Surrogate-based Simulation Optimization

L. Jeff Hong (Fudan University) Xiaowei Zhang (University of Hong Kong)

2021 INFORMS TutORials

Simulation Optimization

$$\max_{x \in \mathcal{X}} \{ f(x) := \mathbb{E}[F(x)] \}$$

- $f(x)$ can only be evaluated via noisy, expensive samples
  - E.g., the expected profit of a complex inventory system
- SO is concerned with designing sampling algorithms to allocate the simulation budget to find a good solution
- Fu and Henderson (2017): history of SO

SO Problem Types

- Ranking and selection (R&S)
  - $\mathcal{X}$ is a set of a relatively small number of feasible solutions with no inherent ordering defined
  - E.g., system configurations with regard to what redundant components to use to design a reliable system
  - Hong, Fan, and Luo (2021)

SO Problem Types

- Ranking and selection (R&S)
    - $\mathcal{X}$ is a set of a relatively small number of feasible solutions with no inherent ordering defined
    - E.g., system configurations with regard to what redundant components to use to design a reliable system
    - Hong, Fan, and Luo (2021)
- Discrete optimization via simulation (DOvS)
    - $\mathcal{X}$ is integer-ordered: $\mathcal{X} \subseteq \mathbb{Z}^d$
    - E.g., inventory decisions (number of units to order) for d products
    - Hong, Nelson, and Xu (2015)

SO Problem Types

- Ranking and selection (R&S)
  - $\mathcal{X}$ is a set of a relatively small number of feasible solutions with no inherent ordering defined
  - E.g., system configurations with regard to what redundant components to use to design a reliable system
  - Hong, Fan, and Luo (2021)

- Discrete optimization via simulation (DOvS)
  - $\mathcal{X}$ is integer-ordered: $\mathcal{X} \subseteq \mathbb{Z}^d$
  - E.g., inventory decisions (number of units to order) for d products
  - Hong, Nelson, and Xu (2015)

- Continuous SO
  - $\mathcal{X} \subseteq \mathbb{R}^d$
  - Algorithms for the continuous setting can often be applied to DOvS

Strategies for Solving Continuous SO

- Sample average approximation (Kim, Pasupathy, and Henderson, 2015)
- Stochastic approximation (Chau and Fu, 2015)
- Random search (Andradóttir, 2015)

Strategies for Solving Continuous SO

- Sample average approximation (Kim, Pasupathy, and Henderson, 2015)
- Stochastic approximation (Chau and Fu, 2015)
- Random search (Andradóttir, 2015)
- Surrogate-based methods (Barton and Meckesheimer, 2006)
    - Flexible to capture complex surface shapes
    - Capable to predict surface values where no simulation samples are observed

Outline

1. Typical Surrogates
2. Locally Convergent Algorithms
3. Globally Convergent Algorithms
4. Computation for Large Datasets
5. Concluding Remarks

Surrogates

- a.k.a. metamodel: an approximation to the response surface (simulation input-output relationship)
- Mitigate the computational burden of running simulation experiments
- Any supervised learning model may, in principle, be used

Criteria for Good Surrogates

- Simple structure, require no “big data” to fit
    - Simulation samples are expensive

Criteria for Good Surrogates

- Simple structure, require no “big data” to fit
    - Simulation samples are expensive
- Computationally easy to fit
    - Often need to be updated as more samples become available

Criteria for Good Surrogates

- Simple structure, require no “big data” to fit
  - Simulation samples are expensive
- Computationally easy to fit
  - Often need to be updated as more samples become available
- Yield a predictor in explicit form
  - Efficient computation of predictions
  - Facilitate theoretical analysis
  - Easy optimization of the surrogate

Typical Surrogates

- Low-order polynomials
- Linear basis function models
- Gaussian processes (GPs)

Polynomials

- The number of terms explodes in multiple dimensions
- Polynomials with orders higher than two are seldom used

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j + \sum_{j=1}^{d} \sum_{k=1}^{d} \beta_{jk} x_j x_k,$$

- Suitable for approximating the surface in a localized region
- $\beta_0, \beta_j, \beta_{jk}$ can be estimated via ordinary least squares (OLS)

Linear Basis Function Models

$$f(x) = \boldsymbol{\beta}^\intercal \phi(x) = \sum_{k=1}^{p} \beta_k \phi_k(x)$$

- $\phi(x)$: e.g., truncated power basis and radial basis
- $\boldsymbol{\beta}$ can be also estimated via OLS:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (\bar{y}_i - \boldsymbol{\beta}^\intercal \phi(x_i))^2 \\ &= \Phi^\intercal (\Phi \Phi^\intercal)^{-1} \bar{y},\end{aligned}$$

  where Φ is the n-by-p matrix with the i-th row being $\phi(x_i)^\intercal$

- The prediction is given by

$$\hat{f}(x) = \hat{\boldsymbol{\beta}}^\intercal \phi(x) = (\Phi \phi(x))^\intercal (\Phi \Phi^\intercal)^{-1} \bar{y}$$

Gaussian Processes

- Generalization of multivariate normal random variables
- Fully characterized by
  - Mean function $\mu : \mathcal{X} \mapsto \mathbb{R}$
  - Covariance function (kernel) $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$

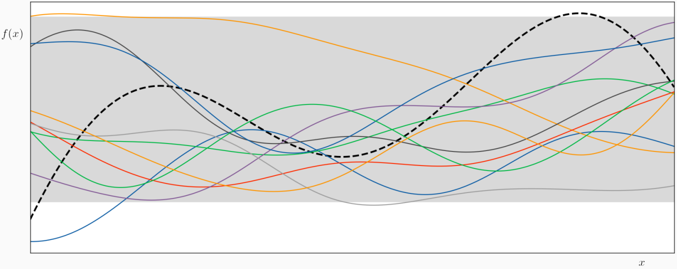
Gaussian Processes

- Generalization of multivariate normal random variables
- Fully characterized by
  - Mean function $\mu : \mathcal{X} \mapsto \mathbb{R}$
  - Covariance function (kernel) $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$

- Bayesian approach: assume prior distribution of f is $\mathrm{GP}(\mu, K)$

$$(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) \sim \mathrm{MVNormal}(\boldsymbol{\mu}, \mathbf{K}),$$

where $\boldsymbol{\mu} = (\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_n))^\intercal$ and $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_{i'}))_{i,i'=1}^{n}$



Mean Functions

- Encode one's prior knowledge about the overall shape of f
- Set $\mu(x) \equiv c$ for some constant c: common practice
- Set $\mu(x) = \boldsymbol{\beta}^\intercal \boldsymbol{\phi}(x)$, where $\boldsymbol{\phi}(x)$ is a vector of known basis functions and $\boldsymbol{\beta}$ is a vector of hyperparameters of compatible dimension
- Set $\mu(x)$ to be a function derived from a simplified model of the same stochastic system

Kernels

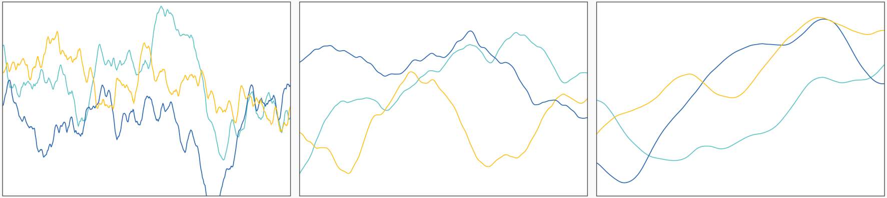
- Gaussian kernels

$$K_{\text{Gaussian}}(x, x') = \tau^2 \exp\left(-\frac{\|x - x'\|^2}{2\eta^2}\right)$$

- Matérn kernels

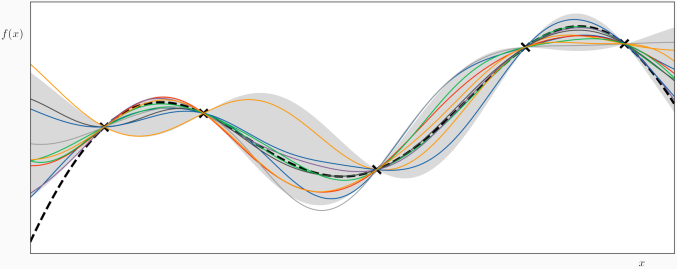
$$K_{\text{Matern}}(x, x'; \nu) = \frac{\tau^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}\|x - x'\|}{\eta}\right)^{\nu} \mathrm{K}_{\nu}\left(\frac{\sqrt{2\nu}\|x - x'\|}{\eta}\right)$$

  - ν: smoothness parameter, usually set to be $1/2, 3/2, 5/2, \ldots$



GP Regression

- Assume simulation noise is Gaussian with known variance
- Posterior of f is also a GP



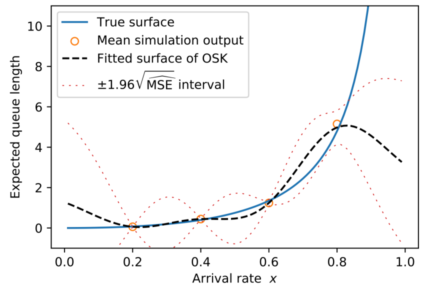
- Use the posterior mean for prediction

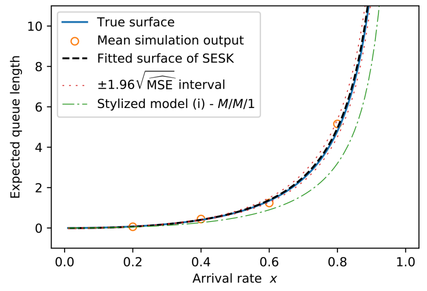
$$\hat{f}(x) = \mu(x) + \boldsymbol{k}(x)^{\intercal}(K + \Sigma)^{-1}(\bar{y} - \boldsymbol{\mu}),$$

where $\boldsymbol{k}(x) = (K(x, x_1), \ldots, K(x, x_n))^{\intercal}$ and Σ is a diagonal matrix

Enhancing Surrogates with Stylized Models

- The main features of a complex system may be captured by a stylized model that yields an analytical expression, say $\psi(x)$
    - Complicated queueing network v.s. independent nodes
- Use ψ in linear basis function models or in the mean functions of GPs
- Shen, Hong, and Zhang (2018); Lin, Matta, and Shanthikumar (2019)





Enhancing Surrogates with Gradient Observations

- Given a sample of $f(x)$, an estimate of $\nabla f(x)$ can often be computed with a negligible extra cost
  - Infinitesimal perturbation analysis or the likelihood ratio method (L'Ecuyer, 1990)
- Surrogate for f induces surrogate for ∇f: jointly estimate the parameters
- Chen, Ankenman, and Nelson (2013); Fu and Qu (2014); Qu and Fu (2014); Huo, Zhang, and Zheng (2018)

Part II: Locally Convergent SO Algorithms

Global Convergence v.s. Local Convergence

- Global: converge to a global optimal solution
- Local: converge to a local optimal solution or a stationary point

Global Convergence v.s. Local Convergence

- Global: converge to a global optimal solution
- Local: converge to a local optimal solution or a stationary point

- Global convergence requires exploring the entire feasible region in the limit
- Local convergence only needs to explore part of the feasible region
    - Advantageous when the simulation budget is very limited

Response Surface Methodology (RSM)

- Stage 1
  - Run a number of experiments in a local region of the current solution
  - Fit a first-order polynomial: $f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$
  - Find a better solution along the ascent direction $\nabla f(\mathbf{x}) = (\beta_1, \ldots, \beta_d)^\intercal$
  - Repeat the process until first-order polynomials are no longer adequate

Response Surface Methodology (RSM)

- Stage 1
  - Run a number of experiments in a local region of the current solution
  - Fit a first-order polynomial: $f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$
  - Find a better solution along the ascent direction $\nabla f(x) = (\beta_1, \dots, \beta_d)^\intercal$
  - Repeat the process until first-order polynomials are no longer adequate

- Stage 2
  - Fit a second-order polynomial: $f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j + \sum_{j=1}^{d} \sum_{k=1}^{d} \beta_{jk} x_j x_k$
  - Maximize the polynomial

Heuristic Nature of RSM

- Procedure involves human judgement
  - In each iteration, the local region over which one optimizes the surrogate is determined based human experience
  - The transition between first- and second-order surrogates relies on human experience
- Typically used for (very) expensive simulation/real experiments, so large-sample properties such as convergence are not clear

Heuristic Nature of RSM

- Procedure involves human judgement
  - In each iteration, the local region over which one optimizes the surrogate is determined based human experience
  - The transition between first- and second-order surrogates relies on human experience
- Typically used for (very) expensive simulation/real experiments, so large-sample properties such as convergence are not clear

- Chang, Hong, and Wan (2013): use the trust-region method to address the heuristic nature of RSM

Stochastic Trust-Region Response-Surface Method (STRONG)

- Iteration k: $\mathbf{x}_k$ is the current solution, Δ_k is the size of the trust region

(1) Fit a surrogate $r_k(\mathbf{x})$ around $\mathbf{x}_k$

- If $\Delta_k \geq \tilde{\Delta}$, $r_k(\mathbf{x})$ is a first-order polynomial
- Otherwise, $r_k(\mathbf{x})$ is a second-order polynomial

(2) Solve $\mathbf{x}_k^* = \text{argmax}\{r_k(\mathbf{x}) : \mathbf{x} \in \mathcal{B}(\mathbf{x}_k, \Delta_k)\}$

(3) Simulate a number of observations at $\mathbf{x}_k^*$ and estimate $f(\mathbf{x}_k^*)$;

(4) Conduct two tests to update $\mathbf{x}_{k+1}$ and Δ_{k+1}

- One tests whether $\mathbf{x}_k^*$ is significantly better than $\mathbf{x}_k$
- The other tests whether the surrogate works well

Two Tests

- If $\mathbf{x}_k^*$ is not significantly better than $\mathbf{x}_k$, then set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and decrease $\boldsymbol{\Delta}_k$
- If $\mathbf{x}_k^*$ is significantly better than $\mathbf{x}_k$, then compute the ratio between the observed and predicted improvements

$$\rho_k = \frac{\bar{f}_k(\mathbf{x}_k^*) - \bar{f}_k(\mathbf{x}_k)}{r_k(\mathbf{x}_k^*) - r_k(\mathbf{x}_k)}$$

  - If ρ_k is large (surrogate works well), then set $\mathbf{x}_{k+1} = \mathbf{x}_k^*$ and increase $\boldsymbol{\Delta}_k$
  - If ρ_k is small (surrogate works poorly), then set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and decrease $\boldsymbol{\Delta}_k$
  - Otherwise: set $\mathbf{x}_{k+1} = \mathbf{x}_k^*$ and keep $\boldsymbol{\Delta}_k$

Rigorous Nature of STRONG

- Procedure involves no human judgement
  - The local region is the trust region and its size is updated based on two tests
  - The transitions between first- and second-order surrogates are based on Δ_k
- The STRONG algorithm converges to a stationary point

Part III: Globally Convergent SO Algorithms

Static Experimental Designs

- Select design points $\mathcal{X} = \{x_1, \ldots, x_n\}$ before any simulation
  - Primary design principle: cover the design space as much as possible
  - E.g., space-filling designs (Santner, Williams, and Notz, 2003)
- Run simulation at each design point, possibly multiple times
- Fit a surrogate with the observations
- Optimize the predicted surface $\hat{f}(x)$—a deterministic function—with any numerical optimization algorithms

Sequential Experimental Designs

- Design points are selected one at a time after each new sample is obtained
- Each new design point is selected based on
    - The updated surrogate reflecting the previous observations
    - Certain criterion that balances exploration and exploitation

Sequential Experimental Designs

- Design points are selected one at a time after each new sample is obtained
- Each new design point is selected based on
  - The updated surrogate reflecting the previous observations
  - Certain criterion that balances exploration and exploitation

- GPs are the favorite because of the need for uncertainty quantification
- Closely related to “Bayesian optimizaiton” (Shahriari et al., 2016; Frazier, 2018)

General Structure

(1) Impose a GP prior on f

(2) Select the next batch of design points subject to a prescribed “criterion”

(3) Run simulation at each of the newly selected design points

(4) Update the GP posterior given the new observations of f

(5) Repeat Steps (2)–(4) until the simulation budget is exhausted

(6) Optimize the posterior mean function and return the optimum

Knowledge Gradient (KG)

- Given $\mathcal{D}_n = \{(x_i, y_i) : i = 1, \ldots, n\}$, the posterior $f|\mathcal{D}_n \sim \mathrm{GP}(\mu_n, K_n)$
    - Updating equations for μ_n and K_n are in closed form
- Select $x_{n+1} = \mathrm{argmax}_{x \in \mathcal{X}} \mathrm{KG}_n(x)$

$$\mathrm{KG}_n(x) := \mathbb{E}\Big[\underbrace{\max_{v \in \mathcal{X}} \mu_{n+1}(v) - \max_{v \in \mathcal{X}} \mu_n(v)}_{\text{increment in belief about } \max_v f(v)} \Big| \mathcal{D}_n, x_{n+1} = x\Big]$$

- Conditional on $\{\mathcal{D}_n, x_{n+1} = x\}$: $\mu_{n+1}(v)$ has a normal distribution that depends on x, while $\mu_n(v)$ is a constant

Maximizing $\mathrm{KG}_n(x)$

- Scott, Frazier, and Powell (2011): discretization

$$\widetilde{\mathrm{KG}}_n(x) := \mathbb{E}\Big[\max_{1\le i\le n+1} \mu_{n+1}(x_i) - \max_{1\le i\le n+1} \mu_n(x_i) \Big| \mathcal{D}_n, x_{n+1} = x\Big]$$

- Wu and Frazier (2016): stochastic approximation

$$\begin{aligned}\nabla_x \mathrm{KG}_n(x) &= \nabla_x \mathbb{E}\Big[\max_{v\in\mathcal{X}} \mu_{n+1}(v) - \max_{v\in\mathcal{X}} \mu_n(v) \Big| \mathcal{D}_n, x_{n+1} = x\Big] \\ &= \mathbb{E}\Big[\nabla_x \max_{v\in\mathcal{X}} \mu_{n+1}(v) \Big| \mathcal{D}_n, x_{n+1} = x\Big]\end{aligned}$$

Upper Confidence Bound (UCB)

- A celebrated class of methods for multi-armed bandit (MAB) problems
  - MAB: online, maximize cumulative reward
  - R&S: offline, maximize terminal reward
- GP-UCB: Srinivas et al. (2012) generalize UCB to the continuous setting
- Selecte $x_{n+1} = \mathrm{argmax}_{x \in \mathcal{X}} \mathrm{UCB}_n(x)$

$$\mathrm{UCB}_n(x) := \mu_n(x) + \sqrt{\gamma_n K_n(x,x)},$$

where $\gamma_n > 0$ is a tuning parameter that varies as a function of n

Upper Confidence Bound (UCB)

- A celebrated class of methods for multi-armed bandit (MAB) problems
  - MAB: online, maximize cumulative reward
  - R&S: offline, maximize terminal reward
- GP-UCB: Srinivas et al. (2012) generalize UCB to the continuous setting
- Selecte $x_{n+1} = \mathrm{argmax}_{x \in \mathcal{X}} \mathrm{UCB}_n(x)$

$$\mathrm{UCB}_n(x) := \mu_n(x) + \sqrt{\gamma_n K_n(x, x)},$$

  where $\gamma_n > 0$ is a tuning parameter that varies as a function of n
- MAB: $\gamma_n \asymp \ln(n)$
- SO: conceivably larger due to emphasis on the terminal reward
  - More exploration is needed

GP-UCB v.s. KG

	GP-UCB	KG
Max. acquisition function	easy	hard
Tuning parameter	γ_n	n.a.

GP-based Search (GPS)

- Given $\mathcal{D}_n = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$, the posterior $f | \mathcal{D}_n \sim \text{GP}(\mu_n, K_n)$
- Use probability of improvement to devise a sampling distribution

$$h_n(\mathbf{x}) \propto \Pr(\text{Normal}(\mu_n(\mathbf{x}), K_n(\mathbf{x}, \mathbf{x})) > f_n^*),$$

  where f_n^* is the current estimated optimal value
- Draw the next batch of design points from $h_n(\cdot)$

GP-based Search (GPS)

- Given $\mathcal{D}_n = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$, the posterior $f|\mathcal{D}_n \sim \mathrm{GP}(\mu_n, K_n)$
- Use probability of improvement to devise a sampling distribution

$$h_n(\mathbf{x}) \propto \Pr(\mathrm{Normal}(\mu_n(\mathbf{x}), K_n(\mathbf{x}, \mathbf{x})) > f_n^*),$$

  where f_n^* is the current estimated optimal value
- Draw the next batch of design points from $h_n(\cdot)$
  - Sun, Hong, and Hu (2014): Markov chain Monte Carlo
  - Sun, Hu, and Hong (2018): Gaussian mixture approximation

GPS v.s. KG/GP-UCB

	GPS	KG/GP-UCB
Criterion	sampling distribution	acquisition function
# points determined	batch	one at a time

Part IV: Computation for Large Datasets

Scalability Issue

- Many computations involve matrix inversion

$$\text{Linear basis function model:}\quad \hat{f}(x) = (\Phi\phi(x))^{\intercal}(\Phi\Phi^{\intercal})^{-1}\bar{y}$$

$$\text{GP regression:}\quad \hat{f}(x) = \mu(x) + k(x)^{\intercal}(K + \Sigma)^{-1}(\bar{y} - \boldsymbol{\mu})$$

- Surrogates become computationally challenging when n is large
    - Time complexity: $\mathcal{O}(n^3)$
    - Space complexity: $\mathcal{O}(n^2)$

Scalability Issue

- Many computations involve matrix inversion

$$\text{Linear basis function model:}\quad \hat{f}(x) = (\Phi\phi(x))^\intercal(\Phi\Phi^\intercal)^{-1}\bar{y}$$

$$\text{GP regression:}\quad \hat{f}(x) = \mu(x) + \boldsymbol{k}(x)^\intercal(K + \Sigma)^{-1}(\bar{y} - \boldsymbol{\mu})$$

- Surrogates become computationally challenging when n is large
  - Time complexity: $\mathcal{O}(n^3)$
  - Space complexity: $\mathcal{O}(n^2)$
- Vast literature on GP approximations
  - Nyström method
  - Random features

Low-rank Approximations and Woodbury Formula

- **Goal:** approximate $(K + \Sigma)^{-1}$, where Σ is a diganoal matrix
- Consider a rank-m matrix of the form $\tilde{K} = UCV$, where $U \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{m \times m}$, and $V \in \mathbb{R}^{m \times n}$ with $m < n$
- Woodbury formula:

$$(K + \Sigma)^{-1} \approx (\tilde{K} + \Sigma)^{-1} = \Sigma^{-1} - \Sigma^{-1}U(\underbrace{C^{-1} + V\Sigma^{-1}U}_{m \times m})^{-1}V\Sigma^{-1}$$

Low-rank Approximations and Woodbury Formula

- **Goal:** approximate $(K+\Sigma)^{-1}$, where Σ is a diganoal matrix
- Consider a rank-m matrix of the form $\tilde{K} = UCV$, where $U \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times m}$, and $V \in \mathbb{R}^{m\times n}$ with $m < n$
- Woodbury formula:

$$(K+\Sigma)^{-1} \approx (\tilde{K}+\Sigma)^{-1} = \Sigma^{-1} - \Sigma^{-1}U(\underbrace{C^{-1} + V\Sigma^{-1}U}_{m\times m})^{-1}V\Sigma^{-1}$$

- Construct another kernel $\tilde{K}(\cdot,\cdot)$ that yields a low-rank $\tilde{K}$

Nyström Method

- Smola and Schölkopf (2000); Rudi et al. (2015); Lu et al. (2020)
- Select m design points out of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
  - $I = \{1, \ldots, n\}$
  - $A \subset I$ is the selected indices
- Let $\boldsymbol{k}_m(\boldsymbol{x}) = (K(\boldsymbol{x}_i, \boldsymbol{x}))_{i \in A}$ and $\boldsymbol{K}_{m,m} := (K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}))_{i \in A, i' \in A}$

$$\tilde{K}(\boldsymbol{x}, \boldsymbol{x}') := \boldsymbol{k}_m(\boldsymbol{x})^\intercal \boldsymbol{K}_{m,m}^{-1} \boldsymbol{k}_m(\boldsymbol{x}')$$

Nyström Method

- Smola and Schölkopf (2000); Rudi et al. (2015); Lu et al. (2020)
- Select m design points out of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
  - $I = \{1, \ldots, n\}$
  - $A \subset I$ is the selected indices
- Let $\boldsymbol{k}_m(\boldsymbol{x}) = (K(\boldsymbol{x}_i, \boldsymbol{x}))_{i \in A}$ and $\boldsymbol{K}_{m,m} := (K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}))_{i \in A, i' \in A}$

$$\tilde{K}(\boldsymbol{x}, \boldsymbol{x}') := \boldsymbol{k}_m(\boldsymbol{x})^\intercal \boldsymbol{K}_{m,m}^{-1} \boldsymbol{k}_m(\boldsymbol{x}')$$

- The covariance matrix associated with evaluating $\tilde{K}$ at $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is

$$\tilde{\boldsymbol{K}} = \boldsymbol{K}_{n,m} \boldsymbol{K}_{m,m}^{-1} \boldsymbol{K}_{m,n}$$

Posterior of $\mathrm{GP}(\mu, \tilde{K})$

$$\mathbb{E}[\tilde{f}(x)|\mathcal{D}_n] = \mu(x) + \boldsymbol{k}_m(x)^\intercal(\underbrace{K_{m,m} + K_{m,n}\Sigma^{-1}K_{n,m}}_{m\times m})^{-1}K_{m,n}\Sigma^{-1}(\bar{y} - \boldsymbol{\mu})$$

$$\mathrm{Cov}[\tilde{f}(x), \tilde{f}(x')|\mathcal{D}_n] = K(x, x') - \boldsymbol{k}_m(x)^\intercal(\underbrace{K_{m,m} + K_{m,n}\Sigma^{-1}K_{n,m}}_{m\times m})^{-1}\boldsymbol{k}_m(x')$$

- Both can be computed with time complexity $\mathcal{O}(m^2 n)$

Random Features

- Rahimi and Recht (2007): random Fourier features (RFF)
- Bochner's theorem: if K is stationary (e.g., Gaussian and Matérn), then

$$\begin{aligned} K(x, x') &= K(0,0) \int_{\mathbb{R}^d} e^{\mathrm{i}\boldsymbol{\omega}^\intercal (x - x')} \mathrm{p}(d\boldsymbol{\omega}) \\ &= K(0,0) \int_{\mathbb{R}^d} \cos\left(\boldsymbol{\omega}^\intercal (x - x')\right) \mathrm{p}(d\boldsymbol{\omega}) \\ &= K(0,0)\, \mathbb{E}_{\boldsymbol{\omega}} \left[\cos\left(\boldsymbol{\omega}^\intercal (x - x')\right)\right], \end{aligned}$$

where $\mathrm{p}(\cdot)$ is a probability measure on $\mathbb{R}^d$

Random Features

- Rahimi and Recht (2007): random Fourier features (RFF)
- Bochner's theorem: if K is stationary (e.g., Gaussian and Matérn), then

$$\begin{aligned} K(x, x') &= K(\mathbf{0}, \mathbf{0}) \int_{\mathbb{R}^d} e^{\mathrm{i}\boldsymbol{\omega}^\intercal (x - x')} \mathrm{p}(d\boldsymbol{\omega}) \\ &= K(\mathbf{0}, \mathbf{0}) \int_{\mathbb{R}^d} \cos\left(\boldsymbol{\omega}^\intercal (x - x')\right) \mathrm{p}(d\boldsymbol{\omega}) \\ &= K(\mathbf{0}, \mathbf{0}) \, \mathbb{E}_{\boldsymbol{\omega}} \left[\cos\left(\boldsymbol{\omega}^\intercal (x - x')\right)\right], \end{aligned}$$

  where $\mathrm{p}(\cdot)$ is a probability measure on $\mathbb{R}^d$

  - If K is Gaussian, then $\mathrm{p}(\cdot)$ is multivariate normal
  - If K is Matérn, then $\mathrm{p}(\cdot)$ is multivariate Student's T

Monte Carlo Approximation

- Straightforward calculation shows: if $b \sim \mathrm{Unif}[0, 2\pi]$, then

$$\mathbb{E}_{\boldsymbol{\omega}}\left[\cos\left(\boldsymbol{\omega}^\intercal(\mathbf{x} - \mathbf{x}')\right)\right] = \mathbb{E}_{\boldsymbol{\omega}, b}\left[\sqrt{2}\cos\left(\boldsymbol{\omega}^\intercal \mathbf{x} + b\right)\sqrt{2}\cos\left(\boldsymbol{\omega}^\intercal \mathbf{x}' + b\right)\right]$$

- Draw $\boldsymbol{\omega}_t$ from $\mathrm{p}(\cdot)$ and b_t from $\mathrm{Unif}[0, 2\pi]$

$$K(\mathbf{x}, \mathbf{x}') \approx K(\mathbf{0}, \mathbf{0}) \cdot \frac{1}{m}\sum_{t=1}^{m} \sqrt{2}\cos\left(\boldsymbol{\omega}_t^\intercal \mathbf{x} + b_t\right)\sqrt{2}\cos\left(\boldsymbol{\omega}_t^\intercal \mathbf{x}' + b_t\right)$$

$$:= \sum_{t=1}^{m} \phi_t(\mathbf{x})\phi_t(\mathbf{x}') := \tilde{K}(\mathbf{x}, \mathbf{x}'),$$

Monte Carlo Approximation

- Straightforward calculation shows: if $b \sim \mathrm{Unif}[0, 2\pi]$, then

$$\mathbb{E}_{\boldsymbol{\omega}}\left[\cos\left(\boldsymbol{\omega}^\intercal(\mathbf{x}-\mathbf{x}')\right)\right] = \mathbb{E}_{\boldsymbol{\omega},b}\left[\sqrt{2}\cos\left(\boldsymbol{\omega}^\intercal\mathbf{x}+b\right)\sqrt{2}\cos\left(\boldsymbol{\omega}^\intercal\mathbf{x}'+b\right)\right]$$

- Draw $\boldsymbol{\omega}_t$ from p(·) and b_t from $\mathrm{Unif}[0, 2\pi]$

$$\begin{aligned} K(\mathbf{x},\mathbf{x}') &\approx K(\mathbf{0},\mathbf{0})\cdot\frac{1}{m}\sum_{t=1}^{m}\sqrt{2}\cos\left(\boldsymbol{\omega}_t^\intercal\mathbf{x}+b_t\right)\sqrt{2}\cos\left(\boldsymbol{\omega}_t^\intercal\mathbf{x}'+b_t\right) \\ &:= \sum_{t=1}^{m}\phi_t(\mathbf{x})\phi_t(\mathbf{x}') := \tilde{K}(\mathbf{x},\mathbf{x}'), \end{aligned}$$

- $\boldsymbol{\phi}_m(\mathbf{x}) = (\phi_1(\mathbf{x}),\ldots,\phi_m(\mathbf{x}))^\intercal$ is a vector of basis functions (i.e., features)
- The covariance matrix associated with evaluating $\tilde{K}$ at $\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ is

$$\tilde{K} = \begin{pmatrix} \boldsymbol{\phi}_m(\mathbf{x}_1)^\intercal \\ \vdots \\ \boldsymbol{\phi}_m(\mathbf{x}_n)^\intercal \end{pmatrix}\begin{pmatrix} \boldsymbol{\phi}_m(\mathbf{x}_1) & \cdots & \boldsymbol{\phi}_m(\mathbf{x}_n) \end{pmatrix} := \boldsymbol{\Phi}_m\boldsymbol{\Phi}_m^\intercal.$$

RFF v.s. Nyström

	RFF	Nyström
Data-dependent?	No	Yes
Kernel-dependent?	Yes	No

Summary

- Low-order polynomials, linear basis function models, and GPs
- Enhancement via stylized models and/or gradient observations

- Locally convergent: RSM, STRONG
- Globally convergent: KG, GP-UCB, GPS

- Scalable GP computations: the Nyström method, random features

Future Studies

- Integrate local search and global search
- Leverage structural info (convexity/smoothness) to accelerate convergence
- Deeper theoretical understanding of SO algorithms
- Parallel computing

References i

Sigrúnr Andradóttir. A review of random search methods. In Michael C. Fu, editor, *Handbook of Simulation Optimization*, pages 277–292. Springer, New York, 2015.

Russell R Barton and Martin Meckesheimer. Metamodel-based simulation optimization. In S.G. Henderson and B.L. Nelson, editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, chapter 18, pages 535–574. Elsevier, 2006.

Kuo-Hao Chang, L. Jeff Hong, and Hong Wan. Stochastic trust-region response-surface method (STRONG)—A new response-surface framework for simulation optimization. *INFORMS Journal on Computing*, 25(2):230–243, 2013.

Marie Chau and Michael C. Fu. An overview of stochastic approximation. In Michael C. Fu, editor, *Handbook of Simulation Optimization*, pages 149–178. Springer, New York, 2015.

Xi Chen, Bruce Ankenman, and Barry L. Nelson. Enhancing stochastic kriging metamodels with gradient estimators. *Operations Research*, 61(2):512–528, 2013.

References ii

Peter I. Frazier. Bayesian optimization. In Esma Gel and Lewis Ntaimo, editors, *Recent Advances in Optimization and Modeling of Contemporary Problems*, INFORMS TutORials in Operations Research, pages 255–278. INFORMS, 2018.

Michael C. Fu and Shane G. Henderson. History of seeking better solutions, AKA simulation optimization. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, , and E. Page, editors, *Proc. 1997 Winter Simulation Conf.*, pages 131–157. IEEE, 2017.

Michael C. Fu and Huashuai Qu. Regression models augmented with direct stochastic gradient estimators. *INFORMS Journal on Computing*, 26(3):484–499, 2014.

L. Jeff Hong, Barry L. Nelson, and Jie Xu. Discrete optimization via simulation. In Michael C. Fu, editor, *Handbook of Simulation Optimization*, pages 9–44. Springer, New York, 2015.

L. Jeff Hong, Weiwei Fan, and Jun Luo. Review on ranking and selection: A new perspective. *Frontiers of Engineering Management*, 8(3):321–343, 2021.

References iii

Haojun Huo, Xiaowei Zhang, and Zeyu Zheng. A scalable approach to enhancing stochastic kriging with gradients. In M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, editors, *Proceedings of the 2018 Winter Simulation Conference*, pages 2213–2224, Piscataway, NJ, 2018. IEEE.

Sujin Kim, Raghu Pasupathy, and Shane G. Henderson. A guide to sample average approximation. In Michael C. Fu, editor, *Handbook of Simulation Optimization*, pages 207–243. Springer, New York, 2015.

Pierre L'Ecuyer. A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science*, 36(11):1364–1383, 1990.

Ziwei Lin, Andrea Matta, and J. George Shanthikumar. Combining simulation experiments and analytical models with area-based accuracy for performance evaluation of manufacturing systems. *IISE Transactions*, 51(3):266–283, 2019.

Xuefei Lu, Alessandro Rudi, Emanuele Borgonovo, and Lorenzo Rosasco. Faster kriging: Facing high-dimensional simulators. *Operations Research*, 68(1):233–249, 2020.

References iv

Huashuai Qu and Michael C. Fu. Gradient extrapolated stochastic kriging. *ACM Transactions on Modeling and Computer Simulation*, 24(4):Article 23, 2014.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 1177–1184, Red Hook, NY, 2007. Curran Associates, Inc.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 1657–1665, Red Hook, NY, 2015. Curran Associates, Inc.

Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments*. Springer, New York, 2003.

Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression. *SIAM Journal on Optimization*, 21(3):996–1026, 2011.

References v

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

H. Shen, L. Jeff Hong, and X. Zhang. Stochastic kriging for queueing simulation with stylized models. *IISE Transactions*, 50(11):943–958, 2018.

Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for mahine learning. In Pat Langley, editor, *Proceedings of the 17th International Conference on Machine Learning*, pages 911–918, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Lihua Sun, L. Jeff Hong, and Zhaolin Hu. Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Operations Research*, 62(6):1416–1438, 2014.

References vi

Wenjie Sun, Zhaolin Hu, and L. Jeff Hong. Gaussian mixture model-based random search for continuous optimization via simulation. In M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, editors, *Proceedings of the 2018 Winter Simulation Conference*, pages 2003–2014, Piscataway, NJ, 2018. IEEE.

Jian Wu and Peter I. Frazier. The parallel knowledge gradient method for batch Bayesian optimization. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 3126–3134, Red Hook, NY, 2016. Curran Associates, Inc.