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
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

- Challenges and opportunities

- Classification
- Regression
- Reinforcement learning
- Dimensionality reduction

- 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 \( \{(x_t, y_t)\}_{t=1}^{T} \), find \( f(\cdot) \):

\[
x_t \rightarrow f(x_t) \rightarrow y_t
\]

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

**Ex1.** Regression: \( y_t = \theta^T x_t + e_t \)  
  Curve fitting for e.g. temperature forecasting

**Ex2.** Classification: \( y_t = \text{sign}(\theta^T x_t + b) \)  
  For e.g., disease diagnosis

---

- Global annual temperature
- Temperature difference (°C) relative to 1981-2010
- Normal brain
- Parkinson's brain

[P. Spetsieris et al PNAS 2015]
**Learning functions with kernels**

**Model:** view $f$ as deterministic from a Hilbert space $\mathcal{H} := \{ f | f(x) = \sum_{t=1}^{\infty} \alpha_t \kappa(x, x_t) \}$

Given data $\{(x_t, y_t)\}_{t=1}^{T}$, find

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{T} \sum_{t=1}^{T} C(f(x_t), y_t) + \lambda \Omega \left( \|f\|_{\mathcal{H}}^2 \right)$$

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

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

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

- Bayesian view is well motivated!
**Goal:** Learn posterior pdf of $f$ using Bayes’ rule

\[
p(f_t | y_t; X_t) \propto p(f_t; X_t)p(y_t | f_t; X_t)
\]

---

**Model:** View learning function $f$ as random with GP prior

\[(a0)\quad f \sim \mathcal{GP}(0, \kappa(x, x')) \iff f_t := [f(x_1), \ldots, f(x_t)]^\top \sim \mathcal{N}(f_t; 0_t, K_t)
\]

\[
[K_t]_{ij} = \text{cov}(f(x_i), f(x_j)) := \kappa(x_i, x_j) \quad \forall t
\]

\[
X_t := [x_1 \ldots x_t]^\top
\]

\[
y_t := [y_1 \ldots y_t]^\top
\]

---

\[(a1)\quad \text{Likelihood} \quad p(y_t | f_t; X_t) = \prod_{\tau=1}^{t} p(y_\tau | f(\tau))
\]

---

GP-based inference

Goal: Given training data \( \{X_t, y_t\} \) and test input \( x_* \), infer (pdf of) \( y_* \)

S1. Posterior pdf of function value at test input

\[
p(f_*|y_t; X_t, x_*) = \int p(f_*|f_t; X_t, x_*)p(f_t|y_t; X_t)df_t
\]

\[
\mathcal{N}(f_*; k_*^T K_t^{-1} f_t, \kappa_{**} - k_*^T K_t^{-1} k_*)
\]

S2. Posterior pdf of test output

\[
p(y_*|y_t; X_t, x_*) = \int p(y_*|f(x_*))p(f_*|y_t; X_t, x_*)df_*
\]

- Numerical or MC sampling for non-Gaussian likelihoods
**GP regression predictor**

- If likelihood also Gaussian, then
  \[ p(y_\ast | y_t, X_t, x_\ast) = \mathcal{N}(y_\ast; \hat{y}_\ast | t, \sigma^2_\ast | t) \]

  ➢ Mean and variance in closed form!

  \[
  \hat{y}_\ast | t = k_\ast^\top (K_t + \sigma^2_n I_t)^{-1} y_t \\
  \sigma^2_\ast | t = \kappa_{**} - k_\ast^\top (K_t + \sigma^2_n I_t)^{-1} k_\ast + \sigma^2_n
  \]

- Wiener filtering

  \[ h_t = \text{cov}^{-1}(y_t) \text{cov}(y_t, y_\ast) = (K_t + \sigma^2_n I_t)^{-1} k_\ast \]
GP-based classifier

**Challenge:** likelihood is non-Gaussian; e.g., logistic \( p(y_t|f(x_t)) = \frac{1}{1 + e^{-y_t f(x_t)}} \)

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

\[ p(f_t|y_t; X_t) \approx \mathcal{N}(f_t; \hat{f}_t, \Sigma_t) \]

\[
\hat{f}_t = \arg \max_{f_t} \ln p(y_t|f_t; X_t) + \ln p(f_t; X_t)
\]

\[
\Sigma_t^{-1} = K_t^{-1} - \nabla^2 \ln p(y_t|f_t; X_t)\big|_{f_t=\hat{f}_t}
\]

**S0.**

\[
p(f_*|y_t; X_t, X_*) = \int p(f_*|f_t; X_t, x_*) p(f_t|y_t; X_t) df_t \approx \mathcal{N}(f_*; \hat{f}_*|t, \sigma^2_{f_*|t})
\]

**S1.**

\[
p(y_*|y_t; X_t, x_*) \approx \int p(y_*|f(x_*)) p(f_*|y_t; X_t, x_*) df_*
\]

**S2.**

Numerical or MC sampling approximation


GP kernel adaptivity and scalability

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

\[
\hat{\alpha} = \arg \max_{\alpha} \quad p(y_t; X_t, \alpha) = \int p(y_t|f_t; X_t)p(f_t; X_t)df_t
\]

- For GP regression \[ p(y_t; X_t, \alpha) = \mathcal{N}(y_t; 0_t, K_t + \sigma_n^2 I_t) \]

  - $K_t$ selection decoupled from $f_t$ estimation; Gaussian approx. for classification

- Curse of dimensionality (CoD)

\[
\hat{y}_{*t} = k_{*t}^\top (K_t + \sigma_n^2 I_t)^{-1} y_t \\
\sigma_{*t}^2 = \kappa_{**} - k_{*t}^\top (K_t + \sigma_n^2 I_t)^{-1} k_{*t} + \sigma_n^2
\]

  - Complexity $O(t^3)$; storage $O(t^2)$
  - 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]

Random features via Fourier spectrum

**RF1.** Draw $D$ random vectors from the kernel’s Fourier transform
\[ \mathbf{v}_i \sim \pi(\mathbf{v}) = \mathcal{F}(\tilde{\kappa}), \quad i = 1, \ldots, D \]

**RF2.** Form $2D \times 1$ random feature (RF) vector
\[
\phi_{\mathbf{v}}(\mathbf{x}) := \frac{1}{\sqrt{D}} \left[ \sin(\mathbf{v}_1^T \mathbf{x}), \cos(\mathbf{v}_1^T \mathbf{x}), \ldots, \sin(\mathbf{v}_D^T \mathbf{x}), \cos(\mathbf{v}_D^T \mathbf{x}) \right]^T
\]

- RF-based linear kernel approximant $\tilde{\kappa}(\mathbf{x}, \mathbf{x}') = \phi_{\mathbf{v}}^T(\mathbf{x})\phi_{\mathbf{v}}(\mathbf{x}')$

**Key idea:** Random linear function
\[
\tilde{f}(\mathbf{x}) = \phi_{\mathbf{v}}^T(\mathbf{x})\theta, \quad \theta \sim \mathcal{N}(\mathbf{0}_{2D}, \sigma_{\theta}^2 \mathbf{I}_{2D})
\]

is a parametric GP with
\[
\text{cov}(\tilde{f}(\mathbf{x}_i), \tilde{f}(\mathbf{x}_j)) = \sigma_{\theta}^2 \phi_{\mathbf{v}}^T(\mathbf{x}_i)\phi_{\mathbf{v}}(\mathbf{x}_j)
\]

- Prior $p(\tilde{f}_t; \mathbf{X}_t) = \mathcal{N}(\tilde{f}_t; \mathbf{0}_t, \sigma_{\theta}^2 \Phi_t \Phi_t^T)$
\[
\Phi_t := \left[ \phi_{\mathbf{v}}(\mathbf{x}_1), \ldots, \phi_{\mathbf{v}}(\mathbf{x}_t) \right]^T
\]

2D-rank approx. of $K_t$

RF-driven parametric GPs

- Parametric generative model

  Vanilla GP: \( f \sim \mathcal{GP}(0, \kappa(x, x')) \)

  RF-based GP: \( \tilde{f}(x) = \phi_v^T(x)\theta \)
  \( \theta \sim \mathcal{N}(0_{2D}, \sigma_\theta^2 I_{2D}) \)

- Batch GPR predictor

  \[
  \hat{y}_*|t = \phi_v^T(x_*) \left( \Phi_t^T \Phi_t + \frac{\sigma_n^2}{\sigma_\theta^2} I_{2D} \right)^{-1} \Phi_t^T y_t \\
  \sigma_*^2|t = \phi_v^T(x_*) \left( \frac{\Phi_t^T \Phi_t}{\sigma_n^2} + \frac{I_{2D}}{\sigma_\theta^2} \right)^{-1} \phi_v(x_*) + \sigma_n^2
  \]

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

Incremental RF-GP learning

- Propagate posterior of $\theta$ as in recursive Bayes [Gijsberts-Metta’13]

$$p(\theta|y_t; X_t) \xrightarrow{\text{predictive pdf \ via \ } x_{t+1}} p(y_{t+1}|y_t; X_{t+1}) \xrightarrow{\text{corrective pdf \ via \ } y_{t+1}} p(\theta|y_{t+1}; X_{t+1})$$

$$p(y_{t+1}|y_t; X_{t+1}) = \int p(y_{t+1}|\theta; x_{t+1})p(\theta|y_t; X_t)d\theta$$

$$p(\theta|y_{t+1}; X_{t+1}) = \frac{p(\theta|y_t; X_t)p(y_{t+1}|\theta; x_{t+1})}{p(y_{t+1}|y_t; X_{t+1})}$$

- **GPR**

$$\mathcal{N}(\theta; \hat{\theta}_t, \Sigma_t) \xrightarrow{} \mathcal{N}(y_{t+1}; \hat{y}_{t+1|t}, \sigma^2_{t+1|t}) \xrightarrow{} \mathcal{N}(\theta; \hat{\theta}_{t+1}, \Sigma_{t+1})$$

$$\hat{y}_{t+1|t} = \phi_{t+1}^T \hat{\theta}_t$$

$$\sigma^2_{t+1|t} = \phi_{t+1}^T \Sigma_t \phi_{t+1} + \sigma_n^2$$

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \sigma^{-2}_{t+1|t} \Sigma_t \phi_{t+1}(y_{t+1} - \hat{y}_{t+1|t})$$

$$\Sigma_{t+1} = \Sigma_t - \sigma^{-2}_{t+1|t} \Sigma_t \phi_{t+1} \phi_{t+1}^T \Sigma_t$$

- **Complexity $O(t(2D)^2)$**

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
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

GP classification for remote sensing

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

GPs for dynamic state estimation

Goal: Given observations $y_t$, estimate $x_t$ (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

Deep neural networks

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

- First layer
  \[ f^1_{\nu}(x) = \sum_{\nu' = 1}^{D_x} w^{1}_{\nu,\nu'} x_{\nu'} + b^1_{\nu}, \quad \nu = 1, \ldots, N_1 \]

- Next layers
  \[ g^{\ell-1}_{\nu}(x) = \phi(f^{\ell-1}_{\nu}(x)), \quad \nu = 1, \ldots, N_{\ell-1} \]
  \[ f^{\ell}_{\nu}(x) = \sum_{\nu' = 1}^{N_{\ell-1}} w^{\ell}_{\nu,\nu'} g^{\ell-1}_{\nu'}(x) + b^\ell_{\nu}, \quad \nu = 1, \ldots, N_\ell \]
Bayesian neural networks (BNNs)

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

Proposition 1 [Neal'96] For $L=2$, if $\{g^1_{\nu}(x)\}_{\nu=1}^{N_1}$ have bounded variances, then as $N_1 \to \infty$ the output $\{f^2_{\nu}(x)\}_{\nu=1}^{N_2}$ (nonlinearity $\varphi$) converges in distr. to a 0-mean GP with
  $\mathbb{E}[f^2_{\nu}(x)f^2_{\nu'}(x')] = \delta_{\nu,\nu'}[\tilde{C}^2_w \mathbb{E}_{w,b} \{\varphi(w^T x + b)\varphi(w^T x' + b)\} + C^2_b]$
Sketch of the proof …

- For $L=2$
  
  \[
  f^2_\nu(x) = \sum_{\nu' = 1}^{D_x} w^2_{\nu,\nu'} x_{\nu'} + b^2_\nu \quad \text{and} \quad g^1_\nu(x) = \varphi(f^1_\nu(x))
  \]

- Gaussian BNN parameters
  \[
  b \sim \mathcal{N}(b; 0, C^1_b), \quad w \sim \mathcal{N}(w; 0, C^1_w I_{D_x})
  \]

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

\[
\mathbb{E}[f^2_\nu(x)] = 0
\]
\[
\mathbb{E}[f^2_\nu(x) f^2_{\nu'}(x')] = \delta_{\nu,\nu'} \left[ \tilde{C}_w^2 \mathbb{E}_{w,b} \left\{ \varphi(w^\top x + b) \varphi(w^\top x' + b) \right\} + C^2_b \right]
\]

- Likewise for $t$ training vectors
  \[
  [f^2(x_1), f^2(x_2), \ldots, f^2(x_t)]^\top
  \]

---

Normal limiting distribution across layers

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

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

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

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

\[
b \sim \mathcal{N}(b; 0, C_b^{\ell-2}), \ w \sim \mathcal{N}(w; 0, C_w^{\ell-2} I_{N_{\ell-2}})
\]

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

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

- Limiting GP has recursively computable kernels

Deep BNNs vis-a-vis GPs

Q. How about finite $N_\ell$?

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

**Theorem.** For a BNN with ReLU as $\varphi$ and any $\{x_\tau\}_{\tau=1}^t$ 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 $\kappa(x, x') = \tilde{C}_w^\ell \mathbb{E}\{\varphi(\epsilon^\ell(x))\varphi(\epsilon^\ell(x'))\} + C_b^\ell$

Deep BNNs versus GPs - Empirical comparison

- Compare \( p_{\text{BNN}}(y; x) \) and \( p_{\text{GP}}(y; x) \) using maximum mean discrepancy metric

\[
\mathcal{MMD}(p_{\text{BNN}}, p_{\text{GP}}, \mathcal{F}) = \sup_{g \in \mathcal{F}} [\mathbb{E}_{y \sim p_{\text{BNN}}}[g(y)] - \mathbb{E}_{y \sim p_{\text{GP}}}[g(y)]]
\]

- Sample estimator over \( \kappa \)-induced RHKS functions (in \( \mathcal{F} \)) [Gretton et al.'12]

\[
\tilde{MMD}^2(p_{\text{BNN}}, p_{\text{GP}}, \mathcal{F}) = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} \kappa(\tilde{y}_i, \tilde{y}_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \kappa(\tilde{y}_i', \tilde{y}_j') - 2 \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \kappa(\tilde{y}_i, \tilde{y}_j')
\]

- Draw \( \tilde{y}_i \sim p_{\text{BNN}} \) and \( \tilde{y}_j \sim p_{\text{GP}} \)

- Sample \( \tilde{MMD}^2 \) versus number of neurons per layer

- Faster convergence for wider and shallower BNNs

Going deep…

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

$$h_{i,\tau}^\ell = f_i^\ell(h_{\tau}^{\ell-1})$$

$$f_i^\ell \sim \mathcal{GP}(0, \kappa_i^\ell)$$

$\ell = 1, \ldots, L$

$i = 1, \ldots, N_\ell$

$\tau = 1, \ldots, t$

**DGP prior (non-Gaussian)**

$$p(H_t^L; X_t) = \int \prod_{\ell=1}^L p(H_t^\ell|H_t^{\ell-1}) \ dH_t^{L-1} \cdots dH_t^1$$

**Likelihood**

$$p(y_t|H_t^L; X_t) = \prod_{\tau=1}^{N_\tau} p(y_{\tau}|h_{1,\tau}^L; x_{\tau})$$

- Intractable integration due to CoD

RF-based DGPs

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

- Parametric layer-to-layer mapping

- Per-datum likelihood $p(y_\tau | \Theta; x_\tau)$, $\Theta := \left\{ \Theta_i^\ell \right\}_{\ell=1}^L \in \mathbb{R}^{N_\ell \times D_\ell}$, $\Theta_i^\ell := [\theta_{1i}^\ell \ldots \theta_{N_i}^\ell]^T \sim \mathcal{N}(0_{2D_\ell}, I_{2D_\ell})$
**Training and testing with DGPs**

**Training**: find \( \{ \alpha^\ell \} \) and \( p(\Theta | y_t; X_t) \) using variational inference

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

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

- Solvable via stochastic optimization

\[
\tilde{\theta}_{di,r}^\ell = \mu_{di}^\ell + \sqrt{s_{di}^\ell} \tilde{\varepsilon}_{di,r}^\ell
\]

\[
\tilde{\varepsilon}_{di,r}^\ell \sim \mathcal{N}(0, 1)
\]

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

\[
p(y_* | y_t; X_t, x_*) \approx \int p(y_* | \Theta; x_*, \{\hat{\alpha}^\ell\}) q(\Theta) d\Theta \approx \frac{1}{R} \sum_{r=1}^{R} p(y_* | \tilde{\Theta}_r; x_*, \{\hat{\alpha}^\ell\})
\]
Testing DGP for regression

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

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

- $x_t$: hourly ambient measurements
- $y_t$: electric energy output

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

- $x_t$: protein structure attributes
- $y_t$: protein functionality

- RF-based DGPs lower RMSE and quantify uncertainty


Testing DGP for classification

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

- \(x_\tau\): frequency of words/characters per email
- \(y_\tau\): 1 (spam) or 0 (not spam)

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

- \(x_\tau\): measurements from 14 electrodes
- \(y_\tau\): 1 (alcoholic) or 0 (not alcoholic)

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

![Graphs showing error rate vs runtime and MNLL for different models](image-url)

![Bar charts showing accuracy and MNLL for MNIST and AIRLINE datasets](image-url)
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 ensembles

- 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 Ensembles of GPs

Ensemble GP learning

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

- GP prior per learner \( m \) \( f \mid i = m \sim \mathcal{GP}(0, \kappa^m(x, x')) \)
- Ensemble (E) GP prior \( f \sim \sum_{m=1}^{M} w^m \mathcal{GP}(0, \kappa^m(x, x')) \) \( \sum_{m=1}^{M} w^m = 1 \)
- RF-based EGP \( \tilde{f} \mid \{\theta^m\}_{m=1}^{M} \sim \sum_{m=1}^{M} w^m \delta(\tilde{f}(x) - \phi^m_\nu(x)\theta^m), \theta^m \sim \mathcal{N}(\theta^m; 0, \sigma^2_{\theta^m}I_{2D}) \)
  (non-Gaussian prior)

- EGPs can model a richer space of learning functions
  - Meta-learner weighs experts using \( w^m_t := \Pr(i = m \mid y_t; X_t) \)
  - Learners seek (in parallel) \( p(\theta^m \mid y_t; X_t) \)

Incremental EGP

Prediction

- Expert $m$ forms RF-based prediction

\[ p(y_{t+1}|y_t, i = m; X_{t+1}) \]

- Ensemble prediction

\[ p(y_{t+1}|y_t; X_{t+1}) = \sum_{m=1}^{M} w_t^m p(y_{t+1}|y_t, i = m; X_{t+1}) \]

Correction

- Expert $m$ updates

\[ p(\theta^m|y_{t+1}, i = m; X_{t+1}) \propto p(y_{t+1}|\theta^m; x_{t+1}) \ p(\theta^m|y_t; X_t) \]

- EGP meta-learner updates weight

\[ w_{t+1}^m = \frac{w_t^m \ p(y_{t+1}|y_t, i = m; X_{t+1})}{p(y_{t+1}|y_t; X_{t+1})} \]

- Gaussian likelihood $\rightarrow$ low complexity $O(M(2D)^2)$ updates
Regret analysis for IE-GP

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

- No assumptions on data generation $\Rightarrow$ valid in adversarial settings

\[
\mathcal{R}(T) := \sum_{t=1}^{T} -\log p(y_t \mid y_{t-1}; X_t) - \sum_{t=1}^{T} -\log p(y_t \mid \hat{f}^*(x_t))
\]

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

**Details:**

- **(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
Switching EGP for global dynamic models

Q. How about global and local dynamics? A. Time-varying learner index $i_t$ and $\theta_t^m$

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

- Weight prediction at meta-learner

$$w_{t+1|t}^m = \sum_{m'=1}^M \Pr(i_{t+1} = m | i_t = m') \Pr(i_t = m'|y_t; X_t) = \sum_{m'=1}^M q_{m,m'} w_{t|t}^m$$

- Used to form ensemble prediction

- Online loss for switching (S) IE-GP

$$\ell_{t+1|t}^{SW} := -\log p(y_{t+1} | y_t; X_{t+1}) = -\log \sum_{m=1}^M w_{t+1|t}^m \exp \left( -l_{t+1|t}^m \right)$$

$$l_{t+1|t}^m := -\log p(y_{t+1} | y_t, i_{t+1} = m; X_{t+1})$$

Regret analysis for global SIE-GP learning

Switching regret: accounting for model shift in the benchmark

\[ R^{SW}(T) := \sum_{\tau=1}^{T} \ell^{SW}_{\tau|\tau-1} - \min_{i_1, \ldots, i_T} \sum_{\tau=1}^{T} \mathcal{L}(\hat{f}^{i_\tau}(x_\tau); y_\tau) \]

(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 \( R^{SW}(T) = \mathcal{O}(\log T) \) w.h.p.
Local dynamic (D) IE-GP models

Q. How each individual GP learners account for dynamics?

A. Time-varying $\theta_t^m$ with state-space (e.g., random walk) evolution

\[
\begin{align*}
\theta_{t+1}^m &= \theta_t^m + \epsilon_{t+1}^m \\
y_{t+1} &= \phi_{\nu}^m(x_{t+1}) \theta_{t+1}^m + n_{t+1}
\end{align*}
\]

- Predictive pdf accounts for state transition

\[
p(\theta_{t+1}^m | y_t; X_{t+1}) = \int p(\theta_{t+1}^m | \theta_t^m) \ p(\theta_t^m | y_t; X_t) \ d\theta_t^m
\]

- Kalman filter (KF) updates exact for Gaussian likelihood

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

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**

\[
\text{nMSE}_t := t^{-1} \sum_{t} (y_{t'} - \hat{y}_{t'|t'-1})^2 / s_y^2
\]

![Graph](image)

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

---

Testing EGP-based classification

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

- (S)IE-GP outperforms alternatives in classification error and running time
**Goal:** Obtain low-dimensional representation $x_t$ for observation $y_t$

GPLVM postulates a nonlinear map $f$ per dimension with GP prior [Lawrence ‘05]

$$[y_t]_d = f_d(x_t) + n_{td} \quad f_d \sim GP(0, \kappa)$$

$$\{n_{td}\} \sim \mathcal{N}(0, \sigma_n^2)$$

- Random feature (RF) approximation for kernel $\kappa$ [Rahimi et al.’08]
  - For (normalized) `stationary’ kernel $\bar{k}(x, x') = k(x - x')$
    - draw $v_i \sim \pi_{\kappa}(v) = \mathcal{F}(\bar{k})$, and form $\phi_v(x) = \frac{1}{\sqrt{D}}[\cos(v_1^\top x) \sin(v_1^\top x) \ldots \cos(v_D^\top x) \sin(v_D^\top x)]^\top$
    - to obtain kernel approximant: $\tilde{k}(x, x') = \phi_v^\top(x)\phi_v(x')$

- RFs turn nonparametric $f_d$ to a linear parametric approximant

$$\tilde{f}_d(x) = \theta_d^\top \phi_v(x) \quad \theta_d \sim \mathcal{N}(0, I)$$

N. Lawrence, “Probabilistic non-linear principal component analysis with Gaussian process latent variable models,” *JMLR*, 2005
RF-based GPLVM

- Conditional likelihood

\[
p(Y|X, \Theta) = \prod_{t=1}^{T} \prod_{d=1}^{D_y} \mathcal{N}(\hat{y}_{td}; \theta_d^T \phi_v(x_t), \sigma^2_n)
\]

\(X := [x_1 \ldots x_T]^T\)

\(Y := [y_1 \ldots y_T]^T \equiv [y:1 \ldots y:D_y]\)

\(\Theta := [\theta_1 \ldots \theta_{D_y}]_{2D \times D_y}\)

- Marginalization over \(\Theta\)

\[
p(Y|X) = \prod_{d=1}^{D_y} \mathcal{N}(y:d; 0, \Phi \Phi^T + \sigma^2_n I)
\]

\(\Phi := [\phi_v(x_1) \ldots \phi_v(x_T)]^T \in \mathbb{R}^{T \times 2D}\)

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

- MAP estimates

\[
\hat{X} = \arg \min_X - \log p(Y|X) - \log p(X)
\]

\[
p(X) = \prod_{t=1}^{T} \mathcal{N}(x_t; 0, \sigma^2_x I)
\]

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

Online RF-based GPLVM

**Goal.** Seek latent representation $x_t$ of new observation $y_t$ given past $\{Y_{t-1}, \hat{X}_{t-1}\}$

- **Conditional likelihood:**
  \[ p(y_t|Y_{t-1}, \hat{X}_{t-1}, x_t) = \mathcal{N}(y_t; \mu_t, \sigma_t^2 I) \]
  \[
  \mu_t = \phi_v^\top(x_t)\theta_{t-1,d} = \phi_v^\top(x_t)A_{t-1}^{-1}B_{t-1}
  \]
  \[
  \sigma_t^2 = \sigma_n^2[1 + \phi_v^\top(x_t)A_{t-1}^{-1}\phi_v(x_t)]
  \]

- **MAP estimate of** $x_t$

  \[
  \hat{x}_t = \arg\max_{x_t} p(y_t|Y_{t-1}, \hat{X}_{t-1}, x_t) \cdot p(x_t)
  \]
  \[
  = \arg\min_{x_t} \frac{1}{2\sigma_t^2}||y_t - \mu_t||^2 + D_y \log \sigma_t + \frac{1}{2\sigma_x^2}||x_t||^2
  \]

- **Recursive updates**

  \[
  B_t = B_{t-1} + \phi_v(\hat{x}_t)y_t^\top
  \]
  \[
  A_t = A_{t-1} + \phi_v(\hat{x}_t)\phi_v^\top(\hat{x}_t)
  \]

  ➢ In practice, updates performed on the Cholesky factor of $A_t$
Ensemble online RF-based GPLVM

**Challenge:** Online choice of kernel?

**Remedy:** Ensemble of $M$ experts, each with a different kernel $\kappa^m$

**Algorithm** for incoming $y_t$

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

$$\hat{x}_t^m := \arg\max_x p(y_t|Y_{t-1}, i=m, \hat{X}_{t-1}^m, x) p(x) \quad m = 1, \ldots, M$$

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

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

- Meta-learner updates expert weights

$$w_t^m := \Pr(i=m|Y_t, \{X_{t}^{(\mu)}\}_{\mu=1}^M) \propto w_{t-1}^m p(y_t|Y_{t-1}, i=m, X_{t-1}^m, \hat{x}_t^m) \quad m = 1, \ldots, M$$

---

GP-based test for dimensionality reduction

- Broadens probabilistic PCA using a GP latent variable model (LVM)
  - An independent GPR per dimension $d$

Goal: Given $D_y \times 1$ vectors $\{y_t\}_{t=1}^T$, find latent $D_x \times 1$ vectors $\{x_t\}_{t=1}^T$

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

Testing (E)RF-GPLVM

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

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 $\mathcal{G} := \{\mathcal{V}, A_N\}$ with vertex set $\mathcal{V}$ and $N \times N$ adjacency matrix $A_N$

- 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 $\mathcal{O}$

**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 $y_n := [y_1, \ldots, y_n]^\top$ to predict $y_{n+1}$ and correct after $y_{n+1}$ is observed
Incremental **Graph-adaptive EGP**

**Idea:** Use one-hop connectivity vector $a_n$ as input: $f_n = f(a_n)$

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

  $N(\theta^m_n; \hat{\theta}^m_n, \Sigma^m_n) \xrightarrow{a_{n+1}} N(y_{n+1}; \hat{y}^m_{n+1|n}, (\sigma^m_{n+1|n})^2) \xrightarrow{y_{n+1}} N(\theta^m_{n+1}; \hat{\theta}^m_{n+1}, \Sigma^m_{n+1})$

  $$\hat{y}^m_{n+1|n} = \phi^m_V(a_{n+1})\hat{\theta}^m_n$$  
  $$\theta^m_{n+1} = \theta^m_n + (\sigma^m_{n+1|n})^{-2} \Sigma_n \phi^m_V(a_{n+1})(y_{n+1} - \hat{y}^m_{n+1|n})$$  
  $$(\sigma^m_{n+1|n})^2 = \phi^m_V(a_{n+1})\Sigma_n \phi^m_V(a_{n+1}) + \sigma^2$$  
  $$\Sigma^m_{n+1} = \Sigma^m_n - (\sigma^m_{n+1|n})^{-2} \Sigma_n \phi^m_V(a_{n+1})\phi^m_V(a_{n+1})^\top \Sigma_n$$

- **Meta-learner**

  $$\sum_{m=1}^{M} w^m_n N(y_{n+1}; \hat{y}^m_{n+1|n}, (\sigma^m_{n+1|n})^2)$$

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

  $$\sigma^2_{n+1|n} = \sum_{m=1}^{M} w^m_n [(\sigma^m_{n+1|n})^2 + (\hat{y}_{n+1|n} - \hat{y}^m_{n+1|n})^2]$$

- **Weight updates**

  $$w^m_{n+1} = \frac{w^m_n N(y_{n+1}; \hat{y}^m_{n+1|n}, (\sigma^m_{n+1|n})^2)}{\sum_{m'=1}^{M} w^m_{n'} N(y_{n+1}; \hat{y}^{m'}_{n+1|n}, (\sigma^{m'}_{n+1|n})^2)}$$

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

### GradEGP vis-à-vis GCNs

Comparison with graph convolutional networks (GCNs)

<table>
<thead>
<tr>
<th>GradEGP</th>
<th>Conventional GCNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental → reduced storage</td>
<td>Batch approach → storage demand</td>
</tr>
<tr>
<td>Scalable online updates</td>
<td>Demanding training phase</td>
</tr>
<tr>
<td>Bayesian → uncertainty quantification</td>
<td>Deterministic → only point estimates</td>
</tr>
<tr>
<td>No need for additional nodal features</td>
<td>Additional nodal features needed</td>
</tr>
<tr>
<td>Input: encrypted version connectivity pattern of nodes → privacy</td>
<td>Input: connectivity pattern of nodes</td>
</tr>
</tbody>
</table>
Testing GradEGP

**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) \(\text{nMSE}_n := n^{-1} \sum_{n'=1}^{n} (y_{n'} - \hat{y}_{n'|n'-1})^2 / s_y^2\)
- Runtime
Performance with uncertainty quantification

- NMSE versus $n$

- GradEGP with uncertainty quantification

- GradEGP outperforms alternatives and estimates stay within confidence intervals

Runtime comparison

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

Higher-order interactions

Q. More informative graph guidance than $a_n$? A. How about per-node “egonet”?

- Egonet of node $n$
  - Node $n$
  - Direct neighbors of node $n$
  - All edges connecting direct neighbors

- $N \times N$ adjacency matrix of node $n$ egonet: $A_n^{ego}$
  - Sparse matrix due to limited connectivity

- “Egonet feature” vector $x_n^{ego}$

Model: Use egonet feature vector $x_n^{ego}$ as input

$$f_n = f(x_n^{ego}) \rightarrow \text{“GradEGP-ego”}$$

Egonet feature vector per node $n$

- $x_{ego}^n$ captures connectivity of node $n$ to all nodes through its egonet
  
  - Degree of node $n$ \[ d_n := \sum_{n' = 1}^{N} A_{n'}^n(n', n) \]
  
  - Connectivity of any node $m$ with node $n$ as a sum of edge weights with its egonet
    \[ c_{Ei}^n(m) = \alpha \sum_{n' \in N_{m}} c_{Ei}^n(m') \]

- Collectively, as eigenvector of max eigenvalue

\[ A_n^e v = \alpha^{-1} v \]

- Our $x_{ego}^n$ comprises degree and eigenvector centralities (a.k.a. `vertex centrality')

\[ x_{ego}^n := \begin{bmatrix} d_n \\ c_{Ei}^n \end{bmatrix} \]

- $x_{ego}^n$ can also include edge centrality, clustering coefficient, network cohesion [Kolaczyk'96]

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 ...

Dr. Qin Lu
UofM

G.-V. Karanikolas
UofM

K. Polyzos
UofM

Prof. Y. Shen
UCI

http://spincom.umn.edu

Questions?

Dr. P. Traganitis
UofM