Balancing plasticity and stability of on-line learning based on hierarchical Bayesian adaptation of forgetting factors

Junichiro Hirayama[1], Junichiro Yoshimoto[2,1] and Shin Ishii[1]
{junich-h, juniti-y, ishii}@is.naist.jp
[1] Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
[2] Initial Research Project, Okinawa Institute of Science and Technology

Abstract

An important character of on-line learning is its potential to adapt to changing environments by properly adjusting meta-parameters that control the balance between plasticity and stability of the learning model. In our previous study, we proposed a learning scheme that address changing environments in the framework of an on-line variational Bayes (VB), which is an effective on-line learning scheme based on Bayesian inference. The motivation of that work was, however, its implications for animal learning, and the formulation of the learning model was heuristic and not theoretically justified. In this article, we propose a new approach that balances the plasticity and stability of on-line VB learning in a more theoretically justifiable manner by employing the principle of hierarchical Bayesian inference. We present a new interpretation of on-line VB as a special case of incremental Bayes that allows the hierarchical Bayesian setting to balance the plasticity and stability as well as yielding a simple learning rule compared to standard on-line VB. This dynamic on-line VB scheme is applied to probabilistic PCA as an example of probabilistic models involving latent variables. In computer simulations using artificial datasets, the new on-line VB learning shows robust performance to regulate the balance between plasticity and stability, thus adapting to changing environments.

1 Introduction

For a short time, external environments surrounding animals can be regarded as static, but they are dynamic over long periods. To adapt to such environments, animals must quickly learn about novel stimuli when the environment changes. In our previous study [12], we proposed a theoretical model of such a dynamic learning scheme possibly realized in animal brains, within the framework of on-line learning. That work, motivated by recent physiological findings that indicate the highly adaptive nature of cortical representation [8, 9, 5, 4], was intended to illustrate the key role of a neuromodulator acetylcholine (ACh) [11, 7] to control representational plasticity. That proposed on-line learning scheme worked well even when the environment dynamically changed, and we discussed the functional role of ACh in relation to the results. However, the formulation of the learning model was heuristic and not theoretically justified. In this article, apart from the implications to cortical learning but still motivated by it, we propose a new approach to dynamic on-line learning in a more theoretically justifiable manner that stands on an engineering viewpoint.

Sato [21] proposed an on-line variational Bayes (VB) method that is an effective on-line learning scheme based on Bayesian inference. A 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 [3, 18], which were originally developed as a batch-type learning scheme, provide an effective approximation. An on-line VB method is an alternative to standard VB in on-line learning scenarios in which the learning model attempts to adapt to new inputs incrementally without retaining the series of past inputs, while batch learning is executed after all the inputs

are given and retaining the past inputs in the memory. On-line 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. Beyond such basic advantages, an important character of on-line learning 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 on-line learning [1, 6, 23, 17, 22, 16], no study has paid special attention to on-line VB learning except for our previous study [12].

Dynamic control between plasticity and stability in our previous study was realized by two aspects: novelty detection that assumed an explicit model of novel inputs (or outliers), and novelty-based scheduling of a forgetting factor, a meta-parameter that modulates the weights of past inference. In the standard formulation of the on-line VB method, the expected sufficient statistics are explicitly maintained and incrementally updated according to a new datum; then the forgetting factor implicitly regulates the updating speed, determining the balance between the plasticity and stability. Novelty-based scheduling of the forgetting factor thus allowed an adaptive plasticity-stability control in the previous model by adjusting the size of forgetting factor from relatively small for novel input and large for familiar input. Since this balancing scheme was a heuristic, however, there was no theoretical justification for it.

In this article, we propose a more theoretically justifiable scheme to control the forgetting factor in on-line VB learning within the framework of hierarchical Bayesian inference. We illustrate that the procedure of on-line VB can be interpreted as a special case of incremental Bayes. Based on this interpretation, we then present a new hierarchical Bayesian way to adaptively schedule the forgetting factor. In addition, the learning rule in this article is formulated as a direct updating of hyperparameters in approximate posterior distribution without explicitly maintaining the expected sufficient statistics as in the standard setting. We apply this scheme to a probabilistic principal component analysis (PPCA) model [24] as an example of probabilistic models involving latent variables, which we also used in the previous study. This new hierarchical Bayesian approach is validated through computer simulations using artificial datasets.

2 Model

2.1 Probabilistic PCA

PPCA is a probabilistic generative model with latent variables, such that its maximum likelihood (ML) estimation is equivalent to standard PCA [24]. PPCA for an n-dimensional observed variable $\boldsymbol{x}_t \in \Re^n$ is given by

$$\boldsymbol{x}_t = \boldsymbol{\Theta}\boldsymbol{y}_t + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}_n\left(\boldsymbol{\xi}_t \mid \boldsymbol{0}, \sigma_x^2 \boldsymbol{I}_n\right), \tag{1}$$

where t denotes the discrete time or the sample index. $\boldsymbol{y}_t \equiv (y_{t,1}, \cdots, y_{t,m})' \in \Re^m$ $(m \leq n)$ is a latent variable corresponding to the principal component score, which is generated independently at each time step from a standard Gaussian distribution. Prime $(')$ denotes the transpose. $\boldsymbol{\xi}_t \in \Re^n$ is 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 that is assumed to be a known constant for simplicity. $\boldsymbol{\Theta} \equiv (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_m) \in \Re^{n \times m}$ is the principal component loading matrix, where $\boldsymbol{\theta}_j \in \Re^n (j = 1, \cdots, m)$ is the principal component vector. For simplicity, the observations are assumed to be normalized to have a zero mean.

[1] These terms follow Grossberg's "plasticity/stability dilemma" [10].

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

2.2 On-line Variational Bayes learning

Model parameter $\boldsymbol{\Theta}$ can be inferred using the on-line VB method [21]. Let $(X_{1:t}, Y_{1:t}) \equiv \{(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau) \mid \tau = 1, \ldots, t\}$ be a series of observations and corresponding latent variables. The objective of 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 on-line variational free energy with a time-dependent forgetting factor $\lambda(s) \in [0, 1]$ $(s = 1, \ldots, t)$ is defined by

$$F^\lambda[q](t) = T^\lambda(t) L^\lambda(t) - H(t) \tag{2a}$$

$$L^\lambda(t) = \eta(t) \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda(s) \right) E\left[\log \frac{p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta})}{q_\tau(\boldsymbol{y}_\tau | \boldsymbol{x}_\tau)} \right] \tag{2b}$$

$$H(t) = E\left[\log \frac{q_\theta(\boldsymbol{\Theta} \mid X_{1:t})}{p_0(\boldsymbol{\Theta})} \right], \tag{2c}$$

where $q(Y_{1:t}, \boldsymbol{\Theta} \mid X_{1:t}) \equiv q_\theta(\boldsymbol{\Theta} \mid X_{1:t}) \prod_{\tau=1}^{t} q_\tau(\boldsymbol{y}_\tau \mid \boldsymbol{x}_\tau)$ denotes a factorized trial distribution to approximate the true posterior distribution $p(Y_{1:t}, \boldsymbol{\Theta} \mid X_{1:t})$, and $E[\cdot]$ denotes expectation over trial distribution q. $T^\lambda(t) \equiv \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda(s) \right)$ is an effective data number and $\eta(t) \equiv 1/T^\lambda(t)$ is the normalization term called the learning rate [21]. Furthermore, $p_0(\boldsymbol{\Theta})$ is the prior distribution of $\boldsymbol{\Theta}$ defined below. The on-line VB method for PPCA is derived as a sequential maximization process of the variational free energy (2). When a datum $\boldsymbol{x}_t$ is observed at time t, F^λ is maximized with respect to q_t in the on-line VB-E step while q_τ $(\tau = 1, \ldots, t-1)$ and q_θ are fixed. In the next step called the on-line VB-M step, F^λ is maximized with respect to q_θ while q_τ $(\tau = 1, \ldots, t)$ 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:

$$q_t(\boldsymbol{y}_t \mid \boldsymbol{x}_t) = \frac{\exp\left(E_{\boldsymbol{\Theta}}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta}) \right] \right)}{\int d\boldsymbol{y}_t \exp\left(E_{\boldsymbol{\Theta}}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta}) \right] \right)}, \tag{3a}$$

$$q_\theta(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\left(\sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda(s) \right) E_{\boldsymbol{y}_\tau}\left[\log p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta}) \right] \right) p_0(\boldsymbol{\Theta})}{\int d\boldsymbol{\Theta} \exp\left(\sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda(s) \right) E_{\boldsymbol{y}_\tau}\left[\log p(\boldsymbol{x}_\tau, \boldsymbol{y}_\tau \mid \boldsymbol{\Theta}) \right] \right) p_0(\boldsymbol{\Theta})}, \tag{3b}$$

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. For the parameters, we use a conjugate prior:

$$p_0(\boldsymbol{\Theta}) = \mathrm{N}_{n \times m}\left(\boldsymbol{\Theta} \mid \boldsymbol{M}_0, \boldsymbol{I}_n, \boldsymbol{G}_0^{-1} \right), \tag{4}$$

where $\mathrm{N}_{n \times m}(\cdot \mid \cdot, \cdot, \cdot)$ denotes a matrix normal distribution[3].

2.3 Recursive learning rule

Instead of directly calculating the VB-M step equation (3b), here we present a recursive updating equation of q_θ, which is obtained as (see Appendix A)

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp(E_{\boldsymbol{y}_t}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta}) \right]) \tilde{q}_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}; \lambda(t))}{\int d\boldsymbol{\Theta} \exp(E_{\boldsymbol{y}_t}\left[\log p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \boldsymbol{\Theta}) \right]) \tilde{q}_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}; \lambda(t))}, \tag{5}$$

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

$$\tilde{q}_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}; \lambda(t)) \propto q_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1})^{\lambda(t)} p_0(\boldsymbol{\Theta})^{1-\lambda(t)}, \tag{6}$$

where the normalization term is omitted. Superscript $^{(\tau)}$ denotes that trial distribution is maximized using observations available at time τ, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\tau$. Eq. (5) suggests that the on-line 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.

[3] $\mathrm{N}_{n \times m}(\boldsymbol{A} \mid \boldsymbol{M}, \boldsymbol{V}, \boldsymbol{K}) = (2\pi)^{-nm/2} |\boldsymbol{K}|^{-n/2} |\boldsymbol{V}|^{-m/2} \exp\left(-\frac{1}{2} \mathrm{tr}\left[(\boldsymbol{A} - \boldsymbol{M})' \boldsymbol{V}^{-1} (\boldsymbol{A} - \boldsymbol{M}) \boldsymbol{K}^{-1} \right] \right)$, where $\boldsymbol{A} \in \Re^{n \times m}$, $\boldsymbol{M} \in \Re^{n \times m}$, $\boldsymbol{K} \in \Re^{m \times m}$, and $\boldsymbol{V} \in \Re^{n \times n}$. $\boldsymbol{M}$ denotes the mean of $\boldsymbol{A}$; $\boldsymbol{K}$ and $\boldsymbol{V}$ are two covariance matrices of $\boldsymbol{A}$ [15].

After a new observation $\boldsymbol{x}_t$ is given at time t, Eq. (5) 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. (5) and the conventional one are: in Eq. (5), 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.

In the case of PPCA, the recursive update of the trial distribution, Eq. (5), is performed by updating only two hyperparameters, since the trial distribution, Eq. (3b), is obtained as a Gaussian. Now let $q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \mathrm{N}_{n\times m}(\boldsymbol{\Theta} \mid \hat{\boldsymbol{M}}_t, \boldsymbol{I}_n, \hat{\boldsymbol{G}}_t^{-1})$, and then the learning rule is derived as

$$\hat{\boldsymbol{G}}_t = \sigma_x^{-2} \langle \boldsymbol{y}_t \boldsymbol{y}_t' \rangle + \lambda(t) \hat{\boldsymbol{G}}_{t-1} + (1-\lambda(t)) \boldsymbol{G}_0, \tag{7a}$$

$$\hat{\boldsymbol{M}}_t = \hat{\boldsymbol{M}}_{t-1} + \left(\sigma_x^{-2} \boldsymbol{x}_t \langle \boldsymbol{y}_t \rangle' + (1-\lambda(t)) \boldsymbol{M}_0 \boldsymbol{G}_0 - \hat{\boldsymbol{M}}_{t-1} \left(\sigma_x^{-2} \langle \boldsymbol{y}_t \boldsymbol{y}_t' \rangle + (1-\lambda(t)) \boldsymbol{G}_0 \right) \right) \hat{\boldsymbol{G}}_t^{-1}, \tag{7b}$$

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 on-line maintenance of the expected sufficient statistics in our previous study [12].

2.4 Forgetting factor adaptation based on the hierarchical Bayes

Forgetting factor $\lambda(t)$ controls the balance between plasticity and stability of on-line VB learning. In a dynamic environment, $\lambda(t)$ should be small (≈ 0), especially when the environment changes, while $\lambda(t)$ should be large (≈ 1) during stationary periods. One possible way to achieve such adaptive scheduling of $\lambda(t)$ is to explicitly use an outlier component in a mixture model prepared for identifying dynamic environments. $\lambda(t)$ is then determined based on the posterior probability of the outlier component [12].

Unlike the previous study, however, in this study we propose a hierarchical Bayesian method to schedule $\lambda(t)$, utilizing the above illustration of the on-line VB learning. According to Eq. (5), $\lambda(t)$ 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. (5), 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. (6) 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

$$p(\lambda(t) \mid \boldsymbol{x}_t) = \frac{L(\boldsymbol{x}_t, \lambda(t)) p(\lambda(t))}{\int_0^1 d\lambda(t)\, L(\boldsymbol{x}_t, \lambda(t)) p(\lambda(t))}, \tag{8}$$

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:

$$\hat{\lambda}(t) = \int_0^1 d\lambda(t)\, p(\lambda(t) \mid \boldsymbol{x}_t) \lambda(t). \tag{9}$$

Practically, however, it is not so easy to calculate the integrals that appeared in Eqs. (8) and (9). In addition, the evaluation of marginal likelihood $L(\boldsymbol{x}_t, \lambda(t))$ also involves intractable integral in calculating the normalization constant of Eq. (6) 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

$$\hat{\lambda}(t) = \frac{\int_0^1 d\lambda(t) L(\boldsymbol{x}_t, \lambda(t)) p(\lambda(t)) \lambda(t)}{\int_0^1 d\lambda(t) L(\boldsymbol{x}_t, \lambda(t)) p(\lambda(t))} \approx \frac{\sum_{\lambda(t)\in\{0,1\}} L(\boldsymbol{x}_t, \lambda(t)) p(\lambda(t)) \lambda(t)}{\sum_{\lambda(t)\in\{0,1\}} L(\boldsymbol{x}_t, \lambda(t)) p(\lambda(t))}. \tag{10}$$

3 Simulations

3.1 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. (1) with $m = 1$ and $\sigma_x = 1$. The number of observations was $T = 600$. True parameter $\boldsymbol{\Theta}$ was fixed in a short time period but occasionally changed as follows: $\boldsymbol{\Theta} = (5, -1)'$ for $t = 1, \ldots, 200$, $(1, 5)'$ for $t = 201, \ldots, 400$, and $(-3, 3)'$ for $t = 401, \ldots, 600$. Prior hyperparameters $\boldsymbol{M}_0$ and $\boldsymbol{G}_0$ were set as $\boldsymbol{M}_0 = \boldsymbol{0}$ 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 1 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 $\boldsymbol{\Theta}$ 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 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 on-line VB learning, the simulation was repeated for 100 runs. The observed dataset for each run was generated by Eq. (1) individually with a random seed number. Figure 2 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 shows the value of the forgetting factor averaged over 100 runs scheduled by our hierarchical Bayesian scheme.

3.2 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 4 as principal component vectors ($n = 25$), and generated 200 observations for each original image according to Eq. (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 5. In this simulation, prior hyperparameters were set as $\boldsymbol{M}_0 = \boldsymbol{0}$ 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 6 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 7 shows estimation error (top panel) and the forgetting factor (bottom panel) averaged over 100 runs.

4 Discussion

We proposed a new balancing scheme between plasticity and stability of on-line VB learning to address feature extraction in dynamic environments. A key to this scheme is the dynamic scheduling of the forgetting factor $\lambda(t)$, as in our previous study [12], while our new scheduling scheme is theoretically justified as a hierarchical Bayesian inference. This could be done by utilizing the new view of the on-line VB method as an incremental Bayes, which is one of the contributions of this study. In this view, the learning model no longer explicitly needs the on-line maintenance of expected sufficient statistics, as in standard on-line VB methods. In addition, the models do not require explicit usage of an outlier component for novelty detection, as assumed in our previous study. Our new hierarchical Bayesian scheme naturally combines novelty detection and the adaptive scheduling of $\lambda(t)$ according to the novelty information in a principled way. Although the integration required for the hierarchical Bayesian estimation of $\lambda(t)$ was intractable and thus approximated simply by summation, simulations showed that the proposed scheme still works well. In simulations using artificial datasets, the new learning model was able to quickly and robustly follow the abrupt changes of input statistics to be accommodated to the new inputs, while the model parameters were improved in stationary periods.

The illustration of the on-line VB method as an incremental Bayesian inference, Eq. (5), is not limited to the case of PPCA. The incremental Bayes update in Eq. (5) 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. (7), 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) [13, 2, 14] and sparse coding [19, 20]. In a future study, the learning rule, Eq. (7), 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 on-line VB-M step, however, Eq. (5) 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.

A On-line VB as an incremental Bayes update

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 $\Psi_t(X_{1:t}, \boldsymbol{\Theta}) \equiv \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda(s)\right) \psi_\tau(\boldsymbol{x}_\tau, \boldsymbol{\Theta})$. Eq. (3b) is then written as

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\big(\Psi_t(X_{1:t}, \boldsymbol{\Theta})\big) p_0(\boldsymbol{\Theta})}{\int d\boldsymbol{\Theta} \exp\big(\Psi_t(X_{1:t}, \boldsymbol{\Theta})\big) p_0(\boldsymbol{\Theta})}. \tag{11}$$

Also, the trial distribution at previous time step $t-1$ is given by

$$q_\theta^{(t-1)}(\boldsymbol{\Theta} \mid X_{1:t-1}) = \frac{\exp\big(\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta})\big) p_0(\boldsymbol{\Theta})}{\int d\boldsymbol{\Theta} \exp\big(\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta})\big) p_0(\boldsymbol{\Theta})}. \tag{12}$$

We here try to obtain a new estimate at time step t, Eq. (11), by only using the previous estimate given by Eq. (12), prior distribution p_0, and forgetting factor $\lambda(t)$. Note that the on-line VB-E step performed between the two maximization steps, Eq. (12) at time $t-1$ and Eq. (11) at time t, is temporally localized so that it does not change the past inference of the latent variable and the forgetting factor; trial distribution $q_\tau(\boldsymbol{y}_\tau)\,(\tau = 1, \ldots, t-1)$ and forgetting factor $\lambda(s)\,(s = 1, \ldots, t-1)$ are fixed at time t. Eq. (12) thus still holds after the on-line VB-E step at time t, because term $\Psi_{t-1}(X_{1:t-1}, \boldsymbol{\Theta})$, which includes the expectations with respect to $q_\tau(\boldsymbol{y}_\tau)\,(\tau = 1, \ldots, t-1)$, does not change through the new inference at time step t.

The logarithm of the numerator of Eq. (12) is given by

$$\Psi_t(X_{1:t},\boldsymbol{\Theta}) + \log p_0(\boldsymbol{\Theta}) = \sum_{\tau=1}^{t}\left(\prod_{s=\tau+1}^{t}\lambda\left(s\right)\right)\psi_\tau(\boldsymbol{x}_\tau,\boldsymbol{\Theta}) + \log p_0(\boldsymbol{\Theta}) \tag{13a}$$

$$= \psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right) + \lambda(t)\sum_{\tau=1}^{t-1}\left(\prod_{s=\tau+1}^{t-1}\lambda\left(s\right)\right)\psi_\tau\left(\boldsymbol{x}_\tau,\boldsymbol{\Theta}\right) + \log p_0\left(\boldsymbol{\Theta}\right) \tag{13b}$$

$$= \psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right) + \lambda(t)\Psi_{t-1}\left(X_{1:t-1},\boldsymbol{\Theta}\right) + \log p_0\left(\boldsymbol{\Theta}\right) \tag{13c}$$

$$= \psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right) + \lambda(t)\Big(\Psi_{t-1}\left(X_{1:t-1},\boldsymbol{\Theta}\right) + \log p_0\left(\boldsymbol{\Theta}\right)\Big) + \left(1-\lambda(t)\right)\log p_0\left(\boldsymbol{\Theta}\right). \tag{13d}$$

Then, Eq. (12) is written as

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\left(\psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right)\right)\Big(\exp\left(\Psi_{t-1}\left(X_{1:t-1},\boldsymbol{\Theta}\right)\right)p_0\left(\boldsymbol{\Theta}\right)\Big)^{\lambda(t)}p_0\left(\boldsymbol{\Theta}\right)^{1-\lambda(t)}}{\int d\boldsymbol{\Theta}\exp\left(\psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right)\right)\Big(\exp\left(\Psi_{t-1}\left(X_{1:t-1},\boldsymbol{\Theta}\right)\right)p_0\left(\boldsymbol{\Theta}\right)\Big)^{\lambda(t)}p_0\left(\boldsymbol{\Theta}\right)^{1-\lambda(t)}}. \tag{14}$$

The denominator of Eq. (12) does not depend on $\boldsymbol{\Theta}$, and then

$$q_\theta^{(t)}(\boldsymbol{\Theta} \mid X_{1:t}) = \frac{\exp\left(\psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right)\right)q_\theta^{(t-1)}\left(\boldsymbol{\Theta} \mid X_{1:t-1}\right)^{\lambda(t)}p_0\left(\boldsymbol{\Theta}\right)^{1-\lambda(t)}}{\int d\boldsymbol{\Theta}\exp\left(\psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right)\right)q_\theta^{(t-1)}\left(\boldsymbol{\Theta} \mid X_{1:t-1}\right)^{\lambda(t)}p_0\left(\boldsymbol{\Theta}\right)^{1-\lambda(t)}} \tag{15a}$$

$$= \frac{\exp\left(\psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right)\right)\tilde{q}_\theta^{(t-1)}\left(\boldsymbol{\Theta} \mid X_{1:t-1};\lambda(t)\right)}{\int d\boldsymbol{\Theta}\exp\left(\psi_t\left(\boldsymbol{x}_t,\boldsymbol{\Theta}\right)\right)\tilde{q}_\theta^{(t-1)}\left(\boldsymbol{\Theta} \mid X_{1:t-1};\lambda(t)\right)}. \tag{15b}$$

This is identical to the updating equation, Eq. (5), which is a special case of the incremental Bayes.

References

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

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

[3] 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.

[4] 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.

[5] M. B. Calford. Dynamic representational plasticity in sensory cortex. *Neuroscience*, 111(4):709–738, 2002.

[6] 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.

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

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

[9] C. D. Gilbert, M. Sigman, and R. E. Crist. The neural basis of perceptual learning. *Neuron*, 31:681–697, 2001.

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

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

[12] J. Hirayama, J. Yoshimoto, and S. Ishii. Bayesian representation learning in the cortex regulated by acetylcholine. *Neural Networks*, 17:1391–1400, 2004.

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

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

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

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] 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.

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

Figures

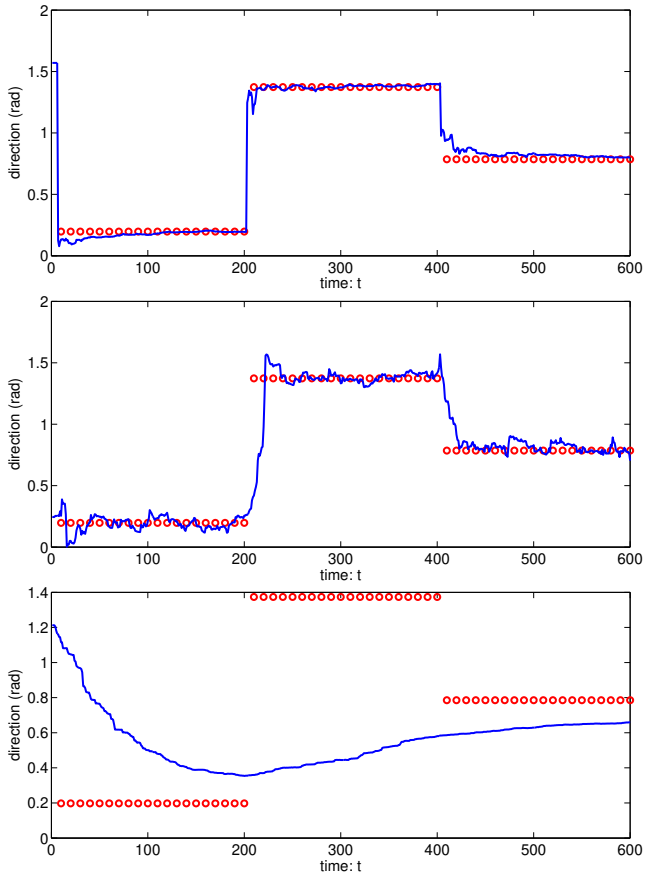

Figure 1:

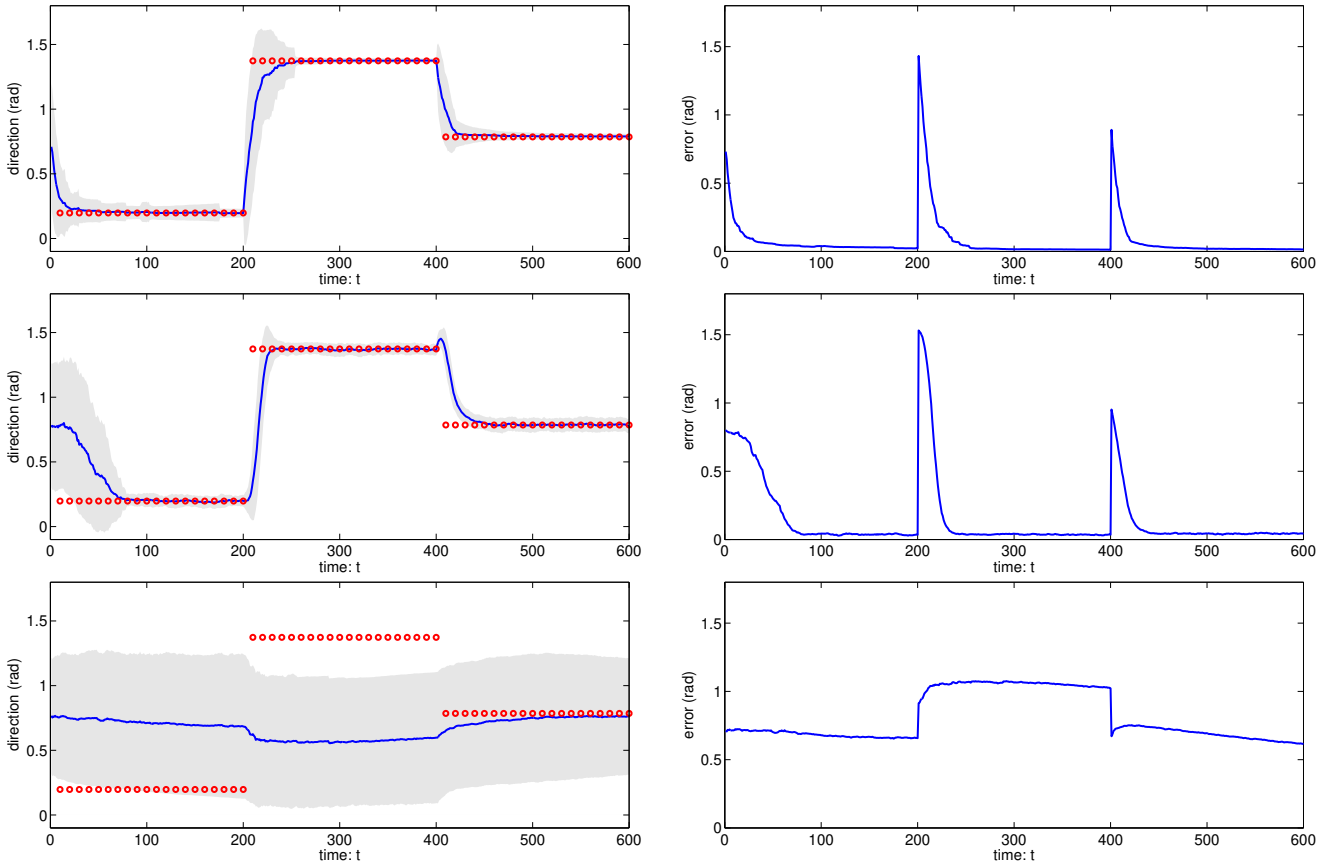

Figure 2:

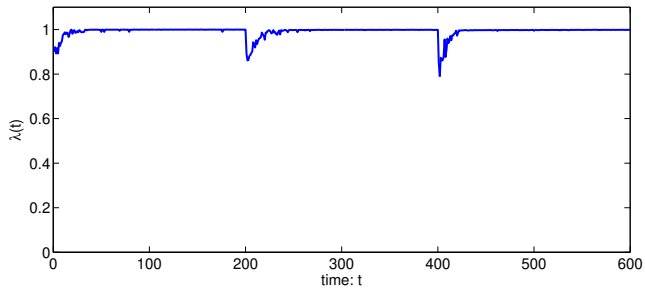

Figure 3:

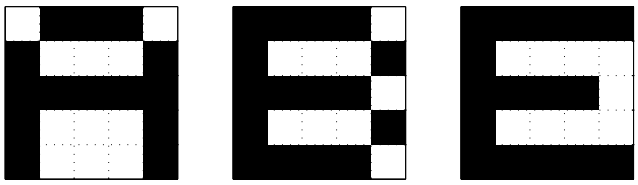

Figure 4:

Figure 5:

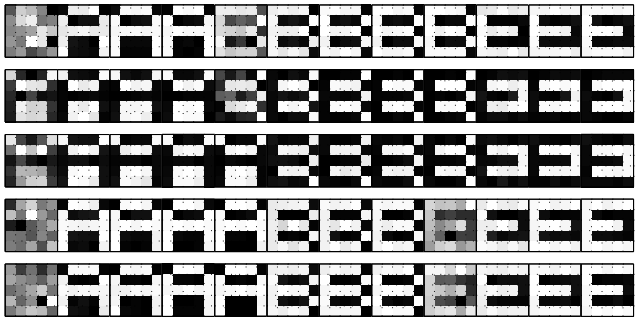

Figure 6:

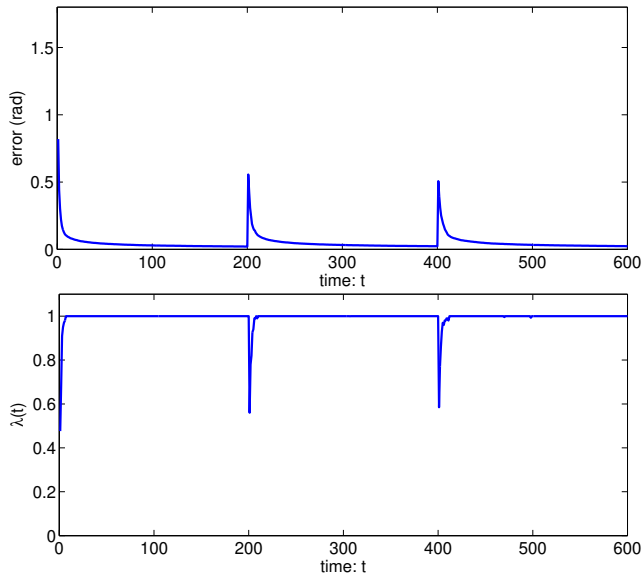

Figure 7:

Figure Legends

Figure 1 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. '$\circ$' represents the real value in each time step.

Figure 2 Direction of estimated principal component vector (left column) and estimation error, i.e., the angle between estimated vector and the true one (right column). We 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).

Figure 3 Forgetting factor $\lambda(t)$ averaged over same 100 runs as in Figure 2.

Figure 4 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. (1), to generate artificial datasets.

Figure 5 Example observations in learning from time steps $1, 51, 101, \ldots, 551$ from right to left.

Figure 6 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 7 Estimation error (top panel) and forgetting factor (bottom panel) averaged over 100 runs.