Computer model calibration, emulation, and discrepancy analysis

M. J. Grosskopf

ISNET-6
October 11, 2018

National Nuclear Security Administration
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

Overview

- Model Calibration
- Emulation (and Gaussian processes)
- Model discrepancy approaches
- Conclusions

- Ask questions throughout!

Computer model calibration

Computer models are developed to better understand physical processes.

- The model encodes a numerical implementation of a mathematical description of the physical processes

$$\mathcal{M}(\mathbf{x}, \mathbf{t})$$

- **x** is a vector of control inputs
- **t** is a vector of calibration and tuning parameters
- Also called *simulator*

What is calibration?

- We have a observable quantity of interest from the physical system: y
- Can extract the same quantity of interest from the model $f(M(\mathbf{x}, \mathbf{t}))$
  - For brevity, $\mathcal{M}$ can be dropped: $f(\mathbf{x}, \mathbf{t})$

- Calibration is learning what parameters of a computer model make its output consistent with observation
  - Generally, θ represents the "true" value of the parameter vector $\mathbf{t}$

What are some approaches to carry out the calibration?

- Optimization:
  - Minimizing some discrepancy metric between the model and observed data (Chi-squared/L2 norm)
  - Equivalently, maximizing some likelihood based on a particular model for stochastic error (Gaussian, Poisson, etc.)
  - Typically, $\hat{\theta}$ represents the optimal parameter vector

- Bayesian inference:
  - Give a prior for θ and likelihood for the observed data
  - Update using rules of probability
  - No one value $\hat{\theta}$ but a distribution summarizing information about θ

Statistical model for calibration

- A common statistical model for simulators is to assume additive Gaussian error:

$$y_i = f(\mathbf{x}_i, \boldsymbol{\theta}) + \epsilon$$

- Seems easy?

Complications

Not uncommon that $\mathcal{M}(\mathbf{x}, \mathbf{t})$ is time consuming to evaluate

- Simulators that take hours on supercomputers are not uncommon these days
- Often not feasible to embed evaluation of the model in a sampler
  - Sometimes not even feasible in an optimizer
- Replace $f(\mathbf{x}, \mathbf{t})$ with approximation: $\eta(\mathbf{x}, \mathbf{t})$

Gaussian processes as emulators

- Being a good Bayesian, when given an unknown function, we can put a prior on it:
  - Gaussian processes are priors on functions:
    $\eta(\mathbf{x}, \mathbf{t}) \sim \mathcal{GP}(\mu(\mathbf{x}, \mathbf{t}), \Sigma(\mathbf{x}, \mathbf{t}))$

- Why?
  - Flexible to capture complex, nonlinear response functions
  - Interpolates(-ish) observed values
  - Gives principled uncertainty estimates at unobserved locations

How Gaussian processes work:

- Pick a mean and covariance function (we'll get to that)
- Observe data
- Posterior mean:

$$\hat{y}(x') = \mu(\mathbf{x}') + k(\mathbf{x}, \mathbf{x}')^T \Sigma(\mathbf{x})^{-1}(y - \mu(\mathbf{x}))$$

- Posterior Covariance:

$$\hat{Cov}(x') = \Sigma(x') + k(\mathbf{x}, \mathbf{x}')^T \Sigma(\mathbf{x})^{-1} k(\mathbf{x}, \mathbf{x}')$$

Picking a mean function

- Common choices:
  - Often pick 0 (not just in machine learning)
  - Constant, μ
  - Linear, $\mathbf{X}\beta$
  - Linear combination of bases, $\Phi(\mathbf{X})\beta$

- All equivalent to fitting a zero mean Gaussian process around the mean function
- Mean functions can help with extrapolation if the mean trend captures the trend at boundaries

What do covariance functions do?

- Looking again at the posterior mean:

$$\hat{y}(x') = \mu(\mathbf{x}') + k(\mathbf{x}, \mathbf{x}')^T \Sigma(\mathbf{x})^{-1}(y - \mu(\mathbf{x}))$$

- Can think of this as:

$$\hat{y}(x') = \mu(\mathbf{x}') + k(\mathbf{x}, \mathbf{x}')^T w(\mathbf{x})$$

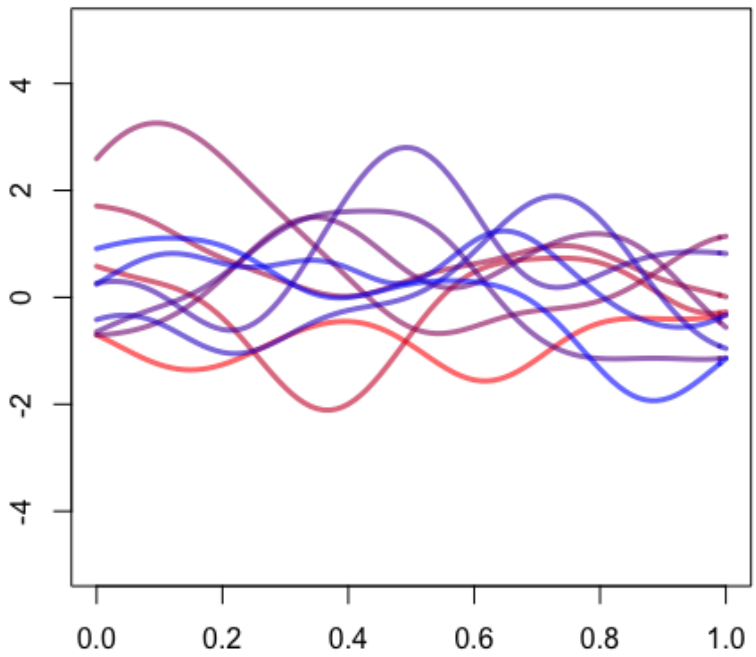
- With a stationary covariance (definition coming later):

$$\hat{y}(x') = \mu(\mathbf{x}') + k(\mathbf{x} - \mathbf{x}')^T w(\mathbf{x})$$

What are some covariance functions?

- Square exponential:

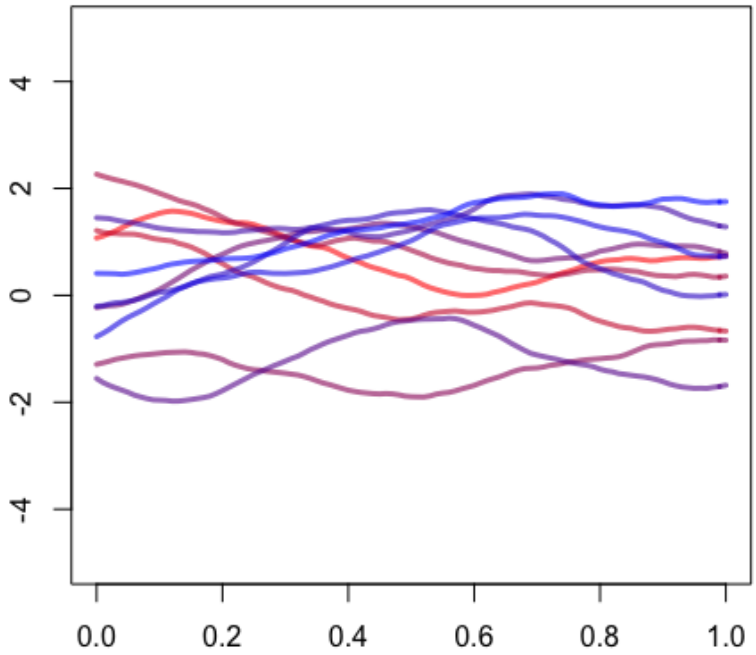
$$k(\mathbf{x}, \mathbf{x}') = \kappa \exp\left(-\frac{(\mathbf{x} - \mathbf{x}')^2}{\rho^2}\right)$$



What are some covariance functions?

- Matern(3/2):

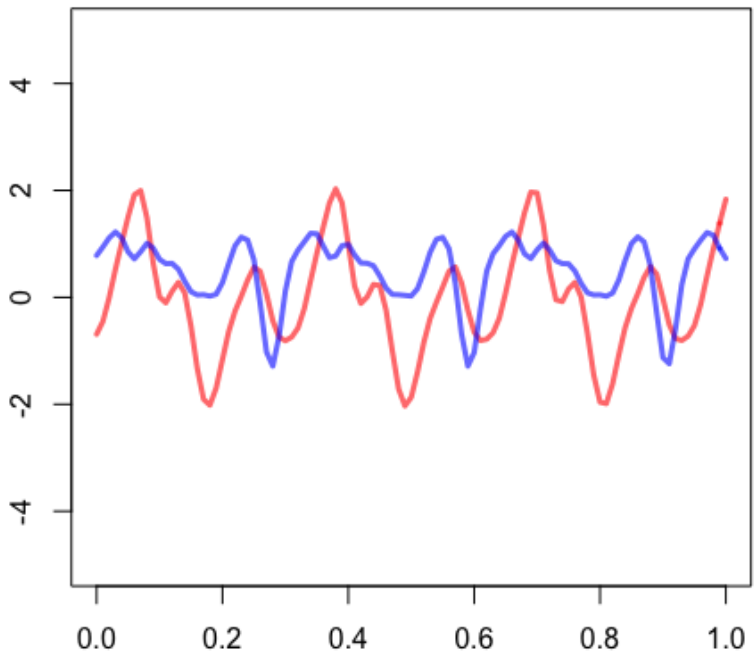
$$k(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{3}|\mathbf{x} - \mathbf{x}'|}{\rho}\right) * \exp\left(-\frac{\sqrt{3}|\mathbf{x} - \mathbf{x}'|}{\rho}\right)$$



What are some covariance functions?

- Periodic:

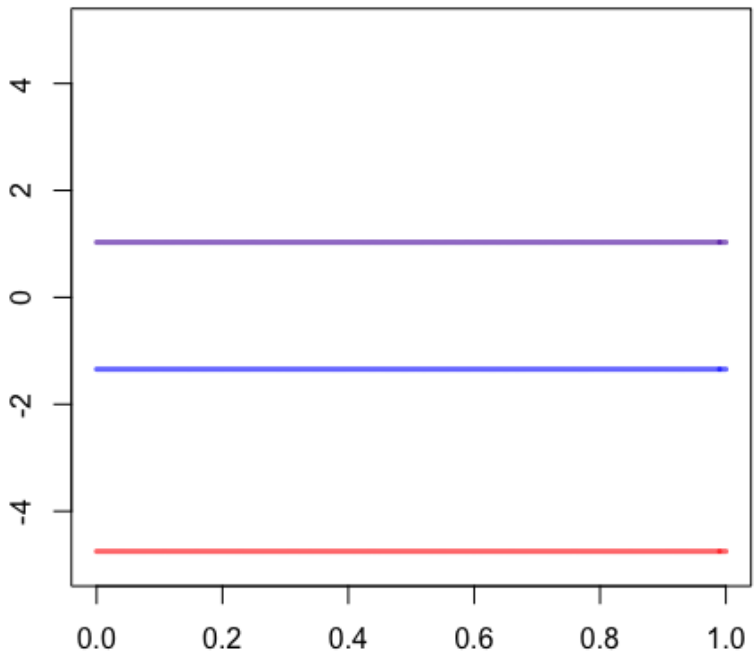
$$\kappa \exp\left(-2\rho \sin\left(\frac{2\pi(\mathbf{x}-\mathbf{x}')}{\lambda}\right)^2\right)$$



What are some covariance functions?

- Pulling the constant mean into the covariance:

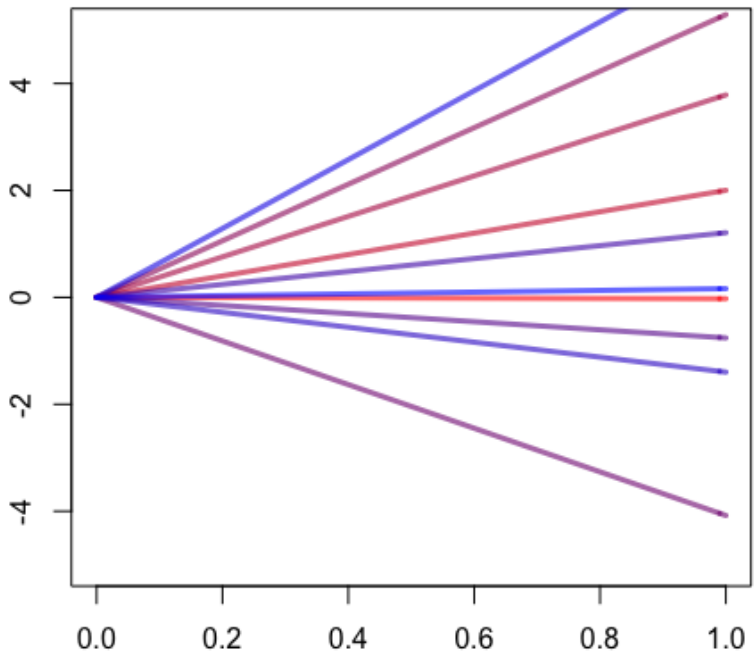
$$\mathbf{1}\Lambda\mathbf{1}^T$$



What are some covariance functions?

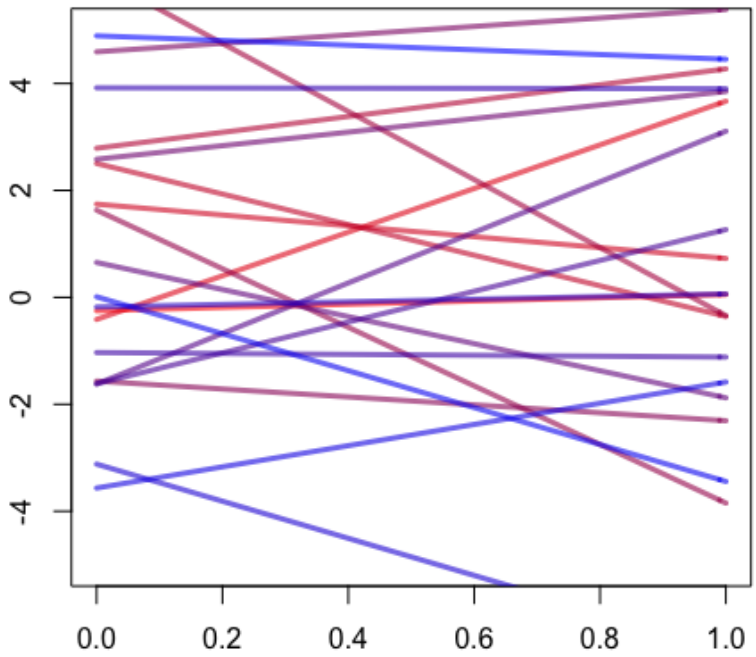
- Pulling the linear mean into the covariance:

$$\mathbf{X}\Lambda\mathbf{X}^T$$



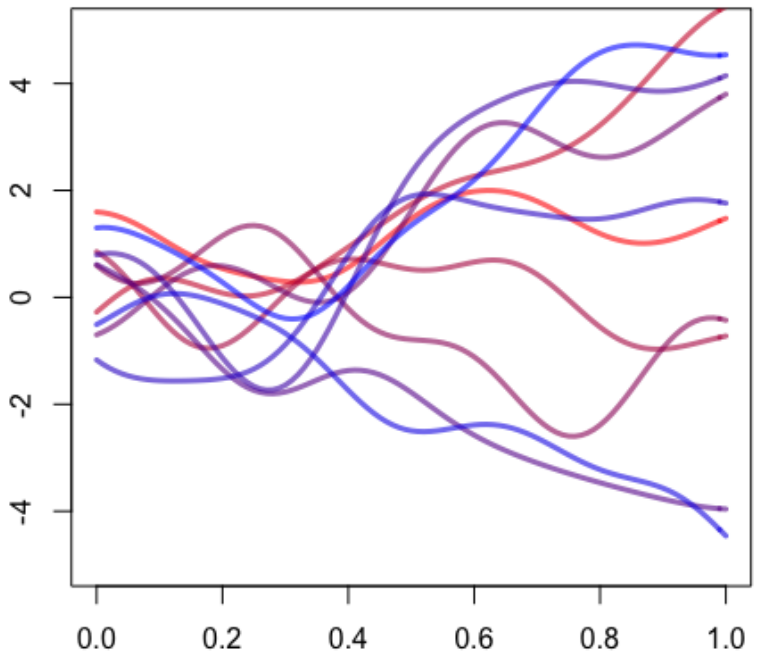
What are some covariance functions?

- Combinations of the above:
- Constant plus
- Linear



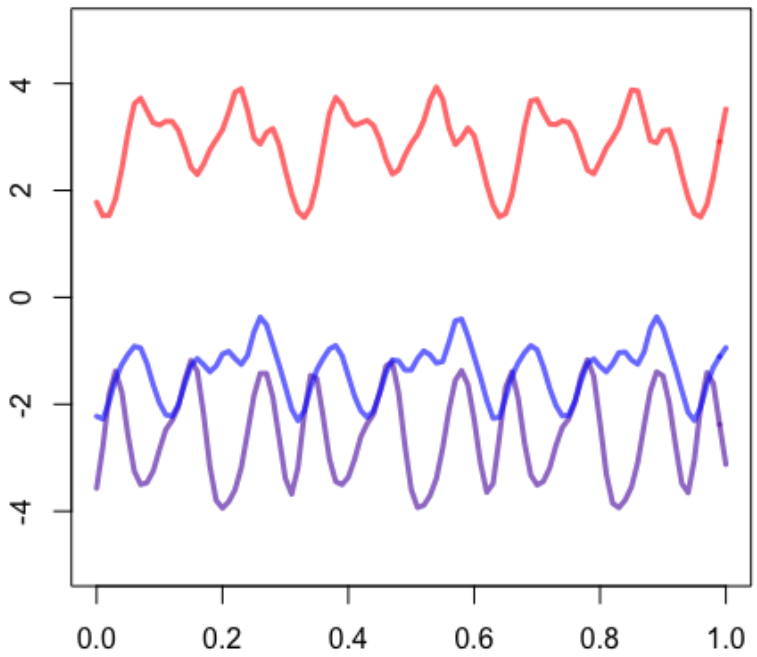
What are some covariance functions?

- Combinations of the above:
- Linear plus
- Square exponential



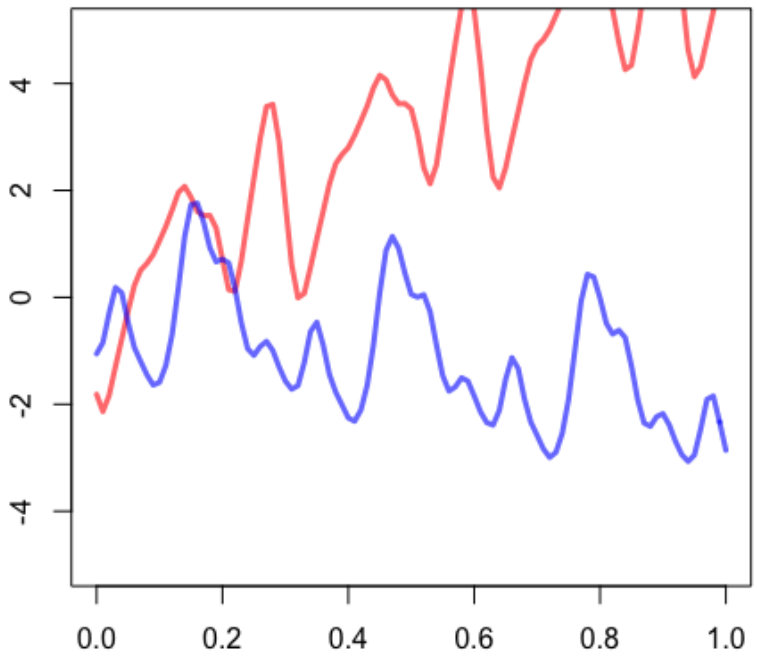
What are some covariance functions?

- Combinations of the above:
- Constant plus
- Periodic



What are some covariance functions?

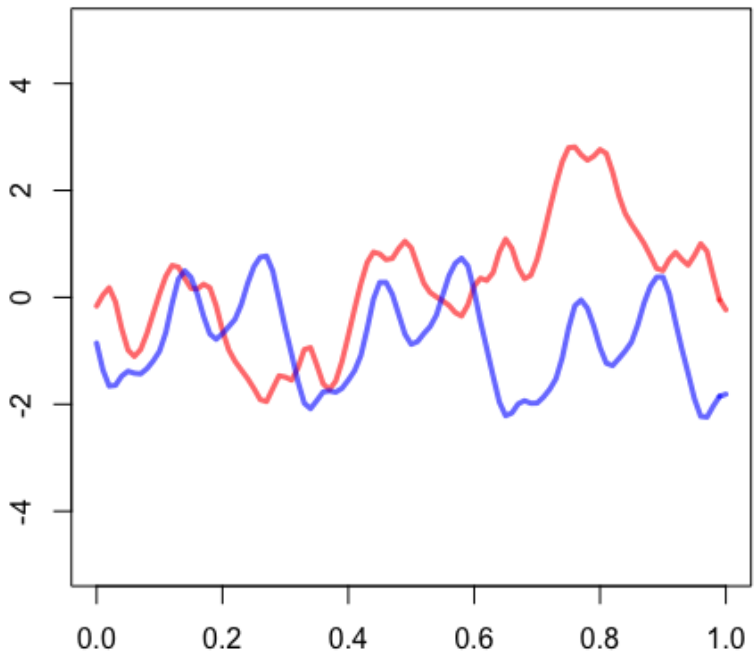
- Combinations of the above:
- Linear plus
- Periodic



What are some covariance functions?

- Combinations of the above:
- Squared Exponential plus
- Periodic

- There are many others



Choosing a covariance function

- Exponential, Matern, square-exponential.... how to choose?
  - Base it on knowledge of the simulator
  - Expected differentiability of the model
  - Data will almost certainly not be informative

- What to think about:
  - Smooth vs non-differentiable
  - Prior structure (periodicity, etc)
  - Stationarity

Choosing a covariance function: Stationarity

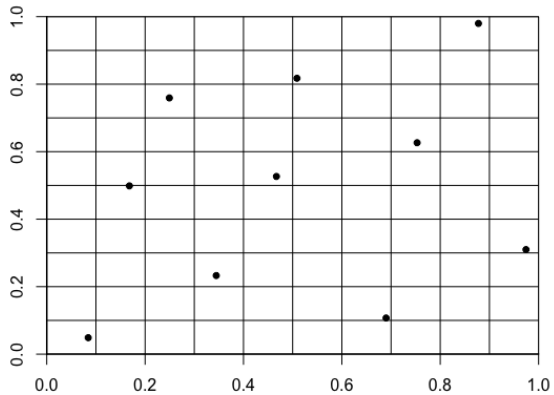
- Many common covariance functions are "stationary"

$$k(\mathbf{x}, \mathbf{x}') \rightarrow k(\mathbf{x} - \mathbf{x}')$$

- Good:
  - Reasonable chance to learn correlation parameters
- Bad:
  - Assumes variability of function is similar across parameter space
  - Strong effect on the uncertainty estimates
- Nonstationary models exist but may need lots of data
  - Process convolution (Higdon 1998)
  - Treed Gaussian processes (Gramacy 2008)

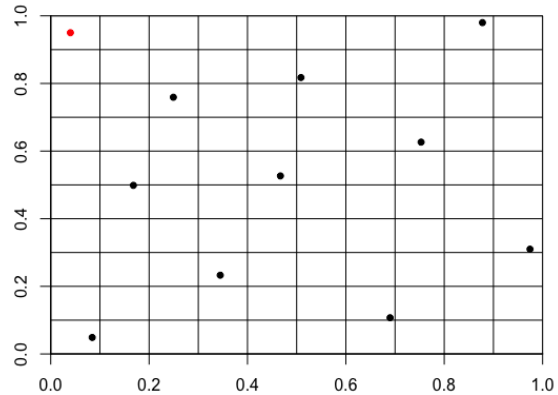
Choosing where to run the model:

- Want to approximate the function as well as possible everywhere?
  - Need to fill the space
  - Maximin Latin Hypercube
- How many locations to choose?
  - More is always better
  - Rule of thumb: 10*dims (Loeppky 2009)



Choosing where to run the model:

- Once you have some information, you can pick more deliberatively
- Sequential design
  - Pick new location to minimize IPMSE for instance
  - Pick new location to maximally reduce posterior entropy



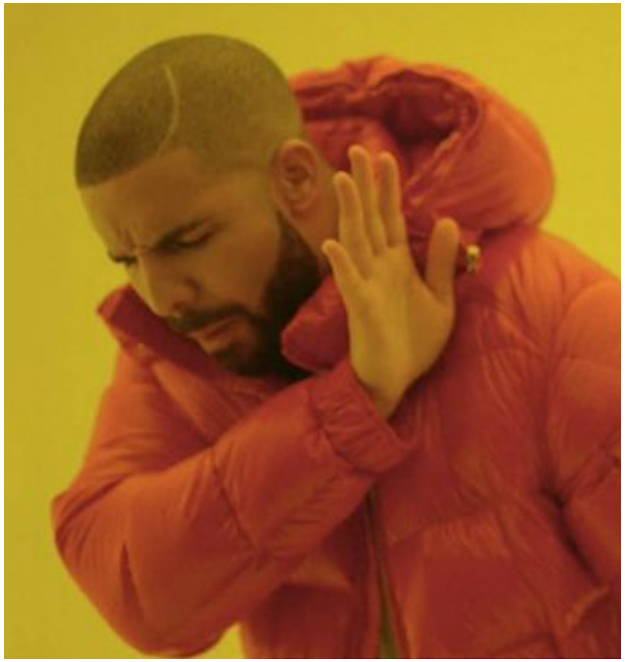
Gaussian processes are not a panacea

- Scalability with data:
  - Inverting that covariance matrix is an $\mathcal{O}(n^3)$ computation
  - When fitting covariance parameters, the matrix needs to be but each iteration/sample
  - Methods exist for sparse GPs:
    - Inducing point methods (Quiñonero-Candela 2005, Titsias 2009)
    - Local GP (Gramacy 2016)
    - Covariance structure (Kaufman 2011, Amambikasaran 2014)
- Scalability with dimension:
  - GPs are essentially local fit $\rightarrow$ need local points
  - To some extent can do well with mean prediction, uncertainty information can suffer
- Extrapolation
  - Stationary GPs are mean-reverting
  - Ability to extrapolate essentially passed to mean function

Packages in R

- mleGP
- tgp
- laGP
- GPfit
- DiceKriging

- Also Stan has GP examples in docs



Packages in Python

- George
- GPy
- PyMC3
- Scikit learn
- PyStan

What about emulating multivariate quantities of interest?

- Sometimes the model gives multidimensional quantities of interest:
  - Discrete spatio-temporal data
  - Functional outputs

- If sparsely sampled, can include functional variable as an input
- Alternatively, decompose multivariate output into bases and weights and emulate the univariate weights
- PCA is very typical but other methods have been tried

Inference in the face of model discrepancy

As much as we'd like, physical models often can't capture reality

- Kennedy-O'Hagan approach to calibration added an additive, systematic bias term:

$$y_i = f(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon$$

- or with an emulator:

$$y_i = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon$$

- $\delta(\mathbf{x})$ is another unknown function
  - Common to use another Gaussian process

Benefits of using Gaussian process for modeling discrepancy

- Goal: Inferring the form of the true response function

- Mentioned this all earlier
- Flexible to capture complex discrepancy
- Quantifies uncertainty in functional response

Drawbacks of using Gaussian process for modeling discrepancy

- Flexibility can be a curse
- Identifiability issue arises because of the flexibility of the Gaussian process
  - Brynjarsdottir (2014), Tuo (2015,2016), Plumlee (2018)

- If no restrictions on δ, any θ will do:

$$y_i = f(\mathbf{x}_i, \theta) + \delta(\mathbf{x}) + \epsilon$$

- Posterior for θ strongly depends on prior for δ

Summary and Conclusion

Conclusions

- Overviewed:
  - Calibration of computer models
  - Value of Gaussian processes
  - Model discrepancy and how it's handled

- More questions?