Kriging Metamodels for Bermudan Option Pricing

Mike Ludkovski

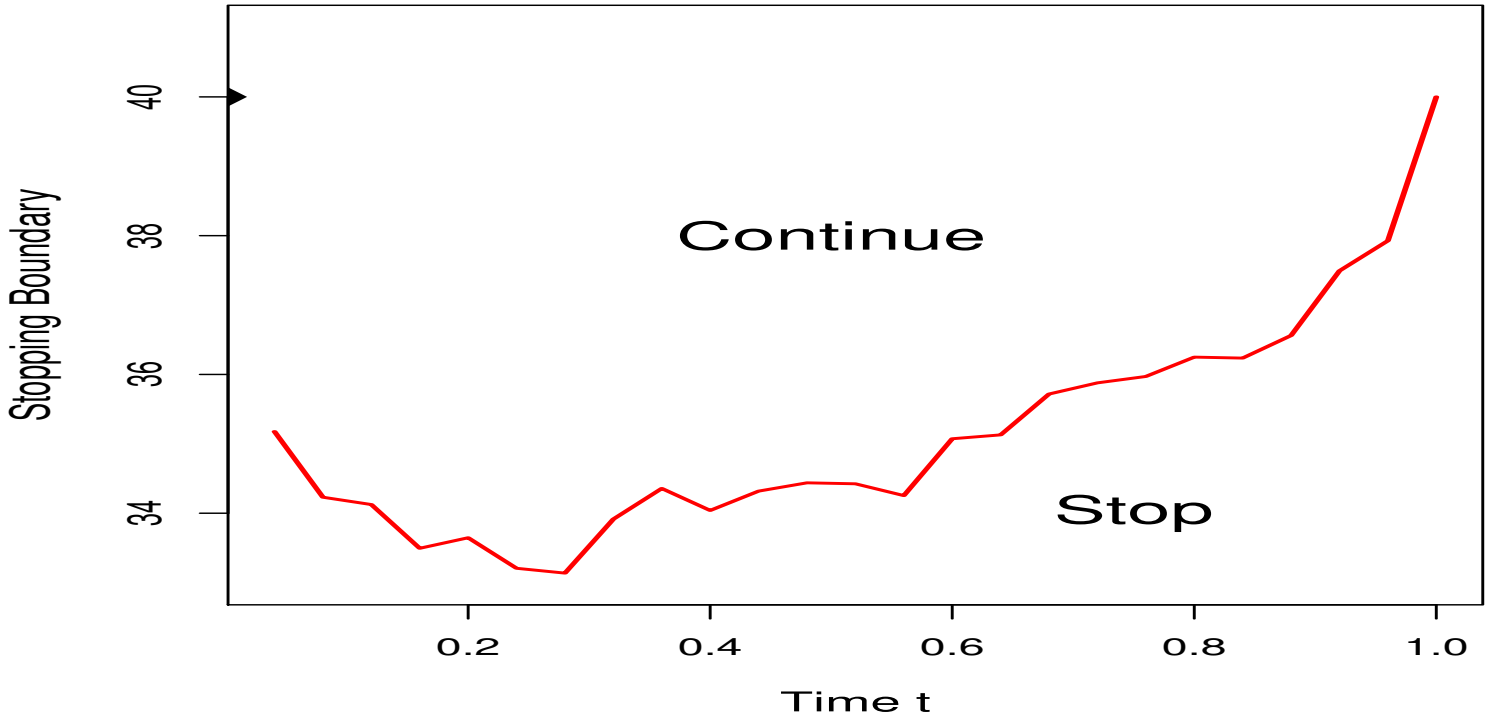
Dept of Statistics & Applied Probability UC Santa Barbara

USC Financial Math Seminar
September 21 2015
Work supported by NSF DMS-1222262

Bermudan Option Pricing/ Optimal Stopping

- State process X, payoff $h(t, X_t)$
- Discrete-time: $t = 1, 2, \ldots$ horizon T
- Value function $V(t, x) = \sup_{\tau \leq T} \mathbb{E}_{t,x} [h(\tau, X_\tau)]$
- Optimization is over stopping times τ
- Solution: $\tau^* = \inf\{t : X_t \in \mathfrak{S}_t\} \wedge T$. Stopping region: $\mathfrak{S}_t = \{x : V(t, x) = h(t, x)\}$
- eg (X_t) is GBM; $h(t, x) = e^{-rt}(K - x)_+$ – Bermudan Put

Stopping Rule via Timing Value



$$T(t,x) := \mathbb{E}_{t,x}\left[V(t+1, X_{t+1})\right] - h(t,x) = \mathbb{E}_{t,x}\left[h(\tau_{t+1}, X_{\tau_{t+1}})\right] - h(x).$$

- Stopping decision is characterized by $\mathfrak{S}_t = \{x : T(t,x) < 0\}$
- To find τ^*, it's sufficient to evaluate the conditional expectation, i.e. **approximate the sign** of $T(t,\cdot)$ for $t = T-1, T-2, \ldots, 0$

Conditional Expectation

$$f(x) := \mathbb{E}[h(X_\cdot)|X_0 = x].$$

- Input: Markov process X with state space $\mathcal{X}$ & (path-) Functional $h(X_\cdot)$
- Output: the conditional mean map $x \mapsto f(x)$
- Generalizes the problem of *pointwise* estimates at a fixed x
- Appears as a building block:
  - Optimal switching/impulse control
  - XVA
  - BSDEs
  - Capital Requirements/Insurance

Regression Monte Carlo

$$f(x) := \mathbb{E}[g(X_\cdot)|X_0 = x].$$

- Classical Monte Carlo for a fixed x_0: $\widehat{f}(x_0) := \frac{1}{N}\sum_{n=1}^{N} h(x_\cdot^n)$ where $x_\cdot^n$ are N *simulated paths*
- Need to be able to predict $f(x)$ for **any** $x \in \mathcal{X}$
- The state space $\mathcal{X}$ is multi-dimensional and continuous

$\rightarrow$ Construct a grid $x^{1:N}$ and borrow information spatially

- Statistical regression: smooth + interpolate

RMC for Optimal Stopping

- Backwards induction in time ($\mathfrak{S}_T = \{x : h(t,x) \geq 0\}$)
- Given stopping sets: $\hat{\mathfrak{S}}_{t+1:T}$
- Starting at $X_t = x$, simulate trajectory $X^x_{t+1:T}$ and take $\tau' = \inf\{s > t : X^x_s \in \hat{\mathfrak{S}}_s\}$
- Pathwise future payoff $y := h(\tau', X^x_{\tau'})$ satisfies

$$\mathbb{E}_{t,x}\left[Y_x\right] = C(t,x) \quad \Leftrightarrow \quad Y_x = C(t,x) + \varepsilon(x)$$

  where $C(t,x) = T(t,x) + h(t,x)$ is the **continuation value**
- Now generate a stochastic grid $(x^n_t)^N_{n=1}$ and paths $x^{1:N}_{t+1:T}$
- Obtain a sample $\{x_t, y_t\}^{1:N}$
- Estimate $\hat{C}(t,\cdot)$ and set $\hat{\mathfrak{S}}_t := \{x : \hat{C}(t,x) - h(t,x) < 0\}$
- Popularized by Longstaff & Schwartz (2001)

Metamodeling

AIM: Build an approximation of $\hat{C}(t, \cdot)$

- Choose an approximation architecture $\mathcal{H}$ and loss function L
- Generate the grid $x_t^{1:N}$: Experimental Design
- Set $\hat{C}(t, \cdot) = \arg\min_{C \in \mathcal{H}} L(C; (x, y)^{1:N})$
- Repeat over $t = T - 1, T - 2, \ldots$

Traditionally:

- Data is generated using the transition density of X ("path-simulation")
- Least-Squares parametric regression, i.e. $\mathcal{H} = span(B_i(x), i = 1, \ldots, r)$
- (The implied loss function is $\mathbb{E}_{0,X_0}[\{\hat{C}(X_t) - C(X_t)\}^2]$)

What is Metamodeling?

- Classical regression – data is given and try to fit the "best curve"
- In metamodeling generating data (through efficient simulations) is part of the solution
- Also, typically look for a non-parametric model (dense $\mathcal{H}$)
- Goes by many other names: response surface modeling, statistical learning, DACE (design and analysis of computer experiments), emulation
- Used extensively in machine learning; simulation optimization, computational statistics
- See eg Kleijnen (2015), Williams and Rasmussen (2006), Powell and Ryzhov (2012)
- Connects to CS, OR, stats communities (language barriers!)

Improving RMC

- Main concerns are Speed/memory – convergence of RMC is slow; often need $\gg 10^5$ paths to obtain a good estimate
- Desire ability to handle a "**black-box**" setting, e.g. 5-D system with implicit dynamics, and limited known structure
- Timing optionality is now embedded in a ton of contracts – wish to have a "universal" algorithm
- Traditional methods offer few performance guarantees (eg. sensitive to the choice of basis functions) and are hard to trust

Contributions

- There has been extensive ongoing research on better regressions: Belomestny, Bouchard, Gobet, Kohler, Oosterlee, Stentoft, Tompaidis, ...
- Also analysis of error propagation through dependent regressions: Egloff (2004), Gobet and Warin (2006), Belomestny (2011), Gerhold (2011), Kohler (2012), Zanger (2013)

Contributions

- There has been extensive ongoing research on better regressions: Belomestny, Bouchard, Gobet, Kohler, Oosterlee, Stentoft, Tompaidis, ...
- Also analysis of error propagation through dependent regressions: Egloff (2004), Gobet and Warin (2006), Belomestny (2011), Gerhold (2011), Kohler (2012), Zanger (2013)
- Contribution 1: investigate impact of RMC experimental designs and suggest several (improved) choices
- Contribution 2: propose use of kriging metamodels
- RMC is often called Least Squares Monte Carlo. This puts misplaced narrow emphasis on a specific regression framework, and tends to ignore the design aspect. We advocate a shift in terminology to better align with the underlying problem.

Modeling Conditional Expectation

$$f(x) := \mathbb{E}[g(X_\cdot)|X_0 = x].$$

Must impose some structure on f (X is a "nice" process, so f is "smooth")

- Project onto basis functions: $f(x) = \sum_{i=1}^{R} a_i H_i(x)$
- Smoothing spline (piecewise cubic)
- Piecewise linear
- Piecewise constant $f(x) = \sum_i a_i 1_{\{x \in R_i\}}$
- Fully nonparametric (kernel): $f(x) = \sum_i K(x, x^i) y^i$
- Gaussian process

Stochastic Kriging

- Data-generating process $Y(x) = C(t, x) + \varepsilon(x)$ where $\varepsilon(x) \sim N(0, \sigma^2(x))$
- Assume the continuation value $C(t, \cdot)$ lives in the function space $\mathcal{H}_K$ – Gaussian RKHS
- Means $C(t, \cdot)$ is a realization of a Gaussian random field with a covariance structure defined by K, $\mathcal{H} = span(K(\cdot, x) : x \in \mathcal{X})$
- $K(x, x') := \mathbb{E}[f(x)f(x')]$ controls the spatial decay of correlation, i.e. smoothness of $C(t, \cdot)$
- e.g Gaussian kernel $K(x, x') = \tau^2 \exp(-\|x - x'\|^2/\theta^2)$ – elements of $\mathcal{H}_K$ are C^∞, with lengthscale θ and fluctuation scale τ.
- Use L^2 projection: $\hat{C}(t, \cdot) = \arg\min_{C \in \mathcal{H}} \sum_{i=1}^N (C(x^i) - y^i)^2$;
- Representer theorem implies that $\hat{C}(t, x) = \sum_{i=1}^N w_i K(x, x^i)$

Stochastic Kriging

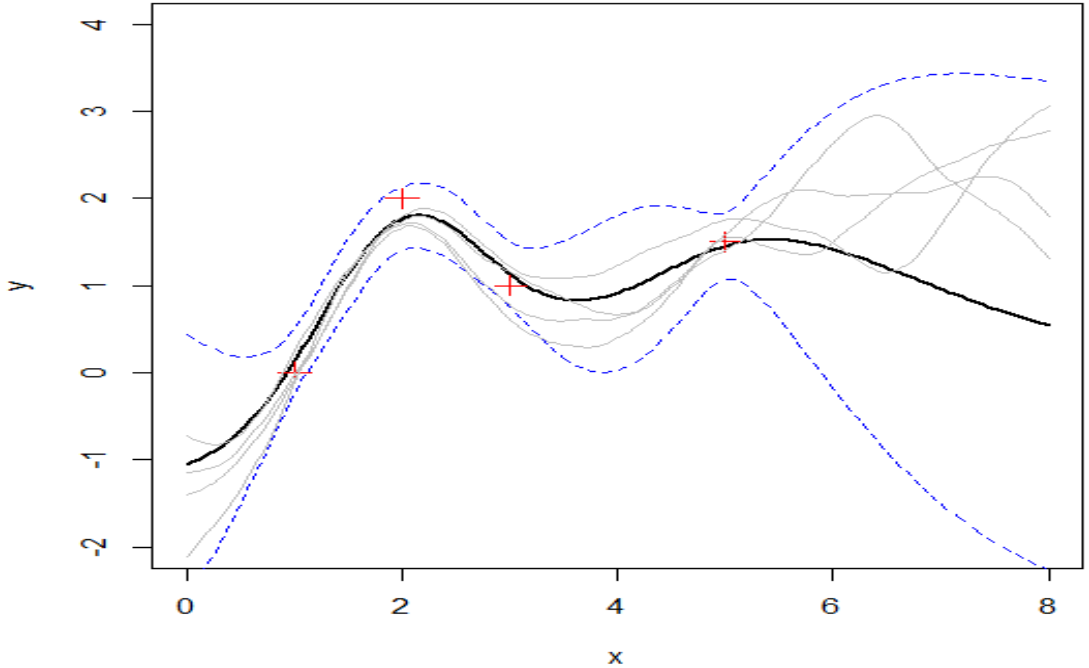
- Think of $C(t, \cdot)$ as a random element in $\mathcal{H}_K$ with a Gaussian prior $C(t, x) \sim N(0, \tau^2)$
- The posterior conditional on $\mathcal{G} \equiv (x, y)^{1:N}$ is also Gaussian
- Marginally $C(t, x)|\mathcal{G} \sim N(m(x), v^2(x))$

$$m(x) = \vec{k}(x)^T(\mathbf{K} + \Sigma)^{-1}\vec{y}$$
$$v(x, x') = K(x, x') - \vec{k}(x)^T(\mathbf{K} + \Sigma)^{-1}\vec{k}(x')$$

- $K_{ij} = K(x^i, x^j)$, $\Sigma = diag(\sigma^2(x^i))$, $k_i = K(x, x^i)$
- Linear model in the infinite basis expansion defined by K

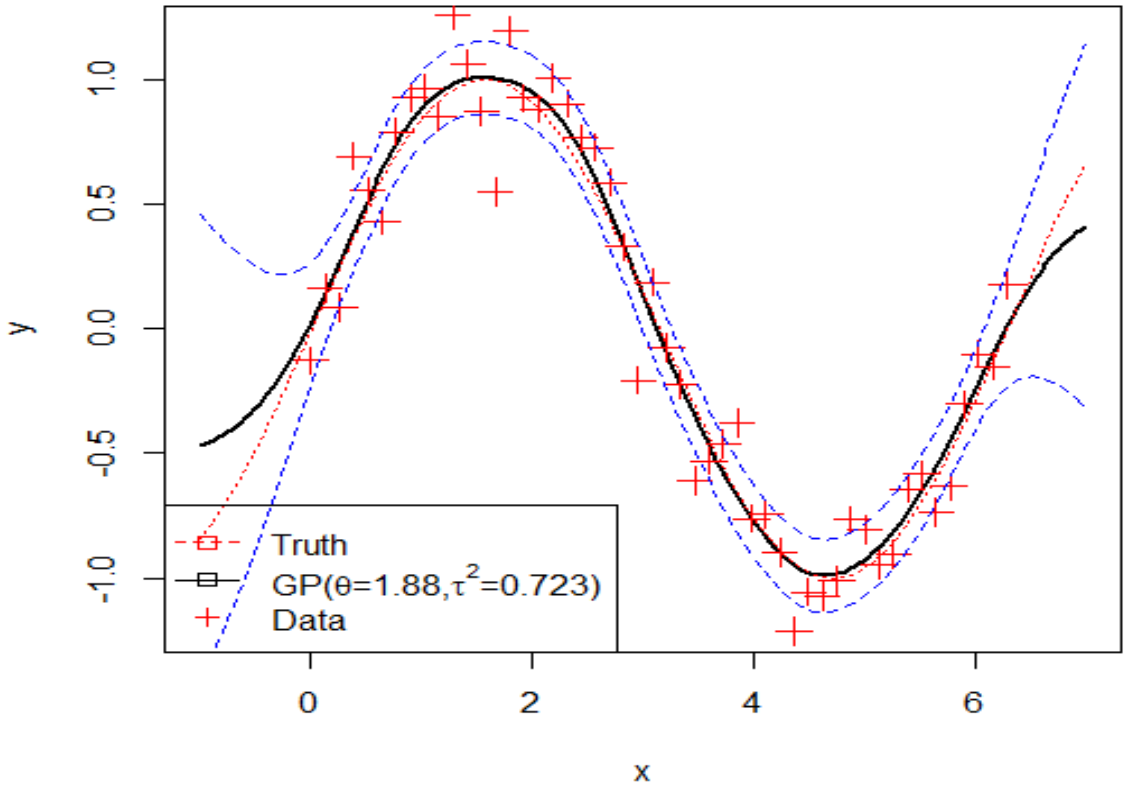
Kriging Example 1

- The posterior is a measure on $\mathcal{H}_K$ (i.e function-valued)
- Visually has a "football" shape– $v^2(x)$ has local minima at x^i's.
- The mean $m(x)$ is a linear combination of kernel eigenfunctions centered at design sites
- Outside the domain $\mathcal{X}'$, revert to prior $m(x) \to 0, v^2(x) \to \tau^2$
- Below: $\theta = 2, \tau = 1.5, \sigma^2(x) \equiv 0.2^2$



Kriging Example 2

- Global consistency – converge to the truth as $N \to \infty$
- Optimized Matern-5/2 kernel
  $K(x, x'; \tau, \theta) = \tau^2 \left(1 + (\sqrt{5} + 5/3)\|x - x'\|_\theta^2\right) \cdot e^{-\sqrt{5}\|x - x'\|_\theta}$



Fitting a GP

- Need to know the kernel hyperparameters – τ, θ, et cetera. Use MLE (nonlinear optimization problem).
- θ is the lengthscale – correlation decay
- τ^2 is the process variance – has analytic MLE once θ is known
- GP is expensive compared to e.g LM; complexity is $O(N^3)$ for a design of size N
- Allows a lot of analytic formulas to understand the fit and its uncertainty
- Kriging is becoming the gold standard in the simulation/DACE communities
- Used `DiceKriging` package in `R` – off-the-shelf use

Simulation Noise

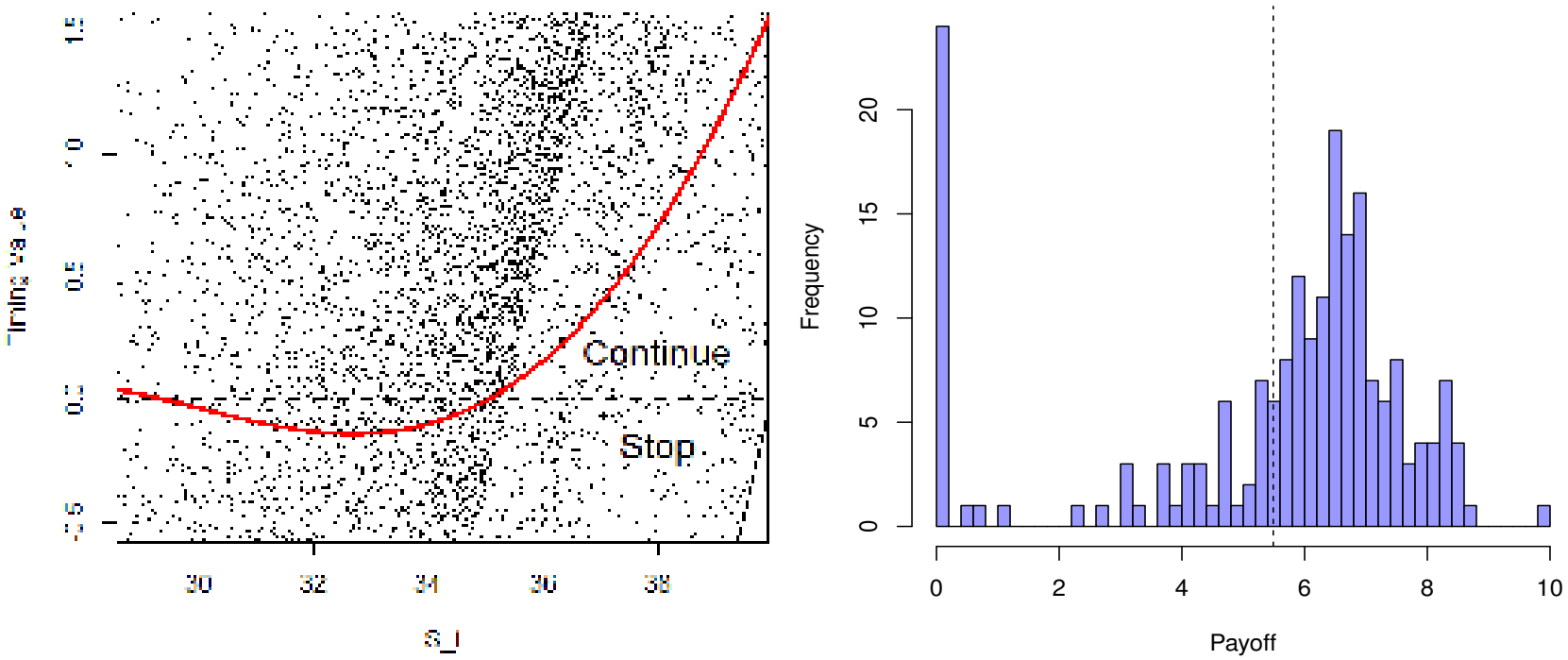


Figure: Left: scatterplot of $(x, H_t(X_\cdot^x) - h(t,x))$ over 10,000 distinct $x \in \mathbb{R}_+$. Right: Histogram of $N = 200$ pathwise future payoffs $y^n \sim H_t(X_\cdot^x)$ starting at $x = 35$ in a 1-D Bermudan Put problem; $t = 0.6$. The vertical dashed line indicates the empirical mean $\mathbb{E}[H_t(X_\cdot^x)|X_t = 35] \simeq Ave(y^{1:N}) = 5.49$. Note that in 24 out of 200 scenarios, the payoff y^n was zero, creating a point mass in the distribution of $H_t(X_\cdot^x)$ and generating a significant negative skew. Other moments were $StDev(y^{1:N}) = 2.45$, $Skew(y^{1:N}) = -1.28$ and $Max(y^{1:N}) = 9.87$.

Simulation Noise

- Knowing the distribution of simulation noise $\varepsilon(x)$ is fundamental for meta-modeling
- Simulation noise is highly state-dependent in RMC
- Also, distribution can be skewed/far from Gaussian
- Solution 1: treat it as a constant σ^2 (so-called "nugget"), can estimate along with other kernel hyper-parameters
- Solution 2: build an empirical estimate through replicating simulations at a fixed site x
- (Resembles a Monte Carlo forest)
- Solution 3: model $x \mapsto \sigma^2(x)$ via an auxiliary metamodel

Batching

- Generate M independent realizations $y^{(i)} \sim Y_x$ of pathwise payoffs starting at $X_t = x$
- Set the average $\bar{y}(x) = \frac{1}{M}\sum_{i=1}^{M} y^{(i)}(x)$
- Empirical $\widetilde{\sigma}^2(x) := \frac{1}{M-1}\sum_{i=1}^{M}(y^{(i)}(x) - \bar{y}(x))^2$
- The averaged simulations still follow the same statistical model but with signal-to-noise ratio improved by factor of M
- Size of macro-design $\mathcal{Z}'$ is N/M – much faster fitting
- Also, $\bar{Y}$ has almost-Gaussian simulation noise

Batched Kriging Metamodel for $T(t, \cdot)$

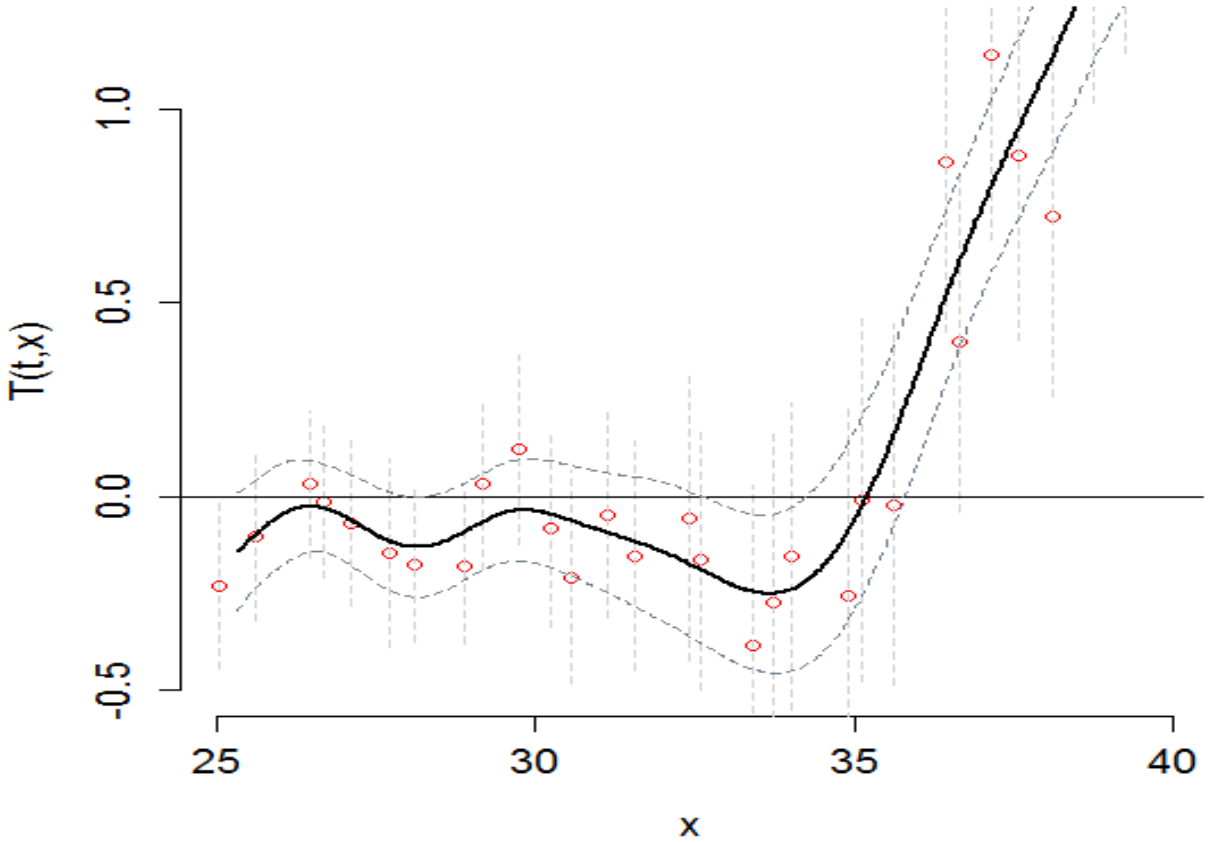


Figure: LHS design $\mathcal{Z}$ of size $N = 3000$ with $M = 100$ replications. The vertical "error" bars indicate the 95% quantiles of the simulation batch at x, while the dotted lines indicate the 95% credibility interval (CI) of the kriging metamodel fit.

Deterministic Kriging

- If M is very large, $\widetilde{\sigma}^2(x)/M \simeq 0$ and can view $\bar{Y}_x$ as deterministic
- Metamodel becomes an interpolator

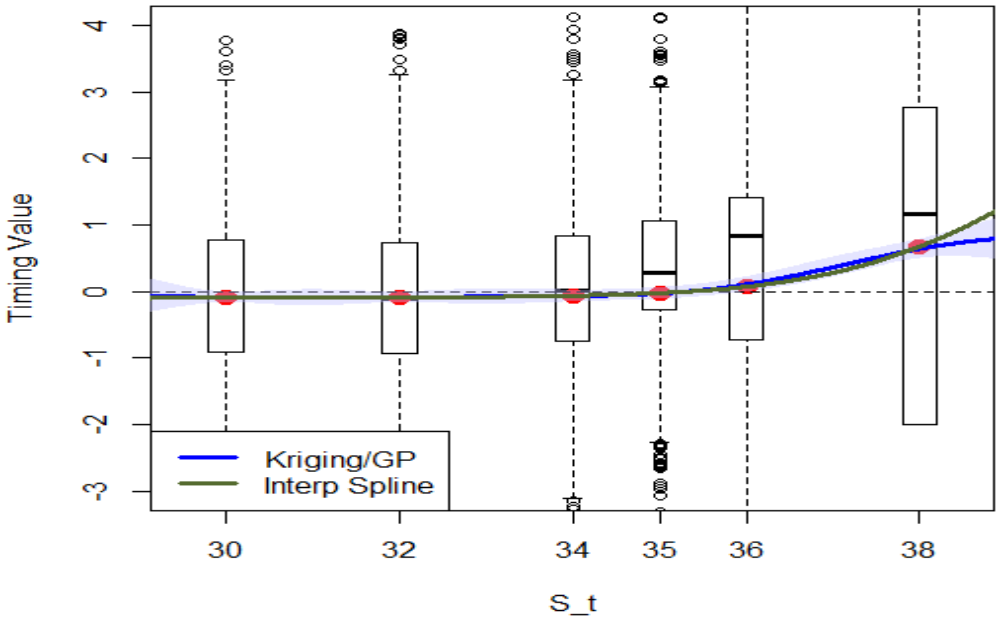
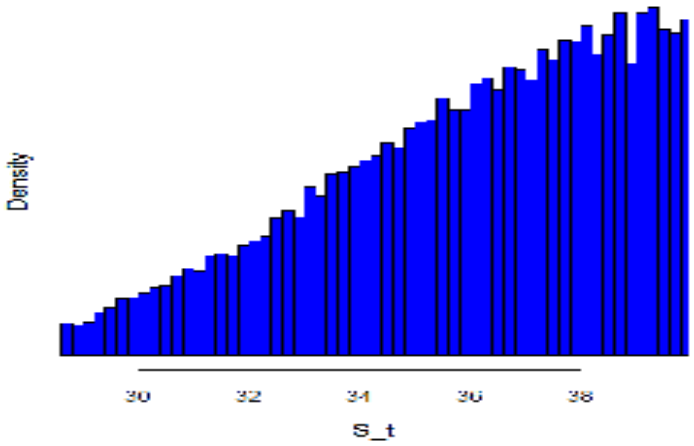
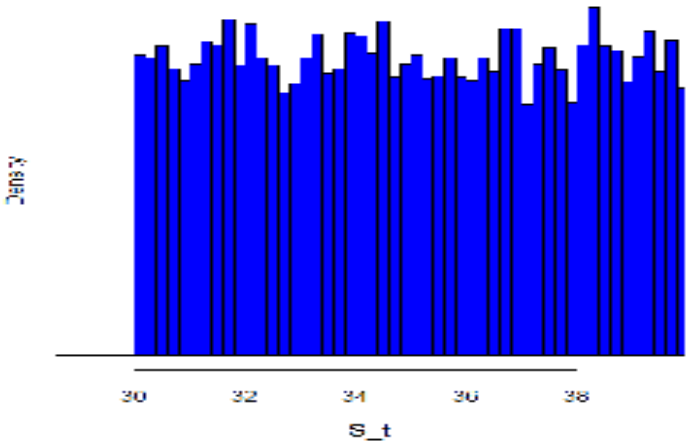


Figure: The boxplots summarize the distribution of $y^{(m)}(x^n)$'s, $m = 1, \ldots, M = 1600$. The dots indicate the batch means $\bar{y}(x^n)$ which are exactly interpolated by the two meta-models. $\mathcal{Z}' = \{30, 32, 34, 35, 36, 38\}$.

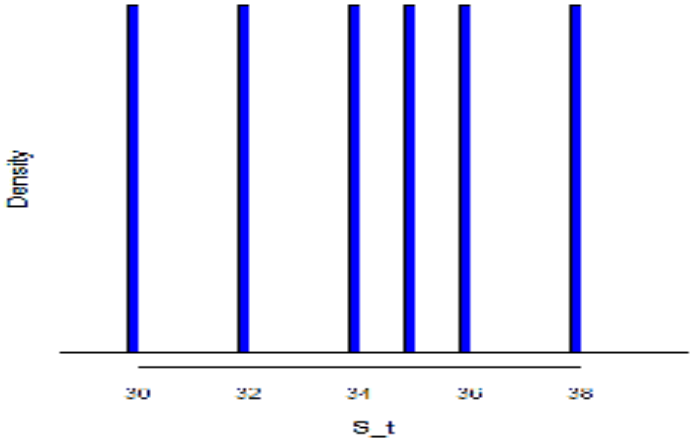
Regression Designs



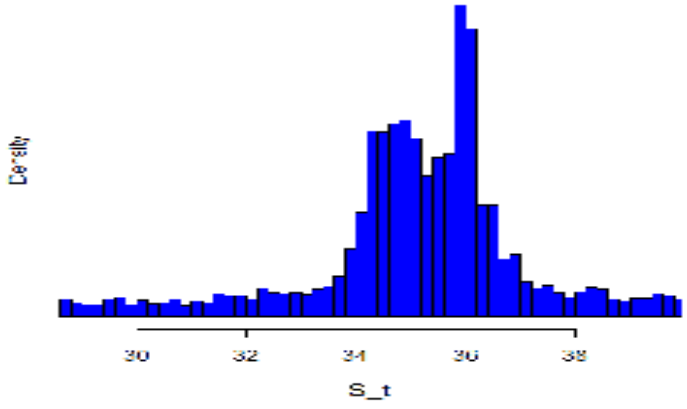
Based on $S_t|S_0$



Uniform in $[30, 40]$



Monte Carlo forest



Adaptive Grid

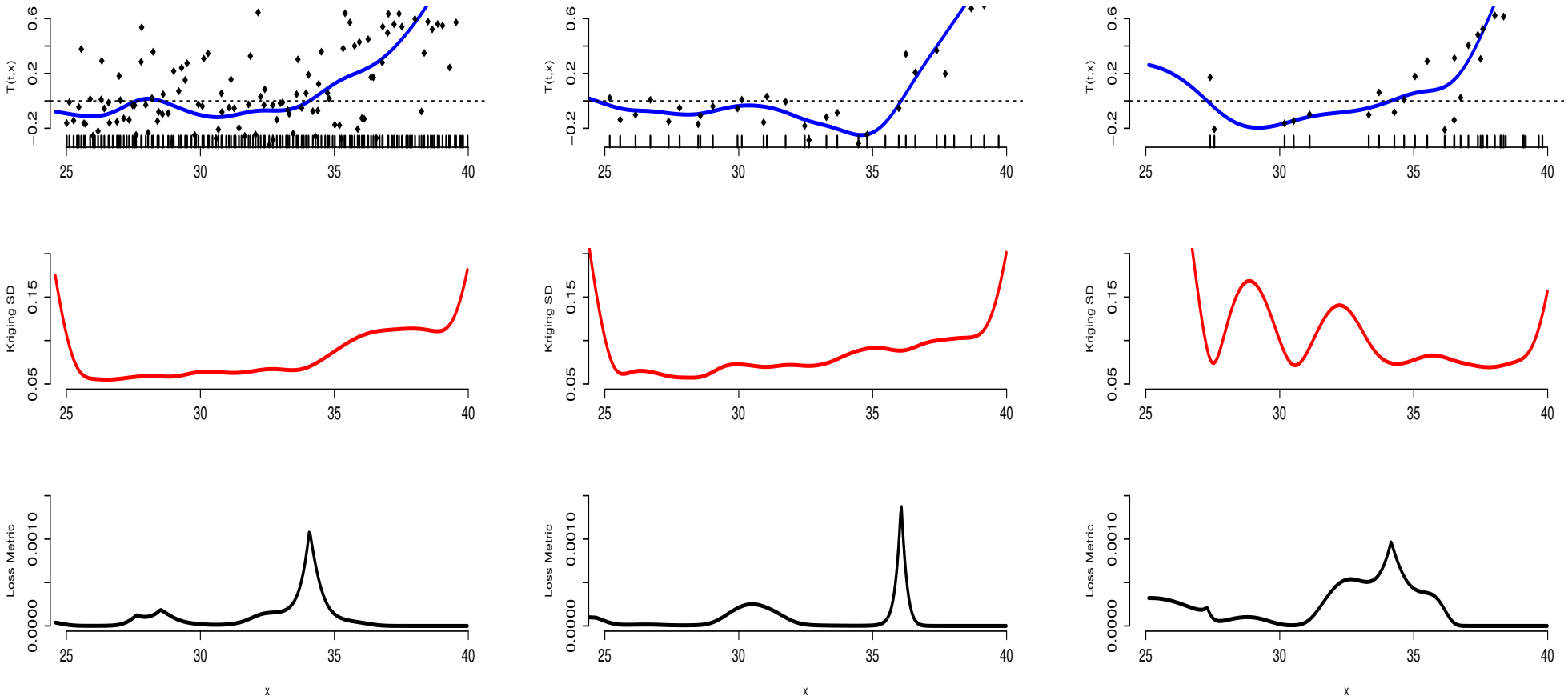
Experimental Design

- The meta-model should learn $C(t, \cdot)$ – $\mathcal{Z}$ should cover the domain $\mathcal{X}$
- Space-filling designs – lattice, low-discrepancy (Sobol)
- **LHS** Latin Hypercube sampling: random space-filling
- User must specify the effective $\mathcal{X}'$ (typically a rectangle)

Experimental Design

- The meta-model should learn $C(t, \cdot)$ – $\mathcal{Z}$ should cover the domain $\mathcal{X}$
- Space-filling designs – lattice, low-discrepancy (Sobol)
- **LHS** Latin Hypercube sampling: random space-filling
- User must specify the effective $\mathcal{X}'$ (typically a rectangle)
- The design should reflect the underlying (X_t)
- **Empirical** sampling: $\mathcal{Z}$ is constructed by drawing from X_t
- Automatically has the right "shape"
- This is the standard approach. Sensitive to X_0 (e.g. OTM Puts)

(Optimal Design is NP-Hard so heuristics are common)



LHS $M = 20, N' = 150$ LHS $M = 100, N' = 30$ Emp $M = 100, N' = 30$

Figure: Three different designs for fitting a kriging metamodel of the continuation value. *Top* panels show the fitted $\hat{T}(t, \cdot)$ as well as the distinct design sites $x^{1:N'}$. *Middle* panels plot the corresponding surrogate standard deviation $v(x)$. *Bottom* panels display the loss metric $\ell(x; \mathcal{Z})$.

Adaptive Design

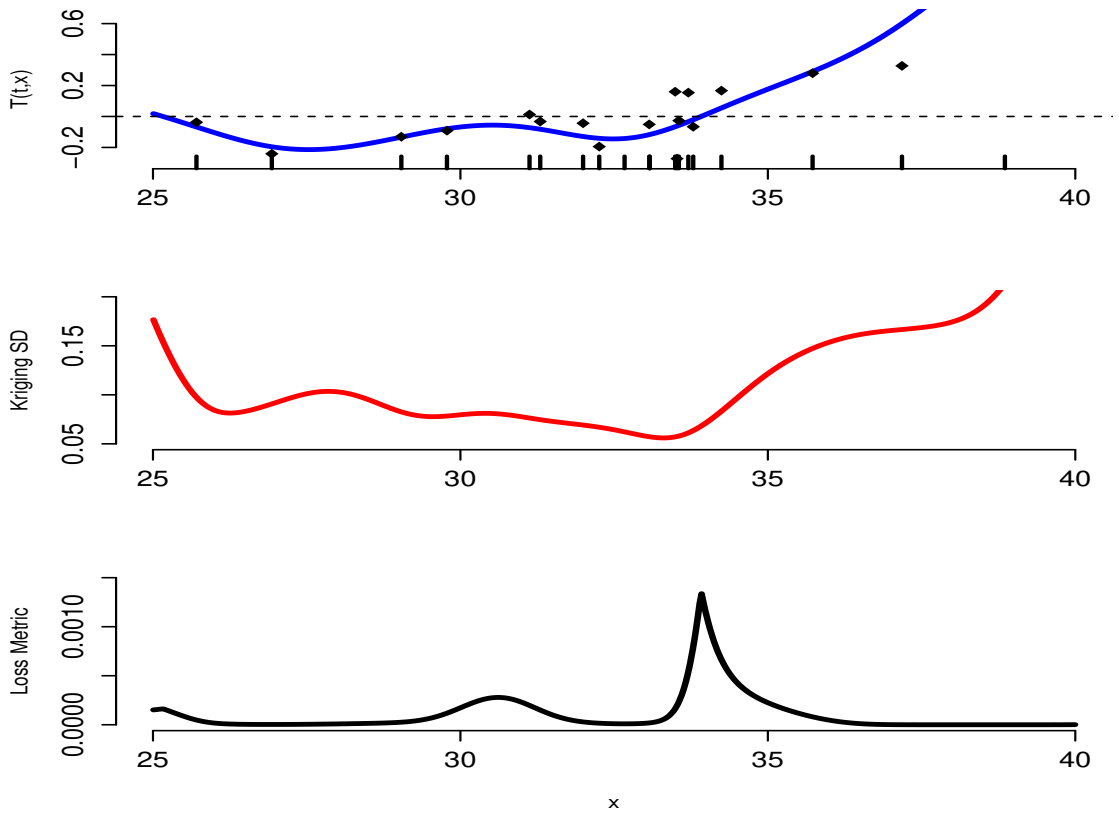
- Recall that aim to learn the sign of $T(t, \cdot)$
- Gradually grow $\mathcal{Z}^{(k)}$, $k = N_0, \ldots, N$
- Add new locations greedily according to acquisition function $x^{k+1} = \arg\max EI_k(x)$
- Favor points where $m^{(k)}(x) \simeq 0$ (close to zero-contour) or $v^{(k)}(x)$ is large (reduce uncertainty)
- Loss from making the wrong stopping decision at (t, x) is

$$\ell(x; \mathcal{Z}) := \int_{\mathbb{R}} |y - h(t,x)| 1_{\{m(x)<h(t,x)<y \bigcup\ y<h(t,x)<m(x)\}} \mathcal{M}_x(dy)$$

- Analytic expression for $EI_k(x) := \mathbb{E}[\ell^{(k)}(x) - \ell^{(k+1)}(x) | \mathcal{Z}^{(k)}, x^{k+1} = x]$
- ZC-SUR strategy: maximizes stepwise expected reduction in loss
- See Gramacy-L. (SIFIN 2015)

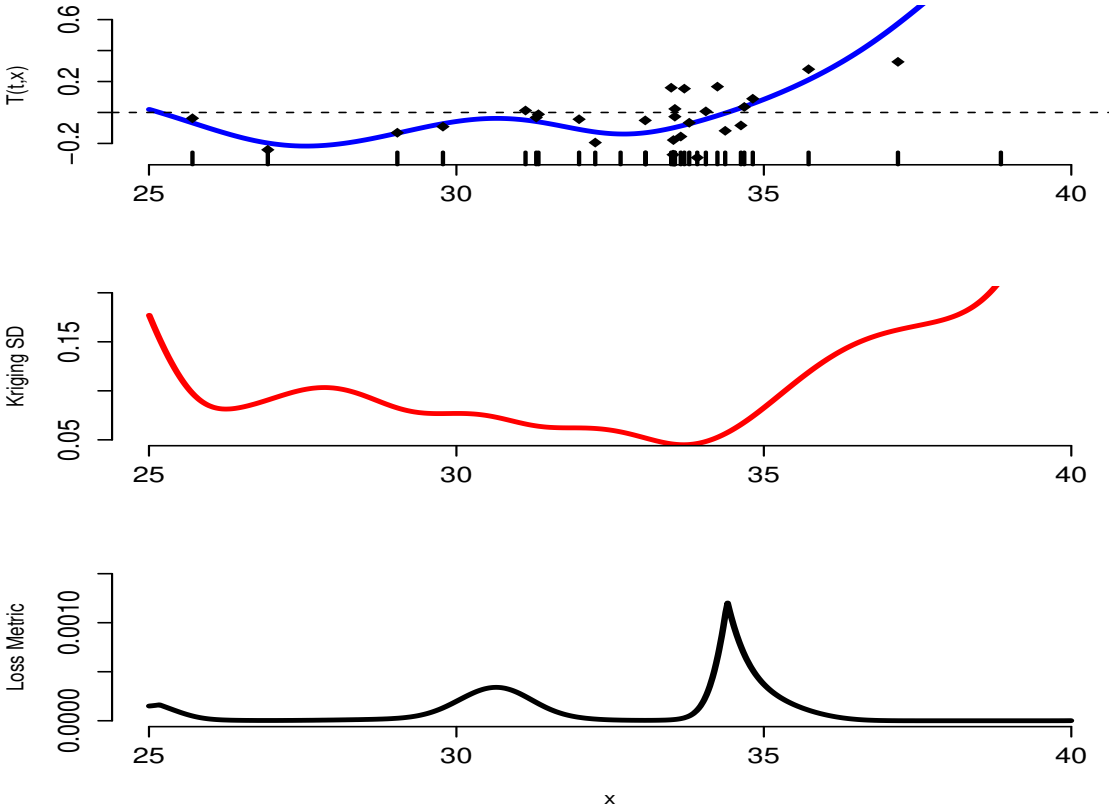
Sequential Design: $K = 20$

Initialize with a LHS design $\mathcal{Z}^{(20)}$



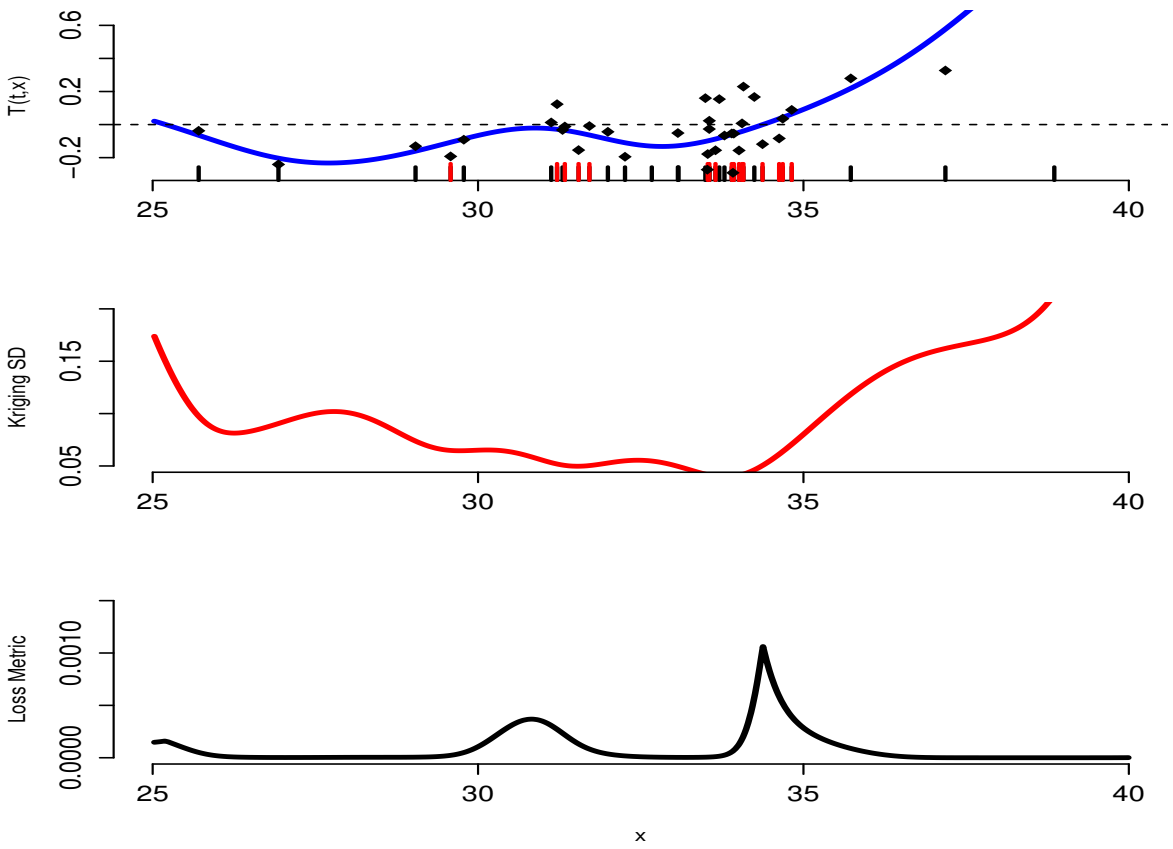
Sequential Design: $K = 30$

Zoom to the stopping boundary



Sequential Design: $K = 40$

Prefer regions that are more likely for X_t



Optimal Stopping for a 2D Stoch Vol Model

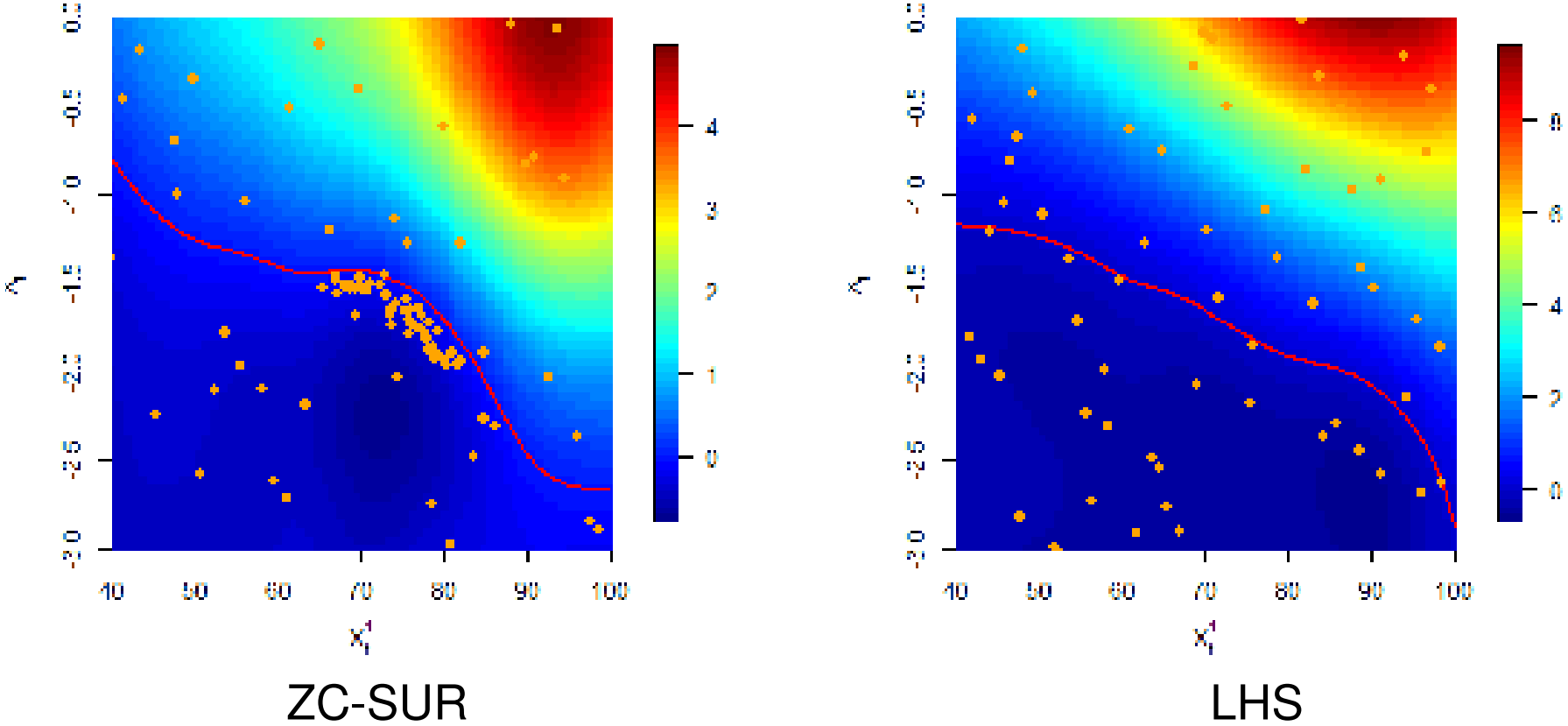


Figure: Adaptive vs LHS designs. Bermudan Put $e^{-rt}(100 - X_1)_+$ with a Heston SV model. Both designs used $N = 10000$ simulations. Color-coded according to $T(t, x)$; contour indicates the stopping boundary.

Comparison in 1-D GBM Put

Batch Size	LHS Spline	LHS Kriging	Emp Kriging	Seq Kriging
$M = 3$	2.306	2.304	2.306	2.303
$M = 8$	2.306	2.306	2.308	2.305
$M = 20$	2.292	2.305	2.286	2.295
$M = 50$	2.302	2.303	2.302	2.309
$M = 100$	2.302	2.303	2.304	2.311
$M = 250$	2.304	2.304	2.303	2.309

Table: Performance of different DoE approaches to RMC in the 1-D Bermudan Put setting, $h(t, x) = e^{-rt}(40 - x)_+$. All methods utilize $|\mathcal{Z}_t| = 3000$. The LHS input space was $\widetilde{\mathcal{X}} = [25, 40]$. Results are based on averaging 100 runs of each method, and evaluating $V(0, X_0)$ on a fixed out-of-sample database of $N_{out} = 50,000$ scenarios.

2D Examples

	Method	$\hat{V}(0, X_0)$	(StDev.)	#Sims	Time
Brockwell Rhambarat SV5					
LSM	$N = 5 \cdot 10^4$	15.98	(0.04)	$2.5 \cdot 10^6$	24
LSM	$N = 1.25 \cdot 10^5$	16.38	(0.03)	$6.25 \cdot 10^6$	52
LHS km	$N = 2500$	16.07	(0.16)	$1.07 \cdot 10^6$	25
LHS km	$N = 10000$	16.48	(0.06)	$4.8 \cdot 10^6$	168
SUR km	$N = 4000$	16.42	(0.11)	$1.67 \cdot 10^6$	65
Agrawal, Juneja and Sircar					
LSM	$N = 5 \cdot 10^4$	18.63	(0.03)	$1.0 \cdot 10^6$	25
LSM	$N = 1.25 \cdot 10^5$	18.81	(0.02)	$2.5 \cdot 10^6$	60
LHS km	$N = 2500$	18.79	(0.04)	$0.20 \cdot 10^6$	11
LHS km	$N = 10000$	18.88	(0.02)	$0.81 \cdot 10^6$	53
SUR km	$N = 4000$	18.86	(0.02)	$0.35 \cdot 10^6$	64
SUR km	$N = 10000$	18.90	(0.01)	$0.80 \cdot 10^6$	103

Kriging Performance

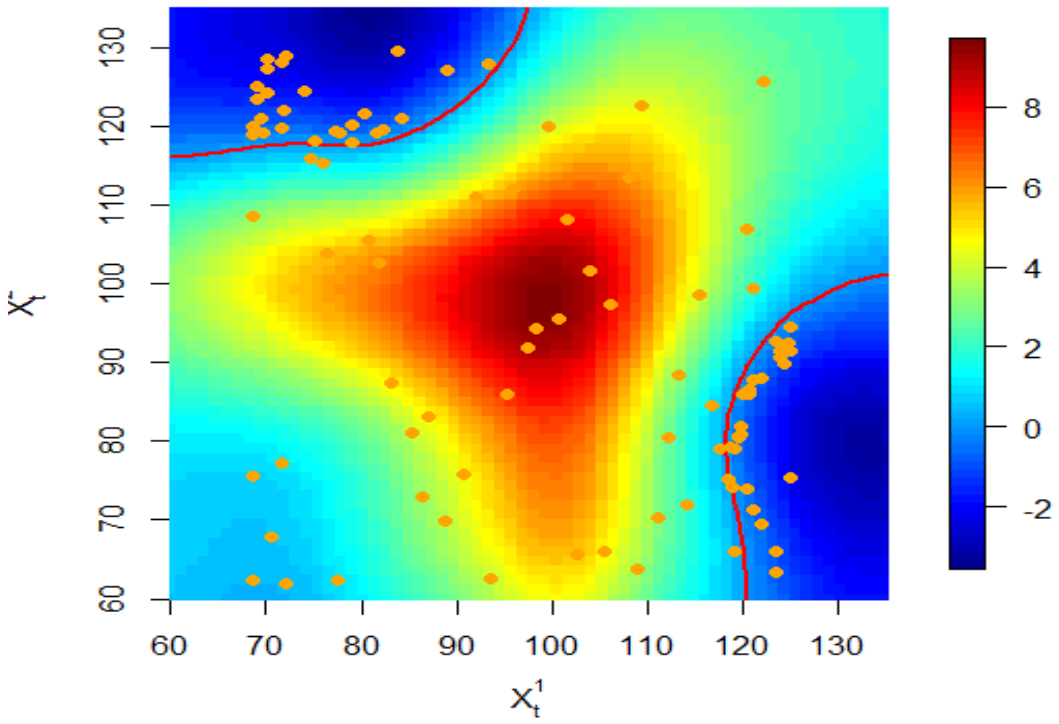
- Kriging appears very promising as a flexible, off-the-shelf regression framework
- Gives smooth, non-parametric fits for $C(t, \cdot)$
- Emphasizes the interpolation vs. smoothing aspect of metamodeling
- Easy implementation via public `R` packages
- Order of magnitude slower than a Least-Squares model (not important if simulations are the bottleneck)

Experimental Design Performance

- Batching has minimal effect on performance (but major effect on speed)
- (Random) space-filling designs allow to reduce size of design by a factor of 3-5
- Compared to standard LSM this reduces simulation budget by 25-50%
- Adaptive sequential designs
  - Yield further substantial savings (up to an order of magnitude)
  - Significant regression overhead as must fit multiple (kriging) metamodels
  - Worth it if in high dimensions $d > 3$ and simulation budget is very constrained

Bermudan Max Call $(\max(X_1, X_2) - K)_+$

- It is often nontrivial to specify a good domain $\mathcal{X}'$
- This is the advantage of the empirical design
- Sequential designs really begin to shine



The Future

- Finding conditional expectations is a metamodeling problem
- Can squeeze (a lot) of extra efficiency by jointly targeting experimental design + regression
- Lots more opportunities in this direction
- e.g. adapt to the BSDE numerical algorithms (Bender, Gobet)
- Also more general control problems (optimal switching, sequential games, et cetera)

RMC = +

The Future

- Finding conditional expectations is a metamodeling problem
- Can squeeze (a lot) of extra efficiency by jointly targeting experimental design + regression
- Lots more opportunities in this direction
- e.g. adapt to the BSDE numerical algorithms (Bender, Gobet)
- Also more general control problems (optimal switching, sequential games, et cetera)

RMC = Regression + Stochastic Grid

The Future

- Finding conditional expectations is a metamodeling problem
- Can squeeze (a lot) of extra efficiency by jointly targeting experimental design + regression
- Lots more opportunities in this direction
- e.g. adapt to the BSDE numerical algorithms (Bender, Gobet)
- Also more general control problems (optimal switching, sequential games, et cetera)

RMC = Regression + Stochastic Grid

THANK YOU!

References

- JPC. KLEIJNEN *Design and Analysis of Simulation Experiments*, 2nd Edition, Springer, 2015.
- CKI. WILLIAMS AND CE. RASMUSSEN *Gaussian processes for machine learning*, MIT Press, 2006.
- M. KOHLER, *A review on regression-based Monte Carlo methods for pricing American options*, in Recent Developments in Applied Probability and Statistics, Springer, 2010, pp. 37–58.
- BECT, J., GINSBOURGER, D., LI, L., PICHENY, V., AND VAZQUEZ, E., *Sequential design of computer experiments for the estimation of a probability of failure*, Statistics and Computing, 22(3), 773–793, (2012).
- R. GRAMACY AND M. LUDKOVSKI *Sequential Design for Optimal Stopping Problems* SIFIN 6(1), 2015, pp. 748–775
- M. LUDKOVSKI *Kriging Metamodels for Bermudan Option Pricing*, preprint, 2015 arXiv:1509.02179
- R. HU AND M. LUDKOVSKI *Sequential Design for Ranking Response Surfaces*, preprint, 2015. arXiv:1509.00980

Require: N – number of initial grid points

$\mathfrak{S}_T \leftarrow \mathcal{X}$
for $t = T-1, T-2, \ldots, 0$ **do**
  $k \leftarrow 0$
  Generate an initial grid $\{x_t^{1:N}\}$, and corresponding classifier $\mathfrak{S}_t^{(0)}$
  while the current grid needs refining **do**
    $k \leftarrow k+1$
    Generate new grid point(s) $\{x_t^{(k),n'}\}$ $n' = 1, \ldots, N^{(k)}$
    Simulate forward trajectories $x_{t+1:T}^{(k),1:N^{(k)}}$. Using $\hat{\mathfrak{S}}_{t+1:T}$ find $y^{(k),1:N^{(k)}}$
    Update the classifier to $\mathfrak{S}_t^{(k)}$ using new samples $(x_t^{(k)}, y^{(k)})^{1:N^{(k)}}$
    (Update the classifiers $\hat{\mathfrak{S}}_{t+1:T-1}$ using $x_{t+1:T-1}^{(k),1:N^{(k)}}$)
    Save the overall grid $\{x_t\} \leftarrow \{x_t\} \cup \{x_t^{(k),1:N^{(k)}}\}$
  end while
  Generate final estimate of the classifier at time step t, $\hat{\mathfrak{S}}_t$
end for
Simulate forward trajectories $X_{0:T}^n$ from $X^n = x_0$ using $\hat{\mathfrak{S}}_{0:T}$
return $V(0, x_0) \simeq \frac{1}{N} \sum_{n=1}^{N} h_{\tau^n}(X_{\tau^n}^n)$
return Estimated policy $\{\hat{\mathfrak{S}}_{0:T}\}$.

Sequential Design for Regression Monte Carlo

Generate the grids adaptively online. [Vanilla RMC re-uses the grids during forward simulations. We regenerate fresh paths at each step]

- Start with initial grid $\mathcal{Z}^{(n_0)} \equiv \{x_t^{1:n_0}\}$
- Build initial approximation $\mathfrak{S}_t^{(n_0)}$
- LOOP for $k = n_0, n_0 + 1, \ldots$
  - Identify promising regions
  - Generate new data $\{x_{t:T}^{k+1}\}$ and costs-to-go $y_t^{k+1} = h(x_{\tau^{k+1}}^k) - h(x_t^{k+1})$.
  - Update the fit to $\mathfrak{S}_t^{(k+1)}$
- END LOOP
- Repeat above at each time-step $t = T - 1, \ldots, 1$