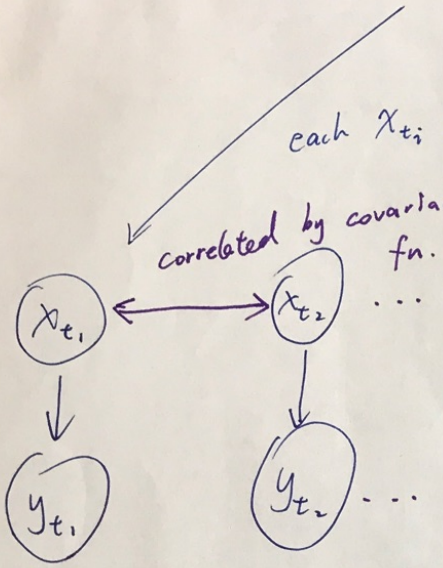
GP ⟶ GPFA ⟶ GPFA + non-rev. = GPFADs

kernels / Covariance fn.



Gaussian Prior: $[x_d \sim GP(0, k_{RBF})]$

① $\log p(\underline{Y}) = \log \int p(\underline{Y}|\underline{X}) p(\underline{X}) \, d\underline{X}$

$= \log N(\text{vec}(\underline{Y}); \mu = 0, \underline{\Sigma} = \underline{K}_{yy})$

② Reversibility Index

$\zeta = \dfrac{\| C^{(-)} \|_F}{\| C^{(+)} \|_F}$ $\underline{\underline{C}}$: space-time covariance

③ Construction of non-reversible planar GPs

GPFADs Problems: generalize beyond latent planes

Bayesian Inference & Predictions in finite regression Models

Data: $\underline{x}, \underline{y}$

Model M: $y = f_w(x) + \varepsilon$

Gaussian likelihood:

$$p(\underline{y} | \underline{x}, \underline{w}, M) \propto \prod_{n=1}^{N} \exp\left(-\frac{1}{2}(y_n - f_w(x_n))^2 / \sigma^2_{noise}\right)$$

Maximum Likelihood:

$$\underline{w}_{ML} = \arg\max_{w} \underbrace{p(\underline{y} | \underline{x}, \underline{w}, M)}_{\text{data | model \& weights \& x}}$$

Predictions: (Max. likelihood)

$$\underbrace{p(y_* | x_*, \underline{w}_{ML}, M)}_{\text{new data | new x, trained weights } \underline{w}_{ML} \text{ \& model}}$$

Posterior (Bayes rule for 3+ variables)

$$p(\underline{w} | \underline{x}, \underline{y}, M) = \frac{\overbrace{p(\underline{w} | M)}^{\text{prior}} \overbrace{p(\underline{y} | \underline{x}, \underline{w}, M)}^{\text{likelihood}}}{p(\underline{y} | \underline{x}, M)}$$

(Marginalizing out the parameters)

Predictions:

$$p(y_* | x_*, \underline{x}, \underline{y}, M) = \int p(y_*, \underline{w} | \underline{x}, \underline{y}, x_*, M)\, dw$$

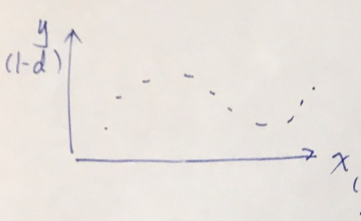
$$= \int \underbrace{p(y_* | \underline{w}, x_*, M)}_{\text{likelihood}} \underbrace{p(\underline{w} | \underline{x}, \underline{y}, M)\, dw}_{\text{posterior}}$$

Marginal Likelihood:

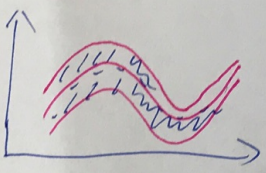
$$p(\underline{y} | \underline{x}, M) = \int \underbrace{p(\underline{w} | \underline{x}, M)}_{\text{prior}} \underbrace{p(\underline{y} | \underline{x}, \underline{w}, M)}_{\text{likelihood}} dw$$

Howard

Coursework



GP

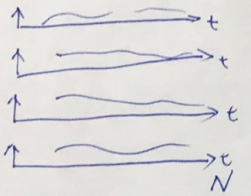


Tensorflow → GPflow (GP Intro)

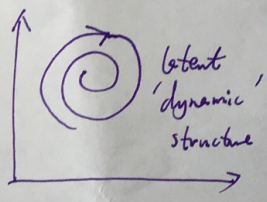
Multi-O/P GP → Covariance fn. design

GPFADs

Project ① ② Non-rev GP Regression



GFFADs



② GPFADS (GP Factor Analysis with DS)

① Plot kernel √√√

↑

build kernel ⟹

$$\begin{bmatrix} [f(\tau)] & [H[f](\tau)] \\ [-H[f](\tau)] & [f(\tau)] \end{bmatrix} \quad \begin{matrix} t_1 \\ t_2 \end{matrix}$$

② Plot trajectories from kernel

$$\frac{\tau}{2} = \left(\frac{1+\tau^2}{\tau}\right)$$

$$K\left(\begin{pmatrix} t_1 \\ t_2 \end{pmatrix}^{l}, \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}^{r}\right)$$

$$= \begin{bmatrix} f^{(+)}(t_1^l - t_1^r) & \alpha f^{(-)}(t_1^l - t_2^r) \\ -\alpha f^{(-)}(t_2^l - t_1^r) & f^{(+)}(t_2^l - t_2^r) \end{bmatrix}$$

$f^{(+)}$: scalar cov function (RQ)

Hilbert transform

should I add "-" here?

K not symmetric?

$K_{xx} \in \mathbb{R}^{MT \times MT}$

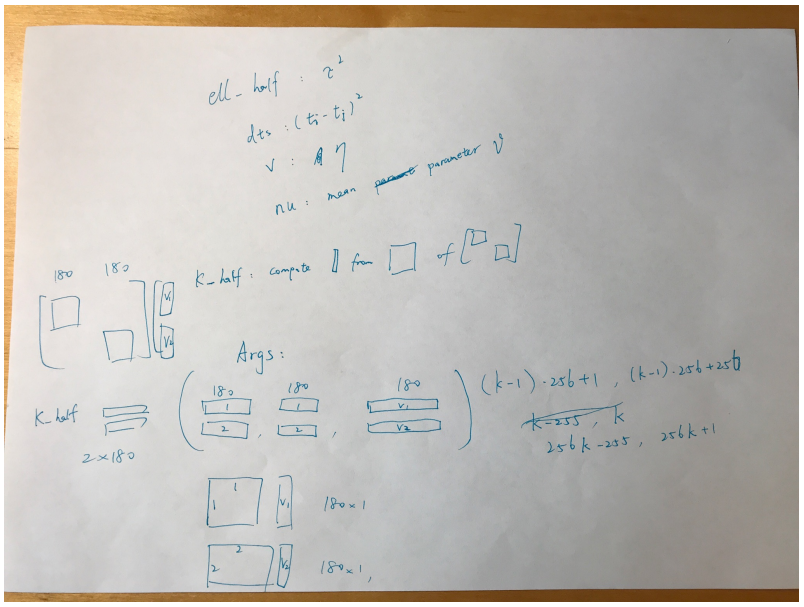
$$F_{circulant} = \begin{bmatrix} f_0 & \cdots & \cdots & \cdots \\ f_1 & & & \\ \vdots & & & \\ f_{T-2} & & & \\ f_{T-1} & & & \\ f_{T-2} & & & \\ \vdots & & & \\ f_1 & & & \end{bmatrix} \triangleq \begin{bmatrix} F & S \\ S^T & F' \end{bmatrix}$$

contains all the info

0 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9

so for each dimension we have a $\begin{bmatrix} \bar{v}_m \\ \bar{0} \end{bmatrix}$ so that $\begin{bmatrix} F & S \\ S^T & F' \end{bmatrix}\begin{bmatrix} \bar{v}_m \\ \bar{0} \end{bmatrix} = \begin{bmatrix} F\bar{v}_m \\ S^T\bar{0} \end{bmatrix}$

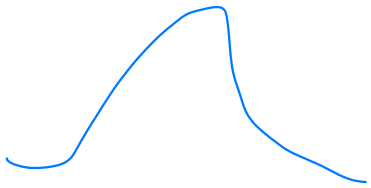
$$= \begin{bmatrix} DFT^{-1}[DFT(F) \odot DFT(\bar{v}_m)] \\ \cdots \end{bmatrix}$$


ell_half : r^2
dts : $(t_i - t_j)^2$
v : η
nu : mean parameter $\hat{\nu}$
180
180
K_half : compute ‖ from ☐ of [☐ ☐]
Args :
K_half
2×180
180
180
180
$(k-1)\cdot 256+1$, $(k-1)\cdot 256+256$
$k-255$, k
$256k-255$, $256k+1$
180×1
180×1,

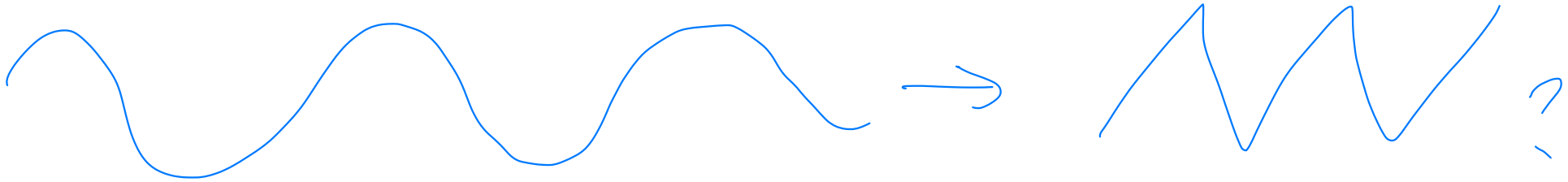
Continuous multidimensional data ⟹ discrete spike trains

what about reversing the process?

spectral mixture



— timescales about neuron



objective?

why do we need non-reversibility?

↙ apply to neural data!

20/10/2021

① get to familiarize GP quickly !!!

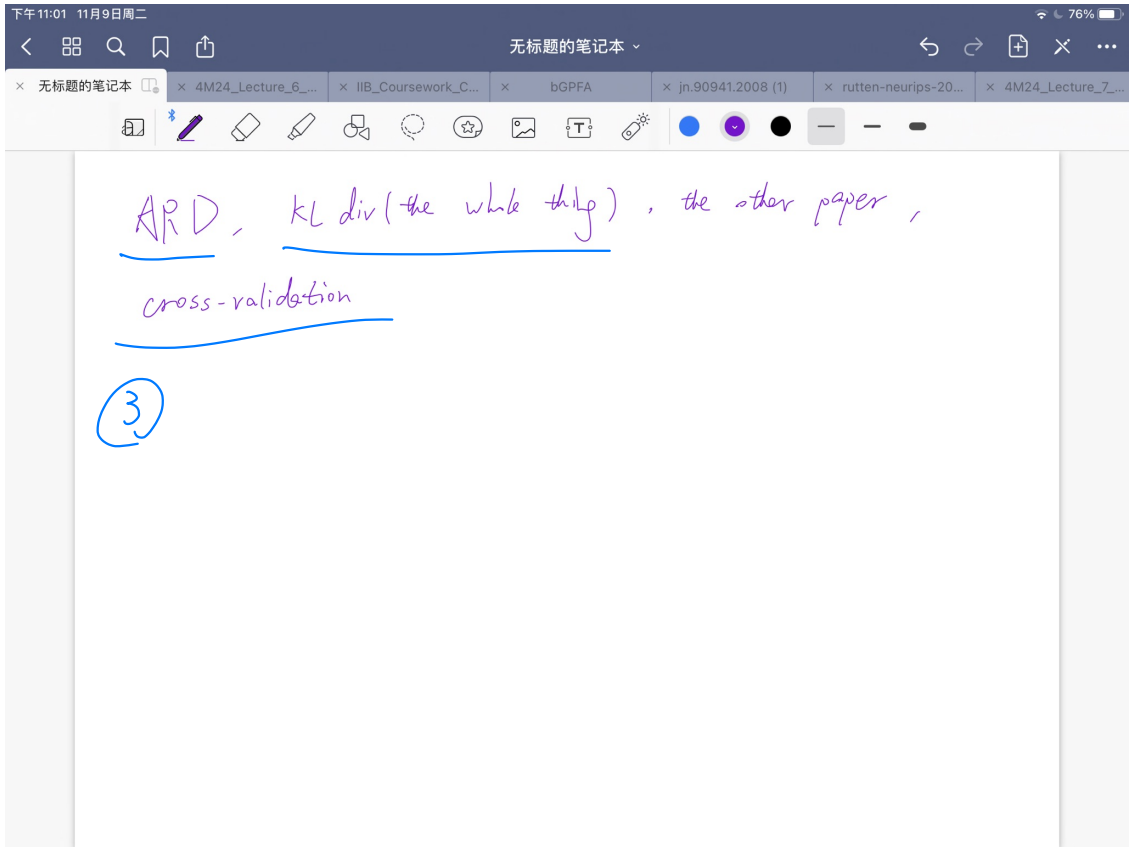
② temporal reversibility !!

③ pendulum dynamics !

④ 2nd order non-reversibility?

Nov 9, 2021

① ELBO

② 

b GPFA:

$$p(Y|t) = \int p(Y|F)\, p(F|x)\, p(x|t)\, dF\, dx$$

latent variables

time

neuron activity

neuron-specific variables (firing rates, etc)

$$p(x|Y,t) = \frac{p(x|t)\, p(Y|t,x)}{p(Y|t)}$$

$+F$ $+F$

$p(Y|t)$ ~ marginal likelihood

Bayesian GPFA is defined by the following generative model:

$\mathbf{Y} \in \mathbb{R}^{N \times T}$ (data)
$\mathbf{X} \in \mathbb{R}^{D \times T}$ (latent variables)
$\mathbf{C} \in \mathbb{R}^{N \times D}$ (readout matrix)

$$\mathbf{x}_d \sim \mathcal{GP}(0, K_{\mathrm{RBF}})$$
$$c_{nd} \sim \mathcal{N}(c_{nd}; \mu = 0, \sigma^2 = s_d^2)$$
$$y_{nt|\mathbf{C},\mathbf{X}} = p(y_{nt}|(\mathbf{CX})_{nt})$$

We learn the scales $\{s_d\}$ by maximizing a lower bound (ELBO) on the model log marginal likelihood:

$$\mathcal{L} \leq \log p(\mathbf{Y}) = \log \int p(\mathbf{Y}|\mathbf{C}, \mathbf{X}) p(\mathbf{C}) p(\mathbf{X}) \, d\mathbf{C} \, d\mathbf{X}$$

This is achieved using variational inference which also provides an estimate of the posterior distribution over our latent variables that we can use for further analyses: $q(\mathbf{X}) \approx p(\mathbf{X}|\mathbf{Y})$.

See Jensen & Kao et al. (2021) for further details about the generative model and inference procedure.

We start by downloading an example dataset which was originally recorded by O'Doherty et al. (2018). Here we consider a single recording session where we have binned the data in 25 ms bins in advance. We have put this data on google drive for ease of access in this tutorial; note that the original dataset is available from https://zenodo.org/record/3854034#.YNCEy5P0nUI.

$$p(\boldsymbol{Y}|\boldsymbol{F}) = \prod_{n,t} \mathcal{N}(y_{nt}; f_{nt}, \sigma_n^2) \tag{2}$$

$$p(\boldsymbol{F}|\boldsymbol{X}) = \delta(\boldsymbol{F} - \boldsymbol{C}\boldsymbol{X}) \tag{3}$$

$$p(\boldsymbol{X}|\boldsymbol{t}) = \prod_{d} \mathcal{N}(\boldsymbol{x}_d; \boldsymbol{0}, \boldsymbol{K}_d) \qquad \text{with } \boldsymbol{K}_d = k_d(\boldsymbol{t}, \boldsymbol{t})) \tag{4}$$

That is, the prior over the d^{th} latent function $x_d(t)$ is a Gaussian process (Rasmussen and Williams, 1996) with covariance function $k_d(\cdot, \cdot)$ (usually a radial basis function), and the observation model $p(\boldsymbol{Y}|\boldsymbol{X})$ is given by a parametric linear transformation with independent Gaussian noise.

In this work, we additionally introduce a prior distribution over the mixing matrix $\boldsymbol{C} \in \mathbb{R}^{N \times D}$ with hyperparameters specific to each latent dimension. This allows us to *learn* an appropriate latent dimensionality for a given dataset using automatic relevance determination (ARD) similar to previous work in Bayesian PCA (Appendix H; Bishop, 1999) rather than relying on cross-validation or ad-hoc thresholds of variance explained. Unlike in standard GPFA, the log marginal likelihood (Equation 1) becomes intractable with this prior. We therefore develop a novel variational inference strategy (Wainwright and Jordan, 2008) which also (i) provides a scalable implementation appropriate for long continuous neural recordings, and (ii) extends the model to general non-Gaussian likelihoods better suited for discrete spike counts.

In this new framework, which we call Bayesian GPFA (bGPFA), we use a Gaussian prior over $\boldsymbol{C}$ of the form $c_{nd} \sim \mathcal{N}(0, s_d^2)$, where s_d is a scale parameter associated with latent dimension d. Integrating $\boldsymbol{C}$ out in Equation 3 then yields the following observation model:

$$p(\boldsymbol{F}|\boldsymbol{X}) = \prod_{n} \mathcal{N}(\boldsymbol{f}_n; 0, \boldsymbol{X}^T \boldsymbol{S}^2 \boldsymbol{X}), \qquad \text{with } \boldsymbol{S} = \text{diag}(s_1, \ldots, s_D). \tag{5}$$

Moreover, we use a general noise model $p(\boldsymbol{Y}|\boldsymbol{F}) = \prod_{n,t} p(y_{nt}|f_{nt})$ where $p(y_{nt}|f_{nt})$ is any distribution for which we can evaluate its density.

2.2 Variational inference and learning

To train the model and infer both $\boldsymbol{X}$ and $\boldsymbol{F}$ from the data $\boldsymbol{Y}$, we use a nested variational approach. It is

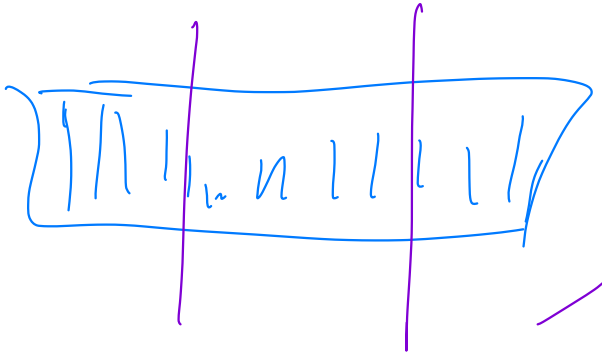
① GP Regression (1) / GPEADs (2) ② (bGPFA) SVGP (GPflow) ✗
has notebooks

also figure 2 GP bad or not?

multi-o/p GP: GPflow? SVGP needed?

③ MIT — bGPFA → kinematic decoding meaning?

④ real-world application, GPFA: expensive for ≥ a few 100 time bins



problems: ② maybe each chunk's represent different neural activities experiment activities, cause difference in parameters?

① How to train all these pieces of samples? Altogether? Separately?

why way to implement Kernel?

Presentation :

1.5 Intro ⟶ Research's background, purpose of neural decoding, contents in this presentation

GPFADs ⟶ GPFA, irrev-kernel: what is it, how to construct, example, comparison, higher-dimension (review, show understanding of the paper)

Implementation ⟶ kernel plot, regression, GPFADs (all planar)

Application to bGPFA ⟶ bGPFA intro

1.5 Next plan, difficulties... ⟶ bGPFA → bGPFADs; other way to express irreversibility (spatial, temporal?) ✓

(GPFADs generate trajectories which ∧ Cross-over, what's the value of irrev index when don't ✗ over

⟶ coding!! Maths!!