Gaussian processes for inference of deep state-space models

Petar M. Djurić

Stony Brook University

Work done with *Yuhao Liu*

Other contributers: *Marzieh Ajirak, Paolo Banelli, Kurt Butler, Tong Chen, Chen Cui, Taraneh Ghanbari, Guanchao Feng, Lingqing Gan, Charles Mikell, Sima Mofakham, Jessica Phillips, Yuri Saalman, Yuanqing Song, Hechuan Wang, Daniel Waxman, Liu Yang*

December 15, 2021

Stony Brook University

Outline

- Motivation
- Gaussian processes, deep Gaussian processes, and deep latent variable Gaussian processes
- Gaussian processes for inference in state-space models
- Ensembles of Gaussian processes
- Extensions to deep state-space models
- Conclusions

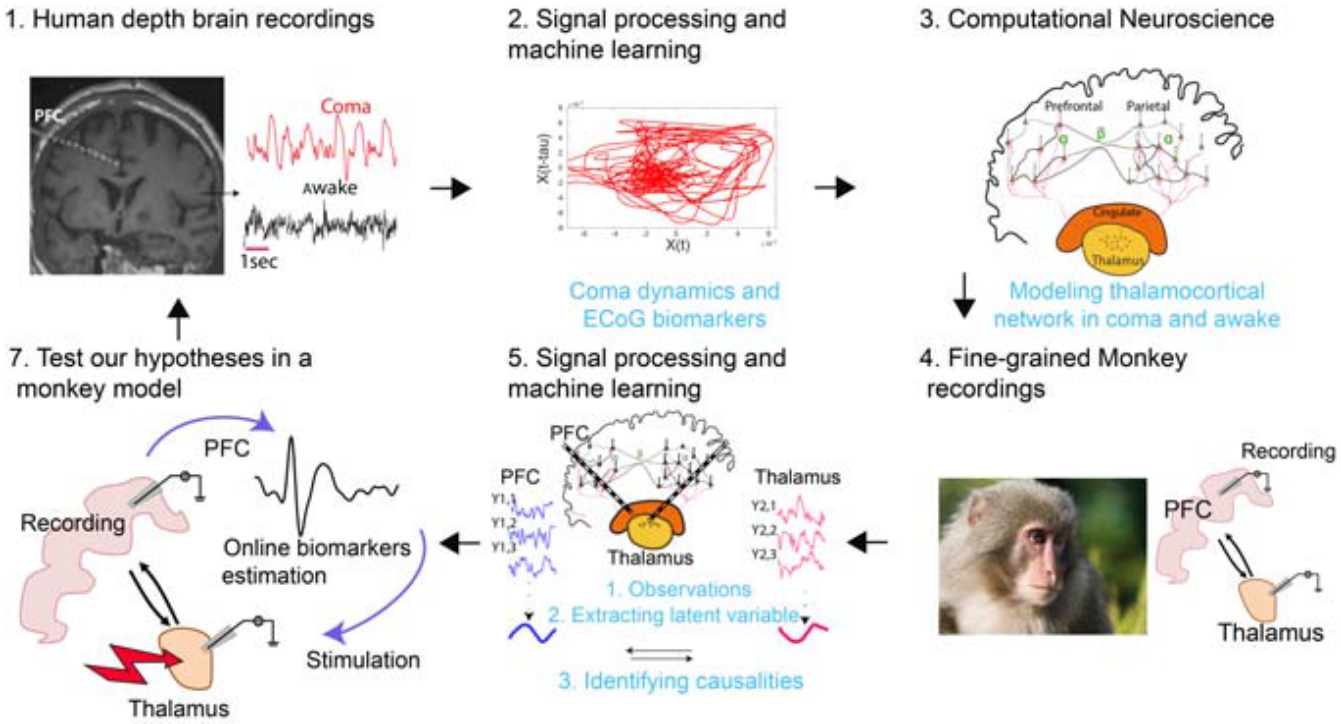
Stony Brook University

Motivation

- Our group has been working on a project where the main objective is to *find the source of consciousness and how consciousness emerges*.
- The measurements that are available for processing are
  - multivariate time series (local field potential signals) and
  - multivariate spike trains.

Stony Brook University

Motivation - contd.



Motivation - contd.

Guiding principles:

- **Compressibility**: The main goal of science is to understand Nature. When the scientific process is successful, a vast array of data can be concisely expressed by compact mathematical expressions. We then say that the data are algorithmically compressible.
- **The principle of locality**: In science, it is well established that local events (e.g., local in time and space) are the most influential. We use this principle in our work by building networks of objects and identifying "neighbors" of each object.

Stony Brook University

Gaussian processes

A Gaussian process, written as $\mathcal{GP}\left(m(\cdot), k(\cdot,\cdot|\boldsymbol{\theta})\right)$, is in essence a distribution over functions.

$m(\cdot)$ is a mean function,

$k(\cdot,\cdot)$ is a kernel or covariance function, and

$\boldsymbol{\theta}$ is the hyper-parameter parameterizing the kernel.

Stony Brook University

Gaussian Processes - contd.

For any set of inputs $\mathbf{X} = [\mathbf{x}_n]_{n=1}^N := [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ in the domain of a real-valued function $f \sim \mathcal{GP}(m, k)$, the function values $\mathbf{f} = [f(\boldsymbol{x}_n)]_{n=1}^N$ are Gaussian distributed, i.e.,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{m}_\mathbf{X}, \mathbf{K}_{\mathbf{XX}})$$

$\mathbf{m}_\mathbf{X} = [m(\mathbf{x}_n)]_{n=1}^N$ is the mean

$\mathbf{K}_{\mathbf{XX}} := k(\mathbf{X}, \mathbf{X}|\boldsymbol{\theta}) = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j}$ is the covariance matrix over all pairs in $\mathbf{X}$.

Stony Brook University

Gaussian Processes (contd.)

Given the observations $\mathbf{f}$ on $\mathbf{X}$, the predictive distribution of $\mathbf{f}_*$ at new inputs $\mathbf{X}_*$ is given by

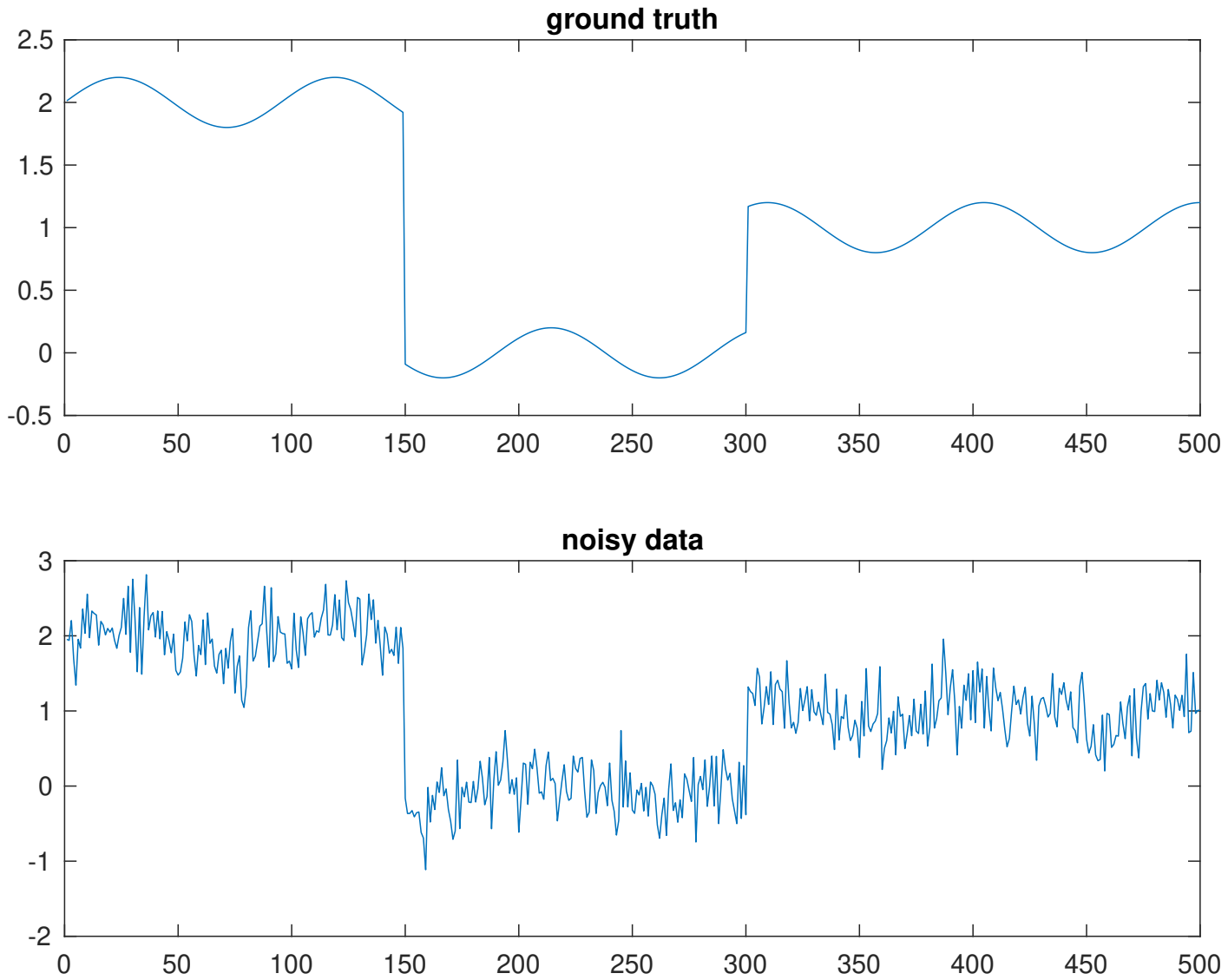
$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{f}, \mathbf{X}) = \mathcal{N}(\mathbf{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

with predictive mean and variance:

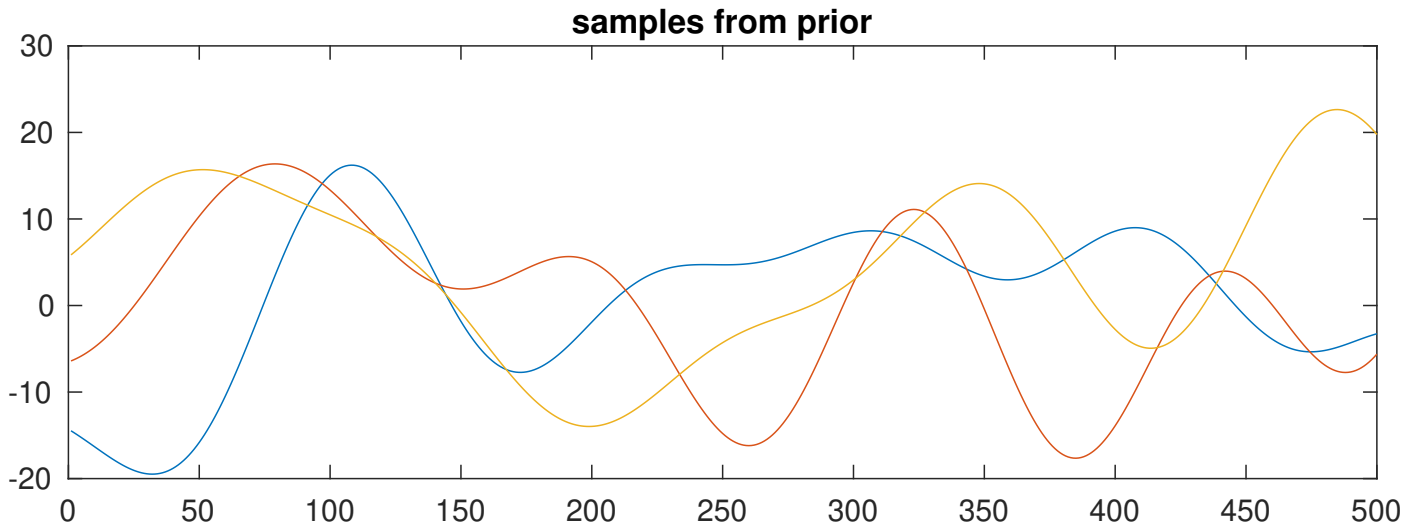
$$\boldsymbol{\mu}_* = \mathbf{m}_{\mathbf{X}_*} + \mathbf{K}_{\mathbf{X}_*\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}(\mathbf{f} - \mathbf{m}_{\mathbf{X}})$$
$$\boldsymbol{\Sigma}_* = \mathbf{K}_{\mathbf{X}_*\mathbf{X}_*} - \mathbf{K}_{\mathbf{X}_*\mathbf{X}}\mathbf{K}_{\mathbf{X}\mathbf{X}}^{-1}\mathbf{K}_{\mathbf{X}\mathbf{X}_*}$$

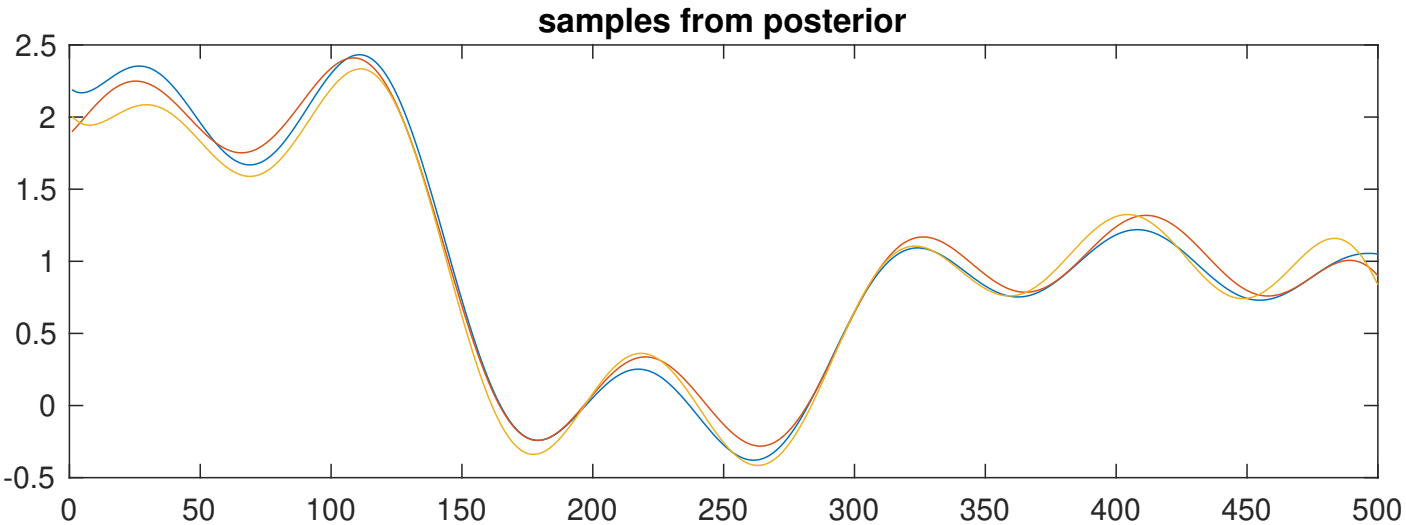
Stony Brook University

Gaussian Processes (contd.)

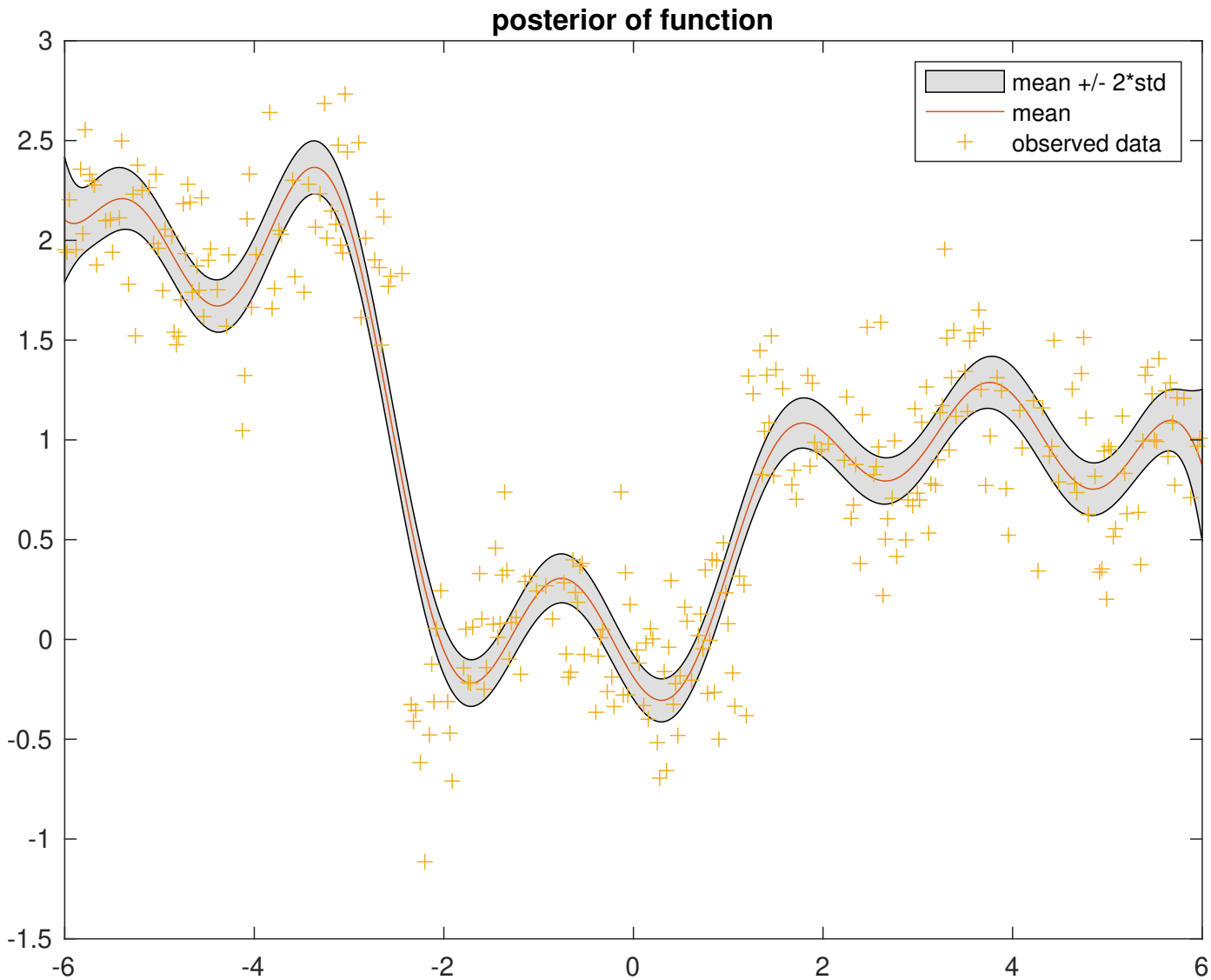


Gaussian Processes (contd.)





Gaussian Processes (contd.)



Gaussian Processes (contd.)

Gaussian processes do not scale up well with N, the number of input sets of data.

One has to invert the $N \times N$ matrix $\mathbf{K_{XX}}$, which for large values of N becomes an issue.

In order to ameliorate the problem, we resort to approximations of the kernels.

Stony Brook University

Gaussian Processes (contd.)

Compared with an approximation in a function space, a Gaussian process with shift-invariant kernel has another way of approximation, which focuses on feature spaces.

One can construct a vector of basis functions, also known as random features, comprised of trigonometric functions and defined by

$$\phi_{\mathbf{v}}(\mathbf{x}) = \frac{1}{\sqrt{R}}[\sin(\mathbf{x}^\top \mathbf{v}_1), \cos(\mathbf{x}^\top \mathbf{v}_1), ..., \sin(\mathbf{x}^\top \mathbf{v}_R), \cos(\mathbf{x}^\top \mathbf{v}_R)]^\top$$

where $\mathbf{v}_{1:R} = \{\mathbf{v}_r\}_{r=1}^R$ are vectors sampled from the power spectral density of the kernel.

Stony Brook University

Gaussian Processes (contd.)

Then the kernel function $k(\mathbf{x}, \mathbf{x}')$ can be approximated by $\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x})^{\top}\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}')$ if the kernel is shift-invariant. It allows for a parametric approximation of the function according to

$$\mathbf{f}(\mathbf{x}) \approx \boldsymbol{\Phi}_{\mathbf{v}}^{\top}\boldsymbol{\eta} \sim \mathcal{GP}(\boldsymbol{\Phi}_{\mathbf{v}}^{\top}\boldsymbol{\mu}, \boldsymbol{\Phi}_{\mathbf{v}}^{\top}\boldsymbol{\Sigma}\boldsymbol{\Phi}_{\mathbf{v}})$$

where $\boldsymbol{\eta} \in \mathbb{R}^{2R\times 1}$ is a parameter vector, which is Gaussian distributed, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and $\boldsymbol{\Phi} \in \mathbb{R}^{2R\times N}$.

Thus, the Gaussian process is approximated by another Gaussian process with a sparse representation $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{v}_{1:R}\}$, where $\mathbf{v}_{1:R}$ are the pre-selected random features.

Stony Brook University

The state-space model

Suppose the observations $\mathbf{y}_t$ are produced by a state space model

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{u}_t$$
$$\mathbf{y}_t = g(\mathbf{x}_t) + \mathbf{v}_t$$

where the functions $f(\cdot)$ and $g(\cdot)$ are unknown, and $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \sigma_u^2 \mathbf{I})$ and $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 \mathbf{I})$ are Gaussian distributed errors (noises).

Stony Brook University

The state-space model (contd.)

We now approximate the state-space model by the following model:

$$\mathbf{x}_t = \boldsymbol{\Psi}^\top \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_{t-1}) + \mathbf{u}_t$$
$$\mathbf{y}_t = \boldsymbol{\Theta}^\top \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_t) + \mathbf{v}_t$$

where $\boldsymbol{\phi}_{\mathbf{v}}$ represents the random features, and $\boldsymbol{\Psi} \in \mathbb{R}^{2R \times d_x}$ and $\boldsymbol{\Theta} \in \mathbb{R}^{2R \times d_y}$, or more specifically, $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_{d_x}]$, and $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{d_y}]$.

We assume that the parameter variables are all independent, i.e., that the columns of $\boldsymbol{\Psi}$ and $\boldsymbol{\Theta}$ are independent from the other columns of $\boldsymbol{\Psi}$ and $\boldsymbol{\Theta}$, respectively.

The state-space model (contd.)

Further, we assume a dynamic setting where the functions $f(\cdot)$ and $g(\cdot)$ change with time. We model $\boldsymbol{\psi}_i$ and $\boldsymbol{\theta}_j$ as random walks, and thus, the model becomes

$$\begin{aligned} \boldsymbol{\psi}_{i,t} &= \boldsymbol{\psi}_{i,t-1} + \mathbf{e}_{i,t} & \boldsymbol{\theta}_{j,t} &= \boldsymbol{\theta}_{j,t-1} + \boldsymbol{\epsilon}_{j,t} \\ x_{i,t} &= \boldsymbol{\phi}_{\mathbf{v}}^{\top}(\mathbf{x}_{t-1})\boldsymbol{\psi}_{i,t} + u_{i,t}, & y_{j,t} &= \boldsymbol{\phi}_{\mathbf{v}}^{\top}(\mathbf{x}_{t})\boldsymbol{\theta}_{j,t} + v_{j,t} \end{aligned}$$

where $\mathbf{e}_{i,t} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2\mathbf{I})$ and $\boldsymbol{\epsilon}_{j,t} \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2\mathbf{I})$ are Gaussian noises.

For the indices i and j, we have $i \in \{1, 2, \cdots, d_x\}$ and $j \in \{1, 2, \cdots, d_y\}$.

Thus, we have two linear state-space models, one for the state and one for the observations. Given the sequence $\mathbf{x}_t$, we can readily apply d_x Kalman filters to estimate $\boldsymbol{\psi}_{i,t}$ and d_y Kalman filters to estimate $\boldsymbol{\theta}_{j,t}$ ($i \in \{1, 2, \cdots, d_x\}$ and $j \in \{1, 2, \cdots, d_y\}$).

The estimation process

Suppose that at $t-1$ we have the estimates of $\mathbf{x}_{t-1}$, $\boldsymbol{\Psi}_{t-1}$, and $\boldsymbol{\Theta}_{t-1}$. Then we estimate $\mathbf{x}_t$, $\boldsymbol{\Psi}_t$, and $\boldsymbol{\Theta}_t$ as follows:

1. Estimate $\mathbf{x}_t$ using particle filtering (note that the model is nonlinear in $\mathbf{x}_t$).

2. Given the estimated $\mathbf{x}_t$, apply the two sets of Kalman filters to estimate $\boldsymbol{\Psi}_t$, and $\boldsymbol{\Theta}_t$.

These steps can be repeated before moving to the next time instant.

The ensemble approach

The use of only a single set of $\mathbf{v}_{1:R}$ might not be accurate enough.

Instead we use an ensemble of different sets of $\mathbf{v}_{1:R}$. Denote $\mathbf{v}_{1:R}^{(m)}$ as the $m-$th set of pre-selected parameters $\mathbf{v}_{1:R}$ sampled from the PSD of the kernel, where $m = 1 : M$.

Thus, we end up with M sets of filters, where each set is characterized by its random features.

Stony Brook University

The ensemble approach - cont.

Next, we invoke particle filtering again. We evaluate each set of filters by weights assigned to the filters.

If the weights at time $t-1$ are all equal, then

$$w_t^{(m)} \propto \mathcal{N}\left(\mathbf{y}_t | \widehat{\boldsymbol{\Theta}}_t^{\top} \boldsymbol{\phi}_{\mathbf{v}}^{(m)}(\widehat{\mathbf{x}}_t), \sigma_v^2 \mathbf{I}\right)$$

where $\widehat{\mathbf{x}}_t$ and $\widehat{\boldsymbol{\Theta}}_t$ are the respective estimates of $\mathbf{x}_t$ and $\boldsymbol{\Theta}_t$ by the $m-$th set of filters.

The estimates of the functions $f(\cdot)$ and $g(\cdot)$ are

$$\widehat{f}_t(\mathbf{x}_{t-1}) = \sum_{m=1}^{M} w_t^{(m)} \widehat{\boldsymbol{\Psi}}_t^{\top} \boldsymbol{\phi}_{\mathbf{v}}^{(m)}(\mathbf{x}_{t-1})$$

$$\widehat{g}_t(\mathbf{x}_t) = \sum_{m=1}^{M} w_t^{(m)} \widehat{\boldsymbol{\Theta}}_t^{\top} \boldsymbol{\phi}_{\mathbf{v}}^{(m)}(\mathbf{x}_t)$$

Stony Brook University

The ensemble approach - cont.

Resampling in particle filtering is a necessary step to avoid deterioration of the particle filtering with time.

Here we can also apply resampling. If a pair of Gaussian processes is resampled more than once, we need to assign to these processes different values for $\mathbf{x}_t$, $\mathbf{\Psi}_t$ and $\mathbf{\Theta}_t$.

A straightforward way of accomplishing this is by drawing from respective Gaussians.

Some results

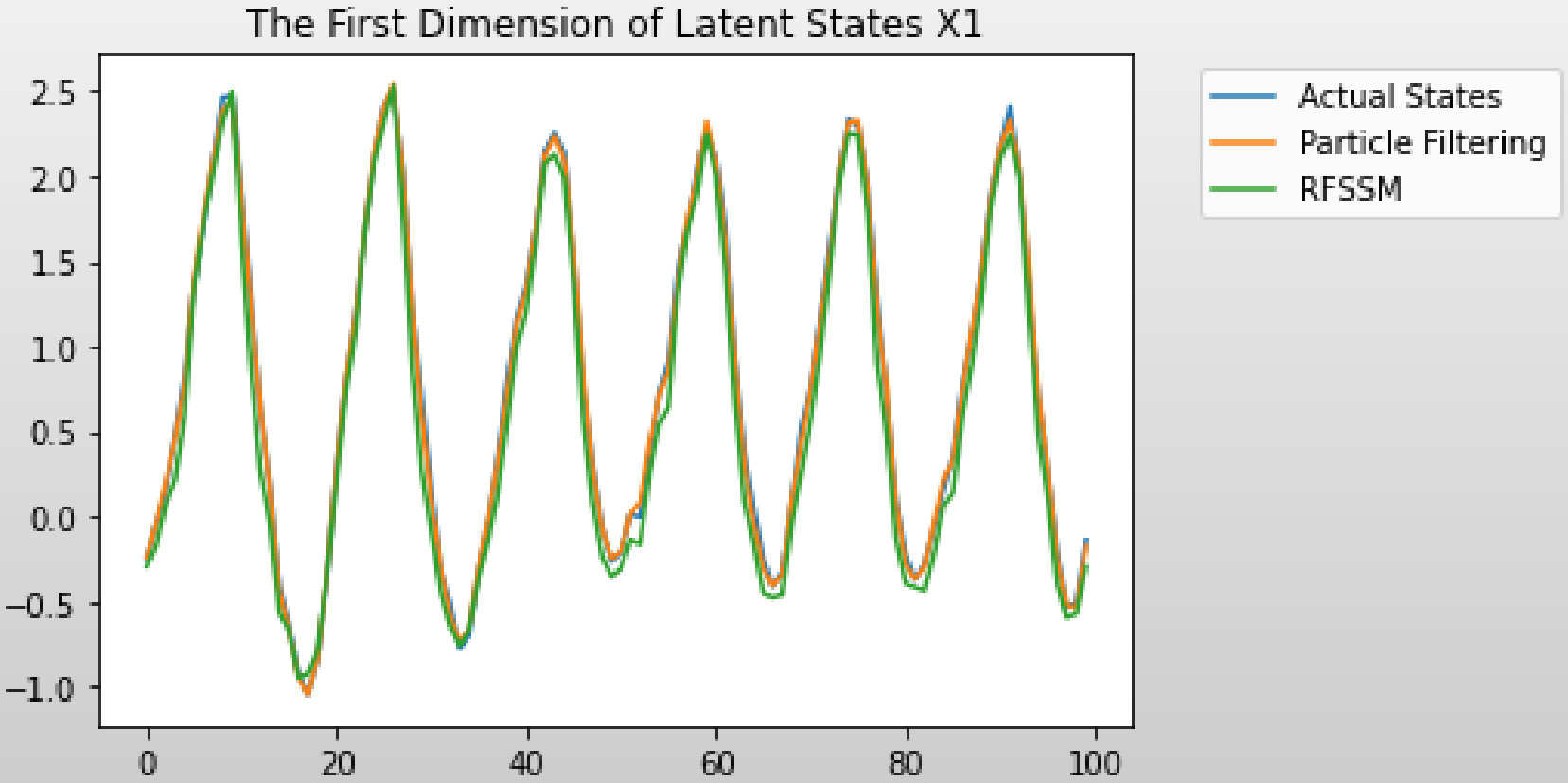
The following state-space model was generated by:

$$x_{1,t} = 0.9x_{1,t-1} + 0.5\sin(x_{2,t-1}) + u_{1,t}$$
$$x_{2,t} = 0.1x_{1,t}^3 - 0.9x_{1,t} + u_{2,t}$$

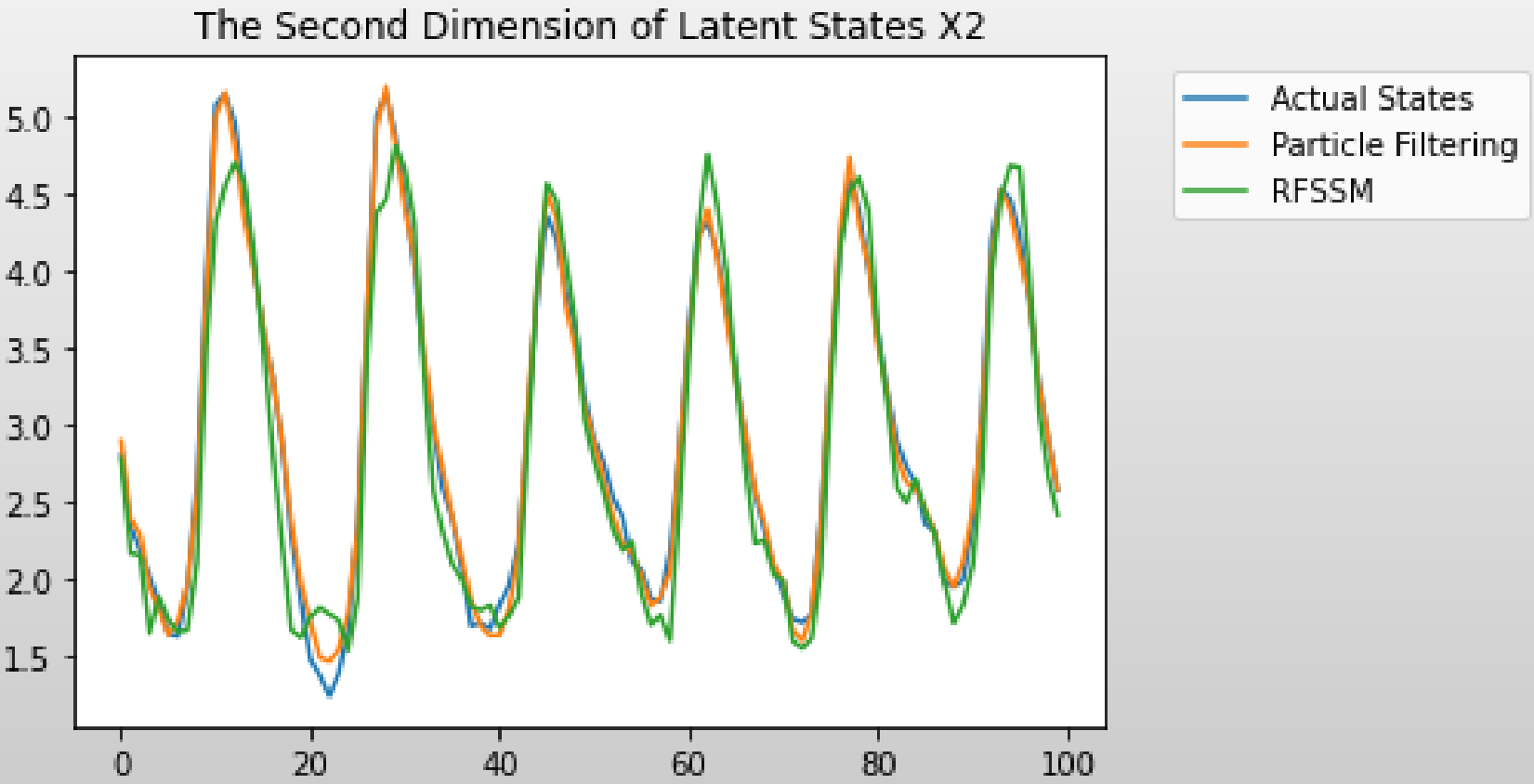
$$y_{1,t} = 1.8\cos(x_{1,t}) - 0.7\sin(x_{2,t}) + v_{1,t}$$
$$y_{2,t} = 0.5x_{1,t} - 1.3\sin(x_{1,t}) + v_{2,t}$$
$$y_{3,t} = 2.0x_{1,t} - 0.4x_{2,t} + v_{3,t}$$
$$y_{4,t} = 0.05x_{1,t}^3 + v_{4,t}$$
$$y_{5,t} = x_{2,t}/(1 + x_{2,t}^2) + v_{5,t}$$

Stony Brook University

Some results - contd.

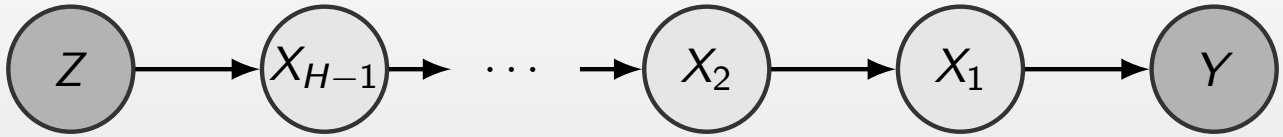


Some results - contd.



Stony Brook University

Deep Gaussian Processes



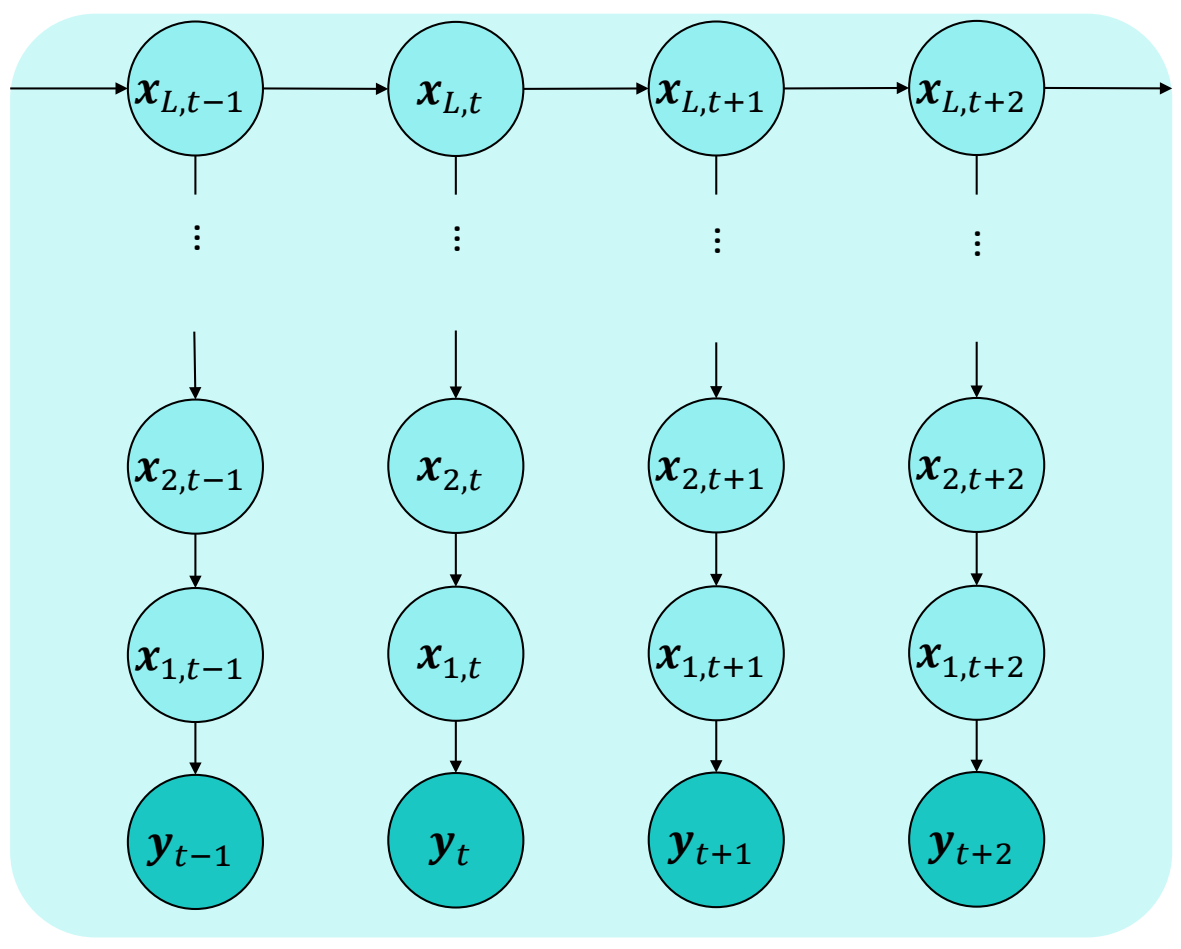
- $\mathbf{Y} \in \mathbb{R}^{N \times d_y}$: observations, output of the network
  - N is the number of observation vectors.
  - d_y is the dimension of the vectors $\mathbf{y}_n$.

- $\{\mathbf{X}_h\}_{h=1}^{H-1}$: intermediate latent states
  - dimensions $\{d_h\}_{h=1}^{H-1}$ are potentially different.

- $\mathbf{Z} \in \mathbb{R}^{N \times d_z}$: the input to the network
  - $\mathbf{Z}$ is observed for supervised learning.
  - $\mathbf{Z}$ is unobserved for unsupervised learning.

Stony Brook University

Deep Gaussian State Space Processes



Stony Brook University

Deep Gaussian State Space Processes - contd.

Now our deep Gaussian state-space model is described by

$$
\begin{aligned}
\boldsymbol{\Psi}_{L,t} &= \boldsymbol{\Psi}_{L,t-1} + \mathbf{u}_{\boldsymbol{\Psi}_{L,t}} \\
\mathbf{x}_{L,t} &= \boldsymbol{\Psi}_{L,t}^{\top} \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_{L,t-1}) + \mathbf{u}_{L,t} \\
&\ldots\ldots \\
\boldsymbol{\Psi}_{2,t} &= \boldsymbol{\Psi}_{2,t-1} + \mathbf{u}_{\boldsymbol{\Psi}_{2,t}} \\
\mathbf{x}_{2,t} &= \boldsymbol{\Psi}_{2,t}^{\top} \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_{3,t}) + \mathbf{u}_{2,t} \\
\boldsymbol{\Psi}_{1,t} &= \boldsymbol{\Psi}_{1,t-1} + \mathbf{u}_{\boldsymbol{\Psi}_{1,t}} \\
\mathbf{x}_{1,t} &= \boldsymbol{\Psi}_{1,t}^{\top} \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_{2,t}) + \mathbf{u}_{1,t} \\
\boldsymbol{\Theta}_t &= \boldsymbol{\Theta}_{t-1} + \boldsymbol{\epsilon}_t \\
\mathbf{y}_t &= \boldsymbol{\Theta}_t^{\top} \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_{1,t}) + \mathbf{v}_t
\end{aligned}
$$

Stony Brook University

Deep Gaussian State Space Processes - estimation

Suppose that we have estimates of all the unknowns at $t-1$. Then we proceed as follows:

1. Using particle filters at each layer, we propagate the states $\mathbf{x}_{l,t-1}$ to $\mathbf{x}_{l,t}$, for $l = 1 : L$ starting from the deepest layer until we reach $\mathbf{x}_{1,t}$. We then weight the states $\mathbf{x}_1^{(m)}$ and using the obtained weights compute the estimates of all the processes, $\widehat{\mathbf{x}}_{l,t}$.

2. Given the estimated states $\widehat{\mathbf{x}}_{l,t}$, we apply Kalman filters to estimate all the parameter processes $\boldsymbol{\Theta}_t$ and $\boldsymbol{\Psi}_{l,t}$ $l = 1 : L$.

3. We may repeat the above two steps one or more times.

Stony Brook University

Conclusions

1. Motivated by studies based on multivariate local field potential signals and spike trains from the brain, deep state space models for tracking latent states were proposed and investigated.
2. Gaussian processes approximated by random features were used to track the latent state processes with particle and Kalman filters.
3. The method was extended to include an ensemble of filters.

Stony Brook University