Module 2: Gaussian Process Models

Jerome Sacks and William J. Welch

National Institute of Statistical Sciences and University of British Columbia

Adapted from materials prepared by Jerry Sacks and Will Welch for various short courses

Acadia/SFU/UBC Course on Dynamic Computer Experiments
September–December 2014

Outline of Topics

Approximation of Computer Codes

Recall

- d-dimensional input: $\mathbf{x} = x_1, \ldots, x_d$
- Deterministic output: $y(\mathbf{x})$

Approximation / prediction / emulation of $y(\mathbf{x})$ is the "engine" of analysis of computer experiments:

- To replace the computer model in future with a fast surrogate
- Sensitivity analysis
- Visualization
- Optimization
- Assessment of reality of the computer model
- ...

We will use a Gaussian process (GP) model for all of the above

What's the Problem?

- Have data $\{\mathbf{x}^{(i)}, y(\mathbf{x}^{(i)})\}$ for $i = 1, \ldots, n$ from running the code.
- Want to predict $y(\mathbf{x})$ at a new $\mathbf{x}$, a standard statistical question; also standard function approximation (no error).
- Don't know much about the function $y(\mathbf{x})$, and if we specify a class (like cubic splines) we need lots of data because of high dimensions.

Our Strategy

- Before collecting data (making computer runs) we have a vague idea of y's properties and so think of y as random.
  - Example: 1-dimensional x on $[0, 1]$; $y(0)$ and $y(1)$ uniform on $[0, 1]$; $y(0)$ and $y(1)$ should be "similar".
- Our prior belief or uncertainty about $y(\mathbf{x})$ is measured by a probability distribution.
- Collect data.
- Now update belief/uncertainty through the conditional distribution of $y(\mathbf{x})$ given the data. In particular, predict $y(\mathbf{x})$ at a new $\mathbf{x}$ as $E[y(\mathbf{x}) \,|\, \text{data}]$.
- Conceptually: prior uncertainty + data $\Rightarrow$ updated (posterior) uncertainty (the Bayesian Paradigm)

Rationale and Technical Needs

- Why is this strategy useful?
  - Lets the data do the talking
  - Copes with data scarcity
  - It works (as we'll see)
  - Has built in uncertainty measures
- What needs to be clarified?
  - Notion of Gaussian process
  - Prior distribution of $y(\mathbf{x})$
  - Computation of posterior distribution

What is a Gaussian Process (GP)?

- A (deterministic) function $y(\mathbf{x})$ coded in a computer model is a "table" $\{\mathbf{x}, y(\mathbf{x})\}$.
- Graph the function by plotting the points of the table. (Plot $y(\mathbf{x})$ at a large number, N, of points, $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ — a scatter diagram — and "connect the dots".)
- Suppose these values are the outcome of a random draw from some joint Gaussian distribution of random variables $y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})$ and plot as above. We will get a realization of a Gaussian process (GP).
- (A new random draw will generate a different function; hence another name, random function statistical model.)
- Alternatively, think of the distribution of $y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})$ as a prior distribution for the function values $y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})$.

The Prior

- We will abuse notation and think of $y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})$ at any N points as random.
- We will work solely with the multivariate normal (MVN) distribution for the $y(\mathbf{x}^{(i)})$.
- Each $y(\mathbf{x}^{(i)})$ has mean μ (can easily be generalized to μ varying according to a regression function)
- The covariance matrix is $\sigma^2\mathbf{R}$ where the correlation matrix

$$\mathbf{R} = \mathrm{Cor}(y(\mathbf{x}^{(i)}), y(\mathbf{x}^{(j)})) \quad (N \times N \text{ matrix})$$

  is specified and absolutely critical to the GP approach.
- Summary: $\mathbf{y} = (y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)}))^T$ is $\mathrm{MVN}(\mu\mathbf{1}, \sigma^2\mathbf{R})$, i.e., has density

$$\frac{1}{(2\pi\sigma^2)^{N/2}(\det\mathbf{R})^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mu\mathbf{1})^T\mathbf{R}^{-1}(\mathbf{y}-\mu\mathbf{1})\right),$$

  where $\mathbf{1}$ is an $N \times 1$ vector of 1's.

Example Correlation Functions in One or More Dimensions

The squared-exponential (Gaussian) correlation function is a popular choice.

Let $\mathbf{x}$ and $\mathbf{x}'$ be two sets of values for the input variables.

For $\theta > 0$:

- 1 dimension, $\mathbf{x} = x$

$$\mathrm{Cor}(y(x), y(x')) \equiv R(x, x') = \exp(-\theta|x - x'|^2)$$

  and

$$\mathbf{R} = [\exp(-\theta|x^{(i)} - x^{(j)}|^2)]$$

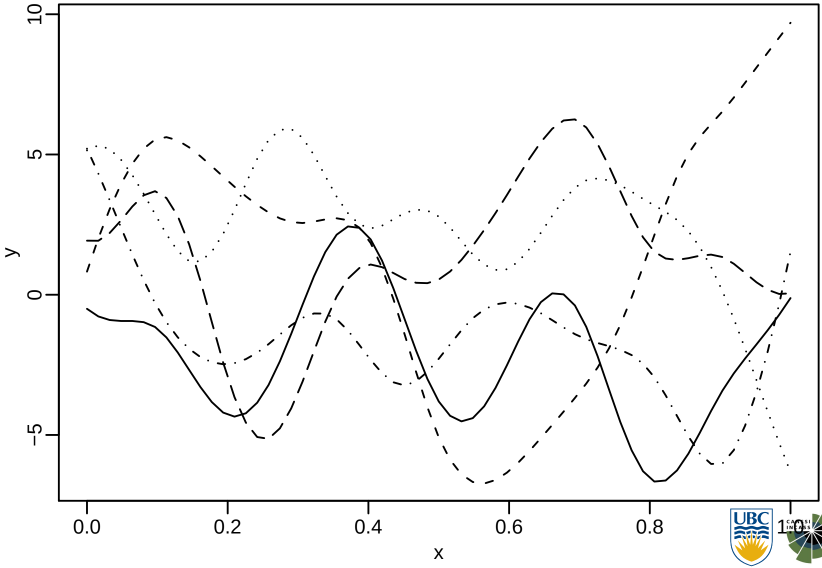
- 2 dimensions, $\mathbf{x} = (x_1, x_2)$

$$R(\mathbf{x}, \mathbf{x}') = \exp(-\theta_1|x_1 - x_1'|^2) \times \exp(-\theta_2|x_2 - x_2'|^2)$$

Simulating Realizations of $y(\mathbf{x})$

- Simulating from a MVN is straightforward (see Appendix A)
- In the next slide are 5 realizations of a Gaussian process with 1-d x, and $\mu \simeq 0$, $\sigma^2 \simeq 19$, and $\theta \simeq 52$ in the squared-exponential correlation function (more in Module 3 about estimation, leading to these values).
- We have simulated at a fine grid of $N = 101$ points. Note that x is 1-dimensional here; the 101-dimensional MVN distribution arises because y is considered at 101 points.

5 Realizations of a Gaussian Process in One Dimension



"The Point"

- The range of possible GP realizations covers enough possibilities that they may be representative of a smooth code output.
- We treat the function $y(\mathbf{x})$ as if it is a realization of a random function.
- Before running the code, the set of possible realizations is large.
- After getting data from running the code, the set of realizations must be narrowed to be consistent with the data.

Damped Sin Wave

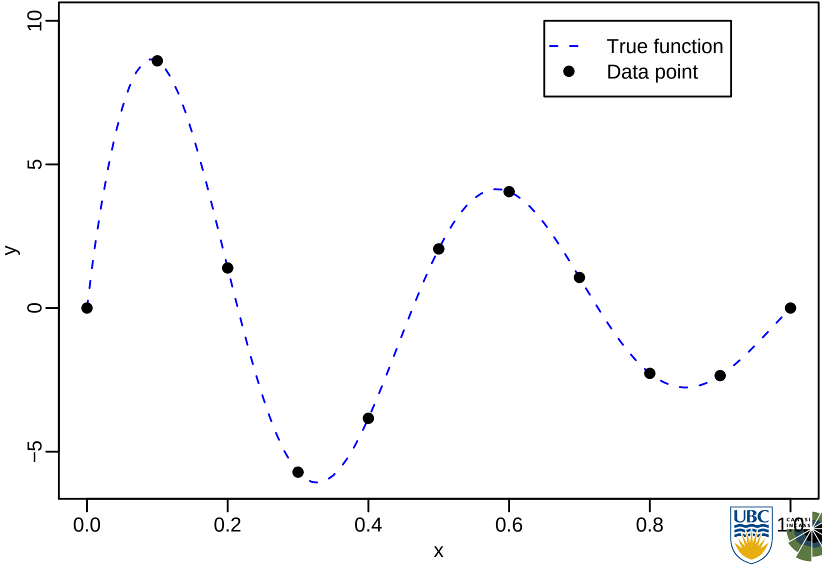
The "damped-sin" function

$$y(x) = 10\sin(4\pi x^{0.9})e^{-1.5x} \qquad (0 \leq x \leq 1)$$

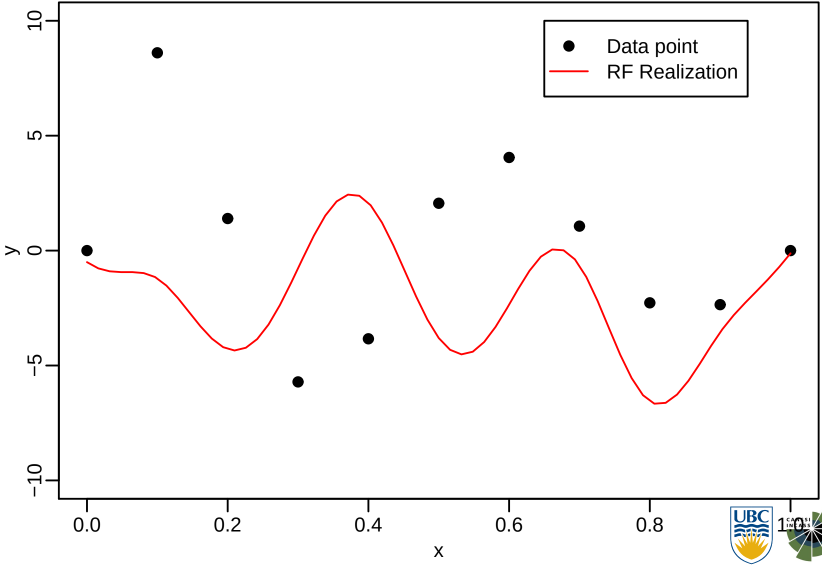
will be used to illustrate the key ideas in approximating a deterministic computer model.

- It is is highly nonlinear and hence complex.
- But it is simple:
  - It is measured without random variability (it represents a deterministic computer model).
  - **x** is only 1-d.
- We will see that the same methodolology extends to high-dimensional **x**.

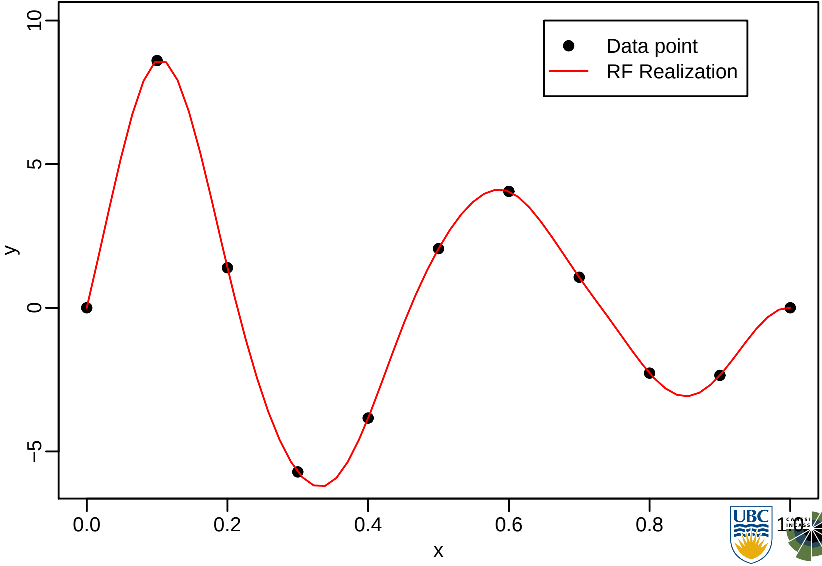
True Function and 11 Runs of the “Code”



A Bad Realization (Inconsistent With Our Data)



A Good Realization (Consistent With Our Data)



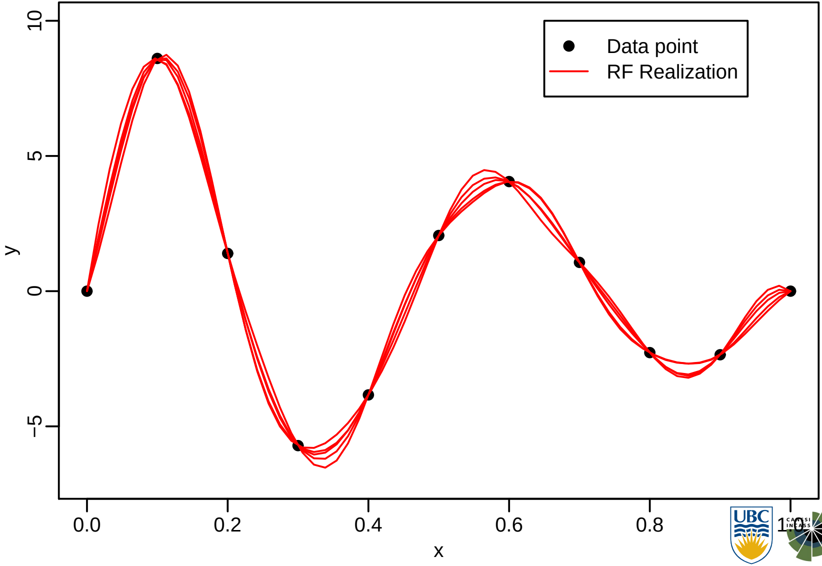
What are Good Realizations?

- $\{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})\}$ at any N "new" points (at which we want to predict) has a prior distribution determined through the MVN distribution.
- Get data $\{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})\}$ by running the code at $n << N$ points (design points).
- Now have a posterior distribution of $y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})$ (or any subset thereof) given the data. It is a conditional MVN distribution

$$\{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})\} \mid \{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})\}.$$

- Good realizations are draws from this posterior distribution (again see Appendix A for details).
- Next slide has five such realizations for the damped sin example.

5 Realizations of the GP Conditional on the Data



Computing the Posterior Distribution

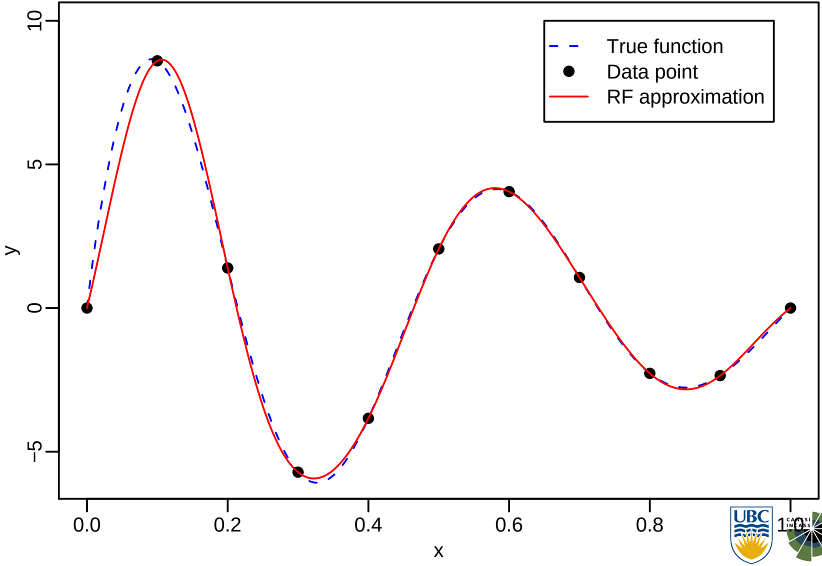
- In practice, we do not have to generate random realizations to predict the function.
- For simplicity, consider predicting y at any single new point, $\mathbf{x}$.
- Given the parameters $(\mu, \sigma^2, \theta, \ldots)$ of the GP, the posterior distribution of $y(\mathbf{x})$ conditional on the data is

$$y(\mathbf{x}) \,|\, \{y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})\} \sim \mathrm{N}(m(\mathbf{x}), v(\mathbf{x})),$$

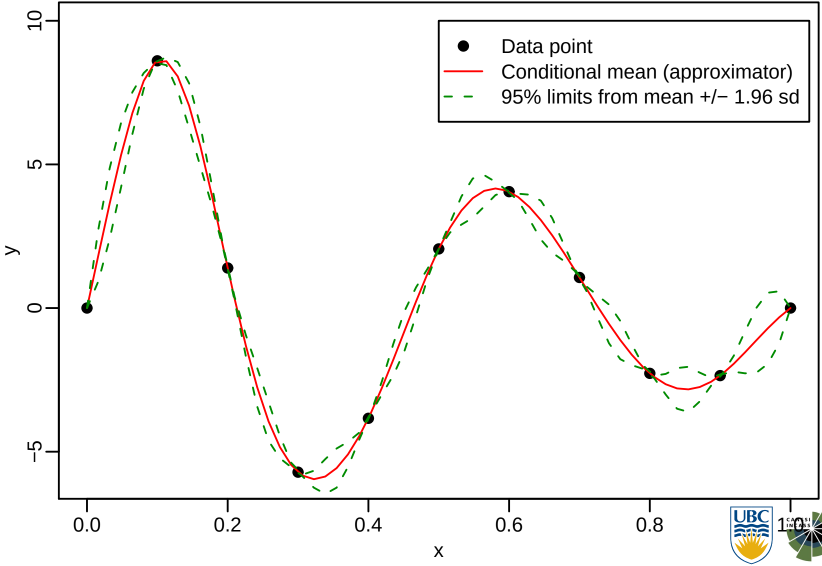
where

- $m(\mathbf{x}) = \mu + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1}(\mathbf{y} - \mu \mathbf{1})$ is the conditional mean, which provides an approximation (prediction) of $y(\mathbf{x})$
- $v(\mathbf{x}) = \sigma^2(1 - \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}))$ is the conditional variance, which provides the variance of the prediction error.
- $\mathbf{R} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ ($n \times n$ matrix)
- $\mathbf{r}(\mathbf{x}) = R(\mathbf{x}^{(i)}, \mathbf{x})$ ($n \times 1$ vector)
- $\mathbf{1}$ is an $n \times 1$ vector of 1's.

Damped Sin: Conditional Mean Approximation



Damped Sin: Approximation and Confidence Limits



What Has / Has Not Been Clarified

Covered:

- Prior uncertainty about y: prior distribution or GP
- Given data from running the code, update uncertainty via Bayes
- Predict at new inputs: posterior mean
- Uncertainty of prediction: posterior variance
- Why Gaussian distribution for prior?
  - Easy to compute

Still to do

- Intuition for using a covariance/correlation function as a prior
- How to estimate the parameters of the GP, including those of the correlation function

Correlation and the Properties of Functions

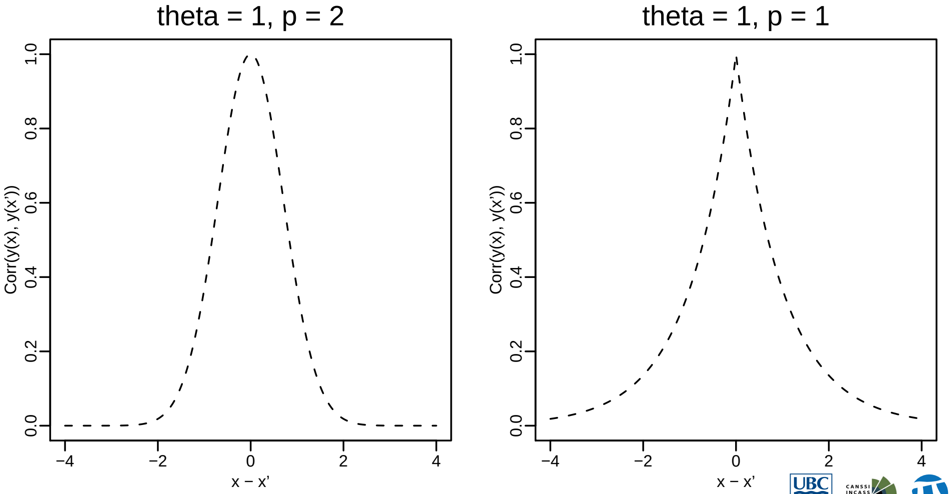
For any two points, $\mathbf{x}$ and $\mathbf{x}'$, in the input space,
$\mathrm{Cor}(y(\mathbf{x}), y(\mathbf{x}')) \equiv R(\mathbf{x}, \mathbf{x}')$ defines the properties of a class of functions.
For a continuous function, $R(\mathbf{x}, \mathbf{x}')$ should be

- 1 when $\mathbf{x} = \mathbf{x}'$
  - (replicates are perfectly correlated)
- Large when $\mathbf{x} \simeq \mathbf{x}'$
  - (two points near to each other in the $\mathbf{x}$ space have highly correlated (similar) function values)
- Small when $\mathbf{x}$ is far from $\mathbf{x}'$
  - (two points far from each other in the $\mathbf{x}$ space have uncorrelated (unrelated) function values).

Power-Exponential Correlation Function

- A popular and flexible class of correlation functions is the power exponential.
- $R(\mathbf{x}, \mathbf{x}') = \prod_{j=1}^{d} \exp(-\theta_j |x_j - x'_j|^{p_j})$.
- $\theta_j \geq 0$ controls the sensitivity of the GP w.r.t. x_j.
  - Larger θ_j gives smaller correlation, i.e., $y(\mathbf{x})$ and $y(\mathbf{x}')$ are less related in the x_j direction and the function is more complex.
  - $\theta_j = 0$ removes x_j (dimension reduction)
- $p_j \in [1, 2]$ affects the smoothness of the GP w.r.t. x_j.
  - $p_j = 2$ (squared-exponential correlation) gives smooth realizations (with infinitely many derivatives).
  - $p_j = 1$ gives much rougher realizations (good for continuous but non-differentiable functions).

Power-Exponential Correlation Function



Parameters of the Prior

- μ, σ^2, θ_j, and p_j are parameters that must be specified to determine the prior.
- They are often called hyperparameters.
- In Module 3 we show how they can be estimated from the data and then used to form the posterior (hence the values for μ, σ^2, and θ used for the damped-sin example).

Module Summary

- Approximate by treating the code input-output function as if it is a realization of a Gaussian process (GP).
- Approximate/predict $y(\mathbf{x})$ by the mean of the conditional distribution given the data and the correlation-function (hyper) parameters.
- Flexible and data adaptive.
- An uncertainty measure comes from the conditional variance.
- How to estimate the (hyper) parameters will be discussed in Module 3.

Appendix A: Simulating Realizations of a GP

Want to generate

$$\mathbf{y}^{(\text{new})} = [y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(N)})]^T,$$

i.e., at N new points, from a MVN distribution with $N \times 1$ mean vector, $\boldsymbol{\mu}$, and $N \times N$ covariance matrix, $\sigma^2 \mathbf{R}$.

- Obtain the Cholesky decomposition, $\mathbf{R} = \mathbf{L}\mathbf{L}^T$
- Generate N iid N(0,1) random variables, $\mathbf{V}$
- Realization $\mathbf{y}^{(\text{new})} = \boldsymbol{\mu} + \sigma \mathbf{L}\mathbf{V}$.
  Note $\text{Cor}(\mathbf{y}) = \mathbf{L}\text{Cor}(\mathbf{V})\mathbf{L}^T = \mathbf{L}\mathbf{I}\mathbf{L}^T = \mathbf{R}$.
- Plot the points $\{\mathbf{x}^{(i)}, y(\mathbf{x}^{(i)})\}$ and connect the dots.

Simulating Realizations Continued

Unconditional and conditional realizations can be generated with appropriate $\boldsymbol{\mu}$ and $\mathbf{R}$ on the previous slide.

- Unconditional realization of $\mathbf{y}^{(\text{new})}$
  - $\boldsymbol{\mu} = 0$ (say)
  - $\mathbf{R} = \mathbf{R}_{N \times N}$
    - $\mathbf{R}_{N \times N} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, an $N \times N$ matrix
- Conditional realization of $\mathbf{y}^{(\text{new})}$ given $\mathbf{y} = [y(\mathbf{x}^{(1)}), \ldots, y(\mathbf{x}^{(n)})]^T$ (data from n code runs)
  - $\boldsymbol{\mu} = \boldsymbol{\mu}^{(0)} + \mathbf{R}^T_{n \times N} \mathbf{R}^{-1}_{n \times n} (\mathbf{y} - \boldsymbol{\mu}^{(0)})$
    - $\boldsymbol{\mu}^{(0)}$ is the unconditional mean vector
    - $\mathbf{R}_{n \times n} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, an $n \times n$ matrix
    - $\mathbf{R}_{n \times N} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, an $n \times N$ matrix
  - $\mathbf{R} = \mathbf{R}_{N \times N} - \mathbf{R}^T_{n \times N} \mathbf{R}^{-1}_{n \times n} \mathbf{R}_{n \times N}$

Appendix B: Dealing With Random Error

Suppose we observe

$$y(\mathbf{x}) + \text{random measurement noise.}$$

Simply model the data as a realization of prior for $y + \epsilon$, where ϵ is independent Gaussian error with mean zero and variance σ_ϵ^2.
In formulas replace

$$\begin{aligned} \sigma^2 \quad &\text{with} \quad \sigma^2_{\text{Total}} = \sigma^2 + \sigma^2_\epsilon \\ \mathbf{R} \quad &\text{with} \quad \frac{\sigma^2}{\sigma^2_{\text{Total}}}\mathbf{R} + \frac{\sigma^2_\epsilon}{\sigma^2_{\text{Total}}}\mathbf{I}_{n\times n} \\ \mathbf{r}(\mathbf{x}) \quad &\text{with} \quad \frac{\sigma^2}{\sigma^2_{\text{Total}}}\mathbf{r}(\mathbf{x}), \end{aligned}$$

where $\mathbf{I}_{n\times n}$ is an $n \times n$ identity matrix.

Realizations of a GP in One Dimension