Bellairs Research Institute
Barbados, December 14, 2021

Ensemble Gaussian Processes for Online, Interactive, and Deep Learning with Scalability and Adaptivity

Georgios B. Giannakis

Acknowledgements: *Drs. Q. Lu, Y. Shen, P. Traganitis; G.-V. Karanikolas, and K. Polyzos*

NSF grant 1901134

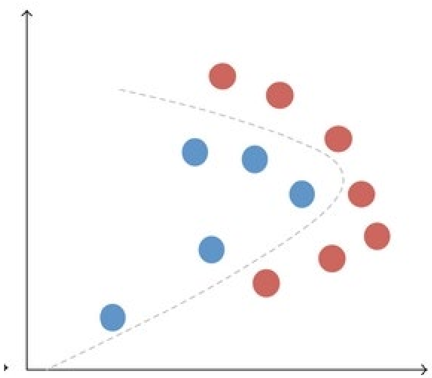
UNIVERSITY OF MINNESOTA
Driven to Discover℠

Agenda

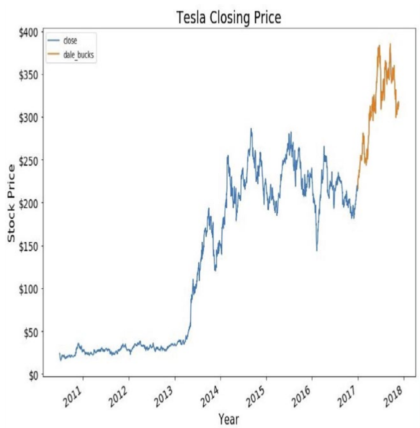
- Part I - Gaussian processes (GPs) and random features (RFs)
- Part II - Incremental (online) and ensemble Gaussian processes (IE-GP)
- Part III.A - Bayesian (black-box or bandit) optimization using GPs
- Part III.B – Reinforcement learning (RL) using (E)GPs
- Closing remarks and outlook

Motivating context

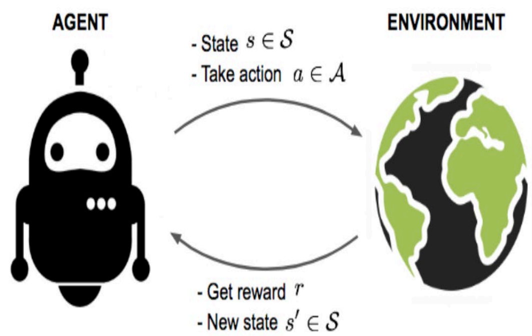
- Nonlinear function models are widespread in real-world applications



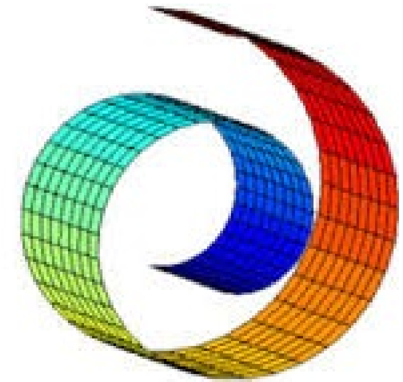
Classification



Regression

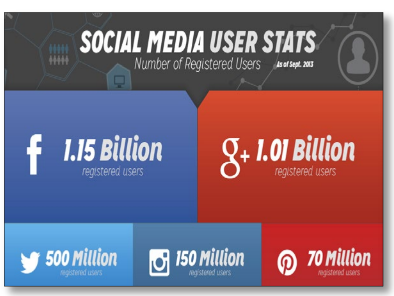


Reinforcement learning

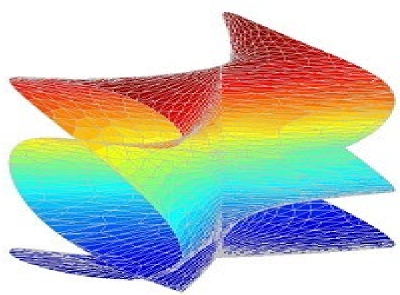


Dimensionality reduction

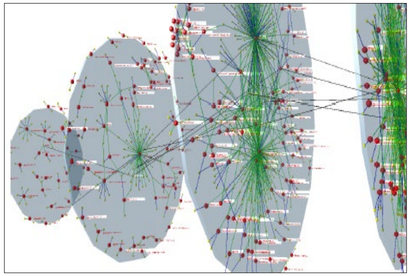
- Challenges and opportunities



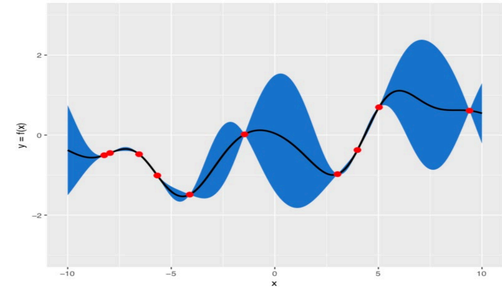
Massive scale



Unknown nonlinearity



Unknown dynamics



Uncertainty quantification

Part I

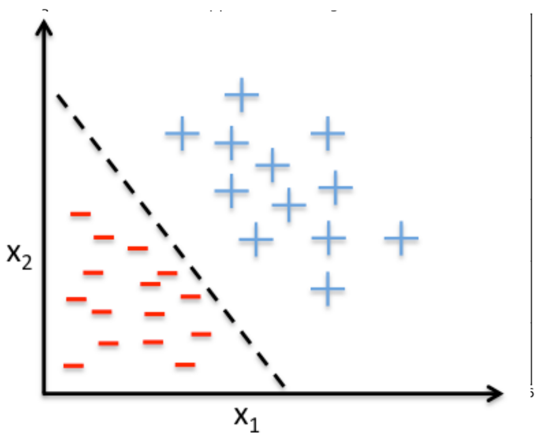
- Gaussian processes (GPs) and random features (RFs)
  - GP/RF basics and applications
  - GP links with wide and deep neural networks (DNNs)
  - Deep GPs

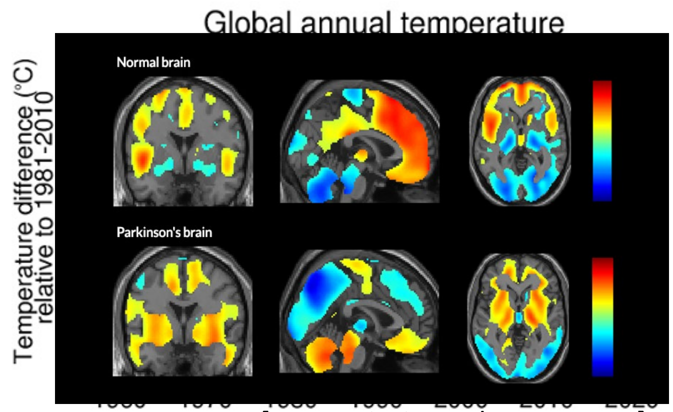
Learning functions from data

Goal: Given data $\{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$, find $f(\cdot)$: $\mathbf{x}_t \rightarrow f(\mathbf{x}_t) \rightarrow y_t$

Ex1. Regression: $y_t = \boldsymbol{\theta}^\top \mathbf{x}_t + e_t$ Curve fitting for e.g. temperature forecasting

Ex2. Classification: $y_t = \mathrm{sign}(\boldsymbol{\theta}^\top \mathbf{x}_t + \mathbf{b})$ For e.g., disease diagnosis





[P. Spetsieris et al PNAS 2015]

- Even unsupervised tasks boil down to function learning
  - E.g., dimensionality reduction, clustering, anomaly detection …

Learning functions with kernels

Model: view f as <u>deterministic</u> from a Hilbert space $\mathcal{H} := \{f \,|\, f(\mathbf{x}) = \sum_{t=1}^{\infty} \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t)\}$

kernel

e.g., $e^{-\|\mathbf{x}-\mathbf{x}_t\|^2/\sigma_\kappa^2}$

- Given data $\{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$, find

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{T} \sum_{t=1}^{T} \underbrace{\mathcal{C}(f(\mathbf{x}_t), y_t)}_{\text{cost}} + \underbrace{\lambda \Omega\left(\|f\|_{\mathcal{H}}^2\right)}_{\text{regularizer}}$$

  - E.g., Least-squares cost and L_2 regularizer $\longrightarrow$ kernel ridge regression

Q1. Kernel selection? **Q2**. Prior information?

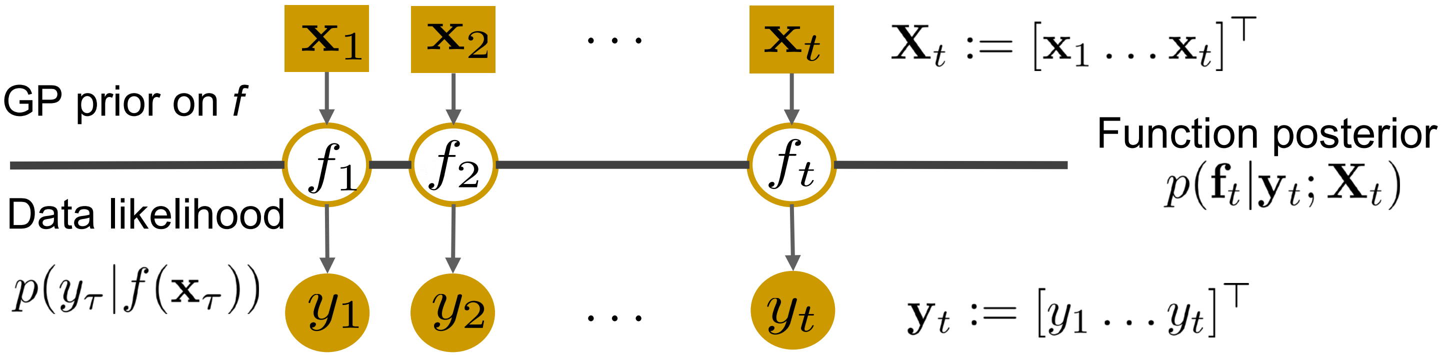
Q3. Efficient solvers? **Q4**. Performance analysis?

- Bayesian view is well motivated!

GP-based learning

Model: View learning function *f* as <u>random</u> with GP <u>prior</u>

(as0) $$f \sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}')) \Leftrightarrow \mathbf{f}_t := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_t)]^\top \sim \mathcal{N}(\mathbf{f}_t; \mathbf{0}_t, \mathbf{K}_t)$$
$$[\mathbf{K}_t]_{ij} = \text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) := \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad \forall t$$



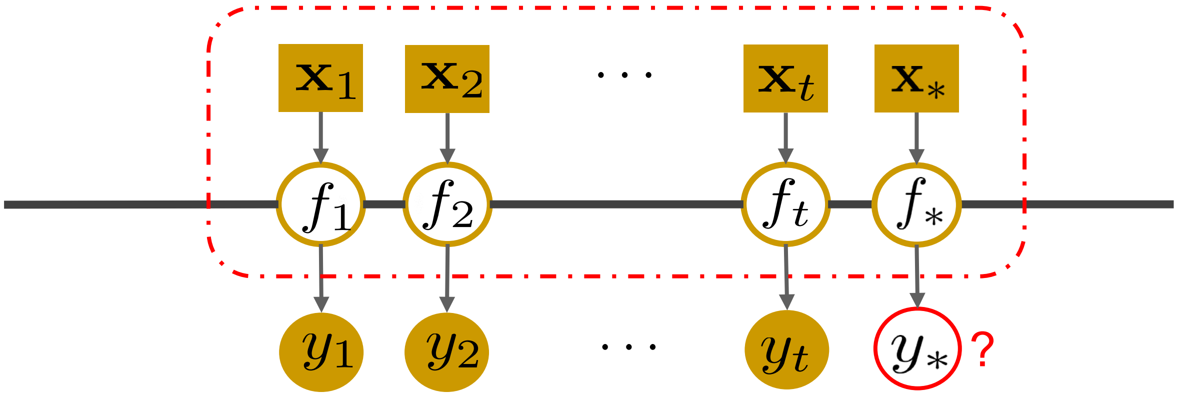
(as1) Likelihood $p(\mathbf{y}_t | \mathbf{f}_t; \mathbf{X}_t) = \prod_{\tau=1}^{t} p(y_\tau | f(\mathbf{x}_\tau))$

Goal: Learn <u>posterior</u> pdf of *f* using Bayes' rule $p(\mathbf{f}_t | \mathbf{y}_t; \mathbf{X}_t) \propto p(\mathbf{f}_t; \mathbf{X}_t) p(\mathbf{y}_t | \mathbf{f}_t; \mathbf{X}_t)$

C. Rasmussen, C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, 2006.

GP-based inference

Goal: Given training data $\{\mathbf{X}_t, \mathbf{y}_t\}$ and test input $\mathbf{x}_*$, infer (pdf of) y_*



S1. Posterior pdf of function value at test input

$$p(f_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) = \int \underbrace{p(f_*|\mathbf{f}_t;\mathbf{X}_t,\mathbf{x}_*)}_{\text{`transition prior'}} \overbrace{p(\mathbf{f}_t|\mathbf{y}_t;\mathbf{X}_t)}^{\text{computable posterior}} d\mathbf{f}_t$$

$$\mathcal{N}(f_*;\mathbf{k}_*^\top \mathbf{K}_t^{-1}\mathbf{f}_t, \kappa_{**} - \mathbf{k}_*^\top \mathbf{K}_t^{-1}\mathbf{k}_*)$$

S2. Posterior pdf of test output

$$p(y_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) = \int \overbrace{p(y_*|f(\mathbf{x}_*))}^{\text{likelihood}} p(f_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) df_*$$

- Numerical or MC sampling for non-Gaussian likelihoods

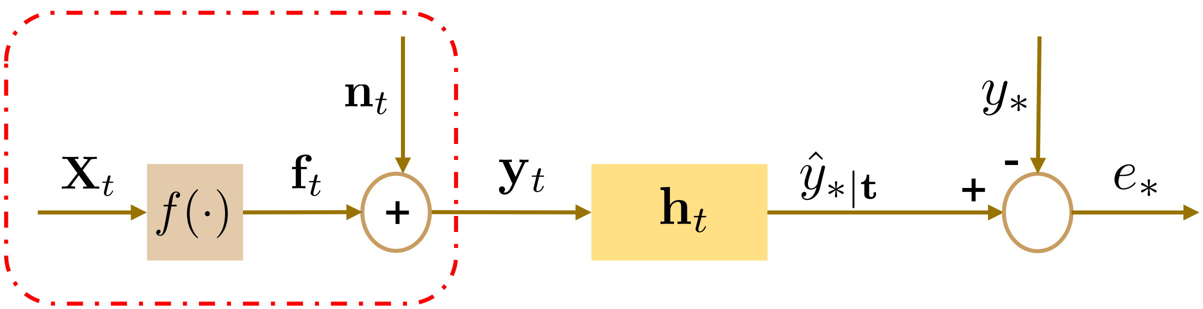
GP regression predictor

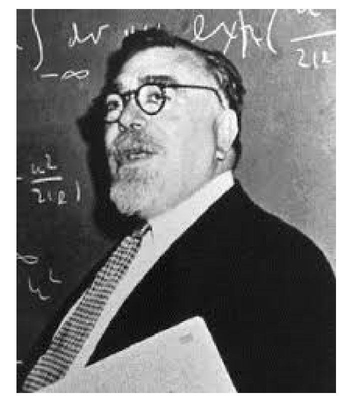
- If likelihood also Gaussian, then $p(y_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) = \mathcal{N}(y_*;\hat{y}_{*|\mathbf{t}},\sigma^2_{*|\mathbf{t}})$

  - Mean and variance in closed form!

$$\hat{y}_{*|\mathbf{t}} = \mathbf{k}_*^\top(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{y}_t$$
$$\sigma^2_{*|\mathbf{t}} = \kappa_{**} - \mathbf{k}_*^\top(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{k}_* + \sigma_n^2$$

- Wiener filtering





  - $\mathbf{h}_t = \mathrm{cov}^{-1}(\mathbf{y}_t)\mathrm{cov}(\mathbf{y}_t, y_*) = (\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\mathbf{k}_*$

GP-based classifier

Challenge: likelihood is non-Gaussian; e.g., logistic $p(y_t|f(\mathbf{x}_t)) = \dfrac{1}{1+e^{-y_t f(\mathbf{x}_t)}}$

- Gaussian approximation of non-Gaussian posterior [Williams et al.'98]

S0. $$p(\mathbf{f}_t|\mathbf{y}_t;\mathbf{X}_t) \approx \mathcal{N}(\mathbf{f}_t;\hat{\mathbf{f}}_t,\mathbf{\Sigma}_t)$$

$$\hat{\mathbf{f}}_t = \arg\max_{\mathbf{f}_t} \ln p(\mathbf{y}_t|\mathbf{f}_t;\mathbf{X}_t) + \ln p(\mathbf{f}_t;\mathbf{X}_t)$$

$$\mathbf{\Sigma}_t^{-1} = \mathbf{K}_t^{-1} - \nabla^2 \ln p(\mathbf{y}_t|\mathbf{f}_t;\mathbf{X}_t)|_{\mathbf{f}_t=\hat{\mathbf{f}}_t}$$

S1. $$p(f_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) = \int p(f_*|\mathbf{f}_t;\mathbf{X}_t,\mathbf{x}_*)p(\mathbf{f}_t|\mathbf{y}_t;\mathbf{X}_t)d\mathbf{f}_t \approx \mathcal{N}(f_*;\hat{f}_{*|\mathbf{t}},\sigma^2_{f_*|\mathbf{t}})$$

S2. $$p(y_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) \approx \int p(y_*|f(\mathbf{x}_*))p(f_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*)df_*$$

  - Numerical or MC sampling approximation

C. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE T-PAMI,* 1998.
T. P. Minka, *A family of algorithms for approximate Bayesian inference*, PhD thesis, MIT, 2001.

GP kernel adaptivity and scalability

- Kernel (hyper) parameters; e.g., $\boldsymbol{\alpha} := [\sigma_\kappa^2, \sigma_n^2]^\top$

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}} \; p(\mathbf{y}_t; \mathbf{X}_t, \boldsymbol{\alpha}) = \int p(\mathbf{y}_t | \mathbf{f}_t; \mathbf{X}_t) p(\mathbf{f}_t; \mathbf{X}_t) d\mathbf{f}_t$$

- For GP regression $p(\mathbf{y}_t; \mathbf{X}_t, \boldsymbol{\alpha}) = \mathcal{N}(\mathbf{y}_t; \mathbf{0}_t, \mathbf{K}_t + \sigma_n^2 \mathbf{I}_t)$
  - ***K**$_t$* selection decoupled from ***f**$_t$* estimation; Gaussian approx. for classification

- Curse of dimensionality (CoD)

$$\hat{y}_{*|\mathbf{t}} = \mathbf{k}_*^\top (\mathbf{K}_t + \sigma_n^2 \mathbf{I}_t)^{-1} \mathbf{y}_t$$
$$\sigma_{*|\mathbf{t}}^2 = \kappa_{**} - \mathbf{k}_*^\top (\mathbf{K}_t + \sigma_n^2 \mathbf{I}_t)^{-1} \mathbf{k}_* + \sigma_n^2$$

  - Complexity $\mathcal{O}(t^3)$; storage $\mathcal{O}(t^2)$
  - CoD also in kernel selection

Remedies: low-rank or structured ***K**$_t$* approximants [Quiñonero-Candela et al.'05], [Titsias'09], [Lázaro-Gredilla et al.'10], [Wilson et al.'15], [Nickisch et al.'18]

J. Quiñonero-Candela et al., "A unifying view of sparse approximate Gaussian process regression," *JMLR*, 2005.
M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," *AISTATS*, 2009.
A. Wilson et al., "Kernel interpolation for scalable structured Gaussian processes (KISS-GP)," *ICML*, 2015.

Random features via Fourier spectrum

RF1. Draw *D* random vectors from the kernel's Fourier transform

$$\mathbf{v}_i \sim \pi(\mathbf{v}) = \mathcal{F}(\bar{\kappa}) \,, \quad i = 1, \ldots, D$$

RF2. Form *2D*x*1* **random feature** (RF) vector

$$\phi_{\mathbf{v}}(\mathbf{x}) := \frac{1}{\sqrt{D}} \left[\sin(\mathbf{v}_1^\top \mathbf{x}), \cos(\mathbf{v}_1^\top \mathbf{x}), \ldots, \sin(\mathbf{v}_D^\top \mathbf{x}), \cos(\mathbf{v}_D^\top \mathbf{x})\right]^\top$$

- RF-based linear kernel approximant $\check{\kappa}(\mathbf{x}, \mathbf{x}') = \phi_{\mathbf{v}}^\top(\mathbf{x})\phi_{\mathbf{v}}(\mathbf{x}')$

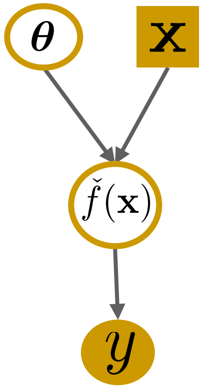
Key idea: Random linear function $\check{f}(\mathbf{x}) = \phi_{\mathbf{v}}^\top(\mathbf{x})\boldsymbol{\theta}$, $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}_{2D}, \sigma_\theta^2 \mathbf{I}_{2D})$

is a parametric GP with $\mathrm{cov}(\check{f}(\mathbf{x}_i), \check{f}(\mathbf{x}_j)) = \sigma_\theta^2 \phi_{\mathbf{v}}^\top(\mathbf{x}_i)\phi_{\mathbf{v}}(\mathbf{x}_j)$

- Prior $p(\check{\mathbf{f}}_t; \mathbf{X}_t) = \mathcal{N}(\check{\mathbf{f}}_t; \mathbf{0}_t, \sigma_\theta^2 \underbrace{\boldsymbol{\Phi}_t \boldsymbol{\Phi}_t^\top}_{2D\text{-rank approx. of } \boldsymbol{K}_t})$ $\quad \boldsymbol{\Phi}_t := [\phi_{\mathbf{v}}(\mathbf{x}_1), \ldots, \phi_{\mathbf{v}}(\mathbf{x}_t)]^\top$

A. Rahimi and B. Recht, "Random features for large scale kernel machines," *Proc. Advances in Neural Info. Process. Syst.*, pp. 117-1184, Canada, Dec. 2008.

RF-driven parametric GPs



- Parametric generative model

Vanilla GP: $f \sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}'))$ $\qquad p(y|f(\mathbf{x}))$

RF-based GP:

$$\check{f}(\mathbf{x}) = \boldsymbol{\phi}_{\mathbf{v}}^{\top}(\mathbf{x})\boldsymbol{\theta}$$
$$\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}_{2D}, \sigma_{\theta}^2 \mathbf{I}_{2D})$$

$$p(y|\boldsymbol{\theta}; \mathbf{x}) := p(y|\check{f}(\mathbf{x}))$$

- Batch GPR predictor

$$\hat{y}_{*|\mathbf{t}} = \boldsymbol{\phi}_{\mathbf{v}}^{\top}(\mathbf{x}_*) \left(\boldsymbol{\Phi}_t^{\top} \boldsymbol{\Phi}_t + \frac{\sigma_n^2}{\sigma_\theta^2} \mathbf{I}_{2D} \right)^{-1} \boldsymbol{\Phi}_t^{\top} \mathbf{y}_t$$

$$\sigma_{*|\mathbf{t}}^2 = \boldsymbol{\phi}_{\mathbf{v}}^{\top}(\mathbf{x}_*) \left(\frac{\boldsymbol{\Phi}_t^{\top} \boldsymbol{\Phi}_t}{\sigma_n^2} + \frac{\mathbf{I}_{2D}}{\sigma_\theta^2} \right)^{-1} \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_*) + \sigma_n^2$$

   - Complexity $\mathcal{O}(t(2D)^2 + (2D)^3)$: scalable especially for $t \gg 2D$

M. Lázaro-Gredilla et al., “Sparse spectrum Gaussian process regression,” *JMLR*, 2010.

Incremental RF-GP learning

- Propagate posterior of θ as in recursive Bayes [Gijsberts-Metta'13]

$$p(\boldsymbol{\theta}|\mathbf{y}_t;\mathbf{X}_t) \xrightarrow[\text{via } \mathbf{x}_{t+1}]{\text{predictive pdf}} p(y_{t+1}|\mathbf{y}_t;\mathbf{X}_{t+1}) \xrightarrow[\text{via } y_{t+1}]{\text{corrective pdf}} p(\boldsymbol{\theta}|\mathbf{y}_{t+1};\mathbf{X}_{t+1})$$

$$p(y_{t+1}|\mathbf{y}_t;\mathbf{X}_{t+1}) = \int p(y_{t+1}|\boldsymbol{\theta};\mathbf{x}_{t+1})p(\boldsymbol{\theta}|\mathbf{y}_t;\mathbf{X}_t)d\boldsymbol{\theta}$$

$$p(\boldsymbol{\theta}|\mathbf{y}_{t+1};\mathbf{X}_{t+1}) = \frac{p(\boldsymbol{\theta}|\mathbf{y}_t;\mathbf{X}_t)p(y_{t+1}|\boldsymbol{\theta};\mathbf{x}_{t+1})}{p(y_{t+1}|\mathbf{y}_t;\mathbf{X}_{t+1})}$$

- GPR

$$\mathcal{N}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}_t,\boldsymbol{\Sigma}_t) \longrightarrow \mathcal{N}(y_{t+1};\hat{y}_{t+1|t},\sigma^2_{t+1|t}) \longrightarrow \mathcal{N}(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}_{t+1},\boldsymbol{\Sigma}_{t+1})$$

$$\hat{y}_{t+1|t} = \boldsymbol{\phi}_{t+1}^\top \hat{\boldsymbol{\theta}}_t \qquad \hat{\boldsymbol{\theta}}_{t+1} = \hat{\boldsymbol{\theta}}_t + \sigma^{-2}_{t+1|t}\boldsymbol{\Sigma}_t\boldsymbol{\phi}_{t+1}(y_{t+1} - \hat{y}_{t+1|t})$$

$$\sigma^2_{t+1|t} = \boldsymbol{\phi}_{t+1}^\top\boldsymbol{\Sigma}_t\boldsymbol{\phi}_{t+1} + \sigma_n^2 \qquad \boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t - \sigma^{-2}_{t+1|t}\boldsymbol{\Sigma}_t\boldsymbol{\phi}_{t+1}\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\Sigma}_t$$

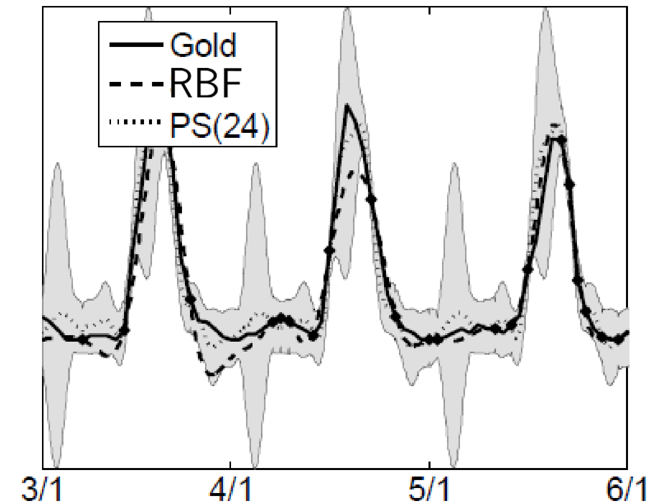
- Complexity $\mathcal{O}(t(2D)^2)$

A. Gijsberts and G. Metta, "Real-time model learning using incremental sparse spectrum GPR," *Neural Networks*, 2013.

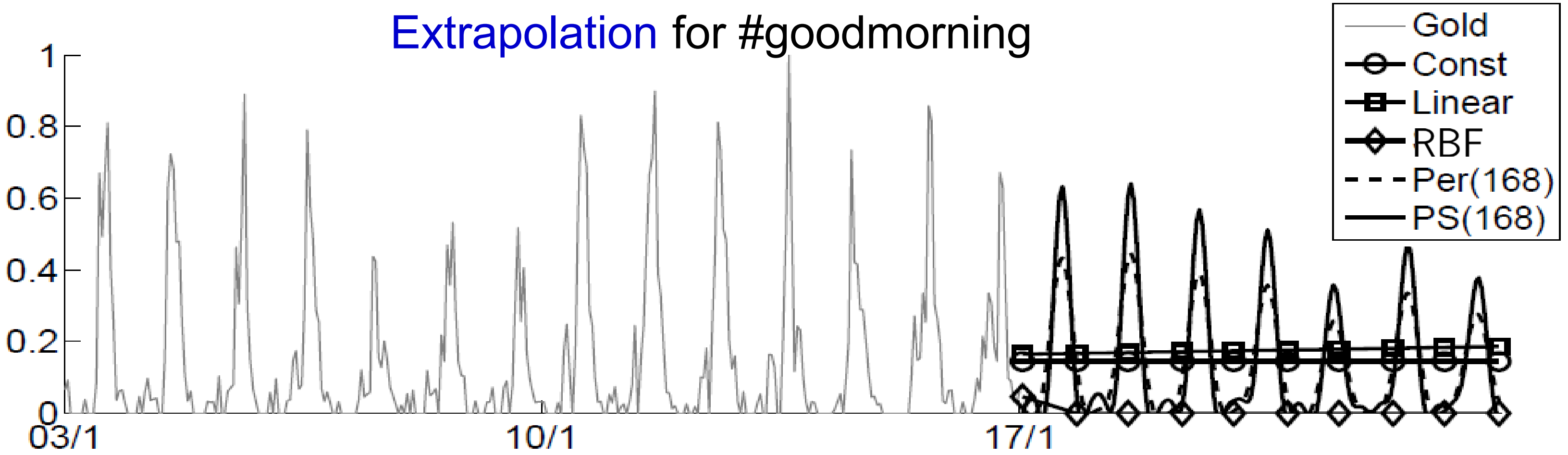
Hashtag popularity

- GPR model trained per hashtag
  - $x_{h,t}$ timestamp of hashtag *h* with $y_{h,t}$ occurrences
- Can also predict hashtag from tweet

Interpolation for #goodmorning



Extrapolation for #goodmorning



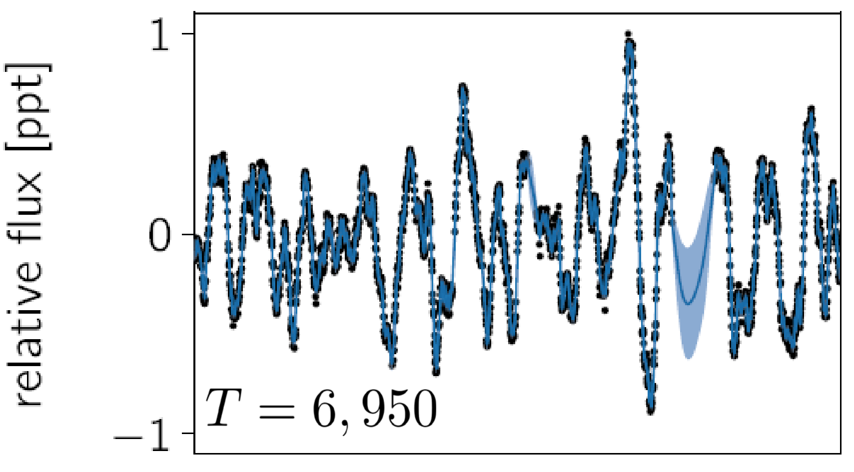
D. Preotiuc-Pietro and T. Cohn, "A temporal model of text periodicities using Gaussian processes," *Proc. Conf. on Empirical Methods of NLP*, pp. 977-988, Seattle, WA, 2013.

Astronomical time series modeling

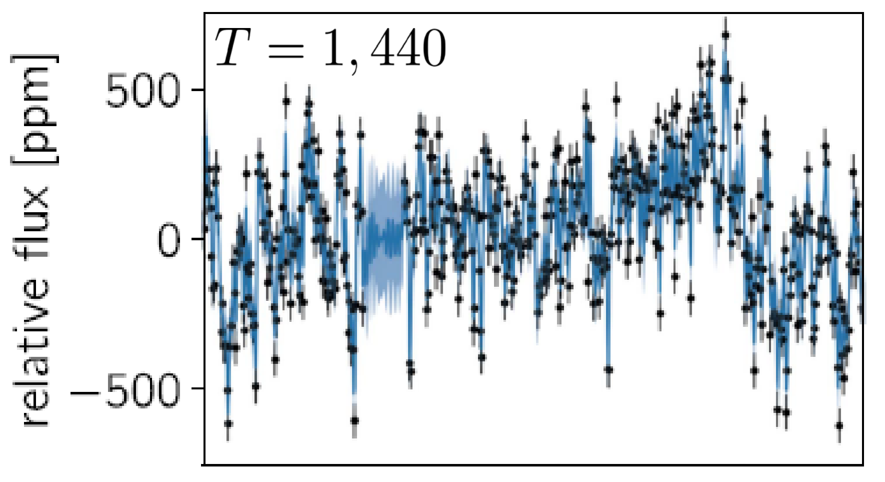
- GPs used for exoplanet discovery and characterization
  - x_t timestamp with y_t astronomical observation at t

- Special kernel matrix (tridiagonal) can afford large-scale KF-type inversion

y_t : Stellar rotation



y_t : Astroseismic oscillations

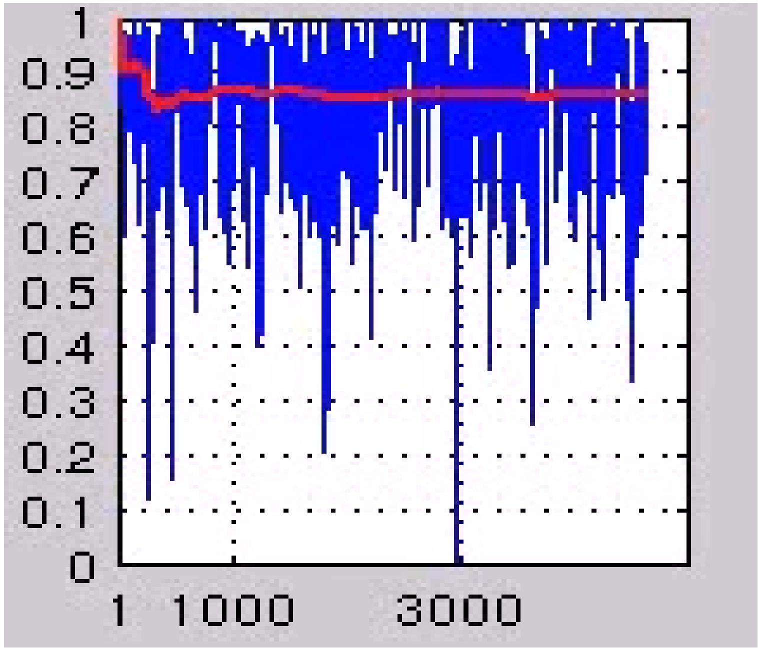


D. Foreman-Mackey, E. Agol, S. Ambikasaran, and R. Angus, Fast and scalable Gaussian process modeling with applications to astronomical time series, *The Astronomical Journal*, 2017.

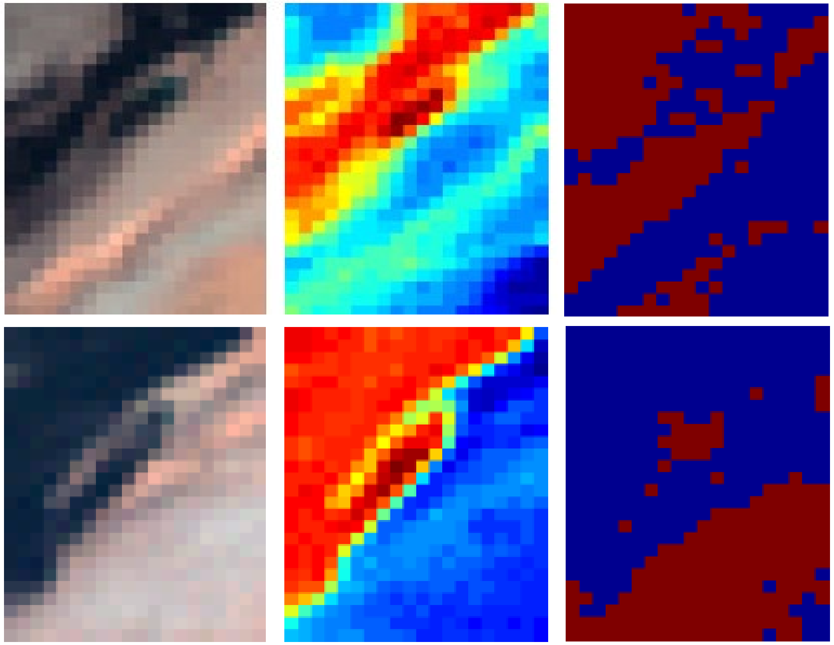
GP classification for remote sensing

- Classify whether pixels of multispectral images belong to clouds or not
- Large-scale imagery prompts RF approximation for GPs
  - $\boldsymbol{x}_t$: multispectral features per pixel; $y_t \in \{0,1\}$ labels (annotated for training)

Classification rate vs frame no.



RGB Infrared GP classifier



P. Morales-Álvarez, A. Pérez-Suay, R. Molina, G. Camps-Valls, "Remote sensing image classification with large-scale Gaussian processes," *IEEE Trans. Geoscience and Remote Sensing*, pp. 1103–14, 2018.

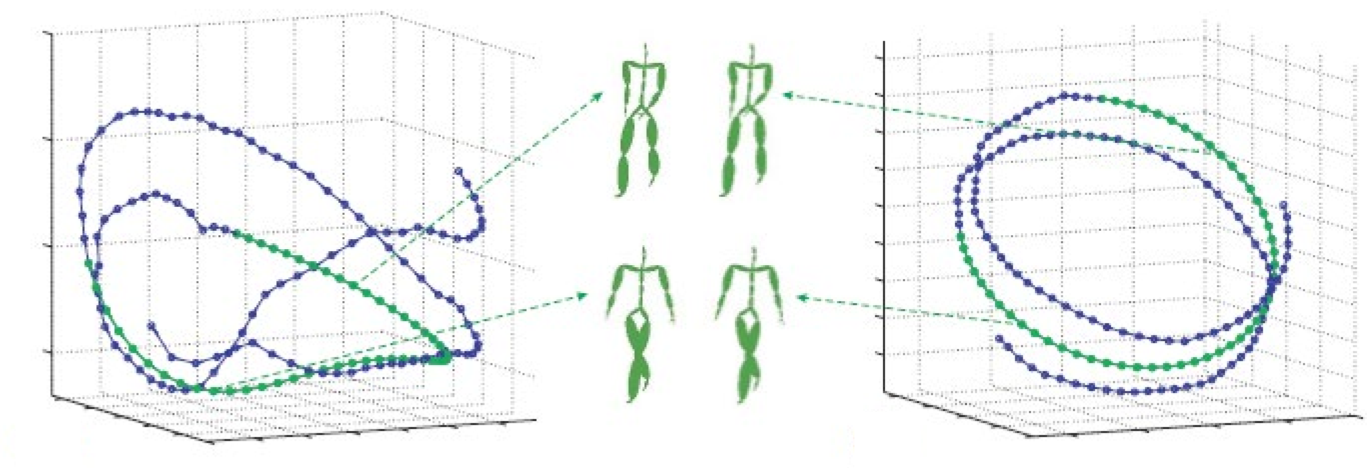
GPs for dynamic state estimation

$$\mathbf{x}_{t+1} = g(\mathbf{x}_t) + \mathbf{w}_{x,t}$$
$$\mathbf{y}_{t+1} = f(\mathbf{x}_{t+1}) + \mathbf{w}_{y,t+1}$$

Goal: Given observations $\boldsymbol{y}_t$, estimate $\boldsymbol{x}_t$ (offline) using GP models for *f* and *g*

- GP models can extrapolate and interpolate missing data

- Blue dots are state estimates
- Green dots are state prediction



- Gaussian kernel
- Linear kernel

J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian Process Dynamical Models," *Proc. NIPS*, pp 1441–1448, 2006.

Deep neural networks

Q. How about parametric function estimators? **A.** E.g., Deep neural nets (DNNs)

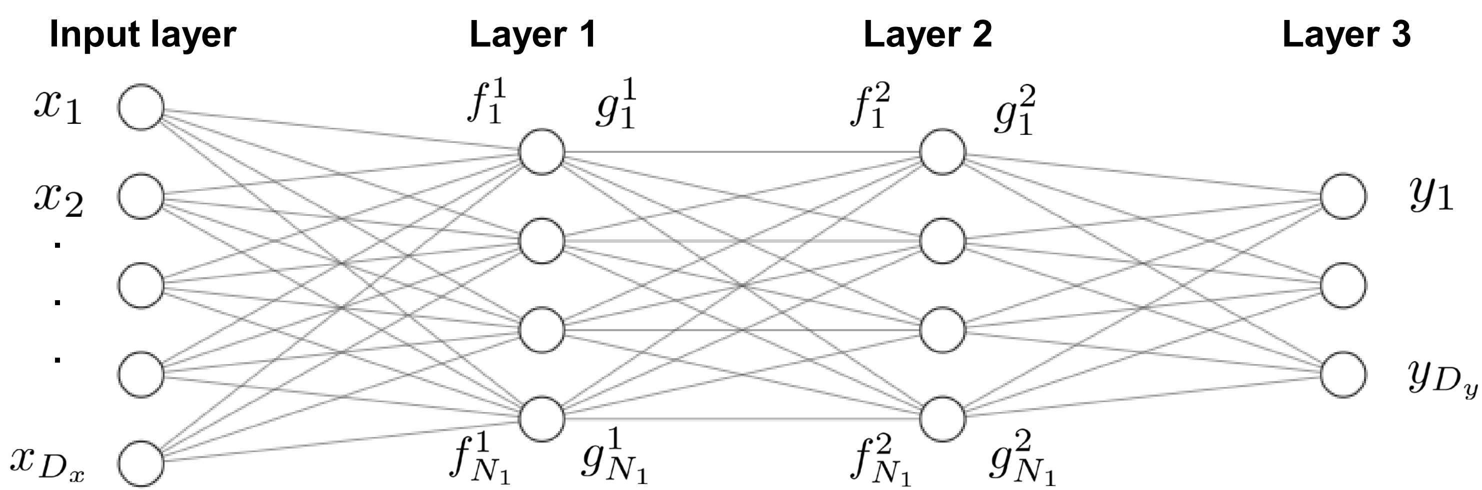
- First layer

$$f^1_\nu(\mathbf{x}) = \sum_{\nu'=1}^{D_x} w^1_{\nu,\nu'} x_{\nu'} + b^1_\nu \,, \qquad \nu = 1, \ldots, N_1$$

- Next layers

$$g^{\ell-1}_\nu(\mathbf{x}) = \varphi(f^{\ell-1}_\nu(\mathbf{x})), \qquad \nu = 1, \ldots, N_{\ell-1}$$

$$f^\ell_\nu(\mathbf{x}) = \sum_{\nu'=1}^{N_{\ell-1}} w^\ell_{\nu,\nu'} g^{\ell-1}_{\nu'}(\mathbf{x}) + b^\ell_\nu, \quad \nu = 1, \ldots, N_\ell$$



Bayesian neural networks (BNNs)

- Zero-mean Gaussian BNN parameters $\{w^{\ell}_{\nu,\nu'}, b^{\ell}_{\nu}\}$ with variances $\{\check{C}^{\ell}_{w}, \check{C}^{\ell}_{b}\}$
  - $w^{\ell}_{\nu,\nu'}, b^{\ell}_{\nu}$ independent across ν, ν'
  - For bounded variance per layer, normalize variances per neuron: $C^{\ell}_{w} := \frac{\check{C}^{\ell}_{w}}{N_{\ell-1}}$

$$C^{\ell}_{b} := \check{C}^{\ell}_{b}$$

$$w^{\ell}_{\nu,\nu'} \sim \mathcal{N}(0, C^{\ell}_{w})$$

$$b^{\ell}_{\nu} \sim \mathcal{N}(0, C^{\ell}_{b})$$

Proposition 1 [Neal'96] For *L*=2, if $\{g^{1}_{\nu}(\mathbf{x})\}_{\nu=1}^{N_1}$ have bounded variances, then as $N_1 \to \infty$ the output $\{f^{2}_{\nu}(\mathbf{x})\}_{\nu=1}^{N_2}$ (nonlinearity φ) converges in distr. to a 0-mean GP with

$$\mathbb{E}[f^{2}_{\nu}(\mathbf{x}) f^{2}_{\nu'}(\mathbf{x}')] = \delta_{\nu,\nu'}[\check{C}^{2}_{w}\mathbb{E}_{\mathbf{w},b}\{\varphi(\mathbf{w}^{\top}\mathbf{x}+b)\varphi(\mathbf{w}^{\top}\mathbf{x}'+b)\} + C^{2}_{b}]$$

R. M. Neal, *Bayesian Learning for Neural Networks,* Springer, 1996.

Sketch of the proof …

- For *L*=2

$$f_\nu^1(\mathbf{x}) = \sum_{\nu'=1}^{D_x} w_{\nu,\nu'}^1 x_{\nu'} + b_\nu^1 \qquad g_\nu^1(\mathbf{x}) = \varphi(f_\nu^1(\mathbf{x}))$$

$$f_\nu^2(\mathbf{x}) = \sum_{\nu'=1}^{N_1} w_{\nu,\nu'}^2 g_{\nu'}^1(\mathbf{x}) + b_\nu^2$$

- Gaussian BNN parameters $b \sim \mathcal{N}(b; 0, C_b^1), \quad \mathbf{w} \sim \mathcal{N}(\mathbf{w}; \mathbf{0}, C_w^1 \mathbf{I}_{D_x})$

- Central limit theorem asserts as $N_1 \to \infty$ a Gaussian pdf with mean and variance

$$\mathbb{E}[f_\nu^2(\mathbf{x})] = 0$$

$$\mathbb{E}[f_\nu^2(\mathbf{x}) f_{\nu'}^2(\mathbf{x}')] = \delta_{\nu,\nu'}[\check{C}_w^2 \mathbb{E}_{\mathbf{w},b}\{\varphi(\mathbf{w}^\top \mathbf{x} + b)\varphi(\mathbf{w}^\top \mathbf{x}' + b)\} + C_b^2]$$

- Likewise for *t* training vectors $[\mathbf{f}^2(\mathbf{x}_1), \mathbf{f}^2(\mathbf{x}_2), \ldots, \mathbf{f}^2(\mathbf{x}_t)]^\top$

R. M. Neal, *Bayesian Learning for Neural Networks,* Springer, 1996.

Normal limiting distribution across layers

Proposition 2. *If the* $(\ell-1)$*st layer input is Gaussian distributed with mean and variance*

$$\mathbb{E}[f_\nu^{\ell-1}(\mathbf{x})] = 0$$
$$\mathbb{E}[f_\nu^{\ell-1}(\mathbf{x}) f_{\nu'}^{\ell-1}(\mathbf{x}')] = \delta_{\nu,\nu'} \kappa(\mathbf{x},\mathbf{x}')$$

$$\kappa(\mathbf{x},\mathbf{x}') := \check{C}_w^{\ell-1} \mathbb{E}_{\epsilon^{\ell-1}(\mathbf{x}),\epsilon^{\ell-1}(\mathbf{x}')} \{\varphi(\epsilon^{\ell-1}(\mathbf{x}))\varphi(\epsilon^{\ell-1}(\mathbf{x}'))\} + C_b^{\ell-1}$$

$$\epsilon^{\ell-1}(\mathbf{x}) := [g_1^{\ell-2}(\mathbf{x}), \ldots, g_{N_{\ell-2}}^{\ell-2}(\mathbf{x})]^\top \mathbf{w} + b$$

$$b \sim \mathcal{N}(b; 0, C_b^{\ell-2}),\ \mathbf{w} \sim \mathcal{N}(\mathbf{w}; \mathbf{0}, C_w^{\ell-2} \mathbf{I}_{N_{l-2}})$$

then as $N_{\ell-1} \to \infty$ *the limiting pdf of the l-th layer input is also Gaussian with*

$$\mathbb{E}[f_\nu^{\ell}(\mathbf{x})] = 0$$
$$\mathbb{E}[f_\nu^{\ell}(\mathbf{x}) f_{\nu'}^{\ell}(\mathbf{x}')] = \delta_{\nu,\nu'} [\check{C}_w^{\ell} \mathbb{E}\{\varphi(\epsilon^{\ell}(\mathbf{x})\varphi(\epsilon^{\ell}(\mathbf{x}'))\} + C_b^{\ell}]$$

- Limiting GP has recursively computable kernels

T. Hazan and T. Jaakkola. "Steps Toward Deep Kernel Methods from Infinite Neural Networks," *ArXiv,* 2015
A.G. Matthews, et al., "Gaussian Processes Behavior in Wide Deep Neural Networks," *ICLR*, 2018.

Deep BNNs vis-a-vis GPs

Q. How about finite N_ℓ?

- *Width function* h_ℓ *:* $N_\ell = h_\ell(t)$

Theorem. *For a BNN with **ReLU** as* φ *and any* $\{\mathbf{x}_\tau\}_{\tau=1}^t$ *there are strictly increasing* $\{h_\ell(t)\}_{\ell=1}^L$ *and thus* $\{N_\ell\}_{\ell=1}^L$ *, so that as* $t \to \infty$ *the NN output pdf converges to a GP with kernel* $\kappa(\mathbf{x}, \mathbf{x}') = \check{C}_w^\ell \mathbb{E}\{\varphi(\epsilon^\ell(\mathbf{x}))\varphi(\epsilon^\ell(\mathbf{x}'))\} + C_b^\ell$

A.G. Matthews, et al., "Gaussian Processes Behaviour in Wide Deep Neural Networks," *ICLR*, 2018.

Deep BNNs versus GPs - Empirical comparison

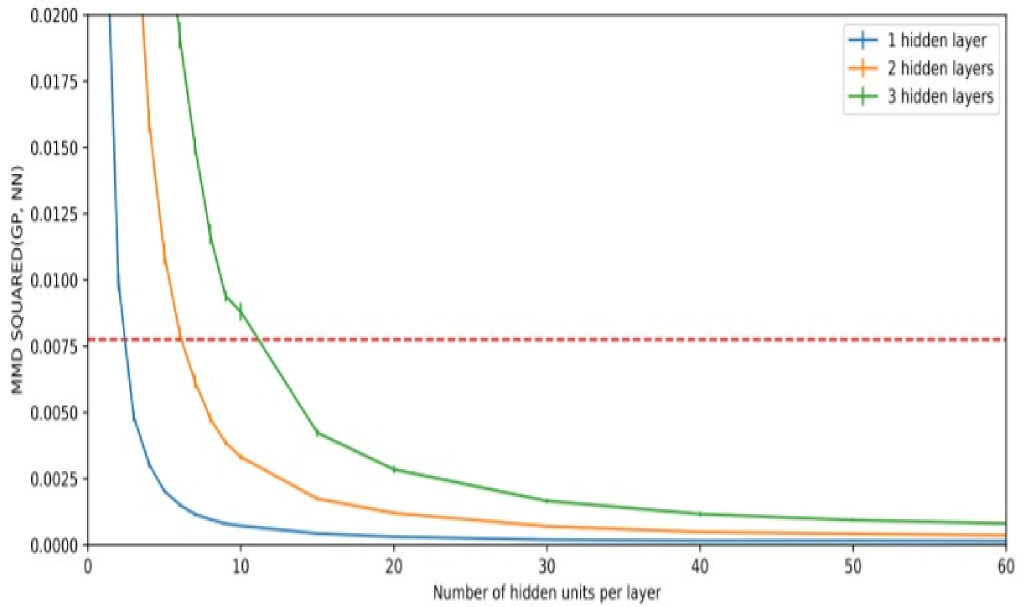
- Compare $p^{\mathrm{BNN}}(\mathbf{y};\mathbf{x})$ and $p^{\mathrm{GP}}(\mathbf{y};\mathbf{x})$ using maximum mean discrepancy metric

$$\mathcal{MMD}(p^{\mathrm{BNN}}, p^{\mathrm{GP}}, \mathcal{F}) = \sup_{g\in\mathcal{F}}[\mathbb{E}_{\mathbf{y}\sim p^{\mathrm{BNN}}}[g(\mathbf{y})] - \mathbb{E}_{\mathbf{y}\sim p^{\mathrm{GP}}}[g(\mathbf{y})]]$$

- Sample estimator over κ-induced RHKS functions (in $\mathcal{F}$) [Gretton etal'12]

$$\widehat{\mathcal{MMD}}^2(p^{\mathrm{BNN}}, p^{\mathrm{GP}}, \mathcal{F}) = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m}\kappa(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\kappa(\tilde{\mathbf{y}}'_i, \tilde{\mathbf{y}}'_j) - 2\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\kappa(\tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}'_j)$$

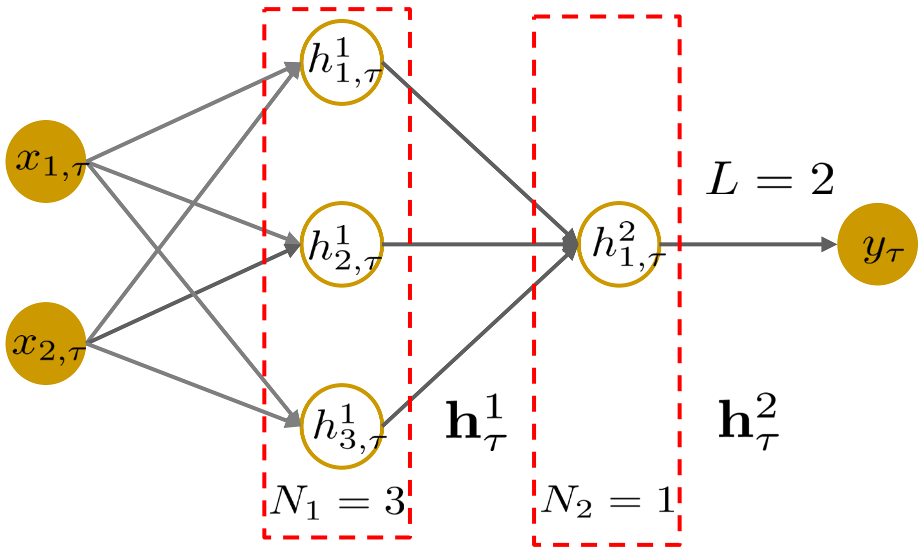
- Draw $\tilde{\mathbf{y}}_i \sim p^{\mathrm{BNN}}$ and $\tilde{\mathbf{y}}'_j \sim p^{\mathrm{GP}}$
- Sample *MMD*^{2} versus number of neurons per layer
- Faster convergence for wider and shallower BNNs



A. Gretton, et al., "*A Kernel Two-sample test,*" *Journal of Machine Learning Research*, 2012.
A.G. Matthews, et al., "Gaussian Processes Behaviour in Wide Deep Neural Networks," *ICLR*, 2018.

Going deep…

- Deep (D) GPs: cascade of *L*-layer GPs to boost expressiveness



$$h^\ell_{i,\tau} = f^\ell_i(\mathbf{h}^{\ell-1}_\tau) \qquad \ell = 1, \ldots, L$$

$$f^\ell_i \sim \mathcal{GP}(0, \kappa^\ell_i) \qquad i = 1, \ldots, N_\ell \qquad \tau = 1, \ldots, t$$

$$\mathbf{h}^\ell_\tau := [h^\ell_{1,\tau} \ldots h^\ell_{N_\ell,\tau}]^\top$$

$$\mathbf{H}^\ell_t := [\mathbf{h}^\ell_1 \ldots \mathbf{h}^\ell_t]^\top \in \mathbb{R}^{N_\ell \times t}$$

$$\mathbf{H}^0_t = \mathbf{X}_t \in \mathbb{R}^{d \times t}$$

DGP prior (non-Gaussian) $$p(\mathbf{H}^L_t; \mathbf{X}_t) = \int \prod_{\ell=1}^{L} p(\mathbf{H}^\ell_t | \mathbf{H}^{\ell-1}_t) \, d\mathbf{H}^{L-1}_t \ldots d\mathbf{H}^1_t$$

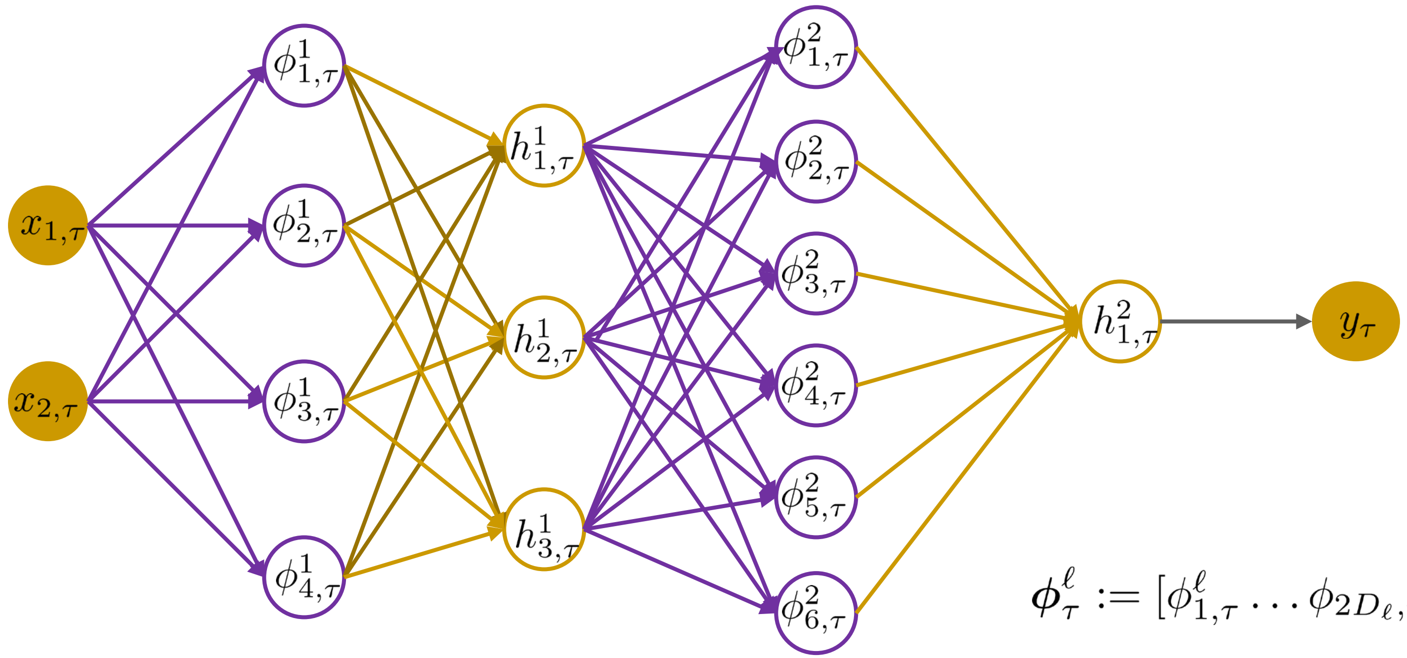
Likelihood $$p(\mathbf{y}_t | \mathbf{H}^L_t; \mathbf{X}_t) = \prod_{\tau=1}^{t} p(y_\tau | h^L_{1,\tau}; \mathbf{x}_\tau)$$

- Intractable integration due to CoD

A. Damianou, and N. Lawrence, "Deep Gaussian processes," *Proc. of AISTATS*, pp. 207-215, Scottsdale, AZ, April 2013.

RF-based DGPs

- Common kernel across each layer nodes $f_i^\ell \sim \mathcal{GP}(0, \kappa^\ell)$



$$\boldsymbol{\phi}^\ell_\tau := [\phi^\ell_{1,\tau} \dots \phi_{2D_\ell,\tau}]^\top$$

Kernel parameters

- Parametric layer-to-layer mapping $\mathbf{h}^\ell_\tau = \boldsymbol{\Theta}^\ell \boldsymbol{\phi}^\ell_\tau(\mathbf{h}^{\ell-1}_\tau; \boldsymbol{\alpha}^\ell, \{\mathbf{v}^\ell_d\}_{d=1}^{D_\ell})$
- Per-datum likelihood $p(y_\tau | \boldsymbol{\Theta}; \mathbf{x}_\tau)$, $\boldsymbol{\Theta} := \{\boldsymbol{\Theta}^\ell\}_{\ell=1}^L$

$$\boldsymbol{\Theta}^\ell := [\boldsymbol{\theta}^\ell_1 \dots \boldsymbol{\theta}^\ell_{N_\ell}]^\top \in \mathbb{R}^{N_\ell \times 2D_\ell}$$

$$\boldsymbol{\theta}^\ell_i \sim \mathcal{N}(\mathbf{0}_{2D_\ell}, \mathbf{I}_{2D_\ell})$$

K. Cutajar, E.V. Bonilla, P. Michiardi, and M. Filippone, "Random feature expansions for deep Gaussian processes," Proc. *ICML*, pp. 884–893, Fort Lauderdale, FL, April 2017.

Training and testing with DGPs

Training: find $\{\boldsymbol{\alpha}^\ell\}$ and $p(\boldsymbol{\Theta}|\mathbf{y}_t;\mathbf{X}_t)$ using variational inference

- Approximate intractable $p(\boldsymbol{\Theta}|\mathbf{y}_t;\mathbf{X}_t)$ with tractable $q(\boldsymbol{\Theta}) = \prod_{\ell=1}^{L}\prod_{i=1}^{N_\ell}\prod_{d=1}^{2D_\ell}\mathcal{N}(\theta_{di}^\ell;\mu_{di}^\ell, s_{di}^\ell)$

$$(\{\hat{\boldsymbol{\alpha}}^\ell\},\{\hat{\mu}_{di}^\ell\},\{\hat{s}_{di}^\ell\}) = \arg\max_{\{\boldsymbol{\alpha}^\ell\},\{\mu_{di}^\ell\},\{s_{di}^\ell\}} \frac{1}{R}\sum_{r=1}^{R}\sum_{\tau=1}^{t}\log p(y_\tau|\tilde{\boldsymbol{\Theta}}_r;\mathbf{x}_\tau,\{\boldsymbol{\alpha}^\ell\}) + \frac{1}{2}\sum_{\ell=1}^{L}\sum_{i=1}^{N_\ell}\sum_{d=1}^{2D_\ell}(1+\log(s_{di}^\ell)-(\mu_{di}^\ell)^2-s_{di}^\ell)$$

- Solvable via stochastic optimization

$$\tilde{\theta}_{di,r}^\ell = \mu_{di}^\ell + \sqrt{s_{di}^\ell}\tilde{\epsilon}_{di,r}^\ell$$
$$\tilde{\epsilon}_{di,r}^\ell \sim \mathcal{N}(0,1)$$

Testing: draw realizations $\tilde{\boldsymbol{\Theta}}_r \overset{i.i.d.}{\sim} q(\boldsymbol{\Theta})$ to obtain output posterior pdf

$$p(y_*|\mathbf{y}_t;\mathbf{X}_t,\mathbf{x}_*) \approx \int p(y_*|\boldsymbol{\Theta};\mathbf{x}_*,\{\hat{\boldsymbol{\alpha}}^\ell\})q(\boldsymbol{\Theta})d\boldsymbol{\Theta} \approx \frac{1}{R}\sum_{r=1}^{R}p(y_*|\tilde{\boldsymbol{\Theta}}_r;\mathbf{x}_*,\{\hat{\boldsymbol{\alpha}}^\ell\})$$

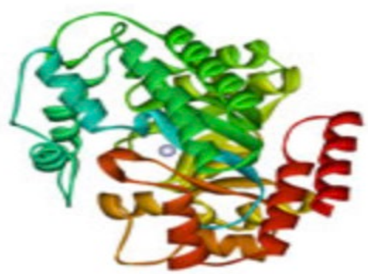
Testing DGP for regression

Benchmarks: DGP-EP [Bui et al.'16], VAR-GP [Hensman et al.'15], dropout-based DNN

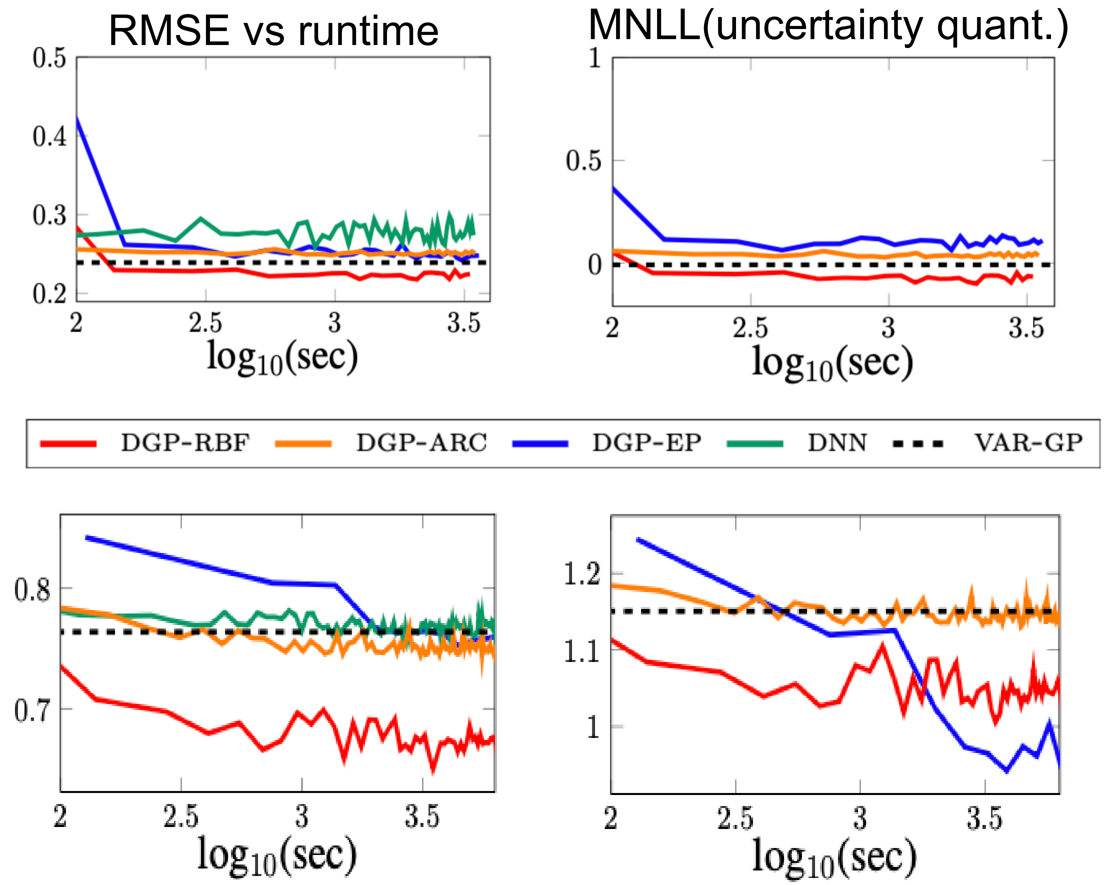
Powerplant (t=9,568, d=4)

$\boldsymbol{x}_\tau$: hourly ambient measurements
y_τ: electric energy output

Protein (t=45,730, d=9)



$\boldsymbol{x}_\tau$: protein structure attributes
y_τ: protein functionality



➢ RF-based DGPs lower RMSE and quantify uncertainty

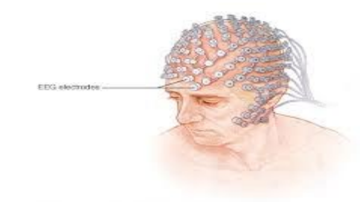
Bui et al., "Deep Gaussian Processes for Regression using Approximate Expectation Propagation," *ICML*, 2016.
Hensman et al., "Scalable variational Gaussian process classification," Proc. *AISTATS*, 2015.

Testing DGP for classification

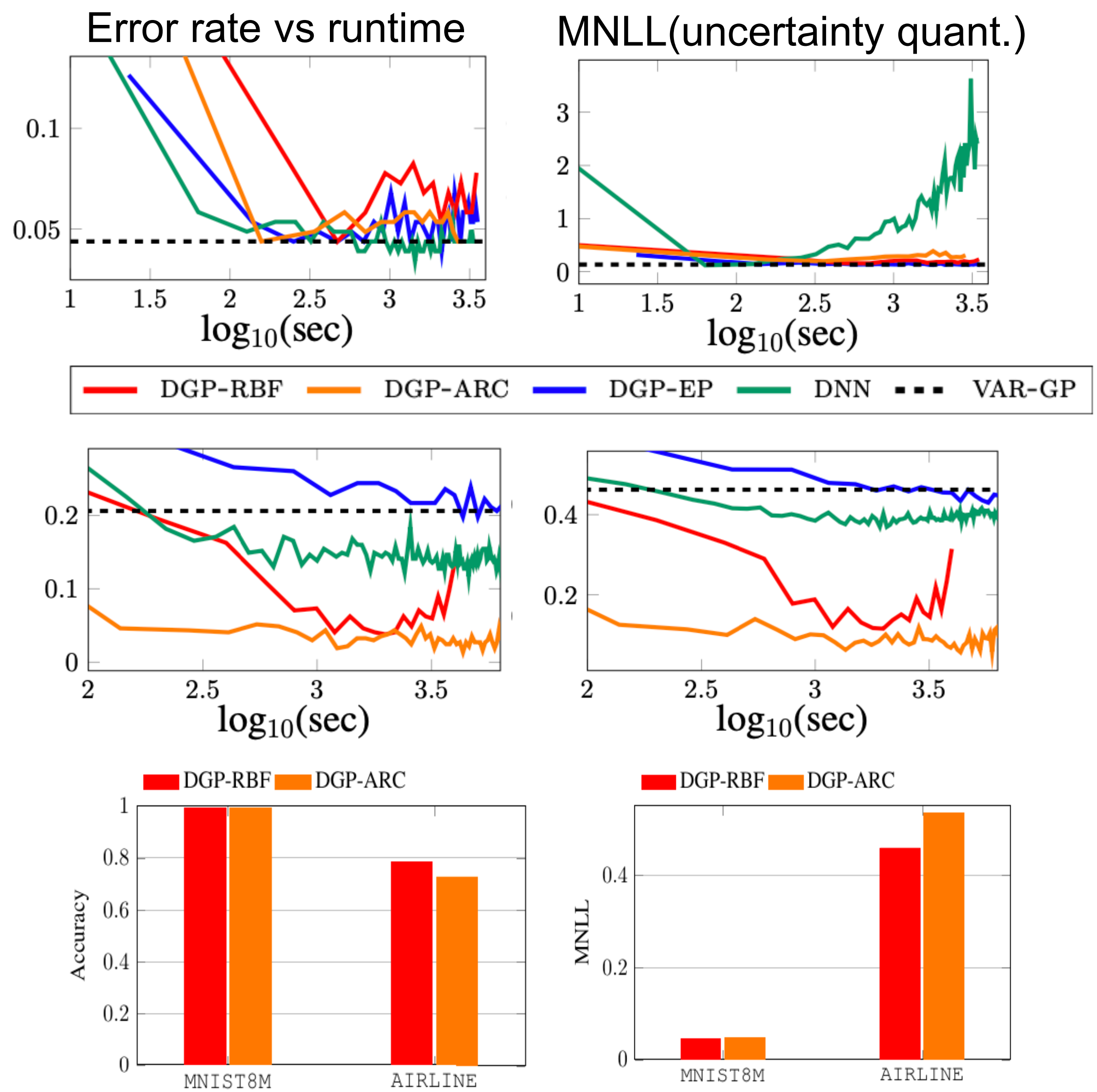
Spam (t=4,601, D_x=57)

$\boldsymbol{x}_\tau$: freq. of words/characters per email
y_τ: 1 (spam) or 0 (not spam)

EEG (t=14,979, D_x=14)



$\boldsymbol{x}_\tau$: measurements from 14 electrodes
y_τ: 1 (alcoholic) or 0 (not alcoholic)



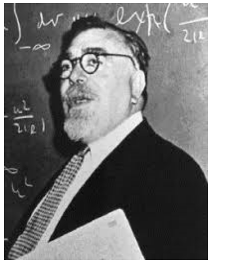
- RF-based DGPs scale well; exhibit lower error; and quantify uncertainty

Part II

- Incremental (online) and ensemble Gaussian processes (IE-GP)
  - IE-GP basics and analysis
  - Dynamic IE-GP learning
  - Unsupervised learning using (E) GPs
  - Graph-guided EGP-based learning

Motivation for incremental emsembles

- ❑ Uncertainty quantification and scalability



- ❑ Robustness to unknown dynamics

- ❑ Performance guarantees valid even in adversarial settings

- ❑ Adaptability to operational environments
  - ➢ Highly expressive model class
  - ➢ Online refinement of the model

Incremental **E**nsembles of **GP**s

Q. Lu, G. V. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian Processes with Spectral Features for Online Interactive Learning with Scalability," *Proc. of AISTATS*, Palermo, Italy, June 2020.

Ensemble GP learning

Q. How expressive is a single GP? **A.** The more the merrier …

- GP prior per learner *m* $\quad f|i=m \sim \mathcal{GP}(0, \kappa^m(\mathbf{x}, \mathbf{x}'))$
- Ensemble (E) GP prior $\quad \boxed{f \sim \sum_{m=1}^{M} w^m \mathcal{GP}(0, \kappa^m(\mathbf{x}, \mathbf{x}'))} \qquad \sum_{m=1}^{M} w^m = 1$
- RF-based EGP (non-Gaussian prior) $\quad \check{f} \mid \{\boldsymbol{\theta}^m\}_{m=1}^{M} \sim \sum_{m=1}^{M} w^m \delta(\check{f}(\mathbf{x}) - \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{x})\boldsymbol{\theta}^m), \; \boldsymbol{\theta}^m \sim \mathcal{N}(\boldsymbol{\theta}^m; \mathbf{0}, \sigma_{\theta^m}^2 \mathbf{I}_{2D})$

w_t^1 $\qquad$ w_t^M

$\cdots$

$p(\boldsymbol{\theta}^1|\mathbf{y}_t; \mathbf{X}_t)$ $\qquad\qquad$ $p(\boldsymbol{\theta}^M|\mathbf{y}_t; \mathbf{X}_t)$

- EGPs can model a richer space of learning functions
  - Meta-learner weighs experts using $\quad w_t^m := \Pr(i=m|\mathbf{y}_t; \mathbf{X}_t)$
  - Learners seek (in parallel) $\; p(\boldsymbol{\theta}^m|\mathbf{y}_t; \mathbf{X}_t)$

Q. Lu, G. V. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian Processes with Spectral Features for Online Interactive Learning with Scalability," *Proc. of AISTATS*, Palermo, Italy, June 2020.

Incremental EGP

Prediction

- Expert m forms RF-based prediction $p(y_{t+1}|\mathbf{y}_t, i=m; \mathbf{X}_{t+1})$
- Ensemble prediction $p(y_{t+1}|\mathbf{y}_t; \mathbf{X}_{t+1}) = \sum_{m=1}^{M} w_t^m p(y_{t+1}|\mathbf{y}_t, i=m; \mathbf{X}_{t+1})$

Correction

- Expert m updates $p(\boldsymbol{\theta}^m|\mathbf{y}_{t+1}, i=m; \mathbf{X}_{t+1}) \propto p(y_{t+1}|\boldsymbol{\theta}^m; \mathbf{x}_{t+1})\, p(\boldsymbol{\theta}^m|\mathbf{y}_t; \mathbf{X}_t)$
- EGP meta-learner updates weight $w_{t+1}^m = \dfrac{w_t^m\, p(y_{t+1}|\mathbf{y}_t, i=m; \mathbf{X}_{t+1})}{p(y_{t+1}|\mathbf{y}_t; \mathbf{X}_{t+1})}$

➢ Gaussian likelihood $\longrightarrow$ low complexity $\mathcal{O}(M(2D)^2)$ updates

Regret analysis for IE-GP

Goal: Bound performance of IE-GP relative to batch benchmark $\hat{f}^*$

- No assumptions on data generation → valid in adversarial settings

$$\mathcal{R}(T) := \sum_{t=1}^{T} \underbrace{\boxed{-\log p(y_t|\mathbf{y}_{t-1};\mathbf{X}_t)}}_{\text{IE-GP prediction loss}} - \sum_{t=1}^{T} \underbrace{\boxed{-\log p(y_t|\hat{f}^*(\mathbf{x}_t))}}_{\text{instantaneous benchmark loss } \mathcal{L}(\hat{f}^*(\mathbf{x}_t);y_t)}$$

(as1) $\mathcal{L}(z;y)$ is convex and continuously twice differentiable wrt z

(as2) $\mathcal{L}(z;y)$ has bounded first two derivatives wrt z

(as3) Kernels $\{\kappa^m\}_{m=1}^{M}$ are shift-invariant and bounded

Theorem. Under (as1)-(as3), IE-GP attains $\mathcal{R}(T) = \mathcal{O}(\log T)$ w.h.p.

Switching EGP for global dynamic models

Q. How about global and local dynamics? **A.** Time-varying learner index i_t and $\boldsymbol{\theta}_t^m$

- Markov chain dynamics at meta-learner: $q_{m,m'} := \Pr(i_{t+1}=m|i_t=m')$

- Weight prediction at meta-learner

$$w_{t+1|t}^m = \sum_{m'=1}^M \Pr(i_{t+1}=m|i_t=m')\Pr(i_t=m'|\mathbf{y}_t;\mathbf{X}_t) = \sum_{m'=1}^M q_{m,m'} w_{t|t}^{m'}$$

  - Used to form ensemble prediction

- Online loss for switching (S) IE-GP

$$\ell_{t+1|t}^{\mathrm{SW}} := -\log p(y_{t+1}|\mathbf{y}_t;\mathbf{X}_{t+1}) = -\log \sum_{m=1}^M w_{t+1|t}^m \exp\left(-l_{t+1|t}^m\right)$$

$$l_{t+1|t}^m := -\log p(y_{t+1}|\mathbf{y}_t, i_{t+1}=m;\mathbf{X}_{t+1})$$

Q. Lu, G. V. Karanikolas, and G. B. Giannakis, ”Incremental Ensemble Gaussian Processes,” submitted to *IEEE-TPAMI*, June 2021.

Regret analysis for global SIE-GP learning

Switching regret: accounting for model shift in the benchmark

$$\mathcal{R}^{\mathrm{SW}}(T) := \sum_{\tau=1}^{T} \ell_{\tau|\tau-1}^{\mathrm{SW}} - \min_{i_1,\ldots,i_T} \sum_{\tau=1}^{T} \mathcal{L}(\hat{f}^{i_\tau}(\mathbf{x}_\tau); y_\tau)$$

(as4) $q_{mm} = q_0, q_{mm'} = \frac{q_1}{M-1}$ for $m, m' \in \mathcal{M}, q_0 + q_1 = 1,$ and $0 \leq q_1 < \frac{1}{2} < q_0 \leq 1$

(as5) Number of model switches $\sum_{\tau=1}^{T} I(i_\tau \neq i_{\tau+1}) \leq S, S \ll T$

Theorem. Under (as1)-(as5), SIE-GP attains $\mathcal{R}^{\mathrm{SW}}(T) = \mathcal{O}(\log T)$ w.h.p.

Q. Lu, G. V. Karanikolas, and G. B. Giannakis, ”Incremental Ensemble Gaussian Processes,” submitted to *T-PAMI*, June 2021.

Local dynamic (D) IE-GP models

Q. How each individual GP learners account for dynamics?

A. Time-varying $\boldsymbol{\theta}_t^m$ with state-space (e.g., random walk) evolution

$$\begin{aligned}\boldsymbol{\theta}_{t+1}^m &= \boldsymbol{\theta}_t^m + \boldsymbol{\epsilon}_{t+1}^m \\ y_{t+1} &= \boldsymbol{\phi}_{\mathbf{v}}^{m^\top}(\mathbf{x}_{t+1})\, \boldsymbol{\theta}_{t+1}^m \;+\; n_{t+1}\end{aligned}$$

- Predictive pdf accounts for state transition

$$p(\boldsymbol{\theta}_{t+1}^m|\mathbf{y}_t;\mathbf{X}_{t+1}) = \int p(\boldsymbol{\theta}_{t+1}^m|\boldsymbol{\theta}_t^m)\; p(\boldsymbol{\theta}_t^m|\mathbf{y}_t;\mathbf{X}_t)\; d\boldsymbol{\theta}_t^m$$

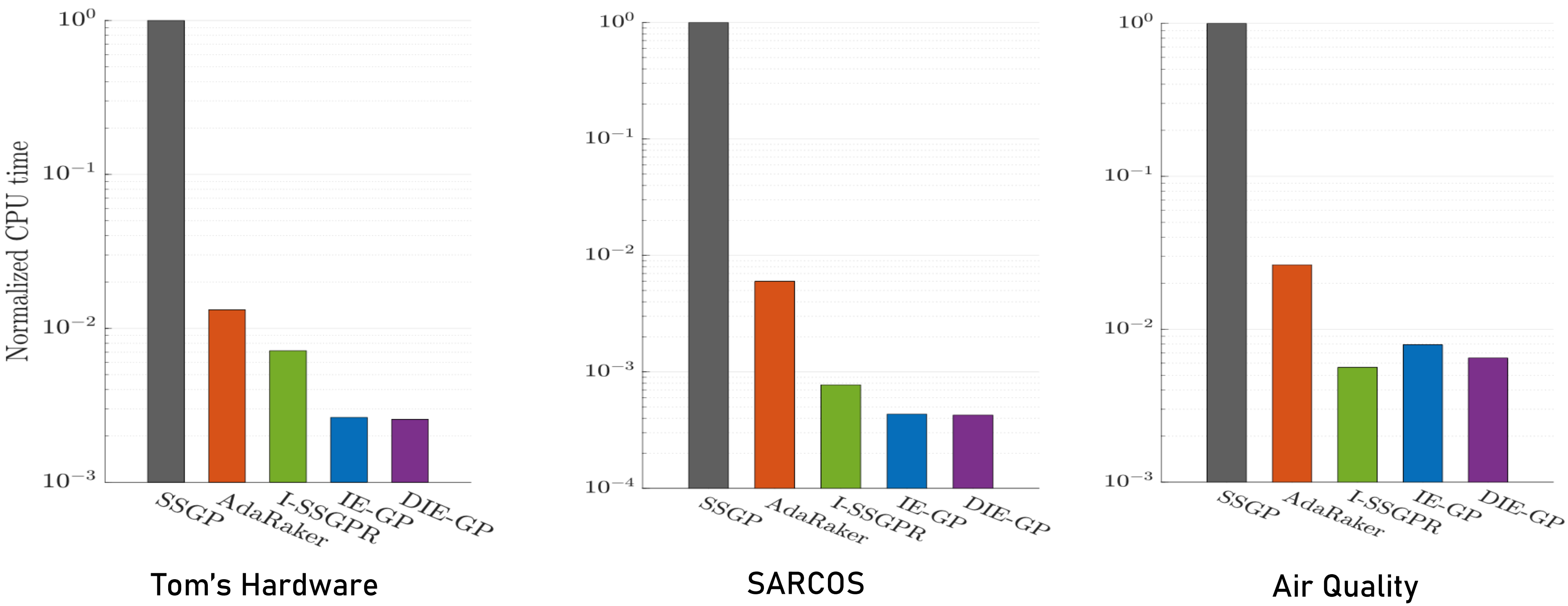
- Kalman filter (KF) updates exact for Gaussian likelihood

Outlook: DI-EGP for extended KF, unscented KF, and particle filtering

Q. Lu, G. V. Karanikolas, Y. Shen, and G. B. Giannakis, "Ensemble Gaussian Processes with Spectral Features for Online Interactive Learning with Scalability," *Proc. of AISTATS*, Palermo, Italy, June 2020.

Testing EGP-based regression

- Benchmarks: SSGP [Bui et al.'17], I-SSGPR [Gijsberts et al.'13], AdaRaker [Shen et al.'19]
- Normalized mean-square error $\mathrm{nMSE}_t := t^{-1}\sum^{t}(y_{t'} - \hat{y}_{t'|t'-1})^2/s_y^2$

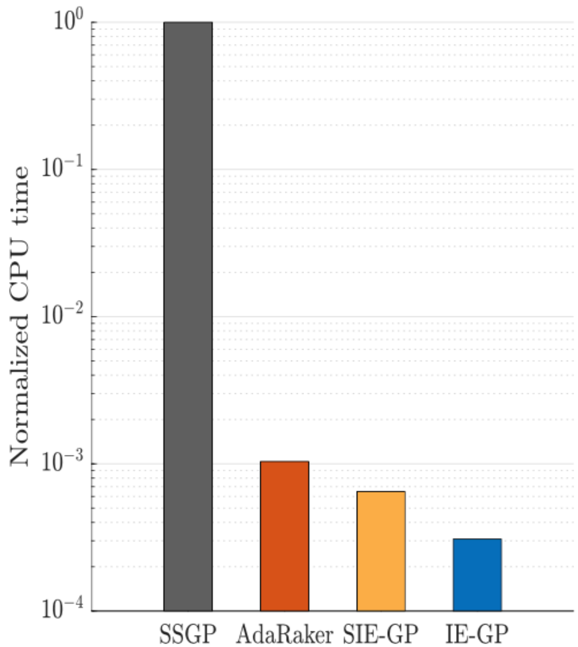


- (D)IE-GP achieve state-of-the-art nMSE and running time

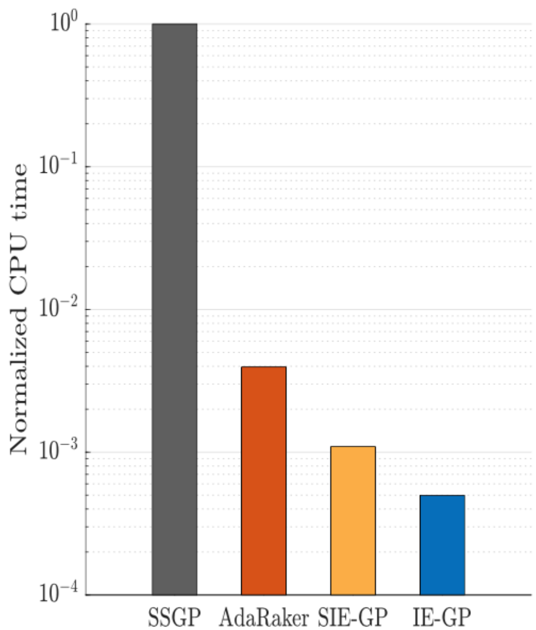
Bui et al., "Streaming sparse Gaussian process approximations," *NIPS*, 2017.
Gijsberts et al., "Real-time model learning using incremental sparse spectrum GPR," *Neural Networks*, 2013.
Shen et al., "RF-based online MKL in environments with unknown dynamics," *JMLR*, 2019.

Testing EGP-based classification

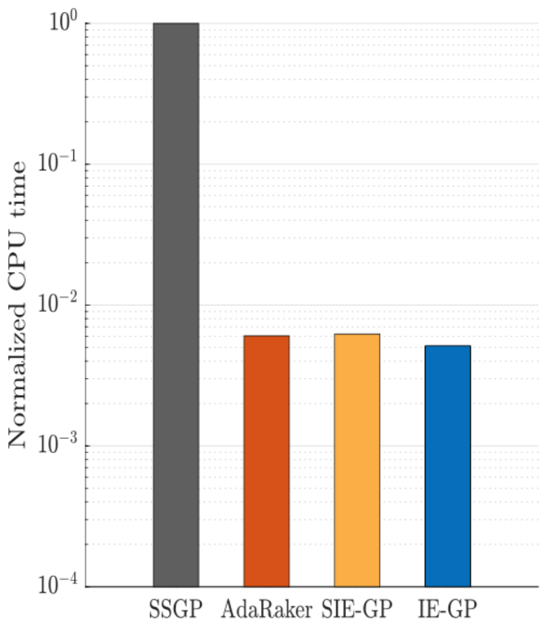
- Benchmarks: SSGP [Bui et al.'17], AdaRaker [Shen et al.'19]



Banana



Musk



Ionosphere

- (S)IE-GP outperforms alternatives in classification error and running time

Dimensionality reduction with RFs and GPs

Goal: Obtain low-dimensional representation $\mathbf{x}_t$ for observation $\mathbf{y}_t$

GPLVM postulates a nonlinear map *f* per dimension with GP prior [Lawrence '05]

$$[\mathbf{y}_t]_d = f_d(\mathbf{x}_t) + n_{td} \qquad \begin{array}{c} f_d \sim \mathcal{GP}(0, \kappa) \\ \{n_{td}\} \sim \mathcal{N}(0, \sigma_n^2) \end{array}$$

- Random feature (RF) approximation for kernel κ [Rahimi et al.'08]
  - For (normalized) `stationary' kernel $\bar{\kappa}(\mathbf{x}, \mathbf{x}') = \bar{\kappa}(\mathbf{x} - \mathbf{x}')$

  draw $\mathbf{v}_i \sim \pi_\kappa(\mathbf{v}) = \mathcal{F}(\bar{\kappa})$, and form $\phi_{\mathbf{v}}(\mathbf{x}) = \frac{1}{\sqrt{D}}[\cos(\mathbf{v}_1^\top \mathbf{x}) \ \sin(\mathbf{v}_1^\top \mathbf{x}) \ldots \cos(\mathbf{v}_D^\top \mathbf{x}) \ \sin(\mathbf{v}_D^\top \mathbf{x})]^\top$

  to obtain kernel approximant: $\check{\bar{\kappa}}(\mathbf{x}, \mathbf{x}') = \phi_{\mathbf{v}}^\top(\mathbf{x}) \phi_{\mathbf{v}}(\mathbf{x}')$

- RFs turn nonparametric f_d to a linear parametric approximant

$$\check{f}_d(\mathbf{x}) = \boldsymbol{\theta}_d^\top \phi_{\mathbf{v}}(\mathbf{x}) \qquad \boldsymbol{\theta}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *JMLR*, 2005

A. Rahimi and B. Recht, "Random features for large scale kernel machines," *NIPS*, 2008

RF-based GPLVM

- Conditional likelihood

$$p(\mathbf{Y}|\mathbf{X},\mathbf{\Theta}) = \prod_{t=1}^{T}\prod_{d=1}^{D_y}\mathcal{N}([\mathbf{y}_t]_d; \boldsymbol{\theta}_d^\top \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_t), \sigma_n^2)$$

$$\mathbf{X} := [\mathbf{x}_1 \ldots \mathbf{x}_T]^\top$$
$$\mathbf{Y} := [\mathbf{y}_1 \ldots \mathbf{y}_T]^\top \equiv [\mathbf{y}_{:1} \cdots \mathbf{y}_{:D_y}]$$
$$\mathbf{\Theta} := [\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_{D_y}]_{2D\times D_y}$$

- Marginalization over $\mathbf{\Theta}$

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^{D_y}\mathcal{N}(\mathbf{y}_{:d}; \mathbf{0}, \mathbf{\Phi}\mathbf{\Phi}^\top + \sigma_n^2\mathbf{I})$$

$$\mathbf{\Phi} := [\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_1)\ldots\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}_T)]^\top \in \mathbb{R}^{T\times 2D}$$

  - RF approximation allows for $\mathcal{O}(TD^2)$ evaluations of likelihood and gradients

- MAP estimates

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} -\log p(\mathbf{Y}|\mathbf{X}) - \log p(\mathbf{X})$$

$$p(\mathbf{X}) = \prod_{t=1}^{T}\mathcal{N}(\mathbf{x}_t; \mathbf{0}, \sigma_x^2\mathbf{I})$$

  - Nonconvex solver using e.g., conjugate gradient method [Møller ‘93]

G. V. Karanikolas, Q. Lu, G. B. Giannakis, “Online Unsupervised Learning using Ensemble Gaussian Processes with Random Features,” *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2021.

Online RF-based GPLVM

Goal. Seek latent representation $\mathbf{x}_t$ of new observation $\mathbf{y}_t$ given past $\{\mathbf{Y}_{t-1}, \hat{\mathbf{X}}_{t-1}\}$

- Conditional likelihood: $p(\mathbf{y}_t|\mathbf{Y}_{t-1}, \hat{\mathbf{X}}_{t-1}, \mathbf{x}_t) = \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_t, \sigma_t^2\mathbf{I})$

$$\boldsymbol{\mu}_t = \phi_{\mathbf{v}}^\top(\mathbf{x}_t)\hat{\boldsymbol{\theta}}_{t-1,d} = \phi_{\mathbf{v}}^\top(\mathbf{x}_t)\mathbf{A}_{t-1}^{-1}\mathbf{B}_{t-1}$$
$$\sigma_t^2 = \sigma_n^2[1 + \phi_{\mathbf{v}}^\top(\mathbf{x}_t)\mathbf{A}_{t-1}^{-1}\phi_{\mathbf{v}}(\mathbf{x}_t)]$$

$$\mathbf{A}_{t-1} := \boldsymbol{\Phi}_{t-1}^\top\boldsymbol{\Phi}_{t-1} + \sigma_n^2\mathbf{I}$$
$$\mathbf{B}_{t-1} := \boldsymbol{\Phi}_{t-1}^\top\mathbf{Y}_{t-1}$$

- MAP estimate of $\mathbf{x}_t$

$$\begin{aligned}\hat{\mathbf{x}}_t &= \arg\max_{\mathbf{x}_t}\ p(\mathbf{y}_t|\mathbf{Y}_{t-1}, \hat{\mathbf{X}}_{t-1}, \mathbf{x}_t)\, p(\mathbf{x}_t) \\ &= \arg\min_{\mathbf{x}_t}\ \frac{1}{2\sigma_t^2}\|\mathbf{y}_t - \boldsymbol{\mu}_t\|^2 + D_y \log\sigma_t + \frac{1}{2\sigma_x^2}\|\mathbf{x}_t\|^2\end{aligned}$$

- Recursive updates

$$\mathbf{B}_t = \mathbf{B}_{t-1} + \phi_{\mathbf{v}}(\hat{\mathbf{x}}_t)\mathbf{y}_t^\top$$
$$\mathbf{A}_t = \mathbf{A}_{t-1} + \phi_{\mathbf{v}}(\hat{\mathbf{x}}_t)\phi_{\mathbf{v}}^\top(\hat{\mathbf{x}}_t)$$

  - In practice, updates performed on the Cholesky factor of $\mathbf{A}_t$

Ensemble online RF-based GPLVM

Challenge: Online choice of kernel?

Remedy: Ensemble of M experts, each with a different kernel κ^m

Algorithm for incoming $\mathbf{y}_t$

- Per expert embeddings $\{\hat{\mathbf{x}}_t^m\}_{m=1}^M$ computed in parallel

$$\hat{\mathbf{x}}_t^m := \arg\max_{\mathbf{x}} \; p(\mathbf{y}_t|\mathbf{Y}_{t-1}, i=m, \hat{\mathbf{X}}_{t-1}^m, \mathbf{x})\, p(\mathbf{x}) \qquad m = 1, \ldots, M$$

- Output "best" embedding across experts $\hat{\mathbf{x}}_t := \hat{\mathbf{x}}_t^{m^*}$ (MAP estimate)

$$m^* := \arg\max_{m \in \{1 \ldots M\}} \; p(\mathbf{y}_t|\mathbf{Y}_{t-1}, i=m, \hat{\mathbf{X}}_{t-1}^m, \hat{\mathbf{x}}_t^m)\, \Pr(i=m|\mathbf{Y}_{t-1}, \{\hat{\mathbf{X}}_{t-1}^{(\mu)}\}_{\mu=1}^M)\, p(\hat{\mathbf{x}}_t^m)$$

- Meta-learner updates expert weights

$$w_t^m := \Pr(i=m|\mathbf{Y}_t, \{\mathbf{X}_t^\mu\}_{\mu=1}^M) \propto w_{t-1}^m\, p(\mathbf{y}_t|\mathbf{Y}_{t-1}, i=m, \mathbf{X}_{t-1}^m, \mathbf{x}_t^m) \quad m = 1, \ldots, M$$

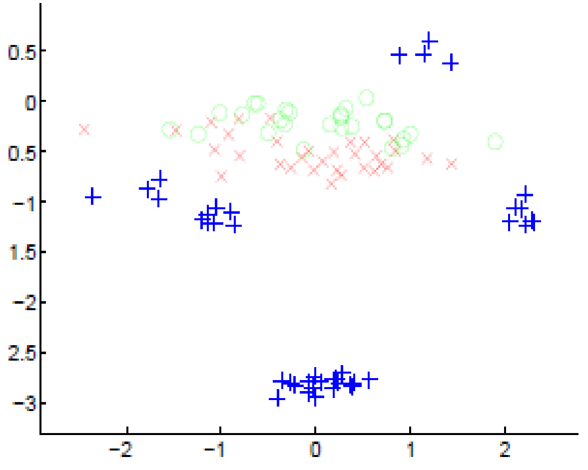
G. V. Karanikolas, Q. Lu, G. B. Giannakis, "Online Unsupervised Learning using Ensemble Gaussian Processes with Random Features," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2021.

GP-based test for dimensionality reduction

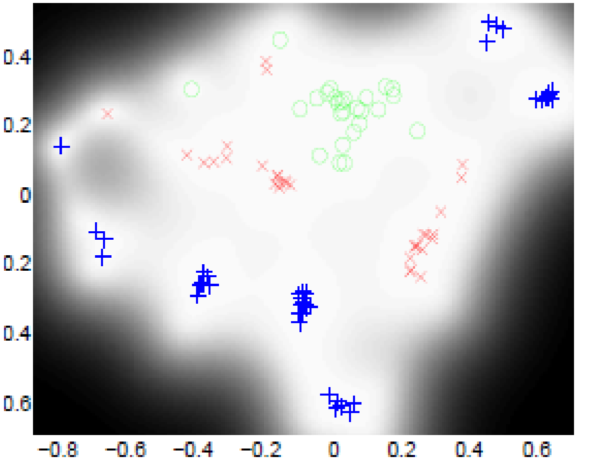
- Broadens probabilistic PCA using a GP latent variable model (LVM)
  - An independent GPR per dimension *d* $$[\mathbf{y}_t]_d = f_d(\mathbf{x}_t) + \varepsilon_t$$

Goal: Given D_y *x 1* vectors $\{\mathbf{y}_t\}_{t=1}^T$, find latent D_x *x* 1 vectors $\{\mathbf{x}_t\}_{t=1}^T$

PCA (D_x=2)



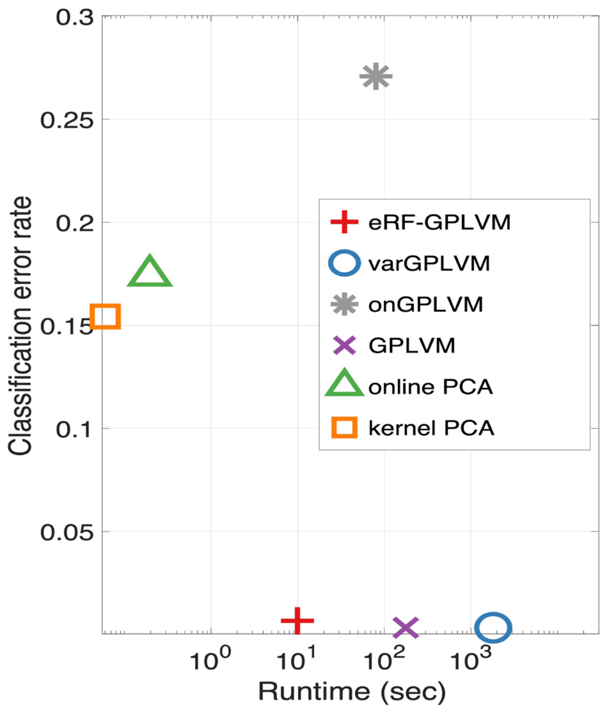
GPLVM (D_x=2)



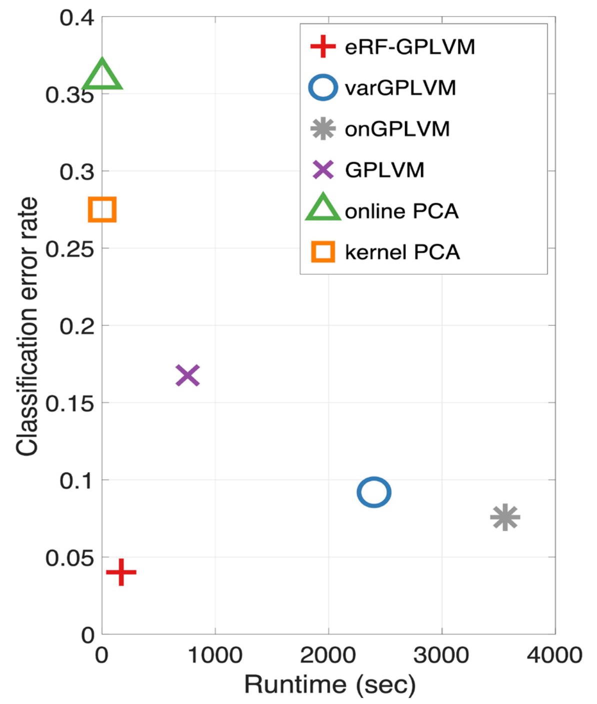
- GPLVM with linear kernel boils down to PCA with quantified uncertainty

N. D. Lawrence, “Probabilistic nonlinear principal component analysis with Gaussian process latent variable models,” *Journal of Machine Learning Research*, pp. 1783–1816, 2005.

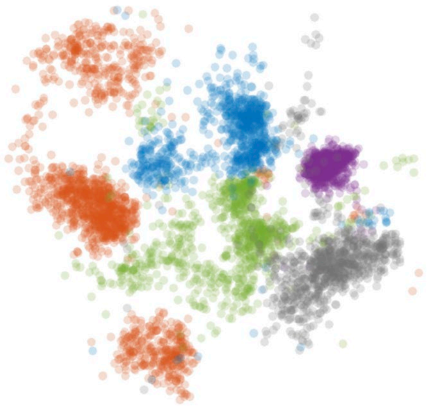
Testing (E)RF-GPLVM



Oil



USPS 0-4



Alternatives: **variational** [Damianou et al. '16], **online** [Yao et al. '11], **GPLVM** [Lawrence '05]

Figure of merit: **error rate** of nearest neighbor classification rule vs **runtime**

- ERF-GPLVM outperforms alternatives on benchmark datasets

A. C. Damianou, M. K. Titsias, and N. D. Lawrence, "Variational inference for latent variables and uncertain inputs in Gaussian processes," *JMLR*, 2016

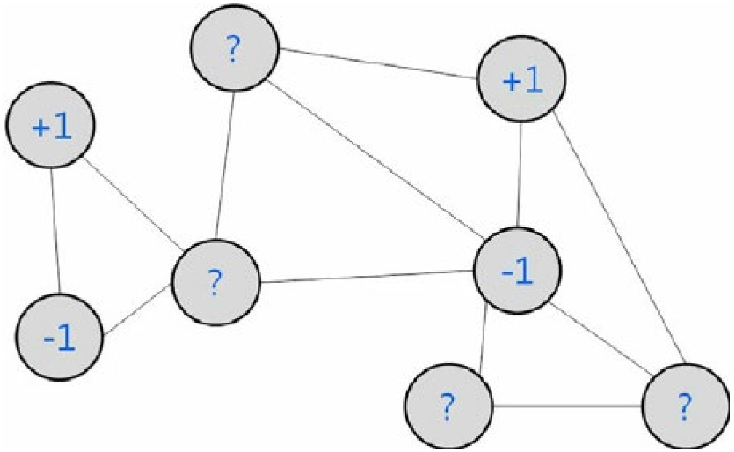
A. Yao, J. Gall, L. V. Gool, and R. Urtasun, "Learning probabilistic non-linear latent variable models for tracking complex activities," NIPS, 2011

N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," JMLR, 2005

Learning functions over graphs

- Graphs: model complex systems

- Graph-guided semi-supervised learning (SSL)



Graph-guided incremental SSL

- Graph $\mathcal{G} := \{\mathcal{V}, \mathbf{A}_N\}$ with vertex set $\mathcal{V}$ and *N x N* adjacency matrix $\mathbf{A}_N$

- Real-valued function on graph $f : \mathcal{V} \rightarrow \mathbb{R}$
  - f_n : feature value at node *n*
  - y_n : nodal value on observed set $\mathcal{O}$

$n \leftrightarrow t$

Goal: Given $\mathcal{G}$ and $\{y_n, n \in \mathcal{O}\}$, predict values $\{y_n, n \in \mathcal{U}\}$, $\mathcal{U} := \mathcal{V} \setminus \mathcal{O}$

- *Incremental* setting: use $\mathbf{y}_n := [y_1, \ldots, y_n]^\top$ to <u>predict</u> y_{n+1} and <u>correct</u> after y_{n+1} is observed

Incremental **Graph-ad**aptive EGP

Idea: Use one-hop connectivity vector $\mathbf{a}_n$ as input: $f_n = f(\mathbf{a}_n)$

- Learner *m* $\quad n \leftrightarrow t$

Prediction $\qquad$ Correction

$$\mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_n^m, \boldsymbol{\Sigma}_n^m) \xrightarrow{\mathbf{a}_{n+1}} \mathcal{N}(y_{n+1}; \hat{y}_{n+1|n}^m, (\sigma_{n+1|n}^m)^2) \xrightarrow{y_{n+1}} \mathcal{N}(\boldsymbol{\theta}^m; \hat{\boldsymbol{\theta}}_{n+1}^m, \boldsymbol{\Sigma}_{n+1}^m)$$

$$\hat{y}_{n+1|n}^m = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{a}_{n+1})\hat{\boldsymbol{\theta}}_n^m$$

$$(\sigma_{n+1|n}^m)^2 = \boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{a}_{n+1})\boldsymbol{\Sigma}_n^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{a}_{n+1}) + \sigma_\varepsilon^2$$

$$\hat{\boldsymbol{\theta}}_{n+1}^m = \hat{\boldsymbol{\theta}}_n^m + (\sigma_{n+1|n}^m)^{-2}\boldsymbol{\Sigma}_n^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{a}_{n+1})(y_{n+1} - \hat{y}_{n+1|n}^m)$$

$$\boldsymbol{\Sigma}_{n+1}^m = \boldsymbol{\Sigma}_n^m - (\sigma_{n+1|n}^m)^{-2}\boldsymbol{\Sigma}_n^m\boldsymbol{\phi}_{\mathbf{v}}^m(\mathbf{a}_{n+1})\boldsymbol{\phi}_{\mathbf{v}}^{m\top}(\mathbf{a}_{n+1})\boldsymbol{\Sigma}_n^m$$

- Meta-learner

$$\sum_{m=1}^M w_n^m \mathcal{N}(y_{n+1}; \hat{y}_{n+1|n}^m, (\sigma_{n+1|n}^m)^2)$$

$$\hat{y}_{n+1|n} = \sum_{m=1}^M w_n^m \hat{y}_{n+1|n}^m$$

$$\sigma_{n+1|n}^2 = \sum_{m=1}^M w_n^m [(\sigma_{n+1|n}^m)^2 + (\hat{y}_{n+1|n} - \hat{y}_{n+1|n}^m)^2]$$

  - Weight updates

$$w_{n+1}^m = \frac{w_n^m \mathcal{N}(y_{n+1}; \hat{y}_{n+1|n}^m, (\sigma_{n+1|n}^m)^2)}{\sum_{m'=1}^M w_n^{m'} \mathcal{N}(y_{n+1}; \hat{y}_{n+1|n}^{m'}, (\sigma_{n+1|n}^{m'})^2)}$$

- Complexity $\mathcal{O}(M((2D)^2 + 2DN))$

K. D. Polyzos, Q. Lu, G. B. Giannakis, "Graph-Adaptive Incremental Learning using an Ensemble of Gaussian Process Experts," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2021.

GradEGP vis-à-vis GCNs

- Comparison with graph convolutional networks (GCNs)

GradEGP

- Incremental $\rightarrow$ reduced storage
- Scalable online updates
- Bayesian $\rightarrow$ uncertainty quantification
- No need for additional nodal features
- Input: encrypted version connectivity pattern of nodes $\rightarrow$ privacy

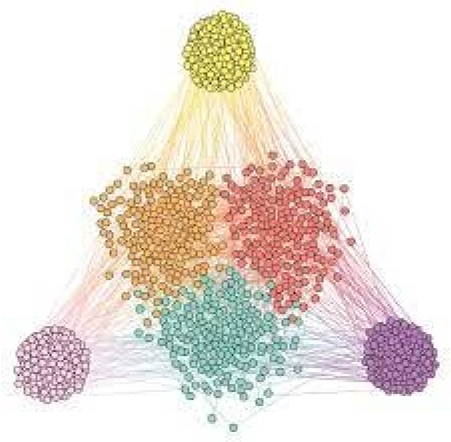
Conventional GCNs

- Batch approach $\rightarrow$ storage demand
- Demanding training phase
- Deterministic $\rightarrow$ only point estimates
- Additional nodal features needed
- Input: connectivity pattern of nodes

T. N. Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *ICLR*, 2016.

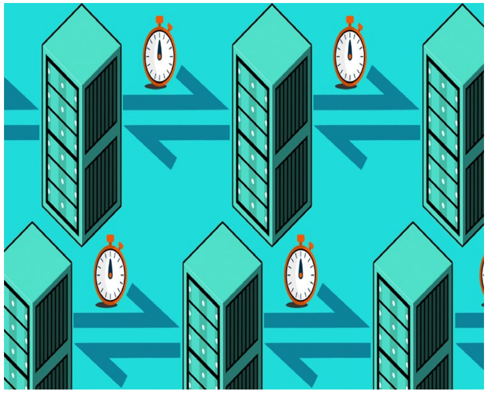
Testing GradEGP

Synthetic SBM *(N=60)*



Email Eu *(N=1,005)*

Network delay *(N=70)*



Benchmarks

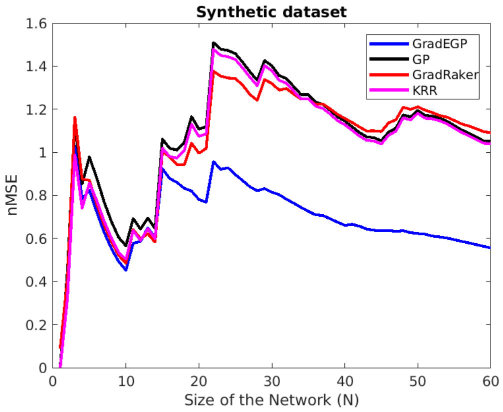
- GP [Rasmussen et al '06]
- Kernel ridge regression (KRR) [Romero et al '16]
- GradRaker [Shen et al '19]

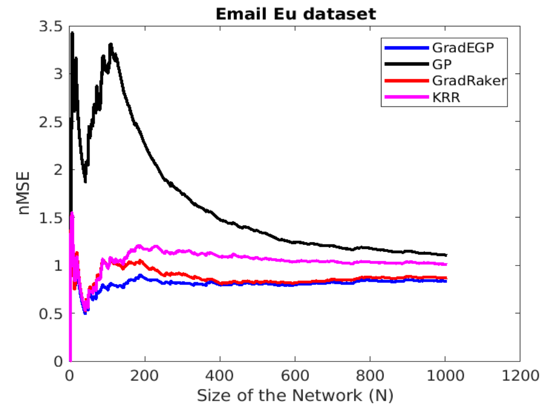
Figures of merit

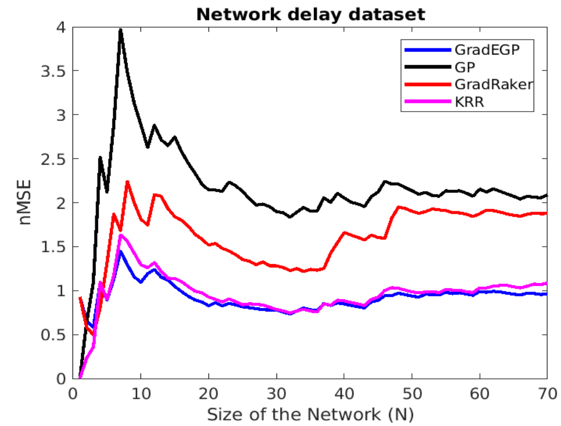
- Normalized mean-square error (NMSE) $\mathrm{nMSE}_n := n^{-1}\sum_{n'=1}^{n}(y_{n'} - \hat{y}_{n'|n'-1})^2/s_y^2$
- Runtime

Performance with uncertainty quantification

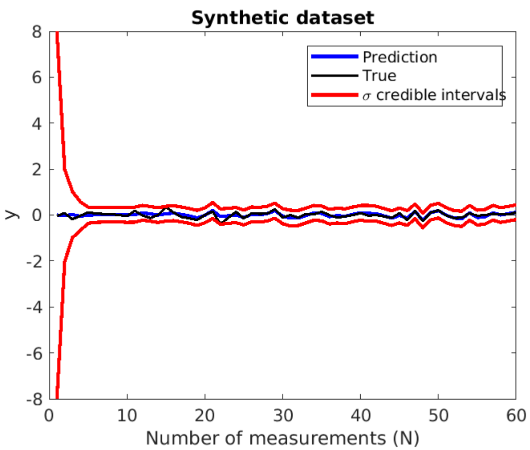
- NMSE versus *n*

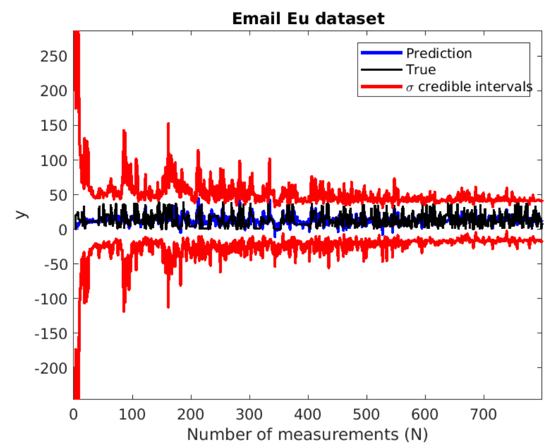


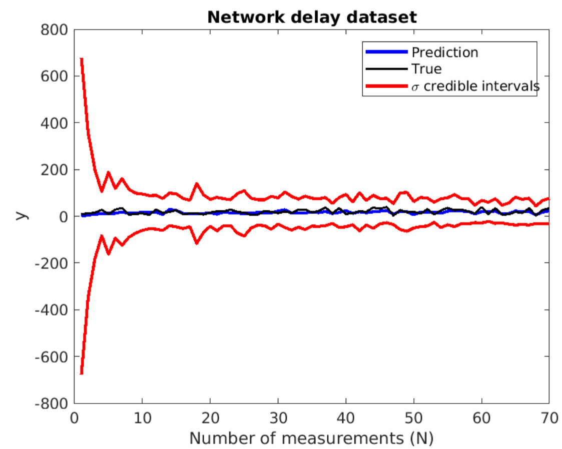




- GradEGP with uncertainty quantification



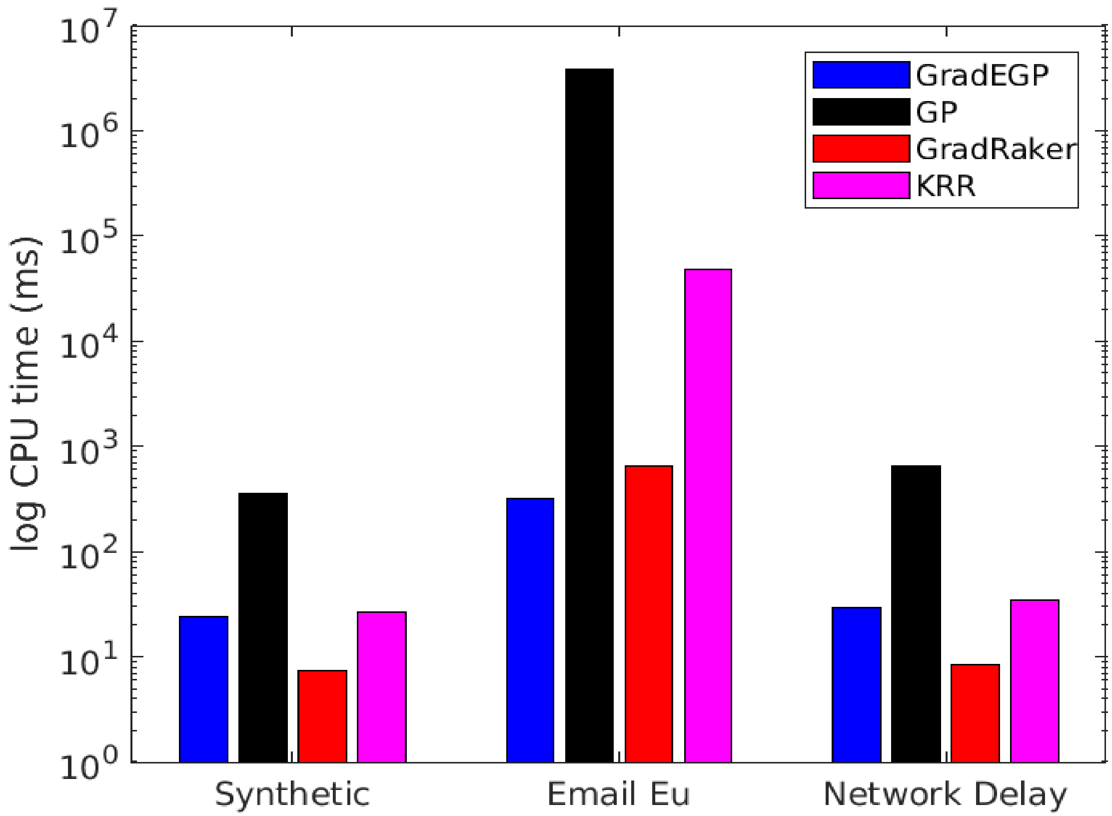




- GradEGP outperforms alternatives and estimates stay within confidence intervals

K. D. Polyzos, Q. Lu, G. B. Giannakis, “Graph-Adaptive Incremental Learning using an Ensemble of Gaussian Process Experts,” *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2021.

Runtime comparison



(N=60) *(N=1,005)* *(N=70)*

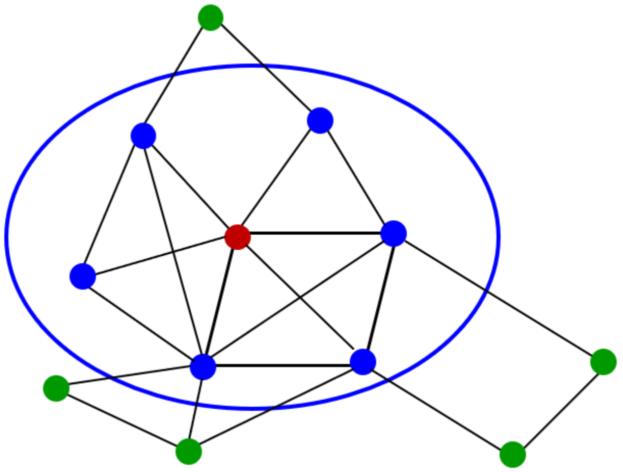
- GradEGP runtime less than scalable GradRaker in large-scale networks

K. D. Polyzos, Q. Lu, G. B. Giannakis, "Graph-Adaptive Incremental Learning using an Ensemble of Gaussian Process Experts," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2021.

Higher-order interactions

Q. More informative graph guidance than $\boldsymbol{a}_n$? **A.** How about per-node "egonet"?

- Egonet of node *n*
  - ✓ Node *n*
  - ✓ Direct neighbors of node *n*
  - ✓ All edges connecting direct neighbors
  - ➢ *N x N* adjacency matrix of node *n* egonet: $\mathbf{A}_n^{\text{ego}}$
    - ✓ Sparse matrix due to limited connectivity
  - ➢ "Egonet feature" vector $\mathbf{x}_n^{\text{ego}}$

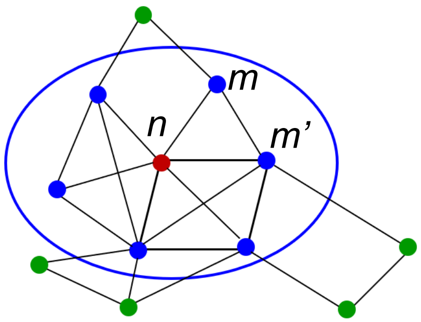


Model: Use egonet feature vector $\mathbf{x}_n^{\text{ego}}$ as input

$$f_n = f(\mathbf{x}_n^{\text{ego}}) \longrightarrow$$ "GradEGP-ego" $\mathbf{a}_n \leftrightarrow \mathbf{x}_n^{\text{ego}}$

K. D. Polyzos, Q. Lu, G. B. Giannakis, "Ensemble Gaussian Processes over Egonet Features for Online Graph-Guided Learning," *Proc. of ASILOMAR*, 2021.

Egonet feature vector per node *n*



- $\mathbf{x}_n^{\text{ego}}$ captures connectivity of node *n* to all nodes through its egonet
  - Degree of node *n* $d_n := \sum_{n'=1}^{N} \mathbf{A}_n^{\text{ego}}(n', n)$
  - Connectivity of any node *m* with node *n* as a sum of edge weights with its egonet

$$c_{\text{Ei}}^{n}(m) = \alpha \sum_{m' \in \mathcal{N}_m^n} c_{\text{Ei}}^{n}(m')$$

✓ Collectively, as eigenvector of max eigenvalue

$$\mathbf{c}_{\text{Ei}}^{n} := [c_{\text{Ei}}^{n}(1), \ldots, c_{\text{Ei}}^{n}(N)]^\top \qquad \mathbf{A}_n^{\text{ego}} \mathbf{c}_{\text{Ei}}^{n} = \alpha^{-1} \mathbf{c}_{\text{Ei}}^{n}$$

- Our $\mathbf{x}_n^{\text{ego}}$ comprises degree and eigenvector centralities (a.k.a. `vertex centrality')

$$\mathbf{x}_n^{\text{ego}} := \begin{bmatrix} d_n \\ \mathbf{c}_{\text{Ei}}^{n} \end{bmatrix}$$

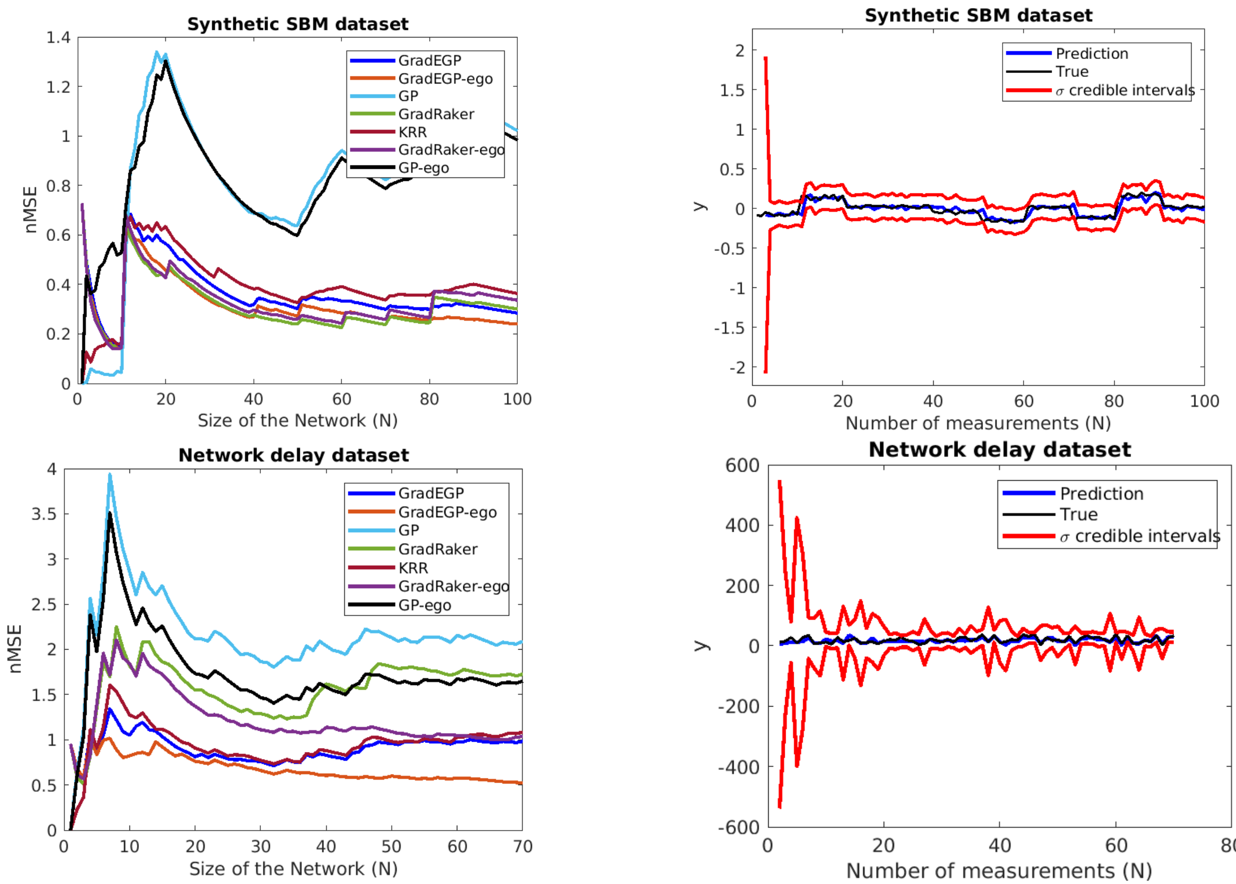
- $\mathbf{x}_n^{\text{ego}}$ can also include edge centrality, clustering coefficient, network cohesion [Kolaczyk'96]

E. D. Kolaczyk, *Statistical Analysis of Network Data,* Springer-Verlag New York,1996.

Testing GradEGP-ego

Benchmarks: GP [Rasmussen et al ’06], KRR [Romero et al ‘16], GradRaker [Shen et al ‘19]

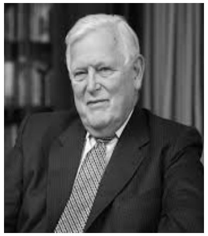
- Prediction performance with confidence intervals



- GradEGP-ego: state-of-the-art prediction performance

Summarizing remarks

- ✓ GPs as priors for nonparametric random function models with DNN links and uncertainty quantification



- ✓ RF offers linear parametric approximate models for online learning with scalability
- ✓ **Deep GP** for richer model expressiveness

- ❑ **Ensemble GPs** offer **wide** adaptability to operational environments
  - ✓ Online expert refinement with performance guarantees
  - ✓ Robustness to (un)modeled global and local dynamics
  - ✓ Supervised, unsupervised, and semi-supervised learning over graphs

- ❑ Interactive open-loop learning (Bayesian optimization) using GPs
- ❑ Interactive closed-loop reinforcement learning via (E) GPs

Research outlook

Q1. Desirable sweet spots by going **wide** *and* **deep**?

Q2. Particle filtering for **nonlinearities and dynamics**?

Q3. **Distributed/federated** IE-GP under computing/communication constraints?

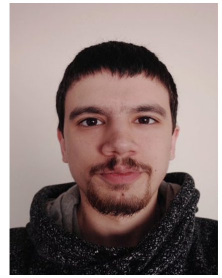
Q4. EGP-based surrogate model for BO with ensemble acquisitions?

Q5. EGP-based value/policy function estimation for **multi-agent** RL?

Q6. **Distributional robust** EGP learning?

Thank You! Stay safe!

Credit to the ensemble that credit is due ...

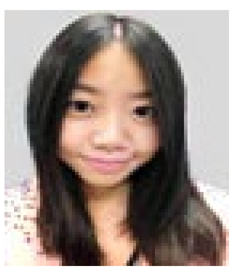


G.-V. Karanikolas
UofM

Dr. Qin Lu
UofM

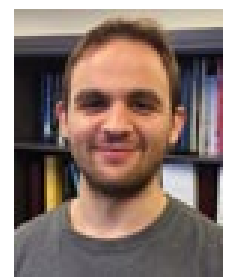
K. Polyzos
UofM

http://spincom.umn.edu



Prof. Y. Shen
UCI

Questions?



Dr. P. Traganitis
UofM