ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746538

Tracking the Dimensions of Latent Spaces of Gaussian Process Latent Variable Models

Yuhao Liu
Department of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY 11794
yuhao.liu.1@stonybrook.edu

Petar M. Djurić
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794
petar.djuric@stonybrook.edu

***Abstract*—Determining the number of latent variables, or the dimensions of latent states, is a ubiquitous problem in dimension reduction. In this paper, we introduce a novel sequential method that relies on the Bayesian approach to estimate the dimension of a latent space of a Gaussian process latent variable model. The proposed method also considers settings where the number of latent variables varies with time. To evaluate our methodology, we compared the estimated dimensions with the true dimensions as they vary with time. Results on synthetic data demonstrate that our method has a very good performance.**

***Index Terms*—Latent Variables, Dimension reduction, Gaussian processes**

I. Introduction

Dimensionality reduction (DR) refers to techniques for projecting high dimensional observed samples to a lower dimensional set of latent input variables in a way that the low dimensional variables preserve important properties of the high dimensional observed samples. The primary problem here is to determine the number of the latent variables, or the dimension of latent states. This problem was initially addressed in [3], which has motivated many researchers to study it. Several most popular methods are documented and described in [9]. One of them, Kaiser's eigenvalue-greater-than-one rule (K1) only keeps the factors that have eigenvalues greater than a benchmark [5]. Despite its simplicity, this work has received criticism because of its unreliability. Parallel Analysis (PA) is another approach that bootstraps on the correlation matrix and then averages the eigenvalues [10]. By this method, the factors that have eigenvalues larger than the eigenvalue of the averaged data set are kept. The Cattell's Scree test [12] is a graphical and subjective approach, which first sorts the eigenvalues in decreasing order and then picks the point where the last significant drop occurs. The minimum average partial method (MAP) is based on a subsequent analysis of the partial correlation matrices in extracting common factors [19]. Silouhette width (SW) is another method that reduces the number of data points [7]. The Bootstrap SVD utilizes both the bootstrap and the singular values to determine the lower dimension [11]. There are also log-likelihood based criteria including ones based on the Akaike information criterion [1]. A common feature of all these methods is that they operate in an offline mode and assume that the number of latent variables is the same for all the observed samples.

The authors thank the support of NSF under Award 2021002.

In order to reduce the dimension in a sequential or online mode, the online principal component analysis (PCA) has been invoked in [2] and implemented by perturbation methods [4], incremental SVD [18], stochastic optimization [14], and randomized algorithms [17]. The probability PCA (PPCA) is an extension of the PCA and assumes a linear relationship between the observations and the latent states. Compared with PPCA, the Gaussian process latent variable model (GPLVM) is more powerful and compatible with nonlinear relationships. The online GPLVM based on random features has been proposed in [6], and the sequential sampling method in [15] to retrieve the latent variables sequentially with variational inference. However, the random feature-based Gaussian processes is limited to have only shift-invariant kernels, and the variational inference combined with the Monte Carlo method produces a bias that cannot be ignored. Moreover, these methods still assume a constant dimension of the latent states and are not compatible with a time-varying system.

To overcome these limitations, we propose a sequential sparse GPLVM (SSGPLVM), which can simultaneously determine the proper dynamic dimension of the latent states and obtain the latent states sequentially.

II. Preliminary

In this section, we provide a brief review of GPLVMs and their sparse version.

A. Gaussian Process Latent Variable Models

Under ordinary PCA [16], we assume that d_y-dimensional observations $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times d_y}$ are a linear function of $\mathrm{d_t}$-dimensional latent variables $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times \mathrm{d_t}}$ with likelihood

$$p(\mathbf{y}_n|\mathbf{W}, \beta) = \int p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \beta) p(\mathbf{x}_n) d\mathbf{x}_n, \tag{1}$$

where $p(\mathbf{x}_n)$ is a Gaussian prior distribution $\mathcal{N}(\mathbf{x}_n|0, \mathbf{I})$, and where $p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I})$ reflects the linear relationship between the latent variables and the observations. Further, we assume that the rows of $\mathbf{W}$ are i.i.d., which implies

the prior $p(\mathbf{W}) = \prod_{i=1}^{d_y} \mathcal{N}(0, \alpha^{-1}\mathbf{I})$. When $\mathbf{W}$ is integrated out, we obtain a marginal distribution for $\mathbf{Y}$ given by

$$p(\mathbf{Y}|\mathbf{X}, \beta) = \frac{1}{(2\pi)^{\frac{N\times d_y}{2}}|\mathbf{K}|^{\frac{d_y}{2}}} \exp\left(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{-1})\right), \quad (2)$$

where $\mathbf{K} = \alpha\mathbf{X}\mathbf{X}^\top + \beta^{-1}\mathbf{I}$. Then we can obtain the optimized $\mathbf{X}$ and the hyper-parameters α, β by maximizing (2).

The GPLVM comes from various modifications of $\mathbf{K}$ [8]. A Gaussian process [13] is defined by a mean function $m(\cdot)$ and a kernel function $k(\cdot,\cdot)$, such that any finite subset of points of the process is Gaussian distributed. A function f that follows a Gaussian process is denoted by

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (3)$$

By choosing a specific nonlinear kernel function $k(\cdot,\cdot|\boldsymbol{\gamma})$ instead of $\mathbf{X}\mathbf{X}^\top$, where $\boldsymbol{\gamma}$ is a hyper-parameter vector in the kernel, we put a nonlinear prior relationship between the latent variable $\mathbf{X}$ and the observations $\mathbf{Y}$. In this paper, we choose the RBF kernel. The equation (2) can still be applied in terms of a nonlinear kernel function, and the latent variable $\mathbf{X}$ and hyper-parameters $\alpha, \beta, \boldsymbol{\gamma}$ are optimized by scaled conjugate gradient (SCG).

B. Sparse Gaussian Process Latent Variable Models

The idea behind sparse GPs is in utilizing a small set of points, which is known as inducing variables and denoted as $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M] = [\mathbf{y}_1, \ldots, \mathbf{y}_M]$. The optimized latent states related to $\mathbf{U}$ are denoted as $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M]$.

The latent state $\mathbf{x}_j$ under an observation point $\mathbf{y}_j$ can be derived from the inducing variables as a Gaussian distribution

$$p(\mathbf{y}_j|\mathbf{x}_j, \alpha, \beta, \gamma) = \mathcal{N}(\mathbf{y}_j|\mathbf{f}_j, \sigma_j^2\mathbf{I}), \quad (4)$$

where $\mathbf{f}_j = \mathbf{U}^\top\mathbf{K}_{\boldsymbol{\Lambda},\boldsymbol{\Lambda}}^{-1}\mathbf{k}_{\boldsymbol{\Lambda},j}$ is the mean vector, $\mathbf{k}_{\boldsymbol{\Lambda},j}$ denotes the kernel vector between the optimized latent states $\boldsymbol{\Lambda}$ and the latent state $\mathbf{x}_j$, and $\sigma_j^2 = k(\mathbf{x}_j, \mathbf{x}_j) - \mathbf{k}_{\boldsymbol{\Lambda},j}^\top\mathbf{K}_{\boldsymbol{\Lambda},\boldsymbol{\Lambda}}^{-1}\mathbf{k}_{\boldsymbol{\Lambda},j}$ is the variance. As a result, we can optimize (4) with respect to $\mathbf{x}_j$ using the scaled conjugate gradients approach. Thus, a sparse GPLVM is determined by its inducing variables $\mathbf{U}$.

III. Choices of Dimensions of the Latent States

We adopt the Bayesian approach to determine the dimension of the latent states. However, in practice, there are settings where the observed data are acquired sequentially and the dimension of the latent states may vary with time. In the rest of the paper, we present our approaches under constant and dynamic dimensions separately.

A. A Constant Dimension

Suppose we know the information that all observations are generated from latent states with a constant dimension. Given a set of candidate dimensions $\mathcal{D}$, we denote the inducing variables as $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M]^\top$ and the corresponding optimized latent states as $\{\boldsymbol{\Lambda}^{(\mathrm{d})}\}_{\mathrm{d}\in\mathcal{D}} = \{[\boldsymbol{\lambda}_1^{(\mathrm{d})}, \ldots, \boldsymbol{\lambda}_M^{(\mathrm{d})}]^\top\}_{\mathrm{d}\in\mathcal{D}}$, where the superscript $^{(\mathrm{d})}$ refers to the specific latent dimension d. The history of observations is denoted as $\mathbf{Y}_{t-1} = [\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}]^\top$, and the corresponding history of latent states are denoted as $\{\mathbf{X}_{t-1}^{(\mathrm{d})}\}_{\mathrm{d}\in\mathcal{D}} = \{[\mathbf{x}_1^{(\mathrm{d})}, \ldots, \mathbf{x}_{t-1}^{(\mathrm{d})}]^\top\}_{\mathrm{d}\in\mathcal{D}}$. The estimate of the latent state and latent dimension at time t are denoted by $(\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}, \widehat{\mathrm{d}}_t)$ and are derived by

$$\begin{aligned}
(\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}, \widehat{\mathrm{d}}_t) &= \underset{\mathbf{x}_t^{(\mathrm{d}_t)}, \mathrm{d}_t}{\mathrm{argmax}}\, p(\mathbf{x}_t^{(\mathrm{d}_t)}, \mathrm{d}_t|\mathbf{y}_t, \mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}}) \quad (5)\\
&= \underset{\mathbf{x}_t^{(\mathrm{d}_t)}, \mathrm{d}_t}{\mathrm{argmax}}\; p(\mathbf{y}_t|\mathbf{Y}_{t-1}, \mathbf{X}_{t-1}^{(\mathrm{d}_t)}, \mathrm{d}_t, \mathbf{x}_t^{(\mathrm{d}_t)})\\
&\quad \times p(\mathbf{x}_t^{(\mathrm{d}_t)}|\mathrm{d}_t, \mathbf{Y}_{t-1}, \mathbf{X}_{t-1}^{(\mathrm{d}_t)}) \times p(\mathrm{d}_t|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}})\\
&= \underset{\mathbf{x}_t^{(\mathrm{d}_t)}, \mathrm{d}_t}{\mathrm{argmax}}\; p(\mathbf{y}_t|\mathbf{U}, \boldsymbol{\Lambda}^{(\mathrm{d}_t)}, \mathrm{d}_t, \mathbf{x}_t^{(\mathrm{d}_t)})\\
&\quad \times p(\mathbf{x}_t^{(\mathrm{d}_t)}|\mathrm{d}_t) \times p(\mathrm{d}_t|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}}),
\end{aligned}$$

where we assume that $p(\mathbf{x}_t^{(\mathrm{d}_t)}|\mathrm{d}_t, \mathbf{Y}_{t-1}, \mathbf{X}_{t-1}^{(\mathrm{d}_t)}) = p(\mathbf{x}_t^{(\mathrm{d}_t)}|\mathrm{d}_t)$ is the prior distribution under dimension d_t, and the last term $p(\mathrm{d}_t|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}})$ denotes the posterior probability of the GPLVM according to latent dimension d_{t-1} at time $t-1$. The optimization process could be decomposed to two steps:

$$\begin{aligned}
\widehat{\mathbf{x}}_t^{(\mathrm{d}_t)} &= \underset{\mathbf{x}_t}{\mathrm{argmax}}\; l(\mathbf{x}_t) \quad (6)\\
&:= \underset{\mathbf{x}_t}{\mathrm{argmax}}\; p(\mathbf{y}_t|\mathbf{U}, \boldsymbol{\Lambda}^{(\mathrm{d}_t)}, \mathrm{d}_t, \mathbf{x}_t)p(\mathbf{x}_t|\mathrm{d}_t), \quad \text{for } \mathrm{d}_t \in \mathcal{D},\\
\widehat{\mathrm{d}}_t &= \underset{\mathrm{d}_t\in\mathcal{D}}{\mathrm{argmax}}\; l(\widehat{\mathbf{x}}_t^{(\mathrm{d}_t)}) \times p(\mathrm{d}_t|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}}), \quad (7)
\end{aligned}$$

where the first step finds the optimal latent states $\widehat{\mathbf{x}}_t^{(\mathrm{d}_t)}$ under each candidate dimension d_t, and the second step chooses the optimal candidate dimension $\widehat{\mathrm{d}}_t$. The posterior probability is updated by

$$\begin{aligned}
&p(\mathrm{d}_t|\mathbf{Y}_t, \{\mathbf{X}_t^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}}) \quad (8)\\
&\propto p(\mathbf{y}_t|\mathbf{U}, \boldsymbol{\Lambda}^{(\mathrm{d}_t)}, \mathrm{d}_t, \widehat{\mathbf{x}}_t^{(\mathrm{d}_t)})p(\widehat{\mathbf{x}}_t^{(\mathrm{d}_t)}|\mathrm{d}_t)p(\mathrm{d}_t|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}})\\
&= l(\widehat{\mathbf{x}}_t^{(\mathrm{d}_t)}) \times p(\mathrm{d}_t|\mathbf{Y}_{t-1}, \{\mathbf{X}_{t-1}^{(\mathrm{d}_t')}\}_{\mathrm{d}_t'\in\mathcal{D}}), \qquad \text{for } \mathrm{d}_t \in \mathcal{D}.
\end{aligned}$$

The implementation under the constant dimension assumption is shown in Algorithm 1.

Algorithm 1: Constant Dimensions

for d *in* $\mathcal{D}$ **do**
- Assign the first M samples as inducing variable $\mathbf{U}$;
- Obtain $\boldsymbol{\Lambda}^{(\mathrm{d})}$ by optimizing (2);

for $t = 1$ *to* T **do**
- Receive $\mathbf{y}_t$;
- Find $\mathbf{x}_t^{(\mathrm{d}_t)}$ via (6) for every $\mathrm{d}_t \in \mathcal{D}$;
- Find $\widehat{\mathrm{d}}_t$ via (7);
- Update posterior via (8).

B. Dynamic Dimensions

In this case, the inducing variables $\mathbf{U}$ consist of observations whose latent states have different dimensions. We assume that the true dimensions of the latent states are randomly sampled from $\mathcal{D}$. The main task is to classify the inducing variables

so that the inducing variables in the same class have the same latent dimension. Our method has three steps:

1) Classify the inducing variables into several classes;
2) Determine the latent dimensions for each class;
3) Predict the dimension and latent states on testing data.

The first two steps are implemented with the training data, which aim to assign the new inducing variables $\mathbf{U}^{(\mathrm{d})} = [\mathbf{u}_1^{(\mathrm{d})}, \ldots, \mathbf{u}_M^{(\mathrm{d})}]^\top$ to each candidate dimension d. Let $\mathbf{\Lambda}^{(\mathrm{d})} = [\boldsymbol{\lambda}_1^{(\mathrm{d})}, \ldots, \boldsymbol{\lambda}_M^{(\mathrm{d})}]^\top$ denote the optimized latent states under $\mathbf{U}^{(\mathrm{d})}$. Notice that the inducing variables are not the same for every candidate dimension compared with the constant dimension assumption. Moreover, we apply the rolling window approach on $\mathbf{U}^{(\mathrm{d})}$ so that the inducing variables $\mathbf{U}_t^{(\mathrm{d})}$ and the latent states $\mathbf{\Lambda}_t^{(\mathrm{d})}$ are time-varying.

Suppose the observation $\mathbf{y}_t$ arrives at time instant t, and we already updated the inducing variables $\{\mathbf{U}_{t-1}^{(\mathrm{d})}\}_{\mathrm{d}\in\mathcal{D}}$ and the embedded latent states $\{\mathbf{\Lambda}_{t-1}^{(\mathrm{d})}\}_{\mathrm{d}\in\mathcal{D}}$ at $t-1$. Specifically, $\mathbf{U}_{t-1}^{(\mathrm{d})} = [\mathbf{u}_{t_1}^{(\mathrm{d})}, \ldots, \mathbf{u}_{t_M}^{(\mathrm{d})}]^\top$ and $\mathbf{\Lambda}_{t-1}^{(\mathrm{d})} = [\boldsymbol{\lambda}_{t_1}^{(\mathrm{d})}, \ldots, \boldsymbol{\lambda}_{t_M}^{(\mathrm{d})}]^\top$, where the subscripts $[t_1, \ldots, t_M]$ represent a strictly increasing time instant sequence. Next, we classify $\mathbf{y}_t$ into a proper class $\widehat{\mathrm{d}}_t$ and update $\mathbf{U}_t^{(\widehat{\mathrm{d}}_t)}$ by arranging $\mathbf{y}_t$. The estimate of $\widehat{\mathrm{d}}_t$ and latent state $\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}$ are derived from

$$\begin{aligned}(\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}, \widehat{\mathrm{d}}_t) &= \underset{\mathbf{x}_t^{(\mathrm{d_t})}, \mathrm{d_t}}{\operatorname{argmax}}\ p(\mathbf{x}_t^{(\mathrm{d_t})}, \mathrm{d_t} | \mathbf{y}_\mathrm{t}, \mathbf{Y}_{\mathrm{t}-1}, \{\mathbf{X}_{\mathrm{t}-1}^{(\mathrm{d}_\mathrm{t}')}\}_{\mathrm{d}_\mathrm{t}'\in\mathcal{D}}) \quad (9)\\ &= \underset{\mathbf{x}_t^{(\mathrm{d_t})}, \mathrm{d_t}}{\operatorname{argmax}}\ p(\mathbf{y}_t | \mathbf{U}_{t-1}^{(\mathrm{d_t})}, \mathbf{\Lambda}_{t-1}^{(\mathrm{d_t})}, \mathrm{d_t}, \mathbf{x}_\mathrm{t}^{(\mathrm{d_t})})\\ &\quad \times p(\mathbf{x}_t^{(\mathrm{d_t})} | \mathrm{d_t}) \times p(\mathrm{d_t} | \mathbf{Y}_{\mathrm{t}-1}, \{\mathbf{X}_{\mathrm{t}-1}^{(\mathrm{d}_\mathrm{t}')}\}_{\mathrm{d}_\mathrm{t}'\in\mathcal{D}})\\ &= \underset{\mathbf{x}_t^{(\mathrm{d_t})}, \mathrm{d_t}}{\operatorname{argmax}}\ p(\mathbf{y}_t | \mathbf{U}_{t-1}^{(\mathrm{d_t})}, \mathbf{\Lambda}_{t-1}^{(\mathrm{d_t})}, \mathrm{d_t}, \mathbf{x}_\mathrm{t}^{(\mathrm{d_t})})\\ &\quad \times p(\mathbf{x}_t^{(\mathrm{d_t})} | \mathrm{d_t}),\end{aligned}$$

where $p(\mathrm{d_t} | \mathbf{Y}_{\mathrm{t}-1}, \{\mathbf{X}_{\mathrm{t}-1}^{(\mathrm{d}_\mathrm{t}')}\}_{\mathrm{d}_\mathrm{t}'\in\mathcal{D}}) \equiv \mathrm{p}(\mathrm{d_t})$ because the true dimension of the latent states is randomly sampled from $\mathcal{D}$. The rolling-window update of $\mathbf{U}_t^{(\widehat{\mathrm{d}}_t)}$ and $\mathbf{\Lambda}_t^{(\widehat{\mathrm{d}}_t)}$ is given by:

$$\mathbf{U}_t^{(\widehat{\mathrm{d}}_t)} = [\mathbf{u}_{t_2}^{(\widehat{\mathrm{d}}_t)}, \ldots, \mathbf{u}_{t_M}^{(\widehat{\mathrm{d}}_t)}, \mathbf{y}_t]^\top, \quad (10)$$

$$\mathbf{\Lambda}_t^{(\widehat{\mathrm{d}}_t)} = [\boldsymbol{\lambda}_{t_2}^{(\widehat{\mathrm{d}}_t)}, \ldots, \boldsymbol{\lambda}_{t_M}^{(\widehat{\mathrm{d}}_t)}, \widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}]^\top. \quad (11)$$

In other words, $\mathbf{y}_t$ is appended into $\mathbf{U}_{t-1}^{(\widehat{\mathrm{d}}_t)}$ while the earliest inducing variable $\mathbf{u}_{t_1}^{(\widehat{\mathrm{d}}_t)}$ is dropped out, and $\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}$ is appended into $\mathbf{\Lambda}_{t-1}^{(\widehat{\mathrm{d}}_t)}$ while the earliest latent state $\boldsymbol{\lambda}_{t_1}^{(\widehat{\mathrm{d}}_t)}$ is dropped out. This rolling-window approach keeps the number of inducing variables as M. In addition, only the sparse GPLVM with $\widehat{\mathrm{d}}_t$ dimension is updated, while the candidate sparse GPLVMs with other dimensions remain the same. Consequently, the class $\widehat{\mathrm{d}}_t$ and its inducing variables $\mathbf{U}_t^{(\widehat{\mathrm{d}}_t)}$ absorb the observations that are similar to it, while their number remains the same.

Suppose the size of training data is t_0, a candidate dimension d is called "fully trained" when the last updated inducing variables $\mathbf{U}_{t_0}^{(\mathrm{d})}$ is totally different from the initial one $\mathbf{U}_0^{(\mathrm{d})}$, i.e., there is no row matched between them. It is acceptable that some candidate dimensions are "partially trained" because these candidate dimensions might be the perturbations. In principal, we suggest to collect enough training data to guarantee that the true latent dimensions are fully trained.

After classifying all the training data into proper classes $\mathrm{d} \in \mathcal{D}$ with inducing variables $\mathbf{U}_{t_0}^{(\mathrm{d})}$, the second step is to determine the optimal latent dimension d_0 under each class d. Notice that the class d does not necessarily have d as the optimal latent dimension. The reason that $\mathrm{d} \neq \mathrm{d}_0$ is that the initial inducing variable $\mathbf{U}_0^{(\mathrm{d})}$ at time 0 causes the wrong arrangement of observations at the beginning, i.e., appending an observation $\mathbf{y}_t$ with true latent dimension d_0 into $\mathbf{U}_t^{(\mathrm{d})}$, which then affects the following observations. However, it does not impact the right classification of observations. The optimal latent dimension d_0 and latent states $\mathbf{\Lambda}_{t_0}^{(\mathrm{d}_0)}$ for class d are given by

$$(\mathbf{\Lambda}_{t_0}^{(\mathrm{d}_0)}, \mathrm{d}_0) = \underset{\mathbf{\Lambda}_{t_0}^{(\mathrm{d}')}, \mathrm{d}'}{\operatorname{argmax}}\ p(\mathbf{U}_{t_0}^{(\mathrm{d})} | \mathbf{\Lambda}_{t_0}^{(\mathrm{d}')}, \mathrm{d}') p(\mathbf{\Lambda}_{t_0}^{(\mathrm{d}')} | \mathrm{d}'). \quad (12)$$

Therefore, we substitute the notation $\mathbf{U}_{t_0}^{(\mathrm{d})}$ with $\mathbf{U}_{t_0}^{(\mathrm{d}_0)}$ and construct the sparse GPLVM under d_0 dimension by $\mathbf{U}_{t_0}^{(\mathrm{d}_0)}$ and $\mathbf{\Lambda}_{t_0}^{(\mathrm{d}_0)}$.

The final step is to do the testing. Given the sparse GPLVMs under different dimensions, the estimated dimension $\widehat{\mathrm{d}}_t$ and latent states $\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}$ at time t is obtained via (9), and the inducing variables are updated via (10) and (11). The procedures are implemented by Algorithm 2.

Algorithm 2: Dynamic Dimensions

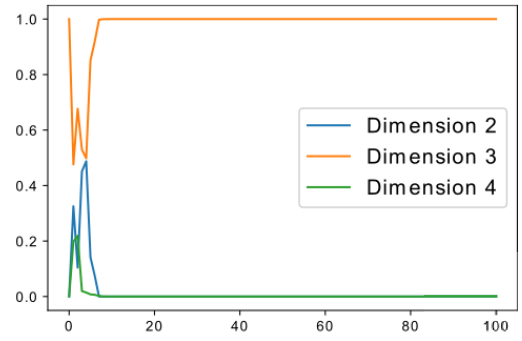
for d *in* $\mathcal{D}$ **do**
  Set the first M samples as inducing variable $\mathbf{U}_0^{(\mathrm{d})}$;
  Obtain $\mathbf{\Lambda}_0^{(\mathrm{d})}$ by optimizing (2);
Training:
for $t = 1$ *to* t_0 **do**
  Receive $\mathbf{y}_t$;
  Find $(\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}, \widehat{\mathrm{d}}_t)$ via (9);
  Update $\mathbf{U}_t^{(\widehat{\mathrm{d}}_t)}$ and $\mathbf{\Lambda}_t^{(\widehat{\mathrm{d}}_t)}$ via (10) and (11);
for d *in* $\mathcal{D}$ **do**
  Determine $(\mathbf{\Lambda}_{t_0}^{(\mathrm{d}_0)}, \mathrm{d}_0)$ for $\mathbf{U}_{t_0}^{(\mathrm{d})}$ via (12);
Testing:
for $t = t_0 + 1$ *to* T **do**
  Receive $\mathbf{y}_t$;
  Find $(\widehat{\mathbf{x}}_t^{(\widehat{\mathrm{d}}_t)}, \widehat{\mathrm{d}}_t)$ via (9);
  Update $\mathbf{U}_t^{(\widehat{\mathrm{d}}_t)}$ and $\mathbf{\Lambda}_t^{(\widehat{\mathrm{d}}_t)}$ via (10) and (11);

IV. Experiments

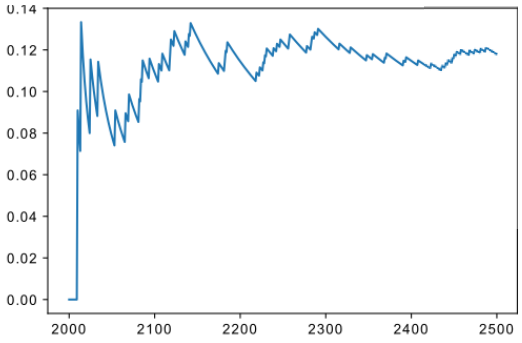
A. Synthetic test with a constant dimension

The observations are generated from

$$\mathbf{y}_t = \mathbf{W}\mathbf{x}_t + \epsilon_t, \quad (13)$$



(a) Constant dimension.



(b) Dynamic dimensions.

Fig. 1. (a) The trajectory of posterior probability for each dimension under a constant dimension assumption; (b) The error rate of predicted dimension on testing data under a dynamic dimension setting.

where $\mathbf{x}_t \in \mathbb{R}^3$ is the latent state, $\mathbf{y}_t \in \mathbb{R}^6$ is the observation, $\epsilon_t \sim \mathcal{N}(0, 0.01\mathbf{I})$, and $\mathbf{W} \in \mathbb{R}^{6\times 3}$. Specifically, the elements in $\mathbf{W}$ are randomly generated from -1 to 1, the $\mathbf{x}_t$s are generated from 5 different Gaussian distributed classes, and the set of candidate dimensions is given by $\mathcal{D} = \{2, 3, 4\}$. We generate 300 samples in total, where the first 200 are set as inducing variables while the remaining 100 are for testing. From Fig. 1(a), the posterior probability $p(\mathrm{d_t}|\mathrm{y}_{1:\mathrm{t}})$ always reaches probability one for $\mathrm{d_t} = 3$, which is the true dimension of the latent states. Moreover, $\mathrm{d_t} = 3$ quickly dominates the other candidates. From Fig. 2, the obtained latent states in the sequential mode converge to those of the offline mode only under $\mathrm{d_t} = 3$, while the other dimensions differ significantly between the sequential and offline modes.

B. Synthetic test with dynamic dimensions

The observations are again generated from

$$\mathbf{y}_t = \mathbf{W}_t\mathbf{x}_t + \epsilon_t, \tag{14}$$

where $\mathbf{y}_t \in \mathbb{R}^6$ is an observation vector at time t, and $\epsilon_t \sim \mathcal{N}(0, 0.01\mathbf{I})$. However, the true dimension of $\mathbf{x}_t$ is randomly sampled from $\mathcal{D}_0 = \{2, 3, 4\}$. The transformation matrices $\mathbf{W}_t$ under the true dimensions $\mathcal{D}_0$ are denoted as

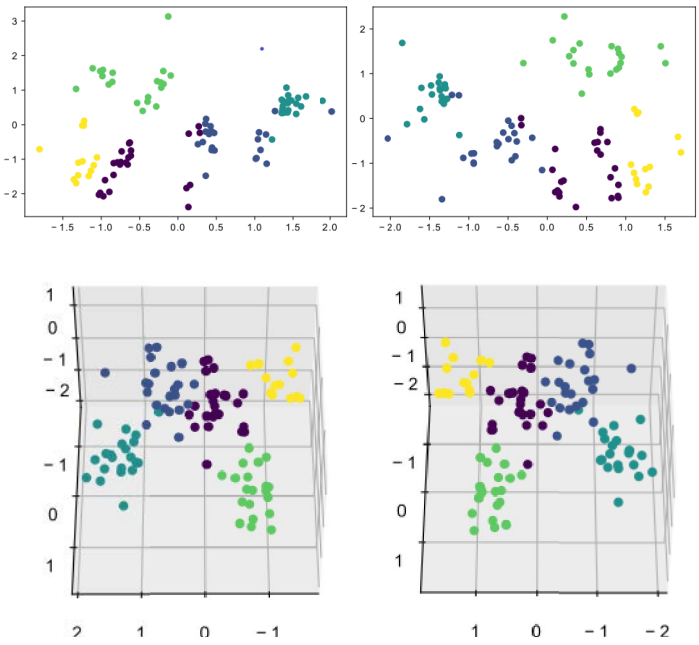


Fig. 2. The panels in the first column are the predicted latent states in online mode while the panels in the second column are the latent states in offline mode. The first row shows our online and offline methods under latent dimension 2, respectively, while the second row presents 3-D plots for the online and offline methods under latent dimension 3, respectively.

$\{\mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{W}^{(4)}\}$, where $\mathbf{W}^{(\mathrm{d})} \in \mathbb{R}^{6\times \mathrm{d}}, \mathrm{d} \in \mathcal{D}_0$. In other words,

$$\mathbf{y}_t = \mathbf{W}^{(\mathrm{d_t})}\mathbf{x}_t + \epsilon_t, \tag{15}$$

when $\mathbf{x}_t \in \mathbb{R}^{\mathrm{d_t}}$, and $\mathrm{d_t}$ is the true latent dimension at t. The elements in $\mathbf{W}_t$ are randomly generated from -2 to 2, and the candidate dimensions are in the range from 2 to 5 (inclusively), where dimension 5 is set to be a candidate of interference. We generated 2500 samples in total, where the first 150 are set as inducing variables, the next 1850 samples are treated as training data to classify the inducing variables, and the remaining 500 are for testing. Figure 1(b) shows the error rate among the last 500 samples, where the error rate on testing data is defined by

$$e_t = 1 - \frac{\sum_{t'=1}^{t}[\widehat{\mathrm{d}}_{t'} = \mathrm{d}_{t'}]}{t}, \tag{16}$$

where $[i = j]$ is Iverson bracket such that $[P] = 1$ when statement P is true while $[P] = 0$ when P is false. The error prediction on testing data mostly comes from predicting the true latent dimension with the perturbation dimension 5.

V. Conclusions

We developed a novel sequential sparse GPLVM by a rolling window. Based on the SSGPLVM, we do not only handle the constant dimension problem but also deal with latent states whose dimensions change with time. In reality, the true latent dimension might stay the same for a while and then increase or decrease. The assumption used in developing our method, however, is more general and can be applied on special cases with minor modifications.

REFERENCES

[1] S. Bacci, S. Pandolfi, and F. Pennoni. A Comparison of Some Criteria for States Selection in the Latent Markov Model for Longitudinal Data. *Advances in Data Analysis and Classification*, 8(2):125–145, 2014.

[2] G. H. Golub. Some Modified Matrix Eigenvalue Problems. *Siam Review*, 15(2):318–334, 1973.

[3] L. Guttman. Some Necessary Conditions for Common-Factor Analysis. *Psychometrika*, 19(2):149–161, 1954.

[4] A. Hegde, J. C. Principe, D. Erdogmus, U. Ozertem, Y. N. Rao, and H. Peddaneni. Perturbation-based Eigenvector Updates for On-line Principal Components Analysis and Canonical Correlation Analysis. *Journal of VLSI signal processing systems for signal, image and video technology*, 45(1):85–95, 2006.

[5] H. F. Kaiser. The Application of Electronic Computers to Factor Analysis. *Educational and psychological measurement*, 20(1):141–151, 1960.

[6] G. V. Karanikolas, Q. Lu, and G. B. Giannakis. Online unsupervised learning using ensemble gaussian processes with random features. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3190–3194. IEEE, 2021.

[7] T. M. Kodinariya and P. R. Makwana. Review on Determining Number of Cluster in K-Means Clustering. *International Journal*, 1(6):90–95, 2013.

[8] N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Nips*, volume 2, page 5. Citeseer, 2003.

[9] R. D. Ledesma and P. Valero-Mora. Determining the Number of Factors to Retain in efa: An Easy-to-Use Computer Program for Carrying Out Parallel Analysis. *Practical assessment, research, and evaluation*, 12(1):2, 2007.

[10] K. R. Mumford, J. M. Ferron, C. V. Hines, K. Y. Hogarty, and J. D. Kromrey. *Factor Retention in Exploratory Factor Analysis: A Comparison of Alternative Methods.* ERIC, 2003.

[11] A. Neishabouri and M. Desmarais. A novel method to determine the number of latent dimensions with svd. 2018.

[12] G. Raîche, T. A. Walls, D. Magis, M. Riopel, and J.-G. Blais. Non-graphical Solutions for Cattell's Scree Test. *Methodology*, 2013.

[13] C. E. Rasmussen. *Gaussian Processes in Machine Learning*. 2003.

[14] T. D. Sanger. Optimal Unsupervised Learning in A Single-layer Linear Feedforward Neural Network. *Neural networks*, 2(6):459–473, 1989.

[15] M. Tegner, B. Bloem-Reddy, and S. Roberts. Sequential Sampling of Gaussian Process Latent Variable models. *arXiv preprint arXiv:1807.04932*, 2018.

[16] M. E. Tipping and C. M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[17] M. K. Warmuth and D. Kuzmin. Randomized Online PCA Algorithms with Regret Bounds that are Logarithmic in the Dimension. *Journal of Machine Learning Research*, 9(Oct):2287–2320, 2008.

[18] H. Zha and H. D. Simon. On Updating Problems in Latent Semantic Indexing. *SIAM Journal on Scientific Computing*, 21(2):782–791, 1999.

[19] W. R. Zwick and W. F. Velicer. Comparison of Five Rules for Determining the Number of Components to Retain. *Psychological bulletin*, 99(3):432, 1986.