Sequential Detection of Convexity from Noisy Function Evaluations

Nanjing Jian† Shane G. Henderson† Susan R. Hunter‡

† School of Operations Research and Information Engineering, Cornell University

‡ School of Industrial Engineering, Purdue University

2014 Winter Simulation Conference, Savannah, GA

December 9, 2014

Supported in Part by NSF CMMI-1200315

Problem Statement

- Consider a function $g : S \subseteq \mathbb{R}^d \to \mathbb{R}$ that can only be evaluated with the presence of noise at r points $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r) \in \mathbb{R}^d$. Let the true values of the function g at $\boldsymbol{x}$ be denoted $\boldsymbol{g} = (g(\boldsymbol{x}_1), g(\boldsymbol{x}_2), \ldots, g(\boldsymbol{x}_r))^T$.
- We wish to determine the convexity/non-convexity of $\boldsymbol{g}$ with some probabilistic guarantee, using only estimates of its values obtained through simulation at the points $\boldsymbol{x}$.

♠ $\boldsymbol{g}$ is convex if a convex function exists that coincides with $\boldsymbol{g}$ at those points.

Motivation

- Learning about black-box functions

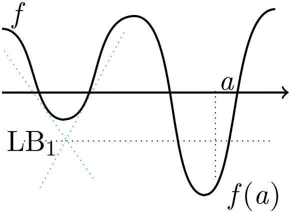
"Sorry, it's curiosity"

Motivation

- Learning about black-box functions

"Sorry, it's curiosity"

- Stopping rule for global (stochastic) optimization algorithms



Motivation

Previous research: One-shot frequentist hypothesis test, with the number of samples predetermined.

Dim	Distance	Regression parameters
1	Juditsky and Nemirovski [2002]	Baraud et al. [2005] Diack and Thomas-Agnan [1998] Meyer [2012] Wang and Meyer [2011]
> 1	Silvapulle and Sen [2001]	Lau [1978] Abrevaya and Jiang [2005]

Motivation

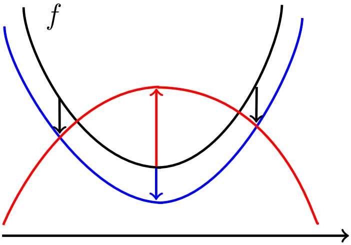
Previous research: One-shot frequentist hypothesis test, with the number of samples predetermined.

Dim	Distance	Regression parameters
1	Juditsky and Nemirovski [2002]	Baraud et al. [2005] Diack and Thomas-Agnan [1998] Meyer [2012] Wang and Meyer [2011]
> 1	Silvapulle and Sen [2001]	Lau [1978] Abrevaya and Jiang [2005]

Our Goal: A sequential algorithm in the Bayesian setting with indefinite number of samples and can be stopped at any time.

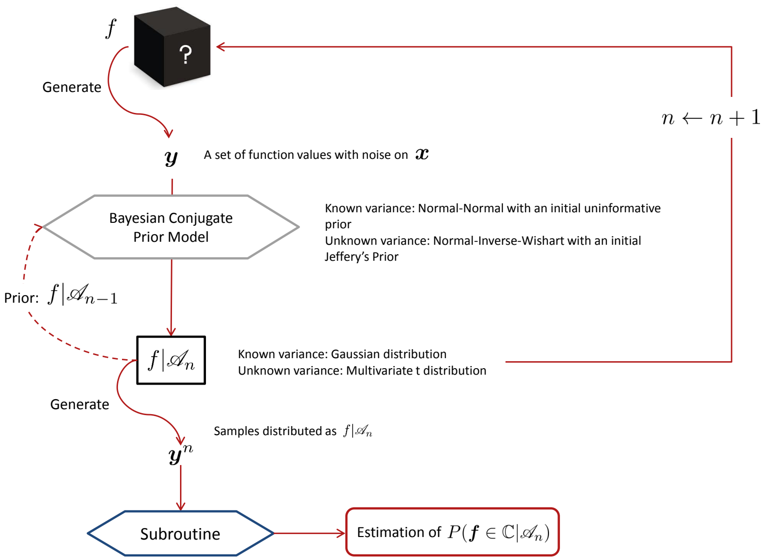
Assumptions

1. We obtain realizations of a random vector $\boldsymbol{Y} = \boldsymbol{f} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim N(\mathbf{0}, \Gamma) \in \mathbb{R}^r$, and Γ positive-definite if known.
2. Conditional on $\boldsymbol{f}$, the samples $(\boldsymbol{y}_n : n = 1, 2, \ldots)$ in each iteration consists of i.i.d. random vectors.



Note that Γ is not necessarily diagonal because using Common Random Numbers can maintain the function structural properties (e.g. Chen et al. [2012]).

Main Idea


f
?
Generate
$n \leftarrow n+1$
$\boldsymbol{y}$
A set of function values with noise on $\boldsymbol{x}$
Bayesian Conjugate Prior Model
Known variance: Normal-Normal with an initial uninformative prior
Unknown variance: Normal-Inverse-Wishart with an initial Jeffery's Prior
Prior: $f|\mathscr{A}_{n-1}$
$f|\mathscr{A}_n$
Known variance: Gaussian distribution
Unknown variance: Multivariate t distribution
Generate
Samples distributed as $f|\mathscr{A}_n$
$\boldsymbol{y}^n$
Subroutine
Estimation of $P(\boldsymbol{f} \in \mathbb{C}|\mathscr{A}_n)$

Convergence

Theorem

Let $p_n = P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n)$ be the n-iteration posterior probability that $\boldsymbol{f}$ is convex. As $n \to \infty$, $p_n - \mathbb{1}\{\boldsymbol{f} \in \mathbb{C}\} \to 0$ a.s.

Convergence

Theorem

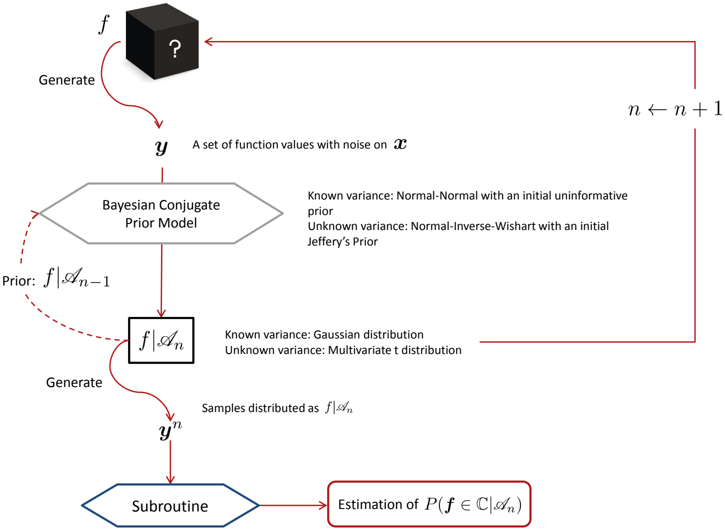
Let $p_n = P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n)$ be the n-iteration posterior probability that $\boldsymbol{f}$ is convex. As $n \to \infty$, $p_n - \mathbb{1}\{\boldsymbol{f} \in \mathbb{C}\} \to 0$ a.s.

Proof sketch (assume Γ known):

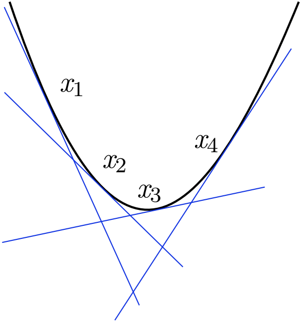
1. $P(\boldsymbol{f} \in \partial\mathbb{C}) = 0$ leaves cases $\boldsymbol{f} \notin \mathbb{C}$ and $\boldsymbol{f} \in \mathbb{C}^\circ$.
2. $\mu_n - \boldsymbol{f} \to 0$ in probability and $\Lambda_n \sim \Gamma/n \to \mathbf{0}$ as $n \to \infty$.
3. When $\boldsymbol{f} \notin \mathbb{C}$, define $D_{\boldsymbol{f}} = \min_{\boldsymbol{h} \in \mathbb{C}} ||\boldsymbol{f} - \boldsymbol{h}||$, then $p_n - \mathbb{1}\{\boldsymbol{f} \in \mathbb{C}\} = P(\mu_n + \Lambda_n^{1/2} Z \in \mathbb{C} | \mathscr{A}_n) \leq P(\|\mu_n + \Lambda_n^{1/2} Z - \boldsymbol{f}\| \geq D_{\boldsymbol{f}} | \mathscr{A}_n) \leq P(\|\mu_n - \boldsymbol{f}\| \geq D_{\boldsymbol{f}}/2 | \mathscr{A}_n) + P(\|\Lambda_n^{1/2} Z\| \geq D_{\boldsymbol{f}}/2 | \mathscr{A}_n) \to 0$ in probability by Markov's Inequality. A lower bound of 0 can be given when $\boldsymbol{f} \in \mathbb{C}^\circ$ similarly.
4. $(p_n : n \geq 0)$ is a uniformly integrable martingale, so the convergence is almost surely.

For the unknown Γ case, a similar proof can be constructed by conditioning on Γ.

Main Idea


f
?
Generate
n ← n + 1
y
A set of function values with noise on x
Bayesian Conjugate Prior Model
Known variance: Normal-Normal with an initial uninformative prior
Unknown variance: Normal-Inverse-Wishart with an initial Jeffery's Prior
Prior: f|𝒜n−1
f|𝒜n
Known variance: Gaussian distribution
Unknown variance: Multivariate t distribution
Generate
Samples distributed as f|𝒜n
y^n
Subroutine
Estimation of P(f ∈ ℂ|𝒜n)

Convexity



$\boldsymbol{g} \in \mathbb{C}$ if and only if each of the following LS(i),$i \in \{1, \ldots, r\}$

$$\boldsymbol{a}_i^T \boldsymbol{x}_i + b_i = g(\boldsymbol{x}_i)$$
$$\boldsymbol{a}_i^T \boldsymbol{x}_j + b_i \leq g(\boldsymbol{x}_j),\ \forall j \in \{1, \ldots, r\} \backslash \{i\}.$$

is feasible in the variables $\boldsymbol{a}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ (Murty [1988]).

Vanilla Monte Carlo Method

In each iteration of the main algorithm, after updating the hyper-parameters of the posterior distribution,

1. Simulate m i.i.d. samples $\boldsymbol{y}_k^n$ from the predictive distribution $\boldsymbol{f}|\mathscr{A}_n$.

Vanilla Monte Carlo Method

In each iteration of the main algorithm, after updating the hyper-parameters of the posterior distribution,

1. Simulate m i.i.d. samples $\boldsymbol{y}_k^n$ from the predictive distribution $\boldsymbol{f}|\mathscr{A}_n$.
2. For each sample $\boldsymbol{y}_k^n$, set $\boldsymbol{g} = \boldsymbol{y}_k^n$ in $\mathrm{LS}(i), i = 1, \ldots, r$. Obtain an indicator $\mathbb{1}\{\boldsymbol{y}_k^n \in \mathbb{C}\}$ that is 0 if any of $\mathrm{LS}(i), i = 1, \ldots, r$ is infeasible and 1 otherwise.

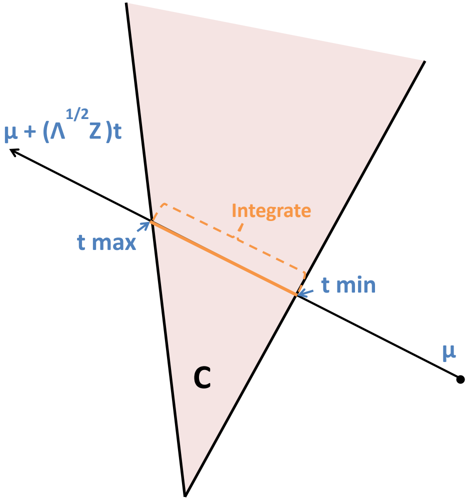
Vanilla Monte Carlo Method

In each iteration of the main algorithm, after updating the hyper-parameters of the posterior distribution,

1. Simulate m i.i.d. samples $\boldsymbol{y}_k^n$ from the predictive distribution $\boldsymbol{f}|\mathscr{A}_n$.
2. For each sample $\boldsymbol{y}_k^n$, set $\boldsymbol{g} = \boldsymbol{y}_k^n$ in LS(i), $i = 1, \dots, r$. Obtain an indicator $\mathbb{1}\{\boldsymbol{y}_k^n \in \mathbb{C}\}$ that is 0 if any of LS(i), $i = 1, \dots, r$ is infeasible and 1 otherwise.
3. Output the estimator $\hat{p_n} = \frac{1}{m}\sum_{k=1}^{m} \mathbb{1}\{\boldsymbol{y}_k^n \in \mathbb{C}\}$ as the average of all indicators.

Conditional Monte Carlo Method

Instead of obtaining a $0-1$ estimator...



Conditional Monte Carlo Method

Why would this work?

$$P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n)$$

Conditional Monte Carlo Method

Why would this work?

$$
\begin{aligned}
&P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n) \\
=&E_n \left(\mathbb{1} \left\{ \Lambda_n^{1/2} X + \mu_n \in \mathbb{C} \right\} \right), \text{ for } X \sim N(0, I) \text{ or } t_{\nu_n}(0, I)
\end{aligned}
$$

Conditional Monte Carlo Method

Why would this work?

$$
\begin{aligned}
&P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n) \\
=&E_n\left(\mathbb{1}\left\{\Lambda_n^{1/2} X + \mu_n \in \mathbb{C}\right\}\right), \text{ for } X \sim N(0, I) \text{ or } t_{\nu_n}(0, I) \\
=&E_n\left(\mathbb{1}\left\{T\Lambda_n^{1/2} Z + \mu_n \in \mathbb{C}\right\}\right), \text{ for } Z \text{ uniform on } S^{r-1}
\end{aligned}
$$

Conditional Monte Carlo Method

Why would this work?

$$
\begin{aligned}
&P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n) \\
=&E_n\left(\mathbb{1}\left\{\Lambda_n^{1/2}X + \mu_n \in \mathbb{C}\right\}\right), \text{ for } X \sim N(0, I) \text{ or } t_{\nu_n}(0, I) \\
=&E_n\left(\mathbb{1}\left\{T\Lambda_n^{1/2}Z + \mu_n \in \mathbb{C}\right\}\right), \text{ for } Z \text{ uniform on } S^{r-1} \\
=&E_n\left(E\left(\mathbb{1}\left\{T\Lambda_n^{1/2}Z + \mu_n \in \mathbb{C}\right\} | Z\right)\right)
\end{aligned}
$$

Conditional Monte Carlo Method

Why would this work?

$$
\begin{aligned}
&P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n) \\
=&E_n\left(\mathbb{1}\left\{\Lambda_n^{1/2} X + \mu_n \in \mathbb{C}\right\}\right), \text{ for } X \sim N(0, I) \text{ or } t_{\nu_n}(0, I) \\
=&E_n\left(\mathbb{1}\left\{T\Lambda_n^{1/2} Z + \mu_n \in \mathbb{C}\right\}\right), \text{ for } Z \text{ uniform on } S^{r-1} \\
=&E_n\left(E\left(\mathbb{1}\left\{T\Lambda_n^{1/2} Z + \mu_n \in \mathbb{C}\right\} | Z\right)\right) \\
=&E_n(P(T \in [t_{\min}(Z), t_{\max}(Z)] | Z))
\end{aligned}
$$

Conditional Monte Carlo Method

Why would this work?

$$
\begin{aligned}
&P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n) \\
=&E_n\left(\mathbb{1}\left\{\Lambda_n^{1/2} X + \mu_n \in \mathbb{C}\right\}\right), \text{ for } X \sim N(0, I) \text{ or } t_{\nu_n}(0, I) \\
=&E_n\left(\mathbb{1}\left\{T\Lambda_n^{1/2} Z + \mu_n \in \mathbb{C}\right\}\right), \text{ for } Z \text{ uniform on } S^{r-1} \\
=&E_n\left(E\left(\mathbb{1}\left\{T\Lambda_n^{1/2} Z + \mu_n \in \mathbb{C}\right\} | Z\right)\right) \\
=&E_n(P(T \in [t_{\min}(Z), t_{\max}(Z)] | Z)) \\
=&E_n(F_{T|Z}(t_{\max}(Z)) - F_{T|Z}(t_{\min}(Z)))
\end{aligned}
$$

Conditional Monte Carlo Method

Why would this work?

$$
\begin{aligned}
& P(\boldsymbol{f} \in \mathbb{C} | \mathscr{A}_n) \\
=& E_n \left(\mathbb{1}\left\{ \Lambda_n^{1/2} X + \mu_n \in \mathbb{C} \right\} \right), \text{ for } X \sim N(0, I) \text{ or } t_{\nu_n}(0, I) \\
=& E_n \left(\mathbb{1}\left\{ T \Lambda_n^{1/2} Z + \mu_n \in \mathbb{C} \right\} \right), \text{ for } Z \text{ uniform on } S^{r-1} \\
=& E_n \left(E \left(\mathbb{1}\left\{ T \Lambda_n^{1/2} Z + \mu_n \in \mathbb{C} \right\} | Z \right) \right) \\
=& E_n(P(T \in [t_{\min}(Z), t_{\max}(Z)] | Z)) \\
=& E_n(F_{T|Z}(t_{\max}(Z)) - F_{T|Z}(t_{\min}(Z)))
\end{aligned}
$$

Known variance: $F_{T|Z}(t) = \frac{1}{2} + \text{sign}(t) F_{\chi^2_r}(t^2)$.
Unknown variance: $F_{T|Z}(t) = \frac{1}{2} + \text{sign}(t) F_{F(r,\nu_n)}(t^2/r)$.

Conditional Monte Carlo Method

At each iteration n,

1. Simulate m i.i.d. samples z_k uniformly on the unit shell S^{r-1}.

Conditional Monte Carlo Method

At each iteration n,

1. Simulate m i.i.d. samples z_k uniformly on the unit shell S^{r-1}.
2. For each sample z_k, set $\boldsymbol{g} = \mu_n + (\Lambda_n^{1/2} z_k) t_i$ in LS(i), $i = 1, \ldots, r$, and use LS(i) as constraints to solve linear programs with optimal objective values $t_{\min}(i) = \min\ t_i$ and $t_{\max}(i) = \max\ t_i$.

Conditional Monte Carlo Method

At each iteration n,

1. Simulate m i.i.d. samples z_k uniformly on the unit shell S^{r-1}.
2. For each sample z_k, set $\boldsymbol{g} = \mu_n + (\Lambda_n^{1/2} z_k) t_i$ in LS(i), $i = 1, \ldots, r$, and use LS(i) as constraints to solve linear programs with optimal objective values $t_{\min}(i) = \min\ t_i$ and $t_{\max}(i) = \max\ t_i$.
3. Set $t_{\min}(z_k) = \max_{i=1,2,\ldots,r} t_{\min}(i)$, $t_{\max}(z_k) = \min_{i=1,2,\ldots,r} t_{\max}(i)$.

Conditional Monte Carlo Method

At each iteration n,

1. Simulate m i.i.d. samples z_k uniformly on the unit shell S^{r-1}.
2. For each sample z_k, set $\boldsymbol{g} = \mu_n + (\Lambda_n^{1/2} z_k) t_i$ in LS(i), $i = 1, \ldots, r$, and use LS(i) as constraints to solve linear programs with optimal objective values $t_{\min}(i) = \min\ t_i$ and $t_{\max}(i) = \max\ t_i$.
3. Set $t_{\min}(z_k) = \max_{i=1,2,\ldots,r} t_{\min}(i)$, $t_{\max}(z_k) = \min_{i=1,2,\ldots,r} t_{\max}(i)$.
4. Calculate $F_{T|Z}(t_{\max}(z_k)) - F_{T|Z}(t_{\min}(z_k))$.

Conditional Monte Carlo Method

At each iteration n,

1. Simulate m i.i.d. samples z_k uniformly on the unit shell S^{r-1}.
2. For each sample z_k, set $\boldsymbol{g} = \mu_n + (\Lambda_n^{1/2} z_k) t_i$ in $\text{LS}(i), i = 1, \dots, r$, and use $\text{LS}(i)$ as constraints to solve linear programs with optimal objective values $t_{\min}(i) = \min\ t_i$ and $t_{\max}(i) = \max\ t_i$.
3. Set $t_{\min}(z_k) = \max\limits_{i=1,2,\dots,r} t_{\min}(i)$, $t_{\max}(z_k) = \min\limits_{i=1,2,\dots,r} t_{\max}(i)$.
4. Calculate $F_{T|Z}(t_{\max}(z_k)) - F_{T|Z}(t_{\min}(z_k))$.
5. Output the estimator $\hat{p_n}$ as a sample average of such integrations obtained from m samples.

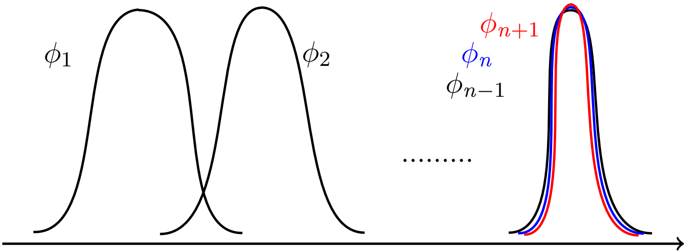
Conditional Monte Carlo Method

At each iteration n,

1. Simulate m i.i.d. samples z_k uniformly on the unit shell S^{r-1}.
2. For each sample z_k, set $\boldsymbol{g} = \mu_n + (\Lambda_n^{1/2} z_k) t_i$ in LS(i), $i = 1, \ldots, r$, and use LS(i) as constraints to solve linear programs with optimal objective values $t_{\min}(i) = \min\, t_i$ and $t_{\max}(i) = \max\, t_i$.
3. Set $t_{\min}(z_k) = \max_{i=1,2,\ldots,r} t_{\min}(i)$, $t_{\max}(z_k) = \min_{i=1,2,\ldots,r} t_{\max}(i)$.
4. Calculate $F_{T|Z}(t_{\max}(z_k)) - F_{T|Z}(t_{\min}(z_k))$.
5. Output the estimator $\hat{p_n}$ as a sample average of such integrations obtained from m samples.

This method achieves variance reduction compared to the vanilla Monte Carlo estimator.

Reusing Samples



As n grows large, we would expect $\frac{\phi_{n+1}}{\phi_n}$ to become close to 1, where ϕ_n is the density of $\boldsymbol{f}|\mathscr{A}_n$. Thus p_{n+1} may be estimated by $\boldsymbol{y}_k^n, k = 1, \ldots, m$, for which $\mathbb{1}\left\{\boldsymbol{y}_k^n \in \mathbb{C}\right\}$ has been calculated in iteration n.

Reusing Samples

At iteration n, calculate $\hat{p_n}$ using the vanilla Monte Carlo method with samples $\boldsymbol{y}_k^n, k = 1, \ldots, m$. In iteration $n + \ell$,

Reusing Samples

At iteration n, calculate $\hat{p_n}$ using the vanilla Monte Carlo method with samples $\boldsymbol{y}_k^n, k = 1, \ldots, m$. In iteration $n + \ell$,

Change of Measure: Reuse all the samples $\boldsymbol{y}_k^n$ in the n-th iteration. Output $\hat{p_{n+\ell}} = \frac{1}{m} \sum_{k=1}^{m} \mathbb{1} \left\{ \boldsymbol{y}_k^n \in \mathbb{C} \right\} \frac{\phi_{n+\ell}(\boldsymbol{y}_k^n)}{\phi_n(\boldsymbol{y}_k^n)}$.

Reusing Samples

At iteration n, calculate $\hat{p_n}$ using the vanilla Monte Carlo method with samples $\boldsymbol{y}_k^n, k = 1, \ldots, m$. In iteration $n+\ell$,

Change of Measure: Reuse all the samples $\boldsymbol{y}_k^n$ in the n-th iteration. Output $\hat{p_{n+\ell}} = \frac{1}{m}\sum_{k=1}^{m} \mathbb{1}\left\{\boldsymbol{y}_k^n \in \mathbb{C}\right\} \frac{\phi_{n+\ell}(\boldsymbol{y}_k^n)}{\phi_n(\boldsymbol{y}_k^n)}$.

Generalized Acceptance/Rejection: Reuse part of the samples by accepting $\boldsymbol{y}_k^n$ with probability $\frac{\phi_{n+\ell}(\boldsymbol{y}_k^n)}{c\phi_n(\boldsymbol{y}_k^n)}$, then generate new samples as needed:

$$\hat{p_{n+\ell}} = \frac{1}{m}\left(\sum_{k\in S} \mathbb{1}\left\{\boldsymbol{y}_k^n \in \mathbb{C}\right\} + \sum_{k=1}^{m-|S|} \mathbb{1}\left\{\boldsymbol{y}_k^{n+\ell} \in \mathbb{C}\right\}\right)$$

Reusing Samples

At iteration n, calculate $\hat{p_n}$ using the vanilla Monte Carlo method with samples $\boldsymbol{y}_k^n, k = 1, \ldots, m$. In iteration $n + \ell$,

Change of Measure: Reuse all the samples $\boldsymbol{y}_k^n$ in the n-th iteration. Output $\hat{p_{n+\ell}} = \frac{1}{m} \sum_{k=1}^{m} \mathbb{1}\left\{\boldsymbol{y}_k^n \in \mathbb{C}\right\} \frac{\phi_{n+\ell}(\boldsymbol{y}_k^n)}{\phi_n(\boldsymbol{y}_k^n)}$.

Generalized Acceptance/Rejection: Reuse part of the samples by accepting $\boldsymbol{y}_k^n$ with probability $\frac{\phi_{n+\ell}(\boldsymbol{y}_k^n)}{c\phi_n(\boldsymbol{y}_k^n)}$, then generate new samples as needed:
$\hat{p_{n+\ell}} = \frac{1}{m}\left(\sum_{k\in S} \mathbb{1}\left\{\boldsymbol{y}_k^n \in \mathbb{C}\right\} + \sum_{k=1}^{m-|S|} \mathbb{1}\left\{\boldsymbol{y}_k^{n+\ell} \in \mathbb{C}\right\}\right)$

By utilizing the samples and results in an earlier iteration, this method saves computational time.

Inverted Bowl

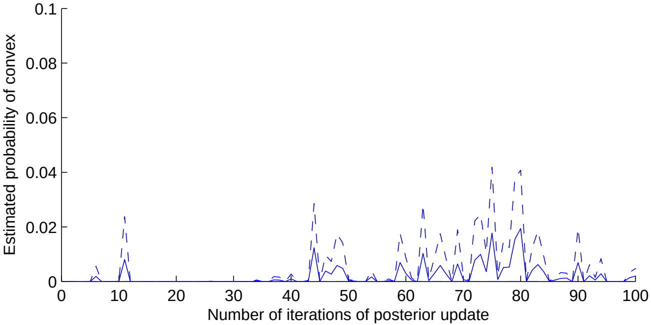


Figure : $\hat{p_n}$ for $g = -\|\boldsymbol{x}\|^2, \boldsymbol{x} \in [-1, 1]^{30}$, $r = 61$, Γ has 10^4 on the diagonal.

Plane

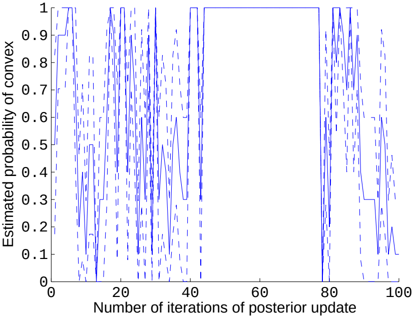


Figure : $\hat{p_n}$ for $g = 0, \boldsymbol{x} \in [-1, 1]^2$, $r = 5$, Γ has 1 on the diagonal.

Where to sample?

An interesting example for $g = \|\boldsymbol{x}\|^2$, $\boldsymbol{x} \in [-1, 1]^{30}$, $r = 60$, and Γ has 10^4 on the diagonal.

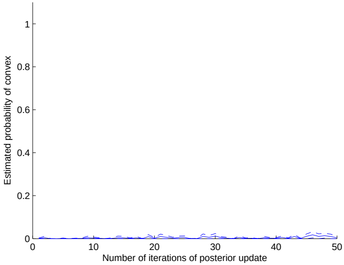


Figure : Sampling the 60 points uniformly at random in space.

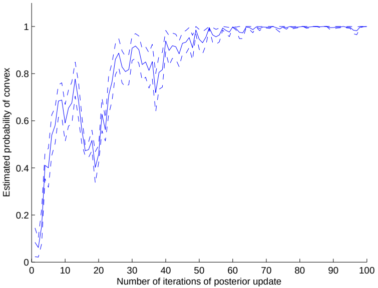


Figure : Sampling along 20 random lines with 3 points on each.

What happened?

For easiness of illustration, consider a 2-dimensional function with the following level sets:

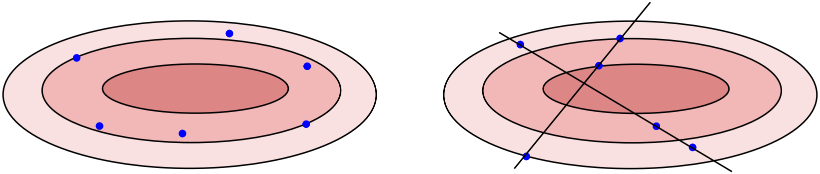


Figure : Sampling uniformly vs. Sampling along random lines.

Ambulances in a Square

A "real" example from SimOpt.org: What does the long run average response time behave like as a function of the ambulance base location?

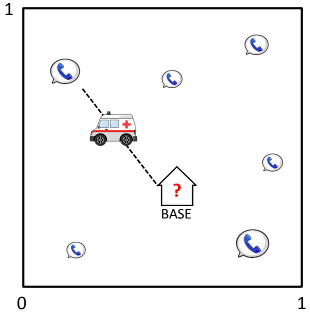


Figure : Problem illustration.

One Base, Two Bases

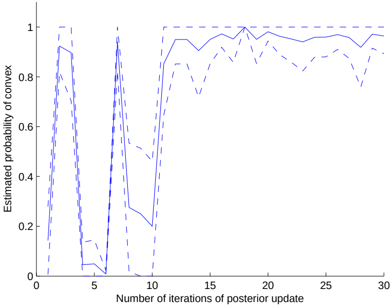


Figure : $\hat{p_n}$ for one ambulance base ($d = 2$).

One Base, Two Bases

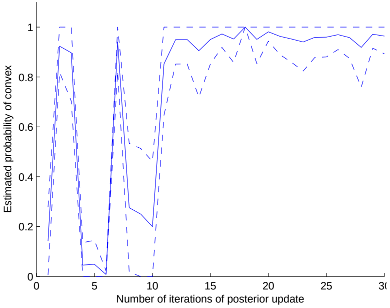


Figure : $\hat{p_n}$ for one ambulance base ($d = 2$).

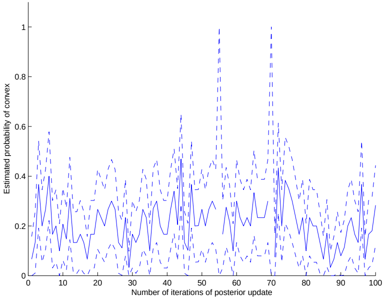


Figure : $\hat{p_n}$ for two ambulance bases ($d = 4$).

Conclusion

We suggested

- a sequential method for detecting convexity/non-convexity of noisy functions
- a Monte Carlo method for estimating probability of convex
- three alternatives for efficiency improvement

Next steps:

- the number and locations of sampled points
- uneven sample size at each sampled point

Reference I

Jason Abrevaya and Wei Jiang. A nonparametric approach to measuring and testing curvature. *Journal of Business & Economic Statistics*, 23:1–19, 2005. URL http://EconPapers.repec.org/RePEc:bes:jnlbes:v:23:y:2005:p:1-19.

Y. Baraud, S. Huet, and B. Laurent. Testing convex hypotheses on the mean of a Gaussian vector. application to testing qualitative hypotheses on a regression function. *The Annals of Statistics*, 33(1):214–257, 2005.

Xi Chen, Bruce E. Ankenman, and Barry L. Nelson. The effects of common random numbers on stochastic kriging metamodels. *ACM Trans. Model. Comput. Simul.*, 22(2):7:1–7:20, March 2012. ISSN 1049-3301. doi: 10.1145/2133390.2133391. URL http://doi.acm.org/10.1145/2133390.2133391.

C. A. T. Diack and C. Thomas-Agnan. A nonparametric test of the non-convexity of regression. *Nonparametric Statistics*, 9:335–362, 1998.

A. Juditsky and A. Nemirovski. On nonparametric tests of positivity/monotonicity/convexity. *The Annals of Statistics*, 30(2):498–527, 2002.

Lawrence J Lau. Testing and imposing monoticity, convexity, and quasi-convexity constraints. *Electronic Journal of Statistics*, 1:409–453, 1978.

Mary C. Meyer. Constrained penalized splines. *Canadian Journal of Statistics*, 40(1):190–206, 2012. ISSN 1708-945X. doi: 10.1002/cjs.10137. URL http://dx.doi.org/10.1002/cjs.10137.

K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin, 1988.

Huashuai Qu, Ilya O. Ryzhov, and Michael C. Fu. Ranking and selection with unknown correlation structures. In *Proceedings of the Winter Simulation Conference*, WSC '12, pages 12:1–12:12. Winter Simulation Conference, 2012. URL http://dl.acm.org/citation.cfm?id=2429759.2429774.

Reference II

Mervyn J. Silvapulle and Pranab K. Sen. *Constrained Statistical Inference: Order, Inequality, and Shape Restrictions*, chapter 3, pages 59–141. John Wiley & Sons, Inc., 2001. ISBN 9781118165614. doi: 10.1002/9781118165614.ch3. URL http://dx.doi.org/10.1002/9781118165614.ch3.

Jianqiang C. Wang and Mary C. Meyer. Testing the monotonicity or convexity of a function using regression splines. *Canadian Journal of Statistics*, 39(1):89–107, 2011. ISSN 1708-945X. doi: 10.1002/cjs.10094. URL http://dx.doi.org/10.1002/cjs.10094.