Gaussian Process Regression for Machine Learning & Statistical Computing

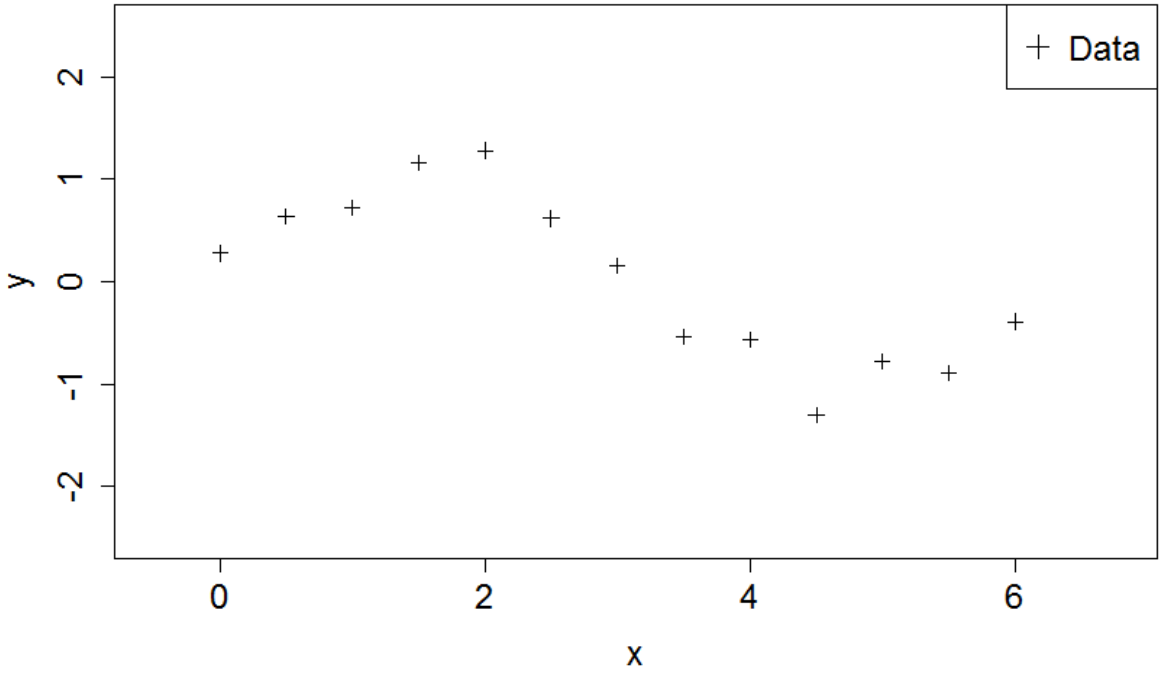
Jimmy Risk

Dept of Statistics & Applied Probability UC Santa Barbara

March 9 2016

Motivation

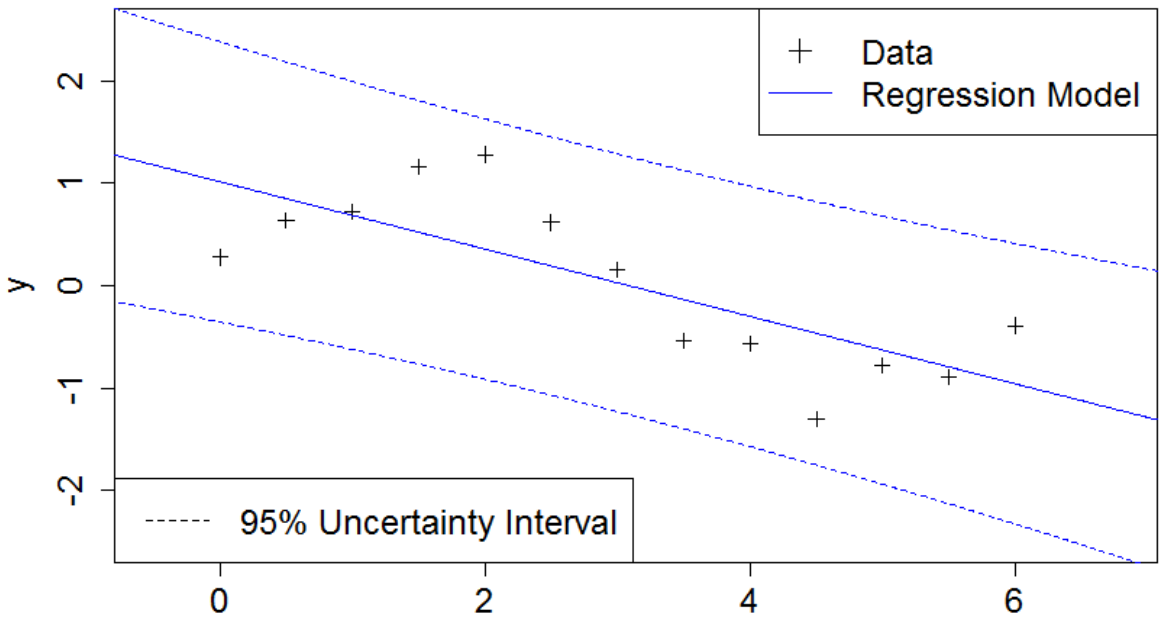
- Given data, how to fit?
- Try simple linear regression



Motivation

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

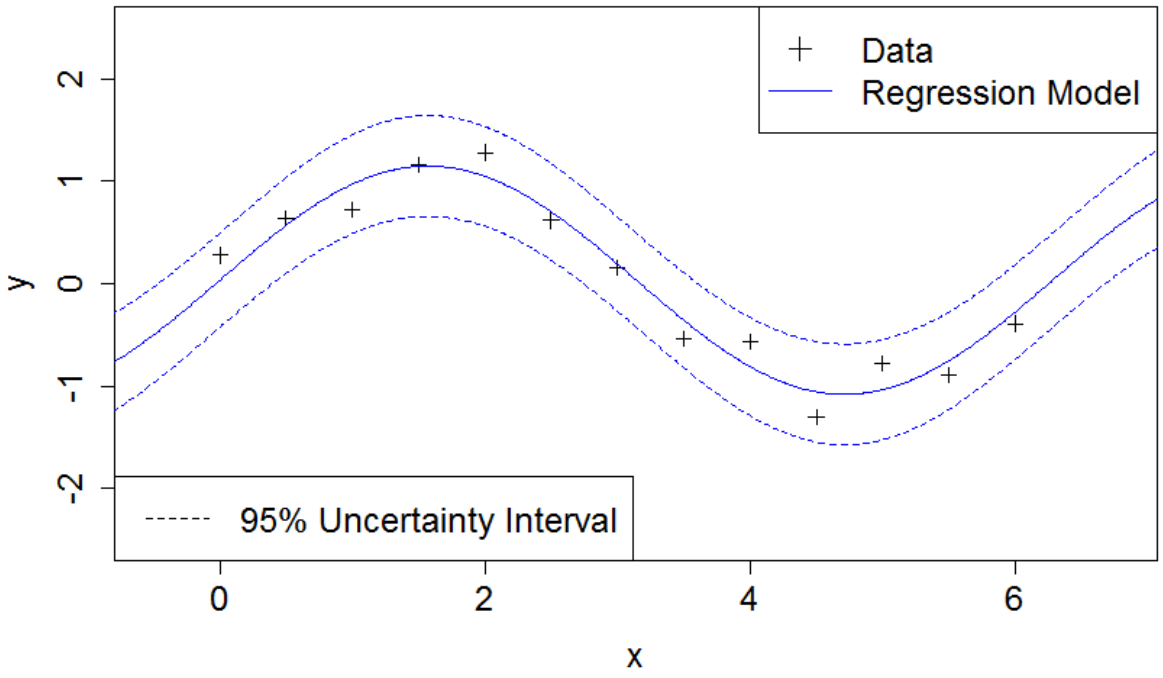
- Parametric - produces line of “best fit," with estimates $\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2$.
- Assumption on $\epsilon_i \sim N(0, \sigma^2)$ yields 95% uncertainty bands
- Could change trend function...



Motivation

$$Y_i = \beta_0 \sin(x) + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

- Better
- Strong assumption on trend



Shifting from the parametric model

Consider

$$Y_i = f(x_i) + \epsilon_i$$

with linear regression

- Heavily dependent on correctly choosing basis functions for $f(x_i)$
  - How many parameters to choose?
  - Should it be a polynomial?
- Difficult to analyze trend in higher dimensions
- Many practitioners blindly choose linear model

Basic GP Idea

For the regression problem of fitting $(x_i, y_i)_{i=1}^N$ to

$$Y = f(x) + \epsilon,$$

Gaussian Process (GP) regression does the following:

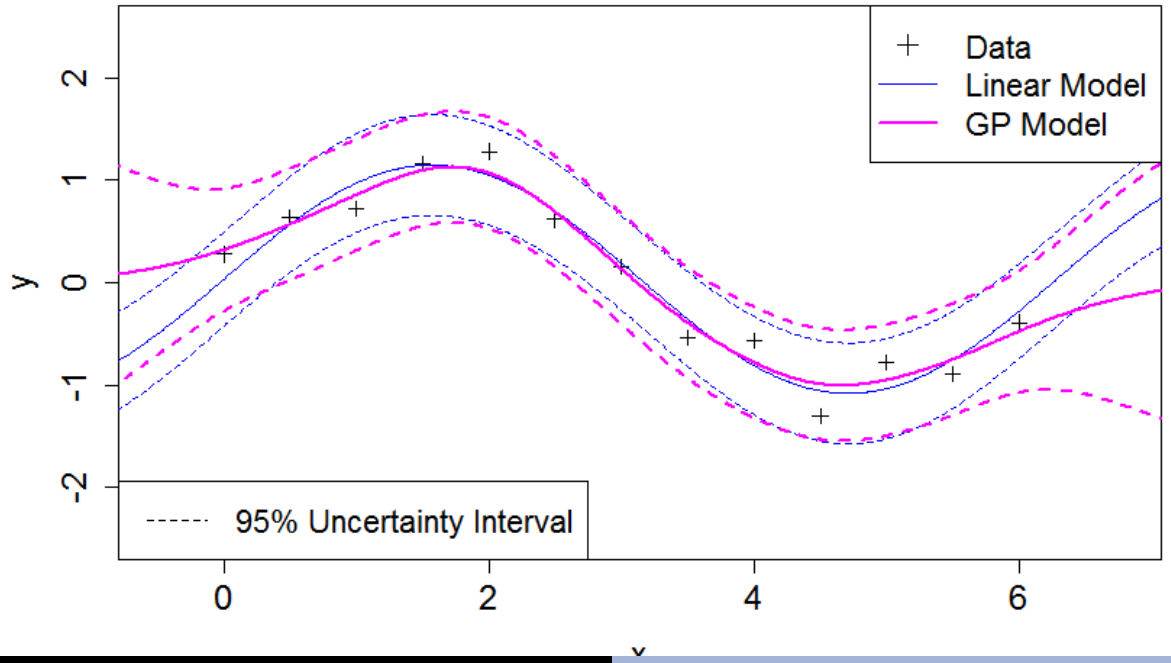
- Assume $f(x)$ has no closed parametric form
- The sample data is one realization of a “random" function
- Finds a distribution over all possible functions $f(x)$ that are consistent with observed data
  - “Output" of the model is a distribution
  - Completely data driven

GP Applied to previous data set

Code input:

```
gp <- km(formula = ~1, design = data.frame(x=x),
response = y, nugget.estim=TRUE)
predict(gp,data.frame(x=xmesh))
```

- `formula = ~1` is the trend assumption (i.e. this model assumes no trend!)

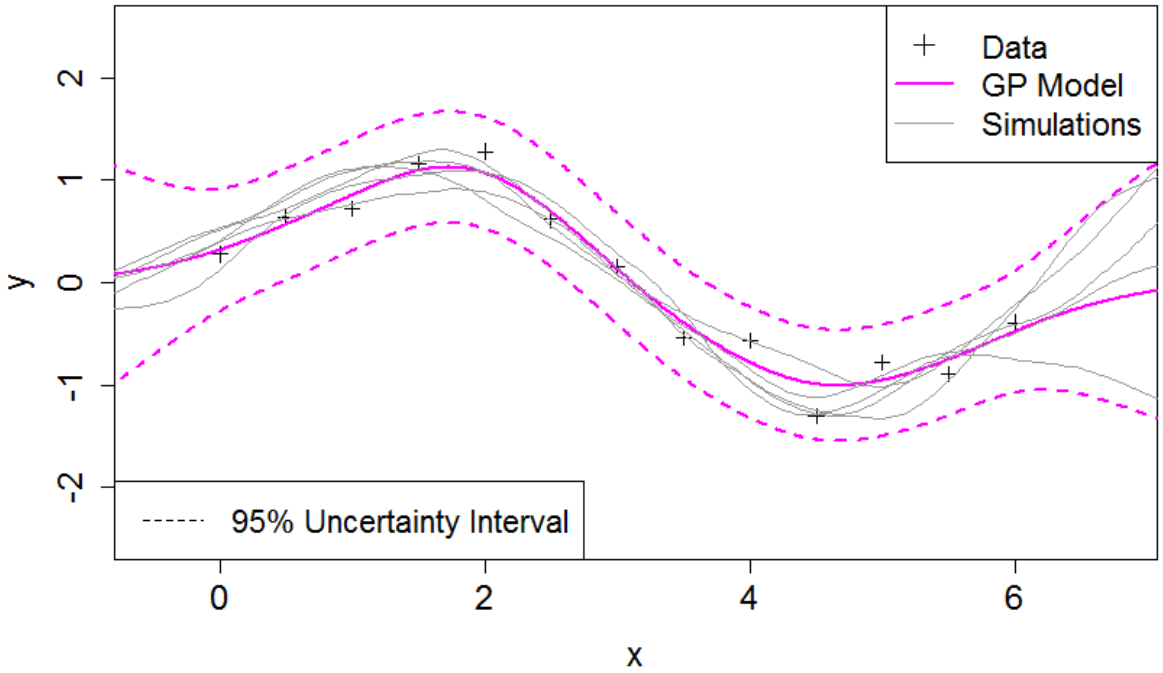


Random sampling of f

5 Realizations (simulations) of $f(x)$. Analogous to

- Flipping a coin
- Taking survey data

The initial data is considered as one realization of f



Technical Details

First, define a Gaussian Process

Definition

$(X_t)_{t \in T}$ is a Gaussian Process if for any finite set of indices $t_1, \ldots, t_k$, the distribution of $(X_{t_1}, \ldots, X_{t_k})$ is multivariate normal.
(X_t) has a covariance kernel C, and the covariance matrix of $(X_{t_1}, \ldots, X_{t_k})$ has entries $\text{cov}(X_{t_i}, X_{t_j}) = C(t_i, t_j), i, j = 1, \ldots, k$. Unless otherwise specified, the mean is assumed to be 0.

Technical Details

Gaussian Process regression attacks the problem of analyzing (for $z \in \mathbb{R}^d$)

$$Y(z) = f(z) + \epsilon(z),$$

where $\epsilon(x)$ is observation noise, by assuming

$$f(z) = \mu(z) + X(z),$$

where

- $\mu : \mathbb{R}^d \to \mathbb{R}$ is a trend function
- X is a mean–zero, square–integrable Gaussian process with covariance kernel C

Covariance Kernel

The covariance kernel determines how locations affect neighboring outputs

- Common assumption is to use:
  - *Stationary* kernels – it depends only on the increment $h = u - v$
  - *Separable* kernels – in higher dimensions, the kernel is a tensor product of 1-d kernels.

For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$,

$$c(\mathbf{h}) := C(\mathbf{u}, \mathbf{v}) = \eta^2 \prod_{j=1}^{d} g(h_j; \theta_j),$$

where $\mathbf{h} = (h_1, \ldots, h_d) = \mathbf{u} - \mathbf{v}$ and g is a 1-d covariance kernel.

Example of covariance Kernel

Example

The Gaussian covariance kernel is defined as

$$g(h) = \exp\left(\frac{-h^2}{2\theta^2}\right).$$

Illustrates how quickly covariance decays as the distance h increases.

Prior Assumptions

The model depends on the following hyperparameters

- θ (characteristic length-scales)
  - Affects the rate at which spatially distant data has an effect on output
- η^2 (process variance)
  - Affects overall fluctuation of the function f
- σ^2 (noise variance)
  - Variance of the observation noise ϵ

These can be prespecified or fitted through MLE (or similar).

An optional trend function $\mu(\cdot)$ can be included

Posterior

- Observe data $\mathcal{D} = (\boldsymbol{y}, \boldsymbol{x}) = ((y_i, x_i)_{i=1}^N)$
  - y is the output, x is the location ($x_i = (x_i^1, x_i^2, \ldots, x_i^d) \in \mathbb{R}^d$)
- Gaussian assumptions imply that marginally for any input x

$$f(x)|\mathcal{D} \sim \mathcal{N}\left(m(x), s^2(x)\right)$$

Posterior

- Observe data $\mathcal{D} = (\boldsymbol{y}, \boldsymbol{x}) = ((y_i, x_i)_{i=1}^N)$
  - y is the output, x is the location ($x_i = (x_i^1, x_i^2, \ldots, x_i^d) \in \mathbb{R}^d$)
- Gaussian assumptions imply that marginally for any input x

$$f(x)|\mathcal{D} \sim \mathcal{N}\left(m(x), s^2(x)\right)$$

- m and s^2 are the posterior mean and variance functions

$$\begin{cases} m(x) \doteq \boldsymbol{c}(x)^T(\boldsymbol{C} + \Sigma)^{-1}\boldsymbol{y}; \\ s^2(x) \doteq C(x,x) - \boldsymbol{c}(x)^T(\boldsymbol{C} + \Sigma)^{-1}\boldsymbol{c}(x), \end{cases} \tag{1}$$

where

$$\begin{cases} \boldsymbol{c}(x) \doteq (C(x, x_i))_{1 \le i \le N} \text{ (covariances between } x \text{ and inputs } \boldsymbol{x}) \\ \boldsymbol{C} \doteq (C(x_i, x_j))_{1 \le i,j \le N} \text{ (covariances between inputs } \boldsymbol{x}) \\ \Sigma \doteq \operatorname{diag}\left(\sigma^2(x_i)\right) \text{ (diagonal matrix of noise variance)} \end{cases} \tag{2}$$

GP Applications

Common application is in coding problems

- Typically in a high dimensional setting, or when
- the code is computationally expensive to run

Examples:

- Numerical solutions to differential equations
- Monte Carlo simulation
  - Engineering
  - Financial math

Also used for spatial modeling

- Originated in geostatistics (where it was called *kriging*)
- Mortality modeling

Quick Motivating Example

Suppose:

- A computer takes x as an input into a function f with output $f(x)$
- It is computationally expensive for the computer to run this code (one evaluation takes e.g. 5 hours).

Quick Motivating Example

Suppose:

- A computer takes x as an input into a function f with output $f(x)$
- It is computationally expensive for the computer to run this code (one evaluation takes e.g. 5 hours).

Further, suppose:

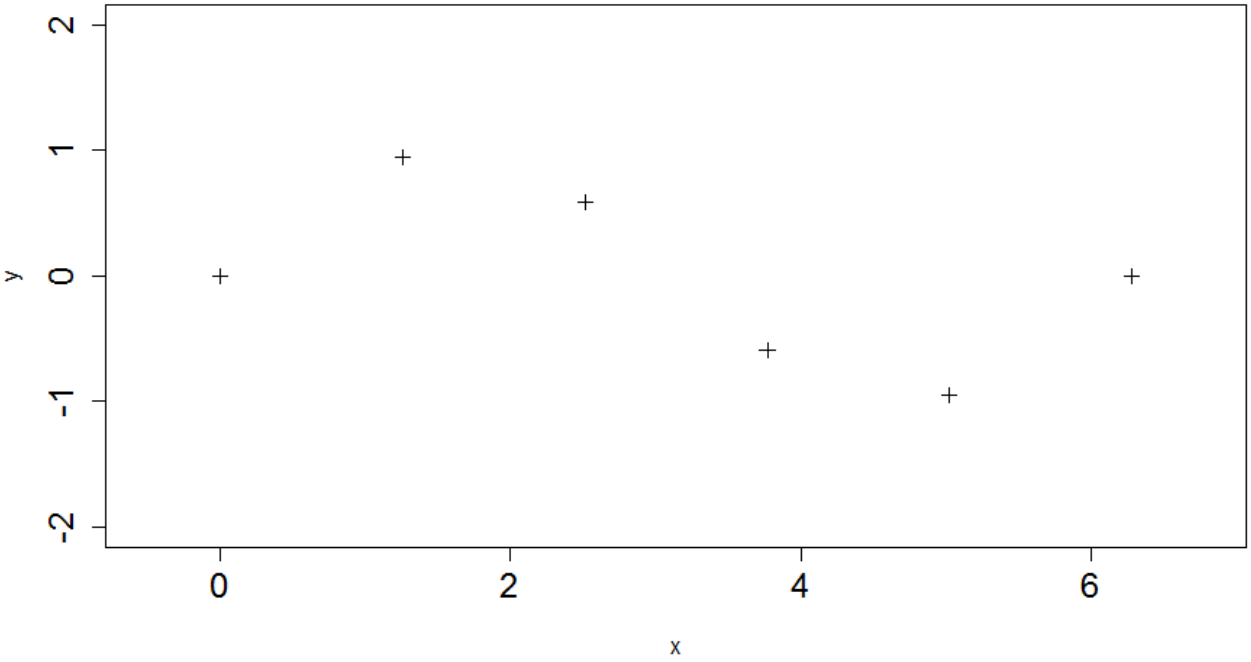
- $f(x) = \sin(x)$ (for simplicity, but we don't know this apriori)
- Want to learn about $f(x)$ over all $x \in [0, 2\pi]$.
- Only have 30 hours (6 runs) to meet a deadline

How can we have a reasonable understanding of the computer function?

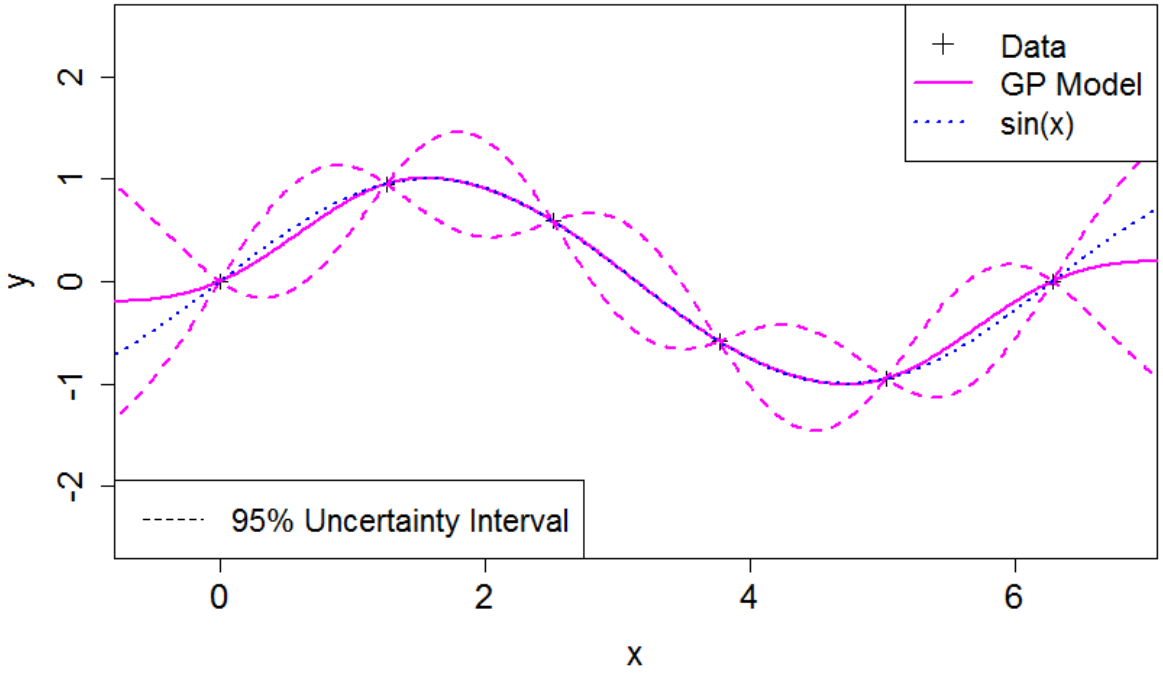
Calibrating f

Use the following strategy:

- Run $f(x)$ at equally spaced points in $[0, 2\pi]$ (i.e. $x_i = 2\pi \frac{i}{5}, i = 0, \ldots, 5$)
- Fit the data $(x_i, y_i)_{i=0}^{6}$ to a GP
- Here, $\epsilon(x) \equiv 0$ since the observations are not noisy

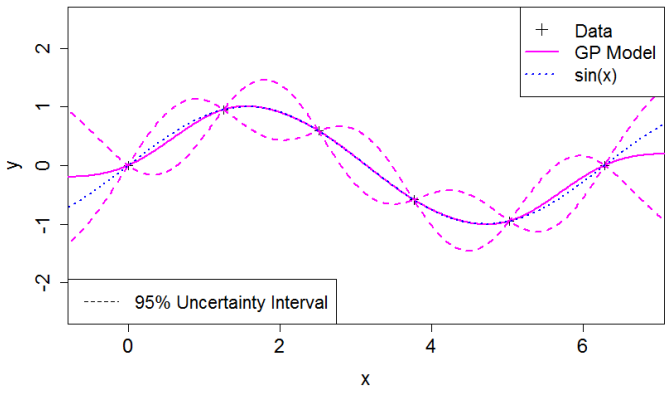


Result

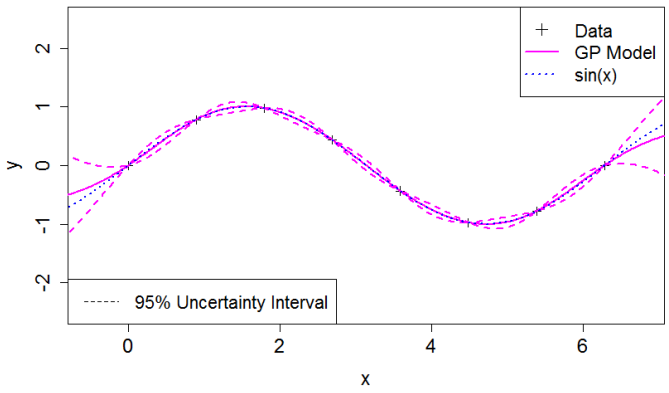


Different scenarios

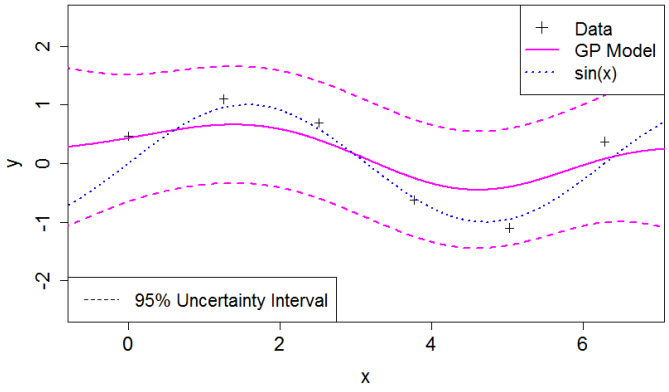
$n = 6, \sigma = 0$



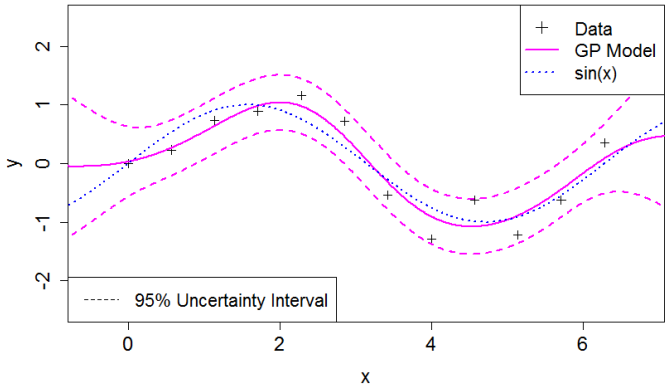
$n = 8, \sigma = 0$



$n = 6, \sigma = 0.25$

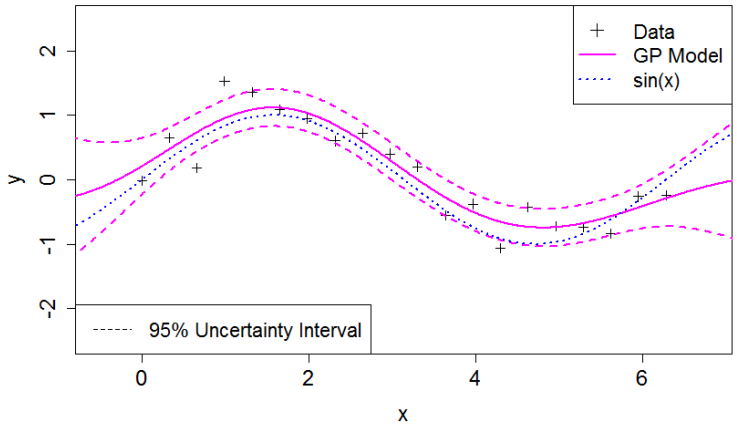


$n = 12, \sigma = 0.25$

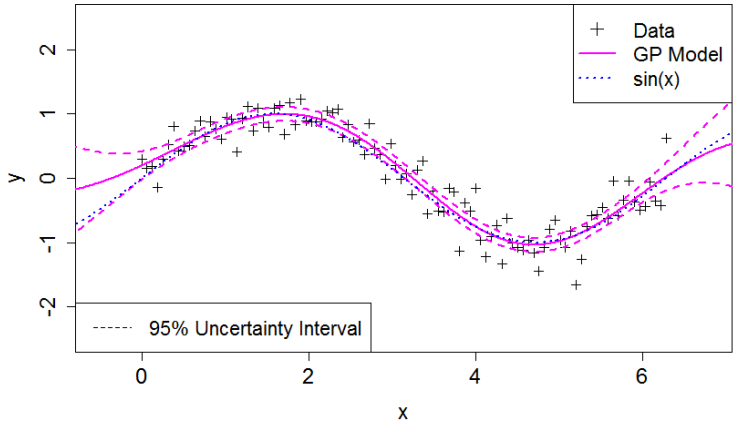


Different scenarios

$$n = 20, \sigma = 0.25$$



$$n = 100, \sigma = 0.25$$



Monte Carlo Applications

- GP regression is useful in Monte Carlo simulation
- Conditional expectation (of a Markov process (Z_t)) can be written as

$$f(z) = \mathbb{E}[\phi(Z_T)|Z_t = z],$$

where we interpret

  - Z_t is the intermediate time t scenario for the process (Z_t),
  - Z_T is what we are interested in at expiration
  - $\phi(\cdot)$ is some function (think payoff)

Monte Carlo Applications

- GP regression is useful in Monte Carlo simulation
- Conditional expectation (of a Markov process (Z_t)) can be written as

$$f(z) = \mathbb{E}[\phi(Z_T)|Z_t = z],$$

where we interpret

  - Z_t is the intermediate time t scenario for the process (Z_t),
  - Z_T is what we are interested in at expiration
  - $\phi(\cdot)$ is some function (think payoff)

$f(z)$ can be *estimated* via simulation (expensive computer code), so we observe

$$Y(z) = f(z) + \epsilon(z),$$

where $\epsilon(z)$ is the Monte Carlo noise whose variance can be easily estimated!

Monte Carlo

$$\mathbb{E}[\phi(Z_T)|Z_t = z]$$

can be estimated using Monte Carlo:

- Simulate Z_T from the distribution of $Z_T|Z_t = z$ (r times)
  - Call the realizations $(Z_T^1, \ldots, Z_T^r)$.
- The law of large numbers says

$$\frac{1}{r}\sum_{i=1}^{r}\phi(Z_T^i) \to \mathbb{E}[\phi(Z_T)|Z_t = z]$$

  as $r \to \infty$
- So, for "large" r,

$$\frac{1}{r}\sum_{i=1}^{r}\phi(Z_T^i) \approx \mathbb{E}[\phi(Z_T)|Z_t = z]$$

Monte Carlo

The approximation error is quantified through the variance of $\phi(Z_T)|Z_t = z$

- $\text{var}(\phi(Z_T)|Z_t = z)$ can be estimated by

$$\hat{\sigma}^2(z) = \frac{1}{r-1}\sum_{i=1}^{r}\left(\phi(Z_T^i) - \frac{1}{r}\sum_{i=1}^{r}\phi(Z_T^i)\right)^2$$

- $\Rightarrow \text{var}\left(\frac{1}{r}\sum_{i=1}^{r}\phi(Z_T^i)|Z_t = z\right)$ is estimated by $\hat{\sigma}^2(z)/r$.

- So, in the problem

$$Y(z) = f(z) + \epsilon(z),$$

  $\hat{\sigma}^2(z)/r$ is an appropriate surrogate for the variance of $\epsilon(z)$.
  - uncertainty at z can be decreased by increasing r at that location

Current Publications

Statistical emulators for pricing and hedging longevity risk products (*Risk, Ludkovski (2016)*) (Insurance: Mathematics and Economics 68)

- Longevity risk is a rising problem
  - Risk associated with people living too long
- Stochastic mortality models have recently boomed in popularity
  - Provide good fit and projections
  - Complicated
    - Accurate analysis often requires time consuming nested simulations
- Pricing many products under stochastic mortality models requires nested Monte Carlo simulations
  - GP assists by fitting at the intermediate time point
    - The typical approach is to use numerical approximations
  - Outperformed industry standard numerical approximations

Current Publications

Gaussian Process Models for Mortality Rates and Improvement Factors (*Ludkovski, Risk, Zail (2016)*) (https://arxiv.org/abs/1608.08291)

- Fit (x_{ag}, x_{yr}) and y as the mortality rate for ages 50–85, years 1999-2015
- Produces a mortality surface in age and year
- Provides easy closed form uncertainty quantification
- GP is differentiable and remains a GP (depending on covariance kernel)
  - Allows to easily analyze mortality improvement $(\frac{d}{dx_{yr}} f(x))$
- Current trends and models say mortality is decreasing near uniformly
  - Our method predicted the increase for middle ages in 2016
  - Other methods did not

“Solvency II" Capital Requirements

Recently implemented (March 2015) in Europe for insurance companies:

- Banking regulations require *Value-At-Risk (VaR)* (quantile) calculations of time $T = 1$ year loss
  - For a given portfolio, what is the 0.5% worst loss that the company could achieve at time $T = 1$?
- Estimating extreme quantiles is difficult
  - Especially with complicated stochastic mortality models
- Big issue in industry, since most do not know how to accurately calculate it
- There is little literature on this topics since it is a new and difficult problem

Setup (Simplified)

The setup:

- 2 stocks with prices (S_t^1) and (S_t^2), where t is time
- Their values can be simulated
- We own a portfolio that gains value as S_t^1 increases and loses value as S_t^2 increases
- Stock prices can be simulated: (for $s < t$)

$$\log(S_t)|S_s \sim N\left(\log(S_s) + r(t-s) - \frac{1}{2}(\sigma_S)^2, (\sigma_S)^2\right)$$

Setup (Simplified)

- $f(z^1, z^2)$ is an unknown function representing the value of a portfolio at time $T = 1$
- Takes in price scenarios
  - i.e. z^i represents a possible value for the stock price S_T^i at $T = 1$
- For given (z^1, z^2), $f(z^1, z^2)$ be estimated to arbitrary degrees of accuracy (by increasing # of simulations at that location)
- Interested in the 0.5% percentile of $f(S_1^1, S_1^2)$
  - i.e. for $N = 10000$ generated scenarios of $(S_1^1, S_1^2) = (z^1, z^2)$, we want the 50th lowest value of $f(z^1, z^2)$

Full Setup

We own 100 call options on S_t^1, and are *short* 50 call options on S_t^2, where a call option has value

$$e^{-\beta(T-t)}\mathbb{E}\left[\left(S_T^i - K^i\right)_+ \middle| S_t^i\right].$$

Therefore, the value of the portfolio at time $T = 1$ when $(S_1^1, S_1^2) = (z^1, z^2)$ is

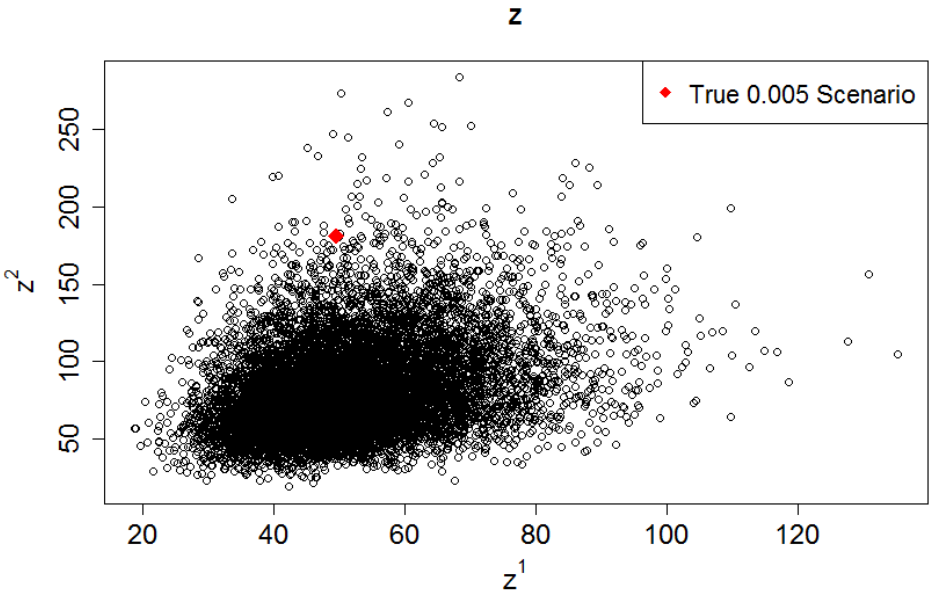
$$f(z^1, z^2) := \mathbb{E}\left[e^{-0.04}\, 100\left(S_2^1 - 40\right)_+ - e^{-2(0.04)} 50\left(S_3^2 - 85\right)_+ \middle| (S_1^1, S_1^2) = (z^1, z^2)\right]. \quad (3)$$

Stock	Position	Initial Price	Strike (K)	Maturity	Volatility
S^1	100	50	40	2	25%
S^2	-50	80	85	3	35%

Assume interest rate of $\beta = 0.04$ and the correlation between S_t^1 and S_t^2 is $\rho = 0.3$

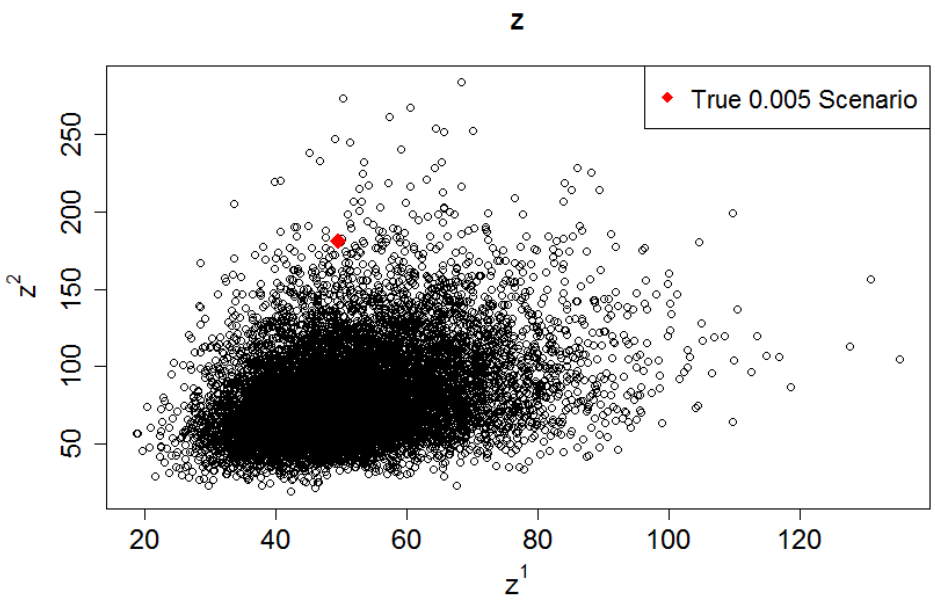
The problem

- First, fix $\mathcal{Z} = \left(z_i^1, z_i^2\right)_{i=1}^{10000}$ as the generated set of future financial scenarios (realizations of (S_1^1, S_1^2))
- For a fixed simulation budget N_{tot}, what is the optimal way to estimate $\mathrm{VaR}_{0.005}$, the 50th lowest value of $f(z_i^1, z_i^2), i = 1, \ldots, 10000$?
- For (z^1, z^2) near to (z_i^1, z_i^2), the approximation quality of $f(z^1, z^2)$ will improve as the number of replications r_i increases at (z_i^1, z_i^2)



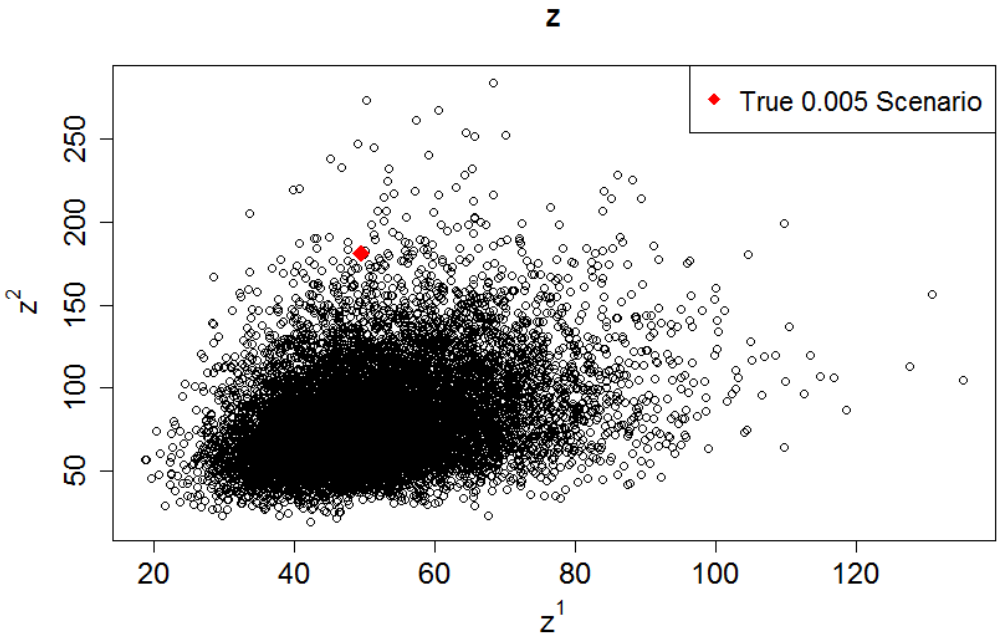
The problem

- Best possible way is to throw all N_{tot} simulations at the true 0.005 scenario
- But its location is unknown
- Initially, f knows nothing since we have no data
- Need an iterative procedure that:
  - learns about f in important regions
  - increases r_i to improve accuracy at important locations (z_i^1, z_i^2)



The problem

- Industry standard is to allocate all simulations equally
  - i.e. each scenario receives $N_{tot}/10000$ simulations
  - The resulting 50th lowest estimate of $f(z^1, z^2)$ is the estimate of $\text{VaR}_{0.005}$.
  - This loses a lot of information, since
    - Nearby points should behave similarly (this method ignores it)
    - Most points are irrelevant and deserve no allocation



Related Ideas (Contour estimation)

The problem boils down to estimation of L, the level of a contour set

$$\mathcal{C} := \{z \in \mathcal{Z} : f(z) = L\}$$

where f is fitted via GPs.

- Investigated in Picheny et. al. (2010)
  - Interest lies in understanding $\mathcal{C}$, not L
  - Little discussion about how to handle unknown L
  - Does not discuss noisy observations

Related Ideas (Tail average)

Investigation of the tail average,

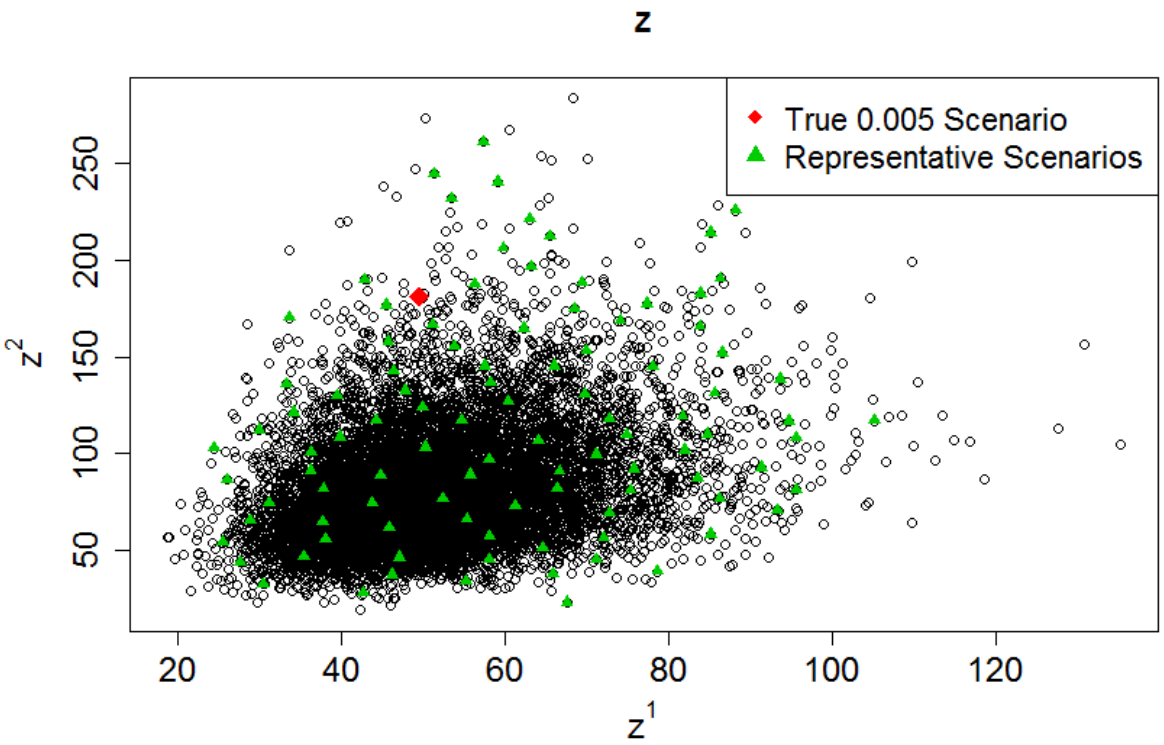
$$\frac{1}{50}\sum_{z\in\Gamma} f(z), \qquad \Gamma = \{z : f(z) \leq \mathrm{VaR}_{0.005}\}$$

using GPs is discussed in Liu & Staum (2010)

- Exact understanding of the contour level is less important
  - Misspecification of edge ordering matters less, since most will be identified correctly and it is an average
- Methods are elementary, since it is the first application of GP in this setting
  - Can be improved significantly
  - Useful as a benchmark

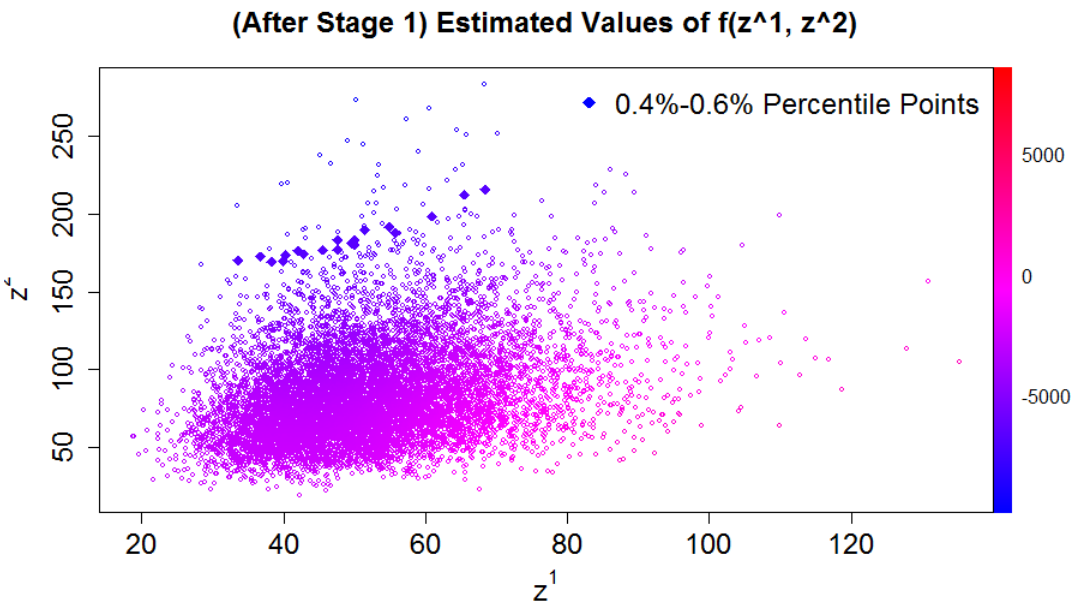
Overall Strategy

- Step 1. Estimate f at representative regions of $\mathcal{Z}$:
  1. Pick 100 scenarios, equally spaced in $\mathcal{Z}$, call them $\mathcal{Z}_1$
  2. Use a small percentage of the budget (5% – 10%) to estimate $f(z), z \in \mathcal{Z}_1$
  3. Call the output $\mathcal{D}_1 = (z_i, y_i, \hat{\sigma}^2(z_i)/r_i)_{i:z_i \in \mathcal{Z}_1}$
  4. Fit a GP to $\mathcal{D}_1$



Overall Strategy

- Sequentially, learn about f in the neighborhood of $\text{VaR}_{0.005}$
- Set $k = 1$. Until budget is depleted, do:
  1. Predict $f(z)|\mathcal{D}_k$ on all of $\mathcal{Z}$.
  2. Estimate $\text{VaR}_{0.005}$ as the 50th ordered prediction
  3. Allocate simulations to point(s) according to some improvement criteria



Total budget $N_{tot} = 10000$ simulations. After stage 1, $N_{remaining} = 9000$.

Choosing Locations

- One improvement criteria is the *targeted integrated mean square error* (timse) from Picheny et. al. (2010)
- L is the current guess for $\mathrm{VaR}_{0.005}$
- The tmse at z is

$$\begin{aligned}\mathrm{tmse}(z) :&= s^2(z)\frac{1}{\sqrt{2\pi(s^2(z)+\varepsilon^2)}}\exp\left(-\frac{1}{2}\left(\frac{m(z)-L}{\sqrt{s^2(z)+\varepsilon^2}}\right)^2\right) \quad (4)\\ &= s^2(z)W(z),\end{aligned}$$

- ε is a quantity parameterizing the uncertainty about L
- $W(z)$ is a weight function that
  - increases both when a location posterior mean close to L, and
  - when it has higher posterior variance

Goal: reduce the posterior variance, but do it for points close to L.

Improvement Criteria

The timse criteria chooses $z \in \mathcal{Z}$ to reduce the average tmse over all $z_i \in \mathcal{Z}$:

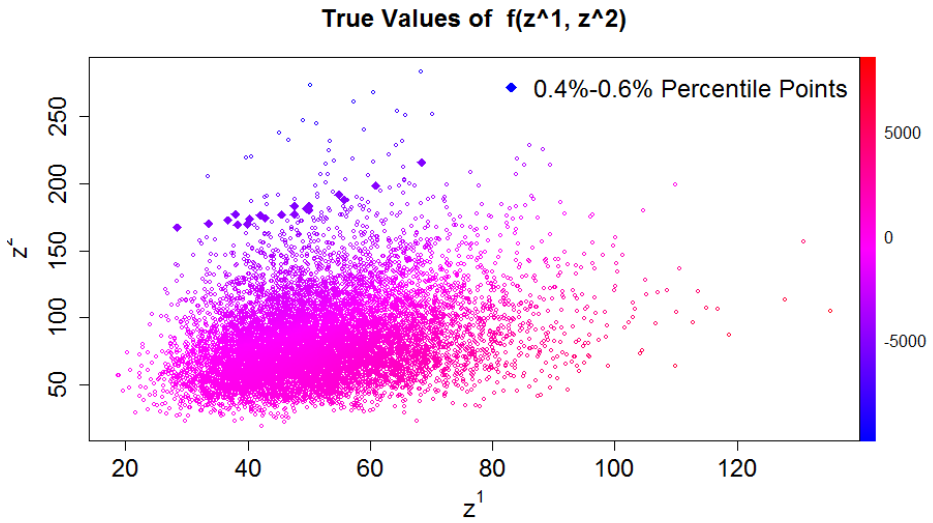
$$\text{timse}(z) := \frac{1}{10000} \sum_{i=1}^{10000} \text{var}(f(z_i)|z^{new} = z) W(z^i) \tag{5}$$

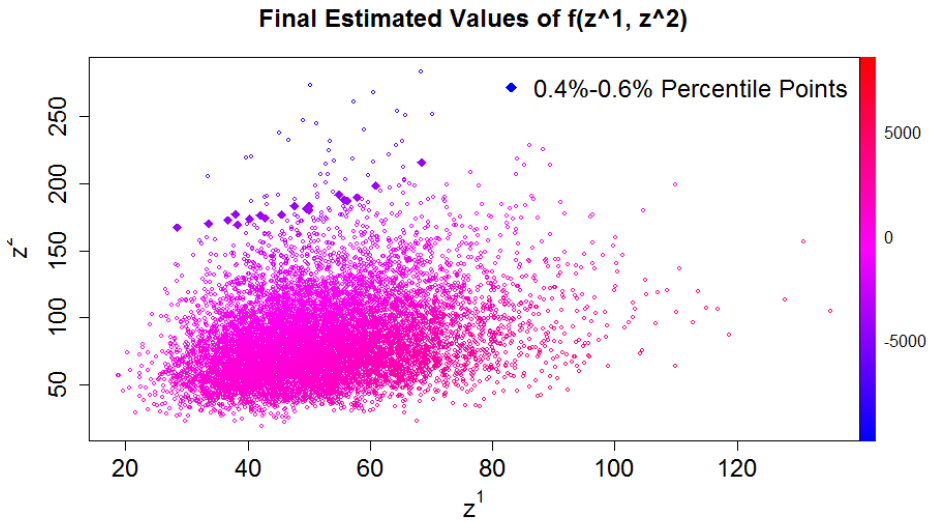
where the notation means the posterior variance of the GP if z is the point to have additional simulations added

Procedure:

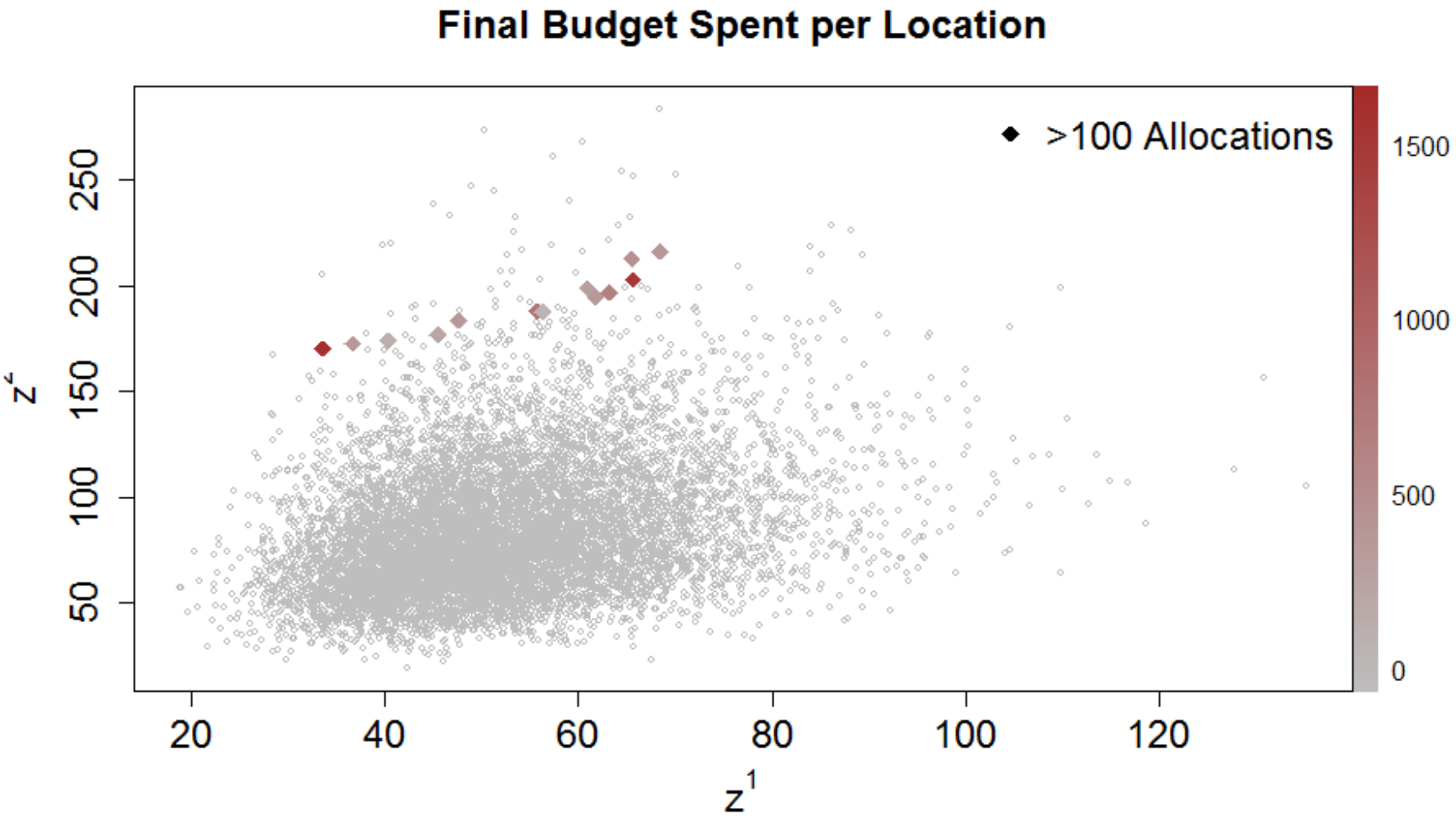
1. Choose $z \in \mathcal{Z}$ that minimizes (5)
2. Allocate simulations to z
3. Fit GP to new data
4. Repeat

True vs Predicted Values of f ($N_{tot} = 10000$)

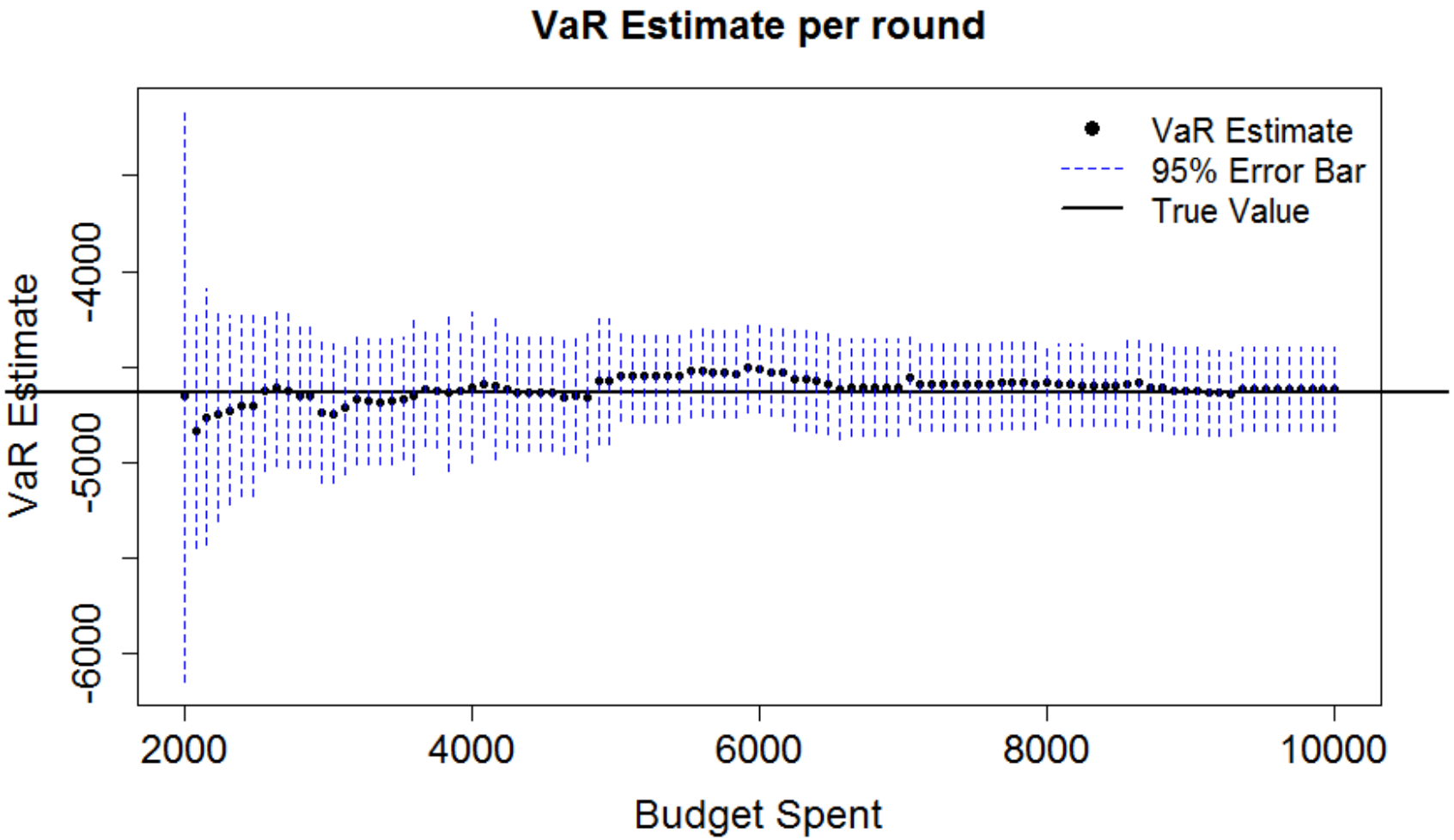




Final Budget Allocation ($N_{tot} = 10000$)



Performance of VaR estimator as budget is depleted



$$\text{True VaR}_{0.005} = -4631.587$$

Numerical Results

True $\text{VaR}_{0.005} = -4631.587$

Method	Bias	$\sqrt{\text{MSE}}$
timse	21.923	69.765
LS	37.357	71.936
Benchmarks		
Plain Monte Carlo	-6834.842	11481.18
Simple Two Stage	64.344	123.971
Best Possible	2.421	48.038

$N_{tot} = 10000$. Bias and $\sqrt{\text{MSE}}$ over 100 macro replications.

- "LS" – Refers to method discussed in tail average paper (Liu Staum (2010))
- "Simple Two Stage" – 10% of budget in round 1, then allocate 90% uniformly to candidate points (those with $W(z) > 10^{-5}$)
- "Best Possible" – Treats location of 0.5% percentile scenario as known. All of N_{tot} is used to simulate at that location. This is the theoretical best possible estimate given the simulation constraint

VaR Conclusion

- Tackling a brand new problem
- Taking existing pieces of related problems, rearranging them in a completely different way
- New methods form as pros and cons of others emerge

Future Plans

- Analysis of other improvement criteria
- Extend to more complicated higher dimensional example

Future Work

Quantile and Level-Set Estimation

- Literature on GP applications in this area has become popular as of late
  - Applications to engineering, computer science
- Most works have little to no work done in the case of noisy observations
  - Importance sampling
  - Convergence results
  - Optimal solutions for given constraints

Future Work

Numerical Solutions to Differential Equations

- Stochastic differential equations
  - Most popular methods use Monte Carlo methods in numerical solutions
- Partial differential equations
  - Duality between PDE's and stochastic processes (by Feynman-Kac formula and others)
  - Solution is a conditional expectation that could be approximated by GPs
  - Least-squares regression methods appear for use in solutions (e.g. Longstaff-Schwarz algorithm)

References

Williams, C. K. and Rasmussen, C. E. 2006.
Gaussian processes for machine learning, the MIT Press.

Adler, Robert J. 2010
The geometry of random fields, Siam

V. Picheny et. al. (2010)
Adaptive designs of experiments for accurate approximation of a target region
Journal of Mechanical Design

M. Liu, J. Staum (2010)
Stochastic kriging for efficient nested simulation of expected shortfall
Journal of Risk

References

J. Risk, M. Ludkovski (2017)
Sequential Design Algorithms for Estimating Value-At-Risk
Work in progress

J. Risk, M. Ludkovski (2016)
Statistical Emulators for Pricing and Hedging Longevity Risk Products
Insurance: Mathematics and Economics 68

M. Ludkovski, J. Risk, H. Zail (2016)
Gaussian Process Models for Mortality Rates and Improvement Factors
https://arxiv.org/abs/1608.08291

J. Risk (2012)
Correlations between Google search data and Mortality Rates
http://arxiv.org/abs/1209.2433