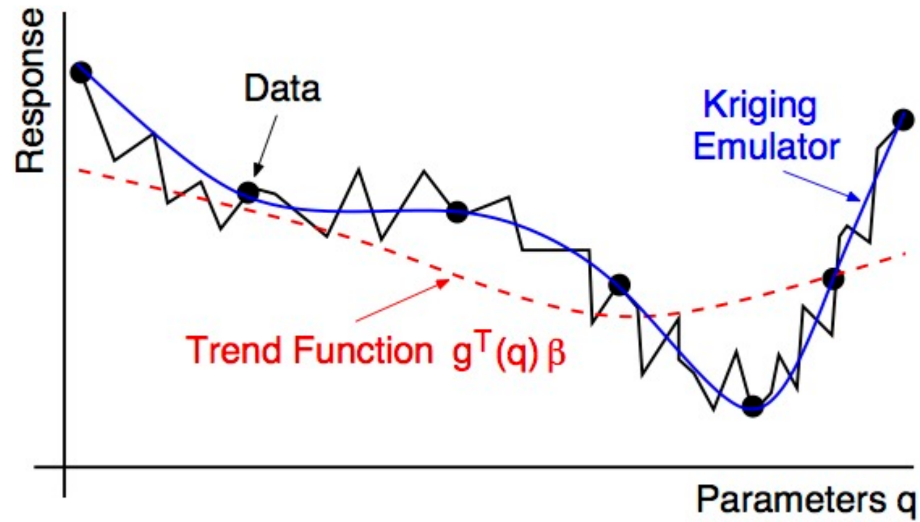
Statistical Surrogate Models



Strategy: Consider high fidelity model

$$y = f(q)$$

with M model evaluations

$$y^m = f(q^m)\ ,\ m = 1, ..., M$$

Statistical Model: $f_s(q)$: Surrogate for $f(q)$

$$y^m = f_s(q^m) + \varepsilon^m\ ,\ m = 1, ..., M$$

Numerical Surrogate: Impose *a priori* structure via basis functions $\Psi_k(q)$

Statistical Surrogate: Define mean and covariance functions that define GP; i.e., $f(q) \sim GP(m(q), c(q, q'))$ with

$$m(q) = \mathbb{E}[f(q)]$$

$$c(q, q') = \mathbb{E}[(f(q) - m(q))(f(q') - m(q'))]$$

Statistical Surrogate Models

Example: Consider M=5 training pairs

$$\{(-3.0, -1.6), (-1.5, 0.5), (-0.2, 1.2), (1.3, -1.0), (2.9, -0.4)\}$$

Take

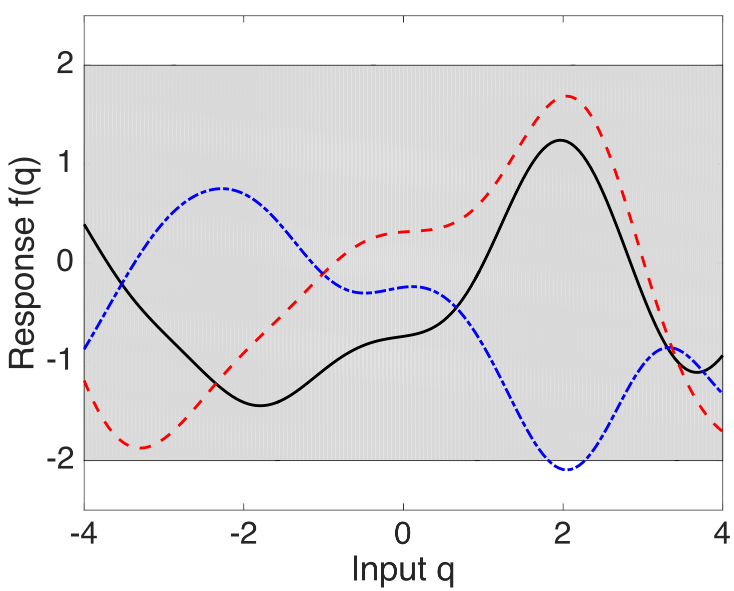
$$m(q) = 0$$

$$c(q, q') = e^{(-q-q')^2/2\ell^2}$$

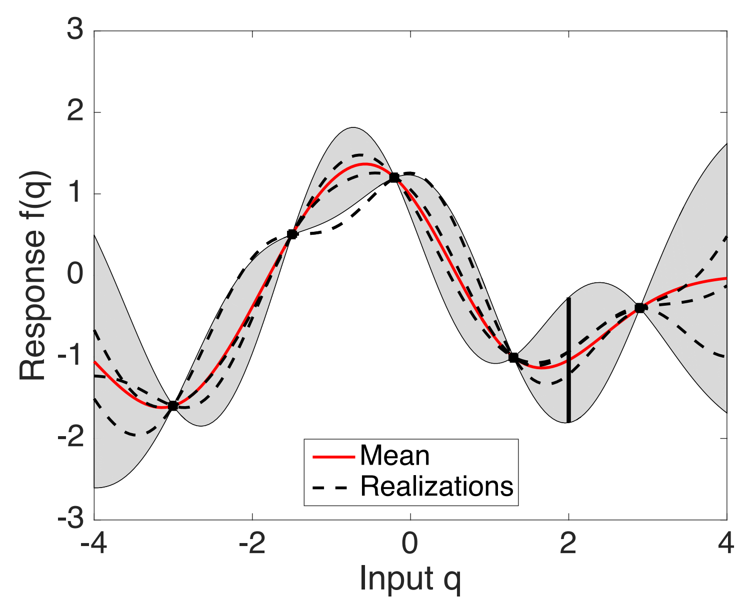
with $\ell = 1$. For $M^* = 161$ test values, form covariance matrix with components

$$[C_{**}]_{ij} = c(q^{*i}, q^{*j})$$

Result: Compare prior functions and functions conditioned on training data



$$f^* \sim \mathcal{N}(0, C_{**})$$



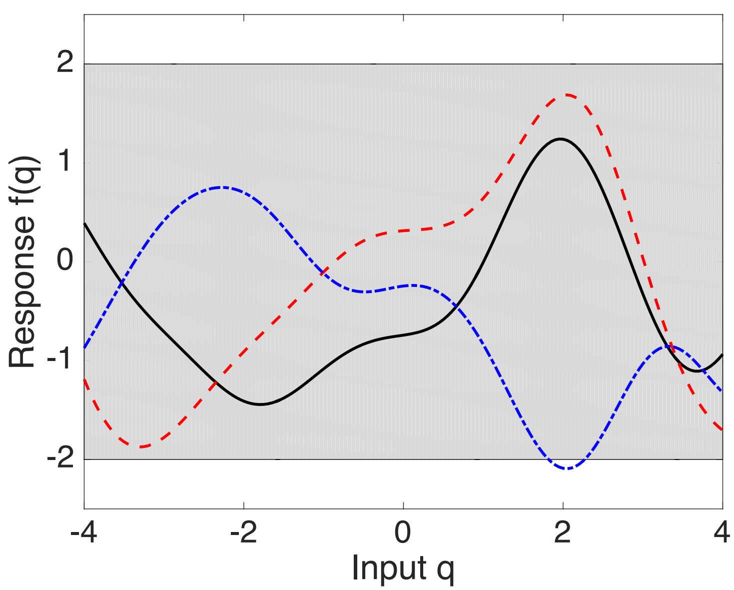
Statistical Surrogate Models

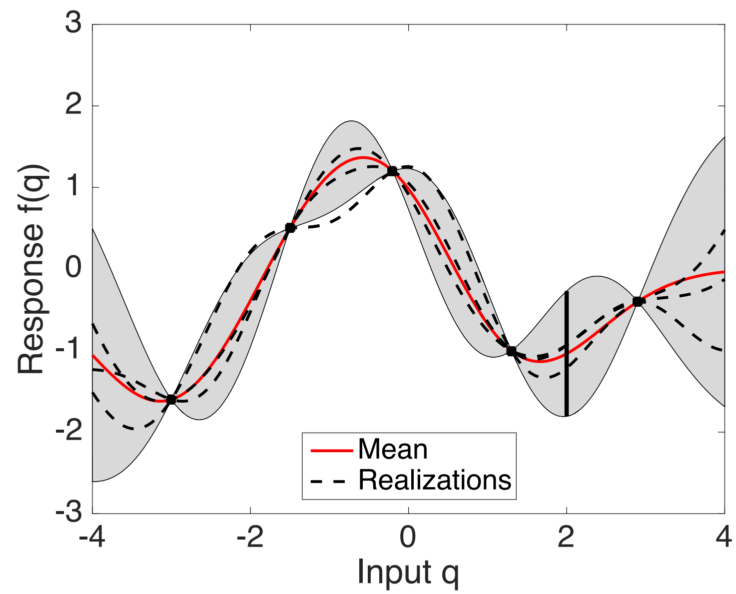
Example: Consider M=5 training pairs. Take

$$m(q) = 0$$

$$c(q,q') = e^{(-q-q')^2/2\ell^2}$$

Result: Compare prior functions and functions conditioned on training data

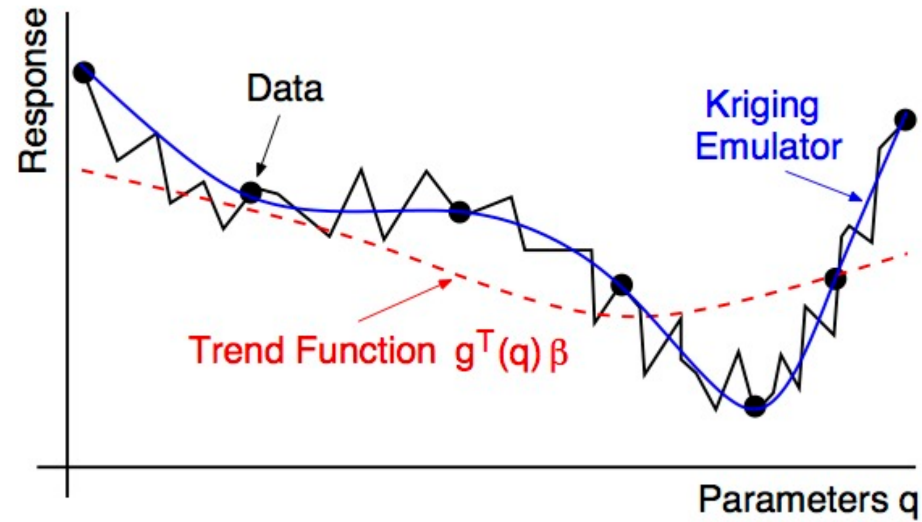




Issues:

- Specification of mean m(q) and covariance function c(q,q')
- Inference of hyperparameters in m(q) and c(q,q'); e.g., ℓ
- Construct joint prior and conditional distributions for training and future data

Gaussian Process Emulator: Mean Function m(q)



High Fidelity Model: M evaluations

$$y^m = f(q^m)\ ,\ \ m = 1, \ldots, M$$

Statistical Model:

$$y^m = f_s(q^m) + \varepsilon^m\ ,\ \ m = 1, \ldots, M$$

Gaussian Process: Kriging

$$f_s(q;\beta) = g^T(q)\beta + Z(q)$$

- $g^T(q)\beta$: Trend function

  Ordinary Kriging: $g^T(q)\beta = \beta_0$

  Universal Kriging: $g^T(q)\beta = \sum_{k=0}^{K} \beta_k g_k(q)$

Gaussian Process Emulator: Covariance Function c(q,q')

Covariance Function:

- Incorporate correlation structure between coefficients;
- Impose regularity or properties such as periodicity;
- Consider exponential and Matérn covariance functions.

Exponential Covariance Function: Anisotropic function

$$r(q,q') = \exp\left(-\sum_{k=1}^{p} \frac{1}{2\ell_k^2} |q_k - q_k'|^{\gamma_k}\right)$$

$$c(q,q') = \sigma^2 r(q,q')$$

Hyperparameters: $\sigma^2, \ell_k, \gamma_k$

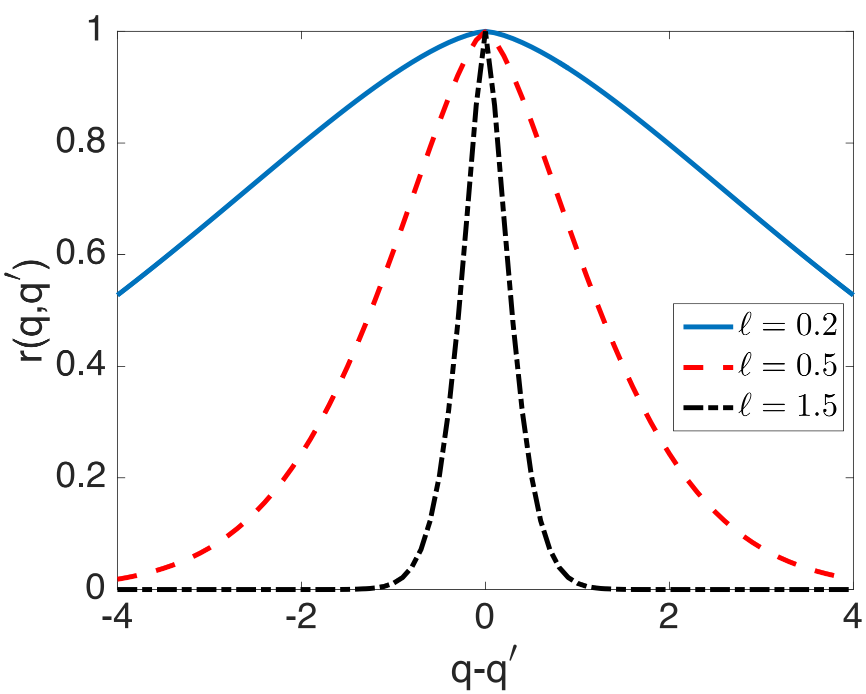
Gaussian Process Emulator: Covariance Function c(q,q')

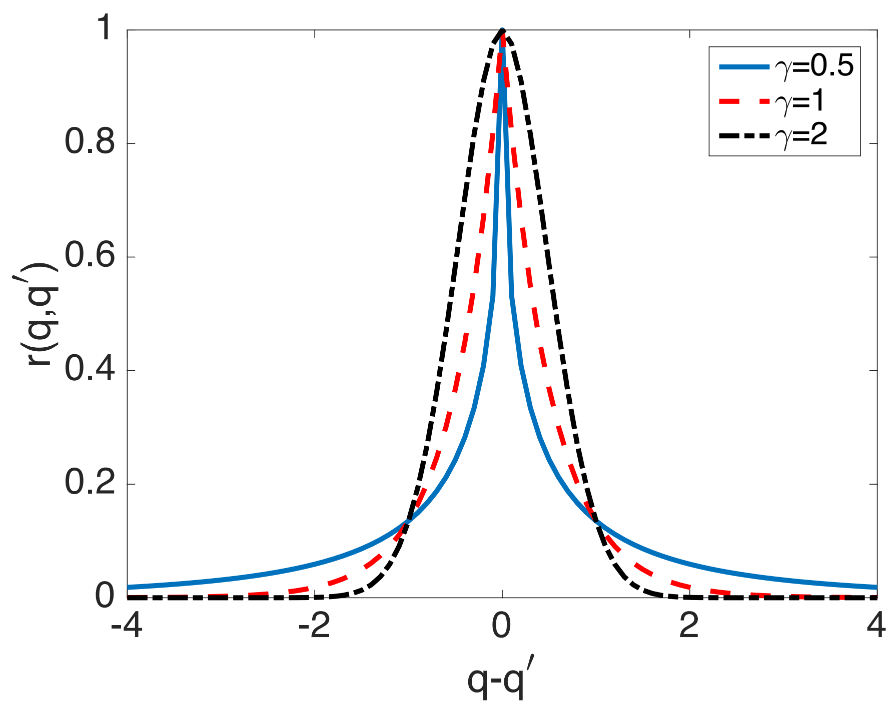
Exponential Covariance Function: Anisotropic function

$$r(q, q') = \exp\left(-\sum_{k=1}^{p} \frac{1}{2\ell_k^2} |q_k - q_k'|^{\gamma_k}\right)$$

$$c(q, q') = \sigma^2 r(q, q')$$

Hyperparameters: $\sigma^2, \ell_k, \gamma_k$





Gaussian Process Emulator: Covariance Function c(q,q')

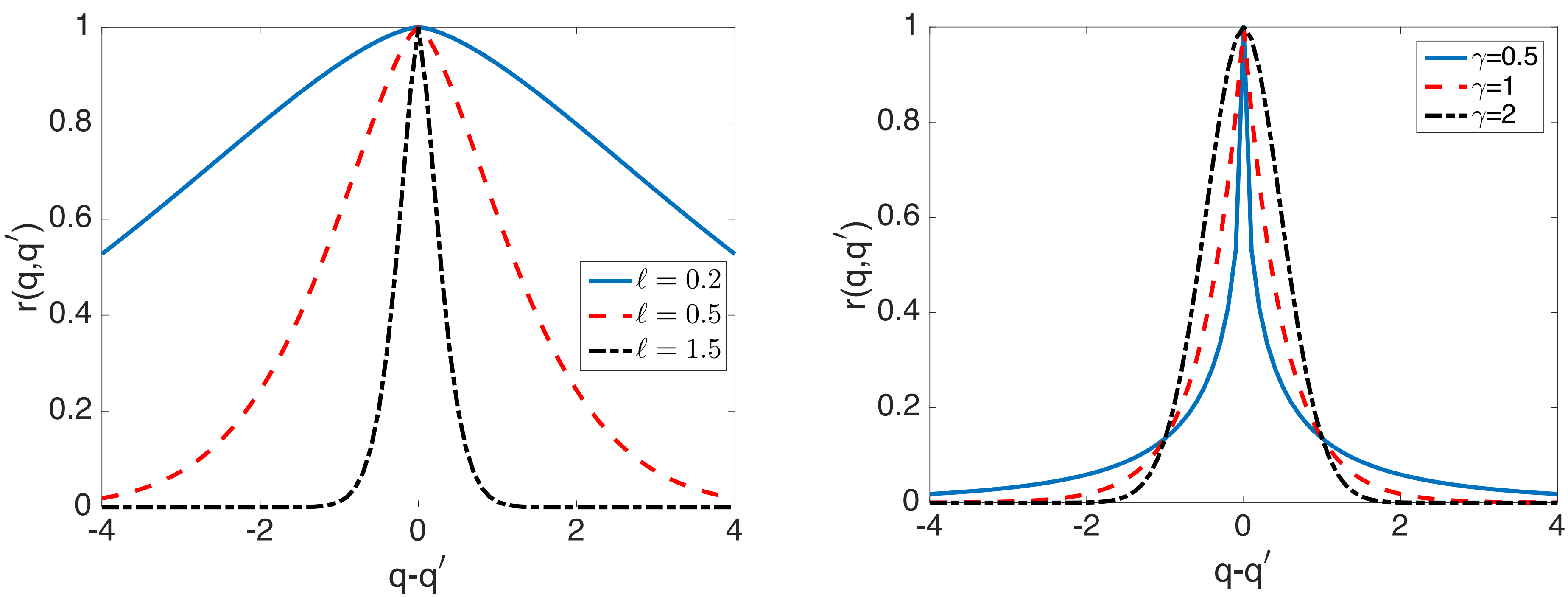
Exponential Covariance Function: Isotropic function -- Take $\ell_k = \ell, \gamma_k = 2$

$$r(q,q') = \exp\left(-\frac{1}{2\ell^2}\sum_{k=1}^{p}(q_k - q_k')^2\right)$$

or

$$r_{SE}(h) = e^{-h^2/2\ell^2}$$

where $h = \|q - q'\|$



Gaussian Process Emulator: Covariance Function c(q,q')

Matérn Covariance Function:

$$r_M(h) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}\, h}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\, h}{\ell} \right)$$

Common Choices:

$$\nu = \frac{1}{2} \quad , \quad r_{1/2}(h) = e^{-h/\ell},$$

$$\nu = \frac{3}{2} \quad , \quad r_{3/2}(h) = \left(1 + \frac{\sqrt{3}\, h}{\ell} \right) e^{-\sqrt{3}\, h/\ell},$$

$$\nu = \frac{5}{2} \quad , \quad r_{5/2}(h) = \left(1 + \frac{\sqrt{5}\, h}{\ell} + \frac{5h^2}{3\ell^2} \right) e^{-\sqrt{5}\, h/\ell}.$$

Gaussian Process Regression

Observation Model: For m = 1, … , M

$$y^m = f(q^m) + \varepsilon^m \quad , \quad \varepsilon^m \overset{iid}{\sim} \mathcal{N}(0, \sigma_0^2)$$

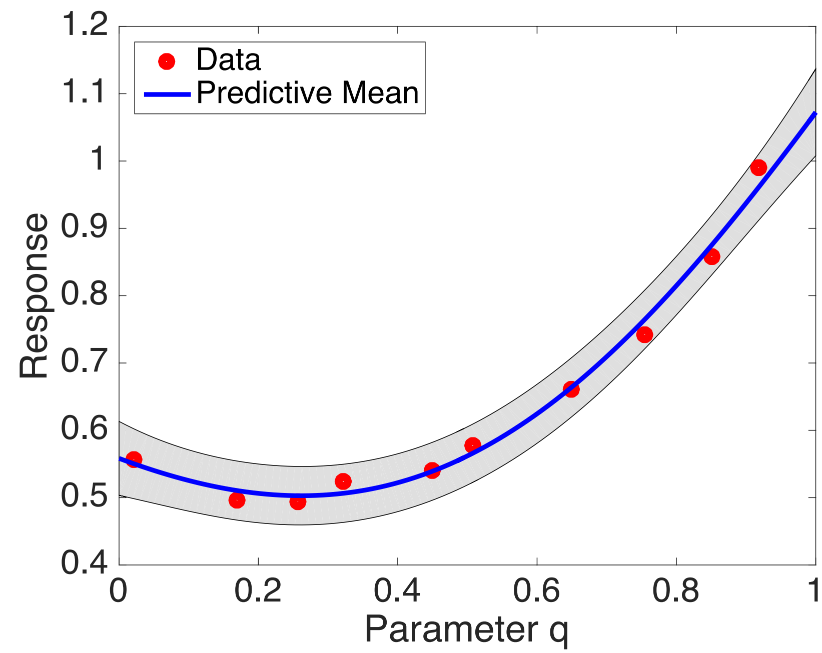
Assumption: ε^m independent from q^m

Result: Error component often termed a 'nugget'

$$y = [f(q^1), \ldots, f(q^M)] \sim \mathcal{N}(m, C + \sigma_0^2 I)$$

$$C_{ij} = c(q^i, q^j) \; , \; C = \sigma^2 R$$

Note: Nugget automatically incorporated in MATLAB GP package



Gaussian Process Emulator: Inference of Hyperparameters

Likelihood: Take $m = \beta_0 1$

$$L = \frac{1}{(2\pi)^{M/2}|C + \sigma_0^2 I|^{1/2}} \exp\left[-\frac{1}{2}(y - m)^T [C + \sigma_0^2 I]^{-1}(y - m)\right]$$

$$= \frac{1}{(2\pi\sigma^2)^{M/2}|R + \delta^2 I|^{1/2}} \exp\left[-\frac{1}{2\sigma^2}(y - \beta_0 1)^T [R + \delta^2 I]^{-1}(y - \beta_0 1)\right]$$

with $\delta^2 = \sigma_0^2/\sigma^2$

Strategy: Maximize log-likelihood

$$\mathcal{L} = -\frac{M}{2}\ln(2\pi) - \frac{M}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|R + \delta^2 I|)$$

$$\frac{1}{2\sigma^2}(y - \beta_0 1)^T [R + \delta^2 I]^{-1}(y - \beta_0 1)$$

by enforcing

$$\frac{\partial \mathcal{L}}{\partial \beta_0} = 0\,, \quad \frac{\partial \mathcal{L}}{\partial \sigma^2} = 0$$

Gaussian Process Emulator: Inference of Hyperparameters

Maximum Likelihood Estimates:

$$\beta_{0_{MLE}}(\ell, \gamma, \delta) = \frac{1^T[R(\ell,\gamma) + \delta^2 I]^{-1} y}{1^T[R(\ell,\gamma) + \delta^2 I]^{-1} 1}$$

$$\sigma^2_{MLE}(\ell, \gamma, \delta) = \frac{1}{M}\left[y - \beta_{0_{MLE}} \underline{1}\right]^T \left[R(\ell,\gamma) + \delta^2 I\right]^{-1} \left[y - \beta_{0_{MLE}} \underline{1}\right]$$

Concentrated or Profile Likelihood: Depends on ℓ, γ, δ

$$-2\mathcal{L}(\ell, \gamma, \delta) = M \ln(\sigma^2_{MLE}(\ell, \gamma, \delta)) + \ln(|R(\ell,\gamma) + \delta^2 I|)$$

Optimization:

- Simulated annealing, genetic algorithms or derivative-based algorithms;
- Inclusion of nugget improves conditioning

Gaussian Process Emulator: Inference of Hyperparameters

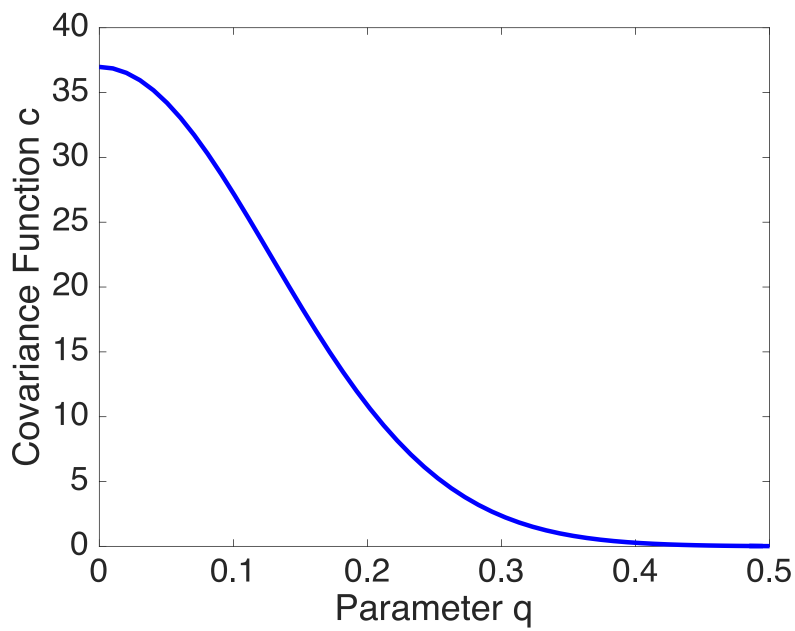
Example: Consider the construction of a covariance function

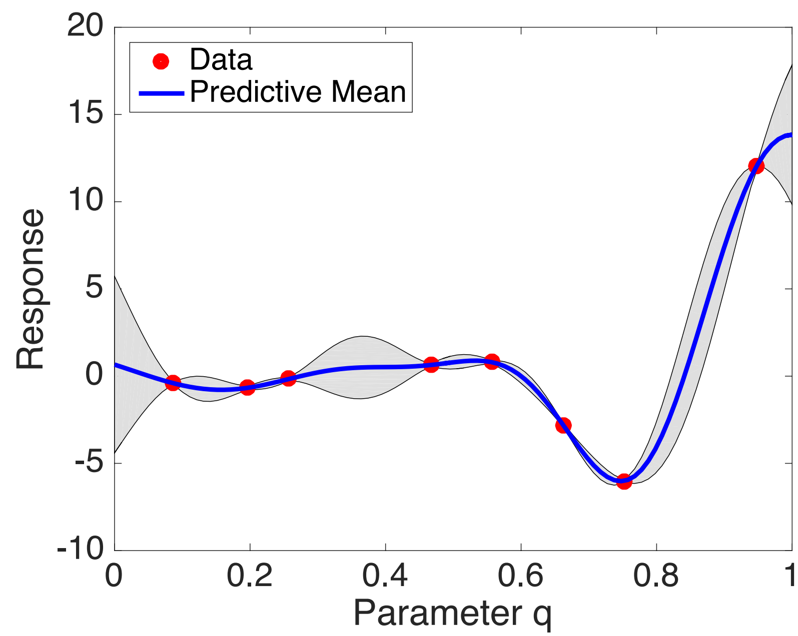
$$c(q,q') = \sigma^2 e^{-(q-q')^2/2\ell^2}$$

for the model

$$f(q) = (6q-2)^2 \sin(12q-4)$$

with $q \in [0,1]$





Question: How does one construct a predictive distribution for future observations?

Gaussian Process Predictions

Objective: Compute mean and covariance structure of future predictions

Case i: Noise-free model $y^* = f^*$

Notation:

$Q^* = [q^{*1}, ..., q^{*M^*}]$: $p \times M^*$ matrix of test input values

$f^* = [f(q^{*1}), ..., f(q^{*M^*})]$: predicted model values

$[C_*]_{ij} = c(q^i, q^{*j})$, $[C_{**}]_{ij} = c(q^{*i}, q^{*j})$: $M \times M^*$,$M^* \times M^*$ covariance matrices

$$\widetilde{\mathfrak{C}} = \begin{bmatrix} C & C_* \\ C_*^T & C_{**} \end{bmatrix}, \; M^* \text{ test points: Augmented covariance matrix}$$

Single Predicted Test Input: q^*

$$c_*(q^*) = \begin{bmatrix} c(q^1, q^*) \\ \vdots \\ c(q^M, q^*) \end{bmatrix} \qquad \widetilde{\mathfrak{C}} = \begin{bmatrix} C & c_* \\ c_*^T & c_{**} \end{bmatrix}, \; 1 \text{ test point,}$$

Gaussian Process Predictions

Example: Consider M=5 training pairs

$$\{(-3.0, -1.6), (-1.5, 0.5), (-0.2, 1.2), (1.3, -1.0), (2.9, -0.4)\}$$

Take

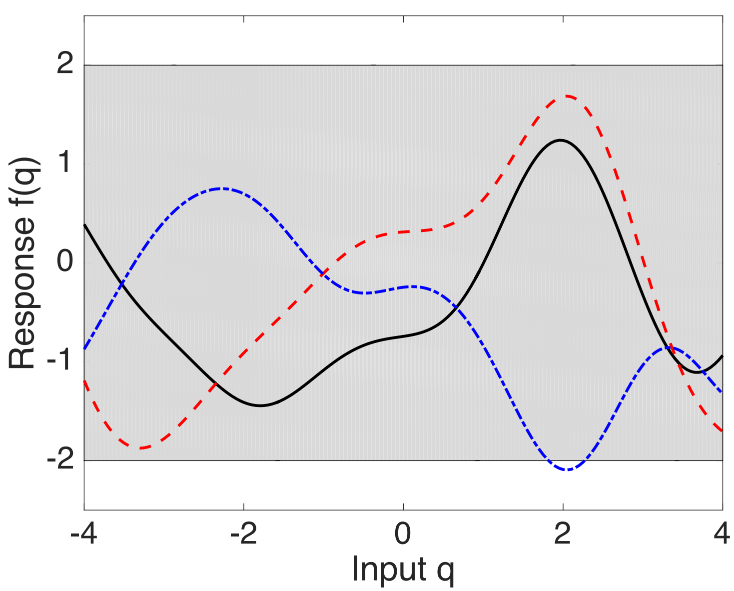
$$m(q) = 0$$

$$c(q, q') = e^{(-q-q')^2/2\ell^2}$$

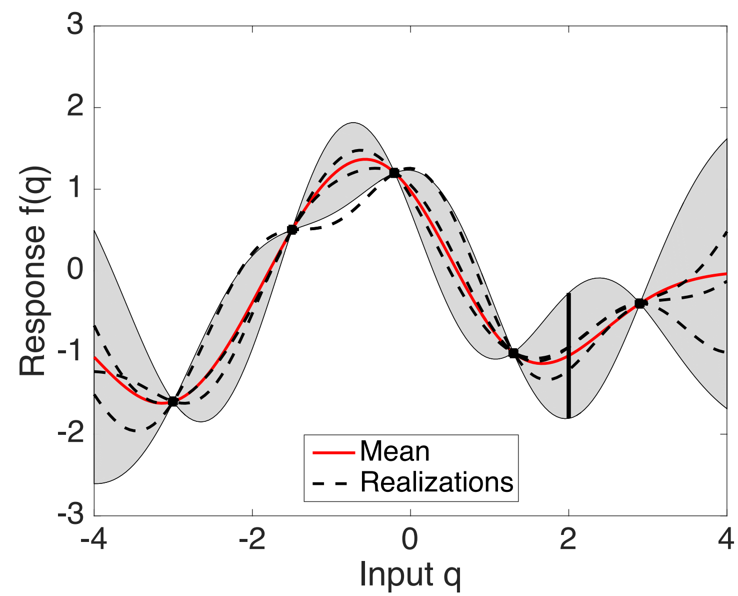
with $\ell = 1$. For $M^* = 161$ test values, form covariance matrix with components

$$[C_{**}]_{ij} = c(q^{*i}, q^{*j})$$

Result: Compare prior functions and functions conditioned on training data



$$f^* \sim \mathcal{N}(0, C_{**})$$



Gaussian Process Predictions: Bayesian Framework

Strategy: Generate

$$f^* \sim \mathcal{N}(\beta_0 1, C_{**})$$

from prior distribution. We then need to condition joint prior distribution between training responses and predictions

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \beta_0 1 \\ \beta_0 1^* \end{bmatrix}, \begin{bmatrix} C & C_* \\ C_*^T & C_{**} \end{bmatrix} \right)$$

to contain only functions that agree with training data

$$\{Q = [q^1, ..., q^M] \, , \; y = [y^1, ..., y^M]\}$$

Conditional Predictive Distribution:

$$f^* | Q, Q^*, y \sim \mathcal{N}\left[\beta_0 1^* + C_*^T C^{-1}(y - \beta_0 1), C_{**} - C_*^T C^{-1} C_*\right]$$

Note:

$$\mathbb{E}[f^*] = \beta_0 1_{M^*} + C_*^T C^{-1}(\mathbf{y} - \beta_0 1)$$

$$\text{cov}[f^*] = C_{**} - C_*^T C^{-1} C_*$$

Gaussian Process Predictions: Bayesian Framework

Note: Consider observations $f^* = f(q^*)$ with $q^* = q^j$ The C_* reduces to

$$c_*^T(q^j) = \left[c(q^1, q^j), \ldots, c(q^M, q^j)\right]$$

for each training input q^j. It then follows that

$$c_*^T(q^j) C^{-1} \left(y - \beta_0 1\right) = y^j - \beta_0$$

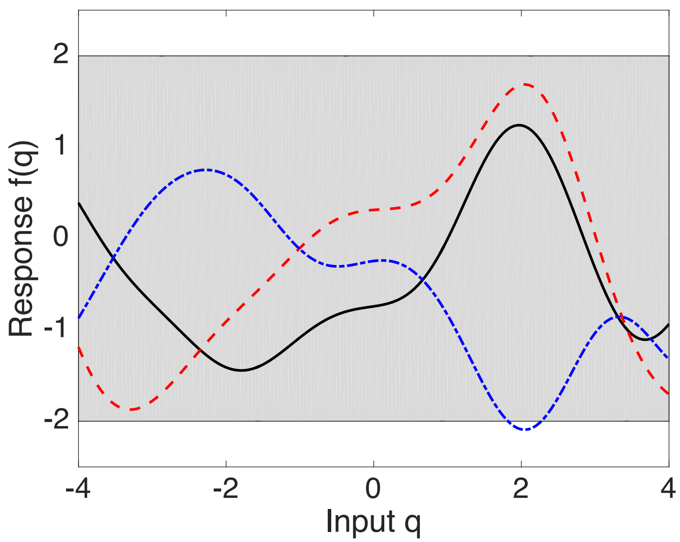
so that $\mathbb{E}[f(q^j)] = y^j$

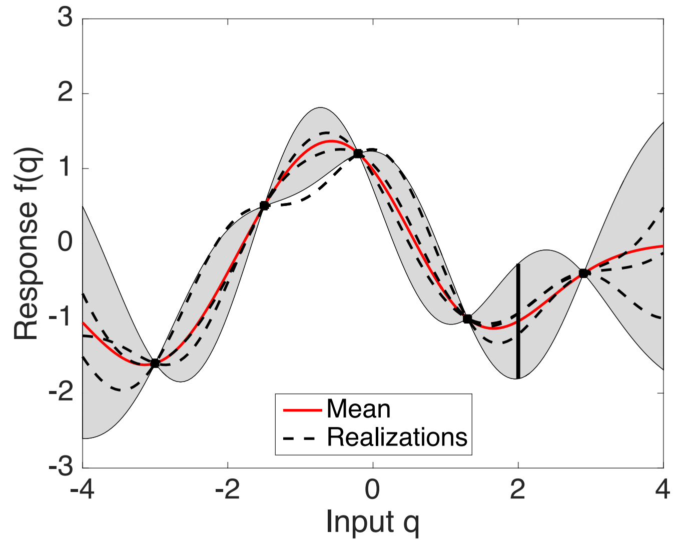
Result: Mean interpolates training data for noise-free model

Gaussian Process Predictions: Bayesian Framework

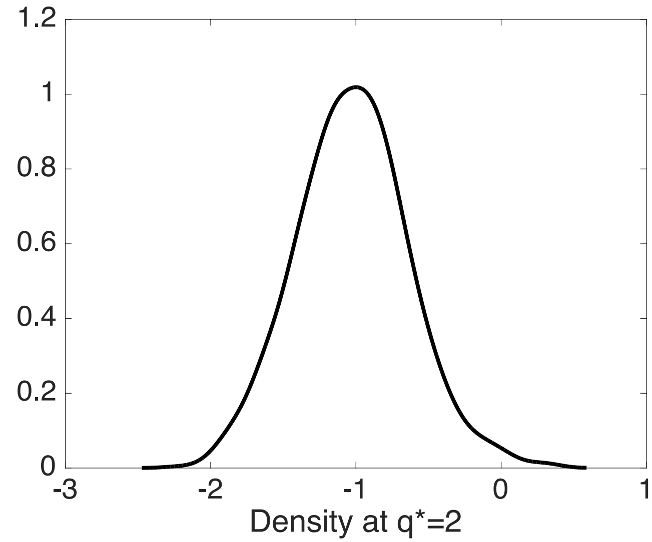
Example: Consider M=5 training pairs

$$\{(-3.0,-1.6),(-1.5,0.5),(-0.2,1.2),(1.3,-1.0),(2.9,-0.4)\}$$





Note: Similar analysis for regression using a nugget.



Gaussian Process Predictions: Bayesian Framework

Example: Consider the modified Branin function

$$f(q) = \left[q_2 - \frac{5.1}{4\pi^2}q_1^2 + \frac{5}{\pi}q_1 - 6\right]^2 + 10\left[\left(1 - \frac{1}{8\pi}\right)\cos q_1 + 1\right] + 5q_1$$

where $q_1 \in [-5, 10]$ and $q_2 \in [0, 15]$.

Anisotropic Covariance Function:

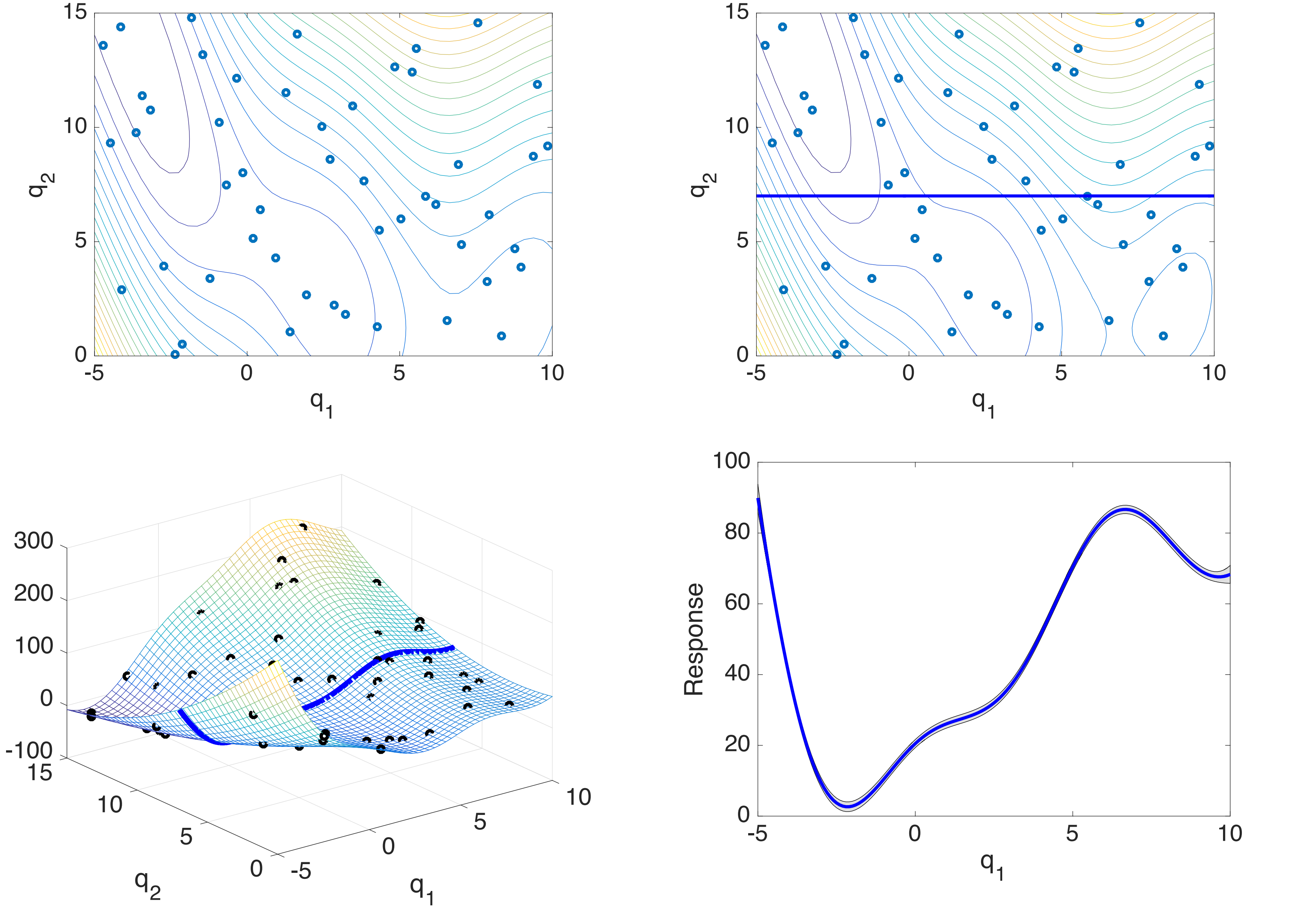
$$c(q, q') = \sigma^2 \exp\left[-\frac{1}{2}\sum_{i=1}^{2}\frac{(q_i - q_i')^2}{\ell_i^2}\right]$$

Hyperparameters: e.g., $\ell_1 = 4.21, \ell_2 = 23.66$

Note: Employed Latin hypercube samples (LHS) for training

Gaussian Process Predictions: Bayesian Framework

Example: Modified Branin function



Gaussian Process Predictions: Frequentist Framework

Strategy: Consider

$$\widetilde{y} = \left[\begin{array}{c} y \\ y^* \end{array} \right]$$

Log-Likelihood for Augmented System:

$$\mathcal{L} = -\frac{M}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathfrak{C}| - \frac{1}{2}(\widetilde{y} - \beta_0\widetilde{1})^T\mathfrak{C}^{-1}(\widetilde{y} - \beta_0\widetilde{1})$$

Note:

$$\begin{aligned} \widetilde{\mathfrak{C}}^{-1} &= \left[\begin{array}{cc} C^{-1} + C^{-1}c_*\left(1 - c_*^T C^{-1} c_*\right)^{-1} c_*^T C^{-1} & -C^{-1}c_*\left(1 - c_*^T C^{-1} c_*\right)^{-1} \\ -\left(c_{**} - c_*^T C^{-1} c_*\right)^{-1} c_*^T C^{-1} & \left(c_{**} - c_*^T C^{-1} c_*\right)^{-1} \end{array} \right] \\ &= \left[\begin{array}{cc} C_{11}^{-1} & C_{12}^{-1} \\ C_{21}^{-1} & C_{22}^{-1} \end{array} \right] \end{aligned}$$

Gaussian Process Predictions: Frequentist Framework

Note:

$$(\widetilde{y} - \beta_0 \widetilde{1})^T \widetilde{\mathfrak{C}}^{-1} (\widetilde{y} - \beta_0 \widetilde{1}) = \begin{bmatrix} y - \beta_0 \mathbf{1} \\ y^* - \beta_0 \end{bmatrix}^T \begin{bmatrix} C_{11}^{-1} & C_{12}^{-1} \\ C_{21}^{-1} & C_{22}^{-1} \end{bmatrix} \begin{bmatrix} y - \beta_0 \mathbf{1} \\ y^* - \beta_0 \end{bmatrix}$$

$$= F(y) + (y^* - \beta_0) C_{21}^{-1} (y - \beta_0 \mathbf{1}) + (y - \beta_0 \mathbf{1})^T C_{12}^{-1} (y^* - \beta_0)$$
$$+ (y^* - \beta_0)^2 C_{22}^{-1}$$

$$= F(y) + \left[\frac{-2 c_*^T C^{-1} (y - \beta_0 \mathbf{1})}{c_{**} - c_*^T C^{-1} c_*} (y^* - \beta_0) + \frac{1}{c_{**} - c_*^T C^{-1} c_*} (y^* - \beta_0) \right]$$

Necessary Condition:

$$\frac{\partial \mathcal{L}}{\partial y^*} = 0$$

yields

$$\frac{-1}{1 - c_*^T C^{-1} c_*} (y^* - \beta_0) + \frac{c_*^T C^{-1} (y - \beta_0 \mathbf{1})}{1 - c_*^T C^{-1} c_*} = 0$$

Gaussian Process Predictions: Frequentist Framework

Necessary Condition:

$$\frac{\partial \mathcal{L}}{\partial y^*} = 0$$

yields

$$\frac{-1}{1 - c_*^T C^{-1} c_*}(y^* - \beta_0) + \frac{c_*^T C^{-1}(y - \beta_0 \mathbf{1})}{1 - c_*^T C^{-1} c_*} = 0$$

Result: Same as for Bayesian framework

$$y^* = \mathbb{E}[f(q^*)] = \beta_0 + c_*^T C^{-1}(y - \beta_0 \mathbf{1})$$

Note:

$$\begin{aligned} \mathbb{E}[f^*] &= \beta_0 + c_*^T C^{-1}(y - \beta_0 \underline{1}) \\ &= \sum_{k=1}^{K} u_k \Psi_k(q^*) + \beta_0 \end{aligned}$$

Basis Functions:

$$\Psi_k(q^*) = c(q^k, q^*)$$

Weights:

$$u = C^{-1}(y - \beta_0 1)$$

Example: Modeling of Volcanic Pyroclastic Flows

Authors: Bayarri, Berger, Calder Dalbey, Lunagomez, Patra, Pitman, Spiller, Wolpert; *Technometrics,* 51(4), 2009; Gu and Berger, *The Annals of Applied Statistics*, 2016.

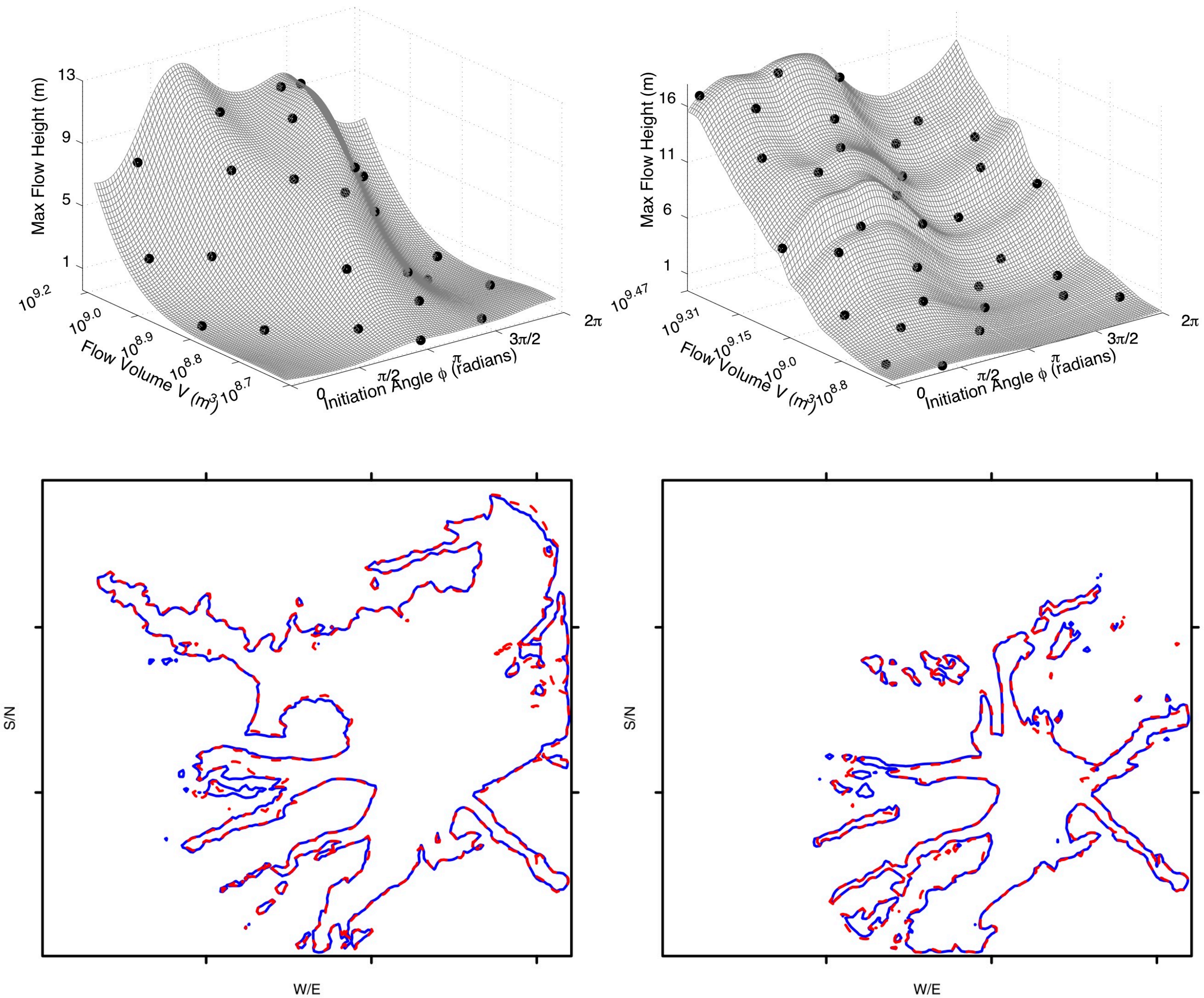
Objectives:

- Employ simulation models and surrogates to assess risk of *rare* catastrophic events; e.g., volcanic eruption.
- Employed TITAN2D to simulate flows.
- Test Case: Soufrière Hills Volcano on Island of Montserrat.
- Use emulator to identify threshold inputs – e.g., critical flow depth – that define catastrophic event.
- Compared GP and mathematical surrogates; GP advantageous for this application.

Example: Modeling of Volcanic Pyroclastic Flows

Objectives:

- Use emulator to identify threshold inputs – e.g., critical flow depth – that define catastrophic event. Employed TITAN2D and GP surrogates.



TITAN2D

GP

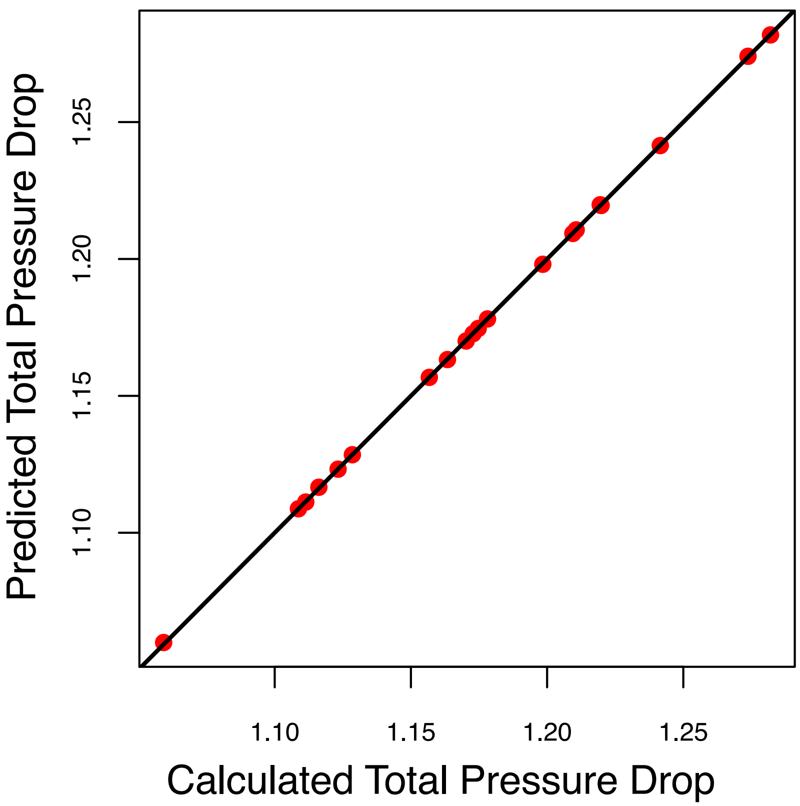
Surrogate Construction: Nuclear Power Plant Design

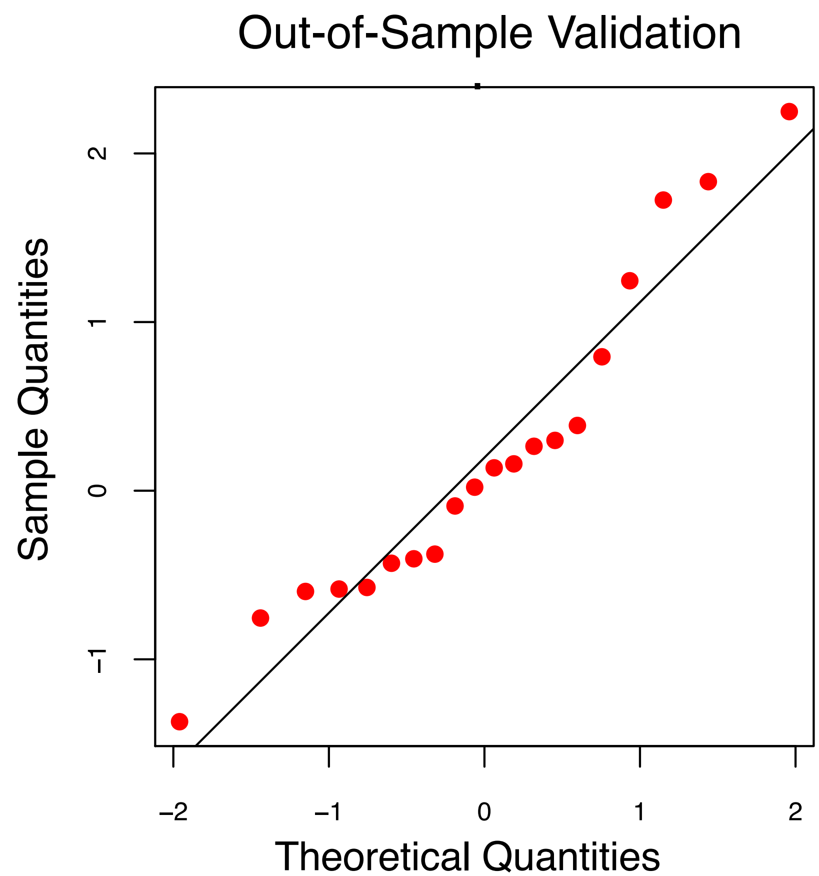
Subchannel Code (COBRA-TF): 33 VUQ parameters reduced to 5 using SA

Surrogate: Total pressure drop

• GP emulator constructed using 50 COBRA-TF runs perturbing 5 active inputs.

• Use remaining computational budget to evaluate quality of surrogate using post-processed Dakota outputs.



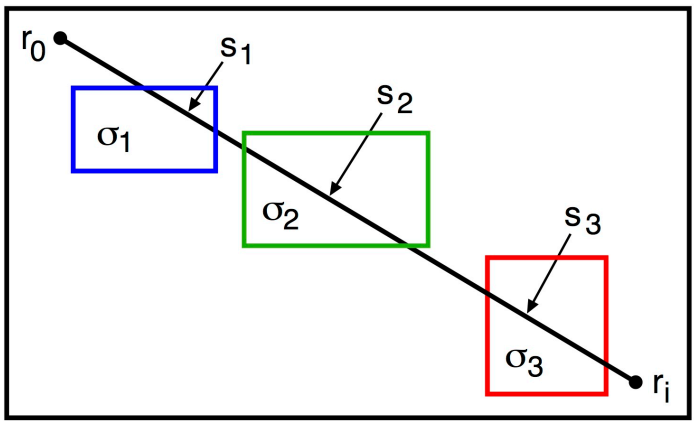


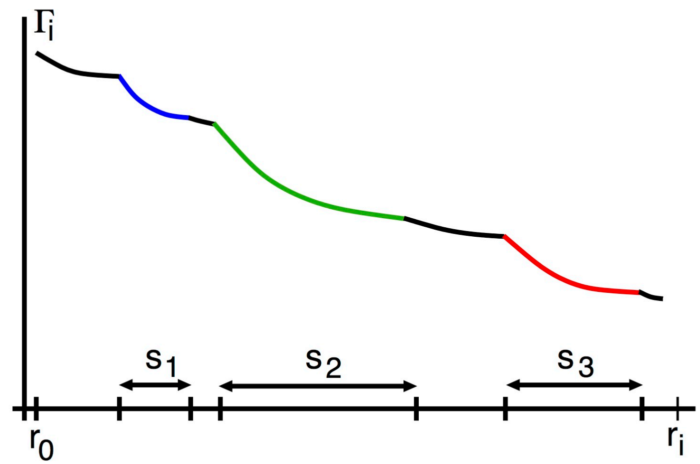
Gaussian Process Emulators

Example: Radiation source localization in an urban environment

Observation Model:

$$y_i(q) = F_i^{MCNP}(q) + \mathbb{E}[B_i \Delta t] \;,\; i = 1, \ldots, N_d,$$





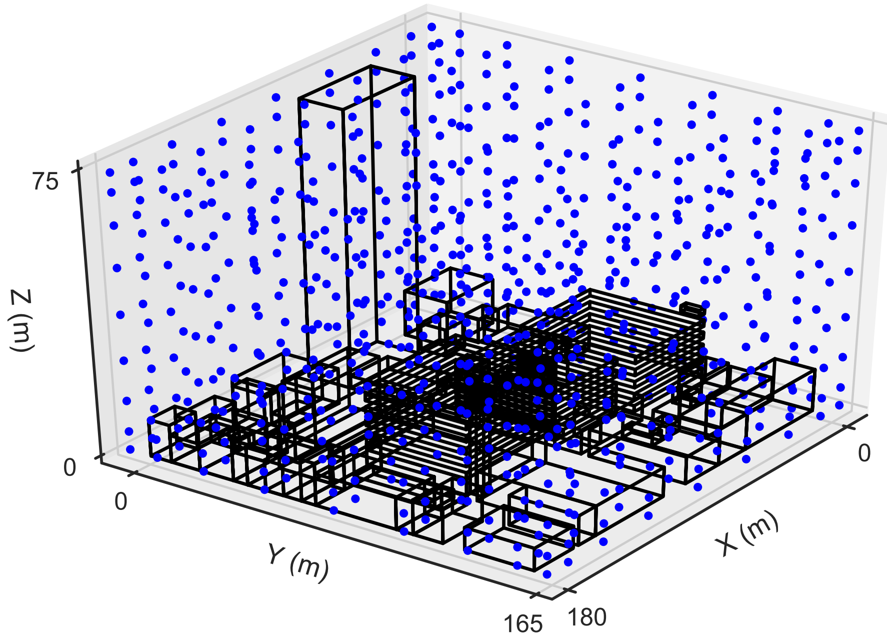
Root Mean Square Error:

$$nRMSE = \frac{1}{N_d} \sum_{i=1}^{N_d} nRMSE_i$$

$$nRMSE_i = \frac{\sqrt{\frac{1}{M^*} \sum_{j=1}^{M^*} \left[F_i^{MCNP}(q^j) - F_i^{SM}(q^j) \right]^2}}{\frac{1}{M^*} \sum_{j=1}^{M^*} F_i^{MCNP}(q^j)}$$

Gaussian Process Emulators

Example: Radiation source localization in an urban environment



Results: Gaussian process (GP) versus neural net (NN)

	Surrogate	nRMSE	Training Time (s)	Eval Time (s)
Training Set 1	GP	5.14	13.1	0.052
	NN	5.17	63.4	0.049
Training Set 2	GP	2.23	915.1	0.267
	NN	3.34	198.7	0.047

Gaussian Process Predictions

References:

- C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006
- A.I.J. Forrester, A.Sóbester and A.J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, Progress in Astronautics and Aeronautics Volume 26, John Wiley and Sons, Chickester, UK, 2007