Random feature expansions guided by input sensitivity

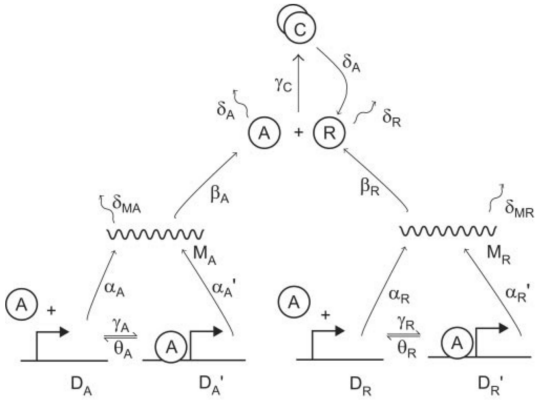
John Darges

Department of Mathematics
Emory University

March 20, 2025

A Motivating Example

Consider the genetic oscillator reaction network modeled by an ODE system[1]



- How does uncertainty in rate parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ lead to uncertainty in estimation of average concentration of R?
- This gives a quantity of interest $R_{avg} = F(\boldsymbol{\theta})$, $\boldsymbol{\theta} \sim \mathcal{D}_\Theta$
- Each evaluation of F requires solving a stiff ODE system

[1] J.G. Vilar, H.Y. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. 2002.

Surrogate Modeling

- **Problem:** Models can be computationally expensive to evaluate while budgets are limited
- **Solution:** Use surrogate $\widetilde{F} \approx F$ that emulates target model but is cheap to evaluate
- Surrogate model built using limited number of model evaluations $\{(\boldsymbol{\theta}_i, F(\boldsymbol{\theta}_i)\}_{i=1}^{M}$
- Target model typically treated as a **black box**
- Any information about model structure can help construct better surrogates

Decompositions of Multivariable Functions

- Multivariable functions $F : \mathbb{R}^n \to \mathbb{R}$ can be decomposed (in different ways) into terms depending on groups of inputs[2]

$$F = \sum_{\boldsymbol{u} \subseteq \{1,\ldots,n\}} F_{\boldsymbol{u}},$$

- Terms $F_{\boldsymbol{u}}$ depends only on subset of inputs θ_j, $j \in \boldsymbol{u}$
- $F_{\boldsymbol{u}}$ and $F_{\boldsymbol{v}}$ are orthogonal
- There are 2^n terms representing all possible input interactions
- Examples include Hoeffding-Sobol' (aka ANOVA) decomposition[3] and anchored decomposition[4]
- Knowing important inputs or interaction terms, e.g. with sensitivity analysis, can make better surrogates

[2] F. Kuo, I. Sloan, G. Wasilkowski, H. Woźniakowski. On Decompositions of Multivariable Functions. 2010.
[3] A.B. Owen. Appendix A: The ANOVA decomposition of $[0,1]^d$ in *Practical Quasi-Monte Carlo Integration*.
[4] I.M. Sobol'. Multidimensional Quadrature Formulas and Haar Functions. 1969.

Random Feature Expansions (RFEs)

- Approximation method based on randomly choosing basis functions called features[5]

$$\widetilde{F}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_k \phi(\boldsymbol{\omega}_k^\top \boldsymbol{\theta}), \quad \boldsymbol{\omega}_k \sim \mathcal{D}_\Omega, \quad \phi : \mathbb{R} \to \mathbb{R}$$

- Feature weights $\boldsymbol{\omega}$ are sampled i.i.d. from chosen probability distribution $\mathcal{D}_\Omega$
- Usually all weights in weight matrix $\mathbf{W} = [\boldsymbol{\omega}_1 \dots \boldsymbol{\omega}_k]$ are sampled i.i.d.
- Coefficients α_k determined by solving linear least squares problem

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{M} \left(F(\boldsymbol{\theta}_i) - \sum_{k=1}^{K} \alpha_k \phi(\boldsymbol{\omega}_k^\top \boldsymbol{\theta}_i) \right)^2$$

- Equivalent to single layer neural networks with random weights[6]

[5] A. Rahimi, B. Recht. Uniform approximation of functions with random bases. 2008.
[6] W. Cao, X. Wang, Z. Ming, J. Gao. A review on neural networks with random weights. 2018.

Sparse Random Features

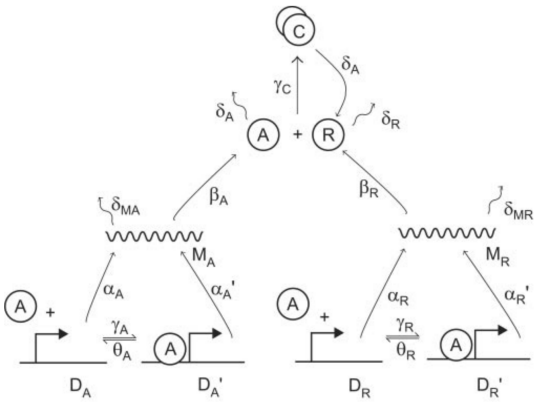
- Recent work proposed imposing interaction structure on RFEs through sparse weights
- Sparse features $\phi(\boldsymbol{\omega}_k^\top \boldsymbol{\theta})$ have weights where most entries in $\boldsymbol{\omega}_k$ are 0
- Sparse random features proposed in[7] assume target model only has low order interactions
- Weight sparsity in[8] determined by choosing a hyperparameter
- Sparse RFEs in[9] sample weights based on ANOVA decomposition of target model
- **Question:** Can we learn model structure without (or along the way to) training an accurate surrogate?

[7] A. Hashemi, H. Schaeffer, R. Shi, U. Topcu, G. Tran, R. Ward. Generalization bounds for sparse random feature expansions. 2023.

[8] J.E.D., A. Alexanderian, P.A. Gremaud. Extreme Learning Machines for Variance-Based Global Sensitivity Analysis. 2024.

[9] L. Weidensager, D. Potts ANOVA-boosting for Random Fourier Features. 2024.

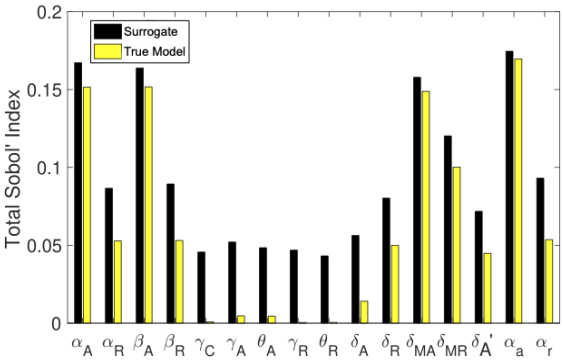
Return to Motivating Example



- Quantity of interest is $R_{avg} = F(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \sim \mathcal{D}_{\Theta}$
- Emulate F using a RFE surrogate $\widetilde{F}$
- Use surrogate to find which parameters have greatest impact on R_{avg} through sensitivity analysis (total Sobol' indices)
- Sensitivity indices can tell us which terms in multivariable decomposition matter and which do not

Sensitivity Analysis of Example

- RFE surrogate uses weights sampled from standard normal distribution
- Sensitivity index measures how much an input parameter θ_i contributes to variance in R_{avg}
- Sensitivity indices are inaccurate but still provide useful insights
- **Takeaway:** Rough surrogate may give hints about true model structure that can be used to build a better surrogate



Proposed Approach

Current Methods

- With no information about model, treat all weights identically and sample i.i.d.
- Require structural information about model before determining how to sample weights for RFE

Our Approach

- Introduce random sparsity to feature weights based on sensitivity of inputs they correspond to
- E.g. If F is not sensitive to θ_i, all its corresponding weights should be sparse (set to 0)
- Sample entries in weight vector $\boldsymbol{\omega}^i$ corresponding to θ_i as random variable $\omega^i = Z_i X$, where $X \sim \mathcal{N}(0, 1)$, $Z_i \sim \text{Bern}(1 - \rho_i)$,
- ρ_i gives sensitivity of θ_i (**Sobol' index**, derivative-based measures, Shapley value, etc.)
- Begin by training a rough surrogate to determine estimates for ρ_i, then build new surrogate with updated sampling distribution

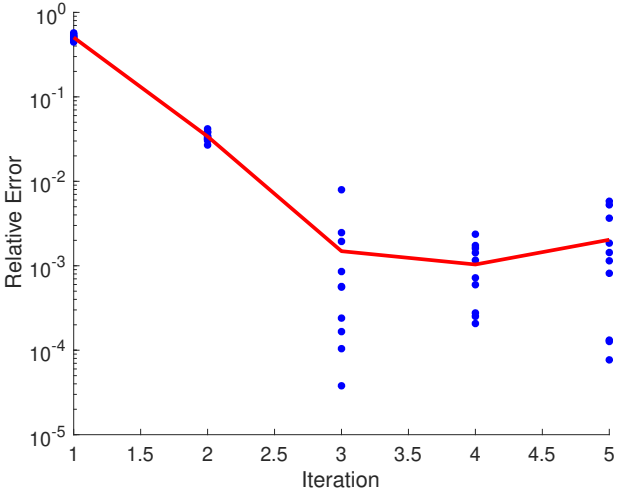
Algorithm

1. Evaluate model to form training set for surrogate
2. Set weight sampling distribution to standard normal distribution, $\boldsymbol{\omega}^i \sim \mathcal{N}(0, 1)$ for all i
3. Sample feature weights and build RFE surrogate
4. Estimate input sensitivities ρ_i using the surrogate
5. Update weight sampling distribution with random sparsity, $\omega^i = Z_i X$, where $X \sim \mathcal{N}(0, 1)$, $Z_i \sim \text{Bern}(1 - \rho_i)$
6. Iteratively repeat 2–5

How does this work in practice?

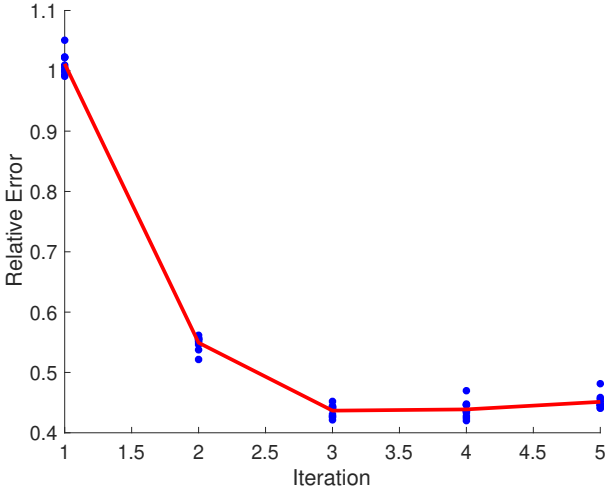
Example: Grienwank Function

$$F(\boldsymbol{\theta}) = \sum_{k=1}^{50} \frac{\theta_k^2}{4000} + \prod_{k=1}^{50} \left(\cos\left(\frac{\theta_k}{\sqrt{k}}\right) + 1\right), \quad \boldsymbol{\theta} \sim \mathcal{U}([-600, 600]^{50}).$$



Example: Sobol' g-function

$$F(\boldsymbol{\theta}) = \prod_{k=1}^{10} \frac{|4x_k - 2| + a_k}{1 + a_k}, \quad a_k = \frac{k-2}{2}, \quad \boldsymbol{\theta} \sim \mathcal{U}([0,1]^{10})$$



Analysis of approach

- Does altering the sampling distribution based on weight sensitivity provide benefits for certain classes of functions?

- Let's try to study what is going on using reproducing kernels

Reproducing Kernels[10]

- A function $\kappa : X \times X \to \mathbb{R}$ is a kernel if the matrix $K_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is symmetric positive definite for any $\{\boldsymbol{x}_i\}_{i=1}^{M} \subset X$.
- Associated to κ is a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\kappa$
- For any $F \in \mathcal{H}_\kappa$, we have that $F(\boldsymbol{x}) = \langle F, \kappa(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}_\kappa}$
- κ defines an operator $T_\kappa : L^2(X) \to L^2(X)$, whose image is $\mathcal{H}_\kappa$

$$T_\kappa g(\boldsymbol{x}) = \int_X g(\boldsymbol{y})\kappa(\boldsymbol{x}, \boldsymbol{y})\, d\mu(\boldsymbol{y})$$

- If $\mathcal{H}_\kappa$ is dense in $L^2(\Omega)$, can approximate functions using

$$\widetilde{F}(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j \kappa(\boldsymbol{x}, \boldsymbol{x}_j)$$

[10] N. Aronszajn. Theory of Reproducing Kernels. 1950.

Feature Map Representation

- Kernels can be obtained through a certain feature map $\Phi : \Omega \times X \to \mathbb{R}$, $\Phi(\boldsymbol{\omega}, \boldsymbol{x}) = \phi(\boldsymbol{\omega}^\top \boldsymbol{x})$ and measure τ on Ω by

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \int_\Omega \phi(\boldsymbol{\omega}^\top \boldsymbol{x}) \phi(\boldsymbol{\omega}^\top \boldsymbol{y}) \, d\tau(\boldsymbol{\omega})$$

- Approximating the kernel by Monte Carlo integration leads to random feature methods[11]
- We can define another transformation $T_\phi : L^2(\Omega) \to L^2(X)$ which has the same image as T_κ

$$T_\phi \beta(\boldsymbol{x}) = \int_\Omega \beta(\boldsymbol{\omega}) \phi(\boldsymbol{\omega}^\top \boldsymbol{x}) \, d\tau(\boldsymbol{\omega})$$

- Approximating the integral by Monte Carlo leads to random feature expansions

[11] A. Rahimi, B. Recht. Random Features for Large Scale Kernel Machines. 2007.

How Sensitivity Changes the Weight Measure

- Suppose we have a feature map $\phi(\boldsymbol{\omega}^\top \boldsymbol{x})$
- Let $\omega_1, \dots \omega_n$ be i.i.d. with law ν
- Then τ is a product measure $\tau = \tau_1 \otimes \cdots \otimes \tau_n$, $\tau_i = \nu$
- Introducing sparsity via the algorithm results in a new measure τ_S
- $\tau_S = \tau_k \otimes \cdots \otimes \tau_n$ with $\tau_i = \rho_i \nu + (1 - \rho_i)\delta_0$, where δ_0 is Dirac measure
- Explicit form for τ_S is

$$\tau_S = \sum_{\boldsymbol{u} \subseteq \{1,\dots,n\}} \Big(\bigotimes_{i \in \boldsymbol{u}} \rho_i \nu \bigotimes_{i \notin \boldsymbol{u}} (1 - \rho_i)\delta_0 \Big)$$

- What kernel do we get when we pair this with the feature map?

Kernel Induced by Sensitivity

- Let $\kappa(\boldsymbol{x}, \boldsymbol{y}) = \int_\Omega \phi(\boldsymbol{\omega}^\top \boldsymbol{x}) \phi(\boldsymbol{\omega}^\top \boldsymbol{y})\, d\tau(\boldsymbol{\omega})$
- When we replace τ with τ_S, we get a new kernel

$$\kappa_S(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{u} \subseteq \{1,\ldots,n\}} \gamma_{\boldsymbol{u}} \kappa_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{y})$$

- With coefficients $\gamma_{\boldsymbol{u}} = \prod_{i \in \boldsymbol{u}} \rho_i \prod_{i \notin \boldsymbol{u}} (1 - \rho_i)$
- And kernels $\kappa_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{y}) = \int_{\Omega^{|\boldsymbol{u}|}} \phi(\boldsymbol{\omega}_{\boldsymbol{u}}^\top \boldsymbol{x}_{\boldsymbol{u}}) \phi(\boldsymbol{\omega}_{\boldsymbol{u}}^\top \boldsymbol{y}_{\boldsymbol{u}})\, d\tau(\boldsymbol{\omega}_{\boldsymbol{u}})$
- $\boldsymbol{\omega}_{\boldsymbol{u}}$ denotes vector of entries of $\boldsymbol{\omega}$ corresponding to indices in $\boldsymbol{u}$

Norm in RKHS of κ_S

- Norm of $F \in \mathcal{H}_S$ is given by

$$\|F\|_S^2 = \min_{F=\sum_{\boldsymbol{u}} F_{\boldsymbol{u}}} \sum_{\boldsymbol{u}\subseteq\{1,\ldots,n\}} \frac{1}{\gamma_{\boldsymbol{u}}} \|F_{\boldsymbol{u}}\|_{\boldsymbol{u}}^2, \quad F_{\boldsymbol{u}} \in \mathcal{H}_{\boldsymbol{u}}$$

- Here $\mathcal{H}_{\boldsymbol{u}}$ is the RKHS that corresponds to $\kappa_{\boldsymbol{u}}$

- Norm bears resemblance to norms of weighted spaces used in analysis of quasi-Monte Carlo[12]

[12] I. Sloan, H. Woźniakowski. When Are Quasi-Monte Carlo Algorithms Efficient for High Dimensional Integrals? 1998.

Summary and Goals

- Proposed an algorithm for random feature expansions that modifes weight sampling by taking relative input sensitivities into account
- Can we prove updating weight sampling distribution based on input sensitivity improves efficiency (fewer features/data points needed to achieve same accuracy) for certain classes of functions?
- Can we prove that input sensitivities can be learned iteratively along the way?
- Are there better approaches to input sensitivity and/or how to alter the sampling distribution?
- Thanks to Laura Weidensager and Elizabeth Newman!