Locally induced GPs for modelling large-scale simulations

Austin Cole

GlaxoSmithKline
formerly of Virginia Tech

UQSay
February 17, 2022

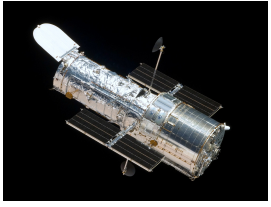
Table of Contents

Table of Contents

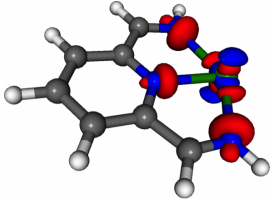
Computer simulation models

- Computer programs that mathematically models a physical process by incorporating domain knowledge
- Often desired over physical experiments due to cost and time savings
- Applications span many different fields:

Engineering



Science



Public Health

Gaussian processes (GPs)

- Assume the response Y_N follows a multivariate Normal distribution:
$$Y_N \sim \mathcal{N}_N(\mu, \Sigma_N), \text{ where } \Sigma_N = \tau^2(K_N + g\mathbb{I}_N)$$
- A common simplification assumes $\mu = 0$, offloading the relationship in the data to Σ_N
- Includes hyperparameters: $\psi = \{\tau^2 \text{ (scale)}, g \text{ (nugget)}, \text{and } \theta \text{ (lengthscale)}\}$
- The correlation matrix K_N is defined as a function of the squared Euclidean distance between pairs of inputs, e.g.

$$k(x_i, x_j) = \exp\left\{ - \frac{||x_i - x_j||^2}{\theta} \right\}$$

GP predictions

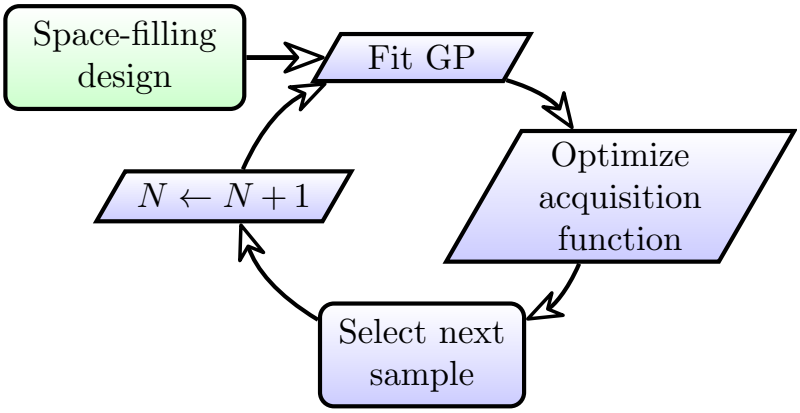
- Conditional predictive equations:

$$\mu(x^\star|X_N, Y_N, \psi) = \Sigma(x^\star, X_N)\Sigma_N^{-1}Y_N$$

$$\sigma^2(x^\star|X_N, \psi) = \Sigma(x^\star, x^\star) - \Sigma(x^\star, X_N)\Sigma_N^{-1}\Sigma(x^\star, X_N)^\top$$

- Interpolates well, providing accurate out-of-sample predictions
- Conditioning on hyperparameters, $\sigma^2(x^\star)$ doesn't depend on Y_N

GP sequential design for computer experiments



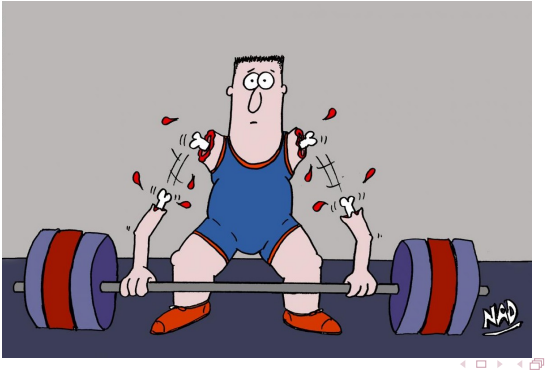
Sequential design acquisition functions

- **Expected Improvement (EI)**: balances information gain close to max/min versus reducing model uncertainty
- **Entropy**: maximizes information gain about events
- **Integrated mean-squared error (IMSE)**: reduces global variance

$$\text{IMSE}(X_N) = \int_{x \in \mathcal{X}} \frac{\sigma_N^2(x)}{\tau^2} dx$$

Limitations of GPs

- The standard GP framework computationally relies heavily on the inversion of the covariance, Σ_N, an $N \times N$ matrix
- Without special structure in Σ_N, the computation is $O(N^3)$, which limits modern computers to data sets in the 1000s



Methods using GPs with Big Data

- Approximation of the covariance or precision matrix:
  - Nearest Neighbor GPs (Datta et al., 2016)
  - Vecchia approximation (Katzfuss and Guinness, 2017)
  - Compactly supported kernels (Kaufman et al., 2011)
  - Inducing points (Snelson and Ghahramani, 2006)
- Local models:
  - Piecewise GPs (Kim et al., 2005)
  - Treed GPs (Gramacy and Lee, 2008)
  - Local GP approximation (Gramacy and Apley, 2015)

Local approximate GPs (LAGP)

- Introduced by Gramacy and Apley (2015)
- Fits a distinct GP for each predictive location
- Uses a **subset of the data** for each local neighborhood around a predictive location
- Produces accurate mean predictions
- Local scope limits uncertainty approximation

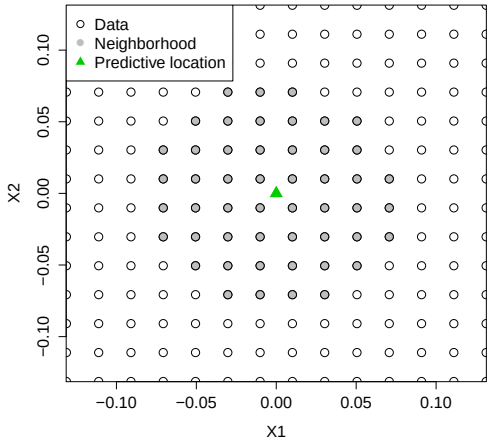


Table of Contents

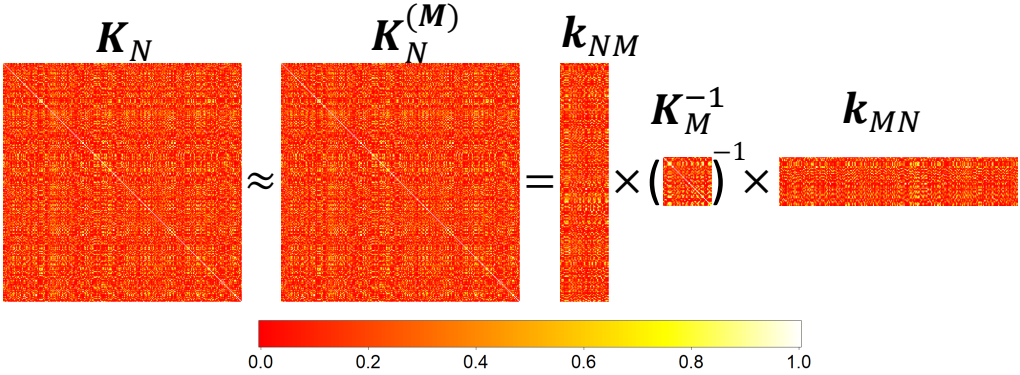
Inducing points formulation

- Let $\bar{X}_M$ be **inducing points**, a set of $M \times d$ ($M \ll N$) latent variables that represent design locations
- Using inducing points, we can creates a low rank covariance matrix with

$$K_N \approx \bar{K} = k_{NM} K_M^{-1} k_{MN} + \text{Diag}(K_N - k_{NM} K_M^{-1} k_{MN})$$

- Woodbury identities allow for more computationally efficient calculations of $\bar{K}^{-1}$ and $|\bar{K}|$
- Reduces the computational burden to $O(NM^2)$

Correlation Approximation with Inducing Points



Selection of Inducing Points

- Early literature used optimization of the negative log-likelihood
- Jointly optimizing the inducing points ($\bar{X}_M$) is very difficult due to high dimensionality and many local minima

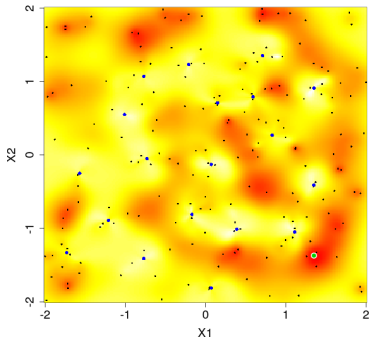


Figure 1: Negative log-likelihood surface

LIGP Framework

1. **Initialize**: neighborhood size (n), number of inducing points (m), and hyperparameters (θ, g)
   For each predictive location $x^\star$:
2. Build a **local neighborhood** of $\mathbf{n}$ points nearest to $x^\star$
3. Create an **inducing point design** centered at $x^\star$
   - Select points with weighted integrated mean-squared error
   - Scale a space-filling design
4. Optimize hyperparameters

Steps 2-4 are computationally independent for each $x^\star$ and can be done in parallel

Log-likelihood used for optimization

$$\begin{aligned}\ell(X_n, Y_n, \bar{X}_m, \tau^2, \theta) \\ \propto & \text{const.} - n\log(\tau^2) - \log|Q_m^{(n)}| + \log|K_m| - \mathbf{1}_n^\top \log(\Omega_n^{(m)})\mathbf{1}_n \\ & - \tau^{-2} Y_n^\top \left(\Omega_n^{-1(m)} - \Omega_n^{-1(m)} k_{nm} Q_m^{-1(n)} k_{nm}^\top \Omega_n^{-1(m)}\right) Y_n.\end{aligned}$$

where $Q_m^{(n)} = K_m + k_{nm}^\top \Omega_n^{-1(m)} k_{nm} + \epsilon_Q \mathbb{I}_m$ and
$\Omega_n^{(m)} = \text{Diag}\{K_n - k_{nm} K_m^{-1} k_{nm}^\top\} + \epsilon_K \mathbb{I}_n$

Weighted Integrated Mean Squared Error (wIMSE)

- For a given predictive location $x^\star$, integrate the weighted predictive variance given a new proposed inducing point over the design space:

$$\int_{\tilde{x}\in\mathcal{X}} k_\theta(\tilde{x}, x^\star)\frac{\sigma^2_{m+1,n}(\tilde{x})}{\tau^2}\, d\tilde{x}$$

- By using a Gaussian kernel for the weight (centered at $x^\star$), we can express wIMSE in closed form (Binois et al., 2018b):

$$\frac{\sqrt{\theta\pi}}{2}\prod_{k=1}^{d}\left(\mathsf{erf}\left\{\frac{x^\star - a_k}{\sqrt{\theta}}\right\} - \mathsf{erf}\left\{\frac{x^\star - b_k}{\sqrt{\theta}}\right\}\right) - \mathsf{tr}\left\{\left(K^{-1}_{m+1} - Q^{-1(n)}_{m+1}\right)W^*_{m+1}\right\}$$

wIMSE-based Inducing Point Design

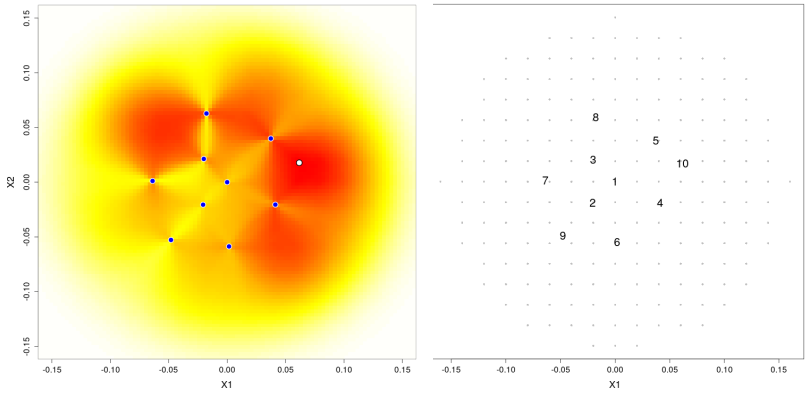


Figure 2: Left: wIMSE surface for selecting 10th inducing point; Right: order of inducing point selection

Inducing Points Template

1. Create an inducing point design centered around the middle of the design space
2. Shift the inducing point design to be centered at each $x^\star$

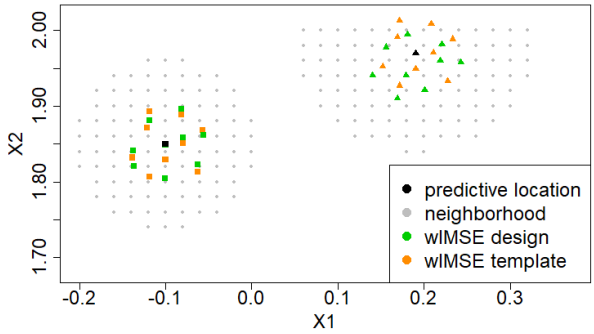


Figure 3: Inducing Point designs for two predictive locations

Space-filling Templates

We can scale a LatinHypercube sample (LHS) to use as the inducing points

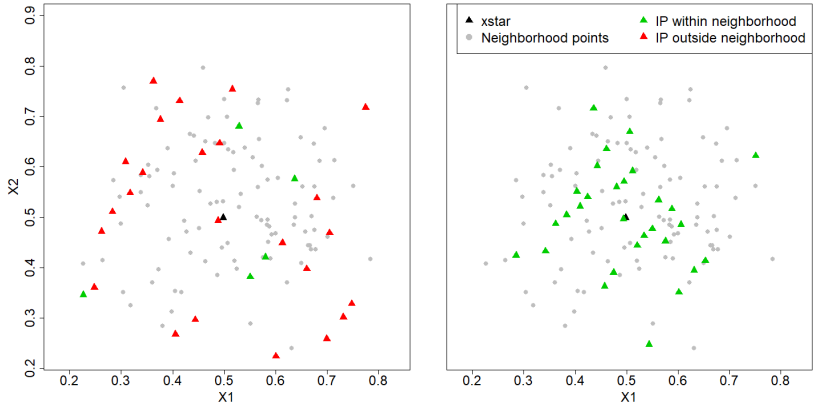
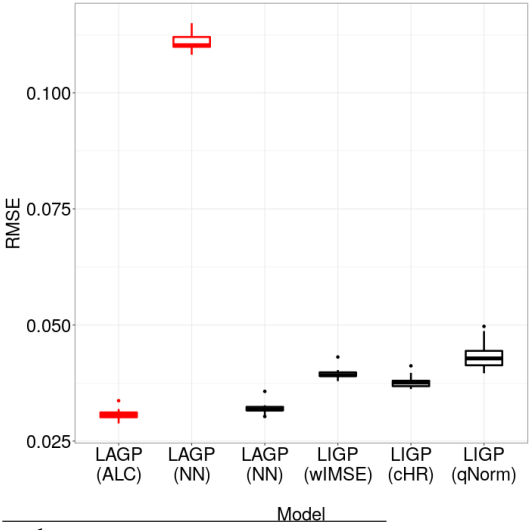


Figure 4: 2D projections of an 8D LHS. Left: Circumscribed HyperRectangle (cHR) scaling; Right: Inverse Gaussian CDF (qNorm) scaling

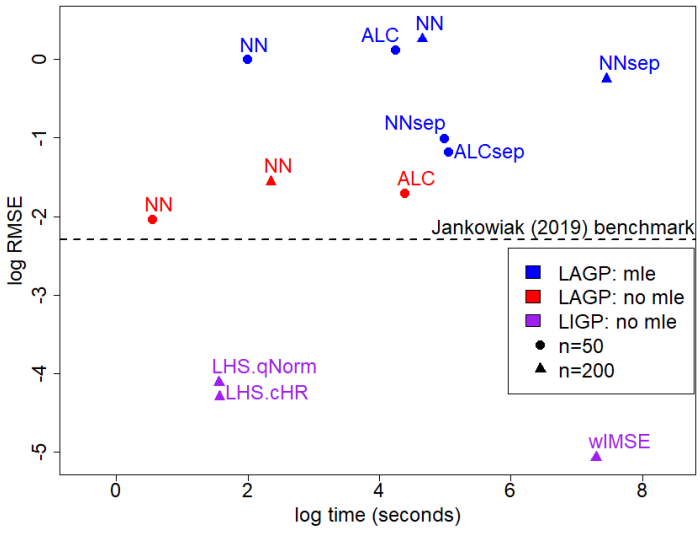
Borehole (8D): 100K training pts, 10K predictions



GP Method	n	Minutes[1]
LAGP(ALC)	50	0.95
LAGP(NN)	50	0.08
LAGP(NN)	150	1.25
LIGP(wIMSE)	150	3.04
LIGP(cHR)	150	0.71
LIGP(qNorm)	150	0.72

[1] LAGP is coded in `C`, while LIGP is coded in `R`

SARCOS (21D): ≈45K training, 4400 predictions



LIGP summary

- Optimizing inducing points globally is fraught with challenges
- Marrying local GP approximations with inducing points provides faster predictions that in some cases are more accurate
- Using an inducing point template (especially a space-filling design) provides further computational savings
- Efficient inducing point designs space-fill around the predictive location $x^\star$ (i.e. the neighborhood's center)

Table of Contents

Stochastic simulations pose a greater challenge

- Distinguishing signal-from-noise requires more data
  - Often include replicated inputs
- Input-dependent noise (heteroskedasticity)
  - Existing methods including stochastic kriging (Ankenman et al., 2010), and heteroskedastic GPs (hetGP; Binois et al., 2018a) buckle with a large number of unique locations

Leveraging the Woodbury identities (again)

From Gneiting and Raftery (2007),

$$(\mathbf{B}+\mathbf{CDE})^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}(\mathbf{D}^{-1}+\mathbf{EB}^{-1}\mathbf{C})^{-1}\mathbf{EB}^{-1} \tag{1}$$
$$\log|\mathbf{B}+\mathbf{CDE}| = \log|\mathbf{D}^{-1}+\mathbf{EB}^{-1}\mathbf{C}| + \log|\mathbf{D}| + \log|\mathbf{B}|,$$

$$\mathbf{B} = \text{Diag}\{K_n - \mathbf{U}k_{\bar{n}m}K_m^{-1}k_{\bar{n}m}^\top\mathbf{U}^\top\} + g\mathbb{I}_n, \quad \mathbf{D} = K_m^{-1} \quad \text{and } \mathbf{E} = \mathbf{C}^\top = k_{\bar{n}m}^\top\mathbf{U}^\top.$$

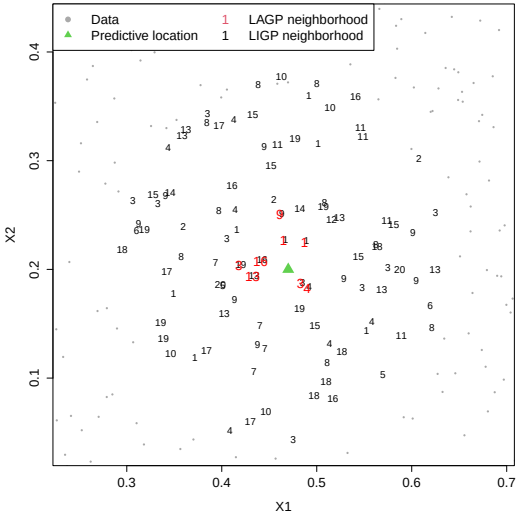
where m is the number of inducing points, n is the local neighborhood size, and $\bar{n}$ is the number of **unique** neighborhood locations $\rightarrow m \ll \bar{n} \ll n$.
$\mathbf{U}$ is a block diagonal matrix comprises of 1-vectors of lengths equal to the number of replicates at each unique design location.

Log-likelihood with low rank matrices

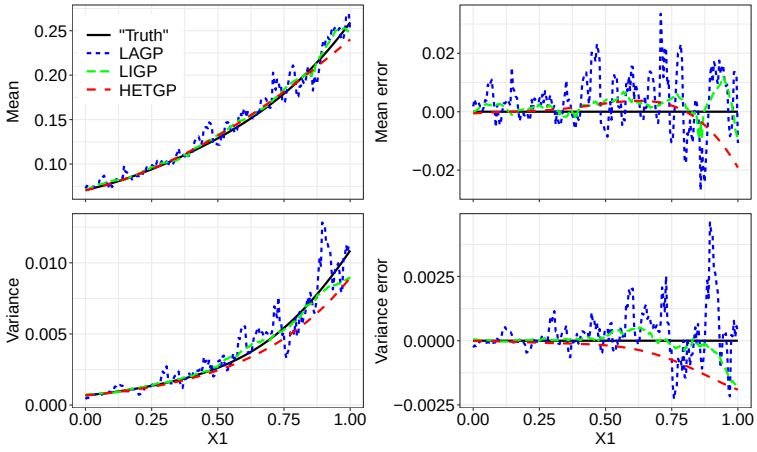
$$\ell(X_n, Y_n, \bar{X}_m; \tau^2, \theta, g) \propto -n\log(\tau^2) - \log|Q_m^{(\bar{n})}| + \log|K_m| - \sum_{i=1}^{\bar{n}} a_i \log \omega_i^{(\bar{n},m)}$$
$$- \tau^{-2}\left(Y_n^\top \Omega_n^{-1(m)} Y_n - \bar{Y}_{\bar{n}}^\top \Lambda_{\bar{n}}^{(m)} k_{\bar{n}m} Q_m^{-1(\bar{n})} k_{\bar{n}m}^\top \Lambda_{\bar{n}}^{(m)} \bar{Y}_{\bar{n}}\right),$$

where $Q_m^{(\bar{n})} = K_m + k_{\bar{n}m}^\top \Lambda_{\bar{n}}^{(m)} k_{\bar{n}m}$, $\Lambda_{\bar{n}}^{(m)} = A_{\bar{n}} \Omega_{\bar{n}}^{-1(m)}$,
$A_{\bar{n}}$ is a diagonal matrix of the number of replicates at each $X_{\bar{n}}$, and $\omega_i^{(\bar{n},m)}$ is the i^{th} diagonal term in $\Omega_{\bar{n}}^{(m)}$. $\bar{Y}_{\bar{n}}$ is a vector of averaged responses for each location in $X_{\bar{n}}$.

Building a local neighborhood with replicates



Estimating mean and variance in SIR model

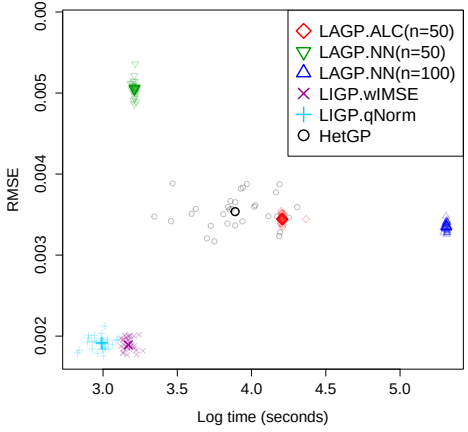


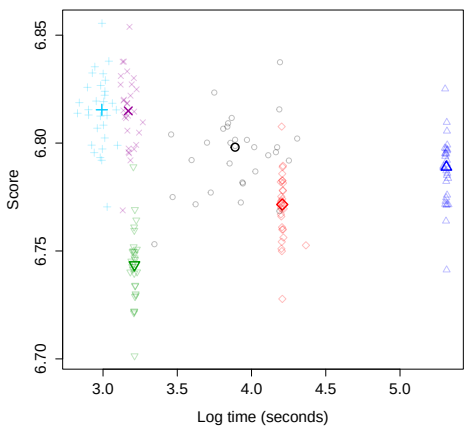
Results versus LAGP, HetGP

- LAGP models fit with neighborhoods $n = 50$ (default)
- HetGP models use random subset of $\bar{N} = 1000$
- LIGP models use $m = 10$ (2d) and $m = 30$ (4d) and $\bar{n} = 100$
- 30 Monte Carlo repetitions were conducted, each with $\bar{N} = 10000$ unique data locations (with varying replication strategies) and $N^\star = 10000$ unique predictive locations
- Comparison metrics are:

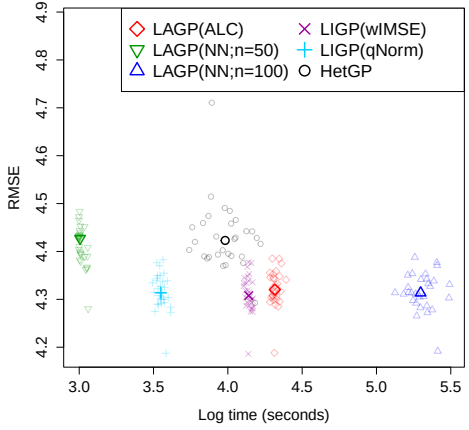
$$\text{RMSE} = \sqrt{\sum_{i=1}^{N^\star} (\hat{\mu}(\mathbf{x}_i) - y(\mathbf{x}_i))^2} \qquad \text{Score} = -\sum_{i=1}^{N^\star} \left(\frac{\hat{\mu}(\mathbf{x}_i) - y(\mathbf{x}_i)}{\hat{\sigma}^2(\mathbf{x}_i)} \right)^2 - \sum_{i=1}^{N^\star} \hat{\sigma}^2(\mathbf{x}_i)$$

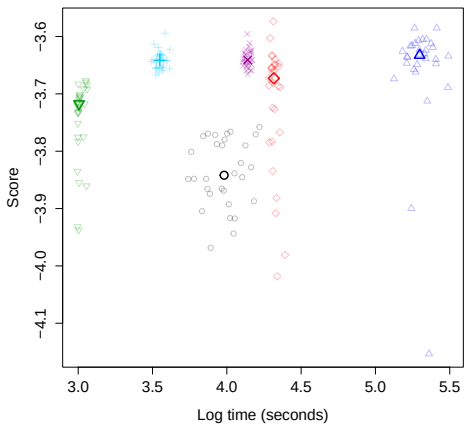
Herbie's Tooth (2d) experiment





Ocean oxygen concentration (4d) experiment





Summary

- LIGP marrying local GP approximations with inducing points, opening the door for **larger local neighborhoods and faster computation**
- When the data includes replicates, building neighborhoods based on the number of **unique points** allows more data to be included
- Efficient inducing point designs space-fill around the predictive location $\mathbf{x}^\star$ (i.e. the neighborhood's center), allowing us to use **template schemes**
- LIGP can be **faster** than LAGP/HetGP and often provide **more accurate mean predictions and noise estimates**

To learn more

- Cole, D. Austin, Ryan B. Christianson, and Robert B. Gramacy. "Locally induced Gaussian processes for large-scale simulation experiments." Statistics and Computing 31.3 (2021): 1-21.
- Cole, D. Austin, Robert B. Gramacy, and Mike Ludkovski. "Large-scale local surrogate modeling of stochastic simulation experiments." arXiv preprint arXiv:2109.05324 (2021).
- liGP `R` package: `https://CRAN.R-project.org/package=liGP`

Special Thanks

- Robert B. Gramacy
- Ryan B. Christianson, Mike Ludkovski
- Julien Bect
- National Science Foundation (Grant DMS-1821258)

Bibliography I

Ankenman, B., Nelson, B. L., and Staum, J. (2010). "Stochastic kriging for simulation metamodeling." *Operations research*, 58, 2, 371–382.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2018a). "Practical heteroskedastic Gaussian process modeling for large simulation experiments." *Journal of Computational and Graphical Statistics*, 27, 4, 808–821.

Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2018b). "Replication or exploration? Sequential design for stochastic simulation experiments." *Technometrics*, 61, 1, 7–23.

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). "Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets." *Journal of the American Statistical Association*, 111, 514, 800–812.

Gneiting, T. and Raftery, A. E. (2007). "Strictly proper scoring rules, prediction, and estimation." *Journal of the American statistical Association*, 102, 477, 359–378.

Bibliography II

Gramacy, R. B. and Apley, D. W. (2015). "Local Gaussian process approximation for large computer experiments." *Journal of Computational and Graphical Statistics*, 24, 2, 561–578.

Gramacy, R. B. and Lee, H. K. (2008). "Bayesian treed Gaussian process models with an application to computer modeling." *Journal of the American Statistical Association*, 103, 483, 1119–1130.

Katzfuss, M. and Guinness, J. (2017). "A general framework for Vecchia approximations of Gaussian processes." *arXiv preprint arXiv:1708.06302*.

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011). "Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology." *The Annals of Applied Statistics*, 5, 4, 2470–2492.

Kim, H. M., Mallick, B. K., and Holmes, C. C. (2005). "Analyzing nonstationary spatial data using piecewise Gaussian processes." *Journal of the American Statistical Association*, 100, 470, 653–668.

Bibliography III

Snelson, E. and Ghahramani, Z. (2006). "Sparse Gaussian processes using pseudo-inputs." *Advances in Neural Information Processing Systems 18*, 1257–1264.