $\nu = 30$.

First, our scheduling scheme of $\lambda(t)$ and the effect of the RP were examined. The principal component dimensionality m was fixed at 14. Figure 3.5 shows the obtained first eigenface with the largest norm during the online learning process with the RP. In the latter half of the frontal condition, at $t = 60$ and 90, the eigenface successfully captured the features of frontal faces. The eigenface was then modified quickly into that of half-profile faces at $t = 130, 150$ and 180, through a transient phase like at $t = 110$. The figure also shows the learning processes with or without the RP, which are evaluated in comparison with the result by usual PCA. The eigenface obtained by our online learning without the RP did not approach that by the usual PCA; in contrast, that with the RP behaved well as the time elapsed within both the frontal and half-profile conditions. The time courses of $\eta(t)$ and $\lambda(t)$ are also shown in this figure. In the case with the RP, $\eta(t)$ and $\lambda(t)$ shift in time to properly adapt to the condition change.

Next, we compared the models with different m, such as $m = 5, \ldots, 30$, where

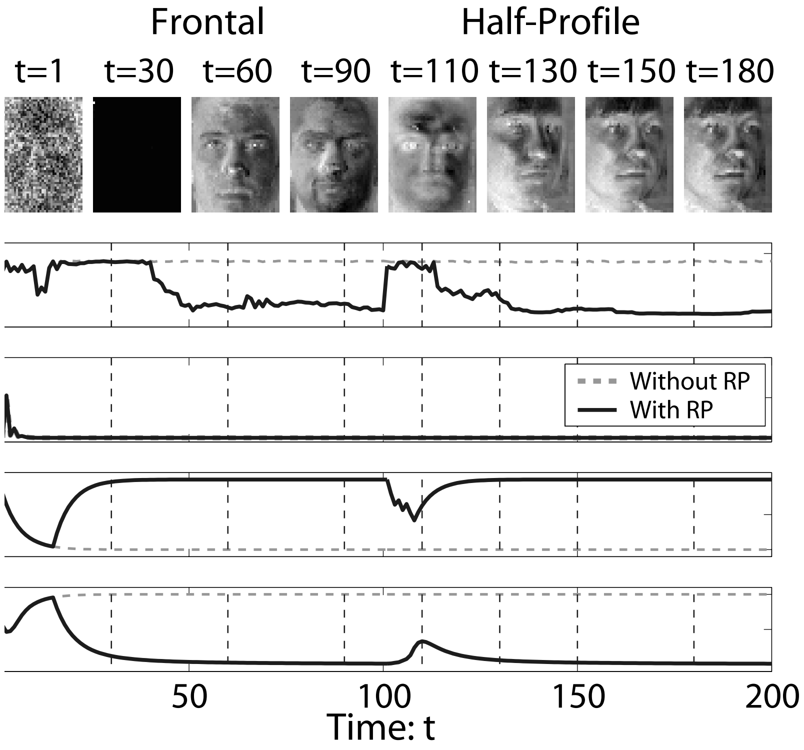

Figure 3.5. Top row: the first eigenface obtained during the online learning process. Appropriate representation was acquired under the frontal and subsequent half-profile conditions. The other four rows, from the second to the bottom: the angle between the first basis vector (i.e., with the largest norm) obtained in our learning and that by the standard PCA, the distance between the estimated mean vector and the true mean, $\lambda(t)$ and $\eta(t)$, where each dash or solid line denotes the case without the refractory period (RP) or with the RP, respectively. A set of dashed vertical lines in each panel denotes the time steps at which the first eigenfaces on the top row are displayed.

the scheduling by Eq. (3.10) was applied with the RP and the same constant parameters as used in the previous simulation. Figure 3.6 shows the normalized online variational free energy $\tilde{F}_m^\lambda[q]$ and the MAP estimator m^* for various m values in our online learning process. After convergence in the frontal condition, m^* was estimated as 9 and also as 9 in the subsequent half-profile condition. These results were consistent with that by the usual PCA; Fig. 3.4 implied that only about 10 bases were significant out of 2,009 bases under both conditions.

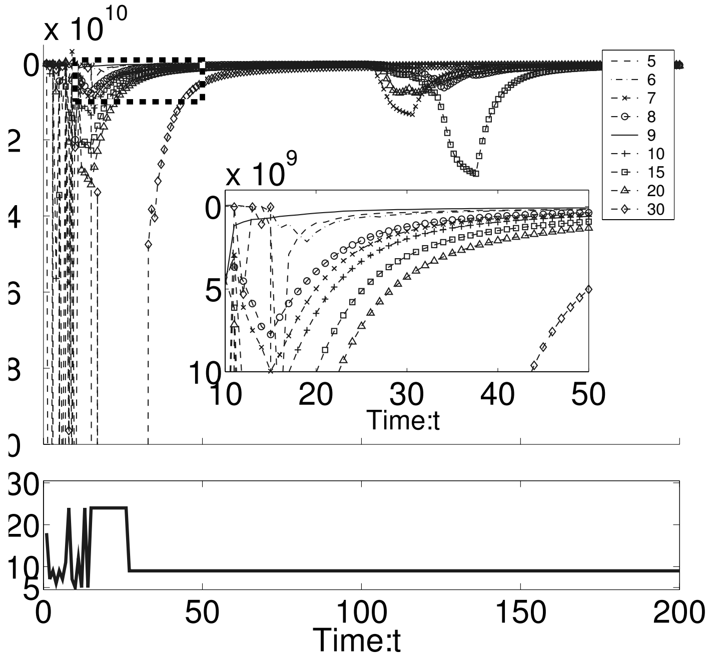

Figure 3.6. Normalized online variational free energy $\tilde{F}_m^\lambda[q](t)$ and the MAP estimator of principal component dimensionality m^*. The inset in the upper panel is a magnified image of the dashed rectangle. m^* was estimated as 9 after about $t = 25$, which was consistent with the result of the standard PCA in Fig. 3.4 that implied only about ten bases were significant out of 2,009.

4. Forgetting factor adaptation based on hierarchical Bayes

4.1 Online VB is a special type of incremental Bayes

Instead of directly calculating the VB-M step equation (3.4b), here we show that the online VB can be regarded as a special type of incremental Bayes, that is, the posterior belief of model parameter at time step t, $q_\theta^{(t)}$, can be recursively calculated according to a new datum based on the previous belief $q_\theta^{(t-1)}$. For notational simplicity, let $\psi_\tau(\boldsymbol{x}_\tau, \boldsymbol{\Theta}) \equiv E_{\boldsymbol{y}_\tau}\left[\log p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta})\right]$ and $\Psi_t(X_{1:t}, \boldsymbol{\Theta}) \equiv \sum_{\tau=1}^{t}\left(\prod_{s=\tau+1}^{t}\lambda\left(s\right)\right)\psi_\tau(\boldsymbol{x}_\tau, \boldsymbol{\Theta})$. Eq. (3.4b) is then written as

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\bigl(\Psi_t(X_{1:t}, \boldsymbol{\Theta})\bigr)p_0(\boldsymbol{\Theta})}{\int d\boldsymbol{\Theta}\exp\bigl(\Psi_t(X_{1:t}, \boldsymbol{\Theta})\bigr)p_0(\boldsymbol{\Theta})}. \tag{3.13}$$

Superscript $^{(\tau)}$ denotes that trial distribution is maximized using observations available at time τ, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\tau$. Also, the trial distribution at previous time step $t-1$ is given by

$$q_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}) = \frac{\exp\bigl(\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta})\bigr)p_0(\boldsymbol{\Theta})}{\int d\boldsymbol{\Theta}\exp\bigl(\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta})\bigr)p_0(\boldsymbol{\Theta})}. \tag{3.14}$$

Note especially that each of the old quantities, $\{\Psi_\tau(X_{1:\tau}, \boldsymbol{\Theta}) \mid \tau = 1, 2, \ldots, t-1\}$, do not changes through the new VB-EM step at time step t, even though it depends on $q_\tau(\boldsymbol{y}_\tau)$ which is also a target quantity of free energy maximization: As described in Sec. 2.2, the online VB-E step performed between the two maximization steps, Eq. (3.14) at time $t-1$ and Eq. (3.13) at time t, is temporally localized so that it does not change the past inference of the latent variable and the forgetting factor. Only the new inference $q_t(\boldsymbol{y}_t)$ is calculated in VB-E step at time step t, while the previous trial distributions $q_\tau(\boldsymbol{y}_\tau)\,(\tau = 1, \ldots, t-1)$ and forgetting factors $\lambda(s)\,(s = 1, \ldots, t-1)$ are thus fixed at time t; and then $\Psi_{t-1}\left(X_{1:t-1}, \boldsymbol{\Theta}\right)$, which includes the expectations with respect to $q_\tau(\boldsymbol{y}_\tau)\,(\tau = 1, \ldots, t-1)$, does not change.

The quantity Ψ_t has the following relation:

$$\Psi_t(X_{1:t}, \boldsymbol{\Theta}) = \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda\left(s\right) \right) \psi_\tau(\boldsymbol{x}_\tau, \boldsymbol{\Theta}) \tag{3.15a}$$

$$= \psi_t\left(\boldsymbol{x}_t, \boldsymbol{\Theta}\right) + \lambda(t) \sum_{\tau=1}^{t-1} \left(\prod_{s=\tau+1}^{t-1} \lambda\left(s\right) \right) \psi_\tau\left(\boldsymbol{x}_\tau, \boldsymbol{\Theta}\right) \tag{3.15b}$$

$$= \psi_t\left(\boldsymbol{x}_t, \boldsymbol{\Theta}\right) + \lambda(t)\Psi_{t-1}\left(X_{1:t-1}, \boldsymbol{\Theta}\right) \tag{3.15c}$$

By substituting this into Eq. (3.13), we obtain

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\left(\psi_t\left(\boldsymbol{x}_t, \boldsymbol{\Theta}\right)\right) \exp(\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta}))^{\lambda(t)} p_0\left(\boldsymbol{\Theta}\right)}{\int d\boldsymbol{\Theta} \exp\left(\psi_t\left(\boldsymbol{x}_t, \boldsymbol{\Theta}\right)\right) \exp(\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta}))^{\lambda(t)} p_0\left(\boldsymbol{\Theta}\right)} \tag{3.16a}$$

$$= \frac{\exp\left(\psi_t\left(\boldsymbol{x}_t, \boldsymbol{\Theta}\right)\right) q_\theta^{(t-1)}\left(\boldsymbol{\Theta} \mid X_{1:t-1}\right)^{\lambda(t)} p_0\left(\boldsymbol{\Theta}\right)^{1-\lambda(t)}}{\int d\boldsymbol{\Theta} \exp\left(\psi_t\left(\boldsymbol{x}_t, \boldsymbol{\Theta}\right)\right) q_\theta^{(t-1)}\left(\boldsymbol{\Theta} \mid X_{1:t-1}\right)^{\lambda(t)} p_0\left(\boldsymbol{\Theta}\right)^{1-\lambda(t)}}. \tag{3.16b}$$

where we used the fact that the denominator of Eq. (3.14) does not depend on $\boldsymbol{\Theta}$. Thus, the VB-M step can be achieved in an incremental manner:

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp(E_{\boldsymbol{y}_t}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta})\right]) \tilde{q}_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}; \lambda(t))}{\int d\boldsymbol{\Theta} \exp(E_{\boldsymbol{y}_t}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta})\right]) \tilde{q}_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}; \lambda(t))}, \tag{3.17}$$

where a modified trial distribution $\tilde{q}_\theta$ is defined as

$$\tilde{q}_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}; \lambda(t)) \propto q_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1})^{\lambda(t)} p_0(\boldsymbol{\Theta})^{1-\lambda(t)}, \tag{3.18}$$

where the normalization term is omitted. Eq. (3.17) suggests that the online VB method is equivalent to the incremental Bayesian inference with a special setting of the prior. This is a new theoretical result of this study. After a new observation $\boldsymbol{x}_t$ is given at time t, Eq. (3.17) incrementally updates the previous posterior belief $q_\theta^{(t-1)}$ into the new posterior $q_\theta^{(t)}$ using the Bayes rule, similarly to conventional incremental Bayesian updates; $\tilde{q}_\theta^{(t-1)}$ is regarded as an improved prior belief based on currently available observations $X_{1:t-1}$ at time $t-1$, starting from the initial prior belief p_0. The differences between the incremental update of Eq. (3.17) and the conventional one are: in Eq. (3.17), log-likelihood term $\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta})$ is replaced by its expectation with respect to latent variable $\boldsymbol{y}_t$, and a forgetting factor is introduced to attenuate previous belief $q_\theta^{(t-1)}$ and to partially restore initial prior belief p_0.

4.2 Recursive learning rule for PPCA

In this section, the observations are assumed to be normalized to have a zero mean in a preprocessing stage. This is just for simplicity, and can be done without loss of generality. The expected mean vector $\boldsymbol{\mu}$ in the generative model of Eq. (3.1) is then set to zero vector, so that only the basis matrix $\boldsymbol{W}$ is the model parameter to be learned. In this section, we use a conjugate prior for $\boldsymbol{W}$:

$$p_0(\boldsymbol{W}) = \mathrm{N}_{n\times m}\left(\boldsymbol{W} \mid \boldsymbol{M}_0, \boldsymbol{I}_n, \boldsymbol{G}_0^{-1}\right), \tag{3.19}$$

where $\mathrm{N}_{n\times m}(\cdot \mid \cdot, \cdot, \cdot)$ is the matrix normal distribution (see Appendix A).

The recursive update equation of Eq. (3.17) for PPCA then results in only two updatings of hyperparameters, since the trial distribution of $\boldsymbol{W}$ is obtained as a Gaussian. Now let $q_\theta^{(t)}(\boldsymbol{W} \mid X_{1:t}) = \mathrm{N}_{n\times m}(\boldsymbol{W} \mid \hat{\boldsymbol{M}}_t, \boldsymbol{I}_n, \hat{\boldsymbol{G}}_t^{-1})$, and then the learning rule is derived as

$$\hat{\boldsymbol{G}}_t = \sigma_x^{-2}\left\langle \boldsymbol{y}_t\boldsymbol{y}_t^T\right\rangle + \lambda(t)\hat{\boldsymbol{G}}_{t-1} + (1-\lambda(t))\boldsymbol{G}_0, \tag{3.20a}$$

$$\begin{aligned}\hat{\boldsymbol{M}}_t = \hat{\boldsymbol{M}}_{t-1} + \Big\{&\sigma_x^{-2}\boldsymbol{x}_t\left\langle \boldsymbol{y}_t\right\rangle^T + (1-\lambda(t))\,\boldsymbol{M}_0\boldsymbol{G}_0 \\ &- \hat{\boldsymbol{M}}_{t-1}\left(\sigma_x^{-2}\left\langle \boldsymbol{y}_t\boldsymbol{y}_t^T\right\rangle + (1-\lambda(t))\boldsymbol{G}_0\right)\Big\}\,\hat{\boldsymbol{G}}_t^{-1},\end{aligned} \tag{3.20b}$$

where $\langle\cdot\rangle$ denotes expectation with respect to trial distribution q. Note that this learning rule directly updates the hyperparameters of the trial distribution, although they were indirectly updated through the online maintenance of the expected sufficient statistics in Sec. 3.

4.3 Hierarchical Bayes inference

In this section, we describe the hierarchical Bayesian method to schedule $\lambda(t)$, utilizing the above illustration of the online VB learning. According to Eq. (3.17), $\lambda(t)$ can be regarded as a hyperparameter of conditional prior $\tilde{q}_\theta^{(t-1)}$ in the incremental updates of trial distribution q_θ. Although $\lambda(t)$ is not a model parameter, one can still perform an inference on $\lambda(t)$ by seeing it as an unknown hyperparameter. Let $L(\boldsymbol{x}_t, \lambda(t))$ be the denominator of Eq. (3.17), and then $L(\boldsymbol{x}_t, \lambda(t))$ corresponds to the marginal likelihood of $\lambda(t)$ given a new observation $\boldsymbol{x}_t$. If prior $p_0(\boldsymbol{\Theta})$ is noninformative, Eq. (3.18) infers the following: when new observation

$\boldsymbol{x}_t$ cannot be explained well under current belief $q_\theta^{(t-1)}$, the marginal likelihood $L(\boldsymbol{x}_t, \lambda(t))$ becomes large for a case that $\lambda(t) \approx 0$, and hence a noninformative prior is used; on the contrary, when $\boldsymbol{x}_t$ can be explained well under $q_\theta^{(t-1)}$, $L(\boldsymbol{x}_t, \lambda(t))$ becomes large for a case that $\lambda(t) \approx 1$, and hence the current belief is used. Then, if the scheduling of $\lambda(t)$ is performed to enlarge the marginal likelihood, it is expected that $\lambda(t)$ becomes low when the environment changes, while it stays high during stationary periods. According to the hierarchical Bayesian inference, therefore, the posterior distribution of $\lambda(t)$ is obtained as

$$p(\lambda(t) \mid \boldsymbol{x}_t) = \frac{L(\boldsymbol{x}_t, \lambda(t))p(\lambda(t))}{\int_0^1 d\lambda(t)\, L(\boldsymbol{x}_t, \lambda(t))p(\lambda(t))}, \tag{3.21}$$

where $p(\lambda(t))$ is a prior distribution of $\lambda(t)$. With this posterior, the actual value of $\lambda(t)$ is estimated as its expectation:

$$\hat{\lambda}(t) = \int_0^1 d\lambda(t)\, p(\lambda(t) \mid \boldsymbol{x}_t)\lambda(t). \tag{3.22}$$

Practically, however, it is not so easy to calculate the integrals that appeared in Eqs. (3.21) and (3.22). In addition, the evaluation of marginal likelihood $L(\boldsymbol{x}_t, \lambda(t))$ also involves intractable integral in calculating the normalization constant of Eq. (3.18) except for special cases with $\lambda(t) = 0$ or 1. In this study, instead of addressing the integrals over the entire range of $\lambda(t) \in [0, 1]$, we evaluate them only at the endpoints, $\lambda(t) = 0$ and 1. The estimator of $\lambda(t)$ is thus obtained by

$$\hat{\lambda}(t) = \frac{\int_0^1 d\lambda(t) L(\boldsymbol{x}_t, \lambda(t))p(\lambda(t))\lambda(t)}{\int_0^1 d\lambda(t) L(\boldsymbol{x}_t, \lambda(t))p(\lambda(t))} \approx \frac{\sum_{\lambda(t)\in\{0,1\}} L(\boldsymbol{x}_t, \lambda(t))p(\lambda(t))\lambda(t)}{\sum_{\lambda(t)\in\{0,1\}} L(\boldsymbol{x}_t, \lambda(t))p(\lambda(t))}. \tag{3.23}$$

4.4 Simulations

Two-dimensional synthesised data

The basic features of our approach were examined by using synthesized data. A two-dimensional vector $\boldsymbol{x}_t$ was generated according to Eq. (3.1) with $m = 1$ and $\sigma_x = 1$. The number of observations was $T = 600$. True parameter $\boldsymbol{W}$ was fixed in a short time period but occasionally changed as follows: $\boldsymbol{W} = (5, -1)^T$ for

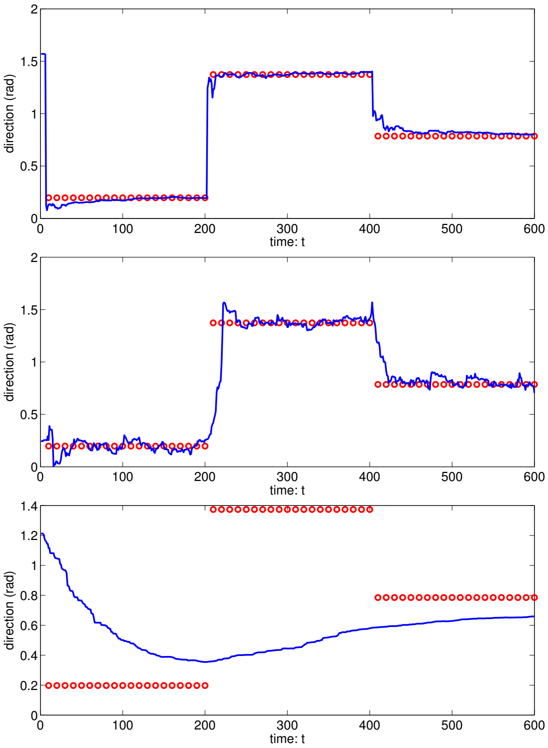

Figure 3.7. Direction of estimated principal component vector in a single trial. Horizontal axis denotes time step t. Panels show the estimator in three cases: 1) $\lambda(t)$ is controlled by our new scheduling scheme; 2) $\lambda(t) = 0.9$; and 3) $\lambda(t) = 1$. Only the direction of the principal component vector, the angle from the x_1-axis, is shown. '∘' represents the real value in each time step.

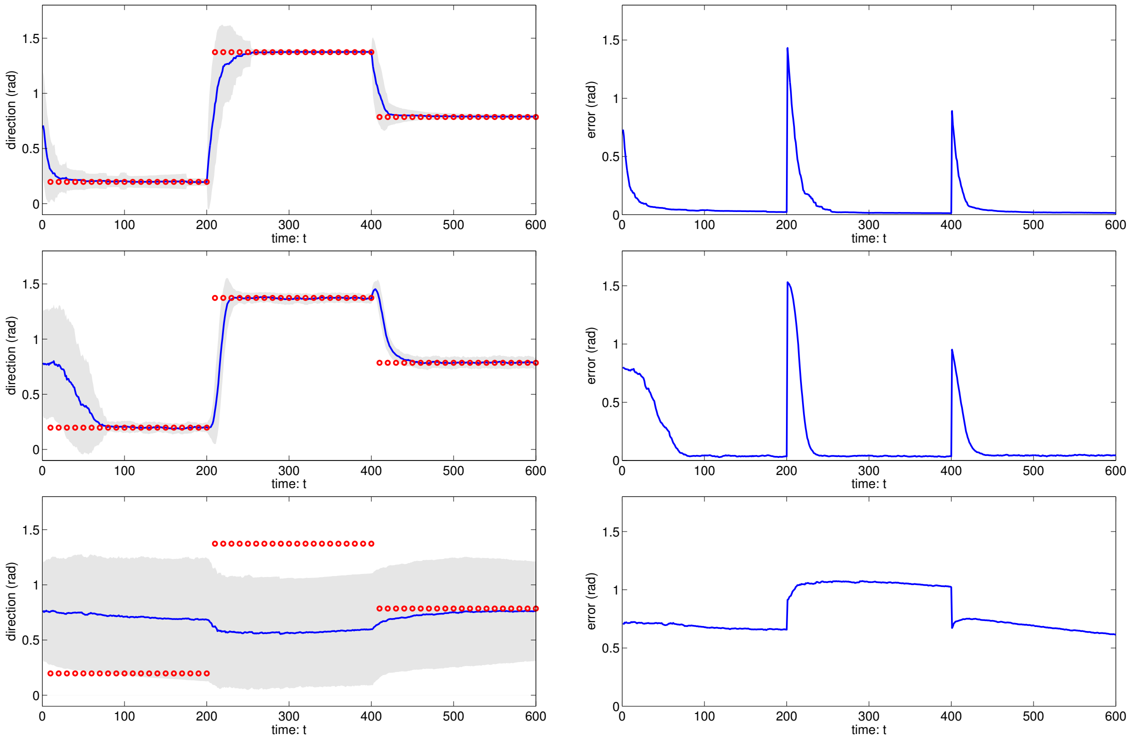

Figure 3.8. Direction of estimated principal component vector (left column) and estimation error, i.e., the angle between estimated vector and the true one (right column). I performed 100 runs by individually preparing 100 different datasets to learn. In the right column, the solid line denotes average over 100 runs, and the dark shade represents errorbar (standard deviation).

$t = 1, \ldots, 200$, $(1, 5)^T$ for $t = 201, \ldots, 400$, and $(-3, 3)^T$ for $t = 401, \ldots, 600$. Prior hyperparameters $\boldsymbol{M}_0$ and $\boldsymbol{G}_0$ were set as $\boldsymbol{M}_0 = \mathbf{0}$ and $\boldsymbol{G}_0 = 1 \times 10^{-3}\boldsymbol{I}_n$, so that the prior became nearly noninformative. The initial hyperparameters of the trial distribution, $\hat{\boldsymbol{M}}_0$ and $\hat{\boldsymbol{G}}_0$, were randomly set.

Figure 3.7 shows learning processes in the following three conditions: 1) our new approach; 2) forgetting factor fixed at $\lambda(t) = 0.9$ for any t; and 3) fixed at $\lambda(t) = 1$ for any t. The direction of the estimated principal component vector is shown in this figure. Here, the estimator of $\boldsymbol{W}$ was given as its expectation, $\hat{\boldsymbol{M}}$. Using our hierarchical Bayesian scheduling of $\lambda(t)$, the inference exhibited high performance compared to the other two conditions. Namely, the estimator could

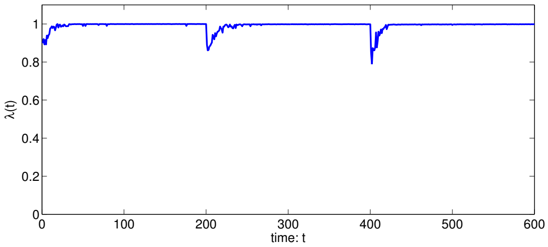

Figure 3.9. Forgetting factor $\lambda(t)$ averaged over same 100 runs as in Figure 3.8.

alter its value rapidly after environmental changes, while it was improved in a stationary period as the number of observed data increased. In cases that the forgetting factor $\lambda(s)$ was set at a constant 1 for all $s = 1, \ldots, T$ (Condition 3), the estimator gradually approached the target value during stationary periods, but the approach speed was too slow. In contrast, in cases that $\lambda(s)$ was set at a smaller constant of 0.9 for all $s = 1, \ldots, T$ (Condition 2), the estimator could alter its value in response to environmental changes, but a high variance remained. Because of this variance, the estimator could not be improved even when time elapsed in a stationary period.

Next, to see the stability of our new online VB learning, the simulation was repeated for 100 runs. The observed dataset for each run was generated by Eq. (3.1) individually with a random seed number. Figure 3.8 shows the learning process (left column) and the estimation error (right column) averaged over 100 runs for each condition. Although the variance of estimator by our approach was relatively large at the beginning of each stationary period, compared to the case of $\lambda(t) = 0.9$, it grew smaller as the stationary period continued. Estimation error also decreased to zero in our approach, while a small bias remained in the case of $\lambda(t) = 0.9$. In the case of $\lambda(t) = 1$, the variance of initial values remained throughout the learning process. Figure 3.9 shows the value of the forgetting factor averaged over 100 runs scheduled by our hierarchical Bayesian scheme.

We also examined a situation where the model parameter gradually changes. In this experiment, $m = 1$, $T = 1,000$, $\sigma_x = 1$, and true parameter $\boldsymbol{W}$ was given as follows: $\boldsymbol{W}$ was fixed at $(5, 0)^T$ in the initial phase with $t = 1, \ldots, 250$, and then gradually changes from $(5, 0)^T$ at $t = 251$ to $(0, 5)^T$ at $t = 500$ (by $\pi/500$ radian

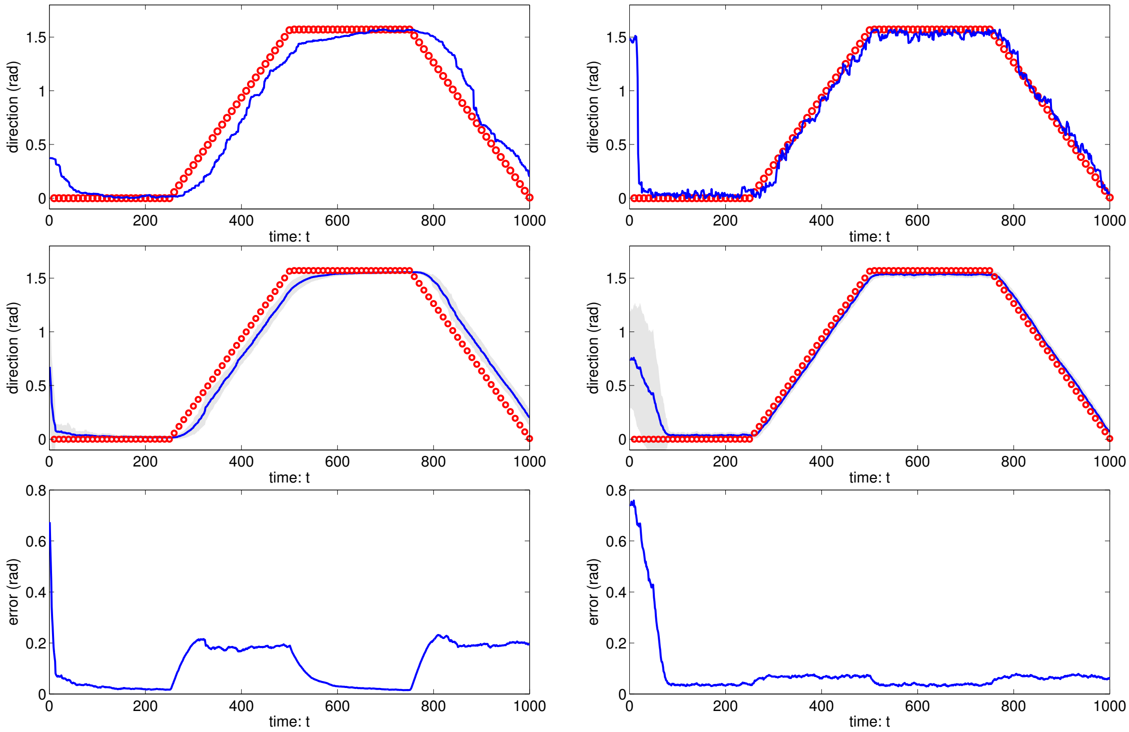

Figure 3.10. Direction of estimated principal component vectors and estimation errors in the case of gradually-changing parameters. Left and right columns respectively show the results of proposed method and the case with $\lambda(t) = 0.9$ (fixed). Top row: Typical examples of single runs. Middle row: The results of 100 runs, where the solid line denotes the average values, and the dark shade represents errorbars (standard deviation). Bottom row: Estimation error, i.e., the angle between estimated vector and the true one.

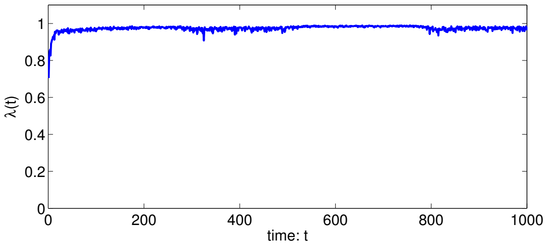

Figure 3.11. Forgetting factor $\lambda(t)$ averaged over the 100 runs by proposed method in the case of gradually-changing parameter.

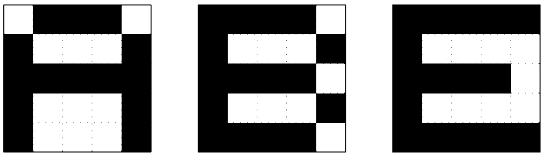

Figure 3.12. Original binary images corresponding to alphabetic characters, 'A', 'B', and 'E', from left to right. These images were used as principal component vectors in the generative model, Eq. (3.1), to generate artificial datasets.

for every time step). It was then fixed again at $(0,5)^T$ for $t = 501, \ldots, 750$ and finally gradually changes from $(0,5)^T$ at $t = 751$ to $(5,0)^T$ at $t = 1,000$. Prior hyperparameters were set as the same as above. We compared the two cases, our new approach and the case of $\lambda(t) = 0.9$, by running each of them for 100 times with different initial hyperparameters of the trial distribution. Figure 3.10 shows typical example of single run (top row), learning process averaged over 100 runs (middle row) and the average estimation error (bottom row). This result shows that the estimates of $\boldsymbol{W}$ by our proposed method is more accurate in the stationary periods than those by the case of fixed $\lambda(t)$, but is more biased in the gradually changing periods. The biased estimates was probably due to the approximation, Eq. (3.23), and thus it would be improved by using a more careful approximation of it. Figure 3.11 shows the forgetting factor averaged over 100 runs scheduled by our hierarchical Bayesian scheme.

Artificially-generated alphabetic characters

Our approach was further evaluated by using a dataset of artificially generated alphabetic characters, consisting of 600 grayscale images of 5×5 pixels. Each image had a feature of 1) 'A', 2) 'B', or 3) 'E'. We used the three binary original images shown in Figure 3.12 as principal component vectors ($n = 25$), and generated 200 observations for each original image according to Eq. (3.1) with $\sigma_x = 0.2$. A learning process consisted of three stages, each of which corresponded to one of the three features of data; 200 data points for each 'A', 'B', and 'E' were provided sequentially through the three stages. Example observations in the learning, those of time steps $1, 51, 101, \ldots, 551$, are presented in Figure 3.13. In this simulation, prior hyperparameters were set as $\boldsymbol{M}_0 = \mathbf{0}$ and $\boldsymbol{G}_0 = 1 \times 10^{-8}\boldsymbol{I}_n$, and so the prior was almost noninformative. $\hat{\boldsymbol{M}}_0$ and $\hat{\boldsymbol{G}}_0$ were set randomly.

Figure 3.14 shows five typical learning processes out of 100 runs; in each the first principal components of time steps $1, 51, 101, \ldots, 551$ are presented. In this figure, time steps 201 and 401 correspond to the changepoints from 'A' to 'B' and 'B' to 'E', respectively. The reversion of black and white occurred in some runs because the signs of principal component vectors were irrelevant to feature extraction. This result shows that the model learned appropriate basis in stationary periods, while it could quickly change the basis to assimilate a new feature when novel inputs were provided. Figure 3.15 shows estimation error (top panel) and the forgetting factor (bottom panel) averaged over 100 runs.

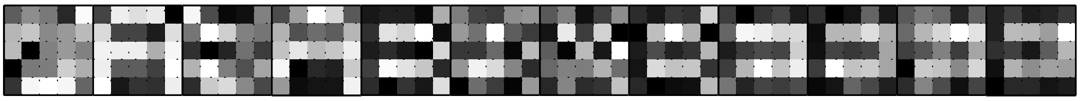

Figure 3.13. Example observations in learning from time steps $1, 51, 101, \ldots, 551$ from right to left.

5. Discussion

In this chapter, we proposed two balancing scheme between plasticity and stability of online VB learning to realize online representation learning, or more specifically the feature extraction, in dynamic environments. A key to these scheme is the dynamic scheduling of the forgetting factor $\lambda(t)$. We proposed 1) a novelty-based scheduling with introducing a method of probabilistic novelty detection based on a mixture model, and 2) a hierarchical Bayesian way to adapt the forgetting factor based on an interpretation of online varational Bayes as a special type of incremental Bayes, which is new theoretical contribution of this study. The simulation results showed that the proposed learning models were able to quickly and robustly follow the abrupt changes of input statistics to be accommodated to the new inputs, together with the model parameters being improved in stationary periods. While both of them have been shown to be effective for online feature

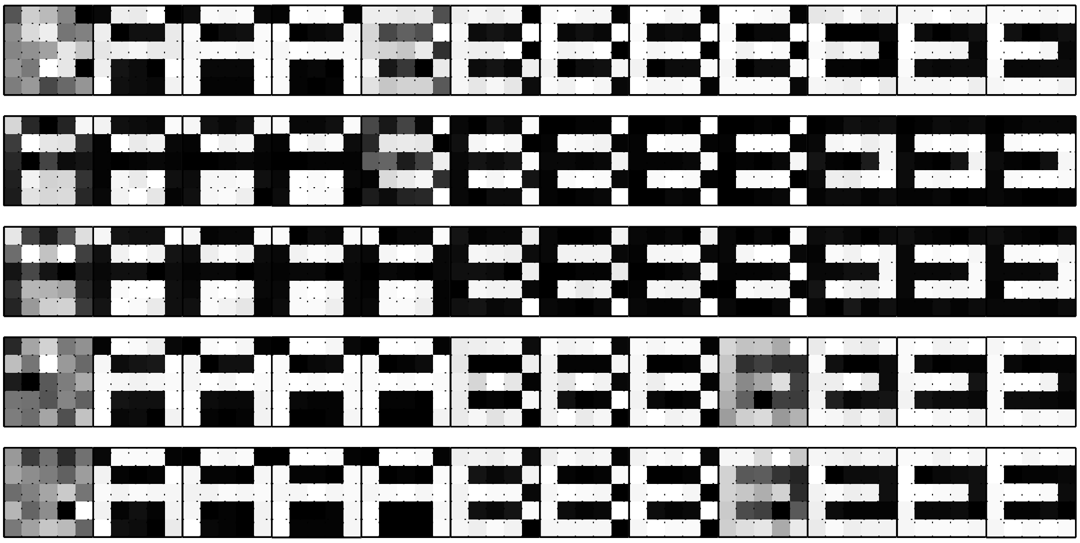

Figure 3.14. Five typical learning processes out of 100 runs, in each of which the first principal components of time steps $1, 51, 101, \ldots, 551$ are presented.

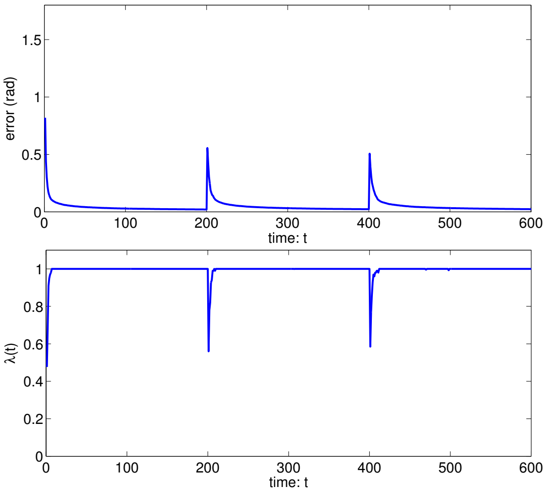

Figure 3.15. Estimation error (top panel) and forgetting factor (bottom panel) averaged over 100 runs.

extraction in changing environments, the second one, the hierarchical Bayesian scheme, is more sophisticated, since it do not require additional mechanism like outlier model explicitly as in the first one, where the novelty detection and the novelty-based scheduling of $\lambda(t)$ is naturally in a principled manner. Although the exact calculation of the hierarchical Bayesian estimates of $\lambda(t)$ is often intractable, a simple approximation was shown to work well in the simulations, at least in the case of abrupt environmental changes. The proposed scheme also works when the environement gradually changes, but the estimates in such periods were shown to be more biased than those by using fixed $\lambda(t)$. This is probably due to the rather crude approximation, Eq. (3.23), which implicitly ruled out the possibility of intermediate belief ($\lambda(t) \in (0,1)$). The results would be improved by using a more accurate approximation of the hierarchical Bayes estimates.

The illustration of the online VB method as an incremental Bayesian inference, Eq. (3.17), is not limited to the case of PPCA. The incremental Bayes update in Eq. (3.17) can be employed for many kinds of models with latent variables, as long as a further factorization of trial distribution of model parameters is not necessary. The simple learning rule we derived for learning PPCA, Eq. (3.20), can also be applied to other linear latent variable models with isotropic Gaussian noise. Extension to allow a general covariance matrix in Gaussian noise is also straightforward. Such models include generative models of independent component analysis (ICA) [49, 7, 60] and sparse coding [72, 73]. In a future study, the learning rule, Eq. (3.20), should be refined to deal with unknown noise variance (or covariance matrix). When further factorization on the model parameters is assumed, that is, more than two trial distributions for distinct subsets of model parameters have to be updated in the online VB-M step, however, Eq. (3.17) cannot be directly applied. In such a case, the mutual dependence of the parameter trial distributions would be an obstacle to individual updates of consistent trial distributions. This problem can be resolved by introducing some additional terms to eliminate mutual dependence from the learning rule; such an investigation remains as our future study.

6. Biological implications

Since animals confront a flood of high-dimensional sensory inputs provided by the environments surrounding them, it is crucial for animals' brains to appropriately transform external information into internal representations. Although those environments can be regarded as static for a short time, they are dynamic over a long time period. To adapt to such a dynamic environment, a brain needs to detect an environmental change and to quickly learn internal representations necessary in a new environment. In this section, we briefly discuss cortical representation learning in dynamic environments and on the functional role of acetylcholine (ACh), which is a neuromodulatory chemical (for reviews on computational neuromodulation, see [44, 34, 31]), in relation to the proposed online learning models.

Cortical representations are probably mediated by neuronal populations. Updating a cortical representation is then likely to require a regulation system that broadly affects the population of related neurons. Since local synaptic modulation like the one in the Hebbian learning is not proper by itself, it is natural to consider that the diffusive effects of neuromodulatory chemicals are related to the regulation system. Hasselmo [44] presented the following computational perspective on the functional role of ACh, based on physiological facts: 1) a high ACh level within a local circuit leads to a predominant influence of external stimuli, which induces learning of new memories or representations; and 2) a low ACh level, in contrast, leads to a predominant influence of local intrinsic activities, which corresponds to recalling of previously-learned information. Moreover, experimental studies have reported that a high ACh level facilitates the plasticity of receptive field and reorganization of representational maps [11, 55, 33, 15, 76]. These works provided evidence that the learning of cortical representational system is regulated by ACh.

The simulations in this chapter showed that our scheduling scheme of the forgetting factor enabled the online VB learning to alter its dynamics in order to re-learn new bases in novel environments. We can interpret the meta-parameters involved in our online leanring model, with the analogy from the function of ACh, as follows: a small $\lambda(t)$ leading to a large $\eta(t)$ induces the influence of current information and facilitates the learning, by suppressing the influence of previously-learned information; in contrast, a large $\lambda(t)$ leading to a small $\eta(t)$ encourages the predominant influence of previously-learned information when updating the expected sufficient statistics. This view suggests the possibility that the functional role of ACh is an inverse effect of the forgetting factor or the effect of learning rate in our online learning, and then ACh modulates the dynamics of representational learning. Several studies actually reported that cortical ACh levels tend to increase when facing novel stimuli or environments [1, 65, 41]. This fact also support the hypothetical view, since the increase of the learning rate in response to detected novelty is a fundamental property of our model.

Our model implicitly assumes that ACh is regulated individually in each local cortical area that corresponds to a local representational unit. Actually, it has been suggested that ACh levels can be regulated selectively within local cortical areas, while diffusive projections of the neuromodulatory system seemingly result in a regulation over a wide range of areas [45, 82, 101]. A major group of cholinergic projection neurons in the central nervous system exists in a subcortical area, the basal forebrain (BF), and the neurons project extensively to the cerebral cortex. According to the anatomical studies by Zaborszky et al. [101, 102], it is possible that the information processing by each local cortical area is regulated by ACh separately from the other areas by means of parallelly-organized feedback loops via the BF. This selective feedback regulation of the cortical ACh level supports our assumption because it can provide a possible implementation of local regulation of representational learning. Zaborszky et al. [102] also reported that the prefrontal cortex (PFC) is the major input source to the BF. The PFC is closely involved with novelty detection [80, 24, 29], which is important in detecting an environmental change.

A relate work have been done by Yu and Dayan [99], in which they suggested that higher-level contextual information may control lower-level ACh release. In their three-layer hidden Markov model, the posterior distribution of a top-level “contextual” hidden state was approximated by means of its MAP estimator and uncertainty, and the latter was supposed to be mediated by an ACh signal. The ACh signal became top-down information such to modulate the representation of the intermediate hidden state. In our model, the novelty information modulates the ACh level, which can be interpreted as a form of top-down information like in

[99] as well as the occurrence probability of an environmental change. However, our model is different from their model in that we primarily focus on the role of ACh in the learning of representational system, while Yu and Dayan [99] were not concerned about learning. Although our model is also related to Yu and Dayan's subsequent work [100] that addressed the issue of learning, their focus was still different to ours in the following way: First, Yu and Dayan's factor analysis model, in which the mean vector of a Gaussian hidden variable shifted in time, focused particularly on the learning of this mean vector. The learning of basis vectors, which was essentially important for obtaining appropriate internal representations, was not explicitly addressed. Our model, in contrast, focuses on the learning of a representational system with special interest in the learning of basis vectors. Second, they advocated that norepinephrine (NE) reports a novelty and drives the cortical ACh release, but our model intends to employ another possibility that the feedback connections from the PFC via the BF report a novelty and regulates local ACh levels as mentioned above. Last, we assumed no dynamics for hidden variables in our model, unlike those introduced in Yu and Dayan's model. In regard to this point, extension and application to more complex situations remain for our future study.

Hasselmo's hypothesis [44] was based essentially on a physiological fact that ACh suppresses intrinsic connections within a local neuronal population, while it has no suppressing effect on afferent connections. In our online VB learning, the ACh level, corresponding to the learning rate, regulates the dynamics of updating the expected sufficient statistics which are maintained explicitly. A computational role of intrinsic connections for the learning of representational system is implied then as to allow the past inference to affect the current learning, through the maintenance of the expected sufficient statistics. A similar idea was also employed in a theoretical study of the population coding [98], which suggested a possible computation of Bayesian-like incremental learning in the brain. On the other hand, there exists a possible extension of our model to incorporating another role of intrinsic connections. As in [26], correlation structures among hidden variables may be due to the intrinsic connections.

We presented a heuristic criterion, the normalized online variational free energy $\tilde{F}_m^\lambda$, for determining the model structure. Although there has so far been no evidence of this criterion in a real brain, we expect that the framework of hierarchical Bayesian inference can also help in our understanding of the biological model selection processes. Reorganization of cortical representational mappings probably involves something like a model selection process, for example, an increase or decrease in the number of associated neurons with specific sensory stimuli [55, 76]. Mercado et al. [53] proposed a computational model of reorganization of auditory maps using an SOM, in which the spatial extent of ACh diffusion regulated the number of associated neurons simultaneously with the learning of current input. This model indicated the possibility that ACh is involved in such a model selection process. Moreover, some types of model selection processes are likely to require global modulatory effects. NE, whose effects are wide-ranged compared to those of ACh [45, 101], can be involved in such processes, because NE is also related to cortical plasticity [33] as is ACh.

Chapter 4

Conclusion

1. Sumamry of this thesis

In this thesis, I proposed two kinds of learning models each of which was intended to address 1) BSS with non-stationary appearance of source signals, and 2) online feature extraction within changing environments, respectively. These two models were commonly based on probabilistic latent variable models, which have recently been a basic framework for understanding representation learning methods. Specifically, I employed the probabilistic (noisy) ICA and PCA as two basic models to be extended.

In the former part of this thesis, I proposed the Switching ICA, which extended the previous probabilistic formulation of noisy ICA with incorporating a dynamic variable selection by employing a special type of HMM/HSMM as the source models. The HMM-based model can be regarded as a special type of an existing HMM-ICA, with newly introducing an explicit model of inactive source. As shown in Chapter 2, this special setting is quite effective when source signals have non-stationary appearances, especially with some amount of observation noise. The method also worked well even in such difficult cases that the number of total sources is unknown and overestimated, or is larger than that of mixtures. In addition, an effective realization of semi-Markov Switching ICA is also an important contribution of this study. The estimation of switching variables were improved by the semi-Markov model, which would be useful for further recognition tasks. It is the first time, at least to my knowleges, to introduce a semi-Markov dynamics into a non-stationary ICA context.

In the latter part of this thesis, I proposed two online Bayesian learning models with specific application to the probabilistic PCA, both of which extended the standard online variational Bayesian learning by means of adaptive regulation of forgetting factors. One is based on probabilistic novelty detection using mixture model, and the other employs a hierarchical Bayesian inference of forgetting factors, based on a new interpretation of the online VB as a special type of incremental Bayes updating. They provides a principled basis of online feature extraction within a difficult non-stationary situations. In the simulation experiments in Chapter 3, each of these methods have been shown to be successful in online feature extraction tasks. I also discussed its connection to the representation learning systems in brain.

2. Future application and open issues

While the proposed methods have so far been demonstrated only in artificial situations, their high performances presented in this thesis indicate their future availabilies in real problems. First, in the broad range of applications that standard ICA methods have applied, Switching ICA would be a promising approach to handle non-stationary cases where the performance of standard ICA degrades, since the particular situation, such that each source abruptly switches on and off, can occur universally in real-world BSS problems. As focused on in Chapter 2, blind separation of audio sources would be a successful application of Switching ICA. In other fields, for example, biomedical signals such as electroencephologram (EEG) data likely have such a non-stationary charecter. In the case of EEG, cortical signals from large brain areas are mixed into resultant data, where ICA have been successfully applied to perform BSS to separate the original signals each of which arise from local neuronal activations [62]. As the neuronal activations can be naturally considered as having much relation to some sort of external/internal events, the induced signals are likely to be dynamically switching on and off according to the occurrence of corresponding events. This speculation indicates the potential availabilities of Switching ICA to improve the analysis of EEG signals. Furthermore, signal separation from such biomedical measurements by using

ICA is an appealing technique for realizing brain-computer interfaces (BCI) [63], where BCI (a.k.a. brain-machine interfaces (BMI)) [56] is a direct communication channel between a human or animal brain and an external device and have recently been an active research topic in the neurosicence field. In this sense, it is also an interesting issue to investigate the capacity of Switching ICA as a fundamental tool for developing effective BCI systems. Second, the online learning models proposed in this thesis provides an practical way to address unsystematic changes of environmental properties in the context of online feature extraction. This charecter becomes important for developing robust recognition systems that operate in real situations. A visual system for face detection and tracking, for example, have popularly used a PCA-like method as to obtain low-dimensional representation of face images. Consider a vision system that successively indentifies a unique person's face and keeps tracking it. To make the performance of PCA-based identification robust, it is important to dynamically update the representation bases according to the environmental changes. In such cases, the proposed methods are probably quite useful.

While the two kinds of methods proposed is thesis has described as to address BSS and feature extraction, respectively, their scopes are not necessarily limited to each of them, since such dynamic, non-stationary characters of data can be emerged in various context of data analysis and processing problems occurred in real situations. Even only focusing the two problems domains, general ideas employed in one of the two learning models can be beneficial for the other problem domain. Actually, the non-stationary appearance of latent variables, which was addressed in the BSS context, is often the case in feature extraction. When a vision system is monitoring a person's face in a frontal view, for example, the features of frontal face would sometimes partly disappear due to occlusions or temporal changes in the head direction. On the other hand, the non-stationary changes of bases themselves, which was focused within online feature extraction, also occurs in non-stationary BSS context, since the position of speakers in a cock-tail party, for example, sometimes changes with time. Applications of each methods to other problem domains beyond the original ones are remained in the future study. Furthermore, these indicate an interesting direction of a future extension of the present work, that is, to combine the dynamic variable selection mechanism employed in Switching ICA with the adaptive control schemes of forgetting factors in an online Bayesian learning model. This also an remained issue for the future study.

I employed the probabilistic ICA and PCA as a basic model to be extended in this thesis, each of which is an fundamental technique in BSS and unsupervised feature extraction. The dynamic variable selection scheme and the dynamic online learning schemes, however, have potentials to be incorporated into other kinds of representation learning methods that is based on latent variable models. Recently, an important variant of such methods, the latent Dirichlet allocation (LDA) [16], have attracted attentions, which have originally developed in the natural language processing field and recently been becoming popular in other areas of machine learning. This can be regarded as a PCA-like method to analyze multinomial dataset [18]. It will be an interesting direction to extend LDA to address dynamic cases in a similar way of this thesis. Another interesting issue is to employ the two proposed schemes collectively. The real-world environments often involves both kinds of non-stationarities as considered in this thesis, that is, the non-stationary appearance of latent variables and the abrupt changes in environmental characters. In this regard, it will be a promising approach to extend basic representation learning methods by means of both the dynamic variable selection and the dynamic online learning scheme.

Acknowledgements

I would like to express my sincere gratitude to Professer Shin Ishii for his constant supervision and support during this work. I thank Professer Yuji Matsumoto for his valuable comments during preparation of this thesis. I also wish to acknowledge the relevant suggestions from Associate Professor Tomohiro Shibata throughout the course of this work.

I am grateful to the current and old members of Theoretical Life-Science Lab. In particular, I would like to gratefully acknowledge Dr. Jun-ichiro Yoshimoto for his kindful guidance at the beggining of this work. The idea of online novelty detection based on a mixture PPCA is initially developed by him. I am also deeply grateful to Dr. Shin-ichi Maeda for his constant guidance and insightful suggestions during the project of Switching ICA.

References

[1] E. Acquas, C. Wilson, and H. C. Fibiger. Conditioned and unconditioned stimuli increase frontal cortical and hippocampal acetylcholine release: effects of novelty, habituation, and fear. *The journal of Neuroscience*, 16(9):3089–3096, 1996.

[2] L. B. Almeida. MISEP–linear and nonlinear ICA based on mutual information. *Signal Processing*, 84(2):231–245, 2004.

[3] S. Amari. Theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, 16(3):299–307, 1967.

[4] S. Amari, T. P. Chen, and A. Cichocki. Nonholonomic orthogonal learning algorithms for blind source separation. *Neural Computation*, 12(6):1463–1484, 2000.

[5] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems*, volume 8, pages 753–763, Cambridge MA, 1996. MIT press.

[6] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *J. Royal Stat. Soc. B*, 36:99–102, 1974.

[7] H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

[8] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proc. 15th Conf. on Uncertainty in AI*, pages 21–30, 1999.

[9] H. Attias. Independent factor analysis with temporally structured sources. In *Advances in Neural Information Processing Systems*, volume 12, pages 386–392, 2000.

[10] H. Attias. A variational bayesian framework for graphical models. In *Advances in Neural Information Processing Systems*, volume 12, pages 209–215, 2000.

[11] J. S. Bakin and N. M Weinberger. Induction of a physiological memory in the cerebral cortex by stimulation of the nucleus basalis. *Proceedings of the National Academy of Sciences of the United States of America*, 93:11219–11224, 1996.

[12] C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, Oxford, UK, 1995.

[13] C. M. Bishop. Latent variable models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 371–403. MIT Press, 1999.

[14] C. M. Bishop. Variational principal components. In *IEE Conference Publication on Artificial Neural Networks*, pages 509–514, 1999.

[15] D.T. Blake, N.N. Byl, and M.M. Merzenich. Representation of the hand in the cerebral cortex. *Behavioural Brain Research*, 135:179–184, 2002.

[16] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] P. J. Brown, M. Vannucci, and T. Fearn. Multivariate Bayesian variable selection and prediction. *Journal of the Royal Statistical Society, Series B*, 60(3):627–641, 1998.

[18] W. L. Buntine. Variational extensions to EM and multinomial PCA. In *ECML*, pages 23–34, 2002.

[19] K. Chan, T.-W. Lee, and T. J. Sejnowski. Variational Bayesian learning of ICA with missing data. *Neural Computation*, 15:1991–2011, 2003.

[20] R. Choudrey and S. Roberts. Flexible Bayesian independent component analysis for blind source separation. In *Proc. Int. Conf. on Independent Component Analysis and Signal Separation (ICA2001)*, pages 90–95, 2001.

[21] R. A. Choudrey and S. J. Roberts. Bayesian ICA with hidden Markov sources. In *International Conference on Independent Component Analysis*, pages 809–814, 2003.

[22] A. Cichocki, S. C. Douglas, and S. Amari. Robust techniques for independent component analysis (ICA) with noisy data. *Neurocomputing*, 22(1–3):113–129, 1998.

[23] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal components analysis to the exponential family. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[24] K. R. Daffner et al. The central role of the prefrontal cortex in directing attention to novel events. *Brain*, 123:927–939, 2000.

[25] C. Darken and J. E. Moody. Note on learning rate schedules for stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 3, pages 832–838, 1991.

[26] P. Dayan. Recurrent sampling models for the Helmholtz machine. *Neural Computation*, 11:653–677, 1999.

[27] P. Dayan and L. F. Abbott. *Theoretical neuroscience : computational and mathematical modeling of neural systems.* MIT Press, 2001.

[28] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society, Series B*, 39(1):1–38, 1977.

[29] R. Dias and R. C. Honey. Involvement of the rat medial prefrontal cortex in novelty detection. *Behaviroral Neuroscience*, 116(3):498–503, 2002.

[30] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo in Practice.* Springer-Verlag, 2001.

[31] K. Doya. Metalearning and neuromodulation. *Neural Network*, 15(4–6):495–506, 2002.

[32] K. Doya, S. Ishii, A. Pouget, and R. P. N. Rao, editors. *Bayesian Brain: Probabilistic Aproaches to Neural Coding.* MIT Press, 2006.

[33] J.-M. Edeline. Learning-induced physiological plasticity in the thalamocortical sensory systems: a critical evaluation of receptive field plasticity, map changes and their potential mechanisms. *Progress in Neurobiology*, 57:165–224, 1999.

[34] J.-M. Fellous and C. Linster. Computational models of neuromodulation. *Neural Computation*, 10:771–805, 1998.

[35] C. Févotte. Bayesian blind separation of audio mixtures with structured priors. In *Proc. 14th European Signal Processing Conference (EUSIPCO'06)*, 2006.

[36] J. D. Freguson. Variable duration models for speech. In *Proc. Symposium on the Application of Hidden Markov Models to Text and Speech*, pages 143–179, 1980.

[37] E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *J. Am. Statist. Ass.*, 88:881–889, 1993.

[38] E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7(2):339–373, 1997.

[39] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[40] J. Geweke. Variable selection and model comparison in regression. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 5*, pages 609–620. Clarendon Press, Oxford, UK, 1996.

[41] M. G. Giovannini, A. Rakovska, R. S. Benton, M. Pazzagli, L. Bianchi, and G. Pepeu. Effects of novelty and habituation on acetylcholine, GABA, and glutamate release from the frontal cortex and hippocampus of freely moving rats. *Neuroscience*, 106(1):43–53, 2001.

[42] R. Gribonval, L. Benaroya, E. Vincent, and C. Févotte. Proposals for performance measurement in source separation. In *Proc. 5th International*

Conference on Independent Component Analysis and Blind Signal Separation (ICA), number 763–768, 2003.

[43] S. Grossberg. Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23:121–134, 1976.

[44] M. E. Hasselmo. Neuromodulation and cortical function: modeling the physiological basis of behavior. *Behavioural Brain Research*, 67:1–27, 1995.

[45] M. E. Hasselmo and C. Linster. Neuromodulation and memory function. In P. S. Katz, editor, *Beyond Neurotransmission: The Role of Neuromodulation in Information Flow and Neuronal Circuit Flexibility*, chapter 9. Oxford University Press, Oxford, U. K., 1998.

[46] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, 1999.

[47] P. A. Højen-Sørensen, O. Winther, and L. K. Hansen. Mean-field approaches to independent component analysis. *Neural Computation*, 14(4):889–918, 2002.

[48] A. Hyvärinen. Independent component analysis in the presence of Gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67, 1998.

[49] A. Hyvärinen. Independent component analysis in the presence of Gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22(49–67), 1998.

[50] A. Hyvärinen. Gaussian moments for noisy independent component analysis. *IEEE Signal Processing Letters*, 6:145–147, 1999.

[51] A. Hyvärinen, P. O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

[52] A. Hyv̈arinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

[53] E. Mercado II, C. E. Myers, and M. A. Gluck. A computational model of mechanisms controlling experience-dependent reorganization of representational maps in auditory cortex. *Cognitive, Affective, & Behavioral Neuroscience*, 1(1):37–55, 2001.

[54] M. T. Johnson. Capacity and complexity of HMM duration modeling techniques. *IEEE Signal Processing Letters*, 12(5):407–410, 2005.

[55] M. P. Kilgard and M. M. Merzenich. Cortical map reorganization enabled by nucleus basalis activity. *Science*, 279:1714–1718, 1998.

[56] M. A. Lebedev and M. A. L. Nicolelis. Brain-machine interfaces: past, present and future. *Trends in Neuroscience*, 29:536–546, 2006.

[57] T.-W. Lee, B. Koehler, and R. Orglmeister. Blind separation of nonlinear mixing models. In *IEEE International Workshop on Neural Networks for Signal Processing*, pages 406–415, 1997.

[58] S. E. Levinson. Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech & Language*, 1(1):29–45, 1986.

[59] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[60] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Comp.*, 12:337–365, 2000.

[61] S. Maeda, W. Song, and S. Ishii. Nonlinear and noisy extension of independent component analysis: Theory and its application to a pitch sensation model. *Neural Computation*, 17:115–144, 2005.

[62] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski. Independent component analysis of electroencephalographic data. In *Advances in Neural Information Processing Systems*, volume 8, pages 145–151, 1996.

[63] S. Makeig, S. Enghoff, J. Tzyy-Ping, and T. J. Sejnowski. A natural basis for efficient brain-actuated control. *IEEE Transactions on Rehabilitation Engineering*, 8(2):208–211, 2000.

[64] T. Minka. Bayesian linear regression. Technical report, MIT, 2000.

[65] M. I. Miranda, L. Ramírez-Lugo, and F. Bermúdez-Rattoni. Cortical cholinergic activity is related to the novelty of the stimulus. *Brain Research*, 882:230–235, 2000.

[66] J. Miskin and D. MacKay. Ensemble learning for blind source separation. In S. Roberts and R. Everson, editors, *Independent Component Analysis: Principles and Practice*, pages 209–233. Cambridge University Press, 2001.

[67] E. Moulines, J.-F. Cardoso, and E. Gassiat. Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proc. ICASSP*, volume 5, pages 3617–3620, 1997.

[68] N. Murata, M. Kawanabe, A. Ziehe, K.-R. Müller, and S. i Amari. Online learning in changing environments with applications in supervised and unsupervised learning. *Neural Networks*, 15(4):743–760, 2002.

[69] N. Murata, K.-R. Müller, A. Ziehe, and S. i Amari. Adaptive on-line learning in changing environments. In *Advances in neural information processing systems*, volume 9, pages 599–605, Cambridge, MA, 1997. MIT Press.

[70] R. Neal and G. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1999.

[71] S. Oba, M. Sato, and S. Ishii. Variational Bayes method for mixture of principal component analyzers. In *proceeding for 7th International Conference on Neural Information Processing*, volume 2, pages 1416–1421, 2000.

[72] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[73] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[74] B. A. Olshausen and K. J. Millman. Learning sparse codes with a mixture-of-Gaussians prior. In *Advances in Neural Information Processing Systems*, volume 12, pages 841–884. MIT Press, 2000.

[75] W. D. Penny, R. M. Everson, and S. J. Roberts. Hidden Markov independent component analysis. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 3–22. Springer, 2000.

[76] S. Penschuck, C. H. Chen-Bee, N. Parakash, and R. D. Frostig. In vivo modulation of a cortical functional sensory representation shortly after topical cholinergic agent application. *J Comp Neurol*, 452(1):38–50, 2002.

[77] K. B. Petersen, O. Winther, and L. K. Hansen. On the slow convergence of em and vbem in low-noise linear models. *Neural Computation*, 17:1921–1926, 2005.

[78] D. T. Pham and J. F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE transactions on signal processing*, 49(9):1837–1848, 2001.

[79] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[80] C. Ranganath and G. Rainer. Neural mechanisms for detecting and remembering novel events. *Nature Review Neuroscience*, 4:193–202, 2003.

[81] R. P. N. Rao, B. A. Olshausen, and M. S. Lewicki, editors. *Probabilistic Models of the Brain: Perception and Neural Function.* MIT Press, 2002.

[82] D. D. Rasmusson. The role of acetylcholine in cortical synaptic plasticity. *Behavioural Brain Research*, 115:205–218, 2000.

[83] S. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2):305–345, 1999.

[84] M. Russell and R. Moore. Explicit modeling of state occupancy in hidden markov models for automatic speech recognition. In *Proceedings of IEEE Conference on Acoustics Speech and Signal Processing*, pages 5–8, 1985.

[85] J Särelä and R. Vigário. A Bayesian approach to overlearning in ICA: A comparison study. Technical Report A70, Helsinki university of technology, Laboratory of computer and information science, 2003.

[86] J. Särelä and R. Vigário. Overlearning in marginal distribution-based ICA: analysis and solutions. *Journal of Machine Learning Research*, 4:1447–1469, 2003.

[87] M. Sato. Online model selection based on the varational Bayes. *Neural Computaion*, 13:1649–1681, 2001.

[88] N. N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14:1723–1738, 2002.

[89] H. Snoussi and A. Mohammad-Djafari. Bayesian unsupervised learning for source separation with mixture of gaussians prior. *Journal of VLSI Signal Processing*, 37(2–3):263–279, 2004.

[90] H. Sompolinsky, N. Barkai, and H. S. Seung. On-line learning of dichotomies: algorithms and learning curves. In *Neural Networks: The Statistical Mechanics Perspective*, pages 105–130. World Scientific, 1995.

[91] A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. In *IEEE Transaction on Signal Processing*, volume 47, pages 2807–2820, 1999.

[92] M. Tipping and C. Bishop. Probabilistic principal component analysis. Technical report, Neural Computing Research Group, Aston University, 1997.

[93] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[94] H. Valpola. *Bayesian Ensemble Learning for Nonlinear Factor Analysis.* PhD thesis, Helsinki University of Technology, 2000.

[95] H. Valpola, M. Harva, and J. Karhunen. Hierarchical models of variance sources. *Signal Processing*, 84:267–282, 2004.

[96] E. Wan, A. Nelson, and R. Peterson. Speech Enhancement Assessment Resource (SpEAR) Database. http://ee.ogi.edu/NSEL/. Beta Release v1.0. CSLU, Oregon Graduate Institute of Science and Technology.

[97] M. Welling and M. Weber. A constrainted EM algorithm for independent component analysis. *Neural Computation*, 13:677–689, 2001.

[98] S. Wu, D. Chen, M. Niranjan, and S. i. Amari. Sequential Bayesian decoding with a population of neurons. *Neural Computation*, 15(5):993–1012, 2003.

[99] A. Yu and P. Dayan. Acetylcholine in cortical inference. *Neural Networks*, 15(4–6):719–730, 2002.

[100] A. Yu and P. Dayan. Expected and unexpected uncertainty: ACh and NE in the neocortex. In *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2002. MIT Press.

[101] L. Zaborszky et al. The basal forebrain corticopetal system revisited. *Ann N Y Acad Sci*, 877:339–367, 1999.

[102] L. Zaborszky, R. P. Gaykema, D. J. Swanson, and W. E. Cullinan. Cortical input to the basal forebrain. *Neuroscience*, 79(4):1051–1078, 1997.

Appendix

A. Probabilistic distributions

A matrix normal distribution is defined as

$$
\begin{aligned}
&\mathrm{N}_{d\times n}\left(\boldsymbol{A} \mid \boldsymbol{M}, \boldsymbol{V}, \boldsymbol{K}\right) \\
&= (2\pi)^{-\frac{dn}{2}} |\boldsymbol{K}|^{-\frac{d}{2}} |\boldsymbol{V}|^{-\frac{n}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left[\left(\boldsymbol{A}-\boldsymbol{M}\right)^T \boldsymbol{V}^{-1}\left(\boldsymbol{A}-\boldsymbol{M}\right)\boldsymbol{K}^{-1}\right]\right),
\end{aligned}
$$

where $\boldsymbol{A} \in \Re^{d\times n}$, $\boldsymbol{M} \in \Re^{d\times n}$, $\boldsymbol{K} \in \Re^{n\times n}$, and $\boldsymbol{V} \in \Re^{d\times d}$. Here, $\boldsymbol{M}$ denotes the mean of $\boldsymbol{A}$; $\boldsymbol{K}$ and $\boldsymbol{V}$ are two covariance matrices of $\boldsymbol{A}$ [64]. A Gamma distribution is defined as

$$
\mathrm{Ga}(x \mid a, b) = \frac{b^a x^{a-1}}{\Gamma(a)} \exp(-bx), \tag{4.1}
$$

where $x \geq 0$, $a > 0$ and $b > 0$. A Beta distribution is defined as

$$
\mathrm{Be}\left(r \mid u, w\right) = \frac{\Gamma(u+w)}{\Gamma(u)\Gamma(w)} r^{u-1}(1-r)^{w-1}, \tag{4.2}
$$

where $0 < r < 1$.

B. Appendix for chapter 3

B.1 Derivations of $q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t)$ and $l(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)$

In the following derivation, we regard the inactive model of source signals $p(s_{i,t} \mid z_{i,t} = 0)$ as a Gaussian $N(s_{i,t} \mid 0, \epsilon)$ with a small variance ϵ, which will take the limit $\epsilon \to 0$ later. Let $\boldsymbol{V}_t = \mathrm{diag}(v_{1,t}, v_{2,t}, \ldots, v_{n,t})$ where $v_{i,t} = \gamma_{i,t}^{-1}$ (if $z_{i,t} = 1$)

or ϵ (if $z_{i,t} = 0$), then

$$\begin{aligned}
&q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t)\\
&\propto \exp(\langle \log p(\boldsymbol{x}_t \mid \boldsymbol{s}_t, \boldsymbol{A}, \beta)\rangle_{\boldsymbol{A},\beta}) \exp\left(\langle \log p(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t, \boldsymbol{\gamma})\rangle_{\boldsymbol{\gamma}}\right) && \text{(4.3a)}\\
&= (2\pi)^{-\frac{d}{2}} \exp\left(\langle \log \beta\rangle\right)^{\frac{d}{2}} \exp\left(-\frac{1}{2}\left(\mathrm{tr}\left[\langle \beta \boldsymbol{A}\boldsymbol{A}^T\rangle \boldsymbol{s}_t\boldsymbol{s}_t^T\right] - 2\mathrm{tr}\left[\langle \beta \boldsymbol{A}\rangle^T \boldsymbol{x}_t\boldsymbol{s}_t^T\right]\right.\right.\\
&\quad \left.\left.+ \mathrm{tr}\left[\langle\beta\rangle \boldsymbol{x}_t\boldsymbol{x}_t^T\right]\right)\right)(2\pi)^{-\frac{n}{2}} \exp\left(\langle \log |\boldsymbol{V}_t|\rangle\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left[\langle \boldsymbol{V}_t\rangle^{-1}\boldsymbol{s}_t\boldsymbol{s}_t^T\right]\right) && \text{(4.3b)}\\
&= (2\pi)^{-\frac{d+n}{2}} \exp\left(\frac{d}{2}\langle \log\beta\rangle\right) \exp\left(-\frac{1}{2}\langle \log|\boldsymbol{V}_t|\rangle\right) \exp\left(-\frac{1}{2}\mathrm{tr}\left[\langle\beta\rangle \boldsymbol{x}_t\boldsymbol{x}_t^T\right]\right)\\
&\quad \times \exp\left(-\frac{1}{2}\left(\mathrm{tr}\left[\left(\langle \beta \boldsymbol{A}\boldsymbol{A}^T\rangle + \langle \boldsymbol{V}_t\rangle^{-1}\right)\boldsymbol{s}_t\boldsymbol{s}_t^T\right] - 2\mathrm{tr}\left[\langle\beta\boldsymbol{A}\rangle^T \boldsymbol{x}_t\boldsymbol{s}_t^T\right]\right)\right) && \text{(4.3c)}\\
&= (2\pi)^{-\frac{d}{2}} \exp\left(\frac{d}{2}\langle \log\beta\rangle - \frac{1}{2}\langle \log|\boldsymbol{V}_t|\rangle - \frac{1}{2}\mathrm{tr}\left[\langle\beta\rangle \boldsymbol{x}_t\boldsymbol{x}_t^T\right]\right)\\
&\quad \times |\hat{\boldsymbol{V}}_t|^{-\frac{1}{2}} \exp\left(\frac{1}{2}\mathrm{tr}\left[\hat{\boldsymbol{V}}_t\hat{\boldsymbol{\mu}}_t\hat{\boldsymbol{\mu}}_t^T\right]\right) \mathrm{N}_n(\boldsymbol{s}_t \mid \hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{V}}_t), && \text{(4.3d)}
\end{aligned}$$

where

$$\hat{\boldsymbol{V}}_t = \left(\langle \beta\boldsymbol{A}\boldsymbol{A}^T\rangle + \langle \boldsymbol{V}_t\rangle^{-1}\right)^{-1}, \quad \text{(4.4a)}$$

$$\hat{\boldsymbol{\mu}}_t = \hat{\boldsymbol{V}}_t\langle\beta\boldsymbol{A}\rangle^T\boldsymbol{x}_t. \quad \text{(4.4b)}$$

Thus, $q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t)$ has the Gaussian form:

$$q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t) = \mathrm{N}_n(\boldsymbol{s}_t \mid \hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{V}}_t). \quad \text{(4.5)}$$

The normalization term, $l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)$, can also be given as

$$\begin{aligned}
l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t) = (2\pi)^{-\frac{d}{2}} \exp&\left(\frac{d}{2}\langle\log\beta\rangle - \frac{1}{2}\langle\log|\boldsymbol{V}_t|\rangle - \frac{1}{2}\log|\hat{\boldsymbol{V}}_t|\right.\\
&\left.- \frac{1}{2}\mathrm{tr}\left[\langle\beta\rangle\boldsymbol{x}_t\boldsymbol{x}_t^T\right] + \frac{1}{2}\mathrm{tr}\left[\hat{\boldsymbol{V}}_t\hat{\boldsymbol{\mu}}_t\hat{\boldsymbol{\mu}}_t^T\right]\right). \quad \text{(4.6)}
\end{aligned}$$

Finally, by taking the limit $\epsilon \to 0$ with respect to the inactive sources under $\boldsymbol{z}_t = \boldsymbol{\zeta}_h$, the corresponding rows and columns of $\hat{\boldsymbol{V}}_t$ and the elements of $\hat{\boldsymbol{\mu}}_t$ become zeros. The Gaussian in Eq. (4.5) then degenerates such that each inactive dimension has a zero mean and zero variance, which yields the Gaussian-Delta form given in Eq. (2.21). The limit of $l(\boldsymbol{s}_t, \boldsymbol{y}_t, \boldsymbol{z}_t)$ is also given as

$$l(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t) = (2\pi)^{-\frac{d}{2}} \exp\left(\frac{d}{2}\langle \log \beta \rangle - \frac{1}{2}\langle \log |\boldsymbol{V}_{\boldsymbol{y}_t^h}| \rangle - \frac{1}{2} \log |\hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h}| \right.$$
$$\left. - \frac{1}{2}\mathrm{tr}\left[\langle \beta \rangle \boldsymbol{x}_t \boldsymbol{x}_t^T \right] + \frac{1}{2}\mathrm{tr}\left[\hat{\boldsymbol{V}}_{\boldsymbol{y}_t^h} \hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h} \hat{\boldsymbol{\mu}}_{\boldsymbol{y}_t^h}^T \right] \right). \qquad (4.7)$$

B.2 Approximate posteriors for model parameters

According to Eq. (2.17), the optimized trial posterior distributions $q(\boldsymbol{A}, \beta)$, $q(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ and $q(\boldsymbol{\pi}, \boldsymbol{\rho})$, are given as follows.

Derivation of $q(\boldsymbol{A}, \beta)$

Using the following notations (cf. [64]):

$$\boldsymbol{R}_{ss} = \langle \boldsymbol{S}\boldsymbol{S}^T \rangle + \boldsymbol{G}_0, \qquad (4.8a)$$
$$\boldsymbol{R}_{xs} = \boldsymbol{X} \langle \boldsymbol{S} \rangle^T + \boldsymbol{M}_0 \boldsymbol{G}_0, \qquad (4.8b)$$
$$\boldsymbol{R}_{xx} = \boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{M}_0 \boldsymbol{G}_0 \boldsymbol{M}_0^T, \qquad (4.8c)$$
$$\boldsymbol{R}_{x|s} = \boldsymbol{R}_{xx} - \boldsymbol{R}_{xs} \boldsymbol{R}_{ss}^{-1} \boldsymbol{R}_{xs}^T, \qquad (4.8d)$$

Equation (2.17a) can be calculated as

$$q(\boldsymbol{A}, \beta)$$
$$\propto \exp\left(\langle \log p(\boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{A}, \beta) \rangle_{\boldsymbol{S}} \right) p_0(\boldsymbol{A}, \beta) \qquad (4.9a)$$
$$\propto \beta^{\frac{d\tau}{2}} \exp\left(-\frac{\beta}{2}\mathrm{tr}\left[\left\langle (\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}) (\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S})^T \right\rangle_{\boldsymbol{S}} \right] \right)$$
$$\times \beta^{\frac{dn}{2}} \exp\left(-\frac{\beta}{2}\mathrm{tr}\left[(\boldsymbol{A} - \boldsymbol{M}_0)^T (\boldsymbol{A} - \boldsymbol{M}_0) \boldsymbol{G}_0 \right] \right) \beta^{\kappa_0 - 1} \exp(-\lambda_0 \beta) \qquad (4.9b)$$
$$\propto \beta^{\frac{d(n+\tau)}{2} + \kappa_0 - 1} \exp\left(-\frac{\beta}{2}\mathrm{tr}\left[\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{R}_{ss} - 2\boldsymbol{A}^T \boldsymbol{R}_{xs} \right] - \frac{\beta}{2}\mathrm{tr}\left[\boldsymbol{R}_{xx} \right] \right)$$
$$\times \exp(-\lambda_0 \beta) \qquad (4.9c)$$
$$= \beta^{\frac{dn}{2}} \exp\left(-\frac{\beta}{2}\mathrm{tr}\left[\left(\boldsymbol{A} - \boldsymbol{R}_{xs}\boldsymbol{R}_{ss}^{-1} \right)^T \left(\boldsymbol{A} - \boldsymbol{R}_{xs}\boldsymbol{R}_{ss}^{-1} \right) \boldsymbol{R}_{ss} \right] \right)$$
$$\times \beta^{\kappa_0 + \frac{d\tau}{2} - 1} \exp\left(-\left(\lambda_0 + \frac{1}{2}\mathrm{tr}\left[\boldsymbol{R}_{x|s} \right] \right) \beta \right). \qquad (4.9d)$$

In Eq. (4.8), the sufficient statistics $\langle \boldsymbol{S} \rangle$ and $\langle \boldsymbol{S}\boldsymbol{S}^T \rangle$ are calculated as explained in Sec. 3.3, using $\langle \boldsymbol{S} \rangle = (\langle \boldsymbol{s}_1 \rangle, \langle \boldsymbol{s}_2 \rangle, \ldots, \langle \boldsymbol{s}_\tau \rangle)$ and $\langle \boldsymbol{S}\boldsymbol{S}^T \rangle = \sum_{t=1}^{\tau} \langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle$. Now we define the following quantities:

$$\hat{\boldsymbol{M}} = \boldsymbol{R}_{xs}\boldsymbol{R}_{ss}^{-1}, \quad \hat{\boldsymbol{G}} = \boldsymbol{R}_{ss} \tag{4.10a}$$

$$\hat{\kappa} = \kappa_0 + \frac{d\tau}{2}, \quad \hat{\lambda} = \lambda_0 + \frac{1}{2}\mathrm{tr}\left[\boldsymbol{R}_{x|s}\right], \tag{4.10b}$$

then, by normalizing Eq. (4.9d) with respect to $\boldsymbol{A}$ and β, the approximate posterior $q(\boldsymbol{A}, \beta)$ is given in the same form as the prior distribution, $p_0(\boldsymbol{A}, \beta)$, in Eq. (2.11a):

$$q(\boldsymbol{A}, \beta) = \mathrm{N}_{d\times n}(\boldsymbol{A} \mid \hat{\boldsymbol{M}}, \beta^{-1}\boldsymbol{I}_d, \hat{\boldsymbol{G}}^{-1})\mathrm{Ga}(\beta \mid \hat{\kappa}, \hat{\lambda}). \tag{4.11}$$

Finally, the expectations required for the VB-E step, $\langle \beta \boldsymbol{A} \rangle$ and $\langle \beta \boldsymbol{A}^T\boldsymbol{A} \rangle$, are given as

$$\langle \beta \boldsymbol{A} \rangle = \hat{\beta}\hat{\boldsymbol{M}}, \tag{4.12a}$$

$$\langle \beta \boldsymbol{A}^T\boldsymbol{A} \rangle = \hat{\beta}\hat{\boldsymbol{M}}^T\hat{\boldsymbol{M}} + d\hat{\boldsymbol{G}}^{-1}, \tag{4.12b}$$

where $\hat{\beta} = \hat{\kappa}/\hat{\lambda}$.

Derivation of $q(\boldsymbol{\alpha}, \boldsymbol{\gamma})$

Equation (2.17b) can be further factorized as $q(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = q(\boldsymbol{\alpha})q(\boldsymbol{\gamma})$, where

$$q(\boldsymbol{\gamma}) = \frac{1}{C_\gamma}\exp\Big(\langle \log p(\boldsymbol{S} \mid \boldsymbol{Y}, \boldsymbol{Z}, \gamma) \rangle_{\boldsymbol{S},\boldsymbol{Y},\boldsymbol{Z}}\Big)\, p_0(\boldsymbol{\gamma}), \tag{4.13a}$$

$$q(\boldsymbol{\alpha}) = \frac{1}{C_\alpha}\exp\left(\langle \log p(\boldsymbol{Y} \mid \boldsymbol{\alpha}) \rangle_{\boldsymbol{Y}}\right) p_0(\boldsymbol{\alpha}). \tag{4.13b}$$

C_γ and C_α are the normalization terms ($C_\phi = C_\gamma C_\alpha$). By regarding the inactive model as having a small variance ϵ as in Sec. B.1, Eq. (4.13a) can be given as

$$\begin{aligned}
q(\boldsymbol{\gamma}) &\propto \exp\Big(\langle \log p(\boldsymbol{S} \mid \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\gamma})\rangle_{\boldsymbol{S},\boldsymbol{Y},\boldsymbol{Z}}\Big)\, p_0(\boldsymbol{\gamma}) \\
&\propto \prod_{t=1}^{T} \exp\left(-\frac{1}{2}\left\langle \log |\boldsymbol{V}_t| \right\rangle_{\boldsymbol{y}_t,\boldsymbol{z}_t}\right) \exp\left(-\frac{1}{2}\left\langle \boldsymbol{s}_t^T \boldsymbol{V}_t^{-1} \boldsymbol{s}_t \right\rangle_{\boldsymbol{s}_t,\boldsymbol{y}_t,\boldsymbol{z}_t}\right) \\
&\quad \times \prod_{i=1}^{n} \gamma_{0i}^{u_{\gamma_{0i}}-1} \exp\left(-w_{\gamma_{0i}}\gamma_{0i}\right) \gamma_{1i}^{u_{\gamma_{1i}}-1} \exp\left(-w_{\gamma_{1i}}\gamma_{1i}\right) &(4.14\text{a}) \\
&= \prod_{i=1}^{n} \exp\left(-\frac{1}{2}\sum_{t=1}^{T} \langle \log v_{i,t}\rangle\right) \exp\left(-\frac{1}{2}\sum_{t=1}^{T} \langle v_{i,t}^{-1} s_{i,t}^2\rangle\right) \\
&\quad \times \prod_{i=1}^{n} \gamma_{0i}^{u_{\gamma_{0i}}-1} \exp\left(-w_{\gamma_{0i}}\gamma_{0i}\right) \gamma_{1i}^{u_{\gamma_{1i}}-1} \exp\left(-w_{\gamma_{1i}}\gamma_{1i}\right). &(4.14\text{b})
\end{aligned}$$

The expectations in Eq. (4.14b) are given as

$$\left\langle \log v_{i,t}\right\rangle_{y_{i,t},z_{i,t}} = (1 - \langle z_{i,t}\rangle) \log \epsilon - \langle z_{i,t}(1 - y_{i,t})\rangle \log \gamma_{0i} - \langle z_{i,t} y_{i,t}\rangle \log \gamma_{1i}, \qquad (4.15\text{a})$$

$$\langle v_{i,t}^{-1} s_{i,t}^2\rangle_{s_{i,t},y_{i,t},z_{i,t}} = \langle z_{i,t}(1 - y_{i,t}) s_{i,t}^2\rangle \gamma_{0i} + \langle z_{i,t} y_{i,t} s_{i,t}^2\rangle \gamma_{1i} + (1 - \langle z_{i,t}\rangle)\epsilon^{-1}. \qquad (4.15\text{b})$$

where we use $v_{i,t} = (\gamma_{i,t}^{-1})^{z_{i,t}} \epsilon^{1-z_{i,t}}$ and $\gamma_{i,t} = \gamma_{0i}^{1-y_{i,t}} \gamma_{1i}^{y_{i,t}}$ in Eq. (4.15a), and $v_{i,t}^{-1} = z_{i,t}\gamma_{i,t} + (1 - z_{i,t})\epsilon^{-1}$ and $\gamma_{i,t} = (1 - y_{i,t})\gamma_{0i} + y_{i,t}\gamma_{1i}$ in Eq. (4.15b). Now let

$$\hat{u}_{\gamma_{0i}} = u_{\gamma_{0i}} + \frac{\tau - \sum_{t=1}^{\tau} \langle y_{i,t}\rangle}{2}; \quad \hat{w}_{\gamma_{0i}} = w_{\gamma_{0i}} + \frac{1}{2}\sum_{t=1}^{\tau} \left\langle z_{i,t}\left(1 - y_{i,t}\right) s_{i,t}^2 \right\rangle, \qquad (4.16\text{a})$$

$$\hat{u}_{\gamma_{1i}} = u_{\gamma_{1i}} + \frac{\sum_{t=1}^{\tau} \langle y_{i,t}\rangle}{2}; \quad \hat{w}_{\gamma_{1i}} = w_{\gamma_{1i}} + \frac{1}{2}\sum_{t=1}^{\tau} \left\langle z_{i,t} y_{i,t} s_{i,t}^2 \right\rangle, \qquad (4.16\text{b})$$

then, according to Eqs. (4.14b), (4.15), and (4.16),

$$q(\boldsymbol{\gamma}) = \prod_{i=1}^{n} \mathrm{Ga}(\gamma_{0i} \mid \hat{u}_{\gamma_{0i}}, \hat{w}_{\gamma_{0i}}) \mathrm{Ga}(\gamma_{1i} \mid \hat{u}_{\gamma_{1i}}, \hat{w}_{\gamma_{1i}}). \qquad (4.17)$$

We note that, in Eq. (4.14b), any term including $\gamma_{\cdot i}$ does not depend on ϵ, and then ϵ is canceled out by the normalization when obtaining Eq. (4.17). The expectation of $\gamma_{\cdot i}$ can be calculated as $\langle \gamma_{\cdot i}\rangle = \hat{u}_{\gamma_{\cdot i}} / \hat{w}_{\gamma_{\cdot i}}$. Updating rules for conjugate

Beta distributions are rather straightforward. The approximate posterior for $\boldsymbol{\alpha}$ is given as

$$q(\boldsymbol{\alpha}) = \prod_{i=1}^{n} \mathrm{Be}(\alpha_i \mid \hat{u}_{\alpha_i}, \hat{w}_{\alpha_i}), \tag{4.18}$$

where the posterior hyperparameters are given as the sum of the prior pseudo-count and the expected count:

$$\hat{u}_{\alpha_i} = u_{\alpha_i} + \sum_{t=1}^{T} \langle y_{i,t} \rangle, \tag{4.19a}$$

$$\hat{w}_{\alpha_i} = w_{\alpha_i} + \sum_{t=1}^{T} \langle 1 - y_{i,t} \rangle. \tag{4.19b}$$

Finally, the expectations are calculated as

$$\langle \log \alpha_i \rangle = \psi(\hat{u}_{\alpha_i}) - \psi(\hat{u}_{\alpha_i} + \hat{w}_{\alpha_i}), \tag{4.20a}$$

$$\langle \log(1 - \alpha_i) \rangle = \psi(\hat{w}_{\alpha_i}) - \psi(\hat{u}_{\alpha_i} + \hat{w}_{\alpha_i}), \tag{4.20b}$$

where $\psi(\cdot)$ denotes the digamma function.

Derivation of $q(\boldsymbol{\pi}, \boldsymbol{\rho})$

The approximate posterior for $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ is also given in the conjugate form as $p_0(\boldsymbol{\pi}, \boldsymbol{\rho})$ in Eq. (2.11c):

$$q(\boldsymbol{\pi}, \boldsymbol{\rho}) = \prod_{i=1}^{n} \mathrm{Be}(\pi_i \mid \hat{u}_{\pi_i}, \hat{w}_{\pi_i}) \mathrm{Be}(\rho_{0i} \mid \hat{u}_{\rho_{0i}}, \hat{w}_{\rho_{0i}}) \mathrm{Be}(\rho_{1i} \mid \hat{u}_{\rho_{1i}}, \hat{w}_{\rho_{1i}}), \tag{4.21}$$

where, for $i = 1, \ldots, n$,

$$\hat{u}_{\pi_i} = u_{\pi_i} + \left\langle z_{i,1} \right\rangle, \quad \hat{w}_{\pi_i} = w_{\pi_i} + 1 - \left\langle z_{i,1} \right\rangle, \tag{4.22a}$$

$$\hat{u}_{\rho_{0i}} = u_{\rho_{0i}} + \sum_{t=2}^{\tau} \left\langle \left(1 - z_{i,t}\right) z_{i,t-1} \right\rangle, \quad \hat{w}_{\rho_{0i}} = w_{\rho_{0i}} + \sum_{t=2}^{\tau} \left\langle z_{i,t} z_{i,t-1} \right\rangle, \tag{4.22b}$$

$$\hat{u}_{\rho_{1i}} = u_{\rho_{1i}} + \sum_{t=2}^{\tau} \left\langle z_{i,t} \left(1 - z_{i,t-1}\right) \right\rangle, \quad \hat{w}_{\rho_{1i}} = w_{\rho_{1i}} + \sum_{t=2}^{\tau} \left\langle \left(1 - z_{i,t}\right) \left(1 - z_{i,t-1}\right) \right\rangle. \tag{4.22c}$$

The expected sufficient statistics are calculated from $q(\boldsymbol{z}_t)$ and $q(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$, which are obtained by the Forward-Backward algorithm in Sec 3.4. The expectations required for the VB-E step are given by

$$\begin{aligned}
\langle \log \pi_i \rangle &= \psi(\hat{u}_{\pi_i}) - \psi(\hat{u}_{\pi_i} + \hat{w}_{\pi_i}), && \text{(4.23a)} \\
\langle \log(1 - \pi_i) \rangle &= \psi(\hat{w}_{\pi_i}) - \psi(\hat{u}_{\pi_i} + \hat{w}_{\pi_i}), && \text{(4.23b)} \\
\langle \log \rho_{0i} \rangle &= \psi(\hat{u}_{\rho_{0i}}) - \psi(\hat{u}_{\rho_{0i}} + \hat{w}_{\rho_{0i}}), && \text{(4.23c)} \\
\langle \log(1 - \rho_{0i}) \rangle &= \psi(\hat{w}_{\rho_{0i}}) - \psi(\hat{u}_{\rho_{0i}} + \hat{w}_{\rho_{0i}}), && \text{(4.23d)} \\
\langle \log \rho_{1i} \rangle &= \psi(\hat{u}_{\rho_{1i}}) - \psi(\hat{u}_{\rho_{1i}} + \hat{w}_{\rho_{1i}}), && \text{(4.23e)} \\
\langle \log(1 - \rho_{1i}) \rangle &= \psi(\hat{w}_{\rho_{1i}}) - \psi(\hat{u}_{\rho_{1i}} + \hat{w}_{\rho_{1i}}). && \text{(4.23f)}
\end{aligned}$$

B.3 Summary of updating rules

A summary of updating rules in the Switching ICA algorithm is given below.

1. VB-E step:
   (a) Calculate $\langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t,\boldsymbol{z}_t}$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t,\boldsymbol{z}_t}$ from $q(\boldsymbol{s}_t \mid \boldsymbol{y}_t, \boldsymbol{z}_t)$.

   (b) Calculate $\langle \boldsymbol{y}_t \rangle_{\boldsymbol{y}_t|\boldsymbol{z}_t}$ from $q(\boldsymbol{y}_t \mid \boldsymbol{z}_t)$, and obtain its normalization term, $e(\boldsymbol{x}_t, \boldsymbol{z}_t)$.

   (c) Calculate $\langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ from $q(\boldsymbol{y}_t \mid \boldsymbol{z}_t)$, $\langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t,\boldsymbol{z}_t}$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t,\boldsymbol{z}_t}$ (Eq. (2.25)).

   (d) Calculate $q(\boldsymbol{z}_t)$ and $q(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$ by the Forward-Backward algorithm based on $e(\boldsymbol{x}_t, \boldsymbol{z}_t)$.

   (e) Calculate $\langle \boldsymbol{s}_t \rangle$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle$ from $q(\boldsymbol{z}_t)$, $\langle \boldsymbol{s}_t \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{z}_t}$ (Eq. (2.27)).

   (f) Calculate $\langle z_{i,t} s_{i,t}^2 \rangle$ and $\langle z_{i,t} y_{i,t} s_{i,t}^2 \rangle$ from $q(\boldsymbol{z}_t)$, $q(\boldsymbol{y}_t \mid \boldsymbol{z}_t)$, $\langle s_{i,t}^2 \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t,\boldsymbol{z}_t}$ (diagonals of $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle_{\boldsymbol{s}_t|\boldsymbol{y}_t,\boldsymbol{z}_t}$).

   (g) Calculate $\langle z_{i,t} \rangle$ and $\langle z_{i,t} z_{i,t-1} \rangle$ from $q(\boldsymbol{z}_t)$ and $q(\boldsymbol{z}_t, \boldsymbol{z}_{t-1})$.

2. VB-M step:
   (a) Update $q(\boldsymbol{A}, \beta)$ with $\langle \boldsymbol{s}_t \rangle$ and $\langle \boldsymbol{s}_t \boldsymbol{s}_t^T \rangle$.

   (b) Update $q(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ with $\langle y_{i,t} \rangle$, $\langle z_{i,t} s_{i,t}^2 \rangle$ and $\langle z_{i,t} y_{i,t} s_{i,t}^2 \rangle$.

(c) Update $q(\boldsymbol{\pi}, \boldsymbol{\rho})$ with $\langle z_{i,t} \rangle$ and $\langle z_{i,t} z_{i,t-1} \rangle$.

(d) Calculate the expectations:
$\langle \beta \boldsymbol{A} \rangle$, $\langle \beta \boldsymbol{A}^T \boldsymbol{A} \rangle$ $\langle \gamma_{\cdot i} \rangle$, $\langle \log \alpha_i \rangle$, $\langle \log(1 - \alpha_i) \rangle$, $\langle \log \pi_i \rangle$, $\langle \log(1 - \pi) \rangle$, $\langle \log \rho_{\cdot i} \rangle$, $\langle \log(1 - \rho_{\cdot i}) \rangle$.

C. Appendix for Chapter 3

C.1 Online VB learning for MPPCA model

In this appendix section, we describe the implementation of the online VB learning for the model (3.8). In this implementation, we assume a natural conjugate prior for $p(\boldsymbol{\Theta} \mid m)$, given by

$$p\left(\boldsymbol{\Theta}|m\right) = \prod_{h=1}^{n} \mathcal{N}_{m+1}\left(\boldsymbol{\theta}_h \mid \boldsymbol{e}_h, \gamma^{-1}\boldsymbol{I}_{m+1}\right). \tag{4.24}$$

Here, $\boldsymbol{\Theta} \equiv (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n)^T$, $\boldsymbol{\theta}_h = (w_{h1}, w_{h2}, \ldots, w_{hm}, \mu_h)^T \in \Re^{(m+1)}$, w_{ij} is the (i,j)-element of matrix $\boldsymbol{W}$ and μ_j is the j-th element of vector $\boldsymbol{\mu}$. $\boldsymbol{e}_h \equiv (\delta_{h,1}, \cdots, \delta_{h,m}, 0)^T \in \Re^{(m+1)}$, $\delta_{i,j}$ is the Kronecker's delta, and γ ($\gamma > 0$) is a constant inverse variance. The mean of each principal component vector $\boldsymbol{w}_j$ ($j = 1, \cdots, m$) over the prior distribution (4.24) becomes orthogonal with the others, and its norm equals 1. Since the principal component vectors are estimated as orthonormal bases when there are no observed data, this prior distribution is suitable for PCA.

The algorithm of our online VB learning for the modified MPPCA is summarized as follows.

1. **Initialization phase**:

   Initialize the trial posterior of parameter, $q_\theta(\boldsymbol{\Theta}|m)$, for each principal component dimensionality, $m = 1, \cdots, n$, such as to be equal to the prior distribution:

$$q_\theta(\boldsymbol{\Theta}|m) = \prod_{h=1}^{n} \mathcal{N}_{m+1}\left(\boldsymbol{\theta}_h \mid \hat{\boldsymbol{\theta}}_h, \hat{\gamma}^{-1}\hat{\boldsymbol{V}}\right), \tag{4.25}$$

   where $\hat{\boldsymbol{\theta}}_h = \boldsymbol{e}_h, \hat{\gamma} = \gamma$ and $\hat{\boldsymbol{V}} = \boldsymbol{I}_{m+1}$. Initialize $\lambda_m(0)$ simply at $\lambda_m(0) = 1$ for each m.

2. **Inference phase**:

After observing a datum $\boldsymbol{x}_t$ at time step t, the following procedure is executed.

(a) **Online VB-Estep**:

For each $m = 1, \cdots, n$, $F_m^\lambda(t)$ is maximized with respect to $q_t(\boldsymbol{y}_t, z_t \mid \boldsymbol{x}_t, m)$. The solution does not depend on $\lambda_m(s)$ $(s = 1, \cdots, t)$ and is given by

$$
\begin{aligned}
&q_t(\boldsymbol{y}_t, z_t = i | \boldsymbol{x}_t, m) \\
&= \frac{\exp\Big[E\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t, z_t = i | \boldsymbol{\Theta})\right]\Big]}{\sum_{j \in \{0,1\}} \int d\boldsymbol{y}_t \exp\Big[E\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t, z_t = i | \boldsymbol{\Theta})\right]\Big]}, \quad i = 0, 1. \qquad (4.26)
\end{aligned}
$$

Based on this posterior distribution, the forgetting factor $\lambda_m(t)$ is given by equation (3.10). The effective data number $T_m^\lambda(t)$ and the learning rate $\eta_m(t)$ are updated by using the forgetting factor $\lambda_m(t)$:

$$
T_m^\lambda(t) = 1 + \lambda_m(t) T_m^\lambda(t), \quad \eta_m(t) = 1/T_m^\lambda(t). \qquad (4.27)
$$

(b) **Online VB-Mstep**:

For each $m = 1, \cdots, n$, $F_m^\lambda(t)$ is maximized with respect to $q_\theta(\boldsymbol{\Theta} | X_{1:t}, m)$. The solution is given by

$$
q_\theta(\boldsymbol{\Theta} | X_{1:t}, m) = \prod_{h=1}^{n} \mathcal{N}_{m+1}\left(\boldsymbol{\theta}_h \mid \hat{\boldsymbol{\theta}}_h, \hat{\gamma}^{-1} \hat{\boldsymbol{V}}\right), \qquad (4.28)
$$

where

$$
\begin{aligned}
\hat{\gamma} =& \sigma_x^{-2} T_m^\lambda(t) \langle (1 - z) \rangle_m(t) + \sigma_\epsilon^{-2} T_m^\lambda(t) \langle z \rangle_m(t) + \gamma \\
& \qquad (4.29) \\
\hat{\boldsymbol{V}} =& \frac{1}{\hat{\gamma}} \big(\sigma_x^{-2} T_m^\lambda(t) \langle (1 - z) \tilde{\boldsymbol{y}} \tilde{\boldsymbol{y}}^T \rangle_m(t) \\
& + \sigma_\epsilon^{-2} T_m^\lambda(t) \langle z \tilde{\boldsymbol{y}} \tilde{\boldsymbol{y}}^T \rangle_m(t) + \gamma \boldsymbol{I}_{m+1}\big) \qquad (4.30) \\
\left(\hat{\boldsymbol{\theta}}_1, \cdots, \hat{\boldsymbol{\theta}}_n\right)^T =& \frac{1}{\hat{\gamma}} \big(\sigma_x^{-2} T_m^\lambda(t) \langle (1 - z) \boldsymbol{x} \tilde{\boldsymbol{y}}^T \rangle_m(t) \\
& + \sigma_\epsilon^{-2} T_m^\lambda(t) \langle z \boldsymbol{x} \tilde{\boldsymbol{y}}^T \rangle_m(t)\big) \hat{\boldsymbol{V}}^{-1}. \qquad (4.31)
\end{aligned}
$$

$\langle f(\boldsymbol{x}, \boldsymbol{y}, z)\rangle_m(t)$ is the expected sufficient statistics defined by

$$\langle f(\boldsymbol{x}, \boldsymbol{y}, z)\rangle_m(t) \equiv \eta_m(t) \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda_m(s) \right) E\left[f(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau, z_\tau) \right]. \quad (4.32)$$

This weighted mean is calculated step-wisely, using that of the previous time step $t-1$:

$$\begin{aligned} \langle f(\boldsymbol{x}, \boldsymbol{y}, z)\rangle_m(t) =& (1-\eta(t))\langle f(\boldsymbol{x}, \boldsymbol{y}, z)\rangle_m(t-1) \\ &+ \eta(t) E\left[f(\boldsymbol{x}_t, \boldsymbol{y}_t, z_t) \right]. \end{aligned} \quad (4.33)$$

(c) **Obtaining the mean model parameter**:

The expectation of parameter $\boldsymbol{\Theta}$ over the trial posterior distribution, $\boldsymbol{\Theta}^*$, has been obtained by equation (4.31), namely, $\boldsymbol{\Theta}^* = \left(\hat{\boldsymbol{\theta}}_1, \cdots, \hat{\boldsymbol{\theta}}_n\right)^T$ for each $m = 1, \ldots, n$. The principal component dimensionality m is then determined as its MAP estimator:

$$m^* = \arg\max_m \tilde{F}_m^\lambda(t). \quad (4.34)$$

List of Publications

Journal Papers

1. J. Hirayama, S. Maeda and S. Ishii, Markov and semi-Markov switching of source appearances for non-stationary independent component analysis, IEEE Transactions on Neural Networks (to appear).

2. J. Hirayama, J. Yoshimoto and S. Ishii, Balancing plasticity and stability of on-line learning based on hierarchical Bayesian adaptation of forgetting factors. Neurocomputing, 69(16-18), 1954-1961, 2006.

3. J. Hirayama, J. Yoshimoto and S. Ishii, Bayesian representation learning in the cortex regulated by acetylcholine, Neural Networks, 17(10), 1391-1400, 2004.

International Conferences (reviewed)

1. S. Osaga, J. Hirayama, T. Takenouchi, S. Ishii, A probabilistic model of MOSAIC. in IEEE Symposium on Foundations of Computational Intelligence (FOCI'07), (to appear).

2. S. Osaga, J. Hirayama, T. Takenouchi, S. Ishii, A probabilistic modeling of MOSAIC learning. in Artificial Life and Robotics (AROB'07), GS16-3, 2007.

3. J. Hirayama, S. Maeda and S. Ishii, A Bayesian approach to blind source separation with variable number of sources. in Artificial Life and Robotics (AROB'06), GS19-6, 2006.

4. J. Hirayama, S. Maeda and S. Ishii, Bayesian noisy ICA for source switching environments. in IEEE Workshop on Statistical Signal Processing (SSP'05), 232, 2005.

5. J. Hirayama, J. Yoshimoto and S. Ishii, Cortical representation learning regulated by acetylcholine. in Brain Inspired Cognitive Systems (BICS'04), ICESS.3, 2004.

Others

1. 大佐賀 智, 平山 淳一郎, 竹之内 高志, 石井 信. (2006). MOSAIC の確率モデル. 日本神経回路学会 第 16 全国大会, pp.92-93.

2. 平山 淳一郎, 前田 新一, 石井 信. (2005). 変動する環境下での独立成分分析と変分ベイズ法による解法. 第 8 回情報論的学習理論ワークショップ (IBIS2005), pp.263-268.

3. 鈴木 郁美, 大羽 成征, 平山 淳一郎, 石井 信. (2005). テスト性能の分散を考慮したロバストな判別器の選択基準. 信学技報, NLP2005-66, NC2005-58, pp.25-30.

4. 平山 淳一郎, 前田 新一, 石井 信. (2005). Bayesian noisy ICA for source switching environments. 電子情報通信学会技術研究報告, NC2004-224, pp.183-187.

5. 平山 淳一郎, 吉本 潤一郎, 石井 信. (2004). 環境変動に追従した基底の獲得と神経修飾物質の機能. 脳と心のメカニズム第 4 回冬のワークショップ.

6. 平山 淳一郎, 吉本 潤一郎, 石井 信. (2004). 動的環境における脳内情報表現の学習と神経修飾. 電子情報通信学会技術研究報告, NC2003-210, pp.97-102.

7. 平山 淳一郎, 吉本 潤一郎, 石井 信. (2003). 神経修飾物質による皮質の学習制御: オンライン変分ベイズ法によるモデル化. 日本神経回路学会第 13 回全国大会, pp.38-39.